

Министерство образования Республики Башкортостан
ГАОУ СПО Стерлитамакский колледж строительства, экономики и права

Учебно-методический комплекс

МДК 02.02. Технология разработки и защиты базы данных

ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ 02 Разработка и администрирование баз данных

основной профессиональной образовательной программы (ОПОП)

по специальности СПО

230115 Программирование в компьютерных системах
базовой подготовки

Разработала : **ДОЛГИХ Е.А.**

Одобрено на заседании предметно-цикловой
комиссии специальности 230115
Программирование в компьютерных системах

Протокол № _____

от «__» _____ 2013г.

Председатель ПЦК

_____ /О.А.Комиссарова/

УТВЕРЖДАЮ

Зав. методическим кабинетом

ГАОУ СПО СКСЭиП

_____ /Н.Б. Дубанова/

«__»__ 2013г

Учебно-методический комплекс МДК 02.02. Технология разработки и защиты базы данных
профессионального модуля ПМ 02 Разработка и администрирование баз данных

Составила

Е.А.Долгих

Преподаватель математики и информатики
ГАОУ СПО Стерлитамакский колледж
строительства, экономики и права

Рецензенты

Э.Р. Гиззатова

К. ф.-м. н., доц. кафедры математического
моделирования физико-математического
факультета Стерлитамакского филиала
БашГУ

О.А.Комиссарова

Председатель предметно-цикловой комиссии
специальности 230115 Программирование в
компьютерных системах ГАОУ СПО
Стерлитамакский колледж строительства,
экономики и права

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
РАЗДЕЛ 1. РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	5
1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	5
1.1. Область применения программы.....	5
1.2. Цели и задачи модуля – требования к результатам освоения дисциплины:.....	5
1.3. Рекомендуемое количество часов на освоение программы профессионального модуля:	6
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	6
3. СТРУКТУРА И ПРИМЕРНОЕ СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	7
3.1. Тематический план профессионального модуля	7
3.2. Содержание обучения по профессиональному модулю (ПМ) «Разработка и администрирование баз данных»	8
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	14
4.1. Требования к минимальному материально-техническому обеспечению	14
4.2. Информационное обеспечение обучения	14
4.3. Общие требования к организации образовательного процесса	15
4.4. Кадровое обеспечение образовательного процесса.....	15
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)	15
РАЗДЕЛ 2. КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН МЕЖДИСЦИПЛИНАРНОГО КУРСА	19
РАЗДЕЛ 3. ЛЕКЦИОННЫЙ МАТЕРИАЛ.....	26
РАЗДЕЛ 4. ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОМУ ПРАКТИКУМУ ПО МДК 02.02. Технология разработки и защиты базы данных.....	121
4.1. Общие положения.....	121
4.2. Методические указания к лабораторному практикуму	121
РАЗДЕЛ 5. КОНТРОЛЬ ЗНАНИЙ ПО МДК 02.02. Технология разработки и защиты базы данных	173
5.1. Организация контроля и оценки освоения программы МДК	173
5.2. Текущий контроль	173
5.3. Рубежный контроль	179
5.4. Итоговый контроль.....	181
РАЗДЕЛ 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ	234
6.1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЮ	234
6.1.1. Организация занятий и контроля знаний	234
6.1.2. Организация и контроль самостоятельной работы	235
6.2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	238
6.2.1. Методические рекомендации по работе с литературой.....	238
6.2.2. Методические рекомендации по подготовке к контрольным работам, зачетам, экзаменам	239
6.2.3. Методические рекомендации по работе над рефератом.....	240
ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ.....	243

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Междисциплинарный курс МДК 02.02. Технология разработки и защиты базы данных является частью профессионального модуля ПМ 02 Разработка и администрирование баз данных, входящего в основную профессиональную образовательную программу в соответствии с ФГОС по специальности 230115 Программирование в компьютерных системах, входящей в состав укрупненной группы специальностей СПО 230000 Информатика и вычислительная техника.

Учебно-методический комплекс представляет собой систему нормативной и учебно-методической документации, средств обучения, необходимых и достаточных для качественной организации освоения образовательной программы профессионального модуля, согласно учебного плана.

Целью создания УМК является разработка методического обеспечения раздела МДК, а также предоставление студенту полного комплекта учебно-методических материалов для самостоятельного изучения МДК.

При разработке УМК учитывается, что его компоненты должны:

- Соответствовать общей идеологии федеральной и региональной политики, содействовать развитию региональной системы среднего профессионального образования
- Предусматривать логически последовательное изложение учебного материала
- Предполагать использование современных методов и технических средств интенсификации учебного процесса, позволяющих обучающимся глубоко осваивать учебный материал и получать навыки по его использованию на практике
- Соответствовать современным научным представлениям в данной предметной области
- Обеспечивать межпредметные связи
- Обеспечивать простоту использования для преподавателей и обучающихся

Представленный учебно-методический комплекс был разработан в соответствии с ФГОС СПО и апробирован в учебном процессе. Своевременно был проведен анализ результатов текущего контроля знаний обучающихся, что позволило внести соответствующие коррективы в содержание учебного материала и контрольно-измерительных материалов.

- Учебно-методический комплекс содержит:
- Рабочую программу междисциплинарного курса (МДК) профессионального модуля
- Комплект контрольно-измерительных материалов по МДК
- Календарно-тематический план МДК,
- Учебно-методическое пособие
- Задания и методические указания по выполнению практических и лабораторных работ
- Методические рекомендации по планированию и организации самостоятельной работы обучающихся
- Задания для текущего, промежуточного и итогового контроля знаний обучающихся

Учебно-методический комплекс междисциплинарного курса МДК 02.02. Технология разработки и защиты базы данных профессионального модуля ПМ 02 Разработка и администрирование баз данных содержит материал и методы, формирующие у студентов – программистов профессиональную культуру и специальное техническое мышление, необходимое для успешной работы в современных областях информационных и компьютерных технологий. Комплекс имеет прикладную направленность, что реализуется через рассмотрение современных моделей баз данных, иллюстрирующих теоретическое содержание программы комплекса. Приводится большое количество заданий различной сложности, предназначенных как для текущего, промежуточного и итогового контроля знаний, так и для начальной исследовательской работы по проблематике развития и модернизации современных баз данных.

Учебно-методический комплекс междисциплинарного курса МДК 02.02. Технология разработки и защиты базы данных профессионального модуля ПМ 02 Разработка и администрирование баз данных предназначен для преподавателей специальных дисциплин укрупненной группы специальностей 230000 Информатика и вычислительная техника и в дополнительном профессиональном образовании повышения квалификации и переподготовки кадров в области разработки программного обеспечения профессиональной деятельности.

РАЗДЕЛ 1. РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ 02 Разработка и администрирование баз данных

Программа профессионального модуля разработана на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования **230115 Программирование в компьютерных системах** (укрупненная группа специальностей 230000 Информатика и вычислительная техника)

Организация-разработчик: ГАОУ СПО Стерлитамакский колледж строительства, экономики и права

Разработчики: Долгих Е.А и Маркова Ю.О., преподаватели высшей категории

Утверждена республиканским экспертным советом по профессиональному образованию ГОУ РУНМЦ РБ, секция СПО (протокол №3/11 от 30.06.2011г.)

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ 02 Разработка и администрирование баз данных

1.1. Область применения программы

Программа профессионального модуля (далее рабочая программа) является частью рабочей профессиональной образовательной программы в соответствии с ФГОС по специальности **230115 Программирование в компьютерных системах** в части освоения основного вида профессиональной деятельности:

Разработка и администрирование баз данных

и соответствующих профессиональных компетенций:

1. Разрабатывать объекты базы данных.
2. Реализовывать базу данных в конкретной системе управления базами данных (СУБД).
3. Решать вопросы администрирования базы данных.
4. Реализовывать методы и технологии защиты информации в базах данных.

Программа профессионального модуля может быть использована в профессиональной подготовке специальностей СПО **230100 Информатика и вычислительная техника** и **230111 Компьютерные сети** и в дополнительном профессиональном образовании повышения квалификации и переподготовки кадров в области инфокоммуникационных систем и сетей.

1.2. Цели и задачи модуля – требования к результатам освоения дисциплины:

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

иметь практический опыт:

- работы с объектами базы данных в конкретной системе управления базами данных;
- использования средств заполнения базы данных;
- использования стандартных методов защиты объектов базы данных;

уметь:

- создавать объекты баз данных в современных системах управления базами данных и управлять доступом к этим объектам;
- работать с современными case-средствами проектирования баз данных;
- формировать и настраивать схему базы данных;
- разрабатывать прикладные программы с использованием языка SQL;
- создавать хранимые процедуры и триггеры на базах данных;
- применять стандартные методы для защиты объектов базы данных;
- *создавать и поддерживать в актуальном состоянии пользовательские учетные записи *;*
- *вести эксплуатационную и техническую документацию *;*

знать:

- основные положения теории баз данных, хранилищ данных, баз знаний;
- основные принципы построения концептуальной, логической и физической модели данных;
- современные инструментальные средства разработки схемы базы данных;
- методы описания схем баз данных в современных системах управления базами данных (СУБД);

- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями;
- основные методы и средства защиты данных в базах данных;
- модели и структуры информационных систем;
- основные типы сетевых топологий, приемы работы в компьютерных сетях;
- информационные ресурсы компьютерных сетей;
- технологии передачи и обмена данными в компьютерных сетях;
- основы разработки приложений баз данных;
- механизмы резервного копирования и восстановления информации*;
- основы CASE-средств и принципы их использования*.

1.3. Рекомендуемое количество часов на освоение программы профессионального модуля:

всего - **531** час, в том числе:

максимальной учебной нагрузки обучающегося - **351** час, включая:

обязательной аудиторной учебной нагрузки обучающегося **234** часа;

самостоятельной работы обучающегося **117** часов;

учебной и производственной практики – **180** часов

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результатом освоения программы профессионального модуля является овладение обучающимися видом профессиональной деятельности «Разработка и администрирование баз данных», в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 2.1.	Разрабатывать объекты базы данных.
ПК 2.2.	Реализовывать базу данных в конкретной системе управления базами данных (СУБД).
ПК 2.3.	Решать вопросы администрирования базы данных
ПК 2.4.	Реализовывать методы и технологии защиты информации в базах данных
ОК 1	Понимать сущность и социальную значимость своей профессии, проявлять к ней устойчивый интерес
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
ОК 10	Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

3. СТРУКТУРА И ПРИМЕРНОЕ СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Тематический план профессионального модуля

Коды профессиональных компетенций	Наименования разделов профессионального модуля	Всего часов (макс. учебная нагрузка и практики)	Объем времени, отведенный на освоение междисциплинарного курса (курсов)					Практика	
			Обязательная аудиторная учебная нагрузка обучающегося			Самостоятельная работа обучающегося		Учебная, часов	Производственная (по профилю специальности), часов <i>если предусмотрена рассредоточенная практика</i>
			Всего, часов	в т.ч. лабораторные работы и практические занятия, часов	в т.ч., курсовая работа (проект), часов	Всего, часов	в т.ч., курсовая работа (проект), часов		
1	2	3	4	5	6	7	8	9	10
ПК 2.1. ПК 2.2. ПК 2.3. ПК 2.4.	Раздел 1. Способы организации инфокоммуникационных систем и сетей	66	20	8	20 <i>(20*)</i>	10	10 <i>(10*)</i>	36	-
	Раздел 2. Реализация базы данных в СУБД Access	180 <i>(33*)</i>	120 <i>(22*)</i>	56		60 <i>(11*)</i>			-
	Раздел 3. Разработка и эксплуатация удаленных баз данных	213 <i>(9*)</i>	94 <i>(6*)</i>	48 <i>(4*)</i>		47 <i>(3*)</i>		72	-
	Производственная практика (по профилю специальности), часов (если предусмотрена итоговая (концентрированная) практика)	72							72
	Всего:	531 <i>(42*)</i>	234 <i>(28*)</i>	112 <i>(4*)</i>	20 <i>(20*)</i>	117 <i>(14*)</i>	10 <i>(10*)</i>	108	72

3.2. Содержание обучения по профессиональному модулю (ПМ) «Разработка и администрирование баз данных»

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. Организации инфокоммуникационных систем и сетей		20	
МДК.02.01. Инфокоммуникационные системы и сети			
Тема 1.1. Основные принципы обработки данных в инфокоммуникационных системах и сетях	Содержание учебного материала	4	
	1. Классификация инфокоммуникационных систем и сетей		1
	2. Принципы централизованной и распределенной обработки данных		1
Тема 1.2. Передача данных в инфокоммуникационных системах	Содержание учебного материала	8	
	1. Принципы пакетной передачи данных		1
	2. Серверы: назначение и функции		1
	3. Совместный доступ к серверу		1
	4. Обеспечение эффективного доступа к данным с помощью проектов		2
	Практические занятия	4	
	1. Настройка параметров соединения с сервером по алгоритму (ОК 2.1.1)		
	2. Обращение к данным сервера		
	Лабораторные работы	4	
	1. Создание проекта		
	2. Подключение проекта к серверу		
Самостоятельная работа Раздела 1 «Способы организации инфокоммуникационных систем и сетей» Составление словаря терминов по разделу «Способы организации инфокоммуникационных систем и сетей» (ОК 4.1.1) Заполнение таблицы «Основные проблемы и перспективы развития инфокоммуникационных сетей» Составление плана подключения проекта с использованием справочной информации (ОК 4.2.1) Реферирование темы «Современные средства совместного пользования данными» (ОК 4.1.1)		10	
Учебная практика «Технические средства организации инфокоммуникационных систем и сетей» Виды работ - Подключение технических средств и сетевого оборудования (оборудование доступа в Интернет, шлюзовое оборудование, серверы, модемы, средства защиты информации) - Использование технологий передачи и обмена данными в компьютерных сетях - Определение конфигурации оборудования при решении ситуационных задач - Проведение анализа и оценки совместимости аппаратного и программного обеспечения - Подключение оборудования для организации и управления многоцелевыми и специализированными базами данных		36	
Раздел 2. Создание объектов базы данных в современных СУБД		100	

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения
1	2	3	4
МДК.02.02. Технология разработки и защиты базы данных			
Тема 2.1. Теория проектирования баз данных	Содержание учебного материала	6	
	1. Базы данных как средство хранения и обработки информации. Цели создания БД.		1
	2. Системы управления базами данных. Основные понятия и терминология. Типы моделей данных. Взаимосвязи в моделях данных.		1
	3. Этапы проектирования баз данных.		2
	Практические занятия	4	
	1. Проектирование структуры базы данных по плану (ОК 2.1.1)		
2. Нормализация таблиц.			
Тема 2.2. Организация баз данных.	Содержание учебного материала	12	
	1. Методы создания новой БД и таблиц БД.		1
	2. Типы данных в БД		1
	3. Управление записями в таблицах БД.		1
	4. Сортировка, поиск и фильтрация данных		2
	5. Взаимосвязи между таблицами. Способы объединения таблиц		2
	6. Обеспечение достоверности, целостности и непротиворечивости данных. Каскадные воздействия	1	
	Лабораторные работы	16	
	1. Создание новой базы данных разными способами с использованием методических указаний (ОК 2.1.1, ОК 2.2.1)		
	2. Создание таблиц в различных режимах. Создание таблиц сложной структуры		
	3. Изменение свойств полей таблицы и форматов отображения данных.		
	4. Создание маски ввода и полей подстановки с использованием справочных данных (ОК 4.1.2)		
	5. Использование методов сортировки, поиска и фильтрации данных в таблице.		
6. Установление взаимосвязей между таблицами.			
7. Создание новой базы данных по индивидуальным заданиям (ОК 3.1.1)			
8. Использование внешних данных.			
Тема 2.3. Организация интерфейса с пользователяем	Содержание учебного материала	12	
	1. Формы, элементы управления формы: свойства, события и методы		1
	2. Способы создания многотабличных форм		2
	3. Типы меню. Работа с меню и окнами: создание, модификация, активация и удаление		1
	4. Кнопочные формы.		2
	5. Формирование и вывод отчетов		2
	6. Итоговые вычисления в отчетах	1	
	Лабораторные работы	16	
	1. Использование простой формы для загрузки базы данных.		
	2. Использование многотабличной формы для загрузки базы данных		
	3. Создание меню различных типов		
	4. Создание многостраничных форм		

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения
1	2	3	4
	5. Создание элементов управления рабочим окном. 6. Создание интерфейса (входной формы) 7. Создание составных отчетов 8. Создание и печать отчетов		
Тема 2.4. Организация запросов в базе данных	Содержание учебного материала 1. Типы запросов в базах данных 2. Групповые операции и выражения в запросах 3. Перекрестные запросы Лабораторные работы 1. Использование запросов для отбора и сортировки записей 2. Создание вычисляемых полей в запросах 3. Использование выражений в запросах с помощью Построителя 4. Редактирование данных с помощью запросов 5. Изменение данных в результирующем множестве запроса 6. Извлечение информации из базы данных и их содержательный анализ с помощью запросов по установленным пользователям критериям (ОК 4.2.2) 7. Создание сводной диаграммы	6 14	 1 1 1
Тема 2.5. Стандартные методы защиты базы данных.	Содержание учебного материала 1. Методы защиты базы данных 2. Контроль и проверка введенных данных 3. Установка системы защиты на уровне пользователей 4. Использование Мастера защиты Лабораторные работы 1. Использование Мастера защиты 2. Создание учетных записей 3. Назначение прав доступа к БД	8 6	 1 2 2 2
Самостоятельная работа Раздела 2 «Создание объектов базы данных в современных СУБД» Заполнение сводной таблицы «Сравнительная характеристика различных СУБД» Создание сравнительной таблицы «Методы создания новой БД: преимущества и недостатки» Создание сравнительной таблицы «Методы создания таблицы в БД: преимущества и недостатки» Создание сравнительной таблицы «Методы создания новой формы БД: преимущества и недостатки» Создание сравнительной таблицы «Методы создания отчетов в БД: преимущества и недостатки» Создание и заполнение сводной таблицы «Типы запросов в БД» Создание сводной таблицы «Виды групповых операций» Реферирование темы «Методы защиты базы данных» Осуществление самоконтроля выполнения работы по заданным критериям		50	
Раздел 3. Разработка и эксплуатация удаленных баз данных		94	

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения	
1	2	3	4	
МДК.02.03. Реализация удаленной базы данных				
Тема 3.1. Теория проектирования удаленных баз данных	Содержание учебного материала	10		
	1. Архитектуры удаленных баз данных.		1	
	2. Двух- и трехуровневые модели архитектуры УБД		1	
	3. Основные технологии доступа к данным		2	
	4. Типовые элементы доступа к данным		1	
	5. Введение в работу с удаленными базами данных	1		
	Практические занятия	2		
	1. Создание и заполнение сравнительной таблицы «Типы архитектур УБД»			
Тема 3.2. Проектирование серверной части приложения баз данных	Содержание учебного материала	6		
	1. <i>Основы CASE-средств и принципы их использования *</i>			
	2. Проектирование структуры базы данных с помощью команд		2	
	3. Визуальные средства проектирования структуры баз данных			
		Практические занятия	6	
	1. Проектирование структуры базы данных с использованием современных case-средств.			
	2. Выполнение процедуры нормализации таблиц (ОК 2.1.1)			
	3. Проведение сравнительного анализа типов SQL серверов и их особенностей (ОК 3.1.1)			
		Лабораторные работы	6	
	1. Создание серверной части приложения: файл базы данных, таблицы.			
2. Визуальное проектирование структуры базы данных: таблицы, индексы.				
3. Применение условий ссылочной целостности, взаимосвязей.				
Тема 3.3. Проектирование клиентской части приложений баз данных.	Содержание учебного материала	18		
	1. Компоненты доступа к данным.		1	
	2. Запросы на выборку и изменение данных. Компоненты отображения данных выборки.		2	
	3. Хранимые процедуры, триггеры и генераторы. Обеспечение достоверности, целостности и непротиворечивости данных. Каскадные воздействия.		1	
	4. Сортировка, фильтрация и поиск данных в базе данных и в выборках.		2	
	5. Управление транзакциями и кэширование памяти		1	
	6. Перехват исключительных ситуаций		1	
	7. Обработка ошибок		2	
	8. Формирование и вывод отчетов		2	
	9. Особенности проектирование клиентской части приложения в различных технологиях доступа к данным	1		
		Лабораторные работы	20	
	1. Размещение визуальных и не визуальных компонентов в клиентской части.			
	2. Создание запросов на редактирование			
	3. Создание запросов на добавление и удаление данных.			
4. Создание генератора, триггеров				

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения
1	2	3	4
	5 Создание хранимых процедур 6 Использование методов сортировки, поиска и фильтрации данных: в базе данных и в выборках. 7 Управление транзакциями. 8 Кэширование изменений при работе с транзакциями. 9 Обеспечение достоверности данных и перехват исключительных ситуаций. 10 Создание отчетов по запросам пользователя		
Тема 3.4. Технология конструирования SQL-запросов.	Содержание учебного материала 1 Особенности технологии конструирования SQL запросов 2 Корректировка данных средствами запросов 3 Запросы к серверу Лабораторные работы 1 Создание запросов SQL 2 Использование инструкций SQL для создания запросов 3 Создание запросов на объединение записей 4 Создание управляющих запросов	6 8	
Тема 3.5. Администрирование и эксплуатация удаленных баз данных.	Содержание учебного материала 1 Установка привилегий доступа к данным. 2 <i>Копирование и перенос данных. Восстановление данных *</i> 3 Копирование клиентской части приложения баз данных Лабораторные работы 1. Установление привилегий доступа. 2. <i>Копирование и восстановление данных.*</i> 3. <i>Создание резервной копии клиентской части.*</i>	6 6	1 1 1
Самостоятельная работа Раздела 3 «Разработка и эксплуатация удаленных баз данных» Осуществление самоконтроля выполнения работы по заданным критериям Разработка сравнительной таблицы «Типы архитектур УБД» Заполнение сводной таблицы «Современные Case-средства»* Реферирование тем: «Особенности проектирования клиентской части приложения в различных технологиях доступа к данным», «Способы копирования и восстановления удаленных данных*»		40	
Учебная практика «Разработка электронной базы данных» Виды работ - Создание таблицы БД в различных режимах по предложенному заданию - Установление стандартных типов данных для указанных по лей - Использование методов ограничения на вводимую информацию - Установление форматов отображения данных - Использование поля подстановки и маски ввода - Определение связи между указанными таблицами базы - Использование методов сортировки и фильтрации данных БД по предложенному заданию - Создание формы в различных режимах по имеющимся в БД объектам - Использование загрузки базы данных с помощью простой и многотабличной форм.		72	

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работ (проект)	Объем часов	Уровень освоения
1	2	3	4
<ul style="list-style-type: none"> - Создание различных видов отчетов - Создание запросов в различных режимах по указанным параметрам - Использование групповых операций и выражений в запросах - Редактирование и анализ данных с помощью запросов - Создание серверной части приложения для решения предложенной задачи - Создание клиентской части приложения для решения предложенной задачи - Создание генераторов, триггеров, хранимых процедур для решения поставленных задач - Управление транзакциями - Обеспечение достоверности данных и перехват исключительных ситуаций.. 			
<p>Производственная практика по профилю специальности «Создание и внедрение информационно-аналитических баз данных»</p> <p>Виды работ</p> <ul style="list-style-type: none"> - Создание файла БД, настраивание интерфейса и способов запуска БД - Использование различных типов связей между созданными таблицами базы - Заполнение БД методом импорта данных , методом непосредственного ввода - Создание меню различных типов - Создание пользовательского интерфейса БД - Определение данных, подлежащих выводу на печать в проекте - Извлечение необходимой информации из базы данных с помощью запросов - Использование групповых операций и выражений в запросах, формах и отчетах - Редактирование и анализ данных с помощью перекрестных запросов - Проектирование серверной и клиентской части приложения - Обеспечение достоверности данных и защиты данных от ошибок пользователя. - Управление учетными записями и правами доступа для приложений, защищённых на уровне пользователей - Оформление пояснительной записки к разработанному проекту 		72	
<p>Примерная тематика курсовых работ (проектов)</p> <p><i>Разработка и проектирование базы данных «Автоматизированное рабочее место специалиста по отраслям» (*)</i></p>			
<p>Обязательная аудиторная учебная нагрузка по курсовой работе (проекту)</p> <p>Самостоятельная работа при выполнении КП</p> <p><i>Определение цели, задач и требований к автоматизированному рабочему месту, согласно заданию</i></p> <p><i>Определение необходимых ресурсов для разработки базы данных*</i></p> <p><i>Формирование требований пользователя к базе данных*</i></p> <p><i>Разработка автоматизированного рабочего места специалиста по отраслям*</i></p> <p><i>Оформление технологической документации базы данных*</i></p> <p><i>Представление разработанной базы на защите курсовой работы</i></p>		20*	
<p>Всего:</p>		531	

Используемые обозначения характеристики уровня освоения учебного материала:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);
2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Требования к минимальному материально-техническому обеспечению

Реализация программы модуля предполагает наличие лабораторий: технологии разработки баз данных, информационно-коммуникационных систем, управления проектной деятельностью; полигона учебных баз практики.

Оборудование учебного кабинета и рабочих мест кабинета: комплект учебной мебели, комплект плакатов; комплект учебно-методической документации.

Технические средства обучения: точки электропитания; сетевое оборудование, обеспечивающее работу локальной сети, мультимедийное оборудование; источники бесперебойного питания; интерактивная доска; подключение к глобальной сети Internet.

Оборудование лаборатории и рабочих мест лаборатории: комплект учебной мебели, персональных компьютеров, точки электропитания; сетевое оборудование, обеспечивающее работу локальной сети, мультимедийное оборудование; источники бесперебойного питания; интерактивная доска; подключение к глобальной сети Internet.

Реализация программы модуля предполагает учебную практику.

Оборудование и технологическое оснащение рабочих мест: комплект учебной мебели, персональных компьютеров, точки электропитания; сетевое оборудование, обеспечивающее работу локальной сети, мультимедийное оборудование; источники бесперебойного питания; интерактивная доска; подключение к глобальной сети Internet.

Реализация программы модуля предполагает обязательную производственную практику.

Оборудование и технологическое оснащение рабочих мест: комплект мебели, персональный компьютер, точка электропитания; сетевое оборудование, обеспечивающее работу локальной сети, мультимедийное оборудование; источники бесперебойного питания; подключение к глобальной сети Internet.

4.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Кузин А.В., Демин В.М. Разработка баз данных в системе Microsoft Access: - учебник. – 2 изд., - М.:Форум: ИНФРА-М, 2007

Дополнительные источники:

1. Кузин А.В., Демин В.М. Разработка баз данных в системе MICROSOFT ACCESS:учебник для ССУЗ.-М.: ФОРУМ: ИНФРА – М,2005
2. Голицина О.Л. Базы данных: учебник для ССУЗ. – М.: ФОРУМ: ИНФРА - М, 2003.
3. Кузин А.В. Разработка базы данных в системе Microsoft Access: учебник для СПО. – М.: ФОРУМ: ИНФРА - М, 2005.
4. Дубнов П.Ю. Access 2000. Программирование баз данных: учебник для ССУЗ. – М.: Вильямс, 2000.
5. Партыка Т.Л., Попов И.И. Информационная безопасность: учеб.пособ. для ССУЗ.–М.: Форум : Инфра.М., 2004
6. Бекаревич Ю., Пушкина Н. Microsoft Access 2000. – СПб, 1999
7. Робинсон С. Microsoft Access 2000: учебный курс –СПб/Питер, 2001
8. Агальцов В.П. Базы данных: Учебное пособие. - М: Мир, 2002.
9. Фаронов В.В. Программирование баз данных в Delphi 7. Учебный курс. – СПб.: Питер, 2005. - 459 с.
10. Фаронов В.В., Шумаков П.В. Delphi 4. Руководство разработчика баз данных. - М.: Нолидж, 1999. - 560 с.
11. Бобровский С. Delphi 5: учебный курс. – СПб: Питер, 2001
12. Голицына О.Л., Максимов Н.В., Попов И.И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2004 – 352с.

4.3. Общие требования к организации образовательного процесса

Перед изучением модуля обучающиеся изучают следующие дисциплины: «Теория алгоритмов», «Операционные системы и среды», «Основы программирования», «Элементы математической логики», «Теория вероятностей и математическая статистика», «Архитектура компьютерных систем», «Технические средства информатизации», «Экономика». До начала учебной и производственно практики необходимо изучить также дисциплины: «Правовое обеспечение профессиональной деятельности», «Безопасность жизнедеятельности»

4.4. Кадровое обеспечение образовательного процесса

Требования к квалификации педагогических кадров, обеспечивающих обучение по междисциплинарному курсу: наличие высшего педагогического образования, соответствующего профилю модуля.

Требования к квалификации педагогических кадров, осуществляющих руководство практикой: наличие высшего педагогического образования, соответствующего профилю модуля.

Инженерно-педагогический состав: средне -профессиональное или высшее инженерное образование, соответствующее профилю модуля

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ПК 2.1. Разрабатывать объекты базы данных	Объекты БД выполнены с соблюдением этапов их разработки и критериями задания	Оценка продукта учебной деятельности (объектов базы данных) по критериям на учебной практике
ПК 2.2. Реализовывать базу данных в конкретной системе управления базами данных (СУБД)	Реализация базы данных в конкретной системе управления базами данных (СУБД) выполнена с учетом требований: 1) управление всеми сведениями базы данных осуществляется из одного файла; 2) в рамках файла базы данных используются следующие объекты: - таблицы для сохранения данных; - запросы для поиска и извлечения только требуемых данных; - формы для просмотра, добавления и изменения данных в таблицах; - отчеты для анализа и печати данных в определенном формате; 3) в файле базы данных настроены параметры запуска БД и параметры автоматизации ввода данных	Оценка результатов стандартизованного тестирования сопоставлением с эталоном (ключом, модельным ответом) на сертификационном экзамене
ПК 2.3. Решать вопросы администрирования базы данных	Решение вопросов администрирования базы данных осуществляется с помощью средств управления доступом к информации базы данных и настройки параметров подключения к файлу базы данных	Оценка продукта профессиональной деятельности (базы данных) по критериям на

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
	<p>Реализация методов защиты информации БД в конкретной системе управления базами данных (СУБД) выполнена с учетом требований:</p> <ol style="list-style-type: none"> 1) файл БД недоступен для чтения с помощью служебных программ или текстовых редакторов; 2) установлены параметры системы защиты стандартными средствами MS Access 3) Параметры системы защиты установлены и настроены средствами SQL Server 4) в файле базы данных предусмотрена защита данных и уровни ограничения доступа 	производственной практике
ПК 2.4. Реализовывать методы и технологии защиты информации в базах данных.	<p>Реализация технологий защиты информации в базе данных в конкретной системе управления базами данных (СУБД) выполнена с учетом требований:</p> <ol style="list-style-type: none"> 1) Обеспечение безопасности осуществляется путем установления параметров шифрования и дешифрования данных БД; 2) Регистрация пользователей и права доступа соответствуют формулировкам целей создания БД и перечню потребностей пользователей 	Оценка продукта профессиональной деятельности (базы данных) по критериям на производственной практике

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся не только сформированность профессиональных компетенций, но и развитие общих компетенций и обеспечивающих их умений.

Результаты обучения (освоенные общие компетенции)		Основные показатели оценки результата	Формы и методы контроля и оценки результатов обучения
Код	Формулировка		
ОК 1	Понимать сущность и социальную значимость своей профессии, проявлять к ней устойчивый интерес	Приведены произвольные примеры социальной значимости своей профессии Дано объяснение сущности профессии	Оценка результатов стандартизованного тестирования на итоговом испытании
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их	Поставленная цель разбита на задачи. Набор ресурсов определен в соответствии с условиями предложенного задания Выбранный метод и способ решения профессиональной задачи соответствует типовому	Оценка продукта деятельности обучающегося (решение практико-ориентированного задания) по критериям на

<i>Результаты обучения (освоенные общие компетенции)</i>		<i>Основные показатели оценки результата</i>	<i>Формы и методы контроля и оценки результатов обучения</i>
<i>Код</i>	<i>Формулировка</i>		
	эффективность и качество.	(известному) алгоритму решения Оценка эффективности и качества метода и способа решения задачи соответствует заданной методике оценивания	Итоговом испытании
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Анализ проблемы проведен в соответствии с условиями заданной стандартной / нестандартной ситуации Принятое решение соответствует результатам проведенного анализа проблемы и позволяет ее решить	Оценка результатов формализованного наблюдения за деятельностью обучающегося по критериям на учебной практике
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	Обращается к информационным системам, базам и банкам компьютерных данных по интересующему вопросу в ходе решения поставленной профессиональной задачи Общается со специалистами по интересующему вопросу	Оценка результатов формализованного наблюдения за учебной / профессиональной деятельностью обучающегося по критериям на учебной / профессиональной практике
ОК 5	Использовать информационно - коммуникационные технологии в профессиональной деятельности.	Использование информационно - коммуникационных технологий в профессиональной деятельности соответствует условиям поставленной профессиональной задачи и характеристикам имеющегося аппаратного обеспечения.	Оценка продукта учебной / профессиональной деятельности обучающегося по критериям на учебной/производственной практике
ОК 6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	Предложена идея для достижения цели групповой работы Озвученная идея аргументирована/разъяснена всему коллективу	Оценка результатов формализованного наблюдения за деятельностью обучающегося по критериям во время выполнения группового проекта на практических занятиях
ОК 7	Брать на себя ответственность за работу членов команды	Общая цель задания четко сформулирована и разъяснена всем членам коллектива/команды	Экспертная оценка продукта учебной деятельности (коллективного)

<i>Результаты обучения (освоенные общие компетенции)</i>		<i>Основные показатели оценки результата</i>	<i>Формы и методы контроля и оценки результатов обучения</i>
<i>Код</i>	<i>Формулировка</i>		
	(подчиненных), за результат выполнения заданий.		проекта) по критериям на защите
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	Задачи профессионального развития определены в соответствии с условиями практико-ориентированного задания Запланированное повышение квалификации соответствует запросам потенциальных работодателей, предъявляемым к специалисту «техник-программист»	Оценка результатов стандартизованного тестирования на итоговом испытании
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Перечисляет современные программные и аппаратные ресурсы, соответствующие условиям поставленной профессиональной задачи Формулирует основные характеристики имеющихся программных и аппаратных ресурсов Осваивает предложенную технологию по имеющейся сопроводительной документации	Оценка результатов стандартизованного тестирования на итоговом испытании
ОК 10	Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).	Приводит произвольные примеры применения полученных знаний при исполнении воинской обязанности	Экспертная оценка результатов стандартизованного тестирования на итоговом испытании

РАЗДЕЛ 2. КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН МЕЖДИСЦИПЛИНАРНОГО КУРСА
МДК 02.02. Технология разработки и защиты базы данных

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
1	Раздел 2. Создание объектов базы данных в современных СУБД								
Тема 2.1. Теория проектирования баз данных									
1	Базы данных как средство хранения и обработки информации. Цели создания БД.	2		1	теор	ММП (мультимедиа проектор)	Составление опорного конспекта (ОК 2.1.1)		
2	Системы управления базами данных. Основные понятия и терминология. Типы моделей данных. Взаимосвязи в моделях данных.	2		2	теор	ММП	Составление опорного конспекта (ОК 2.1.1)		
3	Этапы проектирования баз данных.	2		3	теор	ММП	Составление опорного конспекта (ОК 2.1.1)		
4	Проектирование структуры базы данных по плану (ОК 2.1.1)	2		4	практ	ММП, ДМ (дидактический материал)	Выполнение упражнений		
5	Нормализация таблиц.	2		5	практ	ММП, ДМ	Выполнение упражнений		
Тема 2.2. Организация баз данных									
6	Методы создания новой БД и таблиц БД.	2		6	теор	ММП	Составление конспекта (ОК 2.1.1) Составление сводных таблиц (ОК 3.1.1)		
7	Типы данных в БД	2		7	теор	ММП	Составление конспекта (ОК 4.1.1), Работа с дополнительной литературой, Заполнение сводных таблиц (ОК 2.1.1)		
8	Управление записями в таблицах БД.	2		8	теор	ММП	Составление конспекта (ОК 2.1.1)		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
9	Сортировка, поиск и фильтрация данных	2		9	теор	ММП	Составление конспекта (ОК 2.1.1)		
10	Взаимосвязи между таблицами. Способы объединения таблиц	2		10	теор	ММП	Составление конспекта (ОК 2.1.1)		
11	Обеспечение достоверности, целостности и непротиворечивости данных. Каскадные воздействия	2		11	теор	ММП	Составление конспекта (ОК 2.1.1)		
12	Создание новой базы данных разными способами с использованием методических указаний (ОК 2.1.1, ОК 2.2.1)	2		12	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
13	Создание таблиц в различных режимах. Создание таблиц сложной структуры	2		13	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
14	Изменение свойств полей таблицы и форматов отображения данных.	2		14	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
15	Создание маски ввода и полей подстановки с использованием справочных данных (ОК 4.1.2)	2		15	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
16	Использование методов сортировки, поиска и фильтрации данных в таблице.	2		16	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
17	Установление взаимосвязей между таблицами.	2		17	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
							выполнения работы по заданным критериям		
18	Создание новой базы данных по индивидуальным заданиям (ОК 3.1.1)	2		18	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
19	Использование внешних данных.	2		19	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
Тема 2.3. Организация интерфейса с пользователем									
20	Формы, элементы управления формы: свойства, события и методы	2		20	теор	Эл.учебник	Составление конспекта (ОК 2.1.1)		
21	Способы создания многотабличных форм	2		21	теор	ММП	Составление конспекта (ОК 2.1.1)		
22	Типы меню. Работа с меню и окнами: создание, модификация, активация и удаление	2		22	теор	ММП	Составление конспекта (ОК 2.1.1)		
23	Кнопочные формы.	2		23	теор	ММП	Составление конспекта (ОК 2.1.1) Составление сводной таблицы (ОК 3.1.2)		
24	Формирование и вывод отчетов	2		24	теор	ММП	Составление опорного конспекта(ОК 2.1.1) Создание сравнительной таблицы «Методы создания новой формы БД: преимущества и недостатки» (ОК 3.1.2)		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
25	Итоговые вычисления в отчетах	2		25	теор	ММП	Составление опорного конспекта по теме (ОК 4.2.1) Создание сравнительной таблицы «Методы создания отчетов в БД: преимущества и недостатки» (ОК 3.1.2)		
26	Использование простой формы для загрузки базы данных.	2		26	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
27	Использование многотабличной формы для загрузки базы данных	2		27	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
28	Создание меню различных типов	2		28	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
29	Создание многостраничных форм	2		29	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
30	Создание элементов управления рабочим окном.	2		30	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
31	Создание интерфейса (входной формы)	2		31	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
32	Создание составных отчетов	2		32	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
33	Создание и печать отчетов	2		33	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
Тема 2.4. Организация запросов в базе данных									
34	Типы запросов в базах данных	2		34	теор	ММП	Составление опорного конспекта (ОК 2.1.1) Создание и заполнение сводной таблицы «Типы запросов в БД» (ОК 3.1.2)		
35	Групповые операции и выражения в запросах	2		35	теор	ММП	составление опорного конспекта (ОК 2.1.1) Составление сводной таблицы (ОК 3.1.2)		
36	Перекрестные запросы	2		36	теор	ММП	составление опорного конспекта (ОК 2.1.1)		
37	Использование запросов для отбора и сортировки записей	2		37	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
38	Создание вычисляемых полей в запросах	2		38	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
39	Использование выражений в запросах с помощью Построителя	2		39	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
40	Редактирование данных с помощью запросов	2		40	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
41	Изменение данных в результирующем множестве запроса	2		41	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
42	Извлечение информации из базы данных и их содержательный анализ с помощью запросов по установленным критериям (ОК 4.2.2)	2		42	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
43	Создание сводной диаграммы	2		43	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
Тема 2.5. Стандартные методы защиты базы данных									
44	Методы защиты базы данных	2		44	теор	ММП	Реферирование темы «Методы защиты базы данных» составление опорного конспекта		
45	Контроль и проверка введенных данных	2		45	теор	ММП	составление опорного конспекта (ОК 2.1.1)		
46	Установка системы защиты на уровне пользователей	2		46	теор	ММП	составление опорного конспекта (ОК 2.1.1)		

№	Наименование разделов, тем	Кол. час.	Дата	№ занятия	Вид занятия	Оборудование занятия	Самостоятельная работа студентов	Дом. задание	Прим.
47	Использование Мастера защиты	2		47	теор	ММП	Выполнение упражнений на закрепление материала Выполнение упражнений на закрепление материала		
48	Использование Мастера защиты	2		48	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
49	Создание учетных записей	2		49	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		
50	Назначение прав доступа к БД	2		50	лабор	ДМ , ПК	Выполнение упражнений Осуществление самоконтроля выполнения работы по заданным критериям		

Обязательная аудиторная учебная нагрузка (всего)	100
в том числе:	
лабораторные занятия	52
практические занятия	4

РАЗДЕЛ 3. ЛЕКЦИОННЫЙ МАТЕРИАЛ

План-конспект занятия №1

Базы данных как средство хранения и обработки информации. Цели создания БД.

Цель занятия: познакомить с основными понятиями и типами баз данных.

Вид занятия: теоретическое занятие

Тип занятия: комбинированный урок

Задачи:

- Учебные: 1) разъяснить осн. задачи и области применения БД;
2) дать основные понятие и типы моделей данных;
3) научить решать задачи на создание БД;

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности

Межпредметные связи: Математика, Логика, Алгоритмизация и программирование, Экономика и менеджмент

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта (ОК 2.1.1),

Ход занятия:

1. Организационный момент 1-3 мин
(взаимное приветствие, постановка темы и задач занятия)
2. Объяснение нового материала 60 мин
 - а) Введение в дисциплину,
 - б) Основные понятия и типы моделей данных
3. Закрепление материала 20 мин
(обобщение полученных знаний)
4. Подведение итогов занятия. 1-3 мин
5. Домашнее задание: 1-3мин
 - а) выучить конспект.

Введение

Сегодня трудно себе представить сколько-нибудь значимую информационную систему, которая не имела бы в качестве основы или важной составляющей базу данных. Концепции и технологии баз данных складывались постепенно и всегда были тесно связаны с развитием систем автоматизированной обработки информации. Создание баз данных после появления реляционного подхода превратилось из искусства в науку, но, как показала практика последних лет, все же окончательно его не исключившая. Тем не менее, сейчас это вполне сложившаяся дисциплина (хотя являющаяся скорее инженерной, чем чисто научной), основанная на достаточно формализованных подходах и включающая широкий спектр приемов и методов создания баз данных.

Бадам свойственна «перманентность» данных. Соответственно назначение систем управления базами данных — обеспечение в течение длительного времени их сохранности, а также возможностей выборки и актуализации. Данные существуют всегда, пока есть потребность в их использовании, хотя характер использования, как и пути извлечения практической пользы

могут быть самыми разными: от оперативной актуализации значений до уничтожения данных, от их использования для совершенствования сложных систем управления до формирования «чемоданов компромата».

Базы данных в стремительно, а в какой-то степени и сумбурно развивающихся информационных технологиях — это сравнительно консервативное направление, где СУБД и сами базы представляют собой «долговременные сооружения». Элементная база ЭВМ и парадигмы программирования меняются быстрее, чем хранимые данные теряют актуальность. В таких условиях, в отличие от прикладных программистов, создатели баз данных (от разработчиков СУБД до администраторов БД) должны постоянно помнить о проблеме «наследственности» — о том, как интегрировать в создаваемую систему наследуемые данные, находящиеся под управлением устаревшей СУБД, и о том, как построить систему, чтобы вновь создаваемые данные могли быть, в свою очередь, наследованы следующим поколением систем и разработчиков.

Достаточно консервативны и концепции баз данных. Эта консервативность — следствие не только свойства «долговечности», но и того факта, что базы вторичны по отношению к описываемым ими реальным процессам и объектам, достаточно стабильным и типичным. Кроме того, модели данных строились в значительной степени «по аналогии» с организационными и технологическими структурами — иерархическими, сетевыми, матричными.

Широкое использование баз данных различными категориями пользователей привело, с одной стороны, к созданию интерфейсов, требующих минимум времени на освоение средств управления системой, а с другой — к построению мощных, гибких СУБД, имеющих в том числе развитые средства защиты данных от случайного или преднамеренного разрушения. Появились и средства автоматизации разработки, позволяющие создать базу данных любому пользователю, даже не владеющему основами теории БД.

Но, как было отмечено ранее, база данных — это важная, но не основная (функционально), а обеспечивающая (информационная) составляющая некоторой, обычно достаточно крупной *человеко-машинной* системы. И здесь интересно отметить принципиальное отличие в развитии способностей взаимодействующих субъектов (человек — машина). Разделение информации на *табличную (числовую), текстовую и графическую* отражает последовательность, в которой эти виды информации «осваивались» компьютерами. Первые языки программирования были рассчитаны исключительно на обработку числовой информации (Fortran, Algol). Первыми появляются и табличные базы данных, также преимущественно рассчитанные на обработку числовых таблиц (файлов). Затем осваиваются текстовые файлы и текстовые БД (автоматизированные информационно-поисковые системы с библиографическими и полнотекстовыми базами). Наконец, с существенным повышением быстродействия и емкости памяти компьютеров на сцену выходят графические и мультимедийные базы.

Создание практически полезной «серьезной» базы данных в равной степени зависит как от «фундаментальности» знаний разработчика в области концепций и технологий СУБД, так и от степени понимания им сегодняшних и будущих прикладных задач пользователя, не только от адекватности применения тех или иных типовых или оригинальных решений, но и от качества представления (описания) этих решений, с той или иной степенью успешности позволяющих использовать, сопровождать и развивать систему после разработчика.

Кроме того, возможности накапливать и оперативно обрабатывать большие объемы информации, характеризующие деятельность предприятий за достаточно длительные периоды и в различных аспектах, дали новый импульс к развитию аналитических систем. Такого рода *системы поддержки принятия решений* обычно используются для оценки и выбора альтернативных решений, прогнозирования, идентификации объектов и состояний и т. д. Однако, поскольку для получения необходимых данных в этих случаях нужно использовать сложные SQL-запросы или специализированные процедуры, и при этом обрабатывать большие объемы записей, то уже это может приводить к сознательному отказу от классических нормализованных схем, так как чем выше степень нормализации, тем больше число операций соединения отношений и, соответственно, больше времени необходимо для получения конечного результата.

Понятие базы и банка данных

Развитие вычислительной техники и появление емких внешних запоминающих устройств прямого доступа предопределило интенсивное развитие автоматических и автоматизированных систем разного назначения и масштаба, в первую очередь заметное в области бизнес-приложений. Такие системы работают с большими объемами информации, которая обычно имеет достаточно сложную структуру, требует оперативности в обработке, часто обновляется и в то же время требует длительного хранения. Примерами таких систем являются автоматизированные системы управления предприятием, банковские системы, системы резервирования и продажи билетов и т.д. (рис 1.1). Другими направлениями, стимулировавшими развитие, стали, с одной стороны, системы управления физическими экспериментами, обеспечивающими сверхоперативную обработку в реальном масштабе времени огромных потоков данных от датчиков, а с другой — автоматизированные библиотечные информационно-поисковые системы.

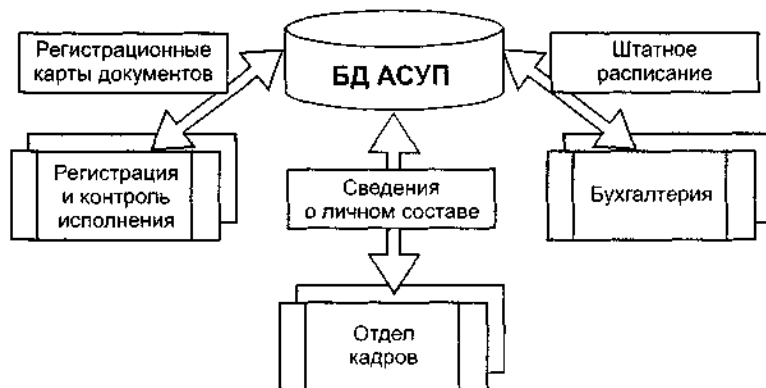


Рис 1.1 Схема автоматизированной информационной системы

Это привело к появлению новой информационной технологии интегрированного хранения и обработки данных — *концепции баз данных*, в основе которой лежит механизм предоставления обрабатывающей программе из всех хранимых данных только тех, которые ей необходимы, и в форме, требуемой именно этой программе. При этом сама форма (структура данных и форматы полей, входящих в эту структуру) описывается на логическом, т.е. «видимом» из программы, уровне. Более того, поскольку различные программы могут по-разному «видеть» (а следовательно, и использовать) одни и те же данные, то система должна сделать «прозрачными» для программы все данные, кроме тех, которые для нее являются «своими».

Банк данных (БД) — это система специально организованных данных, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного многоцелевого использования данных².

Под *базой данных (БД)* обычно понимается именованная совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области. Характерной чертой баз данных является *постоянство*: данные *постоянно* накапливаются и используются; состав и структура данных, необходимых для решения тех или иных прикладных задач, обычно *постоянны* и стабильны во времени; отдельные или даже все элементы данных могут меняться — но и это есть проявление постоянства — *постоянная* актуальность.

Система управления базами данных (СУБД) — это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Иногда в составе банка данных выделяют *архивы*. Основанием для этого является особый режим использования данных, когда только часть данных находится под оперативным управлением СУБД. Все остальные данные (собственно архивы) обычно располагаются на носителях, оперативно не управляемых СУБД. Одни и те же данные в разные моменты времени могут входить как в базы данных, так и в архивы. Банки данных могут не иметь архивов, но если они есть, то в состав банка данных может входить и система управления архивами.

Проблемы совместного использования данных и периферийных устройств компьютеров и рабочих станций быстро породили модель вычислений, основанную на концепции файлового

сервера — сеть создает основу для коллективной обработки, сохраняя простоту работы с персональным компьютером, позволяет совместно использовать данные и периферию.

В этом смысле главной отличительной чертой баз данных является использование централизованной системы управления данными, причем как на уровне файлов, так и на уровне элементов данных. Централизованное хранение совместно используемых данных приводит не только к сокращению затрат на создание и поддержание данных в актуальном состоянии, но и к сокращению избыточности информации, упрощению процедур поддержания непротиворечивости и целостности данных

Эффективное управление внешней памятью является основной функцией СУБД. Эти обычно специализированные средства настолько важны с точки зрения эффективности, что при их отсутствии система просто не сможет выполнять некоторые задачи уже потому, что их выполнение будет занимать слишком много времени. При этом ни одна из таких специализированных функций, как построение индексов, буферизация данных, организация доступа и оптимизация запросов, не является видимой для пользователя и обеспечивает независимость между логическим и физическим уровнями системы прикладной программист не должен писать программы индексирования, распределять память на диске и т. д.

Развитие теории и практики создания информационных систем, основанных на концепции баз данных, создание унифицированных методов и средств организации и поиска данных позволяют хранить и обрабатывать информацию о все более сложных объектах и их взаимосвязях, обеспечивая многоаспектные информационные потребности различных пользователей

Основные требования, предъявляемые к банкам данных, можно сформулировать следующим образом

Множественное использование данных: пользователи должны иметь возможность использовать данные различным образом

Простота пользователи должны иметь возможность легко узнать и понять, какие данные имеются в их распоряжении

Легкость использования' пользователи должны иметь возможность осуществлять (процедурно) простой доступ к данным, при этом все сложности доступа к данным должны быть скрыты в самой системе управления базами данных

Гибкость использования: обращение к данным или их поиск должны осуществляться с помощью различных методов доступа.

Быстрая обработка запросов на данные: запросы на данные, в том числе незапланированные, должны обрабатываться с помощью высокоуровневого языка запросов, а не только прикладными программами, написанными с целью обработки конкретных запросов (разработка таких программ в каждом конкретном случае связана с большими затратами времени). Пользователь должен иметь возможность кратко выразить нетривиальные запросы (в нескольких словах или несколькими нажатиями клавиш мыши). Это означает, что средство формулирования должно быть достаточно «декларативным», т. е. упор должен быть сделан на «что», а не на «как». Кроме того, средство обработки запросов не должно зависеть от приложения, т. е. оно должно работать с любой возможной базой данных.

Язык взаимодействия конечных пользователей с системой должен обеспечивать конечным пользователям возможность получения данных без использования прикладных программ.

База данных — это основа для будущего наращивания прикладных программ: базы данных должны обеспечивать возможность быстрой и дешевой разработки новых приложений.

Сохранение затрат умственного труда: существующие программы и логические структуры данных (на создание которых обычно затрачивается много человеко-лет) не должны переделываться при внесении изменений в базу данных.

Наличие интерфейса прикладного программирования: прикладные программы должны иметь возможность просто и эффективно выполнять запросы на данные; программы должны быть изолированы от расположения файлов и способов адресации данных.

Распределенная обработка данных: система должна функционировать в условиях вычислительных сетей и обеспечивать эффективный доступ пользователей к любым данным распределенной БД, размещенным в любой точке сети.

Адаптивность и расширяемость: база данных должна быть настраиваемой, причем настройка не должна вызывать перезаписи прикладных программ. Кроме того, поставляемый с СУБД набор предопределенных типов данных должен быть расширяемым — в системе должны иметься средства для определения новых типов и не должно быть различий в использовании системных и определенных пользователем типов.

Контроль за целостностью данных: система должна осуществлять контроль ошибок в данных и выполнять проверку взаимного логического соответствия данных.

Восстановление данных после сбоев: автоматическое восстановление без потери данных транзакции. В случае аппаратных или программных сбоев система должна возвращаться к некоторому согласованному состоянию данных.

Вспомогательные средства должны позволять разработчику или администратору базы данных предсказать и оптимизировать производительность системы.

Автоматическая реорганизация и перемещение: система должна обеспечивать возможность перемещения данных или автоматическую реорганизацию физической структуры.

Пользователи баз данных

В информационных системах, создаваемых на основе СУБД, способы организации данных и методы доступа к ним перестали играть решающую роль, поскольку оказались скрытыми внутри СУБД. Массовый, так называемый *конечный пользователь*, как правило, имеет дело только с внешним интерфейсом, поддерживаемым СУБД.

Эти преимущества, как уже понятно, не могут быть реализованы путем механического объединения данных в БД. Предполагается, что в системе обязательно существует специальное должностное лицо (группа лиц) — *администратор базы данных (АБД)*, который несет ответственность за проектирование и общее управление

базой данных. АБД определяет информационное содержание БД. С этой целью он идентифицирует объекты БД и моделирует базу, используя язык описания данных. Получаемая модель служит в дальнейшем справочным документом для администраторов приложений и пользователей. Администратор решает также все вопросы, связанные с размещением БД в памяти, выбором стратегии и ограничений доступа к данным. В функции АБД входят также организация загрузки, ведения и восстановления БД и многие другие действия, которые не могут быть полностью формализованы и автоматизированы.

Администратор приложений (или, если таковой специально не выделяется — администратор БД) определяет для приложений подмодели данных. Тем самым разные приложения обеспечиваются собственным «взглядом», но не на всю БД, а только на требуемую для конкретного приложения («видимую») ее часть. Вся остальная часть БД для данного приложения будет «прозрачна».

Прикладные программисты имеют, как правило, в своем распоряжении один или несколько языков программирования, с помощью которых генерируются прикладные программы.

План-конспект занятия №2

Системы управления базами данных. Основные понятия и терминология. Типы моделей данных. Взаимосвязи в моделях данных.

Цель занятия: познакомить с основными понятиями и типами СУБД.

Вид занятия: теоретическое занятие

Тип занятия: комбинированный урок

Задачи:

Учебные: 1) разъяснить понятие СУБД;

2) дать основные понятие и типы моделей данных;

3) научить решать задачи на создание БД;

Развивающие:

1) развитие логического мышления;

2) развитие памяти

Воспитательные:

1) воспитание аккуратности и внимательности

Межпредметные связи: Математика, Логика, Алгоритмизация и программирование, Экономика и менеджмент

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта (ОК 2.1.1),

Ход занятия:

1. Организационный момент 1-3 мин
(взаимное приветствие, постановка темы и задач занятия)
2. Блиц-опрос (проверка ДЗ) 10 мин
3. Объяснение нового материала 50 мин
 - а) Введение в дисциплину,
 - б) Основные понятия и типы моделей данных
4. Закрепление материала 20 мин
(обобщение полученных знаний)
5. Подведение итогов занятия. 1-3 мин
6. Домашнее задание: 1-3 мин
 - а) выучить конспект.

Блиц-опрос (проверка ДЗ)

1. . Что из перечисленного НЕ является основополагающим в развитии автоматизированных систем:

- А) развитие вычислительной техники
- Б) появление емких внешних носителей
- В) *появление языка BASIC*
- Г) появление информационно-поисковых систем

2. Какое понятие шире – банк данных или база данных:

- А) Банк данных
- Б) база данных

3. Назовите характерную черту баз данных:

- А) архивирование данных
- Б) *Постоянство*
- В) централизованное хранение данных
- Г) многопользовательский интерфейс

4. Возможность обращаться к данным БД с помощью различных методов доступа – это такое требование к БД, как:

- А) многократное использование
- Б) простота
- В) легкость использования
- Г) *гибкость использования*

5. Возможность автоматического восстановления данных такое требование к БД, как:

- А) автоматическая реорганизация
- Б) целостность данных
- В) *восстановление данных после сбоя*
- Г) сохранение затрат умственного труда

1	2	3	4	5
<i>В</i>	<i>А</i>	<i>Б</i>	<i>Г</i>	<i>В</i>

Типология баз данных

Классификация баз и банков данных может быть произведена по разным признакам (относящимся к разным компонентам и сторонам функционирования банков данных (БнД), среди которых выделяют, например, следующие.

По *форме представляемой информации* можно выделить фактографические, документальные, мультимедийные, в той или иной степени соответствующие цифровой, символьной и другим (нецифровой и несимвольной) формам представления информации в вычислительной среде. К последним можно отнести картографические, видео-, аудио-, графические и другие БД.

По *типу хранимой (не мультимедийной) информации* можно выделить фактографические, документальные, лексикографические БД. Лексикографические базы — это классификаторы, кодификаторы, словари основ слов, тезаурусы, рубрикаторы и т. д., которые обычно используются в качестве справочных совместно с документальными или фактографическими БД. Документальные базы подразделяются по уровню представления информации на полнотекстовые (так называемые «первичные» документы) и библиографическо-реферативные («вторичные» документы, отражающие на адресном и содержательном уровнях первичный документ).

По *типу используемой модели данных* выделяют три классических класса БД: иерархические, сетевые, реляционные. Развитие технологий обработки данных привело к появлению постреляционных, объектноориентированных, многомерных БД, которые в той или иной степени соответствуют трем упомянутым классическим моделям.

По *топологии хранения* данных различают локальные и распределенные БД.

По *типологии доступа и характеру использования* хранимой информации БД могут быть разделены на специализированные и интегрированные³.

По *функциональному назначению* (характеру решаемых с помощью БД задач и, соответственно, характеру использования данных) можно выделить операционные и справочно-информационные. К последним можно отнести ретроспективные БД (электронные каталоги библиотек, БД статистической информации и т. д.), которые используются для информационной поддержки основной деятельности и не предполагают внесения изменений в уже существующие записи, например, по результатам этой деятельности. Операционные БД предназначены для управления различными технологическими процессами. В этом случае данные не только извлекаются из БД, но и изменяются (добавляются) в том числе в результате этого использования.

По *сфере возможного применения* можно различать универсальные и специализированные (или проблемно-ориентированные) системы.

По *степени доступности* можно выделить общедоступные и БД с ограниченным доступом пользователей. В последнем случае говорят об управляемом доступе, индивидуально определяющем не только набор доступных данных, но и характер операций, которые доступны пользователю.

Следует отметить, что представленная классификация не является полной и исчерпывающей. Она в большей степени отражает исторически сложившееся состояние дел в сфере деятельности, связанной с разработкой и применением баз данных.

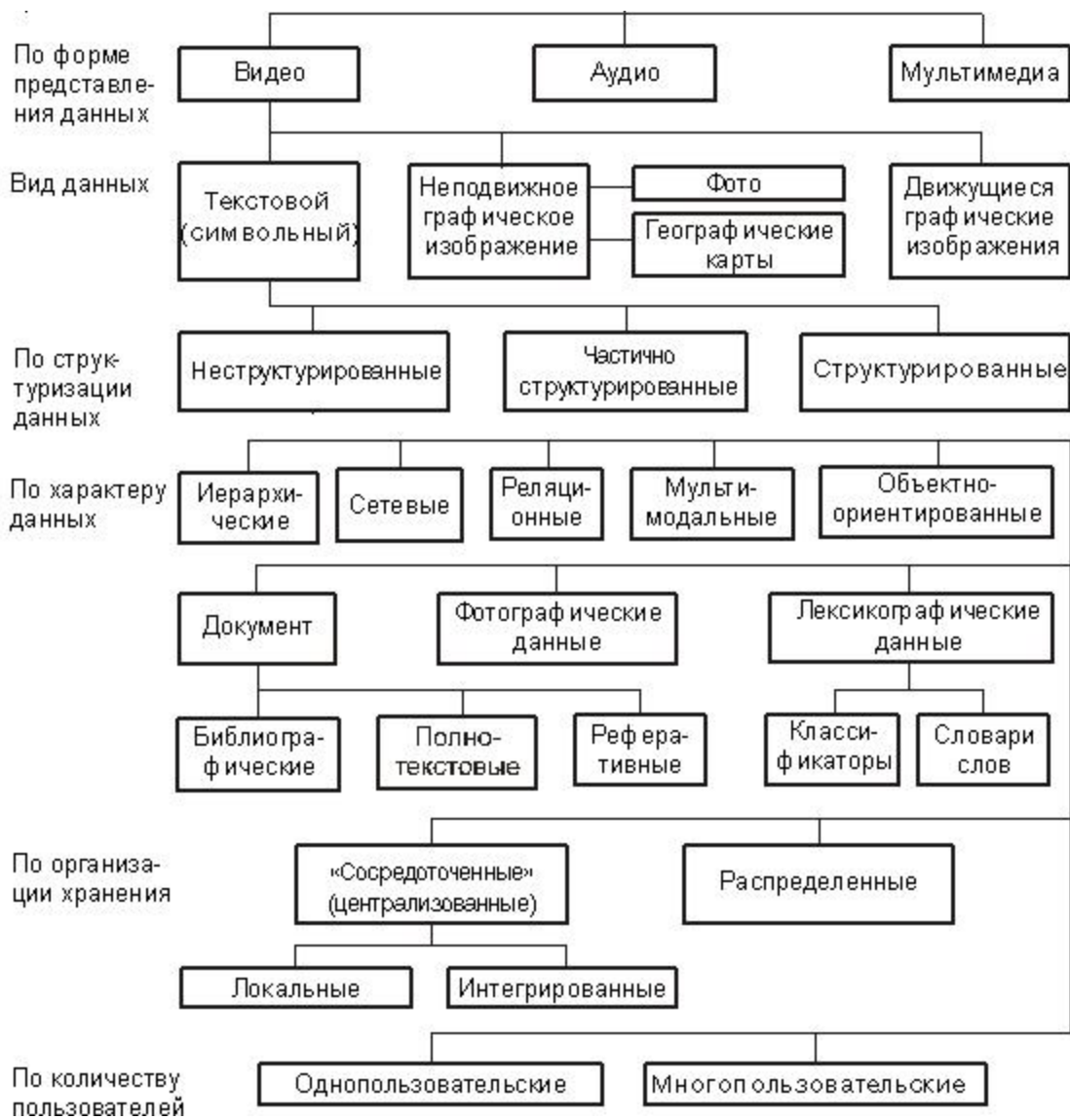


Рис. 1. Классификация баз данных

Типология моделей

Основные отличия любых методов представления информации заключаются в том, каким способом фиксируется семантика предметной области. Однако следует особо отметить, что для всех уровней и для любого метода представления предметной области (нам важен контекст создания и использования машинных баз данных) в основе отображения (т.е. собственно формирования представления) лежит *кодирование* понятий и отношений между понятиями. Многоуровневая система моделей представления информации иллюстрируется рис. 1.7. Рассмотрим далее основные из них.

Ключевым этапом при разработке любой информационной системы является проведение системного анализа, формализация предметной области и представление системы как совокупности компонент. Системный анализ позволяет, с одной стороны, лучше понять «что надо делать» и «кому надо делать» (аналитику, разработчику, руководителю, пользователю), а с другой — отслеживать во времени изменения рассматриваемой модели и обновлять проект.

Декомпозиция как основа системного анализа может быть функциональной (построение иерархий функций) или объектной.

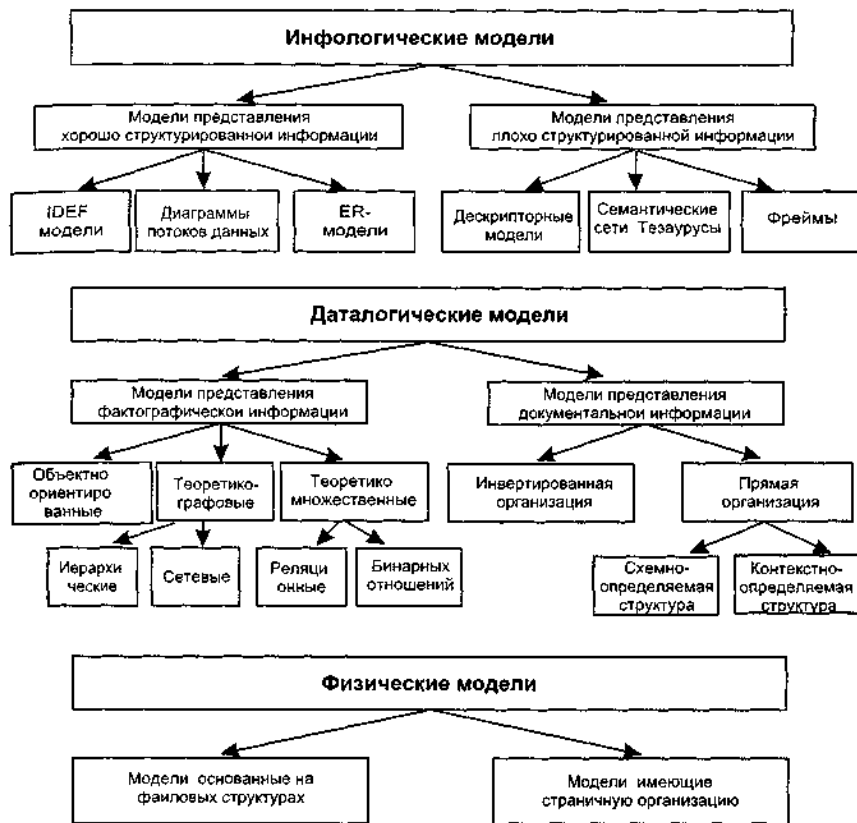


Рис 17 Система моделей представления информации

Однако в большинстве систем, если говорить, например, о базах данных, типы данных являются более статичным элементом, чем способы их обработки. Поэтому получили интенсивное развитие такие методы системного анализа, как диаграммы потоков данных (Data Flow Diagram). Развитие реляционных баз данных в свою очередь стимулировало развитие методик построения моделей данных, и в частности, ER-диаграмм (Entity Relationship Diagram). Однако и Функциональная декомпозиция и диаграммы потоков данных дают только некоторый срез исследуемой предметной области, но не позволяют получить представление системы в целом.

Различаются и методы отображения, используемые на этапе построения даталогических моделей, отражающих способ идентификации элементов и связей, но, что особенно важно — в контексте их будущего представления в одномерном пространстве памяти вычислительной машины. Модели подразделяются на фактографические — ориентированные на представление хорошо структурированной информации, и документальные — представляющие наиболее распространенный способ отражения слабоструктурированной информации. Если в первом случае говорят о реляционной, иерархической или сетевой моделях данных, то во втором — о семантических сетях и документальных моделях.

При проектировании информационных систем свойства объектов (их характеристики) называются атрибутами. Именно значения атрибутов позволяют выделить как в предметной области различные объекты (типы объектов), так и среди объектов одного типа — их различные экземпляры. Представление атрибутов удобнее всего моделируется теоретико-множественными отношениями. Отношение наглядно представляется как таблица, где каждая строка — кортеж отношения, а каждый столбец (домен) представляет множество значений атрибута. Список имен атрибутов отношения образует схему отношения, а совокупность схем отношений, используемых для представления БД, в свою очередь образует схему базы данных.

Представление схем БД в виде схем отношений упрощает процедуру проектирования БД. Этим объясняется создание систем, в которых проектирование БД ведется в терминах реляционной модели данных, а работа с БД поддерживается СУБД одного из упомянутых ранее типов.

Основное отличие методов представления информации заключается в том, каким способом фиксируется семантика предметной области. Первые, фактографические БД, задают

четкую схему соответствия, в рамках которой и отображается предметная область. Подобное построение по сути своей является довольно статичным, требует априорного знания типов отношений. В нем достаточно сложно вводить информацию о новых типах отношений между объектами, но с другой стороны, зафиксированная схема базы данных позволяет довольно эффективно организовать поиск информации.

Во втором случае предметная среда отображается (по крайней мере, на уровне модели) в виде однородной сети, любые изменения которой, как по вводу новых классов объектов, так и новых типов отношений, не связаны с какими-либо структурными преобразованиями сети. В силу большого количества типов отношений манипулирование подобной «элементарной» информацией достаточно затруднено, поэтому для данного случая характерно введение большого количества более общих понятий (и соответствующих им отношений), что упрощает работу с сетью.

Модель данных должна, так или иначе, дать основу для описания данных и манипулирования ими, а также дать средства анализа и синтеза структур данных. Любая модель, построенная более или менее аккуратно с точки зрения математики, сама создает объекты для исследования и начинает жить как бы параллельно с практикой.

Реляционная модель данных в качестве основы отображения непосредственно использует понятие отношения. Она ближе всего находится к так называемой концептуальной модели предметной среды и часто лежит в основе последней.

В отличие от теоретико-графовых моделей в реляционной модели связи между отношениями реализуются неявным образом, для чего используются *ключи отношений*. Например, отношения иерархического типа реализуются механизмом первичных / внешних ключей, когда в подчиненном отношении должен присутствовать набор атрибутов, связывающих это отношение с основным. Такой набор атрибутов в основном отношении будет называться первичным ключом, а в подчиненном — вторичным.

Прогресс в области разработки языков программирования, связанный в первую очередь с типизацией данных и появлением объектно-ориентированных языков, позволил подойти к анализу сложных систем с точки зрения иерархических представлений — с помощью классов объектов со свойствами инкапсуляции, наследования и полиморфизма, схемы которых отображают не только данные и их взаимосвязи, но и методы обработки данных.

В этом смысле объектно-ориентированный подход является гибридным методом и позволяет получить более естественную формализацию системы в целом. В итоге это позволяет снизить существующий барьер между аналитиками и разработчиками (проектировщиками и программистами), повысить надежность системы и упростить сопровождение, в частности, интеграцию с другими системами. Модель будет *структурно объектно-ориентированной*, если она поддерживает сложные объекты, модель будет *поведенчески объектно-ориентированной*, если она обеспечивает процедурную расширяемость, для того чтобы модель была полностью объектно-ориентированной, она должна обладать обоими свойствами.

В заключение отметим, что представленная здесь типология моделей не претендует на полноту, и она не является классификацией в точном смысле этого слова. Она скорее иллюстрирует эклектичность преобладающих в разное время взглядов, методов и решений, используемых при проектировании и реализации баз данных.

Контрольные вопросы и задания

1. Дайте определение понятий «база данных» и «банк данных».
2. Каковы предпосылки создания баз и банков данных?
3. Перечислите преимущества и недостатки использования банков данных.
4. Определите соотношение понятий «информация» и «данные».
5. Перечислите и определите назначение основных компонентов банков данных.
6. Определите основные функции и назначение СУБД.
7. Назовите отличительные особенности банка данных.
8. Перечислите основные требования, предъявляемые к банкам данных.
9. Какие технические средства используются для создания баз данных?
10. Перечислите основные признаки классификации банков данных.
11. Определите понятие и назначение лингвистических средств банков данных.
12. Перечислите основные категории пользователей баз данных.

13. Перечислите основные функции администратора базы данных
14. Укажите взаимосвязь этапов создания базы данных и используемых моделей предметной области
15. В чем различие между структурированной и слабоструктурированной информацией?
16. Приведите классификационную схему моделей баз данных

План-конспект занятия №3

Тема: Этапы проектирования баз данных

Цель занятия: познакомить особенностями проектирования БД

Вид занятия: теоретическое занятие

Задачи:

Учебные: 1) разъяснить основные особенностями проектирования БД

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

1. воспитание аккуратности и внимательности

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта (ОК 2.1.1),

Компоненты банка данных

Определение банка данных предполагает, что с функционально-организационной точки зрения банк данных является сложной человеко-машинной системой, включающей в себя все подсистемы, необходимые для надежного, эффективного и продолжительного во времени функционирования.

В структуре банка данных выделяют следующие компоненты (подсистемы):

- информационная база;
- лингвистические средства;
- программные средства;
- технические средства;
- организационно-административные подсистемы и нормативно-методическое обеспечение.



Информационная база

Данные, отражающие состояние определенной предметной области и используемые информационной системой, принято называть *информационной базой*. Информационная база состоит из двух компонент: 1) коллекции записей собственно данных; 2) описания этих данных — метаданных.

Данные отделены от описаний, но в то же время данные не могут использоваться без обращения к соответствующим описаниям.

Уже из определения базы данных и приведенных ранее основных требований следует, что данные могут использоваться (т. е. представляться) по-разному. С одной стороны, разные прикладные задачи требуют разных наборов данных, в совокупности обеспечивающих функциональную полноту информации, а с другой — они должны быть различны для различных

категорий субъектов (разработчиков или пользователей). Также должны быть различными и способы описания самих данных, их природы, формы хранения, условий взаимной непротиворечивости.

В литературе по базам данных упоминаются три уровня представления данных — концептуальный, внутренний и внешний (рис. 1.2).

Эти уровни представлений введены исходя из различного рассмотрения БД. Например, прикладному программисту требуются не все данные БД, а только некоторая их часть, используемая в его программе. Внешний уровень представления обеспечивает именно эту форму обмена данными.

