

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

ГУМАНИТАРНЫЙ КОЛЛЕДЖ КЫРГЫЗСКОГО ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА имени И. АРАБАЕВА

«Утверждаю»

Директор гуманитарного колледжа

КГУ им. И. Арабаева

Дуйшеналиев Ч. Д.

« \_\_\_\_\_ » 2022 г.



**ТИПОВАЯ ПРОГРАММА**

по дисциплине База данных

Разработчик (должность) преподаватель


Ф.И.О. Осмонова Назира Шамшыбековна

Заведующий отделением «Информатика и Дизайн»

Ф.И.О. Турганбаева Б. Т.

Принято на заседании отделения « \_\_\_\_\_ » 2022 г.

№ протокола \_\_\_\_\_

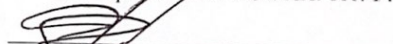
  
Подпись

Рекомендован

Пред. УМС Гум. Колледжа

КГУ им. И. Арабаева

ст. преп. Янгибаева Ж.Т.

  
« \_\_\_\_\_ » 2022 г.

г. Бишкек.

## Типовая программа

Дисциплина База данных

Направление (специальность) ПИ, ПОВ ТАС, АСУ

---

Форма обучения очная

Курс 2 Семестр 4

Часов: всего 65, лекции 22, практических занятий 14

СРС и виду индивидуальных занятий (курсовые работы) 29

Обеспечивающее отделение «Информатика и Дизайн»

## ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

**Целью учебной дисциплины** «Базы данных» является изучение теоретических основ построения и функционирования баз данных, характеристик современных СУБД, современных технологий организации БД и приобретение навыков работы в среде конкретных СУБД.

### Задачи учебной дисциплины

Задачами изучения данного курса является формирование у студента системы знаний о современных методах хранения и обработки информации и выработка у обучающихся устойчивых навыков их использования, что является неотъемлемым признаком высококлассного специалиста. Особое внимание уделяется вопросам автоматизации проектирования БД, практической работе (проектирование, ведение и использование баз данных) в среде выбранных целевых СУБД. Основной задачей дисциплины является формирование у студента четкого понимания о задачах и функциях проектирования баз данных.

### Требования к уровню освоения дисциплины

В результате изучения курса «Базы данных», студенты должны:

#### Иметь представление о:

- современных базах и банках знаний;
- принципах построения БД и БЗ;
- логическом и физическом устройстве современных БД;
- средствах защиты данных, перспективах развития БД.

#### Знать:

- основные понятия БД;
- основы построения и функционирования БД;
- технологии организации БД;
- язык создания и манипулирования данными SQL;
- способы защиты данных;
- приемы работы в распределенных и многопользовательских БД.

#### Уметь:

- построить модель предметной области;
- создать соответствующую модели базу данных в СУБД ACCESS и SQL SERVER;
- организовать ввод информации в базу данных и вывод отчетов;
- формулировать запросы к БД;
- организовать работу в многопользовательской БД.

## МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Изучение дисциплины «Базы данных» основывается на ранее изученных студентами предметах. Дисциплина «Базы данных» дает знания о применении и использовании современных информационных технологий; показывает особенности



технологии банков данных как одной из основных новых информационных технологий, с тем, чтобы студенты понимали тенденции развития современных информационных технологий, видели их преимущества и недостатки, особенности работы в условиях конкретных технологий в их профессиональной деятельности; ориентирует студентов во множестве современных СУБД и связанных с ними технологий; освещает теоретические и организационно-методические вопросы построения и функционирования систем, основанных на концепции баз данных, в том числе различные методологии моделирования и проектирования баз данных.

К началу изучению данной дисциплины студенты должны изучить дисциплины «Информатика и программирование», «Информационные системы». В свою очередь изучения курса «Базы данных» является базой для изучения профессиональных дисциплин.

Дисциплина включает лекционную часть, практические занятия под руководством преподавателя и самостоятельные занятия. Итоговой формой контроля является экзамен по всем темам учебной дисциплины. В качестве текущих форм контроля знаний студентов тестирование по всем темам.

#### Виды учебных занятий по дисциплине и их объёмы (в часах)

Вид учебной работы	Кол-во часов по формам обучения	
	Очная	Заочная
№№ семестров		
Аудиторные занятия		
Лекции	22	
Практические и семинарские занятия	14	
Самостоятельная работа	29	
<b>ВСЕГО ЧАСОВ НА ДИСЦИПЛИНУ</b>		
Текущий контроль (вид текущего контроля и кол-во, №№ семестров)	Тестирование	Тестирование
Виды промежуточного контроля (экзамен, зачет, №№ семестров)	Экзамен- 4сем	Экзамен

**ПРОГРАММА КУРСА ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ**

№	Названия разделов и тем	Виды учеб. занятий		СРС
		Ауд. занятия, в том числе		
		Лекции	Прак. сем. занятия	
	Введение. Основные понятия	2		2
	Введение в банки данных. OLAP и OLTP-системы	2		3
	Принципы построения БД. Инфологическое (концептуальное) моделирование предметной области.	2	1	3
	Теоретико-графовые модели данных. Реляционная модель данных.	2	2	3
	Табличные языки запросов	2	2	3
	Язык SQL.	2	2	3
	Целостность баз данных. Физические модели. Манипулирование данными.	2	2	3
	Организация ввода данных в базу данных. Вывод информации из баз данных. Разработка приложений	2	2	3
	Распределенные БД	4	2	3
	Безопасность данных. Перспективы развития БД и СУБД.	2	1	3
	<b>Всего:</b>	<b>22</b>	<b>14</b>	<b>29</b>

**Темы и планы лекционных занятий**

1. Основы теории баз данных.
  - 1.1. Основы базы данных.
  - 1.2. Функции базы данных.
2. Характеристики СУБД .
  - 2.1. Классы СУБД.
  - 2.2. Производительность СУБД.
  - 2.3. Модели данных в СУБД.
3. Поля базы данных.
  - 3.1. Основные понятия.

