

Федеральное агентство по образованию  
Государственное образовательное учреждение высшего профессионального образования  
*Амурский государственный университет*  
(ГОУВПО «АмГУ»)

УТВЕРЖДАЮ

Зав. кафедрой ИиУС

\_\_\_\_\_ А.В. Бушманов

« \_\_\_ » \_\_\_\_\_ 2007 г.

## УПРАВЛЕНИЕ ДАННЫМИ

### **Учебно-методический комплекс дисциплины**

для специальности

230201 – Информационные системы и технологии

Составитель: Чепак Л.В.

2007 г.

*Печатается по решению  
редакционно-издательского совета  
факультета математики  
и информатики  
Амурского государственного  
университета*

Управление данными для специальности 230201 «Информационные системы и технологии»: учебно-методический комплекс дисциплины. / Чепак Л.В. – Благовещенск. Изд-во Амурского гос. ун-та, 2007. 183 с.

©Амурский государственный университет, 2007

©Кафедра информационных и управляющих систем, 2007

## ОГЛАВЛЕНИЕ

1. Выписка из государственного образовательного стандарта высшего профессионального образования	4
2. Рабочая программа	5
3. График самостоятельной работы студентов	19
4. Методические рекомендации по проведению самостоятельной работы студентов	20
5. Перечень учебников, учебных пособий	22
6. Краткий конспект лекций	24
7. Методические рекомендации по выполнению лабораторных работ	47
8. Методические указания к выполнению практических работ	135
9. Методические указания по применению современных ИТ для преподавания учебной дисциплины	164
10. Комплекты заданий для лабораторных работ	165
11. Методические указания по организации межсессионного и экзаменационного контроля знаний студентов	173
12. Фонд тестовых и контрольных заданий для оценки качества знаний по дисциплине	174
13. Контрольные вопросы к зачету	182
14. Карта кадровой обеспеченности дисциплины	183

# 1. ВЫПИСКА ИЗ ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Направление подготовки дипломированного специалиста

654600 – Информатика и вычислительная техника

Специальность

071900 (230201) – Информационные системы и технологии

Квалификация – *инженер*

Индекс	Наименование дисциплин и их основные разделы	Всего часов
ОПД.Ф.06	Управление данными	136
	Основные понятия банков данных и знаний; информация и данные; предметная область банка данных; роль и место банков данных в информационных системах; пользователи банков данных; преимущества централизованного управления данными; база данных как информационная модель предметной области; система управления базой данных (СУБД); администратор базы данных; архитектура банка данных; инфологическое проектирование базы данных; выбор модели данных; иерархическая, сетевая и реляционная модели данных, их типы структур, основные операции и ограничения; представление структур данных в памяти ЭВМ; современные тенденции построения файловых систем; обзор промышленных СУБД; тенденции развития банков данных.	

## 2. РАБОЧАЯ ПРОГРАММА

по дисциплине "Управление данными"

для специальности 230201 " Информационные системы и технологии "

курс 3 семестр 5

Лекции 36 (час.) Зачет 5

Лабораторные работы 36 (час.)

Практические занятия 18 (час.)

самостоятельная работа 46 (час.)

Всего часов 136 час.

Составитель: доцент Чепак Л.В.

Факультет Математики и информатики

Кафедра Информационных и управляющих систем

# 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

## 1.1. Цели и задачи дисциплины

Использование баз данных и информационных систем становится неотъемлемой составляющей функционирования любого предприятия или организации. От правильного выбора инструментальных средств создания баз данных, определения модели данных, обоснования подходящей схемы построения базы данных, организации запросов к хранимым данным во многом зависит эффективность разрабатываемой системы баз данных. Все это требует осознанного применения теоретических положений и инструментальных средств разработки баз данных.

Курс "Управление данными" знакомит студентов с основами организации данных, методами их проектирования и использования баз данных.

Целью дисциплины является рассмотрение теоретических и прикладных вопросов применения современных систем управления базами данных, изучение и практическое освоение методов создания и ведения баз данных.

## 1.2. Требования к уровню освоения содержания дисциплины

В результате изучения курса студенты должны *знать*:

- современные методы и средства разработки и синтеза структур информационных моделей предметных областей;
- состав информационной модели данных;
- типы логических моделей;
- современные методики синтеза и оптимизации структур баз данных;
- основные принципы организации баз данных информационных систем, способы построения баз данных;
- этапы проектирования базы данных;
- методики оптимизации процессов обработки запросов;

- основные конструкции языков манипулирования данными;
- современные методы обеспечения целостности данных;

*уметь:*

- построить информационную модель для конкретной задачи;
- применять современную методологию на стадии технического проектирования - обследование, выбор и системное обоснование проектных решений по структуре информационных моделей и базам данных, по архитектуре банка данных и его компонентам;
- подобрать наилучшую систему управления базами данных (СУБД);
- проектировать прикладную программу.

*иметь представление:*

- о тенденциях развития банков данных, хранилищ данных ;.

### **1.3.Перечень дисциплин с указанием разделов (тем), усвоение которых студентами необходимо при изучении данной дисциплины**

Изучение данной дисциплины требует от студентов предварительного усвоения таких дисциплин как «Информатика», «Математика», «Информационные технологии» в объеме государственного образовательного стандарта высшего профессионального образования.

## **2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **2.1. Федеральный компонент**

Дисциплина «Управление данными» является дисциплиной, входящей в блок общепрофессиональных дисциплин федерального компонента для специальности 230201 "Информационные системы и технологии". Государственный стандарт – ОПД.Ф.06.

Основные понятия банков данных и знаний; информация и данные; предметная область банка данных; роль и место банков данных в информационных системах; пользователи банков данных; преимущества централизованного

управления данными; база данных как информационная модель предметной области; система управления базой данных (СУБД); администратор базы данных; архитектура банка данных; инфологическое проектирование базы данных; выбор модели данных; иерархическая, сетевая и реляционная модели данных, их типы структур, основные операции и ограничения; представление структур данных в памяти ЭВМ; современные тенденции построения файловых систем; обзор промышленных СУБД; тенденции развития банков данных.

## **2.2. Наименование тем, их содержание, объем в лекционных часах**

### **ТЕМАТИЧЕСКИЙ ПЛАН ЛЕКЦИОННЫХ ЗАНЯТИЙ**

Наименование темы	Кол-во часов
1. Основы построения банков данных	4
2. Архитектура банка данных.	2
3 Инфологическое проектирование БД	6
4. Модели данных.	4
5. Теория проектирования реляционных баз данных.	8
6. Логическое проектирование БД.	4
7. Нормализация схем отношений.	4
8. Физическое проектирование БД.	2
9. Системы управления базами данных.	2
<b>ИТОГО</b>	<b>36</b>

#### **Тема 1. Основы построения банков данных.**

Информация и данные. Предметная область банка данных. Роль и место банков данных в автоматизированных системах. Пользователи банка данных.



Основные требования к банку данных. Преимущества и недостатки централизованного управления данными. Основные компоненты банка данных. База данных как информационная модель предметной области. Система управления базой данных (СУБД). Словарь данных. Администратор базы данных. Два подхода к проектированию баз данных: подход "от реального мира" и подход "от запросов пользователей". Уровни абстрагирования при проектировании процессов обработки данных. Независимость прикладных программ от данных.

## **Тема 2. Архитектура банка данных.**

Архитектура банка данных. Интерфейсы банка данных. Взаимодействие основных компонент банка данных в режиме его эксплуатации. Этапы проектирования базы данных. Анализ и определение требований к БД; инфологическое проектирование БД; датологическое проектирование БД. Задачи инфологического, логического и физического этапов проектирования.

## **Тема 3. Инфологическое проектирование БД**

Сущность инфологического подхода к проектированию информационных систем. Модель "сущность - связь". Типы бинарных связей. Графические диаграммы объектов-связей, ER-диаграммы. Моделирование локальных представлений. Объединение моделей локальных представлений, агрегация и обобщение элементов моделей, устранение выявленных противоречий. Концептуальная инфологическая модель предметной области. Моделирование ограничений предметной области. Ссылочная целостность. Оформление результатов инфологического проектирования.

## **Тема 4. Модели данных**

Понятие модели данных. Структуры данных. Основные операции над данными. Ограничения целостности. Выбор модели данных. Иерархическая, сетевая и реляционная модели данных, их типы структур, основные операции и ограничения. Схема данных.

## **Тема 5. Теория проектирования реляционных баз данных**

Функциональные зависимости. Ключи. Многозначные зависимости. Операции реляционной алгебры. Схема отношения и схема базы данных. Системы реляционного исчисления. Функциональные зависимости и аксиомы Амстронга.

### **Тема 6. Логическое проектирование БД.**

Проектирование реляционной логической модели базы данных. Установление дополнительных логических связей. Отображение инфологической модели на реляционную модель.

### **Тема 7. Нормализация схем отношений.**

Задача нормализации схем отношений. Замыкание множества атрибутов. Покрытие множества функциональной зависимости. Свойство соединения без потерь. Свойство сохранения зависимостей. Нормальные формы: первая, вторая, третья, Бойса-Кодда, четвертая, пятая.

### **Тема 8. Физическое проектирование БД.**

Компоненты этапа физического проектирования. Проектирование формата хранимой записи. Проектирование методов доступа. Хеширование. Жизненный цикл БД.

### **Тема 9. Системы управления базами данных.**

Современные системы управления базами данных. Выбор и установка СУБД. Языки описания (DDL), манипулирования данными (DML), встроенные алгоритмические языки.

## **2.3. Практические занятия, их содержание и объем в часах.**

### **ТЕМАТИЧЕСКИЙ ПЛАН ПРАКТИЧЕСКИХ ЗАНЯТИЙ**

Наименование темы	Кол-во часов
1. Проектирование конкретной БД (изучение предметной области, определение границ предметной области, выявление информационных потребностей пользователей).	4

2. Инфологическое проектирование (определение сущностей, атрибутов, связей, выбор ключей, разработка спецификаций справочник задач, построение концептуальной инфологической модели)	6
3. Логическое проектирование (проектирование реляционной логической модели базы данных, составление матрицы частоты совместного использования сущностей на основе справочника задач, оценка объема лишнего чтения, установление дополнительных логических связей, отображение инфологической модели на реляционную модель, получение совокупности отношений реляционной модели)	4
4. Нормализация отношений (приведение совокупности отношений к 1НФ, 2НФ, 3НФ)	2
5. Физическое проектирование	2
<b>ИТОГО</b>	<b>18</b>

#### **2.4. Лабораторные занятия, их содержание и объем в часах.**

##### **ТЕМАТИЧЕСКИЙ ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ**

Наименование темы	Кол-во часов
1. Создание локальной базы данных, создание таблиц с помощью мастера и путем ввода данных.	2
2. Создание таблиц с помощью конструктора, целостность данных, создание ключей и индексов, определение типов данных. Маски ввода данных.	2
3. Схема базы данных, установление связей между таблицами.	2
4. Обработка данных в таблицах. Сортировка и фильтрация данных. Обычные и расширенные фильтры.	2
5. Создание простых запросов, псевдонимы.	2
6. Создание запросов на основе нескольких таблиц. Выборка данных с условием. Использование выражений в запросах.	4
7. Соединение таблиц. Внутреннее, рекурсивное, внешнее левое и правое соединения, соединение по отношению	2
8. Перекрестные запросы. Использование функций в запросах.	4
9. Запросы на создание, на обновление, на удаление, каскадное удаление и каскадное обновление данных.	4
10. Создание форм. Элементы управления формы. Диаграммы	4
11. Составные и связанные формы. Оформление формы. Ввод	4

данных через форму.	
12. Создание отчетов. Вычисляемые поля в отчете. Сортировка и группировка данных.	4
<b>ИТОГО</b>	<b>36</b>

### **2.7. Самостоятельная работа студентов (46 часов).**

В качестве самостоятельной работы по дисциплине «Управление данными» студенты разрабатывают концептуальную инфологическую и логическую модели, осуществляют нормализацию совокупности отношений и физическое проектирование базы данных. Проектируя базу данных, представляют реализации этапов и объясняют полученные результаты.

Темы самостоятельных работ:

1. Паспортный стол
2. Автомобильный магазин
3. Библиотека
4. Отдел сбыта и маркетинга ОАО Кондитерская фабрика "Зея"
5. Учет преступников
6. Деканат
7. Отдел налоговой полиции
8. Школа
9. Отдел кадров
10. Учет административных нарушений
11. Факультет дистанционного обучения
12. Музыкальный магазин
13. Регистрация транспортных средств
14. Детский сад
15. Отдел управления фирмы "Фармация"
16. Сведения об абитуриентах
17. Складской учет

18. Фирма по продаже компьютерного оборудования
19. Отдел аспирантуры и докторантуры
20. Поликлиника
21. Страхование
22. Станция технического обслуживания "Амур-Лада"
23. Гостиница
24. Ресторан

Отчет по самостоятельной работе должен содержать следующие основные разделы:

1. Описание предметной области.
2. Справочник задач, решаемых пользователями.
3. Концептуальная инфологическая модель.
4. Логическая модель.
5. Нормализация отношений.
6. Физическое проектирование.

В течение семестра студентами должны быть самостоятельно изучены следующие вопросы и подготовлен реферат по заданной теме:

1. Функции, архитектура распределенных БД.
2. Преимущества и недостатки распределенных БД.
3. Фундаментальный принцип, свойства распределенных БД.
4. Технология клиент-сервер.
5. Связь объектно-ориентированных СУБД с общими понятиями объектно-ориентированного подхода.
6. Объектно-ориентированные модели данных.
7. Характеристики, достоинства и недостатки объектно-ориентированных СУБД.

8. Языки программирования объектно-ориентированных СУБД.
9. Языки запросов объектно-ориентированных СУБД.
10. Манифесты БД.
11. Характеристики объектно-реляционных СУБД.
12. Достоинства и недостатки объектно-реляционных СУБД.
13. Сравнительная характеристика объектно-ориентированных и объектно-реляционных СУБД.
14. Требования, предъявляемые к интеграции СУБД в среду Web.
15. Архитектура Web-СУБД.
16. Преимущества и недостатки интеграции СУБД в Web.
17. Основные методы интеграции СУБД в среду Web.
18. Безопасность Web-СУБД.

## **2.6. Вопросы к зачету**

1. Информация и данные. Базы и банки данных. Предметная область банка данных.
2. Пользователи банков данных.
3. Роль и место банков данных в автоматизированных системах.
4. Основные компоненты банков данных.
5. Администратор базы данных.
6. Архитектура банка данных. Трехуровневая архитектура банка данных.
7. Последовательность действий СУБД при обработке запросов.
8. Этапы проектирования базы данных.
9. Инфологическое проектирование БД. Модель "Сущность - Связь".
10. Моделирование локальных представлений. Формулирование сущностей.
11. Выбор идентифицирующего атрибута для каждой сущности. Назначение сущностям описательных атрибутов.
12. Спецификация связей.
13. Объединение моделей локальных представлений. Идентичность, агрегация, обобщение, устранение выявленных противоречий.
14. Иерархическая модель системы.
15. Сетевая модель системы.
16. Реляционная модель данных.
17. Реляционная алгебра и реляционное исчисление.

18. Логическое проектирование.
19. Отображение инфологической модели на реляционную модель.
20. Нормализация отношений.
21. Физическое проектирование.
22. Жизненный цикл БД.

### **2.7. Виды контроля.**

Текущий контроль за аудиторной и самостоятельной работой обучаемых осуществляется во время проведения аудиторных занятий посредством устного опроса, проведения контрольных работ или осуществления лекции в форме диалога. Промежуточный контроль осуществляется два раза в семестр в виде анализа итоговых отчетов по инфологическому и датологическому проектированию. Итоговый контроль осуществляется после успешного прохождения студентами текущего и промежуточного контроля в виде устного или письменного экзамена при ответах экзаменуемого на два вопроса в билете и дополнительные вопросы по желанию экзаменатора.

### **2.8. Требования к знаниям студентов, предъявляемые на зачете**

Зачет принимается по двум частям: 1) лекционный материал; 2) практическая часть. На зачете надо ответить на два теоретических вопроса и ответить на вопрос по проектированию базы данных.

*Критерий оценки ответа на зачете:* зачет ставится за знание теории и содержания основных этапов проектирования баз данных; не зачет получают студенты, которые в течение семестра 1) не сдали отчет по самостоятельной работе и реферат по заданной теме; 2) не сдали лабораторные работы; 3) не ответили на теоретические зачетные вопросы, 4) не смогли ответить на вопросы по проектированию.

## **3. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ**

### **3.1. Перечень обязательной (основной) литературы**

1. Базы данных: учебник для вузов: Рек. УМО вузов. - СПб.: КОРОНА принт, 2000. - 416 с.
2. Дейт К. Введение в системы баз данных: Учебное пособие. – М.: Вильямс, 2001.- 1072 с.
3. Диго С. М. Базы данных: проектирование и использование - М.: Финансы и статистика, 2005. - 592 с.
4. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: Вильямс, 2000.- 1120 с.
5. Кренке Д.М. Теория и практика построения баз данных: Учебное пособие. – СПб.: Питер, 2005. – 786 с.
6. Кузнецов С. Д. Основы баз данных: курс лекций - М. : Интернет- Ун-т Информ. Технологий, 2005. - 485 с.
7. Марков А.С. Базы данных. Введение в теорию и методологию / А.С. Марков, К. Ю. Лисовский. - М.: Финансы и статистика, 2006. - 512 с.
8. Харитонова И. Самоучитель: Office Access 2003. – СПб., Питер, 2004. – 464 с.
9. Харитонова И., Вольман Н. Программирование в Access 2002: учебный курс. – СПб., Питер, 2002. – 480 с.
10. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений - СПб.: Корона принт. -2004. - 736 с.
11. Ульман Дж., Уидом Дж. Введение в системы баз данных. - М.: Издательство "Лори". - 2000, 374 с.
12. Федоров А.П. Базы данных для всех. - М.: Компьютер пресс, 2001.- 256 с.

### **3.2. Перечень дополнительной литературы**

1. Вейскас Д. Эффективная работа с Microsoft Access 2003: учебный курс. – СПб., Питер, 2004. – 398 с.



2.Робинсон С. Microsoft Access 2000: учебный курс. – СПб., Питер, 2001. – 476 с.

3.Хансен Г., Хансен Дж. Базы данных: разработка и управление. М.: Бинном, 1999, 704 с.

4.Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных и знаний: Учебник для вузов по специальности "АСУ" - М.: Высшая школа, 1992 г. - 367 с.

#### 4. НЕОБХОДИМОЕ ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Лекции и практические занятия проводятся в стандартной аудитории, оснащенной в соответствии с требованиями преподавания теоретических дисциплин.

Для проведения лабораторных работ необходим компьютерный класс на 12-14 посадочных рабочих мест пользователей. В классе должен быть установлена СУБД Access..

## 5. УЧЕБНО-МЕТОДИЧЕСКАЯ (ТЕХНОЛОГИЧЕСКАЯ) КАРТА ДИСЦИПЛИНЫ

Номер недели	Номер темы	Вопросы, изучаемые на лекции	Занятия (номера)		метод пособия нагляд. и используемые	Самостоятельная работа студентов		Форма контроля
			(семина.) Практич.	Лабораторные		Содержание	часы	
1	2	3	4	5	6	7	8	9
1	1	1-7	1	1	1,2,3,8 – осн. 3,4 – доп.	Описание предметной области		сб
2	1	8-14		2	1,6,7,8 – осн. 3,4 – доп.			
3	2	1-6	1	3	1,6,7,8 – осн. 3,4 – доп.			
4	3	1-3		4	1,2,3,8 – осн. 3,4 – доп.	Справочник задач, решаемых пользователями		сб
5	3	4-6	2	5	1,2,3,8 – осн. 3,4 – доп.			
6	3	7-10		6	1-3,6-8 – осн. 3,4 – доп.			
7	4	1-5	2	6	1-3,6-8 – осн. 3,4 – доп.	Концептуальная инфологическая модель		к.р.
8	4	6-7		7	1-3,6-8 – осн. 3,4 – доп.			
9	5	1-3	2	8	1-3,6-8 – осн. 3,4 – доп.			
10	5	4-5		8	1-3,6-10 – осн. 3,4 – доп.			
11	5	6	3	9	1,2,3,4 – осн. 1 – доп.	Логическая модель		
12	5	7		9	1,2,3,4 – осн. 1 – доп.			
13	6	1-2	3	10	1,2,3,4 – осн. 1 – доп.			
14	6	3		10	1,2,3,4,10-12 – осн. 1 – доп.	Нормализация отношений		сб
15	7	1-5	4	11	1,2,6 – осн. 3 – доп.	Физическое проектирование		к.р.
16	7	6		11	1,2,6 – осн. 3 – доп.			
17	8	1-5	5	12	1,4,5 – осн. 1,2 – доп.			
18	9	1-3		12	1,4,5,12 – осн. 1,2 – доп.	Реферат		

Условные обозначения:  
к.р. – контрольная работа  
сб. – собеседование

### 3. ГРАФИК САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Содержание	Объем в часах	Сроки и форма контроля
3.1. Разработка для заданной предметной области концептуальной инфологической модели, исследование информационных запросов пользователей	4 час.	Собеседование (3 неделя)
3.1.1. Исследование, анализ и описание заданной предметной области	4 час.	Собеседование (6 неделя)
3.1.2. Составление справочника задач, решаемых пользователями	15 час.	Контрольная работа (7 неделя)
3.1.3. Получение итоговой концептуально-инфологической модели «сущность-связь»		
3.2. Разработка для заданной предметной области реляционной логической и физической моделей базы данных	13 час.	Собеседование (10 неделя)
3.2.1. Установление дополнительных логических связей, отображение концептуально-инфологической модели на реляционную, приведение полученных отношений к трем нормальным формам		
3.2.2. Получение физической модели базы данных, основанной на логической модели	10 час.	Контрольная работа (13 неделя)

#### **4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

По выбранной теме студенты выполняют две контрольные работы. Первая работа связана с построением концептуальной инфологической модели. Вторая контрольная работа описывает датологическое проектирование базы данных.

Контрольная работа № 1 должна включать следующие разделы:

1. Описание предметной области
  - а) описание организации;
  - б) окружение организации;
  - в) организационная структура;
  - г) границы предметной области;
  - д) проблемы организации;
  - е) цели автоматизации.
2. Выявление информационных запросов пользователей
  - а) процессы в отделе;
  - б) основные функции рабочего места;
  - в) задачи пользователей;
  - г) требования пользователей.
3. Инфологическое проектирование
  - а) разработка спецификаций сущностей;
  - б) разработка спецификаций атрибутов;
  - в) выбор идентифицирующих атрибутов;
  - г) разработка спецификаций связей типа «сущность-сущность»;
  - д) разработка спецификаций связей типа «атрибут-атрибут»;
  - е) создание справочника задач;
  - ж) построение концептуальной инфологической модели.

Контрольная работа № 2 должна включать следующие разделы:

1. Логическое проектирование

а) составление матрицы частоты совместного использования сущностей на основе справочника задач;

б) оценка объема лишнего чтения;

в) установление дополнительных логических связей;

г) отображение инфологической модели на реляционную модель»;

д) получение совокупности отношений реляционной модели;

е) приведение полученных отношений к трем нормальным формам.

## 2. Физическое проектирование

а) проектирование базовых отношений в среде целевой СУБД;

б) реализация ограничений предметной области;

в) проектирование физического представления базы данных;

г) определение индексов.

Реферативная работа включает следующие разделы:

1. Обоснование актуальности выбранной тематики и описание целей выполнения работы.

2. Систематизация и анализ найденных в научной печати, в сети Интернет и других источниках материалов.

3. Выводы.

4. Предложения по использованию результатов работы в конкретных областях и возможные направления дальнейших исследований.

## **5. ПЕРЕЧЕНЬ УЧЕБНИКОВ, УЧЕБНЫХ ПОСОБИЙ**

### **5.1. Перечень основной литературы**

1. Базы данных: учебник для вузов - СПб.: КОРОНА принт, 2000. - 416 с.
2. Дейт К. Введение в системы баз данных: Учебное пособие. – М.: Вильямс, 2001.- 1072 с.
3. Диго С. М. Базы данных: проектирование и использование - М. : Финансы и статистика, 2005. - 592 с
4. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: Вильямс, 2000.- 1120 с.
5. Кренке Д.М. Теория и практика построения баз данных: Учебное пособие. – СПб.: Питер, 2005. – 786 с.
6. Марков А.С. Базы данных. Введение в теорию и методологию / А.С. Марков, К. Ю. Лисовский. - М.: Финансы и статистика, 2006. - 512 с.
7. Харитонова И. Самоучитель: Office Access 2003. – СПб., Питер, 2004. – 464 с.
8. Харитонова И., Вольман Н. Программирование в Access 2002: учебный курс. – СПб., Питер, 2002. – 480 с.
9. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений - СПб.: Корона принт. -2004. - 736 с.
10. Ульман Дж., Уидом Дж. Введение в системы баз данных. - М.: Издательство "Лори". - 2000, 374 с.
11. Федоров А.П. Базы данных для всех. - М.: Компьютер пресс, 2001.- 256 с.

### **5.2. Перечень дополнительной литературы**

5. Вейскас Д. Эффективная работа с Microsoft Access 2003: учебный курс. – СПб., Питер, 2004. – 398 с.

6. Робинсон С. Microsoft Access 2000: учебный курс. – СПб., Питер, 2001. – 476 с.
7. Хансен Г., Хансен Дж. Базы данных: разработка и управление. М.: Бинном, 1999, 704 с.
8. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных и знаний: Учебник для вузов по специальности "АСУ" - М.: Высшая школа, 1992 г. - 367 с.



## 6. КРАТКИЙ КОНСПЕКТ ЛЕКЦИЙ

### Лекция 1. Введение в базы и банки данных.

Под *информацией* понимают любые сведения о каком-либо событии, сущности, процессе и т. п., являющиеся объектом некоторых операций: восприятия, передачи, преобразования, хранения или использования.

*Данные* можно определить как информацию, фиксированную в определенной форме, пригодной для последующей обработки, хранения и передачи.

Соответственно двум понятиям — «информация» и «данные» — в банках данных различают два аспекта рассмотрения вопросов: инфологический и датологический.

*Банк данных* (БнД) — это система специальным образом организованных данных — баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

*База данных* (БД) — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

*Предметная область* — это область применения конкретного банка данных.

Банк данных выступает в роли специальной обеспечивающей подсистемы в составе автоматизированных систем (АС).

### Лекция 2. Основные компоненты банка данных.

Компоненты банка данных представлены на рисунке



*Пользователи* выполняют свои задачи с помощью БД: вводят новые данные, модифицируют существующие данные, удаляют ненужные.

*Приложения БД* – представляет собой одну или несколько программ, которые служат посредником между пользователем и СУБД. Приложение создает запросы, формы и отчеты, посылает данные пользователю и получает отчеты. А также, преобразует действие пользователя в запросы для управления данными с помощью СУБД.

*СУБД* – это комплекс программ, предназначенных для ведения и создания, использования БД в условиях принятой технологии обработки данных. Цель СУБД состоит в получении приложений от пользователя и преобразования их в файлы ввода – вывода БД.

Функции приложения базы данных:

- 1) Создание и обработка форм.
- 2) Создание и передача запросов.
- 3) Создание и обработка отчетов.
- 4) Выполнение логики приложений.
- 5) Управление приложением.

Функции СУБД:

- 1) Создание БД.
- 2) Создание таблиц.
- 3) Создание поддерживающих структур.
- 4) Чтение данных из БД.
- 5) Изменение данных в БД.
- 6) Поддержка структур БД.
- 7) Установка правил.
- 8) Управление параллельной обработкой.
- 9) Обеспечение безопасности.
- 10) Сохранение и извлечение резервных копий.

Преимущества СУБД: контроль за избыточностью данных; непротиворечивость данных; больше полезной информации при том же объеме хранимых данных; совместное использование данных; поддержка целостности данных; повышенная безопасность; применение стандартов; повышение эффективности с ростом масштабов системы; возможность нахождения компромисса для противоречивых требований; повышение доступности данных и готовности к работе; улучшение показателей производительности.

Недостатки СУБД: сложность, т. е. усложнение программного обеспечения; размер, стоимость, дополнительные затраты на аппаратное обеспечение, затраты на преобразование.

По степени универсальности СУБД различают на

- 1) системы общего назначения,
- 2) специализированные системы.

### **Лекция 3. Архитектура банка данных.**

Практически любой современный БД имеет архитектуру ANSI/SPARC, которая включает три уровня: внешний, концептуальный и внутренний: 1) внешний, наиболее близкий к пользователям, он связан со способами представления данных для отдельных пользователей;

2) концептуальный, он является промежуточным между внутренним и внешним;

3) внутренний, наиболее близкий к физическому хранению, связанный со способами сохранения информации на физических устройствах.

Основное назначение трехуровневой архитектуры БД является обеспечение независимости от данных, т. е. изменение на нижних уровнях ни как не влияет на верхние уровни.

Различают два вида независимости от данных:

*Логическая независимость* означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему.

*Физическая независимость* от данных означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему.

Администратор БД – это технический специалист, ответственный за реализацию решения администратора данных.

Функции администратора базы данных:

- 1) Определение концептуальной схемы.
- 2) Определение внутренней схемы.
- 3) Взаимодействие с пользователем.
- 4) Определение правил безопасности и целостности.
- 5) Определение процедур резервного копирования и восстановления.
- 6) Управление производительностью и реагирование на изменение.

#### **Лекция 4. Инфологическое проектирование БД. Модель «сущность-связь»**

Сущность инфологического этапа проектирования заключается в установлении соответствия между состояниями предметной области, его восприятием и представлением в БД.

Модель «сущность - связь» - это неформальная модель предметной области, которая используется на этапе инфологического проектирования, и позволяет моделировать основное назначение семантического представления.

Сущность – это собирательное понятие, некоторая абстракция реального существующего объекта, о котором необходимо хранить информацию в системе.

Атрибут – это поименованная характеристика сущности, которая принимает значение из некоторого множества значений.

Конструктивная элементарная связь – это средство, с помощью которого представляется отношение между сущностями. Связь бывает бинарной, тернарной и n-арной.

Классификация связей:

1 «Один к одному» – это такой тип связи между сущностями А и В, при котором данному экземпляру сущности А соответствует один и только один экземпляр сущности В и наоборот. Идентификация сущностей А и В уникальна в обоих направлениях.

2 «Один ко многим» – это такой тип связи, при котором одному экземпляру сущности А может соответствовать 0, 1 или несколько экземпляров сущности В, однако каждому экземпляру сущности В соответствует только один экземпляр сущности А.

3 «Многие ко многим» – это такой тип связи, при котором каждому экземпляру сущности А соответствует 0, 1 или несколько экземпляров сущности В и наоборот.

В зависимости от количественных характеристик различают простую и многозначную связь.

При простой, однонаправленной связи от сущности А к сущности В одному и тому же экземпляру сущности А соответствует 1 экземпляр сущности В, обратная связь не определена. Идентификация экземпляра сущности В экземпляром сущности А – уникальна.

При многозначной однонаправленной связи от сущности А к В, одному экземпляру сущности А соответствует 0, 1 или несколько экземпляров сущности В, при этом обратная связь не определена.

## **Лекция 5. Инфологическое проектирование БД. Моделирование локальных представлений**

Для удобства представления одно локальное представление должно содержать 6 – 7 сущностей. Моделирование каждого локального представления состоит из следующих этапов:

1) Формирование сущностей.

Для каждого локального представления необходимо сформировать сущности, требуемые для его описания.

## 2) Выбор идентифицирующего атрибута.

В качестве идентификатора может служить один атрибут или набор из нескольких атрибутов, причем набор значений этих атрибутов должен быть уникальным.

Значение идентифицирующего атрибута находится во взаимно однозначном соответствии с экземплярами сущности.

## 3) Назначение сущности в описательных атрибутах.

Спецификация атрибутов состоит в указании для каждого атрибута наименования и множества значений, которые он может принимать.

Вторичный ключ – это атрибут или совокупность атрибутов, который идентифицирует объект в наборе или совокупности объектов, имеющих определенное значение этих атрибутов. Вторичный ключ называется ключом поиска.

## 4) Спецификация связей.

Для определения отношений между сущностями используется спецификация связей. Для выявления связей определяются их характеристики, определяется тип связей: «сущность-сущность», «сущность – атрибут», «атрибут – атрибут». Каждая связь типа «сущность-сущность» именуется. Связи типа «сущность – атрибут» служат для описания атрибутов. Связи типа «атрибут – атрибут» - представляют собой отношение между атрибутами, которые относятся к одной и той же сущности.

## **Лекция 6. Инфологическое проектирование БД. Объединение моделей локальных представлений**

При объединении локальных представлений проектировщик может формировать конструкции, являющиеся производными по отношению к используемым локальным представлениям. Образование подобных конструкций выполняется путем введения в рассмотрение понятий более высокого уровня по отношению к понятиям, используемых в локальных представлениях.

При объединении представлений используются основные концепции:

1) Идентичность - два или более моделей идентичны, если они имеют одинаковое смысловое значение.

2) Агрегация - позволяет рассмотреть связь между элементами моделей как новый элемент.

3) Обобщение - абстракция данных, позволяющая трактовать класс различных, подобных объектов, как один обобщенный поименованный тип объекта.

### **Лекция 7. Модели данных. Иерархическая и сетевая модели**

Совокупность операторов декларативного и процедурного типа некоторого алгоритмического языка определяет *модель данных* этого языка, понимаемую как совокупность методов и средств, как некоторый формализм для определения логических структур данных и процессов, динамического изменения их состояния с целью моделирования некоторых реальных процессов или явлений.

Существует большое количество терминов, используемых для обозначения определенных типов структур данных.

*Элемент данных* — наименьшая поименованная единица данных (аналог поля в файловых системах). Элемент данных — это минимальная единица данных, к которой СУБД может адресоваться непосредственно и с помощью которой выполняется построение всех остальных структур

*Агрегат данных* — поименованная совокупность элементов данных внутри записи, которую можно рассматривать как единое целое. Имя агрегата используется для его идентификации в схеме структуры данного более высокого уровня.

*Запись* - поименованная совокупность элементов данных или элементов данных и агрегатов. Имя записи используется для идентификации типа записи в схемах типов структур более высокого уровня. Запись — это агрегат, не входящий в состав никакого другого агрегата. Запись может иметь сложную иерархическую структуру, поскольку допускается многократное применение агрегации.

*Набор* - поименованная совокупность записей, образующих двухуровневую иерархическую структуру. Каждый тип набора представляет собой отношение (связь) между двумя или несколькими типами записей. Для каждого типа набора один тип записи может быть объявлен «владельцем», тогда остальные типы записей - его «члены», т. е. различают «запись - владелец» и «запись - член» набора. Каждый экземпляр набора должен содержать один экземпляр записи, имеющий тип «запись - владелец», и может содержать любое количество экземпляров записей типа «запись - член». Основное назначение набора — представление связей между записями.

*База данных* - поименованная совокупность экземпляров записей различного типа, содержащая ссылки между записями, представленные экземплярами наборов. Описание структуры БД задается ее схемой.

Понятие «группа» одновременно соответствует и понятию «агрегат» и понятию «запись». *Группа* — это поименованная совокупность элементов данных или элементов и других групп. *Групповое отношение* используется для задания связей между группами в базе данных. Групповое отношение определяется как поименованное бинарное отношение, заданное на двух множествах экземпляров рассматриваемых групп. Тип группового отношения задается между двумя типами групп.

Динамические свойства модели данных выражаются множеством операций, которые определяют допустимые действия над некоторой реализацией базы данных для перевода ее из одного состояния в другое. Это множество операций соотносят с *языком манипулирования данными*.

По характеру способа получения результата различают навигационные и спецификационные операции. Если результат операции получается путем прохождения по связям, реализованным в структуре БД, то операции называются *навигационными*. Результат навигационной операции - это единичный объект базы данных, например экземпляр записи.



Если в операции определяются только требования к результату, но не задается способ его получения, то операции называются *спецификационными*.

Логические ограничения, которые накладываются на данные, называются *ограничениями целостности*. Ограничения используются в моделях данных для поддержания целостности данных при функционировании системы, т. е. СУБД должна обеспечивать непротиворечивость данных заданным ограничениям при переводе БД из одного состояния в другое. Большое количество разработанных СУБД связано с существованием различных моделей данных. Проектировщик банка данных, выбирая для своей системы конкретную СУБД общего назначения, прежде всего сталкивается с задачей выбора наиболее подходящей модели данных для своей прикладной области.

### **Иерархическая модель данных.**

Организация данных в СУБД иерархического типа определяется в терминах: *элемент, агрегат, запись (группа), групповое отношение, база данных*. Иерархическая база данных может хранить только такие древовидные структуры.

Корневая запись каждого дерева обязательно должна содержать ключ с уникальным значением. Ключи некорневых записей должны иметь уникальное значение только в рамках группового отношения. Поддерживается только целостность связей между владельцами и членами группового отношения (никакой потомок не может существовать без предка). Не обеспечивается автоматическое поддержание соответствия парных записей, входящих в разные иерархии.

### **Сетевая модель данных**

Сетевая модель данных определяется в тех же терминах, что и иерархическая. Она состоит из множества записей, которые могут быть владельцами или членами групповых отношений. Основное различие этих моделей состоит в том, что в сетевой модели запись может быть членом *более чем одного* группового

отношения. Как и в иерархической модели обеспечивается только поддержание целостности по ссылкам (владелец отношения - член отношения).

## **Лекция 8. Инфологическое проектирование БД. Реляционная модель данных**

*Декартово произведение:* Для заданных конечных множеств  $D_1, D_2, \dots, D_n$  (не обязательно различных) декартовым произведением  $D_1 \times D_2 \times \dots \times D_n$  называется множество произведений вида:  $d_1 * d_2 * \dots * d_n$ , где  $d_i \in D_i, i = 1, \dots, n$ .

*Отношение:* Отношением  $R$ , определенным на множествах  $D_1, D_2, \dots, D_n$  называется подмножество декартова произведения  $D_1 \times D_2 \times \dots \times D_n$ .

Множества  $D_1, D_2, \dots, D_n$  называются *доменами* отношения.

Элементы декартова произведения  $d_1 * d_2 * \dots * d_n$  называются *кортежами*

Число  $n$  определяет степень отношения, количество кортежей называется мощностью отношения

Свойства отношений:

1. Отсутствие кортежей-дубликатов.
2. Отсутствие упорядоченности кортежей.
3. Отсутствие упорядоченности атрибутов.
4. Атомарность значений атрибутов, т.е. среди значений домена не могут содержаться множества значений (отношения).

В реляционной модели данных определены два базовых требования обеспечения целостности:

- целостность ссылок
- целостность сущностей.

## **Лекция 9. Теория проектирования реляционных баз данных. Реляционная алгебра**

Атрибут  $A$  отношения  $R$  *функционально зависит* от атрибута  $B$  этого же отношения, если в каждый момент времени каждому значению атрибута  $A$  соответствует только одно значение атрибута  $B$ .

*Ключ* – атрибут или совокупность атрибутов, значения которых уникально идентифицируют объект в наборе объектов.

Формально, *реляционную алгебру* можно определить в виде множеств, которое состоит из  $\Psi = \{A, D, J, P, R, \Theta, X\}$ , где

$\Psi$  – реляционная алгебра.

$A$  – множество всех атрибутов предметной области (универсальная схема отношений).

$D$  – множество доменов.

$J$  – функция из  $A$  в  $D$  ( $A \rightarrow D$ ), определяющая разнесение атрибутов из  $A$  по доменам из  $D$

$P$  – множество схем отношений.

$R$  – множество всех экземпляров различных таблиц.

$\Theta$  – совокупность операций над доменами из  $D$ .

$X$  – операции над экземплярами таблиц.

К *основным операциям реляционной алгебры* относятся: объединение, разность, выборка, декартово произведение, проекция. К *дополнительным операциям* относятся – пересечение, соединение, деление.

## **Лекция 10. Теория проектирования реляционных баз данных. Реляционное исчисление**

*Запрос* - это операция над отношениями, приводящая к построению результирующих отношений.

Системы реляционного исчисления делятся на:

1) Системы исчисления с переменными картежами.

Система исчисления с переменными картежами: запрос определяет множество картежей таблицы  $\{t:f(t)\}$ , для которых логическое условие поиска  $f(t)$  принимает истинное значение.

2) Система исчисления с переменными на доменах.

Система исчисления с переменными на доменах. Запрос определяет следующее множество:  $\{x_1, x_2, \dots, x_n: f(x_1, x_2, \dots, x_n)\}$  – это множество значений  $x_1, x_2, \dots, x_n$  – из доменов для которых логическое условие поиска  $f(x_1, x_2, \dots, x_n)$  принимает истинное значение.

### **Лекция 11. Теория проектирования реляционных баз данных. Задачи основных этапов**

Процесс проектирования БД представляет собой сложный процесс проектирования отображений описаний проектируемой области до схемы внутренней модели БД.

*Задача инфологического этапа проектирования* состоит в получении смысловых моделей, отражающих информационное содержание концептуальной предметной области. На этом этапе происходит восприятие реальной действительности, абстрагирование, изучение, описание предметной области.

*Задача логического этапа проектирования* состоит в организации данных, выделенных на предыдущем этапе проектирования, в форме, принятой на конкретной СУБД. Т. е. требуется разработать схему концептуальной модели, и схему внешних моделей данных о предметной области, пользуясь только теми типами моделей данных, которые поддерживаются этой СУБД, т. е. все.

*Задача физического этапа проектирования* состоит в выборе рациональной структуры хранения данных, методов доступа к ним.

### **Лекция 12. Теория проектирования реляционных баз данных.**

Ключ – атрибут или совокупность атрибутов, значения которых уникально идентифицируют объект в наборе объектов.

Каждый набор объектов может иметь несколько ключей. Один из них выбирается в качестве идентифицирующего ключа, а другие являются возможными ключами. Выбор ключа – важный момент при проектировании модели данных, с одной стороны ключ должен выполнять свою главную задачу – иденти-

фикацию, с другой стороны, должен включать в свой состав минимальное число атрибутов. Если несколько атрибутов, то ключ называется составным.

Если даны два атрибута  $X$  и  $Y$  некоторого отношения, то говорят, что  $Y$  функционально зависит от  $X$ , если в любой момент времени каждому значению  $X$  соответствует ровно одно значение  $Y$ .

Неключевой атрибут функционально полно зависит от составного ключа если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

В отношении  $R(A, B, C)$  существует многозначная зависимость  $R.A \twoheadrightarrow R.B$  в том и только в том случае, если множество значений  $B$ , соответствующее паре значений  $A$  и  $C$ , зависит только от  $A$  и не зависит от  $C$ .

Множество всех функциональных зависимостей, которые могут быть выведены из заданного множества функциональных зависимостей  $X$ , называется замыканием  $X$  и записывается как  $X^*$ . Для успешной работы, безусловно, необходимо определить ряд правил, позволяющих вычислить  $X^*$  из  $X$ . Набор правил вывода, называемый аксиомами Армстронга, показывает способы вывода новых функциональных зависимостей из заданных. Предположим, что  $A, B$  и  $C$  — подмножества атрибутов отношения  $R$ .

Аксиомы Армстронга:

1. Рефлексивность. Если  $B$  - подмножество  $A$ , то  $A \rightarrow B$ .
2. Дополнение. Если  $A \rightarrow B$ , то  $A, C \rightarrow B, C$ .
3. Транзитивность. Если  $A \rightarrow B$  и  $B \rightarrow C$ , то  $A \rightarrow C$ .

Следует отметить, что каждое из этих трех правил можно обосновать исходя непосредственно из определения понятия функциональной зависимости.

Этот набор правил является полным; это означает, что если задано множество  $X$  функциональных зависимостей, то все функциональные зависимости, производные от  $X$ , можно вывести из  $X$  с помощью только этих правил. Такие правила являются также непротиворечивыми, поскольку они не позволяют вывести какие-либо дополнительные функциональные зависимости, которые не следовали бы из  $X$ . Иными словами, эти правила могут применяться для получения замыкания  $X^*$ .