Внутренний уровень — глобальное представление БД, определяет необходимые условия для организации хранения данных на внешних запоминающих устройствах

Описание БД на концептуальном уровне представляет собой обобщенный взгляд на данные с позиций предметной области (разработчика приложений, пользователя или внешней информационной системы).

Внешний уровень представления данных не затрагивает физической организации (размещения) данных во внешней памяти, поэтому его называют иногда логическим уровнем. Соответственно внутренний уровень называют физическим уровнем.

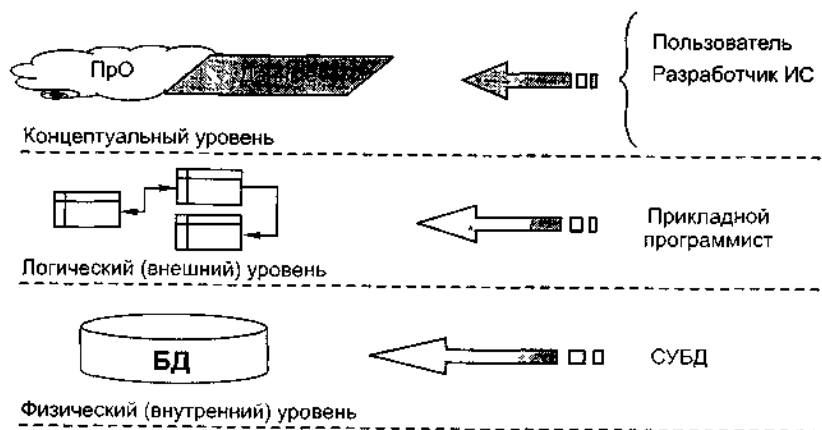


Рис 1.2 Уровни представления данных

Лингвистические средства

Многоуровневое представление БД предполагает соответствующие описания данных на каждом уровне и согласование одних и тех же данных на разных уровнях. С этой целью в состав СУБД включаются специальные языки для описания представлений внутреннего и внешнего уровней. Кроме того, СУБД должна включать в себя язык манипулирования данными (ЯМД). Желательно также наличие тех или иных дополнительных сервисных средств, например средств генерации отчетов.

Работа с базами данных предполагает несколько этапов:

- описание БД;
- описание частей БД, необходимых для конкретных приложений (задач, групп задач);
- программирование задач или описание запросов в соответствии с правилами конкретного языка и использованием языковых конструкций для обращения к БД;
- загрузка БД и т. д.

Для выражения обобщенного взгляда на данные применяют *язык описания данных (ЯОД)* внутреннего уровня, включаемый в состав СУБД³. Описание представляет собой модель данных и их отношений, т. е. структур, из которых образуется БД.

ЯОД позволяет определять схемы базы данных, характеристики хранимых и виртуальных данных и параметры организации их хранения в памяти и может включать в себя средства поддержки целостности базы данных, ограничения доступа, секретности.

ЯМД обычно включает в себя средства запросов к базе данных и поддержания базы данных (добавление, удаление, обновление данных, создание и уничтожение БД, изменение определений БД, обеспечение запросов к справочнику БД).

Исторически первым типом структур данных, который был включен в языки программирования, была иерархическая структура. Некоторые ранние СУБД также предполагали использование в качестве основной модели иерархические структуры типа дерева. Основанием для такого выбора было удобство представления (моделирования) естественных иерархических структур данных, существующих, например, в организациях.

В ряде предметных областей структура данных имеет более сложный вид, в котором поддерживаются связи типа «многие к одному», и которые могут быть представлены ориентированным графом. Такие структуры называют сетевыми. Для управления БД сетевой структуры международной ассоциацией Кодасил была предложена обобщенная архитектура системы с ЯОД схемы (модели БД) и подсхемы (модели части БД для конкретного приложения), а также ЯМД для оперирования данными БД в прикладных программах.

В настоящее время разработаны десятки языков, основанных на реляционной исчислении, различие которых обусловлено особенностями математических теорий, положенных в основу их построения. Среди этих языков можно выделить базирующиеся на С-исчислении, предложенном Коддом, и исчислении, предложенном Пиротти.

С-исчисление базируется на классическом прикладном исчислении предикатов, исчисление представляет собой разновидность прикладного многотипного исчисления предикатов. Существенное различие между этими исчислениями, а следовательно и языками заключается в том, что в С-исчислении в качестве области изменения значений предметной переменной используется множество выборок (кортежей) отношения, а в /"-исчислении каждому типу переменных или констант соответствует определенный домен базы данных.

Функциональные характеристики языков отражают возможности описания данных, средств представления запроса, обновления, поддержки целостности и секретности, включения в языки программирования, управления форматом ответов, средств запроса к словарю данных БД и т. д.

Качественные характеристики языков запросов могут определяться такими свойствами, как полнота, селективная мощность, простота изучения и использования, степень процедурности и модульности, унифицированность, производительность и эффективность. Рассмотрим некоторые из этих понятий.

Селективная мощность языков запросов характеризует возможность выбора данных по разным критериям. Данное понятие плохо поддается формализации; можно сказать, что язык с большей селективной мощностью позволяет сформулировать большинство запросов так, что ответ на них содержит меньше ненужных данных. Языки, обладающие малой селективной мощностью, в общем случае уже требуют привлечения дополнительных средств для анализа ответов на запросы (например, оценки пользователя).

Простота изучения является во многом субъективной оценкой и может быть в некоторой мере охарактеризована степенью его близости к естественному языку, требуемым для его освоения временем и необходимым уровнем подготовки пользователя.

Высокий *уровень процедурности*, свойственный реляционным языкам, определяется присущими реляционной модели свойствами, в частности, полным отделением логической структуры данных от структур хранения и стратегий доступа. Снижение уровня процедурности увеличивает свободу в выборе способов реализации языка, что позволяет осуществить его реализацию более оптимальным способом. Однако необходимо отметить, что меньшая степень процедурности еще не означает автоматически меньшую сложность написания запросов. Некоторые сложные запросы можно более просто сформулировать в виде алгоритма поиска ответа, в то время как его формулировка в декларативном виде может оказаться достаточно трудной.

Модульность построения языка характеризует возможность существования нескольких уровней языка и зависит от специфических свойств математической теории, лежащей в его основе. Минимальный уровень языка, обычно легко понимаемый пользователем, бывает достаточным для формулирования большинства запросов, и лишь формулировка сложных запросов может потребовать использования всех выразительных средств языка, о существовании которых пользователи начального уровня могут и не знать. Языки, не обладающие модульностью, требуют от пользователя знания почти всего объема средств языка, что усложняет процесс их изучения.

Наиболее распространенным языком для работы с базами данных является SQL (Structured Query Language), в своих последних реализациях предоставляющий не только средства для спецификации и обработки запросов на выборку данных, но также и функции по созданию, обновлению, управлению доступом и т. д.

По существу SQL уже соединяет в себе язык описания данных и язык манипулирования данными. Он не является полноценным языком программирования, и в случае его использования для организации доступа к БД из прикладных программ, SQL-выражения встраиваются в конструкции базового языка.

Являясь внутренним языком баз данных, SQL естественно отражает особенности конкретной СУБД. Сегодня это единственный стандартизованный язык фактографических баз данных, достаточно мощный и в то же время простой для понимания и использования. Сочетание этих факторов вместе с поддержкой ведущих производителей, таких как IBM и Microsoft, привели не только к широкому распространению, но и совершенствованию. Сегодня, благодаря независимости от конкретных СУБД и межплатформенной переносимости, SQL стал языком распределенных баз данных и языком шлюзов, позволяющим совместно использовать СУБД разного типа

Программные средства

Обработка данных и управление этой обработкой в вычислительной среде, а также взаимодействие с операционной системой и прикладными программами осуществляется комплексом программных средств, взаимосвязь которых иллюстрируется рис. 1.3 В составе комплекса обычно выделяют следующие компоненты:

- *ядро*, обеспечивающее управление данными во внешней и оперативной памяти, а также протоколирование изменений;
- *процессор языка базы данных*, обеспечивающий обработку (трансляцию или компиляцию) и оптимизацию запросов на выборку и изменение данных;
- *подсистему (библиотеку) поддержки программных вызовов*, которая обслуживает прикладные программы управления данными, взаимодействующие с СУБД через средства пользовательского интерфейса;
- *сервисные программы* (системные и внешние утилиты), обеспечивающие настройку СУБД, восстановление после сбоев и ряд дополнительных возможностей обслуживания.

Большинство СУБД работают в среде операционной системы и тесно с ней связаны. Многопользовательские приложения, обработка распределенных запросов, защита данных требуют эффективно использовать ресурсы, управление которыми обычно является функцией ОС. Использование многопроцессорных систем и мультипоточных технологий обработки данных позволяет эффективно обслуживать параллельно выполняемые запросы, но требует координации использования ресурсов между ОС и СУБД Соответственно, управление доступом и обеспечение защиты также обычно интегрируются с соответствующими средствами операционной системы.

Именно централизованное управление данными обеспечивает:

- сокращение избыточности в хранимых данных;
- совместное использование хранимых данных;
- стандартизацию представления данных, упрощающую эксплуатацию БД;
- разграничение доступа к данным;
- целостность данных, обеспечиваемую процедурами, предотвращающими включение в БД неверных данных, и ее восстановление после отказов системы.

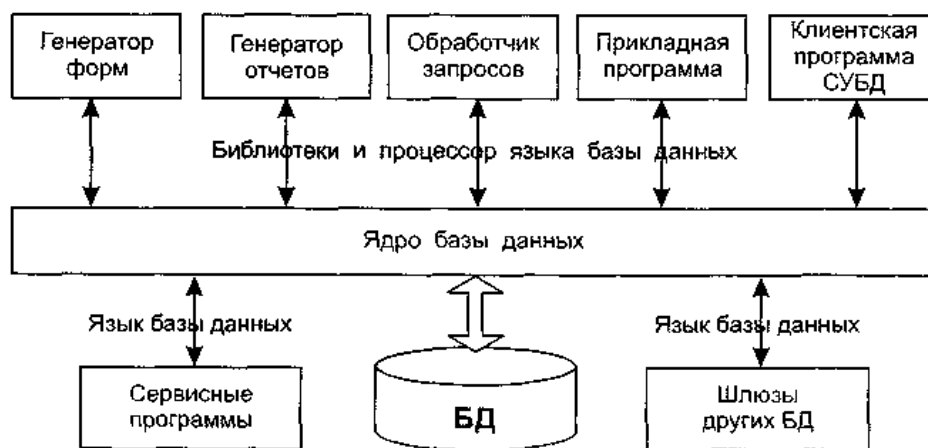


Рис 1.3 Программные средства СУБД

Технические средства

Сегодня большинство банков данных создается и функционирует на основе универсальных вычислительных машин⁴. Однако для больших баз данных, функционирующих в промышленном режиме, обеспечение эффективной и бесперебойной работы должно основываться на использовании адекватных аппаратных средств.

Устройства ввода-вывода и накопители внешней памяти — традиционно «узкое место» любой базы данных. Объем и быстродействие накопителей являются, очевидно, важными параметрами. Однако столь же значима и отказоустойчивость. Здесь следует отметить необходимость согласованных решений при распределении ролей между аппаратными и программными компонентами управления операциями ввода-вывода. Например, наличие буферной памяти в накопителе, ускоряющей ввод-вывод (аппаратное кэширование) при сбоях системы во время выполнения операции записи в БД может привести к потере данных, переданные для записи данные еще будут находиться в буфере, а так как СУБД отметит операцию записи как уже завершившуюся, откат для восстановления данных станет невозможен.

Для повышения надежности хранения часто используют специализированные дисковые подсистемы — RAID (Redundant Array of Inexpensive Disk). Один логический RAID-диск — это несколько физических дисков, объединенных в одно устройство, управляемое специализированным контроллером, что позволяет распределять основные и системные данные между несколькими носителями (дисками), в том числе дублировать данные. Таким образом, в случае повреждения одного из дисков, можно оперативно восстановить потерянные данные.

Не менее значима роль центрального процессора. Многие промышленные СУБД поддерживают многопроцессорную обработку запросов. Теоретически использование еще одного процессора позволит ускорить обработку. Однако на практике многопроцессорные системы требуют повышенного внимания при приобретении оборудования: надежно работают только сертифицированные системы, использующие соответствующие периферийные устройства.

Для распределенных и удаленно используемых баз данных также важно сетевое окружение" связанное оборудование и сетевые протоколы. Здесь важны не только показатели быстродействия, но и поддерживаемые ими возможности обеспечения безопасности.

Организационно-административные подсистемы

Организационно-методические средства не являются технической компонентой системы, однако трудно рассчитывать на устойчивое и долговременное функционирование банка данных, если будут отсутствовать необходимые методические и инструктивные материалы, регламентирующие работу пользователей, различных по своему статусу и уровню подготовленности.

Блиц- опрос №2 (проверка ДЗ)

1. Метаданные – это:

- А) данные, хранящиеся в БД
- Б) данные о данных
- В) данные о процедурах
- Г) данные о СУБД

2. Какой уровень представления данных используется программистом:

- А) концептуальный ~~с различных точек зрения~~
- Б) внутренний ~~для ВЗУ~~
- В) внешний ~~метаданные~~

3. Для какого уровня представления данных используется ЯОД (язык описания данных):

- А) концептуальный ~~с различных точек зрения~~
- Б) внутренний ~~для ВЗУ~~
- В) внешний ~~метаданные~~

4. Какое из лингвистических средств БД служит для организации обработки данных:

- А) ЯОД (язык описания данных)
- Б) ЯМД (язык манипулирования данными)
- В) SQL (Structured Query Language)

5. Какая из качественных характеристик языков запросов позволяет программировать, не зная этот язык полностью:

- А) селективная мощность языка
- Б) простота изучения
- В) уровень процедурности
- Г) модульность построения языка

1	2	3	4	5
Б	В	Б	Б	Г

Основные этапы разработки БД

Этап 1. Уточнение задач

На первом этапе составляется список всех основных задач, которые в принципе должны решаться этим приложением, - включая и те, которые не нужны сегодня, но могут появиться в будущем. Под "основными" задачами понимаются функции, которые должны быть представлены в формах или отчетах приложения.

Этап 2. Последовательность выполнения задач

Для того, чтобы приложение работало логично и удобно, лучше всего объединить основные задачи в тематические группы и затем упорядочить задачи каждой группы так, чтобы они располагались в порядке их выполнения. Может получиться так, что некоторые задачи будут связаны с разными группами или, что выполнение некоторой задачи должно предшествовать выполнению другой, принадлежащей к иной группе.

Этап 3. Анализ данных

После формирования списка задач, наиболее важным этапом является составление подробного перечня всех данных, необходимых для решения каждой задачи. Некоторые данные понадобятся в качестве исходных и меняться не будут. Другие данные будут проверяться и изменяться в ходе выполнения задачи. Некоторые элементы данных могут быть удалены или

добавлены. И наконец, некоторые данные будут получены с помощью вычислений: их вывод будет частью задачи, но в базу данных вноситься они не будут.

Этап 4. Определение структуры данных

После предварительного анализа всех необходимых элементов данных нужно упорядочить их по объектам и соотнести объекты с таблицами и запросами базы данных. Для реляционных баз данных типа Access используется процесс, называемый нормализацией, в результате которого вырабатывается наиболее эффективный и гибкий способ хранения данных.

Этап 5. Разработка макета приложения и пользовательского интерфейса

После задания структуры таблиц приложения, в Microsoft Access легко создать его макет с помощью форм и связать их между собой, используя несложные макросы или процедуры обработки событий. Предварительный рабочий макет легко продемонстрировать заказчику и получить его одобрение еще до детальной реализации задач приложения.

Этап 6. Создание приложения

В случае очень простых задач созданный макет является практически законченным приложением. Однако довольно часто приходится писать процедуры, позволяющие полностью автоматизировать решение всех намеченных в проекте задач. Поэтому, понадобится создать специальные связующие формы, которые обеспечивают переход от одной задачи к другой.

Этап 7. Тестирование и усовершенствование

После завершения работ по отдельным компонентам приложения необходимо проверить функционирование приложения в каждом из возможных режимов. Необходимо проверить работу макросов, для этого используя пошаговый режим отладки, при котором будет выполняться одна конкретная макрокоманда. При использовании Visual Basic для приложений в вашем распоряжении имеются разнообразные средства отладки, позволяющие проверить работу приложения, выявить и исправить ошибки.

По мере разработки автономных разделов приложения желательно передать их заказчику для проверки их функционирования и получения мнения о необходимости внесения тех или иных изменений. После того как заказчик ознакомится с работой приложения, у него практически всегда возникают дополнительные предложения по усовершенствованию, какой бы тщательной не была предварительная проработка проекта. Пользователи часто обнаруживают, что некоторые моменты, о которых в процессе постановки задач, они говорили как об очень важных и необходимых, на самом деле не играют существенной роли при практическом использовании приложения. Выявление необходимых изменений на ранних стадиях разработки приложения позволяет существенно сократить время на последующие переделки.

Реляционный подход к построению инфологической модели

Цель лекции: Ознакомиться с достоинствами и недостатками реляционного подхода к построению инфологической модели. Понять, как влияет нормализация отношений на работу базы данных. Понять различия между простым и составным ключом.

Реляционная модель данных

Реляционная модель есть представление БД в виде совокупности упорядоченных нормализованных отношений.

Для реляционных отношений характерны следующие особенности.

1. Любой тип записи содержит только простые (по структуре) элементы данных.
2. Порядок кортежей в таблице несуществен.
3. Упорядочение значащих атрибутов в кортеже должно соответствовать упорядочению атрибутов в реляционном отношении.
4. Любое отношение должно содержать один атрибут или более, которые вместе составляют уникальный первичный ключ.
5. Если между двумя реляционными отношениями существует зависимость, то одно отношение является исходным, второе - подчиненным.
6. Чтобы между двумя реляционными отношениями существовала зависимость, атрибут, служащий первичным ключом в исходном отношении, должны также присутствовать в подчиненном отношении.

Пример 1. Представим БД "Учебный процесс" в виде реляционной модели ([таблица 1](#)).

Таблица 1.

а) Отношение ГРУППА

Индекс ИГ	Название группы ИГ	Количество ответов КОЛ	Проходной балл ПБАЛЛ
1	A1	16	4,3
2	A2	28	4,0
3	A3	18	4,3

б) Отношение СТУДЕНТ

Номер зачетной книжки НЗ	ИГ	Фамилия И.О. СФИО	Год рождения ГР
--------------------------	----	-------------------	-----------------

Понятие реляционный (англ. *relation* - отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы - один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Пример 2. Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе ([таблица 2](#)).

Таблица 2. Пример реляционной таблицы

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	ИСТ 11
16593	Петрова	Анна	Владимировна	15.03.75	СК 12
16693	Анохин	Андрей	Борисович	14.04.76	ИСТ 11

Понятие информационного объекта

Информационный объект - это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных *реквизитов* (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, материал, вуз, студент, сдача экзаменов и т.д.

Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.

Информационный объект имеет множество реализации - экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется* значением ключа (простого - один реквизит или составного - несколько реквизитов). Остальные реквизиты информационного объекта являются описательными. При этом одни и те же реквизиты в одних информационных объектах могут быть ключевыми, а в других - описательными. Информационный объект может иметь несколько ключей.

Пример 3 В [таблице 2](#) представлен пример структуры и экземпляров информационного объекта Студент.

В информационном объекте Студент ключом является реквизит **Номер** (№ личного дела), к описательным реквизитам относятся: *Фамилия* (Фамилия студента), *Имя* (Имя студента), *Отчество* (Отчество студента), *Дата* (Дата рождения), *Группа* (№ группы). Если отсутствует реквизит **Номер**, то для однозначного определения характеристик конкретного студента необходимо использование составного ключа из трех реквизитов: *Фамилия + Имя + Отчество*.

Таблица 2. Пример структуры и экземпляров информационного объекта

Структура	Номер	Фамилия	Имя	Отчество	Дата	Группа
	16493	Сергеев	Петр	Михайлович	01.01.96	ИСТ 11
Экземпляры инф.объекта Студент	16593	Петрова	Анна	Викторович	15.03.95	СК 12
	16693	Анохин	Роман	Борисович	14.04.96	ИСТ 11

Пример 4 На [рис. 1](#) изображен пример компактного представления информационного объекта Студент с обозначением имени объекта, ключа и указанием максимально возможного числа экземпляров записи.



Рис. 1. Пример компактного представления информационного объекта

Пример 5 Пример представления информационного объекта Студент в виде графа на [рис.](#)

2.

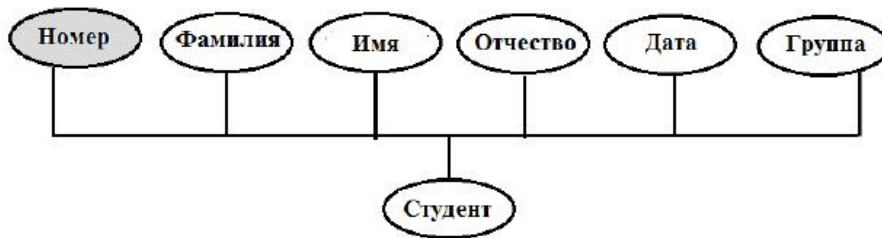


Рис. 2. Пример представления информационного объекта в виде графа

НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

Понятие нормализации отношений

Одни и те же данные могут группироваться в таблицы (отношения) различными способами, т.е. возможна организация различных наборов отношений взаимосвязанных информационных объектов. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование данных и упрощающей процедуры их обработки и обновления.

Определенный набор отношений обладает лучшими свойствами при включении, модификации, удалении данных, чем все остальные возможные наборы отношений, если он отвечает требованиям нормализации отношений.

Нормализация отношений - формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

Е. Коддом выделены три **нормальные формы отношений** и предложен механизм, позволяющий любое отношение преобразовать к третьей (самой совершенной) нормальной форме.

Первая нормальная форма

Отношение называется нормализованным или приведенным к **первой нормальной форме**, если все его атрибуты простые (далее неделимы).

Преобразование отношения к первой нормальной форме может привести к увеличению количества реквизитов (полей) отношения и изменению ключа.

Например, отношение Студент = (*Номер*, *Фамилия*, *Имя*, *Отчество*, *Дата*, *Группа*) находится в первой нормальной форме.

Вторая нормальная форма

Чтобы рассмотреть вопрос приведения отношений ко второй нормальной форме, необходимо дать пояснения к таким понятиям, как функциональная зависимость и полная функциональная зависимость.

Описательные реквизиты информационного объекта логически связаны с общим для них ключом, эта связь носит характер **функциональной зависимости** реквизитов.

Функциональная зависимость реквизитов - зависимость, при которой в экземпляре информационного объекта определенному значению ключевого реквизита соответствует только одно значение описательного реквизита.

Такое определение функциональной зависимости позволяет при анализе всех взаимосвязей реквизитов предметной области выделить самостоятельные информационные объекты.

Пример 6 Пример графического изображения функциональных зависимостей реквизитов Студент показан на [рис. 3](#), на котором ключевой реквизит указан *.



Рис. 3. Графическое изображение функциональной зависимости реквизитов

В случае составного ключа вводится понятие **функционально полной** зависимости.

Функционально полная зависимость неключевых атрибутов заключается в том, что каждый неключевой атрибут функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Таблица находится во **второй нормальной форме**, если она удовлетворяет требованиям первой нормальной формы и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом, то есть любое не ключевое поле однозначно идентифицируется полным набором ключевых полей.

Итак, таблица, находящаяся во второй нормальной форме, должна удовлетворять следующим правилам:

- таблица должна содержать данные об одном типе объектов;
- каждая таблица должна содержать одно поле или несколько полей, образующих уникальный идентификатор (или первичный ключ) для каждой строки;
- все поля, не имеющие ключа, должны определяться полным уникальным идентификатором данной таблицы.

Пример 7

Отношение Студент = (Номер, Фамилия, Имя, Отчество, Дата, Группа)	=	находится в первой и во второй нормальной форме одновременно, так как описательные реквизиты однозначно определены и функционально зависят от ключа Номер .
Отношение Успеваемость = (Номер, Фамилия, Имя, Отчество, Дисциплина, оценка)	=	находится в первой нормальной форме и имеет составной ключ Номер + Дисциплина . Это отношение не находится во второй нормальной форме, так как атрибуты Фамилия, Имя, Отчество не находятся в полной функциональной зависимости с составным ключом отношения.

Третья нормальная форма

Понятие третьей нормальной формы основывается на понятии **нетранзитивной** зависимости.

Транзитивная зависимость наблюдается в том случае, если один из двух описательных реквизитов зависит от ключа, а другой описательный реквизит зависит от первого описательного реквизита.

Таблица находится в **третьей нормальной форме**, если она удовлетворяет определению второй нормальной формы и ни одно из ее неключевых полей функционально не зависит от любого другого неключевого поля. Можно сказать, что таблица находится в третьей нормальной форме, если она находится во второй нормальной форме и каждое неключевое поле нетранзитивно зависит от первичного ключа.

Требование третьей нормальной формы сводится к тому, чтобы все неключевые поля зависели только от первичного ключа и не зависели друг от друга. Другими словами, нужно иметь возможность изменять значение любого неключевого поля, не изменяя значения любого другого поля базы данных. Это требование исключает любое поле, значения в котором получаются как результат вычислений, использующих значения других полей.

Пример 8 Если в состав описательных реквизитов информационного объекта **Студент** включить *фамилию старосты группы (Староста)*, которая определяется только номером группы, то одна и та же фамилия старосты будет многократно повторяться в разных экземплярах данного информационного объекта. В этом случае наблюдаются *затруднения в корректировке фамилии старосты* в случае назначения нового старосты, а также *неоправданный расход памяти* для хранения дублированной информации.

Для устранения транзитивной зависимости описательных реквизитов необходимо провести "**расщепление**" исходного информационного объекта. В результате расщепления часть реквизитов удаляется из исходного информационного объекта и включается в состав других (возможно, вновь созданных) информационных объектов.

Пример 9 "Расщепление" информационного объекта, содержащего транзитивную зависимость описательных реквизитов, показано на [рис. 4](#). Как видно из [рис. 4](#), исходный информационный объект Студент группы представляется в виде совокупности правильно структурированных информационных объектов (**Студент и Группа**), реквизитный состав которых тождественен исходному объекту.

Отношение Студент = (Номер, Фамилия, Имя, Отчество, Дата, Группа) находится одновременно в первой, второй и третьей нормальной форме.

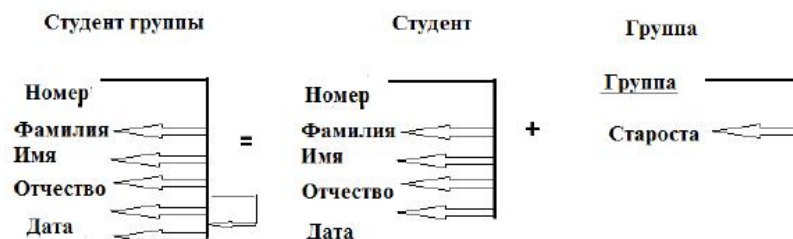


Рис.4. "Расщепление" информационного объекта, содержащего транзитивную зависимость описательных реквизитов

Краткие итоги

Реляционная модель есть представление БД в виде совокупности упорядоченных нормализованных отношений.

Нормализация отношений - формальный аппарат ограничений на формирование отношений. Выделены три нормальные формы отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей нормальной форме.

Реляционные базы данных состоят из нескольких таблиц, связь между которыми устанавливается с помощью совпадающих полей. Каждая запись в таблицах идентифицирует один объект. Отношение между объектами определяет отношение между таблицами. Существует 4 типа отношений:

- Отношение "один-к-одному" (1:1) означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.
- Отношение "один-ко-многим" (1 : M) означает, что каждой записи в одной таблице соответствует одна или несколько записей в другой таблице.
- Отношение "многие-к-одному" (M:1) аналогично рассмотренному ранее типу. Тип отношения между объектами зависит от вашей точки зрения.
- Отношение "многие-ко-многим" (M:M) возникает между двумя таблицами в тех случаях, когда:

Простой или первичный ключ состоит из единственного поля таблицы, значения которого уникальны для каждой записи. Если таблица не имеет единственного уникального поля, первичный ключ составляется из нескольких полей, совокупность значений которых гарантирует уникальность, такой первичный ключ называют составным ключом.

Вопросы для самопроверки

- Назовите характерные особенности реляционных отношений.
- На что ориентирована реляционная модель?
- Какими свойствами обладает реляционная таблица?
- Понятие информационного объекта.
- Понятие нормализации отношений.
- Первая нормальная форма.
- Вторая нормальная форма.
- Третья нормальная форма.

План-конспект занятия №4

Тема: Методы создания новой БД и таблиц БД

Тип занятия: теоретическое занятие.

Цель занятия: ознакомить с методами создания новой БД и таблиц БД

Учебные: 1) объяснить назначение и применение различных методов

2) проверить усвоение материала

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание самостоятельности при решении задач

Развиваемые общие компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.1., ПК 2.1.

Самостоятельная работа студента:

Составление конспекта (ОК 2.1.1)

Составление сводных таблиц (ОК 3.1.1)

Блиц- опрос №7 (проверка ДЗ)

А.В. Кузин. Нормализация таблиц (с.23-27)

1. Наложение ограничений на формирование таблиц с целью устранения дублирования данных и соблюдения их непротиворечивости называется

- А) декомпозиция таблиц
- Б) композиция таблиц
- В) нормализация таблиц
- Г) разложение таблиц

2. Нормализация таблиц НЕ предусматривает выполнения:

- А) устранения ошибочности данных
- Б) устранение дублирования данных
- В) устранение избыточности данных
- Г) непротиворечивости данных

3. Количество нормализованных таблиц в схеме данных равно:

- А) количеству атрибутов в исходной ненормализованной таблице
- Б) количеству известных нормальных форм
- В) количеству записей в исходных данных
- Г) зависит от способов декомпозиции

4. Известно, что таблица находится в третьей нормальной форме. Это означает, что она:

- А) уже находится в первой нормальной форме
- Б) уже находится во второй нормальной форме
- В) имеет три ключевых поля
- Г) связана с тремя таблицами в схеме данных

5. Для ускорения доступа к данным в БД используются методы:

- А) индексирования данных
- Б) композиции таблиц
- В) декомпозиции таблиц
- Г) хеширования данных

6. Какой из способов ускорения доступа к данным в БД требует значительных затрат памяти ПК:

- А) индексирование данных
- Б) композиция таблиц
- В) декомпозиция таблиц
- Г) хеширование данных

1	2	3	4	5	6
В	А	Г	А,Б	А,Г	А

Создание новой базы данных

При первом запуске программы Access или закрытии базы данных без выхода из Access отображается окно **Приступая к работе с Microsoft Office Access**.

Это отправной пункт, из которого можно создать новые базы данных, открыть существующую базу данных или ознакомиться с информацией на веб-узле Microsoft Office Online.

Создание базы данных с помощью шаблона

В приложении Access предусмотрены разнообразные шаблоны, с помощью которых можно быстро создать базу данных. Шаблон – это уже готовая к использованию база данных, включающая все необходимые таблицы, запросы, формы и отчеты для выполнения определенной задачи. Например, предусмотрены шаблоны, которые можно использовать для отслеживания вопросов, управления контактами или учета расходов. Некоторые шаблоны содержат несколько примеров записей, позволяющих продемонстрировать их использование. Шаблоны баз данных можно использовать без изменений или настроить в соответствии с конкретными потребностями.

Если один из этих шаблонов точно соответствует потребностям, с его помощью обычно проще и быстрее всего создать необходимую базу данных. Однако если необходимо импортировать в Access данные из другой программы, возможно, будет проще создать базу данных без использования шаблона. Так как в шаблонах уже определена структура данных, на изменение существующих данных в соответствии с этой структурой может потребоваться много времени.

Создание базы данных без использования шаблона

Если использовать шаблон не имеет смысла, можно создать базу данных с нуля. Для этого нужно создать таблицы, формы, отчеты и другие объекты базы данных. В большинстве случаев необходимо выполнить одно или оба следующих действия:

- Ввод, вставка или импорт данных в таблицу, которая создана при создании новой базы данных, и последующее повторение этой процедуры для новых таблиц, которые создаются с помощью команды **Таблица**, расположенной на вкладке **Создать**.
- Импорт данных из других источников и создание новых таблиц в этом процессе.

Для получения дополнительных сведений о планировании и разработке базы данных или о создании отношений, форм, отчетов или запросов используйте ссылки **См. также** этого раздела.

Создание пустой базы данных

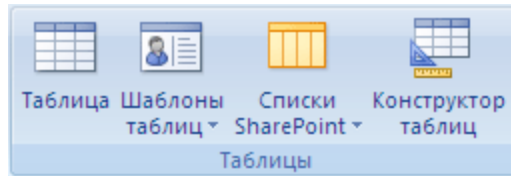
1. На странице **Приступая к работе с Microsoft Office Access** в разделе **Новая пустая база данных** выберите команду **Новая база данных**. Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. Курсор находится в первой пустой ячейке столбца **Добавить поле**.

4. Чтобы добавить данные, начните их вводить или вставьте данные из другого источника

Важно. Если хотя бы один раз закрыть таблицу «Таблица1» без сохранения, она будет удалена полностью, даже если в нее введены данные.

Добавление таблицы

Добавление таблиц к существующей базе данных осуществляется командами группы **Таблицы** на вкладке **Создать**.



Создание таблицы в режиме таблицы

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически. Полям присваиваются имена с последовательными номерами («Поле1», «Поле2» и т. д.). Приложение Access автоматически задает тип данных для каждого поля в зависимости от вводимых данных.

Создание таблицы в режиме конструктора

В режиме конструктора сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например вставку или импорт.

Создание таблицы на основе шаблона В приложении Access имеются шаблоны для часто используемых типов таблиц. Одним щелчком мыши можно создать полную, сконфигурированную и готовую к использованию структуру таблицы. Чтобы привести таблицу в соответствие со своими запросами пользователь может добавлять или удалять поля.

Задание свойств полей в режиме конструктора При создании таблицы в режиме таблицы или конструктора рекомендуется проверить и задать свойства полей. Это можно сделать только в режиме конструктора. Для перехода в режим конструктора в области переходов щелкните таблицу правой кнопкой мыши, а затем выберите команду **Режим конструктора**. Для отображения свойств полей щелкните поле на сетке конструктора. Свойства отображаются под сеткой конструктора в области **Свойства поля**.

Щелкните свойство поля для просмотра его описания в поле, расположенном рядом со списком свойств в области **Свойства поля**. Дополнительную информацию можно получить, нажав клавишу F1.

В следующей таблице описаны некоторые наиболее часто изменяемые свойства полей.

Свойство	Описание
Размер поля (Field Size)	Для текстовых полей это свойство указывает максимально допустимое количество знаков, сохраняемых в этом поле. Максимальное значение — 255. Для числовых полей это свойство указывает тип сохраняемых чисел («Длинное целое», «Двойное с плавающей точкой» и т. д.). Для более рационального хранения данных рекомендуется выделять для хранения данных наименьший необходимый размер памяти. Если потребуется, это значение позже можно изменить.
Формат (Format)	Это свойство определяет формат отображения данных. Оно не влияет на фактические данные, сохраняемые в этом поле. Можно выбрать встроенный формат или задать пользовательский формат.
Маска ввода (Input Mask)	Это свойство используется для определения общего шаблона для ввода любых данных в это поле. Это позволяет обеспечить правильный ввод и нужное количество знаков для всех данных. Для получения справки по созданию маски ввода нажмите кнопку <input type="button" value="..."/> справа от поля свойства.
Значение по умолчанию (Default Value)	Это свойство используется для задания значения по умолчанию, которое будет отображаться в этом поле при каждом добавлении новой записи. Например, для поля «Дата/время», в котором необходимо записывать дату добавления каждой записи, в качестве значения по умолчанию можно ввести «Date()» (без кавычек).
Обязательное	Это свойство указывает, обязательно ли вводить значение в это поле. Если

(Required) для этого свойства задано значение **Да**, невозможно будет добавить новую запись, если в это поле не введено значение.

Копирование данных из другого источника в таблицу Access

Если данные хранятся в другой программе, например Office Excel 2007, их можно скопировать и вставить в таблицу Access. Этот метод работает лучше всего, если данные уже разделены по столбцам, как это сделано в таблице Excel. Если это данные из текстового редактора, перед копированием рекомендуется разделить столбцы данных с помощью табуляции или преобразовать эти данные в таблицу в текстовом редакторе. Если необходимо изменить данные или выполнить другие операции (например, разделить полное имя на имя и фамилию), рекомендуется это сделать перед копированием данных, особенно при отсутствии опыта работы с Access.

При вставке данных в пустую таблицу приложение Access задает тип данных для каждого поля в зависимости от того, какие данные находятся в этом поле. Например, если во вставляемом поле содержатся только значения даты, для этого поля используется тип данных «Дата/время». Если вставляемое поле содержит только слова «Да» и «Нет», для этого поля выбирается тип данных «Логический».

В приложении Access поля называются в зависимости от информации, содержащейся в первой строке вставленных данных. Если в первой строке вставляемых данных находятся данные такого же типа, как и в последующих строках, то в приложении Microsoft Access эта строка используется в качестве строки данных, а полям присваиваются общие имена (F1, F2 и так далее). Если первая строка вставляемых данных отличается от следующих строк, ее содержимое используется только в качестве имен поля и не включается в данные.

Импорт, добавление или связывание данных из другого источника

В Access можно импортировать данные, которые сохранены в другой программе, и поместить эти данные в новую таблицу или добавить в существующую таблицу. В случае совместной работы с другими пользователями, которые применяют иные программы, их данные тоже можно использовать в Access. В любом случае в программе Access импорт данных из других источников не представляет труда. Импорт данных возможен из листа Excel, из таблицы в другой базе данных Access, из списка Windows SharePoint Services 3.0 или других источников.

Если импортируется таблица, программа Access импортирует данные в новую таблицу и отображает эту таблицу в группе **Таблицы** в области переходов. Если выбрано добавление данных к существующей таблице, данные добавляются к этой таблице. Если выбрано связывание данных, в группе **Таблицы** в области переходов создается связанная таблица.

Домашнее задание:

- Создание сравнительной таблицы «Методы создания новой БД: преимущества и недостатки»

<i>Метод создания новой базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание сравнительной таблицы «Методы создания таблицы в БД: преимущества и недостатки»

<i>Метод создания таблицы базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

План-конспект занятия №5

Тема: Типы данных в БД

Цель занятия: изучить типы и свойства данных в БД.

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить назначение и свойства типов данных
 2) разъяснить области применения различных типов данных;
- Развивающие:
 1) развитие логического мышления;
 2) развитие памяти
- Воспитательные:
 1) воспитание аккуратности и внимательности
- Развиваемые общие компетенции: ОК 1, ОК 10, ОК 2.1.1; ОК 3.1.1, ПК 2.1
- Самостоятельная работа студента: Составление конспекта (ОК 4.1.1), Работа с дополнительной литературой, Заполнение сводных таблиц (ОК 2.1.1)

Объекты Microsoft Access

Microsoft Access называет объектами все, что может иметь имя (в смысле Access). В базе данных Access основными объектами являются таблицы, запросы, формы, отчеты, макросы и модули. В других СУБД, как правило, термин база данных обычно относится только к файлам, в которых хранятся данные. В Microsoft Access база данных включает в себя все объекты, связанные с хранимыми данными, в том числе и те, которые определяются для автоматизации работы с ними. Ниже приведен список основных объектов базы данных Access.

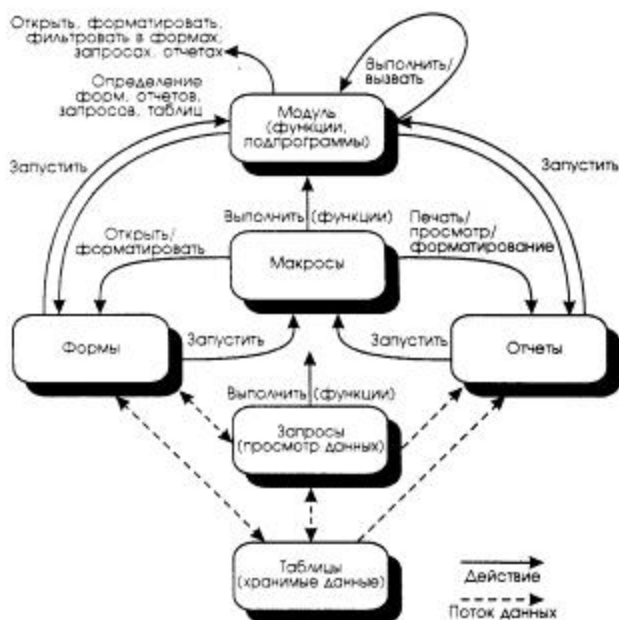


Рис. 1. Взаимосвязи основных объектов в Microsoft Access

1. Таблица. Объект, который определяется и используется для хранения данных. Каждая таблица включает информацию об объекте определенного типа, например о клиентах. Таблица содержит поля (столбцы), в которых хранятся различного рода данные, например фамилия или адрес клиента, и записи (которые называются также строками). В записи собрана вся информация о некотором объекте (человеке, образце продукции и т.п.). Для каждой таблицы можно определить первичный ключ (одно или несколько полей, содержащих уникальные для каждой записи значения) и один или несколько индексов, помогающих ускорить доступ к данным.

2. Запрос. Объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать бланк QBE (запрос по образцу) или инструкции SQL (структурированный язык запросов). Можно создать запросы на выборку, обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц.

3. Форма. Объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения. Формы используются для того, чтобы реализовать требования пользователя к представлению данных из запросов или таблиц. Формы можно также

распечатать. С помощью формы можно в ответ на некоторое событие, например изменение значения определенных данных, запустить макрос или процедуру VBA.

4. Отчет. Объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

5. Макрос. Объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в основной форме открывает другую форму. С помощью другого макроса можно осуществлять проверку значения некоторого поля при изменении его содержимого. В макрос можно включить дополнительные условия для выполнения или невыполнения тех или иных указанных в нем действий. Из одного макроса можно также запустить другой макрос или процедуру VBA.

6. Модуль. Объект, содержащий программы, написанные на языке Visual Basic для приложений. Модули могут быть независимыми объектами, содержащими функции, вызываемые из любого места приложения, но они могут быть и непосредственно "привязаны" к отдельным формам или отчетам для реакции на те или иные происходящие в них изменения.

7. Страницы доступа. Страницы - служат для обеспечения доступа к данным, содержащимся в базе, удалённой от потребителя (например, через Интернет).

Концептуальные взаимосвязи объектов Access показаны на [рис. 1](#).

Работа с таблицами

Создание таблицы в режиме конструктора

1. щёлкнуть по значку **Создание таблицы** в режиме конструктора. Откроется окно **Конструктора** ([рис. 2](#)).

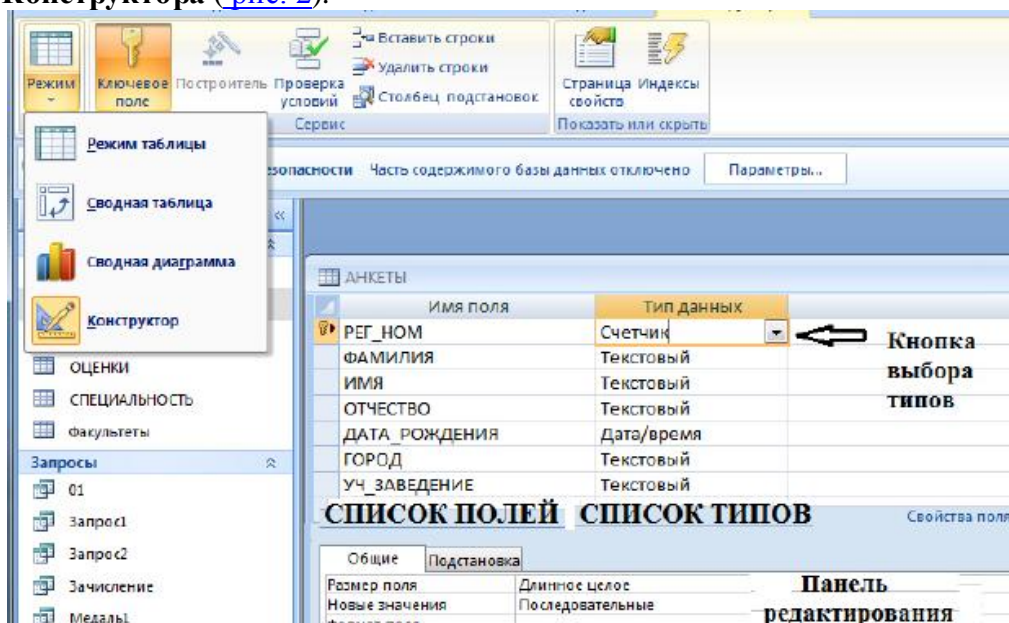


Рис. 2. Вид таблицы в режиме конструктора

2. Заполнить имена полей, (перемещаясь по ячейкам с помощью клавиш **Tab** или стрелками управления курсором);
3. Выбрать из раскрывающегося списка типы данных;
4. Задать ключевое поле:
 - щёлкнуть на его имени правой кнопкой мыши и
 - в контекстном меню выбрать пункт **Ключевое поле**.
5. Бланк закрывают, после чего дают таблице имя.

Созданную таблицу открывают двойным щелчком на её значке. Новая таблица имеет только названия столбцов.

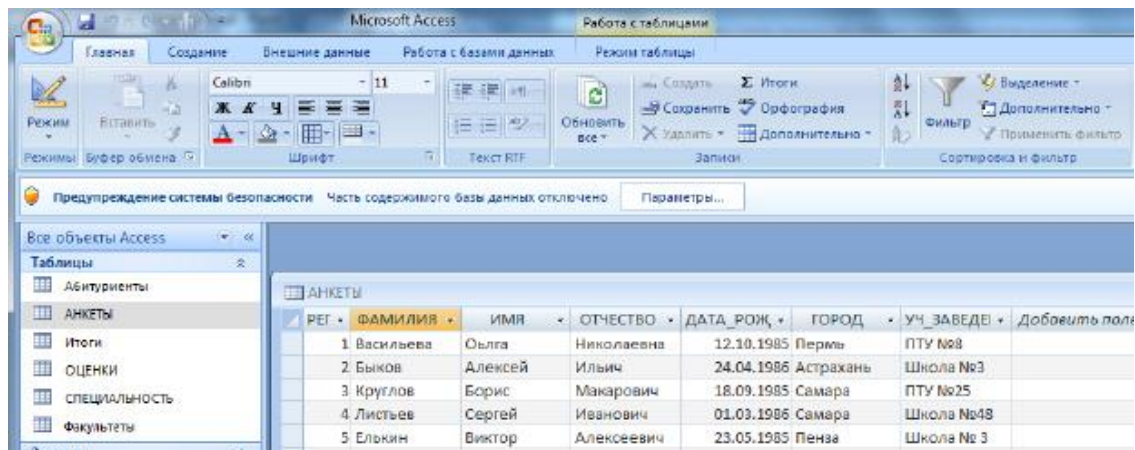


Рис. 3. Вид в режиме таблицы

При заполнении таблицы данными сохранение их происходит автоматически (рис. 3). Но если произошло изменение макета таблицы (ширина столбцов), то СУБД попросит подтверждение сохранения этих изменений.

Для изменения структуры **Таблицы** её надо открыть в режиме **Конструктора**
Введение в типы данных и свойства полей

Любая таблица состоит из полей. Свойства поля описывают характеристики и поведение данных, добавляемых в это поле. Тип данных поля — наиболее важное свойство, потому что оно определяет, какие данные могут храниться в этом поле. В этой статье описаны типы данных и другие свойства полей, доступные в Microsoft Access 2010, и приведены дополнительные сведения в разделе с подробным описанием типов данных.

Разобраться в использовании типов данных может быть непросто: например, поля текстового типа могут хранить как текстовые, так и цифровые данные, однако поле с типом данных "Числовой" может содержать только числовые данные. Таким образом, необходимо понимать, какие свойства используются с каждым типом данных. Тип данных поля определяет и другие важные его свойства, например:

- форматы, которые можно использовать в поле;
- максимальный размер значения в поле;
- варианты использования поля в выражениях;
- возможность индексации поля.

Тип данных поля может быть предопределен либо выбираться в зависимости от способа его создания. Например, при создании поля в режиме таблицы имеет место следующее:

- если используется существующее поле из другой таблицы, его тип данных уже определен в шаблоне или другой таблице;
- если данные вводятся в пустой столбец или поле, Access 2010 назначает полю тип данных на основе вводимого значения либо тип данных и формат для поля выбирает пользователь;
- если на вкладке **Изменение полей** в группе **Поля и столбцы** пользователь щелкает элемент **Добавление полей**, Access 2010 отображает список типов данных, в котором можно выбрать нужный вариант.

Применение тех или иных типов данных

Тип данных поля можно рассматривать как набор характеристик, применяемых ко всем содержащимся в поле значениям. Например, значения в текстовом поле могут содержать только буквы, цифры и ограниченный набор знаков препинания, а максимальное количество знаков в таком поле — 255.

Совет. Иногда фактический тип данных в поле может отличаться от кажущегося. Например, может показаться, что поле содержит числовые значения, хотя на самом деле они являются текстовыми, такими как номера комнат. Сравнить значения различных типов и преобразовывать их из одного типа в другой можно с помощью выражений.

В Access 2010 доступны типы данных **Основные типы**, **Числовой**, **Дата/время**, **Да/Нет** и **Быстрое начало**. В приведенных ниже таблицах указаны форматы, доступные для каждого типа, и описаны эффекты использования параметров форматирования.

Основные типы

Формат	Служит для отображения
Текст	Короткие буквенно-цифровые значения, например фамилия или адрес.
Числовой	Числовые значения, например расстояния. Имейте в виду, что для денежных значений есть отдельный тип.
Денежный	Денежные значения.
Да/Нет	Значения "Да" и "Нет", а также поля, содержащие только одно из двух возможных значений.
Дата/время	Значения даты и времени для годов с 100 по 9999.
Форматированный текст	Текст или сочетания текста и чисел, которые можно отформатировать с помощью элементов управления цветами и шрифтом.
Вычисляемое поле	Результаты вычисления. В вычислении должны использоваться поля в той же таблице. Для создания вычислений используется построитель выражений.
Вложение	Изображения, листы, документы, диаграммы и файлы других поддерживаемых типов, прикрепленные к записям в базе данных (аналогично файлам, вложенным в сообщения электронной почты).
Гиперссылка	Текст или сочетания текста и цифр, хранящиеся в виде текста и используемые в качестве адреса гиперссылки.
МЕМО	Длинные фрагменты текста. Поле типа МЕМО часто используется для хранения подробного описания продукта.
Подстановка	Список значений, возвращаемых запросом или из таблицы, либо набор значений, указанных при создании поля. Поле подстановки можно создать с помощью мастера подстановок. Такое поле обычно имеет текстовый или числовой тип данных в зависимости от того, какие параметры были выбраны в мастере. Примечание. Поля подстановки имеют дополнительный набор свойств, доступных на вкладке Подстановка в области Свойства поля .

Числовой тип

Формат	Служит для отображения
Общий	Числа без дополнительного форматирования (в том виде, в котором они хранятся).
Денежный	Денежные значения в общем формате.
Евро	Общие денежные значения в формате евро.
Фиксированный	Числовые данные.
Стандартный	Числовые данные с дробной частью.
Процентный	Процентные значения.
Экспоненциальный	Вычисления.

Дата и время

Формат	Служит для отображения
Краткий формат даты	Дата в кратком формате. Зависит от региональных настроек даты и времени (например, 3/14/2001 для США).
Средний формат даты	Дата в среднем формате (например, 3-Apr-09 для США).
Длинный формат даты	Дата в длинном формате. Зависит от региональных настроек даты и времени (например, Wednesday, March 14, 2001 для США).
Время в формате ат/рт	Время в 12-часовом формате с учетом региональных настроек даты и времени.
Средний формат времени	Время в формате АМ/РМ.
Время в 24-часовом формате	Время в 24-часовом формате с учетом региональных настроек даты и времени.

Да/Нет

Тип данных	Служит для отображения
Флажок	Поле флажка.
Да/Нет	Варианты "Да" и "Нет".
Истина/ложь	Варианты "Истина" и "Ложь".
Вкл/выкл	Варианты "Вкл" и "Выкл".

Быстрое начало

Тип данных	Служит для отображения
Адрес	Поля для всего почтового адреса.
Телефон	Поля для домашнего, мобильного и рабочего телефонов.
Важность	Раскрывающийся список с вариантами важности: низкая, средняя, высокая.
Состояние	Раскрывающийся список с вариантами "Не начато", "Выполняется", "Завершено", "Отменено".
Теги	От одного до трех тегов.

Объект OLE [Объекты OLE](#), такие как документы Word.

План-конспект занятия №6

Тема: Управление записями в таблицах БД

Цель занятия: познакомить с различными способами работы с записями в БД.

Вид занятия: теоретическое занятие

Задачи:

Учебные: 1) ознакомить с вариантами управления записями в БД

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности

Развиваемые общие компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1, ПК 2.1.

Самостоятельная работа студента:

Конспектирование (ОК 2.1.1)

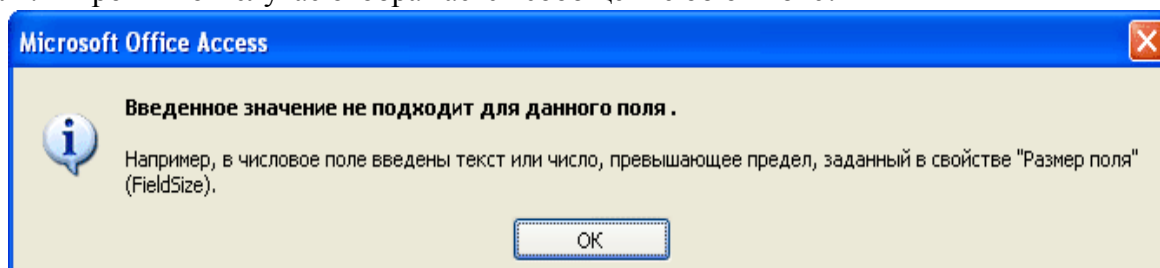
Управление записями в таблицах БД

Добавление записей непосредственно в таблицу в режиме таблицы

Таблица, открытая в режиме таблицы, похожа на лист Excel, так что можно вводить или вставлять данные в одно или несколько полей — как в ячейки листа.

В процессе работы помните о следующем.

- Нет необходимости явно сохранять свои данные. Приложение Access фиксирует изменения, внесенные в таблицу, при перемещении курсора на другое поле той же строки или на другую строку.
- По умолчанию поля в базе данных Access настроены на прием определенного типа данных, например, текста или чисел. Поэтому следует вводить данные того типа, который задан для поля. В противном случае отображается сообщение об ошибке:



- К полю может применяться маска ввода. Маска ввода представляет собой набор текстовых констант и подстановочных знаков, который определяет формат ввода данных.
- За исключением вложений и списков, одновременно допускающих несколько значений, в большинство полей можно ввести только одну запись. Если неизвестно, принимает ли поле вложения, следует посмотреть свойства поля. Список, допускающий одновременно несколько значений, можно легко отличить, так как в Access около каждого элемента такого списка отображается флажок.

Ввод данных в таблицу

1. В области переходов дважды щелкните нужную таблицу.

По умолчанию приложение Access открывает таблицу в режиме таблицы — в виде сетки, похожей на лист Excel.

2. Щелкните первое поле, которое требуется заполнить, или иным образом установите фокус на это поле, и введите данные.

3. Для перемещения на следующее поле в той же строке нажмите клавишу TAB, клавиши ВЛЕВО или ВПРАВО или щелкните ячейку следующего поля.

Если была нажата клавиша TAB, приложение Access по умолчанию использует региональные параметры Windows, чтобы определить, в каком направлении перемещать курсор: влево или вправо. Если используется язык с направлением письма слева направо, при нажатии клавиши TAB курсор перемещается вправо. В противном случае при нажатии клавиши TAB курсор перемещается влево.

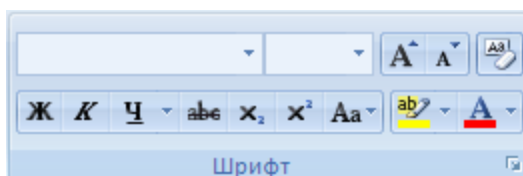
Чтобы перейти к следующей ячейке в столбце, нажмите клавишу ВВЕРХ или ВНИЗ, или щелкните нужную ячейку.

Применение форматирования текста в поле с типом данных «Поле МЕМО»

1. Выберите поле с типом данных «Поле МЕМО» в таблице, открытой в режиме таблицы.

Обычно поле типа «Поле МЕМО» содержит примечания или заметки, так что можно поискать поле с именем «Примечания» или «Заметки». Если таким способом не удастся найти поле типа «Поле МЕМО», см. действия в разделе [Просмотр свойств поля таблицы](#) выше в этой статье.

2. На вкладке **Начальная страница** в группе **Шрифт** воспользуйтесь кнопками и меню для форматирования текста.



Можно применять к тексту различные шрифты и размеры шрифта, курсивное или полужирное начертание, изменять его цвет и т. д.

Добавление записей с использованием формы

С помощью форм ввода данных можно сделать процесс ввода более простым, быстрым и точным. Также формы создаются, чтобы скрыть некоторые поля в таблице, упростить работу с базой данных (снизить стоимость обучения) и обеспечить точность ввода данных пользователями.

Структура формы определяет, как именно эта форма может использоваться для изменения данных. Форма может содержать любое количество элементов управления — списков, полей, кнопок и таблиц, которые выглядят, как листы Excel. Каждый элемент управления в форме может считывать данные из поля базовой таблицы или записывать данные в него. Поведение каждого конкретного элемента управления зависит от типа данных, заданных для поля базовой таблицы, свойств этого поля, и, возможно, от нескольких свойств, которые разработчик базы данных задал для этого элемента управления.

Далее объясняется, как использовать наиболее распространенные элементы управления ввода данных. За сведениями о конкретной базе данных обращайтесь к ее разработчику или к системному администратору.

Ввод или изменение текста в поле

При добавлении или изменении текста в текстовом поле работа с данными ведется в поле с типом данных «Текстовый» или «Поле MEMO». Помните, что можно установить для поля тип данных «Поле MEMO» (или для элемента управления, связанного с полем типа «Поле MEMO»), чтобы поддерживать форматирование текста, так что можно будет применять разные шрифты, размеры, стили и цвета текста.

Изменение текста в поле

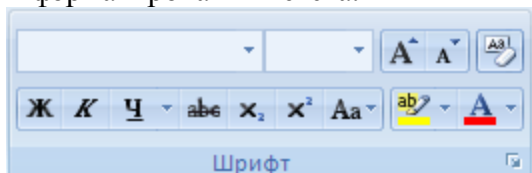
- Поместите курсор в текстовое поле и измените данные. Помните, что над числами в полях с типом данных «Текстовый» или «Поле MEMO» невозможно выполнять вычисления.

Применение форматирования текста

Примечание. Выполнять следующие действия можно, только когда элемент управления «Поле» связан с полем типа «Поле MEMO».

В форме, открытой в режиме формы или макета, или в таблице, открытой в режиме таблицы, выберите поле с типом данных «Поле MEMO».

1. На вкладке **Начальная страница** в группе **Шрифт** воспользуйтесь кнопками и меню для форматирования текста.

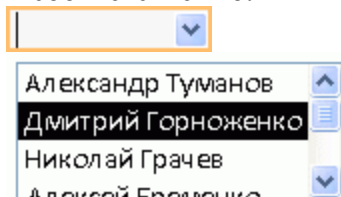


Можно применять к тексту различные шрифты и размеры шрифта, курсивное или полужирное начертание, изменять его цвет и т. д.

Ввод данных с использованием списка

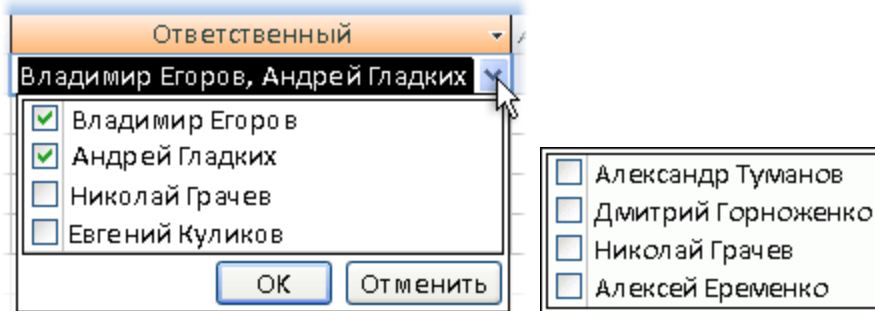
Списки можно использовать в формах, а также в таблицах и в результатах запросов, открытых в режиме таблицы — в виде сетки, похожей на лист Excel.

Список в форме обычно используется в виде элемента управления «Поле со списком» или «Список». Приведенные рисунки изображают элементы управления «Поле со списком» или «Список» соответственно.



При использовании списка в таблице или в результатах запроса приложение Access по умолчанию вставляет поле со списком.

Кроме того, списки могут поддерживать выбор одновременно нескольких значений. Если рядом с каждым элементом списка отображается флажок, то в таком списке можно выбрать одновременно несколько элементов (не более 100). Списки, поддерживающие выбор нескольких значений одновременно, — новая возможность Office Access 2007, позволяющая решить довольно распространенную проблему. Предположим, что база данных Access используется для отслеживания заявок в службу технической поддержки. Список, поддерживающий выбор нескольких значений, можно использовать, если одной заявкой занимаются несколько сотрудников. В более ранних версиях Access для создания подобного средства потребовались бы определенные навыки в программировании. На следующих рисунках показано, как выглядят списки с возможностью выбора нескольких значений.



Теперь можно также изменять элементы в списке. Если эта команда включена, можно непосредственно в списке добавлять, изменять или удалять элементы списка. Однако разработчик базы данных может отключить эту команду, поэтому она может быть доступна не во всех базах данных.

Ниже объясняется, как пользоваться списками и изменять их.

Выбор элементов в поле со списком

1. Откройте форму, таблицу или результаты запроса, в которых содержится список.

Примечание. Формы следует открывать в режиме формы или макета. Таблицы и результаты запросов следует открывать в режиме таблицы.

2. Щелкните стрелку вниз рядом со списком и выберите нужный элемент.
3. Чтобы зафиксировать сделанный выбор в базе данных, переместите курсор в другое поле.

Выбор элементов в списке

1. Откройте форму, содержащую список.
2. Прокрутите список элементов и выберите из них требуемый.
3. Чтобы зафиксировать сделанный выбор в базе данных, переместите курсор в другое поле.

Выбор элементов в поле со списком с флажками

1. Откройте форму, таблицу или результаты запроса, в которых содержится список.

Примечание. Поля со списком с флажками можно использовать в формах, в таблицах и в результатах запросов. Для этого формы следует открывать в режиме формы или просмотра, а таблицы и результаты запросов — в режиме таблицы.

2. Щелкните стрелку вниз рядом со списком.
3. Установите нужные флажки (до 100 флажков) и нажмите кнопку **ОК**.

Выбор элементов из списка с флажками

1. Откройте форму, содержащую список.

Примечание. Форму следует открывать в режиме формы или просмотра.

2. Щелкните стрелку вниз рядом со списком.
3. Установите нужные флажки (до 100 флажков) и нажмите кнопку **ОК**.

Изменение элементов в списке

Примечание. Можно изменять списки любых типов в форме, в таблице или в результатах запроса. Помните, что формы следует открывать в режиме формы или просмотра, а таблицы и результаты запросов — в режиме таблицы. Кроме того, помните, что команда изменения элементов списка может быть отключена разработчиком или администратором базы данных и поэтому не всегда бывает доступна.

Изменение элементов в поле подстановок

Поле подстановок представляет собой список данных, в котором пользователи могут выбирать один или несколько элементов. Можно создавать два типа списков подстановок.

- **Списки значений** содержат постоянный набор значений, которые вводятся вручную. Эти значения хранятся в свойстве **Источник строк (Row Source)** поля подстановок.
- **Списки подстановок** используют запрос для извлечения значений из другой таблицы. Свойство **Источник строк (Row Source)** такого поля вместо постоянного списка значений содержит запрос.

По умолчанию данные подстановки отображаются в поле со списком, хотя для этого можно выбрать элемент управления «Список». Поле со списком открывается для отображения списка и закрывается после того, как сделан выбор. Список, напротив, остается открытым все время.

Поле подстановок в форме

1. В области переходов щелкните форму правой кнопкой мыши и выберите пункт

Конструктор.

-или-

Если форма уже открыта, можно щелкнуть правой кнопкой мыши вкладку формы и выбрать в контекстном меню пункт **Конструктор**.

-или-

На вкладке **Начальная страница** в группе **Вид** щелкните кнопку **Вид** для переключения между доступными представлениями. Или можно щелкнуть стрелку под полем **Вид**, а затем выбрать в меню одно из доступных представлений..

2. Щелкните правой кнопкой мыши элемент управления типа «Список» или «Поле со списком» и выберите команду **Свойства**.

3. Перейдите в окне свойств на вкладку **Все** и найдите свойства **Тип источника строк (Row Source Type)** и **Источник строк (Row Source)**. Свойству **Тип источника строк (Row Source Type)** должно быть присвоено значение **Список значений** или **Таблица/Запрос**, а свойство **Источник строк (Row Source)** должно содержать либо список элементов, разделенных точкой с запятой, либо запрос. Для удобства чтения списка элементов или запроса окно свойств можно увеличить.

Обычно в списках значений используется следующий основной синтаксис:
"элемент";"элемент";"элемент"

В этом случае список представляет собой набор элементов, заключенных в двойные кавычки и разделенных точками с запятой.

В запросах на выборку используется следующий основной синтаксис: **SELECT** [имя_таблицы_или_запроса].[имя_поля] **FROM** [имя_таблицы_или_запроса].

В этом случае запрос состоит из двух предложений (SELECT и FROM). Первое предложение ссылается на таблицу или запрос и поле из этой таблицы или запроса. Второе предложение ссылается только на таблицу или запрос. Следует помнить важное правило: предложения SELECT могут не содержать имя таблицы или запроса, хотя они всегда содержат имя по крайней мере одного поля. Однако любое предложение FROM должно ссылаться на таблицу или запрос. Таким образом, прочитав предложение FROM, всегда можно найти исходную таблицу или исходный запрос для поля подстановок.

4. Выполните одно из следующих действий.

- Если используется список значений, измените элементы списка. Каждый элемент должен быть заключен в двойные кавычки и отделен точкой с запятой.

- Если запрос в списке подстановок ссылается на другой запрос, в области переходов откройте в режиме конструктора этот второй запрос (щелкните правой кнопкой мыши запрос и выберите команду **Конструктор**). Запомните имя таблицы, отображаемое в верхней части конструктора запросов, и перейдите к следующему шагу.

-или-

Если запрос в поле подстановок ссылается на таблицу, запомните ее имя и перейдите к следующему шагу.

5. Дважды щелкните эту таблицу в области переходов, чтобы открыть ее в режиме таблицы, и измените элементы списка.

Поле подстановок в таблице

1. В области переходов щелкните таблицу правой кнопкой мыши и выберите команду

Конструктор.

2. В верхней части сетки конструктора в столбце **Тип данных** щелкните любое поле с типом данных «Текстовый», «Числовой» или «Логический» (или иным образом переместите фокус на это поле).

3. В нижней части сетки конструктора таблицы откройте вкладку **Подстановка** и просмотрите свойства **Тип источника строк (Row Source Type)** и **Источник строк (Row Source)**.

Свойство **Тип источника строк (Row Source Type)** должно иметь значение **Список значений** или **Таблица/Запрос**. Свойство **Источник строк (Row Source)** должно содержать либо список значений, либо запрос.

В списках значений используется следующий основной синтаксис: "элемент";"элемент";"элемент"

В этом случае список представляет собой набор элементов, заключенных в двойные кавычки и разделенных точками с запятой.

Обычно в запросах на выборку используется следующий основной синтаксис: **SELECT** [имя_таблицы_или_запроса].[имя_поля] **FROM** [имя_таблицы_или_запроса].

В этом случае запрос состоит из двух предложений (SELECT и FROM). Первое предложение ссылается на таблицу или запрос и поле из этой таблицы или запроса. Второе предложение ссылается только на таблицу или запрос. Следует помнить важное правило: предложения SELECT могут не содержать имя таблицы или запроса, хотя они всегда содержат имя по крайней мере одного поля. Однако любое предложение FROM должно ссылаться на таблицу или запрос. Таким образом, прочитав предложение FROM, всегда можно найти исходную таблицу или исходный запрос для поля подстановок.

4. Выполните одно из следующих действий.

- Если используется список значений, измените элементы списка. Каждый элемент должен быть заключен в двойные кавычки и отделен точкой с запятой.

- Если запрос в поле подстановок ссылается на другой запрос, в области переходов откройте в режиме конструктора этот второй запрос (щелкните правой кнопкой мыши запрос и выберите команду **Конструктор**). Запомните имя таблицы, отображаемое в верхней части конструктора запросов, и перейдите к следующему шагу.

-or-

Если запрос в поле подстановок ссылается на таблицу, запомните ее имя и перейдите к следующему шагу.

5. Дважды щелкните эту таблицу в области переходов, чтобы открыть ее в режиме таблицы, и измените элементы списка.

Ввод строк нулевой длины

В Access различаются два типа пустых значений: значения Null и строки нулевой длины. Значения Null характеризуют неизвестные значения, а строки нулевой длины характеризуют поля, которые содержат пробел. Допустим, имеется таблица, в которой содержатся данные о заказчиках, и в этой таблице есть поле для номера факса. Если номер факса неизвестен, это поле можно оставить пустым. Поле, оставленное пустым, содержит значение Null, говорящее о том, что значение неизвестно. Если впоследствии выясняется, что у заказчика нет факса, можно ввести в это поле строку нулевой длины, говорящую о том, что известно, что значения нет.

Примечание. Помните, что можно установить свойство, запрещающее принимать значения Null.

Откройте таблицу в режиме таблицы или форму в режиме формы или в режиме просмотра.

1. Выберите нужное поле и введите пару двойных кавычек без пробела между ними ("").

2. Для фиксации изменений в базе данных переместите курсор на другую запись. По умолчанию кавычки будут скрыты.

Добавление записей с помощью средства сбора данных

В Office Access 2007 появилось новое средство: «Сбор данных». Можно создать форму ввода данных в Microsoft Office Outlook 2007, разослать эту форму по электронной почте для заполнения и сохранить эти сведения в базе данных Access. В этой статье не объясняется, как использовать средство «Сбор данных».

Добавление записей путем импорта данных или связывания с ними

Данные, находящиеся в других базах данных Access, книгах Excel, текстовых файлах, документах Word или списках на серверах с Microsoft Windows SharePoint Services 2.0 и более поздними версиями, можно импортировать и создавать с ними связь.

Добавление записи из другой базы данных с помощью запроса на добавление

Запрос на добавление добавляет группу записей из одной или нескольких таблиц исходной базы данных в одну или несколько таблиц конечной базы данных. Например, появилось несколько новых клиентов, а также база данных, содержащая таблицу сведений о них. Чтобы не вводить все новые данные вручную, их можно добавить в одну или несколько таблиц своей базы данных. Запросы на добавление можно использовать также для следующих целей.

- Добавление полей на основе условий отбора. Например, может потребоваться добавить только имена и адреса заказчиков с очень выгодными заказами.
- Добавление записей, когда некоторых полей одной таблицы не существует в другой. Допустим, в таблице «Заказчики» одной базы данных имеется 11 полей, а в таблице «Заказчики» другой базы данных из этих 11 полей имеется только девять. Можно использовать запрос на добавление только тех данных, которые находятся в соответствующих друг другу полях, а все остальные игнорировать.

Способы экономии времени при вводе данных

В приложении Access можно ускорить ввод данных несколькими способами. Процедуры, представленные ниже, поясняют, как настроить значения по умолчанию для полей и списков в формах и как использовать сочетания клавиш для повторного использования значений при сортировке данных.

Задание значения по умолчанию для поля или элемента управления

Если во многих записях определенное поле (например, «Город» или «Страна») имеет одно и то же значение, то можно ускорить ввод данных, задав значение по умолчанию для элемента управления, связанного с этим полем. При открытии формы или создании новой записи это значение по умолчанию появляется в элементе управления.

Задание значения по умолчанию для элемента управления

1. Откройте форму в режиме конструктора.
2. Щелкните правой кнопкой мыши нужный элемент управления и выберите команду

Свойства.

3. На вкладке **Данные** укажите значение для свойства **Значение по умолчанию (Default Value)**.
4. Для задания другого значения по умолчанию повторите шаги 1-3.

Задание другого значения

1. Для значения в поле введите новые данные. Значение по умолчанию всегда можно заменить другим значением.

–или–

Для значения в списке, выберите другое значение из списка.

Повторное использование значений при сортировке

При переходе от записи к записи может возникнуть необходимость сортировки каждой записи по одному и тому же значению. При этом можно экономить время, повторяя с помощью сочетаний клавиш значение, по которому производится сортировка. Допустим, выполняется сортировка списка заказов по типу товара. В первой записи введено **безалкогольные напитки** для сортировки по заказам, где заказчики покупают безалкогольные напитки. При переходе к следующей записи это значение сортировки можно скопировать и использовать снова.

Примечание. В этих действиях предполагается, что форма открыта в режиме формы.

1. Перейдите к полю, в котором повторно используется значение из соответствующего поля предыдущей записи.
2. Нажмите клавиши CTRL+' (апостроф).

План-конспект занятия №7

Тема: Сортировка, поиск и фильтрация данных

Цель занятия: познакомить с различными способами сортировки, поиска и фильтрации данных

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) ознакомить с видами портов
2) научить решать задачи на выбор типа порта

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности

Развиваемые общие компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1

Развиваемые профессиональные компетенции: ПК 2.1.

