

ГОУВПО  
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ  
к лабораторным работам по дисциплине

«Организация баз данных и знаний»

(для студентов направлений подготовки 09.03.02 "Информационные системы и технологии"

(специальностей "Информационные технологии в медиаиндустрии и дизайне", "Системы автоматизированного проектирования"), 02.03.01 "Математика и компьютерные науки"

(специальность "Компьютерное моделирование и дизайн")

ГОУВПО  
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ  
к лабораторным работам по дисциплине

«Организация баз данных и знаний»

(для студентов направлений подготовки 09.03.02 "Информационные системы и технологии"  
(специальностей "Информационные технологии в медиаиндустрии и дизайне", "Системы  
автоматизированного проектирования"), 02.03.01 "Математика и компьютерные науки"  
(специальность "Компьютерное моделирование и дизайн"))

Рассмотрено на заседании кафедры  
программной инженерии

Протокол № 1 от 30.08.2016

Утверждено на заседании

учебно-издательского Совета ДонНТУ  
протокол № от

Донецк –2016

УДК 681.3

Методические указания и задания к лабораторным работам по дисциплине «Организация баз данных и знаний» для студентов направления подготовки 09.03.02 "Информационные системы и технологии" (специальностей "Информационные технологии в медиаиндустрии и дизайне", "Системы автоматизированного проектирования"), 02.03.01 "Математика и компьютерные науки" (специальность "Компьютерное моделирование и дизайн"), Шалдырван Г.Г., Костюкова Н.С., Чернышова А.В., Кузьмичева А.С. Донецк, ДонНТУ, 2016 - 118 стр.

Приведены методические указания и задания к выполнению лабораторных работ по дисциплине «Организация баз данных и знаний». Излагаются вопросы, связанные с проектированием концептуальной модели базы данных, созданием таблиц, реализацией QBE и SQL запросов, разработкой приложения с помощью форм и создание отчетов к базе данных.

Методические указания предназначены для усвоения теоретических основ и формирования практических навыков по курсу «Организация баз данных и знаний».

Составители:

к.т.н., доцент Шалдырван Г.Г.

к.т.н., доцент Костюкова Н.С.

ст.преп. каф. ПИ Чернышова А.В.

ассистент Кузьмичева А.С.

# Лабораторная работа 1

**Тема:** Проектирование концептуальной модели базы данных.

**Цель работы:** изучение вопросов представления данных об объектах предметной области и их связях в рамках реляционной модели.

## Методические указания к выполнению работы

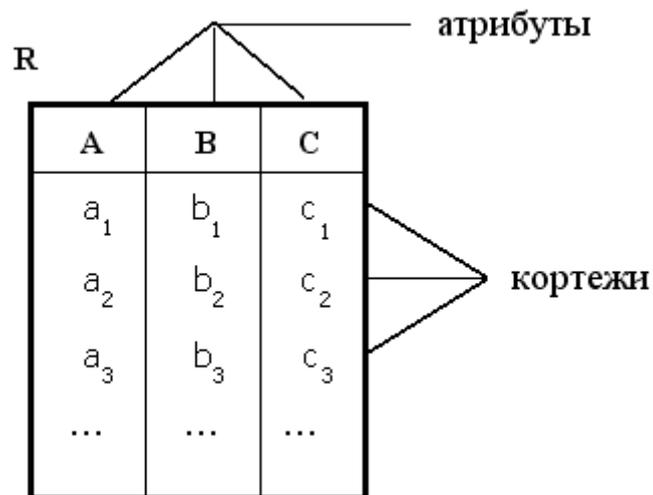
### Основные определения

Каждая СУБД основывается на использовании определенной *модели данных*. **Модель данных** – это форма представления объектов предметной области и связей между ними.

В 1970 г. сотрудник корпорации IBM Э.Ф. Кодд, математик по образованию, использовал для обоснования модели данных математический аппарат теории множеств и предикативной логики. МД, предложенная Коддом, получила название *реляционной модели*. Ее идеи подробно рассмотрены в ставшей классической статье “Реляционная модель данных для больших совместно используемых банков данных”. Предложения Кодда были настолько эффективны для систем баз данных, что за эту модель он был удостоен престижной премии Тьюринга в области теоретических основ вычислительной техники.

Идеи Кодда оказали весьма существенное влияние на все аспекты технологии баз данных, а также на такие области, как искусственный интеллект, обработка естественных языков, разработка аппаратных систем.

Основной структурой данных в реляционной модели является *отношение*, именно поэтому модель получила название *реляционной* (от английского *relation* – отношение). Отношение имеет простую графическую интерпретацию: оно может быть представлено в виде *таблицы*:



**R** – отношение (таблица).

**Кортеж** - строка таблицы (включает данные об одном объекте (например, сведения о банке, о клиентах банка, анкетные данные студента и др.)).

**Атрибут** - столбец таблицы (столбцы представляют различные характеристики объекта (например, ФИО студента, номер его зачетной книжки, адрес, )).

Множество значений, которые может принимать некоторый атрибут, называют *доменом*. Например, атрибут **Год рождения** имеет разные домены для таблицы **Студенты** и таблицы **Пенсионеры**.

Строки таблицы называют *записями*. Все записи имеют одинаковую структуру: они состоят из *полей*, в которых хранятся характеристики объекта. Каждое поле записи имеет строго определенный *тип данных*.

**Определение реляционной модели данных (РМД):** для хранения **данных** об объектах предметной области и для моделирования **взаимосвязей** между ними используется набор таблиц.

Таким образом, БД представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию определенного типа.

Каждая таблица должна иметь *первичный ключ*: одно или несколько полей, которые имеют уникальные значения для каждой строки. Если первичный ключ достаточно сложный (состоит из нескольких полей таблицы

или из одного, но длинного поля), то можно создать *искусственный первичный ключ*, представляющий собой некоторый код, для которого отводится одно поле. Код обычно является числовым или символьным данным. Использование *искусственного первичного ключа* гарантирует уникальность записей в таблице.

### **Представление связей между таблицами**

Для моделирования связей между таблицами используется понятие *внешнего ключа*. *Внешний ключ* - это поле таблицы, которое является *первичным ключом* в другой таблице; служит для организации связи между этими таблицами. Для того чтобы реляционная БД была непротиворечивой, каждому значению внешнего ключа таблицы должно соответствовать значение первичного ключа связанной с ней таблицы. Это условие известно как *ссылочная целостность*. Создание связей между таблицами позволяет выбирать данные сразу из нескольких таблиц.

Между объектами предметной области возможны следующие виды связей:

- ”один-к-одному” ( “1:1” ),
- “один-ко-многим” ( “1:n” ),
- “многие-ко-многим” ( “m:n” ).

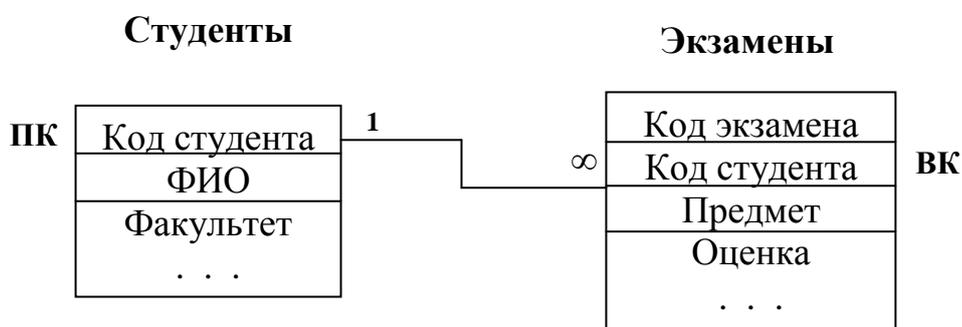
В большинстве случаев две таблицы связаны отношением “**1:n**”. Это означает, что одна запись первой таблицы (таблицы, в которой поле связи является первичным ключом; таблицы со стороны “один”) может быть связана с **несколькими** записями второй таблицы (таблицы, в которой поле связи является внешним ключом; таблицы со стороны “многие”). Однако, любая запись второй таблицы связана только с одной записью первой.

Для пояснения смысла связей между таблицами удобно использовать “родительскую” терминологию. Таблицу, в которой поле связи является первичным ключом, называют “отцом”, а таблицу, в которой поле связи является внешним ключом, - “сыном”. Связь “1:n” означает, что у “отца”

(таблица со стороны “один”) могут быть несколько “сыновей” (таблица со стороны “многие”). Другими словами: одна запись из таблицы – отца может быть связана с несколькими записями из таблицы – сына. Частные случаи этой связи: “отец” имеет одного “сына”, “отец” не имеет ни одного “сына”.

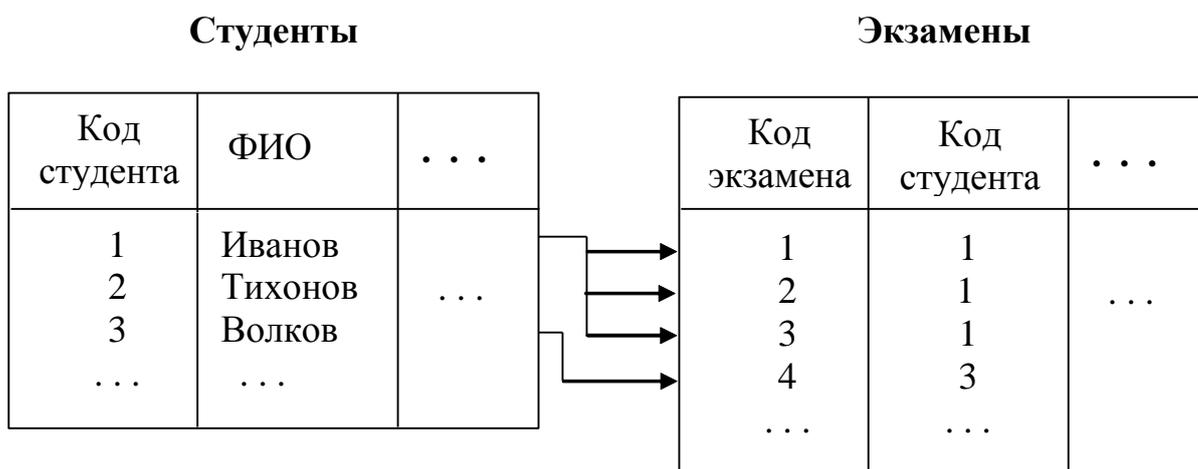
Ссылочную целостность, следуя данной терминологии, можно сформулировать так: не может быть “сына” без “отца”.

**Пример:** таблица **Студенты**, содержащая основные сведения о студентах, и таблица **Экзамены**, в которой содержатся сведения о результатах экзаменов, связаны отношением “1:n”:



Поле связи этих таблиц является поле **Код студента**.

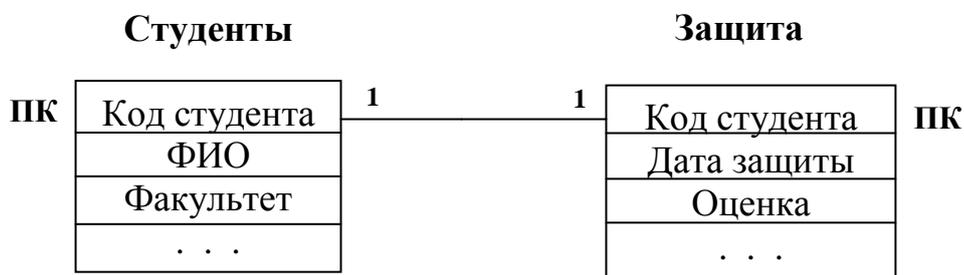
На уровне записей эта связь будет выглядеть так



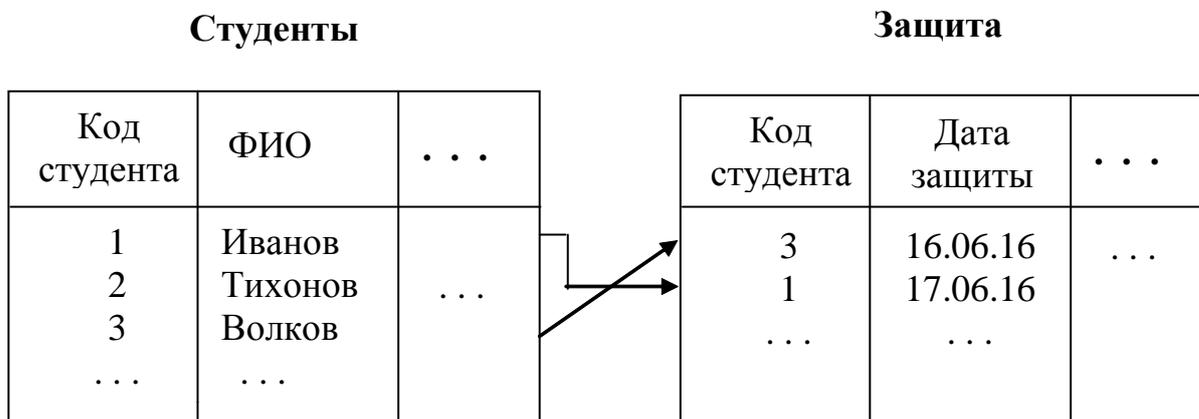
Студент Иванов (код **1**) сдал 3 экзамена (1:n), студент Волков (код **3**) сдал один экзамен (частный случай), студент Тихонов пока не сдал ни одного экзамена (частный случай).

Связь “1:1” означает, что любая запись первой таблицы связана только с одной записью второй таблицы (в частности, ни с одной). В этом случае обе таблицы должны иметь один и тот же первичный ключ.

**Пример:** таблица **Студенты**, содержащая основные сведения о студентах, и таблица **Защита**, в которой содержатся сведения о результатах защиты дипломных проектов, связаны отношением “1:1” (поле связи – **Код студента**):



На уровне записей связь выглядит так:



Студент Тихонов пока не защитил дипломный проект (частный случай)

Связь “m:n” означает, что одна запись первой таблицы может быть связана с несколькими записями второй таблицы и одна запись второй таблицы может быть связана с несколькими записями первой таблицы. Такой вид связи в РМД непосредственно не реализуется. Нужно создать дополнительную таблицу, называемую *таблицей пересечения*, с помощью которой одна связь

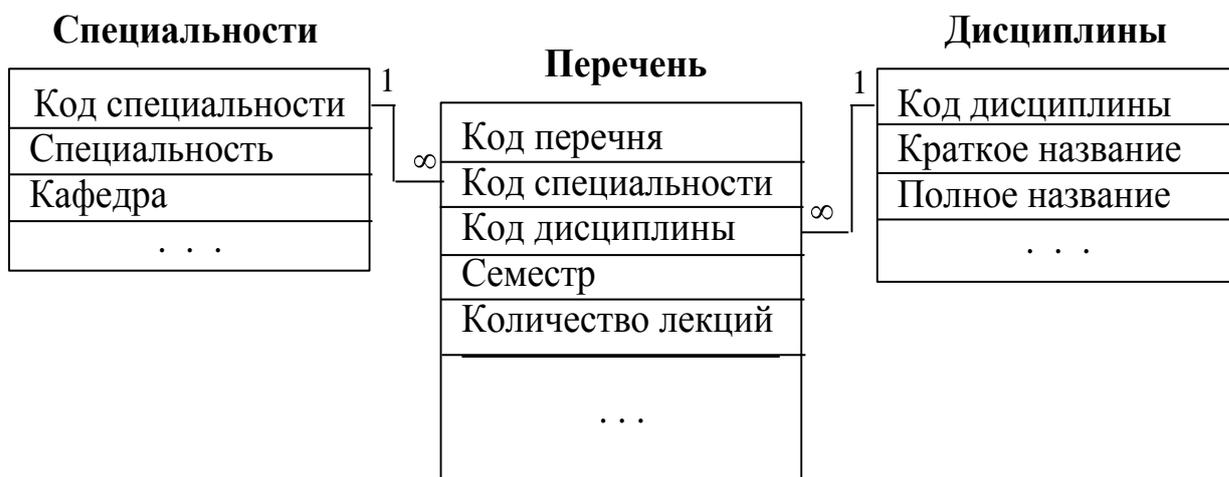
“m:n” будет сведена к двум связям “1:n” (таблица пересечения должна находиться со стороны “многие”).

**Пример:** таблица **Специальности**, содержащая сведения о специальностях института, и таблица **Дисциплины**, в которой приведен перечень читаемых в институте дисциплин, связаны отношением ”m:n” (с одной стороны, на одной специальности читается несколько дисциплин, с



другой стороны, некоторая дисциплина преподается на нескольких специальностях):

Нужно создать таблицу пересечения, в которой приводится перечень дисциплин по специальностям:



В таблице пересечения **Перечень** поля **Код специальности** и **Код дисциплины** являются внешними ключами.

## Нормализация отношений

После анализа данных, характеризующих предметную область, отдельные данные объединяются в таблицы. Группировка данных в таблицы не имеет единственного решения. Можно получить несколько наборов таблиц, которые будут отображать одни и те же данные. Нужно определить оптимальный набор, который обладает лучшими свойствами при включении, удалении и модификации данных, а также позволяет сделать избыточность данных и время ответов на запросы к БД минимальными.

Основоположник РМД Э. Кодд с помощью разработанной им *теории нормализации* показал, какую структуру должны иметь таблицы, чтобы их набор был наиболее эффективным и, следовательно, был выбран в качестве концептуальной модели. Согласно теории нормализации таблица может находиться в одной из трех *нормальных форм*:

- первая нормальная форма (1НФ),
- вторая нормальная форма (2НФ),
- третья нормальная форма (3НФ).

Понятие нормальных форм связано с понятием *однозначной зависимости*: атрибут  $Y$  однозначно зависит от атрибута  $X$ , если каждому значению  $X$  соответствует только одно значение  $Y$ .

Каждая нормальная форма – это, прежде всего, *нормализованное отношение* (*нормализованная таблица*). Оно удовлетворяет следующим требованиям:

- кортежи в отношении (строки в таблице) должны быть различны,
- порядок кортежей (строк) не существенен,
- порядок атрибутов (столбцов) не существенен,
- каждый атрибут должен быть неделимым.

Таблицы, созданные в начале процесса проектирования концептуальной модели, обычно находятся в 1НФ.

**1НФ – это нормализованное отношение, характеризующееся следующими недостатками:**

- сложность в определении первичного ключа,
- избыточность данных,
- неоднозначные зависимости между неключевыми атрибутами и первичным ключом.

Для решения проблем, связанных с 1НФ, нужно привести таблицу ко второй нормальной форме.

**2НФ:** неключевые атрибуты однозначно зависят от первичного ключа, но, в то же время, однозначная зависимость может существовать и между неключевыми атрибутами отношения.

Для решения этой проблемы Кодд предложил для таблиц третью нормальную форму.

**3НФ:** отношение находится в третьей нормальной форме, если неключевые атрибуты его взаимно независимы и однозначно зависят только от первичного ключа.

Процесс преобразования таблиц из одной НФ в другую НФ называется *нормализацией*. Нормализация - это разбиение исходной таблицы на более мелкие таблицы с целью приведения их к 3НФ.

Таким образом, основную идею нормализации можно выразить так: **атрибуты отношения должны однозначно зависеть от первичного ключа, от всего ключа целиком и ни от чего другого, кроме ключа.**

Э. Кодд с помощью математического аппарата теории множеств доказал, что только набор таблиц, находящихся в 3НФ, является оптимальным и может служить моделью реляционной базы данных (3НФ позволяет гораздо эффективнее выполнять операции включения, удаления и модификации записей БД, а также делает избыточность данных минимальной).

### **Выбор первичного ключа**

Каждая таблица реляционной БД должна иметь *первичный ключ* (ПК): одно или несколько полей, однозначно определяющих каждую запись в таблице. Наличие в таблицах первичного ключа поддерживает *целостность*

*данных*: СУБД не позволит ввести в это поле дублирующее значение или значение Null (пусто).

ПК может быть естественным или искусственным. Если поле содержит достаточно короткие уникальные значения, например, номера школ, больниц, почтовых отделений, то это поле можно определить как естественный первичный ключ.

В случае, когда уникальность записей не гарантируется значениями одного поля, нужно создать составной ключ. В случаях, когда имеем длинный естественный ПК, нужно создать искусственный первичный ключ – некий **код**, правила построения которого должны быть достаточно простыми.

**Код** может быть:

- числовым,
- символьным,
- счетчиком.

**Числовой код** (значения 1, 2, 3...) имеет смысл использовать в тех случаях, когда таблица содержит сравнительно немного записей. Он может занимать 1 байт (в таблице не может быть более 255 записей) или 2 байта (в таблице не более 32767 записей). Нарастивание значений числового кода должно выполняться автоматически, без вмешательства пользователя. Для этого нужно написать макрос (модуль), в котором используются соответствующие системные функции.

**Символьный код** должен быть коротким: 1-2 символа. Например, можно взять первую букву длинного названия, если значения ее для записей таблицы будут уникальными (таблица НАЗВАНИЯ ЦЕХОВ примера 1). Однако, программная поддержка символьного кода (выделение букв из названий, проверка их на уникальность) достаточно сложна. Поэтому такой вид ПК может использоваться только для небольших сразу заполненных таблиц (таблиц-справочников).

Для таблиц, имеющих много записей и постоянно пополняющихся, самый удобный вид искусственного ПК – поле типа **Счетчик**. Значение такого поля автоматически увеличивается на единицу при добавлении новой записи в таблицу. Данный способ является самым легким для создания первичного ключа, но следует помнить, что в этом случае система гарантирует обеспечение уникальности только этого поля. Если же в таблице имеется другое поле, значения которого должны быть уникальными, то эту проблему должен решать программист.

Поле первичного ключа выполняет роль связующего поля между двумя таблицами. Кроме того, по умолчанию СУБД сортирует данные по первичному ключу, что значительно ускоряет выполнение запросов, а также поиск и сортировку данных.

Основой для всех современных СУБД является реляционная модель данных, базирующаяся на теории множеств и отношений (англ. *relation* – отношение). Отношение имеет простую графическую интерпретацию в виде таблицы. На логическом уровне БД представляет собой множество взаимосвязанных таблиц. Каждая строка (*запись*, или *кортеж*) таблицы включает данные об одном объекте (например, банке, клиенте, студенте), а столбцы (*поля* или *атрибуты*) таблицы содержат различные характеристики этих объектов (например, наименование банка, фамилия и адрес клиента, номер зачетной книжки студента). Таким образом, *реляционной* называется модель данных, в которой для хранения данных об объектах предметной области и для моделирования взаимосвязей между ними используется набор таблиц.

После анализа предметной области отдельные данные объединяются в таблицы. Такая группировка не имеет единственного решения, поэтому нужно определить оптимальный набор, который обладает лучшими свойствами при включении, удалении и модификации данных, а также позволяет сделать избыточность данных и время ответов на запросы к БД минимальными.

Рассмотрим примеры проектирования концептуальной модели базы данных.

## Пример 1

Для автоматизации учета работы молокозаводов области информационная система должна содержать такие данные о заводах: название завода, город, в котором находится завод, год начала функционирования, директор, телефон, цехи (название цеха, дата ввода в действие), выпускаемая продукция (тип продукции (сметана, кефир, ряженка, творог...), жирность, вес, дата выпуска, объем, цена за 1 кг. ).

После первоначального анализа данных, характеризующих указанную предметную область, можно выделить объекты

### ЗАВОДЫ, ПРОДУКЦИЯ.

Соответствующие им таблицы можно представить так:

#### ЗАВОДЫ

Завод	Город	Начало	Директор	Тел	Цех	Дата ввода
Геркулес	Донецк	1993	Иванов.	...	Творожный	...
Геркулес	Донецк	1993	Иванов	...	Жидких кисломолочных продуктов	
Геркулес	Донецк	1993	Иванов	...	Жидких пастеризованных продуктов	
Маричка	Горловка	1999	Петров	...	Творожный	
Маричка	Горловка	1999	Петров	...	Жидких кисломолочных продуктов	
Маричка	Горловка	1999	Петров	...	Жидких пастеризованных продуктов	
Лактис	Марьинка	1998	Власов	...	Творожный	
Лактис	Марьинка	1998	Власов	...	Жидких кисломолочных продуктов	
...	...	...	...	...	...	...

## ПРОДУКЦИЯ

Тип	Жирность	Вес	Дата выпуска	Цена единицы
Молоко	...	?	?	?
Творог	...	?	?	?
Ряженка	...	?	?	?
.....	...			

Недостатки выполненного проектирования:

1. Избыточность данных (в таблице «Заводы» многократно повторяются данные о каждом заводе).
2. Неадекватность представления данных «Дата выпуска», «Вес» и «Цена единицы» в таблице «Продукция».
3. Отсутствие связи между таблицами .

Нормализуем таблицу «Заводы», разбив ее на 2 таблицы: «Заводы» (в ней будут приведены сведения о заводах; первичный ключ таблицы – поле «Завод») и «Цехи» (сведения о цехах заводов; поле «Завод» - внешний ключ; первичный ключ - составной: «Завод + Цех»).

## ЗАВОДЫ

Завод	Город	Начало	Директор	Телефон
Геркулес	Донецк	1993	...	...
Маричка	Горловка	1999	...	...
Лактис	Марьинка	1998	...	...

## ЦЕХИ

Завод	Цех	Дата ввода
Геркулес	Творожный	...
Геркулес	Жидких кисломолочных продуктов	...
Геркулес	Жидких пастеризованных продуктов	...
Маричка	Творожный	...
Маричка	Жидких кисломолочных продуктов	...
.....	.....	...

Между таблицами «Заводы» и «Цехи» существует связь “один-ко-многим” (один завод имеет несколько цехов). Так как названия цехов повторяются, то целесообразно организовывать справочник названий; в этой таблице поле «Код названия цеха» – искусственный ПК.

## НАЗВАНИЯ ЦЕХОВ

Код названия цеха	Цех
Т	Творожный
К	Жидких кисломолочных продуктов
П	Жидких пастеризованных продуктов
....	.....

В таблице «Заводы» введем искусственный первичный ключ, чтобы в таблице «Цехи» заменить длинное поле «Завод» коротким полем «Код завода»:

## ЗАВОДЫ

Код завода	Завод	Город	Начало	Директор	Телефон
1	Геркулес	Донецк	...	...	...
2	Маричка	Горловка	...	...	...
3	Лактис	Марьинка	...	...	...

...	...	...	...	....	...
-----	-----	-----	-----	------	-----

Для таблицы «Цехи» составной ПК «Завод + Цех» также нужно заменить простым ключом. Для этого введем искусственный ПК «Код цеха»:

### ЦЕХИ

Код цеха	Код завода	Код названия цеха	Дата ввода	Количество
1	1	Т	...	...
2	1	К	...	...
3	1	П	...	...
4	2	Т	...	...
5	2	К	...	...
...	...	...	...	...

Между объектами ЦЕХИ–ПРОДУКЦИЯ существует отношение «m:n» (с одной стороны, каждый цех выпускает несколько видов продукции, с другой стороны, каждый вид продукции выпускается цехами нескольких заводов). Для представления этой связи необходимо ввести таблицу пересечения.

Разобьем таблицу «Продукция» на три: «Виды продукции», «Ассортимент» и «Выпускаемая продукция». «Код вида продукции», «Код продукции» – искусственные первичные ключи.

### ВИДЫ ПРОДУКЦИИ

Код вида продукции	Вид
1	Сметана
2	Кефир
3	Ряженка
...	...

## АССОРТИМЕНТ

Код продукции	Код вида продукции	Жирность
1	1	10
2	1	15
3	1	20
4	2	2,5
5	2	3,2
6	2	1

## ВЫПУСКАЕМАЯ ПРОДУКЦИЯ

Код цеха	Код продукции	Дата выпуска	Объем	Цена
1	2	01.02.2016	120	17.90
1	3	01.02.2017	210	21.20
1	3	01.02.2016	220	26.40
5	2	02.02.2017	112	19.00

Таблица «Выпускаемая продукция» является таблицей пересечения («Код цеха» – внешний ключ, «Код продукции» – внешний ключ).

Естественный первичный ключ – составной: «Код цеха + Код продукции + Дата выпуска», поэтому введем короткий искусственный ПК.

Структура рассмотренных таблиц (все находятся в ЗНФ) и связи между ними приведены на рис. 1.

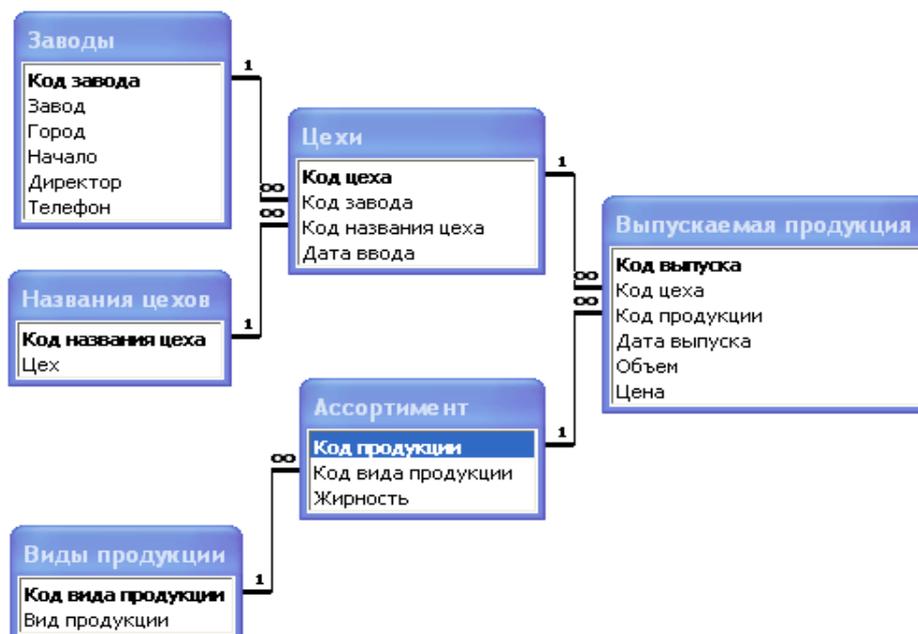


Рисунок 1 Концептуальная модель данных для БД «Продукция молокозаводов»

### Пример 2.

Для учета работы посреднической фирмы, занимающейся продажей вычислительной техники, необходимы сведения о поставщиках (название фирмы, город, страна, форма собственности, телефон, адрес) и о представленных ими товарах (наименование товара, фирма-изготовитель, дата поставки партии, цена единицы, количество единиц)

Первоначально рассмотрим представление БД в виде двух таблиц.