На основе трех правил, приведенных выше, можно вывести несколько дополнительных правил, позволяющих упростить практическую задачу вычисления  $X^*$ . Допустим, что  $D$  — еще одно подмножество атрибутов отношения  $R$ , и сформулируем следующие правила.

4. Самоопределение.  $A \rightarrow A$ .
5. Декомпозиция. Если  $A \rightarrow B, C$ , то  $A \rightarrow B$  и  $A \rightarrow C$ .
6. Объединение. Если  $A \rightarrow B$  и  $A \rightarrow C$ , то  $A \rightarrow B, C$ .
7. Композиция. Если  $A \rightarrow B$  и  $C \rightarrow D$ , то  $A, C \rightarrow B, D$ .

Правило 1 (рефлексивность) и правило 4 (самоопределение) указывают, что множество атрибутов всегда определяет любое из своих подмножеств или само себя. Поскольку с помощью этих правил вырабатываются функциональные зависимости, которые всегда справедливы, они являются тривиальными. Правило 2 (дополнение) указывает, что добавление одного и того же множества атрибутов и к левой, и к правой частям зависимости приводит к получению еще одной действительной зависимости. Правило 3 (транзитивность) указывает, что функциональные зависимости являются транзитивными. Правило 5 (декомпозиция) определяет, что можно удалять атрибуты из правой части зависимости. Повторное применение этого правила позволяет разложить функциональную зависимость  $A \rightarrow B, C, D$  на ряд функциональных зависимостей  $A \rightarrow B$ ,  $A \rightarrow C$  и  $A \rightarrow D$ . Правило 6 (объединение) указывает, что в процессе проектирования мо-

жет быть выполнена обратная операция, при которой ряд зависимостей  $A \rightarrow B$ ,  $A \rightarrow C$  и  $A \rightarrow D$  объединяется в одну функциональную зависимость  $A \rightarrow B, C, D$ . Правило 7 (композиция) является более общим, чем правило 6, и указывает, что для получения еще одной действительной зависимости может объединяться ряд неперекрывающихся зависимостей.

Все действия по определению набора функциональных зависимостей  $F$  для отношения, как правило, начинаются с выявления зависимостей, которые можно определить исходя из семантики атрибутов отношения. Затем применяются аксиомы Армстронга (правила 1-3) для вывода дополнительных функциональных зависимостей, которые также являются справедливыми для этого отношения. Систематический способ определения таких дополнительных функциональных зависимостей состоит в том, что вначале определяется каждое множество атрибутов  $A$ , которое присутствует в левой части некоторых функциональных зависимостей, а затем определяется множество всех атрибутов, зависящих от  $A$ . Поэтому для каждого множества атрибутов  $A$  можно определить множество  $A^*$  атрибутов, функционально определяемых из  $A$  на основе  $F$  ( $A^*$  называется замыканием  $A$  в соответствии с  $F$ )

### **Лекция 13. Логическое проектирование БД. Установление дополнительных логических связей**

Под *логической моделью БД* будем понимать представление о предметной области в виде связей между ними, преобразуя для эффективной реализации в среде конкретной СУБД.

Во-первых, исходным материалом для логического проектирования служит концептуально-инфологическая модель.

Во вторых, целью логического проектирования является построение такой модели БД, которая будет СУБД ориентированной, и с точки зрения выбранного критерия, считаться эффективной.

Перечень основных действий проектировщика:

1 Отображение исходной концептуально-инфологической модели на конкретную логическую модель.

2 Преобразование модели с учетом ограничений конкретной СУБД.

3 Преобразование СУБД, ориентированной модели с целью улучшения ее эксплуатационных характеристик и получение эффективной логической модели.

Реляционно-логическая модель представляет собой совокупность нормализованных отношений, в которых реализованы связи между объектами предметной области и выполнены все преобразования, необходимые в среде организации конкретной СУБД.

Для получения реляционно-логической модели необходимо:

1) Выполнить анализ исходной концептуально-инфологической модели для установления дополнительных логических связей.

2) Провести отображение, полученной на предыдущем этапе модели на реляционную модель путем совместного представления ее отношений в ключевых элементах взаимосвязанной сущности.

3) Выполнить анализ полученных отношений с точки зрения соответствия их третьей нормальной форме.

*Логической связью* называется связь, устанавливаемая между экземплярами сущностей с помощью указателей (ключей).

*Дополнительной* считается связь, установленная на этапе логического проектирования, на основе расчета количественного значения частоты совместного использования сущностей.

Правило установления дополнительных логических связей: Между сущностями исходной инфологической модели устанавливаются дополнительные логические связи если:

1. Частота использования этих сущностей высока



2. Непосредственно связи эти сущности не имеют, а выборка данных согласно имеющимся связям требует лишнего чтения

#### **Лекция 14. Логическое проектирование БД. Отображение концептуальной инфологической модели на реляционную модель**

Для того чтобы провести отображение, полученной на предыдущем этапе модели на реляционную модель используют совместное представление в ее отношениях ключевых элементах взаимосвязанных сущностей.

Общее правило: Ключ порожденной сущности добавляется в исходную.

Правила:

1. Если между сущностями модели существует простая однонаправленная или сложная однонаправленная связь, то порожденной является сущность, к которой она направлена.

2. Если между сущностями модели существует связь один к одному, то для определения исходной сущности необходимо знать частоту использования связей. Исходной будет считаться та сущность, которая будет иметь связь с большей частотой. Если частота использования связи одинакова или неизвестна, то выбор исходной сущности произволен.

3. Если между сущностями существует связь один ко многим, то исходной будет та сущность, от которой исходит простая связь.

4. Если между сущностями существует связь многие ко многим, то создается промежуточная сущность, в которую помещаются ключи взаимосвязанных сущностей и устанавливается связь один ко многим между тремя сущностями.

#### **Лекция 15. Логическое проектирование БД. Нормализация отношений. 1НФ, 2НФ, 3НФ**

В процессе нормализации полученных на предыдущем шаге отношений выполняют анализ реляционных отношений с точки зрения соответствия их третьей нормальной форме.

*Нормализация отношений* – это формальный аппарат ограничений на формирование отношений, который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых данных и уменьшает затраты на ведение БД.

Определение первой нормальной формы: отношение находится в 1НФ, если значения всех его атрибутов атомарны.

Определение второй нормальной формы: Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Определение третьей нормальной формы: Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

#### **Лекция 16. Логическое проектирование БД. Нормализация отношений. НФБК, 4НФ, 5НФ**

Нормальная форма Бойса-Кодда вводит дополнительное ограничение по сравнению с 3НФ.

*Определение нормальной формы Бойса-Кодда:* Отношение находится в НФБК, если оно находится во 3НФ и в ней отсутствуют зависимости атрибутов первичного ключа от неключевых атрибутов.

Ситуация, когда отношение будет находиться в 3НФ, но не в НФБК, возникает при условии, что отношение имеет два (или более) возможных ключа, которые являются составными и имеют общий атрибут. Заметим, что на практике такая ситуация встречается достаточно редко, для всех прочих отношений 3НФ и НФБК эквивалентны.

*Четвертая нормальная форма* касается отношений, в которых имеются повторяющиеся наборы данных. Декомпозиция, основанная на функциональных зависимостях, не приводит к исключению такой избыточности. В этом случае используют декомпозицию, основанную на *многозначных зависимостях*.

*Многозначная зависимость* является обобщением функциональной зависимости и рассматривает соответствия между множествами значений атрибутов.

В отношении  $R(A, B, C)$  существует многозначная зависимость  $R.A \twoheadrightarrow R.B$  в том и только в том случае, если множество значений  $B$ , соответствующее паре значений  $A$  и  $C$ , зависит только от  $A$  и не зависит от  $C$ .

Каждая функциональная зависимость является многозначной, но не каждая многозначная зависимость является функциональной.

*Определение четвертой нормальной формы:* Отношение находится в 4НФ если оно находится в НФБК и в нем отсутствуют многозначные зависимости, не являющиеся функциональными зависимостями.

До сих пор предполагалось, что единственной операцией, необходимой для устранения избыточности в отношении, была декомпозиция его на две проекции. Однако существуют отношения, для которых нельзя выполнить декомпозицию без потерь на две проекции, но которые можно подвергнуть декомпозиции без потерь на три (или более) проекций. Этот факт получил название *зависимости по соединению*, а такие отношения называют *3-декомпозируемые отношения* (ясно, что любое отношение можно назвать " $n$ -декомпозируемым", где  $n \geq 2$ ).

*Определение пятой нормальной формы:* Отношение находится в 5НФ тогда и только тогда, когда любая зависимость по соединению в нем определяется только его возможными ключами. Другими словами, каждая проекция такого отношения содержит не менее одного возможного ключа и не менее одного неключевого атрибута.

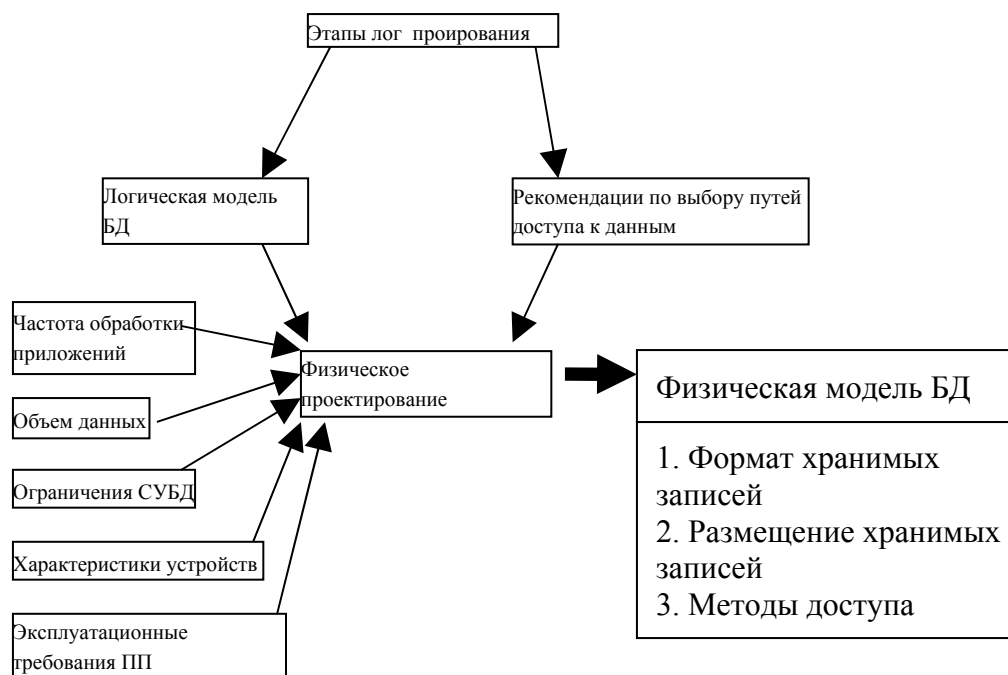
## **Лекция 17. Физическое проектирование БД.**

Физическая организация данных оказывает основное влияние на эксплуатационные характеристики проектируемой БД, т.к. именно на этом этапе осуще-

ствляется отображение логической модели БД на физическую среду хранения данных.

*Физическая БД* представляет собой совокупность совместно хранимых на физических носителях взаимосвязанных данных, состоящую из одного или нескольких типов хранимых записей.

*Проектирование физической модели БД* - это процесс создания эффективно реализуемой структуры физической БД по заданной логической модели. Проектировщик физической модели БД должен хорошо знать особенности СУБД, характеристики технических устройств и операционной системы, эксплуатационные требования прикладных программ.



Метод доступа обеспечивает возможность записи и выборки данных. Можно выделить два основных компонента доступа: структура памяти (так как функционирует под ОС, не рассматривается) и механизм поиска.

Методы вычисления адреса делятся на две группы: методы, в которых адресная процедура реализует взаимно однозначное соответствие адресов и ключей; методы перемешивания (хеширования), в которых адресная процедура реа-

лизует только однозначное преобразование ключа в адрес. Обратное преобразование обычно не имеет места.

Выделяют статические и динамические хеш-функции. Для построения статических хеш-функций используют: метод квадратов; метод, основанный на делении; метод, основанный на умножении. Динамическое хеширование разбивает или сливает вместе блоки при расширении БД или ее сокращении.

Жизненный цикл БД состоит из следующих этапов:

*Предпроектная стадия:* заключение договора о разработке базы данных; предпроектное обследование предметной области.

*Стадия проектирования:* проектирование концептуальной схемы; приведение концептуальной схемы к каноническому виду; проектирование логической модели базы данных; проектирование физической модели базы данных.

*Подготовка описаний и данных:* описание модели базы данных средствами СУБД; подготовка данных к загрузке в базу.

*Стадия реализации:* генерация описаний; ввод и загрузка данных в базу.

*Стадия эксплуатации:* анализ функционирования, поддержка; адаптация базы к изменению условий функционирования.

## **Лекция 18. Системы управления базами данных.**

С развитием компьютерных сетей персональные СУБД стали переходить в разряд многопользовательских. При этом файлы данных размещались на разделяемом сетевом диске. Однако, создание достаточно больших приложений (10-20 одновременно работающих пользователей) показало, что в этом случае резко снижается производительность и возникают проблемы с поддержанием целостности. Поэтому, в настоящее время практически все персональные СУБД дополнены средствами доступа к SQL-серверам (как правило, с использованием ODBC или JDBC). Они могут служить не только средством для создания небольших локальных приложений, но и для разработки клиентских рабочих мест в архитектуре "клиент-сервер".

Любая база данных годна к использованию только тогда, когда ее состояние соответствует состоянию предметной области. *Транзакция* - это последовательность операций, которые должны быть или все выполнены или все не выполнены. Методом контроля за транзакциями является ведение *журнала*, в котором фиксируются все изменения, совершаемые транзакцией в БД. Если во время обработки транзакции происходит сбой, транзакция откатывается - из журнала восстанавливается состояние БД на момент начала транзакции.

Язык SQL имеет общесистемный характер, так как является языком манипулирования и описания данных, выступает как стандартное средство взаимодействия серверов в распределенной системе, используется практически во всех СУБД с архитектурой «клиент-сервер».

В SQL определены два подмножества языка:

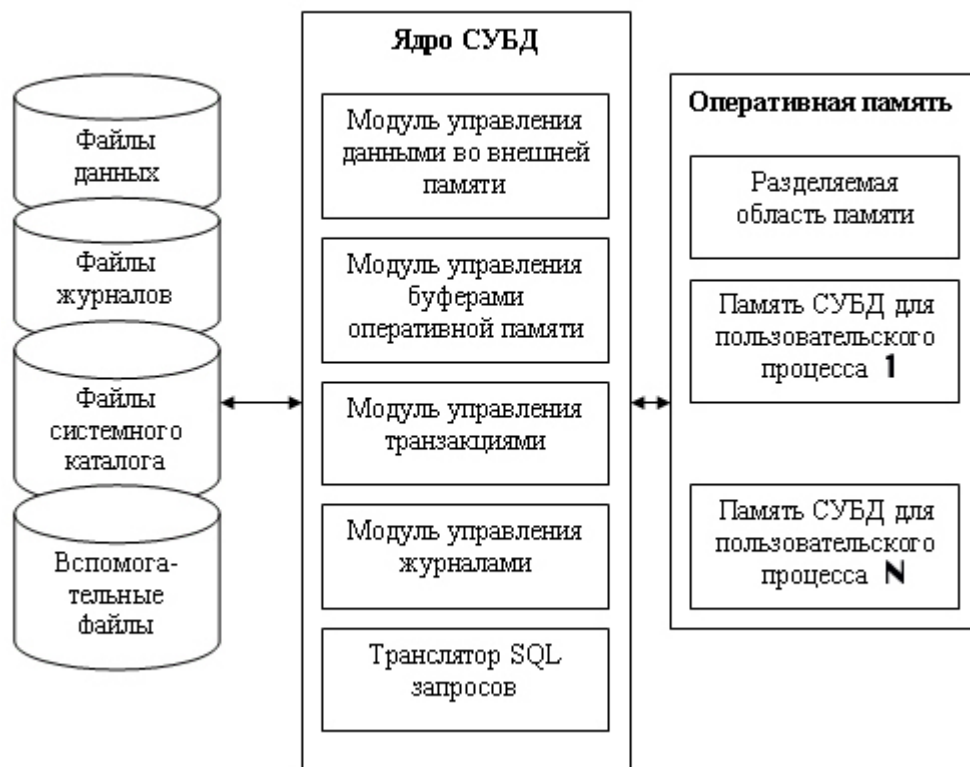
-SQL-DDL (Data Definition Language) - язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.

-SQL-DML (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

Для разработки приложений пользователи могут применять не только инструментальные средства, поставляемые вместе с сервером БД, но и средства сторонних поставщиков. Так, в нашей стране получила большую популярность инструментальная среда Delphi, которая позволяет разрабатывать приложения для различных серверов БД. За рубежом более популярными являются инструментальные системы быстрой разработки приложений (RAF Rapid Application Foundation) продукты компании Advanced Information System, инструментальной среды Power Builder фирмы Power Soft, системы SQL Windows фирмы Gtipta (Taxedo).

На рисунке изображена обобщенная структура СУБД. СУБД должна управлять внешней памятью, в котором расположены файлы с данными, файлы журналов и файлы системного каталога.

С другой стороны, СУБД управляет и оперативной памятью, в которой располагаются буфера с данными, буфера журналов, данные системного каталога, которые необходимы для поддержки целостности и проверки привилегии пользователей. Кроме того, и оперативной памяти во время работы СУБД располагается информация, которая соответствует текущему состоянию обработки запросов, там хранятся планы выполнения скомпилированных запросов и т. д.



## **7. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

Лабораторный практикум, состоящий из 12 лабораторных работ затрагивает основные разделы дисциплины «Базы данных», позволяет студентам получить достаточно полное представление об основных элементах, структуре баз данных и роли управления данными, а также приобрести практические навыки, необходимые для решения ряда практических задач проектирования и разработки баз данных.

Лабораторные работы имеют различный уровень сложности и выполняются два и четыре часа. Каждая предполагает самостоятельную работу студентов по освоению лекций. Текущий контроль знаний осуществляется путем опроса студентов после выполнения работы по вопросам, перечень которых приведен в каждой лабораторной работе.

### **ЛАБОРАТОРНАЯ РАБОТА № 1 (2 часа)**

**Тема.** Создание таблиц и баз данных

Цель лабораторной работы заключается в том, чтобы научиться:

- открывать базу данных
- создавать новую базу данных
- работать с панелями инструментов
- создавать таблицы при помощи Мастера таблиц
- использовать справочную систему Access 2003

#### **Открытие базы данных**

Открытие новой базы данных влечет за собой автоматическое закрытие уже открытой.

Чтобы открыть базу данных:

1. Нажмите кнопку «Открыть» на панели инструментов или выберите команду «Файл, Открыть». Появится диалоговое окно «Открытие файла базы данных», где следует выбрать нужный вам файл базы данных.
2. Выбрать требуемый файл в меню «Файл» из списка ранее открывавшихся файлов.

#### **Создание новой базы данных.**

**1 способ:** запустите Access 2003.



1. В диалоговом окне «Microsoft Access» выберите переключатель «Новая база данных» и нажмите кнопку ОК.

2. Появится диалоговое окно «Файл новой базы данных». По умолчанию, Access 2003 присваивает новой базе данных имя db1, а если база с таким именем уже существует, то db2 и т.д.

3. В раскрывающемся списке «Папка» откройте папку, в которой будет храниться создаваемая база данных. В поле ввода «Имя файла» укажите имя новой базы данных и не добавляйте расширение- Access 2003 автоматически добавит расширение .mdb к имени файла.

4. Нажмите кнопку «Создать» или клавишу Enter для создания базы данных. Если в этот момент была открыта другая база данных, то Access 2003 закроет ее и начнет создавать новую. После завершения процесса создания в главном окне Access 2003 появится окно новой базы данных.

**2 способ:** если Access 2003 уже был запущен:

1. В его главном окне нажмите кнопку «Создать» на панели инструментов или выберите команду «Файл, Создать». Появится диалоговое окно «Создание». Раскройте вкладку «Общие», выделите элемент «База данных» и нажмите кнопку ОК. Продолжайте создание с шага 2 предыдущего способа.

2. Вкладка «База данных» позволяет использовать один из 10 шаблонов, на основе которых могут быть созданы наиболее популярные базы данных.

**3 способ:** запустите Access 2003.

При запуске Access 2003 в диалоговом окне «Microsoft Access» выберите переключатель «Мастера, страницы и проекты базы данных» и нажмите кнопку ОК. Появится диалоговое окно «Создание» с раскрытой вкладкой «Базы данных». Выберите один из шаблонов и нажмите кнопку ОК, а дальше пошагово выполняйте все инструкции для создания типичной базы данных.

### **Работа с панелями инструментов Access 2003**

В Access 2003 команды «Вид, Панели инструментов» делают возможным выбрать текущую видимую панель инструментов. Эти команды показывают список тех панелей инструментов, которые доступны в текущем режиме работы.

Элемент «Настройка» в меню «Вид, Панели инструментов» открывает диалоговое окно «Настройка», которое позволяет вам отображать панели инструментов или скрывать их. Чтобы показать или убрать панель инструмента щелкните по вкладке «Панели инструментов» и поставьте (снимите) флажок в окошке слева от имени. Для получения справки нажмите на кнопку «Помощник» в левом нижнем углу диалогового окна «Настройка».

Диалоговое окно «Настройка» можно также использовать для изменения параметров отображения панелей инструментов. Щелкните по вкладке «Параметры». Эта страница дает возможность выбрать различные параметры отображения панелей инструментов. Можно выбрать параметр «Крупные значки», ко-

торый вызывает изображение кнопок панели инструментов примерно в 2 раза крупнее, что делает их лучше для зрительного восприятия и проще для нажатия мышью.

Параметр «Отображать подсказки для кнопок» на панели инструментов управляет появлением всплывающих подсказок.

От установки опций «Включить в подсказки сочетания клавиш» зависит, показывает ли Access горячие клавиши (если они есть) как часть текста всплывающих подсказок.

Раскрывающийся список «Эффект при выводе меню» позволяет выбрать режим рисования меню. Можно выбрать «Нет» для отсутствия спецэффектов, «Случайный выбор» для случайного выбора анимационного эффекта при каждом новом открытии меню, «Развертывание» (меню разворачивается как веер), «Соскальзывание» (открывается, как катящаяся тень).

Можно добавить или удалить кнопки из панели инструментов:

1. Выберите вкладку «Команды» и в списке «Категории» выберите «Элементы управления». В списке «Команды» появятся дополнительные кнопки.
2. Поместите указатель мыши на кнопку «Диаграмма», нажмите левую кнопку мыши и не отпуская ее перетащите кнопку на панель инструментов.

Диалоговое окно «Настройка» имеет следующие дополнительные возможности:

1. Для удаления кнопок с панели инструментов надо открыть диалоговое окно «Настройка» и перетащить ненужные кнопки в любое место за панелью инструментов.

2. Для того чтобы вернуть панель инструментов к первоначальному виду по умолчанию, следует открыть вкладку «Панели инструментов», выбрать в списке нужную панель и нажать кнопку «Сброс». Появится сообщение, запрашивающее подтверждения на отмену всех изменений.

3. Для создания кнопок, открывающих или запускающих на выполнение объект базы данных, выберите вкладку «Команды», в списке «Категории» выберите, например, «Все таблицы». Рядом в списке «Команды» появятся все таблицы текущей базы данных. Выберите название нужной таблицы и перетащите на панель инструментов. Проверьте работу кнопки.

4. Для изменения текста или значка на кнопках, щелкните правой кнопкой мыши по кнопке. В появившемся контекстном меню выберите команду «Выбрать значок для кнопки» – для смены значка на кнопке; выберите команду «Изменить значок на кнопке» – для редактирования изображения кнопки; выберите «Имя» – для ввода в текстовое поле нужной подписи для отображения ее на кнопке (при включенном пункте «Значок и текст»).

5. Чтобы создать новую пустую панель инструментов для настройки ее пользователем и добавления любого набора нужных кнопок, выберите вкладку «Панели инструментов» и в списке найдите элемент «Служебная программа1»

или «Служебная программа 2» (для появления панели установите галочку слева).

6. Для создания пользовательской панели инструментов, которая станет частью базы данных, откройте вкладку «Панели инструментов» и в ней нажмите кнопку «Создать». Появится диалоговое окно «Создание панели инструментов», запрашивающее название новой панели. Access создаст плавающую панель инструментов, кнопки в которую добавляются из вкладки «Команды».

7. Для удаления пользовательской панели инструментов во вкладке «Панели инструментов» выберите нужную панель и нажмите кнопку «Удалить».

### **Создание новых таблиц при помощи Мастера таблиц.**

Мастер таблиц позволяет создать новую таблицу на основе 25 образцов таблиц делового применения и 20 образцов таблиц личного применения.

Чтобы создать таблицу «Каталог видеозаписей» при помощи Мастера таблиц:

1. Откройте БД Борей. Сделайте активным окно базы данных. Для этого закройте все окна объектов БД или нажмите кнопку «Окно базы данных» на панели инструментов.

2. Раскройте объект «Таблицы» окна базы данных и дважды щелкните на ярлыке «Создание таблицы с помощью мастера» (или нажмите кнопку «Создать» и в появившемся диалоговом окне выделите элемент «Мастер таблиц» и нажмите кнопку ОК).

3. В первом диалоговом окне выберите переключатель «Личное применение». В списке «Образцы таблиц» выделите элемент «Каталог видеозаписей». В списке «Образцы полей» появятся поля, которые могут быть включены в новую таблицу.

4. Чтобы включить все поля нажмите кнопку “>>”. При помощи кнопки “>” в список «Поля новой таблицы» можно добавить выделенный элемент списка «Образцы полей». Кнопка “<” удаляет выделенный элемент списка «Поля новой таблицы», а кнопка “<<” удаляет все поля из этого списка. Нажмите кнопку «Далее».

5. Появится второе диалоговое окно, в котором можно ввести имя новой таблицы и выбрать способ определения ключевого поля. Выберите переключатель «Пользователь определяет ключ самостоятельно». Нажмите кнопку «Далее».

6. Появится следующее диалоговое окно, в котором можно выбрать ключевое поле и указать, какие данные должны содержаться в ключевом поле.

Выбор поля “КодКаталогВидео” в качестве ключевого поля и автоматическое присвоение последовательных чисел каждой записи является в этом случае наиболее подходящим, поэтому выбранные параметры можно оставить без изменений. Нажмите кнопку «Далее».

7. Следующее диалоговое окно появляется только в том случае, если в базе данных уже содержатся другие таблицы. Мастер таблиц позволяет связать новую таблицу с остальными. По умолчанию новая таблица не связывается ни с какой другой. Нажмите кнопку «Далее».

8. Появится последнее диалоговое окно. Выберите переключатель «Изменить структуры таблицы» и нажмите кнопку «Готово». Новая таблица будет открыта в режиме конструктора.

9. Чтобы просмотреть содержимое таблицы, нажмите на кнопке «Вид» или выберите команду «Вид, Режим таблицы».

10. Нажмите кнопку «Заккрыть».

## ЛАБОРАТОРНАЯ РАБОТА №2 (2 часа)

**Тема.** Создание таблиц в режиме конструктора

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать таблицы в режиме конструктора,
- определять значения по умолчанию для полей,
- добавлять условия на значение полей,
- добавлять условия на значение записей,
- выбирать первичный ключ таблицы,
- индексировать таблицу.

### Создание таблицы в режиме конструктора.

1. Нажмите на объекте «Таблицы», а затем нажмите кнопку «Создать». Выберите в списке элемент «Конструктор» и нажмите ОК или откройте ярлык «Создание таблицы в режиме конструктора». Новая пустая таблица будет открыта в режиме конструктора.

2. Каждое поле таблицы имеет набор свойств. О свойствах таблиц и полей более подробно можно узнать, установив текстовый курсор в это поле и нажав на клавишу F1. Введите данные из таблицы 2.1.

Таблица 2.1

Имя_ поля	Тип данных	Описание
Код_сотрудника	Числовой	Код сотрудника, с которым связано мероприятие. Значение присваивается на основе значения поля «Код сотрудника» таблицы «Сотрудники»
Тип мероприятия	Текстовый	Сокращение от названия мероприятия. Д-анализ деятельности, М-мероприятия по продвиже-

		нию, Р-рекламная компания, С-изучение запроса, А-анкетирование, П-прогнозирование, К-анализ конкурентов.
Код_ответственного	Числовой	Код сотрудника, инициировавшего или рекомендовавшего проведение мероприятия.
Дата план	Дата/Время	Запланированная дата проведения мероприятия.
Код_начальника	Числовой	Код сотрудника, утвердившего проведение мероприятия.
Дата_факт	Дата/Время	Дата проведения мероприятия. Поле остается незаполненным, если мероприятие не проведено.
Рейтинг	Числовой	Рейтинг сотрудника по 10-бальной шкале. Незаполненное поле означает отсутствие рейтинга
Сумма	Денежный	Предполагаемый рост дохода от проведенного мероприятия.
Примечание	МЕМО	Полное описание мероприятия. В этом поле ответственный за проведение мероприятия и начальник могут оставлять свои замечания.

3. Заполните свойства полей в соответствии с таблицей 2.2.

Таблица 2.2.

Имя поля	Подпись	Размер поля	Формат поля
Код сотрудника	Сотрудник	Длинное целое	Основной
Тип мероприятия	Мероприятие	1	@>
Код ответственного	Ответственный	Длинное целое	Основной
Дата план	Дата-план	-	Краткий формат даты
Код начальника	Начальник	Длинное целое	Основной
Дата факт	Дата-факт	-	Краткий формат даты
рейтинг	рейтинг	Целое	Основной
Сумма	Сумма	-	0,00р

4. Переключитесь в режимы таблицы для просмотра результата выполненной работы. Появится диалоговое окно, запрашивающее подтверждение сохранения структуры таблицы. Нажмите кнопку ДА. Появится диалоговое окно «Сохранение». Введите имя таблицы: «Задания» и нажмите кнопку ОК или нажмите клавишу Enter. После сохранения структуры таблицы появится окно, сообщающее о том, что в таблице не задано ключевое поле и предлагающее создать его. Сейчас откажитесь от определения ключевых полей, нажав кнопку НЕТ.

### Определение значений по умолчанию для полей.

Обратите внимание на то, что в единственной записи таблицы «Задания» числовым полям присвоены нулевые значения. При создании таблицы, полями с типами данных «Числовой» и «Денежный» автоматически присваивается значение 0, а поля других типов по умолчанию остаются пустыми. Иногда нулевое значение для числовых полей недопустимо. В этом случае значение по умолчанию должно быть изменено. Чтобы увеличить производительность работы при вводе данных в таблицу, необходимо задать значения по умолчанию для ее полей.

1. В таблице 2.3 приведены значения по умолчанию для полей таблицы «Задания», введите их.

Таблица 2.3- Значения по умолчанию для полей таблицы

Поле	Значение по умолчанию	Пояснение
Код_сотрудника		Число 0 не может быть кодом сотрудника, поэтому значение по умолчанию должно быть удалено.
Тип мероприятия	С	Изучение спроса выполняется чаще всего.
Код_ответственного		Число 0 не может быть кодом ответственного.
Дата план	Date()	Выражение возвращает текущую дату
Код начальника		Число 0 не может быть кодом начальника
Дата_факт	Date()+28	Выражение возвращает текущую дату плюс 28 дней.
Рейтинг		Многие мероприятия влияют на рейтинг сотрудника.
Сумма		Поле «Сумма» не должно содержать нулевого значения.
Примечание		Изменений не требуется. Для текстовых полей и полей типа MEMO Access 2003 автоматически устанавливает значение по умолчанию «пустая строка».

Если в свойстве «Значение по умолчанию» ничего не введено, то полю присваивается значение по умолчанию «Пустое значение»(Null).

2. После того как значения по умолчанию присвоены, возвратитесь в режим таблицы. Появится диалоговое окно, запрашивающее подтверждения изменения структуры таблицы. Нажмите кнопку ДА.

#### **Добавление условий на значение полей.**

Условия на значения проверяют корректность данных только в одном поле, независимо от значений других полей.

1. Перейдите в режим конструктора.
2. Добавьте в структуру таблицы «Задания» условия на значение, приведенные в таблице 2.4.

Таблица 2.4

Поле	Условие на значение	Текст сообщения об ошибке
Код_сотрудника	>0	Введите правильный код сотрудника
Тип_мероприятия	“П” Or ”Д” Or ”К” Or ”Р” Or ”М” Or ”С” Or ”А”	Недопустимый код. Допустимы только коды П, Д, М, К, Р, С и А
Код_ответственного	>0	Введите правильный код сотрудника, ответственного за мероприятие.
Дата_план	Between Date()-365 And Date()+365	Мероприятие не может быть запланировано более чем на один год вперед.
Код_начальника	>0 Or Is Null	Введите правильный код начальника или оставьте поле пустым.
Рейтинг	Between 0 And 10 Or Is Null	Введите правильный рейтинг сотрудника (число от 0 до 10) или оставьте поле пустым

3. Установите для свойства «Обязательное поле» полей «Код\_Сотрудника», «Тип\_Мероприятия», «Код\_Ответственного» и «Дата\_План» значение ДА.
4. Сохраните структуру таблицы.

### **Добавление условий на значение записей.**

При нормальной организации работы фирмы значение поля «Дата\_Факт» не должно превышать значение поля «»Дата\_План». Можно задать условие на значение записи таблицы. В таких условиях могут сравниваться одновременно значения нескольких полей. Само условие и текст сообщения об ошибке, появляющийся при его нарушении, можно задать в диалоговом окне «Свойства таблицы».

Чтобы определить условие на значение записи для поля «Дата\_Факт»:

1. Откройте таблицу «Задания» в режиме конструктора и нажмите кнопку «Свойства» на панели инструментов. Появится диалоговое окно «Свойства таблицы».
2. В поле ввода «Описание» введите: Персональные мероприятия по отделу.

3. Поместите текстовый курсор в поле «Условие на значение» и нажмите кнопку «...» справа от него. Появится диалоговое окно «Построитель выражений». В левом списке этого диалогового окна выделена таблица «Задания», а в центральном списке выводится список ее полей.

4. Дважды щелкните левой кнопкой мыши по элементу «Дата\_Факт» в списке полей таблицы. В окне выражения появится строка [Дата\_Факт].

5. Введите: >= в окне выражения и дважды щелкните левой кнопкой мыши по элементу «Дата\_План» в списке полей таблицы, чтобы добавить его в выражение.

6. При создании таблицы «Задания» в поле «Дата\_Факт» планировалось хранить пустые значения для мероприятий, которые еще не произошли. Поэтому добавьте к выражению, задающему условие на значение, строку Or [Дата\_Факт] Is Null.

Таким образом, в построителе выражений у вас должна получиться формула:

[Дата\_Факт] >= [Дата\_План] Or [Дата\_Факт] Is Null

7. Чтобы закрыть окно «Построитель выражений» и добавить условие на значение записей таблицы, нажмите кнопку ОК.

8. Поместите текстовый курсор в поле «Сообщение об ошибке» в диалоговом окне «Свойства таблицы» и введите текст сообщения: «Дата проведения мероприятия не может предшествовать запланированной дате мероприятия».

9. Закройте окно «Свойства таблицы», нажав кнопку «Заккрыть» или нажав кнопку «Свойства» на панели инструментов.

10. Сохраните структуру таблицы.

### **Выбор первичного ключа таблицы**

Чтобы определить первичный ключ таблицы «Задания» и проиндексировать ее по значению ключа:

1. Откройте таблицу «Задания» в режиме конструктора.

2. Выделите поле «Код\_Сотрудника», нажав на область выделения строки в левой части бланка структуры таблицы.

3. Нажмите клавишу <Ctrl> и, удерживая ее нажатой, выделите поле «Тип\_Мероприятия» и поле «Дата\_План».

4. Нажмите кнопку «Ключевое поле» на панели инструментов. Слева от каждого из выделенных полей появится маркер ключевого поля, информирующий о том, что поле включено в первичный ключ таблицы.

5. Чтобы определить последовательность, в которой выделенные поля входят в ключ, нажмите кнопку «Индексы» на панели инструментов. Появится окно «Индексы».

### **Индексирование таблицы**



Каждая таблица Access может иметь до 32 индексов, 5 из которых могут быть составными (в составной индекс может входить не более 10 полей).

Чтобы проиндексировать таблицу «Задания» по значению поля «Дата\_Факт» и создать составной индекс по значениям полей «Тип\_мероприятия» и «Дата\_План».

1. Выделите поле «Дата\_Факт».
2. Сделайте активным свойство «Индексированное поле».
3. Раскройте список «Индексированное поле», щелкнув по кнопке со стрелкой вниз или нажав клавишу <F4>.
4. Поле «Дата\_Факт» может содержать повторяющиеся данные, поэтому в списке нужно выбрать значение «Да(Допускаются совпадения)».
5. Нажмите кнопку «Индексы» на панели инструментов. В этом окне выводятся все индексы таблицы. Чтобы создать составной индекс, введите: Тип\Дата в столбце «Индекс» и выберите поля «Тип\_Мероприятия» и «Дата\_План» в раскрывающихся списках столбца «Имя поля».
6. Перейдите в режим таблицы. Подтвердите сохранение структуры таблицы.

### **ЛАБОРАТОРНАЯ РАБОТА № 3 (2 часа)**

**Тема:** Создание связей, полей подстановок и экспорт данных

Цель лабораторной работы заключается в том, чтобы научиться:

- устанавливать связи между таблицами
- настраивать режим таблицы
- экспортировать данные
- использовать поля подстановок.

#### **Связи между таблицами**

Связи между таблицами базы данных можно определить и просмотреть в графическом окне «Схема данных». Чтобы определить связь между таблицами «Задания» и «Сотрудники»:

1. Закройте все окна таблиц базы данных. Access 2003 не позволяет создать или изменить связь, если открыта хотя бы одна таблица.

2. Выберите команду «Сервис, Схема данных» или нажмите кнопку «Схема данных» на панели инструментов. Появится окно «Схема данных», которое используется для просмотра и изменения существующих связей и для определения новых связей между таблицами и/или запросами.

3. Выберите команду «Связи, Добавить таблиц» или нажмите кнопку «Добавить таблицу» на панели инструментов. Появится диалоговое окно «Добавление таблицы».

4. Раскройте вкладку «Таблицы», в списке таблиц выделите «Задания» и нажмите кнопку «Добавить». Нажмите кнопку «Заккрыть». Таблица «Задания» появится в окне «Схема данных».

5. Связь между таблицами «Задания» и «Сотрудники» строится по значению полей «Код Сотрудника». Поместите указатель мыши на поле «Код Сотрудника» таблицы «Задания», нажмите левую кнопку мыши и, не отпуская ее, перетащите появившийся значок поля на поле «Код Сотрудника» таблицы «Сотрудники». Отпустите левую кнопку мыши. Появится диалоговое окно «Связи»

6. Нажмите кнопку «Объединение». Появится диалоговое окно «Параметры объединения». Выберите переключатель «3». Нажмите кнопку ОК.

7. Для обеспечения ссылочной целостности установите флажок «Обеспечение целостности данных».

8. Нажмите кнопку «Создать», чтобы подтвердить созданию связи перейти в окно «Схема данных».

9. Закройте окно «Схема данных». Появится диалоговое окно, запрашивающее подтверждение изменения схема данных. Подтвердите изменение, нажав кнопку «Да».

Примечание: отношение «многие-ко-многим», например, связь между таблицами «Заказы» и «Товары», реализуется на основе промежуточной таблицы (в данном случае таблицы «Заказано»), которая связана с одной таблице отношением «один-ко-многим» (таблица «Заказы» с таблицей «Заказано»), а с другой таблицей отношением «ногие-к-одному» (таблица «Заказано» с таблицей «Товары»).

### **Изменение связей между таблицами**

Изменение связи может быть вызвано изменением типа данных полей, по которым строится связь (поля, по которым строится связь, должны иметь один тип). В этом случае нужно удалить связь, изменить типы данных и переустановить связь. Чтобы переустановить связь:

1. Сделайте активным окно базы данных.
2. Откройте окно «Схема данных».
3. Щелкните левой кнопкой мыши по связи, которую нужно изменить. Линия выделенной связи становится толще.
4. Чтобы удалить выделенную связь, нажмите клавишу <Delete>. Подтвердите удаление, нажав кнопку «Да» в появившемся диалоговом окне.
5. Если предполагается изменить тип поля, входящего в ключ таблицы, то удалите остальные связи с этой таблицей.
6. Измените типы данных полей таблиц, чтобы они соответствовали новым связям.
7. Переустановите связи.

### Добавление записей в таблицу Задания

Теперь вам предоставляется возможность проверить результаты вашей работы. Заполните таблицу «Задания», используя данные из таблицы 3.1 (поля «Дата план» и «Дата факт» заполняются по умолчанию).

Таблица 3.1

Сотрудник	Мероприятие	Ответственный	Начальник	Рейтинг	Сумма	Примечание
1	С	1		2	2000000	
2	М	1		5	3500000	
3	С	1		8	2250000	
4	С	2	2	1	2250000	
5	К	2	2	3	2250000	
6	Р	5	5	5	4000000	
7	А	5	2	4	3000000	
8	С	2	2	7	2500000	
9	С	5	2	9	3000000	

### Настройка режима таблицы

Чтобы скрыть поле таблицы, выделите его и выберите команду «Формат, Скрыть столбцы». Чтобы снова показать скрытые поля, выберите команду «Формат, Показать столбцы». Появится диалоговое окно «Отображение столбцов». Чтобы отразить скрытое поле, установите флажок слева от его имени.

Чтобы изменить шрифт, которым содержимое таблицы выводится на экран и на принтер, выберите новый шрифт в раскрывающемся списке «Шрифт» на панели инструментов. Можно также выбрать команду «Формат, Шрифт».

Чтобы изменить высоту строк таблицы, перемещайте указатель мыши над кнопками выделения записей в левой части окна таблицы, пока указатель не превратится в двустороннюю стрелку. Нажмите левую кнопку мыши и протащите указатель вверх, чтобы уменьшить высоту строк, или вниз, чтобы увеличить ее. Отпустите левую кнопку мыши. Чтобы задать точную высоту строк таблицы (при печати строки будут иметь именно эту высоту), можно использовать команду «Формат, Высота строки». Появится диалоговое окно «Высота строки». В поле «Высота строки» укажите требуемую высоту (в пунктах) и нажмите кнопку ОК. Чтобы после распечатки таблица выглядела красиво, задайте высоту, равную размеру шрифта, умноженному на коэффициент 1.25.

Чтобы изменить ширину поля таблицы данных, выделите его и выберите команду «Формат, Ширина столбца». Появится диалоговое окно «Ширина столбца». Введите число символов, которые можно уместить в поле. Чтобы установить ширину поля по максимальной длине данных в поле, нажмите кнопку «По ширине данных». Допускается также изменение ширины поля путем перемещения с помощью мыши правой границы области выделения поля.

Линии сетки таблицы можно оформить с помощью команды «Формат, Ячейки». Поэкспериментируйте с разными установками.

В режиме таблицы можно также вывести на экран панель инструментов Формат (режим таблицы). Для этого выберите команду «Вид, Панели инструментов», в появившемся диалоговом окне установите флажок «Формат (режим таблицы)». Измените цвет фона, цвет текста вашей таблицы.

### **Изменение порядка следования полей**

Выделите поле и, удерживая нажатой левую кнопку мыши, перетащите поле на новое место, затем отпустите кнопку. Поле будет перемещено на новое место. Толстая вертикальная черта, появляющаяся при перетаскивании поля, указывает на то место, где будет находиться после перемещения левая граница поля.

### **Закрепление полей таблицы**

Закрепленные поля таблицы всегда остаются видны и при горизонтальной прокрутке таблицы прокручиваются только незакрепленные поля. Чтобы закрепить поля «Код сотрудника» и «Тип Мероприятия», выделите их и выберите команду «Формат, Закрепить столбцы». Разделительная линия между закрепленными и незакрепленными полями станет черной, а не светлой. Чтобы освободить закрепленные столбцы, выберите команду «Формат, Освободить все столбцы».

