

---

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное образовательное учреждение высшего  
профессионального образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РА-  
ДИОЭЛЕКТРОНИКИ» (ТУСУР)

УТВЕРЖДАЮ

Заведующий кафедрой ЭМИС

\_\_\_\_\_ И. Г. Боровской

«\_\_» \_\_\_\_\_ 2012 г.

**С.И. Колесникова**

**УПРАВЛЕНИЕ ДАННЫМИ**

*Методические указания по выполнению практических работ  
и курсовому проектированию*

2012

Методические указания к практическим работам по курсу «Базы данных» содержат только те понятия и определения, которые необходимы для их успешного выполнения. Основное внимание уделяется реляционной модели данных и практическому курсовому проектированию баз данных. Приводится пример проекта базы данных.

Часть практических работ опираются на образцы (таблицы), поставляемые со стандартным приложением (базы данных "Борей", "Заказы"). К пособию прилагаются и другие готовые базы данных, что удобно для практического использования данных методических указаний.

## СОДЕРЖАНИЕ

I. ПРАКТИЧЕСКАЯ РАБОТА №1. Создание и первичная обработка БД.....	4
II. ПРАКТИЧЕСКАЯ РАБОТА №2. Создание запросов, простых и специализированных: на выборку, на создание таблицы, на добавление записей, на обновление и удаление записей. Реляционные и перекрестные запросы.....	5
III. ПРАКТИЧЕСКАЯ РАБОТА №3. Нормализация отношений. Установление связей между таблицами .....	15
IV. ПРАКТИЧЕСКАЯ РАБОТА №4. СОЗДАНИЕ ФОРМ. ВЫЧИСЛЯЕМЫЕ ЭЛЕМЕНТЫ УПРАВЛЕНИЯ В ФОРМАХ. ФИЛЬТРЫ.....	20
V. ПРАКТИЧЕСКАЯ РАБОТА №5. СОЗДАНИЕ ПРОСТЫХ МАКРОСОВ, КНОПОЧНЫХ ФОРМ, МОДУЛЕЙ, ОТЧЕТОВ .....	24
VI. ПРАКТИЧЕСКАЯ РАБОТА №6. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ .....	33
VII. ПРАКТИЧЕСКАЯ РАБОТА №7. SQL-запросы.....	34
VIII. ПРАКТИЧЕСКАЯ РАБОТА №8. Создание Web страницы на основе базы данных	36
IX. Методические рекомендации по написанию курсовой работы.....	37

## I. ПРАКТИЧЕСКАЯ РАБОТА №1. Создание и первичная обработка БД

**Цель работы:** получение навыков работы по созданию базы данных (БД), созданию и редактированию таблиц.

**Темы для предварительного изучения.** Назначение таблиц. Порядок создания БД и таблиц . Разделы конструктора таблиц.

### Задание 1.1. Создать БД "Фирма Лакомка":

При запуске Access появляется диалоговое окно, в котором можно:

- выбрать БД;
- создать новую.

Для создания БД реализуйте один из способов:

- Выбрать пункт **Базы данных (Запуск мастера)**: автоматическое создание типовой БД, в данном случае этот способ игнорируйте.
- Выбрать пункт **Новая база данных**
- Закрыв диалоговое окно (кн. **Отмена**), можно создать БД (**Файл –Создать БД**)

### Задание 1.2. Создать таблицу < Категории товаров > с макетом (рис.1):

	Имя поля	Тип данных	Описание
?	КодТипа	Счетчик	Номер, автоматически присваиваемый новому типу.
	Категория	Текстовый	Наименование категории продуктов.
	Описание	Поле МЕМО	
	Изображение	Поле объекта OLE	Рисунок, представляющий категорию продуктов.

Рис.1 Макет таблиц **Категории товаров**

- Вкладка <Таблицы>, кн.<Создать>, <Конструктор>:
- После заполнения по образцу сохраните таблицу.
- Выполните команду <Вид–Режим таблицы>
- Заполните поля, т.е.создайте несколько записей (5-6)

Пользуясь предыдущим алгоритмом, создайте таблицу **Ассортимент** с макетом

	Имя поля	Тип данных	Описание
?	КодТовара	Счетчик	Уникальное число, автоматически присваиваемое новой записи.
	Марка	Текстовый	
	КодТипа	Числовой	Совпадает со значением поля "КодТипа" в таблице "Типы".
	ЕдиницаИзмерения	Текстовый	(например, в ящиках по 24 бутылки, 1-литровая бутылка и т.п.).
	Цена	Денежный	
	МинимальныйЗапас	Числовой	Минимально допустимый складской запас.
▶	ПоставкиПрекращены	Логический	"Да" означает, что товар больше не поставляется.

(рис.2):

Рис.2. Макет таблиц **Ассортимент**

**Внимание!!!** Обратит внимание при заполнении таблицы **Категории товаров** на содержимое <Код типа>; оно должно совпадать со значением поля <Код типа> в таблице **Ассортимент**. Например, содержимое таблиц может быть следующим (рис.3):

<i>Категории товаров</i>			
Код типа	Категория	Описание	Изображение
1	Фрукты консервированные	Упаковки: тетрапаки, банки	
2	Колбасные изделия	В т.ч. сосиски, запеченное мясо и т.д.	
3	Консервированные молочные продукты	Сгущ.молоко, концентрир. молоко и т.д.	
4	.....	.....	

<i>Ассортимент</i>						
Код товара	Марка	Код типа	Единица измерения	Минимальный запас	Цена	Поставки Прекращены
1	Компот из яблок	1	24 банок по 250 мл	5	69,75р.	Да
2	Компот из персиков	1	48 банок по 450 мл	10	78,52р.	Нет
3	Сгущ.молоко	3	20 банок по 1 кг	35	175,50 р.	Да
4	концентр. молоко	3	40 банок по 250 мл	48	18,25р.	Нет
5	.....	.....	.....		.....	.....

Рис.3. Примеры содержимого таблиц **Категории товаров** и **Ассортимент**

## II. ПРАКТИЧЕСКАЯ РАБОТА №2. Создание запросов, простых и специализированных: на выборку, на создание таблицы, на добавление записей, на обновление и удаление записей. Реляционные и перекрестные запросы

**Цель работы:** получение навыков работы по созданию простых запросов на выборку; запросов с вычисляемыми полями, использованию фильтров; получение навыков работы по созданию специализированных запросов на изменение; запросов с вычисляемыми полями, запросов с параметрами и реляционных запросов. Получение навыков работы по созданию запросов с групповыми операциями.

**Темы для предварительного изучения.** Назначение запросов. Порядок создания запросов. Разделы бланка запросов. Сортировка и группировка. Фильтр, расширенный фильтр. Назначение специализированных запросов. Порядок создания специализированных запросов. Разделы бланка специализированных запросов и команды меню. Назначение групповых операций. Порядок создания запросов с групповыми операциями. Разделы бланка запросов с групповыми операциями и встроенные функции. Конструкция и назначение вычисляемых полей. Назначение запросов с параметрами. Порядок создания: а) запросов с параметрами; б) реляционных запросов.

### **ПРОСТЫЕ ЗАПРОСЫ НА ВЫБОРКУ. ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЯЕМЫХ ПОЛЕЙ. ФИЛЬТРАЦИЯ.**

При выполнении **запроса на выборку** результаты отображаются в **форме динамического набора**, который содержит живые данные, а не статическую копию данных исходных таблиц. Поэтому модификация данных в записях динамического набора приводит к модификации в исходных таблицах. Запрос можно сохранить, но при этом полученные при его выполнении данные не сохраняются.

**Задание 2. 1.** В созданную БД **Фирма "Лакомка"** импортировать все таблицы из БД **Trade**. Далее выполнить последовательность команд:

выбрать вкладку <Таблицы>;

выбрать в контекстном меню команду <Импорт>;

установить путь к БД **Trade**;

выбрать все таблицы, ОК.

**Задание 2. 2.** Выполнить запросы, пользуясь предложенным алгоритмом:

1. Из таблицы **Товары** выбрать товары с ценой выше средней, используя функцию `>(SELECT AVG([Цена]) From Товары)` при этом расположив товары по убыванию цены (в поле <Сортировка> выбрать вид сортировки <По убыванию>).
2. Из таблицы **Товары** выбрать 10 самых дорогих товаров, расположив товары по убыванию цены (воспользоваться кнопкой <Набор значений>, установив сортировку).
3. Из таблицы **Товары** вывести список имеющихся товаров по убыванию цены, при этом товары, поставки которых прекращены, не должны быть выведены в динами-

ческий набор (учесть содержимое поля <Поставки прекращены>, в поле запроса Условие ввести "Нет")

Порядок создания запроса **вручную**:

- В окне БД открыть вкладку <Запросы>.
- Кн. <Создать>. Появится диалоговое окно <Новый запрос>.
- В этом окне выбрать <Конструктор>, кн. **ОК**. Откроется окно <Добавление таблицы>.
- Выбрать таблицу (таблицы) или другой запрос.
- Щелкнуть на кн. <Добавить>.
- Если Вы закрыли диалоговое окно <Добавление таблицы>, а затем решили добавить еще одну таблицу или запрос, то следует выполнить одно из действий:  
1) кн. <Добавить таблицу>, или 2) команда <Запрос– Добавить таблицу>.
- После определения источника данных для запроса в столбцы строки Поле бланка запроса добавляются необходимые поля. Добавить поля можно разными способами:
  - 1) для добавления одного поля переместить его мышью из окна таблицы в строку <Поле>; или дважды щелкнуть на поле в окне таблицы;
  - 2) чтобы добавить несколько полей, выбрать поля, используя кл.<Shift> или <Ctrl>, переместить их мышью;
  - 3) чтобы добавить все поля, переместить мышью звездочку, расположенную вверху окна таблицы; или дважды щелкните на имени таблицы, выделенные при этом поля переместите в бланк запроса.

**Задание 2. 3.      Импортировать таблицу Auto из БД Auto.mdb и выполнить следующие запросы на выборку.**

1. Машина с номером А384ЛЛ, как сообщили, находилась на частной стоянке в течение последних 2 недель. Кто является владельцем и где живет?

2. У Сидорова Юрия Леонидовича украли машину. В рапорте номер машины стерт. Найдите регистрационный номер этой машины.

3. Свидетелями этого случая могли быть люди из машин с номерами А567АЕ и А221ВФ. Найдите имена и адреса владельцев этих машин.

4. Машина после несчастного случая не остановилась, но свидетель утверждает, что это была машина с номером А10... или А19..., и что она была красная. Попробуйте найти машину и владельца.

5. Столкнулись три машины. Люди из двух машин лежат в бессознательном состоянии в больнице. Одна из этих машин - белая, а номер начинается на А23 (остальное - неразборчиво). Другая машина - красная, номер начинается на А40. Найдите, пожалуйста, имена и адреса владельцев этих машин.

6. Милиция хотела бы связаться со всеми владельцами машин "BMW", зарегистрированных после 1991 года. Подготовить соответствующий список.

7. Свидетель несчастного случая, в котором машина не остановилась после удара, сказал, что это была черная или синяя машина с номером, оканчивающемся на букву Т. Подготовить список подозреваемых.

8. Свидетельница несчастного случая помнит, что машина была белого или серого цвета. Она слышала, что пассажир обращался к водителю, называя его "Петрович". Подготовить список подозреваемых.

9. Милиция хочет найти, что общего имеют владелец красной Газели с владельцем вишневого автомобиля с номером, начинающимся с А40. Найдите общую для них информацию.

10. На обочине стоит заброшенная "Нива". Она красного цвета, а в "бардачке" была найдена помятая визитная карточка с едва просматривающейся надписью "Ал....ан...". Номерные знаки отсутствуют. Помогите установить владельца.

**Фильтрация.** Для отбора записей во время просмотра или редактирования можно воспользоваться фильтром. Чтобы создать фильтр, необходимо выбрать пункт меню <Записи–Фильтр–Расширенный фильтр>.

Окно <Фильтра>разделено на две области. В верхней области находится окно со списком полей открытой таблицы. В нижней области находится бланк фильтра. Каждый столбец бланка фильтра описывает одно поле таблицы, для которого записываются условия отбора записей или порядок сортировки.

Строка <Поле> предназначена для выбора имени поля таблицы из раскрывающегося списка. В строке <Сортировка> можно указать порядок сортировки значений данного поля. В строках <Условия отбора>, <или> задаются условия отбора записей в виде логических выражений, например:

(>30), (>= «Москва»), (=10).