- 3.2. Основные типы полей баз данных.
- 3.3. Свойства полей базы данных.
- 4. Объекты СУБД. Таблица как объект СУБД
  - 4.1. Создание таблиц.
  - 4.2. Создание межтабличных связей.
- 5. Запросы как объекты СУБД.
  - 5.1. Работа с запросами.
  - 5.2. Визуальные средства создания запросов.
- 6. Форма как объект СУБД.
  - 6.1. Работа с формами.
  - 6.2. Элементы управления формы.
  - 6.3. Дизайн формы.
- 7. Отчёт как объект СУБД.
  - 7.1. Создание отчетов.
  - 7.2. Редактирование отчетов.
- 8. Страница как объект СУБД.
  - 8.1. Создание страниц.
  - 8.2. Редактирование страниц.

#### **Темы и планы практических занятий.**

- 1. Создание базы данных.
  - 1.1. Создайте новую базу данных.
  - 1.2. Создайте таблицу базы данных.
  - 1.3. Определите поля таблицы в соответствии с табл. 1.
  - 1.4. Сохраните созданную таблицу.
- 2. Создать основные таблицы
  - 2.1. Создать вспомогательные (справочные) таблицы.
  - 2.2. Схема данных (отображение связей между таблицами).
- 3. Разработать формы и заполнить таблицы данными
  - 3.1. Формы разных типов (ленточная, табличная, в столбец);
  - 3.2. Подчиненные формы;
  - 3.3. Многотабличные формы;
  - 3.4. Использовать кнопки для перехода из форм по вводу основных таблиц данных в таблицы-справочники.
- 4. Разработать и выполнить запросы к базе данных:
  - 4.1. Простые;
  - 4.2. Итоговые;
  - 4.3. Перекрестные;
  - 4.4. С использованием параметров;
  - 4.5. Создание вычисляемых полей в запросах;
  - 4.6. Запрос на поиск повторяющихся записей;
  - 4.7. Запрос на поиск записей, не имеющих подчиненных.
- 5. Создать отчеты:
  - 5.1. Простой;
  - 5.2. Подчиненный.



## **Тексты лекций.**

### **1. Основные понятия**

Одним из важнейших условий обеспечения эффективного функционирования любой организации является наличие развитой информационной системы и системы управления базой данных.

Современной формой информационных систем являются банки данных, которые включают в свой состав вычислительную систему, одну или несколько баз данных (БД), систему управления базами данных (СУБД) и набор прикладных программ (ПП).

**Основными функциями банков данных являются:**

- хранение данных и их защита.
- изменение (обновление, добавление и удаление) хранимых данных.
- поиск и отбор данных по запросам пользователей.
- обработка данных и вывод результатов.

База данных обеспечивает хранение информации и представляет собой поименованную совокупность данных, организованных по определенным правилам, включающим общие принципы описания, хранения и манипулирования данными.

С понятием базы данных тесно связано понятие система управления базой данных.

Система управления базами данных представляет собой пакет прикладных программ и совокупность языковых средств, предназначенных для создания, сопровождения и использования баз данных.

Прикладные программы (приложения) в составе банков данных служат для обработки данных, вычислений и формирования выходных документов по заданной форме.

Большинство систем управления базами данных (СУБД) позволяют размещать в своих структурах не только данные, но и методы (то есть программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами. Таким образом, можно говорить о том, что в современных базах данных хранятся отнюдь не только данные, но и информация.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий.

### **2. Характеристики СУБД**

#### **Классы СУБД**

По степени универсальности различают два класса СУБД:

- системы общего назначения СУБД общего значения - это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.
- специализированные системы. Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего значения.

#### **Производительность СУБД**

### **Производительность СУБД оценивается:**

- временем выполнения запросов.
- скоростью поиска информации в неиндексированных полях.
- временем выполнения операций импортирования базы данных из других форматов.
- скоростью создания индексов и выполнения таких массовых операций, как обновление, вставка, удаление данных.
- максимальным числом параллельных обращений к данным в многопользовательском режиме.
- временем генерации отчета.

### **На производительность СУБД оказывают влияние два фактора:**

СУБД, которые следят за соблюдением целостности данных, несут дополнительную нагрузку, которую не испытывают другие программы; производительность собственных прикладных программ сильно зависит от правильного проектирования и построения базы данных.

СУБД, как правило, разделяют по используемой модели данных (как и базы данных) на следующие типы: иерархические, сетевые, реляционные и объектно-ориентированные.

### **Модели данных в СУБД**

Хранимые в базе данные имеют определенную логическую структуру, то есть представлены некоторой моделью, поддерживаемой СУБД.

#### **К числу важнейших относятся следующие модели данных:**

- иерархическая.
- сетевая.
- реляционная.
- объектно-ориентированная.

В **иерархической** модели данные представляются в виде древовидной (иерархической) структуры. Она удобна для работы с иерархически упорядоченной информацией и громоздка для информации со сложными логическими связями.

**Сетевая** модель означает представление данных в виде произвольного графа. Достоинством сетевой и иерархической моделей данных является возможность их эффективной реализации по показателям затрат памяти и оперативности. Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе.

**Реляционная** модель данных (РМД) название получила от английского термина relation — отношение. При соблюдении определенных условий отношение представляется в виде двумерной таблицы, привычной для человека. Большинство современных БД для персональных ЭВМ являются реляционными.

Достоинствами реляционной модели данных являются ее простота, удобство реализации на ЭВМ, наличие теоретического обоснования и возможность формирования гибкой схемы БД, допускающей настройку при формировании запросов.



Реляционная модель данных используется в основном в БД среднего размера. При увеличении числа таблиц в базе данных заметно падает скорость работы с ней. Определенные проблемы использования РМД возникают при создании систем со сложными структурами данных, например, систем автоматизации проектирования.

**Объектно - ориентированные БД** объединяют в себе две модели данных, реляционную и сетевую, и используются для создания крупных БД со сложными структурами данных.