### **Экспорт данных**

Существует несколько методов экспорта данных в другие приложения:

1. Встаньте курсором на таблицу «Типы». Выберите команду «Файл, Экспорт». Появится диалоговое окно «Экспорт объекта». В появившемся диалоговом окне укажите имя файла и формат, в котором будет сохранена таблица (позволяет сохранить таблицы в форматах XLS, RTF, DBF, TXT, HTML и др.). Затем нажмите кнопку «Сохранить». Создастся файл выбранного вами формата.

2. Встаньте курсором на таблицу «Задания». Нажмите на панели инструментов, на кнопке «Связи с Office» стрелку, затем выберите «Анализ в MS Excel», чтобы сохранить данные в рабочем листе Excel или «Публикация в MS Excel», чтобы сохранить данные в формате RFT. Access 2003 запускает соответственно Excel или Word для отображения экспортированного документа.

### **Использование полей подстановок**

Мастер подстановок, впервые появившийся в Access 2000, позволяет заменить обычные поля таблиц полями со списком. Это позволяет создать для определенного поля список допустимых значений, при выборе элемента в котором текущую запись автоматически заносится требуемая величина. Поля подстановки имеют следующие типы:

1. Для поля, содержащего значения внешнего ключа, создается список значений одного или нескольких полей связанной базовой таблицы. Например, таблица «Заказы» БД Борей имеет два поля внешнего ключа: «КодКлиента» и «КодСотрудника». Для поля «КодКлиента» отображается список значений поля «Название» таблицы «Клиенты», а для поля «КодСотрудника» - список значений полей «Фамилия» и «Имя» таблицы «Сотрудники», разделенных запятой и пробелом.

2. Для любого другого поля, исключая поле простого первичного ключа, список содержит фиксированные значения.

### **Добавление списка, задающего значение внешнего ключа, при помощи Мастера подставок**

Рассмотрим построение списка подстановки значений внешнего ключа таблицы Задания, созданной ранее. Чтобы заменить поле «Код сотрудника» полем подстановки:

1. Создайте копию таблицы «Задания» под именем: «Задания (поля подстановки)» с помощью Буфера обмена.

2. Откройте копию в режиме конструктора и выделите поле «Код сотрудника». В ячейке столбца «Тип данных» выберите из списка элемент «Мастер подстановок». Отобразится первое диалоговое окно Мастера подстановок.

3. Поле подстановки должно включать значения из другой таблицы (таблица «Сотрудники»), поэтому изменять установки в первом окне не требуется. Нажмите на кнопку «Далее» для вывода второго диалогового окна Мастера подстановок.

4. Выбрав переключатель «Таблицы», выделите таблицу «Сотрудники», с которой связано поле «Код сотрудника». Нажмите кнопку «Далее» для вывода третьего диалогового окна Мастера подстановок.

5. Добавьте поля «КодСотрудника», «Фамилия» и «Имя» в список подстановки. Включение ключевого поля базовой таблицы, связанное с полем внешнего ключа, является обязательным. Нажмите кнопку «Далее».

6. Измените ширину столбцов, обеспечив достаточный интервал между фамилией и именем. Поле «КодСотрудника» является ключевым и рекомендуется скрыть его. Нажмите кнопку «Далее» для вывода последнего окна Мастера подстановок.

7. Не изменяйте предлагаемой подписи для поля подстановки. Если задать другое значение, то изменится имя поля, а не подпись.

8. Нажмите кнопку «Готово», а затем кнопку ОК в ответ на приглашение сохранить структуру таблицы.

9. Просмотрите результат в режиме таблицы. В поле «КодСотрудника» выводится только первый столбец списка.

### **Добавление списка постоянных значений к полю таблицы**

Добавление списка фиксированных значений к полю таблицы очень похоже на создание списка значений внешнего ключа. Чтобы связать поле «Тип Мероприятия» таблицы «Задания (поля подстановки)» со списком постоянных величин:

1. Выберите поле «Тип Мероприятия». В ячейке столбца «Тип данных» выберите из списка элемент «Мастер подстановок».

2. В появившемся диалоговом окне выберите переключатель «Фиксированный набор значений» и нажмите кнопку «Далее».

3. Во втором диалоговом окне Мастера подстановок введите значение 2 в поле «Число столбцов» и нажмите клавишу <Tab> для создания второго столбца.

4. Ведите значения в столбцы Столбец1 и Столбец2:

П Прогнозирование

Д Анализ деятельности

М Мероприятия по продвижению

Р Рекламная компания

С Изучение спроса

А Анкетирование

К Анализ конкурентов

Измените ширину столбцов, обеспечив достаточный интервал между ними. Нажмите кнопку «Далее» для вывода третьего окна Мастера подстановок.

5. В поле «Тип Мероприятия» используется один символ для обозначения типа мероприятия, поэтому выберите «Столбец1» в качестве столбца, содержащего значения, которые необходимо добавить в таблицу. Нажмите кнопку «Далее» для вывода последнего окна Мастера подстановок.

6. Не изменяя предлагаемой Мастером подстановок подписи для списка подстановки, нажмите «Готово».

7. Нажмите кнопку «Вид» и сохраните внесенные изменения. Увеличьте ширину столбца «Тип» до ширины списка, раскройте список и убедитесь в том, что в нем содержатся фиксированные значения.

Примечание: если необходимо скрыть сокращенное название типа мероприятия, замените первую величину в поле свойства «Ширина столбцов» вкладки «Подстановка» на 0.

Чтобы убрать список подстановки у поля, выберите его, раскройте вкладку «Подстановка» и выделите элемент «Поле» в списке «Тип элемента управления».

## **ЛАБОРАТОРНАЯ РАБОТА № 4 (2 часа)**

**Тема:** Поиск и фильтрация данных

Цель лабораторной работы заключается в том, чтобы научиться:

- сортировать таблицы по значению одного или нескольких полей
- осуществлять поиск записей по значению поля
- осуществлять поиск и замену значений полей таблицы
- использовать обычный фильтр, фильтр по выделенному фрагменту и расширенный фильтр

### **Сортировка таблицы по значению одного поля**

Почтовый список рассылки принято сортировать по почтовому индексу. Чтобы сортировать таблицу «Клиенты» по значению поля «Индекс»:

1. Выделите поле «Индекс».
2. Выделите команду «Записи. Сортировка по возрастанию» или нажмите кнопку «Сортировка по возрастанию» на панели инструментов.

### **Сортировка таблицы по значению нескольких полей**

При создании списка рассылки лучше использовать сортировку таблицы по значениям полей «Страна» и «Индекс». По умолчанию Access 2003 устанавливает приоритеты сортировки слева направо, т.е. сначала таблица сортируется по значению самого левого выделенного поля, затем по значению следующего поля и т.д. изменить приоритеты сортировки можно двумя способами: изменить порядок следования полей в таблице или указать порядок сортировки в окне расширенного фильтра.

Чтобы произвести сортировку таблицы по значению нескольких полей:

1. Переместить поле «Страна» левее поля «Индекс».
2. Выделите поля «Страна» и «Индекс».
3. Выберите команду «Записи, Сортировка, По возрастанию».

### **Отмена сортировки**

Access 2003 предоставляет несколько способов, помогающих придать таблице прежний вид:

1. Выберите команду «Записи, Удалить фильтр».
2. Закройте таблицу без сохранения.

### **Поиск записей по значению поля**

Access 2003 позволяет производить поиск записей, в которых значение определенного поля частично или полностью совпадают с некоторой величиной.

Чтобы найти записи таблицы «Клиенты», в поле «Область» которых содержится «Quebec»:

1. Выделите поле или поместите текстовый курсор в любую ячейку поля.
2. Выберите команду «Правка, Найти» или нажмите кнопку «Найти» на панели инструментов. Появится диалоговое окно «Поиск в поле:...».

3. В поле ввода «Образец» введите искомый текст.
4. В раскрывающемся списке «Совпадение» выберите значение «Поле целиком».
5. Остальные установки изменять не нужно.
6. Нажмите кнопку «Найти», а затем «Найти далее».

Можно использовать другой вариант поиска Quebec:

1. В поле ввода «Образец» введите: **Qu**.
2. В списке «Совпадение» выберите значение «С начала поля».
3. Нажмите кнопку «Найти».

Приведем параметры, которые можно установить в диалоговом окне «Поиск в поле»:

1. Чтобы произвести поиск с условием, что искомый текст должен полностью совпадать с образцом, включает учет того, какие буквы являются прописными, а какие строчными, установите флажок «С учетом регистра».
2. Для поиска данных в указанном формате отображения (например, дата, сохраненная в формате 1/5/92, может выводиться как 5 января 92 г.), установите флажок «С учетом формата полей». Такой поиск является обычно самым медленным.
3. Чтобы найти следующую запись, в поле которой содержится значение, совпадающее со значением поля ввода «Образец», нажмите кнопку «Найти далее».
4. Элементы из раскрывающегося списка «Просмотр» определяют направление поиска. Если параметр «Просмотр» имеет значение «Вниз», то поиск будет начат с первой записи таблицы, независимо от того, где находился указатель текущей записи. То же самое относится к значению «Вверх», но поиск осуществляется от последней записи таблицы к первой. Если параметр «Просмотр» имеет значение «Все», то поиск будет производиться от текущей записи до последней, а затем от первой записи до текущей.

### **Замена вхождений образца а поле**

Чтобы произвести замену вхождений образца в поле, выберите команду «Правка, Заменить». Появится диалоговое окно «Замена в поле:...». Заполните поля ввода, чтобы заменить значения полей, содержащих Quebec на **Qvebec**. Чтобы заменить значения только некоторых полей, удовлетворяющих критерию поиска, нажмите кнопку «Найти далее», а затем, если значение поля должно быть изменено, нажмите кнопку «Заменить». Чтобы изменить все вхождения образца в поле таблицы, нажмите кнопку «Заменить все».

### **Фильтр по выделенному**



Критерий отбора записей устанавливается путем выделения всего значения поля таблицы или его части. После применения такого фильтра Access 2003 отображает в таблице только те записи, которые соответствуют критерию отбора.

Недостатком этого способа является то, что отбор записей можно производить по значению только одного поля.

Чтобы применить фильтр по выделенному к таблице «Клиенты» и вывести на экран только клиентов, находящихся в Германии:

1. Откройте «Клиенты» в режиме таблицы.
2. Встаньте на первую запись поля «Страна».
3. Выделите весь текст в поле «Страна» (в данном случае оно содержит значение «Германия»).
4. Выберите команду «Записи, Фильтр, Фильтр по выделенному» или нажмите кнопку «Фильтр по выделенному» на панели инструментов. В окне таблицы будут выведены записи, удовлетворяющие критерию фильтра.

Фильтр по выделенному можно применять к нескольким полям. Например, после применения фильтра для отображения только тех клиентов, которые находятся в Германии можно перейти к полю «Город» и применить второй фильтр по выделенному, используя в качестве параметра фильтра «Берлин». Результирующая таблица будет включать только тех клиентов, которые находятся в Германии и в Берлине. Прделайте это.

Чтобы удалить фильтр, нажмите кнопку «Удалить фильтр» на панели инструментов. Это та же самая кнопка, что и кнопка «Применить фильтр». Когда кнопка нажата, фильтр применен, когда отжата, фильтр удаляется.

### **Обычный фильтр**

Использование обычного фильтра позволяет быстро отобрать записи по значениям нескольких полей.

Чтобы произвести фильтрацию таблицы «Клиенты» для отбора информации о клиентах из США или Канады при помощи обычного фильтра:

1. Откройте таблицу «Клиенты» в режиме таблицы.
2. Выберите команду «Записи, Фильтр, Изменить фильтр» или нажмите кнопку «Изменить фильтр» на панели инструментов. Появится окно «Клиенты: фильтр».
3. Сделайте текущим поле «Страна», и нажмите на кнопку со стрелой, чтобы раскрыть список возможных значений.
4. Выберите в списке значение «Канада».
5. Раскройте вкладку «Или» в нижней части окна. Критерии поиска, заданные в каждой из вкладок, объединяются при помощи логического оператора «ИЛИ».
6. В поле «Страна» выберите в списке значение «США».

7. Выберите команду «Фильтр, Применить фильтр». Access 2003 отберет все записи, удовлетворяющие критерию фильтра, и выведет их в окне таблицы.

Если в одной из вкладок окна фильтра заполнены критерии отбора для нескольких полей, то критерии отбора объединяются при помощи логического оператора «И». Например, необходимо отобрать информацию обо всех заказах, за которые ответственна Воронова Дарья, и страной получателя для которых является Франция. Выполнить эту работу можно при помощи обычного фильтра. Для этого:

1. Откройте таблицу «Заказы» в режиме таблицы. Закрепите столбцы «Код заказа», «Клиент» и «Сотрудник».
2. Выберите команду «Записи, Фильтр, Изменить фильтр».
3. Выберите команду «Правка, Очистить бланк» или нажмите кнопку «Очистить бланк» на панели инструментов, чтобы удалить предыдущие критерии отбора.
4. В раскрывающемся списке «Сотрудник» выделите значение Воронова Дарья, а в раскрывающемся списке поля «Страна получателя» выделите значение Франция.
5. Выберите команду «Фильтр, Применить фильтр». В окне таблицы появятся записи, удовлетворяющие критерию отбора.

Если при использовании фильтра отбираются не те записи, которые нужны, значит, мог наложиться старый критерий отбора. Чтобы очистить фильтр от всех предшествующих критериев отбора, выберите команду «Правка, Очистить бланк».

### **Расширенный фильтр**

Расширенный фильтр является самым мощным средством фильтрации данных Access 2003. В окне расширенного фильтра можно указать как критерий отбора для различных полей таблицы, так и порядок сортировки этих полей, но при помощи фильтра можно отобрать записи только из одной таблицы.

Чтобы создать фильтр для таблицы «Заказы»:

1. Откройте таблицу «Заказы» в режиме таблицы.
2. Выберите команду «Записи, Фильтр, Расширенный фильтр». Появится окно расширенного фильтра (рис. 4.1). по умолчанию фильтру присваивается имя «Заказы фильтр1». В нижней части окна расширенного фильтра появляется бланк фильтра, в котором можно указать критерии отбора.
3. Перетащите поле «Код Заказа» в первую ячейку строки «Поле» бланка фильтра.
4. В остальные ячейки перетащите поля «Страна Получателя», «Индекс Получателя» и «Дата Размещения».

- Установите параметры сортировки значений полей «Страна Получателя» и «Индекс Получателя»: по возрастанию (рис. 4.1)

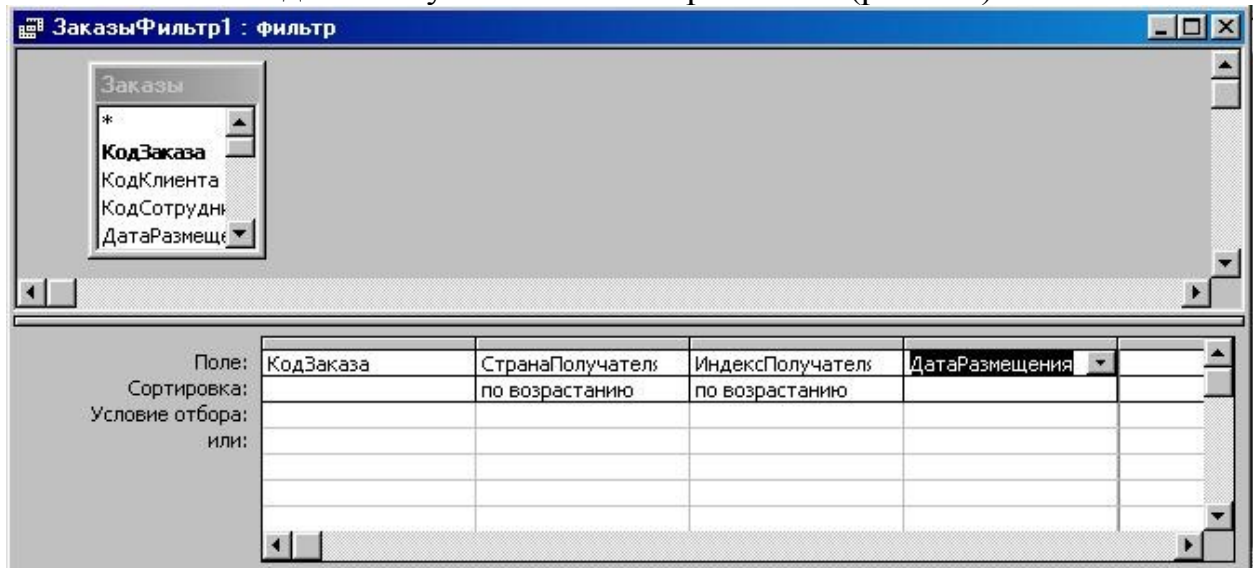


Рис. 4.1

- Выберите команду «Фильтр, Применить фильтр» или нажмите кнопку «Применение фильтра» на панели инструментов.
- Таблица будет отсортирована по значениям полей «Страна получателя» и «Индекс получателя».
- Выберите команду «Записи, Фильтр, Расширенный фильтр».
- Чтобы произвести отбор записей, относящихся к заказам, в которых страной- получателем является США, введите в ячейку, находящуюся на пересечении строки «Условие отбора» и столбца «Страна получателя», слово США. Нажмите клавишу <Enter>. Access 2003 автоматически заключит введенную строку в кавычки.
- Выберите команду «Фильтр, Применить фильтр». В окне и таблицы появятся только те записи, которые относятся к отправке товаров в США.

Чтобы вывести все заказы из стран Северной Америки, полученные не ранее 01.01.94:

- Выберите команду «Записи, Фильтр, Расширенный фильтр».
- Введите Канада в качестве второго критерия отбора для поля «Страна получателя», нажмите клавишу <Enter>, а затем введите Мексика в качестве третьего критерия. При добавлении нескольких критериев отбора для одного поля они группируются при помощи оператора «ИЛИ».
- Введите >=#1/1/94# в качестве первого критерия отбора для поля «Дата Размещения». Если в одной строке бланка фильтра определено нескольких критериев отбора, то они группируются при помощи оператора «И». Таким образом, первую строку критериев отбора можно интерпретировать так: отобрать все заказы из США, поступившие не

ранее 01/01/94. символы # указывают на то, что значение, заключенное в них, имеет тип «Дата/Время».

4. В поле «Дата Размещения» введите тот же критерий отбора для Канады и Мексики. Бланк фильтра примет вид, показанный на рисунке 4.2.

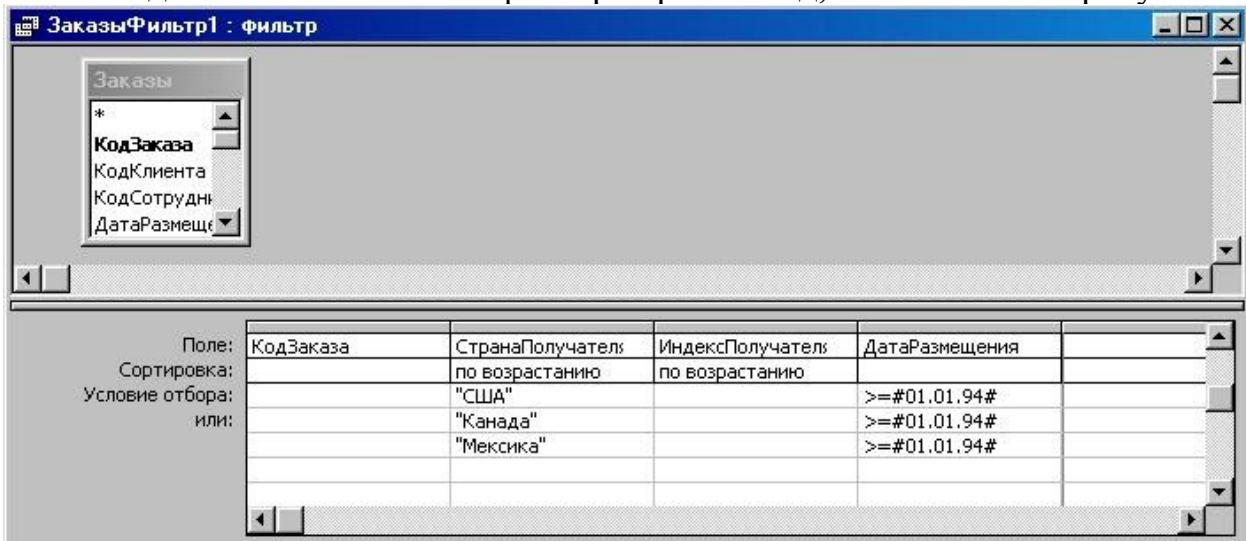


Рис 4.2

5. Нажмите кнопку «Применить фильтр» на панели инструментов. Таблица «Заказы» будет отфильтрована с использованием сложного критерия отбора.

### Сохранение и загрузка фильтра в качестве запроса

Чтобы сохранить фильтр и отменить фильтрацию таблицы «Заказы»:

1. Выберите команду «Записи, Фильтр, Расширенный фильтр».
2. Выберите команду «Файл, Сохранить как запрос».
3. В поле «Имя запроса» диалогового окна «Сохранить как запрос» укажите имя запроса: «Заказы из Северной Америки».
4. Нажмите кнопку ОК.
5. Закройте окно расширенного фильтра.
6. Отмените фильтрацию таблицы и закройте ее.

Чтобы загрузить фильтр, сохраненный в качестве запроса:

1. Откройте таблицу «Заказы» в режиме таблицы.
2. Выберите команду «Записи, Фильтр, Расширенный фильтр».
3. Выберите команду «Файл, Загрузить из запроса». Появится диалоговое окно «Применяемый фильтр». Выберите имя фильтра.
4. Примените фильтр.

Фильтр, сохраненный в качестве запроса, можно использовать как запрос.

Для этого:

1. Закройте таблицу «Заказы».
2. Раскройте вкладку «Запросы» окна базы данных.

3. Дважды щелкните левой кнопкой мыши по запросу «Заказы из Северной Америки».

### **ЛАБОРАТОРНАЯ РАБОТА № 5 (2 часа)**

**Тема:** Создание простых запросов

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать запрос;
- изменять заголовки полей в запросе использовать простой запрос на создание таблицы;
- добавлять в запрос все поля таблицы;
- добавлять параметр к запросу.

#### **Создание запроса**

Конструктор запросов предоставляет больше возможностей, чем фильтры в таблицах: он позволяет выбирать поля, которые будут отображены в результатах запроса, и объединять в одном запросе данные из разных таблиц.

Чтобы создать запрос, формирующий список рассылки для клиентов торговой компании Борей:

1. Откройте базу данных Борей.

2. Раскройте вкладку «Запросы» в окне базы данных и нажмите кнопку «Создать». Появится диалоговое окно «Новый запрос», в котором можно выбрать способ создания запроса: самостоятельно (в режиме конструктора) или используя один из мастеров запросов. Выберите пункт «Конструктор» и нажмите кнопку ОК.

3. Появится диалоговое окно «Добавление таблицы». Оно содержит вкладки, которые позволяют выбрать существующие таблицы или запросы (или и те, и другие) для добавления в новый запрос. По умолчанию раскрыта вкладка «Таблицы».

4. Выберите таблицу «Клиенты» и нажмите кнопку «Добавить», затем нажмите кнопку «Закреть».

Список полей таблицы «Клиенты» появляется в верхней панели, а пустой бланк запроса по образцу - в нижней панели.

По умолчанию создается запрос на выборку, который возвращает данные, выбранные из таблиц или результирующих наборов записей других запросов на выборку. Новому запросу по умолчанию присваивается имя «Запрос1» до тех пор, пока запрос не сохранен и ему не присвоено нужное имя.

5. Текстовый курсор установите в строке «Поле» в первой колонке. Нажмите кнопку списка и выберите поле «Обращаться К».

6. Перейдите во вторую колонку. Дважды щелкните мышью по имени поля «Адрес» в списке полей таблицы «Клиенты» в верхней панели.

7. В следующие колонки добавьте поля: «Название», «Город», «Область», «Индекс» и «Страна».

Флажки в строке «Вывод на экран» устанавливаются автоматически при добавлении новых полей в бланк запроса по образцу.

8. Перейдите в режим таблицы для просмотра результата. Поскольку не задано условие отбора, запрос возвращает все записи таблицы «Клиенты».

9. Вернитесь в режим конструктора.

10. Чтобы запрос возвращал только записи о фирмах из США, введите: США в строке «Условие отбора» в колонке «Страна». Заключать выражение в кавычки не нужно, т.к. это будет сделано автоматически.

11. Для упорядочивания записей в поле «Индекс» выберите способ сортировки «по возрастанию».

12. Просмотрите результат выборки и сортировки и вернитесь в режим конструктора.

13. Чтобы создать список рассылки только для некоторых регионов, можно задать условия отбора для полей «Область» и «Индекс». Например, записи о фирмах из Калифорнии, Орегона и Вашингтона содержат значения, превышающие 90000 в поле «Индекс», а в поле «Область» - значения CA, OR и WA соответственно.

14. Поместите текстовый курсор в колонку «Область» и введите: CA в первую строку условия отбора; в следующей строке условия отбора введите: OR, затем в следующей строке введите: WA.

15. Просмотрите результат запроса.

Примечание: Условия отбора, заданные в одной строке для нескольких полей, должны выполняться одновременно, т. е. в приведенном примере будут выбраны только те записи о фирмах из Калифорнии, у которых в поле «Страна» содержится значение «США», а для выбора фирм из Орегона и Вашингтона это условие не проверяется. В этом примере все записи, удовлетворяющие заданному условию отбора для поля «Область», будут содержать «США» в поле «Страна», но, если учитывать возможность ошибок в данных таблицы «Клиенты» или совпадения названий областей в разных странах, можно повторить условие отбора для поля «Страна» в каждой строке.

### **Изменение заголовков полей в запросе**

Можно заменить заголовок поля в запросе любым удобным именем, при условии, что он не замещается значением свойства «Подпись» для этого поля в исходной таблице. Новый заголовок называется псевдонимом.

1. Вернитесь в режим конструктора. Поместите курсор в колонке «Область» перед первым символом.

2. Введите новый заголовок и за ним двоеточие без пробела: Штат:

Двоеточие отделяет новое имя поля в таблице от существующего, в результате чего ячейка бланка запроса по образцу будет содержать «Штат:Область».

3. Поместите курсор в поле «Индекс» и используйте ZIP в качестве нового заголовка поля.

4. Замените заголовок для поля «Название» - на «Компания».

5. Нажмите кнопку «Запуск» на панели инструментов, чтобы выполнить запрос.

### Сохранение запросов

Чтобы сохранить запрос и присвоить ему имя:

1. Закройте запрос, нажав соответствующую кнопку окна. Access выдаст сообщение для сохранения запроса. Нажмите кнопку «Да», чтобы сохранить запрос, появится диалоговое окно «Сохранение».

2. В текстовом поле «Имя запроса» введите: Список рассылки по США, затем ОК.

Появляется диалоговое окно, в котором нужно ввести условие отбора. Этот тип запроса не является обособленным, т.е. параметр можно добавить к запросу любого типа.

Чтобы создать запрос с параметром:

1. Создайте запрос, аналогичный запросу «Список рассылки по США» и назовите его «Список по США».

2. Откройте запрос «Список по США» в режиме конструктора.

3. В поле «Штат:Область» в первую строку условия отбора введите приглашение пользователю (рис. 5.1). Квадратные скобки, в которые заключено приглашение, означают, что данное выражение будет заменено условием отбора, которое будет введено при запуске запроса.

Поле:	НазваниеКомпании	Адрес	Город	Штат: Область	ZIP: Индекс	Страна	
Имя таблицы:	Клиенты	Клиенты	Клиент	Клиенты	Клиенты	Клиенты	
Сортировка:					по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				[Введите код штата]		"США"	
или:							

Рис. 5.1

4. Сохраните и закройте запрос.

5. Откройте измененный запрос. Появится предупреждающее сообщение об изменении данных таблицы. Выберите «Да».

Примечание: Каждый раз при выполнении запроса, создается новая таблица с именем, определенным в диалоговом окне «Создание таблицы». Если та-

блица с таким именем уже существует, ее содержимое теряется и заменяется на результаты выполнения запроса.

6. Появится диалоговое окно «Введите значение параметра», которое содержит приглашение ввести условие отбора для поля «Штат: Область».

7. Введите: WA и нажмите клавишу <Enter> или кнопку ОК.

8. Раскройте вкладку «Таблицы» и откройте таблицу «Список рассылки». Таблица содержит только записи о фирмах из штата Вашингтон.

## **ЛАБОРАТОРНАЯ РАБОТА № 6 (4 часа)**

**Тема:** Создание запросов с использованием операторов, констант и выражений

Цель лабораторной работы заключается в том, чтобы научиться:

- использовать операторы, константы и переменные в выражениях
- создавать выражения для проверки условий на значения
- задавать выражения в критериях запросов
- использовать выражения для создания вычисляемых полей запросов

### **Использование операторов и выражений**

Имена объектов, содержащие пробелы, например имена полей, должны заключаться в квадратные скобки. Хорошей практикой разработки базы данных является отсутствие пробелов в именах объектов любого типа, а чтобы имена было удобнее читать, разделяйте слова символами подчеркивания (например, Код\_сотрудника).

Для создания выражений в Access существует 6 категорий операторов

1. Арифметические операторы выполняют сложение, вычитание, умножение и деление.
2. Операторы присваивания и сравнения устанавливают и сравнивают значения.
3. Логические операторы работают со значениями, которые могут быть только истинными или ложными.
4. Операторы слияния строковых значений (конкатенации) объединяют строки символов.
5. Операторы идентификации “!” и “.” создают однозначные имена для объектов баз данных, так что можно, например, присвоить одно и то же имя полю в нескольких таблицах и запросах.
6. Другие операторы, такие как Like, Is, In и Between упрощают создание выражений для выборки записей в запросах.

В Access определено около 140 различных функций. Сгруппируем их по назначению:

1. Функции даты и времени управляют значениями Даты/Времени в полях или вводимыми как константы. С их помощью и дату, и время можно извлекать по частям (год или день месяца, часы или минуты).



2. Текстовые функции используются для работы с текстом.
3. Функции преобразования типов данных позволяют назначить наиболее подходящий тип данных для значений числовых полей.
4. Математические и тригонометрические функции выполняют над числовыми значениями операции существенно отличные от стандартных арифметических операторов Access.
5. Финансовые функции вычисляют амортизацию, платежи, банковскую учетную ставку и т.п.
6. Функции смешанного типа нужны для создания запросов, форм и отчетов Access.
7. Другие функции состоят из функций динамического обмена данных (DDE) между приложениями Windows, статистических функций над подмножествами записей и функций, используемых главным образом для программирования в Access VBA.

### **Использование выражений для проверки корректности данных**

Выражение "П" Or "У" Or "К" Or "Г" Or "М" Or "Н" Or "О", примененное для проверки поля «Тип Мероприятия» таблицы «Задания», замените на функцию In(), решающую ту же задачу проще:

In ("П"; "У"; "К"; "Г"; "М"; "Н"; "О")

### **Использование выражений в запросах**

При создании запроса «Список рассылки по США» для отбора записей, относящихся к штатам Калифорния, Орегон и Вашингтон, значения ="CA", ="OR" и ="WA" вводились в разных строках. Лучше введите в той же строке, что и критерий для поля «Страна»=«США», выражение: In("CA"; "OR"; "WA").

### **Задание критерия отбора**

Чтобы поэкспериментировать с критериями отбора, сделайте следующее:

1. Создайте новый запрос.
2. Добавьте таблицы «Клиенты», «Заказы» и «Заказано».
3. Добавьте в бланк запроса из таблицы «Клиенты» поля: «Название», «Индекс» и «Страна».
4. Добавьте в бланк запроса из таблицы «Заказы» поля: «Код Заказа», «Дата Размещения», «Дата Исполнения» и «СтоимостьДоставки».

Поместите курсор в строку «Сортировка» поля «Код Заказа» и установите: «По возрастанию».

5. Нажмите кнопку «Групповые операции» на панели инструментов или выберите «Вид, Групповые операции». При этом в бланк запроса будет добавлена строка «Групповая операция». Значение по умолчанию «Группировка» добавлено к ячейке «Групповая операция» для каждого поля запроса.

6. Установите курсор на столбце «СтоимостьДоставки». Нажмите кнопку над строкой «Поле» для выбора столбца «СтоимостьДоставки» и нажмите клавишу <Insert>, теперь новый столбец вставлен.

7. Введите в ячейку «Поле» нового столбца:

Сумма:  $\text{Sum}([\text{Цена}][\text{Количество}](1-[\text{Скидка}]))$

Это выражение вычисляет чистую сумму за вычетом скидки каждого пункта строки в таблице «Заказано» и присваивает столбцу тип поля «Денежный».

8. Поставьте курсор на строку «Групповая операция» нового столбца и выберите из списка «Sum». Опция «Sum» суммирует все позиции каждого заказа из таблицы «Заказы».

Строка «Групповая операция» всех остальных столбцов бланка запроса содержит запись «Группировка». Убедитесь, что флажки в строке «Вывод на экран» установлены, чтобы новые столбцы запроса появлялись при запуске запроса. Во избежание сообщения об ошибке при запуске запроса не вводите данные в строке «Имя таблицы» нового столбца запроса.

9. Нажмите кнопку «Запуск» и просмотрите результат выполнения запроса.

10. Сохраните ваш запрос под именем: "Суммы".

### **Использование «Построителя выражений» для добавления критерия отбора**

После создания и проверки запроса можно применить критерий, чтобы ограничить количество записей в результирующем множестве запроса. Для упрощения процесса добавления условий отбора записей в запрос можно воспользоваться «Построителем выражений» Access. Прочитайте следующее:

1. Перейдите в режим конструктора.

2. Поместите курсор в строке «Условие отбора» поля «Название».

3. Нажмите кнопку построителя выражений «Построить» на панели инструментов для вывода окна «Построитель выражений» или нажмите правую кнопку мыши в строке «Условие отбора» и выберите из контекстного меню команду «Построить».

4. Введите: Like "\*shop\*"

5. Нажмите кнопку ОК для возврата в бланк запроса. В поле, где расположен курсор, «Построитель выражений» помещает построенное выражение.

6. Просмотрите результат выполнения запроса.

7. Сохраните ваш запрос под именем: «Суммы1».

### **Использование выражений для создания вычисляемых полей**

Вычисляемые поля отображают данные, рассчитанные на основе значений других полей из той же строки таблицы запроса. Для создания запроса, содержащего вычисляемые поля, сделайте следующее:

1. Откройте в режиме Конструктора запрос «Суммы». Перейдите к первому чистому столбцу. Используя «Построитель выражений», введите имя столбца с двоеточием после него, а затем выражение:

ПолнаяСтоимостьСДоставкой: [Сумма]+[СтоимостьДоставки]

Если не ввести имя поля и двоеточие, Access по умолчанию предоставит вычисляемое поле с именем «Выражение1».

2. Поместите курсор в ячейку «Групповая операция» вычисляемого поля и выберите «Выражение» из раскрывающегося списка. Если не выбрать «Выражение», запрос раскроет диалоговое окно «Введите значение параметра» или выведет сообщение об ошибке при выполнении.

3. Запустите запрос и просмотрите результат его выполнения, а затем сохраните его.

Примечание: При попытке выполнить запрос, могут появиться сообщения об ошибке. Сообщение об ошибке «Невозможно вычислить выражение» обычно указывает на орфографическую ошибку в наименовании функции или объекта.

В зависимости от работы функции может появиться и диалоговое окно «Введите значение параметра», оно появляется тогда, когда Access не может идентифицировать некоторый объект или вычислить значение выражения. Обычно причиной этого является ошибка при вводе запроса.

Сообщение об ошибке «Неверный тип данных» чаще всего является результатом попытки применить математические или тригонометрические операторы к значениям типа «Текст» или «Дата/Время». Если выражение ссылается на элемент управления, содержащийся в объекте формы или отчета, они должны быть открыты при выполнении функции.

## ЛАБОРАТОРНАЯ РАБОТА № 7 (2 часа)

**Цель:** Создание соединений таблиц

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать внутреннее соединение по нескольким полям,
- создавать внешнее соединение,
- создавать рекурсивное соединение,
- создавать соединения по отношению.

### Внутреннее соединение

Внутреннее соединение (или эквисоединение) обычно используется при создании запросов на выборку. Результирующее множество запроса содержит записи одной таблицы, имеющие совпадающие значения в связанных полях другой. В большинстве случаев соединения основываются на уникальном значении поля первичного ключа в одной таблице и значениями поля внешнего ключа в другой таблице, если таблицы связаны отношением «один-ко-многим». Если в таблице со стороны «многие» записи с заданной величиной отсутствуют,

то соответствующие записи таблицы со стороны «один» в результирующее множество не включаются.

### **Создание внутреннего соединения по нескольким полям**

Допустим, что требуется вывести имена клиентов, у которых совпадают официальный адрес и адрес доставки. Адрес клиента хранится в поле «Адрес» таблицы «Клиенты», а адрес доставки - в поле «АдресПолучателя» таблицы «Заказы». Таким образом, необходимо сопоставить поля «КодКлиента» в обеих таблицах, а также поля «Адрес» и «АдресПолучателя».

Для вывода имен клиентов, официальный адрес и адрес доставки которых совпадают:

1. Создайте новый запрос.
2. Добавьте в запрос таблицы «Клиенты» и «Заказы».
3. Выберите в таблице «Клиенты» поле «Адрес» и перетащите его на поле «АдресПолучателя» таблицы «Заказы». Это приводит к созданию соединения полей «Адрес» и «АдресПолучателя». Линия, обозначающая связь, имеет с двух сторон точки, которые указывают на то, что соединение выполнено между полями, связь которых в схеме данных не задана, их имена не совпадают и они не являются первичными ключами.
4. Поместите поля «Название» и «Адрес» таблицы «Клиенты», а затем поле «АдресПолучателя» таблицы «Заказы» в бланк запроса.
5. Задайте сортировку по возрастанию в столбце «Название» и просмотрите результат выполнения запроса.
6. Вернитесь в конструктор. Чтобы запретить вывод одинаковых строк, необходимо вызвать бланк свойств запроса. Для этого нажмите кнопку «Свойства» на панели инструментов, либо дважды щелкните по пустой области верхней части окна запроса. Если заголовком бланка является строка «Свойства списка полей» или «Свойства поля», щелкните по пустому месту, чтобы вывести свойства запроса. Кроме того, можно щелкнуть правой кнопкой мыши по свободной области верхней части окна запроса и выбрать в контекстном меню команду «Свойства».
7. Измените значение свойства «Уникальные значения» на «Да».
8. Просмотрите результат выполнения запроса. Результирующее множество не содержит повторяющихся записей.
9. Сохраните запрос.

### **Создание внешнего соединения**

Внешние соединения позволяют вывести данные всех записей таблицы, участвующей в соединении, вне зависимости от того, имеются ли соответствующие им записи в связанной таблице. Они могут быть левыми или правыми.

Запрос, в котором участвуют таблицы с левым внешним соединением, выводит все записи таблицы со стороны «один» с уникальным значением первич-

ного ключа вне зависимости оттого, имеются ли соответствующие им записи в таблице со стороны «многие».

И наоборот, запрос, в котором участвуют таблицы с правым внешним соединением, выводит все записи таблицы со стороны «многие» вне зависимости от того, имеются ли соответствующие им записи в таблице со стороны «один».

Чтобы с помощью левого внешнего соединения создать запрос, который обнаруживает отсутствие записей для некоторого сотрудника в таблице «Задания»:

1. Создайте новый запрос и добавьте в него таблицы «Сотрудники» и «Задания».

2. Перетащите поле «КодСотрудника» таблицы «Сотрудники» на поле «КодСотрудника» таблицы «Задания» для создания внутреннего соединения между ними. Access автоматически создает соединение, если только была установлена связь между этими полями при создании таблицы «Задания».

3. Поместите поля «Фамилия» и «Имя» таблицы «Сотрудники», а также поля «Тип мероприятия» и «ДатаПлан» таблицы «Задания» в бланк запроса.

4. Выделите линию, соединяющую поля «КодСотрудника».

5. Выберите команду «Вид, Параметры объединения». Кроме того, можно вывести диалоговое окно «Параметры объединения», дважды щелкнув по тонкому отрезку соединения. При двойном щелчке по любому из двух толстых отрезков линий по краям соединительной линии выводится бланк свойств запроса. В диалоговом окне «Параметры объединения» переключатель «1» задаст обычное внутреннее соединение, переключатель «2» - левое внешнее соединение и переключатель «3» - правое внешнее соединение.

6. Задайте левое внешнее соединение. Нажмите кнопку ОК.

Обратите внимание, что Access добавляет стрелку на конце линии соединения. Обычно в структуре базы данных таблица со стороны «один» располагается слева от таблицы со стороны «многие». Если это так, то тогда стрелка направлена в сторону таблицы «Задания», что указывает на левое внешнее соединение.

7. Просмотрите результирующее множество.

Несколько сотрудников не имеют записей в таблице «Задания», если это не так, то откройте таблицу «Задания» и удалите несколько записей, а затем повторно выполните запрос.

8. Закройте запрос, сохранив его.

### **Создание рекурсивного соединения**

Рекурсивное соединение связывает данные в одной таблице. Создание этого типа соединения выполняется путем добавления в запрос копии таблицы (Access назначает псевдоним для копии) и связывания полей идентичных таблиц.

В качестве примера рассмотрим запрос, в котором выводятся сотрудники, утвердившие мероприятия, за которые они ответственны, что, предположим, запрещено внутренним распорядком компании Борей.

Для создания рекурсивного соединения:

1. Создайте новый запрос и добавьте в него таблицу «Задания».
2. Добавьте в запрос копию таблицы «Задания», повторно нажав кнопку «Добавить» Access присваивает копии имя «Персональные мероприятия 1». Закройте диалоговое окно «Добавление таблицы».
3. Перетащите поле «КодОтветственного» исходной таблицы на поле копии «КодНачальника». Между таблицами возникает соединение.
4. Поместите поля «КодСотрудника» и «КодОтветственного» исходной таблицы, а также поля «КодНачальника» и «ТипМероприятия» таблицы «Задания\_1» в бланк запроса.
5. При использовании рекурсивных соединений требуется задать вывод только уникальных значений. Присвойте в бланке свойств запроса значение «Да» свойству «Уникальные значения».
6. Нажмите кнопку «Запуск» для вывода записей.
7. Сохраните запрос.

В рассматриваемом примере можно добавить в запрос таблицу «Сотрудники», чтобы вывести имя сотрудника. При этом необходимо создать дополнительное соединение между полем «КодСотрудника» исходной таблицы «Задания» и полем «КодСотрудника» таблицы «Сотрудники», а затем перетащить поле «Фамилия» в бланк запроса. Присвойте значение «Да» свойству «Уникальные значения», а затем выполните и сохраните запрос.

### **Создание соединения по отношению**

Соединение по отношению (или тэта-соединение) связывает данные некоторым отношением (это может быть любое отношение, исключая равенство). Кроме того, в окне конструктора запросов соединения по отношению не выводятся, также как они не отображаются в окне схемы данных. Связь, эквивалентную соединению по отношению, можно задать с помощью условия отбора, которое указывается для одного из двух полей, участвующих в соединении.

Поиск клиентов, которые имеют разный официальный адрес и адрес доставки, является примером, в котором требуется использовать соединение по отношению. Для создания связи, эквивалентной данному соединению:

1. Создайте новый запрос и добавьте в него таблицы «Клиенты» и «Заказы».
2. Поместите поля «Название» и «Адрес» таблицы «Клиенты», а также поле «АдресПолучателя» таблицы «Заказы» в бланк запроса.
3. Введите в строке «Условие отбора» столбца поля «АдресПолучателя» выражение:

<>Клиенты.Адрес

Access автоматически заключает имена таблицы и поля в квадратные скобки вне зависимости от того, включают ли они пробелы или знаки препинания.

Если внести в столбце поля «Адрес» выражение:

◁ Заказы. АдресПолучателя

то будет получен идентичный результат.

4. Присвойте в бланке свойств запроса значение «Да» свойству «Уникальные значения».

5. Выполните запрос. Результирующее множество содержит только те записи, в которых различаются адрес клиента и адрес доставки.

