

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**  
**”БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА”**

**Кафедра "Информационные технологии"**

**Ю. П. ЛЫЧ, Т. Л. ШИНКЕВИЧ**

**ТЕХНОЛОГИИ**  
**ОРГАНИЗАЦИИ, ХРАНЕНИЯ**  
**И ОБРАБОТКИ ДАННЫХ**

**Часть 1 Основы создания баз данных**

**Пособие для самостоятельной работы**

**Гомель 2003**

УДК 681.3(075.8)  
Л889

Рецензент – **В. С. Серегина**, кандидат физико-математических наук, доцент, заведующая кафедрой "Прикладная математика" БелГУТа.

**Лыч Ю. П., Шинкевич Т. Л.**

Л889 Технологии организации, хранения и обработки данных. Ч.1: Основы создания баз данных: Пособие для самостоятельной работы. – Гомель: БелГУТ, 2003. – 157 с.

Рассмотрены теоретические основы формирования экономической информации, информационных совокупностей, информационных систем, вопросы организации баз данных, проектирования реляционных баз данных, системы управления базами данных, в том числе Access 2000. Дан обширный теоретический и практический материал для самостоятельной работы студентов при подготовке к учебным занятиям.

Пособие адресовано студентам экономических специальностей высших учебных заведений, а также может быть рекомендовано студентам, изучающим основы создания баз данных.

**УДК 681.3(075.8)**

© Ю.П. Лыч, Т.Л. Шинкевич, 2003.

# 1 ВВЕДЕНИЕ В ТЕХНОЛОГИИ ОРГАНИЗАЦИИ, ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

## 1.1 Предмет и содержание курса

В философском аспекте *предмет* трактуется как категория, которая означает какую-то выделенную цельную область объектов в процессе деятельности и познания человека.

Предмет учебный, или учебная дисциплина, есть дидактически обоснованная система знаний, умений и навыков, подвергшихся отбору из соответствующих отраслей науки, техники, искусства, производства и др. для их изучения в системе образования. Содержание предмета определяется учебной программой, а последовательность изучения учебных дисциплин – учебными планами.

В условиях развития компьютерных технологий возрастают требования к обобщенному представлению данных (data). Возникает необходимость обобщения структуры данных (имеющих вид различных форм, таблиц и документов), которая является основой для выбора метода размещения и поиска данных в хранилищах информации. Все это и обусловило формирование прикладной дисциплины, изучающей структуру данных и использующей теорию множеств, математическую логику, теорию графов. Рациональные структуры данных – это основа для разработки эффективных алгоритмов обработки данных, оптимизации потоков и ресурсов информации.

Логическим продолжением курса «Основы информатики и вычислительной техники» является курс «Технологии организации, хранения и обработки данных», изучающий современные информационные технологии баз данных (database) и управление ими.

Данный курс — это специальная отрасль научных знаний, посвященная изучению теоретических основ и методологии организации и проектирования баз данных.

*Предмет курса* — научные знания о структуре и организации баз данных, технологии организации, хранения и обработки данных.

*Содержание курса* составляет методика решения комплекса вопросов о базах данных и системах управления базами данных (СУБД).

*Цель курса* – дать студентам теоретические знания технологии организации и хранения данных при формировании баз данных и практические навыки по созданию баз данных и управлению ими.

*Задачи курса* – снабдить студентов знаниями:

- о структуре, свойствах и представлении экономической информации;
- об основах теории баз данных и их проектировании;
- о назначении, архитектуре, функциональных возможностях и тенденциях развития современных СУБД.

*В практическом аспекте задачи этого курса сводятся к выработке:*

- навыков создания баз данных и проектирования их объектов (запросы, формы, отчеты, макросы, модули) в среде СУБД;
- умений применять инструментальные средства СУБД для разработки прикладных программ;
- навыков управления базами данных.

Знания и практические навыки, которые студент получит в процессе изучения курса «Технологии организации, хранения и обработки данных», будут востребованы:

- при изучении дисциплины «Компьютерные информационные технологии» и ряда других специальных дисциплин;
- при выполнении научных, курсовых и дипломных работ;
- в процессе его будущей профессиональной деятельности.

## 1.2 Экономическая информация

**Экономическая информация** – основа для осуществления процессов управления, в своей совокупности образующая экономическую модель объекта управления. С помощью системы показателей экономическая информация отображает состояние отдельных элементов производства, характеризует деятельность как предприятия и любого объекта управления, так и отдельных отраслей и экономики в целом.

Экономическая информация – это совокупность данных, которые являются объектом сбора, передачи, обработки и хранения. Эти данные используются при реализации функций управления на всех уровнях.

Экономическая информация характеризуется рядом специфических особенностей:

- в основном имеет линейную форму (записывается построчно), для ее записи требуется сравнительно ограниченный алфавит;
- в основной массе дискретна;
- относительная однородность и однотипность в сочетании с массовостью и общностью источников ее возникновения позволяет получать по

одним и тем же исходным данным результатную информацию для всех служб и видов хозяйственной деятельности;

- наличие больших объемов исходной информации при сравнительно простых операциях ее обработки;
- основная часть экономической информации подлежит периодической, регулярной обработке, и незначительная часть обрабатывается случайно;
- длительность хранения;
- для массовых экономических расчетов характерен большой диапазон разрядности реквизитов (от 1 до 15 знаков и более);
- преобладание логических операций над арифметическими (в типовых процессах обработки экономической информации сравнения составляют 70 %, а операции сложения и вычитания – 30 %);
- по одним и тем же исходным данным определенного носителя, как правило, получают результатные данные в нескольких разрезах.

**Классификация экономической информации.** Классификация экономической информации имеет важное значение для изучения ее состава, образования и возможности дальнейшей обработки. Экономическую информацию можно классифицировать с учетом различных признаков: момента ее возникновения, насыщенности, полезности, степени необходимости обработки, внутреннего состава, функций управления, объективности отражения действительности, стабильности и т. д.

В процессе деятельности информационная система объекта, сам объект и внешняя среда находятся в информационной взаимосвязи (рисунок 1.1).

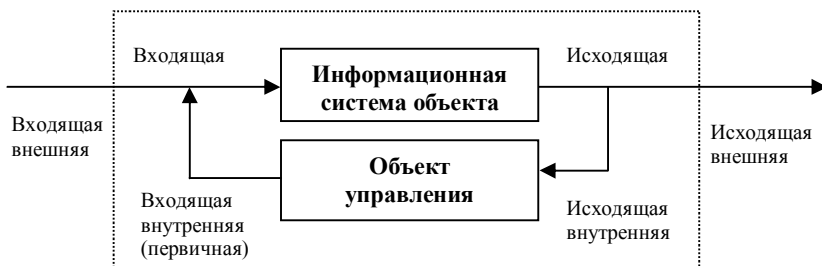


Рисунок 1.1 – Информационные взаимосвязи объекта управления

**По отношению к внешнему миру, объекту управления и информационной системе объекта управления** вся информация системы разделяется на входящую, исходящую, информацию памяти и внутреннюю. Информация, поступающая в информационную систему, называется **входящей**. Она воздействует извне на систему объекта и, в свою очередь, подразделяется на внешнюю и внутреннюю – поступающую от объекта управления. Входящую внутреннюю информацию называют также первичной, так как она отображает происходящие изменения объекта регулирования. Пер-

вичная информация играет важную роль: от ее достоверности и полноты зависит точность резульатной информации.

Информация, которая выходит из данной информационной системы, называется *исходящей* и также разделяется на внешнюю (для других предприятий и вышестоящих организаций) и внутреннюю (для объекта регулирования).

**По отношению к процессу обработки** информация делится на *необрабатываемую*, которая не поступает на обработку, и *обрабатываемую*, которая подвергается обработке. В результате обработки информации появляется *производная* информация, которая, в свою очередь, подразделяется на резульатную и промежуточную (рисунок 1.2).



Рисунок 1.2 – Информация и ее обработка

Промежуточная информация используется для выполнения последующих расчетов и выдачи справок. Она является выходной для одного расчета и входной для групп других задач или для этого же расчета, повторяемого через некоторое время, т. е. обеспечивает связь внутри информационной системы и взаимодействие подсистем.

**По функциям управления** экономическая информация делится на плановую, нормативную, бухгалтерскую, статистическую и оперативно-техническую.

**По числу значений** информационные совокупности делятся на *номенклатуры* (универсальное множество данной информационной совокупности, например, общая потребность в материалах), *массивы* (составные части – подмножества – полная номенклатуры, например, потребность в материалах определенной группы) и *позиции* (отдельные значения номенклатур, например, номенклатурный номер материала).

**По объективности отражения действительности** экономическая информация разделяется на *достоверную* и *недостоверную* (последняя должна быть сведена до минимума), по **целевому назначению** — на *полезную* и *бесполезную*; по **насыщенности** — на недостаточную, достаточную и избыточную. *Недостаточная* информация не содержит сообщений для принятия

квалифицированного решения или управления конкретным явлением, *избыточная* несет в себе дополнительные, ненужные сообщения о конкретном явлении, что усложняет и удорожает ее обработку. При рациональном построении информационных потоков и технологического процесса их обработки не должна возникать недостаточная и избыточная информация.

Для выбора технических средств и процедур обработки информации необходимо определить ее объем и структуру исходных данных. Информацию можно измерять по количеству форм документов и по числу элементов массивов (позиций).

### 1.3 Структурные единицы экономической информации

Семантический подход к экономической информации позволяет рассматривать ее структурный состав по содержанию (состав и форма) и состоянию объекта. Ее обработка связана с информационными совокупностями.

*Информационная совокупность* – это полный (достаточный) объем информации, который связан единой общей формой, о каком-либо процессе, явлении, предмете, операции. Информация должна быть достаточной для всесторонней характеристики изучаемого объекта на конкретном уровне за определенный отрезок времени. Например, данные о стипендии студентов имеют определенный состав и структуру.

Информационные совокупности разнообразны по объему, составу и сложности, имеют определенные форму и содержание. Форма – это структура информационной совокупности, а содержание – значения передаваемых данных.

Каждый объект управления не может функционировать без связанной с ним экономической информационной системы (ЭИС), которая является многоуровневой (рисунок 1.3).

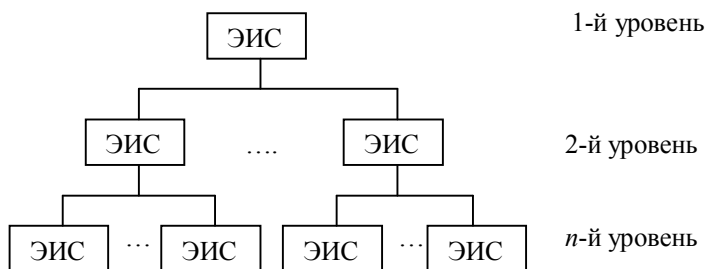


Рисунок 1.3 – Многоуровневая экономическая информационная система

Информационная система *n*-го уровня – это массив документов некоторой формы; она принимается за единицу информации и называется *составной единицей информации* (СЕИ) (рисунок 1.4).