**По характеру использования СУБД делят на:**

- персональные (СУБДП)
- многопользовательские (СУБДМ)

К **персональным СУБД** относятся Visual FoxPro, Paradox, Clipper, dBase, Access и др. К **многопользовательским СУБД** относятся, например, СУБД Oracle и Informix. Многопользовательские СУБД включают в себя сервер БД и клиентскую часть, работают в неоднородной вычислительной среде — допускаются разные типы ЭВМ и различные операционные системы. Поэтому на базе СУБДМ можно создать информационную систему, функционирующую по технологии клиент-сервер. Универсальность многопользовательских СУБД отражается соответственно на высокой цене и компьютерных ресурсах, требуемых для их поддержки.

**СУБДП** представляет собой совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД. Персональные СУБД обеспечивают возможность создания персональных БД и недорогих приложений, работающих с ними, и при необходимости создания приложений, работающих с сервером БД.

### **Язык современной СУБДП**

**Язык современной СУБДП** включает подмножества команд, относившиеся ранее к следующим специализированным языкам:

- язык описания данных — высокоуровневый непроцедурный язык декларативного типа, предназначенный для описания логической структуры данных.
- язык манипулирования данными — командный язык СУБД, обеспечивающий выполнение основных операций по работе с данными — ввод, модификацию и выборку данных по запросам.
- структурированный язык запросов (Structured Query Language, SQL) — обеспечивает манипулирование данными и определение схемы реляционной БДП, является стандартным средством доступа к серверу БД.

Обеспечение целостности БД — необходимое условие успешного функционирования БД. Целостность БД — свойство БД, означающее, что база данных содержит полную и непротиворечивую информацию, необходимую и достаточную для корректного функционирования приложений. Обеспечение безопасности достигается в СУБД шифрованием прикладных программ, данных, защиты паролем, поддержкой уровней доступа к отдельной таблице.

### **3. Поля базы данных**

### Основные понятия

**Поле** - наименьший поименованный элемент информации, хранящейся в БД и рассматриваемой как единое целое.

Поле может быть представлено числом, буквами или их сочетанием (текстом). Например, в телефонном справочнике полями являются фамилия и инициалы, адрес, номер телефона, т.е. три поля, причем все текстовые (номер телефона также рассматривается как некоторый текст).

Поле1	Поле2	...	Поле К-1	Поле К	
					Запись 1
					Запись 2
					...
					Запись N

Рисунок 1. Представление информации в БД

**Запись** - совокупность полей, соответствующих одному объекту. Так, абоненту телефонной сети соответствует запись, состоящая из трех полей.

**Файл** - совокупность связанных по какому-либо признаку записей (т.е. отношение, таблица). Таким образом, в простом случае база данных есть файл.

### Типы полей

Все данные в БД разделены по типам. Вся информация полей, принадлежащих одному столбцу (домену), имеет один и тот же тип. Такой подход позволяет ЭВМ организовать контроль вводимой информации.

Имя поля	Тип данных
услуги рекламного агентства	Текстовый
	Текстовый
	Поле MEMO
	Числовой
	Дата/время
	Денежный
	Счетчик
	Логический
	Поле объекта OLE
	Гиперссылка
	Мастер подстановок...

Рисунок 2

### Основные типы полей баз данных:

- Символьный (текстовый). В таком поле по умолчанию может храниться до 256 символов.



- **Числовой** . Содержит числовые данные различных форматов, используемые для проведения расчетов.
- **Дата / время** . Содержит значение даты и времени.
- **Денежный** . Включает денежные значения и числовые данные до пятнадцати знаков целой части и четырех знаков дробной части.
- **Поле примечание** . Оно может содержать до  $2^{16}$  символов ( $2^{16} = 65536$ ).
- **Счетчик** . Специальное числовое поле, в котором СУБД присваивает уникальный номер каждой записи.
- **Логический** . Может хранить одно из двух значений: true or false.
- **Поле объекта OLE (Object Linking and Embedding** - технология вставки и связывания объекта). Это поле может содержать любой объект электронной таблицы, документ microsoft word, рисунок, звукозапись или другие данные в двоичном формате, внедренные или связанные с СУБД.
- **Гиперссылка** . Может содержать строку, состоящую из букв и цифр, представляющую адрес сайта или web - страницы.
- **Мастер подстановок** . Создает поле, в котором предлагается выбор значений из списка или содержащего набор постоянных значений.

#### **Свойства полей базы данных**

Поля базы данных не просто определяют структуру базы — они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей.

**Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access:**

- **Имя поля** — определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).
- **Тип поля** — определяет тип данных, которые могут содержаться в данном поле.
- **Размер поля** — определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.
- **Формат поля** — определяет способ форматирования данных в ячейках, принадлежащих полю.
- **Маска ввода** — определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).
- **Подпись** — определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).
- **Значение по умолчанию** — то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).
- **Условие на значение** — ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

- **Сообщение об ошибке** — текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство Условие на значение).
- **Обязательное поле** — свойство, определяющее обязательность заполнения данного поля при наполнении базы.
- **Пустые строки** — свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).
- **Индексированное поле** — если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видеоклипов и других объектов OLE, большинство вышеуказанных свойств не имеют смысла.

#### 4. Объекты СУБД

СУБД имеет свою собственную структуру и состоит из основных объектов, используемых при ее создании и работе с ней.

#### 5. Таблица как объект СУБД

##### Создание таблиц

- Работа с любыми объектами начинается с окна База данных (рисунок 3). На левой панели данного окна сосредоточены элементы управления для вызова всех семи типов объектов программы. Создание таблиц начинается с выбора элемента управления Таблицы.