Первая таблица – «Поставщики» - выглядит таким образом:

Фирма	Город	Страна	Собственность	Телефон	Адрес
Альфа	Москва	Россия	частная	...	Ул.Парковая, 334
Гамма	Минск	Беларусь	государственная	...	...
Омега	Киев	Украина	акционерная	...	...
...	...	...	...	...	...

Вторая таблица – «Товары»:

Товар	Изготовитель	Дата поставки	Цена единицы	Количество единиц
Принтер	Hewlett Packard	10.01.15	2000.00	10
DVD-RW	Acme	12.01.16	4.00	100
CD-RW	Memorex	12.02.16	2.00	300
CD-R	Acme	12.04.14	2.00	100
...	...	...	...	...

Данное представление имеет ряд недостатков:

1) Таблицы не связаны, следовательно, нельзя выяснить, кто поставлял заданный товар, какова общая стоимость товаров, представленных поставщиком, и т.п.

2) Избыточность в полях «Город», «Страна», «Собственность» – возможно многократное повторение одних и тех. же заранее известных значений.

3) Присутствует аномалия обновления: если нужно изменить название товара, необходим просмотр всех его поставок.

4) Присутствует аномалия включения: нельзя включить в БД информацию о новом товаре до тех пор, пока не будет выполнена хотя бы одна его поставка/

5) Присутствует аномалия удаления: если некоторый товар исключен из ассортимента поставщиков, то нужно удалять записи обо всех его поставках.

6) Из-за того, что таблица «Поставщики» находится в НФ2 (присутствует зависимость между неключевыми полями «Страна» и «Город»), возможно занесение в таблицу заведомо недостоверных данных).

7) В реальной жизни между объектами «Поставщики» и «Товары» имеет место связь «многие ко многим»: один и тот же поставщик может поставлять разные товары, один и тот же товар может поставляться разными поставщиками. Для реализации этой связи введем таблицу пересечения, содержащую сведения о поставках товаров, поскольку явно такая связь представлена быть не может.

Для устранения избыточности в полях «Город», «Страна», «Собственность» введем справочники «Страны», «Города», «Собственность» с искусственными первичными ключами. Для связи включим одноименные внешние ключи в базовые таблицы.

Чтобы устранить аномалии обновления, включения и удаления, выделим отдельную таблицу, содержащую ассортимент товаров, предлагаемых поставщиками (таблица «Каталог»).

Чтобы избежать зависимости между неключевыми полями, информацию о городах и странах будем хранить в двух связанных таблицах.

Для отражения связи между объектами «Поставщики» и «Товары», присутствующей в реальной жизни, создадим таблицу пересечений «Товары», содержащую данные о дате, объеме поставки каждого товара и цене единицы.

Для реализации связей в каждую таблицу включим искусственный первичный ключ, а в подчиненные таблицы – внешний ключ.

Получим набор таблиц, показанный на рис. 2.

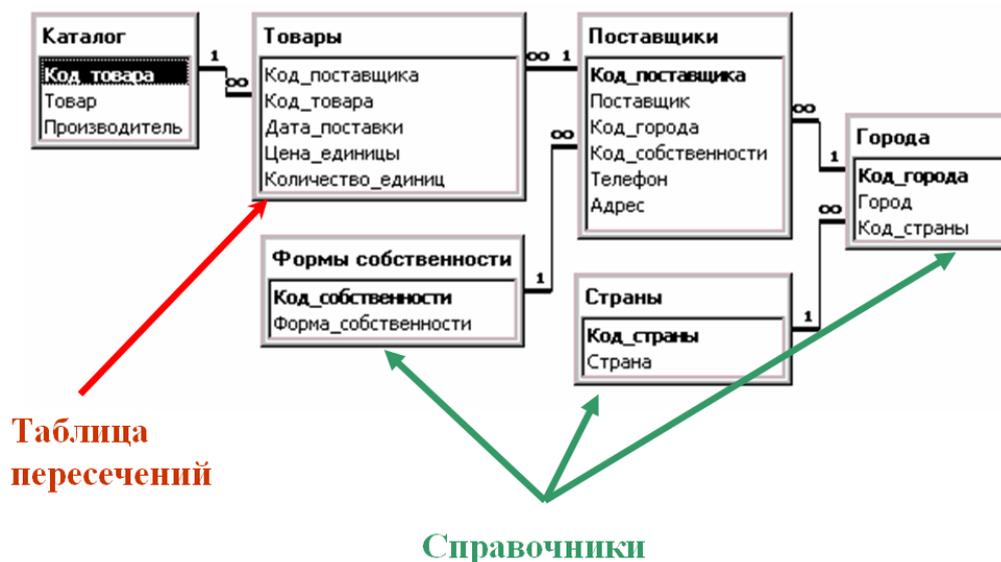


Рисунок 2 – Концептуальная модель данных для БД учета работы посреднической фирмы

### Задание к лабораторной работе

Для предметной области, описанной в заданном варианте (вариант задания см. в приложении к методическим указаниям), выделить объекты и

представить их взаимосвязанными таблицами, находящимися в третьей нормальной форме. Определить базовые таблицы и таблицы-справочники; указать для них первичные и внешние ключи. Описать типы полей.

### **Требования к отчету по лабораторной работе:**

- 1) Концептуальная модель к базе данных по варианту.
- 2) Указать первичные и внешние ключи.
- 3) Указать типы полей в основных таблицах и таблицах пересечения.

### **Контрольные вопросы**

1. Сформулировать следующие определения:
  - предметная область, БД, СУБД;
  - реляционная модель данных;
  - нормализованная таблица, нормализация;
  - нормальные формы таблиц (первая, вторая, третья);
  - первичный, искусственный, внешний ключи;
  - целостность данных, ссылочная целостность.
2. Виды связей между объектами предметной области.

## Лабораторная работа 2

**Тема:** Создание базы данных

**Цель работы:** освоение возможностей создания таблиц и установления связей между ними.

### Методические указания к выполнению работы

При создании базы данных необходимо придерживаться такой последовательности:

1. Создание таблиц базы данных, установка свойств полей и таблиц, создание подстановок для внешних ключей. При этом в первую очередь следует создавать главные таблицы, то есть находящиеся на линии связи со стороны «один», а после них – подчиненные.
2. Установление связей между таблицами, создание схемы данных.
3. Заполнение таблиц данными.

### Создание таблиц базы данных

Рассмотрим в качестве примера создание таблиц для учебной базы данных, процесс проектирования которой рассмотрен в лабораторной работе №1. В начале создадим таблицы «Каталог», «Формы собственности», «Страны», затем таблицы «Города» и «Поставщики», и в последнюю очередь – таблицу «Товары».

При создании таблицы будем использовать режим конструктора – наиболее гибкое средство.

Для создания новой таблицы нужно перейти на вкладку «Создание» и нажать кнопку «Конструктор таблиц» .

Окно конструктора таблицы показано на рис.1. В верхней части окна размещается список полей таблицы с указанием их типов данных и описаний (при необходимости). В нижней части окна приводятся свойства текущего поля. Набор свойств может быть различным для полей разных типов. Значения некоторых свойств задаются по умолчанию.

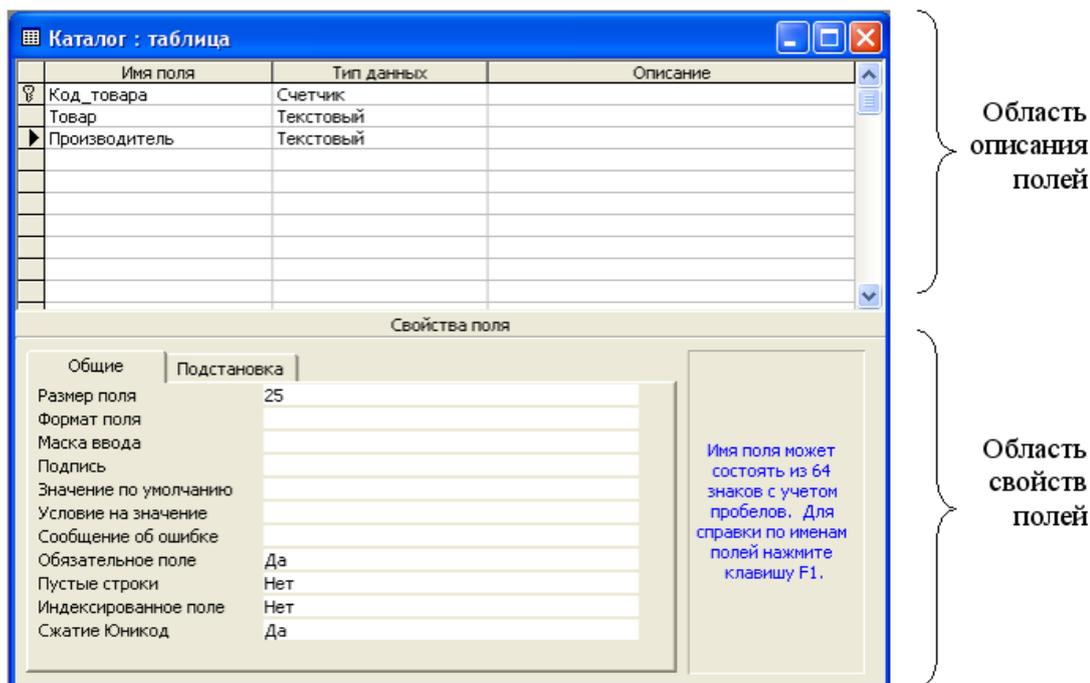


Рисунок 1 – Общий вид окна конструктора таблицы

Рассмотрим этапы создания таблицы «Страны».

Зададим имена полей создаваемой таблицы, их типы и свойства полей.

Перечень типов полей приведен в приложении А.

Поле «Код\_страны» является искусственным первичным ключом таблицы. Выберем для него тип данных «Счетчик», гарантирующий уникальность значений. Чтобы указать, что поле является искусственным ключом, ....

Поле «Страна» будет содержать только алфавитную информацию, поэтому выберем для него тип данных «Текстовый». Размер поля установим равным 25 символам. Кроме этого, укажем в свойстве «Обязательное поле» значение «Да», а в свойстве «Индексированное поле» выберем значение «Да, совпадения не допускаются». Два последних свойства очень важны, так как по сути это поле является естественным первичным ключом.

Сохраним таблицу, указав при этом ее имя «Страны».

Аналогично создадим таблицы «Каталог» и «Формы собственности».

Далее перейдем к созданию таблицы «Города», которая является подчиненной по отношению к таблице «Страны». При создании поля

«Код\_страны» (внешний ключ) воспользуемся возможностью организовать ввод данных в это поле путем выбора значения из раскрывающегося списка. При этом физически в поле внешнего ключа будет сохраняться числовое значение из справочника, а выводиться на экран будет соответствующее ему текстовое значение.

Для типа поля внешнего ключа выберем последний элемент списка – «Мастер подстановок». В ходе создания подстановки пользователь отвечает на вопросы надстройки – мастера. Первый такой вопрос связан с источником данных для списка подстановки (см.рис.2), в нашем случае для всех подстановок используются значения из другой таблицы.

Следующим шагом будет выбор таблицы- источника строк подстановки. В нашем случае это таблица «Страны» (см.рис.3).

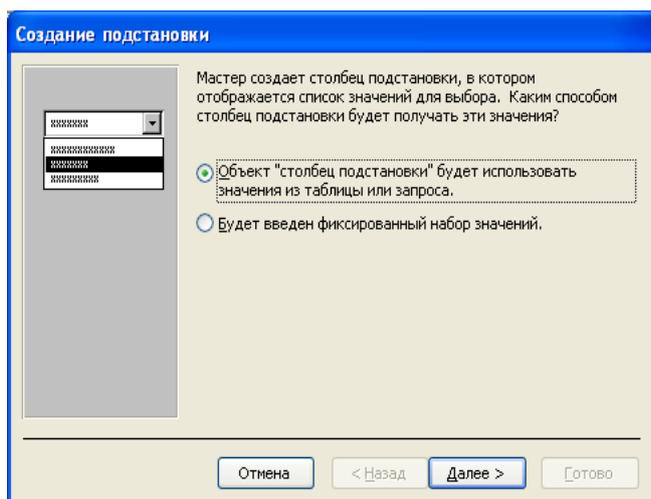


Рисунок 2

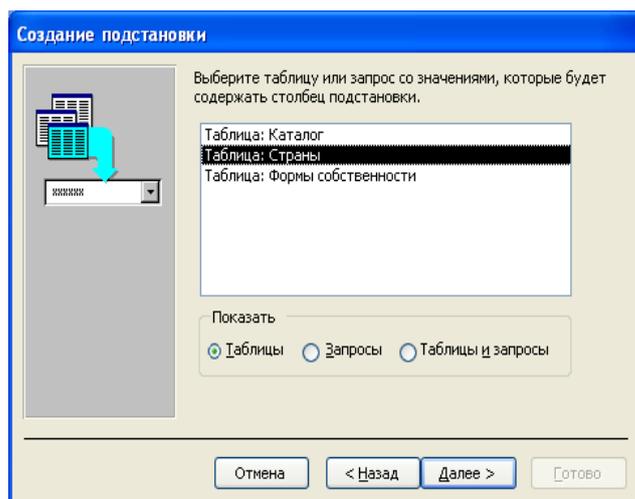


Рисунок 3

После этого нужно указать поля, участвующие в подстановке. Таких поле должно быть как минимум два. Одно – это поле первичного ключа, значение которого заносится в создаваемое нами поле в результате выполнения подстановки. Второе – соответствующее текстовое значение из связанной таблицы, которое будет выводиться на экран вместо малоинформативного кода. Выполнение данного шага показано на рисунке 4.

На следующем шаге можно указать, каким образом будут упорядочены строки в списке подстановки, после чего скрыть ключевой столбец, содержащий малопонятное для пользователя искусственное значение (рис.5).

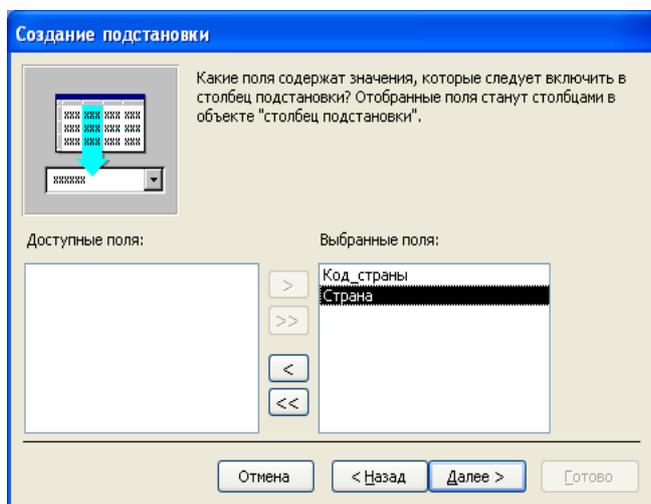


Рисунок 4

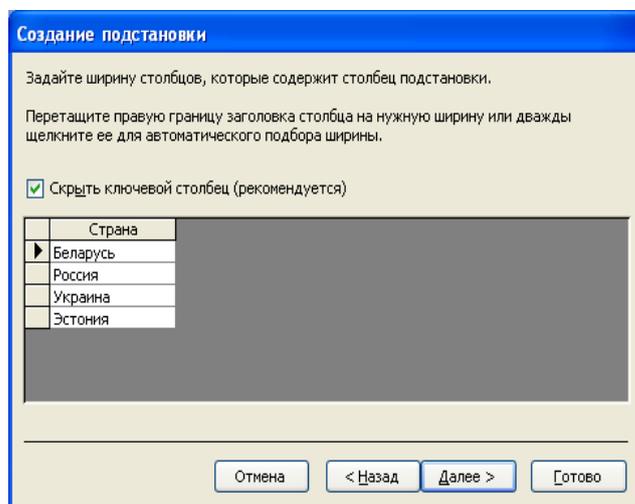


Рисунок 5

Последним шагом при создании подстановки является задание подписи для поля, однако эту операцию лучше выполнить позже, непосредственно в окне конструктора таблицы.

Аналогично создадим и остальные таблицы нашей базы данных: «Поставщики» и «Товары».

При создании таблицы «Поставщики» воспользуемся мастером подстановок для полей «Код\_города» и «Код собственности», задав подписи полей «Город» и «Форма собственности». Для поля «Телефон» выберем тип данных «Текстовый»: хотя значения этого поля и содержат цифровую информацию, в вычислениях они участвовать не будут. Для данного поля укажем размер поля- символов и зададим маску ввода, обеспечивающую ввод только цифр и заполнение поля справа налево:

!###\-##\-##;1

Правила формирования маски ввода и символы, которые для этого используются, приведены в приложении В.

В приложении Б приведены форматы вывода значений разных типов.

При создании таблицы «Товары» обратим внимание на поля «Дата\_поставки», «Цена\_единицы» и «Количество\_единиц».

Для поля «Дата\_поставки» выберем тип «Дата/время», укажем значения свойств «Формат поля», «Маска ввода», «Подпись», «Значение по умолчанию», «Условие на значение», «Сообщение об ошибке», «Обязательное поле».

Для полей «Цена\_единицы» и «Количество\_единиц» укажем значения свойств «Формат поля», «Подпись», «Условие на значение», «Сообщение об ошибке», «Обязательное поле».

В таблице 1 приведено описание полей таблицы «Товары».

Таблица 1. Свойства полей таблицы «Товары».

Имя поля	Тип данных	Свойства
Код_поставки	Счетчик	Первичный ключ
Код_поставщика	Числовой	Размер поля: длинное целое
Код_товара	Числовой	Размер поля: длинное целое
Дата_поставки	Дата/время	Подпись: Дата поставки Формат поля: Краткий формат даты Маска ввода: 99/99/00 Обязательное поле: Да Значение по умолчанию: Date() Условие на значение: <=Date() Сообщение об ошибке: Неправильная дата!
Цена_единицы	Числовой	Размер поля: Одинарное с плавающей Подпись: Цена единицы Число десятичных знаков: 2 Формат поля: Денежный Обязательное поле: Да Условие на значение: >0 Сообщение об ошибке: Цена д.б. положительной
Количество_единиц	Числовой	Размер поля: длинное целое Подпись: Количество единиц Обязательное поле: Да Условие на значение: >0 Сообщение об ошибке: Количество д.б. положительным

## Создание схемы данных

После определения таблиц нужно установить связи между ними. В дальнейшем Access будет использовать эти связи для поддержания целостности данных и при создании запросов по нескольким таблицам. Связи устанавливаются в окне «Схема данных», которое вызывается кнопкой «Схема данных» на вкладке «Работа с базами данных»: .

В окне нужно выполнить следующие действия:

- щелкнуть по кнопке «Отобразить таблицу» ;
- в диалоговом окне «Добавление таблицы» по очереди выбрать необходимые таблицы, дважды щелкнув на имени каждой таблицы;
- закрыть окно «Добавление таблицы».

Чтобы установить связь между двумя таблицами – главной и подчиненной, щелкните по первичному ключу главной таблицы, перетащите и опустите его на соответствующий внешний ключ подчиненной. Когда вы отпустите кнопку мыши, Access откроет окно диалога «Изменение связей» (рис.6). В этом окне содержится информация о полях связи главной и подчиненной таблиц, параметрах обеспечения целостности данных и типе устанавливаемого отношения.

Первоначально нужно включить поддержку целостности данных с помощью опции «Обеспечение целостности данных». Правила поддержки целостности данных будут действовать при выполнении таких операций, как ввод или изменение данных (например, нельзя будет создавать в «сыне» записи, не имеющие «отца»). После этого Access позволит включить опции «Каскадное обновление связанных полей» и «Каскадное удаление связанных записей».

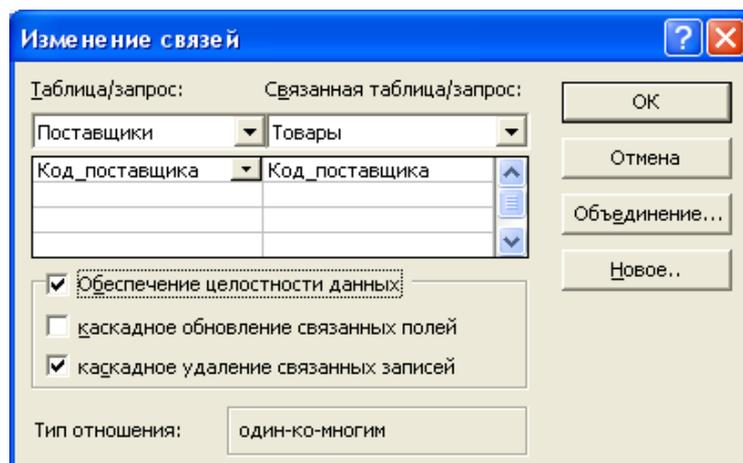


Рисунок 6 – Окно изменения связей

После выбора опций нужно установить связь, щелкнув по кнопке «ОК». После закрытия окна на линии связи возле главной таблицы будет стоять 1, а возле подчиненной – символ бесконечности.

Для редактирования существующей связи нужно дважды щелкнуть по линии связи - снова откроется окно «Изменение связей».

Частично связи создаются в результате работы мастера подстановок. Чтобы вывести эти связи на экран, нужно нажать кнопку «Все связи» . После этого, выполнив двойной щелчок на линии связи, можно изменить свойства этих связей.

На рис.7 показана схема данных, построенная для рассмотренного набора таблиц.

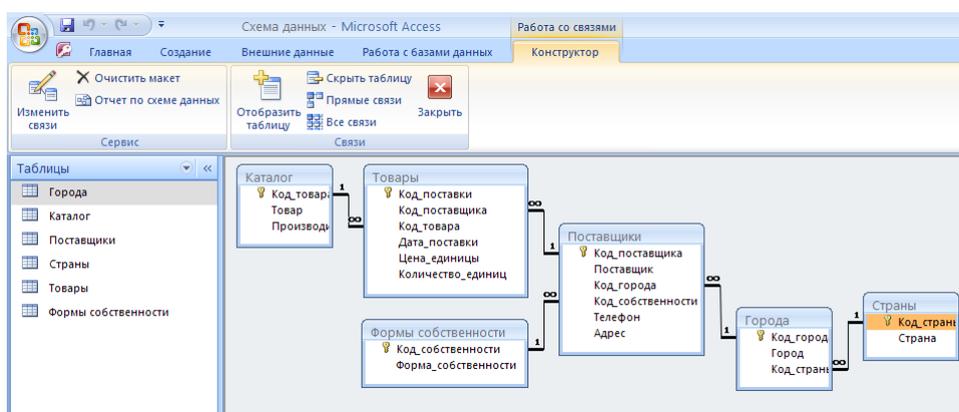


Рисунок 7 – Схема данных в окне Microsoft Access

## **Задание к лабораторной работе**

Для спроектированной реляционной модели выполнить определение таблиц, указать типы и свойства полей (основные свойства, подстановка), создать схему данных.

### **Требования к отчету по лабораторной работе:**

- 1) Привести скриншот созданной схемы данных с указанными типами связей.
- 2) Привести 2-3 скриншота основных таблиц в режиме конструктора.
- 3) Привести скриншот реализации подстановки из таблицы справочника в основную таблицу.

### **Контрольные вопросы**

1. Реализация ограничений целостности данных в Access.
2. Реализация связей между таблицами в Access.
3. Каскадное обновление полей, каскадное удаление записей.
4. Свойства полей (общие, подстановка ).

## Лабораторная работа 3

**Тема:** Разработка простых запросов на выборку средствами QBE

**Цель работы:** изучение способов получения информации с помощью запросов на выборку.

### Методические указания к выполнению работы

Запросы на выборку позволяют выбирать, сортировать, проводить вычисления с данными одной или нескольких взаимосвязанных таблиц.

Наиболее удобным способом создания запроса является использование режима конструктора. Для его активизации следует перейти на вкладку «Создание» и нажать соответствующую кнопку. Далее следует выбрать таблицы, на которых базируется запрос. После выбора таблиц открывается окно конструктора запросов. Оно разделено на две части горизонтальной *полосой раздела* (ее можно использовать для изменения размеров частей). В верхней части помещаются списки полей таблиц (запросов), на основании которых создается запрос. В нижней – располагается бланк QBE (Query By Example – запрос по образцу), в котором формируется запрос.

В бланке запроса первоначально представлены такие строки:

**Поле, Сортировка, Вывод на экран, Условие отбора, или.**

Строка **Поле** служит для представления полей, которые должны присутствовать в наборе записей. Она может содержать имя поля таблицы или вычисляемое поле.

Строка **Сортировка** служит для упорядочения выводимых данных. Для этого в строке **Сортировка** выбирается одно из значений: **По возрастанию** или **По убыванию**. Сортировка может быть многоуровневой, то есть по нескольким полям: условия сортировки применяются в порядке их расположения в бланке запроса слева направо.

В строке **Вывод на экран** с помощью флажков указываются поля, которые должны быть включены в результирующий набор записей. Если же

поле не нужно выводить (оно используется только для отбора записей), то флажок снимается.

Строки **Условие отбора** и **Или** используются для отбора записей с определенными значениями полей. Если условия для разных полей находятся в одной строке (строка **Условие отбора**), то система объединяет их оператором **And**. Если же условия находятся в двух строках (**Условие отбора, или**), то они объединяются оператором **Or**.

Для записи условия отбора используются:

- операторы сравнения = , <> , < , > , <= , >= , **Like**;
- специальные операторы сравнения **In**, **Between**;
- оператор **Is**;
- логические операторы **Not**, **And**, **Or**.

Оператор **In** выполняет множественное сравнение и имеет следующий синтаксис:

*Выражение In* (Значение1;Значение2;...)

Оператор **Between** проверяет вхождение значения в заданный диапазон:

*Выражение Between Значение1 And Значение2*

(*Значение1* – нижняя, *Значение2* – верхняя границы диапазона).

Оператор **Like** проверяет соответствие значения текстового или **МЕМО** поля заданному *шаблону*.

Анализ поля на пустое значение выполняется с помощью оператора **Is Null** и на непустое значение – с помощью **Is Not Null**.

При построении выражений часто возникает необходимость в использовании встроенных функций, например:

**Date()** – возвращает текущую системную дату ,

**Now()** - возвращает текущие системные дату и время,

**Year(дата)** – значение года в виде четырех цифр (целое) ,

**Year(date())** – значение текущего года в виде четырех цифр (целое) ,

**Format(дата; "уууу")** – значение года (например, "2013") ,

**Format(дата; "d-mmmm-уууу")** – полная дата ("1-января-2013") ,

**Format**(“донецк”;”>”) – возвращает “ДОНЕЦК” ,

**Format**(дата; “mmm”) – сокращенное название месяца (янв,фев,...),

**DatePart**(“q”; дата) – определяет номер квартала (1- 4) ,

**DatePart**(“уууу”; дата) – определяет год.

Поля запроса наследуют многие свойства, заданные соответствующим полям таблицы. При необходимости можно изменить значения таких свойств как **Формат поля**, **Число десятичных знаков**, **Маска ввода**, **Подпись** в окне свойств, вызываемом щелчком по нужному полю и по кнопке **Свойства**. Для вычисляемых полей рекомендуется всегда устанавливать эти свойства, так как назначаемые по умолчанию значения могут не подойти.

Рассмотрим пример создания простых запросов на выборку.

**Пример 1.** Вывести сведения о поставщиках и представленных ими партиях товаров (**симметричное объединение без условия**).

Для создания такого запроса перейдем к режиму конструктора, предварительно выбрав таблицы, необходимые для выполнения запроса: «Каталог», «Товары» и «Поставщики». В бланк QBE запроса включим поля «Поставщик» и «Код города» из таблицы «Поставщики», «Товар» из таблицы «Каталог», «Цена\_единицы», «Количество\_единиц» и «Дата\_поставки» из таблицы «Товары». Для включения полей в бланк запроса можно выполнить двойной щелчок по его имени в верхней части окна конструктора. Также можно перетащить поле из списка полей в нужный столбец бланка. Кроме того, можно воспользоваться раскрывающимся списком первой строки бланка и выбрать из него нужное поле.

Далее укажем, что записи, выбранные в результате выполнения запроса, нужно выводить в алфавитном порядке названий поставщиков. Для этого в столбце «Сортировка» поля «Поставщик» из раскрывающегося списка выберем значение «По возрастанию».

На рис. 1 приведен вид данного запроса в режиме конструктора.