При задании условия отбора записей можно использовать:

1. операторы сравнения: <, >, <=, >=, =, <>;
2. логические функции: And, Or, Not, Xor;



3. арифметические операции: + (сложение), – (вычитание), \* (умножение), / (деление), \ (деление нацело), mod (остаток от деления), ^ (возведение в степень);
4. встроенные функции, например, Date() (текущая дата), Cos() (косинус угла) и др.;
5. операторы слияния строковых значений – &, например, "Мос" & "ква";
6. диапазон значений – Between, например, Between 10 and 20;
7. Is Null – определяет, является ли значение пустым, Is Not Null – не пустым;
8. список значений – In, например, In("Москва", "Саратов", "Кострома");
9. сравнение с шаблоном – Like, например, Like "?a\*", означает слова, в которых вторая буква "а". В шаблоне можно использовать символ "?", который замещает только один знак, и символ "\*", который замещает любое число знаков.

В выражениях можно использовать имена других объектов базы данных, например, [КодТипа][Товары] – означает поле <КодТипа> в таблице **Товары**.

Вводить выражения можно не только вручную, но и с помощью <Построителя выражений>, который вызывается нажатием кнопки <Построить>, расположенной на панели инструментов <Конструктор запросов>.

Условия, записанные в одной строке, но в разных столбцах, связываются логической функцией «и», т.е. должны выполняться одновременно. Для задания условий, связанных логической функцией «или», необходимо каждое условие записать на отдельной строке.

После заполнения бланка фильтра необходимо выполнить команду меню <Фильтр–Применить фильтр>. Закрытие окна <Фильтра> приводит к его уничтожению. Чтобы отменить действие фильтра, необходимо выбрать в меню пункт <Записи–Удалить фильтр>.

#### **Задание 2. 4. В таблице Заказано выполните поиск заказов:**

- а). с кодом 10273;
- б). клиентов, коды которых заканчиваются на «09»;
- в). клиентов, коды которых начинаются на «С»;
- г). в коде заказов присутствует значение «30».

**Задание 2. 5.** В таблице **Товары** установите фильтр, позволяющий выделить строки, содержащие сведения о товарах, с кодами типов 1 и 3 и с ценой не менее 45000. Затем отмените фильтрацию.

## СПЕЦИАЛИЗИРОВАННЫЕ ЗАПРОСЫ НА ИЗМЕНЕНИЕ.

Запросы на изменение выполняют определенные действия над извлеченными данными. Разработка начинается с создания запроса на выборку для отбора необходимых данных. Затем происходит выбор соответствующего типа запроса.

В Access можно создавать 4 типа запросов на изменение: *на создание таблицы по результатам запроса, на добавление, на обновление, на удаление.*

### 1. Создание таблиц:

- создать запрос на выборку;
- команда <Запрос—Создание таблицы>;
- ввести имя создаваемой таблицы или выбрать из списка для записи существующей таблицы еще раз.

**Задание 2. 6.** Выполнить запрос на создание таблицы 10 самых дорогих товаров

### 2. Запрос на добавление используется для добавления данных из одной таблицы в другую. При этом типы полей одной таблицы должны соответствовать типам полей другой таблицы.

- Создать запрос на выборку, извлекающий записи, которые нужно добавить в другую таблицу. В бланк запроса добавляется таблица, *из которой* будут выбраны определенные записи.
- Команда <Запрос—Добавление>. Появится диалоговое окно.
- Ввести имя таблицы, *в которую необходимо добавить данные*, или выбрать таблицу из списка.
- Выполнить запрос. Появится диалоговое окно, указывающее, сколько записей будет добавлено в таблицу.

### **Задание 2. 7.**

- 1) Добавить в таблицу **10 самых дорогих товаров** товары с ценой в указанном интервале, например, от 11 000 до 20 000 у.е., по-прежнему выбирая самые дорогие товары. Переименовать запрос в «Самые дорогие товары».
- 2) Добавить в таблицу **Товары** все записи из таблицы **Ассортимент**;
- 3) Добавить в таблицу **Типы** все записи из таблицы **Категории товаров**;
- 4) Удалить таблицы **Ассортимент** и **Категории товаров** (только убедившись, что добавление прошло успешно).

3. Запрос на обновление обновляют все записи, удовлетворяющие определенному условию, т.е. вносят только указанные изменения.

- Создать запрос на выборку, извлекающий записи, которые нужно обновить;
- Команда <Запрос—Обновление>.
- Выполнить запрос. Появится диалоговое окно, указывающее, сколько записей обновит запрос.

**Задание 2. 8.** Увеличить цену товаров в таблице Самые дорогие товары на 6%.

4. Запрос на удаление удаляет все записи, удовлетворяющие определенному условию.

- Создать запрос на выборку, извлекающий записи, которые нужно удалить из таблицы;
- Команда <Запрос—Удаление>. Вести соответствующее условие.
- Выполнить запрос. Появится диалоговое окно, указывающее, сколько записей удалится при выполнении запроса.

**Задание 2. 9.** Удалить из таблицы Самые дорогие товары товары с ценой меньшей заданного числа.

### **ИСПОЛЬЗОВАНИЕ ГРУППОВЫХ ОПЕРАЦИЙ В ЗАПРОСАХ. ВЫЧИСЛЯЕМЫЕ ПОЛЯ**

Выражения, определяющие вычисляемые поля, создаются с помощью мастера простых запросов или вводятся пользователем в строку <Групповая операция> бланка запроса, в которой допускается выбор статистических функций для выполняемых над полем вычислений.

В бланке запроса задают также условия отбора, с помощью которых определяются *группы*, для которых вычисляются *итоговые значения*, записи, включаемые в вычисления, или результаты, отображаемые после выполнения расчетов.

Заполняя ячейки в строке <Групповая операция> в бланке запроса, можно выполнить расчеты *для групп записей* и вычислить сумму, среднее, количество или другой тип итогового значения для вычисляемого поля.

**Задание 2. 10.** Создать запрос «Промежуточная сумма заказа» (рис.5):

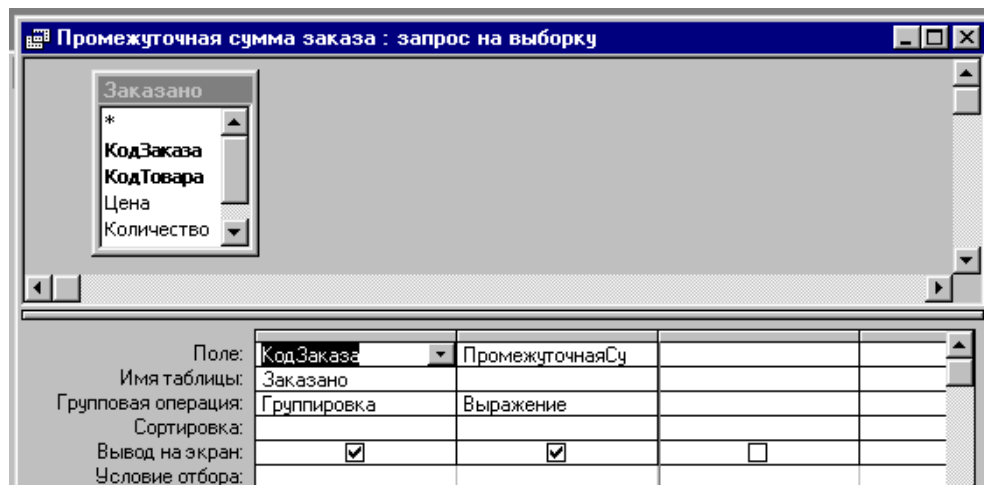


Рис.5. Бланк запроса «Промежуточная сумма заказа»

- в бланк запроса добавить одно поле из таблицы **Заказано**;
- создать вычисляемое поле:

**ПромежуточнаяСумма:  $\text{Sum}(\text{CCur}([\text{Цена}] * [\text{Количество}] * (1 - [\text{Скидка}]) / 100) * 100)$**

и выполнить команду Вид—Групповые операции и для поля Код заказа в строке **Групповые операции** установить операцию *Группировка*, а для вычисляемого поля – *Выражение*.

**Задание 2. 11.** По схеме предыдущего задания создайте запрос "Продажа товаров в 2005г".

- Таблицы, участвующие в запросе: **Товары, Заказы, Заказано**;
- Поля, участвующие в запросе:

<Марка>

<Дата исполнения> - не выводимое поле с условием выбора записей

**Between #01.01.2005# And #31.12.2005#**

Вычисляемое поле <ПродажиТоваров>:

**$\text{Sum}(\text{CCur}([\text{Заказано}].[Цена] * [\text{Количество}] * (1 - [\text{Скидка}]) / 100) * 100)$**

Пояснение. Запись типа [Заказано].[Цена] означает выбор поля <Цена>из таблицы **Заказано**.

- установить сортировку по возрастанию для поля <Код заказа>.

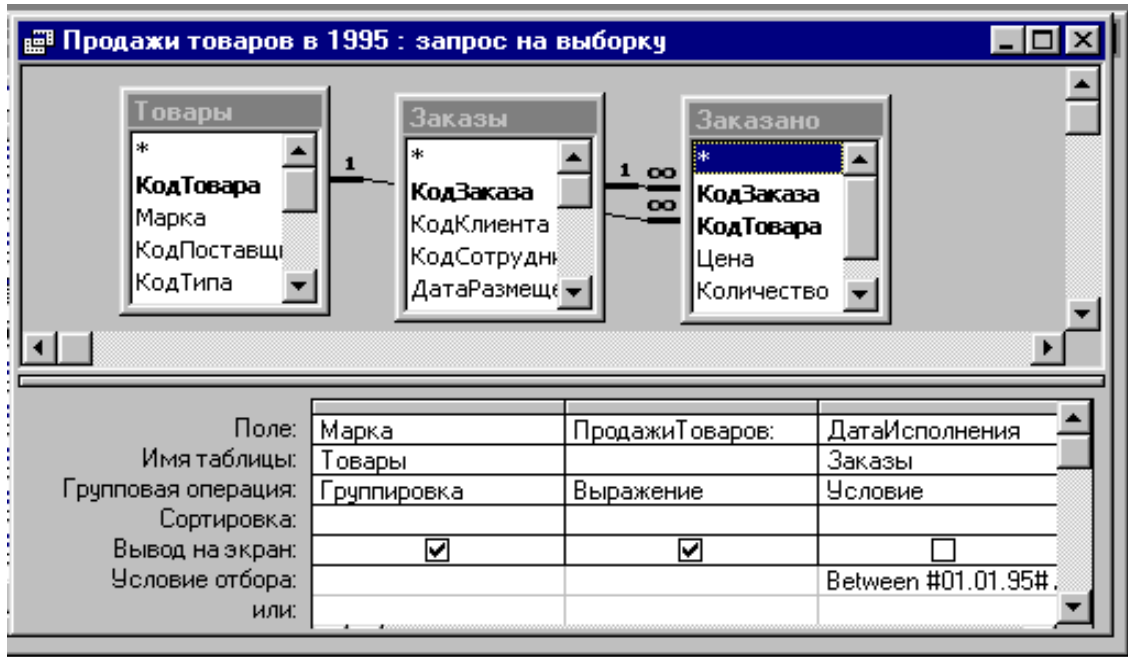


Рис.6. Бланк запроса «Продажа товаров в 2005г»

## ЗАПРОСЫ С ПАРАМЕТРАМИ И РЕЛЯЦИОННЫЕ ЗАПРОСЫ

**Запросы с параметрами** запрашивают необходимые условия при каждом выполнении и создаются следующим образом:

- Разработать запрос на выборку.
- В строке <Условие отбора> введите текст параметра, заключенный в квадратные скобки.

Пример. В поле <Дата заказа> в строке *Условие отбора* ввести выражение [**Введите дату заказа:**]

Если необходимо добавить параметр в поле <Количество>, который запрашивал бы максимальное и минимальное значения, можно использовать выражение **between** [**Введите минимальное количество:**] and [**Введите максимальное количество:**]. Сохранить запрос.

### Задание 2. 12.

- 1) Вывести из таблицы **Клиенты** всех клиентов из определенного города, запрашиваемого с клавиатуры.
- 2) Из таблицы **Сотрудники** вывести сотрудников с определенной датой найма (напр. диапазон за 2002-й, 2003-й года.), запрашиваемой с клавиатуры.
- 3) Из таблицы **Товары**
  - а) уметь выводить записи для определенного типа продукции;

б) вычислить суммарную стоимость каждого типа продукции (т.е. , напр., напитки на общую сумму \_\_\_\_\_).