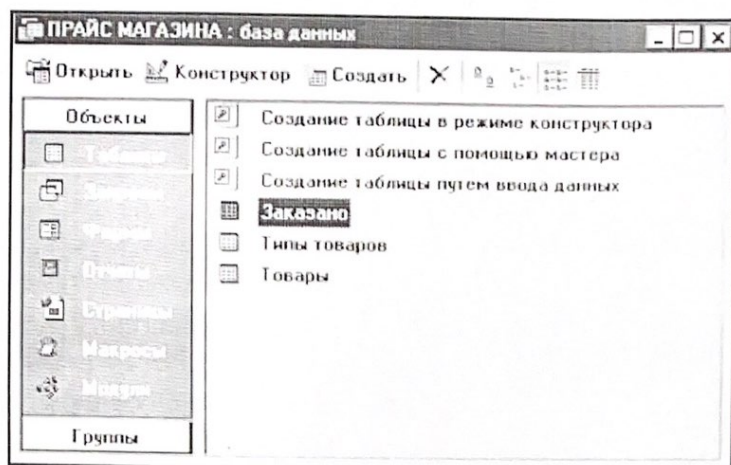


Рисунок 3

- На правой панели представлен список таблиц, уже имеющихся в составе базы, и приведены элементы управления для создания новой таблицы. Чтобы создать таблицу вручную, следует использовать значок Создание таблицы в режиме конструктора.
- При создании таблицы целесообразно (хотя и не обязательно) задать ключевое поле. Это поможет впоследствии, при организации связей между таблицами. Для задания ключевого поля достаточно щелкнуть на его имени правой кнопкой мыши и в открывшемся контекстном меню выбрать пункт Ключевое поле.
- Закончив создание структуры таблицы, бланк закрывают (при этом система выдает запрос на сохранение таблицы), после чего дают таблице имя, и с этого момента она доступна в числе прочих таблиц в основном окне База данных. Оттуда ее и можно открыть в случае необходимости.
- Созданную таблицу открывают в окне База данных двойным щелчком на ее значке. Новая таблица не имеет записей — только названия столбцов, характеризующие структуру таблицы (рисунок 4). Заполнение таблицы данными производится обычным порядком. Курсор ввода устанавливается в нужную ячейку указателем мыши. Переход к следующей ячейке можно выполнить клавишей TAB. Переход к очередной записи выполняется после заполнения последней ячейки.

Исходные данные : таблица			
	Код газеты	Наименование издания	Стоимость размещения рекламы
	1	Ростов	\$10,00
	2	Ва-Банк	\$20,00
	3	Реклама-Юг	\$12,00
	4	Комсомольская правда	\$9,00
	5	Сова	\$16,00
	6	Телеком	\$13,00
*	0		\$0,00

Рисунок 4

В нижней части таблицы расположена Панель кнопок перехода. Ее элементами управления удобно пользоваться при навигации по таблице, имеющей большое число записей.

Начинающим пользователям Microsoft Access доставляет неудобство тот факт, что данные не всегда уместаются в ячейках таблицы. Шириной столбцов можно управлять методом перетаскивания их границ. Удобно использовать автоматическое форматирование столбцов «по содержимому». Для этого надо установить указатель мыши на границу между столбцами (в строке заголовков столбцов), дождаться, когда указатель сменит форму, и выполнить двойной щелчок. Это общесистемный прием Windows 98, и им можно пользоваться в данной программе, как и во многих других.

После наполнения таблицы данными сохранять их не надо — все сохраняется автоматически. Однако если при работе с таблицей произошло редактирование ее макета (например, изменялась ширина столбцов), СУБД попросит подтвердить сохранение этих изменений.

Если возникнет необходимость изменить структуру таблицы (состав полей или их свойства), таблицу надо открыть в режиме Конструктора. Для этого ее следует выделить в окне База данных и щелкнуть на кнопке Конструктор.

Если на этапе проектирования базы данных была четко разработана структура таблиц, то создание таблиц с помощью Конструктора происходит очень быстро и эффективно. Даже без использования автоматизированных средств создание основы для достаточно крупных проектов происходит в считанные минуты — это ценное свойство СУБД Microsoft Access, но оно реализуется при обязательном условии тщательной предварительной подготовки.

### **Создание межтабличных связей**

Если структура базы данных продумана заранее, а связи между таблицами намечены, то создание реляционных отношений между таблицами выполняется очень просто. Вся необходимая работа происходит в специальном окне Схема данных и выполняется с помощью мыши. Окно Схема данных открывают кнопкой на панели инструментов или командой Сервис --> Схема данных (если в меню Сервис не видно соответствующего пункта, следует раскрыть расширенную часть меню).

Образовавшаяся межтабличная связь отображается в окне Схема данных в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается главной, а другая — связанной. Главная — это та таблица, которая участвует в связи своим ключевым полем (название этого поля на схеме данных отображается полужирным шрифтом).

Здесь мы подходим к важному вопросу: «А зачем вообще нужна связь между таблицами?» У связи два основных назначения. Первое — обеспечение целостности данных, а второе — автоматизация задач обслуживания базы. Представим себе, что в таблице Клиенты, где каждый клиент уникален, кто-то удалит запись для одного из клиентов, но не сделает этого в таблице Заказы. Получится, что согласно таблице Заказы, некто, не имеющий ни имени, ни адреса, а только абстрактный код, делал заказы. Узнать по коду, кто же это был на самом деле, будет невозможно — произошло нарушение целостности данных.

В данном случае владелец базы может применить три подхода: либо вообще ничего не делать для защиты целостности данных, либо запретить удаление данных из ключевых полей главных таблиц, либо разрешить его, но при этом адекватно обработать и



связанные таблицы. Вручную сделать это чрезвычайно трудно, поэтому и нужны средства автоматизации.

**Связь между таблицами позволяет:**

- либо исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц.
- либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Для настройки свойств - связи надо в окне Схема данных выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи, после чего выбрать в нем пункт - Изменить связь — откроется диалоговое окно Изменение связи. В нем показаны названия связанных таблиц, имена полей, участвующих в связи (здесь же их можно изменить), а также приведены элементы управления для обеспечения условий целостности данных.

Если установлен только флажок Обеспечение целостности данных, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены флажки Каскадное обновление связанных полей и Каскадное удаление связанных записей, то, соответственно, операции редактирования и удаления данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой стороны — в автоматизации внесения изменений сразу в несколько таблиц, при изменениях в одной таблице.

## **6. Запросы как объекты СУБД**

В целом все запросы решают проблему обновления базы данных. Существуют различные формы запросов. Выбор формы запросов зависит от решаемой рекламной задачи, от системы организации системы базы данных, а так же от пристрастий пользователя. В любом случае пользователь получает из базы данных информацию, требуемую в данный момент времени.