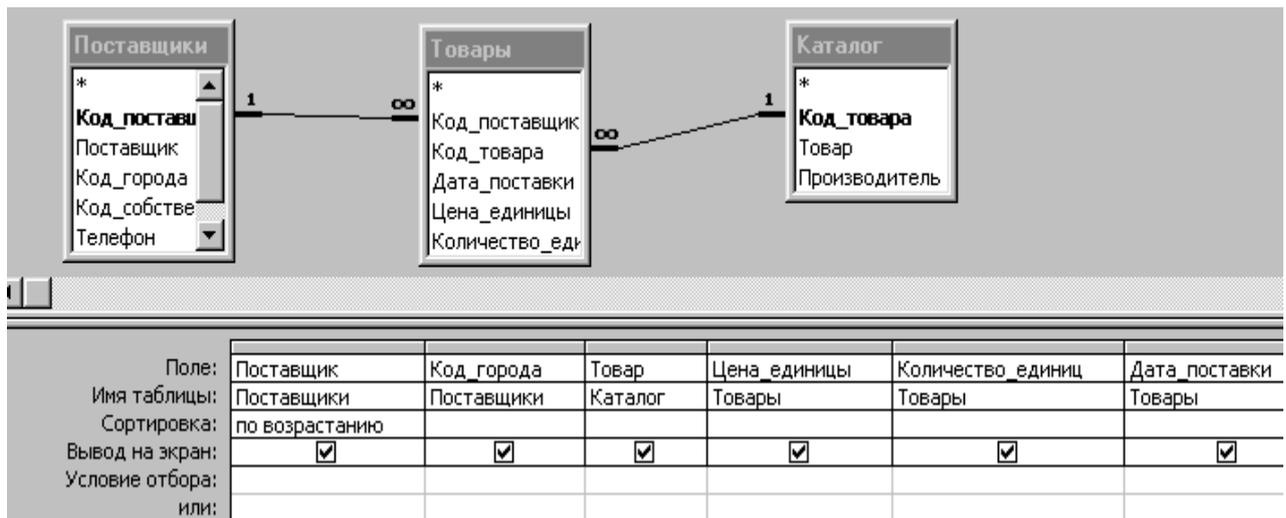


Рисунок 1 – Окно конструктора симметричного объединения без условия

Следует обратить внимание, что для вывода информации о городе поставщика используется поле «Код\_города», справочник «Города» в запрос не включен. Это возможно благодаря подстановке, заданной при создании таблицы «Поставщики».

Результирующий набор записей показан на рис. 2.

Поставщик	Город	Товар	Цена единицы	Количество единиц	Дата поставки
Альфа	Гомель	Материнская плата	690,00	2	07.01.2010
Альфа	Гомель	Монитор	2000,00	12	02.07.2010
Альфа	Гомель	Модуль памяти	90,00	10	02.07.2011
Альфа	Гомель	Клавиатура	320,00	2	05.01.2012
Альфа	Гомель	Материнская плата	870,00	2	05.01.2010
Альфа	Гомель	Монитор	2300,00	1	07.01.2011
Альфа	Гомель	Клавиатура	99,00	1	11.04.2012
Альфа	Гомель	Клавиатура	99,00	25	24.12.2012
Альфа	Гомель	Монитор	3300,00	10	24.12.2012
Альфа	Гомель	Клавиатура	103,95	10	24.12.2012
Альфа	Гомель	Материнская плата	680,00	2	07.01.2010
Альфа	Гомель	Материнская плата	600,00	2	07.06.2012
Альфа	Гомель	Материнская плата	675,00	2	01.01.2012
Альфа	Гомель	Материнская плата	668,00	2	07.06.2012
Альфа	Гомель	Материнская плата	700,00	10	06.02.2012
Альфа	Гомель	Модуль памяти	120,00	2	11.11.2012
Интер-Вест	Донецк	Видеокарта	250,00	4	07.08.2011
Интер-Вест	Донецк	Материнская плата	870,00	2	02.04.2012
Интер-Вест	Донецк	Съемный диск	700,00	10	10.10.2011
Интер-Вест	Донецк	Видеокарта	200,00	2	11.07.2010
Интер-Вест	Донецк	DVD RW	5,00	10	12.05.2010
Интер-Вест	Донецк	CD drive	150,00	5	20.12.2010
Интер-Вест	Донецк	Материнская плата	760,00	1	12.12.2010
Интер-Вест	Донецк	Съемный диск	500,00	12	05.05.2012
МКС	Киев	Материнская плата	524,29	10	01.01.2010
МКС	Киев	Клавиатура	100,00	25	01.01.2011
МКС	Киев	Материнская плата	879,00	10	12.12.2012
МКС	Киев	CD drive	200,00	5	13.02.2011
МКС	Киев	Клавиатура	200,00	2	13.02.2011
МКС	Киев	Клавиатура	180,00	1	01.02.2012
МКС	Киев	DVD RW	5,00	10	14.03.2012

Рисунок 2 – Набор записей, возвращаемый симметричным объединением без условия

**Пример 2.** Вывести сведения обо всех поставках указываемого товара (симметричное объединение с условием отбора по текстовому полю).

Отличие данного запроса от предыдущего состоит в необходимости отбора записей. Чтобы реализовать такой запрос, в строке «Условие отбора» для поля товар запишем имя параметра: [Укажите товар].

**Имя параметра- это текстовая строка, заключенная в квадратные скобки. Имя параметра будет выводиться при запуске запроса. Имя параметра должно отличаться от имени поля, но может включать его.**

На рис.3 показано окно конструктора данного запроса.

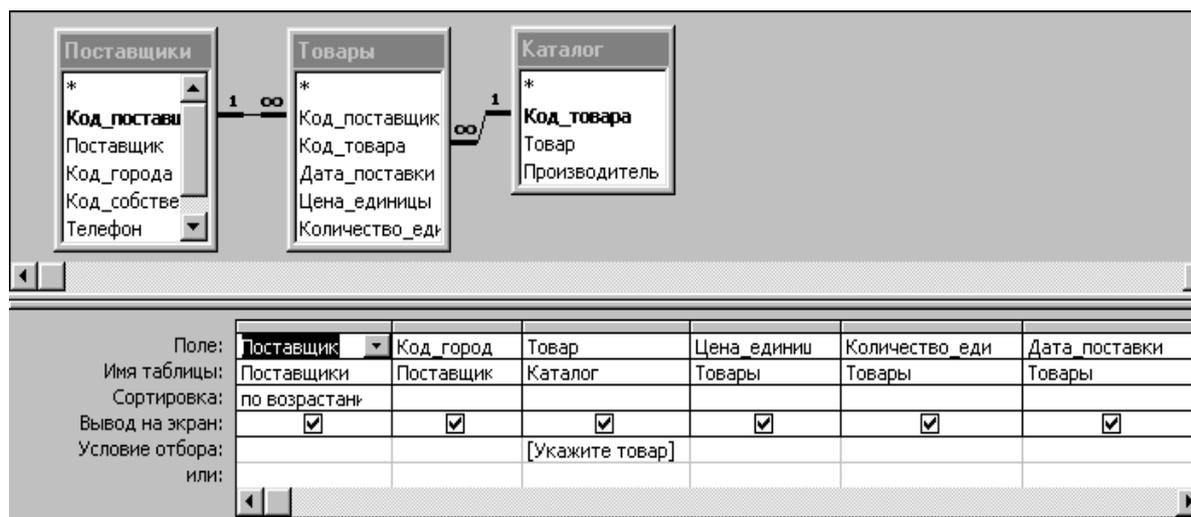


Рисунок 3 – Окно конструктора симметричного объединения с условием

В данном запросе поле «Товар» не только выводится на экран, но и используется для проверки условия отбора. Поэтому, несмотря на наличие подстановки, таблица-справочник обязательно должна быть включена в запрос.

**Пример 3.** Вывести сведения обо всех поставках товаров в заданном году, указав стоимость партии (симметричное объединение с условием отбора по полю типа «Дата/время»).

В данный запрос необходимо включить вычисляемое поле для вывода стоимости партии товаров. Стоимость партии может быть рассчитана как произведение значений полей «Цена\_единицы» и «Количество\_единиц» из таблицы «Товары». Назовем данное вычисляемое поле «Стоимость партии», записав в строке «Поле» бланка QBE следующее выражение:

Стоимость партии: [Товары].[Цена\_единицы]\*[Товары].[Количество\_единиц]

Чтобы в результате запроса выводились только записи о поставках, выполненных в заданном году, создадим вычисляемое поле «Год», в котором из значения поля типа «Дата/время» извлекается целочисленное значение года:

**Год: Year([Дата\_поставки])**

Для этого поля зададим условие отбора с параметром [Введите год].

Длинные выражения удобнее задавать в окне **Область ввода**, которое открывается нажатием **Shift - F2**.

Для вычисляемого поля «Стоимость партии» укажем в окне его свойств формат «Денежный».

На рис.4 показано окно конструктора для данного запроса.

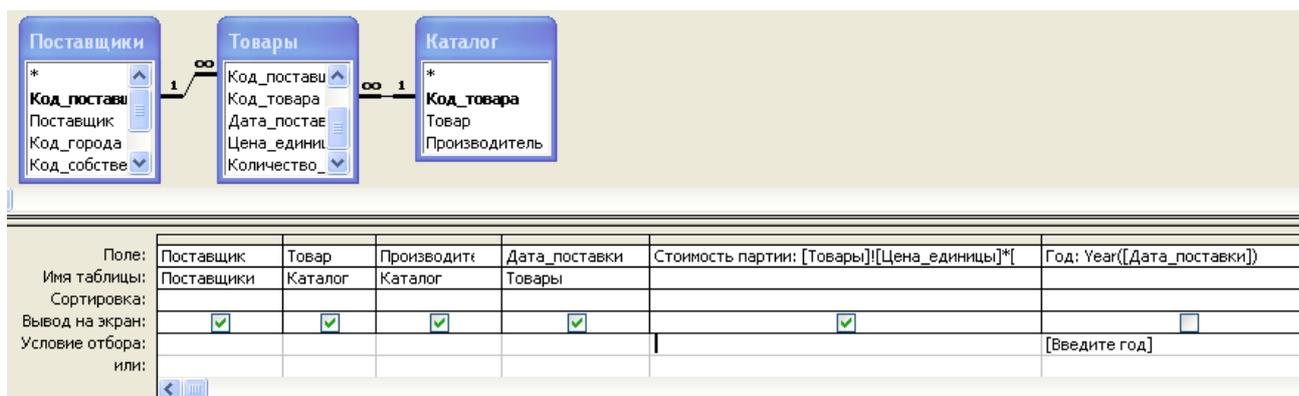


Рисунок 4 – Окно конструктора симметричного объединения с условием

**Пример 4.** Вывести информацию о поставщиках, которые в данный момент не представили товар (левое объединение)

В запросах, рассмотренных в предыдущих примерах, из главной таблицы выбирались только записи, у которых есть подчиненные (то есть данные о поставщиках, не выполнявших поставки, не учитывались). Чтобы определить поставщиков, не выполнявших поставки товаров, нужно изменить тип объединения таблиц. Для этого выполним двойной щелчок по линии связи между таблицами в окне конструктора запроса и в появившемся окне выберем ВТОРОЙ способ объединения (см.рис.5).

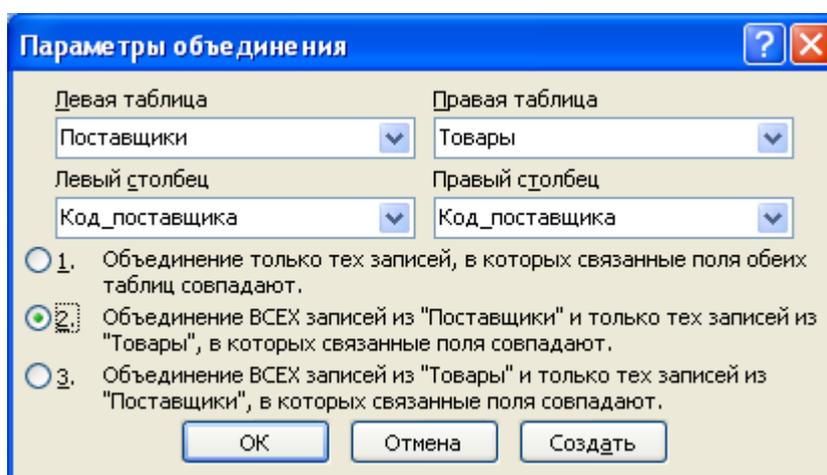


Рисунок 5 – Изменение параметров объединения таблиц

После этого линия связи в окне конструктора изменит свой вид (рис.6). Далее запишем условие отбора записей – пустые значения поля «Код\_поставщика» из таблицы «Товары» (так как значение заведомо пустое, снимем переключатель «Вывод на экран»). Окно конструктора запроса приведено на рис.6, на рис. 7 показан результат его выполнения.

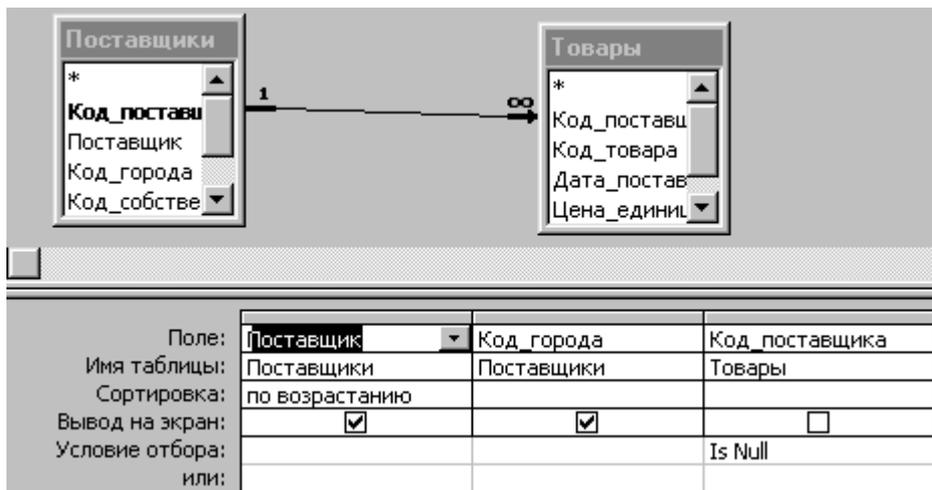


Рисунок 6 - Окно конструктора левого объединения

	Поставщик	Город
	МКС	Донецк

Рисунок 7 – Результат выполнения левого объединения

**Пример 5.** Вывести сведения о поставщиках, которые за указанный период времени не поставили ни одной партии товара (запрос на запросе по принципу левого объединения).

Сначала нужно создать запрос **Товары** за период, определяющий поставщиков, которые сделали поставки товаров в указанный период (см.рис.8).

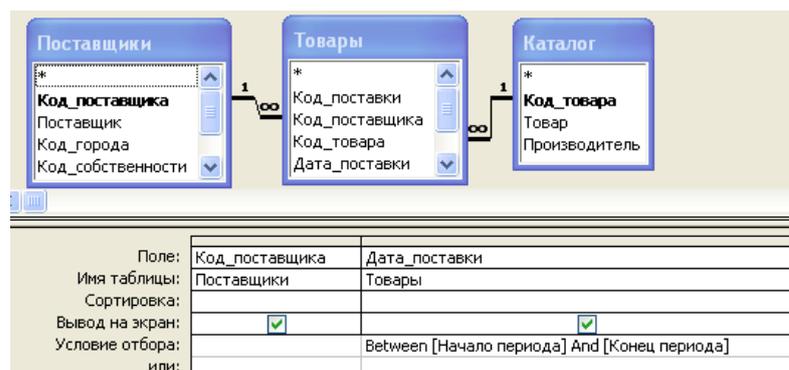


Рисунок 8 – Промежуточный запрос в режиме конструктора

**Для ускорения выполнения рекомендуется в подчиненный запрос включать минимально возможное число полей.**

Теперь на основе таблицы **Поставщики** и запроса **Товары** за период создадим требуемый в примере запрос, используя левое объединение. При

включении таблиц в запрос перейдем на вкладку «Запросы» и выберем запрос «Товары за период» (см.рис.9).

Результат создания запроса показан на рис.10.

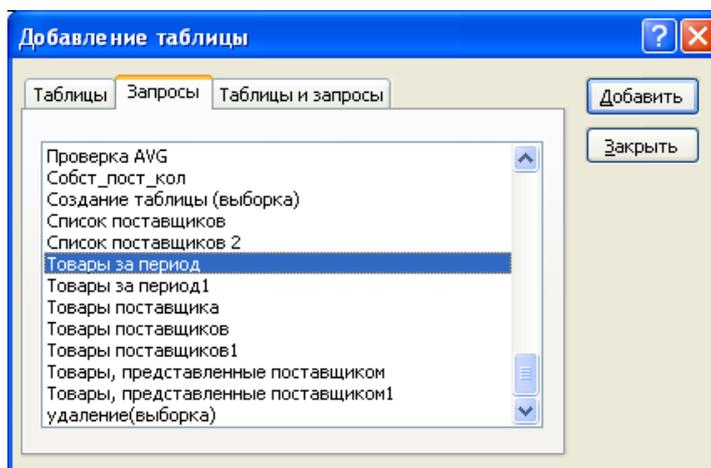


Рисунок 9 – Выбор запроса

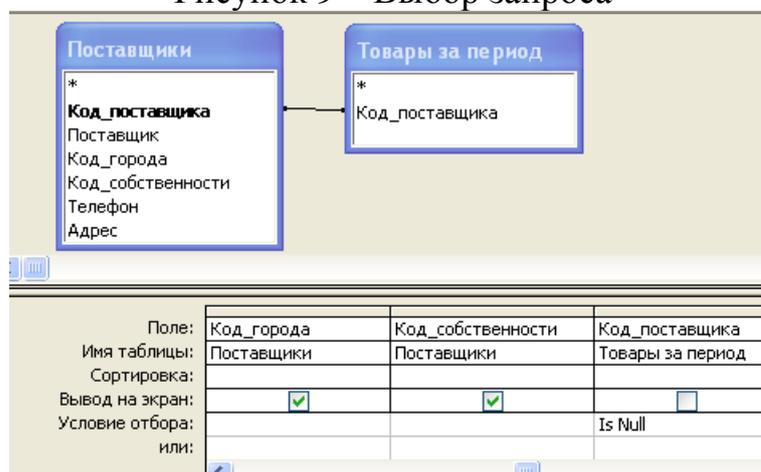


Рисунок 10 – Запрос на запросе в режиме конструктора

## Задание к лабораторной работе

1. Сформулировать следующие виды запросов на выборку:

- симметричное объединение с условием (два запроса с условием отбора по текстовому полю, два - по датам);
- симметричное объединение без условия (3 запроса);
- левое объединение;
- запрос на запросе по принципу левого объединения.

Запросы должны использовать данные из нескольких таблиц, условия отбора следует задавать через параметры.

2. Создать запросы, сформулированные в пункте 1, используя QBE.
3. Дополнить таблицы записями, согласующимися с разработанными запросами.

### **Требования к отчету по лабораторной работе:**

- 1) Сформулировать запросы к своей базе данных (см. вариант задания)
- 2) Привести скриншоты бланка QBE для каждого из запросов по заданию.
- 3) Привести скриншоты работы 1-2 запросов.

### **Контрольные вопросы**

1. Назначение строк бланка QBE.
2. Вычисляемое поле: назначение, синтаксис.
3. Внешние объединения.

## Лабораторная работа 4

**Тема:** Разработка итоговых и перекрестных запросов средствами QBE

**Цель работы:** изучение способов получения итоговых показателей с помощью запросов.

### Методические указания к выполнению работы

#### Итоговые запросы

Если необходимо проанализировать не отдельные записи таблицы, а обобщенные значения по **группам записей**, то нужно создать *итоговый запрос*. Отличием итогового запроса является наличие в бланке QBE строки «Групповая операция», которая включается в бланк нажатием на панели инструментов кнопки «Итоги» с изображением знака суммирования  $\Sigma$ .

В основе выполнения итоговых запросов лежит операция «Группировка». Именно благодаря ей для поля определяются группы записей с одинаковыми значениями; для этих групп будут вычисляться итоговые значения (данное значение назначается по умолчанию). Для вычислений может быть использована одна из итоговых функций Access: **Sum** – вычисляет сумму всех значений поля в каждой группе (для числовых и денежных полей); **Avg** – вычисляет среднее арифметическое значений поля в группе; **Count** – определяет число записей в каждой группе; **Min** – определяет наименьшее значение поля внутри каждой группы; **Max** – определяет наибольшее значение внутри каждой группы; **First** – возвращает значение поля первой записи группы; **Last** – возвращает значение поля последней записи группы.

Рассмотрим примеры создания итоговых запросов.

**Пример 1.** Для каждого вида товара определить количество поставленных партий, общую стоимость партий, среднюю цену за единицу товара (**ИТОГОВЫЙ запрос без условия**).

Для создания такого запроса перейдем к режиму конструктора, предварительно выбрав таблицы, необходимые для выполнения запроса: «Каталог» и «Товары». Нажмем на панели инструментов кнопку «Групповые операции». В данном запросе, как видно из формулировки, содержится одно поле группировки (вид товара) и три вычисляемых поля.

В бланк QBE запроса включим поле «Товар» из таблицы «Каталог», в строке «Групповая операция» выберем для этого поля значение «Группировка», что соответствует выводу значения поля на экран без каких-либо изменений. Укажем, что результаты запроса должны выводиться на экран упорядоченными – в алфавитном порядке названий товаров.

Для определения количества поставленных партий включим в бланк QBE поле «Код\_товара» из таблицы «Товары», выбрав для него групповую операцию Count.

Для расчета средней цены за единицу товара включим в бланк запроса поле «Цена\_единицы», выберем для него групповую операцию Avg. Также для этого поля следует вызвать окно «Свойства поля» и задать значения следующим свойствам: «Формат поля» – Фиксированный, «Число десятичных знаков» – 2.

Для расчета стоимости партии создадим вычисляемое поле с помощью выражения

$$\text{Sum}([\text{Цена\_единицы}] * [\text{Количество\_единиц}]).$$

Обязательно зададим подходящие имена для вычисляемых полей:

**Количество партий:** Count([Код\_товара])

**Общая стоимость:** Sum([Цена\_единицы]\*[Количество\_единиц])

**Средняя цена:** Avg(Цена\_единицы)

Получим запрос, приведенный на рис. 1.

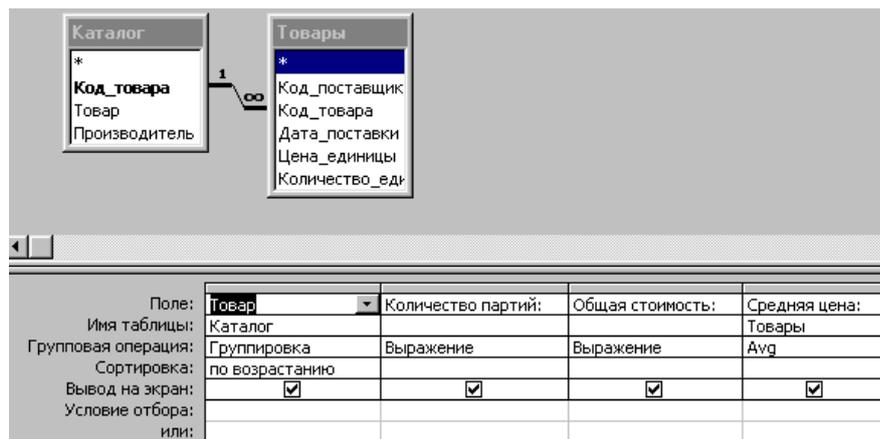


Рисунок 1 – Окно конструктора итогового запроса без условия

**Пример 2.** Для каждого поставщика определить количество партий, поставленных в заданный период, и общую стоимость этих поставок. Указать телефон поставщика (**итоговый запрос с условием на данные**).

Для создания такого запроса перейдем к режиму конструктора, предварительно выбрав таблицы, необходимые для выполнения запроса: «Поставщики» и «Товары». Нажмем на панели инструментов кнопку «Групповые операции». В данном запросе, как видно из формулировки, содержится одно поле группировки (вид товара) и три вычисляемых поля.

В бланк QBE запроса включим поля «Поставщик» и «Телефон» из таблицы «Поставщики», в строке «Групповая операция» выберем для этих полей значение «Группировка».

Для определения количества поставленных партий включим в бланк QBE поле «Код\_поставщика» из таблицы «Товары», выбрав для него групповую операцию Count. Для данного поля зададим имя «Количество партий».

Для расчета стоимости поставок создадим вычисляемое поле с помощью выражения

$$\text{Sum}([\text{Цена\_единицы}] * [\text{Количество\_единиц}]).$$

Для данного поля зададим имя «Общая стоимость поставок».

Для отбора записей включим в бланк запроса поле «Дата\_поставки» из таблицы «Товары». В строке «Групповая операция» для этого поля выберем значение «Условие», а в строке «Условие отбора» запишем:

Between [Начало периода] And [Конец периода].

На рис.2 показано окно конструктора данного запроса.

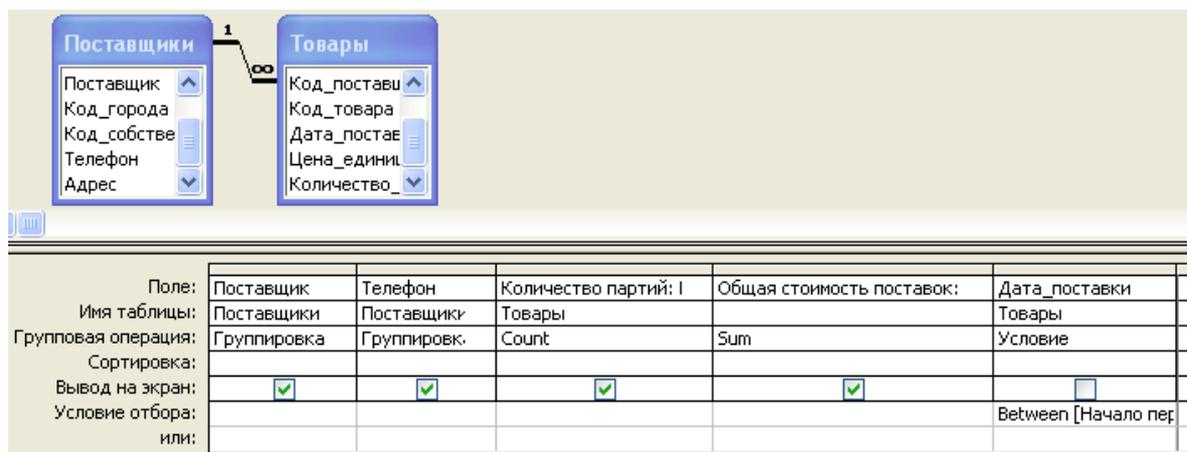


Рисунок 2 – Окно конструктора итогового запроса с условием на данные

**Пример 3.** Для каждого вида товара с общей стоимостью партий, превышающей заданное значение, определить количество поставленных партий, среднюю цену за единицу товара (**итоговый запрос с условием на группы**).

Данный запрос создается так же, как и запрос, приведенный в примере 1. Отличие состоит в необходимости проверки условия по вычисляемому полю «Общая стоимость».

На рис.3 показано окно конструктора для данного запроса.

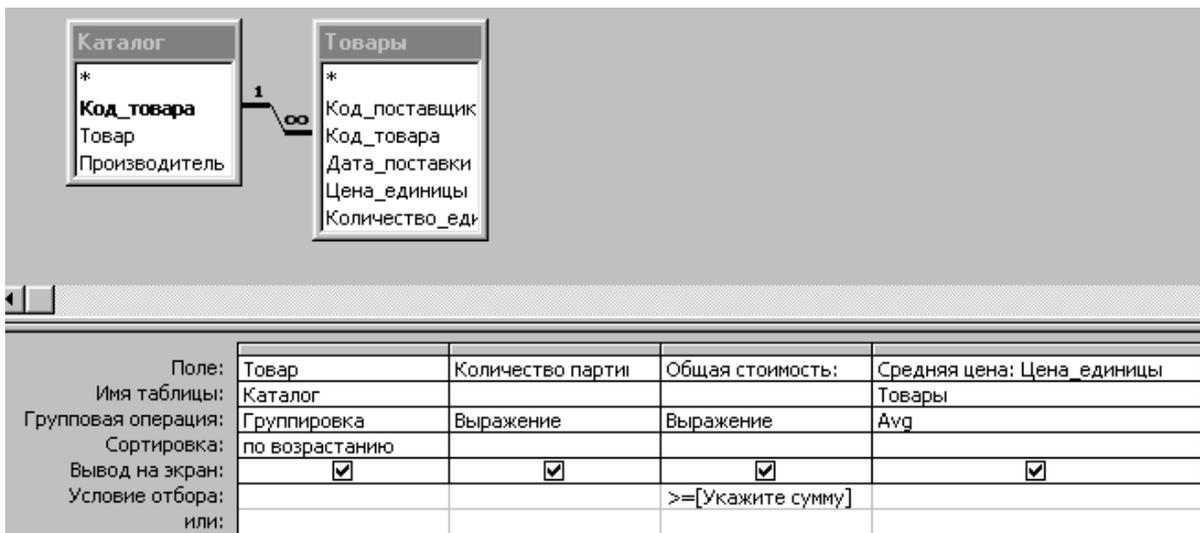


Рисунок 3 – Окно конструктора итогового запроса с условием на группы

*Пример 4. Для поставщиков с количеством партий, поставленных в заданный период, превышающим заданное значение, определить общую стоимость этих поставок. Указать телефон поставщика (итоговый запрос с условием на данные и на группы)*

При создании данного запроса следует поступать так же, как в примере 2. В заключении следует для вычисляемого поля «Количество партий» задать условие отбора:

>[Введите количество].

На рис. 4 приведено окно конструктора данного запроса.