6. Сохраните запрос.

### ЛАБОРАТОРНАЯ РАБОТА № 8 (4 часа)

**Тема:** Использование в запросах статических функций и создание перекрестных запросов

Цель лабораторной работы заключается в том, чтобы научиться:

- использовать статистические функции языка SQL,
- создавать перекрестные запросы.

#### Использование статистических функций языка SQL

Access позволяет практически мгновенно обобщить информацию, содержащуюся в записях таблиц и которая является входными данными для большинства информационных систем. В этих системах можно получить ответы на типичные вопросы, такие, например, как «Какова сумма продаж на заданный день?» или «Как изменилась сумма продаж за текущий месяц по сравнению с суммой продаж за тот же месяц прошлого года?». Чтобы выполнить расчеты на основе информации таблиц, необходимо создать запрос, в котором для вычислений используются статистические функции Access SQL (табл. 8.1).

Эти функции называются статистическими, потому что они выполняют сбор статистики об информации в записях. Имеющихся статистических функций SQL, обычно достаточно для решения большинства задач.

Таблица 8.1

Функция	Описание	Типы полей
Avg()	Вычисляет арифметическое среднее набора чисел, содержащихся в указанном поле запроса	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE
Count()	Вычисляет количество непустых записей, возвращаемых запросом	Все типы полей
First()	Возвращает значения поля из первой записи результирующего набора	Все типы полей

Last()	Возвращает значения поля из последней записи результирующего набора	Все типы полей
Max()	Возвращает максимальное из набора значений, содержащихся в указанном поле	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE
Min()	Возвращает минимальное из набора значений, содержащихся в указанном поле	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE
StDev() StDevP()	Возвращают смещенные и не смещенное значение среднеквадратического отклонения, вычисляемого по набору значений, содержащихся в указанном поле.	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE
Sum()	Возвращает сумму набора значений, содержащихся в заданном поле	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE
Var() VarP()	Возвращают значение смещенной и несмещенной дисперсии, вычисляемой по набору значений, содержащихся в указанном поле	Все типы полей, исключая текстовый, Поле Мемо и Поле объекта OLE

### Вычисления на всем диапазоне записей таблицы

Менеджерам, особенно специализирующимся на продажах и маркетинге, обычно требуется информация о заказах и поставках, размещенных и исполненных за определенный промежуток времени, финансовым директорам - обобщенные величины, такие, например, как общая сумма неоплаченных счетов и среднее число дней, прошедших со времени выставления счета до момента оплаты. В некоторых случаях необходимо выполнить расчеты над всеми записями таблицы, например, вычислить за весь период работы фирмы среднюю сумму выставленных счетов. Для выбора данных, которые требуется обобщить, в запросе используется условие отбора.

Таблица «Заказы» базы данных Борей не содержит поля «Итого», которое содержало бы сумму заказа без стоимости перевозки.

Для создания запроса, в котором используются статистические функции SQL для вывода данных о заказах, общего количества, общей, а также средней, наименьшей и наибольшей суммы, требуется добавить поле, которое бы содержало обобщенную информацию о каждом заказе:



1. Создайте новый запрос и добавьте в него таблицы «Заказы» и «Заказано».

2. Поместите поля «КодЗаказа» и «ДатаРазмещения» таблицы «Заказы» в бланк запроса.

3. Введите в ячейку строки «Поле» пустого столбца выражение:

Итог:  $\text{Sum}([\text{Количество}] * [\text{Цена}] * (1 - [\text{Скидка}]))$

которое вычисляет чистую сумму товаров, включенных в поставку. Нажмите на панели инструментов кнопку «Свойства» и присвойте значение «Денежный» свойству поля «Формат поля».

4. Нажмите на панели инструментов кнопку «Групповые операции»

5. Выберите из списка статистических функций строки «Групповая операция» элемент «Выражение»

6. Проверьте запрос и сохраните его под именем «Итоги по заказам».

Примечание: Если ввести в столбце «Итог» выражение:

$\text{Sum}(\text{Sum}([\text{Количество}] * [\text{Цена}] * (1 - [\text{Скидка}])))$

то суммы заказов выводятся в денежном формате

Чтобы использовать статистические функции SQL для поля «Итог» запроса «Итоги по заказам»

1. Создайте новый запрос и добавьте в него запрос «Итоги по заказам».

2. Перетащите поле «КодЗаказа» на первый столбец, а затем четыре раза поле «Итог».

3. Выберите команду «Вид, Групповые операции».

4. Установите курсор в столбец «КодЗаказа» строки «Групповая операция» и выберите из списка функцию Count().

5. Установите курсор в первый столбец «Итог» строки «Групповая операция» и выберите из списка функцию Sum(), для второго столбца «Итог» задайте функцию Avg(), для третьего - функцию Min(), для четвертого - функцию Max().

6. Установите курсор в столбец «КодЗаказа» и нажмите на панели инструментов кнопку «Свойства». Присвойте значение «Счетчик» свойству «Подпись».

7. Для каждого из четырех столбцов «Итог» присвойте значение «Денежный» свойству «Формат поля», а свойству «Подпись» значения «Суммарная выручка», «В среднем за заказ», «Минимум», «Максимум» соответственно. Значение свойства «Формат поля» задавать не требуется, если в запросе «Итоги по заказам» использована функция Sum().

8. Просмотрите результат выполнения запроса. Условие отбора не было указано, поэтому выведенные данные относятся ко всей таблице.

9. Сохраните запрос под именем «Статистика по заказам».

### **Примечание.**

При запуске запроса «Статистика по заказам» Access определяет, что объект «Итоги по заказам» является не таблицей, а запросом. Поэтому перед

выполнением запроса «Статистика по заказам» запускается запрос «Итоги по заказам». Одно из наиболее важных средств Access заключается в том, что можно выполнить запрос, взяв за исходные данные для него результирующее множество другого запроса, который называется вложенным. По крайней мере, теоретически никаких ограничений на глубину вложенности не накладывается, однако при этом для получения необходимого результата требуется больше времени.

### **Вычисления на выделенных записях таблицы**

В предыдущем примере расчеты производились над всеми имеющимися в таблице «Заказы» записями. Однако обычно требуется выполнить вычисления над определенным набором данных и, основываясь на них, составить статистику. Чтобы произвести расчеты только над теми заказами, которые были размещены в марте 1997 года:

1. Сохраните копию запроса «Статистика по заказам» под именем «Статистика за март» и перейдите в конструктор, чтобы задать условие отбора группы записей.

2. Поместите поле «ДатаРазмещения» на столбец «КодЗаказа», чтобы столбец «ДатаРазмещения» стал первым в бланке запроса. Теперь необходимо определить условие отбора заказов на основе даты.

3. Замените элемент «Группировка» в столбце «ДатаРазмещения» на элемент «Условие». Access сбрасывает флажок «Вывод на экран» столбца «ДатаРазмещения». Если установить этот флажок, то при выполнении запроса будет выведено сообщение об ошибке.

4. Введите выражение: Like «\*.03.1997» в ячейку «Условие отбора» столбца «ДатаРазмещения»

5. Просмотрите результат выполнения запроса.

Данные можно обобщить более удобным образом с помощью функции Format(), например, чтобы получить итоговые данные по месяцам (или по кварталам). Для вывода информации о продажах по каждому месяцу 1997 года, последнему году, для которого в базе Борея имеются данные по всем месяцам:

1. Сохраните копию запроса «Статистика по заказам» под именем «Ежемесячная статистика по заказам» и перейдите в конструктор, а затем выделите столбец «ДатаРазмещения». Нажмите клавишу <Insert> для добавления в запрос пустого столбца

2. Введите выражение:

Месяц: Format([ДатаРазмещения]; "mm-yy") в пустой столбец строки «Поле».

Формат "mm-yy" используется для упорядочивания записей по дате. Кроме того, можно применять формат "m" или "mm", но не "mmm", потому что в последнем случае столбцы упорядочиваются в алфавитном порядке, начиная с августа.

3. Введите выражение: Like «\*.\*.1997» в ячейку «Условие отбора» столбца «ДатаРазмещения».
4. Просмотрите результирующее множество.

### **Добавление параметра в запрос «Ежемесячная статистика по заказам»**

Чтобы преобразовать запрос «Ежемесячная статистика по заказам» в запрос с параметром:

1. Откройте в режиме конструктора запрос «Ежемесячная статистика по заказам».
2. Установите курсор в ячейку «Поле» столбца «Месяц», нажмите клавишу <F2> для выделения содержимого ячейки, а затем поместите выделенное выражение в Буфер обмена. Удалите столбец «Месяц».
3. Установите курсор в ячейку «Поле» столбца «ДатаРазмещения», нажмите клавишу <F2>, а затем заменить его на выражение, содержащееся в Буфере обмена.
4. Установите курсор в ячейку «Условие отбора» столбца «ДатаРазмещения» и замените выражение Like «\*.\*.1997» строкой:  
[Введите месяц и год в формате ММ-ГГ:]
5. Нажмите кнопку «Запуск». В появляющемся диалоговом окне «Введите значение параметра» введите строку: 03-97.
6. Нажмите кнопку ОК для вывода результирующего множества.
7. Сохраните запрос под именем «Ежемесячная статистика с параметром».

### **Определение типа данных параметра**

Все параметры по умолчанию имеют текстовый тип. Если условие отбора относится к данным с типом «Дата/Время» или «Числовой», то необходимо определить тип данных для каждого такого аргумента, используя диалоговое окно «Параметры запроса». Для задания типа данных параметра:

1. Выделите строку «Введите месяц и год в формате ММ-ГГ:» в ячейке «Условие отбора» столбца «ДатаРазмещения» (не включайте квадратные скобки) и скопируйте текст в Буфер обмена.
2. Выберите команду «Запрос, Параметры» для вывода диалогового окна «Параметры запроса».
3. Вставьте текст в столбец «Параметр». Строки в столбце «Параметр» и в ячейке «Условие отбора» должны совпадать в точности, что обеспечивается использованием Буфера обмена. Не указывайте в столбце «Параметр» квадратные скобки!
4. Нажмите клавишу Tab для перемещения к столбцу «Тип данных» и выберите из списка элемент «Дата/время». Нажмите кнопку ОК для закрытия диалогового окна

### **Примечание.**

Задание параметров рекомендуется выполнять по завершении разработки и отладки запроса. Фиксированный критерий отбора облегчает и ускоряет проверку, т.к. при запуске не требуется вводить один или несколько аргументов. Закончив отладку, добавьте параметры.

### **Создание перекрестных запросов**

Перекрестные запросы - это запросы, в которых вычисляются статистические данные и определяется их внешний вид. С помощью рассматриваемого типа запросов можно:

1. Указать поле, которое является заголовком строки, используя операцию «Группировка».
2. Задать поле или поля, которые являются заголовками столбцов, а также условие, определяющее значения в столбцах.
3. Указать место размещения данных в таблице.

Перекрестные запросы обладают следующими достоинствами:

1. Возможностью вывода значительного объема данных в компактном и привычном виде.
2. Итоговые данные представляется в формате, который идеально подходит для автоматического создания графиков и диаграмм с помощью Мастера диаграмм Access.
3. Простота и скорость разработки запросов с несколькими уровнями детализации. Высоко структурированные запросы позволяют, например, посмотреть продажи для конкретного товара, нажав кнопку «Промежуточные итоги».

При использовании перекрестных запросов имеется только одно ограничение: нельзя сортировать таблицу результатов по значениям, содержащимся в столбцах. Таким образом, например, невозможно создать запрос, в котором товары упорядочены в строках по объему продаж: в подавляющем большинстве случаев одновременное упорядочивание данных в столбцах по всем строкам невозможно. Однако можно задать сортировку по возрастанию или по убыванию, либо отсутствие упорядочивания для заголовков строк.

### **Создание перекрестного запроса «Ежемесячная выручка от продаж»**

Для создания перекрестного запроса, в котором в строках выводятся товары, а в столбцах соответствующие им месячные объемы продаж:

1. Создайте новый запрос и добавьте в него таблицы «Товары», «Заказано» и «Заказы».
2. Поместите поля «КодТовара» и «Марка» таблицы «Товары», а затем поле «атаРазмещения» таблицы «Заказы» в бланк запросов.
3. Выберите команду «Запрос, Перекрестный». Заголовок окна запроса Запрос1: на выборку изменится на Запрос1: перекрестный запрос. Кроме того, в бланк запроса будет добавлена строка «Перекрестная таблица».

4. Выберите в списке ячейки «Перекрестная таблица» столбца «КодТовара» значение «Заголовки строк». Выполните то же самое для столбца «Марка». Эти столбцы являются требуемыми заголовками строк перекрестной таблицы.

5. Выберите в списке ячейки «Групповая операция» столбца «ДатаРазмещения» значение «Условие». В ячейке «Условие отбора» этого столбца введите выражение Like "\*.\*.1997" для вывода в перекрестной таблице данных за каждый месяц 1997 года

6. Установите курсор в ячейку «Поле» пустого столбца и введите следующее выражение:

Объем продаж: Sum([Заказано].[Количество]\*[Заказано].[Цена])

Выберите в ячейке «Групповая операция» того же столбца значение «Выражение», а затем в ячейке «Перекрестная таблица» значение «Значение» В столбце «Объем продаж» вычисляется общий объем заказов на каждый товар, который подставляется в ячейки перекрестной таблицы. Ссылка на таблицу «Заказано» обязательна, иначе будет выдано сообщение об ошибке «Неоднозначная ссылка на поле».

7. Установите курсор в ячейку «Поле» пустого столбца и введите выражение:

Format([ДатаРазмещения]; «mmm»)

Access добавляет имя поля «Выражение1», применяемое по умолчанию. Не меняйте его, т.к. функция Format() создаст при запуске запроса имена столбцов, состоящие из трех первых букв названия месяца (формат "mmm"). Выберите из списка в ячейке «Перекрестная таблица» значение «Заголовки столбцов»

8. Просмотрите результирующее множество и сохраните запрос под именем «Ежемесячная выручка от продаж».

Обратите внимание на то, что выведенная перекрестная таблица обладает одним немаловажным недостатком: расположение столбцов определяется алфавитным порядком их заголовков, а не хронологической очередностью.

В этом случае требуется использовать фиксированные заголовки столбцов.

### **Использование фиксированных заголовков столбцов в перекрестных запросах**

В Access можно задать порядок сортировки заголовков строк или столбцов перекрестной таблицы, таким образом, определив ее внешний вид. При использовании названий месяцев столбцы упорядочиваются в алфавитном порядке, а не в хронологической последовательности. Чтобы все-таки вывести столбцы в порядке календарной очередности месяцев, необходимо задать фиксированные заголовки. Для соответствующего изменения перекрестного запроса:

1. Сохраните запрос «Ежемесячная выручка от продаж» под именем «Ежемесячная выручка от продаж2», переключитесь в режим конструктора и

выведите свойства запроса. В бланке свойств появляется свойство «Заголовки столбцов», которое имеется только у перекрестных запросов.

2. Введите краткие названия каждого из месяцев года в поле свойства «Заголовки столбцов»:

"Янв"; "Фев"; "Мар"; "Апр"; "Май"; и т.д.

Необходимо воспроизвести названия без ошибок, в противном случае данные для неправильно заданных месяцев не отображаются. Сокращения требуется заключить в кавычки и разделять названия точками с запятыми. Пробелы не нужны. По завершении ввода всех 12 месяцев закройте бланк свойств.

3. Нажмите кнопку «Запуск» теперь заголовки столбцов перекрестной таблицы соответствуют календарной последовательности.

### **Понижение уровня детализации в перекрестных запросах**

Созданный перекрестный запрос выводит данные по всем товарам, по которым были размещены заказы в одном из месяцев 1997 года. Однако управленческому аппарату более высокого уровня обычно требуется информация в виде графика или диаграммы, с помощью которых можно проследить некоторую тенденцию. Таким образом, необходимо сократить число строк и столбцов, так чтобы создать на основе значений перекрестной таблицы удобный для анализа график.

Для создания перекрестного запроса, который выводит ежеквартальную выручку от продаж по типам товаров (а не по значению поля «КодЗаказа»):

1. Сохраните запрос под именем «Ежеквартальная выручка от продаж товаров по типам».

2. Добавьте в запрос таблицу «Типы».

3. Поместите поля «КодТипа» и «Категория» таблицы «Типы» на столбец «КодТовара». В ячейках «Перекрестная таблица» обоих столбцов выберите значение «Заголовки строк».

4. Удалите столбцы «КодТовара» и «Марка».

5. Введите в столбец «Выражение1» выражение:

Format ([ДатаРазмещения]; ""Квартал ""q")

которое определяет заголовки столбцов «Квартал 1»,... «Квартал 4». Несколько кавычек требуется для указания того, что слово Квартал и пробел рассматриваются как строка, а q - как формат.

6. Откройте бланк свойств и очистите содержимое свойства «Заголовки столбцов», а затем закройте бланк. Если не удалить фиксированные заголовки, то в перекрестной таблице не будет данных. Чтобы ускорить работу запроса, можно задать в поле свойства «Заголовки столбцов» четыре фиксированных названия: "Квартал 1",..., "Квартал 4".

7. Просмотрите результат выполнения запроса.

## **ЛАБОРАТОРНАЯ РАБОТА № 9 (4 часа)**

**Тема:** Создание запросов на создание таблиц; на добавление, удаление, обновление данных.

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать запросы на добавление данных
- использовать запрос для удаления записей
- использовать запрос для обновления данных
- использовать каскадное удаление.
- использовать каскадное обновление

### **Использование запросов на изменение**

Запросы на изменение позволяют создать новые таблицы и обновить данные в имеющихся.

В Access имеются следующие запросы на изменение:

1. Запросы на добавление позволяют добавить в таблицу создаваемые запросом записи.
2. Запросы на удаление удаляют из таблицы записи, которые соответствуют данным результирующего множества запроса.
3. Запросы на обновление изменяют значения существующих полей таблицы в соответствии с записями результирующего множества.

### **Создание новой таблицы при помощи запроса**

Будем создавать таблицу «Адреса доставки», содержащую информацию о клиентах, которые имеют разные почтовый адрес и адрес доставки. При этом требуется удалить адреса доставки, которые для большинства записей таблицы «Заказы» совпадают с адресом клиента в таблице «Клиенты». Удаление повторяющихся данных и копирование их в новые таблицы - важный этап в преобразовании информации из двумерной (не реляционной) базы данных в реляционную базу. Однако самостоятельное удаление повторяющейся информации лучший способ освоить разработку запросов на создание таблицы.

#### **Внимание!**

Перед запуском запроса на изменение обязательно создайте резервную копию таблицы, которую требуется обновить. Исправления, внесенные запросами на изменение, необратимы, а ошибка может привести к тому, что таблицу будет нельзя использовать. Более того, обнаружить записи, измененные запросом, в котором имеется ошибка, в большинстве случаев весьма непросто.

### **Разработка запроса на выборку данных**

Для создания на основе таблицы «Заказы» запроса на выборку, записи которого в последующем используются для заполнения новой таблицы:

1. Создайте новый запрос и добавьте в него таблицы «Клиенты» и «Заказы».

2. Поместите поле «КодКлиента» таблицы «Клиенты» в бланк запроса.

3. Поместите поля «НазваниеПолучателя», «АдресПолучателя», «ГородПолучателя», «ОбластьПолучателя», «ИндексПолучателя» и «СтранаПолучателя» таблицы «Заказы» в бланк запроса.

Теперь требуется задать условие отбора только тех записей таблицы «Заказы», в которых значение поля «НазваниеПолучателя» не равно значению поля «Название» таблицы «Клиенты» или значение поля «АдресПолучателя» не равно значению поля «Адрес».

4. Введите в ячейку «Условие отбора» столбца «НазваниеПолучателя» выражение:

<>[Клиенты].[Название]

5. Введите в ячейку «Или» (OR) столбца «АдресПолучателя» выражение:

<>[Клиенты].[Адрес]

Данное выражение следует указать в строке «Или» (Or), а выражение, заданное на шаге 4, - в строке «Условие отбора», чтобы при образовании общего условия отбора Access объединил оба высказывания оператором Or.

Примечание: Правильнее было бы задать дополнительное условие отбора для проверки неравенства названий городов, областей, индексов и стран. Однако наличие у клиента одного адреса в двух разных городах весьма маловероятно.

6. Откройте бланк свойств запроса и присвойте значение «Да» свойству «Уникальные значения».

7. Просмотрите результат запроса и вернитесь в режим конструктора.

### **Преобразование запроса на выборку в запрос на создание таблицы**

Теперь, убедившись в правильности работы запроса на выборку, для создания на основе его результирующего множества таблицы:

1. Выберите команду «Запрос, Создание таблицы». Появляется диалоговое окно «Создание таблицы». Введите значение: «Адреса доставки» в поле «Имя таблицы». Нажмите кнопку ОК.

2. Нажмите кнопку «Запуск». Выводится приглашение подтвердить добавление указанного количества записей в новую таблицу. Нажмите кнопку «Да» для создания таблицы «Адреса доставки».

3. Сохраните запрос на создание таблицы под именем: «Создание таблицы адресов доставки».

4. Нажмите кнопку «Окно базы данных», раскройте вкладку «Таблицы» и откройте таблицу «Адреса доставки».

Доработайте структуру таблицы «Адреса доставки»:

1. Нажмите кнопку "Конструктор". В новой таблице поле «КодКлиента», скопированное из таблицы «Клиенты», не является первичным ключом.



2. В бланке свойств таблицы в поле «Описание» введите значение: Адреса доставки, отличающиеся от адресов клиентов.

Закройте окно «Свойства».

Одному адресу в таблице «Адреса доставки» соответствует один клиент в таблице «Клиенты». Однако у клиента может быть несколько адресов доставки, поэтому следует связать таблицы «Клиенты» и «Адреса доставки» отношением «один-ко-многим» и разрешить совпадение значений поля «КодКлиента» таблицы «Адреса доставки». При этом первичный ключ таблицы «Адреса доставки», не может быть определен на основе трех полей: «КодКлиента», «НазваниеПолучателя» и «АдресПолучателя». В данном конкретном примере первичный ключ не используется. Присвойте значение «Да (Совпадения допускаются)» свойству «Индексированное поле» поля «КодКлиента». Индексирование ускоряет работу запроса, если клиенты имеют несколько адресов доставки.

4. Присвойте значение «Да» свойству «Обязательное поле» полей «КодКлиента», «НазваниеПолучателя», «АдресПолучателя», «ГородПолучателя» и «СтранаПолучателя», т.к. эти поля обязательны. Поля «ОбластьПолучателя» и «ИндексПолучателя» могут не указываться, т.к. в одних странах нет деления на области, в других не используются почтовые индексы.

5. Установите связи между таблицами «Клиенты» и «Адреса доставки».

### **Создание запросов на добавление данных**

Запрос на создание таблицы создает структуру новой таблицы на основе записей запроса. В таблицу добавляются только те поля, которые участвуют в запросе. Если разработать и сохранить таблицу «Адреса доставки» перед извлечением повторяющихся данных из таблицы «Заказы», то можно использовать для занесения информации в новую таблицу запрос на добавление.

Кроме этого, запрос на добавление позволяет удалить повторяющиеся данные из исходной таблицы. В этом случае после применения запроса на создание таблицы для построения связанных таблиц измените в режиме конструктора тип запроса, выбрав команду «Запрос, Добавление» или команду «Запрос, Удаление», либо щелкнув по кнопке «Тип запроса» и выбрав из списка требуемый тип.

Запрос на добавление данных отличается от запроса на создание таблицы еще и тем, что он может включать меньше полей, чем таблица, в которую заносится информация. В остальном эти типы запросов полностью идентичны.

### **Добавление данных в таблицу «Адреса доставки»**

1. Откройте таблицу «Адреса доставки» в режиме таблицы, выберите команду «Правка, Выделить все записи», а затем нажмите клавишу <Delete> для удаления всех записей из таблицы. Нажмите кнопку «Да» после вывода приглашения на подтверждение удаления.

2. Откройте запрос «Создание таблицы адресов доставки» в режиме конструктора.

3. Выберите команду «Запрос, Добавление». Появится диалоговое окно «Добавление», которое по умолчанию содержит таблицу «Адреса доставки» в раскрывающемся списке «Имя таблицы». Нажмите кнопку ОК для закрытия диалогового окна.

Примечание: Для добавления данных необходимо, чтобы имена полей запроса и полей таблицы, в которую заносится информация, были идентичны, либо требуется указать в столбце бланка запроса поле, используемое для добавления. Если в именах имеется различие, даже вызванное лишним пробелом, то Access не добавляет данные. В бланке запроса на добавление имеется дополнительная строка «Добавление», в которой указываются названия полей таблицы. По умолчанию в этой строке содержатся имена полей запроса и таблицы, которым Access находит соответствие. При необходимости следует изменить значение в ячейке строки «Добавление».

4. Выполните запрос на добавление. В окне сообщения выводится число записей, которые будут занесены в таблицу. Нажмите кнопку «Да» для добавления записей.

5. Откройте таблицу «Адреса доставки» для проверки правильности введенных данных.

Записи, в которых содержатся данные с повторяющимися значениями ключевых полей, добавить нельзя. При попытке добавления выводится сообщение, в котором содержится число недопустимых записей. Однако в отличие от операции «Добавить из буфера» Access не создает таблицу «Ошибки вставки», которая включает неприсоединенные записи.

После добавления записей в существующую таблицу невозможно создать ее первичный ключ. Свойство «Уникальные записи», заданное в бланке свойств запроса, относится только к запросу, а не к таблице, в которую заносятся записи. Чтобы обеспечить возможность добавления повторяющихся данных в таблицу «Адреса доставки», необходимо предварительно задать составной первичный ключ, совпадения значений которого не допускается, а уже затем присоединить записи.

### **Использование запроса для удаления записей**

Перед удалением записей рекомендуется вывести их, используя запрос на выборку, а затем, преобразовав этот запрос в запрос на удаление, выполнить его для того, чтобы уничтожить данные.

Удалять записи из таблицы приходится довольно часто. Например: необходимо уничтожить данные об отмененных заказах или информацию о клиентах, которые не приобретали товаров в течение последних нескольких лет. Удаление записей из таблицы с помощью запроса на удаление является операцией, обратной добавлению. Создайте запрос на выборку и включите в него все поля,

а затем добавьте поля, определяющие критерий отбора уничтожаемых данных. Если не задать условие, то, преобразовав запрос на выборку в запрос на удаление и выполнив его, можно удалить все записи используемой таблицы.

Чтобы освоить операцию удаления записей (фактически не удаляя их), предположим, что имеется информация о том, что австрийские власти объявили фирму ERNST HANDEL (поле «КодКлиента» имеет значение ERNSH) банкротом, и требуется отменить поставку и удалить информацию обо всех размещенных, но еще не исполненных заказах этой фирмы.

Для создания запроса, который отбирает записи о таких заказах:

1. Создайте новый запрос и добавьте в него таблицу «Заказы».
2. Поместите из списка полей элемент \* в строку бланка запроса «Поле».
3. Поместите на бланк запроса поле «КодКлиента». Данное поле требуется для отбора записей. Поля, входящие в запрос, необходимо выводить только один раз, поэтому требуется скрыть поле «КодКлиента», сбросив флажок «Вывод на экран», т.к. оно уже включено в первый столбец бланка запроса.

4. В ячейке «Условие отбора» столбца «КодКлиента» введите код фирмы Ernst Handel - ERNSH.

5. Пустое значение в поле «ДатаИсполнения» указывает на неисполненные заказы. Перетащите это поле на бланк запроса. Скройте его, чтобы повторно не выводить поле «ДатаИсполнения».

6. Введите в ячейку «Условие отбора» столбца «ДатаИсполнения» выражение: Is Null. Чтобы случайно не удалить все записи о фирме Ernst Handel, а также все записи с отсутствующей датой исполнения, необходимо разместить оба критерия отбора в строке «Условие отбора».

7. Запустите запрос на выборку для вывода записей, которые будут удалены после преобразования запроса в запрос на удаление.

#### **Для преобразования запроса на выборку в запрос на удаление**

1. Не закрывая запрос, нажмите на кнопку «Окно базы данных» и создайте копию таблицы «Заказы» (назвав ее «Заказы (копия)») и копию таблицы «Заказано» (назвав ее «Заказано (копия)»).

2. Активизируйте запрос на выборку в режиме конструктора. Выберите команду «Запрос, Удаление». Строки «Сортировка» и «Вывод на экран» исчезают, появляется строка «Удаление». Значение «Из» в первой ячейке строки «Удаление» вместе с выражением, заданным в ячейке «Поле», указывают, что требуется удалить записи из таблицы «Заказы». Два оставшихся столбца со значением «Условие» в строке «Удаление» определяют критерий отбора записей для удаления.

3. Запустите запрос. Выводится приглашение подтвердить уничтожение записей. Нажмите кнопку «Нет» для отмены удаления.

Примечание: Удаление записей в таблице со стороны «один», которым соответствуют записи в таблице со стороны «многие», нарушают условие це-

лостности данных. Связанные записи в таблице со стороны «многие» становятся висячими. В рассматриваемом примере целостность данных обеспечивается связью между таблицами «Заказы» и «Заказано», что предотвращает удаление информации из таблицы «Заказы». Для уничтожения данных необходимо использовать каскадное удаление. Сначала требуется удалить записи из таблицы «Заказано», а уже затем из таблицы «Заказы». Связь между таблицами «Заказы» и «Заказано» допускает каскадное удаление. Однако если сбросить флажок «Каскадное удаление связанных записей», то при попытке удалить информацию из таблицы «Заказы» выводится сообщение об ошибке.

При случайном удалении записей о фирме Ernst Handel восстановите таблицы «Заказы» и «Заказано», переименовав резервные копии.

### **Использование запроса для обновления данных**

Запросы на обновление позволяют изменить данные в таблице. Они используются для обновления значений полей с помощью одного общего выражения. Например, может понадобиться увеличить или уменьшить цену всех товаров, либо изменить стоимость товаров, принадлежащих к определенному типу, на фиксированный процент.

После создания новой таблицы с помощью запроса необходимо провести некоторые подготовительные мероприятия, перед тем как ее использовать:

1. Создать в таблицах «Заказы» и «Адреса доставки» числовое (длинное целое - Longinteger) поле «КодПолучателя». Значение 0 в поле «КодПолучателя» таблицы «Заказы» указывает на то, что адрес клиента и адрес доставки совпадают. Другому значению соответствует некоторый адрес доставки. В рассматриваемом случае поле «КодПолучателя» имеет значение 1 для всех записей таблицы «Адреса доставки». Добавив поле «КодПолучателя» в таблицу «Адреса доставки», можно создать составной первичный ключ на основе этого поля и поля «КодКлиента». Если задать отсутствие совпадений значений для составного ключа, то можно предотвратить случайное повторение значения поля «КодПолучателя».

2. Не удалять поля, которые содержат повторяющиеся данные, извлеченные в новую таблицу, не убедившись, что они верны, а также не изменяйте запросы, формы и отчеты. Проверив правильность информации, можно уничтожить поля.

3. Добавить новую таблицу в запросы, формы, отчеты, макросы и модули, в которых требуется извлеченная информация.

4. Задать в требуемых объектах базы данных ссылки на поля новой таблицы.

Перед удалением информации необходимо проверить правильность извлеченных данных. Кроме того, перед удалением настоятельно рекомендуется создать резервную копию исходной таблицы.

Прежде чем работать с запросом на обновление, требуется провести описанные выше подготовительные мероприятия с таблицей «Адреса доставки». Создание этой таблицы обсуждалось в разделе «Создание новой таблицы при помощи запроса». Кроме того, нужно добавить в таблицы «Заказы» и «Адреса доставки» поле «КодПолучателя». Для этого:

1. Откройте таблицу «Заказы (копия)» в режиме конструктора, предварительно создав резервную копию таблицы «Заказы», если это еще не сделано.

2. Щелкните по области выделения строки «НазваниеПолучателя», а затем нажмите клавишу <Insert> для добавления пустой строки между полями «СтоимостьДоставки» и «НазваниеПолучателя». Access вставляет строку над выбранным полем.

3. Введите в столбце «Имя поля» имя: КодПолучателя, выберите для нового поля числовой тип, а в качестве его размера задайте значение «ДлинноеЦелое». Присвойте значение «Да» свойству «Обязательное поле».

4. Закройте таблицу «Заказы (копия)», сохранив внесенные изменения. Условия целостности данных изменились, когда было задано значение свойства «Обязательное поле», поэтому выводится сообщение об этом. Нажмите кнопку «Нет» для отмены проверки, которая все равно не закончится успешно, т.к. поле «КодПолучателя» не содержит данных.

5. Откройте таблицу «Адреса доставки» в режиме таблицы.

6. Щелкните по кнопке выделения столбца «НазваниеПолучателя» и выберите команду «Вставка, Столбец» для вставки поля «Поле1» между столбцами «КодКлиента» и «НазваниеПолучателя».

7. Введите значение 1 в поле «Поле1» всех записей таблицы «Адреса доставки».

8. Переключитесь в режим конструктора и измените название поля «Поле1» на имя «КодПолучателя». Access определяет на основе введенных значений числовой тип данных поля, и присваивает значение «Длинное целое» свойству «Размер поля».

9. Выделите поля «КодКлиента» и «КодПолучателя», щелкнув по кнопке выделения одной из строк и перетащив указатель мыши на другую.

10. Создайте составной первичный ключ на основе полей «КодКлиента» и «КодПолучателя», а затем закройте таблицу «Адреса доставки», выполнив ее проверку.

Теперь необходимо создать запрос на выборку заказов, в которых требуется указать для поля «КодПолучателя» значение 1, определяющее использование данных таблицы «Адреса доставки». Данный запрос очень похож на тот, который применялся при построении таблицы «Адреса доставки» ранее.

Для разработки запроса на обновление:

1. Создайте новый запрос и добавьте в него таблицы «Клиенты» и «Заказы (копия)».

2. Поместите в бланк запроса поля «Название» и «Адрес» таблицы «Клиенты», а затем поля «НазваниеПолучателя» и «АдресПолучателя» таблицы «Заказы (копия)».

3. Введите в ячейку «Условие отбора» столбца «НазваниеПолучателя» выражение:

<>[Клиенты].[Название]

а в столбец «АдресПолучателя» в ячейку «Или» выражение:

<>[Клиенты].[Адрес].

4. Запустите запрос и проверьте правильность отобранного набора записей. Если имя фирмы Let's Stop N Shop было изменено на название Let's Stop 'N Shop, то в результирующем множестве выводится пять записей о заказах, размещенных этой компанией. Кроме того, в таблицах «Клиенты» и «Заказы» при вводе данных допущена ошибка в названии фирмы Alfred's Futterkiste. Для обнаружения такого рода ошибок и предназначен разработанный запрос на выборку. Однако даже если указанные записи и отображаются, можно внести данные об этих компаниях в таблицу «Адреса доставки». Резервные копии таблиц имеются, поэтому в рассматриваемом случае можно либо сразу перейти к созданию запроса на обновление, либо сначала исправить ошибки в результирующем множестве.

Убедившись в правильности отобранного набора записей таблицы «Заказы (копия)», можно преобразовать запрос на выборку в запрос на обновление. Для этого:

1. Возвратитесь в режим конструктора и перетащите поле «КодПолучателя» таблицы «Заказы (копия)» в первый столбец бланка запроса.

2. Выберите команду «Запрос, Обновление». Строки «Сортировка» и «Вывод на экран» исчезают, появляется строка «Обновление».

3. Введите в ячейку «Обновление» столбца «КодПолучателя» значение 1, которое обозначает ссылку на запись таблицы «Адреса доставки». Пустые значения в остальных ячейках строки «Обновление» указывают на то, что значения в этих полях изменять не требуется.

4. Запустите запрос на обновление. Выводится сообщение, в котором отображается число обновляемых записей.

5. Откройте таблицу «Заказы (копия)», проверьте записи и убедитесь, что в требуемых строках значение поля «КодПолучателя» равно 1.

6. Чтобы задать значение 0 для заказов, в которых адрес клиента и адрес доставки совпадают, закройте запрос на обновление, создайте новый и добавьте в него только таблицу «Заказы (копия)».

7. Перетащите поле «КодПолучателя» на первый столбец бланка запроса и выберите команду «Запрос, Обновление».

8. Введите значение 0 в ячейку «Обновление», а в ячейку «Условие отбора» - выражение Is Null. Затем нажмите кнопку «Запуск», чтобы заменить все пустые значения в поле «КодПолучателя» на 0.

### **Использование каскадного удаления и обновления данных**

При уничтожении записей в главной таблице автоматически удаляются связанные записи в подчиненной таблице. Кроме того, если изменить значение поля первичного ключа записи, то при каскадном обновлении внешнему ключу связанных записей будет присвоено новое значение первичного ключа.

При каскадном удалении и обновлении записей Access автоматически выполняет специальные запросы.

### **Создание тестовых таблиц и установка связей между ними**

Освоение средств баз данных следует выполнять с помощью копий, а не исходных таблиц, для создания которых:

1. Создайте копии таблиц «Заказы» и «Заказано».
2. Тестовым таблицам дайте имена: «Заказы (копия 2)» и «Заказано (копия)».
3. Откройте таблицу «Заказы (копия 2)» в режиме конструктора и для поля «КодЗаказа» установите тип данных «Числовой». Это необходимо для выполнения каскадного обновления.

Для выполнения каскадного удаления и каскадного обновления требуется установить связь между главной и подчиненной таблицей, а также задать обеспечение целостности данных. Чтобы определить каскадное изменение:

1. Выведите окно «Схема данных».
2. Нажмите кнопку «Очистить макет».
3. Добавьте таблицы «Заказы (копия 2)» и «Заказано (копия 2)».
4. Перетащите поле «КодЗаказа» таблицы «Заказы (копия 2)» в поле «КодЗаказа» таблицы «Заказано (копия)» для связывания таблиц отношением «один-ко-многим».
5. В окне «Связи» установите флажок «Обеспечение целостности данных».
6. Установите флажки «Каскадное обновление связанных полей» и «Каскадное удаление связанных записей».
7. Создайте связи и закройте окно «Схема данных».

### **Использование каскадного удаления**

Для выполнения каскадного удаления в тестовых таблицах:

1. Откройте таблицы «Заказы (копия 2)» и «Заказано (копия 2)» в режиме таблицы и разместите так, чтобы были видны обе таблицы (измените их размеры).
2. Щелкните по окну таблицы «Заказы (копия 2)», а затем по области выделения строки для выбора удаляемой из таблицы записи.
3. Нажмите клавишу <Delete>, чтобы удалить запись и связанные с ней данные в таблице «Заказано (копия 2)».

4. Нажмите кнопку «Да» при выводе приглашения подтвердить удаление.

Чтобы убедиться в том, что записи действительно удалены, просмотрите таблицу «Заказано (копия 2)». Если эта таблица была открыта на шаге 1, то в ячейках удаленных записей выводится значение #Удалено.

### **Использование каскадного обновления**

Для выполнения каскадного обновления в тестовых таблицах:

1. Откройте таблицы «Заказы (копия 2)» и «Заказано (копия 2)» в режиме таблицы и разместите так, чтобы были видны обе таблицы.

2. Введите в поле «КодЗаказа» первой записи таблицы «Заказы (копия 2)» код 20000.

3. Установите курсор на другой записи для выполнения обновления. Изменения отражаются в таблице «Заказано (копия 2)».

При выполнении каскадного обновления приглашение подтвердить изменения не выводится, потому что эта операция допускает отмену (команда «Правка, Восстановить текущую запись»).

## **ЛАБОРАТОРНАЯ РАБОТА № 10 (4 часа)**

**Тема:** Работа с формами

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать формы при помощи Мастера форм
- пользоваться конструктором форм.

### **Создание и использование форм**

Формы Access позволяют создавать пользовательский интерфейс для таблиц базы данных. Хотя для выполнения тех же самых функций можно использовать режим таблицы, формы предоставляют преимущества для представления данных в упорядоченном и привлекательном виде. Правильно разработанная форма ускоряет процесс ввода данных и минимизирует ошибки.

Формы создаются из набора отдельных элементов, называемых элементами управления или управляющими объектами. Элементы управления - это компоненты, в которых используются текстовые поля для ввода и редактирования данных, кнопки, флажки, переключатели, списки, метки полей, а также рамки объектов для отображения графики и объектов OLE. Форма состоит из окна, в котором размещаются два типа элементов управления: динамические (отображающие данные из таблиц) и статические (отображающие статические данные, такие, как метки и логотипы).

### **Типы форм Access.**

Содержание и вид формы зависит от того, в каком приложении для работы с базами данных она используется. Приложения для работы с базами данных можно разделить на три основных категории:



### 1. Управление транзакциями.

Такие приложения выполняют функции добавления новых записей в таблицы или изменения существующих записей. Приложения для управления транзакциями требуют наличия доступа «Для записи» к таблицам базы данных, которые присоединены к форме.

### 2. Доступ к данным.

Приложения, входящие в эту категорию, предназначены для представления такой информации, как диаграммы, отчеты, статистические сведения, таблицы или отдельные элементы данных, но не позволяют пользователю добавлять или редактировать данные. Приложения для доступа к данным требуют наличия доступа «Для чтения» к таблицам базы данных, которые присоединены к форме.

### 3. Администрирование баз данных.

Приложения этой категории выполняют административные функции такие, как создание баз данных или таблиц, разграничение прав доступа пользователей к объектам баз данных, обеспечение безопасности с помощью шифрования, периодическое уплотнение базы данных, а также операции резервного копирования. Приложения для администрирования баз данных требуют наличия полного доступа ко всем объектам, содержащимся в базе данных.

Формы являются ключевыми элементами в приложениях для управления транзакциями и для доступа к данным.

## **Создание формы для управления транзакциями при помощи Мастера форм**

Создадим форму «Задания», которая будет обеспечивать добавление новых записей в таблицу «Задания», возможность редактирования существующих записей и будет иметь подчиненную форму, которая нужна для отображения всех предыдущих персональных мероприятий для конкретного сотрудника.

Форма «Задания» будет состоять из форм двух типов: формы для управления транзакциями и формы для доступа к данным. Разработка такой формы может осуществляться двумя способами:

1. Использовать таблицу «Сотрудники» в качестве источника данных для главной формы и использовать подчиненную форму для отображения, добавления и редактирования записей в таблице «Задания». Этот метод позволяет добавить новую запись как в таблицу «Сотрудники», так и в таблицу «Задания».

2. Использовать таблицу «Задания» как источник данных и для главной, и для подчиненной формы.

Простейший путь создания основной и подчиненной форм - использование Мастера форм, который позволяет создавать формы (с подчиненными или без), содержащие поля из одной или более таблиц или запросов. Мастер форм создает базовый внешний вид формы и добавляет текстовые поля для отображения и редактирования значений полей таблиц. Независимо от того, являетесь ли

вы профессиональным разработчиком приложений или начинающим пользователем, использование Мастера форм заметно упрощает и ускоряет процесс создания простых форм, которые затем можно усовершенствовать в режиме конструктора.

Чтобы создать форму «Задания» при помощи Мастера форм:

1. Раскройте вкладку «Формы» окна базы данных, нажмите кнопку «Создать». Появится диалоговое окно «Новая форма».

2. В списке диалогового окна выделите элемент «Мастер форм». Элемент «Конструктор» позволяет открыть новую, пустую форму и режиме конструктора. Выбор элемента «Автоформа» позволяет автоматически создать формы определенных форматов: в столбец, ленточную или табличную. Элемент списка «Диаграмма» вызывает Мастер диаграмм для добавления диаграммы в создаваемую форму. С помощью выбора элемента списка «Сводная таблица» можно создать форму, основанную на данных из сводных таблиц Microsoft Excel.

3. В раскрывающемся списке, находящемся в нижней части диалогового окна «Новая форма», содержатся имена всех таблиц и запросов базы данных, которые могут быть использованы в качестве источника данных для формы. Выберите элемент «Сотрудники». Нажмите кнопку «ОК» и Access отобразит первое диалоговое окно Мастера форм.