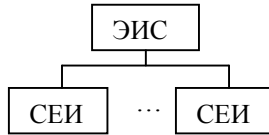


Рисунок 1.4 – Составная единица информации

Структура СЕИ может быть представлена в виде документа конкретной формы. Каждую СЕИ можно последовательно делить на информационные совокупности более низкого уровня и свести деление до реквизитов (полей).

*Реквизит* — это информационная совокупность, которая не поддается дальнейшему расчленению (на смысловом уровне) на единицы информации и характеризует отдельное свойство сущности. Реквизиты являются минимальными единицами информации, из которых образуются все СЕИ (рисунок 1.5).

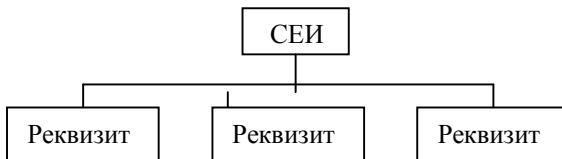


Рисунок 1.5 – Деление СЕИ на реквизиты

Термин «реквизит» общепринят в экономической науке и практике, имеет соответствующий смысл и толкование. В информационных технологиях (табличных процессорах, базах данных) принят адекватный термин «поле». Смысл этих терминов одинаков.

Каждый реквизит состоит из совокупности символов (символьных и числовых). Элементарным символом является бит (двоичный символ 0 или 1).

Отдельный реквизит обладает известной самостоятельностью и имеет особые, характерные для него черты, например, способность фигурировать в самых разнообразных СЕИ, имеющих различную сложность.

По своему возникновению и роли в процессе обработки реквизиты принято делить на основания и признаки.

*Реквизиты-основания* характеризуют количественные свойства сущностей, полученные в результате совершения отдельных операций (подсчета натуральных единиц, вычислений, измерений и т. д.). Например, в документе «Требование» (рисунок 1.6) к основаниям относятся количество, цена, сумма.

*Реквизиты-признаки* характеризуют качественные свойства сущностей (время и место события; обстоятельность, при которых совершилась операция и были получены числа-основания). При помощи признаков достигается индивидуализация сообщений, не допускаются смешивание и искажение информации (фактов). Так, в документе «Требование» признаками являются



номер документа, дата, отправитель, вид операции, наименование единиц измерения, указание, кто затребовал и отпустил товар.

При обработке информации основания подвергаются арифметическим операциям, а при помощи признаков осуществляют поиск информации, ее сортировку, выборку и т. д.

Существенной особенностью реквизитов-оснований является то, что каждый из них вместе с относящимися к нему признаками образует СЕИ особого типа – показатель.

*Показатель* — это информационная совокупность с минимальным составом, достаточным для образования документа, включающая в себя только один реквизит-основание и один или несколько признаков-признаков:

$$\boxed{\text{Показатель}} = \boxed{\text{Основание (одно)}} + \boxed{\text{N признаков}}$$

Элементы показателя, как и показатель в целом, можно рассматривать с двух точек зрения: по форме и по значению. Форма — это наименование граф и строк, а значение — записанные в этих графах и строках конкретные числа, цифры и т. д. Формы показателей в основном стабильны, а значения изменяются.

Деление информационных совокупностей документов на показатели облегчает классификацию информации и играет важную роль при алгоритмизации и размещении информации на носителях, так как СЕИ любой сложности можно свести в конечном счете к определенной группе различных показателей, каждый из которых будет иметь самостоятельный алгоритм вывода.

Изучение состава и структуры СЕИ позволяет правильно отбирать ее элементы, подлежащие компьютерной обработке, и находить наиболее рациональные способы этой обработки.

Рассмотрим структуру составной единицы информации на примере документа «Требование» (см. рисунок 1.6).

ТРЕБОВАНИЕ S					
C11	Номер документа P21	Дата P22	Отправитель C23		Вид операции P24
			Склад P31	Секция P32	
	Наименование P25	Единица измерения P26	Количество Q27	Цена Q28	Сумма Q29
	Затребовал: P210		Отпустил P211		

Рисунок 1.6 – Структура документа

Описание структуры документа рекомендуется выполнять в такой последовательности.

1 *Присвоение идентификаторов информационным единицам документа* (в документе выделены жирным шрифтом):

- S – информационная совокупность;
- C11 – заголовочная часть документа;
- C12 – содержательная часть документа;
- C13 – оформительская часть документа;
- P – реквизиты-признаки;
- Q – реквизиты-основания.

Числовые значения идентификатора состоят из двух цифр: первая – это номер уровня иерархии, вторая – порядковый номер внутри этого уровня.

2 *Отображение информационной совокупности:*

S.(C11,C12,C13)

Точка после буквы (в примере S) обозначает признак (знак) иерархического перечисления. Запишем структуру для каждой части документа:

C11.(P21, P22, C23. (P31, P32), P24)

C12.(P25, P26, Q27, Q28, Q29)

C13.(P210,P211)

В информационной структуре S содержатся составные единицы информации C11, C23, C12, C13, а также составные массивы (в нашем примере – четыре строки).

Запись структуры массива C12:

C12.(1-4).(P25, P26, Q27, Q28, Q29)

Если, например, за месяц выписывается 3500 подобных документов, то информационная совокупность будет записана следующим образом:

S. (1 – 3500). (C11. (P21, P22, C23. (P31, P32), P24);

C12. (1 - 4). (P25, P26, Q27, Q28, Q29);

C13.(P210,P211))

Показатель – составная структурная единица информации. Как уже упоминалось, он включает в себя только один реквизит-основание и один или несколько реквизитов-признаков.

3 *Общая запись показателя:* П.(Q, P1, P2 ..., Pn),

где П – показатель;

Q – реквизит-основание;

Pn – реквизиты-признаки.

В нашем примере (документе) содержится три показателя: П1 – количество; П2 – цена; П3 – сумма.

Запишем структуры показателей:

П1. (Q27, P21, P22, P31, P32, P24, P25, P26);

П2. (Q28, P21, P22, P31, P32, P24, P25, P26);

ПЗ. (Q29, P21, P22, P31, P32, P24, P25, P26).

Нашу информационную структуру можно отобразить в виде графа, в котором концевые вершины – это реквизиты, а остальные – составные структурные единицы (рисунок 1.7).

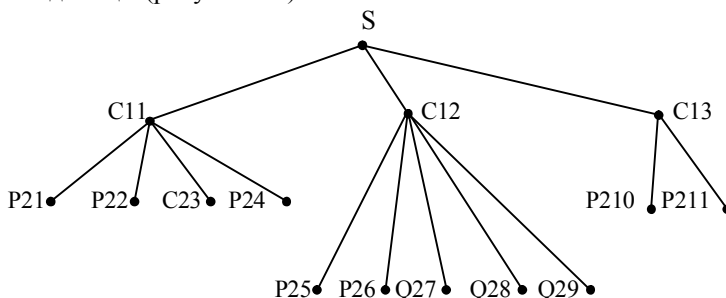


Рисунок 1.7 – Граф информационной структуры

Табличная форма представления структуры информации при организации массивов записей базы данных в памяти компьютера приведена на рисунке 1.8.

P21	P22	P31	P32	P24	P25	P26	Q27	Q28	Q29	P210	P211

Рисунок 1.8 – Табличная форма представления информации

#### 1.4 Внемашинальная организация экономической информации

Функционирование экономики отображается системой экономических показателей, характеризующих все аспекты жизнедеятельности общества. Для их фиксации используются носители информации.

*Носитель информации (media)* – это различные материалы, обладающие соответствующими физическими свойствами (бумага, магнитные ленты, магнитные диски, оптические диски и др.), которые применяются для сохранения информации, т. е. это любое материальное средство, на которое записывается (фиксируется) и с которого считывается информация. Существует более 50 носителей информации.

*Назначение носителей информации:*

- запись первичных данных;
- подготовка исходных данных к обработке;
- непосредственный ввод в компьютер подготовленных данных;
- организация внешней памяти компьютера (системы);
- фиксация результатов обработки данных.

Носители информации характеризуются своими габаритами (длина, ширина, толщина) и емкостью при установленных размерах.

Применение вычислительной техники в системах управления обусловило деление информации на *внутримашинную* и *внемашиинную*.

В условиях современной информационной технологии необходимо разрабатывать новые методы и средства сбора, регистрации и передачи данных. Важную роль призваны играть технические средства, которые обеспечивают механизацию первичного учета и непосредственный ввод информации в машины. В этих условиях происходит переоценка роли традиционных первичных документов.

## 2 ОРГАНИЗАЦИЯ ДАННЫХ В БАЗАХ

### 2.1 Причины, обусловившие появление баз данных

В обществе в целом и в жизнедеятельности любой организации, предприятия, специалиста роль информации постоянно возрастает. С развитием современных информационных технологий получила развитие концепция *интеграции данных* – их централизованного хранения и использования потребителями, имеющими различные запросы и предъявляющими различные требования к информации.

Сложность организации и структуры данных обусловили развитие одного из стратегических направлений развития информационных технологий – баз данных и систем управления базами данных (СУБД). С их помощью создаются хранилища информации для многих пользователей.

Объемы информации с каждым годом увеличиваются, усложняется ее структура. Для обработки и сохранения данных, обеспечения доступа к ним широкого круга пользователей было необходимо разработать общесистемные средства интегрированной обработки, хранения этих данных и управления ими. В связи с этим в конце 60-х гг. появились первые СУБД, которых сегодня на рынке информационных технологий насчитывается великое множество. Разнообразные по своим функциональным возможностям СУБД разрабатываются для массовых моделей и типов компьютеров и разных операционных систем.

Одной из фундаментальных идей современных информационных технологий является *концепция баз данных*: данные должны быть организованы в базы данных с целью адекватного отражения реального объекта и обеспечения информационных нужд пользователей.

**Прикладная система** (application system) предназначена для решения определенной задачи или класса задач или для предоставления пользователю соответствующих услуг. Данная система состоит из прикладных программ. Напомним, что системная программа (system program) входит в со-

став операционных систем, а последняя управляет другими программами и выполняет функции диспетчера.

В процессе эволюции проектирования технологий машинной обработки информации выделялись *индивидуальное* и *типовое* проектирование. Современные информационные технологии максимально ориентированы на применение пакетов прикладных программ (ППП), которые по назначению и функциональным возможностям можно разделить на четыре группы:

1) *пакеты общего назначения* (операционные системы, текстовые процессоры, деловая графика, табличные процессоры, системы управления базами данных, методо-ориентированные пакеты – динамического, линейного программирования и др.);

2) *пакеты функционального типа*, которые реализуют комплекс задач по функциям управления (бухгалтерский учет – учет основных фондов, материалов, заработной платы и т. д.);

3) *интегрированные системы* (integrated system) управления предприятиями и другими объектами – системы обработки информации, которые удовлетворяют широкий комплекс разных информационных и вычислительных потребностей пользователя, поддерживают единый способ взаимодействия пользователя с их компонентами и единый способ представления данных. Интегрированная система есть единый пакет, который включает в себя несколько самостоятельных, наиболее часто используемых программ: текстовый процессор, табличный процессор, базу данных, графику, систему связи. Все модули пакета объединены одной операционной средой в единое целое, что решает проблему совместимости. Интерфейс для пользователя дружелюбный.