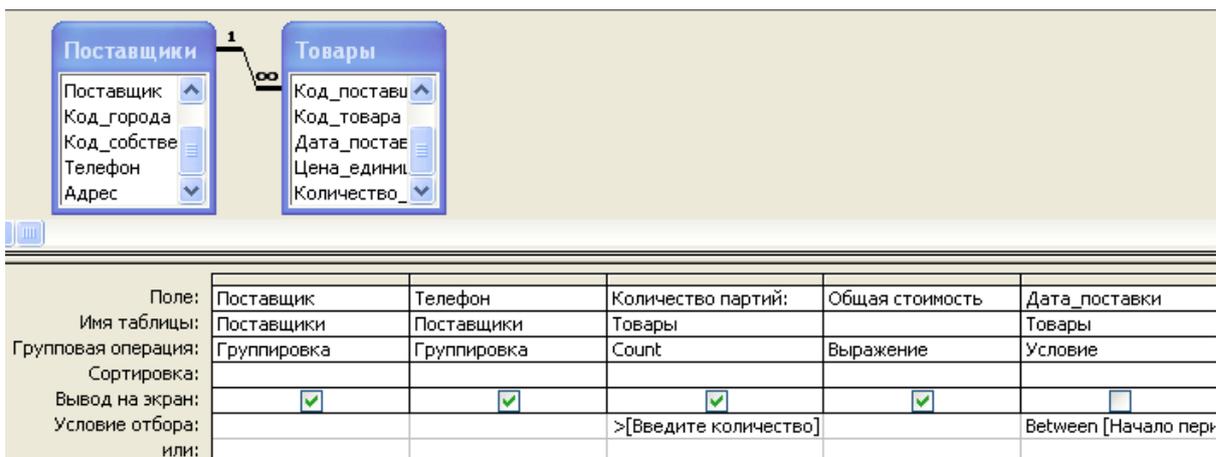


Рисунок 4 - Окно конструктора итогового запроса с условием на данные и группы

## Перекрестные запросы

Перекрестные запросы являются частным случаем итоговых и предназначены для расчета ОДНОЙ итоговой функции с группировкой по значениям двух и более полей, одно из которых выводится в качестве заголовков столбцов, остальные- в качестве заголовков строк.

Рассмотрим примеры создания перекрестных запросов.

**Пример 5.** Для каждого поставщика определить по годам общее количество поставленных партий товаров (перекрестный запрос без условия)

Для создания данного запроса в окне конструктора укажем, что данный запрос является перекрестным. Для этого на вкладке «Конструктор» нажмем соответствующую кнопку:



Включим в запрос таблицы, необходимые для выполнения запроса: «Поставщики» и «Товары». Для полей «Поставщик» и «Код\_города» в строке «Групповая операция» выберем значение «Группировка», а в строке «Перекрестная таблица» - «Заголовки строк». Укажем, что результат выполнения запроса должен быть упорядочен по названию поставщика.

Заголовки столбцов должны показывать годы, когда выполнялись поставки товаров. Создадим вычисляемое поле

Year(Дата\_поставки).

Для этого поля в строке «Групповая операция» выберем значение «Группировка», а в строке «Перекрестная таблица» - «Заголовки столбцов».

Для получения итогового значения в данном запросе включим в него поле «Код\_поставщика». В строке «Групповая операция» выберем для него значение «Count», а в строке «Перекрестная таблица» - «Значение».

На рис. 5 приведено окно конструктора данного запроса.

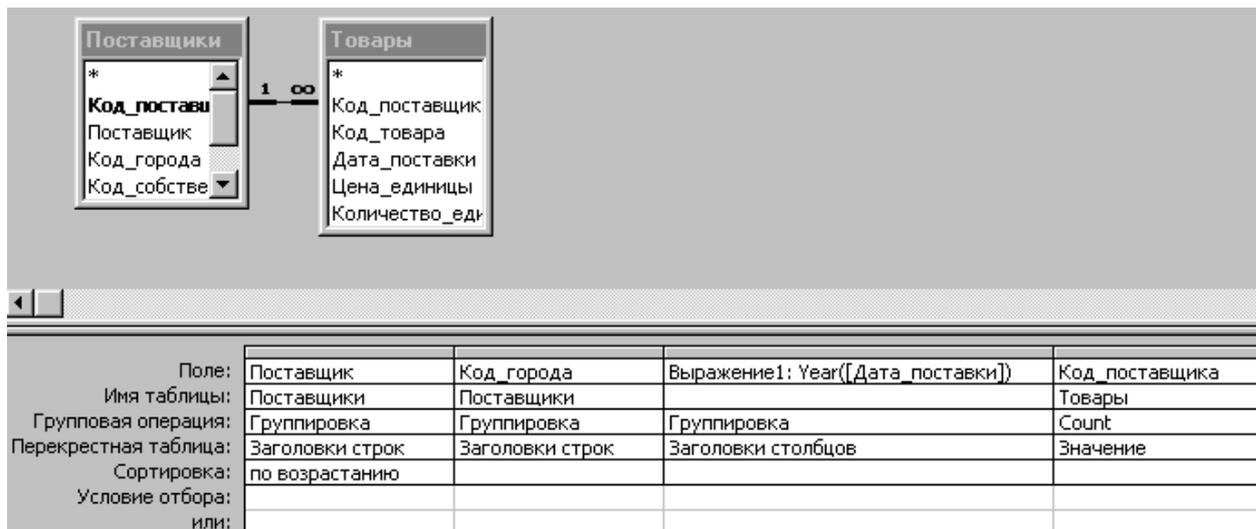


Рисунок 5 – Окно конструктора перекрестного запроса без условия

**Пример 5.** Для каждого поставщика определить общую сумму за поставленные партии товаров по кварталам задаваемого года и общую сумму за год (**перекрестный запрос с условием**).

Включим в запрос таблицы, необходимые для выполнения запроса: «Поставщики» и «Товары», укажем, что запрос является перекрестным. Для поля «Поставщик» в строке «Групповая операция» выберем значение «Группировка», а в строке «Перекрестная таблица» - «Заголовки строк». Укажем, что результат выполнения запроса должен быть упорядочен по названию поставщика.

Для вычисления квартала, к которому относится дата поставки, воспользуемся рассмотренной ранее встроенной функцией. Соответствующее выражение для вычисляемого поля имеет вид:

Выражение1: "Квартал" & DatePart("q"; [Дата\_поставки])

Для этого поля в строке «Групповая операция» выберем значение «Группировка», а в строке «Перекрестная таблица» - «Заголовки столбцов».

Значения, стоящие в ячейках перекрестной таблицы, вычисляются с помощью выражения

Сумма за квартал: Sum([Цена\_единицы]\*[Количество\_единиц]).

В данном случае в строке «Групповая операция» выберем «Выражение», а в строке «Перекрестная таблица» - «Значение».

Чтобы включить в перекрестный запрос сумму по строке (в нашем случае это сумма за год), следует повторно включить в бланк QBE такой же столбец, как и для «Значения», но в строке «Перекрестная таблица» указать для него «Заголовки строк».

Для отбора поставок, осуществленных в заданном году, создадим вычисляемое поле

Year(Дата\_поставки).

В строке «Групповая операция» выберем значение «Условие», в строке «Условие отбора» запишем:

[Укажите год].

Это параметр запроса. Поскольку запрос является перекрестным, укажем явно тип параметра – **целое**.

На рис.6 приведено окно конструктора данного запроса.

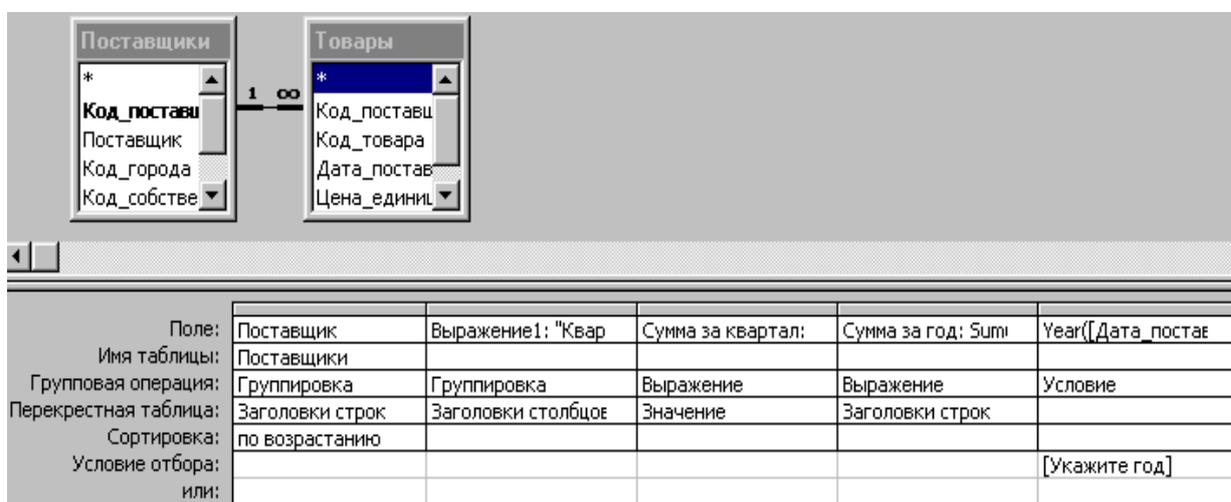


Рисунок 6 – Окно конструктора перекрестного запроса с условием

**Пример 6.** Определить количество поставщиков каждой формы собственности и количество поставок, выполненных поставщиками разных форм собственности (**запрос на запросе по принципу итогового запроса**).

Из-за особенностей выполнения многотабличного запроса невозможно построить запрос, позволяющий получить требуемые показатели. Для вычисления нужно построить запрос на запросе.

Вначале создадим промежуточный запрос «Собст\_пост\_кол», вычисляющий для каждой формы собственности количество поставщиков. На рис. 7 показано окно конструктора этого запроса.

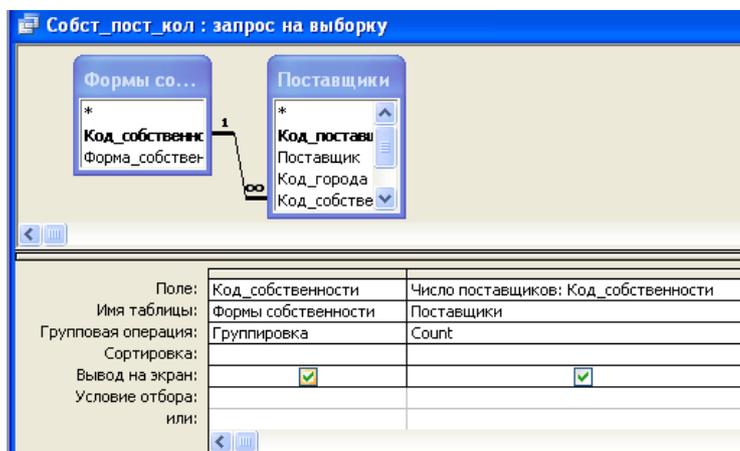


Рисунок 7 – Промежуточный запрос в режиме конструктора

Затем создадим запрос, соответствующий нашей формулировке. Для этого при включении таблиц в запрос перейдем на вкладку «Запросы» и выберем только что созданный запрос. Кроме этого, в запрос необходимо включить таблицы «Товары» и «Товары». Вид запроса в режиме конструктора показан на рис. 8, а на рис.9 приведены результаты его выполнения.

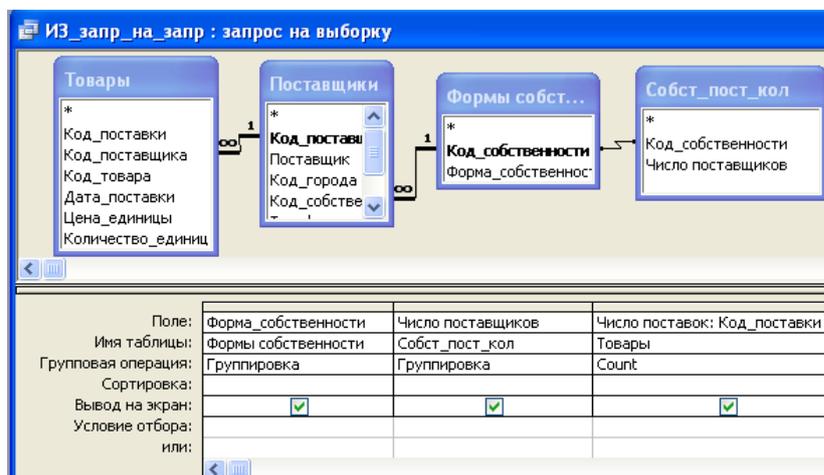


Рисунок 8 – Запрос на запросе в режиме конструктора

Форма собственности	Число поставщиков	Число поставок
Акционерная	1	8
Государственная	1	7
Частная	1	16

Рисунок 9 – Результат выполнения запроса на запросе

## **Задание к лабораторной работе**

1. Сформулировать следующие виды запросов:

- итоговый запрос без условия;
- итоговый запрос с условием на данные;
- итоговый запрос с условием на группы;
- итоговый запрос с условием на данные и на группы;
- перекрестный запрос без условия;
- перекрестный запрос с условием;
- запрос на запросе по принципу итогового запроса.

Запросы должны использовать данные из нескольких таблиц, условия отбора следует задавать через параметры. В результате выполнения запроса на экран должно выводиться не менее трех полей.

2. Создать запросы, сформулированные в пункте 1, используя QBE.

3. Дополнить таблицы записями, согласующимися с разработанными запросами.

### **Требования к отчету по лабораторной работе:**

- 1) Сформулировать запросы к своей базе данных (см. вариант задания)
- 2) Привести скриншоты бланка QBE для каждого из запросов по заданию.
- 3) Привести скриншоты работы 1-2 запросов.

### **Контрольные вопросы**

1. Итоговые запросы: назначение, значения строки «Групповая операция», задание условий отбора.
2. Перекрестный запрос: назначение, совместные значения строк «Перекрестная таблица» и «Групповая операция», задание условий отбора.
3. Особенности использования параметров при создании перекрестных запросов.

## Лабораторная работа 5

**Тема:** Создание запросов на языке SQL.

**Цель работы:** изучение возможностей языка SQL для создания запросов на выборку.

### Методические указания к выполнению работы

Чтобы записать запрос на языке SQL, следует создать запрос в режим конструктора, отказаться от выбора базовых таблиц, вызвать с помощью правой кнопки «мыши» контекстное меню и выбрать пункт **SQL**.

Запрос на языке SQL, как правило, записывается с помощью одной команды. Для всех запросов, выполняющих выборку данных, включая перекрестные и итоговые запросы, эта команда является вариантом одной инструкции языка SQL – инструкции **SELECT**. Данная инструкция может состоять из нескольких разделов, их набор определяется самим запросом.

В **SQL** для запроса на выборку используется инструкция **SELECT**, обобщенный синтаксис которой имеет вид:

```
SELECT <список - выбора>  
FROM <список - таблиц>  
[ WHERE <условие – отбора - записей>]  
[ GROUP BY <список – полей – для - группировки>]  
[ HAVING <условие – отбора - групп>]  
[ ORDER BY <условия - сортировки>;
```

Здесь **FROM**, **WHERE**, **GROUP BY**, **HAVING**, **ORDER BY** – предложения **SQL**.

**FROM** определяет список таблиц, на основе которых строится запрос (обязательное предложение).

**WHERE** указывает условие, которому должны удовлетворять записи результирующего набора.

**GROUP BY** определяет, как должны быть сформированы группы для *итогового* запроса (то есть указывает список полей, по одинаковым значениям которых будет выполняться группировка записей).

**HAVING** используется для указания условия отбора тех групп, которые должны быть включены в результат запроса.

**ORDER BY** указывает условия упорядоченности строк в результирующем наборе.

<список - выбора> определяет, какие поля (столбцы) должен иметь результирующий набор записей (наличие <списка> в SQL – запросе обязательно).

Рассмотрим, как записываются на языке SQL различные запросы

### Запросы на выборку

При записи простых запросов на выборку в инструкции **SELECT** могут присутствовать только предложения **FROM, WHERE, ORDER BY**. Перед списком выбора может быть записан предикат, дополнительно ограничивающий результирующий набор записей. Если в условии отбора выполняется сравнение с параметрами, то перед инструкцией **SELECT** указываются типы этих параметров с помощью инструкции **PARAMETERS/**

**Пример 1.** Вывести сведения о поставщиках и их товарах, поставленных в указанный период.

Для записи данного запроса в начале опишем два параметра. Сравняться со значениями параметров будут значения поля Дата/Время, поэтому укажем для параметров соответствующий тип данных SQL – **DateTime**. Результирующий набор записей будем выводить упорядоченным по дате поставки. Инструкция, реализующая данный запрос, будет выглядеть так:

```
PARAMETERS [Начало периода] DateTime,  
            [Конец периода] DateTime;  
SELECT Поставщик, Город, Товар, Дата_поставки
```

```

FROM (Города INNER JOIN Поставщики
ON Города.Код_города = Поставщики.Код_города )
INNER JOIN (Каталог INNER JOIN Товары
ON Каталог.Код_товара = Товары.Код_товара)
ON Поставщики.Код_поставщика = Товары.Код_поставщика
WHERE Дата_поставки Between[Начало периода] And
[Конец периода]
ORDER BY Поставщик, Товар, Дата_поставки;

```

**Пример 2.** Вывести сведения о пяти поставщиках, которые в течение заданного периода поставили партии товара с наибольшей стоимостью.

Данный запрос отличается от предыдущего двумя особенностями. Во-первых, в него следует включить вычисляемое поле «Стоимость партии» - произведение значений полей «Цена\_единицы» и «Количество\_единиц». Чтобы вывести сведения о поставщиках, представивших пять партий с наибольшей стоимостью, следует упорядочить результирующий набор по убыванию значений вычисляемого поля (**ORDER BY 7 DESC** – сортировка набора записей по седьмому полю из списка выбора) и взять из него только 5 первых записей (это реализуется с помощью предиката **TOP 5**).

Текст запроса:

```

PARAMETERS [Начало периода] DateTime,
[Конец периода] DateTime;
SELECT TOP 5 Поставщик, Город, Товар, Дата_поставки,
Цена_единицы, Количество_единиц,
Цена_единицы*Количество_единиц
AS [Стоимость партии]
FROM (Города INNER JOIN Поставщики
ON Города.Код_города = Поставщики.Код_города )
INNER JOIN ( Каталог INNER JOIN Товары
ON Каталог.Код_товара = Товары.Код_товара)
ON Поставщики.Код_поставщика = Товары.Код_поставщика
WHERE Дата_поставки Between[Начало периода] And
[Конец периода]
ORDER BY 7 DESC;

```

**Пример 3.** Определить, какие виды товаров поставил каждый поставщик в заданном году.

В данном запросе обязательно следует использовать предикат **DISTINCT**, поскольку один и тот же товар мог поставляться одним и тем же поставщиком неоднократно.

Текст запроса:

```
PARAMETERS [Введите год] Short;  
SELECT DISTINCT Поставщик, Товар  
FROM Поставщики INNER JOIN (Каталог  
    INNER JOIN Товары ON Каталог.Код_товара =  
    Товары.Код_товара) ON Поставщики.Код_поставщика =  
    Товары.Код_поставщика  
WHERE year(Дата_поставки)=[Введите год]  
ORDER BY Поставщик,Товар;
```

### Итоговые запросы

Для итоговых запросов инструкция **SELECT** должна дополнительно содержать предложения **GROUP BY** и **HAVING**. Эти предложения записываются после предложения **WHERE**, перед предложением **ORDER BY**. В списке выбора такой инструкции должны быть перечислены поля группировки и вычисляемые поля, получаемые с помощью итоговых функций. В разделе **GROUP BY** повторно перечисляются поля группировки. При необходимости в разделе **HAVING** записывается условие на группы.

**Пример 4.** Определить поставщиков, которые в заданный период поставили товаров на сумму, превышающую заданную. Вывести: наименование поставщика, общую сумму за товары, количество партий товаров.

В данном запросе необходимо проверять два условия. Первое условие – дата поставки должна принадлежать заданному периоду. Это условие на данные, оно записывается в разделе **WHERE**. Второе условие – общая стоимость поставок должна превышать некоторую сумму. Это – условие на группы, оно должно быть записано в предложении **HAVING**.

Текст запроса:

```

PARAMETERS [Начало периода] DateTime,
              [Конец периода] DateTime,
              [Укажите сумму] IEEESingle;
SELECT Поставщик,
        Sum(Цена_единицы*Количество_единиц) AS [Стоимость],
        Count(Товары.Код_поставщика) AS [Количество партий],
FROM Поставщики INNER JOIN Товары
      ON Поставщики.Код_поставщика = Товары.Код_поставщика
WHERE Дата_поставки Between[Начало периода] And
              [Конец периода]

GROUP BY Поставщик
HAVING Sum(Цена_единицы*Количество_единиц)>[Укажите сумму]
ORDER BY 2 DESC;

```

### Перекрестные запросы

Для записи перекрестного запроса используется следующая модификация инструкции **SELECT**:

```

PARAMETERS список параметров;
TRANSFORM значение
SELECT заголовки строк
FROM список таблиц
WHERE условие отбора
GROUP BY заголовки строк
ORDER BY поля сортировки
PIVOT заголовки столбцов in (порядок);

```

Выражение, содержащее итоговую функцию для получения значения в ячейках перекрестной таблицы, записывается в предложении **TRANSFORM**. Как видно, заголовки строк перечисляются дважды: в списке выбора и в предложении **GROUP BY**. Условие отбора при выполнении перекрестного запроса проверяется только для данных, не для групп. Предложения **PARAMETERS** и **WHERE** являются необязательными.

**Пример 5.** Определить для каждого поставщика количество поставленных партий товаров по месяцам задаваемого года и общее количество партий за год.

Особенность данного запроса заключается в порядке вывода на экран столбцов. Названия месяцев года, идущие в естественном порядке, не располагаются ни в алфавитном порядке, ни в порядке, обратном алфавитному. Поэтому укажем явно порядок, в котором должны располагаться столбцы перекрестного запроса, с помощью предложения **IN**.

Текст запроса:

```
PARAMETERS [Укажите год] Short;  
TRANSFORM Count(Товары.Код_поставщика) AS [За месяц]  
SELECT Поставщик,  
           Count(Товары.Код_поставщика) AS [За год]  
FROM Поставщики INNER JOIN Товары ON  
       Поставщики.Код_поставщика = Товары.Код_поставщика  
WHERE Year(Дата_поставки) = [Укажите год]  
GROUP BY Поставщик  
ORDER BY Поставщик  
PIVOT Format(Дата_поставки, "mmm")  
       IN ("янв", "фев", "мар", "апр", "май", "июн", "июл",  
          "авг", "сен", "окт", "ноя", "дек");
```

### Подчиненные запросы

Язык SQL поддерживает возможность включать в условия отбора значения, возвращаемые другим (подчиненным) запросом. В случае, если подчиненный запрос возвращает более одного значения, необходимо использовать предикат **ALL** или **ANY**.

**Пример 5.** Определить поставщиков из заданного города

В данном запросе коды поставщиков из заданного города возвращаются с помощью подчиненного запроса. Использование подчиненного запроса позволяет избежать объединения таблиц.

Текст запроса:

```
PARAMETERS [Введите город] Text;  
SELECT Поставщик, Телефон  
FROM Поставщики  
WHERE Поставщики.Код_города IN  
       (SELECT Города.Код_города  
       FROM Города  
       WHERE Город=[Введите город]);
```

**Пример 6.** Определить товары, у которых цена единицы меньше средней цены всех товаров.

Для проверки условия отбора в данном запросе вначале должен выполняться подчиненный запрос, определяющий среднюю цену товаров.

Текст запроса:

```
SELECT Товар,Дата_поставки, Цена
FROM Товары
WHERE Цена_единицы<
                (SELECT Avg (Цена_единицы)
                FROM Товары)
ORDER BY 3;
```

**Пример 7.** Определить партии товаров, в которых цена единицы товара превышает цену указываемого товара (один и тот же товар может иметь в разных партиях различные цены).

В данном случае подчиненный запрос может возвращать множество значений, поэтому в условии отбора используется предикат **ALL**.

Текст запроса:

```
PARAMETERS [Укажите код товара] Long;
SELECT Товары.Код_товара, Товар,
        Цена_единицы, Дата_поставки
FROM Каталог INNER JOIN Товары
        ON Каталог.Код_товара = Товары.Код_товара
WHERE Цена_единицы > ALL
        (SELECT Distinct Цена_единицы
        FROM Товары
        WHERE Код_товара = [Укажите код товара])
ORDER BY Код_товара;
```

**Пример 8.** Найти производителей товаров, которые ни разу не поставлялись.

В данном запросе необходимо найти записи в таблице «Каталог», у которых нет связанных записей в таблице «Товары». Поскольку один и тот же

производитель может выпускать различные товары, воспользуемся предикатом **DISTINCT**.

Текст запроса:

```
SELECT DISTINCT Производитель  
FROM Каталог  
WHERE Код_товара <> ANY  
      (SELECT Код_товара  
      FROM Товары);
```

### **Задание к лабораторной работе**

Создать на языке SQL следующие виды запросов:

- выборка с условием;
- симметричное объединение с предикатом **DISTINCT**;
- симметричное объединение либо итоговый запрос с предикатом **TOP n**;
- итоговый запрос с условием на данные и на группы,
- перекрестный запрос с условием,
- подчиненный запрос (использовать операции сравнения и один из предикатов **EXISTS**, **ANY**, **ALL**, **IN**).

#### **Требования к выполнению**

1. SQL – запросы не должны совпадать с имеющимися QBE – запросами.
2. Условия отбора должны содержать параметры параметры.

Требование к отчету по лабораторной работе:

#### **В отчет по лабораторной работе включить:**

- 1) Формулировки SQL - запросов по заданию и по Вашей предметной области.
- 2) Привести текст SQL - запросов.
- 3) Привести результаты работы всех SQL - запросов.

### Контрольные вопросы

1. Создание на SQL внешних объединений (назначение опций **INNER**, **LEFT**, **RIGHT**); назначение предикатов **DISTINCT**, **ALL**, **TOP n**, **TOP n PERCENT**.
2. Создание итоговых и перекрестных запросов.
3. Подчиненные запросы (назначение и синтаксис предикатов сравнения, **EXISTS**, **IN**, **ANY**, **ALL**).
4. Назначение и синтаксис инструкции **SELECT**.
5. Оператор **JOIN** (назначение опций **INNER**, **LEFT**, **RIGHT**).
6. Типы **SQL** – запросов, которые не могут быть созданы средствами **QBE**.

## Лабораторная работа 6

**Тема:** Создание запросов на изменение.

**Цель работы:** изучение возможностей системы по изменению данных в БД.

### Методические указания к выполнению работы

#### Определение запроса

Запрос является одним из наиболее важных объектов Microsoft Access. С помощью запросов можно не только выбирать данные из БД, но и изменять их. Для этих целей служат специальные запросы, называемые *запросами на изменение*.

*Запрос на изменение* – это инструмент, который позволяет выполнить следующие действия с данными таблиц:

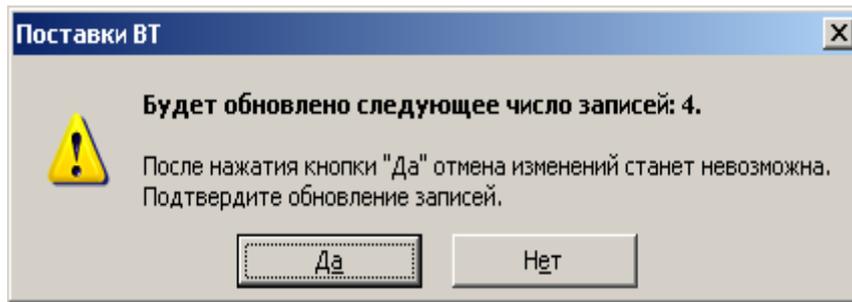
- обновление данных в записи (группе записей),
- удаление записей из таблицы,
- добавление записей из одной таблицы в другую,
- создание таблицы.

В окне БД запросы на изменение помечаются специальными значками с восклицательным знаком:

 - обновление,  - удаление,  - добавление,  - создание таблицы.

Подобно запросам на выборку, запросы на изменение создают динамический набор данных, который можно просмотреть в режиме таблицы (кнопка **Режим таблицы**).

Запуск запроса на выполнение осуществляется с помощью кнопки **Запуск** (восклицательный знак). Появляется диалоговое окно, в котором указывается количество изменяемых записей и запрашивается подтверждение на изменение. Для подтверждения изменения нужно щелкнуть по кнопке **Да**, выбор кнопки **Нет** отменяет эту операцию (записи не будут изменены). Например, при запуске запроса на обновление на экран выведется следующее окно:



Выводимое на экран окно подтверждения замедляет выполнение приложения и может мешать пользователю. Поэтому после отладки всех запросов на изменение вывод этого окна можно отменить. Для этого нужно выполнить команду **Сервис** → **Параметры** и в окне диалога **Параметры** на вкладке **Правка и поиск** в секции **Подтверждение** снять флажок с опции **Запросов на изменение**.

**Следует помнить**, что исправления, вносимые в БД запросами на изменение, необратимы. Поэтому обязательно нужно создавать резервную копию таблицы, записи которой подлежат изменению.

### **Запрос на обновление**

Для изменения значений полей отдельной записи может быть достаточно *режима таблицы*. Если же необходимо произвести одни и те же изменения в группе записей, то нужно создать *запрос на обновление*.