4. В списке «Доступные поля» этого окна выделите элемент «КодСотрудника», затем нажмите кнопку ">" для перемещения элемента «КодСотрудника» из списка «Доступные поля» в список «Выбранные поля». Переместите элементы «Фамилия», «Имя», «Должность», которые являются полями таблицы «Сотрудники». Размещение этих полей в форме позволит редактировать данные таблицы «Сотрудники».

5. Раскройте список «Таблицы/Запросы» и выделите в нем таблицу «Задания».

6. Переместите все поля таблицы «Задания» в список «Выбранные поля».

7. Поскольку поле «КодСотрудника» таблицы «Сотрудники» было включено в список выбранные поля, то нет необходимости включать в форму поле «КодСотрудника» таблицы «Задания». В списке «Выбранные поля» уберите элемент «КодСотрудника». Нажмите кнопку «Далее» для отображения второго диалогового окна Мастера форм.

8. Так как поля, выбранные для отображения в форме, взяты из двух различных таблиц, то Мастер форм предоставляет возможность выбрать тип представления данных. Поскольку форма предназначена для просмотра данных о сотруднике и должна включать подчиненную форму содержащую данные о персональных мероприятиях, относящихся к сотруднику, выделите тип представления данных «по Сотрудникам» и выберите переключатель «Подчиненные формы». В правой верхней части второго диалогового окна Мастера форм отображаются поля создаваемой формы в зависимости от выбранного представления данных. Нажмите кнопку «Далее».

9. В третьем диалоговом окне Мастера форм можно определить параметры оформления подчиненной формы. Выберите переключатель «Ленточный»; в этом случае будет создана подчиненная форма, отображающая данные из таблицы «Задания» в ленточном формате, аналогичном табличному, но предоставляющем возможность изменения параметров форматирования элементов управления. Нажмите кнопку «Далее».

Примечание: В формах, в которых источником данных являются таблицы, связанные отношением «один – ко – многим», подчиненные формы необходимо связывать с основной формой для того, чтобы все записи, отображаемые в подчиненной форме, соответствовали текущей записи основной. Мастер форм использует информацию о типе отношения между таблицами для связывания основной и подчиненной форм. Если отношение между двумя таблицами не установлено, то Мастер форм пропускает шаг, на котором запрашивает тип представления данных в форме, и непосредственно переходит к определению стиля формы. Для корректной работы этой пошаговой процедуры необходимо установить связь между таблицами «Сотрудники» и «Задания».

10. Четвертое диалоговое окно Мастера позволяет выбрать стиль оформления новой формы. Предоставляется несколько встроенных стилей. Поскольку создаваемая форма предназначена для ввода данных и не нуждается в специальных эффектах оформления, выделите стиль «Обычный», а затем нажмите кнопку «Далее».

11. Укажите названия для основной и подчиненной форм. Введите: «Задания» в поле «Форма:» и «Задания подчиненная форма» в поле ввода «Подчиненная форма». Выберите переключатель «Открытие формы для просмотра или ввода данных», а затем нажмите кнопку «Готово» для завершения создания формы.

Совет: Access предлагает имена по умолчанию для формы и подчиненной формы. Убедитесь в том, что введенные имена действительно описывают созданную форму, а также в том, что в имя подчиненной формы включено имя (аббревиатура) основной формы, чтобы была очевидная связь между основной и подчиненной формами.

После того как Мастер форм закончит создание форм, основная форма будет открыта в режиме формы.

#### **Внимание!**

Если на последнем шаге Мастера форм переключатель «Открытие формы для просмотра или ввода данных» не был выбран, то автоматически происходит переход в режим конструктора форм. Access использует имена полей (или значение свойства «Подпись», если оно определено) в качестве меток текстовых полей, а также в качестве заголовков столбцов для подчиненной формы. Имя подчиненной формы Access использует в качестве метки для области подчиненной формы.

## **Использование окна конструктора форм**

Для изменения внешнего вида новой формы нажмите кнопку «Конструктор» на панели инструментов. Плавающая панель инструментов, появившаяся в режиме конструктора, называется «Панель элементов». Она позволяет разместить в форме новые элементы управления. Для отображения панели элементов нажмите кнопку «Панель элементов» на панели инструментов «Конструктор форм» или выберите команду «Вид, Панели инструментов», а затем установите флажок «Панель элементов».

### **Внимание!**

Не сохраняйте форму с теми изменениями, которые будут сделаны при выполнении примеров. Эти изменения имеют лишь демонстративный характер. Сохранение этих изменений навсегда изменит созданную форму. Чтобы предотвратить случайное изменение форм «Задания» и «Задания подчиненная форма», сделайте их резервные копии и работайте с ними. Резервная копия формы создается точно так же, как и резервная копия таблицы.

Форму можно разделить на три основные части: заголовок формы, область данных и примечание формы. Заголовок и примечание формы не являются обязательными частями формы. Для добавления или удаления областей заголовка и примечания выберите команду «Вид, Заголовок/Примечание формы», а затем снимите или установите флажок.

Значения свойств всей формы, разделов формы и каждого элемента управления в форме определяются в окнах свойств соответствующих элементов. Для отображения на экране окна свойств следует выделить нужный объект, щелкнув по нему левой кнопкой мыши, а затем нажать кнопку «Свойства».

Высоту раздела формы можно изменить, перетаскивая границу раздела при помощи мыши. Для этого перемещайте указатель мыши над границей разделов, пока он не превратится в черту с двумя вертикальными стрелками. Нажмите левую кнопку мыши и перетащите границу раздела до получения нужного размера.

## **Изменение размера отдельного элемента управления**

При выделении элемента управления вокруг него появляется рамка с маркером перемещения (в левом верхнем углу) и пятью маркерами изменения размеров.

Маркер перемещения позволяет перемещать отдельно каждый элемент управления (когда указатель мышки примет вид указывающей кисти).

Маркеры изменения размеров позволяют изменить ширину и высоту элемента управления (когда указатель мышки примет вид двусторонней стрелки).

Когда указатель мышки принимает вид раскрытой кисти, то появляется возможность перемещения присоединенных элементов управления.

В форме «Задания» имеются два вида элементов управления, связанных между собой: надпись (где выводится текст) и поле (где отображаются данные таблиц).

Чтобы выделить любой элемент управления следует щелкнуть по нему левой кнопкой мыши. Если выделить элемент управления «Поле», то вместе с ним выделится и элемент управления «Надпись» (если они связаны между собой).

### **Выравнивание элемента управления по сетке**

Сетка состоит из линий, которые проводятся через каждое деление горизонтальной и вертикальной линеек окна Конструктора форм (т.е. через сантиметр). Если сетка видна, то с ее помощью можно выравнивать элементы управления по горизонтали или по вертикали. Даже если сетка не видна, элементы управления можно «привязать к сетке», выбрав команду «Формат, Привязать к сетке». Если перемещать элемент управления при включенной привязке к сетке, то левый верхний угол объекта перемещается дискретно по узлам сетки.

Размеры элементов управления можно также выравнивать по узлам сетки. Для этого надо выделить элемент управления и выбрать команду «Формат, Размер, по размеру данных».

Команда меню «Вид, Сетка» управляет отображением сетки на экране. По умолчанию сетка отображается во всех новых формах. В случае если частота узлов сетки превышает 10 узлов на сантиметр, они не отображаются на экране. Чтобы изменить частоту узлов сетки:

1. Выберите команду «Правка, Выделить форму».
2. Нажмите кнопку «Свойства».
3. В окне свойств формы раскройте вкладку «Макет» и найдите свойства «Число делений по X» и «Число делений по Y». Измените значение этих свойств для оси X - на 10, для оси Y - на 12.

Чтобы выделить или переместить несколько элементов управления одновременно, можно использовать один из следующих способов:

1. Охватить группу элементов управления прямоугольником при помощи мыши. Переместить группу можно, зацепив указателем мыши за границу одного из элементов управления.
2. Щелкнуть левой кнопкой мыши по одному элементу управления, чтобы выделить его и, удерживая нажатой клавишу <Shift>, выделить остальные элементы управления.

### **Изменение текста надписей**

Поместите указатель мыши в пределах границ элемента управления «Надпись» и щелкните левой кнопкой мыши. Измените все надписи, добавив слово: «сотрудника», чтобы, например, надпись стала не: «Фамилия», а «Фамилия сотрудника».

Если при изменении имени элемента управления «Поле» допущена ошибка названия поля, то в режиме формы в этом поле будет выводиться сообщение об ошибке: #имя поля?

Сделайте сознательно ошибку в поле Должность и просмотрите результат в форме.

### **Изменение свойств подчиненной формы**

Чтобы изменить свойства подчиненной формы «Задания подчиненная форма», надо открыть эту форму в режиме Конструктора. Измените расположение и размеры полей так, чтобы все поля были видны в форме «Задания». В окне свойств раскройте вкладку «Данные» и в свойстве «Разрешить изменение» установите значение «Нет». В свойстве «Разрешить удаление» установите значение «Нет». В свойстве «Разрешить добавление» установите значение «Да».

Просмотрите в форме «Задания» результат выполненных вами изменений. Обратите внимание, что в подчиненной форме появилась специальная пустая запись, предназначенная для добавления новой записи в таблицу «Задания».

Примечание: Если в качестве значения свойства «Ввод данных» выбрать «Да», то в подчиненной форме отобразится только пустая запись, предназначенная для добавления новой записи, ни одна из существующих записей отображена не будет.

Задание: Приведите форму к следующему виду, изображенному на рис.10.1.

### **Добавление новых записей, изменение и удаление в таблице «Задания»**

В форме «Задания» введите следующую новую запись, касающуюся Кротова Андрея (при условии, что свойство «Разрешить добавление» имеет значение «Да»):

К	2	02.01.07	2	15.01.07	8	15000	Результаты квартального отчета
---	---	----------	---	----------	---	-------	--------------------------------------

Персональные мероприятия

Код сотрудника: 5 Должность: Менеджер по продажам

Сотрудник: Карпов Андрей

Мероприятия	Ответственный	Дата-план	Начальник	Дата-факт	Рейтинг	Сумма	Прим.
		15.09.98		13.10.98			
К	2	02.01.96	2	15.01.96	8	1500.00	за хор

Запись: 5 из 7

Запись: 1 из 7

Рис. 10.1

Изменение содержимого записи происходит точно так же, как и добавление новых (при условии, что свойство «Разрешить изменение» имеет значение «Да»).

Удаление записей в режиме формы можно: выбрав кнопку «Удалить запись» или выбрав команду «Правка, Удалить запись» или выбрав кнопку выделения записи, находящуюся слева от области данных формы, а затем нажав клавишу <Delete>.

Удалить запись можно при условии, что свойство «Разрешить удаление» имеет значение «Да».

Изменение параметров формы и элементов управления. Изменение последовательности перехода по клавише <Tab>.

Порядок, в котором текстовый курсор перемещается от одного поля к другому, определяется свойством «Переход по Tab» для каждого элемента управления. Для изменения последовательности перехода:

1. Перейдите в режим Конструктора.
2. Выберите команду «Вид, Последовательность перехода». В списке «Последовательность» приведены названия полей в порядке перехода от одного к другому.

3. Щелкните левой кнопкой мыши по кнопке, расположенной слева от названия поля, и перетащите это поле в нужное место последовательности.
4. Нажмите ОК и проверьте результат.

### **Исключение элемента управления из последовательности перехода**

Для удаления элемента управления из последовательности перехода по клавише <Tab> выделите элемент управления, откройте окно свойств этого элемента управления, раскройте вкладку «Другие» и измените значение свойства «Переход по Tab» на «Нет».

Редактировать поле «КодСотрудника» нельзя, поэтому исключите этот элемент управления из последовательности перехода.

### **Изменение источника данных подчиненной формы и сортировка данных в подчиненной форме**

Если мы хотим, чтобы последние данные о мероприятиях появлялись первыми, то для этого источник данных для формы «Задания подчиненная форма» необходимо отсортировать по убыванию. Установить сортировку по убыванию можно, указав порядок сортировки в запросе, являющемся источником записей подчиненной формы.

1. Создайте новый запрос на основе таблицы «Задания».
2. Поместите в запрос все поля таблицы, а для поля «ДатаПлан» укажите порядок сортировки.
3. Сохраните запрос под именем «Задания (для подчиненной формы)».

Для изменения источника данных подчиненной формы:

1. Откройте в режиме Конструктора форму «Задания подчиненная форма».
2. В окне свойств формы выберите вкладку «Данные».
3. Выберите элемент «Источник записей», откройте список и выделите в нем элемент «Задания (для подчиненной формы)» в качестве нового источника данных для подчиненной формы.
4. Просмотрите результат сделанных изменений.

### **Изменение фонового рисунка формы**

Фоновые рисунки из комплекта поставки Access хранятся в папке PROGRAM FILES\MICROSOFT OFFICE\OFFICE\BITMAPS\STYLES.

Для установки или удаления фонового рисунка формы используется окно свойств формы. Чтобы добавить фоновый рисунок и задать его параметры:

1. Откройте форму в режиме «Конструктор».
2. Выделите всю форму. Для этого щелкните левой кнопкой мыши по квадратику в верхнем левом углу окна конструктора формы (в этом месте пересекаются вертикальная и горизонтальная линейки). После выделения формы в месте пересечения линеек появится черный квадратик.



3. Откройте окно свойств и раскройте вкладку «Макет», переместитесь на свойства фоновый рисунок: «Рисунок», «Масштабы рисунка», «Выравнивание рисунка» и «Мозаичное заполнение».

4. Установите значения свойств рисунка. Эти изменения немедленно отображаются в форме.

5. Закройте окно свойств.

### **Использование дополнительных цветов**

Если ни один из 16 системных цветов Windows не подходит для раздела формы или элемента управления, то можно создать свой собственный цвет. Для этого:

1. Щелкните левой кнопкой мыши по одному из свойств цвета элемента формы: «Цвет фона», «Цвет текста» или «Цвет границы» в окне «Свойства».

2. Нажмите кнопку «Построить». Появится диалоговое окно «Цвет». Если среди 48 цветов есть нужный, то выделите его, а затем нажмите кнопку ОК. Если же необходимо создать свой цвет, то нажмите кнопку «Определить цвет».

3. Выберите нужный цвет, перетаскивая перекрестие в схеме определения цвета при помощи мыши.

4. Отрегулируйте яркость с помощью полосы, расположенной справа от схемы определения цвета.

5. Нажмите кнопку «Добавить в набор». Новый цвет будет добавлен в первый из 16 цветных блоков.

### **Создание диаграммы «Рейтинг сотрудников»**

Для того чтобы вставить в форму диаграмму, необходимо:

– переключиться в режим конструктора;

– выполнить команду «Вид, Панель элементов»;

– на этой панели щелкнуть по кнопке <Aa>;

– создать прямоугольник для надписи - заголовка диаграммы. Для этого переведите курсор в левый верхний угол будущего прямоугольника, нажмите левую кнопку мыши и, не отпуская ее, доведите до правого нижнего угла, затем отпустите кнопку;

– ввести надпись «Диаграмма»;

– выполнить команду «Вставка, Диаграмма»;

– на свободном месте формы растянуть прямоугольник для диаграммы (нажмите левую кнопку мыши в левом верхнем углу и, не отпуская ее, растяните прямоугольник до правого нижнего угла, затем отпустите кнопку);

– выбрать таблицу и щелкнуть по кнопке «Далее»;

– выбрать поля;

– щелкнуть по кнопке «Далее»;

–выбрать вид диаграммы «Гистограмма» (по умолчанию он и стоит) и щелкнуть по кнопке «Далее»;

–дважды щелкнуть по надписи «Сумма», выбрать «Отсутствует» и щелкнуть по кнопке ОК;

–щелкнуть по кнопке «Далее»;

–вновь щелкнуть по кнопке «Далее», так как в строке «Поля формы» и в строке «Поля диаграммы» по умолчанию находится то, что нам и нужно;

–стереть название диаграммы (так как мы уже задали надпись для диаграммы) и щелкнуть по кнопке «Далее».

Отредактируйте вид осей диаграммы. Для этого:

–дважды щелкните по диаграмме;

–дважды щелкните по значениям вертикальной оси;

–выберите вкладку «Шкала»;

–уберите «галочку» у надписи «Минимальное значение», а в ячейке справа от этого названия введите 1;

–уберите «галочку» у надписи «Максимальное значение», а в ячейке справа от этого названия введите 9;

–уберите «галочку» у надписи «Цена основных делении», а в ячейке справа от этого названия введите 1 и щелкните по кнопке ОК;

–расширьте область диаграммы, перетащив правую границу окна диаграммы несколько правее (подведя курсор к правой границе до появления двойной стрелки и нажав левую кнопку мыши);

–закройте окно «Microsoft Graph», выбрав в меню «Файл» пункт «Выход и возврат в форму».

–перейдите в режим формы и просмотрите форму для разных записей (щелкая по кнопке перехода к следующей записи в нижней части формы).

Вы увидите изменение диаграмм

–закройте форму.

## **ЛАБОРАТОРНАЯ РАБОТА № 11 (4 часа)**

**Тема:** Создание элементов управления формы

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать группы элементов управления,
- создавать поля со списком,
- создавать набор вкладок.

### **Создание запроса для основной формы.**

Для создания многотабличной формы необходимо создать запрос, который будет источником данных для формы.

Создайте запрос на основе таблиц «Задания» и «Сотрудники». В запрос добавьте все поля таблицы «Задания». Из таблицы «Сотрудники» добавьте

поля»: «Фамилия», «Имя», «Должность», «Фотография». Отсортируйте поле «Фамилия» по возрастанию. Сохраните запрос под именем: «Задания (для формы)».

### **Создание формы без Мастера.**

Чтобы создать пустую форму:

1. В окне базы данных выберите вкладку «Формы» и кнопку «Создать».
2. В поле со списком выберите запрос «Задания (для формы)», а в списке выше выберите элемент «Конструктор».
3. Выберите команду «Вид, Заголовок/примечание формы». Если сетка не появится в форме, то выберите команду «Вид, Сетка».
4. Нажмите кнопку «Надпись» на панели элементов. Поместите указатель мыши в виде крестика в области раздела заголовка формы. Центр крестика определяет позицию верхнего левого угла элемента управления. Нажмите левую кнопку мыши и, удерживая ее, перетащите указатель в нижний правый угол надписи. Число строк и количество символов текущего типа шрифта, которые может отобразить контейнер, выводятся в строке состояния. При достижении нужных размеров отпустите левую кнопку мыши.
5. Внутри контура в качестве текста введите: Ввод персонального мероприятия.
6. С помощью панели инструментов «Панель форматирования» измените надпись (шрифт, размер, цвет и т.д.).
7. Для добавления текстового поля, связанного с полем источника данных формы, используют кнопку «Список полей». Выделите элемент «КодСотрудника» и поместите его в раздел области данных.
8. Иногда возникает необходимость включить в форму длинные текстовые комментарии. Для создания многострочного текстового поля выделите в списке полей элемент «Примечание» и поместите его в раздел области данных. Удалите метку поля «Примечание» и увеличьте размеры текстового поля. Нажмите кнопку «Свойства», выберите вкладку «Макет» и найдите свойство «Полосы прокрутки». Выберите для него значение «По вертикали». К текстовому полю «Примечание» будет добавлена вертикальная полоса прокрутки.
9. Если в дальнейшем форму предполагается печатать, то установите в качестве значения свойств «Расширение» и «Сжатие» значение «Да». В результате вертикальные размеры поля будут изменяться автоматически таким образом, чтобы напечатать все данные, содержащиеся в нем.
10. Для создания вычисляемого поля, отображающего текущую дату и время, на панели элементов выберите кнопку «Поле» и поместите текстовое поле в правую часть раздела заголовка формы. Измените название метки на: «Дата/Время:» и переместите ближе к самому полю. Для отображения в поле текущих даты и времени введите в поле: =Now(). Измените длину текстового поля так, чтобы в нем поместилось необходимое количество символов. Для измене-

ния формата вывода в текстовое поле чисел, дат и времени можно использовать окно свойств, в котором есть свойство «Формат поля».

11. Можно изменить макет формы, используя свойство формы «Режим по умолчанию». Это свойство имеет три значения: простая форма (отображает на экране только одну запись формы); ленточная форма (позволяет выводить в форме одновременно столько записей, сколько их уместится в текущем окне); таблица (позволяет выводить поля формы в табличном формате).

12. Сохраните форму под именем «Ввод персонального мероприятия»

### **Создание группы элементов управления для выбора одного значения.**

Если в форме использовать переключатели, выключатели, кнопки с фиксацией и флажки сами по себе, то в качестве значений они могут возвращать только «Да/Нет» («1/0» или «Истина/Ложь»). В этом случае они используются как свободные элементы управления, что ограничивает возможности возвращаемых в поля таблицы значений. Включение этих элементов управления в группу позволит им возвращать любое число, определенное свойством «Значение параметра» для данного элемента управления.

Возможности, предоставляемые свойством «Значение параметра», позволяют использовать один из трех элементов управления в пределах группы для возвращения значения полю «Рейтинг» таблицы «Задания».

По умолчанию все элементы управления, добавляемые в форму с помощью панели элементов, являются свободными. Для связывания элемента управления с полем выделите его, а затем в списке полей щелкните левой кнопкой мыши по имени поля, с которым необходимо связать элемент управления. Другим способом связывания элемента управления с полем является создание свободного элемента управления с помощью панели элементов, а затем ввод имени поля в текстовое поле ввода свойства «Данные», для этого в окне свойств элемента управления раскройте вкладку «Данные». Для создания группы элементов управления Access предлагает два способа: с использованием Мастера групп и добавление выключателей и переключателей в группу вручную. Рассмотрим оба метода.

### **Использование Мастера групп.**

Для создания группы элементов управления для поля «Рейтинг» таблицы «Задания»:

1. Откройте форму «Ввод персонального мероприятия». Если кнопка «Мастера» не нажата, то щелкните по ней. Эта кнопка нажата, если она утоплена.

2. На панели элементов нажмите кнопку «Группа». Появится первое диалоговое окно «Мастер групп».

3. В строке таблицы «Подписи» введите пять из девяти возможных значений рейтинга: Отлично, Хорошо, Удовлетворительно, Неудовлетворительно и Плохо. Нажмите кнопку «Далее».

4. Во втором диалоговом окне Мастера групп можно определить значение по умолчанию для переключателя. Выберите переключатель: «Да, выбор по умолчанию» и откройте список рядом с этим переключателем. Выберите элемент «Хорошо».

Совет: Можно создать клавиши-акселераторы для переключателей. Для этого в названии переключателя перед буквой, которая будет использоваться в качестве акселератора, надо поставить символ «&». После этого в режиме формы выбор нужного переключателя будет осуществляться одновременным нажатием клавиш <Alt> и выбранной в качестве акселератора буквы.

5. В третьем диалоговом окне Мастера групп можно задать значения для каждого переключателя группы. В пять текстовых полей ввода введите: 9, 7, 5, 3, и 1, а затем кнопку «Далее».

6. В четвертом диалоговом окне Мастера групп можно связать группу элементов управления с полем таблицы или столбцом запроса, являющимся источником записей свободной формы. Выберите переключатель «Сохранить, значение в поле», а в списке рядом с этим переключателем выберите элемент «Рейтинг», являющийся названием столбца запроса, с которым связана форма. Нажмите кнопку «Далее».

7. В пятом диалоговом окне Мастера групп можно определить тип элементов управления, добавляемых в группу: переключатели, выключатели или флажки, а также стиль оформления группы. В области «Образец» можно просмотреть внешний вид группы элементов управления. Выберите «Переключатели», а оформление группы - «Вдавленное». Нажмите кнопку «Далее».

8. В текстовом поле ввода последнего диалогового окна Мастера групп можно ввести название группы переключателей, которое будет являться значением свойства «Подпись» метки группы элементов управления. В это текстовое поле введите: Рейтинг. Нажмите кнопку «Готово».

9. Для тестирования новой присоединенной группы элементов управления добавьте в форму текстовое поле «Рейтинг», связанное со столбцом «Рейтинг» запроса «Задания», являющимся источником данных формы. Выбрав какой-либо переключатель в группе переключателей проследите за значением рейтинга в текстовом поле «Рейтинг».

### **Создание группы без помощи Мастера.**

1. Щелкните левой кнопкой мыши по кнопке «Мастера элементов» на панели элементов, чтобы сделать ее не нажатой (она должна иметь приподнятый вид). Тем самым вы откажетесь от помощи Мастеров элемента. Затем нажмите на панели элементов кнопку «Группа переключателей» и поместите группу в области данных рядом с группой «Рейтинг».

2. На панели инструментов нажмите кнопку «Список полей». В появившемся окне списка полей выделите «Рейтинг» запроса «Задания».

3. Нажмите левую кнопку мыши и перетащите элемент списка в раздел области данных правее группы переключателей «Рейтинг», созданной в предыдущем разделе, затем отпустите левую кнопку мыши. В области данных будет создана группа элементов управления с размерами, заданными по умолчанию. Свойство «Подпись» метки, связанной с группой элементов управления, автоматически получит в качестве значения название поля, которое было выбрано в окне списка полей.

4. Измените размеры созданной группы элементов управления, а шрифт метки группы сделайте полужирным.

Для добавления пяти переключателей, возвращающих значения полю «Рейтинг»:

1. На панели элементов дважды щелкните левой кнопкой мыши по кнопке «Переключатель».

2. Маленький крестик, в который превратится указатель мыши, определяет верхний левый угол расположения переключателя. Поместите его в нужное место в пределах созданной группы.

3. Включите в группу 5 переключателей. Метки кнопок будут нумероваться в порядке их добавления.

4. Для отказа от дальнейшего добавления переключателей еще раз нажмите на панели элементов кнопку «Переключатель».

5. Измените текст меток: Отлично, Хорошо, Удовлетворительно, Неудовлетворительно, и Плохо.

6. Дважды щелкните левой кнопкой мыши по верхнему переключателю, чтобы отобразить на экране окно его свойств. В этом окне раскройте вкладку «Данные» и в списке свойств выделите свойство «Значение параметра». Измените значение этого свойства с 1 на 9. По умолчанию в качестве значения этого свойства для переключателей используются числа, начиная с 1, увеличивающиеся в порядке расположения кнопок в группе.

7. Прделайте то же, что в пункте 6, для четырех оставшихся переключателей, изменив значения свойства «Значение параметра» с чисел 2, 3, 4, 5 на числа 7, 5, 3 и 1 соответственно. Никакие две кнопки группы элементов управления не могут иметь одинаковые значения.

8. Просмотрите внесенные изменения в режиме Формы.

9. Вернитесь в Конструктор и удалите последнюю созданную группу переключателей «Рейтинг».

10. Группы переключателей лучше использовать для выбора значения не более чем из трех-четырех предлагаемых вариантов. Если предстоит выбор из большого числа значений, то наиболее удобным типом элементов управления является поля со списками.

Если в таблице имеется логическое поле типа «Да/Нет», то для ввода нужного значения пользователем можно использовать один переключатель, связанный с полем, не включая его в группу элементов управления.

Примечание: Если добавить кнопку или флажок в присоединенную рамку объекта, перетащив в нее элемент из списка полей, то эта добавленная кнопка будет связана с выбранным полем независимо от того, к какому полю присоединена сама рамка. В этом случае окно свойств данной кнопки или флажка не будет содержать свойство «Значение параметра», а кнопка будет возвращать полю логическое значение типа «Да/Нет».

Независимая присоединенная кнопка внутри присоединенной рамки объекта не подчиняется правилам группы элементов управления, поэтому она может быть выбрана одновременно с другими кнопками и флажками.

Добавление независимых присоединенных кнопок внутри присоединенной рамки объекта в результате может вылиться в присвоение полям противоречивых значений.

Созданные элементы управления можно копировать в другую форму с помощью буфера обмена.

### **Использование Мастера списков**

Списки и поля со списками предназначены для достижения одной и той же цели — предоставления возможности выбора нужного значения из списка вместо его ввода в текстовое поле.

Создадим раскрывающийся список «Ответственный», который позволит выбирать имена сотрудников:

1. Откройте в режиме конструктора форму «Ввод персонального мероприятия», которая была сохранена ранее.

2. Нажмите на панели элементов кнопку «Мастера».

3. На панели элементов выберите «Поле со списком».

4. На панели инструментов нажмите кнопку «Список полей» и перетащите поле «КодОтветственного» в область данных возле группы Рейтинг.

5. В первом диалоговом окне Мастера списков выберем заданный по умолчанию способ: «Поле со списком использует значения из таблицы или запроса». Выбор этого переключателя определяет в качестве значения свойства «Источник записей» поля со списком значение «Таблица/запрос». Нажмите кнопку «Далее».

6. В списке второго диалогового окна выделите таблицу «Сотрудники». Нажмите кнопку «Далее».

7. В следующем диалоговом окне переместите поля «КодСотрудника» и «Фамилия» в список «Выбранные поля». «КодСотрудника» выступает в роли присоединенного поля, а поле со списком будет отображать значения поля «Фамилия». Нажмите кнопку «Далее».

8. В четвертом диалоговом окне Мастера списков отображается список значений для поля со списком. Access правильно определил, что поле «КодСотрудника» является ключевым полем таблицы «Сотрудники»), и поэтому корректно установил, что это поле является присоединенным полем для создаваемого поля со списком. Нажмите кнопку «Далее».

9. Создаваемое поле со списком возвращает значение «КодСотрудника», соответствующее выбранной фамилии в поле «Ответственный». Ранее при перетаскивании поля в область данных формы было определено, что значением свойства «Данные» является поле «КодОтветственного». Мастер списков использует этот факт и устанавливает это поле в качестве значения по умолчанию свойства «Данные», поэтому оставьте в переключателе «Сохранить в поле» значение «КодОтветственного». Нажмите кнопку «Далее».

10. В последнем диалоговом окне Мастера списков можно изменить подпись поля со списком. Введите: Ответственный. Нажмите кнопку «Готово».

11. Перейдите в режим форм и просмотрите результат.

### **Создание поля со списком без помощи мастера.**

Для создания в форме «Ввод персонального мероприятия» поля со списком «Начальник», состоящего из двух столбцов:

1. Откройте форму в режиме конструктора форм. Откажитесь от помощи Мастера.

2. На панели элементов нажмите кнопку «Поле со списком». В окне списка полей выделите поле «КодНачальника» и поместите его в область данных ниже поля со списком «Ответственный».

3. Дважды щелкните левой кнопкой мыши по новому полю со списком. В появившемся окне свойств этого поля раскройте вкладку «Данные».

4. Источником данных для создаваемого поля со списком является таблица «Сотрудники», поэтому заданное по умолчанию значение свойства «Тип источника строк» - «Таблица/запрос» является корректным. Раскройте список свойства «Источник строк» и выберите в нем элемент «Сотрудники» в качестве значения этого свойства.

5. После выбора в свойстве «Источник строк» нужного названия таблицы или запроса все поля таблицы или все столбцы запроса автоматически будут включены как столбцы поля со списком. Первый столбец таблицы «Сотрудники» - «КодСотрудника» - будет возвращаемым значением в поле «Начальник», то значение свойства «Присоединенный столбец» по умолчанию равное 1 является корректным. В этом свойстве можно задать любой номер столбца, который будет возвращаемым значением в поле, к которому присоединено поле со списком. Второй столбец таблицы «Сотрудники» - «Фамилия» - будет идентифицировать начальника. Быть ответственным или начальником какого-либо персонального мероприятия может быть только сотрудник, включенный в таблицу «Сотрудники», поэтому откройте список свойства «Ограничиться



списком» и выберите в нем элемент «Да». Если необходимо позволить пользователю вводить значения, не включенные в список, то значением этого поля должно быть «Нет».

6. Раскройте вкладку «Макет» в окне свойств и в поле свойства «Число столбцов» введите: 2 для создания поля со списком в два столбца. По умолчанию ширина каждого столбца поля со списком составляет 1 дюйм, т.е. 2,54 см. В поле «Ширина столбцов» введите ширину первого и второго столбцов: 0,3;0,8.

7. Просмотрите результат.

8. Для отображения в списке и текстовом поле только фамилии начальника надо изменить в свойстве «Ширина столбцов» первую цифру на 0 и тогда будут выводиться значения только второго столбца.

### **Создание поля со списком с фиксированным набором значений**

Раскрывающийся список для выбора рейтинга сотрудника экономит значительное место в форме по сравнению с элементами управления типа «Переключатель», помещенными в группу и предназначенными для тех же целей. Создадим раскрывающийся список, позволяющий вводить все возможные значения рейтинга:

1. Откройте форму «Ввод персонального мероприятия» в режиме конструктора. Нажмите кнопку «Мастера».

2. На панели элементов выберите «Поле со списком». Перетащите «Рейтинг» из списка полей в область данных под поле со списком «Начальник».

3. В первом диалоговом окне Мастера списков выберите переключатель «Будет введен фиксированный набор значений». Нажмите кнопку «Далее».

4. Для поля со списком «Рейтинг» необходимо два столбца - первый столбец будет содержать возможные значения рейтинга, числа от 0 до 9, а второй столбец будет содержать соответствующие расшифровки каждого кода рейтинга. В поле «Число столбцов» введите: 2. В столбцы 1 и 2 введите:

9	Отлично
8	Очень хорошо
7	Хорошо
6	Средне
5	Удовлетворительно
4	На заметку
3	Неудовлетворительно
2	Плохо
1	Очень плохо
0	Уволен

Установите нужную ширину столбцов. Если нет необходимости отображать код рейтинга в поле со списком, то перетащите правую границу столбца 1 максимально влево, чтобы ширина стала равной 0.

Нажмите кнопку «Далее».

5. В списке «Доступные поля» выделите элемент «Столбец1», содержащий коды рейтинга, в качестве присоединенного столбца для фиксированного списка значений, т.е. этот столбец содержит значения, которые можно сохранить или использовать позднее. Выбранный столбец должен содержать уникальные значения. Нажмите кнопку «Далее».

6. Принимаем заданные по умолчанию значения: Сохранить в поле «Рейтинг». Нажмите кнопку «Далее».

7. Введите подпись поля со списком: Рейтинг. Нажмите кнопку «Готово».

8. Откройте окно свойств поля со списком и раскройте вкладку «Данные». Установите значение «Да» в свойстве «Ограничиться списком». Обратите внимание на свойство «Источник строк». Такой формат ввода нужно использовать при добавлении значений списка вручную.

9. Просмотрите результат.

Ввод персонального мероприятия : форма

Дата/Время 27.09.2004 21:46:51

Ввод персонального мероприятия

Сотрудник: [Поиск]

Рейтинг

Отлично

Хорошо

Удовлетворительно

Не удовлетворительно

Плохо

Ответственный Воронова

Начальник 2

Рейтинг: 3

Рейтинг: 3

### Создание поля со списком для поиска записей

Мастер списков позволяет создавать поля со списками для поиска записи в форме, используя значение, выбранное в списке. Создадим поле «Найти» в форме «Ввод персонального мероприятия», которое будет содержать фамилии всех сотрудников из таблицы «Сотрудники»:

1. Перейдите в режим конструктора форм. Включите кнопку «Мастера».

2. Нажмите кнопку «Поле со списком» и поместите в область данных под полем со списком «Рейтинг».

3. В первом диалоговом окне Мастера списков выберите переключатель «Поиск записи в форме на основе значения, которое содержит поле со списком». Нажмите кнопку «Далее».

4. Переместите в список «Выбранные поля» поле «Фамилия». Нажмите кнопку «Далее».

Замечание: При создании поля со списком для поиска записей выделяйте только одно поле, т.к. при выделении более одного поля поиск записей осуществляться не будет.

5. В следующем диалоговом окне отображен список всех значений выбранного поля. Установите ширину и нажмите кнопку «Далее».

6. Введите подпись создаваемого поля: Найти. Нажмите кнопку «Готово».

7. Просмотрите результат.

### **Создание элемента управления «Набор вкладок»**

С его помощью можно создавать многостраничные формы, похожие на вложенные страницы. Чтобы добавить элемент управления в форму «Ввод персонального мероприятия»:

1. Перейдите в режим конструктора форм. Мастера для этого элемента управления не существует.

2. Нажмите кнопку «Набор вкладок» и поместите в область данных. По умолчанию создается две страницы.

3. Чтобы добавить страницу следует выбрать страницу, перед которой вы хотите вставить новую, вывести контекстное меню и в нем выбрать команду «Добавить вкладку». Так как добавляется новая страница перед текущей, то нельзя добавить новую страницу в конце существующих страниц вкладок. Если же вы хотите, чтобы новая страница была последней необходимо изменить порядок страниц.

4. Для изменения порядка страниц в контекстном меню следует выбрать команду «Последовательность вкладок». Появится диалоговое окно «Порядок страниц». Выберите страницу и нажимайте кнопки «Вверх» или «Вниз», пока страница не окажется в нужной позиции. Затем нажмите ОК.

5. Нам потребуется две страницы, остальные следует удалить. Выберите вкладку страницы, которую надо удалить, в контекстном меню выберите команду «Удалить вкладку».

6. В форме «Ввод персонального мероприятия» одну страницу будем использовать для отображения информации о сотруднике компании. Вторая страница будет отображать историю персональных действий этого сотрудника.

7. Выберите первую страницу элемента управления «Набор вкладок» и нажмите кнопку «Свойства». В свойстве «Подпись» вкладки «Макет» наберите: «Информация компании». Выберите вторую страницу и в свойстве «Подпись» наберите: «История». Закройте окно «Свойства».

8. Из списка полей перетащите поля «Фамилия», «Имя», «Должность», «Фотография» на страницу «Информация компании» и расположите их как вам нравится.

9. Выберите вторую страницу «История». На панели элементов нажмите кнопку «Подчиненная форма/отчет» и поместите в левом верхнем углу второй страницы.

10. В первом диалоговом окне Мастера подчиненных форм выберите переключатель «Таблица или запрос». Нажмите кнопку «Далее».

11. Во втором диалоговом окне можно выбрать таблицу или запрос, который будет источником данных для новой подчиненной формы, а также какие поля необходимо включить в подчиненную форму. В списке «Таблицы и запросы» выберите «Запрос: Задания (для подчиненной формы)». В список «Выбранные поля» поместите все поля запроса, кроме поля «Примечание». Нажмите кнопку «Далее».

12. В третьем диалоговом окне выберите переключатель «Самостоятельное определение». Появятся 4 раскрывающихся списка. В верхнем списке «Поля формы или отчета» выберите элемент «КодСотрудника» (оно будет полем связи); в верхнем списке «Поля подчиненной формы/отчета» также в качестве поля связи выберите поле «КодСотрудника». Нажмите кнопку «Далее».

13. В следующем диалоговом окне введите имя новой подчиненной формы: «Тестовая подчиненная форма». Нажмите кнопку «Готово».

14. Просмотрите результат.

## **ЛАБОРАТОРНАЯ РАБОТА № 12 (4 часа)**

**Тема:** Работа с отчетами

Цель лабораторной работы заключается в том, чтобы научиться:

- создавать автоматические отчеты
- создавать отчеты при помощи Мастера отчетов
- пользоваться конструктором отчетов
- создавать в отчетах вычисляемые поля
- преобразовывать отчеты в другие форматы
- делать снимки отчетов
- отображать в отчетах значения параметров
- выполнять в отчетах сортировку и группировку
- делать шаблоны отчетов.

### **Автоматическое создание отчета на основе таблицы или запроса**

Access предлагает несколько способов создания отчетов. Наиболее простым из них является использование средств автоматического создания отчета. Автоматически создаваемый на основе таблицы или запроса отчет называется *автоотчетом*. Access позволяет автоматически создавать отчеты двух форматов: в столбец и ленточный.

Чтобы создать автоотчет:

1. На панели объектов окна База данных (Database) щелкните по ярлыку Отчеты (Reports) и нажмите кнопку Создать (New). Появится диалоговое окно Новый отчет (New Report).

2. В списке диалогового окна Новый отчет (New Report) выделите один из элементов: Автоотчет: в столбец (AutoReport: Columnar) или Автоотчет: ленточный (AutoReport: Tabular).
3. В поле со списком, находящемся в нижней части диалогового окна Новый отчет (New Report), содержатся имена всех таблиц и запросов базы данных, которые могут быть использованы в качестве источника данных для отчета. Щелкните левой кнопкой мыши по кнопке со стрелкой для открытия списка, а затем выделите в списке необходимый элемент.
4. Нажмите кнопку ОК.

В результате Access автоматически создаст отчет на основе выбранного источника данных, используя ленточный формат или формат в столбец. Ленточный формат располагает поля выводимых записей в строку. Формат в столбец располагает поля выводимых записей в столбец. Автоотчет, созданный с использованием любого из этих двух форматов, будет включать все поля и записи, имеющиеся в выбранном источнике данных.

Чтобы созданный отчет можно было использовать в дальнейшем, его необходимо сохранить. Для этого выберите команду Файл, Сохранить (File, Save) или нажмите кнопку Сохранить (Save) на панели инструментов. Затем, в текстовое поле появившегося диалогового окна Сохранение (Save As) введите название нового отчета (например: Мой отчет) и нажмите кнопку ОК.

**Замечание.** Есть еще один вариант сохранения отчета: с помощью команды меню Файл, Сохранить как (File, Save As). Этой командой отображается диалоговое окно Сохранение (Save As). Введите имя отчета и, прежде чем нажать кнопку ОК, убедитесь, что в раскрывающемся списке Как (As) этого окна выбран элемент Отчет (Report). Выбранный элемент определяет то, как будет сохранен новый отчет, точнее, в виде какого объекта базы данных Access. Дело в том, что в новой версии Access 2002 появилась возможность сохранить отчет в виде другого объекта базы данных — страницы доступа к данным. Сделать это позволяет другой элемент раскрывающегося списка Как — элемент Страница доступа к данным (Data Access Page).

### **Создание отчета с помощью Мастера отчетов**

Еще одним простым и быстрым способом создания отчетов является использование Мастера отчетов. С его помощью можно создавать отчеты (от самых простых до включающих подчиненные отчеты), содержащие поля из одной или более таблиц или запросов. Использование Мастера отчетов позволяет очень быстро познакомиться со многими основными средствами разработки отчетов, предоставляемыми Access 2002.

Чтобы более наглядно представить процесс создания отчета с помощью мастера, опишем процедуру создания отчета на основе сразу нескольких таблиц: "Товары" (Products), "Поставщики" (Suppliers) и "Типы" (Categories) базы

данных "Борей" (Northwind). В отчете будет представлено текущее количество товаров на складе по категориям.

Чтобы создать простой отчет:

1. На панели объектов окна базы данных щелкните по ярлыку Отчеты (Reports) и нажмите кнопку Создать (New). Появится диалоговое окно Новый отчет (New Report).
2. В списке диалогового окна Новый отчет (New Report) выделите элемент Мастер отчетов (Report Wizard). В поле со списком, находящемся в нижней части диалогового окна Новый отчет (New Report), содержатся имена всех таблиц и запросов базы данных, которые могут быть использованы в качестве источника данных для отчета. Раскройте этот список и выделите в нем таблицу "Товары". Нажмите кнопку ОК. Появится первое диалоговое окно Мастера отчетов.
3. Прежде всего нужно определить, какие поля включить в отчет. Из таблицы "Товары" в отчете будут использоваться следующие поля: "Код-Товара" (ProductID), "Марка" (ProductName), "Цена" (UnitPrice), "На-Складе" (UnitsInStock). С помощью кнопки ">" (или дважды щелкнув по элементу списка) поочередно переместите эти поля из списка Доступные поля (Available Fields) в список Выбранные поля (Selected Fields). В отчете поля появляются слева направо, в соответствии с последовательностью их расположения в списке Выбранные поля.

**Замечание.** Если на каком-либо шаге создания отчета при помощи Мастера отчетов была допущена ошибка или возникла необходимость изменения каких-либо параметров, то для возвращения к предыдущему шагу нажмите кнопку Назад (Back). Кроме того, в любой момент можно нажать кнопку Отмена (Cancel) для отказа от создания отчета и возвращения к окну базы данных.