4) Из таблицы **Заказы** уметь выводить записи, удовлетворяющие определенному диапазону даты исполнения заказа.

**Реляционные запросы** создаются на основе таблиц из БД **Фирма Лакомка**. Алгоритм создания реляционного запроса:

1. В окне БД открыть вкладку <Запросы>, щелкнуть на кн. <Создать>. Появится окно <Новый запрос>, в правой части которого перечислены имеющиеся мастера запросов.

2. Выбрать пункт <Конструктор>, ОК. Появится окно запроса вместе с диалоговым окном <Добавление таблицы>.

3. Добавить в запрос нужные таблицы, дважды щелкая на имени каждой из них в диалоговом окне <Добавление таблицы>, затем щелкнуть на кн. <Закрыть>.

4. Создать нужные реляционные связи между таблицами, перенося мышью общее поле (поля) из одной таблицы в другую. После этого на экране появятся линии, соединяющие таблицы (установленные связи между таблицами на уровне БД тоже добавляются по умолчанию).

5. Добавить в запрос необходимые поля, перенося имена полей в столбцы бланка запроса.

6. Задать условия отбора данных, сортировку и вычисляемые поля.

7. Сохранить и выполнить запрос.

**Задание 2. 13.** Создать запрос «Сведения о заказах».

Порядок выполнения работы:

а) добавить в запрос таблицы **Товары** и **Заказано**;

в) поля, участвующие в запросе:

<Код заказа> (**Заказано**)

<Код товара> (**Заказано**)

<Минимальный запас> (**Товары**)

<Количество> (**Заказано**)

<Скидка> (**Заказано**)

*Вычисляемое поле* <Отпускная цена>:

Отпускная цена:  $C_{\text{отп}}([\text{Заказано}].[\text{Цена}]*[\text{Количество}]*(1-[\text{Скидка}])/100)*100$

г) установить сортировку по возрастанию для поля <Код заказа>.

### **III. ПРАКТИЧЕСКАЯ РАБОТА №3. Нормализация отношений. Установление связей между таблицами**

#### **УСТАНОВКА СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ.**

**Цель работы:** получение навыков работы по установлению связей между таблицами; получение навыков в нормализации отношений.

**Темы для предварительного изучения.** Назначение связей между таблицами. Целостность базы данных. Порядок создания связей между таблицами. Схема данных. Три нормальные формы отношений.

Установление связей между таблицами необязательно, но это экономит время, гарантирует получение правильных результатов при разработке сложных форм и отчетов, поддерживает *целостность базы данных* (защита данных от изменений и удалений, которые могут нарушить связи между записями в таблицах).

Реляционные связи между таблицами могут быть разного типа:

- Одно-однозначные (1:1)
- Одно-многочисленные (1:M); схематичное обозначение  $1-\infty$
- Много-многочисленные (M:N)

Одно-однозначные связи (1:1) имеют место, когда каждому экземпляру первого объекта (А) соответствует только один экземпляр второго объекта (В) и наоборот.

#### Алгоритм установления связей:

- Выполнить команду <Сервис—Схема данных>. Способы: 1) вкладка Таблицы—контекстное меню <Схема данных>; 2) команда меню <Сервис--Схема данных>. Если связи устанавливаются впервые, то появится диалоговое окно <Добавление таблицы>.
- Выбрать таблицу, которая используется при установке связи, щелкнуть по кн. <Добавить>. Повторить эти действия для каждой добавляемой таблицы и затем кн. <Закреть>.
- Для создания связей между таблицами переместить поле, которое нужно связать, из исходной таблицы или запроса в соответствующее поле другой таблицы или запроса (как правило, связываются ключевые поля обеих таблиц). Следует учесть, что связываемые поля должны иметь один тип (кроме поля счетчика, которое можно связывать с числовым полем). Появится диалоговое окно <Связи>.

- Если режим <Обеспечение целостности данных> включен, то можно использовать остальные флажки, расположенные ниже. Режим <Каскадное обновление связанных полей> означает: изменение в связанном поле первой таблицы автоматически будут перенесены в поля связанной таблицы, содержащей те же данные. Каскадное удаление связанных полей: удаление в первой таблице автоматически приводит к удалению соответствующих записей связанной таблицы.
- Выбрать тип связи (чаще всего используется Один ко многим)
- Закрыть окно <Связи> с сохранением связи.

**Замечание.**

Установленные связи можно редактировать: 1) открыть окно <Схема данных>; 2) дважды щелкнуть на линии между таблицами; 3) в диалоговом окне <Связи> для удаления связи выделить связывающую линию и кл.<Delete>. Для просмотра связей: команда <Сервис—Схема данных> (рис.4).

**Задание. Установить указанные на рис.4 связи и уметь их объяснить:**

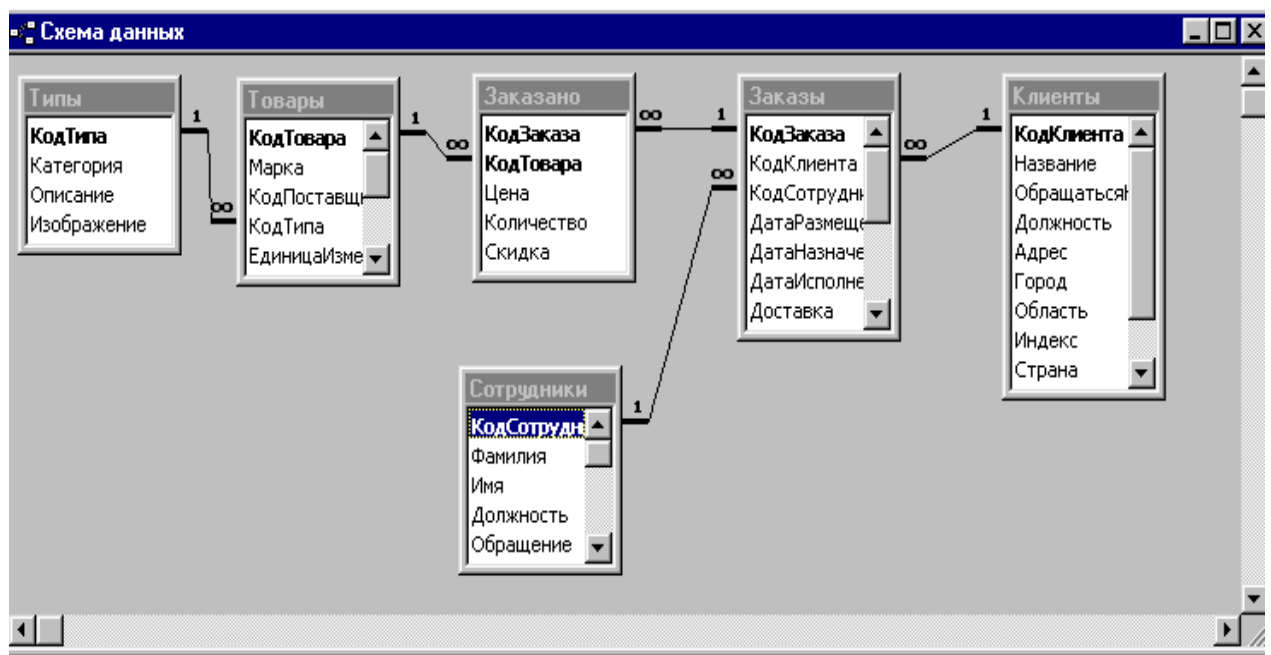


Рис.4. Пример схемы данных

**НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ**

- процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий в хранении данных с целью приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных.



**Задание 3.1.** Предположим, все заказанные продукты регистрируются в таблице

**ЗаказанныеПродукты:**

**ЗаказанныеПродукты**

Idзаказ	Idпро- дукт	Idклиент	Адрес	Количество	ДатаЗаказа
10265	17	BLONP	24,place Kleber	30	07.25.96
10265	70	BLONP	24,place Kleber	20	07.25.96
10278	44	BERGS	Berguvsvagen 8	16	08.12.96
10278	59	BERGS	Berguvsvagen 8	15	08.12.96
10280	75	BERGS	Berguvsvagen 8	30	08.14.96
10289	3	BSBEV	Fauntleroy Circus	30	08.26.96
10289	64	BSBEV	Fauntleroy Circus	9	08.26.96
10297	39	BLONP	24,place Kleber	60	09.04.96
10297	72	BLONP	24,place Kleber	20	09.04.96

### 1 этап. Приведение таблицы к 1НФ.

Структура ненормализованной таблицы:

Idзаказ (ПК)
Idпродукт (ПК)
Idклиент
Адрес
Количество
ДатаЗаказа

Чтобы таблица соответствовала *первой нормальной форме*, все значения ее полей должны быть атомарными и все записи - уникальными. Поэтому *любая реляционная таблица по определению уже находится в первой нормальной форме*.

Тем не менее, эта таблица содержит избыточные данные, например: одни и те же сведения о клиенте повторяются в записи о каждом заказанном продукте. Результатом избыточности данных являются проблемы модификации данных (добавление, изменение, удаление записей). Например, при редактировании данных в таблице **ЗаказанныеПродукты** могут возникнуть следующие проблемы:

- Адрес конкретного клиента может содержаться в БД только тогда, когда клиент заказал хотя бы один продукт.
- При удалении записи о заказанном продукте одновременно удаляются сведения о самом заказе и о клиенте, его разместившем.
- Если заказчик сменил адрес, придется обновить все записи о заказанных им продуктах.

### 2 этап. Приведение таблицы к 2НФ.

Говорят, что реляционная таблица находится во второй нормальной форме, если она находится в первой нормальной форме и ее неключевые поля *полностью зависят от всего первичного ключа*.

Таблица **ЗаказанныеПродукты** находится в первой, но не во второй нормальной форме, так как поля <Idклиент>, <Адрес>, <ДатаЗаказа> зависят только от поля <Idзаказ>, который является только частью составного первичного ключа (<Idзаказ>, <Idпродукт>).

Чтобы перейти от первой нормальной формы ко второй, нужно выполнить следующие шаги:

1. Определить, на какие части можно разбить первичный ключ, так чтобы некоторые из неключевых полей зависели от одной из этих частей (эти части не обязаны состоять из одной колонки!).

2. Создать новую таблицу для каждой такой части ключа и группы зависящих от нее полей и переместить их в эту таблицу. Часть бывшего первичного ключа станет при этом первичным ключом новой таблицы.

3. Удалить из исходной таблицы поля, перемещенные в другие таблицы, кроме тех из них, которые станут внешними ключами.

4. Например, для приведения таблицы **ЗаказанныеПродукты** ко второй нормальной форме нужно переместить поля Idклиент, Адрес, ДатаЗаказа в новую таблицу (назовем ее **ИнфоЗаказы**), при этом поле Idзаказ станет первичным ключом новой таблицы.

**Таблица 5.** ИнфоЗаказы и Заказы

<b>ИнфоЗаказы</b>	<b>Заказы</b>
Idзаказ (ПК)	Idзаказ (ПК, ВК)
Idклиент	Idпродукт (ПК)
Адрес	Количество
ДатаЗаказа	

В результате новые таблицы приобретут вид:

Таблица 6. ИнфоЗаказы и ЗаказыДетали

**ИнфоЗаказы**

<b>Idзаказ</b>	<b>Idклиент</b>	<b>Адрес</b>	<b>ДатаЗаказа</b>
10265	BLONP	24,place Kleber	07.25.96
10278	BERGS	Berguvsvagen 8	08.12.96
10280	BERGS	Berguvsvagen 8	08.14.96
10289	BSBEV	Fauntleroy Circus	08.26.96
10297	BLONP	24,place Kleber	09.04.96

**ЗаказыДетали**

<b>Idзаказ</b>	<b>Idпродукт</b>	<b>Количество</b>
10265	17	30

10265	70	20
10278	44	16
10278	59	15
10280	75	30
10289	3	30
10289	64	9

Однако таблицы, находящиеся во второй, но не в третьей нормальной форме, по-прежнему содержат аномалии модификации данных:

- Адрес конкретного клиента по-прежнему может содержаться в базе данных только тогда, когда клиент заказал хотя бы один продукт.
- Удаление записи о заказе в таблице ИнфоЗаказы приведет к удалению записи о самом клиенте.
- Если заказчик сменил адрес, придется обновить несколько записей (хотя, как правило, их меньше, чем в предыдущем случае).

Устранить эти аномалии можно путем перехода к третьей нормальной форме.