Каждый модуль интегрированной системы удерживает разумно ограниченный перечень функций, используемых в практических применениях. Между модулями реализована внутренняя совместимость (по форматам данных) и внешняя совместимость с другими важнейшими программами, благодаря которой объекты, созданные в среде одного пакета, можно представлять (обрабатывать) в среде другого прикладного пакета. Пример интегрированного пакета, которых создано довольно много, – Microsoft Office;

4) *процедурно-ориентированные пакеты*, реализующие стандартные процедуры обработки информации, – например, генератор ввода-вывода информации; ввод и контроль информации; средства внутримашинной совместной обработки файлов, редактирование и вывод информации.

Как видим, на рынке информационных технологий присутствует большое количество прикладных систем обработки информации, в связи с чем становится актуальной проблема выбора нужной системы. В условиях острого дефицита средств вычислительной техники актуальным был выбор соответствующего комплекса технических средств. В наше время очень важно выбрать нужную прикладную систему обработки информации, по-

сколькx каждая из них ориентирована на работу в среде конкретной операционной системы.

При выборе прикладной системы обработки информации *необходимо учитывать*: качество документации, простоту освоения и внедрения, сервис для пользователя, фирменное сопровождение, стоимость системы.

*Качество документации* считается приемлемым при наличии подробного описания системы и инструкции по применению (учебник с большим количеством примеров, а также краткий словарь по использованию системы).

*Простота освоения.* Система должна содержать большое количество примеров и программ. В учебнике должны быть примеры с фотографиями экрана монитора. Пользователь должен быть обеспечен шаблоном, глоссарием (т. е. словарем, содержащим непонятные и малоиспользуемые слова и фразы) и обычным кратким словарем. Нужно учитывать и возможность установки пакета на компьютере без консультации разработчика. Сервис для пользователя должен быть благоприятным и современным. Под фирменным сопровождением понимается возможность получить консультацию в любое время.

## 2.2 Понятие базы данных

*База данных (database)* — поименованная и организованная (структурированная) совокупность взаимосвязанных данных, которые отражают состояние объектов конкретной предметной области и находятся под центральным программным управлением. База данных удерживает такой минимальный избыток информации, который позволяет удовлетворить запросы каждого из многочисленных пользователей. Данные представлены совокупностью файлов (таблиц). В широком смысле база данных есть любая упорядоченная совокупность данных.

*Табличная структура* состоит из строк, которые соответствуют записям базы данных и столбцов — полей записей. Каждая запись имеет уникальный номер, а каждое поле должно иметь уникальное имя. Одна запись файла соответствует конкретному экземпляру понятия предметной области, например информация о конкретном студенте.

Предметная область (application domain) — часть реальной среды, которая описывается и отражается в базе данных. Это информационная сторона функционирования автоматизированной системы, отражающая множество объектов и связей между ними.

### **Свойства базы данных<sup>1</sup>:**

1 Многоразовое использование: одни и те же данные могут использоваться многими пользователями.

---

<sup>1</sup> Сформулированы Дж. Мартином.

2 Простота обновления – возможность внесения изменений в базу с минимальными затратами.

3 Быстрый поиск и получение необходимой информации по запросу.

4 Уменьшение избыточности – новые задачи должны получать данные из существующей базы, а не путем их повторного ввода.

5 Защита от несанкционированного доступа к данным.

6 Максимальная независимость от прикладных программ: изменения в структуре базы данных не должны, по возможности, приводить к перезаписи пакета программ.

7 Защита от уничтожения и искажения информации (некомпетентного пользователя, злоумышленных действий, сбоев и конфликтных ситуаций).

#### **Требования к базе данных:**

1 Адекватность отражения предметной области:

а) полнота данных;

б) динамичность информационной модели;

в) актуальность информации в данный момент времени.

2 Возможность взаимодействия с пользователями различных категорий и в разных режимах.

3 Обеспечение секретности данных, надежности, целостности, защита от случайного или целенаправленного разрушения базы данных.

4 Обеспечение взаимной независимости программ и данных.

5 Технологичность обработки данных.

6 Совместимость компонентов базы данных.

7 Простота изменения логической и физической структуры базы данных в целях повышения эффективности обработки информации.

8 Способность к расширению и модификации.

База данных функционирует в социальном пространстве со множеством пользователей, которые могут предъявлять ей противоречивые требования. Необходимо постоянно искать компромисс между противоположными требованиями к системе в социальном пространстве пользователей. База данных, как уже упоминалось, находится под централизованным управлением. Система управления базой данных построена на концепции многоуровневой организации. Для слаженной и эффективной работы выполняется администрирование базы данных. Функции администратора баз данных объединены в следующие группы: администрирование предметной области, баз данных, безопасности данных и приложений. Функции администрирования данных поддерживаются специальными служебными программами-утилитами. Для администрации баз данных предусмотрены соответствующие инструкции.

*Пользователь* базы данных (user) — это физическое или юридическое лицо, которое пользуется услугами компьютерной системы для получения информации или решения соответствующих задач. Существуют различные

категории пользователей: аналитик, системный программист, прикладной программист, администратор системы, оператор компьютера, конечный пользователь. Пользователями базы данных могут быть прикладные программы и программные комплексы. Доступ пользователя к базе данных обеспечивается СУБД.

### 2.3 Историческая справка о базах данных

Термин «база данных» был введен в 1963 г. в англоязычной литературе (data base), в 70-х гг. этот термин стали писать через дефис (data-base), а потом – одним словом (database). Имя автора этого термина неизвестно.

Историю развития баз данных можно разделить на три периода.

*Первый период* – 60-е гг. – *переходный*: появление самого понятия и нескольких первоначальных систем. В 1959 г. Мак-Гри предложил использовать файлы исходных данных. Файл, который введен в компьютер, становился общим, и его могли совместно использовать многие пользователи. Компьютеры обеспечивали доступ к данным. Мак-Гри разработал систему баз данных IMS фирмы IBM.

В 1963 г. Бахман разработал первую промышленную систему баз данных IDS: сетевая организация данных на магнитных дисках и многоцелевое использование наборов данных. В середине 60-х гг. началось широкое применение магнитных дисков, а затем появились новые возможности для обработки информации.

В 1967 г. ассоциацией CODASYL была создана рабочая группа по разработке баз данных, которая в 1969 и 1971 гг. выпустила отчеты о своей работе.

В 1968 г. Олле высказал следующие теоретические положения:

1 В условиях применения СУБД лицо, не имеющее профессиональной подготовки, может получить нужную информацию из базы данных, не зная способа размещения данных и доступа к ним.

2 При поиске информации процедурные и непроцедурные системы взаимосвязаны и между ними нет резкой границы.

3 Следует ориентироваться на обработку данных непроцедурным способом.

4 Структура памяти и структура данных – разные понятия. Пользователь должен знать логическую структуру данных. Важно выделить логическую структуру, при которой система эффективна в высокой степени.

*Второй период* – от 70-х гг. до 1981 г. – *период развития*: аппаратное развитие компьютеров и концепции баз данных, а также успешное внедрение систем. Произошло становление баз данных как науки. Были разработаны теоретические обоснования систем баз данных, которые стали основой построения информационных систем.



В 1970 г. Кодд (фирма IBM) предложил реляционную модель данных – реляционную парадигму<sup>1</sup>. Одно из важнейших положений реляционной модели – независимость данных: при управлении базой данных выделяются физический и логический уровни. Отпадает необходимость указывать в программах детали физической организации данных в компьютере; пользователь освобождается от необходимости вникать в тонкости физической организации данных. В 1982 г. появились коммерческие реляционные системы баз данных: SQL/DS, INGRES, ORACLE и др. За успехи в разработке баз данных американская ассоциация ЭВМ в 1973 г. присудила премию Бахману, а в 1981 г. – Кодду.

В 1980 г. был разработан простой язык запросов, ориентированный на пользователя. Быстро расширилось применение баз данных.

*Третий период* – начало в 1981 г. – *период зрелости*: появление коммерческих систем баз данных на основе реляционных моделей. Успешно проводятся работы в сфере аппаратного обеспечения баз данных (расширяются области их применения, развивается технология интегральных схем), возникают новые направления.

## 2.4 Модели данных

Информация, а также организация ее получения — основа проектирования информационных систем. Для более эффективной организации данных применяются информационные модели. Модель данных (понятие которой впервые ввел в 1970 г. Кодд) отражает взаимосвязь между объектами.

База данных, как уже упоминалось, состоит из файлов (таблиц), которые связаны между собой, файлы — из записей и сохраняются на внешней памяти, запись — из полей (реквизитов). Реквизиты объединяются в запись в соответствии с определенными связями. Принцип связей между реквизитами (полями) определяет структуру базы данных. Применяются древовидная, сетевая и реляционная структуры.

**Модель данных** (data model) – это система взаимосвязанных типов объектов, операторов и правил обеспечения целостности, создающая абстрактную структуру, которую поддерживает система управления базой данных; это совокупность правил прохождения структур данных в базе данных, операций над ними, а также ограничений целостности, которая определяет допустимые связи и значения данных, последовательность их изменения. Отражая представление данных и отношений между ними математическими и программными средствами, модель данных есть формализованное описание информационных структур и операций над ними.

---

<sup>1</sup> *Парадигма* — научная теория, которая содержится в системе понятий, отражающих истинные черты действительности (исходная концептуальная схема).

Модель данных должна быть достаточно гибкой, чтобы правильно представлять многие связи реальных объектов. В компьютере концептуальная (общая) структура данных трансформируется в физическую (внутреннюю) структуру.

База данных в соответствии с моделью данных содержит информацию о реальной действительности (рисунок 2.1). Часть реальной действительности, которая отражается в базе данных, называется *предметной областью*.



Рисунок 2.1—Информационная модель

Модель данных — это формально определенная структура представления данных (имитация структуры). Модель должна быть адекватной объекту управления. При проектировании информационных систем она отражается моделями данных, как минимум, в двух уровнях: *логическая* и *физическая структура*. Логическая структура есть представление логической организации данных в виде множества типов записей и связей между ними.