4. Кроме того, чтобы видеть поставщиков товара и категории, в создаваемый отчет необходимо добавить поле "Название" (CompanyName) из таблицы "Поставщики" (Suppliers) и "Категория" (CategoryName) из таблицы "Типы" (Categories). Для добавления первого недостающего поля раскройте список Таблицы и запросы (Tables/Queries) и выберите элемент Таблица: Поставщики (Table: Suppliers). Имя поставщика не должно отображаться в конце отчета. Чтобы столбец "Название" выводился сразу за столбцом "Марка", в списке Выбранные поля (Selected Fields) выделите поле "Марка", затем в списке Доступные поля (Available Fields) выделите поле "Название" и нажмите кнопку ">". Access переместит поле "Название" из списка Доступные поля (Available Fields) и вставит его в список Выбранные поля (Selected Fields) после поля "Марка", перед полем "Цена".
5. Аналогично добавьте в конец списка поле "Категория" из таблицы "Типы". Чтобы перейти ко второму диалоговому окну Мастера отчетов нажмите кнопку Далее.

## Замечание

Можно таким способом добавлять в создаваемый отчет поля из разных таблиц, которые связаны между собой в схеме данных.

6. Во втором диалоговом окне необходимо указать, каким образом будут структурированы данные в отчете. Мастер проверяет связи между таблицами и в зависимости от выбранного представления предлагает несколько вариантов группировки записи в отчете. Чтобы осуществить группировку по товарам, выделите в списке элемент Товары (by Products).
7. Нажмите на кнопку Вывести дополнительные сведения (Show Me More Information) в левой части диалогового окна для отображения одного из вспомогательных диалоговых окон Мастера отчетов. Если в этом диалоговом окне нажать кнопку Показать примеры (Show Me Examples), то Access выведет дополнительные вспомогательные диалоговые окна, в которых используются "примеры из таблиц "Продажи" (Sales Reps), "Клиенты" (Customers) и "Заказы" (Orders), позволяющие продемонстрировать различные способы группировки, которые Мастер отчетов может автоматически применить к создаваемому отчету. Закройте все вспомогательные диалоговые окна и вернитесь к диалоговому окну Мастера отчетов. Нажмите кнопку Далее (Next) для перехода к третьему диалоговому окну Мастера отчетов.
8. В третьем диалоговом окне Мастера отчетов необходимо определить дополнительные уровни группировки в отчете. Выделите в списке поле "Категория" (Category) и нажмите кнопку ">" (или просто дважды щелкните по этому полю в списке), чтобы установить группировку по категории товаров.

**Замечание.** Если потребуется отменить добавление поля группировки, нажмите кнопку "<" или дважды щелкните по названию этого поля (выделенному жирным шрифтом) в области диалогового окна с эскизом отчета.

9. Мастер отчетов позволяет задать интервалы значений, по которым группируются данные. Чтобы изменить их в этом же (третьем) диалоговом окне Мастера отчетов, нажмите кнопку Группировка (Grouping Options). На экране появится диалоговое окно Интервалы группировки (Grouping Intervals). Изменяя интервал группировки, можно влиять на способ группировки данных в отчете. Для числовых полей можно задать группировку по десяткам, сотням и прочим значениям. Для текстовых полей возможна группировка по первой букве, первым трем буквам и т. д.
10. Для создаваемого отчета выберите из списка Интервалы группировки (Grouping Intervals) вариант Обычный (Normal) и нажмите кнопку ОК. Для перехода к четвертому диалоговому окну Мастера отчетов нажмите кнопку Далее (Next).

11. Access позволяет сортировать записи в пределах группы по значению произвольного поля, при этом сортировка проводится не более чем по четырем полям. Выберите в первом раскрывающемся списке поле "КодТовара" (ProductID). По умолчанию принят порядок сортировки по возрастанию; если вы хотите выбрать порядок сортировки по убыванию, то нажмите на кнопку справа от раскрывающегося списка. Для возврата к сортировке по возрастанию снова нажмите на эту кнопку.
12. Чтобы указать для столбца отчета итоговую информацию, нажмите" на кнопку Итоги (Summary Options). В появившемся диалоговом окне Итоги (Summary Options), изображенном на рис. 12.1, можно задать параметры для выбранного столбца. Выбрать можно среди всех числовых полей отчета, не являющихся счетчиками, которые Мастер отчетов перечислит в этом диалоговом окне и позволит установить для любого из них флажки Sum, Avg, Min и Max. В зависимости от установленных флажков Мастер отчетов добавит соответствующие итоговые поля в конец отчета, вычисляющие итоговую сумму, среднее значение, минимальное или максимальное значение данного поля по всем записям. Группа переключателей Показать (Show) позволяет выбрать, следует ли в отчете указывать только итоги или необходимо выдавать полный отчет, добавляя поля с промежуточными итогами в конце каждой группы и поля с полными итогами в конце отчета. Для создаваемого отчета установите флажки Sum для поля "НаСкладе" и Avg для поля "Цена", переключатель данные и итоги (Detail and Summary) и флажок Вычислить проценты (Calculate Percent of Total for Sum), который позволяет выводить долю группы (в процентах) в общем итоге для всех групп (рис. 12.1). Для возврата к диалоговому окну Мастера отчетов нажмите кнопку ОК.
13. Нажмите кнопку Далее (Next), чтобы перейти к пятому диалоговому окну Мастера отчетов. Мастер отчетов предложит на выбор шесть вариантов оформления сгруппированных данных (вариантов макета отчета). В левой части диалогового окна Мастера отображается эскиз выбранного макета. Выберите в группе Макет (Layout) переключатель ступенчатый (Stepped). По умолчанию Мастер отчетов устанавливает флажок Настроить ширину полей для размещения на одной странице (Adjust the field width so all fields fit on a page), что позволяет сэкономить бумагу и сделать отчет более удобным для чтения. В группе Ориентация (Orientation) можно выбрать расположение страницы отчета при печати. Широким отчетам, в которых отображается много полей в одной строке, удобно задавать ориентацию альбомная (Landscape). Для отчета же из нашего примера оставьте выбранным переключатель книжная (Portrait). Чтобы перейти к шестому диалоговому окну Мастера отчетов, нажмите кнопку Далее (Next).



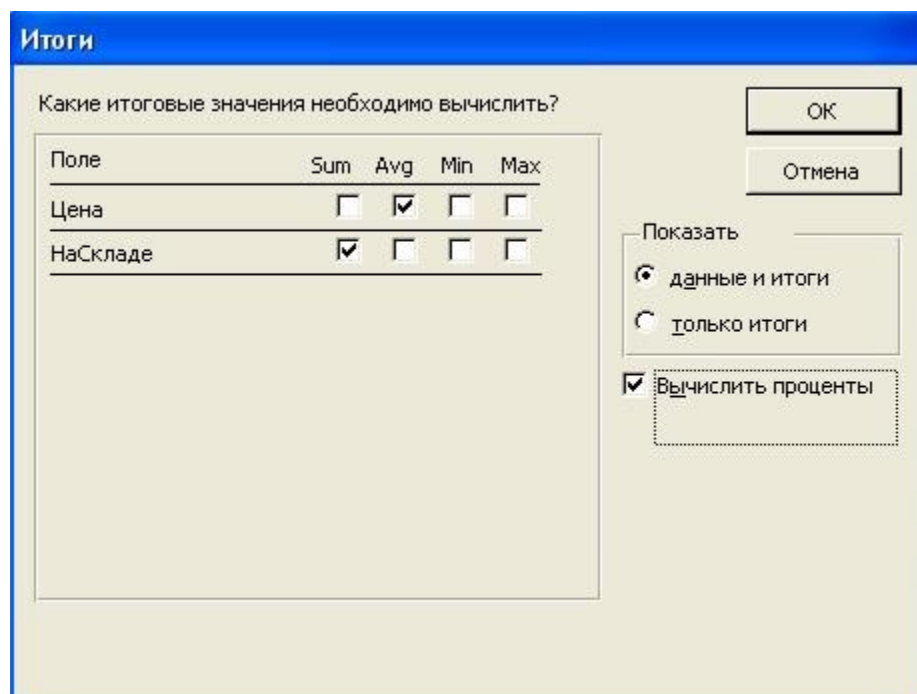


Рис. 12.1. Диалоговое окно Итоги

14. В шестом диалоговом окне Мастера отчетов выберите один из предложенных стилей оформления отчета. В левой части диалогового окна Мастера отображается эскиз отчета выбранного стиля. Для создаваемого отчета выберите стиль Сжатый (Compact), а затем нажмите кнопку Далее (Next) для перехода к последнему диалоговому окну Мастера отчетов.

**Замечание.** В шестом окне Мастера отчетов отображается список всех доступных в данный момент стилей. Кроме шести заранее определенных стандартных стилей, Access позволяет настроить любой стандартный стиль или создать свой собственный для использования при оформлении форм и отчетов, в том числе с помощью мастера. Набор вариантов оформления для отчетов можно пополнить и применить аналогично тому, как это делается для форм.

15. В последнем диалоговом окне Мастера отчетов в качестве заголовка отчета введите: Товары на складе. Мастер отчетов сохранит созданный отчет под этим же именем. Выберите переключатель Просмотреть отчет (Preview the Report) и нажмите кнопку Готово (Finish) для завершения создания отчета. Мастер отчетов создаст отчет и отобразит его в режиме Предварительного просмотра. Если при этом необходимо получить справку по работе с отчетом, предварительно установите флажок Вывести справку по работе с отчетом? (Display Help on Working with the Report?).

На рис. 12.2 изображен отчет, созданный мастером. Для просмотра всех частей отчета воспользуйтесь вертикальной и горизонтальной полосами прокрутки. Этот отчет можно использовать непосредственно в том виде, как его создал мастер, или применить его как основу для разработки более сложного отчета.

Борей - [Товары на складе]

Файл Правка Вид Сервис Окно Справка Борей

100% Закрывать

### Товары на складе

Категория	Код товара	Марка	Название	Цена	На складе
Кондитерские изд					
	2	Pavlova	Pavlova, Ltd.	785,25р.	29
	5	Teatime Chocolate Biscui	Specialty Biscuits, Ltd.	414,00р.	25
	6	Sir Rodney's Marmalade	Specialty Biscuits, Ltd.	645,00р.	40
	7	Sir Rodney's Scones	Specialty Biscuits, Ltd.	450,00р.	3
	11	NuNuCa Nuss-Nougat-Cr	АО Германия-Россия	630,00р.	76
	12	Gumbar Gummibarchen	АО Германия-Россия	405,35р.	15
	13	Schoggi Schokolade	АО Германия-Россия	975,50р.	49
	22	Scottish Longbreads	Specialty Biscuits, Ltd.	562,50р.	6
	48	Zaanse koeken	Zaanse Snoepfabriek	427,50р.	36
	49	Chocolade	Specialty Biscuits, Ltd.	573,75р.	15
	50	Maxilaku	Karkki Oy	900,00р.	10
	51	Valkoinen suklaa	Karkki Oy	731,25р.	65
	63	Tarte au sucre	Forots d'Erables	218,50р.	17
<b>Итого для 'Категория' = Кондитерские изделия (13 записей)</b>					
<b>Sum</b>					<b>386</b>
<b>Avg</b>				<b>1 132,20р.</b>	
<b>С разделителями разрядов</b>					<b>12,19%</b>
Молочные продукт					

Страница: 1

Готово NUM

Рис. 12.2. Отчет "Товары на складе", созданный Мастером отчетов

### Преобразование отчетов

Вы создаете базу данных, накапливаете в ней данные, редактируете и анализируете их, используя средства Access. В какой-то момент, когда результаты обработки данных готовы, вы представляете их в виде отчета Access — внешне выглядящего как обычный документ, но этот документ можно просмотреть лишь в базе данных Access, в которой он сохранен. Этот вариант не годится, когда результаты своей работы в виде отчета Access вы хотите передать кому-либо другому, кто не работает с Access или даже, возможно, вообще не имеет дела с базами данных. Самый простой вариант — напечатать отчет на принтере, и тогда формат, в котором он был создан, не будет иметь значения. Но чаще всего отчет требуется передать именно в электронном виде, например переслать по электронной почте или другим способом предоставить данные из отчета, чтобы в дальнейшем они могли быть использованы в других документах.

Для решения этой задачи Microsoft Access предоставляет целый ряд возможностей преобразования отчетов в другие форматы. Это формат документов RTF, с которым можно работать в Word и некоторых других текстовых редакто-

рах, формат электронных таблиц Excel, страницы HTML, разнообразные текстовые форматы (файлы с расширением txt, csv, tab, asc) с возможностью сохранения содержимого отчета на выбор в кодировке Windows, MS-DOS или Unicode, специальный формат снимков отчетов (snapshot). В новой версии Access 2002 к этим форматам добавились страницы доступа к данным и XML.

### **Преобразование отчета в формат документов Word и Excel**

Access позволяет преобразовать созданный отчет в документ Microsoft Word формата RTF. Чтобы выполнить такое преобразование:

1. Откройте отчет в любом режиме или просто выделите его в окне базы данных.
2. Выберите команду Сервис, Связи с Office, Публикация в Microsoft Word (Tools, Office Links, Publish It with MS Word).

В результате Access 2002 создаст файл формата RTF в текущем каталоге, например в папке \Мои документы (My Documents), по умолчанию используя в качестве имени файла название отчета, а затем откроет его в Word.

С помощью соседней команды меню Сервис, Связи с Office, Анализ в Microsoft Excel (Tools, Office Links, Analyze It with Microsoft Excel) можно преобразовать отчет в электронную таблицу Excel. При выборе этой команды автоматически создается новая рабочая книга Excel, в которую помещается рабочий лист с представлением отчета в виде электронной таблицы. В качестве имени файла XLS используется имя отчета.

**Замечание.** Эти команды преобразования в формат документов Word и Excel доступны также для таблиц, запросов и форм Access. Кроме того, для таблиц и запросов доступна еще команда Сервис, Связи с Office, Слияние с Microsoft Word (Tools, Office Links, Merge It with Microsoft Word). С помощью этой команды вызывается Мастер слияния, который позволяет на основе выделенной таблицы или запроса создать документ Word, связанный с этим источником данных, или связать данную таблицу или запрос с уже существующим документом Word, таким образом, что, когда данные в таблице или запросе будут изменяться, будут соответственно обновляться связанные поля в документе.

### **Преобразование отчета в другие форматы**

В Access 2002 существует средство преобразования отчетов в целый ряд других форматов. Это средство называется *экспортом*. Чтобы экспортировать отчет Access в какой-нибудь другой формат, выделите отчет в окне базы данных или откройте его в любом режиме, а затем выберите команду Файл, Экспорт (File, Export). Появится диалоговое окно экспорта объекта базы данных, похожее на стандартное окно сохранения файла, в котором в качестве типа сохраняемого файла можно выбрать любой формат из предлагаемого списка:

- Microsoft Access (для копирования отчета в другую базу данных);

- Microsoft Excel 5—7 (для преобразования отчета в формат ранних версий Excel);
- Microsoft Excel 97—2002 (для преобразования отчета в формат последних версий Excel);
- страницы HTML (HTML Documents) (для преобразования отчета в формат статических Web-страниц в формате HTML);
- текстовые документы (Text Files) (для преобразования в один из текстовых форматов Windows или MS-DOS: TXT, CSV, TAB, ASC);
- Reach Text Format (для преобразования отчета в документ Word);
- снимок (Snapshot Format) (для преобразования отчета в формат статических снимков);
- документы XML (XML Documents) (для преобразования отчета в формат динамических Web-страниц в формате XML).

Выбрав формат в раскрывающемся списке Тип файла (Save as type), введите в поле Имя файла (File name) имя файла, который будет создан, и нажмите кнопку Экспорт (Export). Предварительно можно установить в этом диалоговом окне флажок Автозапуск (Autostart), если вы хотите, чтобы после экспорта автоматически открылся созданный файл в приложении, в котором этот файл можно просмотреть.

### Снимки отчетов

В Access 2000 появилось новое средство публикации отчетов, отличающееся от экспорта в разнообразные наиболее популярные форматы документов. Это — преобразование отчетов Access в специально разработанный для задачи их публикации формат *снимков отчетов* (report snapshot). Основная идея этого новшества — предоставление компактного представления отчетов Access в виде отдельных от базы данных файлов (снимков отчетов), которые можно просматривать, листать и печатать без использования Microsoft Access, с помощью специально созданного очень небольшого приложения, названного "Просмотр снимков" (Snapshot Viewer). Эта программа совершенно независима от самого Microsoft Access и может быть установлена без установки на компьютер пакета Microsoft Office. Ее дистрибутив входит в состав пакета, а также доступен через Интернет на Web-узле Microsoft: <http://www.microsoft.com>. Электронную версию отчета Access в виде файла снимка отчета можно также переслать по электронной почте или распространить другим способом передачи файлов. Те пользователи, на чьих компьютерах установлена программа "Просмотр снимков", смогут просмотреть полученный в виде снимка отчет. В новой версии Access 2002 это средство сохранилось.

Для преобразования отчета Access в формат снимка требуется воспользоваться тем же, упомянутым в предыдущем разделе, средством экспорта объектов базы данных Access.

Снимок отчета — файл с расширением `snp` — содержит копии всех страниц отчета, включая графические и другие вложенные объекты. Чтобы на основе отчета Access создать снимок:

1. В окне базы данных в списке отчетов выделите нужный отчет и выберите команду **Файл, Экспорт (File, Export)**. Появится диалоговое окно **Экспорт объекта (Export Object)**.
2. В раскрывающемся списке **Тип файла (Save as type)** выберите элемент **Снимок (Snapshot Format)**, в раскрывающемся списке **Папка (Save in)** выберите папку, в которой будет сохранен файл, а в поле **Имя файла (File name)** введите имя файла снимка. Нажмите кнопку **Экспорт (Export)**.
3. Созданный в результате снимок отчета будет сохранен в указанной папке, после чего он будет открыт в автоматически запущенном (если ранее не было запущено) приложении "Просмотр снимков".

В нижней части окна приложения "Просмотр снимков" содержится панель навигации, которая позволяет листать страницы отчета при просмотре. Крайняя справа кнопка на этой панели позволяет выполнить печать отчета. Щелкнув правой кнопкой мыши по области отчета, вы откроете контекстное меню, которое включает команды, дублирующие функции кнопок панели навигации, и команду **Масштаб (Zoom)**, с помощью которой вы сможете регулировать масштаб отображения страниц отчета в окне программы **Просмотра снимков**. Это приложение позволяет одновременно просматривать несколько снимков отчета. Переключаться между снимками и управлять расположением их окон внутри приложения можно с помощью команд меню **Окно (Window)**. Кроме просмотра и печати снимков программа "Просмотр снимков" позволяет также пересылать их по электронной почте. Для этого нужно выбрать команду меню **Файл, Отправить (File, Send)**. Тогда появится окно сообщения **Microsoft Outlook**, в которое автоматически попадет снимок отчета в виде вложенного файла. Останется заполнить необходимые поля и отправить сообщение с помощью кнопки **Отправить (Send)**. Получателю сообщения достаточно дважды щелкнуть мышью по значку вложенного файла, чтобы запустить программу "Просмотр снимков" и открыть в ней полученный снимок.

### **Режимы работы с отчетами**

Работа с отчетами Access 2002 может происходить в трех режимах: в режиме **Просмотра образца**, в режиме **Предварительного просмотра** и в режиме **Конструктора**. Переключиться из одного режима в другой можно либо с помощью команды **Образец (Layout Preview)**, **Предварительный просмотр (Print Preview)** или **Конструктор (Design View)** в меню **Вид (View)** в любом режиме работы с отчетом, либо с помощью кнопки **Вид (View)** панели инструментов **Конструктор отчетов (Report Design)** в режиме **Конструктора**.

Режим Предварительного просмотра предназначен для просмотра отчета в Access в том виде, как он будет выведен на печать. Отчет, открытый в режиме Предварительного просмотра, можно просмотреть в увеличенном или уменьшенном масштабе, отобразить на экране одновременно одну или несколько страниц, а также напечатать, предварительно определив параметры страницы. По умолчанию из окна базы данных (при двойном щелчке по отчету или нажатии кнопки Открыть (Preview) в окне базы данных) отчет открывается именно в этом режиме. Если отчет был открыт в другом режиме, то для перехода в режим Предварительного просмотра выберите команду Вид, Предварительный просмотр (View, Print Preview) или нажмите на стрелку, расположенную справа от кнопки Вид (View) панели инструментов Конструктор отчетов (Report Design) и в открывшемся списке выберите элемент Предварительный просмотр (Print Preview).

Режим Просмотра образца является удобным средством проверки внешнего вида и структуры (макета) создаваемого отчета. В этом режиме в окно просмотра выводятся только те данные, которые необходимы для заполнения каждого из элементов макета отчета. Для просмотра всех данных, выводящихся в отчете, следует использовать режим Предварительного просмотра. Чтобы перейти в режим Просмотра образца, выберите команду Вид, Образец (View, Layout Preview) или нажмите на стрелку, расположенную справа от кнопки Вид (View) панели инструментов Конструктор отчетов (Report Design) и в открывшемся списке выберите элемент Образец (Layout Preview).

В режиме Конструктора можно разрабатывать отчеты, модифицировать их структуру, изменять внешний вид, добавлять и удалять поля. Этим режимом обычно пользуются разработчики отчетов Access. Чтобы перейти в режим Конструктора, выберите команду Вид, Конструктор (View, Design View) или нажмите на стрелку, расположенную справа от кнопки Вид (View) панели инструментов Конструктор отчетов (Report Design) и в открывшемся списке выберите элемент Конструктор (Design View).

**Замечание.** Создать новый отчет можно "с нуля", начиная его разработку сразу в режиме Конструктора, не пользуясь средствами автоматического создания отчетов и Мастером отчетов Access. Для этого достаточно дважды щелкнуть по ярлыку Создание отчета в режиме конструктора (Create report in Design view) над списком отчетов в окне базы данных.

### Структура отчета

Отчет состоит из нескольких частей, называемых *разделами*. Рассмотрим их назначение. Разделы (sections) отчета включают заголовок отчета, примечание отчета, область данных отчета, а также верхний и нижний колонтитулы страниц отчета. Заголовок и примечание отчета, верхний и нижний колонтитулы страницы не являются обязательными элементами структуры отчета. Эти

разделы можно добавить или удалить из отчета с помощью соответствующей команды меню Вид (View) в режиме Конструктора.

*Раздел заголовка* отчета определяет высоту области заголовка отчета. Этот раздел может содержать текст, графику и другие элементы управления, которые будут выводиться в верхней части отчета. В многостраничном отчете раздел заголовка отображается только на первой странице.

*Раздел области данных* предназначен для отображения записей источника данных отчета. Этот раздел может содержать элементы управления для отображения данных из каждой записи в источнике — таблице и запросе.

*Раздел примечаний отчета* определяет высоту области примечания отчета и добавляется вместе с разделом заголовка отчета. При печати многостраничного отчета примечание будет отображено только внизу последней страницы.

Для добавления областей заголовка и примечания отчета выберите команду Вид, Заголовок/примечание отчета (View, Report Header/Footer). Если отчет уже содержит данные области, то команда меню Заголовок/примечание отчета (Report Header/Footer) будет отмечена флажком. В этом случае команда Вид, Заголовок/примечание отчета (View, Report Header/Footer) позволяет удалить области заголовка и примечания из отчета.

**Замечание.** При удалении областей заголовка или примечания отчета, содержащих текст либо другие элементы управления, Access выводит диалоговое окно, предупреждающее об удалении содержимого заголовка и примечания.

При печати многостраничного отчета содержимое разделов верхнего и нижнего колонтитулов страниц отчета выводится на каждой странице. Чтобы добавить к отчету верхний и нижний колонтитулы, выберите команду Вид, Колонтитулы (View, Page Header/Footer). Если отчет уже содержит эти области, то команда меню Колонтитулы (Page Header/Footer) будет отмечена флажком. Для удаления областей верхнего и нижнего колонтитулов страниц отчета снова выберите команду Вид, Колонтитулы (View, Page Header/Footer).

Поскольку отчеты, как правило, включают группировку записей и вывод промежуточных итогов, в них могут присутствовать дополнительные разделы Заголовок группы (Group Header) и Примечание группы (Group Footer). В зависимости от количества уровней группировки в отчете может быть разное количество таких разделов.

### **Инструментальные средства режима Конструктора**

Окно Конструктора отчетов состоит из следующих основных элементов:

- Панель инструментов Конструктор отчетов (Report Design). Кнопки этой панели дублируют некоторые команды меню режима Конструктора отчетов.
- Панель инструментов Формат (форма/отчет) (Formatting (Form/Report)). Кнопки данной панели облегчают форматирование эле-

ментов управления (например, выбор цвета и стиля рамок элементов управления или параметров оформления текста).

- Панель инструментов Панель элементов (Toolbox), с помощью которой в отчет добавляются элементы управления.
- Вертикальная и горизонтальная линейки позволяют производить точную разметку отчета и его разделов.
- Крайняя правая вертикальная линия позволяет определить ширину отчета — расстояние до его правой границы.
- Крайняя нижняя горизонтальная линия позволяет определить высоту нижнего поля отчета.
- С помощью вертикальной и горизонтальной полос прокрутки можно просматривать части отчета, находящиеся за границами области окна Конструктора отчетов.

### **Изменение свойств отчета и его элементов**

Элементами отчета являются разделы отчета и добавленные в них элементы управления. И сам отчет, и его отдельные элементы обладают целым рядом свойств, которые определяют их внешний вид, поведение и данные, которые они отображают. Эти свойства доступны для изменения в режиме Конструктора в специальном окне свойств. Чтобы отобразить окно свойств отчета или его элемента:

1. Откройте отчет в режиме Конструктора. Для этого в окне базы данных щелкните по ярлыку Отчеты (Reports) и выделите в списке отчетов нужный отчет, затем нажмите кнопку Конструктор (Design).
2. Выделите отчет или его элемент и нажмите кнопку Свойства (Properties) на панели инструментов Конструктор отчетов (Report Design), или выберите команду Вид, Свойства (View, Properties), или щелкните по области выделения всего отчета или его элемента правой кнопкой и выберите команду Свойства (Properties) в . контекстном меню.

Например, чтобы отобразить окно свойств отчета, щелкните по квадратику в левом верхнем углу отчета и нажмите кнопку Свойства (Properties) на панели инструментов Конструктор отчетов (Report Design).

На вкладке Макет (Format) окна свойств отчета или его элемента сгруппированы свойства, управляющие внешним видом отчета или его элемента. Например, значение свойства Ширина (Width) определяет ширину отчета или выделенного элемента управления, а в поле Подпись (Caption) задается название, отображаемое в заголовке отчета в режиме его просмотра. На этой же вкладке располагаются поля свойств, позволяющих задать цвет, толщину и стиль рамки элемента управления, определить рисунок, используемый в качестве фона отчета, и настроить его отображение, а также многие другие параметры, отвечающие за внешний вид отчета или его элементов.



Вкладка Данные (Data) позволяет указать источник данных для отчета или выделенного элемента управления, а также настроить их использование.

С помощью вкладки События (Event) окна свойств отчета или его раздела можно определить действия, выполняемые, когда происходят определенные события, например, открытие или закрытие отчета, его активизация или возникновение ошибки.

На вкладке Другие (Other) собраны свойства, которые нельзя отнести ни к одной из перечисленных групп свойств. Например, с помощью этой вкладки окна свойств отчета можно определить привязанные к нему файл справки и панель инструментов, способ группировки данных для отчета. Эта же вкладка в окне свойств элемента управления позволяет, например, изменить его имя или ориентацию отображаемого в нем текста.

На вкладке Все (All) собраны все свойства отчета или его элемента, размещенные на четырех основных вкладках окна свойств. Поскольку отчет и каждый его элемент в отдельности обладают достаточно большим количеством свойств, то список всех свойств на этой вкладке будет достаточно длинным и нахождение нужного свойства на вкладке может оказаться затруднительным.

### **Оформление страниц отчета**

При создании отчета важно не только определить данные, которые должны быть представлены в отчете, но и нужным образом оформить отчет как документ: правильно разбить его на страницы, пронумеровать страницы, выделить наиболее важные данные. В этом разделе будут рассмотрены вопросы, касающиеся оформления страниц отчета.

#### **Добавление разрывов и нумерации страниц**

Если не все нужные данные помещаются на одной странице, Access позволяет вставить принудительный разрыв страницы, чтобы не разрывать группы данных и не нарушать структуру отчета. Для этого:

1. Откройте отчет в режиме Конструктора.
2. На панели элементов нажмите кнопку Разрыв страницы (Page Break). В тех областях отчета, где можно вставить разрыв страницы, курсор мыши будет принимать вид маленького креста, к которому прикреплена страница.
3. Поместите указатель мыши в то место, где необходимо разорвать страницу, и щелкните левой кнопкой мыши. Появится символ разрыва страницы.

Теперь вся информация, расположенная после вставленного разрыва страницы, будет выводиться на новой странице. При необходимости использования принудительных разрывов страниц и в других местах отчета повторите описанную выше процедуру.

Вставленный разрыв страницы можно перемещать в пределах всего отчета. Для этого выделите его, щелкнув по нему левой кнопкой мыши либо выбрав

его название в списке Объект (Object) панели форматирования, и с помощью мыши перетащите в новое место.

Чтобы удалить разрыв страницы, выделите его и нажмите клавишу <Delete> либо выберите команду Правка, Удалить (Edit, Delete).

После того как отчет разбит на страницы, их необходимо пронумеровать. Access 2002 предоставляет несколько возможностей нумерации многостраничных отчетов. Номера страниц могут быть двух форматов, выводиться внизу или вверху каждой страницы с различным выравниванием. Для вставки номеров страниц:

1. Откройте отчет в режиме Конструктора.
2. Выберите команду Вставка, Номера страниц (Insert, Page Numbers).
3. В появившемся диалоговом окне Номера страниц (Page Numbers) установите нужные параметры и нажмите кнопку ОК.

В разделе нижнего или верхнего колонтитулов (в зависимости от выбранного в окне Номера страниц (Page Numbers) значения) появится текстовое поле, источником данных которого будет выражение, задающее номер страницы в выбранном формате. Как и любое другое текстовое поле, номер страницы можно перемещать и форматировать, изменять его свойства и размер.

#### **Использование в отчете линий и прямоугольников**

Чтобы сделать отчет более эффектным и наглядным, Access 2002 позволяет использовать в нем линии и прямоугольники. С помощью этих элементов можно выделять заголовки и группы данных.

Чтобы добавить к отчету линию:

1. Откройте отчет в режиме Конструктора.
2. На панели элементов нажмите кнопку Линия (Line). В тех областях отчета, где можно вставить линию, курсор мыши будет принимать вид маленького креста, к которому прикреплена линия.
3. Щелкните левой кнопкой мыши в том месте отчета, где должна начинаться линия, и, не отпуская кнопку мыши, нарисуйте линию нужной длины, затем отпустите кнопку мыши.

Нарисованную линию можно перемещать, изменять ее размеры и свойства с помощью тех же приемов, что и для любого другого элемента управления. Чтобы удалить линию, выделите ее и нажмите клавишу <Delete>.

Заклучив в прямоугольник несколько элементов управления, можно выделить соответствующую группу данных:

1. Откройте отчет в режиме Конструктора.
2. На панели элементов нажмите кнопку Прямоугольник (Rectangle). В тех областях отчета, где можно вставить прямоугольник, курсор мыши будет принимать вид маленького креста, к которому прикреплен прямоугольник.
3. Щелкните левой кнопкой мыши в том месте отчета, где должен располагаться верхний левый угол прямоугольника, и, не отпуская кнопку

мышью, перетащите указатель мыши в нижний правый угол прямоугольника, затем отпустите кнопку мыши. Внутри нарисованного прямоугольника должны попасть элементы управления, соответствующие той группе данных, которую нужно выделить.

Нарисованный прямоугольник можно перемещать, изменять его размеры и свойства с помощью тех же приемов, что и для любого другого элемента управления. Чтобы удалить прямоугольник, выделите его и нажмите клавишу <Delete>.

Чтобы попавшие в нарисованную прямоугольную область данные еще больше обращали на себя внимание, нужно отформатировать нарисованный прямоугольник:

1. Выделите прямоугольник, щелкнув по нему левой кнопкой мыши.
2. На панели инструментов Конструктор отчетов (Report Design) нажмите кнопку Свойства (Properties).
3. В появившемся окне свойств раскройте вкладку Макет (Format).
4. Чтобы изменить параметры границы прямоугольника, измените значения свойств Тип границы (Border Style), Цвет границы (Border Color) и Ширина границы (Border Width).
5. Чтобы прямоугольник не перекрывал данные, которые он выделяет, установите для свойства Тип фона (Back Style) значение Прозрачный (Transparent) или переместите непрозрачный, залитый цветом, прямоугольник на задний план (по отношению к элементам управления с данными) с помощью команды Формат, На задний план (Format, Send to Back).
6. Чтобы сделать прямоугольник более эффектным и еще больше выделить заключенные в него данные, выберите один из элементов раскрывающегося списка свойства Оформление (Special Effect). С помощью этого свойства прямоугольник можно сделать, например, приподнятым, утопленным либо добавить к нему тень.

### **Создание в отчете вычисляемых полей**

Вычисляемые поля предназначены для отображения в отчетах значений выражений на основе исходных данных. Вычисляемые поля в отчетах можно использовать для получения итоговых значений или проведения специальных вычислений, например, чтобы отобразить общую цену заказа, вычислив произведение количества товара на цену одного изделия. Рассмотрим процедуру создания вычисляемых полей на примере отчета "Товары на складе", созданного ранее.

Чтобы добавить к отчету вычисляемое поле "Сумма":

1. Откройте отчет "Товары на складе" в режиме Конструктора.
2. На панели элементов нажмите кнопку Надпись (Label) и поместите новую надпись справа от надписи "На складе" (Units in Stock) в разделе

верхнего колонтитула. Введите в качестве текста надписи слово "Сумма" (Sum).

3. Чтобы создать вычисляемое поле "Сумма" (Sum), нажмите кнопку Поле (Text Box) на панели элементов и добавьте текстовое поле справа от текстового поля "НаСкладе" (UnitsInStock) в разделе данных. Удалите метку нового текстового поля.
4. Щелкните внутри текстового поля и введите в качестве его значения выражение:  
$$= [\text{НаСкладе}] * [\text{Цена}]$$

Откройте окно свойств поля, дважды щелкнув по нему, и на вкладке Макет (Format) в поле со списком свойства Формат поля (Format) выберите значение Денежный (Currency).

**Замечание.** Ввести выражение в текстовое поле можно несколькими способами: выделите его и нажмите комбинацию клавиш <Shift>+<F2>, чтобы войти в режим редактирования содержимого поля, или сделайте двойной щелчок по полю для отображения диалогового окна Свойства (Properties), а затем введите выражение как значение свойства Данные (Source).

5. Повторите шаги 3 и 4 для того, чтобы создать текстовое поле в разделе примечаний группы "Категория" (Category), но в качестве выражения для вычисления промежуточного значения следует ввести:  $=\text{Sum}([\text{НаСкладе}] * [\text{Цена}])$  ( $=\text{Sum}([\text{UnitsInStock}] * [\text{UnitPrice}])$ ). Нажмите кнопку Полужирный (Bold) на панели форматирования, чтобы установить полужирный шрифт для текста в новом поле. Откройте диалоговое окно Свойства (Properties) и раскройте в нем вкладку Другие (Other), в поле Имя (Name) введите имя текстового поля "Сумма" (Sum). Затем раскройте вкладку Макет (Format) и в поле со списком свойства Формат поля (Format) выберите значение Денежный (Currency).
6. Поскольку ширина области данных нашего отчета увеличилась, уменьшите ширину полей страницы, чтобы все поля поместились на одной странице. Для этого выберите команду Файл, Параметры страницы (File, Page Setup) и задайте вместо ширины левого и правого полей, заданных по умолчанию, значение 10 мм.
7. Нажмите кнопку Вид (View) на панели инструментов Конструктор отчетов (Report Design), чтобы оценить результаты своей работы. Отчет будет выглядеть так, как показано на рис. 12.3. К прежним данным добавился крайний правый столбец со значениями стоимости каждого товара в имеющемся количестве на складе и промежуточных итоговых сумм по стоимости товаров для каждой категории.

**Замечание.** Если при переходе в режим Просмотра отчета появляется диалоговое окно Введите значение параметра (Enter Parameter Value), это означает, что в выражениях неправильно заданы названия одного или нескольких

полей. Нажмите кнопку Отмена (Cancel) и проверьте свойства, установленные на шагах 4 и 5.

Категория	Код товара	Марка	Название	Цена	На складе	Сумма
Кондитерские изд						<b>467 649,00p.</b>
	2	Pavlova	Pavlova, Ltd.	785,25p.	29	22 772,25p.
	5	Teatime Chocolate Biscui	Specialty Biscuits, Ltd.	414,00p.	25	10 350,00p.
	6	Sir Rodney's Marmalade	Specialty Biscuits, Ltd.	645,00p.	40	145 800,00p.
	7	Sir Rodney's Scones	Specialty Biscuits, Ltd.	450,00p.	3	1 350,00p.
	11	NuNuCa Nuss-Nougat-Cr	АО Германия-Россия	630,00p.	76	47 880,00p.
	12	Gumbar Gummibarchen	АО Германия-Россия	405,35p.	15	21 080,25p.
	13	Schoggi Schokolade	АО Германия-Россия	975,50p.	49	96 799,50p.
	22	Scottish Longbreads	Specialty Biscuits, Ltd.	562,50p.	6	3 375,00p.
	48	Zaanse koeken	Zaanse Snoepfabriek	427,50p.	36	15 390,00p.
	49	Chocolade	Specialty Biscuits, Ltd.	573,75p.	15	8 606,25p.
	50	Maxilaku	Karkki Oy	900,00p.	10	9 000,00p.
	51	Valkoinen suklaa	Karkki Oy	731,25p.	65	47 531,25p.
	63	Tarte au sucre	Forots d'Erables	218,50p.	17	37 714,50p.
<b>Итого для Категория' = Кондитерские изделия (13 записей)</b>						
<b>Sum</b>						<b>386</b>

Рис. 12.3. Отчет «Товары на складе» с вычисляемыми полями

### Сортировка и группировка данных

Отчеты становятся более понятными и структурированными, если данные в них организованы в группы и подгруппы. В разд. "Создание отчета с помощью Мастера отчетов" было показано, как задать начальные свойства группировки и сортировки данных. Однако структуру отчета можно изменить. Для этого используется диалоговое окно Сортировка и группировка (Sorting and Grouping).

Чтобы открыть диалоговое окно Сортировка и группировка (Sorting and Grouping), откройте отчет в режиме Конструктора и нажмите кнопку Сортировка и группировка (Sorting and Grouping) на панели инструментов Конструктор отчетов (Report Design).

В этом диалоговом окне можно определить поля или выражения, по которым Access будет группировать данные. Кроме того, сгруппированные данные

можно сортировать по возрастанию или по убыванию. Символ Сортировка и группировка (Sorting and Grouping) на кнопке выделения в левом сером столбце этого диалогового окна указывает на то, что данное поле или выражение (находящееся справа от значка) используется для группировки записей. В данном примере поле "Категория" (CategoryName) используется для группировки строк отчета, а сортировка строк в пределах группы выполняется по полю "КодТовара" (ProductID).

Access позволяет группировать данные двумя способами: по категориям, при этом каждой категории должно соответствовать уникальное значение, и по диапазону значений, который может быть как числовым, так и алфавитным. При этом для группировки данных можно использовать не только значения полей, но и выражения.

**Замечание.** Уровней группировки может быть несколько, т. е. можно создавать подгруппы внутри группы строк. Примером отчета с несколькими уровнями группировки, одним из которых является группировка по значению выражения в качестве категории, является отчет "Продажи по сотрудникам и странам" (Employee Sales by Country) — один из отчетов базы данных "Борей". В нем используются группы (страны) и подгруппы (имена сотрудников).

### Применение шаблона отчетов

Как правило, отчеты, подготавливаемые в одной базе данных, должны быть выполнены в одном стиле. Потратив достаточно много времени на создание и форматирование одного отчета, не хотелось бы терять еще столько же времени на разработку остальных. В качестве решения этой проблемы Access предлагает использовать созданный отчет в качестве шаблона для всех новых отчетов.

Чтобы создать отчет, который послужит шаблоном для всех последующих отчетов:

1. В окне База данных щелкните по ярлыку Отчеты (Reports) и нажмите кнопку Создать (New).
2. В появившемся диалоговом окне Новый отчет (New Report) выделите в списке способов создания элемент Конструктор (Design View) и нажмите кнопку ОК. Откроется окно Конструктора отчетов.
3. Если необходимо, добавьте к отчету разделы верхнего и нижнего колонтитулов, раздел примечания и раздел заголовка отчета. С помощью мыши установите необходимую ширину отчета и высоту его разделов.
4. Выделите весь отчет, щелкнув левой кнопкой мыши в левом верхнем углу в месте пересечения линеек либо выбрав в списке Объект (Object) панели форматирования элемент Отчет (Report).
5. На панели инструментов Конструктор отчетов (Report Design) нажмите кнопку Свойства (Properties).

6. В появившемся окне свойств отчета раскройте вкладку Макет (Format). Определите на этой вкладке те значения свойств, которые должны иметь все создаваемые отчеты. Например, можно указать файл рисунка, который будет использоваться в качестве фона отчетов, определить параметры отображения рисунка, установить точную ширину отчета в сантиметрах, задать частоту сетки, по которой выравниваются объекты отчета.

**Замечание.** Для контроля вносимых изменений используйте режим Предварительного просмотра. Чтобы перейти в этот режим, на панели инструментов Конструктор отчета (Report Design) нажмите кнопку Вид (View), на которой отображен элемент списка режимов Предварительный просмотр (Print Preview).

7. Закройте окно свойств и сохраните отчет: выберите команду Файл, Сохранить (File, Save) и в появившемся диалоговом окне введите название отчета, например Образец. Закройте отчет, щелкнув по кнопке с изображением крестика в правом верхнем углу отчета.
8. Выберите команду Сервис, Параметры (Tools, Options). В появившемся диалоговом окне Параметры (Options) раскройте вкладку Формы и отчеты (Forms/Reports).
9. В текстовом поле Шаблон отчета (Report template) укажите название созданного отчета (в нашем примере: Образец) и нажмите кнопку ОК.

Теперь при создании отчета в режиме Конструктора в качестве шаблона будет использоваться указанный отчет, т. е. по умолчанию свойства создаваемого отчета будут иметь значения, определенные в шаблоне.

### **Отображение в отчете значений параметров**

Если отчет создается на базе запроса с параметрами, то в отчете можно выводить значения параметров, которые вводит пользователь, когда открывает отчет.

С этой целью создайте запрос с параметром. Из таблицы «Заказы» добавьте поля: КодЗаказа, ДатаРазмещения, СтранаПолучателя, а из таблицы «Сотрудники» - Фамилия. В поле «Условие отбора» введите параметр запроса: [Введите страну]. Запустите запрос, проверьте его работу и сохраните запрос под именем «Заказы по странам».

Теперь на основе этого запроса создайте отчет любыми удобными для вас средствами.

1. В отчет добавьте поля: КодЗаказа, ДатаРазмещения, Фамилия. Сгруппируйте заказы по фамилии сотрудника.
2. В области заголовка отчета в режиме Конструктора с помощью кнопки панели элементов создайте текстовое поле. В качестве подписи для поля введите «Страна»
3. Отформатируйте его, установив цвет и шрифт заголовка отчета.

4. В окне Свойства раскройте вкладку Данные и выделите созданный элемент управления. Установите значение свойства Данные равным =[Введите страну]  
Таким образом, в качестве источника данных для этого элемента вы ввели наименование параметра базового запроса.
5. Создадим вычисляемое поле, показывающее общее число заказов по этой стране. В области примечаний отчета в режиме Конструктора с помощью кнопки панели элементов создайте текстовое поле. В качестве подписи для поля введите «Всего заказов». В окне Свойства раскройте вкладку Данные и выделите созданный элемент управления. Установите значение свойства Данные равным =Count([КодЗаказа])
6. Откройте отчет в режиме предварительного просмотра и проверьте, что в поле заголовка отчета выводится то значение параметра, которое вы вводите.



## **8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

### **РАБОТА №1. Описание предметной области.**

#### **1 Введение**

1.1 Название организации (полное и краткое).