### 3 этап. Приведение таблицы к ЗНФ.

Говорят, что реляционная таблица находится в *третьей нормальной форме*, если она находится во второй нормальной форме и все ее неключевые поля зависят только от первичного ключа.

Таблица **ЗаказыДетали** уже находится в третьей нормальной форме. Неключевое поле <Количество> полностью зависит от составного первичного ключа (<Idзаказ>, <Idпродукт> ). Однако таблица **ИнфоЗаказы** в третьей нормальной форме не находится, т.к. содержит зависимость между не ключевыми полями (так называемая *транзитивная зависимость*): поле <Адрес> зависит от поля <IdКлиент>.

Чтобы перейти от второй нормальной формы к третьей, нужно выполнить следующие шаги:

1. Определить все поля (или группы полей), от которых зависят другие поля.
2. Создать новую таблицу для каждого такого поля (или группы полей) и группы зависящих от него полей и переместить их в эту таблицу. Поле (или группа полей), от которого зависят все остальные перемещенные поля, станет при этом первичным ключом новой таблицы.
3. Удалить перемещенные поля из исходной таблицы, оставив лишь те из них, которые станут внешними ключами.

Для приведения таблицы **ИнфоЗаказы** к третьей нормальной форме создадим новую таблицу **Клиенты** и переместим в нее поля <IdКлиент> и <Адрес>. Поле <Адрес> из

исходной таблицы удалим, а поле <IdКлиент> оставим - теперь это внешний ключ (табл.7).

Итак, после приведения исходной таблицы к третьей нормальной форме таблиц стало три - **Клиенты, Заказы, ЗаказыДетали**.

Таблица 7. Клиенты, Заказы и ЗаказыДетали

Клиенты	Заказы	ЗаказыДетали
	Idзаказ (FK)	Idзаказ (ПК, BK)
IdКлиент(ПК)	IdКлиент (BK)	Idпродукт (ПК)
Адрес	ДатаЗаказа	Количество

Заказы		
Idзаказ	IdКлиент	ДатаЗаказа
10265	BLONP	07.25.96
10278	BERGS	08.12.96
10280	BERGS	08.14.96

Клиенты	
IdКлиент	Адрес
ANATR	Avda.de la Constitucion 22
BERGS	Berguvsvagen 8
BLONP	24,place Kleber

- Сведения об адресе клиента можно хранить в БД, даже если это только потенциальный клиент, еще не разместивший ни одного заказа.
- Сведения о заказанном продукте можно удалять, не опасаясь удаления данных о клиенте и заказе.
- Изменение адреса клиента или даты регистрации заказа теперь требует изменения только одной записи.

**Задание 3.2.** Созданы таблицы, содержащие сведения, относящиеся к разным предметным областям. Привести соответствующие таблицы к третьей нормальной форме.

Таблицы находятся в каталоге Student\Колесникова\БД\Нормализация

#### IV. ПРАКТИЧЕСКАЯ РАБОТА №4. СОЗДАНИЕ ФОРМ. ВЫЧИСЛЯЕМЫЕ ЭЛЕМЕНТЫ УПРАВЛЕНИЯ В ФОРМАХ. ФИЛЬТРЫ.

**Цель работы:** получение навыков работы по созданию форм и использованию вычисляемых элементов управления в формах; получение навыков работы по созданию многотабличных и многостраничных форм.

**Темы для предварительного изучения.** Назначение форм. Типы форм и порядок создания форм. Разделы форм в режиме Конструктор. Элементы управления. Форматирование элементов управления. Фильтры в формах. Назначение многотабличных и много-

страничных форм. Порядок создания указанных форм. Разделы форм в режиме <Конструктор>. Свойства используемых элементов.

**Задание 4.1.** Для каждой таблицы рабочей БД создать форму одним из способов:

**1. С использованием кнопки *Новый объект*.**

Если в окне БД выбрать таблицу, затем щелкнуть по данной кнопке и выполнить команду <Автоформа>, то для выбранной таблицы будет создана форма по умолчанию.

**2. С помощью мастера форм, выполнив следующие действия:**

- Вкладка <Формы> в окне БД, кн. <Создать>.
- В ДО <Новая форма> щелкнуть по строке <Мастер форм>, а в списке внизу окна выбрать таблицу.
- Следовать указаниям мастера.

**3. Разработка форм вручную.**

- Вкладка <Формы> в окне БД, кн. <Создать>.
- Выбрать пункт <Конструктор> и выбрать таблицу или запрос, ОК.
- Поместить на форму нужные объекты: поля, текст, графика. Для настройки панели инструментов конструктора форм используйте команду <Вид—Панель элементов>.
- Сохранить форму.

**Задание 4.2.** Перейдя в режим <Конструктор> какой-либо формы, "поиграть" с объектами: изменить размер, переместить на новое место, поместить на форму новый текст, графику, поработать с цветом. Основной инструмент при этом: <Панель элементов>.

**Задание 4.3.** Создать форму по запросу "Сведения о заказах", содержащую поля:

- <Товар> (поле <Марка> переименовать в поле <Товар>),
- <Количество>,
- <Цена>,
- <ОтпускнаяЦена>.

*Указание.* Использовать запрос "Сведения о заказах", изменив его таким образом (или в случае его отсутствия - создать), чтобы в динамическом наборе участвовали поля:

- <Код заказа> (**Заказано**)
- <Код товара> (**Заказано**)

<Марка> (Товары)

<Цена > (Заказано)

<Количество > (Заказано)

<Скидка > (Заказано)

Вычисляемое поле <Отпускная цена>

**Отпускная цена:Сум([Заказано].[Цена]\*[Количество]\*(1-[Скидка])/100)\*100**

**Задание 4.4.** Создать вычисляемый элемент управления <Промежуточная сумма>, подсчитывающий общую сумму заказа по следующей схеме:

- Открыть форму в режиме <Конструктор>;

Товар:	Цена:	
Camembert Pierrot	345 000р.	
Количество:	Скидка:	Отпускная цена:
4	5%	1 311 000р.
Промежуточная сумма: 1 285 353 819р.		

Рис.7. Вид формы Сведения о заказах

- На <Панели элементов> выбрать элемент <Поле>, зафиксировать левый верхний угол на свободном месте формы и, удерживая мышью нажатой, растянуть прямоугольник до правого нижнего.
- Ввести название поля на элементе, расположенном слева, и формулу на элементе справа: **=Sum([ОтпускнаяЦена])**

Оформить макет формы по своему усмотрению, поля расположить, следуя рис. 7

**Задание 4.5.** Создайте форму для просмотра сведений о факультете с указанием названия факультета, списка преподавателей, работающих на этом факультете, списком студентов, обучающихся на факультете и имеющих уровень интеллекта ниже введенного пользователем значения, общим количеством преподавателей, средним интеллектуальным уровнем студентов этого факультета. Выбор названия факультета должен производиться из *раскрывающегося списка*. Используйте подчиненные формы ленточного формата и для

студентов и для преподавателей. В подчиненной форме для студентов должны отображаться поля <ФИО> и <Уровень интеллекта студента>. В подчиненной форме для преподавателей должны отображаться ФИО и название должности, занимаемой преподавателем. Для подсчета количества и среднего используйте элемент управления поле. Сохраните форму под именем **Факультеты**.

*Указание.* Используйте БД **Учебный процесс**.

## СОЗДАНИЕ МНОГОТАБЛИЧНЫХ И МНОГОСТРАНИЧНЫХ ФОРМ

**Задание 4.6.** Для таблицы **Сотрудники** создать дополнительную таблицу **Часы**, содержащую поля:

<Код сотрудника> (в соответствии с таблицей **Сотрудники**, тип поля - числовой)

<Дата оплаты>

<Часы> (количество отработанных часов)

<Сверхурочные часы>

Заполнить таблицу **Часы** (создать записи) таким образом, чтобы для каждого сотрудника было 2-4 записи.

Создать формы для обеих таблиц (можно с помощью мастера).

Создать на основе формы **Сотрудники** многотабличную форму, т.е. такую, которая в одном окне отображает данные из нескольких таблиц. Порядок выполнения задания:

- Открыть *главную форму* в режиме <Конструктор>;
- Перейти в окно БД, выбрать вкладку <Формы>;
- Переместить форму, которая будет использоваться как *подчиненная*, из окна БД в главную форму. В главной форме добавится *элемент управления* подчиненной формой. Его можно переместить в нужное место.
- В *контекстном меню* для элемента управления выбрать команду <Свойства>, вкладку <Данные>, свойство <Подчиненные поля> и убедитесь, что Access создал связь, используя нужные вам поля. Если связь не установлена, введите имя поля, которое будет использоваться для связи записей внутри подчиненной формы.
- Закройте форму и сохраните ее.

### Замечания.

1. При просмотре многотабличной формы в режиме формы записи главной таблицы появятся в главной части формы, а все связанные записи – внутри подчиненной формы.

2. Access правильно устанавливает связь, если имена соответствующих полей совпадают и если связь между таблицами была определена на уровне таблиц (<Схема данных>), при этом поле главной таблицы – ключевое.

Создание **многостраничных форм**. Каждая вкладка такой формы может содержать обычные для форм поля и объекты. Для создания многостраничной формы используется кнопка <Набор вкладок>, которая находится на ПИ конструктора форм. Для создания такой формы:

- Создайте вручную новую форму на основе какой-либо таблицы или запроса (в режиме <Конструктор>)
- Если панели инструментов нет на экране, отобразить ее, выбрав команду <Вид— Панель инструментов>.
- Выберите кнопку <Набор вкладок> ПИ.
- Щелкните на форме на том месте, где должен находиться верхний левый угол элемента управления, и, удерживая кнопку мыши, переместите указатель мыши в нижний правый угол (это делается для определения размеров элемента). Когда Вы отпустите кнопку мыши, в форму будет вставлен элемент управления, состоящий из двух вкладок.
- Чтобы добавить элемент управления в созданную вкладку, откройте нужную вкладку и добавьте в нее элементы, используя список полей и ПИ (кроме элемента <Набор вкладок>).

**Задание 4.7.** Создайте для таблицы **Сотрудники** многостраничную форму, содержащую вкладки <Личная информация> и <Служебная информация>. Поля разместите по своему усмотрению.

## **V. ПРАКТИЧЕСКАЯ РАБОТА №5. СОЗДАНИЕ ПРОСТЫХ МАКРОСОВ, КНОПОЧНЫХ ФОРМ, МОДУЛЕЙ, ОТЧЕТОВ**

**Цель работы:** получение навыков работы по созданию макросов и кнопочных форм; получение навыков работы по созданию отчетов.

**Темы для предварительного изучения.** Назначение макросов и кнопочных форм. Порядок создания макросов. Использование условий. Режимы создания кнопочных форм. Назначение отчетов. Порядок создания отчетов. Разделы бланка отчетов. Элементы управления. Форматирование элементов управления. Сортировка и группировка. Сохранение и печать отчета.



### **Задание 5.1. Создать макросы, открывающие формы, созданные по таблицам.**

- Вкладка <Макросы>—кн. <Создать>.
- Ввести макрокоманды. Для этого щелкнуть на кнопке раскрытия списка на первой пустой ячейке в столбце макрокоманда (или <Alt+ >). Откроется список допустимых макрокоманд.
- Выбрать нужную команду, в частности <Открыть форму>.
- Щелкнуть в нижней части окна (или F6), указать аргументы действия (в нашем случае указать название формы).
- При необходимости снабдить макрокоманды комментариями.

Выполните 1-2 макроса. Макрос на выполнение можно запускать по-разному:

- Самый простой способ – выбор макроса и щелчок на кн. <Запуск>, или двойной щелчок на имени макроса в окне БД.
- Другой способ – команда <Сервис—Запуск макроса>, ввести имя макроса в ДО.
- Третий способ – с помощью кнопок, добавляемых на формы.

Свяжите макрос с кнопкой (способ *перетаскивание*):

- Открыть нужную форму в режиме <Конструктор>. Установите ее размеры таким образом, чтобы форма и окно БД были видны одновременно.
- Открыть вкладку <Макросы>.
- Перетащить макрос в выбранное место формы.

Пользуясь контекстно-зависимым меню, отредактируйте кнопку: попробуйте создать рисунок на кнопке, надпись, измените ее размеры.

Создайте на формах кнопки, открывающие уже созданные запросы (с параметром, с вычисляемым полем и т.д.).

Используйте другой способ создания кнопки (элемент управления <Кнопка>).