Самостоятельная работа студента:

Составление опорного конспекта

Сортировка данных в таблице

Операция сортировки данных используется всегда для удобства нахождения нужной информации. Когда на экране (или на бумаге) отображается таблица, гораздо легче найти нужную строку, если эти строки упорядочены. Вы привыкли к тому, что табличные данные упорядочены по алфавиту, по дате, по увеличению или уменьшению значений в столбцах, содержащих числа. Но в разных ситуациях мы хотели бы сортировать строки по разным признакам (столбцам таблицы). В идеале это должно выполняться легким движением руки. Именно так и позволяет делать Access. По умолчанию, когда таблица открывается в режиме Таблицы, она упорядочивается по значению ключевого поля. Если ключевое поле для таблицы не определено, записи выводятся в порядке их ввода в таблицу. Если нужно отсортировать записи по значению другого поля, достаточно установить курсор на любую строку соответствующего столбца и нажать одну из кнопок на панели инструментов: **Сортировка по возрастанию** (Sort Ascending) или **Сортировка по убыванию** (Sort Descending).

Другой способ выполнения этой операции: щелкнуть правой кнопкой мыши по любой строке нужного столбца и выбрать из контекстного меню (рис.5) соответствующую команду.

Таким образом можно упорядочить записи по любому столбцу таблицы. Ограничения существуют только на тип данных, которые можно упорядочивать — нельзя сортировать значения полей типа MEMO, гиперссылки или объекты OLE.

Чтобы правильно применять сортировку, нужно знать несколько простых правил.

- При сортировке в возрастающем порядке записи, содержащие пустые поля (с пустыми значениями), указываются в списке первыми.
- Числа, находящиеся в текстовых полях, сортируются как строки символов, а не как числовые значения. Если нужно отсортировать их в числовом порядке, все текстовые строки должны содержать одинаковое количество символов. Если строка содержит меньшее количество символов, то сначала нужно вставить незначащие нули.
- При сохранении таблицы сохраняется и порядок ее сортировки.

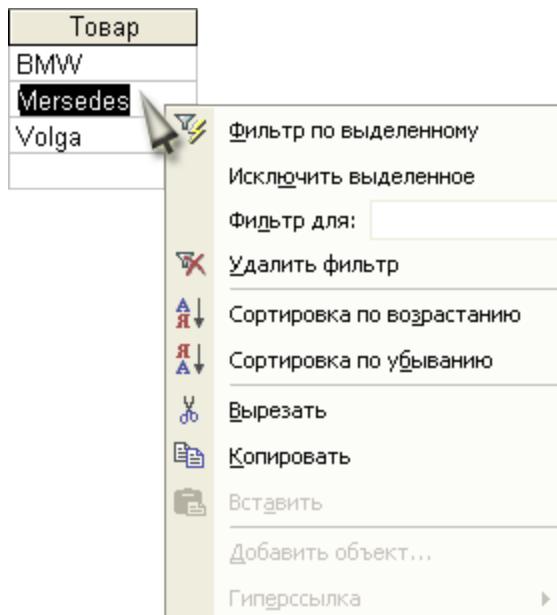


Рис. 5. Контекстное меню в режиме таблицы

Замечание

Порядок сортировки зависит от настроек языка, определенных в диалоговом окне Параметры (Options) при создании базы данных. Чтобы изменить эту настройку, необходимо выбрать в меню Сервис (Tools) команду Параметры (Options), открыть вкладку Общие (General) и в поле Порядок сортировки базы данных (New database sort order) выбрать нужный язык (см. гл. 14).

Совет

Процесс сортировки значительно ускоряется, если предварительно по данному полю был построен индекс.

Если вы изменили порядок сортировки записей в таблице, а потом решили ее закрыть, Access запрашивает, хотите ли вы сохранить изменения макета. Если вы ответите Да (Yes), то новый порядок сортировки сохранится и воспроизведется, когда таблица будет открыта в следующий раз. Для того чтобы вернуться к первоначальному порядку сортировки (по значению ключевого поля), необходимо выполнить команду меню **Записи, Удалить фильтр** (Records, Remove Filter/Sort).

Иногда нужно выполнить сортировку по значению нескольких полей. Для этого переместите сначала сортируемые столбцы таким образом, чтобы они оказались, во-первых, рядом, а во-вторых, с учетом приоритетов, — приоритеты устанавливаются слева направо, т. к. первыми будут сортироваться значения в крайнем левом столбце. После этого нужно выделить все столбцы и нажать, соответственно, кнопку **Сортировка по возрастанию** (Sort Ascending) или **Сортировка по убыванию** (Sort Descending) на панели инструментов.

Поиск и замена данных

В режиме Таблицы предусмотрены возможности поиска данных по образцу либо в конкретном поле, либо во всей таблице.

Чтобы найти необходимые данные, нужно:

1. Открыть таблицу в режиме Таблицы.
2. Если известно, в каком столбце нужно вести поиск, выделить этот столбец (достаточно поместить курсор в любое поле этого столбца).
3. Нажать кнопку **Найти** (Find) на панели инструментов **Режим таблицы** (Table Datasheet) и открыть вкладку **Поиск** (Find) или выполнить команду меню **Правка, Найти** (Edit, Find). Появится диалоговое окно **Поиск и замена** (Find and Replace) (рис. 6).

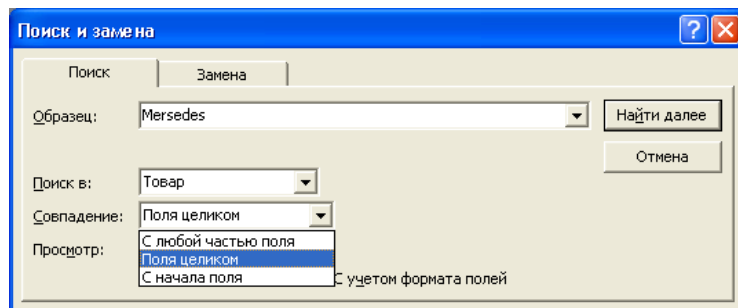


Рис 6. Окно поиска данных

4. Ввести значение, которое требуется найти, в поле **Образец** (Find What). Если точное значение неизвестно, можно использовать подстановочные знаки.

5. Остальные параметры в окне можно изменить или оставить так, как они установлены по умолчанию:

- значение в поле со списком **Поиск в** (Look In), определяющее место поиска, по умолчанию содержит название выбранного столбца;
- значение в поле со списком **Совпадение** (Match) определяет один из трех вариантов совпадения образца со значением поля — **С любой частью поля** (Any Part of Field), **Поля целиком** (Whole Field), **С начала поля** (Start of Field);
- поле со списком **Просмотр** (Search), в котором можно задать направление поиска: **Все** (All), **Вверх** (Up), **Вниз** (Down);
- флажок **С учетом регистра** (Match Case) позволяет при поиске учитывать начертание букв — прописные или строчные;
- флажок **С учетом формата полей** (Search Fields As Formatted) позволяет выполнять поиск данных в указанном формате отображения.

Чтобы найти первое вхождение указанного образца, нажмите кнопку **Найти следующий** (Find Next).

Для поиска следующих вхождений этого значения нажимайте кнопку **Найти далее** (Find Next) до тех пор, пока не будет найдено нужное вхождение.

Для доступа к окну **Поиск и замена** (Find and Replace) можно использовать также команду меню **Правка, Найти** (Edit, Find).

Замечание

Из всех видов поиска самым медленным будет поиск по совпадению с любой частью поля, при этом не используется индекс, даже если поле, по которому выполняется поиск, проиндексировано.

Чтобы выполнить поиск по всем столбцам таблицы, нужно из списка **Поиск в** (Look In) выбрать значение **<Имя таблицы>: таблица**.

Чтобы выполнить не только поиск, но и изменение данных в найденном поле, необходимо в том же диалоговом окне **Поиск и замена** (Find and Replace) раскрыть вкладку **Замена** (Replace) (рис. 7) и в поле **Заменить на** (Replace With) указать значение, на которое нужно заменить искомое значение. После этого нажмите кнопку **Заменить** (Replace) или **Заменить все** (Replace All).

Фильтрация данных

В Microsoft Access предусмотрено четыре способа отбора записей с помощью фильтров: фильтр по выделенному фрагменту, обычный фильтр, поле **Фильтр для** (Filter For) и расширенный фильтр.

Фильтр по выделенному фрагменту, обычный фильтр и поле **Фильтр для** (Filter For) являются очень простыми способами отбора записей, причем самым простым является фильтр по выделенному фрагменту — он позволяет найти все записи, содержащие определенное значение в выбранном поле. Обычный фильтр используется для отбора записей по значениям нескольких полей. Поле **Фильтр для** (Filter For) используется, если фокус ввода находится в поле таблицы и нужно ввести конкретное искомое значение или выражение, результат которого будет применяться в качестве условия отбора. Для создания сложных фильтров следует использовать окно расширенного фильтра.

Набор записей, которые были отображены в процессе фильтрации, называется *результатирующим набором*.

Условия отбора записей, которые вводятся в поля формы **фильтр** (Filter by Form), можно сохранить в базе данных в виде запроса. Для этого необходимо при открытой форме **фильтр** (Filter by Form) нажать кнопку **Сохранить как запрос** (Save As Query) на панели инструментов. Затем в диалоговом окне **Сохранение в виде запроса** (Save As Query) ввести имя запроса и нажать кнопку **ОК**. Когда потребуются повторить установку такого фильтра, нужно опять же при открытой форме **фильтр** (Filter by Form) нажать кнопку **Загрузить из запроса** (Load from Query) на панели инструментов.

Отобранные с помощью фильтра данные можно копировать, экспортировать и рассылать. Копирование данных из выборки осуществляется так же, как копирование данных в таблице.

Экспорт данных из выборки позволяет сохранить их в отдельном файле. Чтобы экспортировать данные, необходимо:

1. Выбрать в меню **Файл** (File) команду **Экспорт** (Export).
2. В окне **Экспорт объекта: Таблица <имя таблицы> в** (Export Table <имя таблицы> to) выбрать папку для экспорта файла.
3. В поле **Имя файла** (File Name) ввести имя файла, в который осуществляется экспорт.
4. В поле **Тип файла** (Save as Type) выбрать тип файла, в который осуществляется экспорт.
5. Нажать кнопку **Сохранить все** (Export All).

В отличие от обычного экспорта данных, рассылка позволяет сразу отправить результаты выборки по электронной почте разным адресатам для дальнейшего ознакомления с материалами, их анализа и т. п. Чтобы отослать результаты выборки, необходимо:

1. Выбрать в меню **Файл** (File) команду **Отправить** (Send To).
2. Из раскрывающегося меню выбрать пункт **Почтовый клиент** (Mail Recipient (as Attachment)).
3. В окне **Послать** (Send) выбрать тип файла.
4. Нажать кнопку **ОК** и далее выполнить стандартную процедуру для отправки почтового сообщения в своем почтовом клиенте.

План-конспект занятия №8

Тема: Взаимосвязи между таблицами. Способы объединения таблиц

Цель занятия: изучить назначение связей между таблицами и способы объединения таблиц

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить виды и назначение связей между таблицами
2) раскрыть особенности и способы объединения таблиц;

Развивающие: развитие логического мышления; развитие памяти

Воспитательные: воспитание аккуратности и внимательности

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1, ОК 3.1.2, ПК 2.1

Самостоятельная работа студента: Составление опорного конспекта (ОК 4.2.1)

ТИПЫ СВЯЗЕЙ. СВОЙСТВА ОТНОШЕНИЙ

Реляционные базы данных состоят из нескольких таблиц, связь между которыми устанавливается с помощью совпадающих полей. Каждая запись в таблицах идентифицирует один объект. Отношение между объектами определяет отношение между таблицами. Существует 4 типа отношений:

- **Отношение "один-к-одному"** (1:1) означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.
- **Отношение "один-ко-многим"** (1 :М) означает, что каждой записи в одной таблице соответствует одна или несколько записей в другой таблице.
- **Отношение "многие-к-одному"** (М:1) аналогично рассмотренному ранее типу "один-ко-многим". Тип отношения между объектами зависит от вашей точки зрения.

- **Отношение "многие-ко-многим" (М:М).** возникает между двумя таблицами в тех случаях, когда каждой записи в одной таблице соответствует 0, 1, 2 и более записей в другой таблице и наоборот.

В большинстве случаев любые две таблицы связаны отношением "один-ко-многим". Это означает, что любая запись в первой таблице может быть связана с несколькими записями во второй, однако любая запись второй таблицы связана только с одной записью в первой.

Пример 1 Дана совокупность информационных объектов, отражающих учебный процесс в ссузе:

СТУДЕНТ (*Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа*) СЕССИЯ (*Номер, Оценка1, Оценка2, Оценка3, Оценка4, Результат*) СТИПЕНДИЯ (*Результат, Процент*) ПРЕПОДАВАТЕЛЬ (*Код преподавателя, Фамилия, Имя, Отчество*)

Связь один к одному (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот. [Рисунок 5](#) иллюстрирует указанный тип отношения.

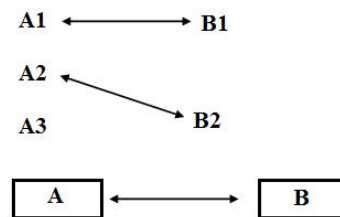


Рис. 5. Графическое изображение реального отношения 1:1

Пример 2 Примером связи 1:1 может служить связь между информационными объектами СТУДЕНТ и СЕССИЯ:

СТУДЕНТ <-> СЕССИЯ Каждый студент имеет определенный набор экзаменационных оценок в сессию.

При связи **один ко многим (1:M)** одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на [рис. 6](#).

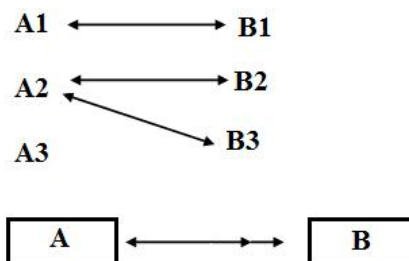


Рис. 6. Графическое изображение реального отношения 1:M

Пример 3 Примером связи 1 :M служит связь между информационными объектами СТИПЕНДИЯ и СЕССИЯ:

СТИПЕНДИЯ <- " СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь **многие ко многим (М:М)** предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот. На [рис. 7](#) графически представлено указанное соответствие.

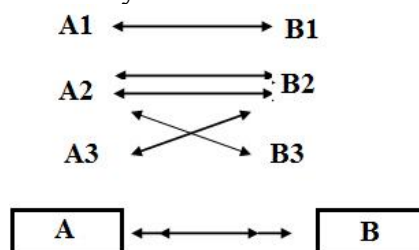


Рис. 7. Графическое изображение реального отношения М:М

Пример 4 Примером данного отношения служит связь между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ:

СТУДЕНТ "-" ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

ПРОСТЫЕ И СОСТАВНЫЕ КЛЮЧИ

Первичный ключ может состоять из единственного поля таблицы, значения которого уникальны для каждой записи. Так, например, на предприятии не может быть двух работников с одинаковыми табельными номерами, поэтому в таблице, содержащей записи о работниках, табельный номер может быть первичным ключом. Такой первичный ключ называют простым ключом.

Если таблица не имеет единственного уникального поля, первичный ключ может быть составлен из нескольких полей, совокупность значений которых гарантирует уникальность. Так, имя, фамилия, отчество, номер паспорта, серия паспорта не могут быть первичными ключами по отдельности, так как могут оказаться одинаковыми у двух и более людей. Но не бывает двух личных документов одного типа с одинаковыми серией и номером. Поэтому в таблице, содержащей записи о людях, первичным ключом может быть набор полей, состоящий из типа личного документа, его серии и номера. Такой первичный ключ называют составным ключом

Все остальные ключи отношения называются возможными ключами.

В отличие от иерархической и сетевой моделей данных в реляционной отсутствует понятие группового отношения. Для отражения ассоциаций между кортежами разных отношений используется дублирование их ключей. Рассмотренный иерархический и сетевой пример базы данных, содержащей сведения о подразделениях предприятия и работающих в них сотрудниках, применительно к реляционной модели будет иметь вид:



Рис. 8. База данных о подразделениях и сотрудниках предприятия

Например, связь между отношениями ОТДЕЛ и СОТРУДНИК создается путем копирования первичного ключа "Номер_отдела" из первого отношения во второе. Таким образом:

- для того, чтобы получить список работников данного подразделения, необходимо из таблицы ОТДЕЛ установить значение атрибута "Номер_отдела", соответствующее данному "Наименованию_отдела" выбрать из таблицы СОТРУДНИК все записи, значение атрибута "Номер_отдела" которых равно полученному на предыдущем шаге.

- для того, чтобы узнать в каком отделе работает сотрудник, нужно выполнить обратную операцию:

- определяем "Номер_отдела" из таблицы СОТРУДНИК
- по полученному значению находим запись в таблице ОТДЕЛ.

Атрибуты, представляющие собой копии ключей других отношений, называются внешними ключами.

Иногда возникает потребность разбить одну таблицу на более мелкие - проблема может заключаться в том, что некоторые сведения из нее используются не слишком часто, или в том, что какие-то данные не предназначены для всеобщего доступа. Например, часть информации о факультетах нужна только для рекламных целей и используется очень редко. С другой стороны, сведения о заработной плате должны быть доступны только определенным сотрудникам. В любом из этих случаев можно создать отдельную таблицу и связать ее с исходной таблицей отношением

типа "один-к-одному". Это означает, что любая запись в первой таблице связана только с одной записью во второй.

Если же между таблицами необходимо организовать связь "многие-ко-многим", то в Access придется создать дополнительную таблицу пересечения, с помощью которой одна связь будет сведена к двум связям типа "один-ко-многим".

Схема данных в Access

Создание схемы данных позволяет упростить конструирование многотаблиц, форм, запросов, отчетов, а также обеспечить поддержание целостности взаимосвязанных данных при корректировке таблиц. Схема данных наглядно отображает таблицы и связи между ними, а также обеспечивает использование связей при обработке данных и целостность БД.

Схема данных задает структуру БД. Она является графическим образом БД. Схема данных базы графически отображается в своем окне, таблицы представлены списками полей, а связи - линиями между полями разных таблиц. Схема данных ориентирована на работу с таблицами, отвечающими требованиям нормализации, между которыми установлены связи 1:М и 1:1 с обеспечением целостности БД. Поэтому схема данных отойдет в соответствии с информационно-логической моделью.

При построении схемы данных Access автоматически определяет по выбранному полю связи тип связи между таблицами. Если поле, по которому нужно установить связь, является уникальным ключом, как в главной таблице, так и в подчиненной Access устанавливает связь «один – к – одному». Если поле связи является уникальным ключом в главной таблице, а в подчиненной таблице является неключевым или входит в составной ключ, Access устанавливает связь «один ко многим» от главной таблице к подчиненной.

Связи - объединения. При выборе в качестве поля связи в главной таблице неключевого поля Access сообщает, что тип отношения не может быть определен. В этом случае между таблицами возможно установление только связи - объединения. **Связь - объединение** обеспечивает объединение записей двух таблиц, имеющих одинаковые значения в поле связи. Причем производится объединение каждой записи из одной таблицы с каждой записью из другой таблицы при условии равенства значений в поле связи. Результатом объединения записей, который определяется выбором одного из трех способов, может быть:

- объединение только тех записей, в которых связанные поля обеих таблиц совпадают;
- объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из первой таблицы, для которых нет связанных во второй, с пустой записью второй таблицы;
- объединение тех записей, в которых связанные поля обеих таблиц совпадают, а также объединение всех записей из второй таблицы, для которых нет связанных в первой, с пустой записью первой таблицы.

Создание схемы данных начинается в окне БД с выполнения команды **Сервис / Схема данных** или нажатием соответствующей кнопки. После этого можно выбрать таблицы, включаемые в схему данных, и приступить к определению связей между ними. Устанавливая связь между парой таблиц в схеме данных, надо выделить в главной таблице уникальное ключевое поле, по которому устанавливается связь, и протащить курсор мыши в соответствующее поле подчиненной таблицы.

Поскольку схема строится для нормализованной БД, в открывающемся окне Связи автоматически установится тип отношения 1:М.

Для такой связи можно задать параметр: *Обеспечение целостности данных*, а затем установить флажки *Каскадное обновление* связанных полей и *Каскадное удаление* связанных записей.

Тема: Обеспечение достоверности, целостности и непротиворечивости данных. Каскадные воздействия

Цель занятия: изучить понятия достоверности и непротиворечивости данных.

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить понятия достоверности и непротиворечивости данных.
2) Разъяснить понятие каскадных воздействий

Развивающие:

- 1) развитие логического мышления;
2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности

Развиваемые компетенции: ОК 1, ОК 10, ОК4.1.1; ОК 4.2.1, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта по теме (ОК 4.1.1)

Блиц-опрос (проверка ДЗ)

А.В. Кузин. Реляционная модель (с.15-17)

1. Тип взаимосвязи в модели БД, при котором каждому экземпляру одного объекта А могут соответствовать несколько экземпляров второго объекта В и наоборот, называется:

- А) один-к-одному
Б) один-ко-многим
В) *многие-ко-многим*

2. Назовите тип взаимосвязи в модели БД, при котором соответствие между объектами не взаимно-равнозначное:

- А) один-к-одному
Б) *один-ко-многим*
В) многие-ко-многим

3. Какая из перечисленных взаимосвязей в модели БД реализуется в Access только посредством дополнительного промежуточного отношения:

- А) один-к-одному
Б) один-ко-многим
В) *многие-ко-многим*

4. Какая из перечисленных взаимосвязей в модели БД может возникнуть в случае, когда оба связываемые атрибуты отношений – ключевые:

- А) *один-к-одному*
Б) один-ко-многим
В) многие-ко-многим

5. Можно ли в качестве ключевого поля использовать текстовые атрибуты?

- А) *да*
Б) нет
В) *свой ответ*

1	2	3	4	5
<i>В</i>	<i>Б</i>	<i>В</i>	<i>А</i>	<i>А,В</i>

Обеспечение целостности данных

Если связываемые в схеме данных таблицы находятся в отношении 1 : 1 и 1 : М, для связи можно задать параметр обеспечения связной Целостности данных. Обеспечение связной целостности данных означает, что Access при корректировке базы данных реализует для связанных таблиц контроль соблюдения следующих условий:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в записях главной таблицы значением ключа связи;

- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;

- изменение значений ключа связи в записи главной таблицы невозможно, если подчиненная таблица имеет связанные с ней записи, в которых соответственно не меняется ключ связи.

При попытке пользователя нарушить эти условия в операциях добавления и удаления записей или обновления ключевых данных в связанных таблицах Access не допускает выполнения операции и выводит соответствующее сообщение. Access не позволяет установить параметр целостности для связи таблиц, если ранее введенные в таблицы данные не отвечают требованиям целостности.

Установление между двумя таблицами связи и задание для нее параметров целостности данных возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип данных, причем имена полей могут быть различными;

- обе таблицы сохраняются в одной базе данных Access;

- главная таблица связывается с подчиненной по первичному простому или составному ключу или уникальному индексу главной таблицы.

Каскадное обновление и удаление связанных записей

Если для выбранной связи обеспечивается поддержание целостности, можно задать режим каскадного удаления связанных записей и каскадного обновления связанных полей. Такие параметры делают возможным в главной таблице, соответственно, удаление записей и изменение значения в ключевом поле, т. к. при этих параметрах система автоматически выполнит необходимые изменения в подчиненных таблицах, обеспечив сохранение свойств целостности базы данных.

В режиме **каскадного удаления связанных записей** при удалении записи из главной таблицы будут автоматически удаляться все связанные записи в подчиненных таблицах. При удалении записи из главной таблицы выполняется каскадное удаление подчиненных записей на всех уровнях, если этот режим задан на каждом уровне.

В режиме **каскадного обновления связанных полей** при изменении значения ключевого поля в записи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях.

План-конспект занятия №10

Тема: Формы, элементы управления формы: свойства, события и методы

Цель занятия: познакомить с назначением и способами создания форм.

Вид занятия: теоретическое занятие

Задачи:

Учебные: 1) ознакомить с формами

2) Изучить элементы управления формы;

Развивающие:

1) развитие логического мышления;

2) развитие памяти

Воспитательные: воспитание аккуратности и внимательности

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 3.1.1, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта по теме (ОК 2.1.1)

Формы MS Access. Общие сведения

Конечному пользователю БД работать непосредственно с таблицами не всегда бывает удобно, а представление в них данных не очень наглядно. Формы MS Access как правило используются для создания пользовательского интерфейса любой БД. Большая часть данных, представленных в форме, берется из таблицы или запроса. Каждая форма представляет собой контейнер, в котором размещаются *элементы управления* для просмотра и редактирования данных, навигации по набору записей и выполнение активных действий, при возникновении разных событий.

Форма — это объект БД, предназначенный для ввода и отображения информации. Формы позволяют выполнить проверку корректности данных при вводе, проводить вычисления, обеспечивают доступ к данным в связанных таблицах с помощью подчиненных форм.

Работа с формами может происходить в трех режимах: в режиме Формы, в режиме Таблицы, в режиме Конструктора. Выбрать режим работы можно при помощи кнопки Вид панели инструментов Конструктор форм либо с помощью команды меню Вид.

В режимах Формы и Таблицы можно осуществлять добавление, удаление и редактирование записей в таблице или в запросе, являющемся источником данных для форм.

В режиме Конструктора можно производить изменение внешнего вида формы, добавление и удаление элементов управления, разработку.

Элемент управления - это визуальный объект, используемый для отображения информации, ввода и изменения данных, выполнения определенных действий или просто для украшения формы.

Первое и наиболее часто встречающееся применение форм - ввод и редактирование информации. Для этого применяются различные элементы управления, чаще всего это поля ввода, в которые вводятся данные различных типов, выпадающие списки с набором значений для подстановки и логические переключатели. Также с помощью формы можно производить любые манипуляции с базой данных: создавать и уничтожать ее объекты, запускать на исполнение другие формы или отчеты и др. Формы по сравнению с обработкой данных в режиме таблицы обладают следующими преимуществами:

1. форма позволяет в каждый момент времени сфокусировать внимание на отдельной записи, поскольку, как правило, отображает все поля одной записи в отличие от режима таблицы, отображающего сразу несколько записей, так что приходится прокручивать окно, чтобы просмотреть содержимое всех полей;

2. элементы управления на форме можно расположить логичным образом, облегчающем чтение и работу с данными;

3. некоторые объекты базы данных, такие как рисунки, анимация, звуки и видео клипы, могут отображаться или воспроизводиться только в режиме формы, а не в режиме таблицы.

Редактирование формы происходит в визуальном режиме, что позволяет экспериментировать с расположением элементов управления и их внешним видом. При этом многие элементы при создании запускают своих "Мастеров", облегчающих процесс настройки их свойств.

Структуры формы. Любая форма может включать следующие разделы:

- заголовок формы — определяет верхнюю часть формы и может содержать текст, графику и другие элементы управления;

- верхний колонтитул — раздел отображается только в режиме предварительного просмотра и обычно содержит заголовки столбцов;

- область данных — определяет основную часть формы, содержащую поля, полученные из источника данных;

- нижний колонтитул — раздел отображается только в режиме предварительного просмотра в нижней части экранной страницы и обычно содержит номер страницы, дату и т. д.;

- примечание формы — отображается внизу последней экранной страницы формы.

Форма может содержать все-разделы или некоторые из них.

Создание форм

Создание формы начинается с щелчка на кнопке "Создать" в окне базы данных, после чего откроется диалоговое окно, представленное на рисунке 7.2.1, в котором следует выбрать способ создания формы.

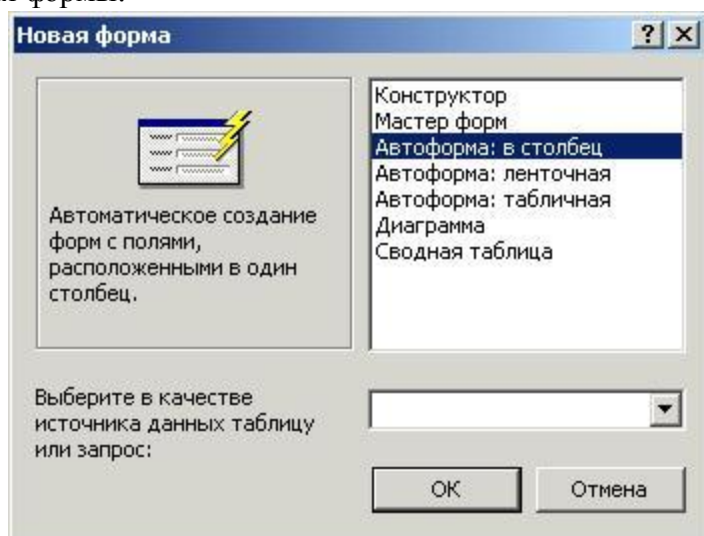


Рисунок 7.2.1 - Окно создания формы

Автоформа - автоматическое создание формы с использованием одного из стандартных шаблонов. Это наиболее простой и быстрый способ создания формы. Если выбран вариант "Автоформа: в столбец", то все поля формы располагаются друг над другом, и в один конкретный момент времени на экране видны данные только одной из записей таблицы базы данных. Вариант "Автоформа: ленточная", больше подходит тогда, когда таблица хранит некоторые значения одного параметра, например, продаж по датам. В одном столбце можно вывести значения дат, а во втором - объем соответствующих им продаж. Ленточная форма похожа на таблицу, но данные в ней расположены не так компактно. Каждая запись занимает отдельную строку - "ленту". Вариант "Автоформа: табличная" предназначен для автоматического создания форм, внешне похожих на таблицы. Для того, чтобы увидеть истинный вид табличной формы, необходимо выполнить команду "Вид" → "Режим формы".

Мастер форм - создание формы с помощью мастера (в зависимости от назначения формы "Мастер" предлагает на выбор стандартные шаблоны и стили оформления). Этот вариант лучше всего использовать в тех случаях, когда из большой таблицы требуется поместить на форму всего несколько отдельных полей и при этом автоматически расположить их на пространстве формы наиболее выгодным образом.

Конструктор - создание формы на основе пустого бланка при помощи инструментальных средств конструктора форм. Этот вариант лучше всего использовать либо для очень простых форм, либо, наоборот, для очень сложных, не предусмотренных встроенным мастером MS Access.

Диаграмма - создание формы с диаграммой на основе выбранных полей таблицы. В этом случае по указанным полям Microsoft Excel автоматически строит диаграмму.

Сводная таблица - создание сводной таблицы Microsoft Excel на основе таблиц или запросов Access. Этот вариант создания формы следует использовать в тех случаях, когда возникает необходимость отобразить некий интегрированный суммарный результат.

Содержимое поля со списком "Таблица/Запрос" показывает, на основе какого объекта MS Access (таблицы или запроса) будет создана форма.

Для настройки элементов формы следует переключиться в режим конструктора.

Элементы управления

Все элементы, добавляемые в форму, являются элементами управления. Примерами разных элементов управления служат поля, надписи, списки, переключатели, кнопки и линии.

При открытии формы в режиме конструктора MS Access отображает панель элементов, предлагающую средства для добавления и работы с различными типами элементов управления.

Если панель элементов отсутствует на экране, то необходимо выполнить команду в меню "Вид"→"Панель элементов". Кнопки панели элементов показаны на рисунке 7.3.1.

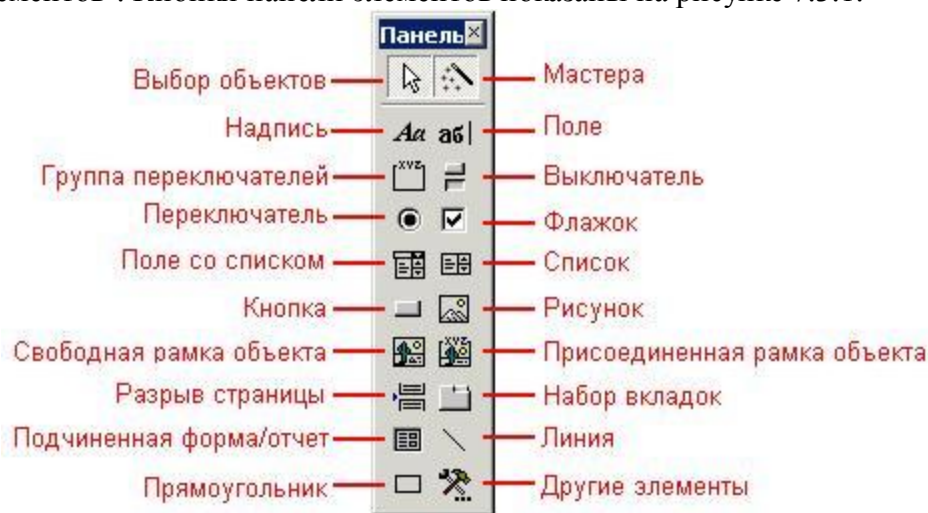


Рисунок 7.3.1 - Кнопки панели элементов и их функции

Описание кнопок панели элементов представлено в следующей таблице.

Название кнопки	Описание
Выбор объектов	Позволяет выделить определенный элемент управления, щелкнув на нем, или группу элементов, поочередно щелкая на них при нажатой клавише Shift. При нажатой кнопке можно выделить элементы управления с помощью прямоугольника выделения
Надпись	Позволяет ввести описательный текст, чтобы назвать элемент или дать инструкцию. Большинству элементов управления надписи придаются автоматически
Группа переключателей	Создается ряд элементов управления, среди которых можно производить выбор и которые присваивают полю числовое значение
Переключатель	Создается круглая кнопка, которая обычно используется в группе взаимоисключающих переключателей. Выделенная кнопка обозначена черным кружком
Поле со списком	Создается элемент управления, состоящий из текстового поля со списком под ним. Пользователь может или ввести текст в текстовое поле, или выбрать пункт из списка
Кнопка	Добавляется прямоугольная кнопка, на которой можно щелкнуть для выполнения действия (например Ok или Закрыть)
Свободная рамка объекта	Отображаются объекты OLE (например, рабочий лист MS Excel). Объект является постоянным, он не меняется с каждой записью)
Разрыв страницы	Форма разделяется на несколько экранов (или несколько страниц при печати)
Подчиненная форма/отчет	В форму добавляется информация из дополнительной таблицы, при этом пользователь может просматривать или изменять ее данные
Прямоугольник	На форме рисуется прямоугольник, чтобы сгруппировать или выделить ряд элементов
Мастера	Если эта кнопка нажата, то при создании нового сложного объекта (например, кнопки) автоматически запускается мастер по созданию

	сложных элементов управления
Поле	Добавляется поле для отображения, ввода или изменения данных
Выключатель	Добавляется прямоугольная кнопка, которую можно использовать для включения и выключения параметра. При включенном параметре кнопка нажата
Флажок	Добавляется маленькое квадратное окошко, которое можно использовать для включения и выключения параметра. При включенном параметре окошко содержит контрольную метку
Список	Добавляется элемент управления, отображающий список, из которого можно выбрать пункт
Рисунок	В форму добавляется постоянный рисунок (он не меняется при изменении записей)
Присоединенная рамка объекта	Отображаются объекты OLE, хранящиеся в записях таблицы, такие как фотографии служащих. При просмотре различных записей объект будет меняться
Набор вкладок	Позволяет разделить форму на отдельные вкладки
Линия	Дает возможность нарисовать на форме отдельную прямую линию для разделения или выделения элементов управления
Другие элементы	Добавляются из списка дополнительных элементов управления, поставляемых с Ms Office. Эти элементы отображаются в меню, появляющемся при нажатии на эту кнопку

Элементы управления можно условно разделить на присоединенные, свободные и вычисляемые.

Присоединенный элемент управления - элемент управления, в котором выводится значение поля базовой таблицы или запроса. Имя поля таблицы или запроса задается в качестве значения свойства "Данные" элемента управления. Присоединенный элемент управления дает возможность просматривать или менять значения в связанном поле и является наиболее распространенным типом элемента управления.

Свободный элемент управления - элемент управления, не связанный с полем. Свободные элементы управления используются для вывода информации, такой как инструкции по работе с формой или рисунки, созданные в других приложениях. Например, линии и прямоугольники являются свободными элементами управления. С помощью свободных элементов управления (например, кнопок или переключателей) можно также организовать обработку некоторых действий пользователя.

Вычисляемый элемент управления - элемент управления, в котором выводится результат расчета выражения, а не сохраненные данные. Значение элемента управления пересчитывается при каждом изменении выражения.

Любой элемент управления в форме имеет **свойства**. Свойства элемента управления определяют структуру, внешний вид и поведение элемента, а также характеристики содержащихся в нем данных. Свойства выбранного объекта отображаются (и могут изменяться) в окне свойств (см. рисунок 7.3.2).

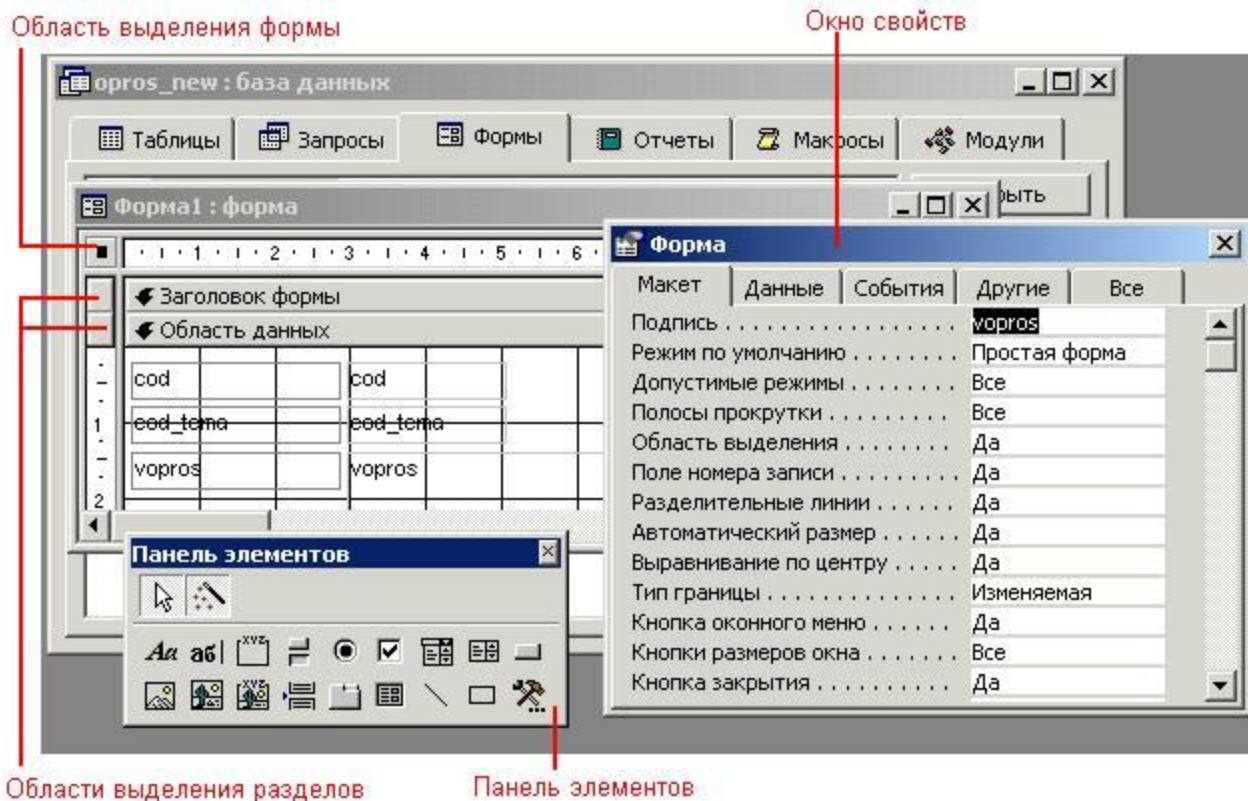


Рисунок 7.3.2 - Форма в режиме конструктора

Изменение значений свойств формы или любого элемента управления формы, а также добавление новых элементов происходит в режиме конструктора.

Кнопочные формы

Как правило, базы данных состоят из большого количества форм и отчетов. Для более простого перемещения по базе данных часто используются кнопочные формы. На кнопочную форму помещаются кнопки, при нажатии которых открываются формы или отчеты (или открываются другие кнопочные формы, с помощью которых открываются дополнительные формы и отчеты), осуществляется выход из MS Access или настраивается сама кнопочная форма.

Кнопочная форма является более простым и предпочитаемым пользователями интерфейсом базы данных по сравнению с окном базы данных. Она предназначена главным образом для тех, кто пользуется БД, предпочитая не работать над ее конструкцией (т.е. для конечного пользователя, а не для разработчика БД).

Чтобы создать кнопочную форму, удобно использовать диспетчер кнопочных форм. Для этого необходимо выполнить команду "Сервис"→"Настройка"→"Диспетчер кнопочных форм". С помощью этой надстройки автоматически создается удобная и функциональная форма, для выполнения основных действий с БД. В диалоговом окне "Диспетчер кнопочных форм" можно создавать, изменять и удалять страницы кнопочной формы. Пример кнопочной формы представлен на рисунке 7.4.1.

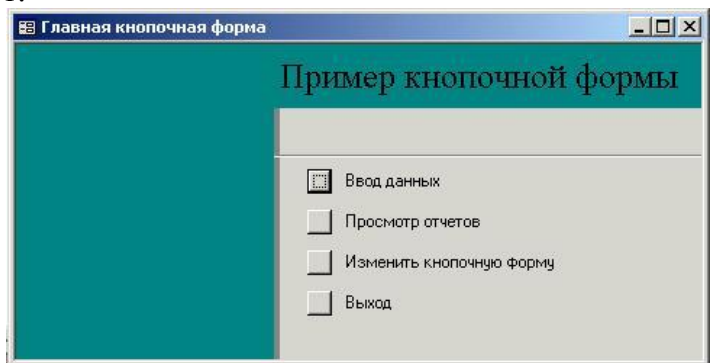


Рисунок 7.4.1 - Пример кнопочной формы

При создании кнопочной формы с помощью диспетчера кнопочных форм MS Access создает таблицу "Элементы кнопочной формы", которая описывает текст и действия кнопок формы. Если затем изменить кнопочную форму в режиме конструктора формы, приложение может перестать работать. Если потребуется значительно изменить кнопочную форму, лучше всего создать ее заново и назначить стартовой.

План-конспект занятия №11

Тема: Способы создания многотабличных форм

Цель занятия: изучить способы создания многотабличных форм.

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить назначение многотабличных форм
2) изучить способы создания многотабличных форм;

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1 , ПК 2.1.

Самостоятельная работа студента:

- Составление опорного конспекта по теме (ОК 4.2.1)

Способы создания многотабличных форм

Составная многотабличная форма создается для работы с несколькими взаимосвязанными таблицами. Она может быть создана в режиме Конструктора или Мастера форм. Наиболее технологичным является первоначальное создание формы с помощью Мастера с последующей доработкой в режиме Конструктора.

Мастер форм предоставляет возможность выбрать включаемые в форму поля из нескольких взаимосвязанных таблиц и на их основе создает многотабличную форму.

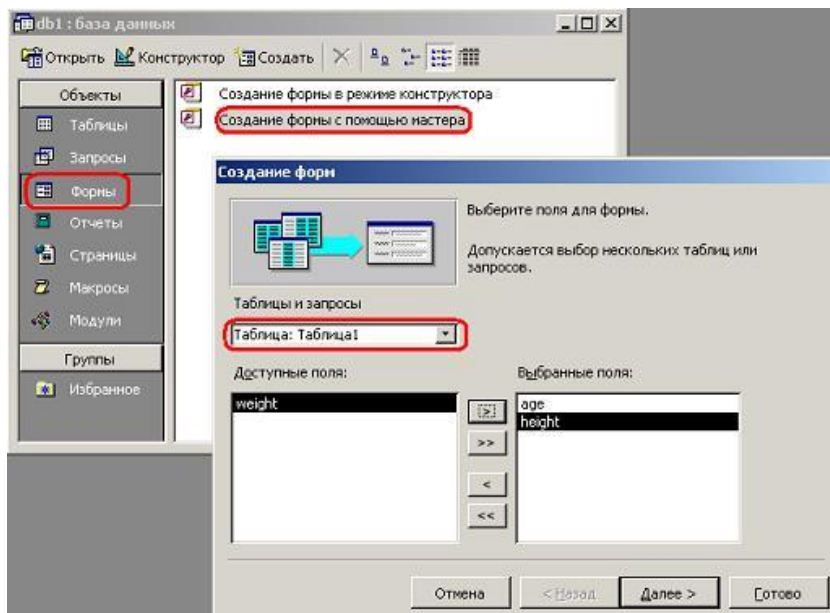
При использовании Мастера форм подчиненная форма строится только для таблицы, подчиненной по отношению к таблице - источнику основной части. Можно создать составную форму, включающую несколько подчиненных форм стандартного вида.

Мастер форм позволяет создать подчиненные формы, которые не включаются непосредственно в основную форму. При этом в форму включается лишь кнопка, представляющая связанную форму. Этот способ построения удобен в сложных формах, а также в случае, когда пользователю нет необходимости постоянно видеть связанные формы.

Составная форма, построенная Мастером, может и не включать подчиненных и связанных форм. Такая форма создается, если необходимо дополнять записи подчиненной таблицы записями из одной или нескольких главных таблиц.

Полученная с помощью Мастера составная форма может быть отредактирована, в том числе и дополнена другими включаемыми формами.

Для создания формы в окне базы данных надо выбрать вкладку «Формы» и нажать кнопку **Создать**. В окне «Новая форма» выбрать режим создания *Мастер форм* и можно в качестве источника данных основной части формы выбрать из списка таблицу или запрос.



В открывшемся в первый раз диалоговом окне «Создание форм» последовательно выбираются таблицы и из них поля, включаемые в форму.

В окне «Создание форм», открывшемся второй раз, в рамке *Выберите тип представления данных* надо выделить таблицу, которая является источником основной части формы. Если таблица была выбрана в окне «Новая форма», то она уже выделена.

- Если таблица-источник основной части формы является главной по отношению к другой таблице, тоже выбранной для формы, в окне «Создания форм» можно выбрать один из двух возможных типов подключения формы:

- * для непосредственного включения подключенной формы надо отметить *Подчиненные формы*;

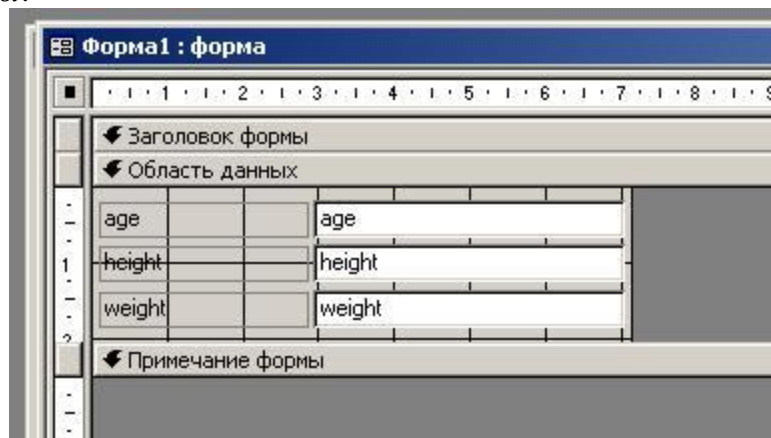
- * для включения кнопки, вызывающей связанную форму, надо выбрать *Связанные формы*.

Далее можно выбрать вид *Ленточная* или *Табличная* для подчиненной или связанной формы.

- Если таблица-источник основной части формы является подчиненной по отношению к другой таблице, тоже выбранной для формы, то создаваемая многотабличная форма не будет включать подчиненную форму. Однако эта форма будет содержать поля из главной таблицы. В окне «Создание форм» автоматически будет установлен тип формы *Одиночная форма*. Далее предоставляется возможность выбрать вид формы: *В один столбец*, *Ленточный* или *Табличный*.

В следующем окне предоставляется возможность выбрать стиль оформления (Обычный или другой), который определяет отображение надписей и значений полей в форме.

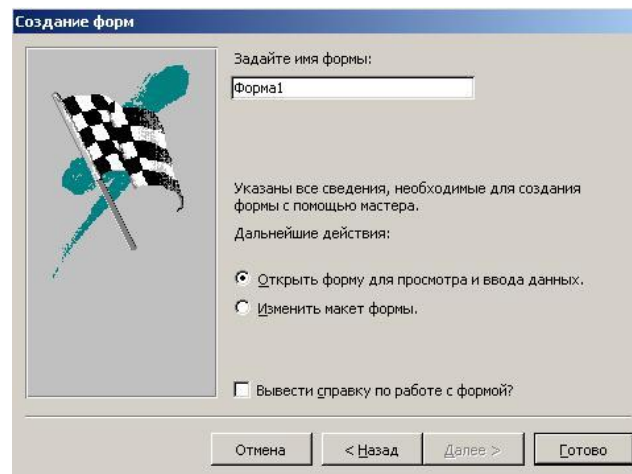
В последнем диалоговом окне «Создание форм» можно отредактировать заголовки форм и выбрать дальнейшие действия - *Открытие формы для просмотра или ввода данных* или *Изменение макета формы*.



Если был выбран вариант *Открытие формы для просмотра или ввода данных*, автоматически выводится форма с данными из таблиц, задействованных в форме. После нажатия

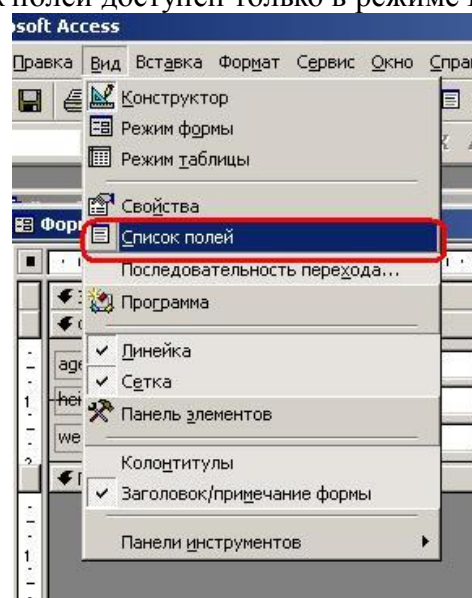
кнопки **Готово**, Мастер завершает создание формы. Сохранение форм производится автоматически. При выборе дальнейших действий - *Изменение макета формы* - форма выводится в режиме Конструктора, позволяющем произвести нужную доработку. После работы Мастера в форме размещены заданные поля из таблицы-источника основной части. Если включена подчиненная форма, то для нее выделена область, в которой указано только присвоенное ей имя.

В режиме Конструктора можно производить создание или редактирование ранее сохраненной формы.



При редактировании могут быть произведены чисто косметические правки по изменению расположения отдельных полей, их отображений и подписей. Могут быть выполнены и более радикальные изменения, включая удаление и дополнение новых полей, удаление встроенных форм, добавление новых подчиненных форм, включение полей со списком, внедрение объектов из других приложений, например, рисунков, диаграмм и т.д.

Чтобы добавить новые поля в форму, необходимо отобразить на экране список полей, доступных для внесения в форму, и перетащить требуемые поля в форму. Для отображения списка полей выполняется команда **Вид | Список полей** или нажимается соответствующая кнопка на панели Конструктора форм. Список полей доступен только в режиме Конструктора форм.

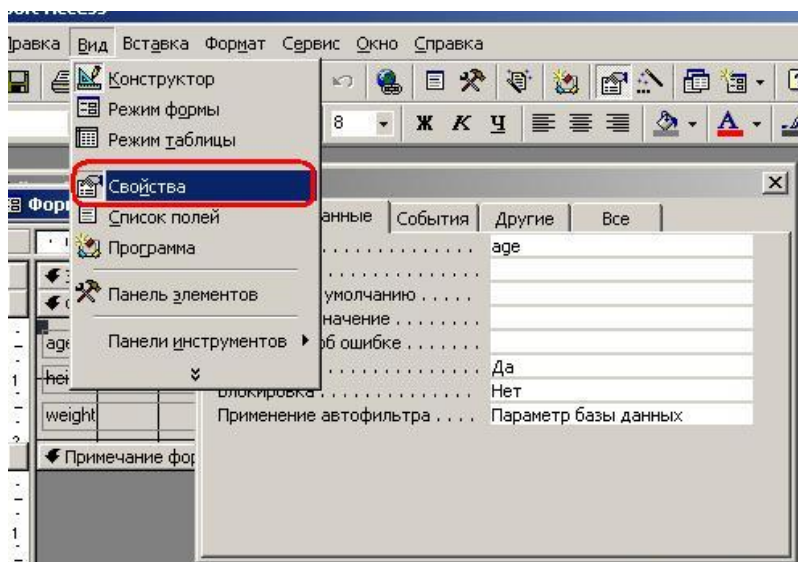


В процессе создания формы или редактирования ранее созданной формы, в том числе однотабличной, можно добавить подчиненную форму. Включение формы на этапе редактирования требует предварительного создания включаемой формы.

Для встраивания подчиненной формы необходимо в режиме Конструктора нажать кнопку панели элементов **Подчиненная форма/отчет**. Затем установить курсор на место размещения создаваемого объекта в форме, нажать кнопку мыши и, не отпуская ее, растянуть рамку подчиненной формы до нужного размера.

Для установления связи с ранее созданной подчиненной формой надо выделить внедренный объект и выбрать пункт меню **Вид | Свойства** или нажать правую клавишу мыши и в

контекстном меню выбрать **Свойства**. В окне «Подчиненная форма/отчет» надо выбрать вкладку «Данные».



В строке *Объект-источник* выбрать из списка имен форм, существующих в базе данных, ту форму, которая будет подчиненной. Имя выбранной формы вставится в рамку подчиненной формы в окне Конструктора.

Связь между основной и подчиненной формой устанавливается автоматически (*Подчиненные поля – Основные поля*) если ранее в схеме данных была определена связь таблиц, соответствующим формам. Если связи между таблицами не определены в схеме данных, необходимо выполнить действия по установке связи. В строках *Подчиненные поля* и *Основные поля* ввести имена полей, по которым связываются таблицы.

Для включения подчиненной формы может быть также использован Мастер, который позволяет не только включить, но и создать нужную подчиненную форму. Чтобы при нажатии кнопки **Подчиненная форма/отчет** запускался Мастер необходимо предварительно на панели элементов нажать кнопку **Мастера элементов**.

Особым способом включения подчиненной формы является использование метода «подтащи и брось». Таким способом можно перетащить в форму из окна базы данных готовую подчиненную форму или таблицу, для которой нужно создать подчиненную форму.

При оформлении подчиненной формы часто полезно избавиться от линий, разделяющих области заголовка, данных и примечаний. Этого можно добиться, например, удалив область заголовка и примечаний с помощью команды **Вид | Заголовок | Примечание формы**. Область данных нужно сделать больше по высоте, чем окно вывода подчиненной формы в основной.

При определении вида формы можно использовать свойства самой формы. Для перехода к просмотру и редактированию свойств формы надо установить указатель мыши в области выделения формы и нажать правую клавишу мыши. Из контекстного меню надо выбрать пункт **Свойства**.

План-конспект занятия №12

Тема : Типы меню. Работа с меню и окнами: создание, модификация, активация и удаление

Цель занятия: изучить назначение и устройство накопителей информации.

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить назначение и устройство накопителей информации
2) разъяснить структуру и стандарты накопителей;

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1 , ПК 2.1.

Самостоятельная работа студента:

- 1) Составление опорного конспекта по теме (ОК 4.2.1)

Классификация меню

Очень важно обеспечить приложение удобным интерфейсом. Access позволяет снабдить базу данных всеми необходимыми элементами интерфейса — меню и панелями инструментов, формами для ввода данных, комбинациями клавиш для быстрого вызова макросов и т. д. Все это делает работу с базой данных удобной и придает приложению законченный вид.

Меню является основной формой диалога в БД.

Существуют и используются много вариантов организации меню. В самом общем случае инструментарий для организации меню можно разбить на два класса: неспециализированные средства (по существу, это команды ввода/вывода) и специализированные средства.

Классификация меню:



Световое меню представляет собой перечень элементов меню, среди которых один текущий отображен иным цветом, чем другие пункты меню ("подсвечен"). Выбор пункта осуществляется перемещением объемного светового курсора с помощью клавиш управления курсором, фиксация пункта – нажатием клавиши, отказ от выбора – нажатием клавиши .

Все виды светового меню допускают работу с мышью, а также выбор по первой букве элемента меню (или другой назначенной букве).

Кнопочное меню широко используется в БД. Термин "кнопка" здесь означает не физическую кнопку на клавиатуре, а некоторую область экрана, которой переданы управляющие свойства.

Макрокоманда «Добавить Меню»

С помощью макрокоманды **Добавить Меню** можно создать:

- пользовательские меню в группе **Команды меню** на вкладке **Надстройки** для определенных [форм](#) или [отчетов](#) или базы данных в целом;
- пользовательское [контекстное меню](#) для формы, [элемента управления](#) формы или для отчета. Пользовательское контекстное меню заменяет встроенное контекстное меню в формах, элементах управления форм и отчетах;
- [глобальное контекстное меню](#). Глобальное контекстное меню заменяет встроенные контекстные меню для полей в таблицах и запросах в режиме таблицы, форм и отчетов, за исключением добавленных пользовательских контекстных меню в формах, элементах управления форм и отчетах.

Макрокоманда **Добавить меню** имеет следующие аргументы.

Аргумент макрокоманды	Описание
Имя меню	Имя раскрывающегося меню, добавляемого в группу Команды меню на вкладке Надстройки . Введите имя меню в поле Название меню в разделе Аргументы макрокоманды окна построителя макросов. Этот аргумент является обязательным для меню на вкладке Надстройки . В пользовательских контекстных меню и глобальных контекстных меню он игнорируется. Чтобы создать клавишу доступа , позволяющую использовать клавиатуру для выбора меню, перед буквой клавиши доступа введите амперсанд (&). Эта буква будет подчеркнута в имени меню на вкладке Надстройки .

Имя [группы макросов](#), которая содержит макросы для команд меню. Этот аргумент является обязательным.

Имя макроса меню

Если макрос, содержащий макрокоманду **ДобавитьМеню**, запускается в [библиотечной базе данных](#), приложение Microsoft Office Access 2007 выполняет поиск группы макросов с этим именем только в текущей базе данных.

Текст строки состояния

Текст, который отображается в [строке состояния](#) при выборе меню. Этот аргумент игнорируется в пользовательских контекстных меню и глобальных контекстных меню.

Создание меню в Access

Для создания меню в Access проделайте следующие действия:

1. Откройте окно **Настройка** (Customize), например, щелкнув правой кнопкой мыши по панели инструментов и выбрав команду **Настройка** (Customize). Таким способом осуществляется переход в режим настройки панелей команд.
2. По умолчанию открывается вкладка **Панели инструментов** (Toolbars). Если у вас открылась другая вкладка, раскройте именно эту вкладку. Пока в списке на этой вкладке отображаются только стандартные панели команд. К этому списку нужно добавить новую панель команд типа **Строка меню** (Menu Bar). Нажмите кнопку **Создать** (New).
3. В диалоговом окне, запрашивающем имя создаваемой панели команд, введите название: **Главное меню** (рис. 12.1) и нажмите кнопку **ОК**.

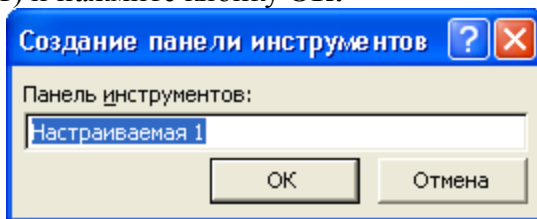


Рис. 12.1. Создание новой панели команд

4. Появится маленькая пустая панель, которая по умолчанию является панелью инструментов. Чтобы превратить ее в строку меню, необходимо изменить ее тип. Для этого выделите элемент **Главное меню** в списке на вкладке **Панели инструментов** (Toolbars) (он добавляется в конец списка) и нажмите кнопку **Свойства** (Properties). Откроется диалоговое окно **Свойства** панели инструментов **Properties** (рис. 12.2).
5. В раскрывающемся списке **тип** (Type) выберите значение **Строка меню** (Menu Bar).

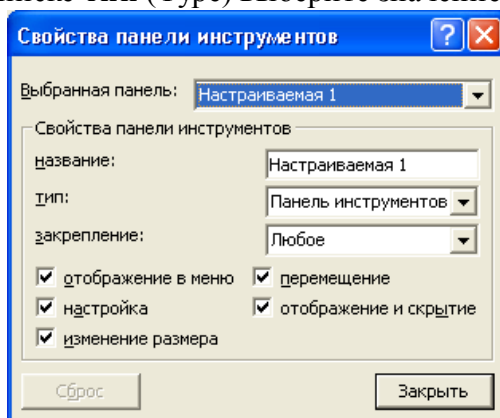


Рис. 12.2. Диалоговое окно **Свойства панели инструментов**

Кроме типа, для данной панели команд можно задать еще ряд свойств:

- **закрепление** (Docking). Можно задать расположение панели команд на экране: в любом месте, только по горизонтали, только по вертикали или вообще запретить его изменять.
- **отображение в меню** (Show on Toolbars Menu). Установите этот флажок, если вы хотите, чтобы название этой панели команд отображалось в списке, появляющемся при выполнении команды меню **Вид, Панели инструментов** (View, Toolbars).

- **настройка** (Allow Customizing). Сброс этого флажка запретит изменение данной панели команд.
- **изменение размера** (Allow Resizing). Если этот флажок не установлен, изменение размеров данной панели команд будет не возможно.
- **перемещение** (Allow Moving). Сбросьте этот флажок, если нужно запретить перемещение панели команд по экрану.
- **отображение и скрывтие** (Allow Showing/Hiding). Когда этот флажок сброшен, пользователь не будет иметь возможности скрывать и снова отображать панель команд на экране.

Сейчас мы не будем менять установленные по умолчанию значения и, изменив только тип, закроем это окно.

Пока наша строка меню пустая. Теперь в нее можно добавить меню — списки команд, раскрывающиеся при выборе меню в строке меню или на панели инструментов. Чтобы добавить меню на панель команд:

1. Раскройте вкладку **Команды** (Commands) в диалоговом окне **Настройка** (Customize).
2. Выделите в списке **Категории** (Categories) значение **Новое меню** (New Menu).
3. Перетащите элемент **Новое меню** (New Menu) из списка **Команды** (Commands) на новую панель.

Теперь вы увидите на ней элемент **Новое меню** (New Menu) (рис. 12.3).

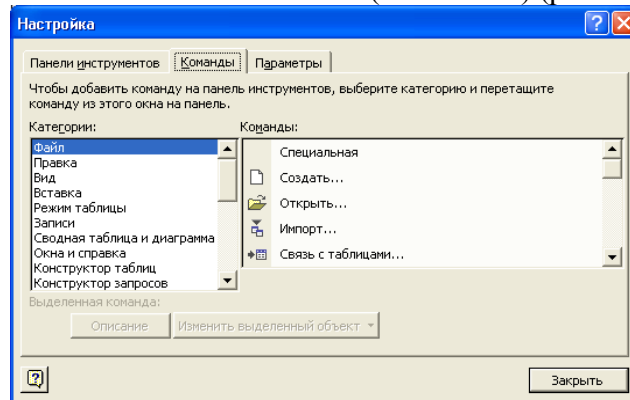


Рис. 12.3. Создание пользовательского меню

Изменить название созданного меню можно в окне свойств. Для этого необходимо щелкнуть правой кнопкой мыши по меню и выбрать в контекстном меню команду **Свойства** (Properties). Появится диалоговое окно **Свойства элемента Главное меню** (Control Properties) (рис. 12.4).

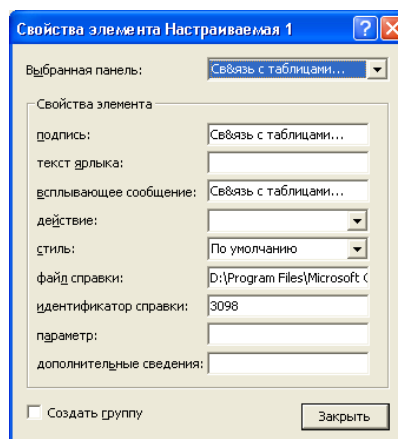


Рис. 12.4. Диалоговое окно **Свойства элемента Главное меню**

В этом окне можно задать следующие значения:

- В поле **подпись** (Caption) введите название меню: **Клиенты**.
- В поле **всплывающее сообщение** (ScreenTip) можно ввести текст всплывающей подсказки.
- Поле **действие** (On Action) следует оставить пустым. Обычно оно определяет действие, выполняемое при выборе данного элемента панели команд. При выборе меню **Клиенты**

должен раскрываться список команд данного меню. Это будет выполняться автоматически, т. к. в качестве элемента строки меню мы добавили меню.

- Поле **стиль (Style)** позволяет определить вид элемента панели команд, в данном случае — меню. Раскрывающийся список имеет следующие значения:
 - **По умолчанию (Default Style)** — на элементе меню выводятся одновременно значок и надпись;
 - **Только текст (всегда) (Text only (Always))** — и на элементе меню и на кнопке выводится только надпись;
 - **Только текст (в меню) (Text only (In Menus))** — выводится надпись на элементе меню и значок на кнопке;
 - **Рисунок и текст (Image and Text)** — и на элементе меню и на кнопке выводится как значок, так и надпись.
- Два поля — **файл справки (Help File)** и **идентификатор справки (HelpContextID)** — позволяют связать с данным элементом панели команд раздел из справочной системы приложения, который будет отображаться при нажатии клавиши <F1>, если данный элемент выделен.
- Поле **параметр (Parameter)** связано с полем **действие (On Action)** и определяет значение аргумента функции, если в поле **действие (On Action)** определен вызов функции. В данном случае оно тоже остается пустым.
- В поле **дополнительные сведения (Tag)** можно ввести информацию, которая может быть доступна из процедуры VBA.
- Установленный флажок **Создать группу (Begin a Group)** означает, что данный элемент панели команд является началом группы и слева от него (в этом случае в строке меню) будет отображаться разделительная линия.

Закройте окно **Свойства элемента меню** и будем добавлять в меню команды. Для этого можно использовать ту же вкладку **Команды (Commands)** диалогового окна **Настройка (Customize)**. Выбрав категорию, а затем команду из выбранной категории, можно перетащить ее в данное меню. Например:

1. Выберите категорию **Все формы (All Forms)**.
2. Перетащите в меню те формы, которые мы использовали при создании кнопочной формы в предыдущем разделе: "Клиенты" (Customers), "Наклейки для клиентов" (Customer Label Dialog), "Телефоны клиентов" (Customer Phone List), "Заказы клиентов" (Customer Orders) (рис. 12.5).

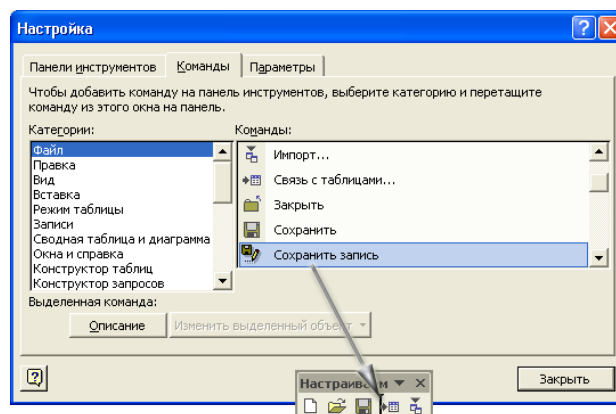


Рис. 12.5. Добавление команд в меню

Создание и удаление контекстных меню

Контекстное меню — это меню специального вида, которые появляются при нажатии правой кнопки мыши. Обычно контекстные меню создаются на последнем этапе разработки приложения. Создание контекстных меню очень похоже на создание обычных строк меню. Только в окне **Свойства панели инструментов** (Toolbar Properties) (см. рис. 14.24) в раскрывающемся списке **Тип** (Type) должно быть выбрано значение **Контекстное меню** (Popup). При этом пользователь получает предупреждение, что данное контекстное меню исчезнет с экрана. Чтобы посмотреть или изменить контекстные меню, нужно выбрать (установить флажок) **Контекстные меню** (Shortcut Menus) в списке панелей команд на вкладке **Панели инструментов** (Toolbars) диалогового окна настройки панелей команд. Тогда на экране появится панель контекстных меню. Создаваемые пользователем контекстные меню отображаются в меню **Настраиваемый** (Custom) (рис. 12.6).

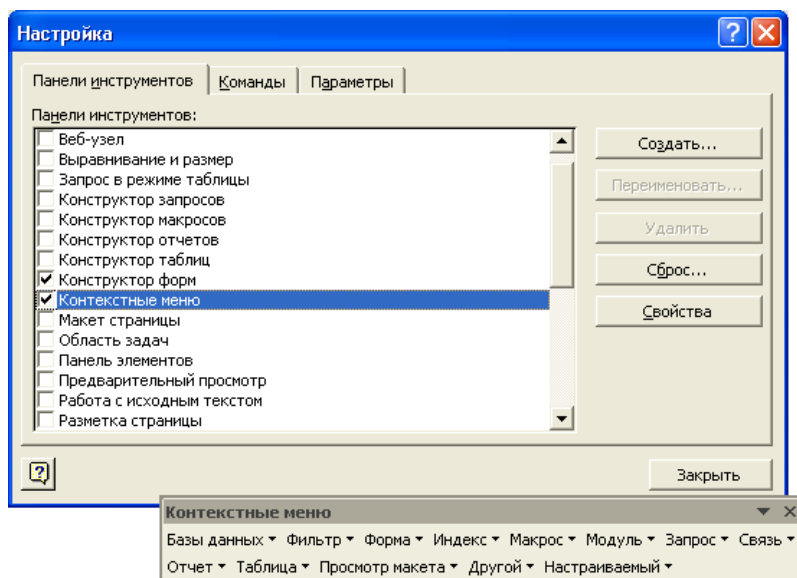


Рис. 12.6. Панель контекстных меню

Добавлять команды в контекстное меню нужно точно так же, как в обычное меню. Чтобы удалить контекстное меню:

1. Удалите все команды из этого меню, перетаскивая их по очереди на свободное место в окне Microsoft Access.
2. Раскройте вкладку **Панели инструментов** (Toolbars) в диалоговом окне **Настройка** (Customize) и нажмите кнопку **Свойства** (Properties).
3. В раскрывающемся списке **Выбранная панель** (Selected toolbar) выберите нужное контекстное меню.
4. В раскрывающемся списке **Тип** (Type) выберите элемент **Панель инструментов** (Toolbar), чтобы изменить тип удаляемого объекта, и нажмите кнопку **Закрывать** (Close).
5. Выделите в списке **Панели инструментов** (Toolbars) соответствующее название панели команд и нажмите кнопку **Удалить** (Delete).

План-конспект занятия №13

Тема: Кнопочные формы

Цель занятия: изучить назначение и способы создания кнопочной формы

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить назначение кнопочной формы
2) изучить способы создания кнопочной формы

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1.

Составление сводной таблицы (ОК 3.1.2), составление опорного конспекта (ОК2.1.1)

Создание главной кнопочной формы

Главная кнопочная форма создается с целью навигации по базе данных, т.е. она может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчетов.

Запросы и таблицы не являются элементами главной кнопочной формы. Поэтому для создания кнопок Запросы или Таблицы на кнопочной форме можно использовать макросы. Сначала в окне базы данных создают макросы «Открыть Запрос» или «Открыть Таблицу» с уникальными именами, а затем в кнопочной форме создают кнопки для вызова этих макросов.

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). Необходимо отметить, что на подчиненных кнопочных формах должны быть помещены кнопки возврата в главную кнопочную форму.

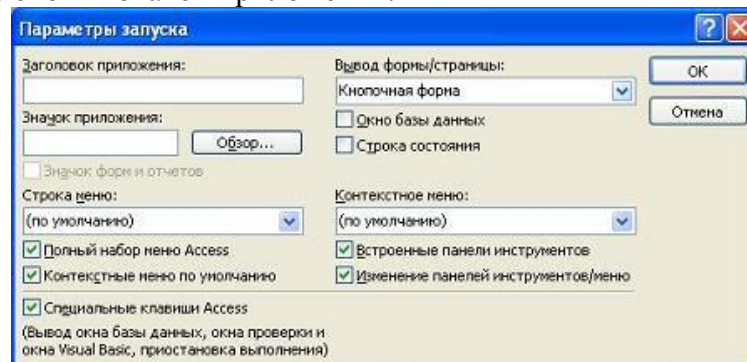
Технология создания кнопочных форм следующая:

- 1) создать страницу главной кнопочной формы (ГКФ);
- 2) создать необходимое количество страниц подчиненных кнопочных форм (например, формы для ввода данных, для отчетов, для запросов и т.д.);
- 3) создать элементы главной кнопочной формы;
- 4) создать элементы для кнопочных форм отчетов и форм ввода или изменения данных;
- 5) создать макросы для запросов или для таблиц с уникальными именами;
- 6) создать элементы для кнопочных форм запросов или таблиц.

Структура кнопочных форм может быть представлена в следующем виде.



Для того чтобы эта форма отображалась при запуске базы данных, необходимо выполнить команду Сервис/Программы запуска, и в открывшемся окне выбрать "Кнопочная форма" из раскрывающегося списка, кроме того, надо снять флажки Окно базы данных и Строка состояния. Можно также ввести заголовок и значок приложения.



План-конспект занятия №14

Тема: Формирование и вывод отчетов

Цель занятия: изучить назначение и способы создания отчетов

Вид занятия: теоретическое занятие

Задачи:

Учебные: 1) разъяснить назначение и способы создания отчетов

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1.

Самостоятельная работа студента:

- Составление опорного конспекта(ОК 2.1.1)
- Создание сравнительной таблицы «Методы создания новой формы БД: преимущества и недостатки» (ОК 3.1.2)

<i>Метод создания новой формы базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

Отчеты MS Access

Отчет - это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести необходимые сведения в том виде, в котором требуется.

Основные сведения в отчете берется из базовой таблицы, запроса или инструкции SQL, являющихся источниками данных для отчета. Другие сведения вводятся при разработке отчета. Для создания связи между отчетом и его источником данных применяются элементы управления (так же, как в формах). Такими элементами могут быть поля, надписи, линии или другие графические объекты.