**Последовательность создания информационной модели.** Прежде всего следует определить концептуальные требования пользователей, интегрируемые в концептуальную модель. Сущность этой модели — структурирование данных и выявление взаимосвязи между ними без рассмотрения особенностей их реализации. Для построения концептуальной модели необходимо проанализировать существующие (решаемые) задачи на объекте. Затем построенная концептуальная модель преобразуется в модель данных в условиях выбранной СУБД. Если принятые взаимосвязи между объектами концептуальной модели не реализуются средствами выбранной СУБД, необходимо менять концептуальную модель.

*Концептуальная структура* (схема) — это абстрагированная общая структура данных без учета физических особенностей используемого компьютера. Версия концептуальной модели, которая может быть обеспечена конкретной системой управления базой данных, называется *логической моделью*. Она отражает логические связи между элементами данных независимо от их содержания и места хранения.

В одной СУБД могут применяться различные структуры и модели данных. Логическая модель отражается в физической памяти (диск, лента и др.). Компьютер превращает концептуальную (общую) структуру в форму, удобную для запоминания или поиска. Такая внутренняя структура данных называется *физической структурой*. Она определяет размещение данных,

методы доступа и технику индексирования. Физическую модель часто называют *внутренней моделью системы*.

Физическая структура данных проектируется с некоторым избытком для обеспечения надежности управления данными. Физическая структура учитывает архитектуру компьютера. Логическое и физическое представление данных приведено на рисунке 2.2.



Рисунок 2.2 – Представление данных под управлением СУБД

**Типы моделей данных.** В зависимости от способа представления взаимосвязей между объектами логическая модель данных может быть *иерархической, сетевой, реляционной*.

В начале 60-х гг., когда не было персональных компьютеров, применялись иерархическая и сетевая модели данных. В начале 70-х гг. появилась реляционная модель. Эти три модели различаются способами представления взаимосвязей между объектами.

**Иерархическая модель данных.** Взаимосвязи между объектами отражаются по принципу иерархии типов объекта в виде связанного графа, вершины которого размещены на разных иерархических уровнях.

Самая высокая вершина называется *корнем* (главный тип объекта), а остальные, находящиеся на нижних уровнях иерархии, — *подчиненными*. Корень (первый уровень) не подчиняется ни одной вершине. Все остальные вершины (типы объектов) связаны с одной и только одной вершиной, которая размещена на более высоком уровне. Взаимосвязь между объектами напоминает генеалогическое дерево (рисунок 2.3).

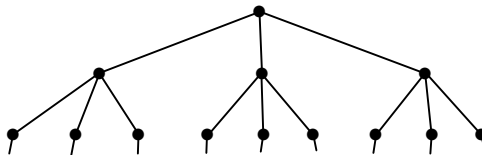


Рисунок 2.3 – Схема иерархической модели данных

Взаимосвязь между главными и подчиненными объектами устанавливается типом «один–ко многим» (1:M). Как видно из схемы иерархической модели данных, ее древовидная структура состоит из узлов и дуг. Узел – совокупность атрибутов, которые описывают объект. Каждый объект характеризуется надежной совокупностью основных атрибутов. Например, объект «студент» может иметь такие атрибуты: фамилия, имя, отчество, номер группы и др.

Вершины графа, которые подчинены другой вершине («имеют отца»), называются *сыновьями*. Любой «сын» на графе может иметь не больше одного «отца», а любой «отец» — множество «детей». Любая вершина может иметь множество подчиненных ей вершин на более низком уровне. Каждая пара вершин соединена одной простой дугой.

Иерархическая модель данных — эффективное средство описания объектов с подобной структурой. В ней существует сильная зависимость между описанием структуры данных и способом их записи на внешние носители (диски).

**Сетевая модель данных.** В сетевой модели понятие главного и подчиненных объектов иное, чем в иерархической модели: любой объект здесь может быть и главным, и подчиненным; каждый объект может участвовать в любом количестве взаимосвязей (рисунок 2.4).

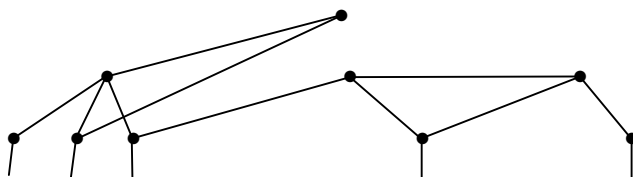


Рисунок 2.4 – Общая схема сетевой модели данных

Данные представляются при помощи записей и связей. Запись (объект) в сетевой модели (в отличие от иерархической) может иметь множество как подчиненных ей записей, так и записей, которым она сама подчинена.

**Реляционная модель данных.** Реляционная структура (модель) представляет данные в виде двумерной таблицы. Табличная структура данных отражает отношения между реальными объектами и их характеристиками. Поиск и обработка записей не зависят от организации хранения данных в памяти компьютера. При этом эффективно используются математическая логика и алгебра. Основным принцип реляционных структур баз данных — получение из таблицы необходимых отношений и формирование новых. На основе первичной таблицы при помощи логических операций формируется новая таблица соответствующей структуры. Каждый файл соответствует какому-нибудь понятию из предметной области.

В реляционных моделях имеются следующие типы объектов: *таблицы* (отношения), *атрибуты* (столбцы) и *домены* (допустимые значения атрибутов). Операторы определяются реляционной алгеброй.

В этой модели объекты и взаимосвязи между ними представлены при помощи таблиц (рисунок 2.5). Одна таблица представляет один объект и состоит из столбцов и строк. Каждая строка таблицы представляет собой одну запись, а каждый столбец — одно поле записей. Таблица обладает следующими свойствами:

- каждый элемент таблицы (на пересечении строки и столбца) есть один элемент данных;
- столбцам присваиваются уникальные имена;
- элементы столбца имеют одинаковую природу (однородны);
- в таблице нет двух одинаковых строк;
- при выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке, независимо от их информационного содержания.

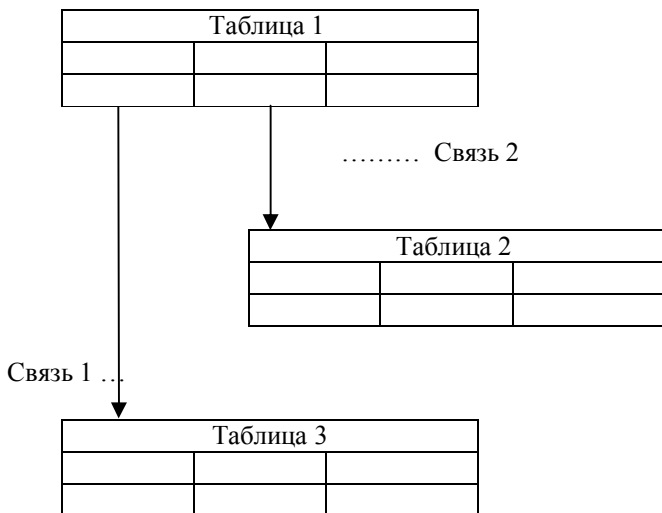


Рисунок 2.5 – Общая схема реляционной модели данных

Информация в реляционных базах данных сохраняется в таблицах, связь между которыми осуществляется путем совпадения значений одного или нескольких полей. Каждая строка таблицы в реляционных базах данных уникальна. Уникальность строк обеспечивают *ключи* (одно или несколько полей таблицы), которые сохраняются в упорядоченном виде. Это обеспечивает прямой доступ к записям в процессе поиска. Каждая таблица должна иметь первичный ключ (ключевое поле одно или более). Поля-ключи уникально идентифицируют каждую строку в таблице.

Предметную область в реляционных моделях данных представляют в виде соответствующего числа таблиц. Классическое понятие «файл» в обработке данных можно ассоциировать с одной таблицей. Таким образом, базу данных можно рассматривать как набор таблиц. Файлы (таблицы) подвергаются классической обработке, так называемому обновлению основного файла.

Таблицы находятся между собой в соответствующих отношениях (связях). Представление предметной области в виде системы отношений — самая сложная и ответственная задача, которая относится к области логического проектирования.

Преимущества реляционных моделей баз данных:

- простота представления данных (таблица);
- минимальный избыток данных, что достигается путем нормализации таблиц;
- независимость приложений пользователя от данных (при включении или удалении таблиц);
- возможность изменения состава атрибутивных отношений;
- отсутствие необходимости описывать схемы данных (в иерархических и сетевых моделях — надо).

Недостатки реляционных моделей: нормализация таблиц приводит к значительной фрагментации данных, а при решении задач, как правило, их необходимо объединять.

**Двенадцать правил Кодда.** Этим правилам, сформулированным Коддом, должна соответствовать реляционная СУБД [7, с. 64-65]:

1 *Правило информации.* Вся информация в базе данных должна быть представлена исключительно на логическом уровне и только одним способом – в виде значений, содержащихся в таблицах.

2 *Правило гарантированного доступа.* Логический доступ ко всем и каждому элементу данных (атомарному значению) в реляционной базе данных должен обеспечиваться путем использования комбинации имени таблицы, первичного ключа и имени столбца.

3 *Правило поддержки недействительных значений.* В настоящей реляционной базе данных должны быть поддержаны недействительные значения, которые отличаются от строки символов нулевой длины, строки пробельных символов и от нуля или другого любого числа и используются для представления отсутствующих данных независимо от типа этих данных.

4 *Правило динамического каталога, основанного на реляционной модели.* Описание данных на логическом уровне должно быть представлено в том же виде, что и основные данные, чтобы пользователи, обладающие соответствующими правами, могли работать с ним при помощи того же реляционного языка, который они применяют для работы с основными данными.

5 *Правило исчерпывающего подъязыка данных.* Реляционная система может поддерживать различные языки и режимы взаимодействия с пользователем (например, режим вопросов и ответов). Однако должен существовать, по крайней мере, один язык, операторы которого можно представить в виде строк символов в соответствии с некоторым четко определенным синтаксисом и который в полной мере поддерживает следующие элементы:

- определение данных;
- определение представлений;
- обработка данных (интерактивная и программная);
- условие целостности;
- идентификация прав доступа;
- границы транзакций (начало, завершение и отмена).

6 *Правило обновления представлений.* Все представления, которые теоретически можно обновить, должны быть доступными для обновления.

7 *Правило добавления, обновления и удаления.* Возможность работать с отношением как с одним операндом должна существовать не только при чтении данных, но и при их добавлении, обновлении и удалении.

8 *Правило независимости физических данных.* Прикладные программы и утилиты для работы с данными на логическом уровне должны оставаться нетронутыми при любых изменениях способов хранения данных или методов доступа к ним.

9 *Правило независимости логических данных.* Прикладные программы и утилиты для работы с данными на логическом уровне должны оставаться нетронутыми при внесении в базовые таблицы любых изменений, которые теоретически позволяют сохранить нетронутыми содержащиеся в этих таблицах данные.

10 *Правило независимости условий целостности.* Должна существовать возможность определять условия целостности, специфические для конкретной реляционной базы данных, на подъязыке реляционной базы данных и хранить их в каталоге, а не в прикладной программе.