1.2 Положение на рынке, (основные конкуренты, аналогичные организации, масштабы деятельности, сфера влияния, доля рынка, количество клиентов, поставщиков, имущества, персонала, виды и объемы выпуска продукции / оказания услуг).

1.3 Основные тенденции развития организации и отрасли, сильные и слабые стороны организации.

1.4 Цели автоматизации, критерии достижения целей.

1.5 Границы обследования, границы планируемой автоматизации

#### **2 Контекст (окружение)**

2.1 Контекстная диаграмма (внешние сущности).

2.2 Основные контрагенты (поставщики, покупатели (клиенты), банк, государственные органы, головные и дочерние организации, партнеры, другие контрагенты)

2.3 Описание контекстной диаграммы.

2.4 Недостатки и достоинства существующего положения дел, требования к новой системе.

#### **3 Отделы внутри фирмы (или основные процессы на фирме)**

3.1 Организационная структура фирмы с точностью до отдела (диаграмма)

3.2 Контактные лица в отделах, телефоны, e-mail

3.3 Основные функции каждого отдела

3.4 Диаграмма DFD «Отделы»

3.5 Описание диаграммы (документооборот между отделами с точностью до документа)

3.6 Общие для нескольких отделов хранилища данных

3.7 Недостатки и достоинства существующего положения дел, требования к новой системе

#### **4. Процессы в отделе (рабочие места, участки работ)**

4.1 Организационная структура отдела с точностью до должности (диаграмма)

4.2 Контактные лица сотрудников, телефоны, e-mail

4.3 Основные функции каждого рабочего места

4.4 Диаграмма DFD «Процессы в отделе»

4.5 Описание диаграммы (документооборот между рабочими местами с точностью до документа)

4.6 Общие для нескольких рабочих мест хранилища данных

4.7 Недостатки и достоинства существующего положения дел, требования к новой системе.

## **РАБОТА №2. Определение границ, выявление информационных запросов пользователей.**

### **1. Функции рабочего места**

1.1 Диаграмма DFD «Функции рабочего места»

1.2 Описание диаграммы (описание процессов, документов, хранилищ по соответствующим методикам).

1.3 Недостатки и достоинства существующего положения дел, требования к новой системе

### **2. Описание существующей системы автоматизации**

2.1 Недостатки и достоинства существующих автоматизированных систем

2.2 Требования заказчика по взаимодействию с существующими автоматизированными системами (требования по совместимости, по обмену информацией, по переносу данных из прежних систем в новую)

2.3 Входящие и исходящие документы для существующих автоматизированных систем

### **3. Формулировка требований заказчика к будущей системе**

3.1 Объединение всех собранных требований и их описание

3.2 Проверка требований на непротиворечивость друг другу, согласование и утверждение компромиссного варианта.

3.3 Проверка каждого требования на корректность и возможность выполнения (достаточность и определенность входной информации, ясный алгоритм обработки информации, четкая структура выходной информации)

## **РАБОТА №3. Разработка спецификаций сущностей, атрибутов.**

Для выбранной предметной области необходимо определить сущности, соответствующие выделенным объектам предметной области, и указать их смысловое описание.

Пусть проектируемая база данных содержит следующие сущности:

Сущность «РабочееМесто» хранит информацию о типе рабочего места.

Сущность «Сотрудник» хранит информацию о сотрудниках: фамилия, имя, отчество, должность, телефон рабочий, телефон сотовый.

Сущность «Комплектующая» хранит данные о комплектующей.

Сущность «Поставщик» хранит информацию о поставщике комплектующей или организации, проводившей ремонт комплектующей.

Сущность «ДокументПоступления» хранит информацию о поступлении комплектующей.

Сущность «ДокументСписания» хранит информацию о причине и дате списания комплектующей.

Сущность «ДокументПеремещения» хранит информацию о перемещении комплектующей с одного рабочего места на другое или на склад.

Сущность «ДокументСдачиНаРемонт» хранит информацию о сдаче комплектующей на ремонт.

Сущность «ДокументПолучСРемонта» хранит информацию о получении комплектующей с ремонта.

Спецификацию атрибутов для каждой сущности удобно представить в виде таблиц.

Атрибуты сущности «РабочееМесто»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодРабМеста</u>	Идентификационный номер, который однозначно определяет рабочее место	>0	-	1
ТипРабМеста	Тип рабочего места	АРМ, НРМ	-	АРМ
ОтветственноеЛицо (КодСотрудника)	Идентификационный номер, который однозначно определяет сотрудника	>0	-	7
НаименованиеРабМеста	Наименование рабочего места	-	-	ОМ1

Атрибуты сущности «Сотрудник»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодСотрудника</u>	Идентификационный номер, который однозначно определяет сотрудника	>0	-	7
Фамилия	Фамилия сотрудника	-	-	Иванова
Имя	Имя сотрудника	-	-	Татьяна
Отчество	Отчество сотрудника	-	-	Александровна
Должность	Наименование должности	-	-	програмист
Подразделение	Наименование подразделения	-	-	Отдел маркетинга
РабочийТелефон	Рабочий телефон сотрудника	-	-	34-45-12

Атрибуты сущности «Комплектующая»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодКомплектующей</u>	Идентификационный номер, который однозначно определяет комплектующую	>0	-	5

СостояниеКомплект	Наименование состо- яния комплектующей	Работает, На ре- монте, Списано	-	На ремонте
РабМесто	Идентификационный но- мер, который однозначно определяет рабочее место	>0	-	1
КодДокПоступления	Идентификационный но- мер, который однозначно определяет документ по- ступления комплектую- щей	>0	-	12
КодДокСписания	Идентификационный но- мер, который однозначно определяет документ списания комплектую- щей	>0	-	12
СерийныйНомер	Серийный номер комплектующей	-	-	1216377598
Цена	Цена комплектующей	-	рубли	153 рубля
Модель	Наименование модели комплектующей	-	-	ASUS A- P800
ТипКомплект	Наименование типа комплектующей	-	-	мате- ринская плата
КоличествоРемонтов	Количество производи- мых ремонтов комплек- тующей	-	-	2

Атрибуты сущности «Поставщик»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодПоставщика</u>	Идентификационный номер, который одно- значно определяет по- ставщика	>0	-	5
НаименованиеПостав- щика	Наименование постав- щика	-	-	ООО «Фор- моза»
АдресОфиса	Адрес офиса фирмы - поставщика	-	-	ул. 50 Лет Октября, 27
ТелефонОфиса	Телефон офиса фирмы - поставщика	-	-	44-95-01
АдресСервЦентра	Адрес сервисного цен- тра фирмы - поставщи- ка	-	-	ул. 50 Лет Октября, 27
ТелефонСервЦентра	Телефон сервисного центра фирмы - постав- щика	-	-	44-95-01

Атрибуты сущности «ДокументПоступления»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодДокПоступления</u>	Идентификационный номер, который однозначно определяет документ поступления комплектующей	>0	-	12
КодПоставщика	Идентификационный номер, который однозначно определяет поставщика	>0	-	5
ДокументПрихода	Наименование и номер документа, подтверждающий факт приобретения комплектующей	-	-	Накладная № 6-76501-541
ДатаПоступления	Дата поступления комплектующей	-	-	02.11.2005

Атрибуты сущности «ДокументСписания»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодДокСписания</u>	Идентификационный номер, который однозначно определяет документ списания комплектующей	>0	-	12
Наименование Док	Наименование документа списания	-	-	№ 2-2006-357
ДатаСписания	Дата списания комплектующей	-	-	12.11.2005
ПричинаСписания	Причина списания комплектующей	-	-	устаревание

Атрибуты для сущности «ДокументПеремещения»

Название атрибута	Значение атрибута	Диапазон значений	Единицы Измерения	Пример атрибута
<u>КодДокПеремещения</u>	Идентификационный номер, который однозначно определяет документ перемещения комплектующей	>0	-	23
Наименование-Док	Наименование документа перемещения	-	-	№ 4-185-057
Куда (КодРаб-Места)	Идентификационный номер, который однозначно определяет рабочее место	>0	-	11

Атрибуты для сущности «ДокументСдачиНаРемонт»

Название атрибута	Значение атрибута	Диапазон значений	Единицы измерения	Пример атрибута
<u>КодДокСдачиНаРем</u>	Идентификационный номер, который однозначно определяет документ сдачи комплектующей на ремонт	>0	-	23
Наименование-Док	Наименование документа сдачи на ремонт	-	-	№ 1-1105-133
КодКомплектующей	Идентификационный номер, который однозначно определяет комплектующую	>0	-	6
ПричинаСдачиНаРем	Причина сдачи комплектующей на ремонт			модем не включается
№Квитанции	Номер квитанции о приеме изделия	>0		№12365321
КодПоставщика	Идентификационный номер, который однозначно определяет поставщика	>0	-	5
ДатаСдачи	Дата сдачи комплектующей на ремонт	-	-	02.11.2005

Атрибуты для сущности «ДокументПолученияСРемонта»

Название атрибута	Значение атрибута	Диапазон Значений	Единицы измерения	Пример атрибута
<u>КодДокПолученияСРем</u>	Идентификационный номер, который однозначно определяет документ получения с ремонта комплектующей	>0	-	25
Наименование-Док	Наименование документа получения с ремонта	-	-	№ 1-2105-153
КодПоставщика	Идентификационный номер, который однозначно определяет поставщика	>0	-	5
КодКомплектующей	Идентификационный номер, который однозначно определяет комплектующую	>0	-	5
ОписаниеРабот	Описание ремонтных работ	-	-	
Стоимость	Стоимость услуги ремонта	-	рубли	230 рублей
ДатаПолучения	Дата получения комплектующей с ремонта	-	-	04.11.2005
№НакладнойПолучения	Номер накладной получения комплектующей	-	-	№ 23-6453-71

**РАБОТА №4. Выбор ключей, разработка спецификаций связей типа «сущность - сущность» и «атрибут-атрибут».**

## 1. Выбор ключей.

Идентифицируемым для сущности «РабочееМесто» является атрибут «КодРабМеста», введенный для исключения дублирования, так как он однозначно идентифицирует каждый экземпляр сущности «РабочееМесто».

Идентифицируемым для сущности «Сотрудник» является атрибут «КодСотрудника», так как он однозначно идентифицирует каждый экземпляр сущности «Сотрудники».

Ключевым для сущности «Комплектующая» выберем атрибут «КодКомплектующей», так как он однозначно идентифицирует каждый экземпляр сущности «Комплектующие».

В сущности «Поставщик» ключевым является атрибут «КодПоставщика», который однозначно идентифицирующий каждого поставщика.

В сущности «ДокументПоступления» ключевым является атрибут «КодДокПоступления», который однозначно идентифицирующий каждый документ поступления комплектующих.

Ключевым для сущности «ДокументСписания» выберем атрибут «КодДокСписания», так как он однозначно идентифицирует каждый экземпляр сущности «ДокументСписания».

В сущности «ДокументПеремещения» ключевым является атрибут «КодДокПеремещения», который однозначно идентифицирующий каждый документ перемещения комплектующей.

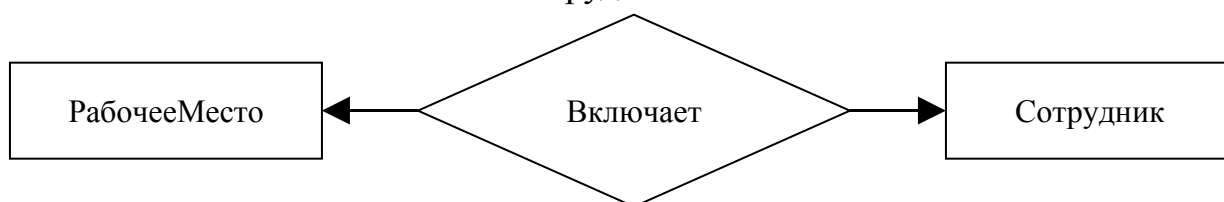
В сущности «ДокументСдачиНаРемонт» ключевым является атрибут «КодДокСдачиНаРем», который однозначно идентифицирующий каждый документ сдачи на ремонт комплектующей.

В сущности «ДокументПолученияСРемонта» ключевым является атрибут «КодДокПолученияСРем», который однозначно идентифицирующий каждый документ получения с ремонта комплектующей.

## 2. Спецификация связей типа «сущность - сущность»

Основное назначение модели «сущность-связь» – семантическое описание предметной области и представление информации для обоснования выбора видов моделей и структур данных, которые в дальнейшем будут использованы в системе.

### 2. 1 Связь «РабочееМесто – Сотрудник»



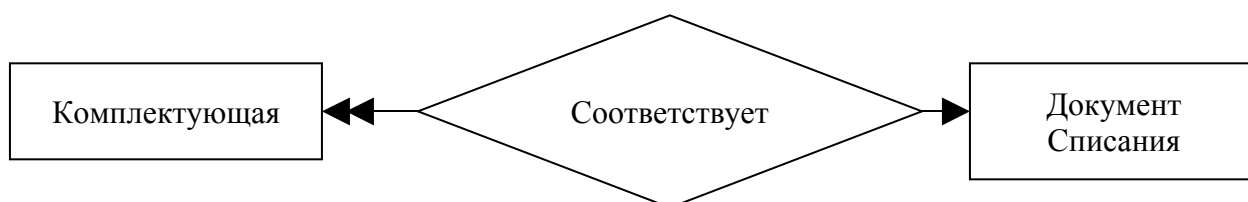
В данном случае получаем связь «один-к-одному», так как каждой записи сущности «РабочееМесто» может соответствовать одна запись сущности «Сотрудник», а каждой записи сущности «Сотрудник» может соответствовать одна

запись сущности «РабочееМесто». То есть, каждое рабочее место включает одного сотрудника, а каждый сотрудник может иметь одно рабочее место.

## 2. 2 Связь «РабочееМесто – Комплектующая»



Получили связь «один-ко-многим», поскольку каждое рабочее место может содержать несколько комплектующих, а каждая комплектующая соответствует лишь одному рабочему месту.



## 2. 3 Связь «Комплектующая – ДокументСписания»

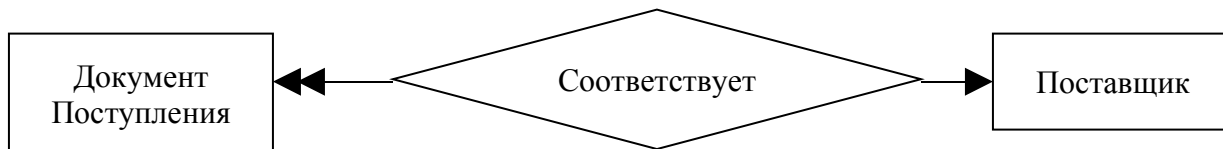
Получили связь «один-ко-многим», поскольку каждой комплектующей соответствует один документ списания, а каждому документу списания может соответствовать несколько комплектующих.

## 2. 4 Связь «Комплектующая – ДокументПоступления»



Получили связь «один-ко-многим», поскольку каждой комплектующей соответствует один документ поступления, а каждому документу списания может соответствовать несколько комплектующих.

## 2. 5 Связь «ДокументПоступления – Поставщик»



Выделили связь «один-ко-многим», поскольку каждому документу поступления соответствует один поставщик, а каждому поставщику может соответствовать несколько документов поступления.

## 2. 6 Связь «РабочееМесто – ДокументПеремещения»





Получили связь «один-ко-многим», поскольку каждое рабочее место может упоминаться в нескольких документах перемещения, а каждый документ перемещения составляется по одному рабочему месту.



#### 2.4.7 Связь «ДокументПеремещения – Комплектующая»

Выделили связь «один-ко-многим», поскольку каждый документ перемещения может описывать одну комплектующую, а каждая комплектующая может упоминаться в нескольких документах перемещения.

#### 2. 8 Связь «ДокументСдачиНаРемонт – Поставщик»



связь «один-ко-многим», поскольку каждый документ сдачи комплектующей на ремонт включает одного поставщика, а каждый поставщик может соответствовать многим документам сдачи на ремонт.

#### 2.9 Связь «Комплектующая – ДокументПолучСРемонта»



Выделили связь «один-ко-многим», поскольку каждая комплектующая может входить в несколько содержаний документа получения с ремонта, а содержание документа содержит данные о получении с ремонта только одной

комплектующей.

## 2. 10 Связь «ДокументПолучСРемонта – Поставщик»



Выделили связь «один-ко-многим», поскольку каждый документ получения комплектующей с ремонта включает одного поставщика, а каждый поставщик может соответствовать многим документам получения с ремонта.

## 2. 11 Связь «Комплектующая – ДокументСдачиНаРемонт»



Выделили связь «один-ко-многим», поскольку каждый документ с сдачи комплектующей на ремонт включает одного поставщика, а каждый поставщик может соответствовать многим документам сдачи на ремонт.

## 3. Спецификация связи «атрибут-атрибут»

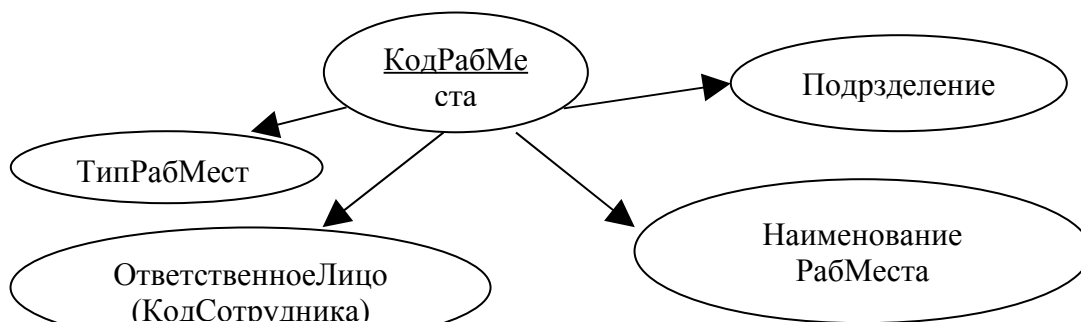


Рисунок 1 – Связи атрибутов сущности «РабочееМесто»

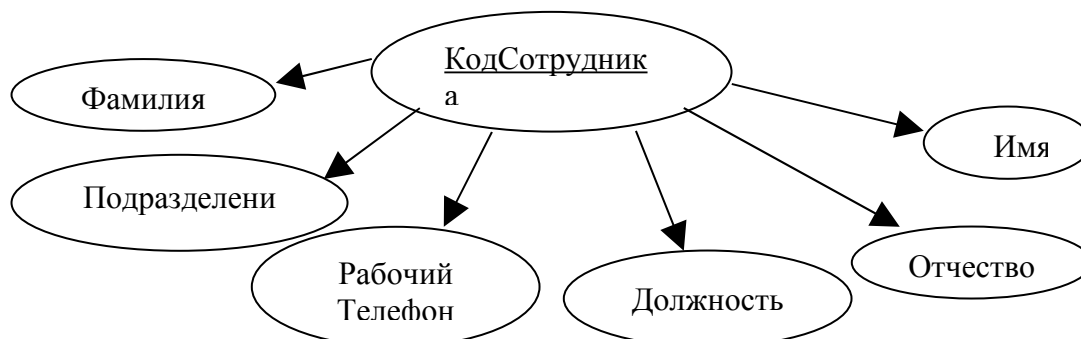


Рисунок 2 – Связи атрибутов сущности «Сотрудник»

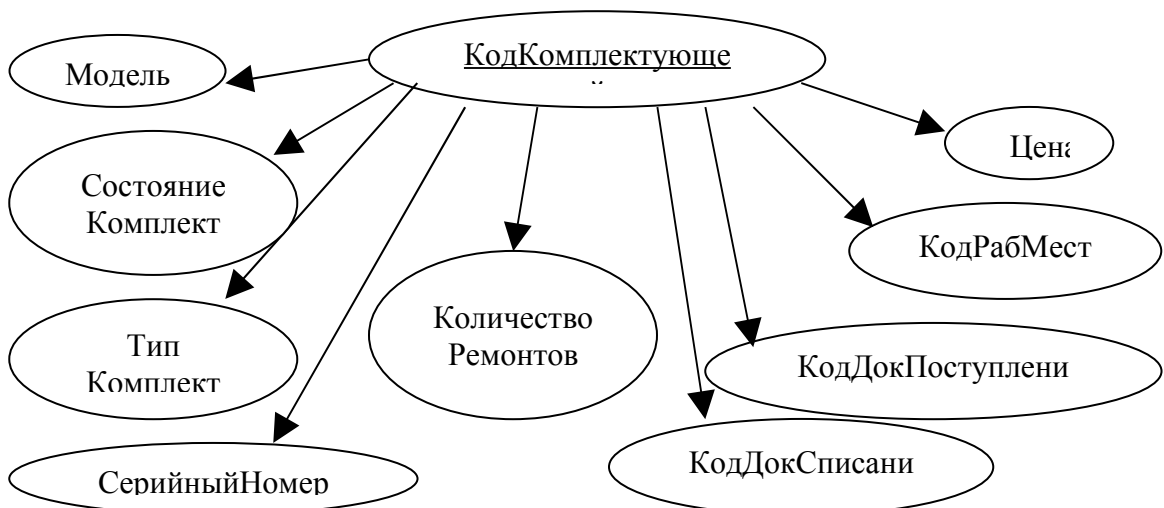


Рисунок 3 – Связи атрибутов сущности «Комплектуящая»



Рисунок 4 – Связи атрибутов сущности «ДокументПеремещения»

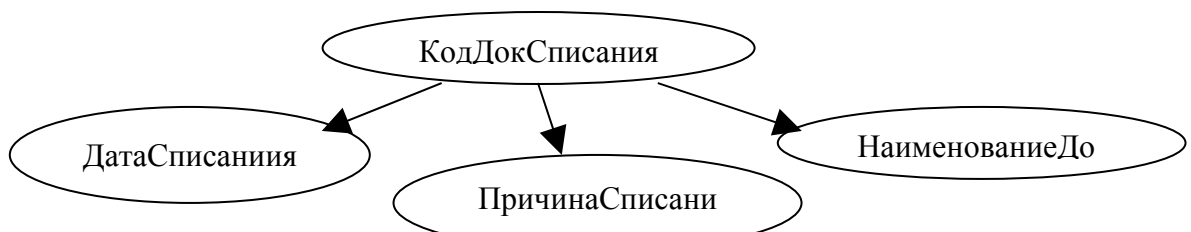


Рисунок 5 – Связи атрибутов сущности «ДокументСписания»



Рисунок 6 – Связи атрибутов сущности «ДокументПоступления»

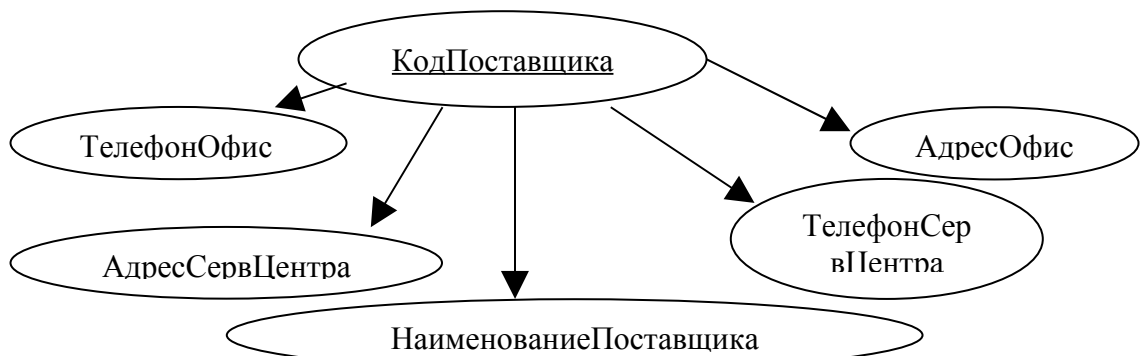


Рисунок 7 – Связи атрибутов сущности «Поставщик»

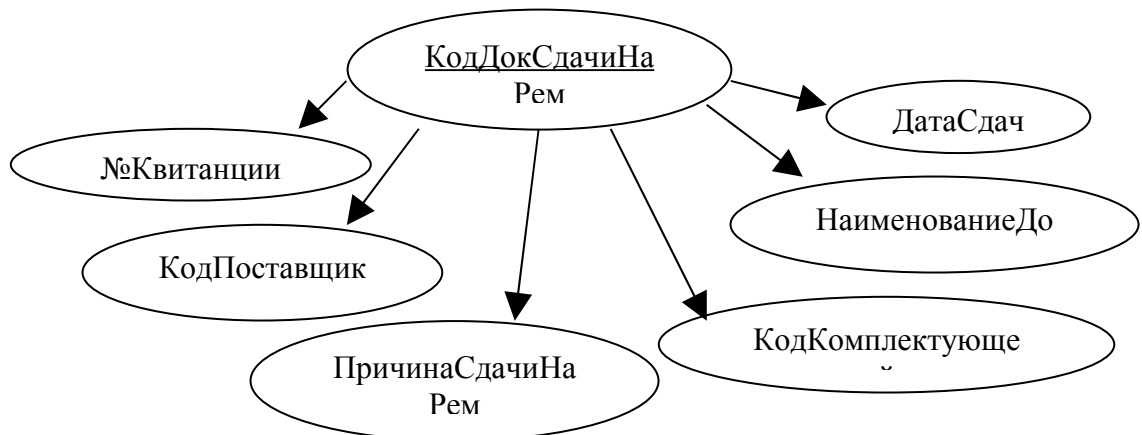


Рисунок 8 – Связи атрибутов сущности «ДокументСдачиНаРемонт»

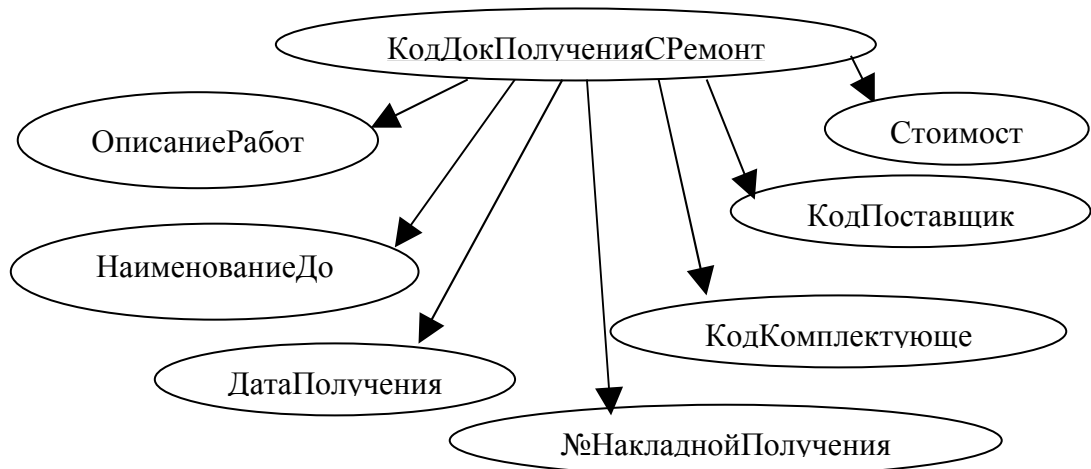


Рисунок 9 – Связи атрибутов сущности «ДокументПолученияСРемонта»

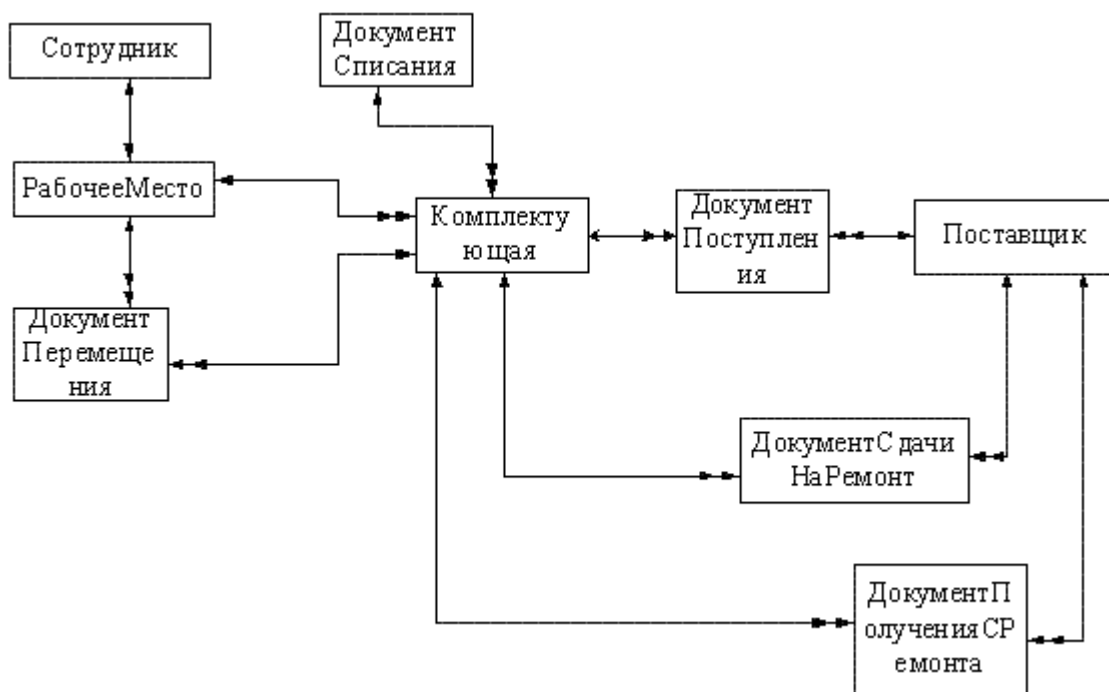
## РАБОТА №5. Создание справочника задач, построение концептуальной инфологической модели

### 1. Справочник задач пользователя

Наименование задачи	Цель решения задачи	Сущности, используемые при решении	Частота решения
---------------------	---------------------	------------------------------------	-----------------

		нии задачи	задачи
Вывод списка комплектующих, находящихся в ремонте	Контроль за местоположением комплектующих	Комплектующая, ДокументСдачиНа-Ремонт	300
Вывод количества комплектующих различных видов, находящихся на складе	Осуществлять контроль за загруженностью склада	Комплектующая, ДокументПоступления	48
Вывод сведений о затратах на ремонт оборудования	Контроль за затратами на ремонт	Комплектующая, ДокументПолученияСРемонта,	12
Получение информации о комплектующей	Анализ состояния комплектующей и принятие решения об её списании	Комплектующая	12
Вывод информации о том, с какими комплектующими работает сотрудник	Обучение правилам работы и контроля за оборудованием	Комплектующая, РабочееМесто, Сотрудник	48
Вывод информации о комплектующих одного типа	Для поиска комплектующей, наиболее подходящей для перемещения	Комплектующая	48
Вывод сведений о комплектующих каждого поставщика	Для составления отчета о проделанной работе	Поставщик, Комплектующая	300
Вывод списка списанных комплектующих	Контроль за комплектующими	Комплектующая, ДокументСписания	12
Вывод сведений о сотруднике	Для связи с сотрудником	Сотрудник, РабочееМесто	12

## 2. Концептуально-инфологическая модель



**РАБОТА №6. Проектирование реляционной логической модели базы данных, составление матрицы частоты совместного использования сущностей на основе справочника задач, оценка объема лишнего чтения, установление дополнительных логических связей.**

Используя данные справочника задач, рассчитаем значения частот совместного использования сущностей. Дополнительные логические связи между сущностями устанавливаются, если частота совместного использования этих сущностей больше средней и имеется большой объем "лишнего" чтения. Для этого нам понадобится матрица суммарной частоты совместного использования сущностей

Сущности	Индексы сущностей								
	1	2	3	4	5	6	7	8	9
1.РабочееМесто	0	60	48	0	0	0	0	0	0
2.Сотрудник	60	0	48	0	0	0	0	0	0
3.Комплектующая	48	48	0	300	48	12	0	300	0
4.Поставщик	0	0	300	48	0	0	0	0	0
5.ДокументПоступления	0	0	48	0	0	0	0	0	0
6.ДокументСписания	0	0	12	0	0	0	0	0	0
7.ДокументПеремещения	0	0	0	0	0	0	0	0	0
8.ДокументСдачиНаРемонт	0	0	300	0	0	0	0	0	0

9.ДокументПолученияСРемонта	0	0	12	0	0	0	0	0	0
-----------------------------	---	---	----	---	---	---	---	---	---

и размерность массивов базы данных

Наименование массива	Размерность массива (в байтах)
1. РабочееМесто	2800
2. Сотрудник	9120
3. Комплектующая	67200
4. Поставщик	5080
5. ДокументПоступления	20500
6. ДокументСписания	93500
7. ДокументПеремещения	70000
8. ДокументСдачиНаРемонт	72500
9.ДокументПолученияСРемонта	54500

Средняя частота совместного использования сущностей определяется по формуле

$$Ч_{\text{ср}} = (Ч_{\text{мин}} + Ч_{\text{макс}}) / 2$$

где  $Ч_{\text{ср}}$  – средняя частота совместного использования сущностей;

$Ч_{\text{мин}}$  – минимальная частота совместного использования сущностей;

$Ч_{\text{макс}}$  – максимальная частота совместного использования сущностей.

Используя данные, получаем, что  $Ч_{\text{ср}} = 150$

Рассмотрим сущности, которые используются с частотой выше средней.

Во-первых, это сущности «Комплектующая» и «Поставщик». Между ними нет связи, а доступ возможен только через сущность «ДокументПоступления». Оценим объем лишнего чтения:  $20500 * 300 = 615 * 10^4$ . Так как эти сущности совместно больше не используются, то необходимость в дополнительной связи отсутствует.

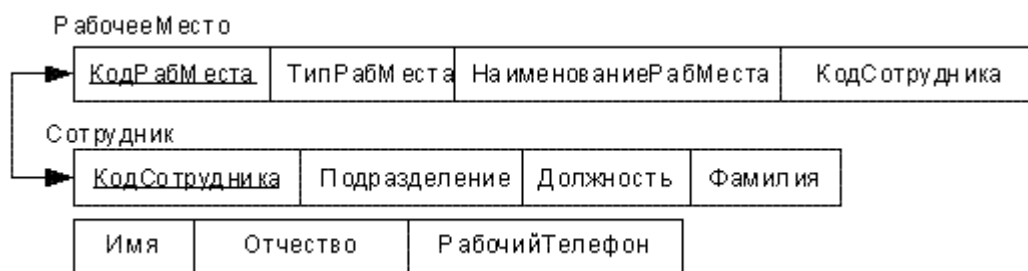
Во-вторых, это сущности «Комплектующая» и «ДокументСдачиНаРемонт», но между ними есть непосредственная связь.

### **РАБОТА №7. Отображение инфологической модели на реляционную модель, получение совокупности отношений реляционной модели.**

#### **1. Отображение инфологической модели на реляционную модель**

Выполним отображение концептуальной инфологической модели на реляционную, пользуясь правилами отображения.

##### **1.1 Связь «РабочееМесто – Сотрудник»**



Данная связь имеет тип «один-к-одному». Исходной будет сущность «РабочееМесто», так как из нее исходит простая связь, следовательно, порожденной является сущность «Сотрудник». По общему правилу ключевой атрибут порожденной сущности добавляется в исходную сущность. Атрибут КодСотрудника сущности «Сотрудник» уже присутствует в сущности «РабочееМесто», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 1

<u>КодРабМеста</u>	ТипРабМеста	НаименованиеРабМеста	КодСотрудника
--------------------	-------------	----------------------	---------------

Отношение 2

<u>КодСотрудника</u>	Подразделение	Должность	Фамилия	Имя	Отчество	РабочийТелефон
----------------------	---------------	-----------	---------	-----	----------	----------------

## 1.2 Связь «РабочееМесто – Комплектующая»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «Комплектующая», так как из нее исходит простая связь, следовательно, порожденной является сущность «РабочееМесто». Атрибут КодРабМеста сущности «РабочееМесто» уже присутствует в сущности «Комплектующая», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 3

<u>КодРабМеста</u>	ТипРабМеста	НаименованиеРабМеста	КодСотрудника
--------------------	-------------	----------------------	---------------

Отношение 4

<u>КодКомплектующей</u>	Модель	КодРабМеста	СостояниеКомплект	ТипКомплект	КодДокПоступления
-------------------------	--------	-------------	-------------------	-------------	-------------------



<u>ющей</u>					
КодДокСписания	СерийныйНо- мер	Цена	КоличествоРе- монтов		

### 1.3. Связь «Комплектующая – ДокументСписания»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «Комплектующая», так как из нее исходит простая связь, следовательно, порожденной является сущность «ДокументСписания». Атрибут КодДокСписания сущности «ДокументСписания» уже присутствует в сущности «Комплектующая», поэтому его повторного дублирования не происходит. В результате получили отношения

#### Отношение 5

<u>Код- Комплекту- ющей</u>	Модель	Код РабМеста	Состояние- Комплект	Тип- Комплект	КодДок Поступления
КодДокСписания	СерийныйНо- мер	Цена	КоличествоРе- монтов		

#### Отношение 6

<u>КодДокСписа- ния</u>	НаименованиеДок	ПричинаСписания	ДатаСписания
-----------------------------	-----------------	-----------------	--------------

### 1.4 Связь «Комплектующая – ДокументПоступления»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «Комплектующая», так как из нее исходит простая связь, следовательно, порожденной является сущность «ДокументПоступления». Атрибут КодДокПоступления сущности «ДокументПоступления» уже присутствует в сущности

«Комплектующая», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 7

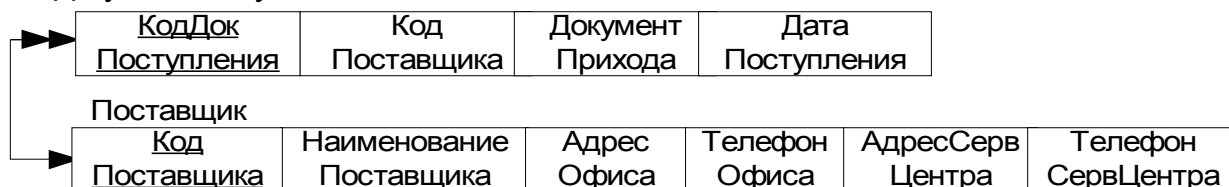
<u>Код-Комплектующей</u>	Модель	Код РабМеста	Состояние-Комплект	Тип-Комплект	КодДок Поступления
КодДокСписания	СерийныйНомер	Цена	КоличествоРемонтов		

Отношение 8

<u>КодДокПоступления</u>	КодПоставщика	ДокументПрихода	ДатаПоступления
--------------------------	---------------	-----------------	-----------------

### 1.5 Связь «ДокументПоступления – Поставщик»

ДокументПоступления



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументПоступления», так как из нее исходит простая связь, следовательно, порожденной является сущность «Поставщик». Атрибут КодПоставщика сущности «Поставщик» уже присутствует в сущности «ДокументПоступления», поэтому его повторного дублирования не происходит. В результате получили отношения

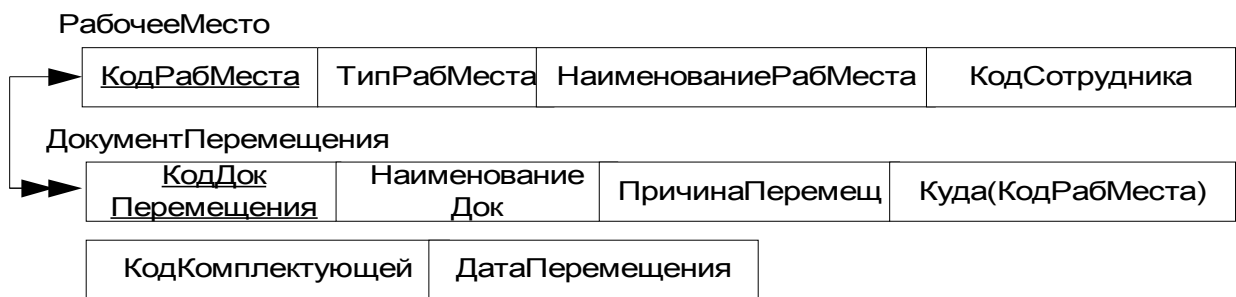
Отношение 9

<u>КодДокПоступления</u>	КодПоставщика	ДокументПрихода	ДатаПоступления
--------------------------	---------------	-----------------	-----------------

Отношение 10

<u>КодПоставщика</u>	Наименование Поставщика	Адрес Офиса	Телефон Офиса	Адрес СервЦентра	Телефон СервЦентра
----------------------	-------------------------	-------------	---------------	------------------	--------------------

### 1.6 Связь «РабочееМесто – ДокументПеремещения»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументПеремещения», так как из нее исходит простая связь, следовательно, порожденной является сущность «РабочееМесто». Атрибут КодРабМеста сущности «РабочееМесто» уже присутствует в сущности «ДокументПеремещения», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 11

<u>КодРабМеста</u>	ТипРабМеста	НаименованиеРабМеста	КодСотрудника
--------------------	-------------	----------------------	---------------

Отношение 12

<u>КодДок Перемещения</u>	Наименование Док	Причина Перемещ	Куда (КодРабМеста)	Код Комплек- ту ющей	Дата Переме- щения
-------------------------------	---------------------	--------------------	-----------------------	-------------------------------	--------------------------

### 1.7 Связь «Комплектующая – ДокументПеремещения»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументПеремещения», так как из нее исходит простая связь, следовательно, порожденной является сущность «Комплектующая». Атрибут КодКомплектующей сущности «Комплектующая» уже присутствует в сущности «ДокументПеремещения», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 13

<u>Код-Комплекту-ющей</u>	Модель	Код РабМеста	Состояние-Комплект	Тип-Комплект	КодДок Поступления
КодДокСписания		СерийныйНо-мер	Цена	КоличествоРе-монтов	

Отношение 14

<u>КодДок Переме-щения</u>	Наименование Док	Причи-на Пере-мещ	Куда (КодРабМе-ста)	Код Комплек-ту ющей	Дата Переме-щения
----------------------------	------------------	-------------------	---------------------	---------------------	-------------------

### 1.8 Связь «Комплектующая – ДокументСдачиНаРемонт»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументСдачиНаРемонт», так как из нее исходит простая связь, следовательно, порожденной является сущность «Комплектующая». Атрибут КодКомплектующей сущности «Комплектующая» уже присутствует в сущности «ДокументСдачиНаРемонт», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 15

<u>Код-Комплекту-ющей</u>	Модель	Код РабМеста	Состояние-Комплект	Тип-Комплект	КодДок Поступления
КодДокСписания		СерийныйНо-мер	Цена	КоличествоРе-монтов	

Отношение 16

<u>Код-Док Сдачи НаРем</u>	Наименование-Док	Код Комплек-ту ющей	Причина СдачиНа-Рем	Код Поставщи-ка	Дата Сдачи	№Квитан-ции
----------------------------	------------------	---------------------	---------------------	-----------------	------------	-------------

### 1.9 Связь «ДокументСдачиНаРемонт – Поставщик»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументСдачиНаРемонт», так как из нее исходит простая связь, следовательно, порожденной является сущность «Поставщик». Атрибут КодПоставщика сущности «Поставщик» уже присутствует в сущности «ДокументСдачиНаРемонт», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 17

<u>Код- Док Сдачи НаРем</u>	Наименование- Док	Код Комплек- ту ющей	Причина СдачиНа- Рем	Код Поставщи- ка	Дата Сдачи	№Квитан- ции
---	----------------------	-------------------------------	----------------------------	------------------------	---------------	-----------------

Отношение 18

<u>КодПостав- щика</u>	Наименова- ние Поставщика	Адрес Офиса	Телефон Офиса	Адрес СервЦен- тра	Телефон СервЦен- тра
----------------------------	---------------------------------	----------------	------------------	--------------------------	----------------------------

### 1.10 Связь «Комплектующая – ДокументПолучСРемонта»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументПолучСРемонта», так как из нее исходит простая связь, следовательно, порожденной является сущность «Комплектующая». Атрибут КодКомплектующей сущности «Комплектующая» уже присутствует в сущности «ДокументПолучСРемонта», поэтому его повторного дублирования не происходит. В результате получили отношения

Отношение 19

<u>Код-Комплекту-ющей</u>	Модель	Код РабМеста	Состояние-Комплект	Тип-Комплект	КодДок Поступления
КодДокСписания	СерийныйНо-мер	Цена	КоличествоРе-монтов		

#### Отношение 20

<u>КодДок Получе-ния СРем</u>	Наиме-нование Док	Код По-став щика	Дата Получе-ния	Код Компле к тую-щей	Описа-ние Работ	Стои мост ь	№Наклад-ной Получе-ния
-------------------------------	-------------------	------------------	-----------------	----------------------	-----------------	-------------	------------------------

#### 1.11 Связь «ДокументПолучСРемонта – Поставщик»



Данная связь имеет тип «один-ко-многим». Исходной будет сущность «ДокументПолученияСРемонта», так как из нее исходит простая связь, следовательно, порожденной является сущность «Поставщик». Атрибут КодПоставщика сущности «Поставщик» уже присутствует в сущности «ДокументПолученияСРемонта», поэтому его повторного дублирования не происходит. В результате получили отношения

#### Отношение 21

<u>КодДок Получе-ния СРем</u>	Наиме-нование Док	Код По-став щика	Дата Получе-ния	Код Компле к тую-щей	Описа-ние Работ	Стои мост ь	№Наклад-ной Получе-ния
-------------------------------	-------------------	------------------	-----------------	----------------------	-----------------	-------------	------------------------

#### Отношение 22

<u>КодПостав-щика</u>	Наименова-ние Поставщика	Адрес Офиса	Телефон Офиса	Адрес СервЦен-тра	Телефон СервЦен-тра
-----------------------	--------------------------	-------------	---------------	-------------------	---------------------

## 2. Итоговый набор отношений

Проанализировав все отношения, полученные на этапе отображения концептуально-инфологической модели на реляционную, выделяем отношения, относящиеся к одной сущности.