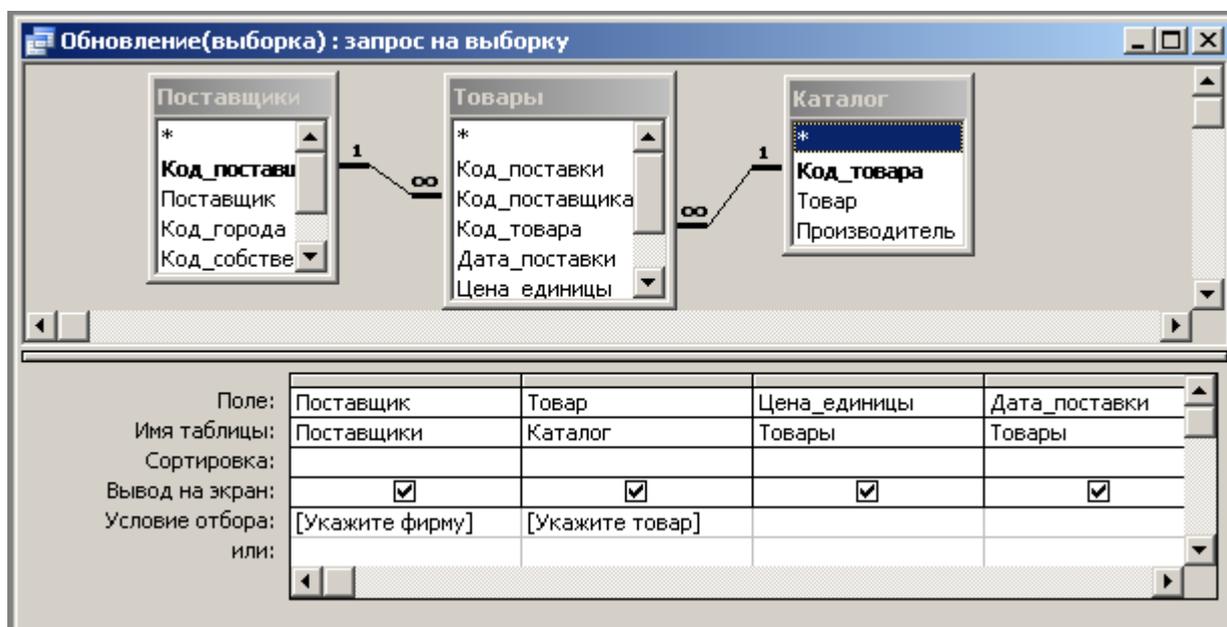
#### **Пример 1.**

**Некоторая фирма – поставщик желает снизить (повысить) цену на указываемый товар (коэффициент изменения задается).**

Для составления такого запроса выполним следующие действия.

1. Создадим резервную копию изменяемой таблицы (таблица **Товары**).

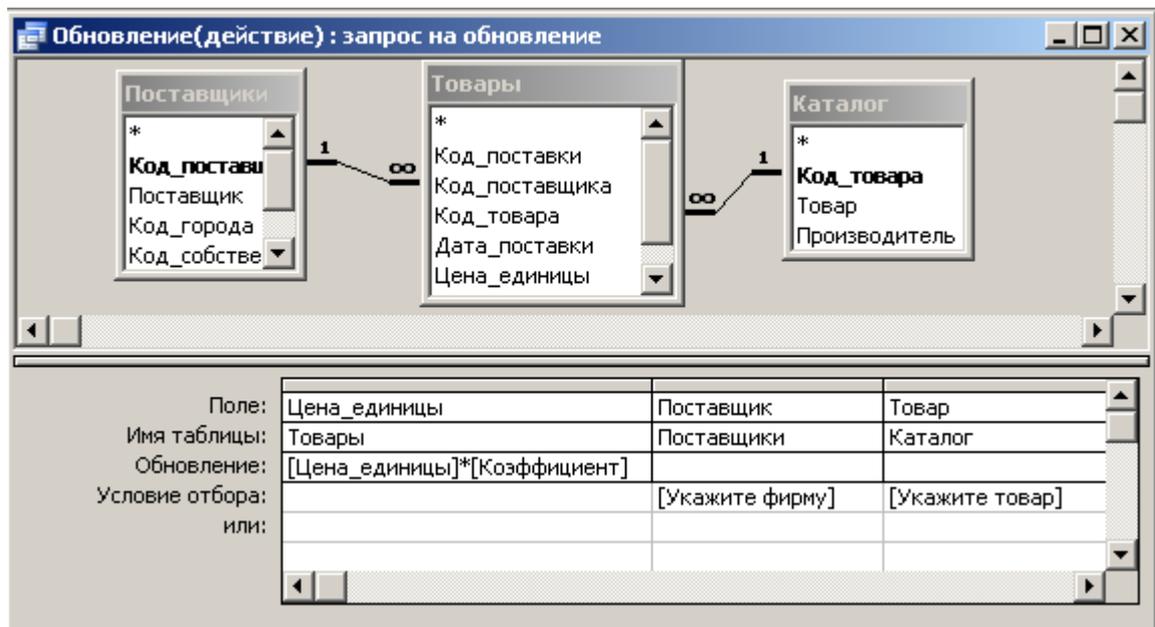
2. Создадим запрос на выборку с условием, позволяющим найти все подлежащие обновлению записи (не забудем указать при этом типы параметров запроса):



3. Просмотрим в режиме таблицы результат запроса на выборку с тем, чтобы убедиться, что выбраны нужные записи.

4. Сделаем копию запроса на выборку (для контроля над операцией обновления).

5. В режиме конструктора преобразуем запрос на выборку в запрос на обновление, выполнив команду **Запрос → Обновление** или используя кнопку **Тип запроса**. При этом изменится заголовок запроса, в бланке QBE исчезнут строки **Сортировка** и **Вывод на экран**, добавится строка **Обновление**. Далее в строку **Обновление** вводим выражение замены для обновляемого поля:



6. Проверим результаты выполнения запроса, воспользовавшись копией запроса на выборку и копией таблицы **Товары**.

**SQL** – инструкция, соответствующая рассмотренному запросу на обновление, имеет следующий вид:

**PARAMETERS** Коэффициент **Ieesingle**,  
 [Укажите фирму] **text** ( 255 ), [Укажите товар] **text** ( 255 );  
**UPDATE** Поставщики **INNER JOIN** (Каталог **INNER JOIN** Товары  
**ON** Каталог.Код\_товара = Товары.Код\_товара)  
**ON** Поставщики.Код\_поставщика = Товары.Код\_поставщика  
**SET** Товары.Цена\_единицы = [Цена\_единицы]\*[Коэффициент]  
**WHERE** (Поставщики.Поставщик = [Укажите фирму])  
**AND** (Каталог.Товар = [Укажите товар]);

**Замечания:**

1. В запросе на обновление можно изменить сразу несколько полей, в том числе и поле (поля), которое используется для условия отбора. При этом для вычисления нового значения некоторого поля можно использовать значения других полей.

2. Если было изменено поле, содержащее условие отбора, для проверки изменений необходимо и в запросе на выборку (копии) изменить условие отбора новым.

3. Если для обновляемого поля установлено свойство **Условие на значение**, то вновь вводимое значение не должно нарушать это условие.

4. Для обновления значения первичного ключа таблицы не нужно создавать специальный запрос: Access обеспечивает выполнение такого изменения автоматически. Кроме того, если при установлении связи между таблицей–отцом и таблицей–сыном включена опция **Каскадное обновление связанных полей**, то Access выполнит системный запрос, с помощью которого обновляются значения внешнего ключа всех связанных записей в таблице-сыне. Однако следует помнить, что изменять значение может только естественный первичный ключ.

### **Запрос на создание таблицы**

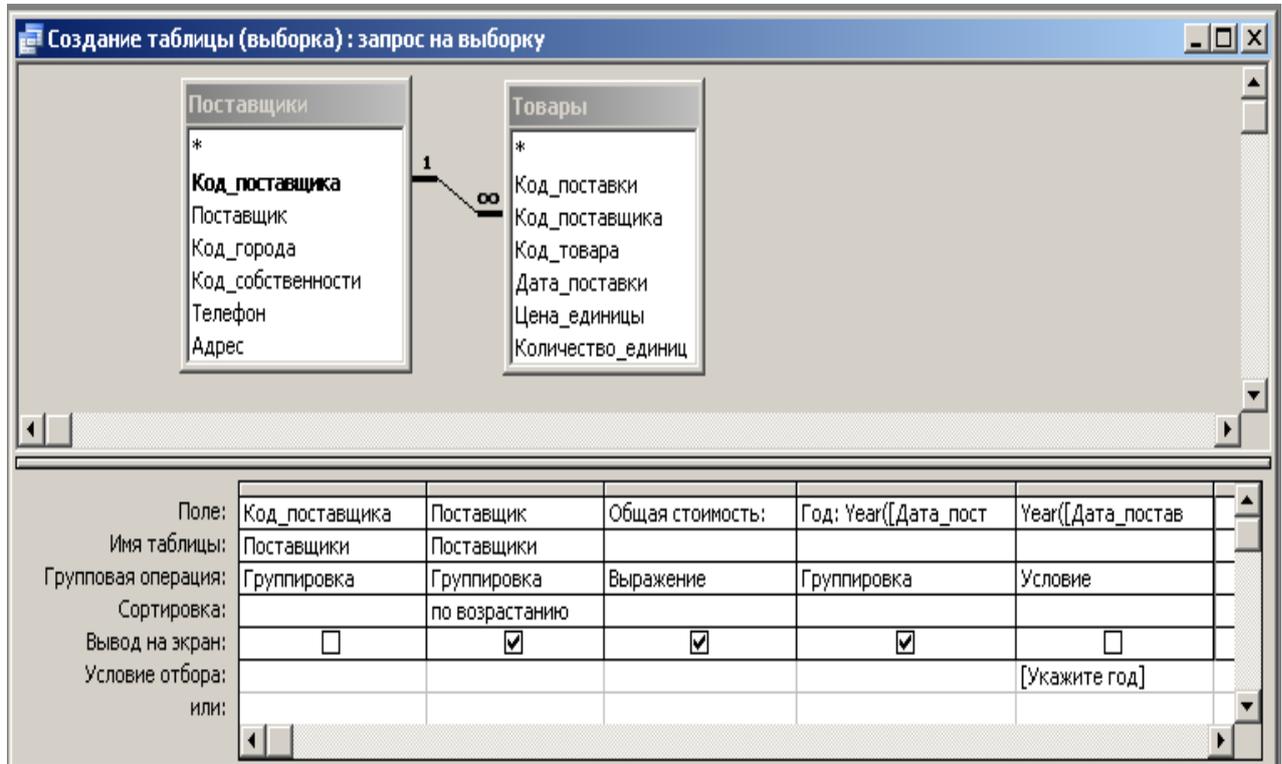
При выполнении запроса на выборку создается *динамический набор записей*, который при выводе на экран выглядит как таблица, но ею не является. Этот набор является временным и не хранится в БД. После закрытия запроса набор “рассыпается”. Для того чтобы сохранить выбранные из таблиц данные, нужно использовать специальный запрос – *запрос на создание таблицы*. С его помощью результаты выборки помещаются в новую таблицу, структура которой определяется структурой выбранных записей.

#### **Пример 2.**

**Вычислить для каждого поставщика общую стоимость товаров, поставленных в указанном году, и представить полученные данные в виде таблицы.**

Для реализации поставленной задачи выполним такую последовательность действий.

1. Создадим сначала следующий итоговый запрос:



Общая стоимость: **Sum**(Товары!Цена\_единицы\*Товары!Количество\_единиц)

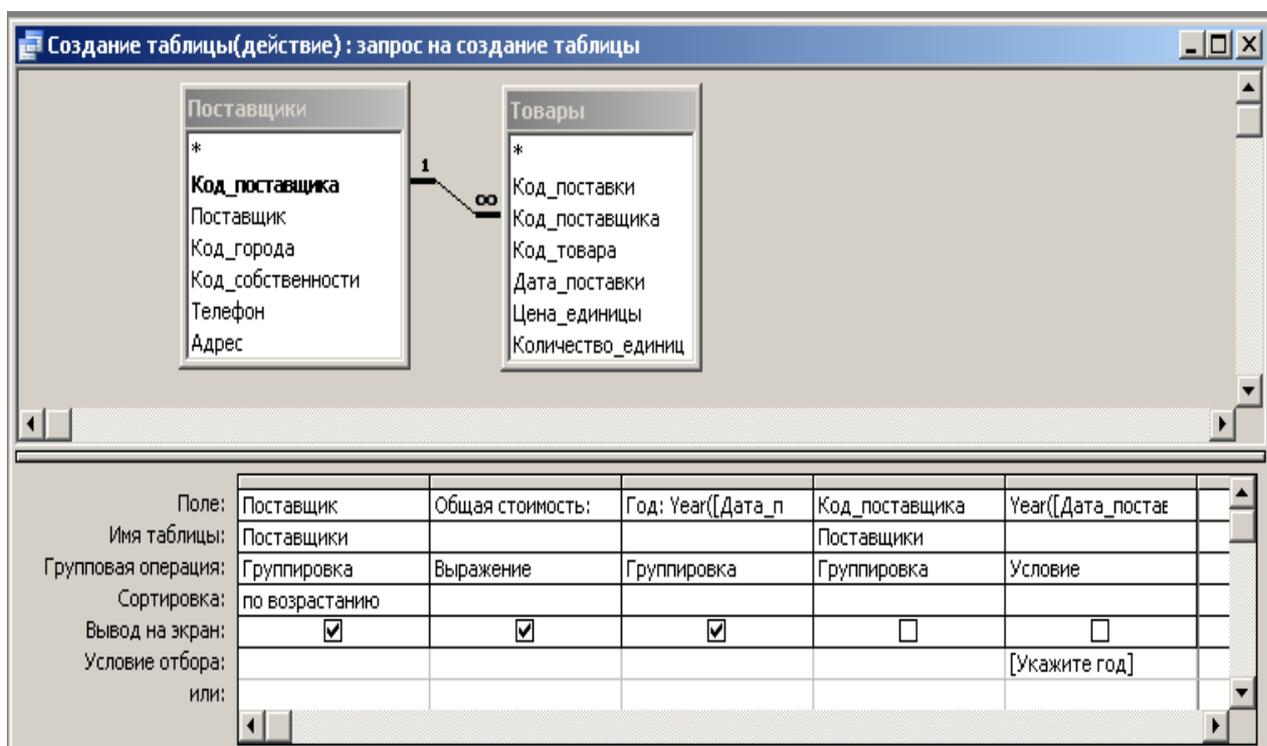
Для полей, которые должны быть скопированы в новую таблицу, обязательно нужно установить опцию **Вывод на экран**. У ключевого поля **Код\_поставщика** снимем флажок, так как нет смысла заносить это поле в новую таблицу (но в формировании результатов запроса это поле должно участвовать). Не забудем также указать тип параметра запроса.

2. Просмотрим в режиме таблицы результаты созданного запроса.

3. Сделаем копию итогового запроса.

4. В режиме конструктора преобразуем запрос на выборку в *запрос на создание таблицы*, выполнив команду **Запрос** → **Создание таблицы** или воспользовавшись кнопкой **Тип запроса**. На экран выводится окно диалога, в котором нужно указать имя таблицы и определить, где она должна быть создана (в текущей или в другой БД).

При выборе имени новой таблицы следует помнить, что в базе данных Access не должно быть таблицы и запроса с одинаковыми именами.



5. Выполним полученный запрос с помощью кнопки **Запуск**. Access создаст записи для новой таблицы и проинформирует об их количестве (выводит на экран окно подтверждения). По кнопке **Да** записи помещаются в созданную таблицу.

6. Проверим результаты запроса, переключившись на вкладку **Таблицы**.

7. Редактируем структуру новой таблицы, так как Access копирует при создании только основные свойства: типы данных и размеры полей. Можно,

например, изменить названия полей на более содержательные, задать формат вывода, создать, если нужно, первичный ключ.

Для запроса **Создание таблицы(действие)** генерируется следующая инструкция **SQL**:

```
PARAMETERS [Укажите год] short;  
SELECT Поставщики.Поставщик,  
    Sum(Товары!Цена_единицы*Товары!Количество_единиц) AS [Общая  
    стоимость], Year([Дата_поставки]) AS Год  
INTO [Итоговые данные]  
FROM Поставщики INNER JOIN Товары  
ON Поставщики.Код_поставщика = Товары.Код_поставщика  
WHERE Year([Дата_поставки])=[Укажите год]  
GROUP BY Поставщики.Поставщик, Year([Дата_поставки]),  
    Поставщики.Код_поставщика  
ORDER BY Поставщики.Поставщик;
```

### **Замечания:**

1. Сохранение результатов запроса в виде таблицы полезно в том случае, если нужно долго хранить полученную информацию (например, итоговую) после удаления из БД данных, на которых базировался запрос.

2. Если запрос на создание таблицы используется повторно, то будет создаваться новая таблица и удаляться старая.

## **Запрос на добавление**

*Запрос на добавление* позволяет скопировать выбранные записи и вставить их в другую таблицу (таблицу - приемник). Структуру этой таблицы нужно определить заранее, причем добавляемые данные должны соответствовать типам полей таблицы – приемника. Поле **Счетчик** копировать не следует, так как при этом возможно дублирование значений первичного ключа. Кроме того, в новой таблице это поле не имеет смысла.

Примером использования запроса на добавление является извлечение устаревших данных из рабочей таблицы и копирование их в архивную таблицу.

Запрос на добавление удобно использовать также для накапливания периодически получаемых итоговых значений. В примере 2, например, требовалось вычисление итоговых значений по поставкам товаров за указываемый год. Если же нужно накапливать такие данные за несколько лет, то следует создать отдельную таблицу и вставлять в нее итоговые данные за каждый год. Такое получение итогов будет более эффективным, так как работа с вычисленными данными, сохраненными в таблице, выполняется быстрее, чем подсчитывание их каждый раз с помощью запроса.

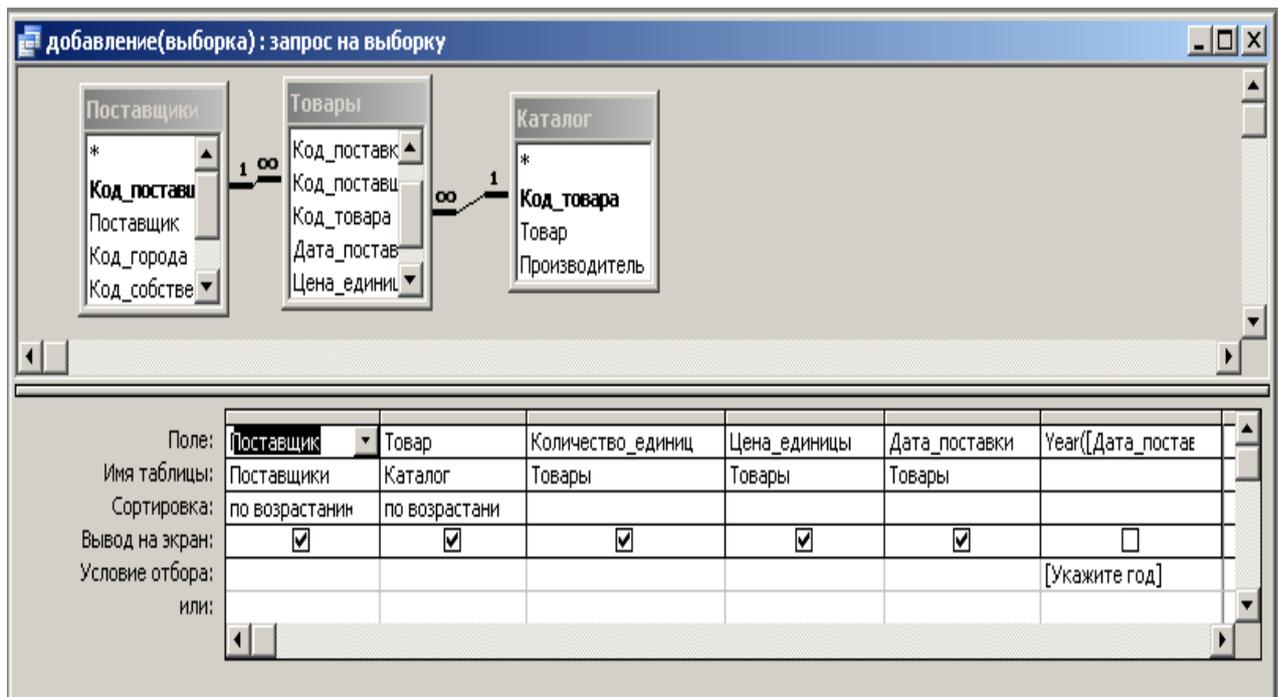
### **Пример 3.**

**Занести в архив сведения о поставках, сделанных в указанном году.**

Для реализации этого запроса выполним следующие действия.

1.Создадим в режиме конструктора структуру таблицы – преемника. Если же таблица-преемник должна иметь структуру, совпадающую со структурой базовой таблицы, то можно скопировать в буфер обмена эту базовую таблицу и при вставке из буфера обмена в окне **Вставка таблицы** выбрать опцию **Только структура**. Укажем имя новой таблицы (пока эта таблица будет пустой).

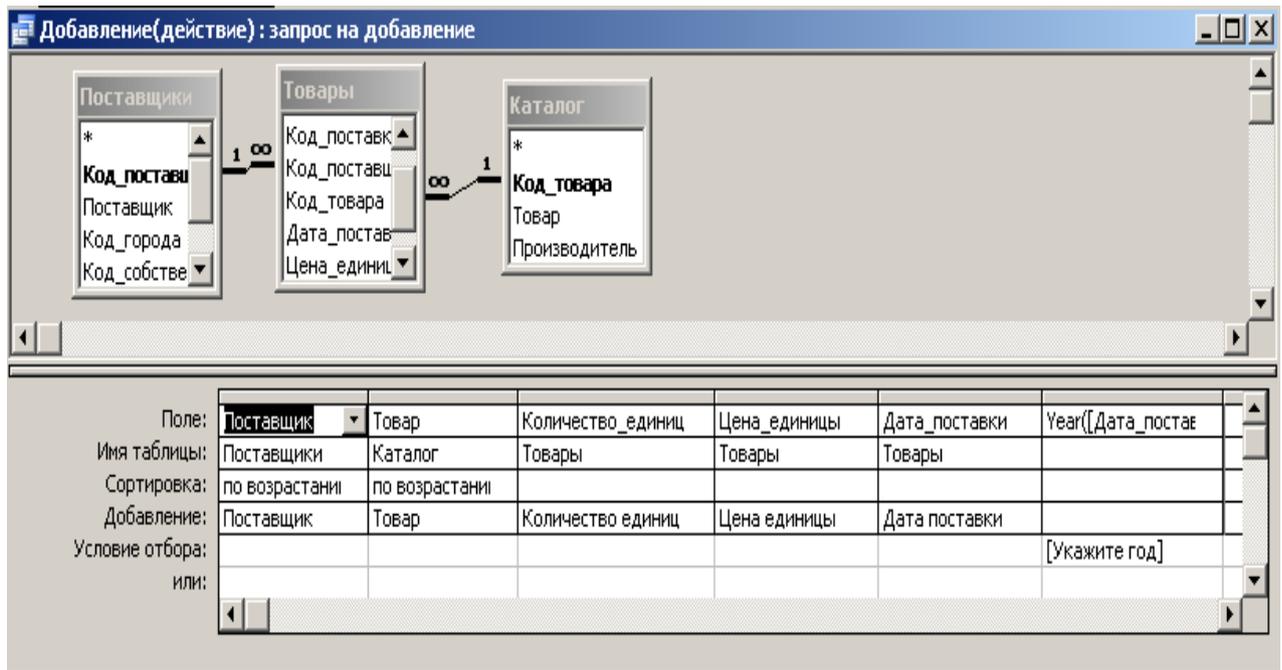
2.Создадим запрос на выборку для получения нужных данных, указывая соответствующий критерий отбора записей и задавая тип параметра:



3. Выполним запрос с тем, чтобы убедиться, что записи отобраны правильно; сделаем копию запроса.

4. Преобразуем запрос на выборку в *запрос на добавление*, выполнив команду **Запрос** → **Добавление** или выбрав **Добавление** в раскрывающемся списке кнопки **Тип запроса**. Access выведет окно добавления, в котором нужно указать имя таблицы – приемника.

5. В бланке запроса в строке **Добавление** нужно указать соответствующие имена полей таблицы – приемника. Если эти имена совпадают с именами полей базовой таблицы, то заполнение строки произойдет автоматически.



6. Щелкнем по кнопке **Запуск** и в диалоговом окне подтвердим добавление (кнопка **Да**).

7. Проверим результат выполнения запроса на добавление, просмотрев содержимое таблицы – приемника.

Для запроса на добавление генерируется следующая инструкция **SQL**:

**PARAMETERS** [Укажите год] **Short**;

**INSERT INTO** Архив ( Поставщик, Товар, [Количество единиц],  
[Цена единицы], [Дата поставки] )

**SELECT** Поставщики.Поставщик, Каталог.Товар,  
Товары.Количество\_единиц, Товары.Цена\_единицы,  
Товары.Дата\_поставки

**FROM** Поставщики **INNER JOIN** (Каталог **INNER JOIN** Товары  
**ON** Каталог.Код\_товара = Товары.Код\_товара)

**ON** Поставщики.Код\_поставщика = Товары.Код\_поставщика

**WHERE** Year([Дата\_поставки])=[Укажите год]

**ORDER BY** Поставщики.Поставщик, Каталог.Товар;

### Замечания:

1. Если вставляемые записи не удовлетворяют условию на значение для некоторого поля или для таблицы, то Access выведет сообщение об ошибке и не произведет вставку подобных записей.

2. При добавлении данных в текстовое поле или поле МЕМО может произойти усечение вставляемых данных, если поле таблицы – приемника имеет размер, не достаточный для хранения поступающих данных. О возникновении такой ситуации Access не предупреждает, так как не считает это ошибкой.

### **Запрос на удаление**

Из всех запросов на изменение *запрос на удаление* наиболее опасен, так как он удаляет записи из таблицы безвозвратно. Подобно другим запросам на изменение в этом запросе для определения удаляемых записей должны использоваться условия отбора, указываемые через параметр.

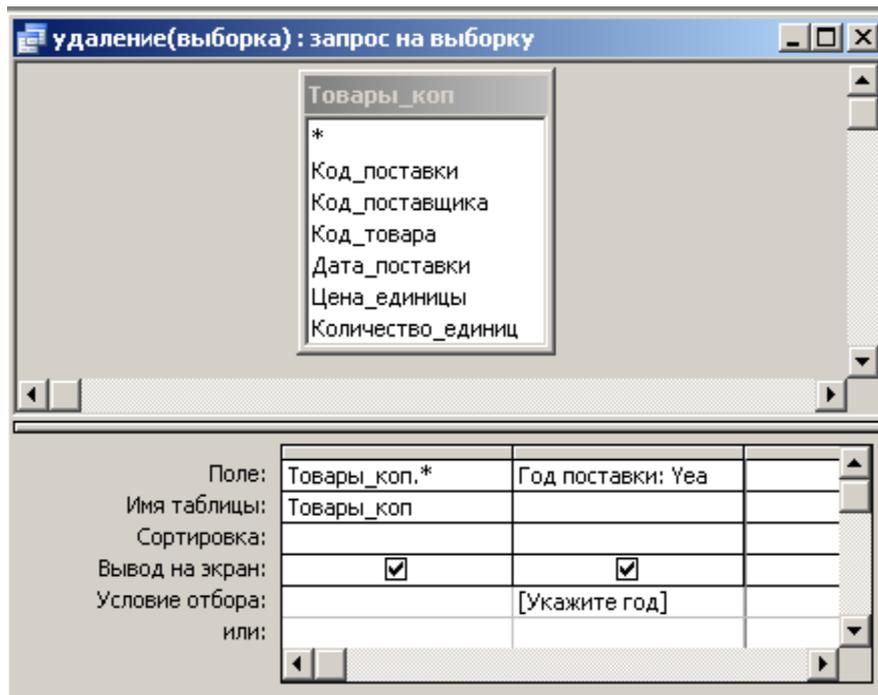
#### **Пример 4.**

**Удалить из базовой таблицы записи о поставках, занесенные в архив (смотри пример 3).**

Выполним следующую последовательность действий.

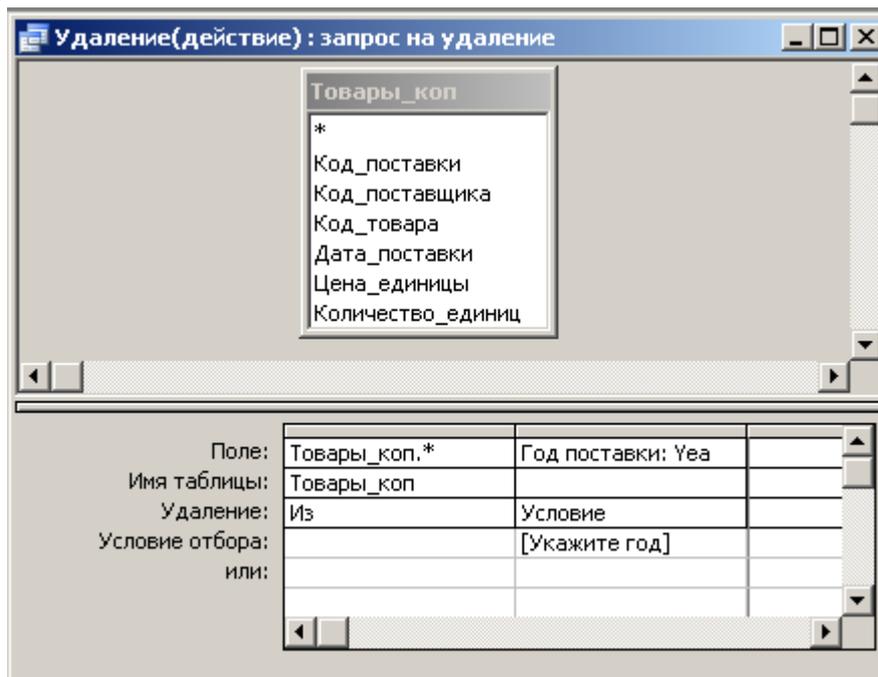
1. Создадим резервную копию таблицы, из которой удаляются записи.

2. Создадим запрос на выборку, позволяющий найти все подлежащие удалению записи. Обычно в таком запросе используется всего два поля. В первом указывается **имя-таблицы.\*** (что означает: **все поля таблицы**), второе поле используется для задания условия отбора (не забудем указать тип параметра):



3. Проверим результаты выполнения запроса.