11 *Правило независимости распространения.* Реляционная СУБД не должна зависеть от потребностей конкретного клиента.

12 *Правило единственности.* Если в реляционной системе есть низкоуровневый язык (обрабатывающий одну запись за один раз), то должна отсутствовать возможность использования его для того, чтобы обойти правила и условия целостности, выраженные на реляционном языке высокого уровня (обрабатывающем несколько записей за один раз).

## 2.5 Средства представления связей данных

Проектирование базы данных требует профессиональной подготовки и может быть разделено на два этапа:

1) *логическое* (проектирование концептуальной схемы);

2) *физическое* (проектирование внутренней схемы).

В реляционных базах данных все данные сохраняются в виде прямоугольных таблиц и все операции над базой данных сводятся до манипуляции таблицами.

Основные понятия в реляционных базах данных: таблица, отношение, строка, столбец, первичный и внешний ключи.

*Таблица* имеет уникальное имя. База данных включает в себя множество таблиц, связь между которыми устанавливается при помощи совпадающих полей. Каждая таблица удерживает информацию о каких-либо объектах одинакового типа. Каждая запись в таблице идентифицирует один объект группы.

Сведения об объекте предметной области в базах данных связаны между собой соответствующими отношениями. В общем виде их можно представить так:

$$A \text{ F}(x)/\text{S}(x)B,$$

где  $A$  и  $B$  – объекты;

$F(x)$  – виды связи объекта  $A$  и объекта  $B$ ;

$S(x)$  – связь объекта  $B$  и объекта  $A$ .

Функции  $F(x)$  и  $S(x)$  могут принимать значения единой связи (1) и множества связей ( $M$ ).

Между информационными объектами существуют четыре типа *отношений*, для которых введены следующие обозначения:

«один-к-одному» (1:1);

«один-ко-многим» (1: M);

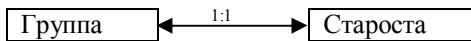
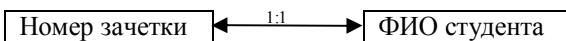
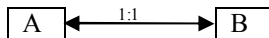
«многие-к-одному» (M : 1);

«многие-ко-многим» (M: M).

Рассмотрим эти типы отношений между парами связанных элементов данных.

*Отношение (отражение) «один-к-одному» (1:1)* означает, что в каждый момент времени одной записи в одной таблице  $A$  соответствует не более чем одна запись в таблице  $B$ , к которой направлена связь, и наоборот. Связь поддерживается при помощи совпадающих полей, которые могут иметь разные имена. Связь между таблицами устанавливается не по совпадению имен полей, а на основе значений совпадающих полей. Связь уникальная как слева направо, так и справа налево.

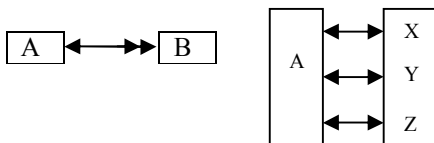
*Пример.*





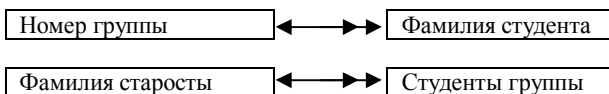
Отношение «один-ко-многим» (1: M). Совпадающие поля в обеих таблицах имеют одинаковые имена.

Пример.



Одна запись с выходящей А-связью идентифицирует несколько записей, которые принимают эту связь из таблицы В, а запись в таблице В не может иметь более одной соответствующей ей записи в таблице А. Связь является уникальной лишь справа налево, следовательно, идентифицируется только одна запись (экземпляр объекта).

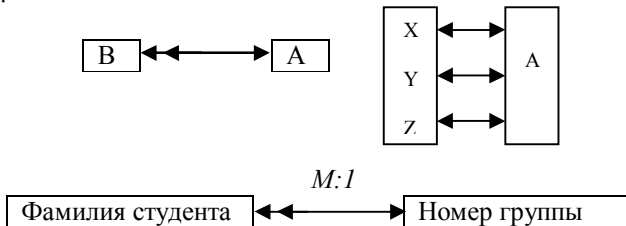
Пример.



В академической группе один староста руководит всеми студентами, но каждый студент имеет только одного старосту.

Отношение «много-к-одному» (M:1) схоже с отношением «один-ко-многим», но является уникальной лишь связь слева направо.

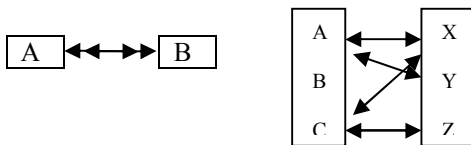
Пример.



Отношение «много-ко-многим» (M:M) возникает между двумя таблицами, когда:

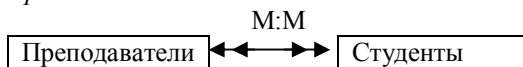
- одна запись с первой таблицы А (выходная связь) может быть связана больше чем с одной записью другой таблицы В (принимающая);
- одна запись с другой таблицы может быть связана больше чем с одной записью первой таблицы. Идентификация не является уникальной в обоих направлениях.

Пример.



Эта схема реализуется только при помощи третьей соединительной таблицы, ключ связи которой состоит, как минимум, из двух полей. Эти поля являются полями внешнего ключа в таблицах А и В. Первичный ключ для соединительной таблицы – это обычно комбинация из внешних ключей.

*Пример.*



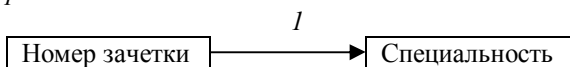
Один преподаватель обучает множество студентов, один студент учится у многих преподавателей. Как видно, отношения отражают связи между парами элементов, направленные в обе стороны. Между объектами «преподаватели» и «студенты» существуют отношения М: М, потому что каждый преподаватель может работать более чем с одним студентом и каждый студент может посещать занятия более чем одного преподавателя.

Если между таблицами имеются связи типа М: М, нужно создать дополнительную таблицу пересечений, с помощью которой связь М: М будет сведена к двум связям типа 1: М. Заметим, что Access не позволяет определить прямую связь М : М между двумя таблицами.

*Ассоциация* (association) в базах данных – это зависимость между двумя элементами данных, односторонняя связь между парами элементов данных. Существуют три типа ассоциаций: простая (тип 1); сложная (тип М); условная (тип С).

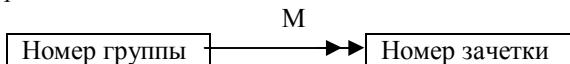
Ассоциация называется *простой* (однозначной), если экземпляр элемента данных с выходной связью идентифицирует один и только один экземпляр элементов данных, который принимает эту связь. Обозначается числом 1 и одинарной стрелкой 1.

*Пример.*



Ассоциация называется *сложной* (многозначной), если экземпляр элементов данных с выходной связью идентифицирует несколько (один или более) элементов данных, к которым эта связь направлена. Сложную ассоциацию обозначают символом М и двойной стрелкой.

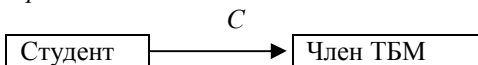
*Пример.*



Ассоциация называется *условной*, если для элемента данных с выходной связью может не существовать соответствующего элемента данных, который принимает эту связь, но когда связь существует, то относится к одному экземпляру элементов данных. Идентификация, когда она существует, явля-

ется уникальной. Каждый экземпляр объекта А может иметь один или не иметь ни одного экземпляра объекта В. Условная ассоциация обозначается буквой С и одинарной стрелкой:

*Пример.*

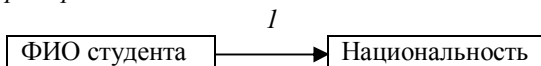


В управлении базами данных *ключ* — это идентификатор записей или группы записей в файле данных. Ключ является содержимым одиночного поля. Ключи сохраняются в таблице ключей (индексные файлы) и специально идентифицируются для ускорения поиска необходимой записи. Ключ — это идентификатор, который уникально определяет запись об объекте.

Ключи делятся на два вида: первичный и вторичный. Первичные ключи бывают *простыми* (одно поле) и *составными* (несколько полей). Первичный ключ однозначно идентифицирует запись; по нему находится только один экземпляр-запись. Вторичный ключ — значение ключевого поля — может повторяться в нескольких записях файла (не уникальный), по нему находим несколько экземпляров записей с одинаковым значением вторичного ключа.

*Простой ключ* состоит из одного элемента. Значение простого ключа уникально в таблице.

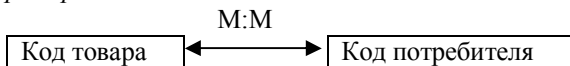
*Пример.*



Простой ключ состоит из одного поля (в нашем примере — «ФИО студента»), а каждое значение этого поля уникально в отношении и однозначно определяет понятие «Национальность».

*Составной ключ* состоит из нескольких полей. Связь между атрибутами М:М. Атрибуты сложного ключа не зависят друг от друга, ни один из них не является дополнительным квалификационным признаком другого атрибута.

*Пример.*



Целиком составной ключ включает два поля: «Код товара» и «Код потребителя», значения которых однозначно идентифицируют объект.

*Полусоставной ключ* содержит несколько атрибутов с функциональной зависимостью М:1. Каждый последующий атрибут является дополнительным квалификационным признаком предыдущих атрибутов.

*Пример.* Полусоставной ключ включает атрибуты «Преподаватель» и «Студент».

## 2.6 Файловая модель

Информация может иметь вид совокупности не связанных между собой файлов. Каждый файл состоит из записей одинаковой линейной структуры. Основными элементами структуры файла являются поле, запись, файл. В файле (файловой модели) структурной основной единицей является запись, которая состоит из полей.

*Поле* – это неделимая единица информации, элементарная единица логической организации данных. Термин «поле» в системе управления базой данных соответствует понятию «реквизит» в теории информации. Каждое поле содержит определенные данные.

*Запись* – это конечная логическая связанная совокупность полей. Состав (перечисление) и принятая последовательность полей, которые входят в запись, определяют структуру записи. Каждая запись в файле однозначно идентифицируется уникальным ключом записи.

**Индексирование.** Именно ключи являются эффективным средством доступа к соответствующим записям в файле. Сущность индексирования — создание специального индексного файла, который содержит в заданном порядке все значения ключа файла данных. Индексный файл содержит указания на соответствующие записи файла данных. Указатели создаются для каждого значения ключа.

*Описание логической организации данных файловой модели.* После выделения логической структуры записи необходимо представить ее в наглядной документальной форме. Каждому файлу присваивается уникальное имя, каждому полю – имя поля. Для каждого поля задается его формат, который включает тип данных, длину поля, точность.

Логическую структуру данных файловой модели можно представить разными способами, например в виде таблицы.

**Компьютерная реляционная база данных.** Теория и практика баз данных и СУБД развиваются в компьютерной среде и являются составными частями современных информационных технологий. Одна из идей реляционной базы данных сводится к смене ее физической организации без изменения логической организации с целью повышения эффективности использования базы данных.