К категории отчетов в MS Access относятся и другие объекты, включая наклейки с адресами (и другие типы наклеек), создаваемые Мастером наклеек, и диаграммы, создаваемые мастером диаграмм.

Стандартные отчеты делятся на две разновидности: столбцовые и ленточные.

Столбцовый отчет больше напоминает обычную форму: каждое поле выводится в отдельной строке, в которой слева расположено название, а справа - содержимое. В зависимости от количества полей в базе каждая запись может располагаться на одном или нескольких листах.

Ленточный отчет построен так, что каждое поле расположено в отдельном столбце, а каждая запись представлена одной строкой. Такой отчет напоминает режим таблицы. На различных уровнях структуры отчета могут находиться итоговые значения (промежуточные суммы, средние значения и т.д.).

В режиме конструктора отчеты делятся на несколько разделов (см. рисунок 8.1.1). Информация, содержащаяся в каждом разделе, выводится в определенной позиции печатного отчета. Раздел *Заголовок отчета* выводится в начале первой страницы. Раздел *Примечание отчета* - в конце последней страницы. Содержимое раздела *Верхний колонтитул* выводится в верхней части каждой страницы отчета, а раздел *Нижний колонтитул* - в нижней части каждой страницы. Информация из раздела *Область данных* выводится для каждой записи, входящей в отчет.

Отчет с группировкой данных позволяет вычислить итоговые значения для групп, а также представить информацию в удобном для использования виде. Заголовок группы печатается в начале каждой новой группы записей и может содержать имя группы и другие сведения, применимые ко всей группе целиком. Примечание группы печатается в конце группы записей и обычно содержит результаты групповых вычислений.

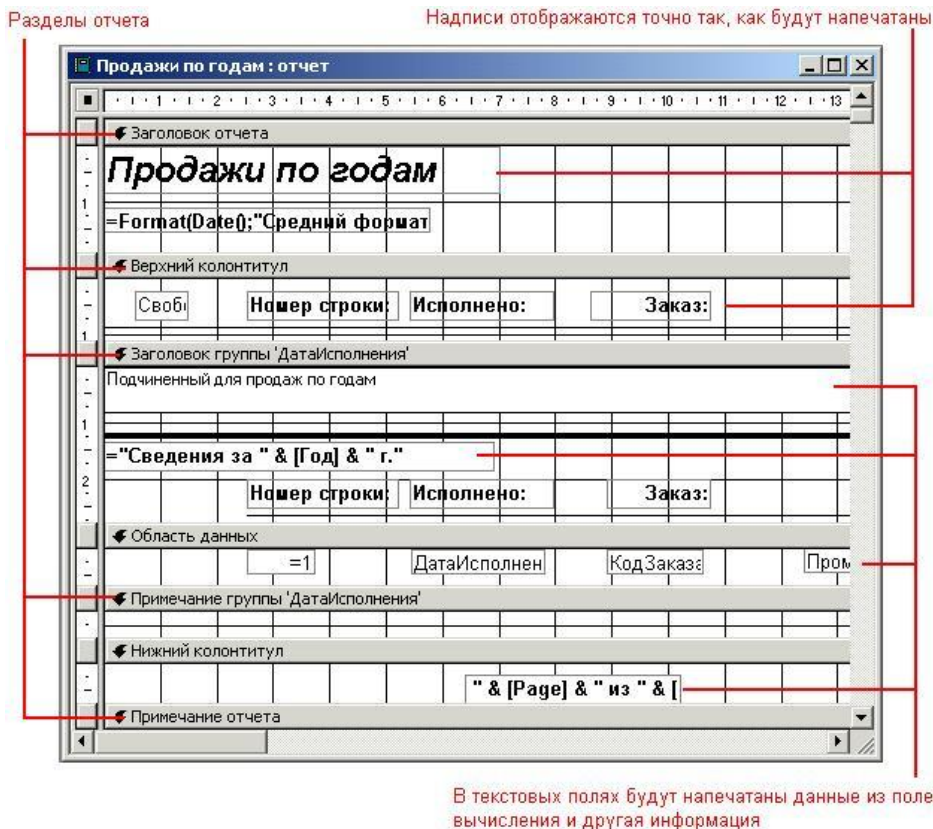


Рисунок 8.1.1 - Отчет в режиме конструктора

Создание отчетов

Для создания отчета необходимо выбрать в окне базы данных вкладку "Отчеты" и нажать кнопку "Создать", после чего откроется диалоговое окно "Новый отчет", представленное на рисунке 8.2.1.

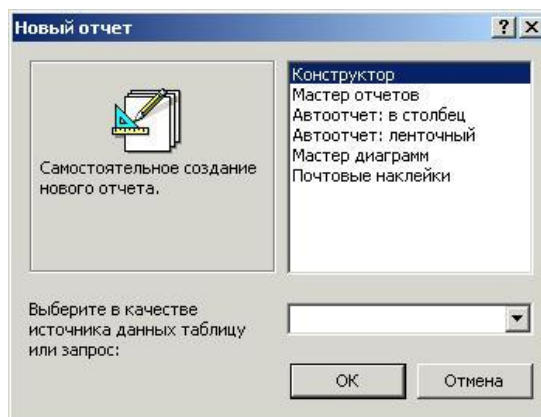


Рисунок 8.2.1 - Окно создания отчета

Вариант *Конструктор* позволяет создавать отчеты "с нуля" в режиме конструктора.

Мастер отчетов представляет собой наиболее простой и гибкий способ разработки отчетов. При этом отчеты, созданные с помощью Мастера отчетов обладают более широким диапазоном возможностей при выборе полей и проектировании отчета, чем автоотчеты.

Варианты *Автоотчет: в столбец* и *Автоотчет: ленточный* предназначены для быстрого создания столбцовых и ленточных отчетов. В автоотчеты включаются *все* поля, принадлежащие таблице или запросу из источника данных отчета.

Мастер диаграмм предназначен для создания диаграмм средствами программы Microsoft Graph/

Вариант *Почтовые наклейки* предназначен для создания почтовых и других наклеек.

Часто требуется изменить внешний вид всего отчета или его части. Например, довольно часто MS Access не может разместить имя поля по ширине столбца и требуется изменить шрифт или его размер. Для изменения внешнего вида одного элемента управления, например надписи,

необходимо его выделить и на панели инструментов *Форматирование* выбрать другой шрифт, размер шрифта или другие параметры. Чтобы изменить весь отчет, требуется нажать кнопку *Автоформат* на панели инструментов и выбрать новый внешний вид отчета.

План-конспект занятия №15

Тема : Итоговые вычисления в отчетах

Цель занятия: изучить способы создания итоговых вычислений в отчетах.

Вид занятия: теоретическое занятие

Задачи:

Учебные: 1) разъяснить способы создания итоговых вычислений в отчетах

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1; ОК 4.1.1 , ПК 2.1

Самостоятельная работа студента:

- 1) Составление опорного конспекта по теме (ОК 4.2.1)
- 2) Создание сравнительной таблицы «Методы создания отчетов в БД: преимущества и недостатки» (ОК 3.1.2)

<i>Метод создания отчетов в базе данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

Создание групповых и итоговых отчетов

Сгруппированная информация часто бывает более понятна, чем представленная в неподготовленном виде. Например, отчет, в котором продажи сгруппированы по региону, позволяет выявить тенденции, которые иначе остались бы незамеченными. Вы знали о том, что горох лучше продается в Торжке? Пользуются ли успехом помидоры в Череповце? А итоговые цифры по каждой группе избавят от необходимости длительного общения с калькулятором. Если вы печатаете одним пальцем, вам следует обратить внимание на эту статью.

Access упрощает создание групповых отчетов. После того, как выбран источник записей (например, таблица или запрос, из которых берутся данные для отчета), полдела уже сделано. Остальное поможет доделать специальный мастер. Он представляет собой функцию Access, которая выводит на экран последовательность страниц с вопросами. На основе полученных ответов мастер создает отчет. Отвечая на вопросы, вы укажете, какое поле или поля должны использоваться для группировки отчета.

Выбор источника записей

Отчет состоит из данных, полученных из таблицы или запроса, а также данных макета отчета — надписей, заголовков и рисунков. Таблица или запрос с данными для отчета называются также источником записей отчета. Перед началом создания отчета необходимо определиться с тем, какие сведения в него будут включены. В числе выбранных полей должны быть те поля, которые будут сгруппированы, или поля, по которым будет подводиться итог. Если все нужные поля находятся в одной таблице, в качестве источника записей нужно использовать эту таблицу. Если данные находятся в разных таблицах, в качестве источника записей используется запрос — он, возможно, уже существует в базе данных; в противном случае придется создать запрос специально для отчета.

Создание группового отчета с помощью мастера отчетов

Мастер отчетов позволяет значительно ускорить создание отчета. Такой отчет можно использовать в том виде, в каком он есть, а можно внести в него изменения, которые кажутся вам необходимыми.

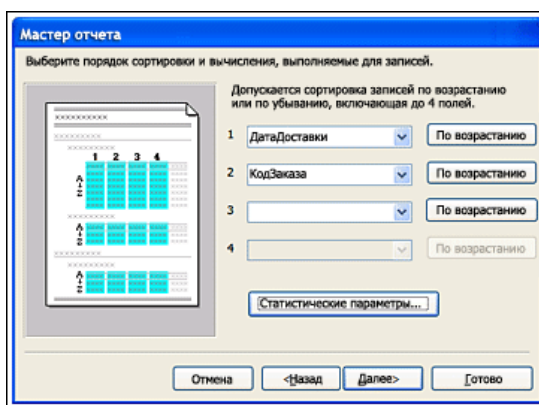
Группировка записей с помощью мастера отчетов

Группировка позволяет организовать записи в группы, например по региону или по продавцу. Группы можно вкладывать друг в друга, что позволит легко выявить связи между группами и быстро найти нужные сведения. Группировка также используется для подсчета итоговых и процентных показателей.

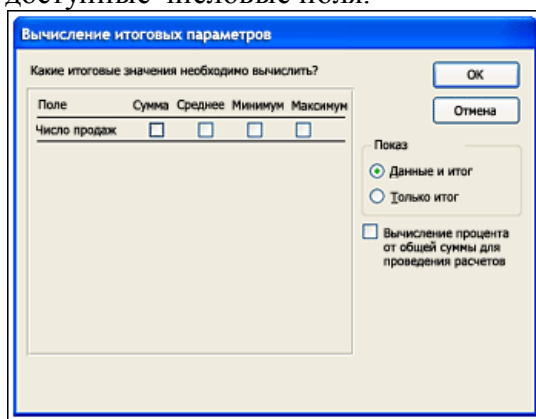
Сортировка записей и подведение итогов с помощью мастера отчетов

Мастер отчетов позволяет сортировать записи по нескольким полям (до четырех). Можно выбрать порядок сортировки: по возрастанию или по убыванию.

1. Выберите первый раскрывающийся список и выберите поле, по которому будет выполняться сортировка. Кнопка справа от списка позволяет выбирать порядок сортировки: по возрастанию или убыванию (по умолчанию выполняется сортировка **по возрастанию**). Выбрать второе, третье и четвертое поля сортировки при необходимости можно в соответствующих раскрывающихся списках.



2. Чтобы подсчитать итог по числовым полям, выберите **Итоги**. Мастер выведет на экран доступные числовые поля.



3. Чтобы включить соответствующую функцию в примечание группы, установите флажок **Sum**, **Avg**, **Min** или **Max**. Можно также установить отображение сведений и итога или только итога. В последнем случае будут показаны итоги по каждому значению даты отгрузки (если, например, установлен флажок **Sum**), но сведения о порядке будут опущены. Можно также задать отображение процентных показателей.

4. Нажмите кнопку **ОК**.

5. Выполните оставшиеся инструкции мастера отчетов. На последней странице нажмите кнопку **Готово**.

Разделы отчета

Отчет в Access состоит из нескольких разделов. Чтобы ваши отчеты несли полезную информацию, необходимо понимать, как работает каждый из них. Например, способ расчета результатов в Access определяется тем, в каком разделе размещается вычисляемый элемент управления. Далее следуют краткие сведения о типах разделов и их использовании.

- **Заголовок отчета.** Ставится один раз в начале отчета. В заголовке отчета отображаются элементы, которые могли бы размещаться на титульной странице, такие как надпись или заголовок с датой. Если в заголовке отчета размещается вычисляемый элемент управления с функцией суммирования **Sum**, рассчитанная сумма отражает итог по всему отчету. Заголовок отчета печатается до заголовка страницы.

- **Заголовок страницы.** Печатается в верхней части каждой страницы. С помощью заголовка страницы можно, например, на каждой странице разместить название отчета.

- **Заголовок группы.** Печатается в начале каждой новой группы записей. Используется для печати имени группы. Например, в отчете с группировкой по продукту в качестве заголовка группы можно указать наименование продукта. Если в заголовке группы размещается вычисляемый элемент управления с функцией суммирования **Sum**, рассчитанная сумма отражает итог по всей группе.

- **Область сведений.** Печатается один раз для каждой строки исходной записи. В разделе «Подробности» размещаются элементы управления, составляющие тело отчета.

- **Примечание группы.** Печатается в конце каждой группы записей. Здесь размещаются итоги по группе.

- **Примечание страницы.** Печатается в конце каждой страницы. Здесь размещаются номера страницы или иные сведения о странице.

- **Примечание отчета.** Печатается один раз в конце отчета. В примечании отчета печатаются итоговые сведения и иная статистика по всему отчету. Обратите внимание на то, что примечание отчета в представлении «Конструктор» идет последним, но печатается до примечания последней страницы.

По окончании создания общего вида отчета его можно изменить в режиме конструктора. С помощью [панели элементов](#) в отчете можно разместить [элементы управления](#), а с помощью [списка полей](#) — [поля](#). В [окне свойств](#) можно работать со многими [свойствами](#) и изменять их в соответствии с потребностями.

С помощью панели элементов в отчет добавляются элементы управления. Они представляют собой объекты, отображающие данные, выполняющие действия и позволяют улучшить внешний вид отчета посредством надписей и изображений. В Access предусмотрена работа с тремя типами элементов управления: присоединенными, свободными и вычисляемыми.

- **Присоединенные элементы управления.** Элемент управления, источником данных которого является поле таблицы или запроса, называется [присоединенным](#). В присоединенных элементах управления отображаются значения полей базы данных. Это могут быть текстовые значения, даты, числа, значения да/нет, рисунки и графики. Самым распространенным типом присоединенного элемента управления является текстовое поле. Например, в текстовое поле в форме, содержащее фамилию сотрудника, эти данные могут быть добавлены из поля «Фамилия» таблицы «Сотрудники».

- **Свободные элементы управления.** Элемент управления, не имеющий источника данных (поля или выражения), называется [свободным](#). В свободных элементах управления отображаются данные, линии, многоугольники и рисунки. Например, надпись с названием отчета — это свободный элемент управления.

- **Вычисляемые элементы управления.** Элемент управления, источником данных которого является выражение, а не поле, называется [вычисляемым](#). Значение такого элемента управления определяется *выражением*, которое является его источником данных. Выражение представляет собой сочетание операторов (таких как = и +), имен элементов управления, имен полей, функций, возвращающих одно значение, и постоянных величин. Например, следующим выражением рассчитывается цена товара со скидкой 25 % путем умножения значения поля «Цена за единицу» на постоянную величину (0,75).

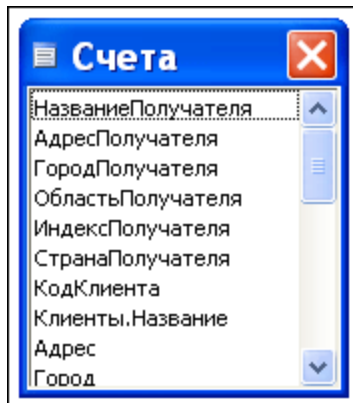
$$= [\text{Цена за единицу}] * 0.75$$

В выражении могут использоваться данные из поля исходной таблицы или запроса отчета или элемента управления, содержащегося в отчете.

Текстовое поле может быть как присоединенным, так и свободным или вычисляемым элементом управления. Если в отчете есть разные виды элементов управления, разумно сначала добавить и организовать все присоединенные элементы, особенно если их число значительно по

сравнению с остальными элементами отчета. Затем с помощью панели элементов можно добавить свободные и вычисляемые элементы управления, завершающие конструкцию отчета.

Чтобы присоединить текстовое поле к полю, нужно указать поле, из которого этот элемент будет получать данные. Чтобы создать элемент управления, присоединенный к выбранному полю, достаточно перетащить это поле из списка полей в отчет. (В списке полей отображаются поля исходной таблицы или запроса отчета. Чтобы вывести на экран список полей, выберите в меню **Вид** команду **Список полей**.) При перетаскивании поля из списка полей по умолчанию создается текстовое поле.



Другой способ присоединить поле к элементу управления — ввести имя поля в самом элементе управления или в поле **ControlSource** в окне свойств элемента управления, где задаются характеристики элемента управления, такие как имя, источник данных и формат.


Создание присоединенного текстового поля с помощью списка полей является предпочтительным по двум причинам:

- К присоединенному текстовому полю присоединена надпись, которая по умолчанию воспринимает имя поля (или заголовок, определенный для этого поля в исходной таблице или запросе) в качестве его заголовка, поэтому вводить заголовок отдельно не придется.
- Присоединенное текстовое поле наследует многие параметры поля исходной таблицы или запроса (например, свойства **Формат**, **Число десятичных знаков** и **Маска ввода**). Поэтому можно быть уверенным, что эти свойства поля останутся такими же при создании текстового поля, присоединенного к этому полю.

Если созданный свободный элемент управления нужно присоединить к полю, установите имя поля в качестве значения свойства **ControlSource** элемента. Чтобы получить подробные сведения о свойстве **ControlSource**, выполните в справке поиск по слову «ControlSource».

Добавление полей в отчет

Чтобы добавить поле в отчет, убедитесь сначала, что отображается список полей. Он содержит перечень всех полей источника записей.

- В меню **Вид** выберите **Список полей** (или нажмите кнопку **Список полей**  на панели инструментов).
- Чтобы добавить одно поле, перетащите его из списка полей в тот раздел отчета, где оно будет находиться.
- Чтобы добавить несколько соседних полей одновременно, щелкните первое поле и, удерживая нажатой клавишу SHIFT, щелкните последнее поле. Перетащите выделенные поля в отчет.
- Чтобы добавить одновременно несколько полей из разных частей списка, щелкните первое поле, а затем, удерживая нажатой клавишу CTRL, щелкните остальные поля. Перетащите выделенные поля в отчет.

При перетаскивании полей в отчет Access создает для каждого из них присоединенное текстовое поле и автоматически размещает около каждого текстового поля элемент управления «надпись».

Добавление элементов управления в отчет

Все данные отчета содержатся в элементах управления. Элемент управления — объект, отображающий данные, выполняющий действие или являющийся элементом оформления отчета.

Некоторые элементы управления присоединены к полям исходной таблицы или запроса, поэтому можно либо вводить данные в поля, либо выводить данные из них на экран. Например, вводить и отображать информацию можно с помощью текстового поля. В других элементах управления отображаются данные, которые хранятся только в макете отчета и не связаны с источником данных. Например, для отображения описаний используются надписи, а линии и многоугольники позволяют сделать отчет более привлекательным.

Некоторые элементы управления создаются автоматически, такие как присоединенное текстовое поле, которое создается при добавлении в отчет поля из списка полей. Другие элементы управления можно создать с помощью панели элементов.

Чтобы создать вычисляемый или свободный элемент управления, щелкните отчет в том месте, где должен находиться левый верхний угол элемента управления. Чтобы создать элемент управления стандартного размера, щелкните отчет один раз; чтобы создать элемент управления нужного размера, щелкните его и перетащите в отчет.

Чтобы создать присоединенный элемент управления, нужно выбрать поле из списка полей и перетащить его в отчет.

Использование окна сортировки и группировки

Данные в отчетах, таблицах и формах можно сортировать. Сортировка означает организацию записей по значениям одного или нескольких полей. Например, можно отсортировать список продуктов в алфавитном порядке. В отчетах отсортированные записи можно отделить в группы.

Группировка позволяет организовать записи в группы, например по региону или по продавцу. Группы можно вкладывать друг в друга, что позволит легко выявить связи между группами и быстро найти нужные сведения. Группировка также используется для подсчета итоговых и процентных показателей.

При печати отчета записи обычно организуются в алфавитном порядке. Например, если распечатывается список поставщиков, записи можно отсортировать по наименованию организации. При работе в режиме конструктора порядок сортировки в отчете задается в окне «Сортировка и группировка».

Чтобы открыть окно «Сортировка и группировка», выполните следующие действия:

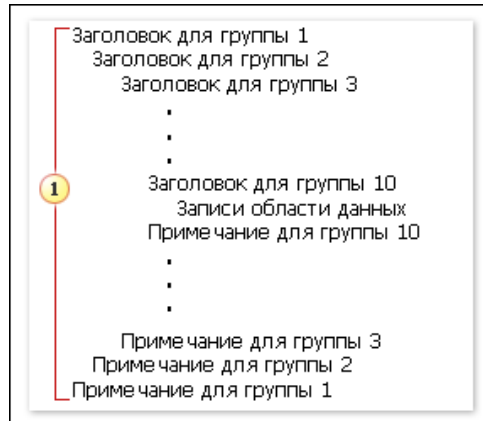
В верхней части окна «Сортировка и группировка» задается порядок сортировки записей в отчете. Сортировку можно проводить не более чем по 10 полям и выражениям, а сортировку по одному полю или выражению можно проводить несколько раз. Например, поля длиной в пять знаков можно отсортировать по возрастанию первых двух знаков и по убыванию последних трех знаков.

- **Поле/выражение.** Здесь указывается имя поля или выражение, по которым будет выполнена сортировка. Поле или выражение в первой строке столбца **Поле/выражение** задает первый уровень сортировки; поле или выражение во второй строке задает второй уровень и т. д. Например, можно отсортировать список сотрудников по фамилии, а затем — по имени.

- **Порядок сортировки.** Здесь указывается способ сортировки данных по полю или выражению. **По возрастанию** — от меньшего значения к большему. **По убыванию** — от большего значения к меньшему. Например, сортировка текстовых полей по возрастанию означает сортировку по алфавиту. Сортировка числовых полей по возрастанию означает сортировку от 0 до максимального значения.

В окне «Сортировка и группировка» можно добавлять и удалять поля и выражения, а также изменять их порядок.

Группировку можно выполнить по любым полям и выражениям, по которым выполнялась сортировка (максимум по 10). По одному полю или выражению можно группировать несколько раз. Если группировка осуществляется по нескольким полям или выражениям, Access вкладывает группы друг в друга в соответствии с их уровнем. Первое поле, по которому выполняется группировка, принадлежит первому и наиболее значимому уровню; второе поле, по которому выполняется группировка, принадлежит следующему уровню и так далее. На следующем рисунке демонстрируется принцип вложения групп в Access.



1 Каждому заголовку группы соответствует примечание группы.

После того, как указаны поля в окне «Сортировка и группировка», чтобы создать группу, достаточно добавить заголовок и примечание группы. Для добавления и удаления заголовков групп используется свойство **Заголовок группы**, а для добавления и удаления примечаний — свойство **Примечание группы**. Для каждого поля и выражения, по которому выполняется группировка, можно задать как оба этих элемента, так и один из них.

В заголовке группы обычно отображаются сведения с описанием группы в отдельном разделе в начале группы. В примечании группы обычно излагаются обобщающие сведения о группе в отдельном разделе в конце группы.

- **Свойство «Группировка».** Задаёт способ группировки значений. По каждому значению можно группировать поля следующих типов данных: текстовые, счетчики, денежные, числовые и даты и время. Текстовые поля можно группировать по первым нескольким знакам их значений. Поля счетчика, даты и времени, денежные и числовые поля можно группировать по диапазонам значений.

- **Свойство «Интервал группировки».** Задаёт интервал, действующий для значений поля или выражения, по которым выполняется группировка.

Свойства **Группировка** и **Интервал группировки** влияют на то, как записи организуются в отчете, поэтому никакие изменения в макете отчета при установке этих свойств не отображаются. Результат изменения свойств виден во время предварительного просмотра и печати отчета.

То, какие параметры Access будет отображать для свойства **Группировка**, зависит от типа данных поля, по которому выполняется группировка. В таблицах следующих разделов приведены предусмотренные параметры для каждого типа данных. Если группировка выполняется по выражению, Access выводит все параметры для всех типов данных.

Если группировка выполняется по текстовому полю, для свойства **Группировка** можно установить один из диапазонов, приведенных в следующей таблице. Если присвоить свойству **Группировка** значение **По первым знакам**, свойству **Интервал группировки** можно присвоить любое число, допустимое для значений поля, по которому выполняется группировка. Если присвоить свойству **Группировка** значение **По полному значению**, значением свойства **Интервал группировки** будет **1**.

Значение свойства

«Группировка» **Группируемые записи**

По полному значению Содержащие то же значение в поле или выражении

По первым знакам Содержащие те же первые *n* знаков в поле или выражении

Если группировка выполняется по полю даты и времени, свойству **Группировка** присваивается один из диапазонов из списка, приведенного в следующей таблице. Для всех диапазонов, кроме диапазона **По полному значению**, в качестве значения свойства **Интервал группировки** можно установить любое число, допустимое для значений поля или выражения, по которым выполняется группировка. (Значением свойства **Интервал группировки** для диапазона **По полному значению** всегда является **1**.)

Значение свойства

«Группировка»	Группируемые записи
По полному значению	Содержащие то же значение в поле или выражении
Год	Даты того же календарного года
Квартал	Даты того же календарного квартала
Месяц	Даты того же месяца
Неделя	Даты той же недели
День	Даты того же дня
Час	Время с тем же часом
Минута	Время с теми же минутами

Дополнительные сведения

При задании свойства **Группировка** для поля даты и времени записи группируются в хронологическом порядке, то есть по дате. Чтобы сгруппировать записи по тому же периоду времени за несколько лет (например, доходы за несколько лет по кварталам), используется функция **DatePart**. Дополнительные сведения см. в статье [Функция DatePart \(на английском языке\)](#)

Если группировка выполняется по полю счетчика, денежному или числовому полю, свойству **Группировка** присваивается один из диапазонов, приведенных в следующей таблице. Если свойству **Группировка** присвоить значение **Интервал**, в качестве значения свойства **Интервал группировки** задается любое число, допустимое для значений поля, по которому выполняется группировка. Если свойству **Группировка** присвоить значение **По полному значению**, значением свойства **Интервал группировки** будет **1**.

Значение свойства

«Группировка»	Группируемые записи
По полному значению	То же значение в поле или выражении
Интервал	Значения, попадающие в указанный интервал

По умолчанию Access начинает группы счетчика, денежные и числовые группы со значения 0. Например, если свойство **Группировка** имеет значение **Интервал**, а свойство **Интервал группировки** — **5**, Access сгруппирует записи следующим образом: 0–4, 5–9, 10–14 и т. д.

Добавление сумм, счетчиков и итогов в отчет

Access позволяет добавлять в отчет суммы, счетчики и итоги. Предположим, в конец отчета нужно добавить сумму объемов заказов. Для этого в примечании отчета нужно создать вычисляемый элемент управления. Источником данных вычисляемого элемента управления является выражение, а не поле. При открытии отчета с вычисляемым элементом управления Access вычисляет его значение на основе текущих данных. Например, чтобы напечатать на отчете текущую дату, в текстовое поле вводится выражение **=Date()**. **Date** — это встроенная функция, которая возвращает текущую дату по показаниям системных часов компьютера. При запуске отчета рассчитывается значение выражения в вычисляемом элементе управления, **Date()**, и текущая дата отображается в элементе управления отчета.

Использование функции «Sum»

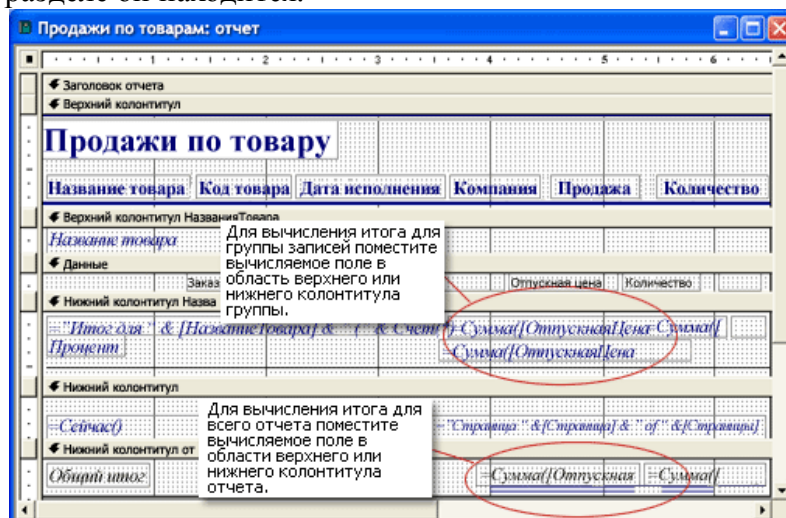
Чтобы вычислить сумму объемов заказов по всему отчету, поместите в примечании отчета текстовое поле и присвойте свойству **ControlSource** текстового поля следующее выражение:

```
=Sum([QuantityOrdered])
```

Функция **Sum** вычисляет итог по набору значений источника записей — в данном случае, столбца «Объем заказов». **Sum** — статистическая функция Access, которая позволяет вычислять итоговые значения.

Большинство итоговых значений представляют собой сумму по группе записей или по всем записям. При вычислении суммы важно то, в каком разделе отчета размещается вычисляемый элемент управления, содержащий выражение суммирования. Access определяет

способ подсчета суммы на основании места размещения элемента управления, то есть на основании того, в каком разделе он находится.



Например, чтобы вычислить итог по одной записи, в области сведений создается вычисляемое текстовое поле. В качестве примера можно взять расчет налога с продаж. Чтобы рассчитать итог по группе записей, например по каждому продукту в сгруппированном по продуктам отчете, вычисляемое текстовое поле создается в заголовке или примечании группы. Чтобы распечатать общий итог по всему отчету, вычисляемое текстовое поле создается в заголовке или примечании отчета.

Продажи по товару					
Название товара	Код заказа	Дата исполнения	Компания	Продажа	Количество
	10832	19 января 1998	Дом Азии	3 000 р.	3
	10836	21 января 1998	Серьезная торговля	30 000 р.	30
	10845	30 января 1998	БЫСТРАЯ остановка	45 000 р.	48
	10859	02 февраля 1998	Честная сделка	23 500 р.	30
	10872	08 февраля 1998	Отменные товары	15 000 р.	15
	10876	12 февраля 1998	Чудесный выбор	20 000 р.	20
	10857	27 марта 1998	Удачная покупка	8 000 р.	8
	10868	01 апреля 1998	Серьезная торговля	4 000 р.	4
	11031	24 апреля 1998	Верный способ сэкономить	20 000 р.	20
	11053	29 апреля 1998	Пижоло и другие	26 000 р.	25
	11072		Серьезная торговля	120 р.	130
	11077		Бакалейные Гремучая земля	2 000 р.	2
Итого для Удачной покупки (30 заказов)				658 710 р.	74 740
Процент				1,73%	
Свободный выбор					
Итоговые значения группы					
	10956	20 марта 1998	Деликатесы всех морей	4 000 р.	14
	10977	10 апреля 1998	Народный % НВ	8 000 р.	30
Итого для Свободного выбора (21 заказ)				93 000 р.	48,485
Процент				0,31%	
Общий итог					
Итоговые значения отчета					
				37 973 376 р.	51 317

Обратите внимание: если итоги будут подводиться в вычисляемом элементе управления, необходимо повторить выражение, записанное в вычисляемом элементе управления, в функции **Sum**. Например, если в вычисляемом элементе управления «Налог» записано выражение

$$=[\text{Цена за единицу}] * 0,085,$$

чтобы рассчитать итог по значениям элемента управления «Налог», нужно записать

$$=Sum([UnitPrice]*0,085),$$

а не

$$=Sum(Tax)$$

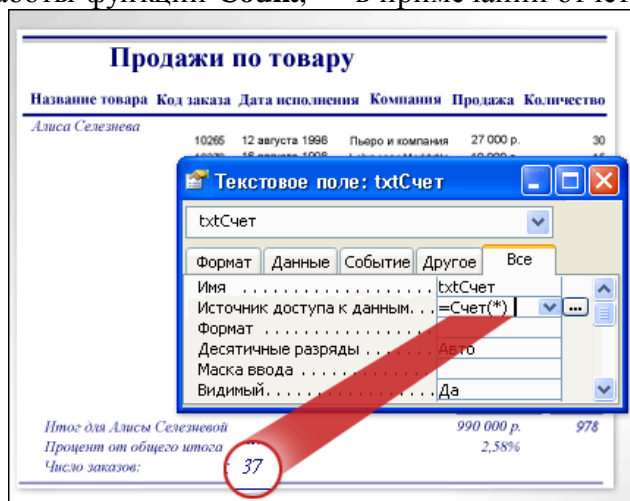
Иногда необходимо подсчитать не только сумму, но также и количество элементов. Например, в дополнение к отдельным суммам выручки может понадобиться вывести число заказов на конкретный продукт. Чтобы подсчитать заказы, можно воспользоваться функцией **Count**. Access запрашивает базу данных на языке SQL, а функция **Count** является одной из статистических функций SQL, с помощью которой можно производить вычисления по столбцам.

Использование функции «Count»

Пусть в отчет «Выручка по продуктам» нужно добавить счетчик количества заказов. Чтобы подсчитать число заказов, создается текстовое поле, а свойству **ControlSource** задается выражение, которое выглядит следующим образом:

=Count (*)

Звездочка означает, что Access будет проводить подсчет по всем строкам области сведений, но результат будет зависеть от того, где будет размещено текстовое поле. Как и при подсчете суммы, способ работы счетчика в Access зависит от того, в каком разделе отчета будет находиться текстовое поле. Если разместить текстовое поле в заголовке или примечании группы, счетчик обработает только строки этой группы. Если текстовое поле будет находиться в заголовке или примечании отчета, счетчик обработает строки по всему отчету. На следующей диаграмме выражение «Count(*)» отображается в поле свойства **ControlSource** текстового поля, а итог, полученный по результатам работы функции **Count**, — в примечании отчета.



Сохранение результатов

Сохраненный макет отчета можно будет многократно использовать повторно. Макет остается прежним, но отчет каждый раз при печати состоит из новых данных. Общую структуру отчета при необходимости можно изменить или создать новый подобный отчет на основе оригинала.

Предварительный просмотр отчета

Предварительный просмотр позволяет увидеть, как отчет будет выглядеть на печати. Можно, например, проверить выравнивание и размер столбцов данных, а также возвращает ли отчет ожидаемые данные. Предварительный просмотр можно проводить в двух режимах:

- **Режим просмотра образца.** Используется для проверки шрифта, размера шрифта и общей структуры. В режиме просмотра образца отображаются все разделы отчета и некоторые записи.
- **Режим предварительного просмотра.** Используется для просмотра отчета точно в том виде, как он будет выглядеть на печати.

В режимах предварительного просмотра и просмотра образца отчет отображается крупным планом.

Печать отчета

Отчет можно распечатать из окна предварительного просмотра, просмотра образца, в режиме конструктора или из окна базы данных. Перед печатью необходимо еще раз проверить параметры страницы (например, поля и ориентацию). Access сохраняет параметры страницы с отчетом, поэтому они устанавливаются один раз, а потом только вносятся необходимые изменения.

Отправка отчета по электронной почте

Отчет можно не передавать на бумажном носителе, а отправить по электронной почте.

План-конспект занятия №16

Тема Типы запросов в базах данных

Тип занятия: *теоретическое занятие.*

Цель занятия: познакомить с основными типами запросов в БД

Задачи:

- Учебные: 1) разъяснить назначение основных типов запросов
2) ознакомить с вариантами создания запросов

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1.

Самостоятельная работа студента:

- Составление опорного конспекта (ОК 2.1.1)
- Создание и заполнение сводной таблицы «Типы запросов в БД» (ОК 3.1.2)

Типы запросов в базе данных	Назначение	Способ создания

Запросы MS Access

Основное назначение таблицы в MS Access - хранение внесенных данных. При этом она играет роль пассивного хранилища, т. е. сама не может инициировать ни процесс запроса новых записей, ни передачу информации конечному потребителю. Для этого в СУБД служат другие объекты - запросы.

Все запросы делятся на запросы-выборки и запросы-действия.

После выполнения *запроса-выборки* СУБД создает виртуальную таблицу, в которую заносит выбранную информацию и хранит ее до тех пор, пока сгенерированная таблица не будет закрыта. Когда этот запрос закрывается, полученная таблица уничтожается, освобождая занимаемую память.

В отличие от запроса-выборки, *запрос-действие* вносит изменения в существующие данные. При помощи запроса-действия можно автоматически создать новую таблицу, внести данные в уже имеющуюся таблицу, а также удалить или изменить какой-либо набор записей из существующей таблицы.

Любой вновь создаваемый запрос в MS Access автоматически считается запросом-выборкой. В том случае, когда пользователю или разработчику нужно получить запрос-действие, следует специально это указать. Это помогает избежать случайных действий с данными, которые нельзя отменить.

Фактически, запрос в MS Access является инструкцией на языке *SQL* (Structured Query Language), который является на сегодняшний день основным средством составления запросов в наиболее распространенных реляционных СУБД. Посредством механизма SQL становится возможным как обращаться к содержимому баз MS Access из других приложений (в том числе и

приложений от других производителей), так и получать данные для MS Access из внешних приложений, поддерживающих SQL.

Запросы в MS Access доступны в трех режимах: табличном, в котором запрос выглядит так же, как обычная таблица, режиме конструктора, где запрос предстает в виде схемы связанных объектов, и в виде инструкций SQL.

Краткие сведения об SQL

Структурированный язык запросов (Structured Query Language) - это язык запросов, ориентированный специально на реляционные базы данных. SQL был разработан в 70-е годы фирмой IBM и стандартизован ANSI (American National Standards Institute). Поэтому его часто также называют ANSI-SQL. Одно из достоинств SQL - его широкое распространение. При этом следует помнить, что на различных СУБД диалект SQL может отличаться от стандарта (ANSI SQL89 и ANSI SQL92). Диалект SQL, используемый в MS Access, называется Jet SQL.

Небольшой набор команд SQL и определенный синтаксис составления выражений позволяет совершать сколь угодно сложные операции. Самой распространенной из них является выборка связанных данных из одной или нескольких таблиц при помощи оператора SELECT, однако в случае необходимости SQL может вносить информацию, корректировать хранящиеся записи и создавать новые объекты базы данных. SQL применяется не только в MS Access, но и во всех современных СУБД.

SQL-запросы построены на лексическом диалекте, основанном на семантике обычного языка (английского). При создании языка предполагалось, что конструкцию на нем сможет написать любой неподготовленный пользователь, руководствуясь исключительно своими потребностями и исходя из существующей структуры данных. Среди базовых инструкций, на которых основываются запросы, стоит выделить следующие наиболее распространенные команды:

1. SELECT - предназначается для выборки из базы данных набора записей, соответствующих указанному критерию. В качестве входных параметров инструкции передаются названия полей таблиц, которые должны быть внесены в результирующий набор.
2. UPDATE - служит для редактирования записей (одной или нескольких). Пользователь указывает названия полей таблиц и их новые значения.
3. DELETE - инструкция, полностью удаляющая из базы данных все указанные записи (а не только значения отдельных полей).
4. CREATE - позволяет создавать новые объекты базы данных.

Вспомогательные конструкции языка, например, WHERE задает критерий уточняющего поиска среди отобранных записей.

В языке запросов, реализованном в Microsoft Access, поддерживаются четыре дополнительных расширения: TRANSFORM (позволяет строить перекрестные запросы), IN (создание связи с удаленной базой данных), DISTINCTROW (определение запроса-объединения с потенциальной возможностью объединения данных), WITH OWNER-ACCESS OPTION (позволяет создавать специализированные запросы, которые может выполнять пользователь, не имеющий права доступа к таблицам, которыми оперирует данный запрос). В качестве функции могут быть использованы итоговые функции SQL, например такие, как AVG, COUNT, MAX, MIN, STDEV, STDEVP, SUM, VAR или VARP. Кроме того, в качестве функции могут применяться встроенные функции MS Access и встроенного языка VBA.

Создание запросов

Для того чтобы создать запрос-выборку, нужно перейти на вкладку "Запросы" в окне базы данных и нажать кнопку "Создать", в результате откроется диалоговое окно, представленное на рисунке 6.3.1.

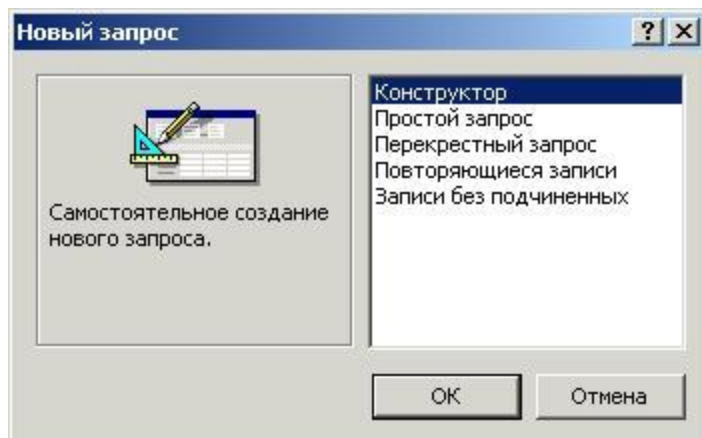


Рисунок 6.3.1 - Создание нового запроса

При создании запроса с помощью *Конструктора* разработчику необходимо выбрать таблицы или запросы, содержащие нужные данные и заполнить бланк запроса. *Бланк запроса* расположен в нижней части окна конструктора и содержит описание запроса в табличной форме (см. рисунок 6.3.2). Этот вариант следует выбирать для создания запроса "с нуля" в режиме конструктора.

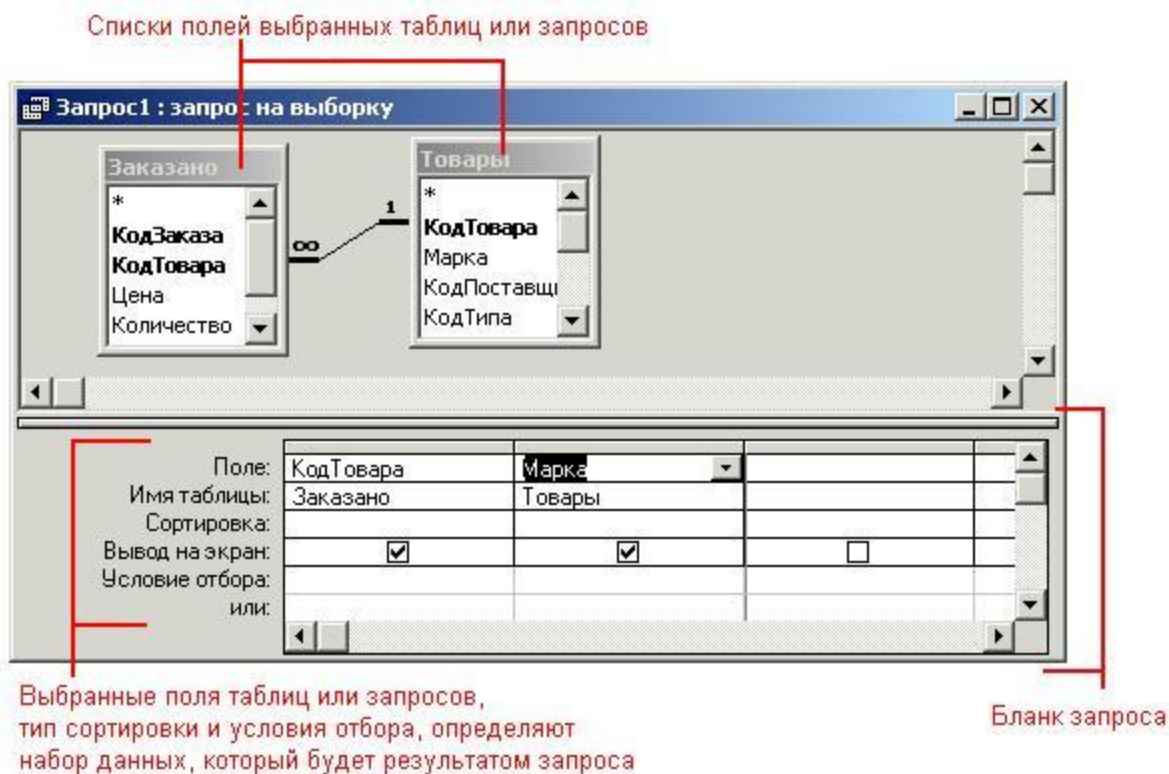


Рисунок 6.3.2 - Режим конструктора для запроса

Каждая колонка в бланке запроса соответствует одному полю. Строки "Поле" и "Имя таблицы" содержат списки, которые позволяют определить нужное поле. Таблица бланка запросов расширяется вправо автоматически при добавлении в запрос новых полей. Контекстное меню, связанное с бланком запроса, позволяет скрывать или, наоборот, показывает строки "Имя таблицы" и "Групповая операция". Под "*Групповой операцией*" понимается обработка и обобщение значений данного поля при помощи определенной функции (Например функция "Count" позволяет подсчитать число записей).

Если воспользоваться вариантом *Простой запрос*, то откроется диалоговое окно Мастера простых запросов, в котором следует выбрать участвующие в запросе поля из любых таблиц или запросов, входящих в базу данных. В последнем диалоговом окне Мастера пользователю предлагается выбор выполнить запрос или просмотреть его структуру в режиме конструктора. Основное преимущество этого способа заключается в том, что Мастер автоматически заполняет

бланк запроса). Если полученный запрос не соответствует требованиям, в него можно внести изменения в режиме конструктора.

Перекрестный запрос - это операция построения таблицы для вычисления итоговых значений на основе существующей таблицы или запроса. Перекрестный запрос создается в предположении, что исходная таблица содержит необходимые данные для формирования заголовков строк и столбцов новой таблицы. Например, из всего объема продаж товаров и услуг за всю историю предприятия необходимо выбрать показатели по конкретным позициям за конкретный год, несколько определенных месяцев или кварталов.

В том случае, когда из всего массива данных одной или нескольких таблиц нужно выбрать лишь те данные, которые повторяются, следует воспользоваться вариантом "*Повторяющиеся записи*". Например, выбрать те товары, количество продаж которых "больше чем один".

Если нужно найти среди всех записей базы данных те, которые не имеют подчиненных, следует выбирать вариант, "*Записи без подчиненных*". Например, это достаточно типовая ситуация для любой базы данных, фиксирующей торговые операции по конкретным клиентам или по конкретным товарам. Незачем хранить информацию по конкретной поставке товара, если сам товар уже давно продан. Также незачем хранить в списке "постоянных клиентов" фамилии тех людей, которые уже длительное время не осуществляли никаких покупок. Очевидно, что чем меньше записей приходится просматривать, тем быстрее работает вся СУБД.

Запрос с параметрами - это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, например условие для возвращения записей или значение, которое должно содержаться в поле. Можно создать запрос, в результате которого выводится приглашение на ввод нескольких данных, например, двух дат. В результате будут возвращены все записи, находящиеся между указанными двумя датами. Также запросы с параметрами удобно использовать в качестве основы для форм и отчетов. Например, на основе запроса с параметрами можно создать месячный отчет о доходах. При выводе данного отчета, на экране появится приглашение ввести месяц, доходы которого интересуют пользователя. После ввода месяца на экране будет представлен требуемый отчет. Можно создать специальную форму или диалоговое окно, которое вместо диалогового окна запроса с параметрами будет выводить приглашение на ввод параметров запроса.

Существует ряд вычислений, которые можно выполнить в запросе, например, найти сумму или среднее по значениям одного поля, перемножить значения двух полей или вычислить дату, отстоящую на заданное количество месяцев от текущей даты.

Результаты вычислений, выводимые в поле, не запоминаются в базовой таблице. Вместо этого, вычисления снова производятся всякий раз, когда выполняется запрос, поэтому результаты всегда представляют текущее содержимое базы данных. Обновить вычисленные результаты вручную невозможно.

Для определения вычисляемого поля можно использовать встроенные функции MS Access или выражения, создаваемые пользователем. Для каждого вычисляемого поля следует выбрать одно выражение. Выражения, определяющие вычисляемые поля, создаются с помощью мастера простых запросов или вводятся пользователем в строку "Групповая операция" бланка запроса, в которой допускается выбор статистических функций для выполняемых над полем вычислений. В бланке запроса задают также условия отбора, с помощью которых определяются группы, для которых вычисляются итоговые значения, записи, включаемые в вычисления, или результаты, отображаемые после выполнения расчетов.

Для расчетов с использованием формул, определяемых пользователем, требуется создать новое вычисляемое поле прямо в бланке запроса. Вычисляемое поле создается с помощью выражения, которое вводится в пустую ячейку "Поле" в бланке запроса. Результаты вычислений не обязательно должны отображаться в поле. Вместо этого, их можно использовать в условиях отбора для определения записей, которые выбираются в запросе, или для определения записей, над которыми производятся какие-либо действия.

Условия отбора записей

Условия отбора записей могут задаваться для одного или нескольких полей в строке "Условие отбора" бланка запроса (см. рисунок 6.3.2). Условием отбора является выражение,

которое состоит из операторов сравнения и операндов, используемых для сравнения. В качестве операндов выражения могут использоваться: литералы и константы.

Литералом называют значение в явном представлении, например, число, строковое значение или дату. Примерами значений в явном представлении могут служить "Москва", 100 и #1-января-94#. Даты необходимо заключать в символы (#), а строковые значения в прямые кавычки (").

Константа представляет не изменяющееся значение. True, False, Истина, Ложь и Null являются примерами констант, автоматически определяемых в MS Access. Допускается также использование *подстановочных знаков*, представленных в следующей таблице.

Знак	Пример	Использование
*	ко* ищет "кот", "котелок" и "кофе", *ка ищет "кнопка", "папка" и "практика"	Соответствует любому количеству символов и может быть использован в любом месте текстовой строки
?	п?ник находит "подлинник", "помощник" и "противник"	Соответствует любому одиночному символу
#	5#2 находит 502, 572, 592	Соответствует любой одиночной цифре
[]	л[ай]па находит "лапа" и "липа", но не "лупа"	Соответствует любому одному символу из заключенных в квадратные скобки
!	л[!ай]па находит "лупа", но не "лапа" и не "липа"	Соответствует любому одному символу, кроме заключенных в скобки
-	б[а-в]д находит "бад", "ббд" и "бвд"	Соответствует любому символу из диапазона символов. Необходимо указывать этот диапазон по возрастанию (например от А до Я, но не от Я до А)

Оператор *BETWEEN...AND* определяет принадлежность значения выражения указанному диапазону

Оператор *IN* проверяет, совпадает ли значение выражения с одним из элементов указанного списка, который задается в круглых скобках.

Оператор *AND* требует одновременного выполнения всех выражений, которые он соединяет.

Оператор *OR* требует выполнения хотя бы одного из тех выражений, которые он соединяет.

Оператор *NOT* требует невыполнения того выражения, перед которым он стоит.

Примеры определения диапазонов значений с помощью операторов представлены в следующей таблице

Пример выражения	Описание
>234	Числа, превышающие 234
Between #02.02.93# And #01.12.93#	Даты в диапазоне от 2-фев-93 до 1-дек-93
<1200,45	Числа, меньшие чем 1200,45
>="Иванов"	Все фамилии, начиная с фамилии "Иванов" и до конца алфавита

Еще одним встроенным элементом поиска нужной информации в Access являются функции. Функция возвращает значение, которое является результатом расчетов или выполнения других операций. В Microsoft Access определен ряд встроенных функций, например:

функция *Sum* возвращает сумму набора значений поля; функция *Avg* вычисляет арифметическое среднее набора чисел, содержащихся в указанном поле запроса; функция *Count* вычисляет количество записей, возвращаемых запросом.

Запросы-действия

Как уже говорилось ранее, существует еще одна категория запросов - запросы-действия, которые вносят изменения в данные. Эта категория состоит из четырех типов, в которых условия, заданные в бланке, служат для отбора изменяемых записей, а дополнительная информация определяет характер изменений.

Запрос на *удаление* -удаляет группу записей из одной или нескольких таблиц. С помощью запроса на удаление можно удалять только всю запись целиком, а не отдельные поля внутри нее.

Запрос на *создание таблицы*-создает новую таблицу на основе всех или части данных из одной или нескольких таблиц и заносит в новую таблицу выбранные записи. Этот тип запроса может быть полезен при создании резервных копий информации. Например, отобрать все старые записи для копирования в отдельную резервную таблицу, после чего все эти записи удалить с помощью запроса на удаление из основной таблицы.

Запрос на *добавление*- добавляет группу записей из одной или нескольких таблиц в одну или несколько других.

Запрос на *обновление* записей - вносит общие изменения в группу записей одной или нескольких таблиц. Запрос на обновление записей позволяет изменять данные в существующих таблицах.

При работе с запросом в режиме конструктора можно преобразовать существующий запрос к одному из этих четырех типов. Для это следует выбрать тип запроса из меню "Запрос" (см. Рисунок 6.5.1).

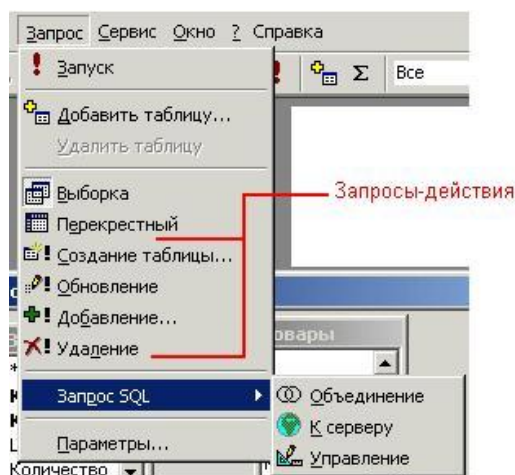


Рисунок 6.5.1 - Меню "Запрос"

При преобразовании запроса к другому типу состав строк бланка запросов соответствующим образом меняется. В двух типах запросов "Создание таблицы" и "Добавление" MS Access выводит диалоговое окно для получения дополнительной информации перед переходом в режим конструктора.

Пользователь может открыть в режиме конструктора различные запросы: запрос на выборку, перекрестный запрос и запрос на изменение. Запрос на выборку и перекрестный запрос также можно открыть в режиме таблицы для просмотра результатов. Запрос на изменение нельзя открыть в режиме таблицы. Однако в этом режиме можно выполнить предварительный просмотр данных, которые будут задействованы в запросе на изменение. Такой запрос отличается тем, что в окне базы данных рядом с его именем стоит восклицательный знак (!).

Запросы SQL

Запрос SQL - это запрос, создаваемый при помощи инструкций SQL. Примерами запросов SQL могут служить запросы на объединение, запросы к серверу, перекрестные и подчиненные запросы.

Запрос на *объединение* - комбинирует поля из одной или нескольких таблиц или запросов в одно поле в результатах запроса. Например, если шесть поставщиков ежемесячно посылают новые списки оборудования, то с помощью запроса на объединение эти списки можно объединить в один. А затем результаты поместить в новую таблицу, созданную с помощью запроса на создание таблицы, основанного на запросе на объединение.

Запрос *к серверу* - отправляет команды непосредственно в базы данных ODBC, например Microsoft SQL Server, причем используются только команды, поддерживаемые данным сервером.

Запрос *управление* создает или вносит изменения в объекты базы данных, такие как таблицы Microsoft Access или Microsoft SQL Server.

Запросы и фильтры

Запрос на выборку содержит условия отбора данных и возвращает выборку, соответствующую указанным условиям, без изменения возвращаемых данных. В Microsoft Access существует также понятие *фильтра*, который в свою очередь является набором условий, позволяющих отбирать подмножество записей или сортировать их. Сходство между запросами на выборку и фильтрами заключается в том, что и в тех и в других производится извлечение подмножества записей из базовой таблицы или запроса. Однако между ними существуют различия, которые нужно понимать, чтобы правильно сделать выбор, в каком случае использовать запрос, а в каком - фильтр.

Основные отличия запросов и фильтров заключаются в следующем.

- Фильтры не позволяют в одной строке отображать данные из нескольких таблиц, т. е. объединять таблицы.
- Фильтры не дают возможности указывать поля, которые должны отображаться в результирующем наборе записей, они всегда отображают все поля базовой таблицы.
- Фильтры не могут быть сохранены как отдельный объект в окне базы данных (они сохраняются только в виде запроса).
- Фильтры не позволяют вычислять суммы, средние значения, подсчитывать количество записей и находить другие итоговые значения.

Запросы могут использоваться только с закрытой таблицей или запросом. Фильтры обычно применяются при работе в режиме **Формы** или в режиме **Таблицы** для просмотра или изменения подмножества записей. Запрос можно использовать:

- для просмотра подмножества записей таблицы без предварительного открытия этой таблицы или формы;
- для того чтобы объединить в виде одной таблицы на экране данные из нескольких таблиц;
- для просмотра отдельных полей таблицы;
- для выполнения вычислений над значениями полей.

План-конспект №17

Тема занятия: Групповые операции и выражения в запросах

Тип занятия: *теоретическое занятие.*

Цель занятия: познакомить с основными приемами групповых операций и выражений в запросах

Учебные: 1) разъяснить основные понятия, связанные с групповыми операциями

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1.

Самостоятельная работа студента: составление опорного конспекта (ОК 2.1.1)

Составление сводной таблицы (ОК 3.1.2)

Запросы. Вычисления и групповые операции

Очень часто при формировании набора записей требуется произвести вычисления над данными (определить возраст сотрудника, стоимость заказа, процент от продаж, выделить часть номенклатурного кода и т. п., т. е. вывести информацию, которая не хранится в базе данных) или выполнить определенные операции по непосредственной обработке отбираемых данных.

В QBE СУБД MS Access такие возможности предоставляются через вычисляемые поля и групповые операции.

Вычисляемые поля

Вычисляемое поле представляет собой выражение, состоящее из операторов (арифметических, сравнения, логических, конкатенации) и операндов. В качестве операндов могут быть использованы константы, встроенные или определенные пользователем функции и идентификаторы, например

Стоимость: Товары! Цена * Количество * (1-Скидка)

КоличествоМужчин: Sum(If(Пол = "м"; 1; 0))

ФИО: Фамилия &" "& Left(Имя;1) &" "& Left(Отчество;1) &"."

Стоимость, КоличествоМужчин и ФИО являются именами вычисляемых полей и отображаются в режиме таблицы в заголовке столбца, символ двоеточие исполняет роль разделителя между именем вычисляемого поля и выражением.

Если имя таблицы или поля содержит пробелы, то его идентификатор обязан в выражении заключаться в квадратные скобки, например

Стоимость: Цена*[Количество товара]

Вычисляемое поле создается прямо в бланке QBE путем внесения выражения в ячейку **Поле** любого свободного столбца. Результаты вычислений, выводящиеся в поле, не запоминаются в базовой таблице. Вычисления снова производятся всякий раз, когда выполняется запрос, поэтому результаты всегда представляют текущее содержимое базы данных.

Обновить вычисленные результаты вручную невозможно.

Для построения сложных выражений в СУБД MS Access входит утилита, называемая **Построитель выражений**.

Создание выражений с помощью Построителя выражений

Построитель выражений можно запустить щелкнув по клавише **Построить** на панели инструментов **Конструктор запросов** или выбрав в контекстном меню поля бланка запроса QBE команду меню **Построить....**

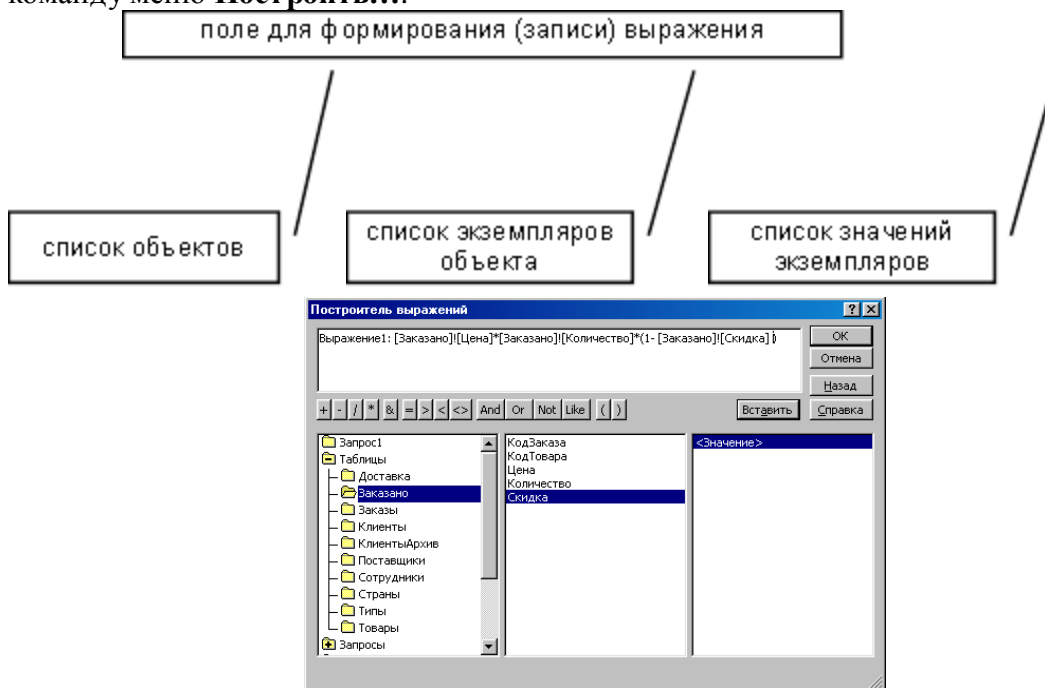


Рис.6.2. Диалоговое окно Построителя выражений со сформированным выражением

Обзор встроенных функций СУБД MS Access

СУБД MS Access содержит более 100 встроенных функций (рис.6.3), которые можно использовать при формировании вычисляемого поля или при задании условия отбора.

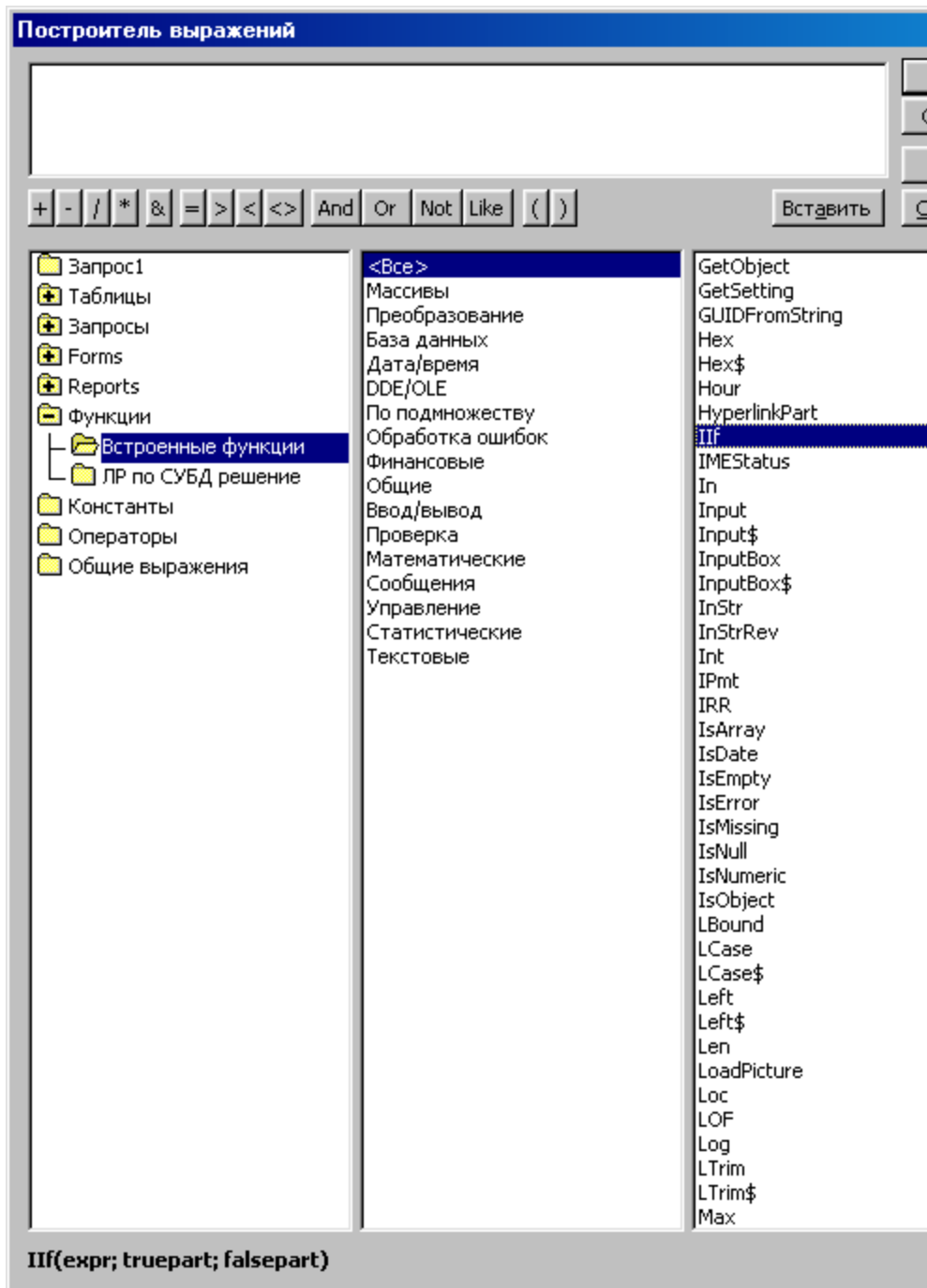


Рис.6.3. Диалоговое окно Построителя выражений со списком встроенных функций

Все встроенные функции в Построителе выражений сгруппированы по функциональному назначению.

При выборе функции в правом окне построителя, в нижней области окна Построителя выражений отображается синтаксис этой функции. Для более детальной справки необходимо щелкнуть по клавише **Справка**.

Таблица 6.1 Функции категории Дата/время

Day(дата)	Возвращает значение дня месяца от 1 до 31
Month(дата)	Возвращает значение месяца от 1 до 12
MonthName(месяц[:флаг])	Возвращает название месяца соответствующего номеру месяца: 1 – январь, 2 – февраль, и т. д. Если значение аргумента флаг Истина, то функция

	возвращает аббревиатуру месяца: 1 – янв, 2 – фев и т. д.
Year(дата)	Возвращает значение года от 100 до 9999
Weekday(дата[; число])	Если аргумент число не указан, возвращает значение дня недели от 1 (воскресенье) до 7 (суббота). Если аргумент число имеет значение 0, то возвращает значение дня недели от 1 (понедельник) до 7 (воскресенье)
Hour(дата)	Возвращает целое число от 0 до 23, представляющее значение часа
DatePart(интервал; дата)	Возвращает числовое значение в зависимости от значения аргумента интервал: "q" – квартал (от 1 до 4); "m" – месяц (от 1 до 12); "уууу" – год (от 100 до 9999); "ww" – неделя (от 1 до 53); и т. п. (см. справку по функции).
Date()	Возвращает текущую системную дату

Функции категории Проверка

IsNull(переменная)	Возвращает значение Истина если значение переменной (значение поля в источнике данных) равно Null
IsNumeric(переменная)	Возвращает значение Истина если значение переменной (значение поля в источнике данных) имеет один из числовых типов данных

Функции категории Управление

If(условие; выр1; выр2)	Возвращает значение выражения выр1 если значение аргумента условие Истина и возвращает значение выражения выр2 если значение аргумента условие Ложь. Данная функция аналогична функции ЕСЛИ в MS Excel.
-------------------------	---

Функции категории Текстовые

Left(текст; n)	Возвращает n левых символов аргумента текст
Right(текст; n)	Возвращает n правых символов аргумента текст
Mid(текст; нач_поз[; n])	Возвращает n символов начиная с позиции нач_поз аргумента текст. Если аргумент n не указан, то возвращает все символы до конца строки начиная с позиции нач_поз аргумента текст.
Len(текст)	Возвращает количество символов (длину строки) в аргументе текст
LTrim(текст)	Возвращает строковое значение аргумента текст без начальных пробелов
RTrim(текст)	Возвращает строковое значение аргумента текст без заключительных пробелов
Trim(текст)	Возвращает строковое значение аргумента текст без начальных и заключительных пробелов
Str(число)	Возвращает строковое значение аргумента число
Format(переменная; формат)	Возвращает значение аргумента переменная в формате заданным аргументом формат

Итоговые запросы

При анализе данных очень часто интересуют не отдельные записи, а итоговые значения по группам данных, например:

- количество сделок с Партнерами за определенный промежуток времени;
- средний объем продаж по каждому месяцу за предыдущий год.

Ответы на такие вопросы дает итоговый запрос.

Для вычисления итоговых значений необходимо нажать кнопку **Групповые операции** на панели инструментов **Конструктор запросов**, чтобы в бланке QBE появилась строка **Групповая операция** (после имени таблицы).

По умолчанию для каждого поля, занесенного в бланк запроса, устанавливается значение **Группировка** (итоги не подводятся).

Для подведения итогов необходимо заменить установку **Группировка** на конкретную итоговую функцию. В СУБД MS Access предусмотрено 9 функций, обеспечивающих выполнение групповых операций.

Таблица. Функции категории Статистические

Функция	Назначение
Sum	Возвращает сумму набора значений
Avg	Возвращает среднее арифметическое набора значений
Min	Возвращает наименьшее значение из набора значений
Max	Возвращает наибольшее значение из набора значений
Count	Возвращает количество записей в наборе значений отличных от Null
First	Возвращает первое значение поля в группе
Last	Возвращает последнее значение поля в группе
StDev	Возвращает среднеквадратичное отклонение набора значений
Var	Возвращает дисперсию набора значений

В раскрываемом списке строки **Групповая операция** имеется установка **Выражение**. Данная установка применяется, когда в выражении (строка **Поле**) используется несколько итоговых функций.

В раскрываемом списке строки **Групповая операция** имеется установка **Условие**. Данная установка применяется, когда в строке Условие отбора записано условие выборки, но данные столбца (поля) не должны участвовать в групповой операции.

Для решения более сложных статистических задач в СУБД MS Access предусмотрен специальный тип запроса – перекрестный запросов.

План-конспект №18

Тема занятия: Перекрестные запросы

Тип занятия: *теоретическое занятие.*

Цель занятия: изучить назначение и принцип создания перекрестных запросов

Вид занятия: теоретическое занятие

Задачи:

- Учебные: 1) разъяснить назначение перекрестных запросов
2) объяснить принцип разработки перекрестного запроса;

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1.

Самостоятельная работа студента: Составление опорного конспекта по теме

Перекрестные запросы

В **Access** имеется возможность создания запросов, в которых используются групповые операции, в виде электронных таблиц, т.е. в качестве названий строк и столбцов такой "таблицы" используются те или иные поля таблиц, а в ее "ячейках" будут отображаться вычисляемые значения, которые соответствуют значениям, указанным в названии строки и столбца. Например, в качестве примера подобной электронной таблицы можно рассмотреть набор оценок, которые получил каждый студент по всем предметам. Другими словами, заголовками строк такой таблицы будут фамилии студентов, заголовками столбцов - названия предметов, а на пересечении строк и столбцов будет отображаться оценка того или иного студента по данному предмету, аналогично расположению данных в рабочих листах **Excel**.

Примечание. Вычисляемые значения не могут быть отредактированы, поэтому перекрестный запрос нет смысла использовать с целью изменения данных. Однако перекрестные запросы могут оказаться весьма полезными при создании отчетов.

Для создания перекрестного запроса, отображающего таблицу оценок студентов по предметам, следует, вначале создать новый запрос в режиме конструктора. Затем нужно в макет запроса добавить таблицы, в которых содержатся исходные данные запроса: **Студенты**, **Успеваемость** и **Предметы**.

После этого необходимо выполнить команду **Запрос | Перекрестный**, в результате чего в макете запроса среди параметров появится строка **Перекрестная таблица**, в которой для полей запроса можно с помощью разворачивающегося списка выбрать одно из следующих значений.

- **Заголовки строк** - выбирается для поля запроса, значения которого нужно использовать в качестве заголовков строк. В качестве заголовков строк необходимо определить хотя бы одно поле, при этом в строке **Групповые операции** необходимо выбрать значение **Группировка**, одну из **итоговых функций** или **Выражение**.

- **Заголовки столбцов** - выбирается для поля запроса, значения которого нужно использовать в качестве заголовков столбцов. С этой целью используется только одно поле, требования к которому аналогичны требованиям к полю, в котором установлено значение **Заголовки Строк**.

- **Значение** - выбирается для поля запроса, в котором вычисляется итоговое значение, отображаемое в "ячейках" перекрестного запроса. Такое поле должно быть единственным, при этом в строке **Групповые операции** для него необходимо выбрать одну из **итоговых функций** или задать **выражение**, в котором используются итоговые функции.

- **(не отображается)** - выбирается для того поля, значения которого не должны отображаться в перекрестном запросе.

В первом столбце запроса нужно выбрать поле **Фамилия** таблицы **Студенты**, при этом следует задать для него значение **Группировка** в поле **Групповая операция**, а также **Заголовки строк** в поле **Перекрестная таблица**. Также можно выбрать направление сортировки **по возрастанию**.

Для второго столбца запроса необходимо выбрать поле **Название** таблицы **Предметы** и установить для него значение **Группировка** в поле **Групповая операция** и **Заголовки столбцов** в поле **Перекрестная таблица**.

В третьем столбце запроса нужно выбрать поле **Оценка** таблицы **Успеваемость** и задать для него функцию **Sum** в поле **Групповая операция**, а также **Значение** в поле **Перекрестная таблица**.

Созданный запрос можно сохранить под именем **ОценкиПоПредметам** (рис. 1).