Отношения 1, 3, 11 включают сущность «РабочееМесто», отношение 2 – сущность «Сотрудник», отношения 4, 5, 7, 13, 15, 19 – сущность «Комплектующая», отношение 6 – сущность «ДокументСписания», отношения 8, 9 – сущность «ДокументПоступления», отношения 10, 18, 22 – сущность «Поставщик», отношения 12, 14 – сущность «ДокументПеремещения», отношения 16, 17 – сущность «ДокументСдачиНаРемонт», отношения 20, 21 – сущность «ДокументПолученияСРемонта».

Составляем итоговый набор отношений, исключая дублирование отношений (исключение повторяющихся отношений).

Отношение 1 – «РабочееМесто»

<u>КодРабМеста</u>	ТипРабМеста	НаименованиеРабМеста	КодСотрудника
--------------------	-------------	----------------------	---------------

Отношение 2 – «Сотрудники»

<u>КодСотрудника</u>	Подразделение	Должность	Фамилия	Имя	Отчество	Рабочий Телефон
----------------------	---------------	-----------	---------	-----	----------	-----------------

Отношение 3 – «Комплектующие»

<u>КодКомплектующей</u>	Модель	КодРабМеста	ТипКомплект	СостояниеКомплект	КодДокПоступления
КодДокСписания	СерийныйНомер	Цена	КоличествоРемонтов		

Отношение 4 – «ДокументСписания»

<u>КодДокСписания</u>	НаименованиеДок	ПричинаСписания	ДатаСписания
-----------------------	-----------------	-----------------	--------------

Отношение 5 – «ДокументПоступления»

<u>КодДокПоступления</u>	КодПоставщика	ДокументПрихода	ДатаПоступления
--------------------------	---------------	-----------------	-----------------

Отношение 6 – «Поставщики»

<u>КодПоставщика</u>	НаименованиеПоставщика	АдресОфиса	ТелефонОфиса	АдресСервЦентра	ТелефонСервЦентра
----------------------	------------------------	------------	--------------	-----------------	-------------------

Отношение 7 – «ДокументПеремещения»

<u>КодДокПеремещения</u>	НаименованиеДок	ПричинаПеремещения	Кда(КодРабМеста)	КодКомплектующей	ДатаПеремещения
--------------------------	-----------------	--------------------	------------------	------------------	-----------------

Отношение 8 – «ДокументСдачиНаРемонт»

<u>КодДокСдачиНаРем</u>	НаименованиеДок	КодПоставщика	ДатаСдачи	КодКомплектующей	ПричинаСдачиНаРем	№Квитанции
-------------------------	-----------------	---------------	-----------	------------------	-------------------	------------

Отношение 9 – «ДокументПолученияСРемонта»

<u>КодДок</u>	Наимено	Код	Дата	Код	Описание	Стои	№Накладной
---------------	---------	-----	------	-----	----------	------	------------

<u>Получе нияСРем</u>	вание Док	Постав щика	Получения	Комплек тующей	Работ	мость	Получения
---------------------------	--------------	----------------	-----------	-------------------	-------	-------	-----------

## РАБОТА №8. Нормализация отношений (приведение совокупности отношений к 1НФ, 2НФ, 3НФ)

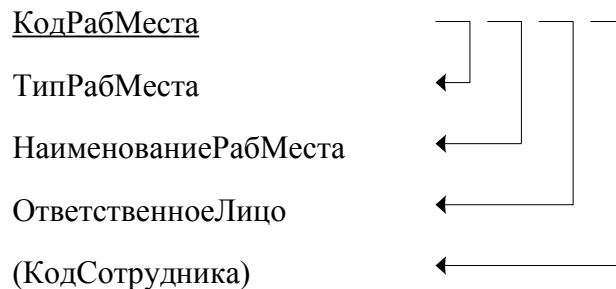
### 1 Приведение к первой нормальной форме

Все отношения, полученные на этапе отображения концептуальной инфологической модели на реляционную модель, соответствуют первой нормальной форме, так как все атрибуты простые.

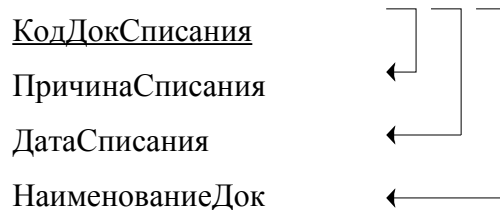
### 2 Приведение ко второй нормальной форме

Рассмотрим функциональные зависимости в отношениях

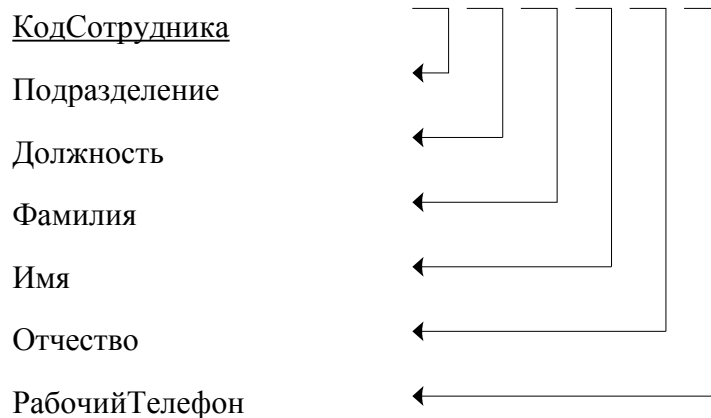
Функциональная зависимость атрибутов отношения «РабочееМесто»



Функциональная зависимость атрибутов отношения «ДокументСписания»

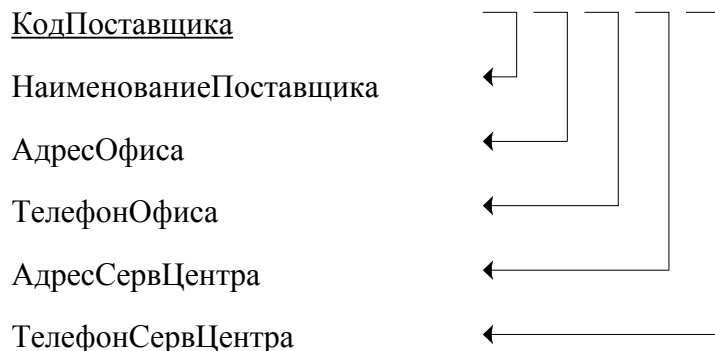


Функциональная зависимость атрибутов отношения «Сотрудник»

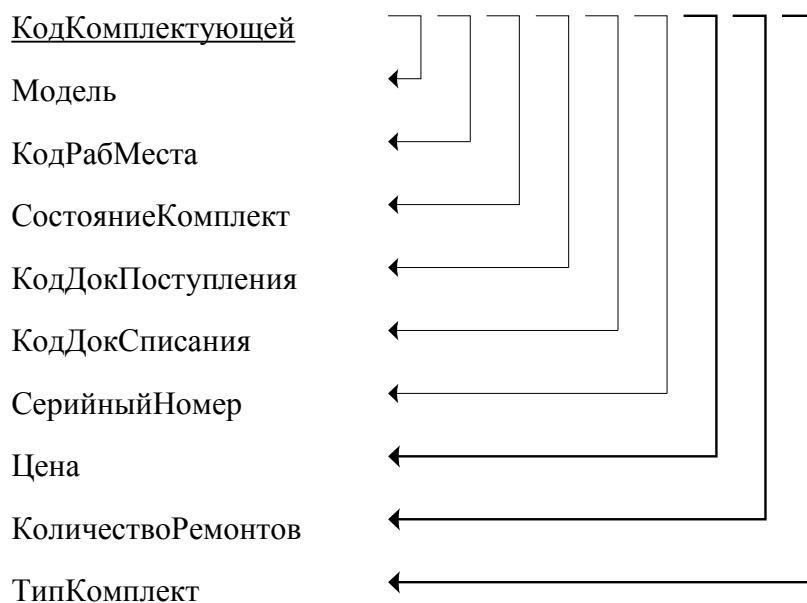




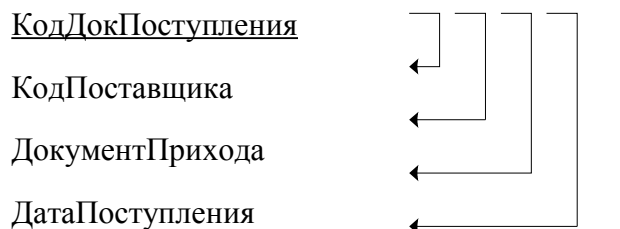
Функциональная зависимость атрибутов отношения «Поставщик»



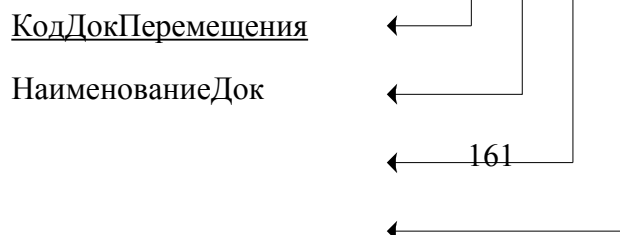
Функциональная зависимость атрибутов отношения «Комплектующая»



Функциональная зависимость атрибутов отношения «ДокументПоступления»

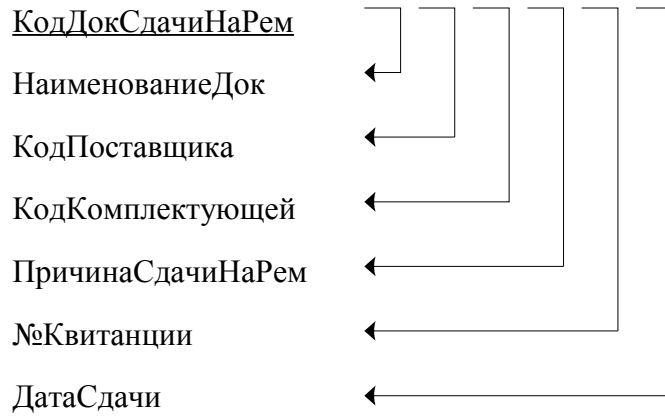


Функциональная зависимость атрибутов отношения «ДокументПеремещения»

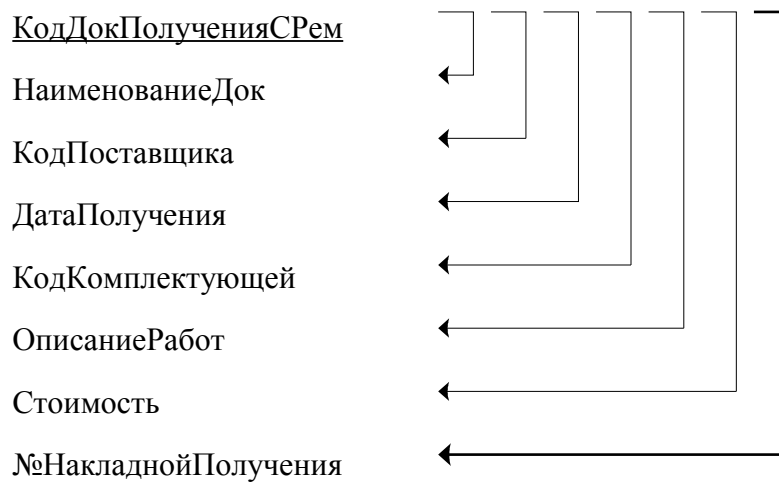


ПричинаПеремещения  
Куда(КодРабМеста)  
ДатаПеремещения  
КодКомплектующей

Функциональная зависимость атрибутов отношения «ДокументСдачиНаРемонт»



Функциональная зависимость атрибутов отношения «ДокументПолученияСРемонта»



Отношения находятся во второй нормальной форме, так как они находятся в первой нормальной форме, нет составных ключей и каждый атрибут, который, функционально полно зависит от ключа.

### 3 Приведение к третьей нормальной форме

Проанализировав отношения, можно сделать вывод, что они находятся в третьей нормальной форме, так как они находятся во второй нормальной форме и все атрибуты, которые не являются ключевыми, не имеют транзитивной зависимости от ключевых атрибутов.

## РАБОТА №9. Физическое проектирование

Целью физического проектирования является представление результатов логического проектирования в форме, пригодной для реализации в конкретной СУБД. При физическом проектировании происходит трансформация сущностей в таблицы, а атрибутов в поля; определяется тип данных, хранящихся в полях реляционных таблиц; осуществляется преобразование ограничений предметной области в ограничения СУБД и определяются индексы, позволяющие ускорить доступ к данным.

Таблица «РабочееМесто»

Название поля	Тип данных	Ограничение	Индексация
<u>КодРабМеста</u>	Счетчик	>0	Да
ТипРабМеста	Текстовый	-	Нет
ОтветственноеЛицо (КодСотрудника)	Числовой	>0	Нет
НаименованиеРабМеста	Текстовый	-	Да

Таблица «Сотрудник»

Название поля	Тип данных	Ограничение	Индексация
<u>КодСотрудника</u>	Счетчик	>0	Да
Подразделение	Текстовый	-	Нет
Должность	Текстовый	-	Нет
Фамилия	Текстовый	-	Нет
Имя	Текстовый	-	Нет
Отчество	Текстовый	-	Нет
РабочийТелефон	Текстовый	-	Нет

Таблица «Комплетующая»

Название поля	Тип данных	Ограничение	Индексация
<u>КодКомплектующей</u>	Счетчик	>0	Да
ТипКомплектующей	Текстовый	-	Нет
Модель	Текстовый	-	Нет

РабМесто	Числовой	>0	Да
Состояние	Текстовый	-	Нет
КодДокПоступления	Числовой	>0	Да
КодДокСписания	Числовой	>0	Нет
СерийныйНомер	Текстовый	-	Нет
КоличествоРемонтов	Числовой	-	Нет
Цена	Денежный	-	Нет

Таблица «Поставщик»

Название поля	Тип данных	Ограничение	Индексация
<u>КодПоставщика</u>	Счетчик	>0	Да
НаименованиеПоставщика	Текстовый	-	Нет
АдресОфиса	Текстовый	-	Нет
ТелефонОфиса	Текстовый	-	Нет
АдресСервЦентра	Текстовый	-	Нет
ТелефонСервЦентра	Текстовый	-	Нет

Таблица «ДокументСписания»

Название поля	Тип данных	Ограничение	Индексация
<u>КодДокСписания</u>	Счётчик	>0	Да
НаименованиеДок	Текстовый	-	Нет
ПричинаСписания	Текстовый	-	Нет
ДатаСписания	Дата/Время	<=Date()	Нет

Таблица «ДокументПоступления»

Название поля	Тип данных	Ограничение	Индексация
<u>КодДокПоступления</u>	Счётчик	>0	Да
КодПоставщика	Числовой	>0	Да
ДокументПрихода	Текстовый	-	Нет
ДатаПоступления	Дата/Время	<=Date()	Нет

Таблица «ДокументПеремещения»

Название поля	Тип данных	Ограничение	Индексация
<u>КодДокПеремещения</u>	Счетчик	>0	Да
НаименованиеДок	Текстовый	-	Да
ПричиныПеремещ	Текстовый	-	Нет
Куда(РабМесто)	Числовой	>0	Да
ДатаПеремещения	Дата/Время	<=Date()	Да
КодКомплектующей	Числовой	-	Да

Таблица «ДокументСдачиНаРемонт»

Название поля	Тип данных	Ограничение	Индексация
<u>КодДокСдачиНаРем</u>	Счетчик	>0	Да
НаименованиеДок	Текстовый	-	Да
КодПоставщика	Числовой	>0	Да
ДатаСдачи	Дата/Время	<=Date()	Нет
КодКомплектующей	Числовой	>0	Да
ПричинаСдачиНаРем	Текстовый	-	Нет
№Квитанции	Текстовый	-	Нет

Таблица «ДокументПолученияСРемонта»

Название поля	Тип данных	Ограничение	Индексация
<u>КодДокПолученияСРем</u>	Счетчик	>0	Да
НаименованиеДок	Текстовый	-	Нет
КодПоставщика	Числовой	>0	Да
ДатаПолучения	Дата/Время	<=Date()	Нет
КодКомплектующей	Числовой	>0	Нет
ОписаниеРабот	Текстовый	-	Нет
Стоимость	Денежный	-	Нет
№НакладнойПолучения	Текстовый	-	Нет

## **9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ПРЕПОДАВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Для выполнения лабораторных работ используется Microsoft Access, т.к. является наиболее распространённой персональной СУБД. Это объясняется, прежде всего, тем, что она входит в состав офисного пакета Microsoft Office, используемого практически на всех IBM-совместимых компьютерах с операционной системой Windows. СУБД Microsoft Access предоставляет пользователю широкие возможности ввода, обработки и представления данных. Эти средства не только удобны, но и высокопродуктивны, что обеспечивает высокую скорость разработки приложений. СУБД Microsoft Access имеет дружественный интерфейс и позволяет использовать широкий спектр инструментов: от самых простых до инструментов профессионального разработчика. Поэтому СУБД Microsoft Access используется при выполнении лабораторных работ по дисциплине «Базы данных» может быть использована в дальнейшей деятельности выпускников.

## 10. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

### Лабораторная работа № 1

#### Задания

1. Самостоятельно изучить контекстно-зависимую справку: показать справочную информацию о кнопке «Запросы», о кнопке «Вырезать».
2. Разобраться со справочной системой Access 2003: показать работу вкладок «Содержание», «Предметный указатель», «Поиск».
3. Разобраться с возможностями «Помощника»: способы вызова, параметры.
4. Создайте пользовательскую панель инструментов.
5. Создайте кнопки открытия таблиц «Товары» и «Заказы» на вашей панели инструментов. Переименуйте их.
6. Создайте таблицу путем ввода данных. В качестве примера наберите почтовые адреса 5 человек: Фамилия, Имя, адрес, город, индекс, страна. Таблицу назовите «Адреса».
7. Создайте с помощью Мастера таблицу «Список рассылки» и свяжите ее с таблицей «Клиенты», установив переключатель на: одной записи таблицы «Клиенты» будет соответствовать много записей таблицы «Список рассылки».

#### Контрольные вопросы

1. Назвать основные и дополнительные функции Access 2003.
2. Перечислить характеристики приложения Access 2003.
3. Перечислить и дать характеристику каждого элемента базы данных Access 2003.
4. Перечислить способы создания новой базы данных.
5. Перечислить способы создания новой таблицы базы данных.

### Лабораторная работа № 2

#### Задания

1. Показать результаты по лабораторной работе (таблицу «Задание»).
2. В таблице «Заказы» установить, чтобы значение поля «ДатаИсполнения» было меньше значения поля «ДатаНазначения» и выводилось сообщение об ошибке.
3. В таблице «Сотрудники» для поля «ДатаНайма» установить, чтобы дата выглядела следующим образом: Август 4, 1998 - 4:33
4. В таблице «Сотрудники» для поля «Домашний телефон» создайте маску ввода (например, для ввода номера телефона 567-77-22).

5. Скопируйте из базы данных Борея таблицы «Товары» и «Поставщики» и поместите их в свою базу данных под другими именами.
6. Удалите скопированные таблицы в присутствии преподавателя.

### **Контрольные вопросы**

1. Перечислите типы данных.
2. Опишите типы данных.
3. Опишите свойства таблиц.
4. Опишите свойства полей.

### **Лабораторная работа № 3**

#### **Задания**

1. Изменить ключ в таблице «Задания»: Код ответственного и Дата\_факт.
2. Добавить третье поле к индексу «Тип/Дата» - поле Рейтинг.
3. Закрыть таблицу «Задания» не сохраняя изменения.
4. В «Схеме данных» удалить связь между таблицами «Типы» и «Товары», затем восстановить эту связь обеспечивая целостность данных.
5. Экспортировать таблицу «Задания» в HTML.
6. Сделать поле подстановки для поля Код ответственного.
7. Удалить поле подстановки для поля Сотрудник таблицы «Задания».
8. Оформить таблицу «Задания», используя все возможности настройки режима таблицы.

#### **Контрольные вопросы:**

1. Что понимается под ссылочной целостностью данных.
2. Рассказать о режиме каскадного обновления, режиме каскадного удаления.
3. Сколько таблица может иметь индексов, сколько может быть составных индексов, сколько полей может входить в составной индекс.
4. Типы полей подстановки.
5. Как убрать поле подстановки.
6. Как скрыть одно из значений в поле подстановки.
7. Как устанавливается связь “многие-ко-многим”.

### **Лабораторная работа № 4**

#### **Задания**

1. В таблице «Клиенты» найдите все записи, содержащие бес в поле «Область».
2. В таблице «Клиенты» в поле «Область» замените Qvebec на Quebec.



3. В таблице «Клиенты» в поле «Страна» отберите с помощью фильтра по выделенному данные, в которых имеется фрагмент –ан-.
4. В таблице «Задания» с помощью расширенного фильтра выберите все мероприятия, запланированные на сегодня, но еще не выполненные. Сохраните как запрос.

### Контрольные вопросы.

1. Дайте характеристику фильтра по выделенному.
2. Дайте характеристику обычного фильтра.
3. Дайте характеристику расширенного фильтра.
4. Почему при использовании фильтра могут отбираться не те записи.
5. когда критерии отбора объединяются при помощи логического оператора «И».
6. Когда критерии поиска объединяются при помощи логического оператора «ИЛИ».

## Лабораторная работа № 5

### Задания

1. Создать запрос с параметром: по фамилии сотрудника, вывести информацию о заказах, которые во Францию, Италию и Германию доставляются почтой.
2. Сохранить запрос как «Заказы сотрудников».
3. Для поля СтранаПолучателя в запросе сделать псевдоним.

### Контрольные вопросы

1. Чем запрос отличается от фильтра.
2. Что такое запрос на выборку.
3. Объяснить результат выполнения запроса «Список рассылки по США».
4. Что называется псевдонимом.
5. Объясните работу запроса «Список по США».
6. Что такое запрос с параметром.
7. Как включить в запрос все поля таблицы

## Лабораторная работа № 6

### Задание

Создайте, просмотрите результат и сохраните все запросы на выборку, которые находятся в таблице 6.1 (для каждой строки таблицы создайте отдельный запрос)

Таблица 6.1

Таблица	Поле	Выражение	Возвращае-
---------	------	-----------	------------

			мые записи
Клиенты	Страна	Not “США” And Not “Канада” или Not (“США” Or “Канада”)	Фирмы, не находящиеся в США и Канаде
Клиенты	Должность	Like “[Н-Я]*”	Должности с названиями на Н-Я
Клиенты	Название	Like С* Or Like Т*	Фирмы с названиями С или Т
Клиенты	Индекс	>=90000	Фирмы с почтовыми индексами, большими или равными 90 000
Заказы	ДатаРазмещения	Year ([ДатаРазмещения])=1996 или Like “*. *.1996”	Заказы, полученные в 1996 году
Заказы	ДатаРазмещения	Like “*.01.1997”	Заказы, полученные в январе 1997
Заказы	ДатаРазмещения	Like “0?.01.1997”	Заказы, полученные с 1 по 9 января 1997
Заказы	ДатаРазмещения	Year ([ДатаРазмещения])=1997 And DatePart(“q”;[ДатаРазмещения])=1 или Between #01.01.1997# And #31.03.1997#	Заказы, полученные в первом квартале 1997
Заказы	ДатаРазмещения	Year ([ДатаРазмещения])=1997 And DatePart(“ww”;[ДатаРазмещения])=10	Заказы, полученные на десятой неделе 1997
Заказы	ДатаРазмещения	>=DateValue(“15.01.1997”)	Заказы, полученные 15.1.97 и позднее

## Лабораторная работа № 7

### Задания

1. Создайте запрос для отображения категорий товаров, закупленных некоторым клиентом.
2. Создайте запрос с правым внешним соединением и объясните результат выполнения запроса.

3. Объясните результат выполнения всех созданных в лабораторной работе запросов.
4. Создать запрос, отражающий информацию о заказах, которые назначаются и исполняются в один день.

#### **Контрольные вопросы:**

1. Назовите основные типы соединений.
2. Рассказать о внутреннем соединении.
3. Рассказать о левом внешнем соединении.
4. Рассказать о правом внешнем соединении.
5. Рассказать о рекурсивном соединении.
6. Рассказать о соединении по отношению.

### **Лабораторная работа № 8**

#### **Задания**

1. Объяснить работу всех созданных в работе запросов.
2. Создать перекрестный запрос, по строкам – фамилии сотрудников, по столбцам – названия стран получателей заказов. Запрос должен отражать информацию о том, сколько заказов каждый сотрудник исполняет в каждой стране.

#### **Контрольные вопросы**

1. Назовите основные статистические функции языка SQL.
2. Почему статистические функции языка SQL применяются с группировкой?
3. Какой запрос является перекрестным?
4. В чем отличие запросов на выборку от перекрестных запросов?
5. Как изменить заголовки столбцов в перекрестном запросе?

### **Лабораторная работа № 9**

#### **Задания**

1. Создать запрос на создание таблицы, в которой будут храниться данные о заказах, которые исполняются в 1997 году и за которые отвечают сотрудники с фамилией, начинающейся на буквы А, В, К.
2. Сохранить таблицу по имени «Заказы 1997»
3. Создать запрос на удаление: из таблицы «Заказы 1997» удалить все записи за осень 1997 года.
4. Создать запрос на добавление: в таблицу «Заказы 1997» добавить записи за декабрь 1996 года.

5. Создать запрос на обновление: в таблице «Заказы 1997» изменить фамилию сотрудника Воронова на Иванова.
6. Продемонстрировать каскадное удаление и обновление данных.

### **Контрольные вопросы**

1. Для чего используются запросы на создание таблицы, на удаление, на добавление, на обновление?
2. Поясните каскадное удаление.
3. Поясните каскадное обновление.

## **Лабораторная работа № 10**

### **Задания**

1. Поменяйте цвета фона заголовка, области данных, примечания формы.
2. Смените шрифт текста.
3. Выровняйте текст.
4. Измените ширину формы.
5. Поменяйте фоновый рисунок формы с использованием свойств фонового рисунка (масштабы рисунка, выравнивание рисунка, мозаичное заполнение).
6. Измените цвет текста.
7. Измените цвет и ширину границы элемента формы.
8. Измените стиль границы элемента управления формы.
9. Измените стиль оформления элементов управления.
10. Используйте дополнительные цвета.
11. Выведите на экран и просмотрите свойства раздела: Заголовок формы.
12. Выведите на экран и просмотрите свойства раздела: Область данных.
13. Выведите на экран и просмотрите свойства формы.
14. Для основной формы «Задания» сделайте, чтобы отображались записи в алфавитном порядке по фамилии вместо отображения их в порядке, определяемом значением поля «КодСотрудника».

### **Контрольные вопросы**

1. Перечислить и описать типы форм.
2. Перечислить и описать элементы окна конструктора форм.
3. Перечислить и описать свойства фоновых рисунков

## **Лабораторная работа № 11**

### **Задания**

1. На форме создать группу переключателей, соответствующих типу выполняемого мероприятия для каждого сотрудника. Воспользоваться данными из таблицы «Задания».

2. На форме создать вкладки: на первой из них должна быть отражена информация о заказах (КодЗаказа, ДатаИсполнения, СтранаПолучателя), а на второй вкладке – информация о товарах в этом заказе (Категория, Марка, Цена Количество Скидка и Общая стоимость каждого товара с учетом скидки).

### **Контрольные вопросы**

1. Перечислить и описать типы элементов управления.
2. Какие способы используются для создания группы элементов
3. Для чего предназначены списки и поля со списками

## **Лабораторная работа № 12**

### **Задания**

1. Создать запрос с параметром, который позволяет получить КодЗаказа, Название клиента, Фамилию сотрудника, Страну получателя, Марку, Цену и Количество товаров в заказе, а также стоимость товара со скидкой. Должна выдаваться информация о заказах, ДатаИсполнения которых находится в интервале между двумя датами, задающимися пользователем с клавиатуры в формате ДД.ММ.ГГГГ. Сохранить запрос под именем «Заказы по клиентам».
2. На основе запроса «Заказы по клиентам» создать отчет, который содержит поля: КодЗаказа, Название клиента, Марку, Цена, Количество, Цена со скидкой.
3. Осуществите группировку по Названию клиента, затем укажите дополнительный уровень группировки ДатаИсполнения по месяцам.
4. Указать в качестве итоговой информации отчета Sum для Цены, Количество, Цены со скидкой.
5. В поле заголовка отобразите значения вводимых параметров, т.е. Начальную и Конечную дату исполнения заказов.
6. Оформление отчета выберите самостоятельно.
7. Сохраните и просмотрите полученный отчет.
8. В отчете измените размер шрифта, расстояние между строками и размеры полей, чтобы вся информация была видна в отчете.
9. Создайте в отчете вычисляемое поле «Сумма», в котором вычисляется общая стоимость заказов и общая стоимость заказов со скидкой по клиентам и месяцам.
- 10.Отсортируйте Марку товаров в алфавитном порядке.
11. Сделайте снимок отчета и экспортируйте отчет HTML.
- 12.Создайте и сохраните свой шаблон отчета.

### **Контрольные вопросы**

1. Что такое и для чего предназначен автоотчет?

2. Какие существуют способы создания отчетов?
3. В какие форматы можно преобразовывать отчеты?
4. Что такое снимки отчетов?
5. Какие существуют режимы работы с отчетами?
6. Из каких разделов состоит отчет?
7. Как осуществляется группировка данных в отчетах?
8. Каким образом осуществляется сортировка данных в отчете?
9. С какой целью в отчетах применяют линии и прямоугольники:
10. Что такое шаблон отчетов?
11. Как изменить параметры страницы отчета?

## **11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ МЕЖСЕССИОННОГО КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ**

1. Межсессионная аттестация студентов проводится дважды в семестр на 7 и 13 неделях 5-го семестра.
2. Аттестационная оценка складывается из оценок, полученных аттестационных занятиях по лабораторным работам и оценки за контрольную работу.
3. Организация аттестации студентов, проводится в соответствии с положением АмГУ о курсовых экзаменах и зачетах.

## 12. ФОНД ТЕСТОВЫХ И КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЗНАНИЙ ПО ДИСЦИПЛИНЕ

Утверждено на заседании

Кафедры ИУС

« \_\_\_\_ » \_\_\_\_\_ 2007г.

Зав. каф. \_\_\_\_\_ Бушманов А.В.

Инструкция: выберите из четырех предложенных вариантов правильный на ваш взгляд, и обведите его. Исправления в тесте НЕ ДОПУСКАЮТСЯ! Перед тем, как выбрать ответ, хорошо подумайте!

### Вариант 1

1. Система информационных, математических, программных, языковых, организационных, и технических средств, предназначенных для централизованного накопления и коллективного многоаспектного использования данных для получения необходимой информации - это

- |                |                |
|----------------|----------------|
| 1) данные      | 2) информация  |
| 3) база данных | 4) банк данных |

2. Какие сведения не хранятся в словаре данных

- |                              |  |
|------------------------------|--|
| 1) о данных, хранимых в базе | 2) о возможных значениях и форматах представления данных |
| 3) о запросах пользователей  | 4) об источниках возникновения данных                    |

3. Укажите функцию, которую банк данных не выполняет в составе автоматизированных систем

- |  |  |
|--|--|
| 1) специальная обеспечивающая подсистема                   | 2) механизм реализации решений                       |
| 3) поддержание информационной модели в требуемом состоянии | 4) обеспечение информационных запросов пользователей |

4. Укажите признак, по которому не выделяют пользователей банка данных



- 1) постоянство общения с банком данных      2) уровень компетенции  
3) скорость обработки запросов                      4) форма представления запросов

5. Какой тип связи не поддерживается в иерархической модели данных

- 1) один к одному    2) один ко многим  
3) многие ко многим    4) простая однонаправленная

6. Какой из русских эквивалентов наиболее точно отражает смысл понятия "реляционный" применительно к теории баз данных?

- 1) относительный    2) табличный  
3) родственный    4) взаимосвязанный

7. Элемент декартового произведения доменов - это

- 1) кортеж    2) степень отношения  
3) отношение    4) мощность отношения

8. В базе данных имеется таблица Отделы и таблица Сотрудники. При условии, что в каждый отдел включено несколько сотрудников, но каждый сотрудник может быть включен только в один отдел, какой тип связи между таблицами нужно установить?

- 1) один к одному    2) один ко многим от таблицы Отделы к таблице Сотрудники  
3) многие ко многим    4) один ко многим от таблицы Сотрудники к таблице Отделы

9. Укажите уровень архитектуры системы баз данных, о котором справедливо утверждение – этот уровень описывает, какие данные хранятся в базе данных

- 1) внешний    2) концептуальный  
3) информационный    4) внутренний

10. Укажите этап, задача которого состоит в следующем: разработать схему концептуальной модели и схемы внешних моделей данных о ПО, пользуясь только теми типами моделей данных, которые поддерживаются конкретной СУБД

- 1) логическое проектирование                                      2) инфологическое проектирование  
3) выбор СУБД    4) физическое проектирование

11. Средство, с помощью которого представляются отношения между объектами предметной области - это

- 1) сущность
- 2) предметная область
- 3) связь
- 4) атрибут

12. Отношение, в котором определен этот ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом. Укажите ключ, о котором идет речь в высказывании

- 1) вторичный ключ
- 2) возможный ключ
- 3) первичный ключ
- 4) внешний ключ

13. Укажите, какое действие не принадлежит этапу логического проектирования

- 1) установление дополнительной логической связи
- 2) отображение исходной концептуальной инфологической модели на конкретную логическую модель
- 3) преобразование модели с учетом ограничений конкретной СУБД
- 4) выбор рациональной структуры хранения данных и методов доступа к ним

14. В какой нормальной форме находится отношение?

<u>Код сотрудника</u>	<u>ФИО</u>	<u>Должность</u>	<u>Дата рождения</u>
12	Иванов О.П.	конструктор	10.12.1968
15	Орлова И.М.	бухгалтер	12.01.1972
18	Петров И.Р.	инженер	02.05.1978
25	Сидоров Е.В.	ассистент	31.06.1987

- 1) 1 НФ
- 2) 2 НФ
- 3) 3 НФ
- 4) НФ Бойса-Кодда

15. Если в каждый момент времени каждому значению атрибута А соответствует не более одного значения атрибута В, то зависимость между атрибутами А и В называется

- 1) полной функциональной
- 2) транзитивной
- 3) функциональной
- 4) многозначной

16. Укажите действия, которые не выполняются в процессе эксплуатации базы данных

- 1) обеспечение доступа к данным                      2) реорганизация базы данных  
 3) создание структуры базы данных                      4) контроль за состоянием базы данных

17. Если отношения R и S определены в виде

R =

A1	A2	A3	A4
5	4	0	8
7	3	-5	1

S =

A5	A6	A7
3	4	1
4	5	8
7	6	3
8	3	6

то результат естественного соединения равен

1)

A1	A2	A3	A4	A5	A6	A7
5	4	0	8	3	4	1
7	3	-5	1	8	3	6

2)

A1	A2	A3	A4	A5	A7
5	4	0	8	3	1
7	3	-5	1	8	6

3)

A1	A3	A4	A5	A7
5	0	8	3	1
7	-5	1	8	6

4)

A1	A2	A3	A4	A5	A6	A7
5	4	0	8	3	4	1
7	3	-5	1	3	4	1
5	4	0	8	8	3	6
7	3	-5	1	8	3	6

18. Если отношения R и S определены в виде

R =

A1	A2	A3
7	10	4
3	6	5
5	7	1

S =

A1	A2	A3
5	7	1
8	3	6

то разность R – S указанных отношений равно

1)

A1	A2	A3	A1	A2	A3
7	10	4	5	7	1
3	6	5	8	3	6
5	7	1			

2)

A1	A2	A3
7	10	4
3	6	5
5	7	1
5	7	1
8	3	6

3)

A1	A2	A3
5	7	1

4)

A1	A2	A3
7	10	4
3	6	5

19. Этот элемент базы данных представляет собой специальный тип веб-страниц, который обеспечивает веб-интерфейс к данным в базе данных

- 1) формы
- 2) макросы
- 3) отчеты
- 4) страницы доступа

### Вариант 2

1. Совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области - это

- 1) данные
- 2) информация
- 3) база данных
- 4) предметная область

2. Укажите функции, которые не являются функциями администратора базы данных

- 1) взаимодействие с пользователями
- 2) определение процедур резервного копирования и восстановления
- 3) определение правил безопасности и целостности
- 4) принимать решение, какие данные нужно вносить в базу данных

3. Укажите элемент, который не является компонентом банка данных

- 1) словарь данных
- 2) база данных
- 3) внешние пользователи
- 4) система управления базой данных

4. Какие требования не предъявляются к банку данных со стороны внешних пользователей?

- 1) удовлетворять актуальным информационным потребностям
- 2) обеспечивать доступ к данным всем пользователям
- 3) обеспечивать заданный уровень достоверности хранимой информации
- 4) обеспечивать выдачу информации пользователям в различной форме

5. Укажите модель данных, которая обладает корневой записью

- 1) сетевая
- 2) иерархическая
- 3) объектно-ориентированная
- 4) реляционная

6. Укажите правила ограничения целостности, которыми обладает реляционная модель данных

- 1) целостность связей
- 2) целостность ссылок
- 3) целостность кортежей
- 4) целостность атрибутов

7. Множество значений атрибута - это

- 1) кортеж
- 2) отношение

3) домен

4) мощность отношения

8. В базе данных имеется таблица Заказы и таблица Товары. При условии, что в каждый заказ включено несколько товаров, и каждый товар может быть включен в несколько заказов, какой тип связи между таблицами нужно установить?

1) один к одному

2) один ко многим от таблицы Заказы к таблице Товары

3) многие ко многим

4) один ко многим от таблицы Товары к таблице Заказы

9. Укажите принцип, который не имеет отношение к архитектуре системы баз данных: для базы данных имеется

1) одна внутренняя модель

2) столько внешних моделей, сколько существует различных представлений пользователей

3) одна внешняя модель

4) одна концептуальная модель

10. Укажите этап, задача которого состоит в следующем: получение смысловых моделей, отражающих информационное содержание конкретной предметной области

1) логическое проектирование

2) инфологическое проектирование

3) выбор СУБД

4) физическое проектирование

11. Укажите, какая из диаграмм является ER-диаграммой

1) диаграмма потоков данных

2) диаграмма производительности базы данных

3) диаграмма бинарных связей

4) диаграмма "сущность - связь"

12. Укажите свойство, которым должен обладать первичный ключ

1) однозначная идентификация

2) значения могут быть не уникальными

3) максимальное число атрибутов

4) может быть неопределенным

13. Укажите, что является целью логического этапа проектирования базы данных

1) построение СУБД ориентированной, эффективной модели базы данных

2) выбор рациональной структуры хранения данных и методов доступа к ним

- 3) построение концептуальной инфологической модели базы данных 4) построение внешней модели базы данных

14. В какой нормальной форме находится отношение?

<u>ФИО</u>	Должность	Оклад	Стаж	Надбавка за стаж	Кафедра
Иванов О.П.	доцент	5000	12	100	ИУС
Петров И.Р.	ст. преп	2000	3	20	ОМИ

- 1) 1 НФ 2) 2 НФ  
3) 3 НФ 4) НФ Бойса-Кодда

15. Пусть отношение  $R = (A, B, C)$ . Если атрибут  $C$  зависит от атрибута  $B$ , а атрибут  $B$  зависит от атрибута  $A$ , то зависимость атрибута  $C$  от атрибута  $A$  называется

- 1) многозначной 2) зависимостью соединения  
3) полной функциональной 4) транзитивной

16. Укажите исходные данные для этапа физического проектирования

- 1) инфологическая модель 2) запросы пользователей  
3) логическая модель 4) методы доступа

17. Если отношения  $R$  и  $S$  определены в виде

R =

A1	A2	A3	A4
5	4	0	8
2	7	1	2
3	8	1	9
9	8	4	8
7	3	-5	1

S =

A5	A6	A7
3	4	1
8	3	6

то результат эквисоединения равен

1)

A1	A2	A3	A4	A5	A6	A7
5	4	0	8	3	4	1
7	3	-5	1	8	3	6

2)

A1	A2	A3	A4	A5	A7
5	4	0	8	3	1
7	3	-5	1	8	6

3)

A1	A3	A4	A5	A7
5	0	8	3	1
7	-5	1	8	6

4)

A1	A2	A3	A4	A5	A6	A7
5	4	0	8	3	4	1
7	3	-5	1	3	4	1
5	4	0	8	8	3	6
7	3	-5	1	8	3	6

18. Если отношение R определено в виде

R =

A1	A2	A3
8	5	2
3	6	5
5	7	1
10	9	5

то селекция  $\sigma_{A1 \leq A2}(R)$  равна

1)

A1	A2	A3
8	5	2
3	6	5
5	7	1

2)

A1	A2
8	5
3	6
5	7

3)

A1	A2
3	6
5	7

4)

A1	A2	A3
3	6	5
5	7	1

19. Этот элемент базы данных позволяет автоматизировать повторяющиеся операции

1) формы

2) макросы

3) отчеты

4) страницы доступа

### 13. КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЧЕТУ

1. Информация и данные. Базы и банки данных. Предметная область банка данных.
2. Пользователи банков данных.
3. Роль и место банков данных в автоматизированных системах.
4. Основные компоненты банков данных.
5. Администратор базы данных.
6. Архитектура банка данных. Трехуровневая архитектура банка данных.
7. Последовательность действий СУБД при обработке запросов.
8. Этапы проектирования базы данных.
9. Инфологическое проектирование БД. Модель "Сущность - Связь".
10. Моделирование локальных представлений. Формулирование сущностей.
11. Выбор идентифицирующего атрибута для каждой сущности. Назначение сущностям описательных атрибутов.
12. Спецификация связей.
13. Объединение моделей локальных представлений. Идентичность, агрегация, обобщение, устранение выявленных противоречий.
14. Иерархическая модель системы.
15. Сетевая модель системы.
16. Реляционная модель данных.
17. Реляционная алгебра и реляционное исчисление.
18. Логическое проектирование.
19. Отображение инфологической модели на реляционную модель.
20. Нормализация отношений.
21. Физическое проектирование.
22. Жизненный цикл БД.



#### **14. КАРТА КАДРОВОЙ ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ**

Лектор – к.т.н., доцент Чепак Лариса Владимировна.

Руководитель лабораторных работ – к.т.н., доцент Чепак Лариса Владимировна.

Руководитель практических работ – к.т.н., доцент Чепак Лариса Владимировна