4. В режиме конструктора преобразуем запрос на выборку в *запрос на удаление* с помощью команды **Запрос** → **Удаление**:



5. Щелкнем по кнопке **Запуск** и в диалоговом окне подтвердим удаление (кнопка **Да**).

Для запроса **Удаление(действие)** генерируется следующая инструкция **SQL**:

```
PARAMETERS [Укажите год] Short;  
DELETE Товары_коп.*, Year([Дата_поставки]) AS [Год поставки]  
FROM Товары_коп  
WHERE Year([Дата_поставки])=[Укажите год];
```

**Замечание:**

Если таблица-отец и таблица-сын связаны отношением “1:n” и включены опции **Обеспечение целостности данных** и **Каскадное удаление связанных записей**, то при удалении записей из таблицы-отца автоматически удаляются связанные с ними записи таблицы-сына.

Если же опция **Каскадное удаление связанных записей** не включена, то удалить записи из отца невозможно, так как опция **Обеспечение целостности данных** поддерживает ссылочную целостность, что не позволяет оставлять в сыне записи, не имеющие отцов. В случае, когда семантика данных предметной области позволяет содержать в сыне записи, не имеющие отцов, опция **Обеспечение целостности данных** не должна быть включена.

### **Задание к лабораторной работе**

Создать на языке QBE следующие виды запросов:

- обновление данных в записи (группе записей),
- удаление записей из таблицы,
- добавление записей из одной таблицы в другую,
- создание таблицы.

### **Требования к отчету по лабораторной работе:**

- 1) Привести скриншоты QBE запросов на обновление данных в записи, удаление записей из таблицы, добавление записей из одной таблицы в другую, создание таблицы
- 2) Привести скриншоты результатов работы всех запросов.

### **Контрольные вопросы**

Определение, виды и назначение запросов на изменение.

## Лабораторная работа 7

**Тема:** Разработка форм для работы с данными.

**Цель работы:** Овладение навыками построения экранных форм средствами MS Access.

### Методические указания к выполнению работы

Для ввода и отображения система Access предоставляет объекты специального типа – формы.

Формы применяются в следующих случаях:

- работа с данными (ввод, вывод, редактирование, удаление);
- управление ходом выполнения приложения (формы для работы с макросами и функциями Visual Basic);
- вывод сообщений (о работе приложения, об ошибках, предупреждения);
- печать информации.

Рассмотрим примеры построения некоторых форм для работы с данными.

#### Пример 1. Создание формы для ввода данных

Построим форму для ввода данных в таблицу «Поставщики».

Перейдем на вкладку «Создание» и нажмем кнопку «Конструктор форм».

Откроем окно свойств формы и на вкладке «Данные» укажем источник записей формы – таблицу «Поставщики». Нажмем кнопку «Добавить поля» на вкладке «Конструктор» для вывода списка полей.

Создадим в верхней части формы элемент управления «Надпись», содержащий текст «Новый поставщик». В окне свойств данного элемента управления зададим свойства, определяющие внешний вид надписи: выберем шрифт, цвет и размер символов.

Перетащим поля из списка полей источника записей в нужное место формы, предварительно выделив их с помощью мыши и клавиши Shift.. Следует

помнить, что первый элемент помещается в то место формы, где была отпущена кнопка мыши.

Каждый элемент состоит из надписи и поля (надпись соответствует значению свойства «Подпись» или имени поля таблицы, если подпись отсутствует). Access автоматически связывает элемент «Надпись» с ЭУ «Поле». С этими элементами можно работать как совместно, так и с каждым отдельно: их можно выделять, перемещать, удалять. Зададим свойства элементов, определяющие их внешний вид: шрифт, цвет и размер символов.

При таком способе создания поля в форме, оно наследует многие свойства связанного с ним поля таблицы.

Если в таблице «Поставщики» для поля «Код\_города» была выполнена подстановка (использовалась таблица-справочник «Города»), то для ввода в это поле системой будет использоваться элемент управления «Поле со списком».

Далее изменим свойства элемента управления для ввода значений в поле «Код\_поставщика». Данное поле является искусственным первичным ключом и имеет тип «Счетчик», поэтому для непосредственного ввода оно недоступно. Выделив данный элемент управления, откроем правой кнопкой мыши контекстное меню, выберем в нем пункт «Свойства». В открывшемся окне свойств перейдем на вкладку «Данные» и установим значения свойств «Блокировка» и «Доступ» равными «Нет». Теперь поле и связанная с ним надпись будут изображаться затененными и не смогут получить фокус ввода

Зададим значения некоторых свойств самой формы. На вкладке «Макет» в свойстве «Подпись» введем текст «Форма для ввода данных». На вкладке «Данные» зададим для свойства «Ввод данных» значение «Да».

В результате выполнения этих действий получим форму, показанную на рис. 1.

The image shows a software window titled "Форма для ввода данных" (Data Entry Form) with a subtitle "Новый поставщик" (New Supplier). The form contains the following fields and controls:

- Код поставщика** (Supplier Code): A text input field with "(Счетчик)" (Counter) to its right.
- Поставщик** (Supplier): A text input field.
- Город** (City): A dropdown menu.
- Фирма собственности** (Ownership Company): A dropdown menu.
- Телефон** (Phone): A text input field.
- Адрес** (Address): A text input field.
- Занести** (Save): A button with a dashed border.
- Отменить** (Cancel): A button.

Рисунок 1 – Форма для ввода данных в таблицу «Поставщики»

При открытии данной формы все ее элементы управления будут либо пустыми, либо заполненными значениями по умолчанию. При вводе значений будут проверяться условия на значения, будут действовать свойства «Обязательное поле», «Маска ввода» и «Формат поля», заданные при создании таблицы «Поставщики». При закрытии формы либо при переходе к следующей записи значения, введенные в элементах управления формы, будут сохранены в новой записи таблицы «Поставщики».

## Пример 2. Создание ленточной формы

Особенность ленточной формы состоит в том, что она одновременно представляет на экране список записей. Особенно удобно такие формы использовать для представления результатов итоговых и перекрестных запросов.

Для отображения данных итогового запроса без условия построим ленточную форму с помощью мастера. Для этого на вкладке «Создание» выберем меню «Другие формы», а в нем – пункт «Мастер форм». Далее, отвечая на вопросы, укажем источник записей – итоговый запрос без условия, в списке полей выберем все поля запроса. В окне выбора внешнего вида формы выберем пункт «Ленточная форма». Затем укажем стиль оформления формы (в примере выбран стиль «Обычная») и введем ее имя. Построенная мастером

форма показана на рис. 2. Как видим, она мало пригодна для практического использования.

Товар	Ичество партий	Общая стоимость	едная цена
CD-R	2	1750	175,00
DVD-R	2	100	5,00
Видео	2	1400	225,00
Клави	7	7333,49996948242	157,42
Матер	11	31878,8796386719	718,75
Модул	2	1140	105,00
Монит	3	59300	2533,33
Съемк	2	13000	600,00

Рисунок 2 - Пример ленточной формы

Перейдем в режим конструктора формы и в окне ее свойств на вкладке «Макет» зададим в свойстве «Подпись» текст «Обобщенные показатели по товарам».

В построенной форме заголовки столбцов помещаются отдельно от записей – в заголовке формы, при необходимости заголовки столбцов можно указывать несколькими строками, нажав в месте разрыва комбинацию клавиш Ctrl+Enter.

Сами поля размещаются в области данных. Поля могут быть любой ширины и высоты; к ним можно применять специальные эффекты (например, тени или эффекты объема).

Увеличим ширину поля «Товар», уменьшим ширину полей «Количество партий», «Общая стоимость», «Средняя цена». Подписи полей в заголовке формы разобьем на несколько строк. Выполним выравнивание полей и их надписей, воспользовавшись командами контекстного меню. Для поля «Общая стоимость» в окне свойств на вкладке «Макет» укажем:

Формат поля – фиксированный,

Число десятичных знаков – 2.

Изменим текст надписи в заголовке формы.

В разделе примечаний ленточной формы обычно помещаются командные кнопки и вычисляемые элементы управления для подсчета итогов по всем записям.

Свойство формы «Режим по умолчанию» должно иметь значение «Ленточная форма».

Поскольку источником записей для данной формы является итоговый запрос, результаты выполнения которого не могут редактироваться, изменим некоторые свойства построенной формы, открыв ее в режиме конструктора. Выделим все элементы управления в области данных, откроем окно их свойств с помощью контекстного меню, в свойстве «Доступ» укажем значение «Нет», а в свойстве «Блокировка» - значение «Да». Теперь элементы управления не смогут получить фокус ввода, сохранив настройки макета, заданные разработчиком. Построенная форма приведена на рис. 3.

Товар	Средняя цена	Количество партий	Общая стоимость
CD drive	175,00	2	1750,00
DVD RW	5,00	2	100,00
Видеокарта	225,00	2	1400,00
Клавиатура	157,42	7	7333,50
Материнская плата	718,75	11	31878,88
Модуль памяти	105,00	2	1140,00
Монитор	2533,33	3	59300,00
Съёмный диск	600,00	2	13000,00

Рисунок 3 - Пример ленточной формы

**Пример 3.** Составная форма для работы с данными, находящимися в отношении "один-ко-многим".

Особенность составной формы заключается в том, что в форму, называемую *главной*, включается другая форма, называемая *подчиненной*. В главной форме размещаются данные из таблицы-отца, в подчиненной – из таблицы-сына, причем в ней будут отображены только те записи, которые связаны с записями главной формы. При переходе к следующей записи главной формы автоматически обновляется содержимое всех подчиненных форм.

Составная форма, построенная только на таблицах, может использоваться также для ввода и редактирования данных в таблицах.

Данные в составной форме можно отображать в разных форматах, при этом главную форму принято представлять в виде простой формы, подчиненную – в виде ленточной или табличной форм.

Создадим форму для вывода сведений о поставщиках и представленных ими товарах. В главной форме будут выводиться сведения о поставщике, в подчиненной – о его товарах. Эти две формы связаны: при переходе к другому поставщику (то есть к другой записи главной формы) в подчиненной отобразится новый перечень товаров.

В начале создадим источники данных для главной и подчиненной форм. Чтобы в главной форме указать среди сведений о поставщике не только поле «Город», но и поле «Страна», не содержащееся в таблице «Поставщики», нужно создать запрос на выборку и включить в него поле «Страна». Этот запрос и будет источником записей для главной формы. Аналогично, в таблице «Товары» содержится только код товара, а производитель его указан в таблице «Каталог». Для того чтобы к сведениям о товаре добавилось поле «Производитель», нужно также создать запрос на выборку.

Разработку составной формы нужно начинать с подчиненной формы.

Подчиненную форму создадим в виде ленточной (можно воспользоваться мастером). Далее откроем подчиненную форму в режиме конструктора и дополним ее вычисляемым элементом управления «Поле». Данный элемент управления разместим в области примечаний подчиненной формы. Это поле предназначено для представления общей суммы за все партии товаров каждого поставщика. В окне свойств данного элемента управления на вкладке «Другие» в свойстве «Имя» укажем значение «Общая сумма», на вкладке «Макет» в свойстве «Формат» укажем значение «Фиксированный», в свойстве «Число десятичных знаков» - значение 2. На вкладке «Данные» в свойстве «Данные» этого поля нужно указать выражение

$$=Sum([Цена\_единицы]*[Количество\_единиц]).$$

Далее создадим главную форму. Для ее создания можно воспользоваться мастером. Результат работы мастера изменим, разместив по своему вкусу элементы управления, добавив надпись «Сведения поставщике и его товарах» и изменив подпись формы. Увеличим высоту формы.

Следующий шаг – связывание форм. Для этого оставим главную форму в режиме конструктора, а ярлык подчиненной формы перетащим в главную из окна базы данных и разместим по своему вкусу. Сохранив форму и переключившись в режим ее просмотра, получим изображение, как на рис. 4.

#### **Пример 4. Форма с набором вкладок для представления справочников.**

Для представления справочников создадим форму, нажав кнопку «Конструктор форм» на вкладке «Создание». Далее, в режиме конструктора, зададим для формы на вкладке «Макет» следующие значения свойств:

- Подпись – Справочники,
- Область выделения – нет,
- Кнопки перехода – нет,
- Полосы прокрутки - отсутствуют.

Товар	Производитель	Дата поставки	Цена единицы	Количество единиц
Винчестерские диски	Quantum	02.12.2002	120	10
Клавиатура	NEC	24.04.2002	120	10
Клавиатура	NEC	24.08.2002	50	50
Материнская плата	Acer	07.06.1999	12	3
Монитор	Samsung	12.12.1999	220	5

Общая сумма: 24109

Запись: 1 из 8

Запись: 1 из 6

Рисунок 4 – Составная форма

Далее, на вкладке «Конструктор» в группе «Элементы управления» нажмем набор «Вкладка» и укажем положение и размер вкладок на форме. С помощью контекстного меню (команды «Вставить вкладку» и «Удалить вкладку») создадим набор из трех вкладок.

Выделив каждую вкладку, в окне свойств зададим необходимое значение в свойстве «Подпись», и, при желании, в свойстве «Рисунок».

Далее создадим ленточные формы для отображения записей каждого справочника и внедрим их в разрабатываемую форму, перетащив ярлык из окна БД на соответствующую вкладку. Получим форму, аналогичную показанной на рис.5.

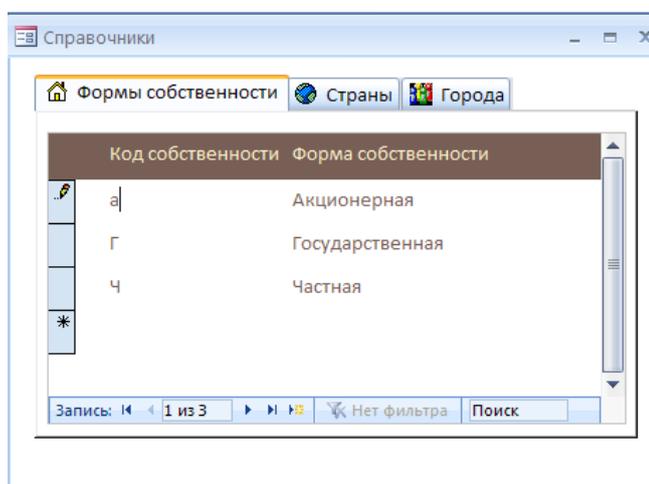


Рисунок 5- Форма вкладки

### Пример 5. Построение объемной диаграммы

Создадим пустую форму (кнопка «Конструктор форм» на вкладке «Создание»). Далее, в режиме конструктора, зададим для формы на вкладке «Макет» следующие значения свойств:

Подпись – Объемная диаграмма,

Область выделения – нет,

Кнопки перехода – нет,

Полосы прокрутки - отсутствуют.

На вкладке «Конструктор» в группе «Элементы управления» нажмем кнопку «Вставить диаграмму», укажем положение и размер диаграммы. Для гистограммы и других видов диаграмм с двумя осями (трехмерных диаграмм) *поле категорий*

располагается по оси X, а *поле значений* – по оси Y (обычно ось абсцисс представляет текстовое поле, ось ординат – числовое значение). Роль третьей координаты играет дополнительное третье поле, на основании которого значения по оси Y будут разбиты на подкатегории. В результате получаются *ряды данных*.

Построим диаграмму по результатам следующего перекрестного запроса: для каждого поставщика определить по годам общее количество поставленных партий товаров.

Мастер начинает работу и предлагает следующие экраны:

1) Выбор полей, которые необходимо отобразить в диаграмме (если в запросе несколько полей представляют заголовки строк, то для построения диаграммы нужно взять только одно поле).

2) Выбор типа диаграммы (для перекрестного запроса нужно выбрать трехмерную диаграмму, например, гистограмму). В правой части окна появляется описание выбранного типа диаграммы.

3) Выбор способа отображения данных в диаграмме. Мастер определяет, что поле «Поставщик» используется в качестве *поля категорий*, то есть делений по оси X, а значения поля «Общее количество» определяют размеры столбцов гистограммы, то есть используются в качестве делений по оси Y. Для выделения рядов берется поле «Год».

На данном этапе окно «Создание диаграммы» выглядит, как на рис.6.

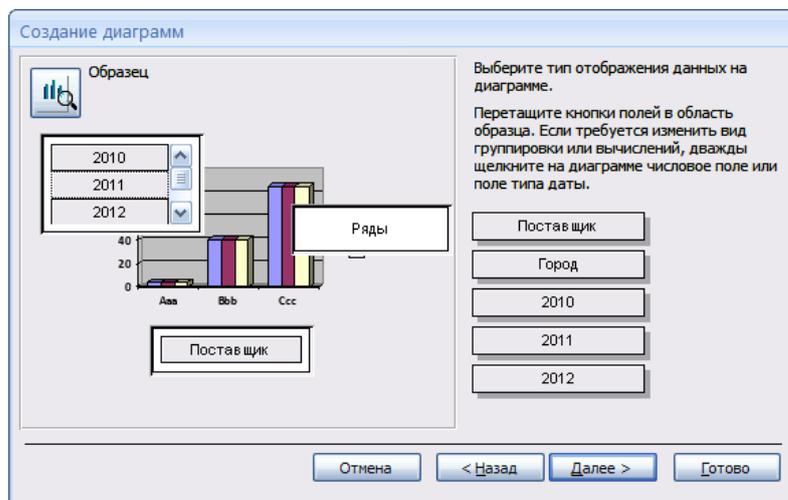


Рисунок 6 - Создание диаграммы

Окно разделено на две части: в правой отображаются выбранные для построения диаграммы поля, левая часть содержит макет диаграммы.

Так как при создании диаграммы было выбрано несколько числовых полей, их нужно перетащить в левую часть окна для создания рядов диаграммы (имена этих рядов появятся в *легенде*). Перетаскиваем поля в левую часть и присоединяем их к списку. По умолчанию для них назначается операция «Сумма» (например, «Сумма\_2010»). Суммирующую операцию нужно выбрать для каждого поля списка. Дважды щелкаем по полю - открывается окно «Вычисление итоговых значений», в котором указываются итоговые операции. **Но так как итоговые значения уже вычислены в запросе, то для всех полей выбираем опцию «Отсутствует».**

5) В последнем окне Мастера нужно указать заголовок диаграммы. Если заголовок должен состоять из двух строк, то после ввода первой строки нужно нажать Ctrl – Enter, а затем ввести вторую строку. Пустую строку, которая появится между первой и второй строками заголовка, можно удалить, используя Microsoft Graph.

В этом окне нужно также определить, будет ли выводиться *легенда* (в *легенде* содержится перечень условных обозначений для рядов данных гистограммы или для категорий сегментов круговой диаграммы). Для вывода легенды опции «Показать условные обозначения?» следует присвоить значение «Да». Работа Мастера завершена, и нужно щелкнуть на кнопке «Готово».

После выполнения описанных выше действий в окне конструктора форм появится диаграмма, созданная для рассматриваемого запроса (рис.7).

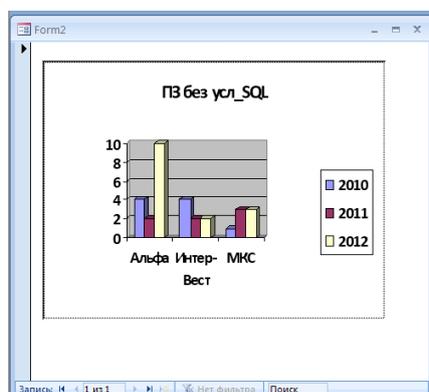


Рисунок 7 - Пример построенной диаграммы мастером

В режимах формы и конструктора диаграмма отображается с помощью *рамки диаграммы*. Чтобы размер диаграммы соответствовал *рамке*, свойству **Установка размеров**, имеющему первоначальное значение **Фрагмент**, нужно задать значение **Вписать в рамку** или **По размеру рамки** (однако следует помнить, что диаграмму рекомендуется отображать пропорционально).

Так как диаграмма предназначена только просмотра, то в окончательном ее варианте нужно установить следующие значения свойств:

Доступ – Нет

Блокировка – Да

После выполнения некоторой настройки начальная диаграмма будет иметь более привлекательный вид (рис.8).

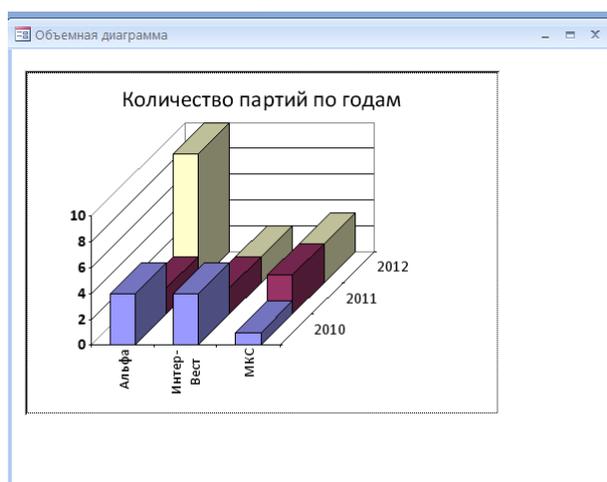


Рисунок 8 - Настройка диаграммы

### **Пример 5. Построение круговой диаграммы.**

Круговая диаграмма всегда строится для сопоставления величин, в совокупности представляющих единое целое. В качестве примера построим круговую диаграмму, иллюстрирующую вклад каждого поставщика в поставки. Будем отображать на диаграмме количество партий для каждого поставщика, используя итоговый запрос без условия.

Для вставки диаграммы поступим так же, как и в предыдущем примере: создадим пустую форму и вызовем мастер построения диаграмм.

Далее выберем тип диаграммы и укажем ряды данных (см. рис. 9). При построении двумерных диаграмм, например, круговой диаграммы, нужно

выбрать *поле категорий*, которое будет определять название сегментов диаграммы, и *поле значений* - для определения величины сегмента каждой категории.

При указании рядов обязательно устраним вычисления итогов, поскольку они выполняются в источнике данных – запросе.

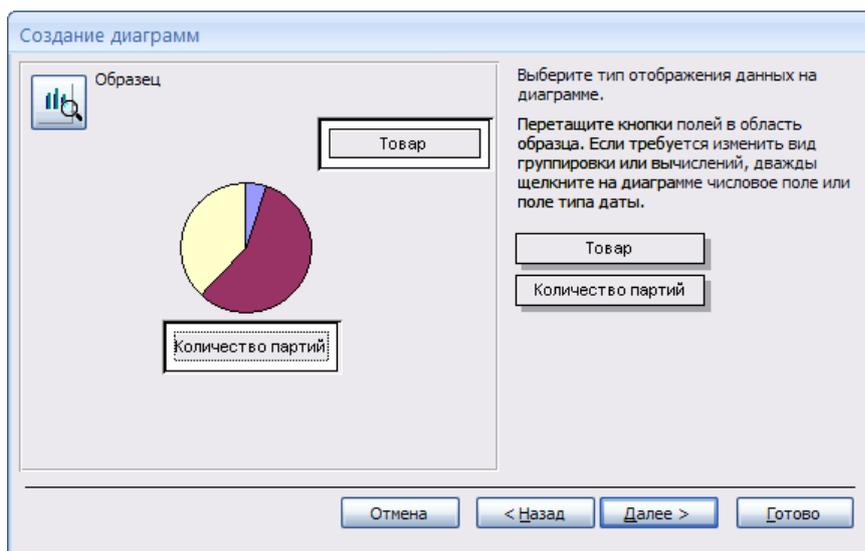


Рисунок 9 - Выбор типа диаграммы, определение рядов

Построенная мастером диаграмма, как правило, не отображает всю необходимую информацию. Выполнив двойной щелчок по диаграмме в режиме конструктора, запустим программу Microsoft Graph, с помощью которой можно изменить внешний вид диаграммы. Выбрав в контекстном меню пункт «Параметры диаграммы», на вкладке «Подписи данных» отметим флажок «Значения», чтобы рядом с каждым сектором диаграммы выводилось соответствующее числовое значение. Выбрав в контекстном меню пункт «Формат области построения», изменим заливку. Получим форму, показанную на рис. 10.



Рисунок 10 - Пример круговой диаграммы

### Задание к лабораторной работе

Разработать следующие виды форм:

- для ввода данных в базовую таблицу (использовать элементы управления «Поле», «Список»/ «Поле со списком»),
- ленточную форму для просмотра результатов перекрестного или итогового запроса (с помощью Мастера),
- составную форму на основании связанных таблиц с вычисляемым элементом в подчиненной форме,
- объемная диаграмма по результатам перекрестного запроса,
- круговая диаграмма по результатам итогового запроса
- набор вкладок для представления справочников.

Одну из форм дополнить фоновым рисунком, в одной форме рисунок включить в виде отдельного элемента управления.

### Требования к отчету по лабораторной работе:

Привести скриншоты всех разработанных Вами форм.

## **Контрольные вопросы**

1. Определение формы: структура, виды, основные свойства.
2. Элементы управления: назначение, правила построения, основные свойства.
3. Диаграммы: назначение, типы, правила построения

## Лабораторная работа №8

**Тема:** Разработка отчетов.

**Цель работы:** Овладение навыками построения отчетной документации средствами MS Access.

### Методические указания к лабораторной работе

Основная функция отчетов – подготовка данных в пригодном для печати виде. Отчеты обладают значительной частью тех же свойств, что и формы. Элементы управления, которые используются в отчетах, также аналогичны тем, которые применяются в формах.

Для создания отчетов используются соответствующие кнопки на вкладке «Создание». МА позволяет создавать отчет самостоятельно или с помощью мастера. Мастер позволяет быстро разработать макет, служащий основой создаваемого отчета. После этого можно переключиться в режим конструктора и внести изменения в стандартный макет.

Указанным способом можно создавать **диаграммы** (с помощью «**Мастера диаграмм**»), **почтовые наклейки**, **ленточный автоотчет** (по аналогии с ленточной формой) или **автоотчет в столбец** (аналог простой формы), а также другие виды отчетов.

Существует также возможность сохранения формы в виде отчета. Для этого необходимо в окне базы данных выбрать сохраняемую форму, в появившемся контекстном меню выбрать команду «**Сохранить как отчет**». После этого следует ввести имя для отчета в поле «**Имя отчета**». Построенный таким образом отчет будет иметь те же элементы управления и свойства, что и форма.

Отчеты, как и формы, можно создавать вручную. При таком способе необходимо выполнить определенную последовательность действий:

1. Создание пустого отчета и выбор источника записей.
2. Сортировка и группировка данных в отчете.
3. Добавление заголовка и примечаний.

4. Отображение данных, текста и итогов в текстовых полях, надписях и других элементах управления.
5. Форматирование, перемещение, изменение размеров и выравнивания надписей, текстовых полей и других элементов управления.
6. Просмотр и распечатка отчета.

Рассмотрим примеры создания различных отчетов.

### Пример 1. Отчет с группировкой записей.

В качестве источника данных для отчета будем использовать запрос «Поставщики\_товары», представляющий собой симметричное объединение таблиц «Поставщики» и «Товары».

Первоначально создадим вручную отчет, вид которого в режиме конструктора показан на рис. 1, а в режиме предварительного просмотра – на рис. 2.

Как видим, в данном случае имеется большая избыточность: многократно повторяются названия поставщика и города. Для устранения избыточности введем группировку по полю «Поставщик» (см.рис. 3).

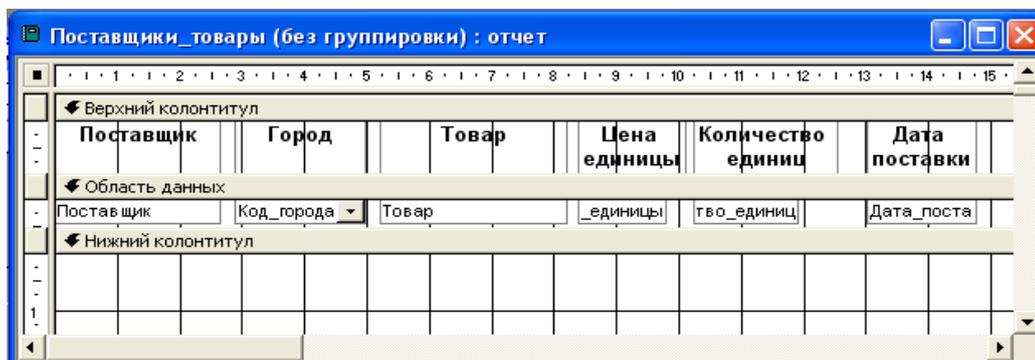


Рисунок 1– Отчет в режиме конструктора

Поставщик	Город	Товар	Цена единицы	Количество единиц	Дата поставки
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Видеокарта	422,00	1	01.01.2001
SunGroup	Гомель	Видеокарта	25,00	1	05.04.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
SunGroup	Гомель	CD drive	315,00	3	08.12.2002
Альфа	Петербург	Клавиатура	55,00	10	24.12.2002
Альфа	Петербург	Flash-карта	0,00	10	06.02.2002
Альфа	Петербург	Flash-карта	25,50	10	02.07.2001
Альфа	Петербург	Клавиатура	70,00	2	05.01.2002
Альфа	Петербург	Видеокарта	230,00	1	07.01.2001
Альфа	Петербург	Клавиатура	34,00	1	11.04.2002
Альфа	Петербург	Видеокарта	330,00	10	24.12.2002
Альфа	Петербург	Клавиатура	100,00	25	24.12.2002
Артемон	Донецк	Flash-карта	12,00	5	12.09.2002
Артемон	Донецк	Видеокарта	10,00	6	18.05.2002

Рисунок 2– Отчет в режиме предварительного просмотра

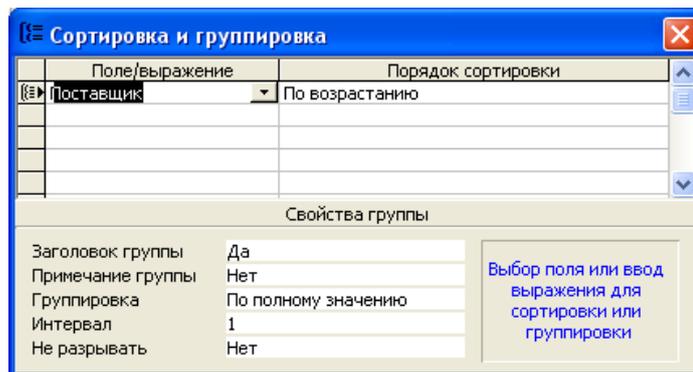


Рисунок 3– Определение группировки по полю отчета

После этого в макете отчета появится дополнительный раздел – **Заголовок группы «Поставщик»**. Перенесем в этот раздел из области данных элементы управления, отображающие поля «Поставщик» и «Код\_города», и в режиме конструктора отчет будет выглядеть, как показано на рис.4, а в режиме предварительного просмотра – как на рис.5. Как видим, после внесения изменений название и город каждого поставщика указываются однократно, только в начале каждой группы записей.