В реляционной базе данных информация записана в таблицах (строки и колонки данных), а для поиска данных в таблице используют данные из колонок другой таблицы. Строки таблицы представляют собой записи (набор информации об отдельном элементе), а колонки – поля (отдельные атрибуты записи).

Во время поиска реляционная база данных связывает информацию поля одной таблицы с информацией в соответствующем поле другой таблицы для обработки третьей таблицы, в которой комбинируются запрошенные

данные с обеих таблиц. Предположим, таблица А содержит следующие поля: Специальность, ФИО, Год рождения, а таблица В – Специальность, Курс, Стипендия. Компьютерная реляционная база данных может связать таблицы А и В по значению поля «Специальность», находящегося в обеих таблицах, чтобы отыскать необходимую информацию, например, такой структуры: *ФИО, Стипендия*.

## 2.7 Централизованные и распределенные базы данных

СУБД развивались в условиях единства и взаимовлияния компьютеров, программного обеспечения, средств связи, а также науки управления. На этом пути были разработаны две научно-практические концепции технологии обработки информации: *централизованная и распределенная базы данных*.

*Централизованная база данных* (centralized database) разрабатывается и функционирует на принципах централизации в одном месте. База данных находится на одном компьютере, в виде одного информационного массива. В этом случае говорят о централизованном или монопольном владении данными. Такая база данных доступна только одному пользователю. Применяется в локальных сетях персональных компьютеров.

При функционировании сети персональных компьютеров к информации централизованной базы данных обеспечивается одновременный доступ нескольких пользователей со своих рабочих мест. База данных при этом размещается на машине-сервере. Работая в сети персональных компьютеров и пользуясь централизованной базой данных, каждый пользователь при необходимости может сделать локальную базу данных для воплощения своих профессиональных функций.

*Распределенная база данных* (distributed database) представляет собой совокупность баз данных, которые физически распределены (разнесены) по взаимосвязанным ресурсам вычислительной сети и доступны для совместного применения в разных местах. Распределенная база данных разъединена только физически, а не логически. Вся база данных потенциально доступна с любого конечного абонента (пользователя). На логическом уровне (модели) распределенная база данных описывается как целая информационная совокупность (логическая интеграция).

Распределенная база данных развернута в виде баз данных (подсистем), компоненты которых размещены по разным узлам сети. При наличии одинаковых компонентов (подсистем) база данных считается *однородной*, в противном случае – *неоднородной*.

Распределенная база данных может создаваться по одной идеологии, проектироваться и функционировать как единое целое (однородная). Предметной областью в этом случае являются крупная организация, предприятие, ведомство. Может возникнуть ситуация, когда несколько данных, ко-

торые раньше разрабатывались автономно, продолжают работать в своем режиме, но при этом между ними организуются связи, которые не требуют жестких ограничений на их функционирование. Таким образом, создается неоднородная распределенная база данных.

Распределенные базы данных требуют высокого уровня развития национальных и международных систем связи между компьютерами. Распределенная база данных – это целая логическая информационная система, имеющая администратора интегрированной базы данных и администраторов локальных баз данных.

По способу доступа к информации базы данных делятся на базы данных с *локальным доступом* и *отдаленным (сетевым) доступом*.

Системы централизованных баз данных с отдаленным доступом организуются по двум концепциям (архитектурам): «файл–сервер» и «клиент–сервер».

*«Файл–сервер»*. В компьютерной сети один компьютер выделяют в качестве центрального (сервер файлов) для хранения совместно используемой централизованной базы данных. Все остальные компьютерные системы выполняют функции рабочих станций, с которых пользователи поддерживают доступ с центральной базой данных. По запросам пользователей соответствующие файлы базы данных передаются на рабочие станции для основной обработки и использования. На рабочих станциях могут создаваться и локальные базы данных. Концепция «файл–сервер»: файлы базы данных с файл-сервера передаются для обработки на рабочих станциях.

*«Клиент–сервер»*. Концепция: сохранение централизованной базы данных на центральном компьютере (сервер базы данных) и основная обработка данных. По запросу клиента (рабочей станции) на сервере ищутся и выбираются необходимые данные, которые транспортируются по сети от сервера к рабочим станциям. Основная обработка данных выполняется центральной машиной.

### **3 ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ**

#### **3.1 Понятие процесса проектирования**

*Проектирование базы данных* – это взаимосвязанный комплекс работ: от определения общих параметров создаваемой базы данных, анализа объекта управления и существующей системы управления на объекте, построения информационной модели до создания реальной базы данных.

Проектирование базы данных начинается с исследования объекта управления с целью получения необходимой информации и построения модели

системы управления. Проводится диагностический анализ объекта управления, а затем собственно исследование для построения модели.

В информационную модель должны включаться все задачи объекта управления. Информационную модель можно отображать в виде информационной схемы. На первом этапе работы надлежит разработать принципиальную модель системы управления — основу для построения полной рабочей модели.

Исследование и анализ объекта управления с целью построения информационной модели объекта управления проводят квалифицированные специалисты, которые имеют соответствующие знания и опыт работы. На стадии обследования и изучения объекта управления изучаются и анализируются потоки информации с целью формализации информационных процессов.

Изучаются формы документов, техника их заполнения и обработки:

- назначение документа;
- количество экземпляров при выписке;
- наименование обязательных реквизитов и показателей документа;
- правила формирования показателей;
- периодичность составления документа;
- кто заполняет реквизиты документа.

Уточняются маршруты движения документов. Информационные потоки описываются графическим методом (графические схемы). Основным элементом информационного потока являются документы.

До перехода к непосредственной компьютерной разработке базы данных очень важно сосредоточить внимание на постановке задачи, определении перечня задач и принципов работы системы.

### 3.2 Этапы проектирования реляционной базы данных

Проектирование базы данных независимо от типа информационной модели можно разделить на три этапа: а) *концептуальное*; б) *логическое*; в) *физическое* проектирование.

*Концептуальное проектирование.* Описываются взаимосвязи между элементами данных, которые отражают взаимосвязи в предметной области. Проводятся сбор информации, её упорядочение, составляются локальные и композиционные модели данных. Описание данных на соответствующем языке называется схемой.

*Логическое проектирование.* Описывается, как выбранные взаимосвязи будут представлены в структурах записей базы данных. Выбираются модели СУБД и логические структуры (таблицы, файлы, списки и др.) для описания данных.

*Физическое проектирование.* Описывается, как структуры записей будут представлены в памяти компьютера, на дисках. Реализация баз данных на

дисках мало интересует пользователя — это в основном прерогатива администратора системы и базы данных. Пользователь видит логическую организацию базы данных (рисунок 3.1).

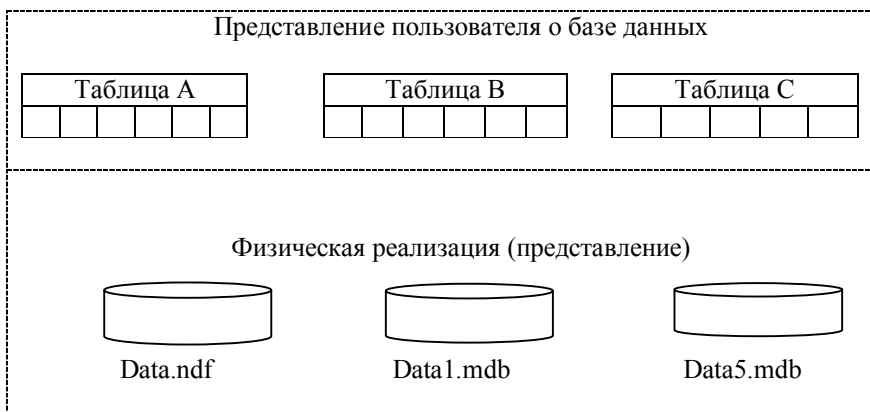


Рисунок 3.1– Логическая и физическая структуры базы данных

Создание базы данных начинается с определения ее структуры, которая непосредственно влияет на эффективность функционирования базы данных. Рекомендуются следующие этапы проектирования баз данных.

*Первый этап* — уточнение задачи. Принимается решение о необходимости и целесообразности автоматизации обработки информации на объекте управления. Определяются цели создания баз данных, их функции, информация, которая должна содержаться в них.

На объекте управления, для которого разрабатывается база данных, уточняется и составляется полный список основных задач с позиции системного подхода, а также с учетом перспектив развития объекта управления и появления новых задач. Для каждой задачи заполняется рабочий бланк, в котором ей присваивается номер, даются название, краткое описание и указывается связь с другими задачами.

*Второй этап* — последовательность выполнения задачи. Основные задачи (например, учет материалов, учет готовой продукции) объединяются в тематические группы; внутри каждой из них задачи располагаются в порядке их выполнения. Последовательность выполнения задач можно отображать графически. Нужно иметь в виду, что некоторые задачи имеют межгрупповые связи.

Определяется и утверждается количество таблиц, в которых будет содержаться база данных. Рекомендуется предварительно разработать таблицы на бумаге. При этом необходимо учитывать принципы проектирования таблиц:



- а) каждое поле должно быть связано с темой таблицы;
- б) каждая таблица должна содержать информацию только на одну тему.

*Третий этап* – анализ данных. Определение структуры каждой таблицы: перечень полей таблицы и последовательность их размещения. Каждая таблица должна содержать информацию по отдельной теме, а каждое поле – отдельные данные по теме таблицы. При выборе полей необходимо руководствоваться следующим:

- а) каждое поле должно быть связано с темой таблицы;
- б) в таблицу не включаются поля, которые являются результатом вычислений;
- в) в таблице должна присутствовать вся необходимая информация для удовлетворения запросов пользователей;
- г) каждое поле должно содержать наименьшие логические единицы информации (не «Товар», а «КодТовара» и «НаименованиеТовара»).

Для каждой задачи составляется подробный перечень всех элементов данных (реквизитов), которые нужны для ее решения. Указывается назначение каждого элемента: а) входного (Input); б) выходного (Output); в) сменяемого (Update); г) удаляемого (Delete); д) вычисляемого (Calculate).

*Входной* реквизит необходимо прочитать без его изменения с базы данных для выполнения задачи; *выходной* реквизит вводится или вычисляется на основе входных данных, а затем сохраняется; *сменяемый* реквизит читается в базе данных, а затем сменяется и записывается опять на старое место; *удаляемый* реквизит удаляется из базы данных; *вычисляемый* реквизит – это новое значение, которое получается на основе входных данных, не сохраняется в базе данных, но выводится на дисплей, принтер.

*Четвертый этап* — определение структуры данных. Элементы (реквизиты) данных каждой задачи после предварительного анализа нужно упорядочить по объектам и создать эффективный и гибкий способ сохранения информации в базе данных. Это достигается путем *нормализации таблицы* (см. п. 3.3). Следует задать индивидуальное значение каждому полю в каждой записи таблицы. Каждая таблица должна содержать поле (несколько полей), указывающее индивидуальное значение каждой записи в таблице — *основной ключ*.