## **СОЗДАНИЕ ОТЧЕТОВ**

### **Задание 5.2. Создать простой отчет, отображающий результаты обработки информации для прикладной области, выбранной в соответствии с вариантом задания.**

Пример выполнения задания. Создадим простой отчет, содержащий:

- список сотрудников предприятия с полями: <Идентификационный код>, <Фамилия>, <Имя>, <Отчество>, <Телефон>;
- Сгруппировать данные по 1-ой букве фамилии;

- Вычислить количество фамилий в каждой группе.

Примерный вид требуемого отчета представлен на табл. 1.

Таблица 1. Вид простого отчета

СПИСОК СОТРУДНИКОВ ПРЕДПРИЯТИЯ 26.02.06				
Идентификационный код	Фамилия	Имя	Отчество	Телефон
<i>Группа фамилий, начинающихся на букву</i>		<b>А</b>		
1545678990	<b>Архипов</b>	Сергей	Иванович	нет
<i>Количество в группе</i>		<b>1</b>		
<i>Группа фамилий, начинающихся на букву</i>		<b>Б</b>		
2314743296	<b>Бородулин</b>	Андрей	Васильевич	27-14-12
1955443781	<b>Безродный</b>	Владимир	Михайлович	32-32-14
2055894321	<b>Бронзов</b>	Станислав	Иванович	23-10-70
<i>Количество в группе</i>		<b>3</b>		
....				

Порядок выполнения действий:

1. Открыть базу данных (например, **Профсоюз**), для которой создается отчет;
2. Начать создание отчета в режиме <Конструктор>:

Выбрать вкладку <Отчеты>, нажать кнопку <Создать>.

Появляется окно <Новый отчет>, в котором выбрать <Конструктор>(для самостоятельного создания отчета), указать источник данных - таблицу **Сотрудник** (выбрать таблицу из раскрывающегося списка с перечнем таблиц и запросов), ОК.

На экран выводятся окна, представленные на рис. 8:

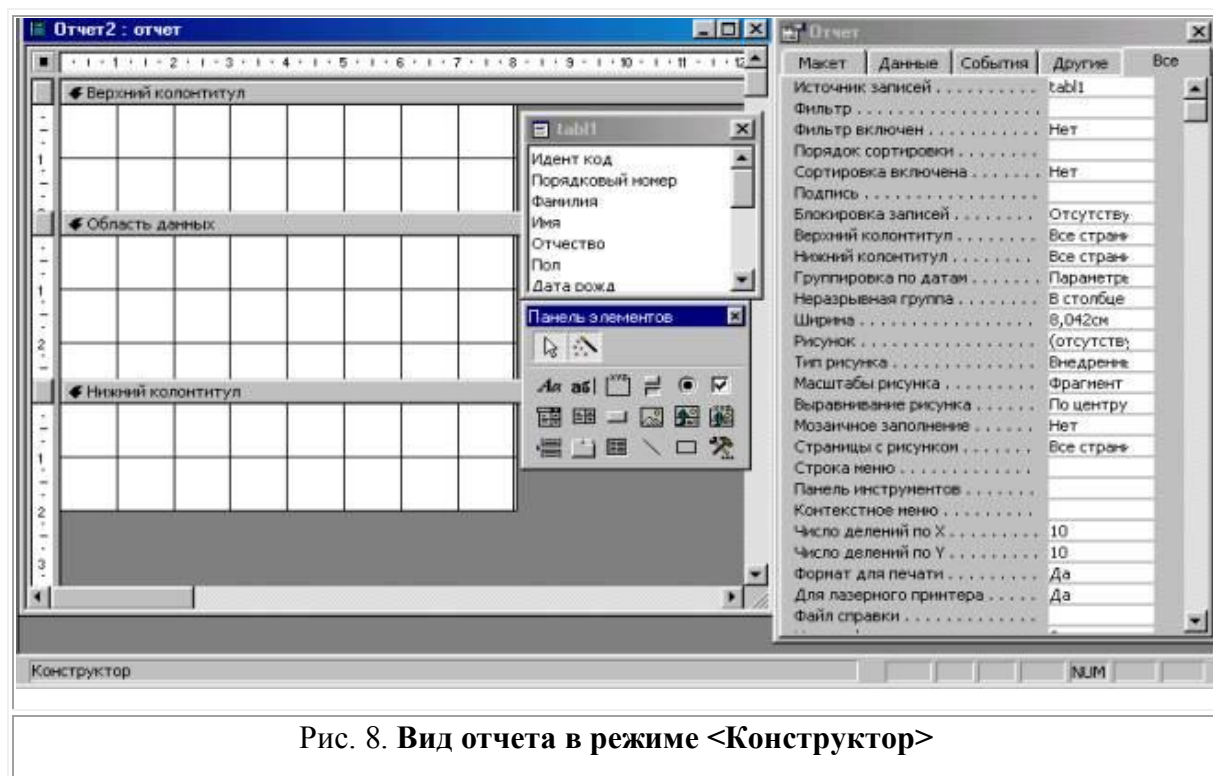


Рис. 8. Вид отчета в режиме <Конструктор>

а) Окно отчета в режиме конструктора с заголовком **Отчет1: отчет** и со следующими областями: <Верхний колонтитул>; <Область данных>; <Нижний колонтитул>.

б) *Панель элементов*, содержащая кнопки для создания элементов управления, которые можно включить в отчет (панель элементов можно закрыть или вывести, выполнив щелчок по соответствующей кнопке на панели инструментов или выполнив команду <Вид – Панель элементов>;

в) *Список полей* базовой таблицы или запроса (список полей таблицы **Сотрудник**) (список полей можно вывести или закрыть, выполнив команду <Вид – Список полей или выполнив щелчок по соответствующей кнопке на панели инструментов).

Перемещая окна, можно расположить их в удобном для работы порядке, например: окно отчета - слева, список полей в правой области экрана, ниже -панель элементов.

Дополнительно можно вывести окно свойств создаваемого отчета (<Вид – Свойства>) или выполнив щелчок по пиктограмме Свойства.

### 3. Установить размеры отчета:

- Переместить правую границу окна создания отчета с помощью указателя мыши так, чтобы на верхней линейке было видно число 19 (размер отчета 18 см);
- Выполнить <Файл – Параметры страницы>;
- При выбранной вкладке <Страница> установить книжную ориентацию листа и размер А4 (210x297);

- При выбранной вкладке <Поля> установить размеры левого и правого поля по 10 мм;
  - При выбранной вкладке <Столбцы> установить: количество столбцов-1; ширина столбца-18 см; высота - 3 см;
  - Переместить правую границу области данных отчета до значения 18 на верхней линейке;
4. Добавить в бланк отчета области <Заголовок отчета> и <Примечание отчета> (выполнить <Вид – Заголовок/Примечание отчета>).
  5. Переместить из таблицы в <Область данных> список нужных полей:
    - В окне таблицы **Сотрудник** выделить в комбинации с клавишей <Ctrl> поля <Идентификационный код>, <Фамилия>, <Имя>, <Отчество>, <Телефон> и отбуксировать их в <Область данных>. В <Области данных> появятся связанные элементы управления, т.е. элементы, связанные с полями таблицы **Сотрудник** (слева - подпись, справа - значение поля). Выполнить щелчок мышью на свободном пространстве в области данных, чтобы убрать выделение вставленных элементов управления;
    - Можно просмотреть содержимое отчета на данном этапе, выбрав <Файл – Предварительный просмотр>. В дальнейшем можно использовать эту команду для просмотра содержимого отчета после внесения каких-либо изменений;
    - Переместить заголовки столбцов в область <Верхний колонтитул>, выполнив следующие действия:
      - Выделить подписи элементов управления (слева) в <Области данных>, для чего нажать клавишу <Shift> и выполнить щелчок на каждой подписи (или обвести их слева направо с нажатой левой кнопкой мыши). Выполнить команду <Вырезать>;
      - Активизировать <Верхний колонтитул>щелчком мыши по заголовку и выполнить команду <Вставить>. Подписи будут вставлены в область <Верхнего колонтитула>;
      - Расставить заголовки столбцов следующим образом: подпись <Идентификационный код> переместить в левый верхний угол области. Остальные подписи расставить так, чтобы расстояние между левыми границами подписей было равно 3 см;
      - Выполнить редактирование и форматирование заголовков столбцов (в соответствии с табл. 1). Для этого выделить все подписи в строке (поместить курсор мыши слева от строки, чтобы курсор принял форму стрелки, направленной вправо и выполнить щелчок мышью), щелкнуть правой кнопкой мыши на выделении, в появившемся окне выбрать команду <Свойства> и установить во вкладке <Макет> следующие значения: Ширина 3 см, Высота 1 см, Тип границы Сплошная, Размер шрифта 12 см. Для редактирования под-

писи выделить элемент, выполнить щелчок мышью на подписи и внести нужные изменения.

*Замечания:* Форматировать можно и отдельный элемент, выделив только его. При форматировании можно использовать пиктограммы панели форматирования на панели инструментов, или команды меню <Формат – Выровнять>, <Формат – Размер>, или установить соответствующие свойства для данного элемента.

- Уменьшить размер области <Верхний колонтитул> по размеру высоты заголовков столбцов, добавив приблизительно 0,5 см, переместив границу следующей области Область данных вверх;

- Разместить поля в <Области данных> в ряд под заголовками соответствующих столбцов. Уменьшить размер <Области данных> мышкой переместив нижнюю границу области вверх. Отформатировать каждое поле в соответствии с рис.1;

- Просмотреть содержимое отчета на данном этапе, выбрав <Файл – Предварительный просмотр>. Внести изменения, если есть несоответствия табл.1 или ошибки.

6. Определить поля, по которым будет производиться группировка и сортировка данных:

- Выполнить щелчок по пиктограмме <Сортировка и группировка> панели инструментов или выбрать <Вид – Сортировка и группировка>. Открывается окно <Сортировка и группировка>:

- в столбце <Поле/выражение> (левый столбец) открыть список полей и выбрать поле <Фамилия>;

- в столбце <Порядок сортировки> установить порядок сортировки (по возрастанию)

- Выполнить установку свойств в области <Свойства группы>:

- Для <Заголовок группы> и <Примечание группы> установить значение Да. При этом в окне отчета появляется дополнительная область <Заголовок группы> "Фамилия" перед <Областью данных> и <Примечание группы> после <Области данных>;

- Для группировки по первому символу установить в строке <Группировка> значение <По первым символам>;

- В строке <Интервал> установить число начальных символов, по которым хотим образовывать группы, для группировки по одному первому символу это значение равно 1;

- В строке <Не разрывать> установить значение <Полную группу>;

- Закрывать окно <Сортировка и группировка>.

7. Вставить в область <Заголовок группы> бланка отчета текст "*Группа фамилий, начинающихся на букву*", а затем должна выводиться буква, по которой создавалась очередная группа:

- Выделить область <Заголовок группы> в бланке отчета (щелкнуть на заголовке области);
- Из окна <Список полей> перетащить поле <Фамилия> в бланк отчета в область <Заголовок группы>. Поместить указатель с изображением руки с вытянутым указательным пальцем на маркер, расположенный в левом верхнем углу левого поля (подпись) и отбуксировать это поле в левый верхний угол области Заголовка группы, отступив 0,25см слева. Аналогично переместить правое поле вправо на 8см от левой границы области. В левом поле набрать текст "*Группа фамилий, начинающихся на букву*" и нажать клавишу <Enter>. Установить параметры форматирования: курсив, размер 10. Затем выполнить действия: <Формат–Размер–По размеру данных>. Изменить размеры правого поля, перемещая маркер правой границы влево так, чтобы видна была одна буква Ф и немного следующая.

- В окне <Панель элементов> щелкнуть по кнопке Линия, переместить курсор в область Заголовок группы под набранный текст и провести линию, подчеркнув оба поля;

- Просмотреть отчет, выбрав <Файл–Просмотр>. Если видны 2 буквы фамилии, то уменьшить поле Фамилия, если где-либо не видно буквы, то поле Фамилия увеличить (как описано предыдущем пункте).

9. Вставить в область <Примечание группы> бланка отчета текст "*Количество в группе*", а затем должно выводиться количество фамилий, относящихся к данной группе:

- Создать элемент управления. Для этого выполнить щелчок на панели элементов по кнопке **ab|**, а затем в области <Примечание группы> в месте расположения элемента. Появляется элемент, состоящий из 2-х частей. Переместить правую часть элемента вправо. В левую часть поля (подпись) ввести текст "*Количество в группе*". В правую часть элемента ввести формулу **=Count([Фамилия])**. Произвести форматирование.