## Пример 2. Отчет с расчетом итоговых показателей для групп записей

Видоизменим отчет, разработанный в предыдущем примере. В заголовок группы «Поставщик» добавим вычисляемое поле «ИтогПоставщика». В свойстве «Данные» этого поля укажем выражение

$$=Sum([Цена_единицы]*[Количество_единиц]),$$

а в свойстве «Формат поля» – «Денежный». Скопируем это поле в примечание отчета, где зададим его имя «ОбщийИтог» (см.рис.6).

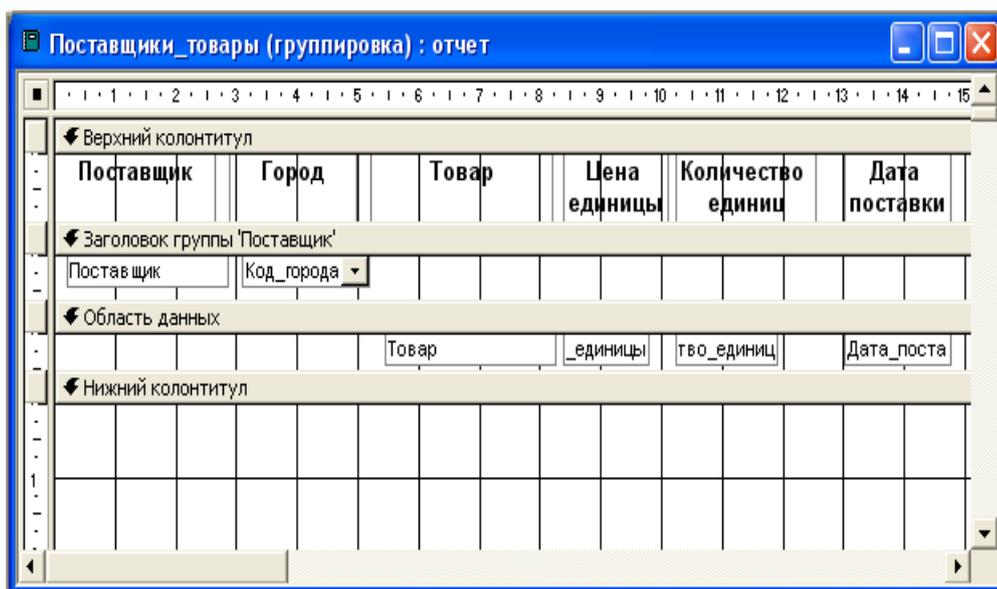


Рисунок 4– Отчет после включения заголовка группы

Поставщик	Город	Товар	Цена единицы	Количество единиц	Дата поставки
SunGroup	Гомель	Flash-карта	1,00	1	06.06.2002
		Видеокарта	422,00	1	01.01.2001
		Видеокарта	25,00	1	05.04.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		Flash-карта	1,00	1	06.06.2002
		CD drive	315,00	3	08.12.2002
Альфа	Петербург	Клавиатура	55,00	10	24.12.2002
		Flash-карта	0,00	10	06.02.2002
		Flash-карта	25,50	10	02.07.2001
		Клавиатура	70,00	2	05.01.2002
		Видеокарта	230,00	1	07.01.2001
		Клавиатура	34,00	1	11.04.2002
		Видеокарта	330,00	10	24.12.2002
		Клавиатура	100,00	25	24.12.2002
Артемон	Донецк	Flash-карта	12,00	5	12.09.2002
		Видеокарта	10,00	6	18.05.2002
Интер-Вест	Донецк	CD drive	12,00	1	12.12.2002
		Клавиатура	167,00	5	11.05.2002
		Видеокарта	27,00	10	16.12.2002
		Клавиатура	167,00	4	07.05.2002
Омега	Донецк	Flash-карта	14,00	4	29.09.2002
		Клавиатура	34,00	7	12.10.2002
		CD drive	26,00	5	10.10.2002

Рисунок 5– Отчет с группировкой в режиме предварительного просмотра

Поставщики\_товары (итоги) : отчет

Поставщик	Товар	Цена единицы	Количество единиц	Дата поставки
Заголовок отчета				
Верхний колонтитул				
Заголовок группы 'Поставщик'				
Поставщик				
=Sum(Цена_еди				
Область данных				
	Товар	_единицы	во_единиц	Дата_поста
Нижний колонтитул				
Примечание отчета				
Итого:		=Sum(Цена_ед		

Рисунок 6 - Отчет с общим итогом по поставщику

В режиме предварительного просмотра полученный отчет будет содержать значение общей стоимости товаров для каждого поставщика и общую стоимость по всем поставкам (рис.7).

Поставщик	Товар	Цена единицы	Количество единиц	Дата поставки
<b>SunGroup</b>				
<b>1 399,00 грн.</b>				
	Flash-карта	1,00	1	06.06.2002
	Видеокарта	422,00	1	01.01.2001
	Видеокарта	25,00	1	05.04.2002
	Flash-карта	1,00	1	06.06.2002
	Flash-карта	1,00	1	06.06.2002
	Flash-карта	1,00	1	06.06.2002
	Flash-карта	1,00	1	06.06.2002
	Flash-карта	1,00	1	06.06.2002
	Flash-карта	1,00	1	06.06.2002
	CD drive	315,00	3	08.12.2002
<b>Аль фа</b>				
<b>7 009,00 грн.</b>				
	Клавиатура	55,00	10	24.12.2002
	Flash-карта	0,00	10	06.02.2002
	Flash-карта	25,50	10	02.07.2001
	Клавиатура	70,00	2	05.01.2002
	Видеокарта	230,00	1	07.01.2001
	Клавиатура	34,00	1	11.04.2002
	Видеокарта	330,00	10	24.12.2002
	Клавиатура	100,00	25	24.12.2002
<b>Артемон</b>				
<b>120,00 грн.</b>				
	Flash-карта	12,00	5	12.09.2002
	Видеокарта	10,00	6	18.05.2002
<b>Интер-Вест</b>				
<b>1 785,00 грн.</b>				
	CD drive	12,00	1	12.12.2002
	Клавиатура	167,00	5	11.05.2002
	Видеокарта	27,00	10	16.12.2002
	Клавиатура	167,00	4	07.05.2002
<b>Омега</b>				
<b>424,00 грн.</b>				
	Flash-карта	14,00	4	29.09.2002
	Клавиатура	34,00	7	12.10.2002
	CD drive	26,00	5	10.10.2002
<b>Итого:</b>				
	<b>10 737,00 грн.</b>			

Рисунок 7

**Пример 3.** Отчет с расчетом итоговых показателей и долей для групп записей

В отчет, разработанный в примере 2, внесем следующие изменения. В заголовок группы «Поставщики» добавим поле «ДоляПоставщика», у которого в свойстве «Данные» указано выражение

$$=[\text{ИтогПоставщика}]/[\text{ОбщийИтог}],$$

а в свойстве «Формат поля» – «Процентный». В область данных отчета добавим вычисляемое поле «Номер», у которого свойство «Данные» имеет значение =1, а свойство «Сумма с накоплением» – «Для группы». Получим отчет, показанный на рис.8.

Поставщик	Товар	Цена единицы	Количество единиц	Дата поставки	Стоимость партии
<b>SunGroup</b>	<b>1 399,00 грн.</b>	<b>13,03%</b>			
1	Flash-карта	1,00	1	06.06.2002	1,00 грн.
2	Видеокарта	422,00	1	01.01.2001	422,00 грн.
3	Видеокарта	25,00	1	05.04.2002	25,00 грн.
4	Flash-карта	1,00	1	06.06.2002	1,00 грн.
5	Flash-карта	1,00	1	06.06.2002	1,00 грн.
6	Flash-карта	1,00	1	06.06.2002	1,00 грн.
7	Flash-карта	1,00	1	06.06.2002	1,00 грн.
8	Flash-карта	1,00	1	06.06.2002	1,00 грн.
9	Flash-карта	1,00	1	06.06.2002	1,00 грн.
10	CD drive	315,00	3	08.12.2002	945,00 грн.
<b>Альфа</b>	<b>7 009,00 грн.</b>	<b>65,28%</b>			
1	Клавиатура	55,00	10	24.12.2002	550,00 грн.
2	Flash-карта	0,00	10	06.02.2002	0,00 грн.
3	Flash-карта	25,50	10	02.07.2001	255,00 грн.
4	Клавиатура	70,00	2	05.01.2002	140,00 грн.
5	Видеокарта	230,00	1	07.01.2001	230,00 грн.
6	Клавиатура	34,00	1	11.04.2002	34,00 грн.
7	Видеокарта	330,00	10	24.12.2002	3 300,00 грн.
8	Клавиатура	100,00	25	24.12.2002	2 500,00 грн.
<b>Артемон</b>	<b>120,00 грн.</b>	<b>1,12%</b>			
1	Flash-карта	12,00	5	12.09.2002	60,00 грн.
2	Видеокарта	10,00	6	18.05.2002	60,00 грн.
<b>Интер-Вест</b>	<b>1 785,00 грн.</b>	<b>16,62%</b>			
1	CD drive	12,00	1	12.12.2002	12,00 грн.
2	Клавиатура	167,00	5	11.05.2002	835,00 грн.
3	Видеокарта	27,00	10	16.12.2002	270,00 грн.
4	Клавиатура	167,00	4	07.05.2002	668,00 грн.
<b>Омега</b>	<b>424,00 грн.</b>	<b>3,95%</b>			
1	Flash-карта	14,00	4	29.09.2002	56,00 грн.
2	Клавиатура	34,00	7	12.10.2002	238,00 грн.
3	CD drive	26,00	5	10.10.2002	130,00 грн.
<b>Итого:</b>	<b>10 737,00 грн.</b>				

Рисунок 8 - Отчет с расчетом итоговых показателей и долей для групп записей

### **Задание к лабораторной работе:**

В ходе выполнения лабораторной работы студенты должны на базе **различных** запросов разработать следующие отчеты:

- ленточный автоотчет;
- автоотчет в столбец;
- отчет с группировкой записей;
- отчет с группировкой записей и расчетом итоговых показателей и долей для групп.

### **Требование к отчету по лабораторной работе:**

В отчет включить вид всех разработанных отчетов в режиме конструктора и в режиме печати с пояснениями, какую информацию предоставляет отчет.

### **Контрольные вопросы**

1. Структура, режимы просмотра отчета.
2. Сортировка и группировка записей в отчете, создание вычисляемых полей
3. Способы повышения скорости печати отчетов

## Приложение А

### Описание наиболее используемых типов полей таблиц Microsoft Access

Тип	Описание
Текстовый	используется для представления алфавитно-цифровых (символьных) данных, например, имен, адресов, описаний, а также числовых данных, которые не используются в вычислениях (номера телефонов, документов, почтовые индексы и т.п.). Длина их не должна превышать 255 байт (символов).
Числовой	применяется для представления данных, используемых в вычислениях. Числовые данные могут быть разного размера: 1, 2, 4, 8 или 12 байт, целыми либо вещественными.
Денежный	используется для хранения данных о денежных суммах (занимает 8 байт). Целая часть данного типа может содержать до 15 десятичных знаков, дробная – до четырех.
Счетчик	предназначен для представления искусственного первичного ключа. Значения задаются путем автоматической генерации последовательных целых чисел. Размер – 4 байта (значения до 2 миллиардов); значения счетчика обновлять нельзя.
Логический	используется для хранения данных, имеющих только два значения: <b>Да/Нет (Истина/Ложь)</b> . Значению <b>Да</b> соответствует <b>-1</b> , значению <b>Нет</b> – <b>0</b> .
Дата/Время	используется для хранения календарных дат или значений времени, относящихся к годам с 100 по 9999. Размер поля – 8 байт. Данные хранятся в специальном фиксированном числовом формате. Позволяет выполнять вычисления в единицах измерения времени: секундах, минутах, часах, днях, месяцах и годах. Для дат допустима операция вычитания (при вычитании дат результатом является количество дней, при вычитании (сложении) из даты целого числа получается новая дата) и специальные системные функции.
МЕМО	текст объемом от 256 до 65 535 байт. Поле <b>МЕМО</b> занимает 8 байт для хранения не самих данных, а ссылок на них. Операции сортировки и поиска для полей типа <b>МЕМО</b> выполнены быть не могут.
Поле объекта OLE	позволяет использовать в БД данные других приложений Windows (такие приложения называются OLE-серверами), например, текстовые документы (MS Word), электронные таблицы (MS Excel), рисунки, звуковые и видеофайлы. В поле хранится ссылка на OLE-объект; объем объекта ограничивается имеющимся в наличии дисковым пространством.

## Приложение Б

### Форматы вывода значений различных типов

#### Б.1 Стандартные форматы для значений типа «Числовой» и «Денежный»

Формат	Описание
Основной	дробные числа с точностью до 7 цифр Пример: 2,234 -11,89. Устанавливается по умолчанию
Денежный	числа с разделителем разрядов и с символом денежной единицы. Пример: 4 256, 45 грн. Символ денежной единицы зависит от установки в Windows ( <b>Язык и стандарты</b> на панели управления)
Евро	символ денежной единицы "евро". Пример: 4 256, 45 €
Фиксированный	числа выводятся без разделителей групп разрядов. Пример: 4256,45
С разделителями разрядов	разделяет пробелом группы разрядов Пример: 4 256,45
Процентный	умножает значение на 100 и добавляет в конце символ процента. Пример: Значение 0,56 будет выводиться как 56%
Экспоненциальный	указывает при выводе мантиссу и порядок. Пример: 4,3E+03

#### Б.2 Стандартные форматы для значений типа «Дата/время»

Значение	Описание
Полный формат даты (Значение по умолчанию.)	Если значением является только дата, то время не отображается; если значением является только время, то не отображается дата. Это значение представляет собой комбинацию краткого формата даты и длинного формата времени. Примеры: 03.04.93, 17:34:00 и 03.04.93 17:34:00.
Длинный формат даты	То же, что и длинный формат даты, определенный в языковых и региональных параметрах панели управления Microsoft Windows. Пример: суббота, 3 апреля 1993 г.
Средний формат даты	Пример: 3-апр-93.
Краткий формат даты	То же, что и краткий формат даты, определенный в языковых и региональных параметрах панели управления Microsoft Windows.

<b>Значение</b>	<b>Описание</b>
	Пример: 03.04.93. Предупреждение. Значения краткого формата данных предполагают, что даты от 01.01.00 до 31.12.29 являются датами 21-го столетия (т. е. относятся к 2000-2029 годам). Даты с 01.01.30 по 31.12.99 считаются датами 20-го столетия (1930-1999 гг.).
Длинный формат времени	То же, что и параметр вкладки Время в языковых и региональных параметрах панели управления Microsoft Windows. Пример: 17:34:23.
Средний формат времени	Пример: 17:34.
Краткий формат времени	Пример: 17:34.

### **Б.3 Символы для создания пользовательских форматов даты и времени**

<b>Символ</b>	<b>Описание</b>
:	Разделитель времени. Символ, используемый для разделения компонентов текстовых или числовых значений. Разделители определены в языковых и региональных параметрах панели управления Microsoft Windows.
/	Разделитель компонентов даты (в зависимости от языковых и региональных параметров).
c	То же, что и встроенный полный формат даты.
d	Число месяца, состоящее из одной или двух цифр (от 1 до 31).
dd	Число месяца, состоящее из двух цифр (от 01 до 31).
ddd	Сокращенное название дня недели (пн-вс).
dddd	Полное название дня недели (понедельник-воскресенье).
ddddd	То же, что и встроенный краткий формат даты.
dddddd	То же, что и встроенный длинный формат даты.
w	Номер дня в неделе (от 1 до 7).
ww	Номер недели в году (от 1 до 53).
m	Номер месяца, состоящий из одной или двух цифр (от 1 до 12).
mm	Номер месяца, состоящий из двух цифр (от 01 до 12).
mmm	Первые три буквы названия месяца (янв-дек).
mmmm	Полное название месяца (январь-декабрь).
q	Номер квартала в году (от 1 до 4).
y	Номер дня в году (от 1 до 366).
yy	Последние две цифры номера года (от 01 до 99).
yyyy	Номер года полностью (от 0100 до 9999).
h	Значение часов, состоящее из одной или двух цифр (от 0 до 23).
hh	Значение часов, состоящее из двух цифр (от 00 до 23).
n	Значение минут, состоящее из одной или двух цифр (от 0 до 59).
nn	Значение минут, состоящее из двух цифр (от 00 до 59).

Символ	Описание
s	Значение секунд, состоящее из одной или двух цифр (от 0 до 59).
ss	Значение секунд, состоящее из двух цифр (от 00 до 59).
tttt	То же, что и встроенный длинный формат времени.
AM/PM	12-часовой формат времени с добавлением прописных букв «AM» или «PM».
am/pm	12-часовой формат времени с добавлением строчных букв «am» или «pm».
A/P	12-часовой формат времени с добавлением прописных букв «A» или «P».
a/p	12-часовой формат времени с добавлением строчных букв «a» или «p».
AMPM	12-часовой формат времени с использованием соответствующего указателя «до полудня/после полудня», определенного в региональных настройках Microsoft Windows.

#### **Б.4 Символы для создания пользовательских форматов типа «Текстовый» и «Поле МЕМО»**

Символ	Описание
@	для вывода в позиции любого символа или пробела;
&	для вывода в позиции любого символа;
<	преобразует все символы к нижнему регистру;
>	преобразует все символы к верхнему регистру

#### **Б.5 Символы для создания пользовательских форматов типа «Числовой» и «Денежный»**

Символ	Описание
0	для вывода в позиции цифры; если на этой позиции в числе нет цифры, выводится 0;
#	для вывода в позиции цифры (или пробела);
%	значение умножается на 100 и дополняется знаком %

#### **Б.6 Форматирование логических значений**

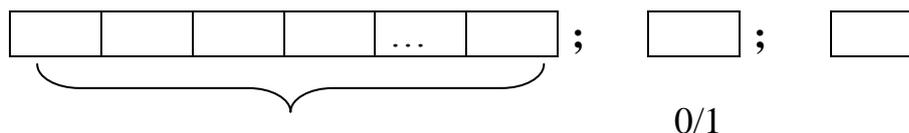
Для вывода данных типа «Логический» можно выбрать один из трех стандартных форматов: «Да/Нет» (по умолчанию), «Истина/Ложь», «Вкл/Выкл». Можно также определить собственные слово или фразу, выводимые в качестве логических значений «Да» и «Нет», и цвет символов при выводе. Например, “проживает в общежитии”[синий]; “не проживает в общежитии” [красный].

## Приложение В

### Символы для формирования маски ввода

Свойство **Маска ввода** позволяет указать шаблон, облегчающий ввод данных в поле. С помощью маски ввода можно обеспечить соответствие вводимых данных определенному формату, а также добавление к данным постоянных символов, например, точки (.) при указании даты или дефиса (-) при вводе номера телефона.

Маска ввода может состоять из трех частей, разделенных точкой с запятой:



Собственно маска ввода  
(символы маски и постоянные символы)

0/1

символ-  
заполнитель

Необязательная вторая часть указывает Access, нужно ли сохранять постоянные символы маски в этом поле (0 – сохранять, 1 – не сохранять).

Основные символы, используемые для задания маски:

Символ	Описание
<b>0</b>	в данную позицию должна быть введена только цифра;
<b>9</b>	цифра или пробел;
<b>#</b>	цифра, пробел, знаки + и -;
<b>L</b>	только буква;
<b>A</b>	буква или цифра;
<b>&lt;</b>	преобразует все символы к нижнему регистру;
<b>&gt;</b>	к верхнему регистру;
<b>!</b>	указывает, что маска ввода заполняется справа налево;
<b>\</b>	указывает, что следующий символ является постоянным;
<b>“литерал”</b>	указывает, что приведенные символы являются постоянными;
<b>., : ; - /</b>	десятичный разделитель, разделитель групп разрядов, даты и времени (согласуются с символами разделителей, определенных в окне <b>Язык и стандарты</b> панели управления Windows)

Маска ввода	Образцы значений
(000) 000-0000	(206) 555-0248
(999) 999-9999	(206) 555-0248 ( ) 555-0248
(000) AAA-AAAA	(206) 555-TELE
#999	-20 2000
>L????L?000L0	GREENGR339M3 MAYR452B7
>L0L 0L0	T2F 8M4
00000-9999	98115- 98115-3007
>L<???????????????	Мария Ольга

## Приложение Г - Варианты заданий

1. Информационная система «Экологическая служба страны» содержит данные об отделениях ЭСС (название отделения, город, телефон, год создания), о предприятиях, по вине которых произошли экологические катастрофы (ЭК) (название предприятия, город, отрасль промышленности (химическая, металлургическая, деревообрабатывающая, ...), год создания, телефон), экологических катастрофах (предприятие, тип катастрофы (выброс окиси углерода, сброс отходов в реку, утечка отравляющих веществ, взрыв, пожар,...), дата, количество пострадавших людей, последствия (вид(гибель животных, гибель людей, отравление, радиоактивное заражение, загрязнение атмосферы,...), материальный ущерб)), сведения об участии ЭСС в устранении ЭК (отделение ЭСС, количество людей, принимавших участие, количество дней).

2. Для автоматизации учета шефской помощи предприятиями детским садам города необходимы следующие сведения: о детских садах (номер детсада, название, район города, тип собственности (государственный, частный, предприятия,...), год открытия, телефон, дети (ФИО, группа (младшая, средняя, старшая, подготовительная), дата рождения)), о предприятиях, оказывающих помощь (название, форма собственности (государственное, частное, ЗАО, ОАО, ООО,...), телефон, год начала работы, количество сотрудников), об оказанной помощи детским садам (вид помощи (поставка продуктов, мебели, медикаментов, ремонт, оплата экскурсии,...), стоимость, дата оказания) и о помощи отдельным детям (вид помощи (оплата лечения, денежная помощь, покупка детского питания, покупка одежды,...), стоимость, дата оказания).

3. В информационной системе об ассортименте книжных магазинов города обрабатываются данные о магазинах (номер магазина, район города, тип собственности (государственный, частный, ОАО, ЗАО, ООО,...), год открытия), книгах (название книги, автор (ФИО, страна, дата рождения, дата смерти), издательство (название издательства, город, год создания, телефон), год издания, язык, в какие магазины поступила (цена экземпляра, количество экземпляров, дата поступления в магазин)).

4. База данных «Кондитерские фабрики страны» содержит такую информацию: название фабрики, город, телефон, год введения в строй, тип собственности (ОАО, ЗАО, ООО, частная, государственная), выпускаемая фабрикой продукция (наименование продукции, тип (карамель, ирис, шоколад, мармелад, печенье ...), расфасовка (коробка, весовая, штучная,...)), поступления партий продукции в магазины (магазин (номер или название, район города, год открытия, телефон, число сотрудников), продукция, дата поступления, объем (в кг или штуках), цена единицы (кг или штуки)).

5.База данных «Зоопарки» содержит следующую информацию: название зоопарка, тип (частный, государственный, ОАО, ЗАО, ООО,...), город, год открытия, телефон, сотрудники (ФИО, дата рождения, должность, оклад), имеющиеся животные (кличка, класс (земноводные, пресмыкающиеся, птицы, млекопитающие, рыбы,...), вид (медведь, волк, слон,...), родина (указать часть света, остров, океан,...), средняя продолжительность жизни, сотрудник, под наблюдением которого находится животное, дневной рацион (количество кг), дата поступления в зоопарк, случаи оказания животным медицинской помощи (вид помощи (прививка, операция, лечение, ...), стоимость, дата оказания)).

**Примечание: данные вид и класс связаны между собой.**

6.Информационная система о функционировании кинологовических клубов города должна обрабатывать данные о клубах (название клуба, район города, год создания, телефон, вступительный взнос), сведения о собаках, которые прикреплены к клубам (кличка, порода, год рождения, стоимость щенка, хозяин (ФИО, социальное положение (служащий, учащийся, предприниматель, пенсионер, рабочий, ...), дата рождения, адрес)), сведения об участии собак в соревнованиях (соревнование (название соревнования, дата проведения, взнос, количество зрителей), награда (вид награды, денежный эквивалент)).

7.Система учета кредитной деятельности банков страны должна содержать такие данные: название банка, тип (государственный, коммерческий, акционерный, ...), год создания, уставной фонд, телефон, сведения о выданных кредитах (клиент (ФИО или название фирмы, город, телефон, номер счета), вид кредита (потребительский, под векселя, под ценные бумаги, краткосрочный, долгосрочный,...), сумма, годовой процент, дата выдачи), возврат кредитов (сумма, дата возврата).

**Примечание: клиент может брать кредиты в нескольких банках; кредит может возвращаться по частям.**

8.Для автоматизации учета издательской деятельности в стране необходимы сведения об издательствах (название, город, тип собственности (государственное, частное, ООО, ОАО, ЗАО,...), год образования, телефон, адрес) и об изданных книгах (название книги, автор (фамилия, страна, дата рождения, дата смерти), жанр (детектив, фантастика, исторический, приключения,...), язык издания, цена, год издания, переплет (твердый, мягкий, целлофанированный), тип издания (отдельное произведение, собрание сочинений, избранные произведения), тираж).

**Примечание: одна и та же книга может издаваться разными изданиями (и по разной цене).**

9.База данных «Ботанические сады страны» включает следующую информацию: название ботанического сада, город, тип (частный, государственный, ...), год открытия, сотрудники (ФИО, должность, дата рождения, оклад), растения (номер растения, номер участка произрастания,

семейство (лилейные, крестоцветные, розоцветные, пасленовые, бобовые,...), вид (тюльпан, гиацинт, лилия, яблоня, груша, роза, боярышник,...), срок жизни (количество лет), дата посадки, стоимость саженца, сотрудник, ухаживающий за растением), сведения об удобрении растений (удобрение (название, объем упаковки, стоимость упаковки), количество упаковок, дата выполнения).

**Примечание: данные семейство и вид связаны между собой.**

**10.**Для системы, автоматизирующей учет наличия лекарств в аптеках города, необходимы следующие сведения: наименование лекарства, фирма-производитель (название фирмы, тип собственности (государственная, частная, акционерная,...), страна, год образования, адрес), форма выпуска лекарства (капли, таблетки, мазь,...), фармакологическая группа (сердечно-сосудистое, болеутоляющее, противовоспалительное,...), в какие аптеки поступило лекарство (сведения об аптеке (номер аптеки, тип (государственная, частная, акционерная,...), район города, телефон), дата поступления партии в аптеку, количество упаковок лекарства в партии, цена одной упаковки).

**Примечание: для упрощения можно считать, что каждое лекарство производится только одной фирмой.**

**11.** Для автоматизации учета пользователей электронной почты необходима информация о фирмах-провайдерах (название фирмы, тип собственности (государственная, частная, ЗАО, ОАО,...), адрес, телефон, год начала работы), о заключенных с ними договорах (абонент (ФИО или название предприятия, тип (частное лицо, ВУЗ, школа, предприятие,...), физический адрес, адрес электронной почты), дата подключения, стоимость подключения, стоимость пересылки 1 Мб информации) и о предоставленных провайдерами услугах (абонент, дата предоставления, объем сообщения (в Мб)).

**12.**Информационная система, предназначенная для учета результатов техосмотров автомобилей, должна содержать: сведения о ГАИ (район города, телефон, адрес, сотрудники (ФИО, звание, год рождения)), владельцах автомобилей (ФИО, социальное положение (предприниматель, служащий, учащийся, рабочий, пенсионер, домохозяйка,...), дата рождения, телефон, автомобили (номер, марка, год приобретения, цена)), техосмотрах (дата, стоимость, кто проводил, выявленные недостатки (вид (не отрегулированное освещение, плохие тормоза, люфт руля, отсутствие аптечки, разбитые фары, ...), штраф)).

**Примечание: каждый владелец автомобилей прикреплен только к одному ГАИ.**

**13.**Для автоматизации работы администратора гостиницы необходима информация о номерах гостиницы (номер, этаж, тип (люкс, полулюкс, отдельный, номер на несколько человек,...), размер оплаты за сутки), о вселении в гостиницу (ФИО, город проживания, дата рождения, социальное положение (банковский служащий, предприниматель, рабочий, учащийся,...),

дата поселения, на сколько суток), о предоставленных дополнительных услугах (название услуги (использование телевизора, холодильника, междугородний телефонный разговор, отправка телеграммы,...), цена за условную единицу (условные единицы: сутки, минута, слово,...), дата предоставления, продолжительность (количество условных единиц)).