*Пятый этап* – разработка макета базы данных (задачи) в целом и ее объектов (форм, запросов, отчетов, макросов и др.). Необходимо определить связи между таблицами. После определения структуры таблиц и ключевых полей в них надо установить связи между таблицами, которые находятся в логико-информационных отношениях.

*Шестой этап* – создание базы данных (приложения). Для полной автоматизации отношений с базой данных пишутся нужные процедуры, специальные формы связи для перехода между задачами, разрабатываются специальные меню. Непосредственно перед заполнением таблиц информацией

необходимо еще раз проверить, проанализировать их структуру и внести необходимые уточнения, изменения – с помощью контрольного примера.

*Седьмой этап* – тестирование и усовершенствование созданного. Проверяется функционирование базы данных в возможных режимах, а также работа макросов и процедур, выявляются и исправляются ошибки.

Добавляются и создаются другие объекты базы данных (таблицы, запросы, формы, отчеты, макросы, модули). После утверждения структуры таблиц базы данных и их взаимосвязей можно приступить к заполнению их информацией, на основе которой MS Access создаст любые объекты (запросы, формы, отчеты, макросы, модули).

Для оптимизации таблиц базы данных MS Access имеет два инструмента: Мастер анализа таблиц и Анализатор быстрогодействия базы данных.

Мастер анализа таблиц анализирует таблицу и предлагает ее новую структуру и связи, а также переделывает таблицы (нормализация таблиц). Для этого подается команда Сервис – Анализ – Таблица, далее следует выполнять инструкции Мастера.

Анализатор быстрогодействия исследует всю базу данных и выдает рекомендации по ее усовершенствованию. Для этого необходимо:

а) открыть базу данных, которая предназначена для усовершенствования;

б) дать команду **Сервис→Анализ→Быстродействие**;

в) выбрать вкладку, на которой находится объект базы данных; выбрать вкладку «Все» для просмотра списка всех объектов базы данных;

г) выбрать имена активизируемых объектов базы данных (для всех – выбрать кнопку «Выбрать все»);

д) кнопка **«ОК»** выдает список трех типов предложений по оптимизации: «Рекомендации», «Предложения», «Идея»;

ж) выбрать вид рекомендации и нажать кнопку **«Оптимизировать»**.

Руководствуясь итогами функционирования (эксплуатации) базы данных, пользователь вносит свои предложения относительно ее усовершенствования. База данных передается в эксплуатацию, разработчики наблюдают за ней и производят нужные усовершенствования.

**Методы разработки базы данных.** При разработке базы данных применяются два метода:

- проектирование начинается с определения основных функций и задач («сверху вниз») и продолжается до определения структуры таблиц и других объектов;

- проектирование начинается с анализа данных и определения их структуры («снизу вверх»).

На практике эти методы применяются не в чистом, а комбинированном виде.

**Информация и данные в базе данных.** Эти понятия хорошо известны из школьных предметов и курса «Информатика и вычислительная техника». Вместе с тем целесообразно подчеркнуть важные нюансы. В базах данных сохраняются данные.

*Данные* – это статичные значения, которые *сохраняются* в таблицах базы данных. Ключевое слово в определении – «статичные».

*Информация* – это данные, которые пользователь базы данных *запрашивает* и получает в соответствующем виде. Ключевое слово в определении – «запрашивает».

Таким образом, в базе данных информация запрашивается, а данные сохраняются. Именно такой смысл понятий «данные» и «информация» следует учитывать при разработке базы данных в целом и ее отдельных объектов. Рекомендуется сначала определить, что надо получить из базы данных (какую информацию) и что следует запрашивать. Получив ответ на эти вопросы, следует уточнить, что нужно для этого, какие данные должны быть в базе данных, чтобы пользователь получил нужную ему информацию в надлежащем виде.

**Взаимодействие задач базы данных.** Описание элементов каждой задачи базы данных и назначения элементов (входной, выходной, сменяемый, удаляемый, вычисляемый) используется для установления и отображения взаимосвязей задач базы данных, которые приводятся в виде схемы.

*Анализ данных.* Элементы данных объединяются в объекты. Устанавливается тип связей («один-к-одному», «один-ко-многим»). Для каждого элемента данных указывается его тип (числовой, символьный и т. д.), длина поля, дается кратко описание элемента. Задаются условия для определения конкретного поля.

### 3.3 Нормализация отношений

*Нормализация* (normal form) – это процесс уменьшения избыточности информации в базе данных с целью получения более эффективных и гибких способов хранения данных. Это понятие имеет отношение к зависимости в реляционных базах данных.

При проектировании реляционной базы данных необходимо создать наиболее эффективную структуру данных, которая должна обеспечить:

- а) быстрый доступ к данным в таблицах базы данных;
- б) исключение из структуры таблицы ненужного повторения данных, которое приводит к нерациональному употреблению дискового пространства и ошибкам при вводе данных;
- в) целостность данных: при изменении одних объектов автоматически соответствующим образом изменяются связанные с ними объекты.

Теорию нормализации отношений в реляционной модели разработал Кодд. Сначала использовались три нормализованные формы; позже были

выделены еще две. Таким образом, разработаны пять нормализованных форм таблиц.

Отношения, которые соответствуют выделенным условиям ограничения, имеют нормальную форму. Для всех нормальных форм условием ограничения является следующее: отношения не должны носить характер вложения.

Нормализация отношений представляет собой структурный метод ликвидации избыточности и несогласованности, а также обеспечения эффективной обработки, хранения и обновления информации.

Все формы нормализации отношений предназначены для уменьшения избыточности информации.

Каждая последующая (от первой до пятой) нормальная форма должна удовлетворять требованиям предыдущей и некоторым дополнительным условиям. Четвертая и пятая формы при практическом проектировании, как правило, не применяются.

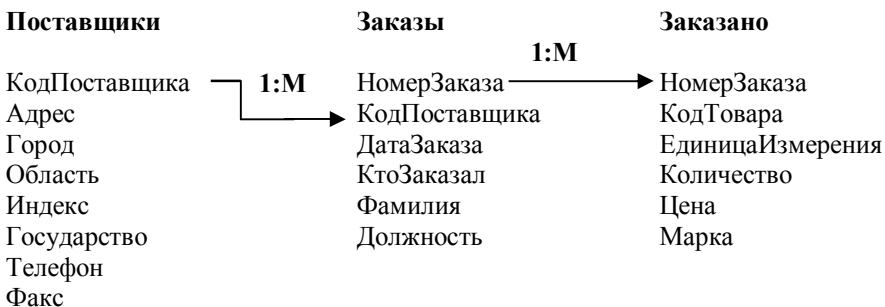
**Первая нормальная форма таблицы.** Эта форма наименее структурирована; каждое поле содержит уникальную информацию. Исходная таблица, как правило, ненормализована.

Требования, предъявляемые к таблице в первой нормальной форме:

- а) таблица не должна содержать повторяющиеся записи;
- б) в таблице не должно быть повторяющихся групп полей;
- в) строки должны быть неупорядоченными;
- г) столбцы должны быть неупорядоченными.

Рассмотрим эти теоретические положения на примере. Предположим, таблица «Поставщики» имеет следующую структуру (перечисление полей): Поставщик, Адрес, Город, Область, Индекс, Страна, Телефон, Факс, НомерЗаказа, ДатаЗаказа, КтоЗаказал, Фамилия, Должность, Марка, КодТовара, ЕдиницаИзмерения, Цена, Количество, Тариф.

Исходная таблица «Поставщики» ненормализована. На основании одной ненормализованной таблицы в соответствии с теорией нормализации сделаем многотабличную базу данных эффективной структуры.



Поскольку согласно требованиям (п. «а») каждая таблица должна иметь уникальный индекс, чтобы избежать записей, которые повторяются, введем универсальный индекс поля «КодПоставщика».

Чтобы ликвидировать группы полей, которые повторяются (п. «б»), необходимо разбить исходную таблицу на три отдельные таблицы и установить совпадающие поля для связывания таблиц «один-ко-многим»:

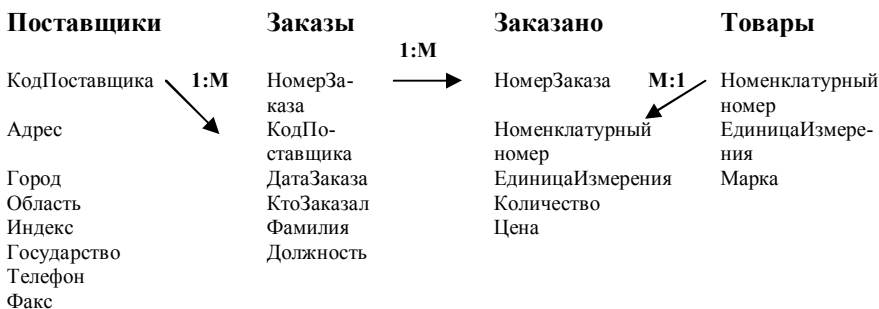
В таблице «Поставщики» поле «КодПоставщика» является ключевым. В таблице «Заказы» ключевым полем также является «КодПоставщика». В таблице «Заказано» содержатся сведения о товарах, которые включены в конкретный заказ.

Таблицы «Поставщики», «Заказы», «Заказано» находятся в первой нормальной форме, потому что часто повторяющиеся объекты разделены и определены поля-ключи в каждой таблице для связей между ними.

**Вторая нормальная форма таблицы.** Таблица относится ко второй нормальной форме, если:

- а) она соответствует условиям первой нормальной формы;
- б) любое не ключевое поле однозначно идентифицируется полным набором ключевых полей.

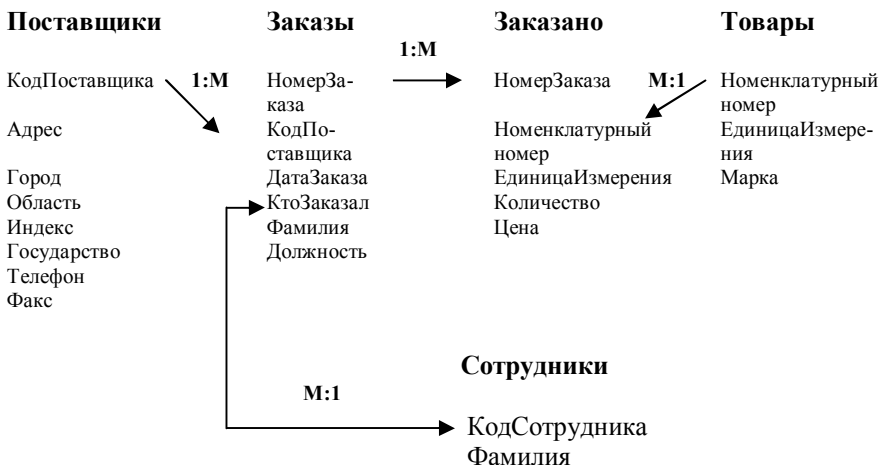
Вторая нормальная форма применима только для таблицы с составным ключом. В нашем примере это таблица «Заказы», в которой два таких поля: «НомерЗаказа» и «КодПоставщика». Поля «Единица Измерения», «Цена» однозначно определяются только одним из ключевых полей, а именно «НомерЗаказа». Поэтому таблица «Заказы» не является таблицей второй нормальной формы. Чтобы привести ее к этому виду, необходимо выделить из таблицы «Заказано» таблицу «Товары» с полями, которые характеризуют товар каждого номенклатурного номера. В итоге получаем:



**Третья нормальная форма таблицы.** Таблица относится к третьей нормальной форме, если:

- а) она соответствует условиям второй нормальной формы;
- б) ни одно из ключевых полей таблицы не идентифицируется с помощью иного не ключевого поля.