10. Вставить при необходимости рисунок в заголовок отчета:

- <Вставка–Рисунок–Из файла>. В открывшемся окне выбрать подходящий графический файл, ОК.

11. Вставить текст заголовка отчета и рядом дату создания отчета:

- На панели элементов выбрать кнопку **ab|**, переместить курсор в область <Заголовок отчета> справа от рисунка. Отодвинуть правое поле ближе к правой границе области. В левое поле ввести текст заголовка. В правом поле с надписью <Свободный> ввести

формулу =Date(). Отформатировать поля. Вставить номер страницы в области <Нижний колонтитул>:

- Аналогично добавить элемент <Номер страницы>: в левое поле ввести текст "Страница", а в правое =Page. Сохранить отчет.

### Задание 5.3. Создать сложный отчет на основе ранее созданного запроса.

Создадим сложный отчет, содержащий:

- список сотрудников по подразделениям, имеющих в составе семьи более 1 человека;

- суммарный оклад и средний по подразделениям;
- суммарный оклад по всему предприятию.
- Вид отчета представлен на табл. 2.

Таблица 2. Вид сложного отчета

<u>Список по подразделениям сотрудников, имеющих в составе семьи более 1 чел.:</u>					26.02.06	
					15:15:00	
<i>Фамилия:</i>	<i>Имя:</i>	<i>Отчество:</i>	<i>Должность:</i>	<i>Оклад:</i>	<i>Состав семьи:</i>	
<u>Название подразделения:</u>			<u>Дирекция</u>			
Царева	Анна	Николаевна	гл.бухгалтер	430	1	
Каменева	Татьяна	Дмитриевна	начальник ОК	150	1	
Безродный	Владимир	Михайловна	зам.директора	500	1	
Архипов	Сергей	Иванович	директор	530	1	
<u>Суммарный оклад по подразделению:</u>			<u>Дирекция</u>	<b>1610</b>		
<u>Средний оклад по подразделению:</u>			<u>Дирекция</u>	<b>402,5</b>		
<u>Суммарный оклад по предприятию:</u>				<b>1610</b>		
<u>Название подразделения:</u>			<u>уч. кафедра</u>			
Строков	Олег	Викторович	преподаватель	350	2	
Старченко	Светлана	Борисовна	статистик	100	2	
Садчиков	Аркадий	Викторович	диспетчер	100	1	
Мапошенко	Юрий	Николаевич	специалист	150	2	
Бронзов	Станислав	Иванович	преподаватель	350	1	
<u>Суммарный оклад по подразделению:</u>			<u>уч. кафедра</u>	<b>1050</b>		

<u>Средний оклад по подразделению:</u>	<u>уч. кафедра</u>	210
<u>Суммарный оклад по предприятию:</u>		2660

- В связи с тем, что создаваемый отчет использует информацию из всех таблиц базы данных, будем строить отчет на базовом запросе "Запрос для отчета", созданном в предыдущей практической работе.

Действия будут аналогичны действиям, описанным в п. I данной практической работы при создании простого отчета.

1. Начать создание отчета в режиме <Конструктор>:

Выбрать вкладку <Отчеты>, нажать кнопку <Создать>. Появляется окно <Новый отчет>, в котором выбрать <Конструктор> (для самостоятельного создания отчета), указать источник данных - запрос "Запрос для отчета". Вид экрана - как и при создании отчета по п I.

2. Установить размеры отчета.

3. Добавить в бланк отчета области <Заголовок отчета> и <Примечание> отчета.

4. Определить поля, по которым будем группировать и сортировать данные:

- Установить группировку по полю <Название подразделения>, порядок сортировки по возрастанию.

5. Добавить области <Заголовок группы> и <Примечание группы>;

6. Выбрать из окна запроса "Запрос для отчета" в <Область данных> список всех полей, кроме <Название подразделения> и перенести заголовки колонок таблицы в <Верхний колонтитул>. Выполнить размещение и форматирование всех полей в областях <Верхний колонтитул> и <Область данных> в соответствии с табл. 2.

7. Заполнить область <Заголовок группы>:

8. Выбрать из окна запроса "Запрос для отчета" в <Заголовок группы> поле <Название подразделения> и изменить содержимое поля <подпись> (левое поле) на <Название подразделения>. Выполнить форматирование полей в соответствии с табл. 2.

9. Заполнить область <Примечание группы>:

- Переместить из списка полей запроса "Запрос для отчета" в <Примечание группы> поле <Название подразделения> 2 раза, расположив связанные поля друг под другом. Изменить название полей <подпись> (левое поле) на <Суммарный оклад> по подразделению для первой строки и <Средний оклад> по подразделению для второй строки;

- Рядом с 1-ой и 2-ой строками создать по одному не заполненному полю, щелкнув сначала на Панели элементов по кнопке **ab|**, а затем в области <Примечание группы> в



соответствующей строке. Убрать левые поля (подписи), выделив их и затем, щелкнув на кнопке <Вырезать> на панели элементов. В верхней строке в поле с надписью <Свободный> набрать формулу =Sum([Оклад]) (для вычисления суммарного оклада по подразделению), в нижней - =Avg([Оклад]) (для вычисления среднего оклада по подразделению). Разместить поля с формулами под заголовками соответствующих столбцов;

- Аналогично создать строку для вывода в отчет "Суммарного оклада по предприятию" в разделе <Примечание отчета>, создав сначала связанное поле и введя соответствующую подпись и формулу. Для поля с формулой в качестве значения свойства <Сумма с накоплением> установить значение, отличное <Отсутствует>.

10. Вставить в раздел <Заголовок отчета> текст заголовка отчета, а рядом дату и время создания отчета, вставить номер страницы. Для вывода даты и времени использовать в формуле функцию Now().

11. Выполнить форматирование полей в соответствии с табл. 2. Просмотреть отчет. Сохранить отчет.

**Задание 5.4.** Создайте отчет, содержащий информацию о читаемых каждым преподавателем предметах. Сгруппируйте предметы по коду преподавателя. Для каждой группы должны выводиться ФИО преподавателя, его код, название занимаемой им должности и общее количество часов нагрузки. Для каждого предмета должны выводиться его название и количество часов. Сохраните отчет под именем **Нагрузка преподавателей**.

*Указание.* Используйте БД "Учебный процесс".

## VI. ПРАКТИЧЕСКАЯ РАБОТА №6. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

**Цель работы:** получение навыков работы по проектированию базы данных, практическое освоение основных приемов и правил методологии информационного моделирования

**Темы для предварительного изучения.** Основные этапы разработки базы данных. Современные средства проектирования и программирования баз данных, освоение среды разработки концептуальной модели Erwin и реализация модели средствами Access.

*Пример проекта* базы данных гостиницы.

- 1 Введение
- 2 Постановка задачи
  - 2.1 Описание предметной области
  - 2.2 Формализованное описание задачи

- 3 Концептуальная модель данных
- 3.1 Сущности и связи ( ER-уровень)
- 3.2 Логика взаимосвязей данных ( KB-уровень)
- 3.3 Состав атрибутов сущностей (FA-уровень)
- 4 Глоссарий модели
- 5 Приложение. Описание типовых запросов
- 6 Заключение
- 7 Список литературы

## VII. ПРАКТИЧЕСКАЯ РАБОТА №7. SQL-запросы

**Цель работы:** получение навыков работы по созданию базы данных (БД), созданию и редактированию таблиц, созданию запросов на языке реляционных баз данных SQL.

**Темы для предварительного изучения.** Назначение SQL. Порядок создания БД и таблиц. Функциональные категории команд SQL: DDL, DML, DCD.

### Задание 7.1.

1. Создайте таблицы Salepeople(Продавцы), Customers(Покупатели), Orders(Заказы) с полями указанными в приложении.
2. Создайте таблицу Заказов так, чтобы все значения поля ONum, а также все комбинации полей CNum и SNum отличались друг от друга, и так что бы значения NULL исключались из поля даты.
3. Создайте таблицу Продавцов так, чтобы комиссионные по умолчанию составляли 10%, не разрешались значения NULL, чтобы поле SNum являлось первичным ключом.
4. Создайте таблицу Заказов, будучи уверенными в том, что значение в поле ONum больше чем значение в поле CNum, а CNum больше чем SNum. Запрещены значения NULL в любом из этих трех полей.
5. Заполните таблицы Salepeople, Customers, Orders значениями, указанными в приложении. Напишите команду, которая удаляет все заказы заказчика Clemens из таблицы Заказов.
6. Напишите команду, которая увеличивает рейтинг всех заказчиков в Риме на 100.
7. Продавец Serres оставил компанию. Переназначьте его заказчиков продавцу Motika.
8. Напишите команду SELECT, которая выводит номер заказа, сумму и дату для всех строк из таблицы Заказов.
9. Напишите запрос, который вывел бы все строки из таблицы Заказчиков, для которых номер продавца 11.
10. Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: City, SName, SNum, Comm.
11. Напишите команду SELECT, которая вывела бы оценку(Rating), сопровождаемую именем каждого заказчика в San Jose.
12. Напишите запрос, который вывел бы значения SNum всех продавцов в текущем порядке из таблицы Заказов, без каких бы то ни было повторений.

## Задание 7.2.

- 1.Создайте таблицы Salepeople(Продавцы), Customers(Покупатели), Orders(Заказы) со связями между ними (чтобы ваша база была в состоянии справочной целостности).
  - 2.Создайте таблицу Orders (Заказы). Считаем, что когда продавцы оставляют компанию, их текущие заказы не передаются другому продавцу и автоматически отменяются все заказы для заказчиков, чьи счета удаляются, а изменив номера продавца или заказчика можно автоматически передать заказы ему.
  - 3.Передайте другому пользователю право на изменение оценки(rating) заказчика.
  - 4.Передайте какому то менеджеру право передавать другим пользователям право делать запросы в таблице Заказов.
  - 5.Предоставьте какому то пользователю право вставки в таблицу Продавцов, а затем отмените эту привилегию.
  - 6.Передайте какому то пользователю право вставлять или модифицировать таблицу Заказчиков с сохранением его возможности задавать значения Rating в диапазоне от 100 до 500.
- Для выполнения заданий используйте таблицы указанные в приложении. Строки таблиц можете заполнить по своему усмотрению.

## Задание 7.3.

- 1.Напишите запрос, который вывел бы номер заказа и сумму комиссионных продавца для каждого заказа заказчика с рейтингом выше 100.
  - 2.Напишите запрос который вывел бы все пары номеров заказов по данным заказчикам с именами этих заказчиков, и исключал бы дубликаты из вывода.
  - 3.Напишите запрос, который вывел бы имена и оценки всех заказчиков, у которых величина приобретений выше среднего.
  - 4.Напишите запрос, который выберет имена и номера всех заказчиков с максимальными для их городов оценками, используя соотнесенный подзапрос.
  - 5.Напишите запрос, использующий оператор EXIST, который выберет всех продавцов с заказчиками, размещенными в их городах которые ими не обслуживаются.
  - 6.Напишите запрос который бы выбирал все заказы с суммой больше чем любая для заказчиков в некотором городе.
  - 7.Напишите команду, которая бы вывела имена и номера каждого продавца и каждого заказчика, которые имеют больше чем один заказ. Результат представьте в алфавитном порядке.
- Для выполнения задания используйте таблицы с полями указанными в приложении. Содержание таблиц может быть произвольным.

## Приложение.

Таблицы используемые в примерах.

Таблица Продавцы (Salepeople):

Snum	Sname	City	Comm
11	Peel	London	0.12
12	Serres	San Jose	0.13
14	Motika	London	0.11
17	Rifkin	Barcelona	0.15

13 Axelrod New York 0.10

**SNum** - уникальный номер назначенный каждому продавцу,

**SName** - имя продавца

**City** - расположение продавца( город ).

**Comm** - комиссионные продавцов в десятичной форме.

Таблица Заказчики (Customers):

Snum	Sname	City	Rating	SNum
21	Hoffman	London	100	11
22	Giovanni	Rome	200	13
23	Liu	SanJose	200	12
24	Grass	Berlin	300	12
26	Clemens	London	100	11
28	Cisneros	SanJose	300	17
27	Pereira	Rome	100	14

**CNum** - уникальный номер назначенный каждому заказчику,

**CName** - имя заказчика,

**City** - расположение заказчика( город ).

**Rating** - код указывающего уровень предпочтения данного заказчика перед другими ( рейтинг ).

**SNum** - номер продавца назначенного этому заказчику.

Таблица Заказы (Orders):

Onum	AMT	Odate	CNum	Snum
31	18.69	2007/10/03	28	17
33	767.19	2007/10/03	21	11
32	1900.10	2007/10/03	27	14
35	5160.45	2007/10/03	23	12
36	1098.16	2007/10/03	28	17
39	1713.23	2007/10/04	22	13
37	75.75	2007/10/04	24	12
38	4723.00	2007/10/05	26	11
310	1309.95	2007/10/06	24	12
311	9891.88	2007/10/06	26	11

**ONum** - уникальный номер данный каждому приобретению,

**AMT** - значение суммы приобретений,

**ODate** - дата приобретения,

**CNum** - номер заказчика делающего приобретение,

**SNum** - номер продавца продающего приобретение

## VIII. ПРАКТИЧЕСКАЯ РАБОТА №8. Создание Web страницы на основе базы данных

**Цель работы:** получение навыков работы по созданию Web-страницы на основе базы данных.

**Темы для предварительного изучения.** Настройка Internet Information Server, знакомство с HTML, ASP, настройка DSN по эл. источнику: Сороковиков В. Н. Разработка Web-сайта на основе базы данных. – Томск. – 2002.