**14.** Система обработки информации о деятельности страховых компаний страны использует сведения о компаниях (название компании, тип (частная, государственная, акционерная,...), год получения лицензии, город, в котором находится центральное отделение, телефон, филиалы (название филиала, город, адрес, телефон, количество сотрудников)) и заключенных с филиалами договоров (клиент (ФИО, дата рождения, социальное положение (служащий, предприниматель, учащийся, рабочий, пенсионер,...), телефон), вид страхования (недвижимость, имущество, от несчастного случая, к бракосочетанию, до совершеннолетия,...), сумма страхования, дата заключения договора).

**Примечание: клиент может заключать договора с филиалами разных компаний.**

**15.** Информационная система «Мясокомбинаты области» должна обрабатывать сведения о мясокомбинатах (название мясокомбината, город области, тип собственности (частный, государственный, ООО, ЗАО, ОАО,...), год начала работы, телефон, выпускаемая мясокомбинатом продукция (вид продукции (буженина, колбаса, рулет, корейка, паштет,...), название, годовой объем выпуска (в кг), цена 1 кг)) и заказах на продукцию (заказчик (название организации, город, телефон, адрес), дата заказа, объем).

**16.** Для учета деятельности рекламных агентств страны используются сведения об агентствах (название агентства, город, тип (частное, государственное, акционерное,...), год создания, расчетный счет, телефон, виды производимой в агентстве рекламной продукции (видеоклип, радиорепортаж, рекламный щит, реклама на общественном транспорте, в газете, журнале, листовка,...)) и заказах на изготовление рекламной продукции (заказчик (название предприятия или ФИО, город, телефон, адрес), вид рекламной продукции, цена экземпляра, количество экземпляров, дата заказа, срок изготовления (в днях), форма расчета (наличная, безналичная)).

**Примечание: заказчик может заказывать различные виды рекламы в разных агентствах.**

**17.** База данных «Станции техобслуживания города» включает информацию о станциях (название станции, форма собственности (государственная, частная, ОАО, ЗАО, ООО,...), район расположения, год открытия, ремонтные услуги, оказываемые станцией (вид ремонта (замена масла, регулировка тормозной системы, регулировка электрооборудования, переборка двигателя, замена рессор,...), номинальная стоимость ремонта)), о владельцах автомобилей, пользующихся услугами станций (ФИО, автомобили

(номер, марка, стоимость), социальная группа (предприниматель, банковский служащий, инженер, рабочий,...), адрес, телефон) и о произведенных на станциях ремонтах (вид ремонта, где и когда производился, реальная стоимость ремонта).

**18.**Для автоматизации учета работы СОБЕСов используются данные о самих учреждениях (район города, год открытия, количество сотрудников, адрес, телефон, клиенты (ФИО, дата рождения, тип (ветеран ВОВ, ветеран труда, инвалид 1 группы, инвалид 2 группы, инвалид 3 группы, пенсионер,...), размер пенсии, имеющиеся льготы (бесплатный проезд в общественном транспорте, бесплатный проезд в междугородних автобусах, в пригородном транспорте, 50% оплата квартиры, 50% оплата коммунальных услуг, бесплатные медикаменты, 50% оплата за медикаменты,...))), о предоставлении клиентам СОБЕСов помощи (вид помощи (денежная, продуктовая, предметы быта, оплата лечения, санаторная путевка,)), денежный эквивалент, дата предоставления).

**19.**Для автоматизации деятельности органов государственной криминалистики необходимы следующие сведения о нарушениях: анкетные данные правонарушителя (ФИО, дата рождения, номер дела в картотеке, адрес последнего места проживания, судимости (дата, название суда (районный, городской, областной, верховный), статья (206- хулиганство, 93- умышленное убийство, 84 - кража государственного имущества, 140 - кража имущества граждан,...), количество лет, тип приговора (лишение свободы, ссылка, исправительные работы, штраф, оправдательный приговор,...), место заключения (колония строгого режима, колония общего режима, колония для несовершеннолетних,...), стоимость процесса)), жалобы осужденных на решение суда (дата подачи жалобы, ее направленность (несогласие со свидетельскими показаниями, несогласие с протоколом №.., несогласие с приговором,...), в какой суд подана, дата разбирательства жалобы, результат (отменить приговор, подтвердить, изменить)).

**20.**Для автоматизации обработки информации о работе крупнейших предприятий автомобилестроения необходимы данные об этих предприятиях (название, страна, тип собственности (государственное, частное, ЗАО, ОАО, ООО,...), год начала функционирования, телефон, факс, выпускаемые автомобили (марка, год начала выпуска, цена единицы, технические характеристики (мощность, максимальная скорость, расход топлива))) и о заказах на автомобили (заказчик (ФИО или название предприятия, страна, город, телефон), марка автомобиля, количество единиц, дата заказа).

**21.**Система учета поставки овощей в магазины города содержит информацию о предприятиях, производящих продукцию (название предприятия, район области, тип собственности (государственная, частная, ЗАО, ОАО, ООО,...), число сотрудников, год начала работы, телефон,

производимая предприятием продукция (вид (огурцы, помидоры, картофель, капуста,...), наименование сорта)), и о закупках продукции (магазин (название или номер магазина, район города, адрес, телефон), продукция, количество килограмм, цена килограмма, дата поставки).

**22.** Для автоматизации деятельности горно-спасательной службы (ГСС) Донбасса необходимы данные об отделениях службы (название отделения, город, телефон, год создания, сотрудники (ФИО, должность (командир отряда, командир взвода, командир роты, боец,...), оклад, с какого года работает в отделении, дата рождения)) и шахтах (название шахты, город, максимальная глубина, площадь выработок, участки (номер, длина, тип выработки (вертикальная, горизонтальная, наклонная), тип угля (антрацит, коксующийся, энергетический, рядовой, ...), год ввода в действие, чрезвычайные происшествия на участках (тип (обвал лавы, пожар, взрыв газа, затопление, ...), причина (объективная, субъективная), дата, материальный ущерб, сотрудники ГСС, участвующие в устранении ЧП (сколько дней работал каждый из них))).

**23.** Для автоматизации учета обеспеченности населения города хлебобулочными изделиями необходима информация о предприятиях, выпекающих хлеб (название или номер предприятия, район города, тип собственности (государственное, частное, ООО, ОАО, ЗАО,...), год начала работы, телефон, выпускаемая предприятием продукция (название продукции (хлеб “Дарницкий”, хлеб “Ромашка”, хлеб с отрубями, бублик, булочка с изюмом, батон,...), сорт муки (белая высшего сорта, белая первого сорта, ржаная, с отрубями, ...), вес единицы)) и о поставках в магазины (магазин (номер или название магазина, район города, адрес, телефон), продукция, количество единиц, цена единицы, дата поставки).

**24.** База данных «Банки города» содержит сведения об отделениях банков (номер (название) отделения, банк (название банка, тип (государственный, акционерный, коммерческий,...), уставной фонд, операции с валютой (возможны или нет), сфера обслуживания (юридические лица, физические лица, те и другие), дата получения лицензии), телефон, район города, в котором находится отделение, год открытия) и выполненных операциях (клиент (ФИО или наименование организации, юридическое или физическое лицо, телефон, номер счета, дата открытия счета), вид операции (банковский перевод, инкассо, аккредитив отзывной, аккредитив безотзывной,...), сумма, дата выполнения).

**25.** Для учета заболеваемости детей, посещающих детские сады города, необходимы следующие сведения о детях: ФИО, группа (младшая, средняя, старшая, подготовительная), дата рождения, пол, посещаемый детский сад (номер, название, район города, тип собственности (государственный, частный, предприятия,...), год открытия), имеющиеся прививки (против каких болезней и когда сделаны), перенесенные заболевания (название заболевания, дата начала, продолжительность болезни (количество дней), стоимость лечения,

полученные осложнения (новое заболевание (воспаление легких, ангина, ...), стоимость лечения)).

**26.** Для автоматизации учета лечения больных стационара используется информация о больных (ФИО, дата рождения, социальное положение (предприниматель, служащий, рабочий, пенсионер,...), заболевание (пневмония, инфаркт, инсульт, гепатит,...), дата поступления в больницу, лечащий врач, назначенные процедуры (вид (инъекции, массаж, лечебная физкультура, электросон, прогревание,...), интенсивность (ежедневно, 3 раза в день, через час,...), количество дней, стоимость 1 дня)) и о врачах (ФИО, отделение (название отделения (терапевтическое, хирургическое, лор, дерматологическое,...), число коек, телефон), категория (высшая, первая, вторая, третья), дата поступления на работу, оклад).

**27.** Для автоматизации учета профсоюзной помощи сотрудникам ВУЗов города необходима информация о ВУЗах (краткое и полное название, район города, в котором ВУЗ находится, телефон профкома, адрес, факультеты (краткое и полное название, год создания, телефон, сотрудники (ФИО, должность, оклад, дата рождения, дата поступления на работу))) и случаях оказания помощи (вид помощи (премия, денежная помощь, путевка,...), сумма, дата).

**28.** База данных «Футбольные клубы страны» содержит следующие сведения о клубах: название клуба, город, год создания, принадлежность к лиге (высшая, первая, вторая, третья), ФИО председателя клуба, контактный телефон, игроки (ФИО, положение в команде (вратарь, нападающий, защитник, полузащитник,...), дата рождения, с какого года играет в команде, стоимость контракта), игры с зарубежными командами (противник (название команды, страна, старший тренер), дата проведения игры, количество пропущенных мячей, количество забитых мячей, участие в игре (игрок, денежная премия, допущенные нарушения (желтая карточка, красная карточка, пенальти, штрафной,...))).

**29.** Информационная система, предназначенная для учета деятельности международных благотворительных фондов в стране, содержит сведения о фондах (название фонда, страна, год создания, контактный телефон, источники формирования капитала фонда (тип пожертвования (пожертвования частных лиц, пожертвования организаций, государственная помощь, пожертвования церкви,...), сумма, тип валюты, дата внесения пожертвования)) и об оказании фондами помощи (гражданин (ФИО, город, категория (ветераны разных войн, инвалид войны, инвалид труда, инвалид детства, многодетная семья,...), дата рождения), сумма помощи, дата оказания).

**30.** В БД «Международные выставки» содержится информация о выставках, проведенных на территории страны: название выставки («Уголь-96»,

“Медицина-98”, “Средства защиты окружающей среды”,...), выставочный центр (название (“Экспо-Донбасс”, “Кристалл”, “Дружба”,...), город, год создания, площадь (кв.м.), телефон), тематика выставки, дата начала, продолжительность (количество дней), количество посетителей, участники выставки (название компании, страна, форма собственности (государственная, частная, акционерная,...), телефон, представленные экспонаты (название, тип экспоната (прибор, машина, предмет быта,...), страховочная стоимость экспоната, год создания)).

**31.**Для автоматизации учета работы налоговой службы Донецкой области необходима следующая информация: об отделениях налоговой инспекции (номер отделения, тип (городская, районная, областная), название (Донецкая область, г. Донецк, Ворошиловский район, Киевский район,...), год начала функционирования, телефон, адрес, сотрудники (ФИО, дата рождения, должность, оклад, образование (высшее, среднетехническое, среднее, среднее специальное))) и о предприятиях, выплачивающих налоги (название предприятия, форма собственности (государственное, частное, ЗАО, ОАО, ООО,...), телефон, год начала работы, количество сотрудников, платежи налоговой службе (вид платежа (налог на добавленную стоимость, налог с прибыли, штраф за неуплату налогов,...), дата, сумма, кто оформлял (сотрудник инспекции))).

**32.**Для автоматизации учета мировой добычи минералов необходима информация о месторождениях (название месторождения, страна, год начала функционирования, тип собственности (государственное, частное, ОАО, ЗАО,...), выработки (номер выработки, длина, добываемый камень (алмаз, топаз, гранат, хризолит, аметист, ...), вид добычи (подземная, наземная))) и сведения о добыче минералов (где, какой камень, тип камня (ювелирный, поделочный, технический,...), количество карат, цена 1 карата, дата добычи).

**33.**База данных «Подписка города» содержит такую информацию: почтовые отделения (номер отделения, район города, телефон, год создания отделения, количество почтальонов), подписчики (ФИО, социальное положение (служащий, предприниматель, рабочий, пенсионер, учащийся,...), дата рождения, адрес), выписанные подписчиками издания (сведения об издании (название, тип (газета ежедневная, газета еженедельная, журнал, приложение к газете, книжное обозрение,...), язык издания, цена за месяц, издательство (название, тип (частное, государственное, акционерное,...), город, телефон)), срок подписки (количество месяцев), дата подписки).

**Примечание: каждый подписчик связан только с одним почтовым отделением.**

**34.**Для учета работы ломбардов города необходима информация о ломбардах (название, тип собственности (государственный, частный, ЗАО, ОАО,...), адрес, телефон) и выданных в них ссудах (клиент (ФИО, дата

рождения, социальное положение (домохозяйка, предприниматель, рабочий, служащий,...), домашний адрес, телефон), размер ссуды, название предмета, взятого под залог (часы, ювелирное изделие, картина,...), дата выдачи ссуды, дата возврата, пеня за каждый просроченный день (в процентах отобщей стоимости), возврат (возвращена, не возвращена)).

**35.** Для системы учета деятельности туристических агентств страны необходимы следующие сведения: агентства (название агентства, город, телефон, номер лицензии, форма собственности (частное, государственное, акционерное,...), дата создания), маршруты (название маршрута, дата начала, транспорт (вид (самолет, поезд, автобус,...), стоимость), проживание (название отеля, разряд), питание (имеется, не имеется), продолжительность (количество дней), стоимость), приобретенные на маршрут путевки (ФИО клиента, дата рождения, социальное положение (домохозяйка, предприниматель, рабочий, служащий,...), домашний адрес, телефон).

**36.** Для учета платных занятий в детских садах БД должна содержать следующую информацию: название детского сада, район города, тип собственности (частный, государственный, предприятия,...), год начала функционирования, телефон, дети (ФИО, дата рождения, пол, группа (младшая, средняя, старшая, подготовительная), участие детей в дополнительных платных занятиях (вид занятия (плавание, бальные танцы, изучение иностранного языка,...), время, отводимое на занятие (мин), стоимость одного занятия, дата проведения занятия)).

**37.** В БД, содержащей сведения о международных выставках художественных произведений, должны быть представлены такие сведения: музеи (название музея (Лувр, Эрмитаж, Русский музей, Третьяковская галерея,...), страна, год создания музея, телефон, экспонаты (название экспоната, тип (картина, скульптура, ювелирное изделие, изделие из керамики,...), автор, век создания, в каком году экспонат попал в музей, страховочная цена)), участие экспонатов в выставках (экспонат, выставка, страна, дата начала, дата завершения, количество посетителей)).

**38.** Для автоматизации учета заказов лечебных заведений города на лекарственные препараты информационная система должна содержать следующие данные: номер больницы, тип (областная, городская, районная, ведомственная), район города, в котором больница находится, телефон, год создания, отделения больницы (название отделения (терапевтическое, кардиологическое, хирургическое, ...), число мест, количество врачей, заказы на лекарства (наименование лекарства, форма выпуска (капли, таблетки, мазь,...), фармакологическая группа (сердечно-сосудистое, болеутоляющее, противовоспалительное,...), фирма-производитель (название, тип собственности (государственная, частная, акционерная,...), страна, год образования), дата заказа, количество упаковок, цена одной упаковки)).

**39.** Для учета рекламной деятельности телеканалов система должна обрабатывать следующую информацию: сведения о телеканалах (название канала, город, тип собственности (государственный, частный, ЗАО, ООО, ...)), начало работы (время суток), конец работы, стоимость 1 минуты рекламного времени (тип времени (утреннее, дневное, вечернее, ночное), стоимость)), заказы на рекламу (заказчик (название заказчика, город, телефон, адрес), канал, дата заказа, заказываемое время (тип времени, количество минут), продолжительность (количество дней)).

**40.** Информационная система «Кинопрокат» обрабатывает следующие сведения: о кинотеатрах (название, район города, тип собственности (государственный, частный, ОАО, ЗАО, ООО,...)), число мест, телефон, год ввода в эксплуатацию), фильмах (название, киностудия (название киностудии, страна, год создания), режиссер, год создания, продолжительность (в минутах)) и прокате фильмов (в каком кинотеатре, сеансы (утренний- 0.5, дневной- 1, вечерний- 1.5 от стоимости билета на дневной сеанс), цена билета на дневной сеанс, количество зрителей, дата).

**41.** Система учета работы комиссионного магазина должна содержать данные о комитентах (ФИО, район проживания, социальное положение (рабочий, служащий, пенсионер, домохозяйка, предприниматель,...)), место работы, дата рождения, телефон), принимаемых товарах (номер квитанции, группа товара (посуда, одежда, обувь, игрушки,...)), наименование товара, дата приема, количество единиц, цена за единицу) и о реализации товаров (товар, дата, количество проданных единиц).

**42.** База данных «Мебельные фабрики страны» содержит следующую информацию: сведения о фабриках (название фабрики, город, тип собственности (частная, государственная, ОАО, ЗАО, ООО,...)), число сотрудников, год открытия, телефон, адрес, выпускаемая фабрикой продукция (наименование продукции, оптовая цена, годовой объем выпуска (в штуках)), сведения о поставках продукции (заказчик (название фирмы, город, адрес, телефон), продукция, количество единиц, дата поставки).

**43.** Для автоматизации учета работы косметических магазинов города необходимы такие данные: магазины (номер магазина, район города, год начала работы, количество сотрудников, телефон), поставки косметической продукции (фирма-производитель (название фирмы, тип собственности (государственный, частный, ОАО, ЗАО,...)), город, телефон), продукция (название, группа (крем, мыло, шампунь, духи, пудра,...)), дата поставки, количество единиц, цена единицы).

**44.** Для учета работы парикмахерских города необходима информация о парикмахерских (номер, район города, разряд (высший, первый, второй), тип собственности (частная, государственная, акционерная,...)), год начала

функционирования, телефон), клиентах (ФИО, дата рождения, социальная группа (предприниматель, банковский служащий, инженер, рабочий,...), домашний адрес)) и оказанных клиентам услугах (парикмахерская, название услуги (стрижка, завивка, укладка, маникюр, массаж,...), стоимость (зависит от разряда парикмахерской), дата оказания услуги).

**45.** Для учета междугородних перевозок грузов на автотранспортных предприятиях (АТП) города необходимы сведения о предприятиях (название АТП, район города, тип собственности (государственное, частное, ЗАО, ОАО,...), год начала функционирования, телефон, водители (ФИО, дата рождения, дата поступления на работу, категория (А, В, С), оклад)) и о перевозках (груз (наименование, вес), куда перевозится (город), дата выезда, стоимость перевозки, автомобиль (номерной знак, технические характеристики (марка, грузоподъемность, номинальный расход топлива), водитель)).

**46.** Для автоматизации учета предоставления услуг связи фирмами-провайдерами Internet необходима информация о фирмах-провайдерах (название, тип собственности (государственный, частный, ЗАО, ОАО,...), адрес, телефон, год начала работы), о заключенных с ними договорах (абонент (ФИО или название предприятия, тип (частное лицо, школа, ВУЗ, предприятие,...), физический адрес, IP-адрес, телефон), дата подключения, стоимость подключения, стоимость 1 Мб информации) и о сеансах модемной связи (дата и время сеанса, количество Мб, тариф (бизнес-время- 100% оплата, вечерний тариф- скидка 20%, ночной тариф- скидка 50%,...)).

**Примечание: один абонент может заключать несколько договоров.**

**47.** БД «Видеотеки города» содержит следующую информацию: название видеотеки, телефон, район города, адрес, номер лицензии, режим работы (начало и конец (в час.)), предоставленные видеотекой услуги (вид услуги (прокат, запись, продажа кассет,...), дата предоставления, цена, фильм (киностудия (название, страна, год создания), год выхода, режиссер, главный актер (ФИО, страна, дата рождения, гонорар актера), продолжительность фильма)).

**48.** БД «Фотоателье» содержит следующую информацию о фотомастерских города: номер или название фотоателье, разряд (высший, первый, второй), тип собственности (государственное, частное, ООО, ОАО, ЗАО,...), район города, год открытия, телефон, предоставляемые услуги (вид (фотография цветная, фотография черно-белая, проявка пленки, печать,...), размер, срок выполнения (количество дней)), сведения о заказах (заказчик (ФИО, район города, социальное положение (предприниматель, студент, школьник, рабочий, ...), дата рождения), количество штук, цена 1 шт., дата выполнения).

**49.** Для автоматизации учета работы сети химчисток города необходимы такие сведения: название химчистки, район города, тип собственности (государственная, частная, ЗАО, ОАО,...), год открытия, телефон, предоставляемые услуги (вид услуги (чистка верхней одежды, чистка меховых изделий, удаление пятен, стирка, окраска одежды,...), срок выполнения), заказы на чистку (заказчик (ФИО, дата рождения, социальное положение (домохозяйка, предприниматель, рабочий, служащий,...), телефон, домашний адрес), вид услуги, тариф (обычный или срочный (на 50% дороже)), цена, дата заказа).

**50.** Для автоматизации деятельности ветеринарной службы города необходимы такие данные: пункты ветеринарной помощи (номер пункта, район города, телефон, адрес, сотрудники (ФИО, должность (хирург, ветеринар, медсестра, ...), оклад, дата рождения)), сведения об оказании в пунктах медицинской помощи животным (животное (кличка, вид (собака, кошка, птица, грызун, ...), год рождения, стоимость, хозяин (ФИО, социальное положение (служащий, пенсионер, предприниматель, учащийся, ...), дата рождения, телефон)), выполненные при оказании помощи процедуры (вид процедуры (прививка, лечение, операция, ...), стоимость, дата оказания, кто выполнил (сотрудник пункта))).

**51.** Для автоматизации учета ЧП на автомобильных дорогах Донецкой области необходима следующая информация: сведения о ГАИ (место нахождения (Донецк или район области (Волновахский, Марьинский,...)), телефон, адрес, сотрудники (ФИО, звание, дата рождения)), сведения о ДТП (место (район области или город), дата, время, участники (номер машины, марка, стоимость машины, размер страховки, владелец (ФИО, государство, социальное положение (служащий, пенсионер, предприниматель, рабочий, ...), дата рождения), последствия для машины (вид поломки (дверцы, двигатель, рулевое колесо, бампер, ...), стоимость ремонта), перечень сотрудников, участвующих в разборе ДТП (сотрудник, количество затраченных часов))).

**52.** Для автоматизации учета работы службы скорой помощи города нужны такие сведения: пункты скорой помощи (номер пункта, район города, количество сотрудников, телефон, адрес), заявки на вызов скорой помощи (больной (ФИО, дата рождения, социальное положение (пенсионер, рабочий, служащий, предприниматель, ...), телефон, адрес), дата и время вызова, причина вызова (высокая температура, сердечный приступ, головная боль, высокое давление, боль в животе, ...), оказанные при вызове процедуры (процедура (укол, электрокардиограмма, кислородная подушка, таблетки, ...), денежный эквивалент процедуры)).

**53.** Для автоматизации учета работы нефтеперерабатывающих заводов страны необходима следующая информация: сведения о заводах (название, город, тип собственности (государственный, частный, ЗАО, ООО, ...), год

начала функционирования, телефон, страны-поставщики нефти (Казахстан, Россия, Туркмения, ...), виды выпускаемого топлива (вид (дизельное, бензин А76, бензин А80, А92, А95, авиационный керосин,...), годовой объем (в тоннах), цена 1 тонны)), сведения о заказах на нефтепродукты (заказчик (название предприятия, город, телефон), вид топлива, объем (в тоннах), дата заказа).

**54.** Система учета деятельности компьютерных клубов города обрабатывает такие данные: сведения о клубах (название клуба, район города, начало работы (час), конец работы (час), количество компьютеров, стоимость 1 часа времени (тип времени (утреннее, дневное, вечернее, ночное), стоимость), загруженность компьютеров (пользователь (ФИО, дата рождения, социальное положение (школьник, студент ВУЗа, служащий, рабочий, безработный,...), место работы (учебы)), используемая программа (Word, СУБД, Internet-программы, игры, графика, ... ), дата, тип времени, количество часов).

**55.** В информационной системе, содержащей сведения о музыкальных магазинах города нужны данные о магазинах (название магазина, район города, тип собственности (государственная, частная, ЗАО, ООО, ...), год открытия, телефон) и музыкальных альбомах (название альбома, тип среды (CD, DVD, кассета,...), какой фирмой выпущен (название фирмы, город, телефон), дата выпуска, время звучания (в минутах), количество песен, в каких магазинах продается (дата поступления в магазин, количество экземпляров, цена за альбом).

**56.** Для автоматизации учета работы сети кондитерских магазинов города необходимы такие данные: магазины (номер магазина, район города, год начала работы, количество сотрудников, телефон), поставки продукции (поставщик (название, тип собственности (государственный, частный, ОАО, ЗАО,...), город, телефон), продукция (название, тип (печенье, конфеты, мармелад, шоколад,...)), расфасовка (коробка, весовая, штучная,...)), дата поставки, объем (в кг или штуках), цена единицы (кг или штуки)).

**57.** Для работы справочной системы о партиях и общественных организациях страны необходимы такие сведения: организации (название, год основания, тип (партия, движение, фронт,...), город, где находится штаб, телефон, адрес), члены организаций (ФИО, дата рождения, социальное положение (рабочий, предприниматель, домохозяйка, студент,...), дата вступления в организацию), документы, выпущенные организациями (название документа, направленность (защита окружающей среды, охрана труда,...), дата выпуска, перечень лиц, участвовавших в составлении документа), проведение акций (дата проведения, тип акции (митинг, забастовка, акция протеста, благотворительная акция,...), число участников, стоимость подготовки, лицо, ответственное за проведение акции).

**58.** Для автоматизации учета работы сети АЗС города необходима следующая информация: сведения о АЗС (название, район города, год открытия, адрес, телефон), поставки топлива (вид (дизельное, бензин А76, бензин А80, А92, А95, авиационный керосин,...), поставщик (название завода, город, тип собственности (государственный, частный, ЗАО, ООО, ...), год начала функционирования, телефон), цена за 1 л., количество литров, дата поступления).

**59.** Для автоматизации учета работы служб ремонта города необходима следующая информация: ремонтные фирмы (название, тип собственности (государственная, частная, ОАО, ЗАО,...), год начала работы, телефон, количество сотрудников, предоставляемые фирмой услуги (тип услуги (окраска стен, поклейка обоев, настил линолеума, ...), цена за 1 кв. метр)), выполненная работа (заказчик (название фирмы или ФИО физического лица, район города, адрес, телефон), вид работы, дата начала работы, дата окончания, количество кв. метров, сумма израсходованных материалов).

**60.** Для автоматизации учета пострадавших в авариях шахтеров необходимы такие данные: шахты (название шахты, город, телефон, максимальная глубина, площадь выработок, участки шахты (номер, длина, тип выработки (вертикальная, горизонтальная, наклонная), тип угля (антрацит, коксующийся, энергетический, рядовой,...), дата ввода в действие)), шахтеры (ФИО, дата рождения, адрес), чрезвычайные происшествия на участках шахт (тип (обвал лавы, пожар, взрыв газа, затопление, ...), причина (объективная, субъективная), дата, материальный ущерб, пострадавшие шахтеры (вид травмы (перелом конечностей, отравление газом, летальный исход,...), стоимость лечения)).

## Литература:

1. Иващенко В.П., Швачич Г.Г., Овсянников А.В. Системы управления базами данных. Учебное пособие.–Днепропетровск:ИМА-пресс, 2011. – 362 с.
2. Карпова Т.С. Базы данных: модели, разработка, реализация - СПб.: Питер, 2001. - 304 с: ил.
3. Тимошок Т. Самоучитель Access 2007- "Диалектика-Вильямс", 2008. - 464 с.:ил.
4. Днепров А. Microsoft Access 2007. - СПб.: Питер, 2008. - 240 с: ил.
5. Дейт К. Введение в системы баз данных. К.; М.;СПб.; Издательский дом "Вильямс", 2000.-848 с.:ил
6. Грох М., Стокман Г., Пауэлл Г. Microsoft Office Access 2007. Библия пользователя. - М.: "Диалектика", 2008. - 1200 с.
7. Иллюстрированный самоучитель по Microsoft Access.  
<http://www.taurion.ru/access>
8. Microsoft Office: справка и учебные материалы.  
<http://office.microsoft.com/ru-ru/support/?CTT=97>  
[http://www.e-reading.mobi/bookreader.php/1008436/Dneprov\\_-\\_Microsoft\\_Access\\_2007.html](http://www.e-reading.mobi/bookreader.php/1008436/Dneprov_-_Microsoft_Access_2007.html)

## Содержание:

Лабораторная работа №1 Проектирование концептуальной модели базы данных	
Лабораторная работа № 2 Создание базы данных.....	
Лабораторная работа №3 Разработка простых запросов на выборку средствами QBE .....	
Лабораторная работа №4 Разработка итоговых и перекрестных запросов средствами QBE .....	
Лабораторная работа №5 Создание запросов на языке SQL .....	
Лабораторная работа №6 Создание запросов на изменение .....	
Лабораторная работа №7 Разработка форм для работы с данными. ....	
Лабораторная работа №8 Разработка отчетов .....	
Приложение А Описание наиболее используемых типов полей таблиц Microsoft Access .....	
Приложение Б Форматы вывода значений различных типов .....	
Приложение В Символы для формирования маски ввода .....	
Приложение Г Варианты заданий	
Литература .....	

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ**  
к лабораторным работам по дисциплине

**«Организация баз данных и знаний»**

(для студентов направлений подготовки 09.03.02 "Информационные системы и технологии"  
(специальностей "Информационные технологии в медиаиндустрии и дизайне", "Системы  
автоматизированного проектирования"), 02.03.01 "Математика и компьютерные науки"  
(специальность "Компьютерное моделирование и дизайн"))

Составители:

Галина Георгиевна Шалдырван  
Наталья Стефановна Костюкова  
Алла Викторовна Чернышова  
Анна Сергеевна Кузьмичева