### **Работа с запросами**

**Запросы представляют собой набор выборочной информации из общей базы данных.**

Если структура базы данных предприятия хорошо продумано, то исполнители, работающие с базой, должны навсегда забыть о том, что в базе есть таблицы, а ещё лучше, если они об этом вообще ничего не знают. Таблицы – слишком ценные объекты базы, чтобы с ними имел дело кто-либо, кроме разработчика базы.

Если исполнителю надо получить данные из базы, он должен использовать специальные объекты – запросы. Все необходимые запросы разработчик базы должен подготовить заранее. Если запрос подготовлен, надо открыть панель Запросы в окне базы данных, выбрать его и открыть двойным щелчком на значке – откроется результирующая таблица, в которой исполнитель найдёт то, что его интересует.

В общем случае результирующая таблица может не соответствовать ни одной из базовых таблиц базы данных. Её поля могут представлять набор из полей разных таблиц, а её записи могут содержать отфильтрованные и отсортированные записи таблиц, на основе которых формировался запрос. Лишь в тех случаях, когда исполнитель не находит нужных данных в результирующей таблице, возникает необходимость готовить новый запрос – это задача разработчика базы.

#### Управление отображением данных в результирующей таблице.

В нижней части бланка запроса по образцу имеется строка Вывод на экран . По умолчанию предполагается, что все поля, включённые в запрос, должны выводиться на экран, но это не всегда целесообразно. Например, бывают случаи, когда некое поле необходимо включить в запрос, на пример потому, что оно является полем сортировки, но в то же время, не желательно, чтобы пользователь базы видел его содержание. В таких случаях отображения содержимого на экране подавляют сбросом флажка Вывод на экран.

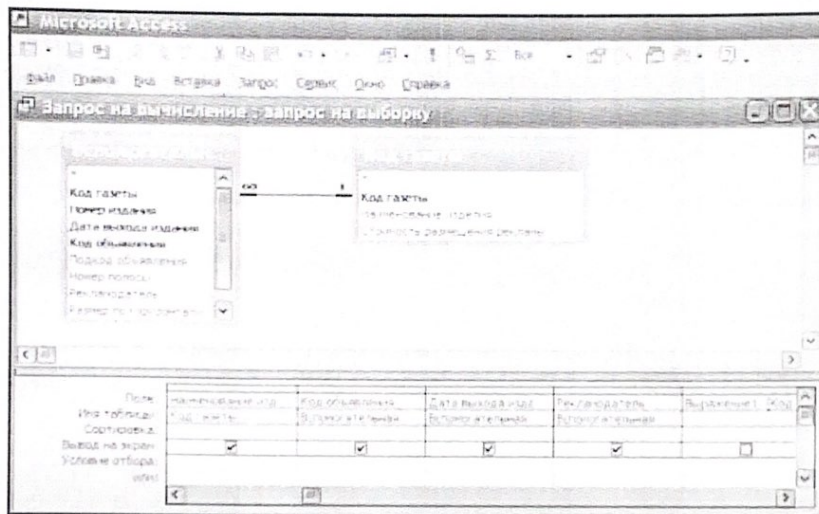
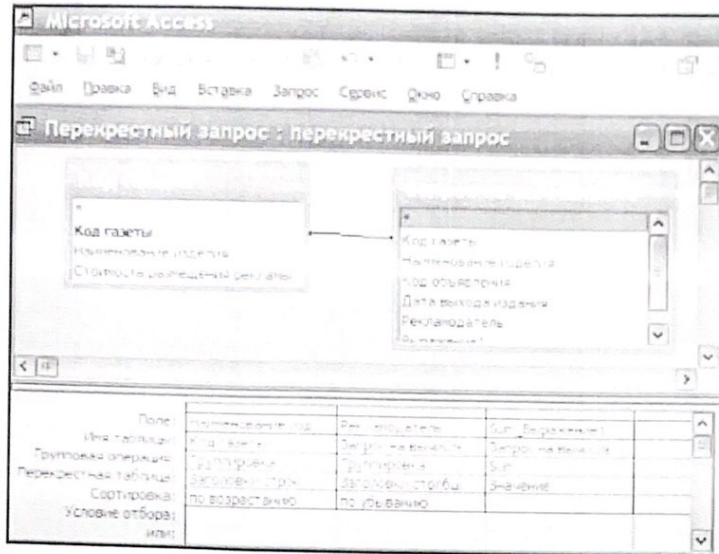


Рисунок 5

#### Упорядочение записей в результирующей таблице.

Если необходимо, чтобы данные, отобранные в результате работы запроса по образцу, были упорядочены по какому-либо полю, применяется сортировка. В нижней части бланка имеется специальная строка Сортировка . При щелчке на этой строке открывается кнопка раскрывающегося списка, в котором можно выбрать метод сортировки: по возрастанию или по убыванию. В результирующей таблице данные будут отсортированные по тому полю, для которого задан порядок сортировки.





**Рисунок 6.** В приведённом примере в поле Наименование изделия сортировка будет производиться по возрастанию, а в поле Рекламодатель – по убыванию.

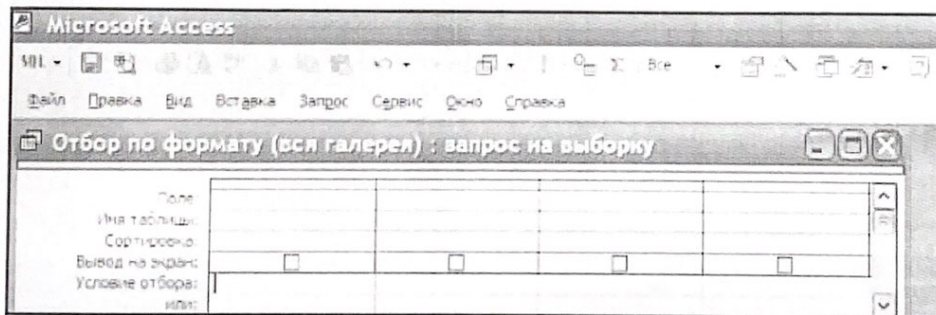
Возможна многоуровневая сортировка – сразу по нескольким полям. В этом случае данные сначала сортируются по тому полю, которое в бланке запроса по образцу находится левее, за тем по следующему полю, для которого включена сортировка, и так далее слева направо. Соответственно, при формировании запроса надо располагать поля результирующей таблицы не как попало, а с учётом будущей сортировки. В крайнем случае, если запрос уже сформирован, и надо изменить порядок следования столбцов, пользуются следующим приёмом:

Выделяют столбец щелчком на его заголовке (кнопку мыши отпускают).