**Задание. Опубликовать спроектированную в задании 6 БД.**

### **IX. Методические рекомендации по написанию курсовой работы.**

Курсовой проект выполнен с целью закрепления знаний, полученных в ходе изучения начального курса “Базы данных”, а также освоения практических навыков работы с современными средствами проектирования и программирования баз данных, а именно освоение среды разработки концептуальной модели Etwin и реализация модели средствами Access.

Основная цель системы обработки данных заключается в повышении эффективности работы компании, учреждения или организации. В нашем случае разрабатываемая система предназначена для повышения эффективности работы гостиницы (назовем ее «XXX»). Система обработки данных должна:

- обеспечивать получение общих или детализированных данных по итогам работы;
- позволять легко определять тенденции изменения важнейших показателей;
- обеспечивать получение информации, критической по времени, без существенной задержки;
- выполнять точный и полный анализ данных.

Среди наиболее ярких представителей систем управления базами данных можно отметить: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а также баз данных Microsoft SQL Server и Oracle, используемые в приложениях, построенных по технологии «клиент-сервер». Фактически, у любой современной СУБД существует аналог, выпускаемый другой компанией, имеющий аналогичную область применения и возможности, любое приложение способно работать со многими форматами представления данных, осуществлять экспорт и импорт данных благодаря наличию большого числа конвертеров. Наиболее интересной чертой этих пакетов являются их большие возможности интеграции, совместной работы и использования данных, так как данные пакеты являются продуктами одного производителя, а также используют сходные технологии обмена данными.

Одним из самых популярных среди СУБД является Microsoft Access. Основными преимуществами являются: популярность среди многих конечных пользователей и осуществление высокой устойчивости данных, простота в освоении, использовании непрофессиональными программистами, возможность подготавливать отчеты из баз данных различных форматов произвольной формы на основании различных данных; возможность разработки некоммерческих приложений.

**Анализ предметной области.** Гостиница «XXX» получает прибыль за предоставление номеров на определенный срок. Клиент - это физическое лицо, которое забронировало номер в гостинице. Работу гостиницы можно представить следующим образом:

Клиент приходит в гостиницу «XXX» для того, чтобы остановиться в ней на какой-то срок и обращается к сотруднику, который предоставляет возможность выбора интересующего клиента номера. После того, как клиент сделал свой выбор, сотрудник заносит сведения о клиенте и интересующем его номере. Клиент также имеет право выбрать определенные услуги, которые предоставляет гостиница «XXX». Эти услуги будут записаны на имя клиента и также включены в его счет.

Гостиница периодически готовит и направляет руководству отчеты о заказах предлагаемых услуг, имеющихся номерах. На основании этих отчетов руководство гостиницы обсуждает различные вопросы и принимает соответствующие решения по реализациям различных задач.

**Основные производственные функции гостиницы:**

- работа с клиентами;
- предоставление номеров и услуг клиентам.

В настоящей работе рассматривается только фрагмент бизнеса, связанный с непосредственным обслуживанием клиентов.

**Основными компонентами этого фрагмента являются:**

- клиенты;
- номера;
- дополнительные услуги.

**Основные бизнес-процессы:**

- предоставление номеров клиентам;
- предоставление различных услуг клиентам;
- контроль качества работы гостиницы.

**Временная последовательность выполнения процессов следующая:**

- обслуживание клиентов
- обновления сведений об имеющихся свободных номерах – ежедневно;
- отчеты о заказах – каждые три месяца.

База данных должна поддерживать накопление и хранение информации об основных компонентах бизнеса и автоматизированное выполнение бизнес-процессов.

**Формализованное описание задачи.**

**Наименование задачи:** Автоматизация управления гостиницей.

**Цель работы гостиницы:** Предоставление клиентам интересующих их номеров, обслуживание клиентов.

**Функции:**

- регистрация нового клиента;
- регистрация факта предоставления номера;
- обновление сведений о свободных номерах;
- регистрация заказов для клиентов;
- обслуживание клиентов.

**Бизнес – правила:**

- номер предоставляется клиенту после полной оплаты счета за номер;
- история продаж сохраняется в течение одного года;
- стоимость полного счета клиента определяется предоставленным номером и дополнительными услугами, также предоставленными клиенту;
- полный счет производится клиентом по истечению срока пребывания клиента в гостиницы.

**Перечень вводимой информации:**

- фамилия клиента
- имя клиента
- отчество клиента
- серия паспорта
- номер паспорта
- кем выдан паспорт
- дата выдачи
- контактный телефон клиента
- оплачивает
- класс номера
- количество мест
- стоимость номера
- описание
- дата прибытия клиента
- дата отъезда клиента
- наименование услуги
- стоимость услуг
- описание услуг.

**Основной пользователь:** основными пользователями БД являются сотрудники гостиницы. Сведения об имеющихся номерах могут быть доступны клиентам.

**Концептуальная модель данных.** В качестве предметной области разрабатываемой базы данных выбрана гостиница, занимающаяся предоставлением номеров, выбранных клиентами, и оказанием (по желанию клиентов) определенных услуг за отдельную плату. В базе данных должны фиксироваться сведения о клиентах, данные о свободных и занятых номерах гостиницы, сведения о заказах, а также информация об услугах, предоставляемых гостиницей и т.д.

Таблица **Клиенты** содержит информацию: код клиента (уникальный номер, при регистрации клиента), фамилия, имя, отчество, серия паспорта, номер паспорта, кем выдан, дата выдачи, телефон, оплата.

При учете номеров учитываются данные: код номера, наличие удобств, класс, стоимость, описание, фото. По этим данным формируется таблица «Номера».

В таблицу **Заказы** входят такие данные, как: код заказа, код услуги, код клиента, код сотрудника (выполняющий данный заказ), количество заказов.

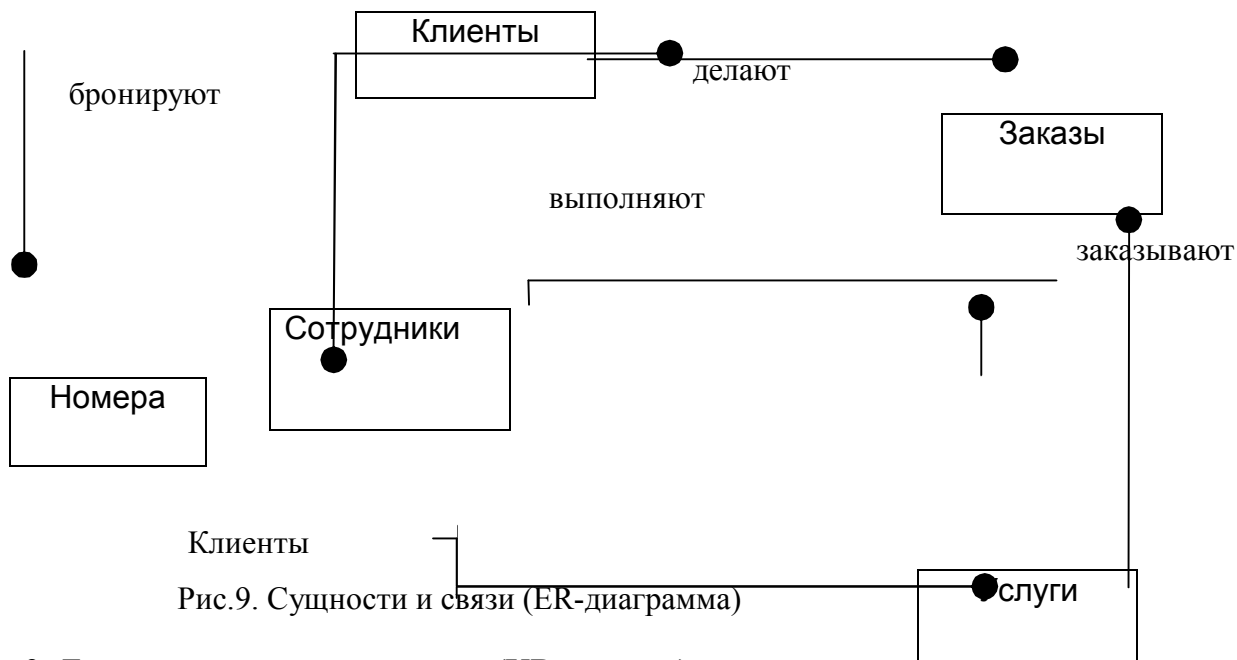
Таблица **Услуги** формируется следующим образом: код услуги, наименование, стоимость, описание.

Таблица **Сотрудники** содержит сведения: код сотрудника, фамилия, имя, дата рождения, дата найма, адрес, телефон, кому подчиняется.

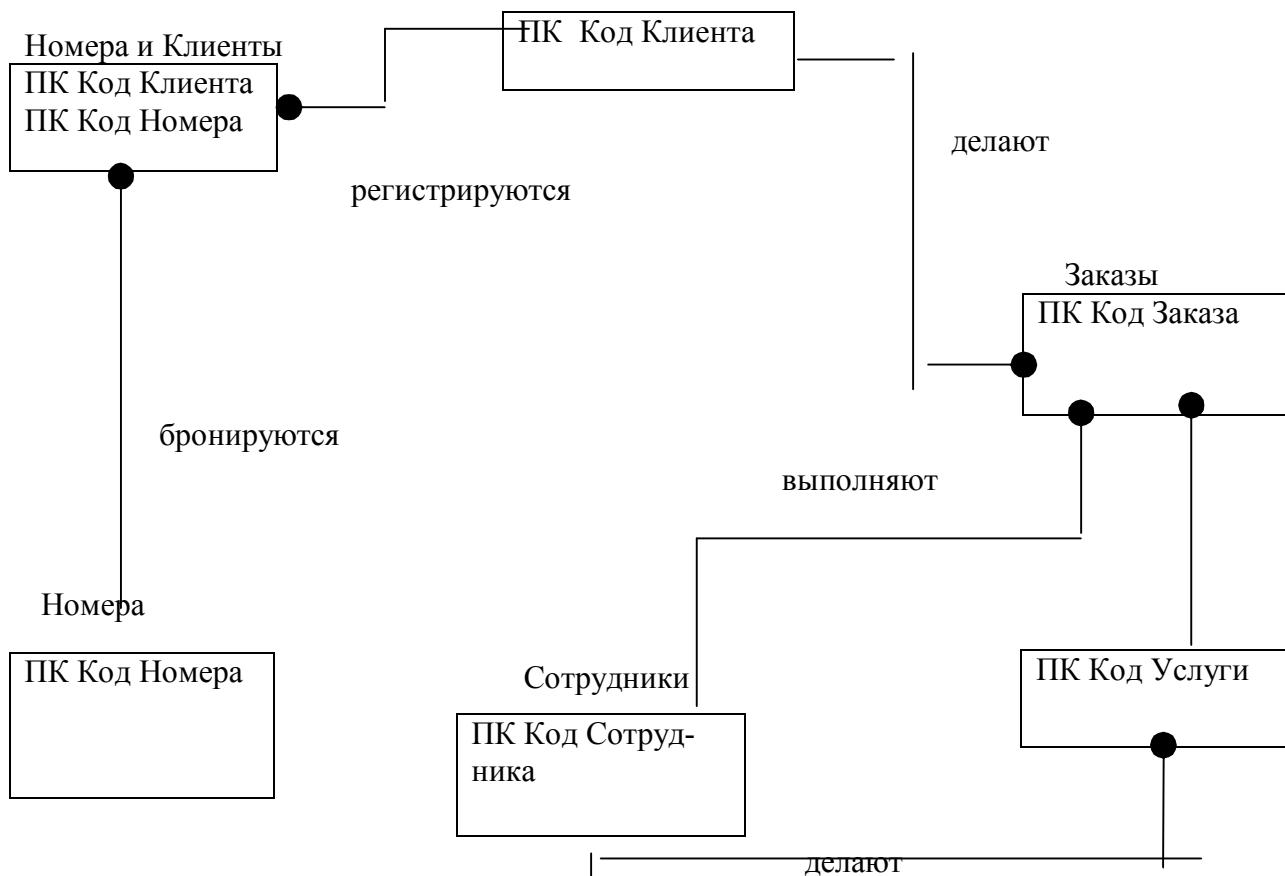
**Задание.** В каталоге Student\Колесникова\БД\Курсовой проект находятся варианты с заданной тематикой для проектирования базы данных.