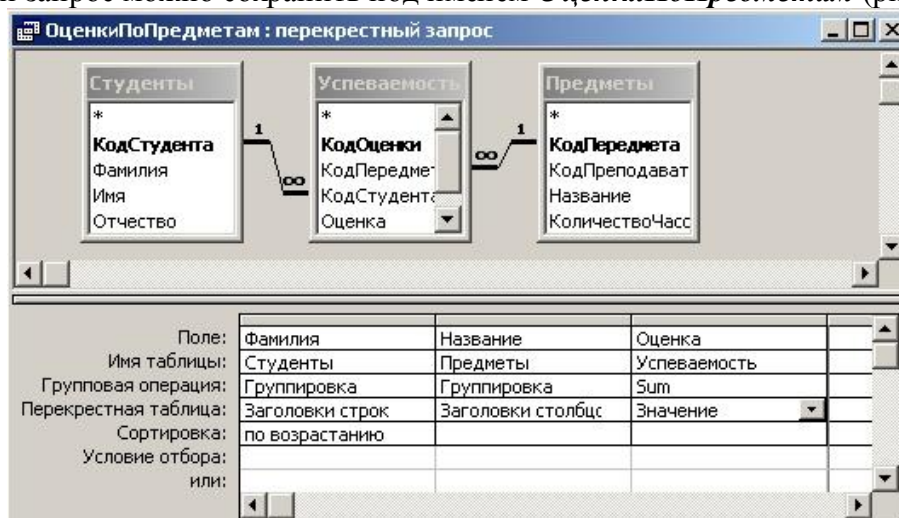


Рис. 1. Макет перекрестного запроса **ОценкиПоПредметам**

В результате выполнения перекрестного запроса **ОценкиПоПредметам** будет получена электронная таблица, содержащая оценки студентов по всем предметам (рис. 2). Данный запрос

может быть использован, например, в отчете, который должен отображать итоги прошедшей сессии.

ОценкиПоПредметам : перекрестный запрос						
	Фамилия	Математика	Физика	Философия	Химия	Экономика
▶	Жукова		5		5	
	Игнатов		5		4	
	Морозов		4		4	
	Петров	5		3		4

План-конспект занятия №19

Тема занятия: Методы защиты базы данных

Тип занятия: *теоретическое занятие.*

Цель занятия: изучить назначение и методы защиты базы данных

Задачи:

- Учебные: 1) разъяснить назначение и устройство сканеров
2) разъяснить структуру и стандарты сканирования;

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1., ПК 2.4.

Самостоятельная работа студента:

- Реферирование темы «Методы защиты базы данных»
- составление опорного конспекта

Защита информации в базах данных

В современных СУБД поддерживается один из двух наиболее общих подходов к вопросу обеспечения безопасности данных: избирательный подход и обязательный подход. В обоих подходах единицей данных или "объектом данных", для которых должна быть создана система безопасности, может быть как вся база данных целиком, так и любой объект внутри базы данных.

Эти два подхода отличаются следующими свойствами:

- В случае избирательного управления некоторый пользователь обладает различными правами (привилегиями или полномочиями) при работе с данными объектами. Разные пользователи могут обладать разными правами доступа к одному и тому же объекту. Избирательные права характеризуются значительной гибкостью.

- В случае неизбирательного управления, наоборот, каждому объекту данных присваивается некоторый классификационный уровень, а каждый пользователь обладает некоторым уровнем допуска. При таком подходе доступом к определенному объекту данных обладают только пользователи с соответствующим уровнем допуска.

- Для реализации избирательного принципа предусмотрены следующие методы. В базу данных вводится новый тип объектов БД — это пользователи. Каждому пользователю в БД присваивается уникальный идентификатор. Для дополнительной защиты каждый пользователь кроме уникального идентификатора снабжается уникальным паролем, причем если идентификаторы пользователей в системе доступны системному администратору, то пароли пользователей хранятся чаще всего в специальном кодированном виде и известны только самим пользователям.

- Пользователи могут быть объединены в специальные группы пользователей. Один пользователь может входить в несколько групп. В стандарте вводится понятие группы PUBLIC,

для которой должен быть определен минимальный стандартный набор прав. По умолчанию предполагается, что каждый вновь создаваемый пользователь, если специально не указано иное, относится к группе PUBLIC.

- Привилегии или полномочия пользователей или групп — это набор действий (операций), которые они могут выполнять над объектами БД.

- В последних версиях ряда коммерческих СУБД появилось понятие "роли". Роль — это поименованный набор полномочий. Существует ряд стандартных ролей, которые определены в момент установки сервера баз данных. И имеется возможность создавать новые роли, группируя в них произвольные полномочия. Введение ролей позволяет упростить управление привилегиями пользователей, структурировать этот процесс. Кроме того, введение ролей не связано с конкретными пользователями, поэтому роли могут быть определены и сконфигурированы до того, как определены пользователи системы.

- Пользователю может быть назначена одна или несколько ролей.

- Объектами БД, которые подлежат защите, являются все объекты, хранимые в БД: таблицы, представления, хранимые процедуры и триггеры. Для каждого типа объектов есть свои действия, поэтому для каждого типа объектов могут быть определены разные права доступа.

На самом элементарном уровне концепции обеспечения безопасности баз данных исключительно просты. Необходимо поддерживать два фундаментальных принципа: проверку полномочий и проверку подлинности (аутентификацию).

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

Система назначения полномочий имеет в некотором роде иерархический характер. Самыми высокими правами и полномочиями обладает системный администратор или администратор сервера БД. Традиционно только этот тип пользователей может создавать других пользователей и наделять их определенными полномочиями.

СУБД в своих системных каталогах хранит как описание самих пользователей, так и описание их привилегий по отношению ко всем объектам.

Далее схема предоставления полномочий строится по следующему принципу. Каждый объект в БД имеет владельца — пользователя, который создал данный объект. Владелец объекта обладает всеми правами-полномочиями на данный объект, в том числе он имеет право предоставлять другим пользователям полномочия по работе с данным объектом или забирать у пользователей ранее предоставленные полномочия.

В ряде СУБД вводится следующий уровень иерархии пользователей — это администратор БД. В этих СУБД один сервер может управлять множеством СУБД (например, MS SQL Server, Sybase).

В СУБД Oracle применяется однобазовая архитектура, поэтому там вводится понятие подсхемы — части общей схемы БД и вводится пользователь, имеющий доступ к подсхеме.

В стандарте SQL не определена команда создания пользователя, но практически во всех коммерческих СУБД создать пользователя можно не только в интер активном режиме, но и программно с использованием специальных хранимых процедур. Однако для выполнения этой операции пользователь должен иметь право на запуск соответствующей системной процедуры.

В стандарте SQL определены два оператора: GRANT и REVOKE соответственно предоставления и отмены привилегий.

Оператор предоставления привилегий имеет следующий формат:

```
GRANT {<список действий> | ALL PRIVILEGES }  
ON <имя_объекта>  
TO {<имя_пользователя> | PUBLIC }  
[WITH GRANT OPTION ]
```

Здесь список действий определяет набор действий из общедопустимого перечня действий над объектом данного типа.

Параметр ALL PRIVILEGES указывает, что разрешены все действия из допустимых для объектов данного типа.

<имя_объекта> — задает имя конкретного объекта: таблицы, представления, хранимой процедуры, триггера.

<имя_пользователя> или PUBLIC определяет, кому предоставляются данные привилегии.

Параметр WITH GRANT OPTION является необязательным и определяет режим, при котором передаются не только права на указанные действия, но и право передавать эти права другим пользователям. Передавать права в этом случае пользователь может только в рамках разрешенных ему действий.

Рассмотрим пример, пусть у нас существуют три пользователя с абсолютно уникальными именами user1, user2 и user3. Все они являются пользователями одной БД.

User1 создал объект Tab1, он является владельцем этого объекта и может передать права на работу с этим объектом другим пользователям. Допустим, что пользователь user2 является оператором, который должен вводить данные в Tab1 (например, таблицу новых заказов), а пользователь user3 является большим начальником (например, менеджером отдела), который должен регулярно просматривать введенные данные.

Для объекта типа таблица полным допустимым перечнем действий является набор из четырех операций: SELECT, INSERT, DELETE, UPDATE. При этом операция обновление может быть ограничена несколькими столбцами.

Общий формат оператора назначения привилегий для объекта типа таблица будет иметь следующий синтаксис:

```
GRANT {[SELECT][,INSERT][,DELETE][,UPDATE (<список столбцов>)]}  
ON <имя_таблицы>  
TO {<имя_пользователя> | PUBLIC } [WITH GRANT OPTION ]
```

Тогда резонно будет выполнить следующие назначения:

```
GRANT INSERT  
ON Tab1  
TO user2  
GRANT SELECT  
ON Tab1  
TO user3
```

Эти назначения означают, что пользователь user2 имеет право только вводить новые строки в отношении Tab1, а пользователь user3 имеет право просматривать все строки в таблице Tab1.

При назначении прав доступа на операцию модификации можно уточнить, значение каких столбцов может изменять пользователь. Допустим, что менеджер отдела имеет право изменять цену на предоставляемые услуги. Предположим, что цена задается в столбце COST таблицы Tab1. Тогда операция назначения привилегий пользователю user3 может измениться и выглядеть следующим образом:

```
GRANT SELECT, UPDATE (COST)  
ON Tab1  
TO user3
```

Если наш пользователь user1 предполагает, что пользователь user4 может его замещать в случае его отсутствия, то он может предоставить этому пользователю все права по работе с созданной таблицей Tab1.

```
GRANT ALL PRIVILEGES  
ON Tab1  
TO user4  
WITH GRANT OPTION
```

В этом случае пользователь user4 может сам назначать привилегии по работе с таблицей Tab1 в отсутствие владельца объекта пользователя user1. Поэтому в случае появления нового оператора пользователя user5 он может назначить ему права на ввод новых строк в таблицу командой

```
GRANT INSERT  
ON Tab1
```

```
TO user5
```

Если при передаче полномочий набор операций над объектом ограничен, то пользователь, которому переданы эти полномочия, может передать другому пользователю только те полномочия, которые есть у него, или часть этих полномочий. Поэтому если пользователю user4 были делегированы следующие полномочия:

```
GRANT SELECT, UPDATE, DELETE
```

```
ON Tab1
```

```
TO user4
```

```
WITH GRANT OPTION,
```

то пользователь user4 не сможет передать полномочия на ввод данных пользователю user5, потому что эта операция не входит в список разрешенных для него самого.

Кроме непосредственного назначения прав по работе с таблицами эффективным методом защиты данных может быть создание представлений, которые будут содержать только необходимые столбцы для работы конкретного пользователя и предоставление прав на работу с данным представлением пользователю.

Так как представления могут соответствовать итоговым запросам, то для этих представлений недопустимы операции изменения, и, следовательно, для таких представлений набор допустимых действий ограничивается операцией SELECT. Если же представления соответствуют выборке из базовой таблицы, то для такого представления допустимыми будут все 4 операции: SELECT, INSERT, UPDATE и DELETE.

Для отмены ранее назначенных привилегий в стандарте SQL определен оператор REVOKE. Оператор отмены привилегий имеет следующий синтаксис:

```
REVOKE {<список операций> | ALL PRIVILEGES }
```

```
ON <имя_объекта>
```

```
FROM {<список пользователей> | PUBLIC }
```

```
{CASCADE | RESTRICT }
```

Параметры CASCADE или RESTRICT определяют, каким образом должна производиться отмена привилегий. Параметр CASCADE отменяет привилегии не только пользователя, который непосредственно упоминался в операторе GRANT при предоставлении ему привилегий, но и всем пользователям, которым этот пользователь присвоил привилегии, воспользовавшись параметром WITH GRANT OPTION.

Например, при использовании операции:

```
REVOKE ALL PRIVILEGES
```

```
ON Tab1
```

```
FROM user4 CASCADE
```

будут отменены привилегии и пользователя user5, которому пользователь user4 успел присвоить привилегии.

Параметр RESTRICT ограничивает отмену привилегий только пользователю, непосредственно упомянутому в операторе REVOKE. Но при наличии делегированных привилегий этот оператор не будет выполнен. Так, например, операция:

```
REVOKE ALL PRIVILEGES
```

```
ON Tab1 TO user4 RESTRICT
```

не будет выполнена, потому что пользователь user4 передал часть своих полномочий пользователю user5.

Посредством оператора REVOKE можно отобрать все или только некоторые из ранее присвоенных привилегий по работе с конкретным объектом. При этом из описания синтаксиса оператора отмены привилегий видно, что можно отобрать привилегии одним оператором сразу у нескольких пользователей или у целой группы PUBLIC.

Поэтому корректным будет следующее использование оператора REVOKE:

```
REVOKE INSERT ON Tab1 TO user2,user4 CASCADE
```

При работе с другими объектами изменяется список операций, которые используются в операторах GRANT и REVOKE.

По умолчанию действие, соответствующее запуску (исполнению) хранимой процедуры, назначается всем членам группы PUBLIC.

Если вы хотите изменить это условие, то после создания хранимой процедуры необходимо записать оператор REVOKE.

```
REVOKE EXECUTE
ON COUNT_EX
TO PUBLIC CASCADE
```

И теперь мы можем назначить новые права пользователю user4.

```
GRANT EXECUTE
ON COUNT_EX
TO user4
```

Системный администратор может разрешить некоторому пользователю создавать и изменять таблицы в некоторой БД. Тогда он может записать оператор предоставления прав следующим образом:

```
GRANT CREATE TABLE,
ALTER TABLE,
DROP TABLE
ON DB_LIB
TO user1
```

В этом случае пользователь user1 может создавать, изменять или удалять таблицы в БД DB_LIB, однако он не может разрешить создавать или изменять таблицы в этой БД другим пользователям, потому что ему дано разрешение без права делегирования своих возможностей.

В некоторых СУБД пользователь может получить права создавать БД. Например, в MS SQL Server системный администратор может предоставить пользователю main_user право на создание своей БД на данном сервере. Это может быть сделано следующей командой:

```
GRANT CREATE DATABASE
ON SERVER_0
TO main_user
```

По принципу иерархии пользователь main_user, создав свою БД, теперь может предоставить права на создание или изменение любых объектов в этой БД другим пользователям.

В СУБД, которые поддерживают однобазовую архитектуру, такие разрешения недопустимы. Например, в СУБД Oracle на сервере создается только одна БД, но пользователи могут работать на уровне подсхемы (части таблиц БД и связанных с ними объектов). Поэтому там вводится понятие системных привилегий. Их очень много, 80 различных привилегий.

План-конспект занятия №20

Тема: Контроль и проверка введенных данных

Тип занятия: *теоретическое занятие.*

Цель занятия: изучить способы контроля и проверки введенных данных

Задачи:

Учебные: 1) разъяснить способы контроля и проверки введенных данных

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти
- 3) развитие профессиональных навыков

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1., ПК 2.4.

Самостоятельная работа студента: составление опорного конспекта (ОК 2.1.1)

Контроль и проверка введенных данных

Microsoft Access может осуществлять проверку данных, вводимых в поле или запись таблицы. Чтобы проверить значение, вводимое в поле или запись, необходимо задать условие на значение. Кроме того, можно указать текст сообщения, которое должно появиться на экране, если

введенные данные не соответствуют данному условию. В Microsoft Access существует два типа условий на значение:

- условия на значение поля;
- условия на значения записи.

Очень важно, что оба типа условий вводятся в систему на уровне таблицы. Это означает, что проверка вводимых данных на соответствие этим условиям будет выполняться независимо от способа ввода данных, будь то прямое редактирование записей в режиме Таблицы, или импорт (копирование) из другой таблицы, или редактирование записи в форме (на странице доступа к данным). Это позволяет надежно защищать данные от ошибок ввода.

Условие на значение поля позволяет обеспечить ввод корректных данных в поле записи. Проверка введенного значения осуществляется при переходе к другому полю или другой записи. Если введенное значение не соответствует заданному условию, на экран выводится сообщение, указанное пользователем, или стандартное сообщение, если пользователь не определил свое. Чтобы определить условие на значение поля, необходимо:

1. Открыть таблицу в режиме Конструктора.
2. В верхней части окна выбрать нужное поле.
3. Открыть на панели свойств поля вкладку **Общие** (General) и выбрать свойство **Условие на значение** (Validation Rule), а затем ввести необходимое условие. Задать условие на значение можно при помощи Построителя выражений (рис. 2.35), нажав на кнопку Построителя справа. Выражения, используемые в условиях, возвращают значения **Истина** (True), и в этом случае данные корректны, или **Ложь** (False) — данные некорректны. Одним из самых простых выражений, которое может встретиться в условии, является > 0 , т. е. в поле могут вводиться только положительные числа. Для ввода более сложных выражений нужно знать правила построения выражений в Access.

4. Выбрать свойство **Сообщение об ошибке** (Validation Text) и ввести сообщение, которое будет выводиться на экран, если данные не соответствуют заданному условию.

Если условие на значение накладывается на поле, которое уже содержит данные, то при сохранении таблицы Microsoft Access выдаст вопрос, нужно ли применять новое условие к существующим данным.

Замечание

Чтобы проверить соответствие текущих данных в таблице условиям на значение, следует на заголовке таблицы в режиме конструктора нажать правую кнопку мыши, а затем выбрать команду **Проверка условий** (Test Validation Rules).

Условия на значение записи отличаются от условий на значение поля тем, что в условиях на значение записи допускаются ссылки на разные поля. Это позволяет использовать такие условия для сравнения значений, введенных в разные поля таблицы. Условия на значение записи проверяются при сохранении записи. Чтобы определить условие на значение записи, необходимо:

1. Открыть таблицу в режиме Конструктора.
2. Нажать на панели инструментов **Конструктор таблиц** (Table Design) кнопку **Свойства** (Properties) или, щелкнув правой кнопкой мыши по заголовку таблицы, выбрать из контекстного меню команду **Свойства** (Properties). Откроется диалоговое окно **Свойства таблицы** (Table Properties).

3. Третья строка в этом окне — **Условие на значение** (Validation Rule), которое позволяет ввести выражение для условия на значение записи. Например, в таблице "Заказы" (Orders) нужно указать, что дата исполнения заказа должна быть больше, чем дата размещения. Если воспользоваться Построителем выражений, то видно, что в окне Построителя появилась папка Заказы, а рядом — список полей таблицы "Заказы" (Orders) (рис. 2.36).

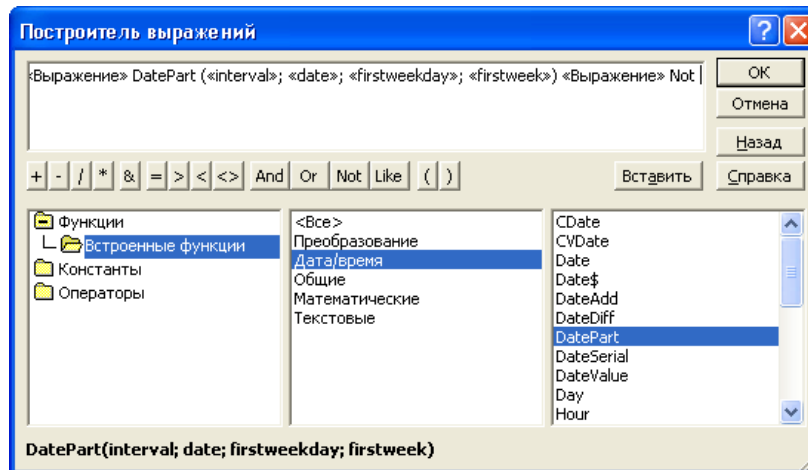


Рис. 2.36. Ввод условия с помощью Построителя выражений

Выделите в списке значение **ДатаИсполнения** и нажмите на кнопку **Вставить** (Paste) или просто дважды щелкните мышью по этому элементу. В верхнем поле окна Построителя появится имя поля в квадратных скобках. Нажмите кнопку **>**. Аналогичным способом выберите из списка элемент **ДатаРазмещения**. В результате в верхнем поле появится выражение [ДатаРазмещения] > [ДатаИсполнения]. Нажмите кнопку **ОК**.

Замечание

Если в выражениях используются ссылки на название полей таблицы, то они окружаются квадратными скобками.

4. Выбрать свойство **Сообщение об ошибке** (Validation Text) и ввести сообщение, которое будет выдаваться на экран, если данные не соответствуют указанному условию.

Замечание

Для таблицы можно указать только одно выражение, определяющее условия на значения. Если же необходимо наложить более одного условия, то в ячейке свойства **Условие на значение** (Validation Rule) выражения можно объединить при помощи оператора **And**.

Вне зависимости от того, заданы условия на значение поля или записи, при попытке выхода из текущего поля, после ввода данных Microsoft Access проверяет, являются ли введенные данные допустимыми для этого поля. Если значение не является допустимым, появляется предупреждающее сообщение. Для того чтобы выйти из поля, необходимо ввести правильное значение или отменить внесенные изменения.

Значение поля может оказаться недопустимым по следующим причинам.

- Значение несовместимо с типом данных поля (например, в числовых и текстовых полях или в поле даты). Если введено значение, не соответствующее типу данных поля, Microsoft Access пытается преобразовать его в правильный тип данных. Однако некоторые данные нельзя преобразовать в данные другого типа. Например, нельзя хранить текстовое значение в числовом поле.
- Данные нарушают условие на значение, заданное в свойстве **Условие на значение** (Validation Rule) этого поля.
- В поле необходимо ввести данные, а они не были введены.

Данные проверяются также при переходе на другую запись. Перед сохранением записи Microsoft Access проверяет все условия, заданные в свойстве **Условие на значение** (Validation Rule) записи. Кроме того, если таблица, в которую вносятся изменения, имеет ключ или уникальный индекс, то проверяется, что изменяемая запись не содержит в этом поле или полях значений, совпадающих со значениями в других записях таблицы. Если запись нельзя сохранить, необходимо внести требуемые изменения или нажать кнопку **Отменить** (Undo) для отмены всех изменений, внесенных в запись.

Помимо проверки условий на значения и корректности ввода данных в Microsoft Access имеется возможность проверки орфографии. Чтобы проверить орфографию в режиме таблицы, необходимо:

1. Выбрать записи, столбцы, поля или часть текста поля, в которых требуется проверить орфографию.
2. Нажать на кнопку **Орфография** (Spelling) на панели инструментов.

Кроме проверки орфографии, в Microsoft Access возможно автоматическое исправление орфографических ошибок при вводе данных. Для этого служит режим Автозамены (AutoCorrect), который позволяет автоматически исправлять слова, при вводе которых часто допускаются ошибки, а также автоматически заменять сокращения полными значениями. То есть можно ввести аббревиатуру, а Автозамена сама подставит расшифрованное наименование в поле. Чтобы включить режим Автозамены (AutoCorrect) необходимо:

1. Выбрать в меню **Сервис** (Tools) команду **Параметры автозамены** (AutoCorrect Options) (рис. 2.37).
2. Установить флажок **Заменять при вводе** (Replace text as you type).
3. В поле **Заменять** (Replace) ввести образец, который должен заменяться автоматически.
4. В поле **на** (With) ввести текст, который подставляется вместо образца.
5. Нажать кнопку **Добавить** (Add).

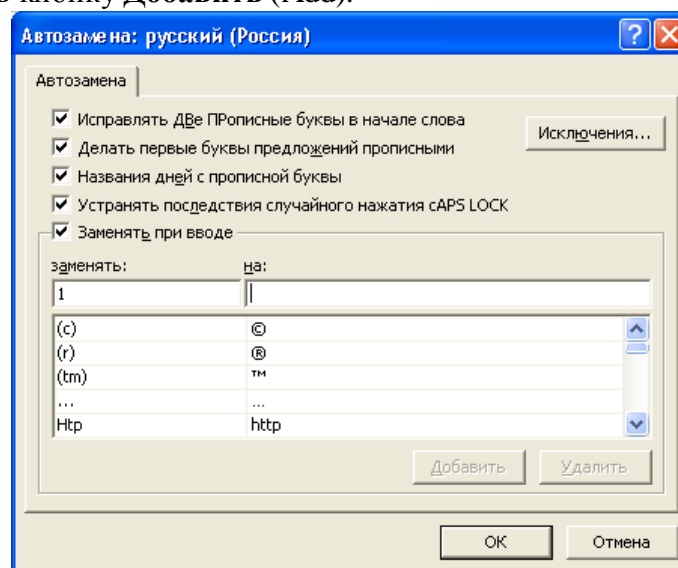


Рис. 2.37. Диалоговое окно Автозамена

План-конспект занятия №21

Тема: Установка системы защиты на уровне пользователей

Тип занятия: *теоретическое занятие.*

Цель занятия: познакомить с идеей защиты данных на уровне пользователей

Задачи:

Учебные: 1) разъяснить основные понятия, связанные с системой защиты

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые компетенции: ОК 1, ОК 10, ОК2.1.1, ОК 3.1.2, ПК 2.1., ПК 2.4.

Самостоятельная работа студента: составление опорного конспекта (ОК 2.1.1)

Защита на уровне пользователей

Microsoft Access напоминает механизмы безопасности, используемые в серверных системах. С помощью паролей и разрешений предоставляется или ограничивается доступ пользователей или групп к объектам базы данных. Учетные записи системы безопасности определяют пользователей и группы, которым разрешен доступ к объектам. Эта информация, которую называют сведениями о рабочей группе, сохраняется в файле рабочей группы.

Самым лучшим способом защиты базы данных является защита на уровне пользователей. Двумя основными причинами использования защиты на уровне пользователей являются:

- защита приложения от повреждения из-за неумышленного изменения пользователями таблиц, запросов, форм, отчетов и макросов, от которых зависит работа приложения;
- защита конфиденциальных сведений в базе данных.

При использовании этого типа защиты пользователь должен ввести пароль при запуске Microsoft Access. Затем Access читает файл рабочей группы, в котором каждый пользователь идентифицируется уникальным кодом. В файле рабочей группы пользователи по их личным кодам и паролям идентифицируются как авторизованные индивидуальные пользователи и как члены конкретных групп. В Microsoft Access определены две стандартные группы: администраторы ([группа «Admins»](#)) и пользователи ([группа «Users»](#)), но допускается определение дополнительных групп.

Хотя установка защиты на уровне пользователей для большинства баз данных является сложной задачей, мастер защиты обеспечивает быстрое и легкое применение новейших средств безопасности [базы данных Access](#). Более того, благодаря использованию общих схем защиты мастер позволяет уменьшить или даже вообще исключить необходимость использования команды **Защита** в меню **Сервис**.

Мастер защиты помогает назначить разрешения и создать [учетные записи пользователей](#) и [учетные записи групп](#). Однако после запуска мастера для базы данных и существующих в ней таблиц, запросов, форм, отчетов и макросов имеется возможность вручную назначить или удалить разрешения на доступ для учетных записей пользователей и групп в рабочей группе. Также могут быть установлены разрешения на доступ, по умолчанию присваиваемые вновь создаваемым объектам базы данных.

Группам и пользователям предоставляются разрешения, определяющие возможность их доступа к каждому объекту базы данных. Например, члены группы «Users» могут иметь разрешения на просмотр, ввод или изменение данных в таблице «Клиенты», но им не будет разрешено изменять структуру этой таблицы. Члены группы «Users» могут быть допущены только к просмотру данных в таблице, содержащей сведения о заказах, и не иметь доступа к таблице «Платежная ведомость». Члены группы «Admins» имеют разрешения на доступ ко всем объектам базы данных. Имеется возможность установить более разветвленную структуру управления, создавая собственные учетные записи групп, предоставляя этим группам соответствующие разрешения и добавляя пользователей в группы.

Если для системы защиты достаточно группы администраторов и группы пользователей, то нет необходимости создавать другие группы. Можно воспользоваться стандартными группами «Admins» и «Users». В этом случае необходимо присвоить соответствующие разрешения на доступ стандартной группе «Users» и добавить дополнительных администраторов в стандартную группу «Admins». Каждый новый пользователь автоматически добавляется в группу «Users». Типовые разрешения на доступ для группы «Users» могут включать «Чтение данных» и «Обновление данных» для таблиц и запросов и «Открытие/запуск» для форм и отчетов

В случае необходимости более разветвленной структуры управления для различных групп пользователей, имеется возможность создания новых групп, присвоения группам различных наборов разрешений на доступ и добавления новых пользователей в соответствующие группы. Для упрощения управления разрешениями на доступ рекомендуется присваивать разрешения только группам (а не отдельным пользователям), а затем добавлять пользователей в соответствующие группы.

Например, можно организовать защиту базы данных «Заказы» с помощью создания учетных записей «Управляющие» для руководства фирмы, «Представители» для торговых представителей и «Персонал» для административного персонала. После этого следует присвоить наибольшее количество разрешений группе «Управляющие», промежуточное количество группе «Представители» и минимальное группе «Персонал». При создании учетной записи для нового сотрудника эта запись будет добавляться в одну из групп и сотрудник автоматически получит разрешения, принадлежащие этой группе.

Рабочие группы и файлы рабочих групп

Рабочая группа Microsoft Access — это группа пользователей, которые совместно используют данные в многопользовательской среде. Если определена защита на уровне пользователей, члены рабочей группы регистрируются с помощью [учетных записей пользователей](#) и [учетных записей групп](#), которые сохраняются в файле рабочей группы Microsoft Access. Пароли пользователей также запоминаются в файле рабочей группы. Этим учетным записям системы защиты могут быть предоставлены разрешения на доступ к базе данных и ее таблицам, запросам, формам, отчетам и макросам. Сами разрешения сохраняются в базе данных с включенными средствами безопасности.

Когда пользователь в первый раз запускает Microsoft Access после установки Microsoft Office, Access автоматически создает файл рабочей группы Microsoft Access, который идентифицируется по указанным пользователем имени и названию организации. Относительное расположение файла рабочей группы записывается в следующие параметры реестра:

HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\Access\Jet\4.0\Engines\SystemDB

и

HKEY_USERS\DEFAULT\Software\Microsoft\Office\11.0\Access\Jet\4.0\Engines\SystemDB

Следующие пользователи наследуют стандартный путь к файлу рабочей группы из раздела реестра HKEY_USERS. Так как получение этой информации часто не представляет особого труда, существует вероятность того, что кто-то сможет создать другую версию этого файла и приобрести неотъемлемые [разрешения на доступ](#) учетной записи администратора (члена [группы Admins](#)) в рабочей группе, определяемой этим файлом рабочей группы. Чтобы избежать этого, создайте новый файл рабочей группы и укажите [код рабочей группы \(WID\)](#). Только тот, кто знает этот код, сможет создать копию файла рабочей группы.

Любые создаваемые учетные записи пользователей и групп и пароли сохраняются в этом файле рабочей группы до тех пор, пока пользователь не присоединится к другой рабочей группе с помощью администратора рабочих групп.

Примечание. В Access 2002 и более поздних версиях программу «Администратор рабочих групп» можно запустить из подменю **Защита** меню **Сервис**.

Обязательно запишите введенное имя, название организации и код рабочей группы с учетом регистра (для всех трех элементов) и храните их в надежном месте. Если потребуется заново создать файл рабочей группы, необходимо будет ввести точно такие же сведения. Если они забыты или потеряны, восстановить их и получить доступ к базам данных будет невозможно.

Разрешения на доступ: как они работают и кто может их назначать

Существуют два типа [разрешений на доступ](#): явные и неявные. Разрешения называются явными, если они непосредственно присвоены [учетной записи пользователя](#); такие разрешения не влияют на разрешения других пользователей. Неявными называются разрешения на доступ, присвоенные [учетной записи группы](#). Пользователь, включенный в такую группу, получает все разрешения, предоставленные группе; удаление пользователя из этой группы лишает его всех разрешений, присвоенных данной группе.

При попытке пользователя выполнить какую-либо операцию с защищенным объектом базы данных его текущие разрешения определяются комбинацией явных и неявных разрешений на доступ. На уровне пользователей всегда действуют минимальные ограничения из налагаемых явными разрешениями для пользователя и для всех групп, к которым принадлежит данный пользователь. Поэтому простейшим способом управления рабочей группой является создание новых групп и определение разрешений на доступ для этих групп, а не для индивидуальных пользователей. После этого изменение разрешений для отдельных пользователей осуществляется путем добавления пользователей в группы или удаления их из групп. Кроме того, при необходимости предоставить новые разрешения они предоставляются сразу всем членам группы в одной операции.

Изменять разрешения других пользователей на отдельные объекты базы данных могут следующие пользователи:

- члены [группы «Admins»](#), определенной в файле рабочей группы, который использовался при создании конкретной базы данных;
- [владелец](#) объекта;

- любой пользователь, получивший на этот объект разрешения администратора.

В ряде случаев пользователи, не имеющие разрешения на выполнение какого-либо действия, имеют возможность присвоить их себе. Такая ситуация имеет место для пользователя, являющегося членом группы «Admins» или владельцем объекта.

Пользователь, создавший таблицу, запрос, форму, отчет или макрос, является владельцем этого объекта. Кроме того, группа пользователей, которые могут изменять разрешения в базе данных, может также сменить владельца объектов или заново создать эти объекты, что является альтернативным способом смены владельца объектов. Для создания объекта заново достаточно импортировать или экспортировать этот объект в другую базу данных или сделать копию объекта. Этот прием является простейшим способом смены владельца объектов, в том числе и всей базы данных.

Примечание. Копирование, импорт или экспорт не приводит к изменению прав владельца для запросов, у которых свойство **При запуске предоставляются права (RunPermissions)** имеет значение **Владельца**. Изменение владельца допускается только для запросов, у которых свойство **При запуске предоставляются права (RunPermissions)** имеет значение **Пользователя**.

Организация учетных записей

Файл рабочей группы Microsoft Access содержит следующие встроенные учетные записи.

Учетная запись	Функция
Admin	Стандартная учетная запись пользователя . Данные записи являются одинаковыми для всех экземпляров Microsoft Access и других приложений, использующих ядро базы данных Microsoft Jet , таких как Microsoft Visual Basic для приложений (VBA) и Microsoft Excel.
Admins	Учетная запись группы администраторов. Данная запись является уникальной в каждом файле рабочей группы. По умолчанию пользователь «Admin» является членом группы «Admins» . В каждый момент в группу «Admins» должен входить по крайней мере один пользователь.
Users	Учетная запись группы, содержащая все учетные записи пользователей. Microsoft Access автоматически добавляет учетные записи пользователей в группу «Users» при их создании членом группы «Admins». Данные записи являются одинаковыми в каждом файле рабочей группы, однако этот файл содержит только учетные записи пользователей, созданные членами группы «Admins», входящими в данную рабочую группу. По умолчанию данная учетная запись имеет все разрешения на все новые объекты. Единственным способом удаления учетной записи из группы «Users» является удаление пользователя членом группы «Admins».

В действительности, система защиты Microsoft Access всегда включена. До активизации пользователем процедуры подключения для рабочей группы Microsoft Access при запуске автоматически подключает всех пользователей с помощью встроенной [учетной записи пользователя «Admin»](#) с пустым паролем. Неявно Microsoft Access использует учетную запись «Admin» как учетную запись администратора рабочей группы, а также как [владельца](#) всех создаваемых баз данных и таблиц, форм, отчетов и макросов.

Наличие администраторов и владельцев является необходимым, поскольку им предоставляются разрешения, которые невозможно отобрать.

- Администраторы (члены группы «Admins») всегда могут получить все разрешения на объекты, созданные в рабочей группе.
- Учетные записи, являющиеся владельцами таблиц, запросов, форм, отчетов или макросов всегда могут получить все разрешения на доступ к этим объектам.
- Учетная запись, являющаяся владельцем базы данных, всегда может открыть базу данных.

Поскольку учетные записи пользователя «Admin» совершенно одинаковы для всех экземпляров Microsoft Access, то первым шагом при организации системы защиты является определение учетных записей администратора и владельца (или единой учетной записи,

являющейся записью и администратора, и владельца). После этого следует удалить учетную запись пользователя «Admin» из группы «Admins». Если этого не сделать, любой пользователь Microsoft Access сможет подключиться к рабочей группе с помощью учетной записи «Admin» и получить все разрешения на доступ к таблицам, запросам, формам, отчетам и макросам рабочей группы.

В группу «Admins» разрешается добавлять произвольное число учетных записей, однако владельцем базы данных может быть только одна учетная запись — та, что была активной при создании базы данных, либо та, что была активной при передаче права владельца путем создания новой базы данных и импорта в нее всех объектов из исходной базы данных. Однако учетные записи групп могут являться владельцами таблиц, запросов, форм, отчетов и макросов базы данных.

План-конспект занятия №22

Тема: *Использование Мастера защиты*

Вид занятия: *теоретическое занятие*

Цель занятия: познакомить с Мастером защиты.

Задачи:

Учебные: 1) разъяснить основные понятия, связанные с Мастером защиты

Развивающие:

- 1) развитие логического мышления;
- 2) развитие памяти

Воспитательные:

- 1) воспитание аккуратности и внимательности
- 2) воспитание вдумчивости при принятии решений

Развиваемые общие компетенции: ОК 1, ОК 10, ОК2.1.1, ПК 2.1., ПК 2.4.

Самостоятельная работа студента: Выполнение упражнений на закрепление материала

Использование Мастера защиты

Мастер защиты позволяет установить [защиту на уровне пользователя](#) с общей схемой защиты и закодировать [базу данных Microsoft Access](#).

1. Откройте базу данных, которую требуется защитить.
2. В меню **Сервис** выберите команду **Защита**, а затем команду **Мастер**.
3. Выполняйте инструкции, выводимые в диалоговых окна мастера.

Примечания

- Мастер защиты на уровне пользователя создает резервную копию текущей базы данных с тем же самым именем и расширением имени файла .bak, а затем применяет средства защиты к выделенным объектам текущей базы данных.
- Если в текущей базе данных программы [Microsoft Visual Basic для приложений \(VBA\)](#) защищены паролем, мастер запросит ввод пароля или неудачно завершит свою работу.
- Все пароли, созданные при работе мастера, распечатываются в отчете, формируемом мастером защиты по окончании его работы. Следует хранить этот отчет в безопасном месте. Можно использовать этот отчет для повторного создания [файла рабочей группы](#).

РАЗДЕЛ 4. ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОМУ ПРАКТИКУМУ ПО МДК 02.02. Технология разработки и защиты базы данных

4.1. Общие положения

В настоящих методических указаниях приводится описание практических и лабораторных работ, выполняемых обучающимися. Каждая работа посвящена рассмотрению одного из вопросов МДК.

Описание всех лабораторных работ составлено по единому плану: сначала даются основы теории, затем формулируется цель работы, приведен порядок выполнения работы, предложены вопросы для самоконтроля понимания материала и задания для самостоятельного решения.

Предполагается, что отчет по работе должен содержать следующие пункты:

- Цель работы
- Исходные данные
- Алгоритм выполнения
- Результаты работы
- Выводы

На каждом занятии обучающийся получает задание, которое выполняет самостоятельно под руководством преподавателя. В конце каждой лабораторной работы приведены варианты заданий, контрольные вопросы и примеры, демонстрирующие способы решения поставленных задач.

Таким образом, методические указания позволяют, во-первых, интенсифицировать практическую составляющую обучения и, во-вторых, обучить студентов навыкам использования основных универсальных действий.

Методические указания могут также быть использованы для проведения лабораторно-практических занятий в лаборатории вычислительной техники.

4.2. Методические указания к лабораторному практикуму

План-конспект занятия №1

Тема : Проектирование структуры базы данных по плану (ОК 2.1.1)

Вид занятия: *практическое занятие №1*

Цель:

- познакомиться с алгоритмом проектирования структуры БД;
- научиться производить настройку свойств полей.

Ход работы:

Лабораторная работа посвящена разработке базы данных для учёта абитуриентов и результатов вступительных испытаний высшего учебного заведения. База данных должна состоять из четырёх взаимосвязанных таблиц: ***факультет, специальность, группа, контингент***.

В таблице ***факультет*** содержится информация о каждом факультете вуза, в том числе о плане набора на факультет в текущем учебном году.

В таблице ***специальность*** содержится информация о всех специальностях вуза, принадлежности каждой специальности тому или иному факультету, план набора на специальность.

В таблице ***группа*** содержится сведения об экзаменационных группах и датах экзаменов для каждой группы. Группы сформированы по факультетам. Каждая группа содержит только абитуриентов одного факультета.

В таблице ***контингент*** содержится полная информация о каждом абитуриенте, в том числе результаты вступительных испытаний.

При первом сеансе работы с базой данных необходимо создать файл базы данных. Для этого:

- На диске **C:** создадим папку **Базы данных** (в случае если папка еще не создана)
- Запустите программу Access
- В правой части окна Access выбираем «Создать файл» и затем «Новая база данных»
- В появившемся окне присваиваем новой базе данных имя, под которым мы в дальнейших сеансах работы будем открывать разрабатываемую базу данных. Файл помещаем в папку **Базы данных**

Разработка таблицы «факультет»

§ В окне **база данных** выбрать вкладку **Таблицы**.

§ Из трёх предлагаемых способов разработки таблицы выбираем режим – **Конструктор**.

§ Создать в таблице поля с именами:

АббревиатураФакультета;

НазваниеФакультета;

ПланНабораФакультета.

Примечание: В названии полей исключить пробелы. Имена полей заносятся в виде первого столбца таблицы. Типы полей определяются во втором столбце таблицы.

§ Установить для каждого поля следующие типы данных:

для поля *АббревиатураФакультета* – **Текстовый**;

для поля *НазваниеФакультета* – **Текстовый**;

для поля *ПланНабораФакультета* – **Числовой**.

Установить поле *АббревиатураФакультета* – **ключевым**, для чего установить на данное поле курсор и на панели инструментов **Стандартная** нажать кн. **Ключевое поле**. После чего рядом с именем поля должно появиться изображение ключа.

§ Используя окно «**свойство поля**» для созданных полей определим их свойства.

Вначале определим **общие** свойства. Перечень этих свойств приведен в окне свойств.

Для каждого введённого выше поля определяем те свойства из списка, которые необходимы с точки зрения возможности размещения данных и будущих операций в разрабатываемой базе данных.

Определяем свойства для текстового поля *АббревиатураФакультета*.

Размер поля – *3 символа*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения не допускаются)*

Определяем свойства для текстового поля *НазваниеФакультета*.

Размер поля – *40 символов*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения не допускаются)*

Определяем свойства для числового поля *ПланНабораФакультета*.

Размер поля – *целое*

Обязательное поле – *да*

Индексированное поле - *да (совпадения допускаются)*

В результате указанных действий должна быть заполнена форма приведенная ниже. Сохраним разработанную таблицу под именем **факультет**.

Разработка таблицы «специальность»

§ В окне **база данных** выбрать вкладку **Таблицы**.

§ Из трёх предлагаемых способов разработки таблицы выбрать режим – **Конструктор**.

§ Создать в таблице поля с именами:

АббревиатураСпециальности;

АббревиатураФакультета;

НазваниеСпециальности;

ПланНабораСпециальности.

- Установить для каждого поля следующие типы данных:
для поля *АббревиатураСпециальности* – **Текстовый**;
для поля *АббревиатураФакультета* – **Текстовый**;
для поля *НазваниеСпециальности* – **Текстовый**;
для поля *ПланНабораСпециальности* – **Числовой**.

Установить поле *АббревиатураСпециальности* – **ключевым**.

§ Используя окно «**свойство поля**» определить свойства полей таблицы.

Определяем свойства для текстового поля *АббревиатураСпециальности*.

Размер поля – *3 символа*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения не допускаются)*

Определяем свойства для текстового поля *АббревиатураФакультета*.

Размер поля – *3 символа*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения допускаются)*

Определяем свойства для текстового поля *НазваниеСпециальности*.

Размер поля – *40 символов*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения не допускаются)*

Определяем свойства для числового поля *ПланНабораСпециальности*.

Размер поля – *целое*

Обязательное поле – *да*

Индексированное поле - *да (совпадения допускаются)*

Сохраним разработанную таблицу под именем **специальность**

Разработка таблицы «группы»

§ В окне **база данных** выбрать вкладку **Таблицы**.

§ Из трёх предлагаемых способов разработки таблицы выбрать режим –

Конструктор.

§ Создать в таблице поля с именами:

НомерГруппы;

АббревиатураФакультета;

ДатаЭкзамена;

· Установить для каждого поля следующие типы данных:

для поля *НомерГруппы* – **Текстовый**;

для поля *АббревиатураФакультета* – **Текстовый**;

для поля *ДатаЭкзамена* – **Дата/время**.

Установить поле *НомерГруппы* **ключевым**.

§ Используя окно «**свойство поля**» определить свойства полей таблицы.

Определяем свойства для текстового поля *НомерГруппы*.

Размер поля – *3 символа*

Обязательное поле – *да*

Пустые строки – *нет*

Индексированное поле - *да (совпадения не допускаются)*

Определяем свойства для текстового поля *АббревиатураФакультета*.

Размер поля – *3 символа*

Обязательное поле – *да*

Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
Определяем свойства для текстового поля *ДатаЭкзамена*.
Размер поля – *Краткий формат даты*
Маска ввода – *00.00.0000;0;_ (вводится выбором Краткий формат даты)*
Значение по умолчанию - *#17.07.2009#*
Условие на значение – *Between #30.06.2009# AND #30.07.2009#*
Сообщение об ошибке – *«Дата вне заданного диапазона»*
Обязательное поле – *да*
Индексированное поле - *да (совпадения допускаются)*

Сохраним разработанную таблицу под именем **группы**

Разработка таблицы «контингент»

Добавим в базу данных третью таблицу **контингент** путем импорта в Access таблицы Excel, из файла **списки.xls** в папке *Базы данных*. Для этого в окне базы данных щелкните правой клавишей для вызова контекстного меню.

В контекстном меню выбираем **Импорт**. Указываем **сохранить в новой таблице, создать автоматически ключ**. В качестве имени таблицы указываем **контингент**.

Открываем вновь созданную таблицу в режиме конструктора. Проверим наименования полей в новой таблице, созданной путем импорта таблицы Excel.

КодАбитуриента;
Фамилия;
Имя;
Отчество;
АббревиатураСпециальности;
НомерГруппы;
Пол;
Общежитие;
ДатаРождения;
ГодОкончанияШколы;
Медаль;
ОценкаМатематика;
ОценкаФизика;
ОценкаРусский;
СреднийБаллАттестата;
ПризнакЗачисления.

Установить поле *КодАбитуриента* **ключевым**.

- Установить для каждого поля следующие типы данных:
 - для поля *КодАбитуриента* - **Счётчик**
 - для поля *Фамилия* - **Текстовый**
 - для поля *Имя* - **Текстовый**
 - для поля *Отчество* - **Текстовый**
 - для поля *АббревиатураСпециальности* - **Текстовый**
 - для поля *НомерГруппы* - **Текстовый**
 - для поля *Пол* - **Текстовый**
 - для поля *Общежитие* - **Логический**
 - для поля *ДатаРождения* - **Дата/Время**
 - для поля *ГодОкончанияШколы* - **Числовой**
 - для поля *Медаль* - **Логический**
 - для поля *ОценкаМатематика* - **Числовой**
 - для поля *ОценкаФизика* - **Числовой**
 - для поля *ОценкаРусский* - **Числовой**

для поля *СреднийБалАттестата*- **Числовой**
для поля *ПризнакЗачисления*- **Логический**

- § Используя окно «**свойство поля**» определить свойства полей таблицы.
- Определяем свойства для поля *КодАбитуриента*.
Размер поля – *ДлинноеЦелое*
Новые значения – *Последовательные*
Индексированное поле - *да (совпадения не допускаются)*
- Определяем свойства для текстового поля *Фамилия*.
Размер поля – *30 символов*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для текстового поля *Имя*
Размер поля – *15 символов*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для текстового поля *Отчество*
Размер поля – *15 символов*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для текстового поля *АббревиатураСпециальности*
Размер поля – *3 символа*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для текстового поля *НомерГруппы*
Размер поля – *3 символа*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для текстового поля *Пол*
Размер поля – *1 символ*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для логического поля *Общежитие*
Формат поля – *Да/нет*
Обязательное поле – *да*
- Определяем свойства для текстового поля *ДатаРождения*.
Размер поля – *Краткий формат даты*
Маска ввода – *00.00.0000;0;_ (вводится выбором Краткий формат даты)*
Обязательное поле – *да*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для числового поля *ГодОкончанияШколы*.
Размер поля – *целое*
Обязательное поле – *да*
Пустые строки – *нет*
Индексированное поле - *да (совпадения допускаются)*
- Определяем свойства для логического поля *Медаль*
Формат поля – *Да/нет*
Обязательное поле – *да*

Определяем свойства для числового поля *ОценкаМатематика*.

- Размер поля – *целое*
- Подпись - *Математика*
- Обязательное поле – *Нет*
- Индексированное поле - *нет*

Определяем свойства для числового поля *ОценкаФизика*.

- Размер поля – *целое*
- Подпись - *Физика*
- Обязательное поле – *нет*
- Индексированное поле - *нет*

Определяем свойства для числового поля *ОценкаРусский*.

- Размер поля – *целое*
- Подпись - *Русский*
- Обязательное поле – *нет*
- Индексированное поле - *нет*

Определяем свойства для числового поля *СреднийБалАттестата*.

- Размер поля – *Одинарное с плавающей точкой*
- Формат поля – *фиксированный*
- Подпись – *Ср. бал ат.*
- Число десятичных знаков - *1*
- Обязательное поле – *да*
- Индексированное поле - *да (совпадения допускаются)*

Определяем свойства для логического поля *ПризнакЗачисления*

- Формат поля – *Да/нет*
- Подпись - *Зачислен*
- Обязательное поле – *нет*

Закройте окно конструктора таблиц. В окне **Базы данных** должны быть указаны четыре созданные таблицы.

Ввод данных в таблицы

1. Из списка объектов левой части окна **Базы данных** выберете тип объекта «Таблица». Из четырёх созданных таблиц выберете таблицу **Факультеты**. Откройте её двойным щелчком клавиши мышки, либо выбрав опцию «открыть» в меню окна **Базы данных**.

Введите данные в таблицу. В качестве данных введите следующие значения:

Аббревиатура	Название факультета	План набора
ктэ	Компьютерных технологий и электроники	100
эу	Экономики и управления	75
ээ	Энергетики и электротехники	125

2. Из списка объектов левой части окна **Базы данных** выберете тип объекта «Таблица». Из четырёх созданных таблиц выберете таблицу **Специальности**. Откройте её в режиме **Конструктора**.

Для поля *АббревиатураФакультета* вместо типа данных **Текст** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Таблицы**. В появившемся списке таблиц выбираем таблицу **Факультеты**, в качестве доступного поля выбираем *АббревиатураФакультета*. В окне **Порядок сортировки** выбрать поле *АббревиатураФакультета* по возрастанию. Подтвердите сохранение таблицы.

3. Откройте таблицу **Специальности** в режиме **Таблица**. Введите данные, используя значения:

Аббревиатура Специальности	Аббревиатура Факультета	Название специальности	План набора специальности
ас	ктэ	Автоматика и системотехника	25

бу	эу	Бухучёт	20
вм	ктэ	Вычислительные машины	25
м	эу	Менеджмент	20
оэс	ктэ	Оптико электронные системы	25
пи	эу	Прикладная информатика	25
пэ	ктэ	Промышленная электроника	25
э	ээ	Электрические сети	25
эм	ээ	Электрические машины	30
эо	ээ	Электроснабжение	25
эп	ээ	Электропривод	20
эс	ээ	Электрические системы	25

Откройте таблицу **Группы** в режиме Таблица.

Для поля *АббревиатураФакультета* вместо типа данных **Текст** выберите **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберите значения **Таблицы**. В появившемся списке таблиц выбираем таблицу **Факультеты**, в качестве доступного поля выбираем *АббревиатураФакультета*. В окне **Порядок сортировки** выбрать поле *АббревиатураФакультета* по возрастанию. Подтвердите сохранение таблицы.

Введите данные в таблицу **Группы**, используя значения:

Группа	факультет	Дата экзамена	Группа	факультет	Дата экзамена
001	ээ	01.07.2014	031	ктэ	16.07.2014
002	ээ	01.07.2014	032	ктэ	16.07.2014
003	ээ	01.07.2014	033	ктэ	17.07.2014
004	ээ	01.07.2014	034	ктэ	17.07.2014
005	ээ	01.07.2014	041	эу	30.06.2014
006	ээ	01.07.2014	042	эу	01.07.2014
011	ээ	16.07.2014	043	эу	01.07.2014
012	ээ	16.07.2014	044	эу	01.07.2014
013	ээ	17.07.2014	045	эу	01.07.2014
014	ээ	17.07.2014	051	эу	18.07.2014
015	ээ	17.07.2014	052	эу	18.07.2014
016	ээ	17.07.2014	053	эу	18.07.2014
021	ктэ	30.06.2014	061	эу	17.07.2014
022	ктэ	18.07.2014	062	эу	17.07.2014
023	Ктэ	18.07.2014	063	эу	17.07.2014
024	ктэ	18.07.2014	064	эу	17.07.2014
025	ктэ	18.07.2014	065	эу	17.07.2014
026	ктэ	18.07.2014	066	эу	17.07.2014

4. Откройте таблицу **Контингент** в режиме Таблица.

Для поля *АббревиатураСпециальности* вместо типа данных **Текст** выберите **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберите значения **Таблицы**. В появившемся списке таблиц выбираем таблицу **Специальности**, в качестве доступного поля выбираем *АббревиатураСпециальности*. В окне **Порядок сортировки** выбрать поле *АббревиатураСпециальности* по возрастанию. Подтвердите сохранение таблицы.

Для поля *НомерГруппы* вместо типа данных **Текст** выберите **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберите значения **Таблицы**. В появившемся списке таблиц выбираем таблицу **Группы**, в качестве доступного поля выбираем *НомерГруппы*. В окне **Порядок сортировки** выбрать поле *НомерГруппы* по возрастанию. Подтвердите сохранение таблицы.

Для поля *Пол* вместо типа данных **Текст** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Фиксированный набор значений**. В появившейся колонке вводим две строчки со значениями: м и ж.. Подтвердите сохранение таблицы.

Для поля *Медаль* вместо типа данных **Логический** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Фиксированный набор значений**. В появившейся колонке вводим две строчки со значениями: да и нет. Подтвердите сохранение таблицы.

Для поля *Общежитие* вместо типа данных **Логический** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Фиксированный набор значений**. В появившейся колонке вводим две строчки со значениями: да и нет. Подтвердите сохранение таблицы.

Для поля *ОценкаМатематика* вместо типа данных **Числовой** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Фиксированный набор значений**. В появившейся колонке вводим четыре строчки со значениями: 5,4,3,2. Подтвердите сохранение таблицы.

Для полей *ОценкаФизика* и *ОценкаРусский* выполнить такие же подстановки.

Для поля *ПризнакЗачисления* вместо типа данных **Логический** выберете **Мастер подстановок**. В открывшемся окне в качестве объекта подстановки выберете значения **Фиксированный набор значений**. В появившейся колонке вводим две строчки со значениями: да и нет. Подтвердите сохранение таблицы.

План-конспект занятия №2

Тема **Нормализация таблиц**

Вид занятия: *практическое занятие №2*

Цель работы. Изучить вопросы, связанные с нормализацией данных, созданием таблиц в БД и заданием схемы данных.

Описание базы данных "Склад"

База данных предназначена для учета товаров, которые поступают по заказам на оптовый склад. *Сотрудник* оформляет *закупку* товаров нескольких наименований у одного *поставщика*. Общая закупка включает в себя несколько *сделок*, отображаемых набором сведений о закупке каждого *товара*. Каждый товар относится к тому или иному *типу* товаров. Закупленные товары должны прибыть на склад при помощи одного из возможных способов *доставки*.

При создании базы выделяем следующие сущности:

Товары - содержит сведения о товарах;

Типы - справочник групп (типов) товаров;

Сделки - содержит сведения о заказах (проведенных операциях по закупке) каждого из товаров;

Закупки - содержит сведения о заказах нескольких товаров от одного поставщика;

Сотрудники - содержит сведения о сотрудниках, оформивших заказ;

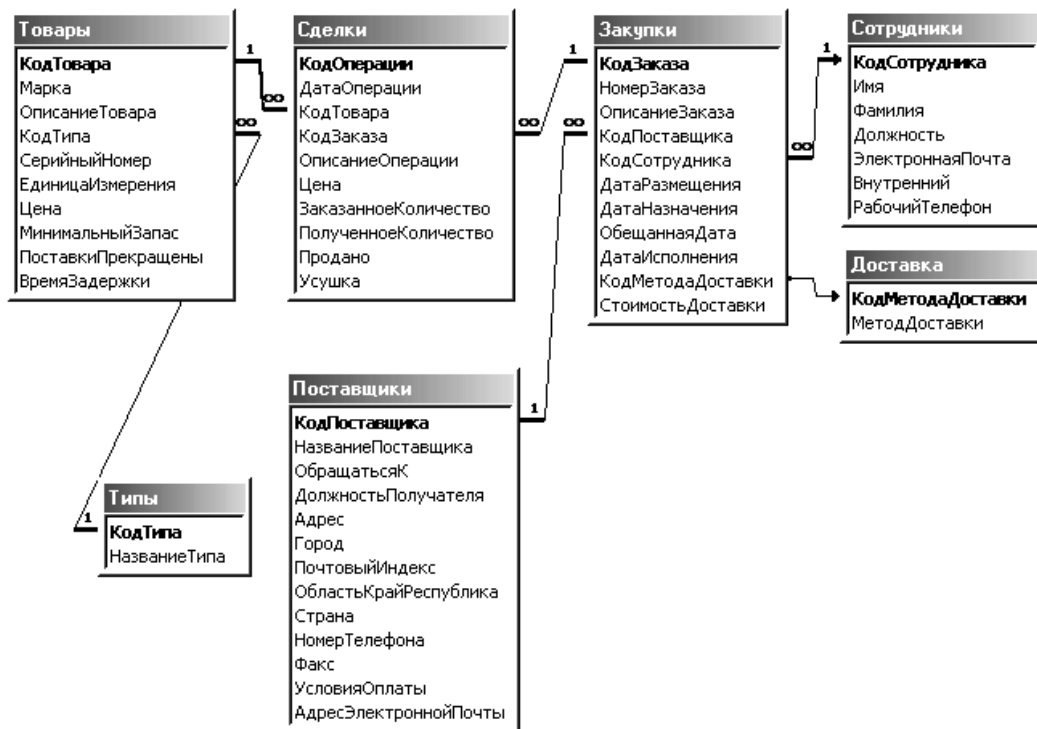
Поставщики - содержит сведения о каждой организации - поставщике товаров

Доставка - справочник видов доставки.

Порядок выполнения.

1. Создать таблицы БД (список таблиц приведен ниже). Задать требуемые свойства полей.
2. Создать схему данных.
3. Сделать подстановки в связанных полях
4. Ввести тестовые данные (по 10-20 строк в таблицы "Товары" и др., по 5 строк в таблицы "Типы" и др.

Схема данных



Список таблиц и свойств полей таблиц
 Таблица "Товары"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодТовара	Счетчик	Дл целое	Код товара	Да (Совп. не доп.)		
Марка	Текстовый	50	Марка	Да (Совп. доп.)	Нет	
ОписаниеТовара	Текстовый	255	Описание товара	Нет	Нет	
КодТипа	Числовой	Дл целое	Код типа	Да (Совп. доп.)	Нет	
СерийныйНомер	Текстовый	50	Серийный номер	Да (Совп. доп.)	Нет	
Цена	Денежный	Денежн ый	Цена	Нет	Нет	
ЕдиницаИзмерения	Текстовый	50	Единица измерения	Нет	Нет	
МинимальныйЗапас	Числовой	Дл целое	Минимальный запас	Нет	Нет	
ПоставкиПрекращены	Логический	Да/Нет	Поставки прекращены	Нет	Нет	
ВремяЗадержки	Текстовый	30	Срок	Нет	Нет	

Таблица "Типы"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодТипа	Счетчик	Дл целое	Код типа	Да (Совп. не доп.)		
НазваниеТипа	Текстовый	50	Категория	Да (Совп. доп.)	Нет	

Таблица "Сделки"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодОперации	Счетчик	Дл целое	Код операции	Да (Совп. не доп.)		
ДатаОперации	Дата/время	Краткий формат даты	Дата операции	Да (Совп. доп.)	Нет	Маска ввода 99.99.00;0
КодТовара	Числовой	Дл целое	Код товара	Да (Совп. доп.)	Нет	
КодЗаказа	Числовой	Дл целое	Код заказа	Да (Совп. доп.)	Нет	
ОписаниеОперации	Текстовый	255	Описание операции	Нет	Нет	
Цена	Денежный	Денежный	Цена	Нет	Нет	
ЗаказанноеКоличество	Числовой	Длинное целое	Заказанное количество	Нет	Нет	
ПолученноеКоличество	Числовой	Длинное целое	Полученное количество	Нет	Нет	
Продано	Числовой	Длинное целое	Продано	Нет	Нет	
Усушка	Числовой	Длинное целое	Усушка	Нет	Нет	

Таблица "Закупки"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодЗаказа	Счетчик	Дл целое	Код заказа	Да (Совп. не доп.)		
НомерЗаказа	Текстовый	30	Номер Заказа	Нет	Нет	
ОписаниеЗаказа	Текстовый	255	Описание Заказа	Нет	Нет	
КодПоставщика	Числовой	Дл целое	Код поставщика	Да (Совп. доп.)	Нет	
КодСотрудника	Числовой	Дл целое	Код сотрудника	Да (Совп. доп.)	Нет	
ДатаРазмещения	Дата/время	Краткий формат даты	Дата размещения	Да (Совп. доп.)	Нет	Маска ввода 99.99.00;0
ДатаНазначения	Дата/время	Краткий формат даты	Дата назначения	Нет	Нет	Маска ввода 99.99.00;0
ОбещаннаяДата	Дата/время	Краткий формат даты	Обещанная дата	Нет	Нет	Маска ввода 99.99.00;0
ДатаИсполнения	Дата/время	Краткий формат даты	Дата исполнения	Нет	Нет	Маска ввода 99.99.00;0
КодМетодаДоставки	Числовой	Длинное целое	Код доставки	Да (Совп. доп.)	Нет	
СтоимостьДоставки	Денежный	Денежный	Цена доставки	Нет	Нет	

Таблица "Поставщики"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодПоставщика	Счетчик	Дл целое	Код поставщика	Да (Совп. не доп.)		

НазваниеПоставщика	Текстовый	50	Название	Да (Совп. доп.)	Нет	
ОбращатьсяК	Текстовый	50	Обращаться к	Да (Совп. доп.)	Нет	
ДолжностьПолучателя	Текстовый	50	Должность	Нет	Нет	
Адрес	Текстовый	255		Нет	Нет	
Город	Текстовый	50		Нет	Нет	
ПочтовыйИндекс	Текстовый	20	Индекс	Да (Совп. доп.)	Нет	
ОбластьКрайРеспублика	Текстовый	20	Регион	Нет	Нет	
Страна	Текстовый	50		Нет	Нет	
НомерТелефона	Текстовый	30	Телефон	Нет	Нет	
Факс	Текстовый	30		Нет	Нет	
УсловияОплаты	Текстовый	255	Условия оплаты	Нет	Нет	
АдресЭлектроннойПочты	Текстовый	50	Электронная почта	Да (Совп. доп.)	Нет	

Таблица "Доставка"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодМетодаДоставки	Счетчик	Дл целое	Код доставки	Да (Совп. не доп.)		
МетодДоставки	Текстовый	20	Метод доставки	Нет	Нет	

Таблица "Сотрудники"

Имя поля	Тип данных	Размер/Формат	Подпись	Индексир. поле	Обязат. поле	Другие
КодСотрудника	Счетчик	Дл целое	Код сотрудника	Да (Совп. не доп.)		
Имя	Текстовый	50		Нет	Нет	
Фамилия	Текстовый	50		Да (Совп. доп.)	Нет	
Должность	Текстовый	50		Нет	Нет	
ЭлектроннаяПочта	Текстовый	50	Имя электронной почты	Да (Совп. доп.)	Нет	
Внутренний	Текстовый	30		Нет	Нет	
РабочийТелефон	Текстовый	30		Нет	Нет	

Содержание отчета.

1. Описание сущностей и их атрибутов.
2. Описание таблиц и определение свойств полей таблиц.
3. Модель данных «сущность – связь».
4. Описание типов связей и связываемых полей.

Контрольные вопросы

1. Основные понятия информационно-логических моделей: сущность, атрибут (реквизит), информационный объект, информационный элемент.
2. Основные понятия баз данных: поле, запись, таблица.
3. Что такое нормализация данных?
4. Ключи в БД (простые, составные, первичные, вторичные, внешние) и их назначение.
5. Индексация таблиц и ее назначение.
6. Какими свойствами обладают таблицы, которые находятся в 1-й, во 2-й и в 3-й нормальной форме?
7. Какими свойствами обладают связи "один-к-одному", "один-ко-многим", "мноغو-ко-

многим"?

- Каковы свойства внутреннего объединения таблиц, левого и правого внешнего объединений?
- Реализация отношения "много-ко-многим" в базе данных.

План-конспект занятия №3

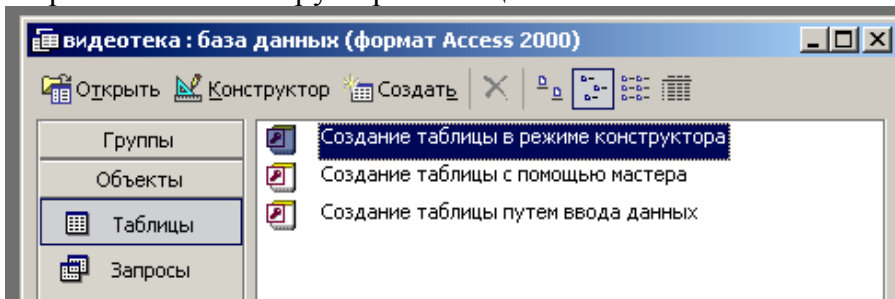
Тема: **Создание новой базы данных разными способами с использованием методических указаний (ОК 2.1.1, ОК 2.2.1)**

Вид занятия: лабораторное занятие №1

Цель: создание таблицы в режиме конструктор в режиме таблицы при работе с БД в среде MS Access.

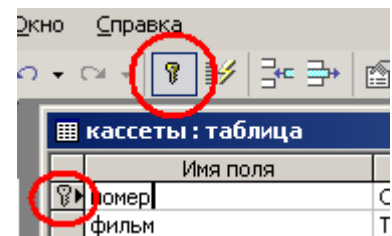
План выполнения работы:

- Запустите MS Access. (Пуск → Программы → MS Access)
- Выполните команду **Файл → Создать → Новая база данных**. В файловом окне укажите имя файла «Видеотека».
- В режиме **Таблицы** выберите **Создание таблицы в режиме конструктор**. На экране откроется окно конструктора таблиц.



- Задайте структуру таблицы БД «Видеотека»: Введите имена полей и описание, выберите Типы данных, щелкнув мышкой по строчке Типы данных. Появится раскрывающаяся кнопка, щелкните по ней. В появившемся списке выберите тип.

Имя поля	Тип данных	
номер	Счетчик	номер кассеты
фильм	Текстовый	название фильма
код страны	Числовой	код страна-производительность
время	Числовой	продолжительность
код жанра	Числовой	



- Выбор первичного ключа.
 - Выберите поля для первичного ключа
 - Нажмите на кнопку **Ключевое поле** на панели инструментов или выберите команду **Правка - Ключевое поле**.

Напротив ключевого поля должен появиться ключик

- Закрываем режим конструктора и задаем имя таблицы «Кассеты».
- Повторив шаги с 3 по 6 создаем таблицу «Страна», «Жанры».



Таблица «Страна»

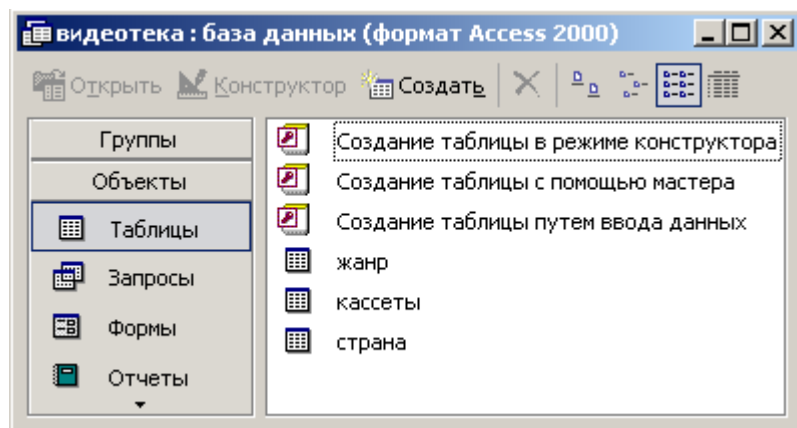
Имя поля	Тип данных	Описание
 Код страны	Числовой	Код страны

Страна	Текстовый	Название страны
--------	-----------	-----------------

Таблица «Жанр»

8. В результате получите базу данных «Видеотека», состоящую из трех таблиц: Кассеты, Страна и Жанр.

Имя поля	Тип данных	Описание
 Код жанра	Числовой	Код жанра
Жанр	Текстовый	Название жанра



План-конспект занятия №4

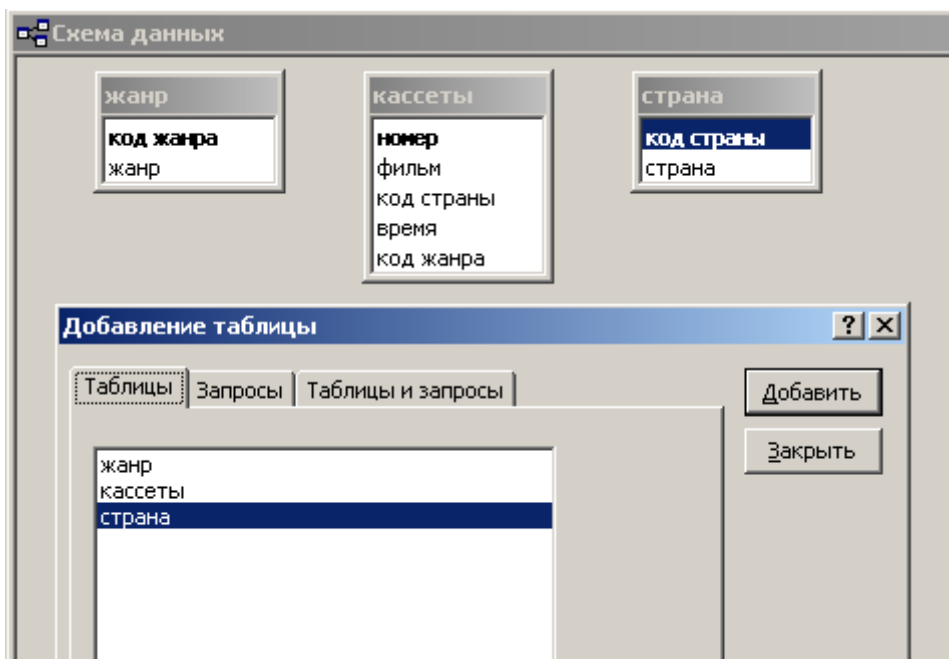
Тема: *Создание таблиц в различных режимах. Создание таблиц сложной структуры*

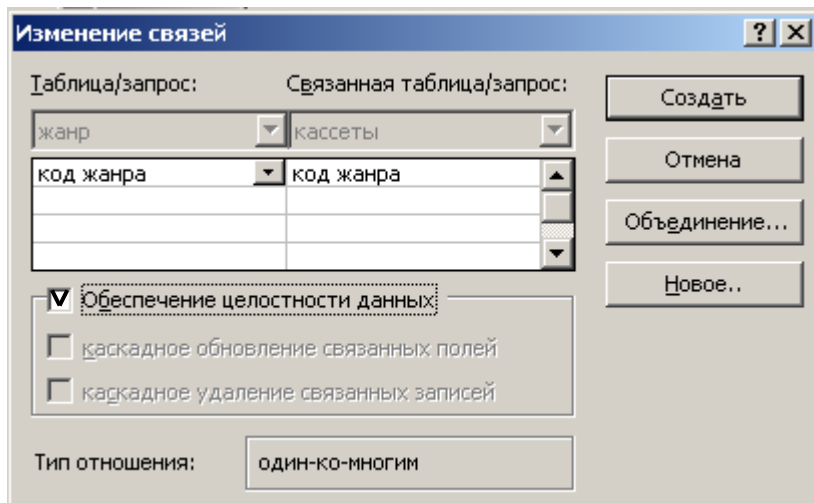
Вид занятия: *лабораторное занятие №2*

Цель: создание связей между таблицами в режиме таблиц при работе с БД в среде MS Access.

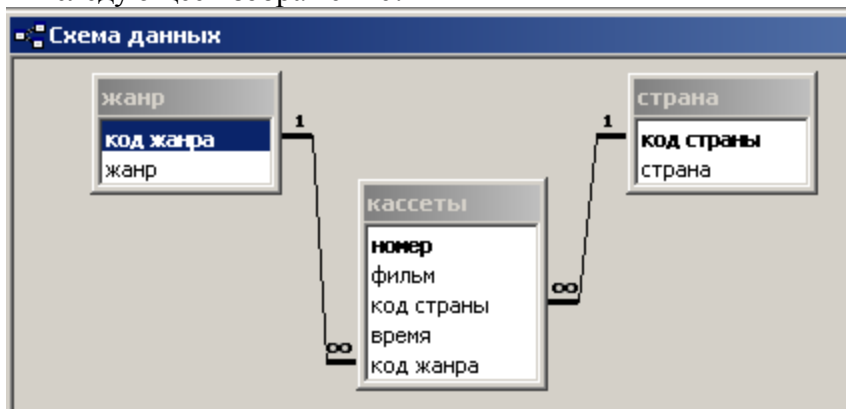
План выполнения работы:

1. Выбрать **Сервис-Схема данных**
2. Выделить таблицу “Жанр” и щелкнуть по кнопке **Добавить**.
3. Выделить таблицу “Страна” и щелкнуть по кнопке **Добавить**.
4. Выделить таблицу “Кассеты” и щелкнуть по кнопке **Добавить**.
5. Щелкнуть по кнопке **Заккрыть**.
6. Щелкнуть на поле “Код жанра” в таблице “Жанр” и держа нажатой переместить на поле “Код жанра” в таблице “Кассеты”. Включить опцию “Обеспечение целостности”. Щелкнуть по кнопке **Создать**.