Ещё раз щёлкают на заголовке уже выделенного столбца (но кнопку не отпускают) Перетаскивают столбец уже в другое место.

#### **Использование условия отбора.**

Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является так называемое Условие отбора . Соответствующая строка имеется в нижней части бланка запроса по образцу.



**Рисунок 7**

**Вычлняемые поля в запросах.**

Допустимо использование вычисляемых полей. Для этого применяется «построитель выражений». В этой структуре используются следующие обозначения:

Sum – определяет сумму значений числового поля.

Count – определяет количество группируемых записей.

Min и Max – определяют минимальное и максимальное значение поля.

First – выбирает первое значение.

#### Визуальные средства создания запросов.

Определяющим моментом формированием запросов является связь между таблицами. Связи в информационных технологиях показывают, какие поля, из каких таблиц связаны и каким образом. Синонимом слова Связь является слово Отношение.

В системе управления базы данных рассматриваются три типа отношений:

- «Один-к-одному». При данном типе отношения в запросе по образцу происходит объединение только совпадающих записей.

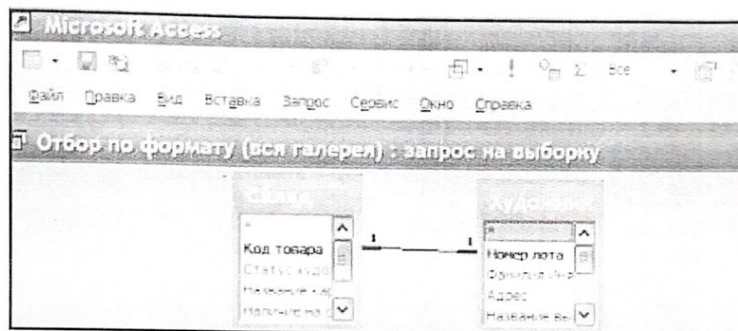


Рисунок 8

- «Один-ко-многим». При данном типе отношения происходит объединение всех записей из одной таблицы, с совпадающими записями из другой.

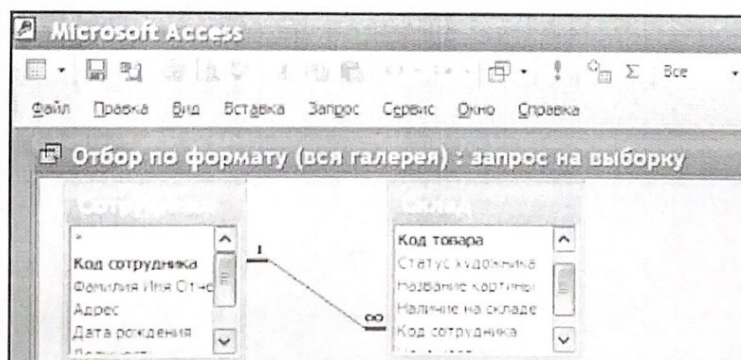


Рисунок 9

- «Многие-ко-многим».

#### Особенности запросов



Особенность запроса состоит в том, что по запросу выбираются данные из базовых таблиц, и на их основе создаются временные результирующие таблицы.

Эти временные таблицы находятся в оперативной памяти. При запросе пользователь имеет дело с отражением требуемых полей оперативной памяти, потому что работы с запросами происходит гораздо эффективнее, так как время доступа к базе данных, хранящейся на винчестере гораздо больше времени доступа к запросам в оперативной памяти.

В запрос заносятся только требуемые поля из общей таблицы, поэтому просмотр результата в значительной мере упрощается.

#### **Безопасность хранения данных в базе данных.**

При обращении к винчестеру могут возникнуть ситуации, приводящие к разрушению структуры базы данных:

- Неопытность пользователя.
- Хакерство.
- Выключение электричества.
- Повреждение вычислительной системы.
- Повреждение программного обеспечения (вирус).

### **7. Виды запросов, применяемых в рекламной практике**

#### **Запрос по образцу**

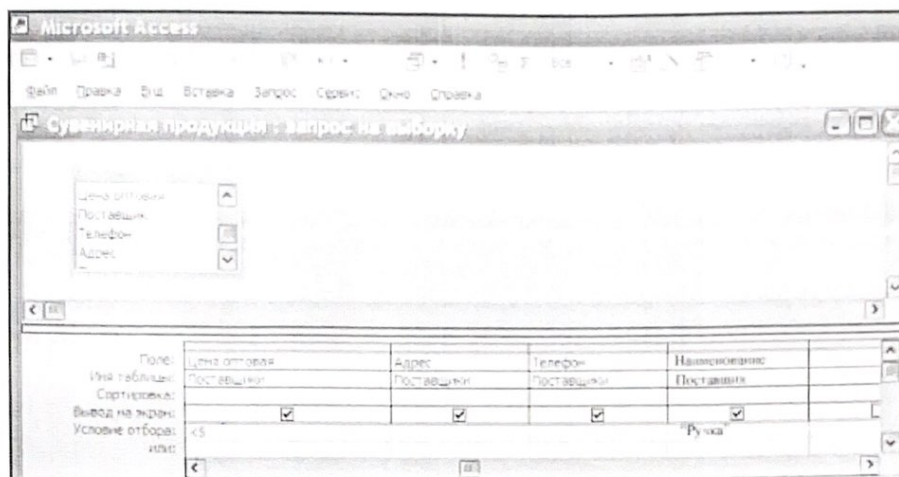
Этот запрос может быть разработан с помощью специального языка SQL (Structured Query Language). Однако пользователям MS Access изучать данный язык программирования необязательно, а большинство операций можно выполнить щелчками кнопок мыши и приёмом перетаскивания в бланке, так как пакет MS Access имеет визуальные средства разработки запроса по образцу.