## 1. Сущности и связи(ER-уровень)



## 2. Логика взаимосвязей данных (КВ-уровень)



### 3. Состав атрибутов сущностей (ФА-уровень)

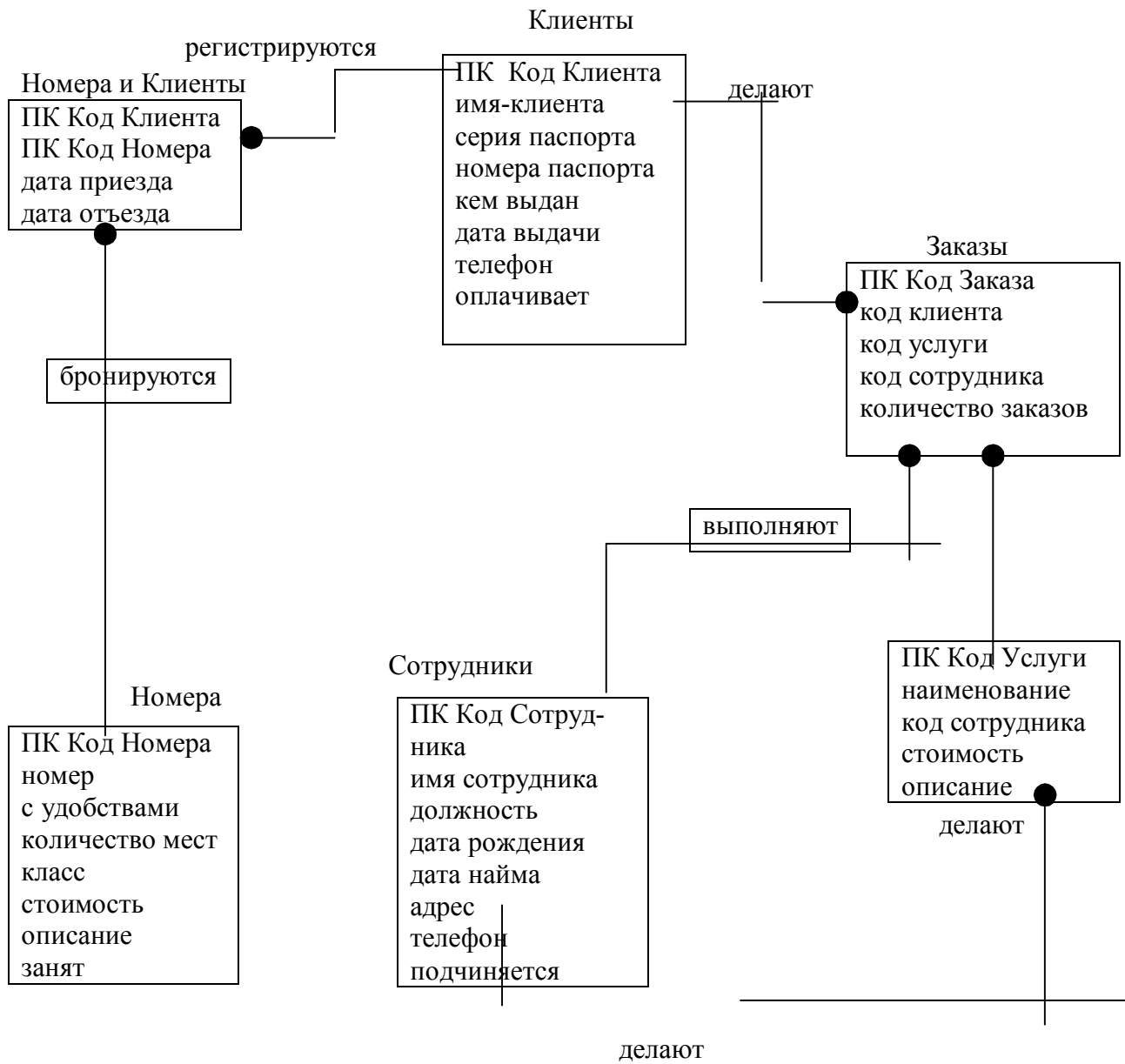


Рис.11. Состав атрибутов сущностей

#### 4. Глоссарий модели.

Таблица 6. **Сущности**

Имя	Определение
КЛИЕНТ	Физическое или юридическое лицо, которое бронируют номер в гостинице.
ЗАКАЗ	Зарегистрированный факт оплаченного заказа.
НОМЕРА	Имеющиеся номера в гостинице.
СОТРУДНИКИ	Физическое лицо, работающее в гостинице по договору найма.
УСЛУГИ	Перечень услуг в номер, представляемой гостиницей клиентам.

Таблица 7. **Атрибуты**

Имя	Домен	Определение	Владелец
Код-клиента	коды	Уникальный код клиента	КЛИЕНТ
Серия паспорта-клиента	числовые	Серия паспорта Клиента	КЛИЕНТ
имя-клиента	имена	Фамилия, имя и отчество Клиента наименование организации Клиента.	КЛИЕНТ
телефон-клиента	телефоны	Полный номер телефона Клиента	КЛИЕНТ
Номер паспорта-клиента	числовые	Номера паспорта Клиента	КЛИЕНТ
Кем выдан паспорт-клиента	имена	Наименование учреждение, выдавшее паспорт Клиенту.	КЛИЕНТ
Дата выдачи паспорта-клиента	даты	Дата выдачи паспорта Клиенту.	КЛИЕНТ
Оплачивает-клиента	Истина-ложь	Истина-клиент оплачивает номер, ложь- не оплачивает	КЛИЕНТ
Дата прибытия-клиента	даты	Дата прибытия клиента в номер	НОМЕРА и КЛИЕНТЫ
Дата отъезда-клиента	даты	Дата отъезда клиента из номера	НОМЕРА и КЛИЕНТЫ
Код-номера	коды	Уникальный код номера гостиницы.	НОМЕРА
Номер-номера	числовые	Номер гостиницы	НОМЕРА
С удобствами- номера	истина-ложь	Истина- номер с удобствами, ложь – без удобств.	НОМЕРА
Класс- номера	имена	Соответствующий класс номера.	НОМЕРА
Количество мест- номера	Числовые	Количество мест в номере.	НОМЕРА
Стоимость -номера	Цены	Стоимость номера в сутки	НОМЕРА
Описание - номера	имена	Краткое описание но-	НОМЕРА

		мера	
Занят - номер	Истина-ложь	Истина- номер занят, ложь- номер свободен.	НОМЕРА
Код-услуги	коды	Уникальный код услуги	УСЛУГИ
Наименование-услуги	имена	Наименование услуги	УСЛУГИ
Стоимость- услуги	цены	Стоимость определенной услуги.	УСЛУГИ
Описание - услуги	имена	Краткое описание определенной услуги.	УСЛУГИ
Код -заказа	коды	Уникальный код заказа	ЗАКАЗЫ
Количество - заказов	числовые	Количество заказов у клиента	ЗАКАЗЫ
Код - сотрудника	коды	Уникальный код сотрудника.	СОТРУДНИКИ
имя- сотрудника	имена	Фамилия, имя и отчество СОТРУДНИКА	СОТРУДНИКИ
адрес-сотрудника	адреса	Полный адрес сотрудника.	СОТРУДНИКИ
телефон-сотрудника	телефоны	Полный номер телефона сотрудника	СОТРУДНИКИ
дата-найма сотрудника	даты	Дата начала действия договора найма между гостиницей и сотрудником	СОТРУДНИКИ
Должность сотрудника	имена	Должность, занимаемая сотрудником.	СОТРУДНИКИ
Дата рождения сотрудника	даты	Дата рождения сотрудника	СОТРУДНИКИ
Подчиняется	имена	Имя начальника для определенного сотрудника	СОТРУДНИКИ

Таблица 8. Домены.

Имя	Тип (<длина>)	Определение
Адреса	Текстовый(100)	<индекс>, <город>, <улица>, <дом>, [<квартира>]
Имена	Текстовый(200)	Последовательности букв русского алфавита, возможно, содержащие пробелы и дефисы
Коды	Счетчик	Последовательность чисел, начиная с 1
Цены	Денежный	Специальный числовой тип, интерпретируемый как <рубли>.<копейки>
Числовые	Числовые	Целые числа
Телефоны	Числовые	[(<код города>)] <номер телефона>
Даты	Дата/время	Специальный числовой тип, интерпретируемый как <день><месяц><год>
Истина-ложь	Логический (Да/Нет)	[Y] – занят номер, [ ] – не занят номер

## 5. Приложение. Типовые запросы к БД “Гостиница ХХХ”.

### 1. Получить информацию об оплате за номера клиентов.

```

SELECT Клиенты.КодКлиента, Клиенты.[Фамилия клиента], Клиенты.Имя, Клиен-
ты.Отчество, Номера.Номер, [Номера и Клиенты].[Дата отъезда]-[Номера и Клиен-
ты].[Дата прибытия] AS Срок, Номера.Стоимость, ([Номера и Клиенты].[Дата отъезда]-
[Номера и Клиенты].[Дата прибытия])*Номера.Стоимость AS [Оплата за номер]
FROM Номера INNER JOIN (Клиенты INNER JOIN [Номера и Клиенты] ON Клиен-
ты.КодКлиента = [Номера и Клиенты].КодКлиента) ON Номера.КодНомера = [Номера и
Клиенты].КодНомера
WHERE (((Клиенты.Оплачивает)=Yes))

```

### **2. Получить информацию об оплате за услуги клиентов.**

```

SELECT Заказы.[Код Клиента], Клиенты.[Фамилия клиента], Клиенты.Имя, Клиен-
ты.Отчество, Sum ([Дополнительные услуги]. Стоимость * Заказы. [Количество заказов])
AS Оплата
FROM Клиенты INNER JOIN ([Дополнительные услуги] INNER JOIN Заказы ON [Допол-
нительные услуги].[Код Услуги] = Заказы.[Код Услуги]) ON Клиенты.КодКлиента = Зака-
зы.[Код Клиента]
GROUP BY Заказы.[Код Клиента], Клиенты.[Фамилия клиента], Клиенты.Имя, Клиен-
ты.Отчество

```

### **3. Получить информацию о полном счете клиентов.**

```

SELECT [Оплата за номер].КодКлиента, [Оплата за номер].[Фамилия клиента], [Оплата за
номер].Имя, [Оплата за номер].Отчество, [Оплата за номер].[Оплата за номер]+[Счет за
услуги].Оплата AS [Полный счет]
FROM [Оплата за номер] INNER JOIN [Счет за услуги] ON [Оплата за номер].КодКлиента
= [Счет за услуги].[Код Клиента]
ORDER BY [Оплата за номер].КодКлиента

```

### **4. Получить информацию о свободных номерах.**

```

SELECT Номера.КодНомера, Номера.Номер, Номера.[С удобствами], Номера.Класс, Но-
мера.[Количество мест], Номера.Стоимость, Номера.Описание, Номера.Фото FROM Но-
мера
WHERE (((Номера.Занят)=No))

```

**Заключение.** В ходе курсовой работы были получены: а) знания работы с базами данных, сбором и обработкой информации б) навыки по разработке баз данных в) навыки работы на Visual Basic.

Разработанная база данных предназначена для сотрудников в сфере гостиничного бизнеса.

### **Литература**

1. Карпова Т.А. Базы данных: модели, разработка, реализация. – Санкт-Петербург, 2001.
2. Сибилев Д.С. Разработка концептуальной модели данных - Томск, 2002.
3. Базы данных. Учебное пособие/Красина Ф.А.- Томск: Томский межвузовский центр дистанционного образования, 2002.-114 с.