7. Щелкнуть на поле “Код страны” в таблице “Страна” и держа нажатой переместить на поле “Код страны” в таблице “Кассеты”. Включить опцию “Обеспечение целостности”. Щелкнуть по кнопке **Создать**. Если все действия выполните правильно, то появится следующее изображение:



8. Закрыть окно схемы данных, сохранив.

План-конспект занятия №5

Тема: *Изменение свойств полей таблицы и форматов отображения данных.*

Вид занятия: *лабораторное занятие №3*

Используя разработанную в Практическом занятии №3 модель, создать таблицы, изменяя свойства полей согласно модели:

План-конспект занятия №6

Тема: *Создание маски ввода и полей подстановки с использованием справочных данных (ОК 4.1.2)*

Вид занятия: *лабораторное занятие №4*

Используя разработанную в Практическом занятии №3 модель, создать таблицы, используя подстановки и маски ввода:

Таблица подстановок в связанных полях

Таблица	Товары	Сделки	Закупки
Поле	Код типа	Код товара	Код поставщика
Подстановки			

Тип элемента управления	Поле со списком	Поле со списком	Поле со списком
Тип источника строк	Таблица или запрос	Таблица или запрос	Таблица или запрос
Источник строк	SELECT [Типы].* FROM [Типы] ORDER BY [Типы].[НазваниеТипа];	Товары	SELECT [Поставщики].* FROM [Поставщики] ORDER BY [Поставщики].[НазваниеПоставщика];
Присоединенный столбец	1	1	1
Число столбцов	2	2	2
Заглавия столбцов	Нет	Да	Нет
Ширина столбцов	0см;5см	1см;5см	0см;5см
Число строк списка	8	8	8
Ширина списка	5	5	5
Ограничиться списком	Да	Да	Да

Таблица	Закупки	Закупки
Поле	Код сотрудника	Код метода доставки
Подстановки		
Тип элемента управления	Поле со списком	Поле со списком
Тип источника строк	Таблица или запрос	Таблица или запрос
Источник строк	SELECT [КодСотрудника], [Фамилия]&"&[Имя] FROM [Сотрудники] ORDER BY [Фамилия]&"&[Имя];	SELECT [Доставка].* FROM [Доставка] ORDER BY [Доставка].[МетодДоставки];
Присоединенный столбец	1	1
Число столбцов	2	2
Заглавия столбцов	Нет	Нет
Ширина столбцов	0см;5см	0см;5см
Число строк списка	8	8
Ширина списка	5	5
Ограничиться списком	Да	Да

План-конспект занятия №7

Тема: Использование методов сортировки, поиска и фильтрации данных в таблице.

Вид занятия: лабораторное занятие №5

Цель занятия: Приобрести навыки сканирования графических изображений сканером Epson; Освоить приёмы работы с программой оптического распознавания текстов.

Отсортировать таблицу "Специальность" поле "Специальность" по возрастанию. На панели выбрать команду "Главная" вкладка **Сортировка и фильтр**. [Рисунок 9.7.](#)

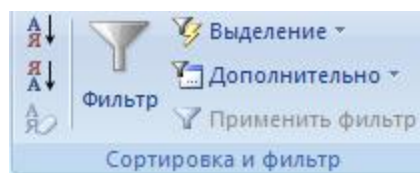


Рис. 9.7.

- С помощью фильтра отобразить следующие специальности "Управление и инновации", "Программирование", "Радиотехника". Для этого следует применить оператор "or" ([рис. 9.8](#)) или воспользоваться кнопкой "Или".

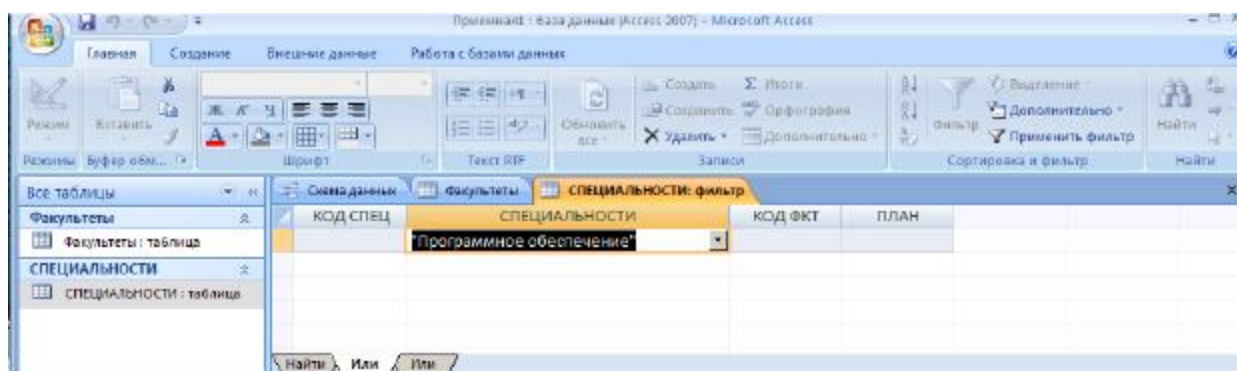


Рис. 9.8.

- С помощью фильтра отобразить специальности находящиеся в диапазоне З до Р, где З и Р начальные буквы названия специальности. Для этого используется оператор Between "З*" and "С*" следует учитывать что С* в этот диапазон не входит, З* и С* обозначает, что это первая буква текстового поля ([рис. 9.9](#)).

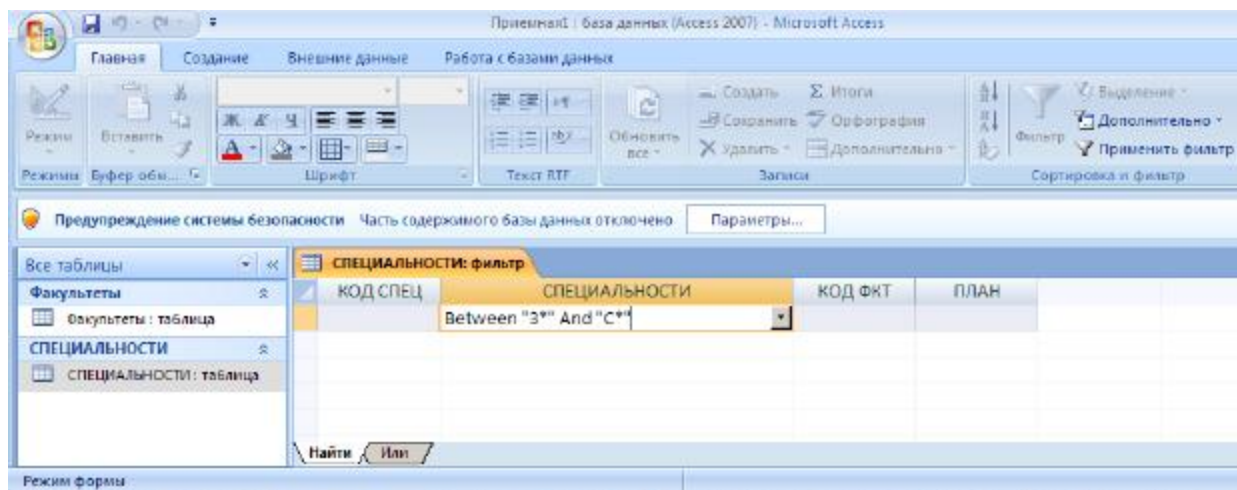


Рис. 9.9.

Самостоятельно:

1. Отсортируйте таблицу Специальность по возрастанию кодов специальности.
2. С помощью фильтра отобразите специальности, которые относятся к факультетам 01 и 04.
3. С помощью фильтра отобразите специальности, которые находятся в диапазоне от С до У где С и У начальные буквы текстового поля.

Тема: Установление взаимосвязей между таблицами.

Вид занятия: лабораторное занятие №6

Цель занятия: Приобрести навыки работы со схемой данных.

Подготовка к работе

Изучить литературу о СУБД Microsoft Access, приемах работы и создание форм и их назначении.

Контрольные вопросы

1. Создание формы.
2. Понятие простого, составного и внешнего ключей.
3. Охарактеризовать и привести примеры связей 1:1, 1:M, M:M.
4. Способы ввода и редактирования и обработки данных в формах БД.
5. Способы создания форм.
6. Типы форм.

Все действия по созданию новых таблиц производим в той базе данных, которую мы создали на первой лабораторной работе №5.

1. Создать [таблицу 10.1 "АБИТУРИЕНТЫ"](#) следующей структуры, ключевое поле РЕГ_НОМ:

Таблица 10.1. АБИТУРИЕНТЫ			
Имя поля	Тип поля	Размер поля	
РЕГ_НОМ	Счетчик	Длинное целое	
КОД_СПЕЦ	Текстовой	6	Выполнить как поле подстановки
МЕДАЛЬ	Логический	Да/Нет	
СТАЖ	Числовой	Одинарное с плавающей точкой	

2. Поле подстановки выполняется, как показано на [рис. 10.1](#)

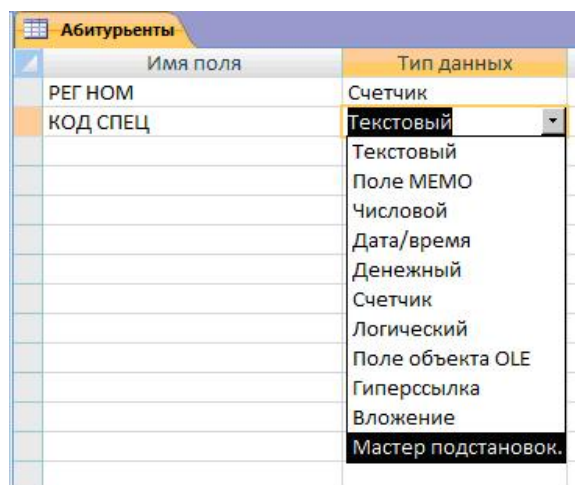


Рис. 10.1.

3. Создать [таблицу 10.2](#) "АНКЕТЫ" следующей структуры, ключевое поле РЕГ_НОМ:

Таблица 10.2. АНКЕТЫ		
Имя поля	Тип поля	Размер поля
РЕГ_НОМ	Счетчик	Длинное целое
ФАМИЛИЯ	Текстовой	30
ИМЯ	Текстовой	20
ОТЧЕСТВО	Текстовой	20
ДАТА_РОЖДЕНИЯ	ДАТА	Краткий формат даты
ГОРОД	Текстовой	30
УЧ_ЗАВЕДЕНИЕ	Текстовой	50

4. Организовать связь таблиц **АБИТУРИЕНТЫ** с таблицей **СПЕЦИАЛЬНОСТИ** через поле **КОД_СПЕЦ**, а таблицы **АНКЕТЫ** и **АБИТУРИЕНТЫ** через поле **РЕГ_НОМ**. Обратите внимание, что создается два типа связей **КОД_СПЕЦ** с **КОД_СПЕЦ** связь один ко многим, а **РЕГ_НОМ** с **РЕГ_НОМ** связь один к одному.
5. Ввести записи в таблицу **АБИТУРИЕНТЫ**. Содержание таблицы показано на [рис. 10.2](#).

РЕГ_НОМ	КОД_СПЕЦ	МЕДАЛЬ	СТАЖ
1	030502	<input type="checkbox"/>	1
2	090106	<input checked="" type="checkbox"/>	
3	210302	<input type="checkbox"/>	0,5
4	210401	<input type="checkbox"/>	
5	210402	<input type="checkbox"/>	
6	210403	<input checked="" type="checkbox"/>	
7	210405	<input type="checkbox"/>	
8	210406	<input checked="" type="checkbox"/>	
9	220601	<input type="checkbox"/>	2
10	230105	<input type="checkbox"/>	
11	230201	<input type="checkbox"/>	1
12	210302	<input checked="" type="checkbox"/>	
13	210403	<input type="checkbox"/>	0,5
14	090106	<input checked="" type="checkbox"/>	
15	230105	<input type="checkbox"/>	
16	230201	<input checked="" type="checkbox"/>	
17	210406	<input type="checkbox"/>	
18	210302	<input checked="" type="checkbox"/>	
19	030502	<input type="checkbox"/>	
20	210405	<input checked="" type="checkbox"/>	
*	(№)	<input type="checkbox"/>	

Рис. 10.2.

Пояснения логическое значение **Истина** обозначена галочкой, **ЛОЖЬ** - пустым квадратом.

Тема: Создание новой базы данных по индивидуальным заданиям (ОК 3.1.1)

Вид занятия: лабораторное занятие №7

Цель занятия: Закрепление навыков создания БД

Задание:

Для вашего варианта (см. Приложение 1) создать базу данных в MS Access. Составить базовую таблицу. Определить **ключевые атрибуты**. Используя **нормализацию** базовой таблицы разработать структуру базы данных, состоящей не менее чем из 5 таблиц, предусмотреть необходимые связи между таблицами. Определить все поля данных (**атрибуты** или столбцы) таблиц. С учетом зависимостей между **атрибутами**, установить связи и создать требуемую **схему БД**. Применить необходимые методы поддержки целостности при описании связей. Ввести не менее четырех ограничений.

При создании таблиц использовать метод построения таблиц с помощью мастера и/или вручную – полностью самостоятельно. При создании таблиц учесть все ограничения, которые заданы в постановке задачи. Предусмотреть необходимые связи между таблицами, определить все поля данных (атрибуты или столбцы) таблицы. Задать схему БД. Установить требуемые связи (между пятью таблицами) и создать требуемую схему БД. Применить необходимые методы поддержки целостности при описании связей.

Контрольные вопросы:

1. Дайте определения базы данных и реляционной базы данных.
2. Перечислите основные функции СУБД.
3. В файле какого типа сохраняется база данных Access?
4. Какие способы создания таблиц существуют в Access.
5. Что такое объект типа “таблица”, как его можно создать, как задать свойства объекта “таблица”, отличные от свойств его отдельных полей?
6. Какой режим позволяет наиболее полно определить параметры структуры таблицы?
7. Перечислите основные свойства поля таблицы.
8. Что такое первичный ключ?
9. Как связано понятие внешний ключ (Foreign key) с поддержкой целостности в СУБД MS Access?
10. Дайте характеристику свойству «Формат поля».
11. Каким образом осуществляется настройка свойств полей?
12. В каком режиме осуществляется ввод данных в таблицу?
13. Для каких целей используется «маска ввода»?
14. Как влияют друг на друга свойства “формат” и “маска ввода”?
15. Как изменить таблицу?
16. Дайте определение информационно-логической модели.
17. Дайте определение информационного объекта.
18. Охарактеризуйте типы отношений информационных объектов.
19. Перечислите требования нормализации.
20. Назовите правила выделения информационных объектов.

Тема: Использование простой формы для загрузки базы данных

Вид занятия: лабораторное занятие №9

Цель занятия: Приобрести навыки создания простой формы для ввода данных

Цель работы

Освоение приемов работы с Microsoft Access, создание формы, работа с ней.

Подготовка к работе

Изучить литературу о СУБД Microsoft Access, приемах работы и создание форм и их назначении.

Контрольные вопросы

1. Создание формы.
2. Понятие простого, составного и внешнего ключей.
3. Охарактеризовать и привести примеры связей 1:1, 1:M, M:M.
4. Способы ввода и редактирования и обработки данных в формах БД.
5. Способы создания форм.
6. Типы форм.

Расширение базы данных "Приемная комиссия" Работа с формой

Все действия по созданию новых таблиц производим в той базе данных, которую мы создали на первой лабораторной работе.

1. Создать [таблицу 10.1](#) "АБИТУРИЕНТЫ" следующей структуры, ключевое поле РЕГ_НОМ:

Таблица 10.1. АБИТУРИЕНТЫ			
Имя поля	Тип поля	Размер поля	
РЕГ_НОМ	Счетчик	Длинное целое	
КОД_СПЕЦ	Текстовой	6	Выполнить как поле подстановки
МЕДАЛЬ	Логический	Да/Нет	
СТАЖ	Числовой	Одинарное с плавающей точкой	

2. Поле подстановки выполняется, как показано на [рис. 10.1](#)

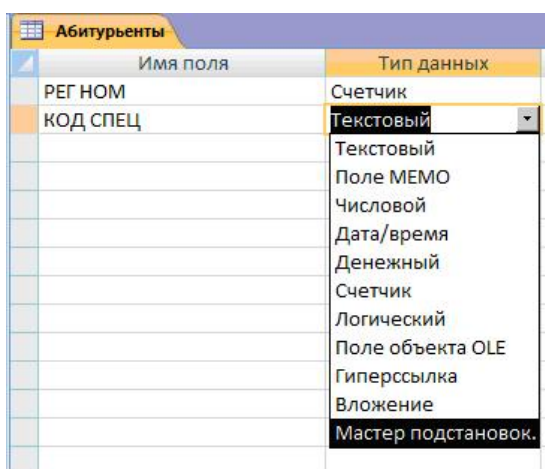


Рис. 10.1.

3. Создать [таблицу 10.2](#) "АНКЕТЫ" следующей структуры, ключевое поле РЕГ_НОМ:

Таблица 10.2. АНКЕТЫ

Имя поля	Тип поля	Размер поля
РЕГ_НОМ	Счетчик	Длинное целое
ФАМИЛИЯ	Текстовой	30
ИМЯ	Текстовой	20
ОТЧЕСТВО	Текстовой	20
ДАТА_РОЖДЕНИЯ	ДАТА	Краткий формат даты
ГОРОД	Текстовой	30
УЧ_ЗАВЕДЕНИЕ	Текстовой	50

4. Организовать связь таблиц **АБИТУРИЕНТЫ** с таблицей **СПЕЦИАЛЬНОСТИ** через поле **КОД_СПЕЦ**, а таблицы **АНКЕТЫ** и **АБИТУРИЕНТЫ** через поле **РЕГ_НОМ**. Обратите внимание, что создается два типа связей **КОД_СПЕЦ** с **КОД_СПЕЦ** связь один ко многим, а **РЕГ_НОМ** с **РЕГ_НОМ** связь один к одному.
5. Ввести записи в таблицу **АБИТУРИЕНТЫ**. Содержание таблицы показано на [рис. 10.2](#).

РЕГ_НОМ	КОД_СПЕЦ	МЕДАЛЬ	СТАЖ
1	030502	<input type="checkbox"/>	1
2	090106	<input checked="" type="checkbox"/>	
3	210302	<input type="checkbox"/>	0,5
4	210401	<input type="checkbox"/>	
5	210402	<input type="checkbox"/>	
6	210403	<input checked="" type="checkbox"/>	
7	210405	<input type="checkbox"/>	
8	210406	<input checked="" type="checkbox"/>	
9	220601	<input type="checkbox"/>	2
10	230105	<input type="checkbox"/>	
11	230201	<input type="checkbox"/>	1
12	210302	<input checked="" type="checkbox"/>	
13	210403	<input type="checkbox"/>	0,5
14	090106	<input checked="" type="checkbox"/>	
15	230105	<input type="checkbox"/>	
16	230201	<input checked="" type="checkbox"/>	
17	210406	<input type="checkbox"/>	
18	210302	<input checked="" type="checkbox"/>	
19	030502	<input type="checkbox"/>	
20	210405	<input checked="" type="checkbox"/>	
*	[No]	<input type="checkbox"/>	

Рис. 10.2.

Пояснения логическое значение **Истина** обозначена галочкой, **ЛОЖЬ** - пустым квадратом.

План-конспект занятия №13

Тема: *Создание меню различных типов*

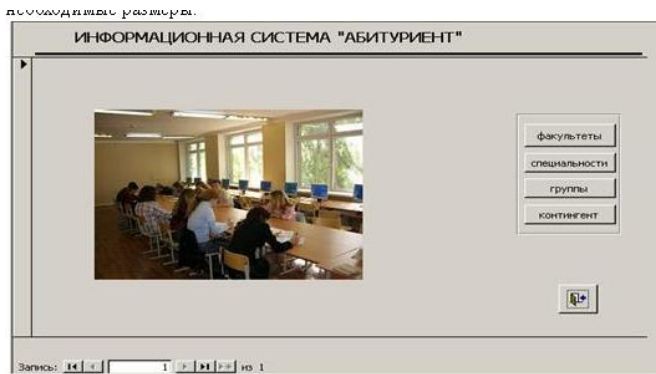
Вид занятия: *лабораторное занятие №11*

Цель занятия: Приобрести навыки создания меню

Кнопочная форма – это форма, открывающая другие формы, или отчеты. Создать кнопку форму можно двумя способами. Первый способ Вы уже должны были освоить, он основан на включении кнопок в форму в режиме конструктора. Второй способ

предусматривает использование специальной служебной программы в меню «Сервис» команда «Служебные программы», а затем «Диспетчер кнопочных форм». Вы можете воспользоваться любым способом. Примерный вид главной кнопочной формы представлен на следующем рисунке.

В качестве элемента оформления формы использована фотография. Для размещения фотографии или другого рисунка необходимо выбрать на панели элементов объект *рисунок* и установить его на планшете, определив необходимые размеры.



План-конспект занятия №14

Тема: Создание многостраничных форм

Вид занятия: лабораторное занятие №12

Цель занятия: Приобрести навыки создания многостраничных форм.

Лабораторная работа посвящена разработке пользовательских форм в базе данных «Абитуриенты».

Примечание: Во время выполнения предыдущей работы должны быть созданы:

1. База данных в папке «Базы данных» с уникальным именем, которое Вы присвоили ей при сохранении. База данных должна быть полностью заполнена данными, приведенными в описании к предыдущей работе.
2. Запросы к базе данных

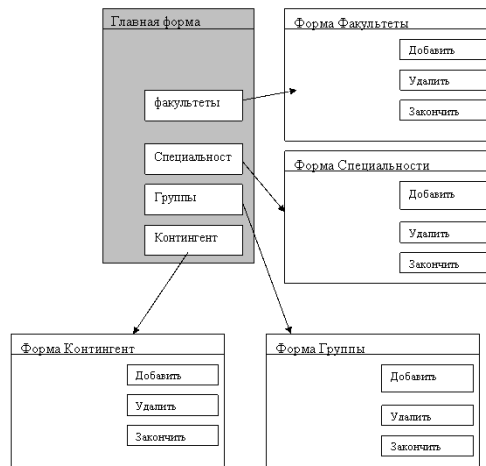
Формы являются основным способом взаимодействия пользователя с базой данных, являясь основой пользовательского интерфейса. В формах могут быть использованы одна или несколько таблиц. Существуют различные виды форм.

Для разработки форм существуют два способа:

- Создание форм в режиме конструктора
- Создание форм с помощью мастера

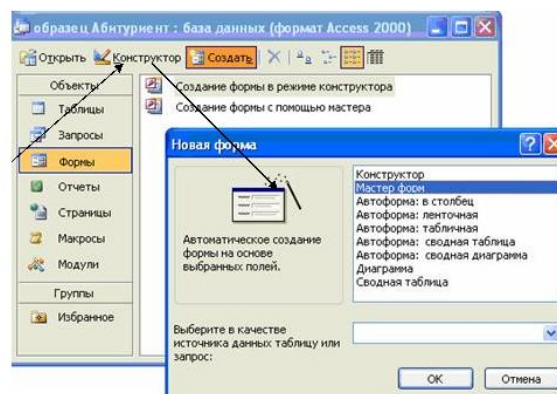
Разработку формы можно начать с использования конструктора, а затем доработать в конструкторе форм. Будем далее придерживаться такой технологии.

В течение занятия необходимо разработать приложение, состоящее из четырёх форм. Условно связи между формами можно представить в следующем виде:



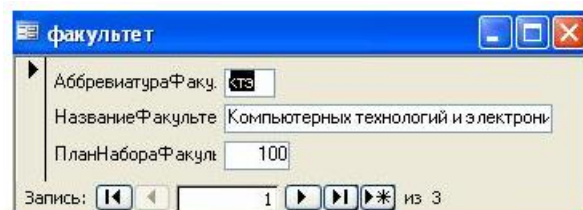
Разработка формы *факультеты*

Последовательность действий при разработке формы указана стрелками.

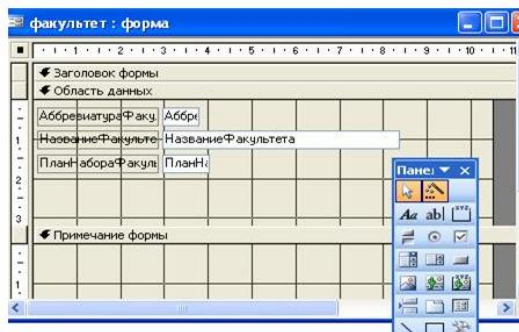


- Для создания формы в режиме мастера форм в окне свойств формы выбираем в качестве источника данных таблицу *факультеты*
- Из списка полей перетаскиваем в область данных формы все поля

Выполнив все оставшиеся шаги в мастере форм, получим готовую форму.



Откроем форму в конструкторе и усовершенствуем ее. Увеличим размер, увеличим размер надписей, разместим на форме кнопку закрытия формы, кнопку удаления записи и кнопку добавления записи. В конструкторе все элементы формы распределены на планшете, площадь которого можно изменить при помощи мышки. Дополнительные элементы, которые мы хотели бы расположить в форме, можно перетащить с панели элементов при помощи мышки.



Из панели элементов перетаскиваем в область данных пиктограмму кнопки

- При помощи мастера командных кнопок для каждой кнопки определить выполняемые операции. В данном примере это категория кнопки – работа с формой, действие – закрыть форму. Аналогичные действия для двух других копков.
- Для изменения свойств объектов формы, расположенных на планшете необходимо мышкой выделить элемент формы и нажать правую клавишу мышки. В контекстном меню выбираем свойства.
- Открывается таблица свойств выделенного объекта. В этой таблице можно задать размер шрифта, его насыщенность и другие характеристики.

Примечание: Кнопки: Список полей, Панель элементов, Свойства - расположены на панели инструментов конструктора форм.

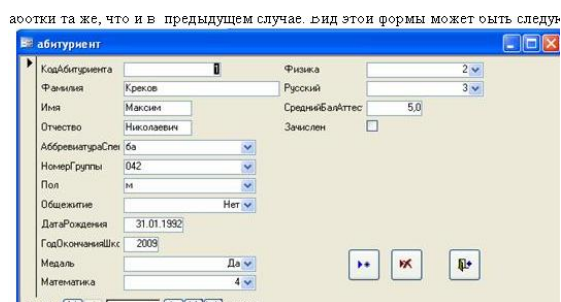


Пример доработанной в конструкторе формы приведен на рисунке. При разработке формы для копков можно вместо рисунков задать соответствующие надписи.

Аналогичным образом разработайте формы **Специальность** и **Группы**.

Разработка формы **Контингент**

Вначале разработаем форму для ввода данных нового абитуриента. Назовем эту форму **Абитуриент**. Методика разработки та же, что и в предыдущем случае. Вид этой формы может быть следующим.



Кроме полей ввода в форму добавлены три кнопки: добавление новой записи, удаление записи и закрытие формы.

Формы *Абитуриент*, *Факультеты* и ей подобные формы относятся к типу одиночных форм. Такие формы предназначены для просмотра по одной записи. Когда записей в таблице не много это вполне приемлемый вариант. Когда число записей достаточно большое и пользователю необходимо одновременно просматривать несколько записей, то для этого лучше использовать ленточные формы. Рассмотрим методику разработки такой формы на примере формы *список*.

Для этого воспользуемся, как и ранее мастером форм. Пункт меню Создать, тип формы Ленточная, источник данных таблица Контингент. Присвойте имя этой форме *список*.

Запустите форму на выполнение и познакомьтесь с результатом Вашей работы. Хотя в форме выводится таблица с записями, но многие поля можно было бы исключить из формы. Поэтому откроем форму в режиме конструктора и доработаем ее, удаляя лишние поля, расставляя по шире оставшиеся. Кроме того, желательно откорректировать заголовки, выделить их жирным шрифтом. В таблице свойств формы изменяем подпись на текст «список абитуриентов». В качестве примера Ваша форма может иметь следующий вид.

Код	Фамилия	Имя	Отчество	Спец.сть	Медаль	Математ.	Физика	Русский
1	Креков	Максим	Николаевич	ба	Да	4	2	3
2	Михаил	Михаил	Юрьевич	н	Нет	0	0	0
3	Волков	Евгений	Валерьевич	н	Нет	5	4	3
4	Киев	Евгений	Виссарионовна	ба	Нет	2	3	4
5	Николаева	Ирина	Юрьевна	н	Да	5	0	0
6	Рагобыльская	Валентина	Сергеевна	н	Нет	3	2	3
7	Щирая	Анастасия	Владимировна	ба	Да	5	4	4
8	Щирая	Анна	Владимировна	ба	Да	5	5	3
9	Козлова	Наталья	Владимировна	н	Да	0	0	0

Часто ленточные формы используются в качестве подчиненных форм, т.е. входят как отдельный элемент в другую форму. Рассмотрим такой вариант на примере формы контингент.

Будем разрабатывать форму *контингент* в режиме конструктора. Разместим на планшете в области данных ряд элементов формы. Для этого при помощи мышки увеличим площадь области данных. После этого выберем на панели элементов элемент *подчиненная форма*, перетащим его в область данных планшета. При этом для подчиненной формы на планшете мышкой очерчиваем область, где будет размещена подчиненная форма. После этого Вам предлагается выбрать из перечня форм ту форму, которую будете использовать в качестве подчиненной. Выбираем форму *список*.

Запустите форму *контингент* и оцените результат предыдущих операций. После этого откройте эту форму в конструкторе и откорректируйте при необходимости расположение и размер подчиненной формы. Кроме того, в форму *контингент* добавьте кнопку закрытия формы и кнопку открытия формы для ввода данных нового абитуриента, т.е. форму *абитуриент*. В качестве примерного вида результата разработки формы приведем следующий вариант.

Код	Фамилия	Имя	Отчество	Спец.сть	Медаль	Математ.	Физика	Русский
1	Креков	Максим	Николаевич	ба	Да	4	2	3
2	Михаил	Михаил	Юрьевич	н	Нет	0	0	0
3	Волков	Евгений	Валерьевич	н	Нет	5	4	3
4	Киев	Евгений	Виссарионовна	ба	Нет	2	3	4
5	Николаева	Ирина	Юрьевна	н	Да	5	0	0
6	Рагобыльская	Валентина	Сергеевна	н	Нет	3	2	3
7	Щирая	Анастасия	Владимировна	ба	Да	5	4	4

Кнопки: новый абитуриент, закрыть форму

Тема: Создание составных отчетов

Вид занятия: лабораторное занятие №15

Цель работы

Освоение приемов работы с Microsoft Access, создание отчетов.

Подготовка к работе

Изучить литературу о СУБД Microsoft Access, приемах работы и создание отчетов.

Контрольные вопросы

1. Способы создания отчетов.
2. Понятие отчета.
3. Исходные данные.
4. Ввод формул в отчет.
5. Назначение панели инструментов.
6. Основные приемы работы с отчетом.
7. Способы ввода редактирования и обработки данных в отчетах БД.


Ход работы:

Создание отчета

Сформируйте отчет для приемной комиссии о результатах вступительных экзаменов, используя **мастер отчетов**.

Прежде чем создавать сложный Отчет, стоит создать запрос к этому отчету, для того чтобы не возникло трудностей с созданием отчета. Предлагается попробовать два способа первый без создания предварительного запроса и второй способ создать предварительно запрос и по нему создать отчет.

Первый способ

- Перейти на вкладку **Отчеты**, выполнить команду **Создать**, выбрать  **Мастер отчетов**.
- Простой запрос
- Выбрать поля, с указанных таблиц и запросов в следующем соответствии *Факультет, Экзамен_1, Экзамен_2, Экзамен_3*(из таблицы *ФАКУЛЬТЕТЫ*), *Сумма баллов* (из запроса "Сумма баллов", *Фамилия* (из таблицы *АНКЕТЫ*), *РЕГ_НОМ*, *КОД_СПЕЦ*, *Медаль*, *Стаж_раб.* (из таблицы *АБИТУРЬЕНТЫ*), щелкнуть на кнопке **Далее**.
- Задать один уровень группировки по полю **Факультет** ([рис. 12.1](#)), щелкнуть на кнопке **Далее**.

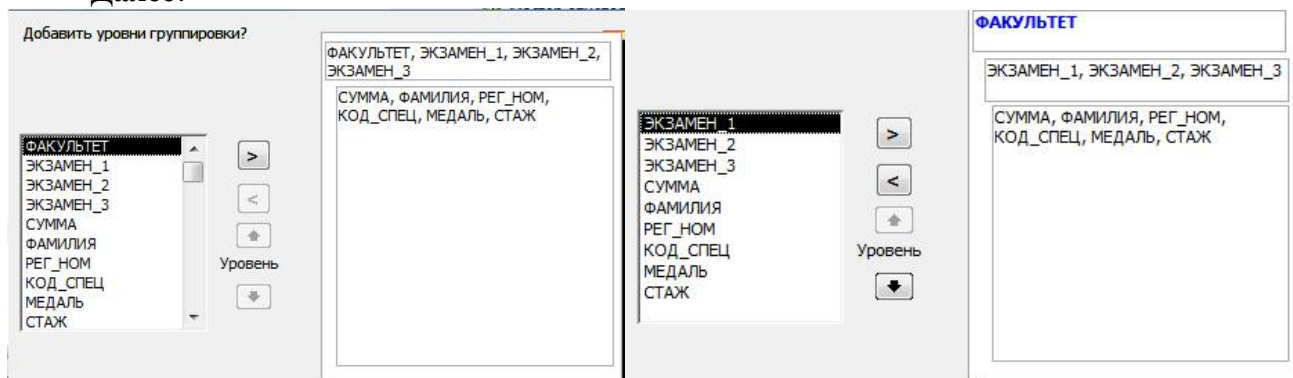


Рис. 12.1.

- Указать порядок сортировки 1). Сумма по убыванию; 2). Фамилия по возрастанию ([рис. 12.2](#)), щелкнуть кнопку **Далее**.

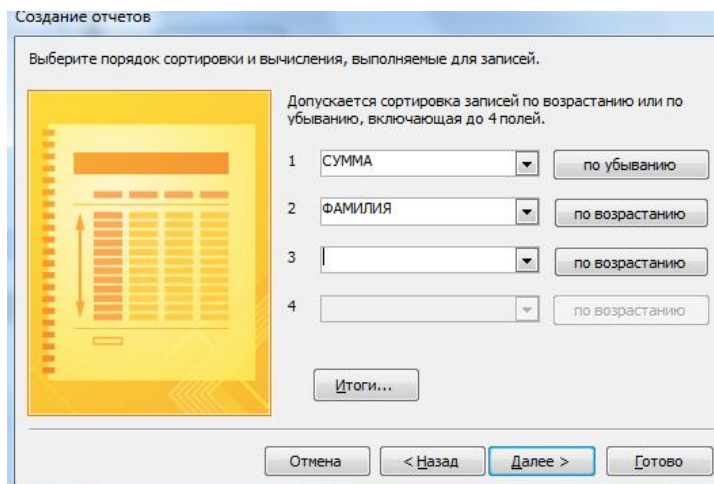


Рис. 12.2.

- Выбрать вид макета **Ступенчатый**, ориентация **Альбомная**, Настроить ширину полей для размещения на одной странице ([рис. 12.3](#)) Далее.

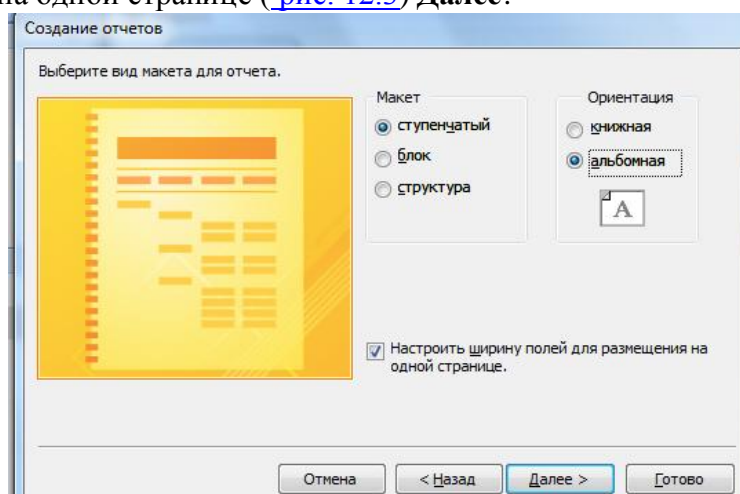


Рис. 12.3.

- Выбрать стиль **Стандартный**, Далее.
- Указать имя отчета "**К зачислению**", щелкнуть на кнопке **Готово** ([рис. 12.4](#)).

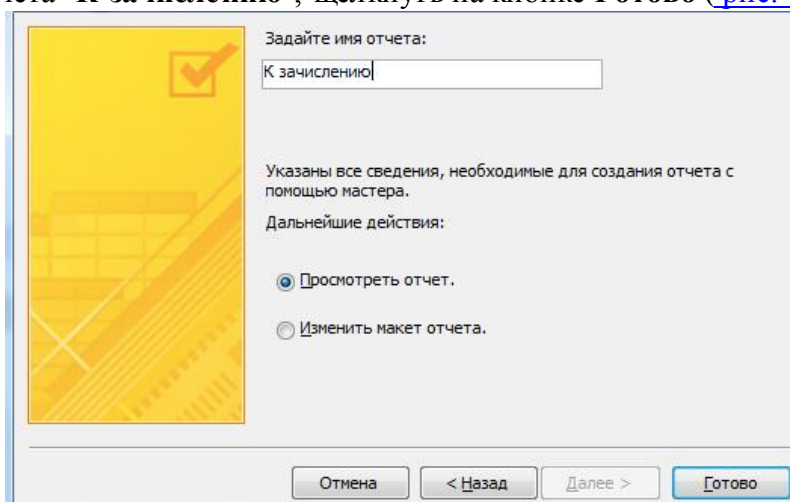


Рис. 12.4.


- Результат отчета представлен на ([рис. 12.5](#)).

ФАКУЛЬТЕТ	ЭКЗАМЕН_1	ЭКЗАМЕН_2	ЭКЗАМЕН_3	СУММА	ФАМИЛИЯ	РЕГ_НОМ	КОД_СПЕЦ	МЕДАЛ	СТАЖ
Безового Тел									
	математика	физика	русский язык						
				15	Алексеев	14	210405	<input checked="" type="checkbox"/>	0
				15	Григорьев	7	210406	<input checked="" type="checkbox"/>	1,5
				11	Новалев	11	210406	<input type="checkbox"/>	0
				9	Павлов	19	210302	<input type="checkbox"/>	0
Зачного обр									
	математика	физика	русский						
				15	Лысенков	17	030602	<input checked="" type="checkbox"/>	0
				13	Ильин	10	030602	<input type="checkbox"/>	0
				12	Маркеев	16	210403	<input type="checkbox"/>	1,5
				10	Аношкин	9	090106	<input type="checkbox"/>	1
				10	Кузнецов	20	090106	<input type="checkbox"/>	1,6

Рис. 12.5.

Обратите внимание на то, что данные в графу "Сумма баллов" выбирается из запроса "Сумма баллов". Других полей из этого запроса извлекать не следует. В режиме предварительного просмотра отчет появится на экране, чтобы отредактировать внешний вид следует войти в **Конструктор**.

Ввод формулы для вычисления общего количества поступивших абитуриентов, в отчет производится в конструкторе следующим образом (рис. 12.6):

- на **Панели элементов** нажать на кнопку **Поле** ;
- выбрать с помощью мыши в области **Примечание** отчета место, в которое следует поместить поле;
- поместить курсор в поле и ввести выражение, вычисляющее количество поступивших абитуриентов $=\text{Count}(\text{РЕГ_НОМ})$.

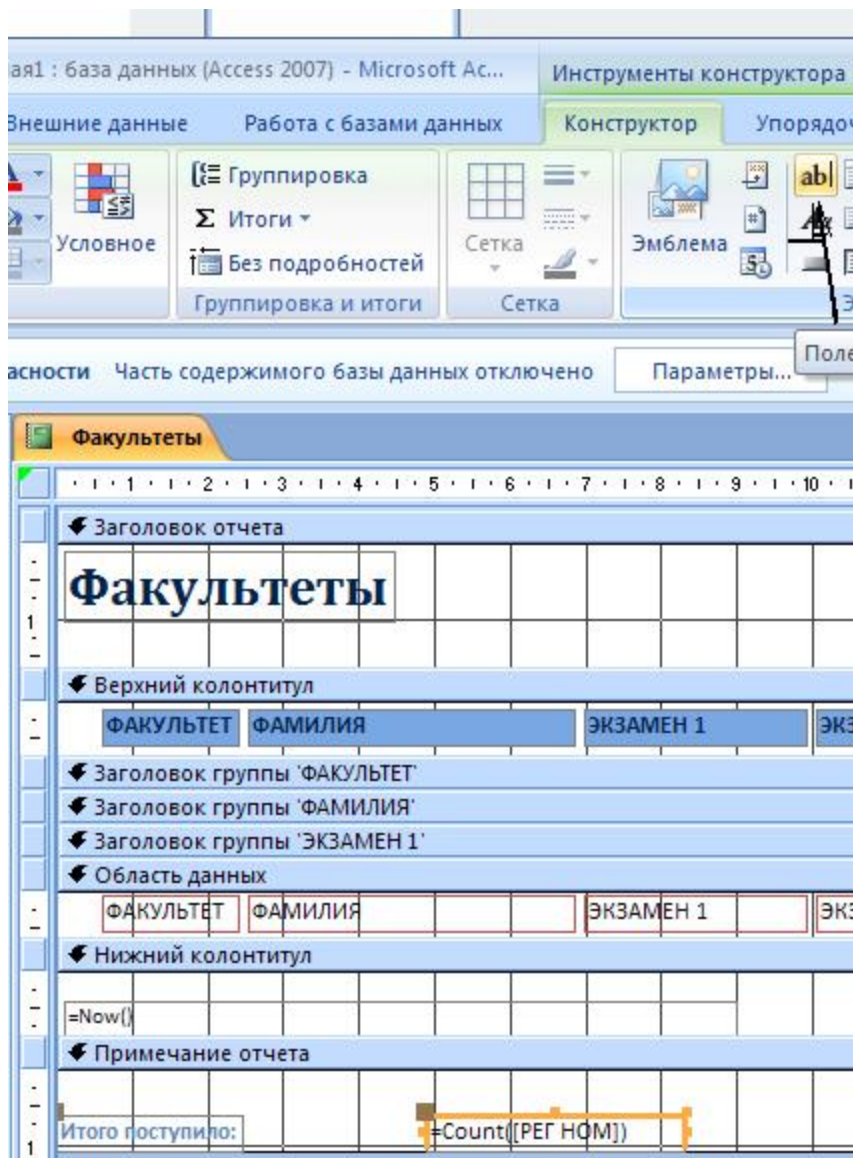


Рис. 12.6.

Самостоятельно:

1. Выполните отчет вторым способом, сравните результаты, сделайте выводы.
2. Определите критерий, по которому будет проходить прием студентов. Например, набранные баллы не ниже 12, в виде исключения принимаются студенты, имеющие стаж работы, даже если балл ниже.
3. Занести в таблицу **ИТОГИ** сведения о поступлении в университет. В поле **ЗАЧИСЛЕНИЕ** выставить флажки в записях для принятых в университет абитуриентов, с помощью запроса.
4. На основе запроса о принятых студентах самостоятельно получите отчет со списком абитуриентов, принятых в университет в котором должны отражаться следующие поля Фамилия, Имя, Отчество, количество баллов, название факультета и специальности. Причем Фамилия Имя и отчество сведите в одно поле, применив вычисляемое поле в запросе следующим образом ([рис. 12.7](#))

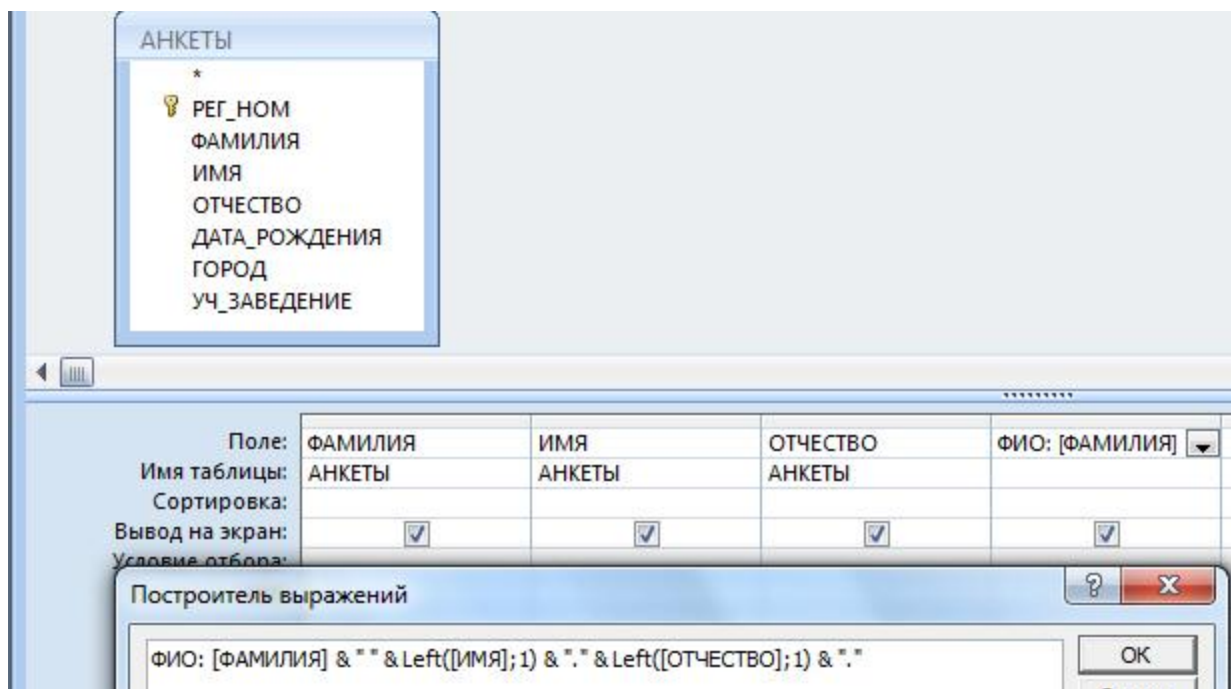


Рис. 12.7.

ФИО: [ФАМИЛИЯ] & " " & Left([ИМЯ];1) & "." & Left([ОТЧЕСТВО];1) & "."

Примечание. Это формула означает, что фамилия берется полностью, а с имени и отчества первая буква слева.

Результат применения такого запроса показан на [рис. 12.8.](#)

ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ФИО
Васильева	Ольга	Николаевна	Васильева О.Н.
Быков	Алексей	Ильич	Быков А.И.
Круглов	Борис	Макарович	Круглов Б.М.
Листьев	Сергей	Иванович	Листьев С.И.
Елькин	Виктор	Алексеевич	Елькин В.А.
Мухин	Олег	Иванович	Мухин О.И.
Григорьева	Анастасия	Алексеевна	Григорьева А.А.
Зубова	Ольга	Николаевна	Зубова О.Н.
Анохина	Анна	Петровна	Анохина А.П.
Ильин	Петр	Викторович	Ильин П.В.
Ковалева	Инна	Сергеевна	Ковалева И.С.
Мрясова	Тамара	Юрьевна	Мрясова Т.Ю.
Серов	Олег	Петрович	Серов О.П.
Алексеева	Александра	Петровна	Алексеева А.П.
Галкин	Петр	Сергеевич	Галкин П.С.
Моресьев	Семен	Игнатьевич	Моресьев С.И.
Лысенков	Анатолий	Иванович	Лысенков А.И.
Алимов	Александр	Сергеевич	Алимов А.С.
Павлова	Валентина	Николаевна	Павлова В.Н.
Кузнецов	Иван	Петрович	Кузнецов И.П.

Рис. 12.8.

- Посчитайте, сколько абитуриентов поступило на каждый факультет.
- Постройте круговую диаграмму по результатам поступления в ВУЗ укажите % поступивших абитуриентов, на каждый факультет.

Тема: Создание и печать отчетов

Вид занятия: лабораторное занятие №16

Цель занятия: Приобрести навыки создания и печати отчетов по образцу

Задание:

Отчёт должен содержать фамилию и инициалы всех студентов одной группы, а также оценки по трём экзаменам. Вид отчёта представлен на рисунке.

ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ					
Группа – 02					
Дата экзамена 27.07.2010					
№	Фамилия И.О.	математика	физика	русский	Роспись преподавателя
1	Архипов М.В.	4	3	2	
2	Владимиров С.П.	5	4	3	
3	Гришина Е.А.	4	5	3	
4	Дунаева О.Л.	3	2	4	
5	Ершов П.Д.	2	4	4	
6	Зымина К.Е.	4	3	5	
Средний бал		3.7	3.2	3.5	

Председатель приёмной комиссии В.И.Григорьев

В качестве первого этапа разработки отчета разработаем запрос к базе данных, содержащий следующие данные необходимые для отчета:

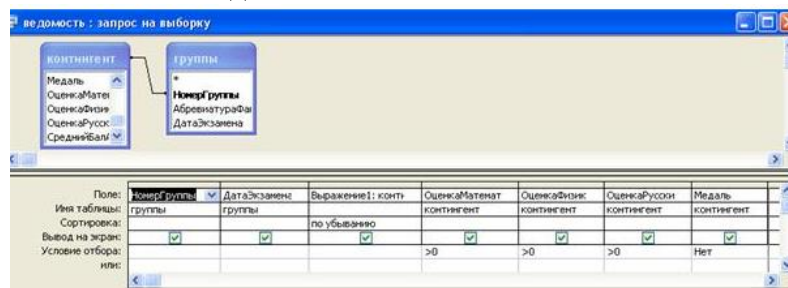
- Номер группы;
- Дату экзамена;
- Фамилию и инициалы абитуриента;
- Оценку по математике;
- Оценку по физике;
- Оценку по русскому языку.

Примечание: В запросе для формирования и объединения инициалов абитуриента сформируйте выражение

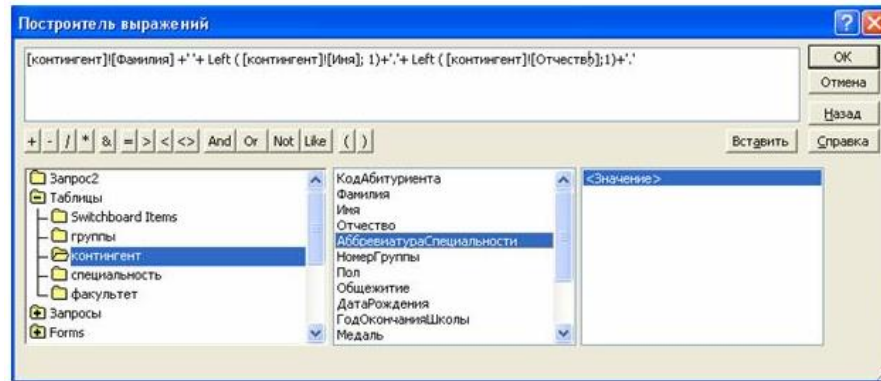
$[Контингент].[Фамилия]+' '+Left([Контингент].[Имя];1)+' '+Left([Контингент].[Отчество];1)+' '$

Функция *Left(поле, n)* вырезает *n* первых символов поля.

Назовем запрос **Ведомость**. Макет запроса представлен на рисунке. В запрос включены только абитуриенты сдававшие все три экзамена, т.е. оценки должны быть больше нуля. Кроме того, из запроса исключены все медалисты.



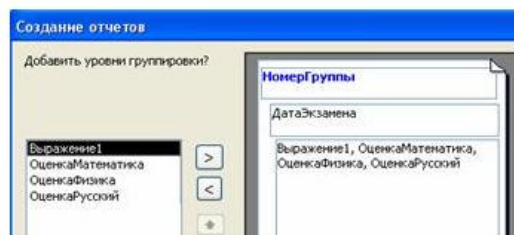
Для формирования в запросе выражения рекомендуется использовать построитель выражений.



Результат запроса будет иметь следующий вид.

НомерГруппы	ДатаЭкзамена	Выражение1	Математика	Физика	Русский	Медаль
022	30.06.2010	Аветисян А.С.	2	2	2	Нет
005	01.07.2010	Азаренков С.А.	5	5	3	Нет
023	30.06.2010	Айвазян Р.А.	4	4	4	Нет
004	01.07.2010	Акимов А.А.	3	4	5	Нет
063	17.07.2010	Акимова Ю.В.	3	5	5	Нет
005	01.07.2010	Аксенов И.О.	4	5	3	Нет
023	30.06.2010	Александренко	5	4	3	Нет

Приступим к разработке отчета. Воспользуемся мастером создания отчетов. В качестве уровней группировки выберем номер группы и дату экзамена.



Полученный отчет будет иметь вид

группы

НомерГруппы	Экзамена	Выражение1	Математик	Физика	Русский
002	01.07.2010	Андреев В.А.	4	5	3
		Байдакова Ю.С.	3	3	3
		Вилонов А.С.	5	4	4
		Гайдж Р.А.	2	3	4
		Галенин П.И.	4	4	4
		Ефременов И.	4	4	3
		Клиня Ю.М.	3	5	3

Доработаем отчет в конструкторе до требуемого вида. Для этого воспользуемся возможностями по переносу отдельных элементов отчета в бланке отчета, а также добавим необходимые разделительные линии.

Экзаменационная ведомость			
Номер группы -	НомерГруппы		
Дата экзамена -	ДатаЭкз		
Фамилия И.О.	Математика	Физика	Русский
Выражение1	ОценкаМа	ОценкаФи	ОценкаРус
Средний бал	=Avg	=Avg/C	=Avg/C

Тема: *Использование запросов для отбора и сортировки записей*

Вид занятия: лабораторное занятие №17

Цель работы

Освоение приемов работы с Microsoft Access, создание простых и сложных запросов.

Подготовка к работе

Изучить литературу о СУБД Microsoft Access, приемах работы и создание простых и сложных запросов.

Контрольные вопросы

1. Создание запросов.
2. Простые запросы.
3. Сложные запросы.
4. Применение операторов "or", "and", "between".
5. Запрос на удаление.
6. Использование групповых операций.
7. Использование вычисляемых полей.

Ход работы

Реализация простых и сложных запросов к базе данных "Приемная комиссия"

1. Построить и выполнить запрос к базе данных "Приемная комиссия": получить список всех экзаменов на всех факультетах. Список отсортировать в алфавитном порядке названий факультетов. Для выполнения достаточно одной таблицы **ФАКУЛЬТЕТЫ**.
 - открыть вкладку **Создание**, в открывшемся панели выбрать **Конструктор запросов**;
 - в поле схемы запроса поместить таблицу **ФАКУЛЬТЕТЫ**. Для этого в окне **Добавление таблицы**, вкладке **Таблицы** выбрать название таблицы **ФАКУЛЬТЕТЫ**, щелкнуть на кнопках **Добавить** и **Заккрыть**. Запрос сохранить под именем "Список экзаменов";
 - заполнить бланк запроса с помощью контекстного меню в верхней половине бланка открываются те таблицы, к которым обращён запрос. В этих таблицах дважды щёлкают на названиях тех полей, которые должны войти в результирующую таблицу. При этом автоматически заполняются столбцы в нижней части бланка. Сформировав структуру запроса, его закрывают;
 - для сортировки данных в запросе следует щелкнуть на строке **Сортировка**. Появляется кнопка раскрывающегося списка, в котором можно выбрать метод сортировки по возрастанию или по убыванию;
 - возможна многоуровневая сортировка (сразу по нескольким полям), но в строгой очерёдности слева на право. Поля надо располагать с учётом будущей сортировки, при необходимости перетаскивая их мышью на соответствующие места;
 - управление отображением данных осуществляется установкой (или сбросом) флажка **Вывод на экран** (рис. 11.1).

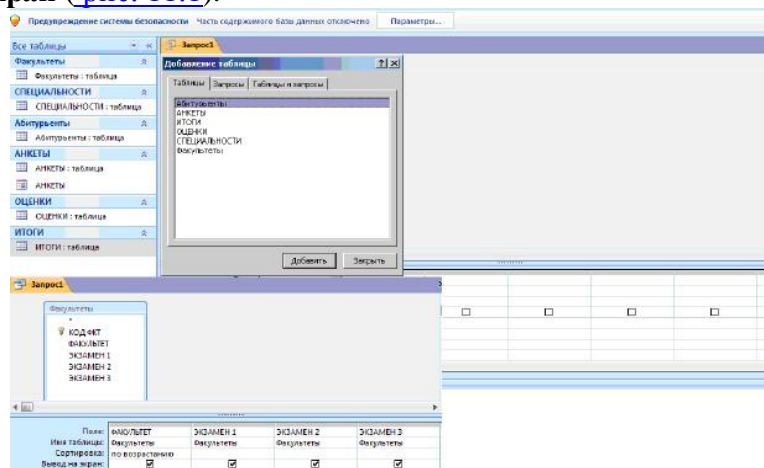


Рис. 11.1.

- Сменить заголовки граф запроса. Заголовками граф таблицы являются имена полей. Имеется возможность замены их на любые другие надписи, при этом имена полей в БД не изменятся. Делается это через параметры **Свойства поля**, для полей соответствующей таблицы. Для этого нужно открыть конструктор и в списке свойств каждого поля добавить в строке **Подпись** соответствующий текст. Замените поле **ФАКУЛЬТЕТ** на **ФАКУЛЬТЕТЫ**, **ЭКЗАМЕН_1** на **1-й экзамен** и т.д. ([рис. 11.2](#)).

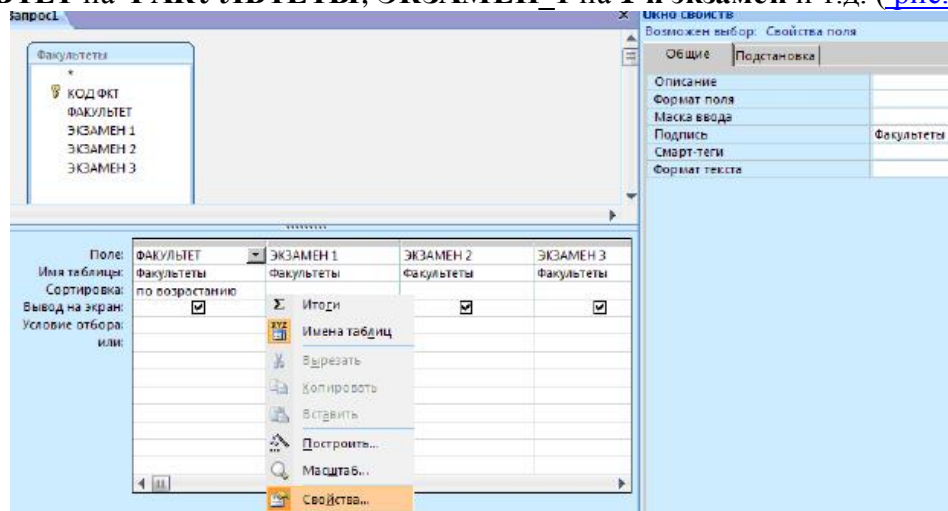


Рис. 11.2.

После этого вернуться к запросу "Список экзаменов", выполнив команду **Открыть** получим таблицу с результатом запроса, которая отличается от предыдущей лишь заголовками. ([рис. 11.3](#)). Обратите внимание, что заголовки меняются только в просмотрном режиме в конструкторе они остаются прежними.

Факультеты	1-й Экзамен	2-й Экзамен	3-й экзамен
Базового Телекоммуникационного образования	математика	физика	русский язык
Заочное	математика	физика	русский
Информационных систем и технологий	математика	английский	физика
Телекоммуникации и радиотехники	математика	физика	информатика

Рис. 11.3.

- Выведите список всех специальностей с указанием факультета и плана приема. Отсортировать список в алфавитном порядке по двум ключам: названию факультета (первый ключ) и названию специальности (второй ключ). Напомним, что сортировка сначала происходит по первому ключу и, в случае совпадения у нескольких записей его значения, они упорядочиваются по второму.
 - Построить запрос в конструкторе запросов в виде, показанном на рисунке ([рис. 11.4](#)).

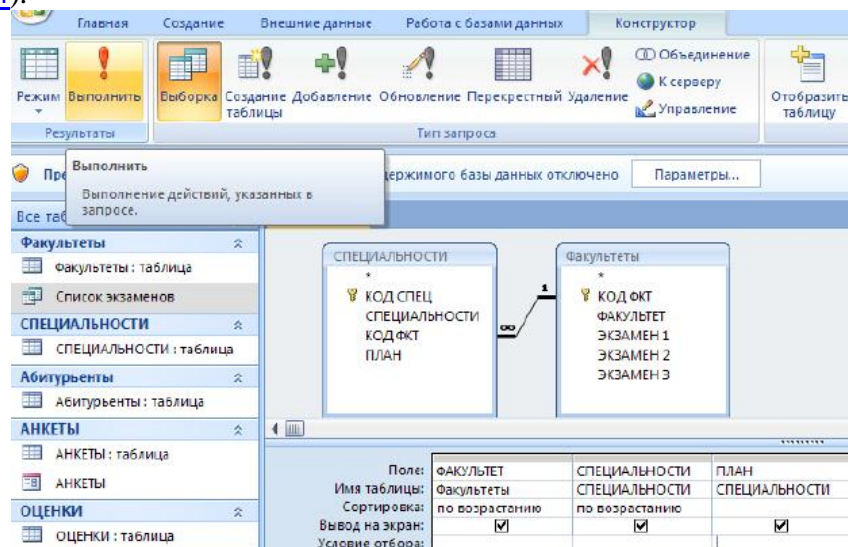


Рис. 11.4.

- Исполнить запрос. В результате должна получиться следующая таблица ([рис. 11.5](#)).

ФАКУЛЬТЕТ	СПЕЦИАЛЬНОСТИ	ПЛАН
Базового Телекоммуникационного образования	Радиосвязь, радиовещание и телевидени	25
Базового Телекоммуникационного образования	Радиотехника	25
Базового Телекоммуникационного образования	Сети связи и системы коммутации	50
Заочное	Защищенные системы связи	10
Заочное	Связь с общественностью	10
Заочное	Экономика в телекоммуникации	10
Информационных систем и технологий	Информационные системы и технологии	15
Информационных систем и технологий	Программное обеспечение	25
Информационных систем и технологий	Управление инновациями	15
Телекоммуникации и радиотехники	Средства связи с подвижными объектами	10
Телекоммуникации и радиотехники	Физика и техника оптической связи	10

Рис. 11.5.

- Получить список всех абитуриентов, живущих в Самаре и имеющих медали. В списке указать фамилию, номер школы и факультет на который они поступают. Отсортировать список в алфавитном порядке фамилий.

- Для реализации данного запроса информация берется из трех таблиц **АНКЕТЫ**, **ФАКУЛЬТЕТЫ**, **АБИТУРИЕНТЫ**.

В конструкторе запросов это будет выглядеть так (см. [рис. 11.6](#))

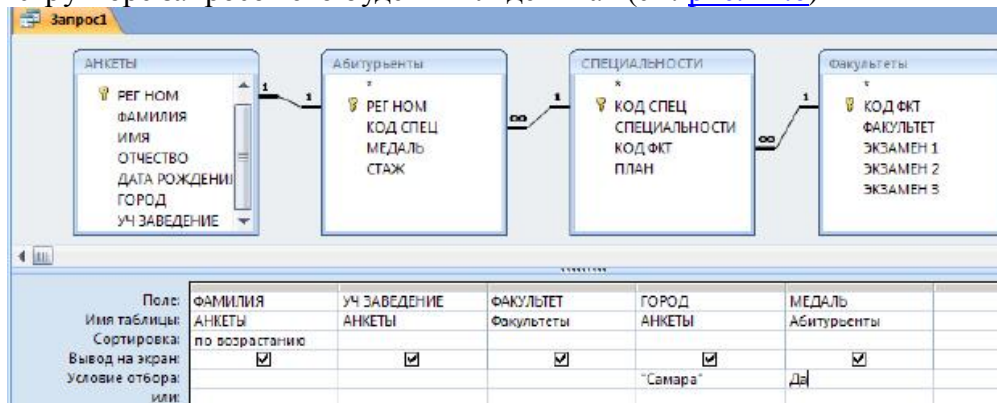


Рис. 11.6.

Обратите внимание на то, что, в запросе используются поля только из трех таблиц **АНКЕТЫ**, **ФАКУЛЬТЕТЫ** и **АБИТУРИЕНТЫ**, в реализации запроса участвует таблица **СПЕЦИАЛЬНОСТИ**, т.к. таблица **АБИТУРИЕНТЫ** связана с таблицей **ФАКУЛЬТЕТЫ** через таблицу **СПЕЦИАЛЬНОСТИ**.

Результатом запроса должна быть следующая таблица ([рис. 11.7](#)):

ФАМИЛИЯ	УЧ ЗАВЕДЕНИЕ	ФАКУЛЬТЕТ	ГОРОД	МЕДАЛЬ
Алексеева	ПТУ №25	Заочное	Самара	<input checked="" type="checkbox"/>
Алимов	Школа № 45	Базового Телекоммуникационного образования	Самара	<input checked="" type="checkbox"/>
Кузнецов	Школа № 48	Базового Телекоммуникационного образования	Самара	<input checked="" type="checkbox"/>
Мрясова	Школа № 93	Базового Телекоммуникационного образования	Самара	<input checked="" type="checkbox"/>
Мухин	СОМЛИ	Заочное	Самара	<input checked="" type="checkbox"/>

Рис. 11.7.

Самостоятельно:

- Получить список всех абитуриентов, поступающих в ВУЗ имеющих производственный стаж. Указать фамилию, город, специальность, стаж и факультет на который поступают. Отсортировать фамилии по возрастанию.
- Получить список абитуриентов, поступающих в ВУЗ имеющих производственный стаж и медаль. Указать фамилию, специальность и факультет на который поступают. Отсортировать фамилии по возрастанию.

Тема: Создание вычисляемых полей в запросах

Вид занятия: лабораторное занятие №18

Цель занятия: Приобрести навыки разработки сложных вычисляемых запросов

Ход работы:

1. Удалите из таблицы **ОЦЕНКИ** сведения об абитуриентах, получивших двойки или не явившихся на экзамены. Для этой цели будет использоваться второй вид запроса: запрос на удаление. Алгоритм выполнения запроса.

- перейти на вкладку **Создать**, далее **Конструктор запросов**;
- Добавить таблицу **ОЦЕНКИ**;

- установить тип запроса **Удаление** с кнопки  (рис. 11.8);

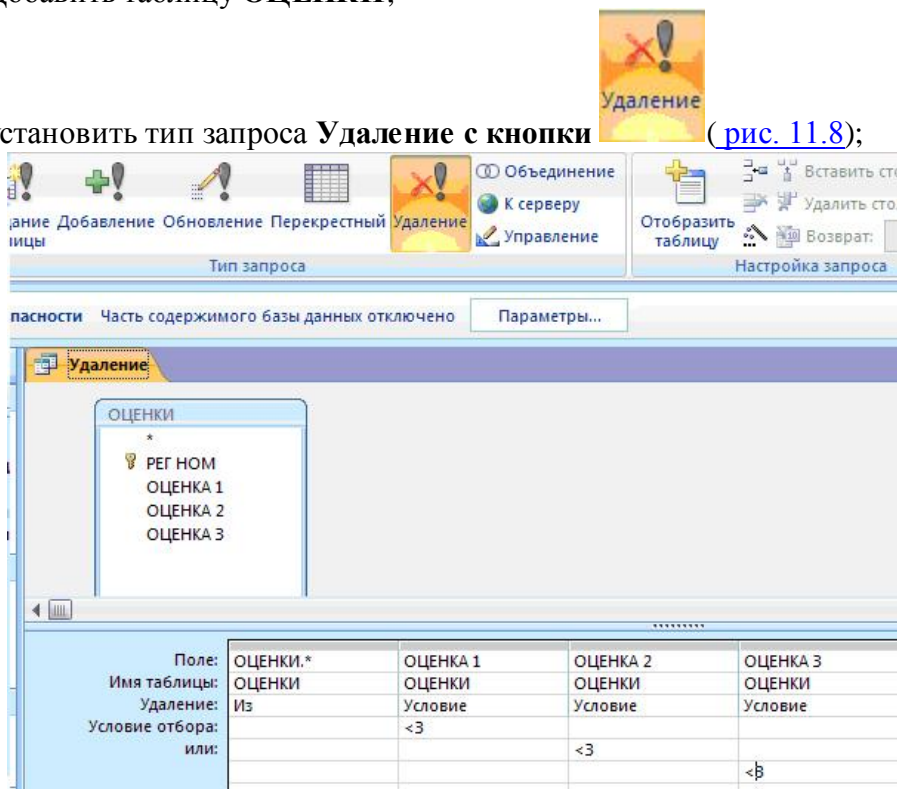


Рис. 11.8.

2. Получить список всех абитуриентов, сдавших физику с оценкой хорошо и отлично.

- В данном запросе следует применить оператор **or**. Конструктор запроса показан на (рис. 11.9).

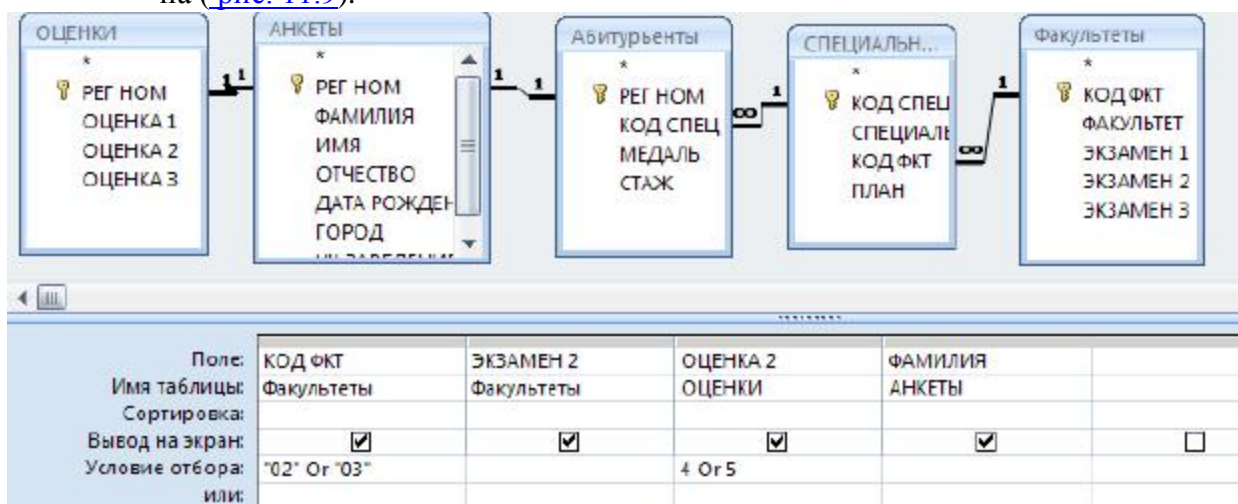


Рис. 11.9.

Как вы могли заметить в поле **КОД_ФКТ**, тоже стоит условие, т.к. физику сдавали на трех факультетах, и добавлены две лишние таблицы, чтобы не нарушать связь. Результат представлен на [рис. 11.10](#).

КОД ФКТ	ЭКЗАМЕН 2	ОЦЕНКА 2	ФАМИЛИЯ	РЕГ НОМ
02	физика	4	Круглов	3
02	физика	4	Зубова	8
02	физика	4	Алимов	18
02	физика	4	Кузнецов	20

Рис. 11.10.

3. Выведите таблицу со значениями суммы баллов, включив в неё регистрационный номер, фамилию и сумму баллов. Отсортировать по убыванию суммы:
 - В данном запросе используется вычисляемое поле СУММА;
 - Данный запрос в конструкторе будет выглядеть следующим образом ([рис. 11.11](#)).

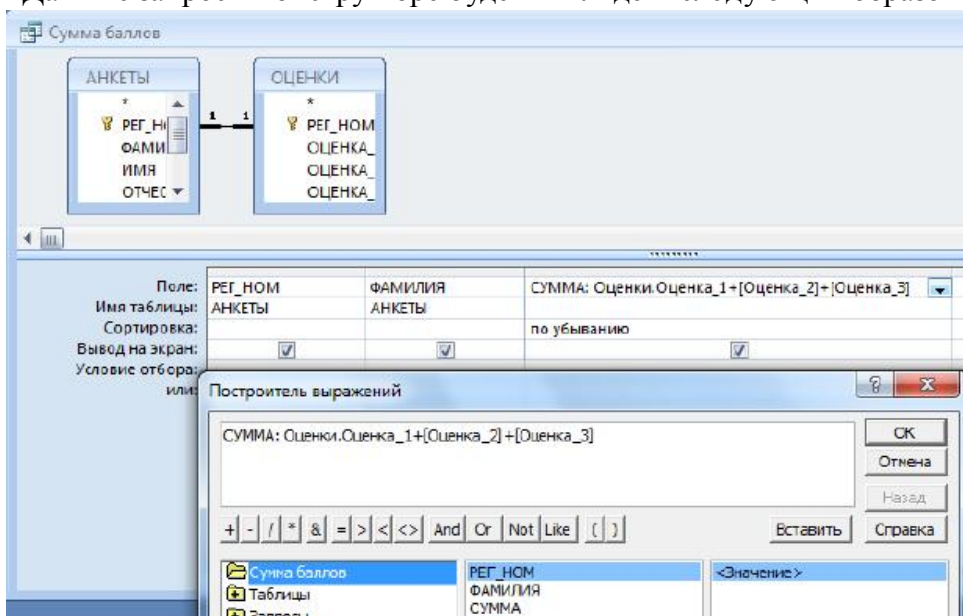



Рис. 11.11.

Выражение можно вводить, как непосредственно в ячейке конструктора, так и воспользовавшись построителем выражений .

- Квадратные скобки обозначают значения соответствующего поля.

Примечание. Вычисляемое поле представляется в следующем формате: <имя поля> <выражение>.

В результате выполненного запроса таблица будет выглядеть следующим образом ([рис. 11.12](#)).