Пример:

Рекламному агентству по производству сувенирной продукции необходимо закупить партию ручек для нанесения логотипа фирмы. Стоимость данных ручек не должна превышать пять рублей за единицу продукции.

#### **Технология разработки запроса по образцу в MS Access:**

- Запуск программы MS Access (Пуск-->Программы-->MS Access).
- В окне MS Access включите переключатель открыть базу данных, выберете ранее созданную базу Сувенирная продукция и щёлкните на кнопку ОК.
- В окне Сувенирная продукция: База данных откройте панель Запросы. Дважды щёлкните на значке Создание запроса в режиме Конструктора - откроется бланк запроса по образцу.
- Добавьте в созданный запрос таблицу Поставщики.
- В строке Условие отбора поле Наименование введите: Ручка.
- Строку Условие отбора для поля Цена оптовая надо заполнить таким образом, чтобы при запуске запроса выдавались данные о ручках стоимостью до пяти рублей. Для этого необходимо в строке Условие отбора для поля Цена оптовая написать: <=5



**Рисунок 10**

- Закройте запрос. При закрытии сохраните его под именем Отбор ручек стоимостью до пяти рублей.
- В окне Сувенирная продукция: База данных откройте панель Запросы и запустите запрос Отбор ручек стоимостью до пяти рублей. На экране появится результирующая таблица, которая содержит информацию о ручках стоимостью до пяти рублей.
- Закройте все объекты базы данных. Закройте программу MS Access.

### **Запрос с параметрами**

Специальный тип запросов, называемый запросами с параметрами, позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приёмом обеспечивается гибкость работы с базой.

Пример:

Рекламному агентству по производству печатной рекламы необходимо выбрать информацию из ранее созданной базы данных о поставщиках цветных принтеров для выпуска новой партии рекламных буклетов.

### **Технология разработки запроса с параметрами MS Access:**

- Запуск программы MS Access (Пуск->Программы->MS Access).
- В окне MS Access включите переключатель открыть базу данных, выберите ранее созданную базу печатная техника и щёлкните на кнопку ОК.
- В окне Печатная техника: База данных откройте панель Запросы. Дважды щёлкните на значке Создание запроса в режиме Конструктора - откроется бланк запроса по образцу.
- Добавьте таблицу Поставщики, из которой выберите поля: Компонент, Модель, Цена оптовая, Поставщик, Телефон.
- В строке Условие отбора поля Компонент введите: Принтер.
- Строку Условие отбора для поля Цена оптовая надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение. Текст, обращённый к пользователю, должен быть заключён в квадратные скобки. Если бы



хотели отобразить принтеры, цена которых более 200 у.е., мы бы написали: >200. если бы нам нужны были принтеры дешевле 200 у.е., мы бы написали: <200. но если мы хотим дать пользователю возможность выбора, мы должны написать «Введите цену».

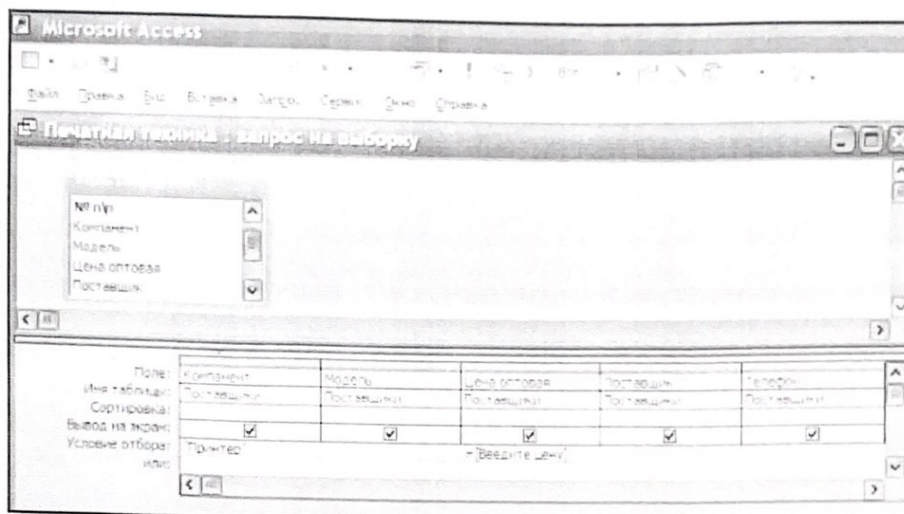


Рисунок 11

- Закройте запрос. При закрытии сохраните его под именем Отбор принтера.
- В окне Печатная техника: База данных откройте панель Запросы и запустите запрос Отбор принтера – на экране появится диалоговое окно Введите значение параметра.
- Введите какое-либо число и щёлкните по кнопке ОК. В зависимости оттого, что реально содержится в таблице Поставщики, по результатам запросам будет, сформирована результирующая таблица.

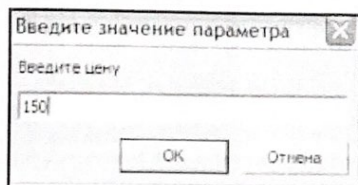


Рисунок 12

- Закройте все объекты базы данных. Закройте программу MS Access.

### Итоговый запрос

Запрос, выполняющий вычисление по всем записям для какого-либо числового поля, называются итоговыми запросами. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле, может, так же исполняться иная итоговая функция. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Пример:

Полнопрофильному рекламному агентству необходимо определить наиболее востребованный вид услуг за прошедший период времени, для того чтобы выявить сильные и слабые стороны своей деятельности. Рекламное агентство оказывает следующие виды услуг:

- Создание видеороликов.
- Создание аудиороликов.
- Наружная реклама.
- Неоновая реклама.
- Планирование и реализация рекламных кампаний.
- Реклама в прессе.
- Реклама на транспорте.
- Изготовление сувенирной продукции с нанесением логотипа фирмы.