РЕГ_НОМ	ФАМИЛИЯ	СУММА
17	Лысенков	15
15	Галкин	15
14	Алексеева	15
12	Мрясова	15
6	Мухин	15
7	Григорьева	15
1	Васильева	15
3	Круглов	14
5	Елькин	14
10	Ильин	13
2	Быков	13
8	Зубова	13
4	Листьев	12
16	Моресьев	12
11	Ковалева	11
20	Кузнецов	10
9	Анохина	10
19	Павлова	9
*	(№)	

Рис. 11.12.

4. При обработке данных бывает необходимо учитывать в запросах итоги, которые вычисляются определенным образом для различных групп записей в таблице. Для итоговых вычислений необходимо создать новый запрос в режиме конструктора, добавить в него необходимые таблицы, а затем воспользоваться кнопкой **Групповые операции** Σ . При этом в строке **Групповые операции** бланка запроса по умолчанию устанавливается значение **Группировка**. Из списка **Групповые операции** выбираются необходимые и запрос закрывается.
- При выполнении групповых операций можно использовать итоговые функции, которые следует выбирать из списка в добавленном поле **Групповые операции**. Основные итоговые функции:
- **Sum** - суммирование числа значений в группе (в столбце),
 - **Avg** - среднее значение для группы,
 - **Min** - минимальное значение для группы,
 - **Max** - максимальное значение для группы,
 - **Count** - подсчет числа значений для группы,
 - **First** - значение поля в первой записи группы,
 - **Last** - значение поля в последней записи группы.
5. Найдите Количество абитуриентов набравших 14 баллов. Для этого необходимо применить групповые операции, и в зависимости от условий для каждого поля, следует выбрать из списка необходимую функцию ([рис. 11.13](#)).

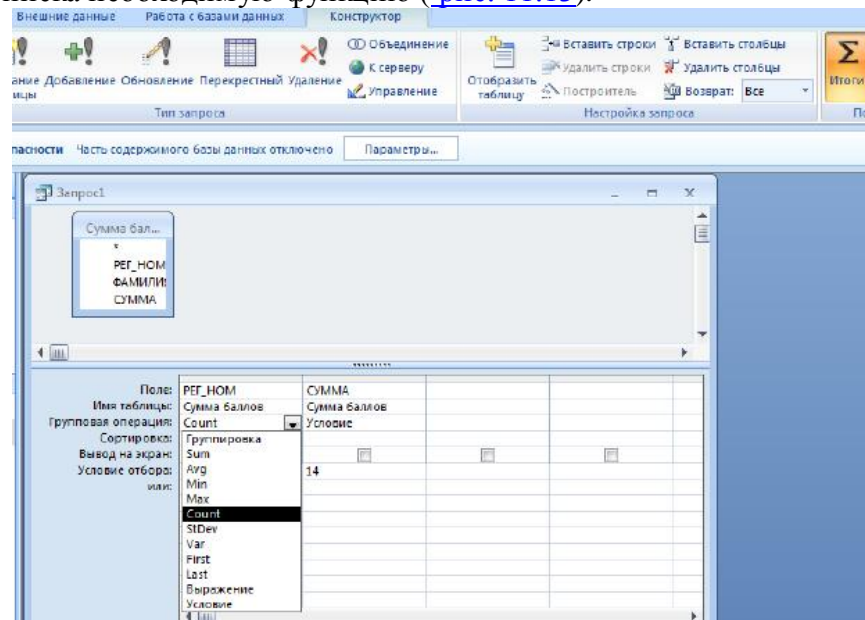


Рис. 11.13.

Самостоятельно:

1. Получите список студентов сдавших математику с оценкой хорошо и отлично по факультетам 01 и 03.
2. Сделайте запрос таким образом, чтобы остались абитуриенты, набравшие 12 баллов и более, с полем зачисление. Обратите внимание, что таблица Итоги заполнится автоматически.
3. Найдите среднюю сумму баллов.
4. Найдите фамилию студента получившего min балл при поступлении.
5. Найдите количество студентов сдавших русский язык на 5.

Тема: Использование выражений в запросах с помощью Построителя

Вид занятия: лабораторное занятие №19

Цель занятия: Приобрести навыки работы с Построителем выражений для создания сложных запросов

Ход работы:

Для БД, созданной на прошлом занятии, вычислить возраст студентов
Это можно сделать по формуле в Построителе выражений

Возраст: Year(Date()) - Year(Дата рождения)

Тема: Редактирование данных с помощью запросов

Вид занятия: лабораторное занятие №20

Цель занятия: Приобрести навыки редактирования данных с помощью запросов.

Обновление данных в одной таблице данными из другой таблицы

При обновлении данных в одной таблице данными из другой необходимо соблюдать основное правило: типы данных исходного и целевого полей должны совпадать или быть совместимыми.

Кроме того, при обновлении данных в одной таблице данными из другой и использовании совместимых типов данных вместо совпадающих приложение Access преобразует типы данных этих полей в целевой таблице. В результате некоторые данные в целевых полях могут быть обрезаны (удалены). В разделе [Ограничения преобразования типов данных](#) указаны случаи, когда можно и нельзя выполнять преобразование типов данных. В таблице, приведенной в этом разделе, также рассмотрены случаи, когда преобразование типа данных может изменить или удалить некоторые или все данные в поле, и указано, какие данные могут быть удалены.

Процесс обновления данных в одной таблице данными из другой включает основные этапы, указанные ниже.

Ход работы:

1. Создание запроса на обновление и добавление в него исходной и целевой таблиц.
2. Объединение этих таблиц по полям, содержащим связанные сведения.
3. Добавление имен целевых полей в строку **Поле** бланка запроса.
4. Добавление имен исходных полей в строку **Обновление** бланка запроса с использованием следующего синтаксиса: *[исходная таблица].[исходное поле]*.

Действия, описанные в данном разделе, предполагают, что используются две похожих таблицы. В этом примере таблица «Клиенты» располагается в унаследованной базе данных и содержит более свежие данные, чем таблица «Заказчики». Несколько имен менеджеров и адресов были изменены. Поэтому было принято решение обновить таблицу «Заказчики» данными из таблицы «Клиенты».

Таблица «Покупатели»

Код покупателя	Имя	Адрес	Город	Область	Почтовый индекс	Страна	Телефон	Контакт
1	Музей научных достижений	Загородное шоссе, д. 150	Москва	Московская обл.	987654	РФ	(551) 999-01-01	Верный Григорий
2	Blue Yonder Airlines	Ул. Гарибальди, д. 170	Пермь	Пермская обл.	995522	РФ	(5501) 99-01-01	Корепин Вадим
3	Винный завод	Ул. Кедрова, Красная	Красноярски	Красноярская обл.	876444	РФ	(5502)	Ожогова Инна

		д. 54	рск	й край			99-02-01	
4	Фармацевтическая фабрика	Ул. Губкина, д. 233	Тверь	Тверская обл.	897884	РФ	(5503) 99-01-03	Новиков Николай
5	Кофейная фабрика	Пл. Хо Ши Мина, д. 15, кв. 5	Москва		876555	РФ	(5504) 99-01-04	Шашков Руслан
6	Почтовая служба	Лихов пер., 991	Красноярск	Красноярский край	876444	РФ	(5502) 99-02-01	Горноженко Дмитрий
7	НИИ железа	Ул. Тенистая, д. 698	Москва	Московская обл.	344554	РФ	(5505) 99-03-01	Грачев Николай
8	Корпорация «Софт»	Ул. Заречная, 503	Москва	ИЛИ	134158	РФ	(5501) 99-01-01	Ильина Юлия
9	Лесопитомник	Ул. Заречная, 504	Москва	ИЛИ	134158	РФ	(5501) 99-01-01	Сергиенко Мария

Таблица «Клиенты»

Код клиента	Имя	Адрес	Город	Область	Почтовый индекс	Страна	Телефон	Руководитель
1	Музей научных достижений	Загородное шоссе, д. 150	Москва	Московская обл.	987654	РФ	(551) 999-01-01	Песоцкий Станислав
2	Blue Yonder Airlines	Ул. Гарибальди, д. 170	Пермь	Пермская обл.	995522	РФ	(5501) 99-01-01	Корепин Вадим
3	Винный завод	Ул. Кедрова, д. 54	Красноярск	Красноярский край	876444	РФ	(5502) 99-02-01	Ожогина Инна
4	Фармацевтическая фабрика	Ул. Губкина, д. 233	Тверь	Тверская обл.	897884	Россия	(5503) 99-01-03	Новиков Николай
5	Кофейная фабрика	Орликов пер, д. 896	Москва		876555	Россия	(5504) 99-01-04	Шашков Руслан
6	Почтовая служба	Лихов пер., 991	Красноярск	Красноярский край	876444	РФ	(5502) 99-02-01	Подколзина Екатерина
7	НИИ железа	Ул. Строителей, д. 150, кв. 78	Москва	Московская обл.	344554	РФ	(5505) 99-03-01	Попкова Дарья
8	Корпорация «Софт»	Ул. Заречная, 503	Москва	ИЛИ	134158	РФ	(5501) 99-01-01	Климов Сергей
9	Лесопитомник	Ул. Заречная, 504	Москва	ИЛИ	134158	РФ	(5501) 99-01-01	Сергиенко Мария

Помните, что хотя типы данных каждого поля таблицы могут не совпадать, они должны быть совместимыми, т. е. приложение Access должно быть в состоянии преобразовать данные исходной таблицы в тип данных, используемый в целевой таблице. Иногда в процессе преобразования некоторые данные могут быть удалены. Дополнительные сведения об

ограничениях преобразования типов данных см. в разделе [Ограничения преобразования типов данных](#).

Создание и выполнение запроса на обновление

Примечание. В описанных ниже действиях предполагается, что используются два приведенных выше образца таблиц. Описанные действия можно адаптировать к конкретным данным.

1. На вкладке **Конструктор** в группе **Макрос & код** нажмите кнопку **Конструктор запросов**.
2. В диалоговом окне **Добавление таблицы** откройте вкладку **Таблицы**.
3. Дважды щелкните исходную и целевую таблицы, чтобы добавить их в запрос, а затем нажмите кнопку **Заккрыть**. Каждая таблица появится в окне конструктора запросов.
4. В большинстве случаев Access автоматически объединяет связанные поля в запросе. Чтобы объединить поля со связанными сведениями вручную, перетащите их из одной таблицы в другую на место аналогичных полей.

Например, при использовании приведенных выше образцов таблиц перетащите поле «Код клиента» на поле «Код заказчика». Access создаст связь между этими полями в двух таблицах и будет использовать ее для объединения связанных записей.

5. На вкладке **Конструктор** в группе **Тип запроса** нажмите кнопку **Обновить**.
6. В целевой таблице дважды щелкните поля, которые нужно обновить. Каждое поле будет появляться в строке **Поле** бланка запроса.

При использовании образцов таблиц добавьте все поля, *за исключением* поля «Код заказчика». Обратите внимание, что имя целевой таблицы отображается в строке **Таблица** бланка запроса.

7. В строке запроса **Обновление** в каждый столбец, содержащий целевое поле, добавьте имя исходной таблицы и имя ее поля, соответствующего полю в целевой таблице, используя синтаксис **[Таблица].[Поле]**, где имена таблицы и поля заключены в квадратные скобки и разделены точкой.

На рисунке показано, как выглядит часть бланка запроса при использовании образцов таблиц. Обратите внимание на синтаксис имен таблицы и поля в строке **Обновление**.

Поле:	Имя	Адрес	Город	Республика и об
Таблица:	Клиенты	Клиенты	Клиенты	Клиенты
Обновление:	[Клиент].[Имя]	[Клиент].[Адрес]	[Клиент].[Город]	[Клиент].[Респу
Условия:				
или:				

Помните, что имена таблицы и поля в строке **Обновление** должны быть написаны правильно и что должна быть соблюдена любая пунктуация в именах исходной таблицы и поля. Регистр букв можно не учитывать.

8. На вкладке **Конструктор** в группе **Результаты** нажмите кнопку **Выполнить**.
9. В ответ на запрос подтверждения нажмите кнопку **Да**.

План-конспект занятия №24

Тема: *Извлечение информации из базы данных и их содержательный анализ с помощью запросов по установленным пользователям критериям (ОК 4.2.2)*

Вид занятия: лабораторное занятие №22

Цель занятия: Приобрести навыки работы с запросами

Ход работы:

Рассмотрим этапы разработки базы данных на примере создания базы данных «Контроль знаний».

Подготовим описание свойств полей для каждой из таблиц.

Таблица 2.2.1 - Описание полей таблицы ПРЕПОДАВАТЕЛИ

Имя поля	Тип данных	Размер
Код сотрудника	Текстовый	3
Фамилия	Текстовый	20
Имя	Текстовый	20
Отчество	Текстовый	20
Должность	Текстовый	15
Читаемые дисциплины	Текстовый	50
Телефон	Текстовый	10

Таблица 2.2.2 - Описание полей таблицы СТУДЕНТЫ

Имя поля	Тип данных	Размер
Код студента	Текстовый	3
Фамилия	Текстовый	20
Имя	Текстовый	20
Отчество	Текстовый	20
Группа	Текстовый	30
Год рождения	Дата/время	
Средний балл	Числовой	

Таблица 2.2.3 - Описание полей таблицы ФОРМЫ КОНТРОЛЯ ЗНАНИЙ

Имя поля	Тип данных	Размер
Номер	Счетчик	
Дата	Дата/время	
Код преподавателя	Текстовый	3
Код студента	Текстовый	3
Форма контроля знаний	Текстовый	20
Тема	Текстовый	50
Оценка	Числовой	

Создание запросов

Запрос на выборку

С помощью **запроса** можно выбрать необходимые данные из нескольких взаимосвязанных таблиц. Запрос можно сохранить под определенным именем для того, чтобы использовать его в любой момент или создать на его базе форму либо отчет. Через запрос можно производить обновление данных в таблицах, добавление, удаление записей, а также вычисления. Для создания запроса можно выбрать **Режим конструктора** или использовать **Мастер запросов**. Выберем объект **Запрос** и строку **Создание запроса в режиме конструктора**. В появившемся окне **Добавление таблицы** необходимо выбрать одну или несколько таблиц, из которых будут выбираться данные, и нажать кнопку **Добавить**. Далее закрыть окно **Добавление таблицы** и в окне **<Имя запроса>**: **запрос на выборку** перенести нужные поля из таблицы в строку **Поле** бланка запроса. Затем надо задать условия отбора, которые записываются в соответствующие столбцы первой строки **Условие отбора**. Между условиями в разных полях одной строки выполняется логическая операция «и».

Например, создадим запрос для вывода списка форм контроля, проведенных за последнюю неделю преподавателем Ивановым. (в качестве текущей даты примем 15 декабря 2005 года). Выберем строку **Создание запроса в режиме конструктора**. В окне **Добавление таблицы** выберем таблицы **ПРЕПОДАВАТЕЛИ** и **ФОРМЫ КОНТРОЛЯ ЗНАНИЙ**, нажмем кнопку **Закрыть**. В следующем окне перенесем в бланк запроса из таблицы **ПРЕПОДАВАТЕЛИ** поле **Фамилия**, из таблицы **ФОРМЫ КОНТРОЛЯ ЗНАНИЙ** поле **Форма контроля знаний** и поле **Дата** два раза, т.к. у нас два условия по дате: первое **Дата** <15

декабря, второе **Дата** > 8 декабря. Снимем галочку в поле **Дата** в третьем столбце для того, чтобы информация о дате не выводилась на экран два раза. Окно конструктора разработанного запроса представлено на рисунке 2.2.22.

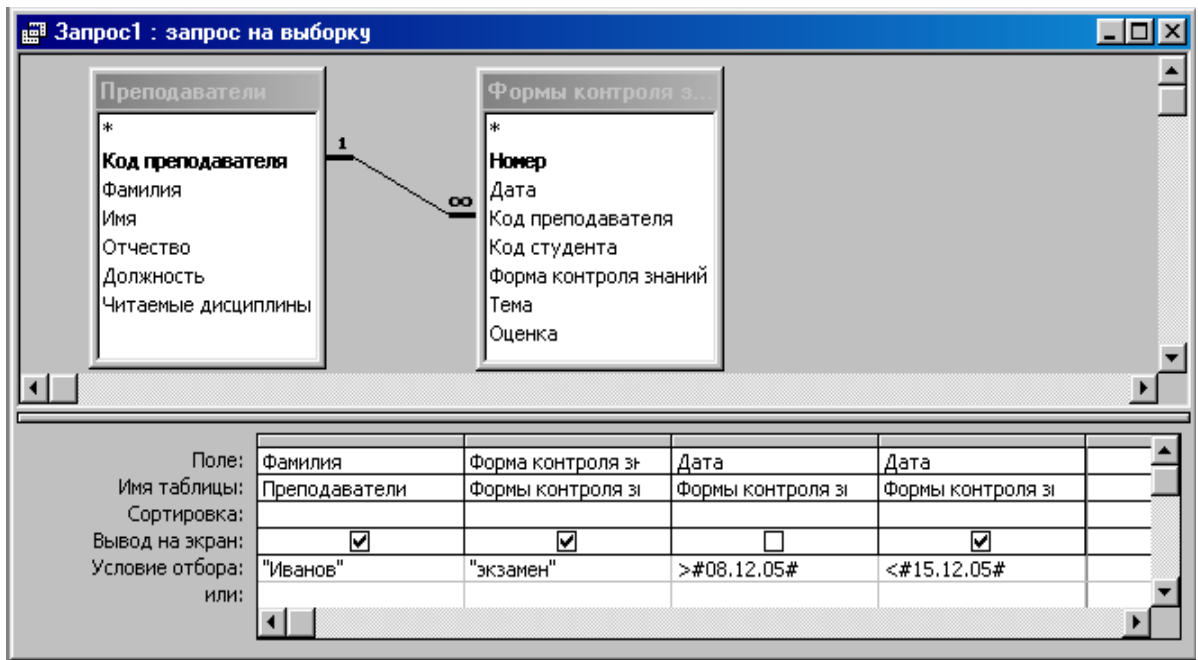


Рисунок 2.2.22 - Запроса на выборку в режиме конструктора

Для того чтобы выполнить запрос, надо нажать на панели конструктора запросов кнопку **Запуск** или кнопку **Вид/Режим таблицы**. На экране появится окно запроса в режиме таблицы с данными из таблиц ПРЕПОДАВАТЕЛИ и ФОРМЫ КОНТРОЛЯ ЗНАНИЙ, отвечающими заданным условиям отбора.

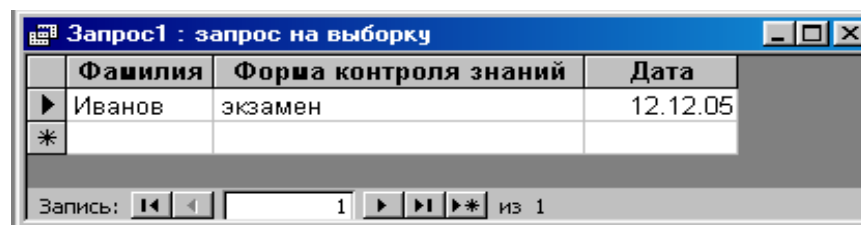


Рисунок 2.2.23 - Запроса на выборку в режиме таблицы

Запрос на обновление

Позволяет обновить все записи таблицы, а не редактировать каждую запись отдельно. Допустим, у всех сотрудников, имеющих номер телефона 23-45-67, нужно заменить номер на 34-56-78. Создадим запрос на обновление.

Выберем объект **Запросы** и **Режим конструктора** создания запроса. Добавим одну таблицу ПРЕПОДАВАТЕЛИ. Перенесем в бланк запроса поле **Телефон**. Выполним команду **Запрос/Обновление**. В строку **Условие отбора** введем старый номер, а в строку **Обновление** – новый номер телефона, см. рисунок 2.2.24. Щелкнем по кнопке **Запуск** на панели инструментов и в ответ на предложение обновить записи в таблице ПРЕПОДАВАТЕЛИ щелкните по кнопке **Да**.

Аналогично можно составить запрос на удаление записей по заданному условию.

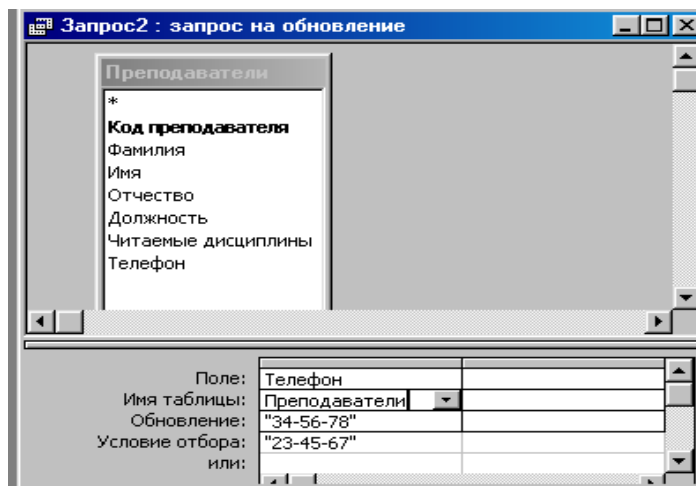


Рисунок 2.2.24 - Запроса на обновление в режиме конструктора

Итоговый запрос

Позволяет обобщить данные таблицы и рассчитать для определенного поля сумму, среднее, минимальное и максимальное значения. Для этих целей можно использовать Мастер простых запросов или создать запрос самостоятельно в режиме конструктора. Подсчитаем общее количество форм контроля, проведенных преподавателями. Создадим сначала базовый запрос в режиме конструктора, в который включим нужные нам поля из двух таблиц: поле **Фамилия** из таблицы ПРЕПОДАВАТЕЛИ и поле **Дата** из таблицы ФОРМЫ КОНТРОЛЯ ЗНАНИЙ. Сохраним запрос с именем **База** (см. рисунок 2.2.25).

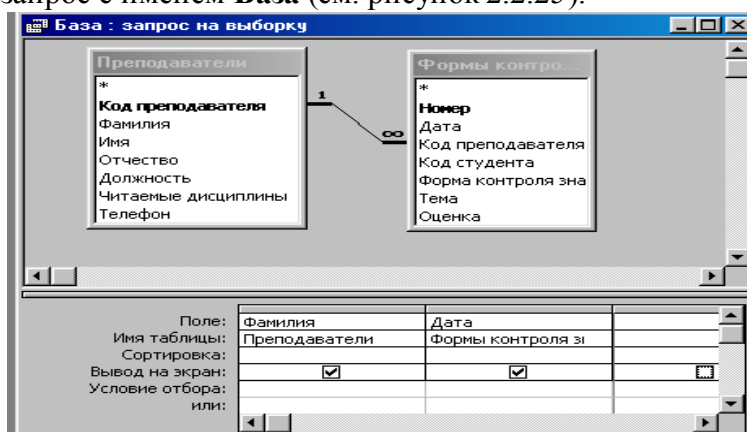


Рисунок 2.2.25 - Базовый запрос в режиме конструктора

Затем создадим **итоговый запрос**. Выбрав объект **Запросы**, нажмем на кнопку **Создать** и выберем **Простой запрос**. В раскрывающемся списке выберем наш базовый запрос **База**. Нажмем на кнопку **>>** для того, чтобы выбрать оба поля, эти поля переместятся в правое окошко. Нажмем кнопку **Далее**. В следующем окне установим переключатель **Итоговый** и щелкнем по кнопке **Итоги**. Установим необходимый флажок, в нашем примере это **Подсчет записей**, нажмем на **ОК**, затем на кнопку **Далее**, выберем интервал группировки дат и в следующем окне введем имя запроса и нажмем кнопку **Готово**, Результат см. на рисунок 2.2.26.

Фамилия	First - Дата	Count - Формы контроля знаний
Дьяконов	12.03.05	2
Иванов	12.12.05	1
Никитина	12.12.04	1

i. Рисунок 2.2.26 - Окно итогового запроса

Перекрестный запрос

Обеспечивает подведение итогов по данным из таблицы или запроса, группируя их по двум или нескольким полям. Результат представляется в виде перекрестной таблицы. В первом столбце перекрестной таблицы отображаются значения из полей базовой таблицы или запроса; количество полей для первого столбца может достигать до трех. Из значений другого поля формируются заголовки столбцов.

Например, построим перекрестный запрос для анализа форм контроля, проведенных преподавателями со студентами. Перед тем как запустить **Мастер перекрестных запросов** следует сформулировать **базовый запрос**, объединяющий нужные нам поля из разных таблиц. Ниже показан базовый запрос, который объединяет поля таблиц, см. рисунок 2.2.27.

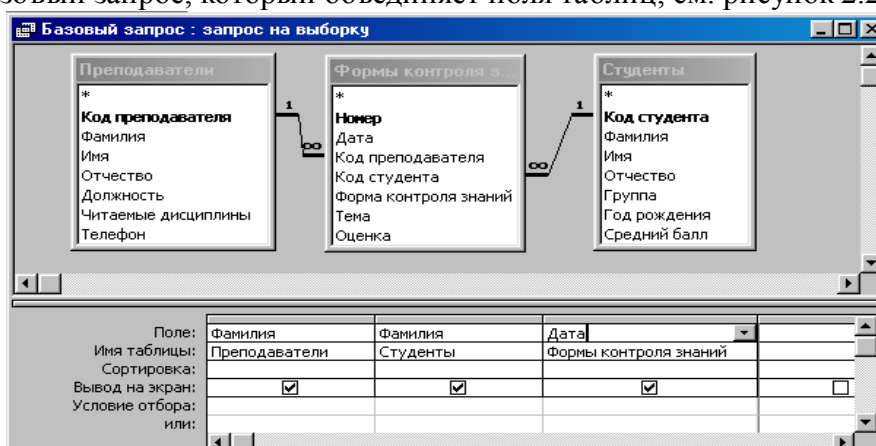


Рисунок 2.2.27 - Базовый запрос в режиме конструктора

После создания базового запроса, который включает в себя все необходимые для перекрестного запроса поля, можно запускать Мастер перекрестных запросов. В окне базы данных выберите объект **Запросы**, а затем щелкните по кнопке **Создать**. В диалоговом окне **Новый запрос** дважды щелкните на строке **Перекрестный запрос**. На первом шаге мастера установите переключатель **Запросы** и в списке запросов выделите базовый запрос. Нажмите на кнопку **Далее**. На втором шаге мастера задайте поля для заголовков строк, выберите **Фамилию преподавателя** и нажмите на **Далее**. На следующем шаге выделите поле **Фамилия студента**, значения этого поля будут использоваться в качестве заголовков столбцов. На следующем шаге выберите в списке функций функцию **Число**, нажмите **Далее**, задайте имя запроса и нажмите на кнопку **Готово**.

Перекрестная таблица, полученная в результате, показана на рисунке 2.2.28.

Фамилия	Итоговое значение	Иванов	Петров	Сидоров
Дьяконов	2	1		1
Иванов	1			1
Никитина	1		1	

Запись: 3 из 3

Рисунок 2.2.28 - Результат выполнения перекрестного запроса.

Тема: Создание сводной диаграммы

Вид занятия: лабораторное занятие №23

Цель занятия: Приобрести навыки построения сводной диаграммы в БД.

Задание:

1. Для БД, разработанной на прошлом занятии, построить сводную диаграмму

Порядок выполнения работы:

Диаграммы позволяют наглядно представить данные из таблиц или запросов. Диаграмму можно добавить в форму или отчет. Перед тем как запустить **Мастер диаграмм**, нужно убедиться в том, что в отчете или запросе имеются все данные, необходимые для создания диаграммы.

Построим диаграмму для наглядного представления количества форм контроля, проведенных преподавателями, на основании созданного нами перекрестного запроса. В окне базы данных выберем объект **Формы** и нажмем на кнопку **Создать**. В окне **Новая форма** выберем **Диаграмма**, в раскрывающемся списке выделим имя перекрестного запроса (см.рисунок 2.2.32) и нажмем кнопку **ОК**.

В окне мастера поместим поля **Фамилия** и **Итого** в список **Поля диаграммы** и нажмем на **Далее**. В следующем окне выберем тип диаграммы **Объемная гистограмма** и нажмем на **Далее**. В следующем окне нажмем на кнопку **Далее**, а в появившемся окне наберем название диаграммы и нажмем на кнопку **Готово**. Для редактирования диаграммы откройте форму с диаграммой в режиме конструктора и вызовите для нее контекстное меню. В контекстном меню выполните команду **Объект Диаграмма/Изменить**. Диаграмма, демонстрирующая количество форм контроля проведенная преподавателями, представлена на рисунке 2.2.33.

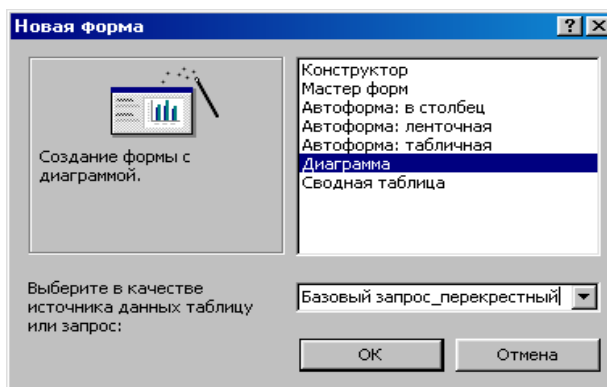


Рисунок 2.2.32 – Окно новой формы

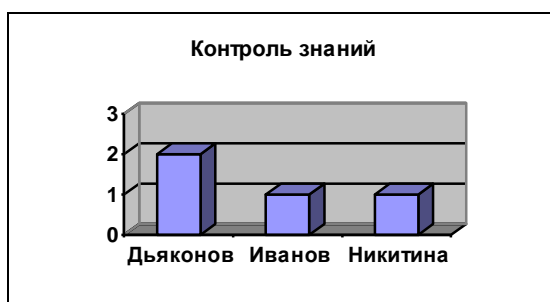


Рисунок 2.2.33 - Построенная диаграмма.

Построенную диаграмму можно сохранить, задав ей имя.

Тема: Создание учетных записей

Вид занятия: лабораторное занятие №24

Цель занятия: Приобрести навыки защиты информации в БД

Учетные записи пользователей предоставляют отдельным пользователям определенные привилегии доступа к сведениям и ресурсам базы данных. Учетные записи групп содержат несколько учетных записей пользователей и предоставляют средства контроля и управления разрешениями и доступом этих групп к объектам базы данных.

Создание учетных записей, добавление пользователей и печать сведений

–Создание учетной записи администратора

Для выполнения этих действий необходимо войти в базу по учетной записи члена группы «Admins».

1. Запустите Microsoft Access в рабочей группе с включенными средствами безопасности.

–Инструкции

Файл рабочей группы Microsoft Access содержит список пользователей, совместно использующих данные. Такую группу пользователей называют рабочей группой. Пароли пользователей также запоминаются в файле рабочей группы. Для управления доступом к базе данных необходимо создать новый файл рабочей группы.

1. Запустите Microsoft Access.

2. В меню **Сервис** выберите команду **Защита**, а затем команду **Администратор рабочих групп**.

3. В диалоговом окне **Администратор рабочих групп** нажмите кнопку **Создать**.

4. В диалоговом окне **Сведения о владельце рабочей группы** введите свое имя, название организации и любое сочетание букв и цифр длиной до 20 знаков в качестве кода рабочей группы (WID).

Обязательно запишите введенное имя, название организации и код рабочей группы с учетом регистра (для всех трех элементов) и храните их в надежном месте. Если потребуется заново создать файл рабочей группы, необходимо будет ввести точно такие же сведения. Если они забыты или потеряны, восстановить их и получить доступ к базам данных будет невозможно.

5. Введите имя нового файла рабочей группы. Чтобы сохранить его в другой папке, введите новый путь или нажмите кнопку **Обзор**, чтобы выбрать новый путь.)

6. Нажмите кнопку **ОК**.

Новый файл рабочей группы будет использоваться при следующем запуске Microsoft Access. Любые создаваемые учетные записи пользователей и учетные записи групп, а также их пароли, сохраняются в новом файле рабочей группы. Чтобы присоединить к рабочей группе, определенной новым файлом рабочей группы, других пользователей, скопируйте этот файл в общую папку (если он не был сохранен на шаге 5 в общей папке); после этого каждый пользователь должен будет запустить «Администратор рабочих групп» и присоединиться к новому файлу рабочей группы.

Важно. Для защиты базы данных не следует использовать стандартную рабочую группу, которая определяется файлом рабочей группы, созданным при установке Microsoft Access. Рекомендуется использовать файл рабочей группы, созданный с помощью уникального кода рабочей группы (WID), а при отсутствии такого файла следует его создать.

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду **Пользователи и группы**.

4. На вкладке **Пользователи** нажмите кнопку **Создать**.

5. В диалоговом окне **Новый пользователь или группа** введите имя учетной записи администратора и [личный код \(PID\)](#), а затем нажмите кнопку **ОК**, чтобы создать новую учетную запись.

–[Соглашения о паролях](#)

Используйте надежные пароли, представляющие собой сочетание прописных и строчных букв, цифр и символов. Пароли, не содержащие набор таких элементов, являются ненадежными. Надежный пароль: Y6dh!et5. Ненадежный пароль: House27. Пароли должны состоять не менее чем из 8 знаков. Рекомендуется использовать фразу-пароль, состоящую из 14 или более знаков. Дополнительные сведения см. в разделе [Защита личных сведений с помощью надежных паролей](#).

Важно помнить свой пароль. Если вы забыли пароль, корпорация Майкрософт не сможет его восстановить. Записывайте пароли и храните в безопасном месте отдельно от данных, для защиты которых они используются.

Имена учетных записей могут иметь длину от 1 до 20 знаков и могут состоять из букв алфавита, цифр, пробелов и знаков из расширенных наборов, за исключением следующих:

- знаки "\ [] : | < > + = ; , . ? *
- пробелы в начале имени;
- управляющие знаки (с кодами ASCII от 10 до 31).

Примечание. В паролях [учитывается регистр знаков](#).

Запишите имя и код учетной записи, с учетом регистра знаков, и храните их в надежном месте. Эти сведения потребуются для воссоздания учетной записи. Если имя или код учетной записи потеряны или забыты, восстановить их будет невозможно.

Примечания

- Имя учетной записи пользователя не может совпадать с именем учетной записи группы, и наоборот.
- Созданный на шаге 5 личный код не является паролем. Microsoft Access использует код и имя пользователя как исходные значения для алгоритма шифрования, создающего защищенный идентификатор [учетной записи пользователя](#).

1. В списке **Имеющиеся группы** выберите **Admins** и нажмите кнопку **Добавить**.

Новая учетная запись администратора будет добавлена в группу «Admins», а в списке **Участие в группе** появится имя группы **Admins**.

–[Создание учетной записи пользователя](#)

Для выполнения этих действий необходимо войти в базу по учетной записи члена [группы «Admins»](#).

Примечание. Проще всего организовать систему защиты, распределяя пользователей по группам и определяя [разрешения](#) на доступ для групп, а не для отдельных пользователей.

1. Запустите Microsoft Access с использованием той [рабочей группы](#), которой будет принадлежать учетная запись.

Важно. Учетные записи, созданные для пользователей базы данных, должны сохраняться в том [файле рабочей группы](#), который эти пользователи будут использовать для работы с базой данных. Если для создания базы данных используется другая рабочая группа, измените рабочую группу перед созданием учетных записей. Найти используемый файл рабочей группы можно в администраторе рабочих групп (команда **Защита** в меню **Сервис**).

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду

Пользователи и группы.

4. На вкладке **Пользователи** нажмите кнопку **Создать**.

5. В диалоговом окне **Новый пользователь или группа** введите имя новой учетной записи и [личный код \(PID\)](#), и нажмите **ОК** для создания новой учетной записи, которая автоматически будет добавлена в [группу «Users»](#)

Запишите имя и код учетной записи с учетом регистра знаков и храните их в надежном месте. Для воссоздания учетной записи, которая была удалена или создана в другой рабочей

группе, необходимо будет ввести точно такие же сведения. Если имя или код учетной записи потеряны или забыты, восстановить их будет невозможно.

Примечания

- Имя учетной записи пользователя не может совпадать с именем учетной записи группы, и наоборот.
- Созданный на шаге 5 личный код не является паролем. Microsoft Access использует код и имя пользователя как исходные значения для алгоритма шифрования, создающего защищенный идентификатор [учетной записи пользователя](#).

–Создание учетной записи группы

Частью системы защиты базы данных являются [учетные записи групп](#), входящих в [рабочую группу](#) Microsoft Access. Создание группы позволяет присваивать одинаковый набор [разрешений на доступ](#) сразу нескольким пользователям.

Для выполнения этих действий необходимо войти в базу по учетной записи члена [группы «Admins»](#).

1. Запустите Microsoft Access с использованием той рабочей группы, которой будет принадлежать учетная запись.

Важно. Учетные записи, созданные для пользователей базы данных, должны сохраняться в том [файле рабочей группы](#), который эти пользователи будут использовать для работы с базой данных. Если для создания базы данных используется другая рабочая группа, измените рабочую группу перед созданием учетных записей. Сменить рабочую группу или найти используемый файл рабочей группы можно в администраторе рабочих групп (команда **Защита** в меню **Сервис**).

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду

Пользователи и группы.

4. На вкладке **Группы** нажмите кнопку **Создать**.

5. В диалоговом окне **Новый пользователь или группа** введите имя новой учетной записи и [личный код \(PID\)](#).

Запишите имя и код учетной записи с учетом регистра знаков и храните их в надежном месте. Для воссоздания учетной записи, которая была удалена или создана в другой рабочей группе, необходимо будет ввести точно такие же сведения. Если имя или код учетной записи потеряны или забыты, восстановить их будет невозможно.

Примечание. Имя учетной записи пользователя не может совпадать с именем учетной записи группы, и наоборот.

6. Для создания новой учетной записи группы нажмите кнопку **ОК**.

Примечание. Созданный на шаге 5 личный код не является паролем. Microsoft Access использует код и имя пользователя как исходные значения для алгоритма шифрования, создающего защищенный идентификатор [учетной записи пользователя](#).

–Добавление пользователей в группы

Для выполнения данной процедуры необходимо войти в базу по учетной записи члена [группы «Admins»](#).

1. Запустите Microsoft Access с использованием [рабочей группы](#), содержащей [учетные записи пользователей](#) и [учетные записи групп](#).

Для проверки имени текущей рабочей группы или для изменения рабочей группы используйте служебную программу «Администратор рабочих групп».

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду

Пользователи и группы.

4. На вкладке **Пользователи** введите в поле **Имя** имя пользователя, добавляемого в группу.

5. В списке **Имеющиеся группы** выберите группу, в которую добавляется пользователь, и нажмите кнопку **Добавить**.

Имя выбранной группы появится в списке **Участие в группе**.

6. Повторите шаг 5, если требуется добавить текущего пользователя в другие группы. Повторите шаги 4 и 5 для добавления в группы других пользователей.

–Печать сведений о пользователях и группах

Примечание. Если файл рабочей группы создан в Microsoft Access 95 или более поздней версии, то все пользователи, входящие в рабочую группу, могут выводить на печать сведения об учетных записях пользователей и групп. Для печати сведений об учетных записях пользователей и групп при использовании файла рабочей группы, созданного в Microsoft Access 2.0, необходимо подключиться как член группы «Admins».

1. Запустите Microsoft Access с использованием файла рабочей группы, сведения из которого требуется напечатать.

Для проверки имени текущего файла рабочей группы или для изменения рабочей группы следует использовать администратор рабочих групп.

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду

Пользователи и группы.

4. На вкладке **Пользователи** нажмите кнопку **Распечатать отчет**.

5. В диалоговом окне **Печать сведений о защите** выполните одно из следующих действий.

Чтобы распечатать отчет, содержащий сведения о всех пользователях, входящих в текущую рабочую группу, выберите вариант **только пользователей**.

Чтобы распечатать отчет, содержащий сведения о всех группах, входящих в текущую рабочую группу, выберите вариант **только групп**.

Чтобы распечатать сведения об учетных записях как пользователей, так и групп, выберите вариант **пользователей и групп**.

Удаление учетных записей и пользователей

–Удаление пользователей из групп

Для выполнения этих действий необходимо войти в базу по учетной записи члена группы «Admins».

Примечания

- Не допускается удаление пользователей из стандартной группы «Users». Microsoft Access автоматически добавляет в группу «Users» всех пользователей. Для удаления любой учетной записи пользователя из группы «Users» необходимо удалить саму учетную запись.

- В каждый момент времени по крайней мере один пользователь должен входить в стандартную группу «Admins».

1. Запустите Microsoft Access в рабочей группе, содержащей учетные записи пользователей и групп.

Для проверки имени текущей рабочей группы или для изменения рабочей группы используйте служебную программу «Администратор рабочих групп».

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду

Пользователи и группы.

4. В поле **Имя** на вкладке **Пользователи** введите имя учетной записи пользователя, которую требуется удалить.

5. В списке **Участие в группе** выберите имя группы, из которой удаляется пользователь, и нажмите кнопку **Удалить**.

6. Для удаления текущего пользователя из других групп повторите шаг 5. Повторите шаги 4 и 5 для удаления из групп других пользователей.

–Удаление учетной записи пользователя

Для выполнения этих действий необходимо войти в базу по учетной записи члена группы «Admins».

Примечание. Учетную запись пользователя «Admin» удалить нельзя.

1. Откройте базу данных.

2. В меню **Сервис** выберите команду **Защита**, а затем команду **Пользователи и группы**.

3. На вкладке **Пользователи** выберите имя пользователя в списке **Имя** и нажмите кнопку **Удалить**.

4. Нажмите кнопку **Да**, чтобы удалить учетную запись пользователя.

5. Повторите шаги 3 и 4 для удаления других учетных записей; затем нажмите кнопку **ОК**.

–Удаление учетной записи группы

Для выполнения этих действий необходимо войти в базу по учетной записи члена группы [«Admins»](#).

Примечание. Учетные записи групп «Admins» и [Users](#) удалить нельзя.

1. Запустите Microsoft Access с использованием той [рабочей группы](#), которой принадлежит удаляемая учетная запись.

Для проверки имени текущей рабочей группы или для изменения рабочей группы используйте служебную программу «Администратор рабочих групп».

2. Откройте базу данных.

3. В меню **Сервис** выберите команду **Защита**, а затем команду **Пользователи и группы**.

4. На вкладке **Группы** выберите имя группы в раскрывающемся списке **Имя** и нажмите кнопку **Удалить**.

5. Нажмите кнопку **Да**, чтобы удалить учетную запись группы.

6. Повторите шаги 4 и 5 для удаления других учетных записей группы.

План-конспект занятия №28

Тема: Назначение прав доступа к БД

Вид занятия: *лабораторное занятие №25*

Цель занятия: Приобрести навыки сканирования графических изображений сканером Epson; Освоить приёмы работы с программой оптического распознавания текстов.

Просмотр или передача прав владельца

–Просмотр и передача прав собственности на отдельные объекты в базе данных со средствами безопасности

При наличии разрешения администратора на доступ к таблице, запросу, форме, отчету или макросу можно получить сведения о [владельце](#) объекта и передать его права другому пользователю или группе.

Примечание. Если права владельца таблицы, запроса, формы, отчета или макроса передаются учетной записи группы, то эти права автоматически получают все пользователи, принадлежащие к группе.

1. Откройте базу данных.

2. В меню **Сервис** выберите команду **Защита**, а затем команду **Разрешения**.

На вкладке **Смена владельца** отображается список таблиц, запросов, форм, отчетов и макросов, выведенных в данный момент в [окне базы данных](#), и имена текущих владельцев этих объектов.

3. Выберите тип объектов в поле **Тип объекта**.

4. В списке **Объект** выберите один или несколько объектов, для которых требуется произвести смену владельца. Для одновременного выделения нескольких имен объектов проведите указатель по именам объектов при нажатой кнопке мыши или выбирайте имена объектов при нажатой клавише CTRL.

5. В раскрывающемся списке **Новый владелец** выберите имя учетной записи пользователя или группы, которой требуется передать права владельца объекта или объектов.

6. Нажмите кнопку **Сменить владельца**.

–Передача прав владельца всей базы данных другому администратору

Примечание. Для импорта базы данных необходимо иметь разрешение «Открытие/запуск» для базы данных и «Чтение макета» на ее объекты. Для импорта таблиц

необходимо, кроме того, иметь разрешение «Чтение данных». Если разрешения имеются не на все таблицы, запросы, формы, отчеты и макросы, Microsoft Access импортирует только те объекты, на которые имеются достаточные разрешения.

1. Запустите Microsoft Access с помощью [рабочей группы с включенными средствами безопасности](#), в которую входит учетная запись пользователя, которой требуется передать права владельца базы данных и ее объектов.

–Инструкции

[Файл рабочей группы](#) Microsoft Access содержит список пользователей, совместно использующих данные. Такую группу пользователей называют [рабочей группой](#). Пароли пользователей также запоминаются в файле рабочей группы. Для управления доступом к базе данных необходимо создать новый файл рабочей группы.

1. Запустите Microsoft Access.

2. В меню **Сервис** выберите команду **Защита**, а затем команду **Администратор рабочих групп**.

3. В диалоговом окне **Администратор рабочих групп** нажмите кнопку **Создать**.

4. В диалоговом окне **Сведения о владельце рабочей группы** введите свое имя, название организации и любое сочетание букв и цифр длиной до 20 знаков в качестве [кода рабочей группы \(WID\)](#).

Обязательно запишите введенное имя, название организации и код рабочей группы с учетом регистра (для всех трех элементов) и храните их в надежном месте. Если потребуются заново создать файл рабочей группы, необходимо будет ввести точно такие же сведения. Если они забыты или потеряны, восстановить их и получить доступ к базам данных будет невозможно.

5. Введите имя нового файла рабочей группы. По умолчанию файл рабочей группы сохраняется в папке языка версии. Чтобы сохранить его в другой папке, введите новый путь или нажмите кнопку **Обзор**, чтобы выбрать новый путь.)

Новый файл рабочей группы будет использоваться при следующем запуске Microsoft Access. Любые создаваемые [учетные записи пользователей](#) и [учетные записи групп](#), а также их пароли, сохраняются в новом файле рабочей группы. Чтобы присоединить к рабочей группе, определенной новым файлом рабочей группы, других пользователей, скопируйте этот файл в общую папку (если он не был сохранен на шаге 5 в общей папке); после этого каждый пользователь должен будет запустить «Администратор рабочих групп» и присоединиться к новому файлу рабочей группы.

Для проверки имени текущей рабочей группы или для изменения рабочей группы используйте служебную программу «Администратор рабочих групп».

2. Подключитесь с помощью этой учетной записи.

3. Создайте новую базу данных.

4. Импортируйте в новую базу данных все объекты из исходной базы данных.

–Инструкции

1. Откройте базу данных или переключитесь в [окно базы данных](#) открытой базы данных.

2. Выберите в меню **Файл** команду **Внешние данные** и подкоманду **Импорт**.

3. Убедитесь, что в поле со списком **Тип файла** выбран элемент **Microsoft Access (*.mdb; *.adp; *.mda; *.mde; *.ade)**.

4. Щелкните стрелку справа от поля **Папка**, выберите диск и папку, где расположены [база данных Microsoft Access \(.mdb\)](#) или [проект Microsoft Access \(.adp\)](#), предназначенные для [импорта](#), и дважды щелкните значок базы данных.

5. В диалоговом окне **Импорт объектов** выберите вкладку, соответствующую типу импортируемого объекта, и выделите объекты, которые нужно импортировать, либо нажмите кнопку **Выделить все**, чтобы импортировать все объекты данного типа. Повторите этот шаг для всех нужных типов объектов.

РАЗДЕЛ 5. КОНТРОЛЬ ЗНАНИЙ

ПО МДК 02.02. Технология разработки и защиты базы данных

5.1. Организация контроля и оценки освоения программы МДК

Текущий контроль при освоении МДК осуществляется в ходе учебных (аудиторных) занятий в следующих формах:

- устные или письменные тематические опросы,
- самостоятельная работа студента (составление сравнительной (сводной) таблицы, реферирование)
- проверка выполнения домашних заданий (аналитический обзор литературы определенной тематики, составление опорного конспекта)

Промежуточный контроль (аттестация) студентов осуществляется ежемесячно в рамках накопительной системы оценивания.

Рубежный контроль осуществляется в форме проверочно-итоговых занятий по разделам, предусмотренных рабочей программой.

Итоговый контроль освоения МДК осуществляется при проведении итогового тестирования.

Оценка усвоенных знаний при итоговом контроле осуществляется с помощью следующих форм заданий:

- ✓ с выбором правильного ответа из фиксированного набора вариантов.

Оценка усвоенных умений при итоговом контроле осуществляется с помощью:

- ✓ тестовых заданий с развернутым ответом.

Условием допуска к дифференцированному зачету по учебной дисциплине «Технические средства информатизации» является положительная оценка промежуточной аттестации по учебной дисциплине.

Условием положительной аттестации студента по МДК является положительная оценка освоения всех умений и знаний по всем контролируемым показателям.

5.2. Текущий контроль

В данном разделе представлены задания для текущего контроля по темам рабочей программы **МДК 02.02. Технология разработки и защиты базы данных**

Текущий контроль осуществляется в виде так называемых блиц-опросов, которые позволяют преподавателю достаточно быстро и результативно оценить знания обучающихся. В каждом блиц-опросе 5 вопросов. Количество правильных ответов сразу дает оценку по 5-балльной системе. Каждый вопрос сопровождается несколькими вариантами ответов с множественным выбором.

Блиц-опрос №1

Учебник: Голицина (с.9-13)

1. . Что из перечисленного НЕ является основополагающим в развитии автоматизированных систем:

- А) развитие вычислительной техники
- Б) появление емких внешних носителей
- В) появление языка BASIC
- Г) появление информационно-поисковых систем

2. Какое понятие шире – банк данных или база данных:

- А) Банк данных
- Б) база данных

3. Назовите характерную черту баз данных:

- А) архивирование данных
- Б) Постоянство

- В) централизованное хранение данных
- Г) многопользовательский интерфейс

4. Возможность обращаться к данным БД с помощью различных методов доступа – это такое требование к БД, как:

- А) многократное использование
- Б) простота
- В) легкость использования
- Г) гибкость использования

5. Возможность автоматического восстановления данных такое требование к БД, как:

- А) автоматическая реорганизация
- Б) целостность данных
- В) восстановление данных после сбоя
- Г) сохранение затрат умственного труда

1	2	3	4	5
<i>В</i>	<i>А</i>	<i>Б</i>	<i>Г</i>	<i>В</i>

Ответы:

Блиц-опрос №2

Учебник: Голицина (с. 13-17)

1. Нарисовать схему архитектуры клиент-сервер

1. Метаданные – это:

- А) данные, хранящиеся в БД
- Б) данные о данных
- В) данные о процедурах
- Г) данные о СУБД

2. Какой уровень представления данных используется программистом:

- А) концептуальный ~~с различных точек зрения~~
- Б) внутренний ~~для ВЗУ~~
- В) внешний ~~метаданные~~

3. Для какого уровня представления данных используется ЯОД (язык описания данных):

- А) концептуальный ~~с различных точек зрения~~
- Б) внутренний ~~(для ВЗУ)~~
- В) внешний ~~метаданные~~

4. Какое из лингвистических средств БД служит для организации обработки данных:

- А) ЯОД (язык описания данных)
- Б) ЯМД (язык манипулирования данными)
- В) SQL (Structured Query Language)

5. Какая из качественных характеристик языков запросов позволяет программировать, не зная этот язык полностью:

- А) селективная мощность языка
- Б) простота изучения
- В) уровень процедурности
- Г) модульность построения языка

1	2	3	4	5
<i>Б</i>	<i>В</i>	<i>Б</i>	<i>Б</i>	<i>Г</i>

Ответы:

Блиц-опрос №3

Учебник: Голицина (с. 18-22)

1. Какое из программных средств БД позволяет восстановить ее после сбоя:

- А) ядро БД
- Б) процессор языка БД
- В) библиотека поддержки вызовов
- Г) сервисные программы

2. Какое из программных средств БД обеспечивает обработку запросов:

- А) ядро БД
- Б) процессор языка БД
- В) библиотека поддержки вызовов
- Г) сервисные программы

3. У какого пользователя БД больше прав доступа:

- А) конечный пользователь
- Б) администратор БД
- В) администратор приложений
- Г) прикладной программист

4. БД «Телефонный справочник» по типу хранимой информации относится к:

- А) фактографическим
- Б) документальным
- В) лексикографическим

5. БД «Каталог древних книг» по функциональному назначению и степени доступности относится к:

- А) операционная с ограниченным доступом пользователей
- Б) справочно-информационная с ограниченным доступом пользователей
- В) справочно-информационная общедоступная
- Г) операционная общедоступная

Ответы:

1	2	3	4	5
<i>Г</i>	<i>Б</i>	<i>Б</i>	<i>Б</i>	<i>В</i>

ЗАДАНИЕ

Охарактеризовать предложенную БД по различным типам классификации

1 вариант

БД Продажа ж/д билетов

2 вариант

БД Электронный каталог библиотек

Ответы:

<i>№</i>	<i>1 вариант БД Продажа ж/д билетов</i>	<i>1 вариант БД Электронный каталог библиотек</i>
I	Фактографическая	Документальная
II	Фактографическая	Документальная
III	Реляционная	Реляционная
IV	Распределенная	Локальная
V	Интегрированная	Интегрированная
VI	Операционная	Справочно-информационная
VII	Универсальная	Универсальная
VIII	С ограниченным доступом пользователей	общедоступная

Блиц-опрос №4

А.В. Кузин. Модели данных (с.11-13)

1. Назовите количество классических моделей баз данных:

- А) 5
- Б) 4
- В) 3
- Г) 2

2. В какой модели БД поиск данных ведется всегда с корня модели:

- А) сетевая
- Б) иерархическая
- В) объектно-ориентированная
- Г) реляционная

3. Существует ли таблица в реляционной БД, которая не содержит строк:

- А) да
- Б) нет

4. Какая из названных моделей БД является в настоящее время промышленным стандартом в программировании:

- А) сетевая
- Б) иерархическая
- В) объектно-ориентированная
- Г) реляционная

5. Назовите самую «новую» модель БД в настоящее время:

- А) сетевая
- Б) иерархическая
- В) объектно-ориентированная
- Г) реляционная

Ответы:

1	2	3	4	5
<i>В</i>	<i>Б</i>	<i>А</i>	<i>Г</i>	<i>В</i>

Блиц-опрос №5

Учебник: *А.В. Кузин. Реляционная модель* (с.13-15)

1. Может ли существовать реляционная таблица, не содержащая ни одного поля:
А) да
Б) *нет*
2. Могут ли две таблицы одной БД иметь одинаковые имена? Почему?
А) да
Б) *нет*
3. Могут ли две таблицы одной БД иметь одинаковые ключевые поля?
А) да
Б) нет
В) *свой ответ*
4. Столбец (несколько столбцов), значения которых однозначно характеризуют значения первичного ключа другого отношения, называется:
А) первичным ключом
Б) индексом
В) составным ключом
Г) *внешним ключом*
5. Столбец (несколько столбцов), значения которых однозначно идентифицируют каждую строку таблицы, называется:
А) *первичным ключом*
Б) индексом
В) составным ключом
Г) *внешним ключом*

1	2	3	4	5
<i>Б</i>	<i>Б</i>	<i>А</i>	<i>Г</i>	<i>А</i>

Ответы:

Блиц-опрос №6

Учебник: *А.В. Кузин. Реляционная модель* (с.15-17)

1. Тип взаимосвязи в модели БД, при котором каждому экземпляру одного объекта А могут соответствовать несколько экземпляров второго объекта В и наоборот, называется:
А) один-к-одному
Б) один-ко-многим
В) *многие-ко-многим*
2. Назовите тип взаимосвязи в модели БД, при котором соответствие между объектами не взаимно-равнозначное:
А) один-к-одному
Б) *один-ко-многим*
В) *многие-ко-многим*
3. Какая из перечисленных взаимосвязей в модели БД реализуется в Access только посредством дополнительного промежуточного отношения:
А) один-к-одному
Б) *один-ко-многим*
В) *многие-ко-многим*
4. Какая из перечисленных взаимосвязей в модели БД может возникнуть в случае, когда оба связываемые атрибуты отношений – ключевые:
А) *один-к-одному*
Б) *один-ко-многим*
В) *многие-ко-многим*
5. Можно ли в качестве ключевого поля использовать текстовые атрибуты?
А) да
Б) нет
В) *свой ответ*

Ответы:

1	2	3	4	5
<i>В</i>	<i>Б</i>	<i>В</i>	<i>А</i>	<i>А,В</i>

Блиц-опрос №7

Учебник: *А.В. Кузин. Нормализация таблиц* (с.23-27)

1. Наложение ограничений на формирование таблиц с целью устранения дублирования данных и соблюдения их непротиворечивости называется

- А) декомпозиция таблиц
- Б) композиция таблиц
- В) *нормализация таблиц*
- Г) разложение таблиц

2. Нормализация таблиц НЕ предусматривает выполнения:

- А) *устранения ошибочности данных*
- Б) устранение дублирования данных
- В) устранение избыточности данных
- Г) непротиворечивости данных

3. Количество нормализованных таблиц в схеме данных равно:

- А) количеству атрибутов в исходной ненормализованной таблице
- Б) количеству известных нормальных форм
- В) количеству записей в исходных данных
- Г) *зависит от способов декомпозиции*

4. Известно, что таблица находится в третьей нормальной форме. Это означает, что она:

- А) *уже находится в первой нормальной форме*
- Б) *уже находится во второй нормальной форме*
- В) имеет три ключевых поля
- Г) связана с тремя таблицами в схеме данных

5. Для ускорения доступа к данным в БД используются методы:

- А) *индексирования данных*
- Б) композиции таблиц
- В) декомпозиции таблиц
- Г) *хеширования данных*

6. Какой из способов ускорения доступа к данным в БД требует значительных затрат памяти ПК:

- А) *индексирование данных*
- Б) композиция таблиц
- В) декомпозиция таблиц
- Г) *хеширование данных*

Ответы:

1	2	3	4	5	6
<i>В</i>	<i>А</i>	<i>Г</i>	<i>А,Б</i>	<i>А,Г</i>	<i>А</i>

5.3. Рубежный контроль

Варианты контрольной работы

Вариант №1

- 1.Перечислите типы организации баз данных.
- 2.Чем отличаются поля и записи таблицы?
- 3.В каком диалоговом окне создают связи между полями таблиц базы данных?
- 4.Какие элементы управления форм вы знаете?
- 5.Какие способы автоматической генерации отчетов вы знаете?

Вариант №2

- 1.Какую базу данных называют реляционной?
- 2.Какие типы полей вы знаете?
- 3.Почему при закрытии таблицы программа Access не предлагает выполнить сохранение внесенных данных?
- 4.При создании элемента управления формы рядом с ним автоматически образуется элемент управления, называемый присоединенной надписью. Как оторвать эту надпись от элемента управления, если разработчик не хочет, чтобы она располагалась слева, а желает поместить ее ниже элемента управления?
- 5.Что следует указать программе Access для автоматической генерации отчета? на базе чего выполняется отчет?

Вариант №3

- 1.Дайте сравнительную характеристику различных СУБД.
- 2.Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?
- 3.Понятие, виды и назначение окон.
- 4.Какой командой можно выровнять элементы управления формы, расположенные неаккуратно?
- 5.С какими еще объектами кроме таблиц, запросов, макросов, форм и отчетов работает СУБД Access? Для чего они предназначены?

Вариант №4

- 1.Что представляет собой нормализованная таблица?
- 2.Содержит ли какую-либо информацию таблица, в которой нет полей? Может ли существовать такая таблица?
- 3.Команды по созданию и управлению рабочим окном.
- 4.При заполнении таблиц удобно не вводить часто повторяющиеся данные, а выбирать их из заранее заготовленного списка. Как называются элементы управления формы, позволяющие это сделать? Как их установить?
- 5.Виды каскадных воздействий.

Вариант №5

- 1.Что представляет собой ключ? Какие виды ключей вам известны? Как их задать?
- 2.В чем состоит особенность поля типа Счетчик?
- 3.Для чего предназначены формы?
- 4.Можно ли использовать формы не только для ввода, но и для вывода данных? Если да, то на какое устройство компьютерной системы выполняется этот вывод?
- 5.Расскажите о требованиях, предъявляемых к базе данных.

Вариант №6

1. Типы взаимосвязей в модели данных.
2. В чем состоит особенность поля типа Метод?
3. Какие методы автоматического создания форм вы знаете?
4. Для чего предназначены отчеты?
5. Назначение, виды и хранение процедур.

Вариант №7

1. Расскажите о требованиях, предъявляемых к базе данных
2. Какое поле можно считать уникальным?
3. Расскажите о технологии разработки однотобличной формы.
4. На какое устройство компьютерной системы происходит вывод данных при работе отчета?
5. Перечислите типы организации баз данных.

Вариант №8

1. Расскажите об этапах проектирования и создания базы данных.
2. Каким должно быть поле первичного ключа?
3. Расскажите о технологии разработки формы для загрузки двух взаимосвязанных таблиц.
4. Из чего состоят отчеты?
5. Какую базу данных называют реляционной?

Вариант №9

1. Расскажите об этапах построения информационно-логической модели данных.
2. В чем преимущество поля, являющегося первичным ключом?
3. Из чего состоят формы?
4. Какие разделы отчетов вы знаете?
5. Типы взаимосвязей в модели данных.

Вариант №10

1. В каких объектах хранятся данные базы?
2. Поле какого типа является естественным кандидатом на роль ключевого поля?
3. Какие разделы форм вы знаете?
4. Что общего и в чем различие между разделами отчетов и разделами форм?
5. Что представляет собой нормализованная таблица?

5.4. Итоговый контроль

Комплект материалов для итоговой оценки сформированности умений и знаний

ЗАДАНИЕ ДЛЯ ОБУЧАЮЩИХСЯ

количество вариантов 31

Билет № 1

1. База данных - это:

- а) совокупность данных, организованных по определенным правилам;
- б) совокупность программ для хранения и обработки больших массивов информации;
- в) интерфейс, поддерживающий наполнение и манипулирование данными;
- г) определенная совокупность информации.

2. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без отчетов;
- в) без таблиц;
- г) без макросов.

3. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на выборку
- в) запрос на обновление,
- г) запрос на добавление

4. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

5. Представлена база данных "Телефонный справочник"

Фамилия И.О.	Телефон
Иванов И.И.	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке возрастания запись, содержащая номер телефона 568-98-00, переместится на

- а) не переместится;
- б) 1 строку вверх;
- в) 1 строку вниз;
- г) 3 строки вверх.

6. Представлена база данных "Классы школы"

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 8б классе переместятся на:

- а) 0 строк
- б) 1 строку вверх
- в) 1 строку вниз
- г) 2 строки вверх

7. Запросу (Физика = 5 или Информатика = 5) и Математика=5 и Первая_буква (Фамилия) = "И" в базе данных

№ записи	Фамилия	Имя	Математика	Физика	Информатика
1	Ипатов	Егор	4	4	5
2	Киреев	Александр	5	5	5
3	Илюхин	Сергей	5	4	4
4	Ивашкина	Татьяна	5	5	4

отвечают записи

- а) 1,3,4
- б) 4
- в) 3,4
- г) 2,4

Билет № 2

1. Наиболее распространенными в практике являются:

- а) распределенные базы данных;
- б) иерархические базы данных;
- в) сетевые базы данных;
- г) реляционные базы данных.

2. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

3. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без отчетов;
- в) без таблиц;
- г) без запросов

4. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Представлена база данных "Телефонный справочник"

Фамилия И.О.	Телефон
Иванов И.И-	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке убывания запись, содержащая сведения о телефоне 384-15-15, переместится на:

- а) 2 строки вверх
- б) 3 строки вверх;
- в) 4 строки вверх;
- г) не переместится

6. Представлена база данных "Классы школы"

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев

8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 8б классе переместятся на:

- а) 1 строку вверх
- б) 1 строку вниз
- в) 2 строки вверх
- г) 2 строки вниз

7. Запросу (Физика = 5 или Информатика = 5) и Математика=5 и Первая_буква (Фамилия) = "И" в базе данных

№ записи	Фамилия	Имя	Математика	Физика	Информатика
1	Ипатов	Егор	4	4	5
2	Киреев	Александр	5	5	5
3	Илюхин	Сергей	5	4	4
4	Ивашкина	Татьяна	5	5	4

отвечают записи

- а) 1,3,4
- б) 4
- в) 3,4
- г) 2,3

Билет № 3

1. Наиболее точным аналогом реляционной базы данных может служить:

- а) неупорядоченное множество данных;
- б) вектор;
- в) генеалогическое дерево;
- г) двумерная таблица.

2. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без отчетов;
- в) без таблиц;
- г) без форм;

3. Укажите, в каком из видов запроса результат сохраняется в виде отдельной таблицы:

- а) запрос на выборку
- б) запрос на создание таблицы
- в) запрос на обновление,
- г) запрос на добавление

4. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Представлена база данных "Телефонный справочник"

Фамилия И.О.	Телефон
Иванов И.И-	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке возрастания запись, содержащая номер телефона 568-98-00, переместится на

- а) 1 строку вверх;
- б) 2 строки вверх;
- в) 1 строку вниз;
- г) 3 строки вверх.

6. Представлена база данных "Отделы"

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Ракитский, переместятся на:

- а) 1 строку вверх
- б) 1 строку вниз
- в) 2 строки вверх
- г) 3 строки вверх

7. База данных "Зоопарк" содержит в полях Животное, Тип, Количество сведения о наименовании животного, типе его рациона (возможны два типа: П - плотоядное или Т - травоядное) и количестве пищи в день.

Для получения информации о животных, съедающих в день от 5 до 10 кг мясных продуктов, запрос имеет вид:

- а) Тип $\langle \rangle$ Т и $10 \leq \text{Количество} \geq 5$
- б) Тип = П и ($\text{Количество} \geq 5$ и $\text{Количество} \leq 10$)
- в) Тип $\langle \rangle$ Т и $10 \leq \text{Количество} \langle \rangle 5$
- г) Тип =П и ($\text{Количество} \geq 5$ и $\text{Количество} \geq 10$)

Билет № 4

1. **В каком диалоговом окне создают связи между полями таблиц базы данных:**
 - а) таблица связей;
 - б) схема связей;
 - в) схема данных;
 - г) таблица данных.

2. **В каких элементах таблицы хранятся данные базы:**
 - а) в строках;
 - б) в столбцах;
 - в) в записях;
 - г) в ячейках.

3. **Укажите, какой из фильтров использует бланк запроса:**
 - а) фильтр по выделенному
 - б) обычный фильтр
 - в) расширенный фильтр
 - г) фильтр на исключение выделенного

4. **Обеспечение целостности данных НЕ предусматривает:**
 - а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
 - б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
 - в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
 - г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

5. **Представлена база данных "Волшебные страны"**

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	148	46.9
ОЗ	155	95.3
ШВАМБРАНИЯ	132	53.5
ЛУКОМОРЬЕ	199	47.7
ЗАЗЕРКАЛЬЕ	211	76.2

После проведения сортировки сведения о НАРНИИ переместились на одну строку вниз. Сортировка проводилась в порядке

- а) убывания по полю ПЛОЩАДЬ;
- б) возрастания по полю ПЛОЩАДЬ;
- в) убывания по полю СТРАНА;
- г) возрастания по полю НАСЕЛЕНИЕ

6. Представлена база данных "Отделы"

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Ракитский, переместятся на:

- а) 0 строк
- б) 1 строку вниз
- в) 2 строки вверх
- г) 3 строки вверх

7. Запросу Серия = Для чайников или Год_издания >= 1996 в базе данных

N	Автор	Серия	Наименование	Год_издания	Кол_стр
1	Уолш Р.	Для начинающих	Windows 95	1996	128
2	Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
3	Кирсанов Д.	Для чайников	Word 7.0	1996	236
4	Визе М.	Компьютер для носорога	Access 2.0	1994	255

отвечают записи:

- а) только 3
- б) только 1,2,3;
- в) только 1,3
- г) только 2,3

Билет № 5

1. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в столбцах;
- в) в записях;
- г) в ячейках.

2. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без таблиц;
- в) без запросов
- г) без макросов.

3. **Обеспечение каскадного обновления связанных полей предусматривает:**
- отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
 - изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
 - запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
 - запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
4. **Укажите, какой из перечисленных типов запросов не является запросом действия:**
- запрос на удаление
 - запрос на выборку
 - запрос на обновление,
 - запрос на добавление
5. **Представлена база данных "Волшебные страны"**

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	148	46.9
ОЗ	155	95.3
ШВАМБРАНИЯ	132	53.5
ЛУКОМОРЬЕ	199	47.7
ЗАЗЕРКАЛЬЕ	211	76.2

После проведения сортировки сведения о НАРНИИ переместились на одну строку вниз. Сортировка проводилась в порядке

- возрастания по полю СТРАНА;
- возрастания по полю ПЛОЩАДЬ;
- убывания по полю СТРАНА;
- возрастания по полю НАСЕЛЕНИЕ

6. **Представлена база данных "Отделы"**

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Антонов, переместятся на:

- 1 строку вверх
- 1 строку вниз
- 2 строки вниз
- 3 строки вниз

7. *Запрос к базе данных с полями Фамилия, Год_рождения, Класс, Оценка для вывода списка учеников 10 классов, 1981 года рождения, имеющих оценки 4 или 5, содержит выражение*

- а) $\text{Класс} = 10$ или $\text{Оценка} > 4$ или $\text{Год_рождения} = 1981$
- б) $\text{Оценка} \geq 4$ и $\text{Год_рождения} = 1981$ и $\text{Класс} = 10$
- в) $\text{Оценка} \geq 4$ и $\text{Год_рождения} > 1980$ или $\text{Класс} = 10$
- г) $\text{Класс} = 10$ и $\text{Оценка} > 4$ или $\text{Год_рождения} = 1981$

Билет № 6

1. *Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?*

- а) пустая таблица не содержит ни какой информации;
- б) пустая таблица содержит информацию о структуре базы данных;
- в) пустая таблица содержит информацию о будущих записях;
- г) таблица без записей существовать не может.

2. *Каскадное удаление связанных записей предусматривает:*

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

3. *Укажите, какой из перечисленных типов запросов не является запросом действия:*

- а) запрос на удаление
- б) запрос на выборку
- в) запрос на обновление,
- г) запрос на добавление

4. *В каких элементах таблицы хранятся данные базы:*

- а) в полях;
- б) в строках;
- в) в записях;
- г) в ячейках.

5. *Представлена база данных "Телефонный справочник"*

Фамилия И.О.	Телефон
Иванов И.И-	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке убывания запись, содержащая сведения о телефоне 384-15-15, переместится на:

- а) не переместится
- б) 1 строки вверх
- в) 2 строки вверх;
- г) 3 строки вверх;

6. Представлена база данных "Отделы"

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Антонов, переместятся на:

- а) 0 строк
 - б) 1 строку вверх
 - в) 2 строки вниз
 - г) 3 строки вниз
7. **База данных "Зоопарк" содержит в полях Животное, Тип, Количество сведения о наименовании животного, типе его рациона** (возможны два типа: П - плотоядное или Т - травоядное) и количестве пищи в день. Для получения информации о животных, съедающих в день от 5 до 10 кг мясных продуктов, запрос имеет вид:
- а) Животное = (Тип=П или Количество=5)
 - б) Тип = П и (Количество >=5 и Количество <=10)
 - в) Тип <> Т и 10 <= Количество <> 5
 - г) Тип =П и (Количество >=5 и Количество >=10)

Билет № 7

1. Содержит ли какую-либо информацию таблица, в которой нет полей?

- а) содержит информацию о структуре базы данных;
- б) не содержит ни какой информации;
- в) таблица без полей существовать не может;
- г) содержит информацию о будущих записях.

2. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в строках;
- в) в столбцах;
- г) в записях

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице

- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. Какой из мастеров не предусмотрен в Access?

- а) мастер запросов
- б) мастер макросов
- в) мастер таблиц
- г) мастер отчетов

5. Представлена база данных "Волшебные страны"

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46.9
ОЗ	55	95.3
ШВАМБРАНИЯ	32	53.5
ЛУКОМОРЬЕ	99	47.7
ЗАЗЕРКАЛЬЕ	51	76.2

После проведения сортировки сведения о ЗАЗЕРКАЛЬЕ переместились на две строки вверх. Сортировка проводилась в порядке

- а) убывания по полю ПЛОЩАДЬ;
- б) возрастания по полю ПЛОЩАДЬ;
- в) убывания по полю СТРАНА;
- г) возрастания по полю НАСЕЛЕНИЕ

6. Представлена база данных "Отделы"

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Чеботарев, переместятся на:

- а) 1 строку вверх
- б) 1 строку вниз
- в) 2 строки вниз
- г) 2 строки вверх

7. Запросы Серия = Для чайников или Год_издания >= 1996 в базе данных

N	Автор	Серия	Наименование	Год_издания	Кол_стр
1	Уолш Р.	Для начинающих	Windows 95	1996	128
2	Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
3	Кирсанов Д.	Для чайников	Word 7.0	1996	236
4	Визе М.	Компьютер для носорога	Access 2.0	1994	255

отвечают записи:

- а) только 3
- б) только 1,2,3;
- в) только 1,3
- г) только 1

Билет № 8

1. Что из перечисленного не является объектом Access:

- а) модули;
- б) таблицы;
- в) макросы;
- г) ключи;

2. В чем состоит особенность поля "счетчик"?

- а) служит для ввода числовых данных;
- б) служит для ввода действительных чисел;
- в) имеет ограниченный размер;
- г) имеет свойство автоматического наращивания.