#### **Технология создания итогового запроса в MS Access:**

- Запустите программу MS Access (Пуск-->Программы -->MS Access).
- В окне MS Access включите переключатель открыть базу данных, выберите ранее созданную базу Рекламное агентство и щёлкните на кнопку ОК.
- Щёлкните откройте панель Запросы щёлчком на одноимённой кнопке окна Рекламное агентство: База данных.
- Выполните двойной щелчок на значке Создание запросов в режиме конструктора. В открывшемся диалоговом окне Добавление таблицы выберите таблицу Виды услуг и заказы, на основе которой будет разрабатываться итоговый запрос. Закройте окно Добавление таблицы.
- В бланке запроса по образцу введите следующие поля таблицы Виды услуг и заказы: Виды услуг, Количество заказов, Общая прибыль.
- На панели инструментов MS Access щёлкните на кнопке Групповые операции или воспользуйтесь строкой меню (Вид-->Групповые операции). Эта команда необходима для создания в нижней части бланка строки Групповые операции. Именно на её базе создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение Группировка.
- Для поля, по которому производится группировка записей (в нашем случае – Вид услуг), оставьте в строке Групповые операции значения Группировка. Для остальных полей щёлкните в этой строке – появится кнопка раскрывающего списка, из которого можно выбрать итоговую функцию для расчёта значений в данном поле.
- Для поля Количество заказов выберите итоговую функцию Count, определяющую общее количество заказов на конкретный вид услуг.
- Для поля Общая прибыль выберите итоговую функцию Sum для определения общей прибыли по конкретному виду услуг.



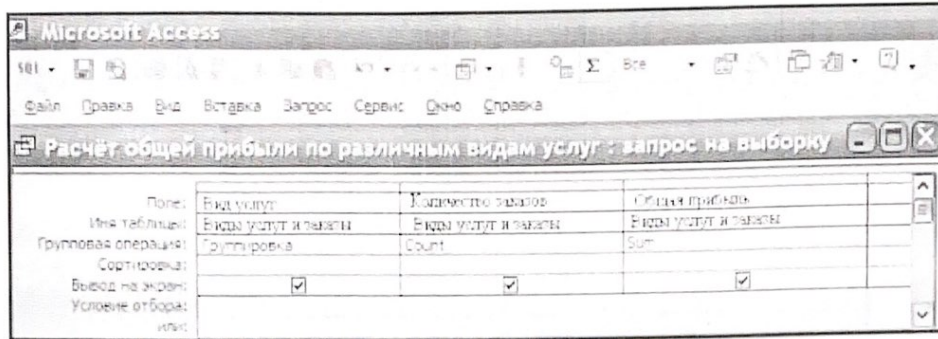


Рисунок 13

- Закройте бланк запроса по образцу и дайте ему имя Расчёт общей прибыли по различным видам услуг.
- Запустите запрос и убедитесь, что он правильно работает.

### Перекрёстный запрос

Он позволяет создать результирующие таблицы на основе результатов расчётов, полученных при анализе группы таблиц.

В перекрёстном запросе отображаются результаты расчетов (таких как суммы, количество записей и среднее значения), выполненных по данным из одного поля таблицы. Эти результаты группируются по двум наборам данных, один из которых расположен в левом столбце таблицы, а второй в верхней строке. Для примера, при создании перекрёстного запроса мы используем таблицу Код газеты и Запрос на вычисление. Запрос на вычисление был создан на основе таблиц Вспомогательная 1 и Код газеты.

Код газеты	Номер изд	Дата выхода	Код объявлен	Подкод объявл	Номер полосы	Рекламодатель	Размер по гориз	Размер по верт
1	1	01.01.2001	1	1	1	1 Sunrise	4	1
2	2	11.01.2001	1	1	1	2 Forte	2	2
3	3	10.01.2001	1	1	1	3 WVV	3	1
4	2	11.01.2001	1	1	1	2 Forte	2	2
5	1	01.01.2001	1	1	1	1 Sunrise	4	1
6	3	10.01.2001	1	1	1	3 WVV	4	1
7	1	01.01.2001	1	1	1	1 Sunrise	4	1
8	0		0	0	0	0	0	0

Microsoft Access

Код газеты : таблица

Код газеты	Наименование изделия	Стоимость размещения рекламы
1	Вечерний ростов	10р
2	Ва-Банк	20р
3	Реклама Юг	12р
4	Комсомольская правда (на Дону)	9р
5	Что Где Почем	16р
6	Телеком	13р

Записи: 14 из 7

Microsoft Access

Запрос на вычисление : запрос на выборку

Код газеты	Наименование изделия	Код объявления	Дата выхода	Рекламодатель	Выражение1
1	Вечерний ростов	1	01.01.2001	Sunrise	120.00р
1	Вечерний ростов	1	11.01.2001	Forte	40.00р
2	Ва-Банк	1	10.01.2001	VVV	180.00р
3	Реклама Юг	1	11.01.2001	Forte	40.00р
4	Комсомольская правда (на Дону)	1	01.01.2001	Sunrise	108.00р
5	Что Где Почем	1	10.01.2001	VVV	192.00р
6	Телеком	1	01.01.2001	Sunrise	156.00р

Записи: 14 из 8

Перекрестный запрос : перекрестный запрос

Код газеты: Наименование изделия, Стоимость размещения рекламы

Запрос на вычисление: Код объявления, Дата выхода издания, Рекламодатель, Выражение1

Имя таблицы:	Наименование изд.	Рекламодатель	Sum_Выражение1:
Код газеты	Вечерний ростов	Sunrise	120
Код газеты	Вечерний ростов	Forte	40
Код газеты	Ва-Банк	VVV	180
Код газеты	Реклама Юг	Forte	40
Код газеты	Комсомольская правда (на Дону)	Sunrise	108
Код газеты	Что Где Почем	VVV	192
Код газеты	Телеком	Sunrise	156

Имя таблицы: Код газеты  
Имя таблицы: Запрос на вычисление  
Групповая операция: Группировка  
Перекрестная таблица: Заголовки строк, Заголовки столбцов, Значение  
Сортировка:  
Условие отбора:  
или:

Рисунок 14

Для строк перекрёстного запроса выберем поле Наименование изделия из таблицы Код газеты, а так же поле Рекламодатель из Запроса на вычисления. Для вывода расчетов используем поле Наименование изделия.