3. Укажите несуществующий тип связи информационных объектов:

- а) один - к – одному
- б) один – ко- многим
- в) многие – ко - многим
- г) один – ко- всем

4. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. *Представлена база данных "Волшебные страны"*

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46.9
БУЯН	156	47.7
ОЗ	155	95.3
ГУСТОЙ ЛЕС	99	53.5
ЗАЗЕРКАЛЬЕ	51	76.2

После проведения сортировки сведения о ГУСТОМ ЛЕСЕ переместились на две строки вверх. Сортировка проводилась в порядке:

- а) возрастания по полю СТРАНА;
- б) убывания по полю ПЛОЩАДЬ;
- в) возрастания по полю ПЛОЩАДЬ;
- г) убывания по полю СТРАНА

6. *Представлена база данных "Отделы"*

Отдел	Кол_сотр	Нач_отд
310а	27	Шпак
101в	26	Антонов
215	30	Чеботарев
101г	18	Ракитский
112	24	Кабанов

После сортировки в порядке возрастания по полю "Отдел" сведения об отделе, которым руководит Чеботарев, переместятся на:

- а) 0 строк
- б) 1 строку вверх
- в) 1 строку вниз
- г) 2 строки вниз

7. *Запрос к базе данных с полями Фамилия, Год_рождения, Класс, Оценка для вывода списка учеников 10 классов, 1981 года рождения, имеющих оценки 4 или 5, содержит выражение*

- а) $\text{Класс} > 10$ и $\text{Оценка} = 4$ и $\text{Год_рождения} = 1981$
- б) $\text{Оценка} \geq 4$ и $\text{Год_рождения} = 1981$ и $\text{Класс} = 10$
- в) $\text{Оценка} \geq 4$ и $\text{Год_рождения} > 1980$ или $\text{Класс} = 10$
- г) $\text{Класс} = 10$ и $\text{Оценка} > 4$ или $\text{Год_рождения} = 1981$

Билет № 9

1. Какое поле можно считать уникальным?

- а) поле, значения в котором не могут повторяться;
- б) поле, которое носит уникальное имя;
- в) поле, значение которого имеют свойство наращивания
- г) поле для ввода текстовых данных

2. Что из перечисленного не является объектом Access:

- а) ключи;
- б) формы;
- в) отчеты;
- г) страницы

3. Какой из мастеров не предусмотрен в Access?

- а) мастер форм
- б) мастер подстановок
- в) мастер диаграмм
- г) мастер макросов

4. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Представлена база данных "Вошебные страны"

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46.9
ОЗ	55	95.3
ШВАМБРАНИЯ	32	53.5
ЛУКОМОРЬЕ	99	47.7
ЗАЗЕРКАЛЬЕ	51	76.2

После проведения сортировки сведения о ЗАЗЕРКАЛЬЕ переместились на две строки вверх. Сортировка проводилась в порядке

- а) возрастания по полю ПЛОЩАДЬ;
- б) возрастания по полю НАСЕЛЕНИЕ
- в) возрастания по полю СТРАНА;
- г) убывания по полю ПЛОЩАДЬ

6. *Представлена база данных "Гостиница"*

Номер	Кол_комнат	Цена
210л	27	500
101о	26	30
215д	30	80
101д	18	50
112пл	24	250

После сортировки в порядке возрастания по полю "Номер" сведения о номере ценой 500 переместятся на:

- а) 4 строки вниз
- б) 3 строки вниз
- в) 2 строки вниз
- г) 1 строку вниз

7. *База данных "Зоопарк" содержит в полях Животное, Тип, Количество сведения о наименовании животного, типе его рациона (возможны два типа: П - плотоядное или Т - травоядное) и количестве пищи в день. Для получения информации о животных, съедающих в день от 5 до 10 кг мясных продуктов, запрос имеет вид:*

- а) Животное = (Тип=П или Количество=5)
- б) Тип <> Т и 10 <= Количество >=5
- в) Тип = П и (Количество >=5 и Количество <=10)
- г) Тип =П и (Количество >=5 и Количество >=10)

Билет № 10

1. Таблицы в базах данных предназначены:

- а) для автоматического выполнения группы команд
- б) для хранения данных базы;
- в) для отбора и обработки данных базы;
- г) для ввода данных базы и их просмотра;

2. Что из перечисленного не является объектом Access:

- а) модули;
- б) ключи;
- в) таблицы;
- г) формы;

3. Какой из мастеров не предусмотрен в Access?

- а) мастер запросов
- б) мастер макросов
- в) мастер таблиц
- г) мастер отчетов

4. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

5. Представлена база данных "Волшебные страны"

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46.9
БУЯН	156	47.7
ОЗ	155	95.3
ГУСТОЙ ЛЕС	99	53.5
ЗАЗЕРКАЛЬЕ	51	76.2

После проведения сортировки сведения о ГУСТОМ ЛЕСЕ переместились на две строки вверх. Сортировка проводилась в порядке:

- а) убывания по полю ПЛОЩАДЬ;
- б) возрастания по полю СТРАНА;
- в) возрастания по полю ПЛОЩАДЬ;
- г) возрастания по полю НАСЕЛЕНИЕ

6. Представлена база данных "Гостиница"

Номер	Кол_комнат	Цена
210л	27	500
101о	26	30
215д	30	80
101д	18	50
112пл	24	250

После сортировки в порядке возрастания по полю "Номер" сведения о номере ценой 500 переместятся на:

- а) 0 строк
- б) 1 строку вниз
- в) 2 строки вниз
- г) 3 строки вниз

7. Запрос к базе данных с полями Фамилия, Год_рождения, Класс, Оценка для вывода списка учеников 10 классов, 1981 года рождения, имеющих оценки 4 или 5, содержит выражение

- а) $\text{Класс} > 10$ и $\text{Оценка} = 4$ и $\text{Год_рождения} = 1981$
- б) $\text{Класс} = 10$ или $\text{Оценка} > 4$ или $\text{Год_рождения} = 1981$
- в) $\text{Оценка} \geq 4$ и $\text{Год_рождения} = 1981$ и $\text{Класс} = 10$
- г) $\text{Класс} = 10$ и $\text{Оценка} > 4$ или $\text{Год_рождения} = 1981$

Билет № 11

1. База данных - это:

- а) совокупность данных, организованных по определенным правилам;
- б) совокупность программ для хранения и обработки больших массивов информации;
- в) интерфейс, поддерживающий наполнение и манипулирование данными;
- г) определенная совокупность информации.

2. Сколько файлов достаточно для хранения всех объектов базы данных в Access?

- а) одного
- б) двух
- в) по количеству объектов
- г) по количеству типов объектов

3. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на выборку
- в) запрос на обновление,
- г) запрос на добавление

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	8
VAZ21099	10	12
Ford	22	2
UAZ	6	3

После проведения сортировки сведения об автомобиле MERSEDES500 переместились на одну строку вниз. Сортировка проводилась в порядке:

- а) возрастания по полю Цена
- б) убывания по полю Продано
- в) возрастания по полю Модель
- г) возрастания по полю Продано

6. Даны две электронные таблицы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Иванова Е.И.	ноябрь	Скорпион
Зайцева И.О.	апрель	Телец
Мальцев А.А.	декабрь	Стрелец
Малышев Е.Н.	май	Близнецы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Зайцева И.О.	апрель	Телец
Малышев Е.Н.	май	Близнецы

Вторая таблица получена из первой после удаления записей, у которых:

- а) Поле Зодиак = «Стрелец» или Первая буква (ФИО) = «И»
- б) Поле Зодиак = «Стрелец» и Первая буква (ФИО) = «И»
- в) Первая буква (Месяц) <> «а»
- г) Месяц <> «август» или Месяц <> «апрель»

7. **База данных "Зоопарк" содержит в полях Животное, Тип, Количество сведения о наименовании животного, типе его рациона** (возможны два типа: П - плотоядное или Т - травоядное) и количестве пищи в день. Для получения информации о животных, съедающих в день от 5 до 10 кг мясных продуктов, запрос имеет вид:

- а) Животное = (Тип=П или Количество=5)
- б) Тип <> Т и 10 <= Количество >=5
- в) Тип = П и (Количество >=5 и Количество <=10)
- г) Тип <> Т и 10 <= Количество <> 5

Билет № 12

1. **Что из перечисленного не является объектом Access:**

- а) модули;
- б) таблицы;
- в) макросы;
- г) ключи;

2. **Ключами поиска в системах управления базами данных (СУБД) называются:**

- а) диапазон записей файла БД, в котором осуществляется поиск;
- б) логические выражения, определяющие условия поиска;
- в) поля, по значению которых осуществляется поиск;
- г) номер первой по порядку записи, удовлетворяющей условиям поиска

3. **Каскадное удаление связанных записей предусматривает:**

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	4
BA3 21099	10	12
Ford	22	2
УАЗ	6	3

После проведения сортировки сведения об автомобиле MERSEDES 500 переместились на две строки вниз. Сортировка проводилась в порядке

- а) возрастания по полю Модель
- б) возрастания по полю Цена
- в) возрастания по полю Цена;
- г) убывания по полю Продано

6. Даны две электронные таблицы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Иванова Е.И.	ноябрь	Скорпион
Зайцева И.О.	апрель	Телец
Мальцев А.А.	декабрь	Стрелец
Малышев Е.Н.	май	Близнецы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Зайцева И.О.	апрель	Телец
Малышев Е.Н.	май	Близнецы

Вторая таблица получена из первой после удаления записей, у которых:

- а) Поле Зодиак = «Стрелец» и Первая буква (ФИО) = «И»
- б) Поле Зодиак = «Стрелец» или Первая буква (ФИО) = «И»
- в) Первая буква (Месяц) <> «а»
- г) Первая буква (ФИО) <> «М»

7. Запрос к базе данных с полями Товар, Продавец, Цена для получения информации о продавцах, продающих товар X по цене выше Y, содержит выражение

- а) Товар = X и Цена <= Y
- б) Товар = X или Цена <= Y
- в) Товар > X и Цена <= Y
- г) Цена >= Y или Товар = X

Билет № 13

1. Для чего предназначены формы:

- а) для отбора и обработки данных базы;
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для выполнения сложных программных действий

2. Ключами поиска в системах управления базами данных (СУБД) называются:

- а) диапазон записей файла БД, в котором осуществляется поиск;
- б) логические выражения, определяющие условия поиска;
- в) поля, по значению которых осуществляется поиск;
- г) номера записей, удовлетворяющих условиям поиска

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES500	27	8
VAZ21099	10	12
Ford	22	2
UAZ	6	3

После проведения сортировки сведения об автомобиле MERSEDES500 переместились на одну строку вниз. Сортировка проводилась в порядке:

- а) возрастания по полю Цена
- б) возрастания по полю Модель
- в) убывания по полю Цена
- г) возрастания по полю Продано

6. Даны две электронные таблицы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Иванова Е.И.	ноябрь	Скорпион
Зайцева И.О.	апрель	Телец
Мальцев А.А.	декабрь	Стрелец
Малышев Е.Н.	май	Близнецы

ФИО	Месяц	Зодиак
Матросов И.И.	август	Дева
Зайцева И.О.	апрель	Телец
Малышев Е.Н.	май	Близнецы

Вторая таблица получена из первой после удаления записей, у которых:

- а) Поле Зодиак = «Стрелец» и Первая буква (ФИО) = «И»
- б) Первая буква (Месяц) \in «а»
- в) Первая буква (ФИО) \in «М»
- г) Первая буква (Зодиак) \in «С»

7. Запрос к базе данных с полями Товар, Продавец, Цена для получения информации о продавцах, продающих товар X по цене выше Y, содержит выражение

- а) Продавец = (Товар = X или Цена < Y)
- б) Товар = X или Цена \leq Y
- в) Товар > X и Цена \leq Y
- г) Цена \geq Y или Товар = X

Билет № 14

1. Без каких объектов не может существовать база данных:

- а) без отчетов;
- б) без таблиц;
- в) без форм;
- г) без запросов

2. Ключами поиска в системах управления базами данных (СУБД) называются:

- а) диапазон записей файла БД, в котором осуществляется поиск;
- б) логические выражения, определяющие условия поиска;
- в) поля, по значению которых осуществляется поиск;
- г) номера записей, удовлетворяющих условиям поиска

3. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. Укажите, какой из фильтров использует бланк запроса:

- а) фильтр по выделенному
- б) обычный фильтр
- в) расширенный фильтр
- г) фильтр на исключение выделенного

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	4
BA3 21099	10	12
Ford	22	2
УАЗ	6	3

После проведения сортировки сведения об автомобиле MERSEDES 500 переместились на две строки вниз. Сортировка проводилась в порядке

- а) возрастания по полю Цена;
- б) убывания по полю Продано
- в) возрастания по полю Модель
- г) убывания по полю Цена

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания Переместившееся на максимальное расстояние значение в столбце "ВЕС" в базе данных

N	ФИО	РОСТ	ВЕС
1	Иванов	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Петров	167	66

равно:

- а) 66
- б) 67
- в) 68
- г) 73

7. Запрос к базе данных с полями Фамилия, Год_рождения, Класс, Оценка для вывода списка учеников 10 классов, 1981 года рождения, имеющих оценки 4 или 5, содержит выражение

- а) Класс > 10 и Оценка = 4 и Год_рождения = 1981
- б) Класс = 10 или Оценка > 4 или Год_рождения = 1981
- в) Оценка >= 4 и Год_рождения = 1981 и Класс = 10
- г) Оценка >= 4 и Год_рождения > 1980 или Класс = 10

Билет № 15

1. Что из перечисленного не является объектом Access:

- а) ключи;
- б) формы;
- в) отчеты;
- г) страницы

2. Ключами поиска в системах управления базами данных (СУБД) называются:

- а) диапазон записей файла БД, в котором осуществляется поиск;
- б) логические выражения, определяющие условия поиска;
- в) поля, по значению которых осуществляется поиск;
- г) номер первой по порядку записи, удовлетворяющей условиям поиска

3. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. Укажите, в каком из видов запроса результат сохраняется в виде отдельной таблицы:

- а) запрос на выборку
- б) запрос на создание таблицы
- в) запрос на обновление,
- г) запрос на добавление

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	4
BA3 21099	10	12
Ford	22	2
УАЗ	6	3

После проведения сортировки сведения об автомобиле УАЗ переместились на три строки вверх. Сортировка проводилась в порядке:

- а) возрастания по полю Цена
- б) убывания по полю Модель
- в) убывания по полю Цена
- г) возрастания по полю Продано.

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания Переместившееся на максимальное расстояние значение в столбце "ВЕС" в базе данных

N	ФИО	РОСТ	ВЕС
1	Иванов	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Петров	167	66

равно:

- а) 66
- б) 68
- в) 73
- г) 75

7. Запрос к базе данных с полями Товар, Продавец, Цена для получения информации о продавцах, продающих товар X по цене выше Y, содержит выражение

- а) Продавец = (Товар = X или Цена < Y)
- б) Товар = X и Цена <= Y
- в) Товар > X и Цена <= Y
- г) Цена >= Y или Товар = X

Билет № 16

1. Для чего предназначены формы:

- а) для хранения данных базы;
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для выполнения сложных программных действий

2. Сколько файлов достаточно для хранения всех объектов базы данных в Access?

- а) одного
- б) двух
- в) по количеству объектов
- г) по количеству типов объектов

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в столбцах;
- в) в записях;
- г) в ячейках.

5. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	4
BA321099	10	12
Ford	22	2
УАЗ	6	3

После проведения сортировки сведения об автомобиле УАЗ переместились на три строки вверх. Сортировка проводилась в порядке:

- а) возрастания по полю Цена
- б) убывания по полю Продано
- в) убывания по полю Цена
- г) возрастания по полю Продано.

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания Переместившееся на максимальное расстояние значение в столбце "ВЕС" в базе данных

N	ФИО	РОСТ	ВЕС
1	Иванов	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Петров	167	66

равно:

- а) 75
- б) 74
- в) 73
- г) 66

7. Запрос к базе данных с полями Автор, Наименование, Серия, Год_издания для получения списка книг автора X в серии "Сказки", изданных не ранее 1996 года, содержит выражение

- а) Серия = "Сказки" и Год_издания \geq 1995 или Автор = X
- б) Серия = "Сказки" и Год_издания $>$ 1995 и Автор = X
- в) Серия = "Сказки" или Год_издания $>$ 1995 и Автор = X
- г) Серия = "Сказки" и Год_издания $>$ 1995 или Автор = X

Билет № 17

1. Таблицы в базах данных предназначены:

- а) для автоматического выполнения группы команд
- б) для хранения данных базы;
- в) для отбора и обработки данных базы;
- г) для ввода данных базы и их просмотра;

2. Без каких объектов не может существовать база данных:

- а) без отчетов;
- б) без таблиц;
- в) без форм;
- г) без макросов.

3. Какой из мастеров не предусмотрен в Access?

- а) мастер запросов
- б) мастер макросов
- в) мастер таблиц
- г) мастер отчетов

4. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Представлена база данных "Чемпионат ХХХ по УУУ"

Команда	Страна	Всего_игр	Выиграно	Проиграно
Безумные медведи	Швеция	6	1	5
Пронырливые волки	Германия	7	5	2
Угрюмые слоны	Норвегия	5	1	3
Голодные лисы	Франция	5	2	2
Грозные буйволы	Испания	7	4	3

После проведения сортировки сведения о команде из Швеции переместились на две строки вниз. Сортировка проводилась в порядке:

- а) возрастания по полю Выиграно
- б) убывания по полю Всего_игр
- в) возрастания по полю Страна
- г) возрастания по полю Команда

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние переместилось значение в столбце "РОСТ" :

N	ФИО	РОСТ	ВЕС
1	Иванов	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Петров	167	66

- а) 165
- б) 167
- в) 168
- г) 171

7. Запрос к базе данных с полями Автор, Наименование, Серия, Год_издания для получения списка книг автора X в серии "Сказки", изданных не ранее 1996 года, содержит выражение

- а) Автор = (Серия = "Сказки" или Год_издания < 1995);
- б) Серия = "Сказки" и Год_издания >=1995 или Автор = X
- в) Серия = "Сказки" и Год_издания >1995 и Автор = X
- г) Серия = "Сказки" или Год_издания >1995 и Автор = X

Билет № 18

1. Что из перечисленного не является объектом Access:

- а) модули;
- б) ключи;
- в) таблицы;
- г) формы;

2. В чем состоит особенность поля "счетчик"?

- а) служит для ввода числовых данных;
- б) служит для ввода действительных чисел;
- в) имеет ограниченный размер;
- г) имеет свойство автоматического наращивания.

3. Какой из мастеров не предусмотрен в Access?

- а) мастер форм
- б) мастер подстановок
- в) мастер диаграмм
- г) мастер макросов

4. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Поле Наименование в базе данных

Автор	Серия	Наименование	Год издания	Кол. стр
Уолш Р.	Для начинающих	Windows 95	1996	128
Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
Кирсанов Д.	Для чайников	Word 7.0	1996	236
Визе М.	Компьютер для носорога	Access 2.0	1994	255

имеет тип:

- а) целочисленный;
- б) числовой с плавающей точкой;
- в) символьный;
- г) дата

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние переместилось значение в столбце "РОСТ" :

N	ФИО	РОСТ	ВЕС
1	Иванов	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Петров	167	66

- а) 184
- б) 171
- в) 168
- г) 167

7. Запросы Серия = Для чайников и Год_издания = 1996 в базе данных

N	Автор	Серия	Наименование	Год_издания	Кол_стр
1	Уолш Р.	Для начинающих	Windows 95	1996	128
2	Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
3	Кирсанов Д.	Для чайников	Word 7.0	1996	236
4	Визе М.	Компьютер для носорога	Access 2.0	1994	255

отвечают записи

- а) 2 и 3
- б) 3
- в) 4
- г) 1 и 3

Билет № 19

1. Таблицы в базах данных предназначены:

- а) для хранения данных базы;
- б) для отбора и обработки данных базы;
- в) для ввода данных базы и их просмотра;
- г) для выполнения сложных программных действий.

2. Укажите несуществующий тип связи информационных объектов:

- а) один - к - одному
- б) один - ко- многим
- в) многие - ко - многим
- г) один - ко- всем

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. Укажите, какой из фильтров использует бланк запроса:

- а) фильтр по выделенному
- б) обычный фильтр
- в) расширенный фильтр
- г) фильтр на исключение выделенного

5. Представлена база данных "Чемпионат ХХХ по УУУ"

Команда	Страна	Всего_игр	Выиграно	Проиграно
Безумные медведи	Швеция	6	1	5
Пронырливые волки	Германия	7	5	2
Угрюмые слоны	Норвегия	5	1	3
Голодные лисы	Франция	5	2	2
Грозные буйволы	Испания	7	4	3

После проведения сортировки сведения о команде из Швеции переместились на две строки вниз. Сортировка проводилась в порядке:

- а) убывания по полю Всего_игр
- б) возрастания по полю Страна
- в) убывания по полю Страна
- г) возрастания по полю Команда

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние переместилось значение в столбце "ВЕС"

№	ФИО	РОСТ	ВЕС
1	ИВАНОВ	165	67
2	ШУБИН	170	75
3	СИДОРОВ	168	73
4	ЯШИН	184	74
5	СИЗОВ	171	68
6	ПЕТРОВ	167	66

равно:

- а) 66
- б) 67
- в) 68
- г) 74

7. Запросу Серия = Для чайников и Год_издания = 1996 в базе данных

N	Автор	Серия	Наименование	Год_издания	Кол_стр
1	Уолш Р.	Для начинающих	Windows 95	1996	128
2	Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
3	Кирсанов Д.	Для чайников	Word 7.0	1996	236
4	Визе М.	Компьютер для носорога	Access 2.0	1994	255

отвечают записи

- а) 2 и 3
- б) 3
- в) 1
- г) 1 и 3

Билет № 20

1. База данных - это:

- а) совокупность данных, организованных по определенным правилам;
- б) совокупность программ для хранения и обработки больших массивов информации;
- в) интерфейс, поддерживающий наполнение и манипулирование данными;
- г) определенная совокупность информации.

2. Укажите несуществующий тип связи информационных объектов:

- а) один - к - одному
- б) один - ко- многим
- в) многие - ко - многим
- г) один - ко- всем

3. Укажите, какой из фильтров использует бланк запроса:

- а) фильтр по выделенному
- б) обычный фильтр
- в) расширенный фильтр
- г) фильтр на исключение выделенного

4. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. Поле Наименование в базе данных

Автор	Серия	Наименование	Год издания	Кол. стр
Уолш Р.	Для начинающих	Windows 95	1996	128
Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
Кирсанов Д.	Для чайников	Word 7.0	1996	236
Визе М.	Компьютер для носорога	Access 2.0	1994	255

имеет тип:

- а) числовой с плавающей точкой;
- б) символьный;
- в) дата;
- г) командный.

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние переместилось значение в столбце "ВЕС"

№	ФИО	РОСТ	ВЕС
1	ИВАНОВ	165	67
2	ШУБИН	170	75
3	СИДОРОВ	168	73
4	ЯШИН	184	74
5	СИЗОВ	171	68
6	ПЕТРОВ	167	66

равно:

- а) 75
- б) 74
- в) 73
- г) 66

7. Запрос к базе данных с полями *Фамилия, Год рождения, Класс, Оценка по информатике* для вывода списка учеников 11 классов, 1980 года рождения, имеющих оценки по информатике 4 или 5, имеет вид

- а) Класс =10 или Оценка ≥ 4 или Год рождения =1980
- б) Оценка ≥ 4 и Год рождения =1980 и Класс =11
- в) Класс =10 и Оценка ≥ 4 или Год рождения =1980
- г) Оценка ≥ 4 и Год рождения =1980 или Класс =11

Билет № 21

1. Наиболее распространенными в практике являются:

- а) распределенные базы данных;
- б) иерархические базы данных;
- в) сетевые базы данных;
- г) реляционные базы данных.

2. Для чего предназначены формы:

- а) для хранения данных базы;
- б) для отбора и обработки данных базы;
- в) для ввода данных базы и их просмотра;
- г) для выполнения сложных программных действий

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

5. По какому полю базы данных упорядочены записи

№	Фамилия	Улица	Дом	Квартира	№ телефона
1	Иванов	Советская	42	15	258-36-19
2	Петров	Пушкина	15/2	366	366-56-98
3	Сидоров	Гоголя	35	25	255-41-88
4	Кузьмин	Гафури	69	38	564-89-71

- а) фамилия;
- б) улица
- в) № телефона
- г) Квартира

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние в БД переместилось значение в столбце "ВЕС":

№	ФИО	РОСТ	ВЕС
1	Чикин	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Титов	167	66

- а) 66
- б) 67
- в) 74
- г) 75

7. Запрос к базе данных с полями Фамилия, Год рождения, Класс, Оценка по информатике для вывода списка учеников 11 классов, 1980 года рождения, имеющих оценки по информатике 4 или 5, имеет вид

- а) Класс >10 и Год рождения =1980 и Оценка =5 и Оценка =4
- б) Класс =10 или Оценка >=4 или Год рождения =1980
- в) Оценка >=4 и Год рождения =1980 и Класс =11
- г) Класс =10 и Оценка >=4 или Год рождения =1980

Билет № 22

1. Для чего предназначены запросы:

- а) для хранения данных базы;
- б) для отбора и обработки данных базы;
- в) для ввода данных базы и их просмотра;
- г) для автоматического выполнения группы команд;

2. Что из перечисленного не является объектом Access:

- а) модули;
- б) таблицы;
- в) ключи;
- г) отчеты;

3. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. Наиболее распространенными в практике являются:

- а) распределенные базы данных;
- б) иерархические базы данных;
- в) сетевые базы данных;
- г) реляционные базы данных.

5. Поле *Наименование* в базе данных

Автор	Серия	Наименование	Год издания	Кол. стр
Уолш Р.	Для начинающих	Windows 95	1996	128
Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
Кирсанов Д.	Для чайников	Word 7.0	1996	236
Визе М.	Компьютер для носорога	Access 2.0	1994	255

имеет тип:

- а) числовой с плавающей точкой;
- б) символьный;
- в) дата;
- г) командный.

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние в БД переместилось значение в столбце "ВЕС":

№	ФИО	РОСТ	ВЕС
1	Чикин	165	67
2	Алешин	170	75
3	Сидоров	168	73
4	Яшин	184	74
5	Сизов	171	68
6	Титов	167	66

- а) 74
- б) 75
- в) 67
- г) 66

7. Запрос к базе данных, содержащей поля Автор, Наименование, Ключевое_слово и Год_издания, для получения списка книг автора W на тему "Сети", изданных не ранее 1995 г., содержит выражение

- а) Ключевое_слово = "Сети" и Год_издания \geq 1995 и Автор = W
- б) Ключевое_слово = "Сети" и Год_издания $>$ 1995 и Автор = W
- в) Ключевое_слово = "Сети" или Год_издания \geq 1995 и Автор = W
- г) Ключевое_слово = "Сети" и Год_издания \geq 1995 или Автор = W

Билет № 23

1. Без каких объектов не может существовать база данных:

- а) без отчетов;
- б) без таблиц;
- в) без запросов
- г) без макросов.

2. Для чего предназначены формы:

- а) для хранения данных базы;
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для выполнения сложных программных действий

3. Сколько файлов достаточно для хранения всех объектов базы данных в Access?

- а) одного
- б) двух
- в) по количеству объектов
- г) по количеству типов объектов

4. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. По какому полю базы данных упорядочены записи

№	Фамилия	Улица	Дом	Квартира	№ телефона
1	Иванов	Советская	42	15	258-36-19
2	Петров	Пушкина	15/2	366	366-56-98
3	Сидоров	Гоголя	35	25	255-41-88
4	Кузьмин	Гафури	69	38	564-89-71

- а) Фамилия;
- б) Улица
- в) № телефона
- г) Дом

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние в БД передвинулось значение в столбце "ВЕС"

№	ФИО	РОСТ	ВЕС
1	ИВАНОВ	165	67
2	ЯКУШЕВ	170	75
3	СИДОРОВ	168	73
4	ПЕТРОВ	184	74
5	СИЗОВ	171	68
6	ШИТОВ	167	66

- а) 67
- б) 68
- в) 74
- г) 75

7. Запрос к базе данных, содержащей поля Автор, Наименование, Ключевое_слово и Год_издания, для получения списка книг автора W на тему "Сети", изданных не ранее 1995 г., содержит выражение

- а) Автор = (Ключевое_слово = "Сети" или Год_издания < 1995
- б) Ключевое_слово = "Сети" и Год_издания >= 1995 и Автор = W
- в) Ключевое_слово = "Сети" и Год_издания > 1995 и Автор = W
- г) Ключевое_слово = "Сети" и Год_издания >= 1995 или Автор = W

Билет № 24

1. Что из перечисленного не является объектом Access:

- а) ключи;
- б) формы;
- в) отчеты;
- г) страницы

2. Укажите, какой из фильтров использует бланк запроса:

- а) фильтр по выделенному
- б) обычный фильтр
- в) расширенный фильтр
- г) фильтр на исключение выделенного

3. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в строках;
- в) в столбцах;
- г) в записях

5. Записи в базе данных

Автор	Серия	Наименование	Год издания	Кол. стр
Уолш Р.	Для начинающих	Windows 95	1996	128
Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
Кирсанов Д.	Для чайников	Word 7.0	1996	236
Визе М.	Компьютер для носорога	Access 2.0	1994	255

упорядочены по полю:

- а) кол. стр
- б) год издания
- в) серия
- г) наименование

6. После сортировки таблицы по атрибуту "ФИО" в порядке возрастания на максимальное расстояние в БД передвинулось значение в столбце "ВЕС"

№	ФИО	РОСТ	ВЕС
1	ИВАНОВ	165	67
2	ЯКУШЕВ	170	75
3	СИДОРОВ	168	73
4	ПЕТРОВ	184	74
5	СИЗОВ	171	68
6	ШИТОВ	167	66

- а) 75
- б) 74
- в) 73
- г) 68

7. Представлена база данных "Расписание уроков".

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение №_урока>2 и Класс> «8а», удовлетворяют только записи:

- а) 1,2,3,4,5,6,7
- б) 3, 5
- в) 5
- г) 3,4,5

Билет № 25

1. Таблицы в базах данных предназначены:

- а) для автоматического выполнения группы команд;
- б) для отбора и обработки данных базы;
- в) для хранения данных базы;
- г) для выполнения сложных программных действий.

2. Укажите, в каком из видов запроса результат сохраняется в виде отдельной таблицы:

- а) запрос на выборку
- б) запрос на создание таблицы
- в) запрос на обновление,
- г) запрос на добавление

3. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в строках;
- в) в записях;
- г) в ячейках.

4. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

5. Записи базы данных

Фамилия	Имя	Отчество	Должность	Оклад
Иванов	Петр	Алексеевич	Вахтер	250000
Зайцев	Иван	Петрович	Директор	1200000
Романов	Вадим	Сергеевич	Нач. отдела	680000
Абдо	Тимур	Махмутович	Программист	945600

упорядочены по полю:

- а) Фамилия
- б) Должность
- в) Оклад
- г) Имя

6. *Представлена база данных "Телефонный справочник"*

Фамилия И.О.	Телефон
Иванов И.И.	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке возрастания запись, содержащая номер телефона 568-98-00, переместится на

- а) не переместится;
- б) 1 строку вверх;
- в) 1 строку вниз;
- г) 3 строки вверх.

7. *Представлена база данных "Расписание уроков".*

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение №_урока>2 и Класс> «8а», удовлетворяют только записи:

- а) 3,4,5,6
- б) 3, 5
- в) 5
- г) 3,4,5

Билет № 26

1. Для чего предназначены формы:

- а) для хранения данных базы;
- б) для отбора и обработки данных базы;
- в) для ввода данных базы и их просмотра;
- г) для автоматического выполнения группы команд;

2. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

3. Обеспечение каскадного обновления связанных полей предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. В каких элементах таблицы хранятся данные базы:

- а) в полях;
- б) в столбцах;
- в) в записях;
- г) в ячейках.

5. Записи в базе данных

Автор	Серия	Наименование	Год издания	Кол. стр
Уолш Р.	Для начинающих	Windows 95	1996	128
Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
Кирсанов Д.	Для чайников	Word 7.0	1996	236
Визе М.	Компьютер для носорога	Access 2.0	1994	255

упорядочены по полю:

- а) автор
- б) кол. стр
- в) год издания
- г) наименование

6. *Представлена база данных "Телефонный справочник"*

Фамилия И.О.	Телефон
Иванов И.И.	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке возрастания запись, содержащая номер телефона 568-98-00, переместится на

- а) 1 строку вверх;
- б) 2 строки вверх;
- в) 1 строку вниз;
- г) 3 строки вверх.

7. *Представлена база данных "Расписание уроков"*.

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение Кабинет<30 или День> «Вт», удовлетворяют только записи:

- а) 5,6,7,9
- б) 2,3,4,5,6,7,9
- в) 1,2,3,4,5,6,7,8
- г) все записи

Билет № 27

1. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без таблиц;
- в) без форм;
- г) без макросов.

2. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на выборку
- в) запрос на обновление,
- г) запрос на добавление

3. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. В каких элементах таблицы хранятся данные базы:

- а) в строках;
- б) в столбцах;
- в) в записях;
- г) в ячейках.

5. Записи базы данных

Фамилия	Имя	Отчество	Должность	Оклад
Иванов	Петр	Алексеевич	Вахтер	250000
Зайцев	Иван	Петрович	Директор	1200000
Романов	Вадим	Сергеевич	Нач. отдела	680000
Абдо	Тимур	Махмутович	Программист	945600

упорядочены по полю:

- а) Должность
- б) Оклад
- в) Отчество
- г) Имя

6. Представлена база данных "Телефонный справочник"

Фамилия И.О.	Телефон
Иванов И.И-	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов И.К.	568-98-00
Иванников П.П.	384-15-15

После проведения сортировки по полю Фамилия И.О. в порядке убывания запись, содержащая сведения о телефоне 384-15-15, переместится на:

- а) 2 строки вверх
- б) 3 строки вверх;
- в) 4 строки вверх;
- г) не переместится

7. Представлена база данных "Расписание уроков".

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение Кабинет<30 или День> «Вт», удовлетворяют только записи:

- а) 2,3,4,9
- б) 2,3,4,5,6,7,9
- в) 1,2,3,4,5,6,7,8
- г) все записи

Билет № 28

1. Без каких объектов не может существовать база данных:

- а) без таблиц;
- б) без форм;
- в) без запросов
- г) без макросов.

2. Для чего предназначены формы:

- а) для отбора и обработки данных базы;
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для выполнения сложных программных действий

3. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на выборку
- в) запрос на обновление,
- г) запрос на добавление

4. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

5. Представлена база данных "Классы школы"

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 9а классе переместятся на:

- а) 4 строки вниз
- б) 1 строку вниз
- в) 3 строки вниз
- г) 2 строки вниз

6. Представлена база данных "Волшебные страны"

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	148	46.9
ОЗ	155	95.3
ШВАМБРАНИЯ	132	53.5
ЛУКОМОРЬЕ	199	47.7
ЗАЗЕРКАЛЬЕ	211	76.2

После проведения сортировки сведения о НАРНИИ переместились на одну строку вниз. Сортировка проводилась в порядке

- а) убывания по полю ПЛОЩАДЬ;
- б) возрастания по полю ПЛОЩАДЬ;
- в) убывания по полю СТРАНА;
- г) возрастания по полю НАСЕЛЕНИЕ

7. Представлена база данных "Расписание уроков".

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение №_урока<4 или День> «Вт», удовлетворяют только записи:

- а) 2,4,6,7,8,9
- б) 1,2,4,5,6,7,8,9
- в) все записи
- г) 2,4,6,7

Билет № 29

1. Таблицы в базах данных предназначены:

- а) для выполнения сложных программных действий.
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для хранения данных базы;

2. Что из перечисленного не является объектом Access:

- а) модули;
- б) таблицы;
- в) макросы;
- г) ключи;

3. Укажите, какой из перечисленных типов запросов не является запросом действия:

- а) запрос на удаление
- б) запрос на обновление,
- в) запрос на создание таблицы
- г) запрос на добавление

4. **Обеспечение каскадного обновления связанных полей предусматривает:**

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) изменение значений в подчиненных записях при изменении значений поля связи в записи главной таблицы
- в) запрет на обновление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

5. **Представлена база данных "Классы школы"**

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 9а классе переместятся на:

- а) 0 строк
- б) 1 строку вниз
- в) 2 строки вниз
- г) 3 строки вниз

6. **Представлена база данных "Волшебные страны"**

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46.9
БУЯН	156	47.7
ОЗ	155	95.3
ГУСТОЙ ЛЕС	99	53.5
ЗАЗЕРКАЛЬЕ	51	76.2

После проведения сортировки сведения о ГУСТОМ ЛЕСЕ переместились на две строки вверх. Сортировка проводилась в порядке:

- а) возрастания по полю СТРАНА;
- б) убывания по полю ПЛОЩАДЬ;
- в) возрастания по полю ПЛОЩАДЬ;
- г) убывания по полю СТРАНА;

7. Представлена база данных "Расписание уроков".

№	День	№_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубев	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение №_урока<4 или День> «Вт», удовлетворяют только записи:

- а) 2,4,6,7,8,9
- б) 1,2,5,6,7
- в) 1,2,4,5,6,7,8,9
- г) все записи

Билет № 30

1. Таблицы в базах данных предназначены:

- а) для выполнения сложных программных действий.
- б) для ввода данных базы и их просмотра;
- в) для автоматического выполнения группы команд;
- г) для хранения данных базы;

2. Укажите, в каком из видов запроса результат сохраняется в виде отдельной таблицы:

- а) запрос на выборку
- б) запрос на создание таблицы
- в) запрос на обновление,
- г) запрос на добавление

3. Каскадное удаление связанных записей предусматривает:

- а) отказ в удалении записи в подчиненной таблице с существующим в главной таблице значением ключа связи
- б) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы
- в) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- г) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи

4. В чем состоит особенность поля "счетчик"?

- а) служит для ввода числовых данных;
- б) служит для ввода действительных чисел;
- в) имеет ограниченный размер;
- г) имеет свойство автоматического наращивания.

5. Представлена база данных "Классы школы"

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 8б классе переместятся на:

- а) 1 строку вверх
- б) 1 строку вниз
- в) 2 строки вверх
- г) 2 строки вниз

6. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES500	27	8
VAZ21099	10	12
Ford	22	2
UAZ	6	3

После проведения сортировки сведения об автомобиле MERSEDES500 переместились на одну строку вниз. Сортировка проводилась в порядке:

- а) возрастания по полю Цена
- б) убывания по полю Продано
- в) возрастания по полю Модель
- г) возрастания по полю Продано

7. Дана база данных "Фрукты":

N	Страна	Бананы	Ананасы	Авокадо
1	Боливия	60	39	39
2	Болгария	54	71	39
3	Австрия	47	51	62
4	Ботсвана	63	45	62

Укажите вариант ответа, содержащий номера записей, отвечающих следующему запросу к базе данных: (Авокадо = 62 или Бананы > 54)

- а) 3
- б) 1, 3, 4
- в) 2, 3, 4
- г) 3, 4;

Билет № 31

1. База данных - это:

- а) совокупность данных, организованных по определенным правилам;
- б) совокупность программ для хранения и обработки больших массивов информации;
- в) интерфейс, поддерживающий наполнение и манипулирование данными;
- г) определенная совокупность информации.

2. Укажите, какой из фильтров использует бланк запроса:

- а) фильтр по выделенному
- б) обычный фильтр
- в) расширенный фильтр
- г) фильтр на исключение выделенного

3. Обеспечение целостности данных НЕ предусматривает:

- а) отказ в добавлении записи в подчиненную таблицу с несуществующим в главной таблице значением ключа связи
- б) запрет на удаление записи, если не удалены связанные с ней записи в подчиненной таблице
- в) запрет на изменение значений ключа связи в записи главной таблицы, если в подчиненной таблице имеются связанные с ней записи
- г) автоматическое удаление всех связанных записей в подчиненных таблицах при удалении записи из главной таблицы

4. Какое поле можно считать уникальным?

- а) поле, значения в котором не могут повторяться;
- б) поле, которое носит уникальное имя;
- в) поле, значение которого имеют свойство наращивания
- г) поле для ввода текстовых данных

5. Представлена база данных "Классы школы"

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю "Класс" сведения о 8б классе переместятся на:

- 0 строк
- 1 строку вверх
- 1 строку вниз
- 2 строки вверх

6. Представлена база данных "Продажа автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES 500	27	4
BA3 21099	10	12
Ford	22	2
УАЗ	6	3

После проведения сортировки сведения об автомобиле MERSEDES 500 переместились на две строки вниз. Сортировка проводилась в порядке

- возрастания по полю Цена;
- убывания по полю Продано
- возрастания по полю Модель
- убывания по полю Цена

7. Дана база данных "Фрукты":

N	Страна	Бананы	Ананасы	Авокадо
1	Боливия	60	39	39
2	Болгария	54	71	39
3	Австрия	47	51	62
4	Ботсвана	63	45	62

Укажите вариант ответа, содержащий номера записей, отвечающих следующему запросу к базе данных:

(Авокадо = 39 или Ананасы > 50) и Первая_буква (Страна) = "Б"

- 1
- 1, 2
- 1, 2, 4
- 1, 3.

БЛАНК ОТВЕТОВ ДЛЯ ОБУЧАЮЩИХСЯ

для итогового контроля по МДК 02.02. Технология разработки и защиты базы данных

обучающихся в группе ____ курса _____

по специальности 230115 Программирование в компьютерных системах базовой подготовки

ФИО _____

Дата _____

Вариант _____

<i>вопрос</i>	A1	A2	A3	A4	A5	A6	A7
<i>ответ</i>							

КЛЮЧИ для итогового контроля
по МДК 02.02. Технология разработки и защиты базы данных

	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
<i>1.</i>	А	В	Б	Г	Б	Г	Б
<i>2.</i>	Г	В	В	Б	Г	В	Б
<i>3.</i>	Г	В	Б	Б	А	В	Б
<i>4.</i>	В	В	В	Г	Г	В	Б
<i>5.</i>	В	Б	Б	Б	Г	А	Б
<i>6.</i>	Б	Б	Б	В	А	Б	Б
<i>7.</i>	В	Г	Г	Б	Б	Б	Б
<i>8.</i>	Г	Г	Г	Б	А	В	Б
<i>9.</i>	А	А	Г	Б	А	Б	В
<i>10.</i>	Б	Б	Б	Г	Б	Г	В
<i>11.</i>	А	А	Б	Б	В	А	В
<i>12.</i>	Г	В	Б	В	В	Б	Г
<i>13.</i>	Б	В	Г	В	Б	Г	Г
<i>14.</i>	Б	В	Б	В	А	А	В
<i>15.</i>	А	В	Б	Б	Г	А	Г
<i>16.</i>	Б	А	Г	В	Г	Г	Б
<i>17.</i>	Б	Б	Б	В	Б	Б	В
<i>18.</i>	Б	Г	Г	Б	В	Г	Б
<i>19.</i>	А	Г	Г	В	Б	А	Б
<i>20.</i>	А	Г	В	Б	Б	Г	Б
<i>21.</i>	Г	В	Г	В	Б	Б	В
<i>22.</i>	Б	В	Б	Г	Б	В	А
<i>23.</i>	Б	Б	А	Б	Б	Г	Б
<i>24.</i>	А	В	Б	Г	В	А	Б
<i>25.</i>	В	Б	В	Г	Б	Б	Б
<i>26.</i>	В	В	Б	В	А	А	В
<i>27.</i>	Б	Б	Б	В	А	Г	В
<i>28.</i>	А	Б	Б	Г	Б	Г	Б
<i>29.</i>	Г	Г	В	Б	Б	А	В
<i>30.</i>	Г	Б	Б	Г	В	В	Б
<i>31.</i>	А	В	Г	А	Г	А	Б

5.5. Задания для самостоятельной работы обучающихся

- Заполнение сводной таблицы «Сравнительная характеристика различных СУБД»

<i>Вид СУБД</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание сравнительной таблицы «Методы создания новой БД: преимущества и недостатки»

<i>Метод создания новой базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание сравнительной таблицы «Методы создания таблицы в БД: преимущества и недостатки»

<i>Метод создания таблицы базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание сравнительной таблицы «Методы создания новой формы БД: преимущества и недостатки»

<i>Метод создания новой формы базы данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание сравнительной таблицы «Методы создания отчетов в БД: преимущества и недостатки»

<i>Метод создания отчетов в базе данных</i>	<i>Преимущества</i>	<i>Недостатки</i>

- Создание и заполнение сводной таблицы «Типы запросов в БД»

<i>Типы запросов в базе данных</i>	<i>Назначение</i>	<i>Способ создания</i>

- Создание сводной таблицы «Виды групповых операций»

<i>Виды групповых операций в базе данных</i>	<i>Назначение</i>	<i>Способ создания</i>

- Реферирование темы «Методы защиты базы данных»
- Осуществление самоконтроля выполнения работы по заданным критериям

РАЗДЕЛ 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

6.1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЮ

6.1.1. Организация занятий и контроля знаний

Преподавание МДК 02.02. Технология разработки и защиты базы данных предусматривает:

- проведение лекций;
- проведение практических занятий;
- проведение лабораторных занятий;
- выполнение домашних заданий;
- реферирование ;
- проведение контрольных работ по разделам;
- аналитический обзор литературы определенной тематики
- проведение зачетных испытаний
- самостоятельная работа студентов (изучение теоретического материала, подготовка к практическим занятиям, выполнение домашних лабораторных заданий, реферирование, составление сводной таблицы, подготовка к контрольной работе).

В рамках изучения МДК 02.02. Технология разработки и защиты базы данных необходимо предусмотреть развитие форм самостоятельной работы студентов.

Пакет базовых заданий для самостоятельной работы (индивидуальные типовые расчеты во время лабораторно-практических заданий, вопросы для подготовки к зачету, тематику контрольных работ, тематику и вопросы для подготовки рефератов) следует выдавать в начале семестра, определив предельные сроки выполнения и сдачи. Задания для самостоятельной работы желательно составлять из базовой и дополнительной частей. Организуя самостоятельную работу, необходимо постоянно обучать студентов методам такой работы.

Содержание лекции должно отвечать следующим дидактическим требованиям:

- изложение материала от простого к сложному, от известного к неизвестному;
- логичность, четкость и ясность в изложении материала;
- возможность проблемного изложения, дискуссии, диалога с целью активизации деятельности студентов;
- связь теоретических положений и выводов с практикой.

Преподаватель, читающий лекционные курсы в колледже, должен знать существующие в педагогической науке и используемые на практике варианты лекций, их дидактические и воспитывающие возможности, а также их методическое место в структуре процесса обучения.

Преподаватель должен рекомендовать студентам изучать разделы модуля путем прослушивания и конспектирования лекций и материалов практических занятий, а также путем самостоятельной работы с рекомендуемой учебной литературой.

Лекции по курсу МДК 02.02. Технология разработки и защиты базы данных целесообразно читать в аудитории, оснащённой проекционной аппаратурой для демонстрации заранее подготовленных компьютерных презентаций. Презентации должны содержать опорный материал для конспектирования: отражать логику изложения в виде иерархической структуры, содержать основные определения, табличный и графический иллюстрационный материал. Определяющим требованием к презентации является её способность привить базовые навыки отражения смысла изучаемых процессов, а также дать необходимые основы для выполнения заданий лабораторного практикума.

В начале каждой лекции и практического занятия рекомендуется кратко напомнить основные положения материала предыдущего занятия, а в конце – обобщить изложенный материал и ответить на вопросы студентов. При проведении практических занятий с разбором решений типовых задач целесообразно акцентировать внимание студентов на распространенных ошибках и пояснять причины их возникновения.

Самостоятельная работа по курсу используется:

- для проработки конспектов лекций и обязательной учебной литературы по курсу;
- при необходимости – для ознакомления с рекомендуемой литературой;
- для написания реферата, предусмотренного данной рабочей программой;
- для выполнения расчётного задания по теме ;
- для выполнения тех заданий лабораторного практикума, которые, как правило, не вызывают затруднений у студентов и потому могут быть выполнены в отсутствие преподавателя;

Выполнение контрольной работы, выполнение и защита индивидуальных типовых работ и рефератов являются необходимым условием положительной оценки промежуточной и итоговой аттестации студента по дисциплине.

Порядок подготовки и защиты индивидуальных типовых расчетов изложен в методических указаниях для студентов.

При защите индивидуальных типовых работ, выполненных во время лабораторно-практических занятий, можно использовать следующие критерии (показатели) оценки ответов:

1. полнота и конкретность ответа, его обоснованность и доказательность;
2. последовательность и логика изложения;
3. уровень культуры речи (при защите в форме собеседования);
4. при выполнении практического задания: умение правильно определить возможные методы и способы решения задачи и выбрать из них наиболее оптимальный.

По результатам защиты индивидуальных типовых расчетов рекомендуется дать общую оценку результатов как каждого студента, так и всей группы в целом, обратив особое внимание на следующие аспекты:

5. качество подготовки;
6. степень усвоения знаний;
7. положительные стороны и недостатки в работе студентов;
8. задачи и пути устранения недостатков.

Также рекомендуется давать подобную оценку по результатам защиты рефератов, выполнения контрольной работы и в конце каждого практического занятия со студентами.

При изложении материала важно помнить, что почти половина информации на лекции передается через интонацию. Учитывать тот факт, что первый кризис внимания студентов наступает на 15-20-й минутах, второй - на 30-35-й минутах.

При проведении аттестации студентов важно всегда помнить, что систематичность, объективность, аргументированность - главные принципы, на которых основаны контроль и оценка знаний студентов. Проверка, контроль и оценка знаний студента, требуют учета его индивидуального стиля в осуществлении учебной деятельности. Знание критериев оценки знаний обязательно для преподавателя и студента.

Характеристика используемых форм, методов и технологий контроля учебной работы (аттестации) студента

Порядок проведения текущего контроля и промежуточной аттестации должен проводиться в строгом соответствии с положением о проведении текущего контроля успеваемости и промежуточной аттестации студентов в колледже. Требования к итоговой аттестации, определяются требованиями к итоговой аттестации, установленными федеральными государственными образовательными стандартами среднего профессионального образования по направлению подготовки 230000 Информатика и вычислительная техника

1. Промежуточная аттестация.

Промежуточная аттестация проводится по результатам выполнения домашних заданий

2. Домашние задания.

На каждом практическом занятии студент получает домашнее задание — набор заданий.

3. Выполнение контрольной работы.

Контрольная работа выполняется на аудиторном занятии. Примерный вариант заданий для контрольной работы приведен в разделе «Рубежный контроль».

4. Выполнение и защита индивидуальных типовых работ.

Индивидуальные типовые работы выполняются студентами на аудиторных лабораторно-практических занятиях (в рамках самостоятельной работы). Защита индивидуальных типовых расчетов проводится только после правильного выполнения всех заданий. При защите индивидуальных типовых расчетов студенту задают два вопроса по теоретическим материалам соответствующего раздела дисциплины

5. Итоговая аттестация по дисциплине

Итоговой аттестацией по дисциплине является тестирование. Для его проведения имеются контрольно-оценочные средства (представлены в разделе «Итоговый контроль по дисциплине»)

6.1.2. Организация и контроль самостоятельной работы

В современный период востребованы высокий уровень знаний, академическая и социальная мобильность, профессионализм специалистов, готовность к самообразованию и самосовершенствованию. В связи с этим должны измениться подходы к планированию, организации учебно-воспитательной работы, в том числе и самостоятельной работы студентов.

Прежде всего, это касается изменения характера и содержания учебного процесса, переноса акцента на самостоятельный вид деятельности, который является не просто самоцелью, а средством достижения глубоких и прочных знаний, инструментом формирования у студентов активности и самостоятельности.

Целью методических рекомендаций является повышение эффективности учебного процесса, в том числе благодаря самостоятельной работе, в которой студент становится активным субъектом обучения, что означает:

- способность занимать в обучении активную позицию;
- готовность мобилизовать интеллектуальные и волевые усилия для достижения учебных целей;
- умение проектировать, планировать и прогнозировать учебную деятельность;
- привычку инициировать свою познавательную деятельность на основе внутренней положительной мотивации;
- осознание своих потенциальных учебных возможностей и психологическую готовность составить программу действий по саморазвитию.

Организация и контроль самостоятельной работы

Для успешного выполнения самостоятельной работы студентов необходимо планирование и контроль со стороны преподавателей.

Аудиторная самостоятельная работа выполняется студентами на лекциях, лабораторно-практических занятиях, и, следовательно, преподаватель должен заранее выстроить систему самостоятельной работы, учитывая все ее формы, цели, отбирая учебную и научную информацию и средства (методических) коммуникаций, продумывая роль студента в этом процессе и свое участие в нем.

Вопросы для самостоятельной работы студентов, указанные в рабочей программе дисциплины, предлагаются преподавателями в начале изучения дисциплины. Студенты имеют право выбирать дополнительно интересующие их темы для самостоятельной работы.

Содержание деятельности преподавателя и студента при выполнении самостоятельной работы представлено в таблице [2].

+Виды самостоятельной работы студентов

<i>Репродуктивная самостоятельная работа</i>	Самостоятельное прочтение, просмотр, конспектирование учебной литературы, прослушивание лекций, магнитофонных записей, заучивание, пересказ, запоминание, Интернет-ресурсы, повторение учебного материала и др.
<i>Познавательно-поисковая самостоятельная работа</i>	Подготовка сообщений, докладов, выступлений на семинарских и практических занятиях, подбор литературы по дисциплинарным проблемам, написание рефератов, контрольных, курсовых работ и др.
<i>Творческая самостоятельная работа</i>	Написание рефератов, научных статей, участие в научно-исследовательской работе, подготовка дипломной работы (проекта). Выполнение специальных заданий и др., участие в студенческой научной конференции.

Внеаудиторная самостоятельная работа студентов (далее самостоятельная работа) –планируемая учебная, учебно-исследовательская, научно-исследовательская деятельность студентов, осуществляемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Она включает в себя:

- подготовку к аудиторным занятиям (лекциям, практическим, семинарским, лабораторным работам и др.) и выполнение соответствующих заданий;
- самостоятельную работу над отдельными темами учебных дисциплин в соответствии с учебно-тематическими планами;
- написание рефератов, докладов, эссе;
- подготовку ко всем видам практики и выполнение предусмотренных ими заданий;
- выполнение письменных контрольных работ;
- подготовку ко всем видам контрольных испытаний, в том числе к экзаменам;
- работу в студенческих научных обществах, кружках, семинарах и др.;
- участие в работе факультативов, спецсеминаров и т.п.;
- участие в научных и научно-практических конференциях, семинарах, конгрессах и т.п.;

-другие виды деятельности, организуемой и осуществляемой колледжем.

Самостоятельная работа

Основные характеристики	Деятельность преподавателя	Деятельность студентов
Цель выполнения СР	- Объясняет цель и смысл выполнения СР; - дает развернутый или краткий инструктаж о требованиях, предъявляемых к СР и способах ее выполнения; - демонстрирует образец СР	- Понимает и принимает цель СР как личностно значимую; - знакомится с требованиями к СР
Мотивация	- Раскрывает теоретическую и практическую значимость выполнения СР, тем самым формирует у студента познавательную потребность и готовность к выполнению СР; - мотивирует студента на достижение успеха	- Формирует собственную познавательную потребность в выполнении СР; - формирует установку и принимает решение о выполнении СР
Управление	- Осуществляет управление путем целенаправленного воздействия на процесс выполнения СР; - дает общие ориентиры выполнения СР	На основе владения обобщенным приемом сам осуществляет управление СР (проектирует, планирует, рационально распределяет время и т.д.)
Контроль и коррекция выполнения СР	- Осуществляет предварительный контроль, предполагающий выявление исходного уровня готовности студента к выполнению СР; - осуществляет итоговый контроль конечного результата выполнения СР	- Осуществляет текущий операционный самоконтроль за ходом выполнения СР; - выявляет, анализирует и исправляет допущенные ошибки и вносит коррективы в работу, отслеживает ход выполнения СР; - ведет поиск оптимальных способов выполнения СР; - осуществляет рефлексивное отношение к собственной деятельности; - осуществляет итоговый самоконтроль результата СР
Оценка	- На основе сличения результата с образцом, заранее заданными критериями дает оценку СР; - выявляет типичные ошибки, подчеркивает положительные и отрицательные стороны, дает методические советы по выполнению СР, намечает дальнейшие пути выполнения СР; - устанавливает уровень и определяет качество продвижения студента и тем самым формирует у него мотивацию достижения успеха в учебной деятельности	- На основе соотнесения результата с целью дает самооценку СР, своим познавательным возможностям, способностям и качествам

Выполнение любого вида самостоятельной работы предполагает прохождение студентами следующих этапов:

- определение цели самостоятельной работы;
- конкретизация познавательной (проблемной или практической) задачи;
- самооценка готовности к самостоятельной работе по решению поставленной или выбранной задачи;
- выбор адекватного способа действий, ведущего к решению задачи (выбор путей и средств для ее решения);
- планирование (самостоятельно или с помощью преподавателя) самостоятельной работы по решению задачи;
- реализация программы выполнения самостоятельной работы.

6.2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

6.2.1. Методические рекомендации по работе с литературой

Важной составляющей самостоятельной внеаудиторной подготовки является работа с литературой ко всем видам занятий: лекционным, практическим, при подготовке к зачетам, экзаменам, тестированию, участию в научных конференциях.

Умение работать с литературой означает научиться осмысленно пользоваться источниками. Прежде чем приступить к освоению научной литературы, рекомендуется чтение учебников и учебных пособий.

Существует несколько методов работы с литературой.

Один из них –самый известный –метод повторения: прочитанный текст можно заучить наизусть. Простое повторение воздействует на память механически и поверхностно. Полученные таким путем сведения легко забываются.

Наиболее эффективный метод –метод кодирования: прочитанный текст нужно подвергнуть большей, чем простое заучивание, обработке.

Чтобы основательно обработать информацию и закодировать ее для хранения, важно произвести целый ряд мыслительных операций: прокомментировать новые данные; оценить их значение; поставить вопросы; сопоставить полученные сведения с ранее известными.

Для улучшения обработки информации очень важно устанавливать осмысленные связи, структурировать новые сведения.

Изучение научной, учебной и иной литературы требует ведения рабочих записей.

Форма записей может быть весьма разнообразной: простой или развернутый план, тезисы, цитаты, конспект.

План – первооснова, каркас какой-либо письменной работы, определяющие последовательность изложения материала.

План является наиболее краткой и потому самой доступной и распространенной формой записей содержания исходного источника информации. По существу, это перечень основных вопросов, рассматриваемых в источнике. План может быть простым и развернутым. Их отличие состоит в степени детализации содержания и, соответственно, в объеме.

Преимущество плана состоит в следующем. Во-первых, план позволяет наилучшим образом уяснить логику мысли автора, упрощает понимание главных моментов произведения.

Во-вторых, план позволяет быстро и глубоко проникнуть в сущность построения произведения и, следовательно, гораздо легче ориентироваться в его содержании.

В-третьих, план позволяет –при последующем возвращении к нему –быстрее обычного вспомнить прочитанное.

В-четвертых, с помощью плана гораздо удобнее отыскивать в источнике нужные места, факты, цитаты и т. д.

Выписки–небольшие фрагменты текста (неполные и полные предложения, отдельные абзацы, а также дословные и близкие к дословным записи об излагаемых в нем фактах), содержащие в себе квинтэссенцию содержания прочитанного.

Выписки представляют собой более сложную форму записей содержания исходного источника информации. По сути, выписки –не что иное, как цитаты, заимствованные из текста. Выписки позволяют в концентрированной форме и с максимальной точностью воспроизвести в произвольном (чаще последовательном) порядке наиболее важные мысли автора, статистические и даталогические сведения. В отдельных случаях —когда это оправданно с точки зрения продолжения работы над текстом –вполне допустимо заменять цитирование изложением, близким к дословному.

Тезисы–сжатое изложение содержания изученного материала в утвердительной (реже опровергающей) форме.

Отличие тезисов от обычных выписок состоит в следующем.

Во-первых, тезисам присуща значительно более высокая степень концентрации материала.

Во-вторых, в тезисах отмечается преобладание выводов над общими рассуждениями.

В-третьих, чаще всего тезисы записываются близко к оригинальному тексту, т. е. без использования прямого цитирования.

Исходя из сказанного, нетрудно выявить основное преимущество тезисов: они незаменимы для подготовки глубокой и всесторонней аргументации письменной работы любой сложности, а также для подготовки выступлений на защите, докладов и пр.

Аннотация – краткое изложение основного содержания исходного источника информации, дающее о нем обобщенное представление.

К написанию аннотаций прибегают в тех случаях, когда подлинная ценность и пригодность исходного источника информации исполнителю письменной работы окончательно неясна, но в то же время о нем необходимо оставить краткую запись с обобщающей характеристикой. Для указанной цели и используется аннотация.

Характерной особенностью аннотации наряду с краткостью и обобщенностью ее содержания является и то, что пишется аннотация всегда после того, как (хотя бы в предварительном порядке) завершено ознакомление с содержанием исходного источника информации. Кроме того, пишется аннотация почти исключительно своими словами и лишь в крайне редких случаях содержит в себе небольшие выдержки оригинального текста.

Резюме – краткая оценка изученного содержания исходного источника информации, полученная, прежде всего, на основе содержащихся в нем выводов.

Резюме весьма сходно по своей сути с аннотацией. Однако, в отличие от последней, текст резюме концентрирует в себе данные не из основного содержания исходного источника информации, а из его заключительной части, прежде всего выводов.

Но, как и в случае с аннотацией, резюме излагается своими словами – выдержки из оригинального текста в нем практически не встречаются.

Конспект – сложная запись содержания исходного текста, включающая в себя заимствования (цитаты) наиболее примечательных мест в сочетании с планом источника, а также сжатый анализ записанного материала и выводы по нему.

Для работы над конспектом следует: определить структуру конспектируемого материала, чему в значительной мере способствует письменное ведение плана по ходу изучения оригинального текста; в соответствии со структурой конспекта произвести отбор и последующую запись наиболее существенного содержания оригинального текста — в форме цитат или в изложении, близком к оригиналу; выполнить анализ записей и на его основе – дополнение записей собственными замечаниями, соображениями, "фактурой", заимствованной из других источников и т. п. (располагать все это следует на полях тетради для записей или на отдельных листах-вкладках); завершить формулирование и запись выводов по каждой из частей оригинального текста, а также общих выводов

Систематизация изученных источников позволяет повысить эффективность их анализа и обобщения. Итогом этой работы должна стать логически выстроенная система сведений по существу исследуемого вопроса.

Необходимо из всего материала выделить существующие точки зрения на проблему, проанализировать их, сравнить, дать им оценку.

Кстати, этой процедуре должны подвергаться и материалы из Интернета во избежание механического скачивания готовых текстов. В записях и конспектах студенту очень важно указывать названия источников, авторов, год издания. Это организует его, а главное, пригодится в последующем обучении. Безусловно, студент должен взять за правило активно работать с литературой в библиотеке не только СКСЭиП, но и в других библиотеках, используя, в том числе, их компьютерные возможности (электронная библиотека в сети Интернет).

6.2.2. Методические рекомендации по подготовке к контрольным работам, зачетам, экзаменам

Приступая к изучению МДК, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке СКСЭиП, получить в библиотеке рекомендованные учебники и учебно-методические пособия, завести новую тетрадь для конспектирования лекций и работы с первоисточниками.

Помимо учебной, научной литературы студентами должны активно использоваться хрестоматии – сборники текстов, иллюстрирующих содержание учебника, а также словари, справочники. В хрестоматиях собраны материалы, которые позволяют расширить кругозор. При подготовке к занятиям, зачетам, экзаменам следует в полной мере использовать академический курс учебника, рекомендованного преподавателем. Они дают более углубленное представление о проблемах, получивших систематическое изложение в учебнике. Работа с хрестоматией позволит студенту самостоятельно изучить документы, фрагменты источников, другие произведения, разъясняющие сущность изучаемого вопроса.

Студентам рекомендуется самостоятельно выполнять доклады, индивидуальные письменные задания и упражнения, предлагаемые при подготовке к занятиям. Работа, связанная с решением этих задач и упражнений, представляет собой вид интеллектуальной практической деятельности. Она способствует выработке умения и привычки делать что-либо правильно, а также закреплению навыков и знаний по проблеме.

Доклад – это вид самостоятельной работы студентов, заключающийся в разработке студентами темы на основе изучения литературы и развернутом публичном сообщении по данной проблеме.

Отличительными признаками доклада являются:

- передача в устной форме информации;
- публичный характер выступления;
- стилевая однородность доклада;
- четкие формулировки и сотрудничество докладчика и аудитории;
- умение в сжатой форме изложить ключевые положения исследуемого вопроса и сделать выводы.

Такая работа, в процессе которой студенту приходится сравнивать, сопоставлять, выявлять логические связи и отношения, применять методы анализа и синтеза, позволит успешно в дальнейшем подготовиться к зачетам, экзаменам и тестированию. Тестирование ориентировано в целом на проверку блоков проблем, способствует систематизации изученного материала, проверке качества его усвоения.

Серьезная и методически грамотно организованная работа по подготовке к семинарским занятиям, написанию письменных работ значительно облегчит подготовку к экзаменам и зачетам. Основными функциями экзамена, зачета являются: обучающая, оценочная и воспитательная. Экзамены и зачеты позволяют выработать ответственность, трудолюбие, принципиальность. При подготовке к зачету, экзамену студент повторяет, как правило, ранее изученный материал. В этот период сыграют большую роль правильно подготовленные заранее записи и конспекты.

Студенту останется лишь повторить пройденное, учесть, что было пропущено, восполнить пробелы при подготовке к семинарам, закрепить ранее изученный материал.

6.2.3. Методические рекомендации по работе над рефератом

Реферат – краткое изложение содержания документа или его части, научной работы, включающее основные фактические сведения и выводы, необходимые для первоначального ознакомления с источниками и определения целесообразности обращения к ним.

Современные требования к реферату – точность и объективность в передаче сведений, полнота отображения основных элементов как по содержанию, так и по форме.

Цель реферата – не только сообщить о содержании реферируемой работы, но и дать представление о вновь возникших проблемах соответствующей отрасли науки.

В учебном процессе реферат представляет собой краткое изложение в письменном виде или в форме публичного доклада содержания книги, учения, научного исследования и т.п.

Иначе говоря, это доклад на определенную тему, освещающий её вопросы на основе обзора литературы и других источников.

Рефераты в рамках учебного процесса в вузе оцениваются по следующим основным критериями:

- актуальность содержания, высокий теоретический уровень, глубина и полнота анализа фактов, явлений, проблем, относящихся к теме;
- информационная насыщенность, новизна, оригинальность изложения вопросов;
- простота и доходчивость изложения;
- структурная организованность, логичность, грамматическая правильность и стилистическая выразительность;
- убедительность, аргументированность, практическая значимость и теоретическая обоснованность предложений и выводов.

Составление списка использованной литературы.

В соответствии с требованиями, предъявляемыми к реферату, докладу, необходимо составить список литературы, использованной в работе над ним.

Основные этапы работы над рефератом

В организационном плане написание реферата – процесс, распределенный во времени по этапам. Все этапы работы могут быть сгруппированы в три основные: подготовительный, исполнительский и заключительный.

Подготовительный этап включает в себя поиски литературы по определенной теме с использованием различных библиографических источников; выбор литературы в конкретной библиотеке; определение круга справочных пособий для последующей работы по теме.

Исполнительский этап включает в себя чтение книг (других источников), ведение записей прочитанного.

Заключительный этап включает в себя обработку имеющихся материалов и написание реферата, составление списка использованной литературы.

Написание реферата.

Определен список литературы по теме реферата.

Изучена история вопроса по различным источникам, составлены выписки, справки, планы, тезисы, конспекты. Первоначальная задача данного этапа - систематизация и переработка знаний. Систематизировать полученный материал - значит привести его в определенный порядок, который соответствовал бы намеченному плану работы.

Структура реферата

Введение

Введение - это вступительная часть реферата, предваряющая текст. Оно должно содержать следующие элементы:

- а) очень краткий анализ научных, экспериментальных или практических достижений в той области, которой посвящен реферат;
- б) общий обзор опубликованных работ, рассматриваемых в реферате;
- в) цель данной работы;
- г) задачи, требующие решения.

Объем введения при объеме реферата 10-15 может составлять одну страницу.

Основная часть.

В основной части реферата студент дает письменное изложение материала по предложенному плану, используя материал из источников. В этом разделе работы формулируются основные понятия, их содержание, подходы к анализу, существующие в литературе, точки зрения на суть проблемы, ее характеристики. В соответствии с поставленной задачей делаются выводы и обобщения. Очень важно не повторять, не копировать стиль источников, а выработать свой собственный, который соответствует характеру реферируемого материала.

Заключение

Заключение подводит итог работы. Оно может включать повтор основных тезисов работы, чтобы акцентировать на них внимание читателей (слушателей), содержать общий вывод, к которому пришел автор реферата, предложения по дальнейшей научной разработке вопроса и т.п. Здесь уже никакие конкретные случаи, факты, цифры не анализируются.

Заключение по объему, как правило, должно быть меньше введения.

Список использованных источников

В строго алфавитном порядке размещаются все источники, независимо от формы и содержания: официальные материалы, монографии и энциклопедии, книги и документы, журналы, брошюры и газетные статьи.

Список использованных источников оформляется в той же последовательности, которая указана в требованиях к оформлению рефератов, курсовых, дипломных работ.

Порядок сдачи и защиты рефератов.

1. Реферат сдается на проверку преподавателю за 1-2 недели до зачетного занятия.
2. При защите реферата преподаватель учитывает:
 - качество
 - степень самостоятельности студента и проявленную инициативу
 - связность, логичность и грамотность составления
 - оформление в соответствии с требованиями ГОСТ.
3. Защита тематического реферата может проводиться на выделенном одном занятии в рамках часов учебной дисциплины или конференции или по одному реферату при изучении соответствующей темы, либо по договоренности с преподавателем.
4. Защита реферата студентом предусматривает
 - доклад по реферату не более 5-7 минут
 - ответы на вопросы оппонента.

На защите запрещено чтение текста реферата.

Требования к оформлению рефератов, курсовых, дипломных работ

Работа должна быть выполнена с помощью ПК через 1,5 интервала. Тексты работ печатают с соблюдением размеров полей: справа не менее 2 см, слева 3 см, снизу, сверху – 2 см, размер шрифта TimesNewRoman–14.

Главы и параграфы курсовой и дипломной работ (проектов) нумеруются арабскими цифрами. Рядом с номером подраздела проставляется и номер раздела, они при этом разделяются между собой точкой, например, 2.1 (первый параграф, второй раздел). Слово «раздел» можно и не писать, введение и заключение не нумеруются. Номер соответствующего раздела или подраздела ставится в начале заголовка. Каждый раздел работы должен начинаться с нового листа, а новые подразделы продолжают на той же странице, на которой закончен предыдущий подраздел. Заголовки глав печатаются

прописными буквами по центру, заголовки подразделов -строчными. Если заголовок включает несколько предложений, то их разделяют точками. Переносы слов в заголовках не допускаются. В конце заголовка точки не ставятся. Полужирный шрифт не используется.

Расстояние между заголовками и текстом должно быть в одну пустую строку. Абзацы начинаются отступами в 1,5 см.

Страницы нумеруются арабскими цифрами, нумерация страниц должна быть сквозной. Титульный лист включается в общую нумерацию, однако номер на нем не ставится. Иллюстрации и таблицы, расположенные на отдельных листах, а также все приложения включают в общую нумерацию страниц работы. Номер страницы проставляется внизу посередине.

Иллюстрации (графики, схемы, диаграммы) располагаются непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. Все иллюстрации обозначаются словом «Рисунок» и в тексте на них делаются ссылки. Иллюстрации нумеруются арабскими цифрами или двумя цифрами (напр. 2.1), где 1-я цифра указывает номер главы, 2-я – номер рисунка, но сквозной нумерацией в пределах всей работы.

Если ссылки приводятся в конце страницы, используются знаки сносок, как правило, цифры, в том месте, где заканчивается мысль автора. Например, в тексте: Речевой период, который некоторые называют синтаксической конструкцией, создается по принципу кругообразно замыкающихся и ритмически организованных частей¹.

В сноске:

¹Ефимов А.И. О мастерстве речи пропагандиста. -М., 1997. Изд-во Юрайт, с. 42.

Цифровой материал рекомендуется оформлять в виде таблиц, каждую из которых размещают после упоминания о ней. Таблица должна иметь номер (арабскими цифрами) и заголовок, написанный с заглавной буквы.

Слово «Таблица» помещается с красной строки с номером, затем ставится пробел, тире, пробел и заголовок таблицы с прописной буквы без кавычек.

Тексты желательно иллюстрировать графиками, диаграммами, рисунками. При ссылке на таблицы и рисунки указывают их полный номер.

Список использованных источников оформляется в определенной последовательности. Вначале приводятся: 1. Федеральные законы, указы Президента РФ, постановления Правительства РФ, нормативные материалы, изданные органами власти и управления различных уровней. 2. Монографии, научные сборники, журнальные статьи в алфавитном порядке, с указанием ф.и.о. авторов; названия; года издания; издательства; номеров журналов, номеров страниц начала и окончания статьи. Для научной и учебной литературы –общее число страниц.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ

Основные источники:

1. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - 2-е изд., испр. и доп. - М.: Форум: ИНФРА-М, 2009.
2. Кузин А.В., Демин В.М. Разработка баз данных в системе Microsoft Access: - учебник. – 2 изд., - М.:Форум: ИНФРА-М, 2007

Дополнительные источники:

3. Кузин А.В. Разработка базы данных в системе Microsoft Access: учебник для СПО. – М.: ФОРУМ: ИНФРА - М, 2005.
4. Дубнов П.Ю. Access 2000. Программирование баз данных: учебник для ССУЗ. – М.: Вильямс, 2000.
5. Партыка Т.Л., Попов И.И. Информационная безопасность: учеб.пособ. для ССУЗ.–М.: Форум : Инфра.М., 2004
6. Бекаревич Ю., Пушкина Н..Microsoft Access 2000. – СПб, 1999
7. Робинсон С. Microsoft Access 2000: учебный курс –СПб/Питер, 2001
8. Агальцов В.П. Базы данных: Учебное пособие. - М: Мир, 2002.

Базы данных, Интернет-ресурсы, информационно-справочные и поисковые системы:

1. Сайт: <http://www.intuit.ru/studies/courses/3439/681/info#>
2. Сайт: <http://office.microsoft.com/ru-ru/access-help>