В таблице «Заказы» поле «Фамилия» однозначно определяет значение поля «КтоЗаказал». Так как не ключевое поле «Фамилия» определяется другим не ключевым полем «КтоЗаказал», то таблица «Заказы» не является таблицей третьей нормальной формы. Чтобы привести ее к этому виду, необходимо создать новую таблицу «Сотрудники»:



## 4 СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

### 4.1 Понятие системы управления базами данных

Базы данных создаются для дальнейшего интегрированного использования. Их поддержание в компьютерной среде осуществляют программные средства – СУБД (database management system), которые представляют собой совокупность программных и языковых средств общего или специализированного назначения, необходимых для создания баз данных на машинных носителях, поддержания их в актуальном состоянии и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных. СУБД – это управляющие программы, которые обеспечивают все манипуляции с базами данных: создание базы, ее ведение, ее использование многими пользователями и др., т. е. реализуют сложный комплекс функций по централизованному управлению базой данных и обслуживают интересы пользователей.

СУБД можно рассматривать как программную оболочку, которая находится между базой данных и пользователем. Она обеспечивает централизованный контроль защиты и целостности данных, доступ к данным, их обра-

ботку, формирование отчетов на основе базы данных и другие операции и процедуры. СУБД является универсальным программным продуктом общего назначения (рисунок 4.1).



Рисунок 4.1– Система управления базой данных

Понятия «база данных» и «система управления базой данных» образуют во взаимосвязи понятие «банк данных» (date bank) – хранилище реальной совокупности данных, система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования. База данных и банк данных имеют архивы, которые в совокупности образуют фонд данных.

**Языковые средства систем управления базами данных.** Каждая СУБД функционирует в конкретной языковой среде. Именно с помощью языковых средств и посредством их реализуются заложенные функциональные возможности СУБД и удовлетворяются потребности, запросы как профессионалов (администрирование баз данных, разработка прикладных программ этой СУБД и для этой СУБД, описание представления баз данных и др.), так и конечных пользователей.

В зависимости от уровня профессионализма пользователей языковые средства реализуются следующим образом: для разработчика (системщика) сложных приложений языковая среда представлена в явной синтаксической форме, а для непрофессионалов-программистов – в виде меню, диалога. В последнем случае интерфейсные средства автоматически формируют адекватные синтаксические конструкции языка интерфейса, которые выполняются системой. В современном интерфейсе в СУБД языковые средства применяются в неявном виде (команд в классическом смысле не видно).

Язык СУБД выполняет две группы функций: представление баз данных на уровне системной архитектуры (язык описания данных); осуществление операций манипулирования с данными.

**Архитектура СУБД** – это описание на соответствующем общем уровне структуры СУБД и базы данных.

Современные СУБД имеют общий функциональный состав, состоящий из трех основных частей:

- диалоговые средства пользователя для работы с базой данных;

- средства для разработки приложений пользователя;
- дополнительные средства для разработки приложений (программного продукта).

Каждая функциональная часть СУБД включает в себя компоненты, отображенные в общей функциональной схеме интерфейса СУБД (рисунок 4.2).

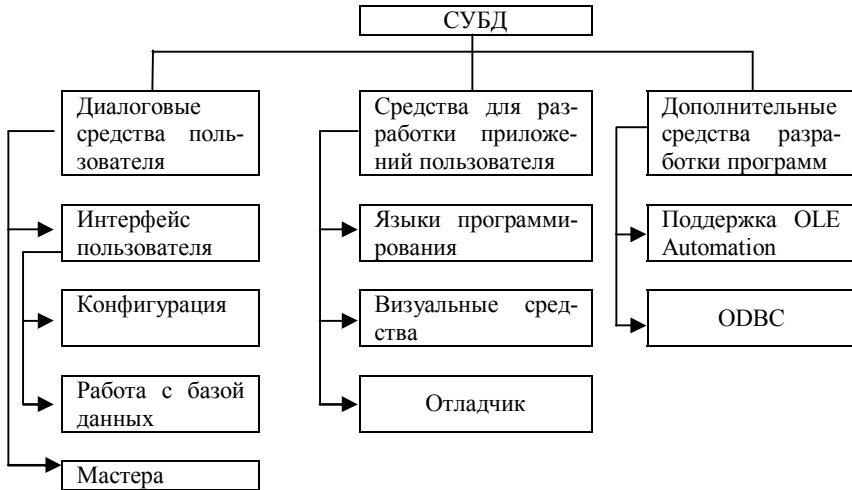


Рисунок 4.2 – Функциональный состав СУБД

**ODBC** (Open Database Connectivity – взаимодействие с открытыми базами данных) – это технология работы с внешними данными (доступ к данным) в системе «клиент – сервер». В этой системе приложения делятся:

- на клиентскую часть (Font-end), размещенную на компьютере пользователя в виде интерактивного графического интерфейса;
- сервер (Back-end), обеспечивающий управление базой данных, разделение информации, администрирование и безопасность. Сервер находится на базовом компьютере (mainframe – мэйн-фрэйм) или специально выделенных компьютерах.

Сущность технологии «клиент – сервер»: приложение клиента (front-end) формирует запрос к серверу базы данных (back-end), который выполняет все команды и их результаты посылает клиенту.

#### 4.2 Классификация систем управления базами данных

Базы данных и СУБД созданы для экономии ценного для человека ресурса – времени. В мире существует много СУБД, основанных на многочисленных методических и организационных принципах и ориентированных на различные технические и системные программные средства.



Все СУБД построены на идеях базы данных CODASYL, разработанных группой американских специалистов в 1971 г. Разработка СУБД осуществлялась с ориентацией на развитие вычислительной техники. Так, для ЭВМ единой системы (ЕС ЭВМ) и операционной системы ОС ЕС было разработано семейство СУБД «БАНК», а также «СЕТОР», «СЕТАН», «ОКА», «ИНЕС». Для ЭВМ семейства малых машин (СМ ЭВМ) разработана диалоговая многотерминальная система «ДИАМС» и др.

В мире насчитывается более 50 типов СУБД для компьютеров IBM PC и совместимых с ними. Наиболее популярной является Microsoft (MS) Access, которая входит в комплект MS Office. Для персональных компьютеров реляционного типа применяются MS SQL Server, Oracle, DB2, Rdb и др.

С точки зрения универсальности СУБД делятся на системы общего назначения и специализированные системы.

СУБД *общего назначения* функционируют на различных объектах, разрабатываются без твердой привязки к конкретной предметной области. Эти СУБД имеют средства настройки (привязки) для работы с базой данных на конкретном объекте управления, что обеспечивает экономию времени при создании автоматизированных информационных систем.

Безусловно, универсальность СУБД общего назначения предусматривает наличие широкого, даже избыточного диапазона функций, на основе которых средствами настройки создается информационная технология для конкретной предметной области. СУБД общего назначения выполняют всю совокупность функций по созданию базы данных.

*Специализированные* СУБД разрабатываются для конкретного применения в том случае, когда путем привязки СУБД общего назначения нельзя или сложно создать систему с нужными параметрами. Учитываются специфические особенности предметной области. Разработка специализированных СУБД по сравнению с привязкой СУБД общего назначения очень трудоемка.

*Признаки классификации СУБД<sup>1</sup>:*

- среда функционирования, класс компьютеров и операционных систем (платформ);
- тип модели базы данных, которую поддерживает СУБД (сетевая, иерархическая, реляционная);
- возможность и наличие встроенного языка СУБД, возможность ее перенесения в другие приложения (SQL, Visual Basic и др.);
- наличие развитых средств конструирования (таблицы, формы, отчеты, запросы, макросы) и средств работы с базой данных;

---

<sup>1</sup> Экономическая информатика: Учебник / Под ред. В.В.Евдокимова. – СПб.: Питер Пабблишинг, 1997.

- возможность работы с нетрадиционными данными в корпорационных сетях (страница HTML – Hypertext Markup Language – язык описания, разметки документа, который позволяет задавать структуру и формат документа): сообщения электронной почты, отображение, звуковые файлы, видеоклипы и др.;

- применение концепции с нетрадиционными данными (объектно-реляционной и объектной);

- уровень применения (локальные для настольных систем, архитектура «клиент–сервер», с параллельной обработкой данных – многопроцессорные);

- применение объектной технологии OLE 2.0;

- возможность интегрирования данных в различных СУБД;

- степень поддержки стандартного языка программирования запросов для работы с реляционными базами данных SQL (Structured Query Language) и возможность работы с сервером базы данных (SQL)<sup>1</sup>;

- наличие средств отчуждения приложений, которые позволяют не проводить полной инсталляции СУБД для тиражируемых приложений пользователя.

#### **4.3 Краткий исторический обзор систем управления базами данных для персональных компьютеров**

База данных есть интегрированная совокупность недублированных данных соответствующей предметной области.

Современные СУБД имеют средства организации запросов к базе данных, которая ориентирована на пользователя-программиста, и вывод данных. Все СУБД имеют диалоговые системы для работы пользователя с данными. До повсеместного применения Windows распространенными были реляционные СУБД dBASE-группы (семейство стандартов Xbase). СУБД этого стандарта имели относительно простой командный язык для манипулирования данными, интерфейс, представленный в виде меню, генераторы отчетов и экранных форм. Программы писались на базовом языке и выполнялись в режиме интерпретации.

Базы данных, которые разрабатываются и функционируют по методике сетевых моделей, можно выделить в отдельную группу (направление развития). Эти СУБД предназначены для создания больших баз данных с ориентацией на широкий диапазон компьютерной техники и программного обеспечения. Пример – система db-VISTA, которая включает универсальный язык C, а также СУБД AdabasD.

В конкурентном развитии семейства dBASE на рынке появилась СУБД Paradox, которая включает язык запросов по образцу QBE (Query By Exam-

---

<sup>1</sup> Сервер баз данных – вид программного обеспечения для создания и применения при работе в сети интегрированных баз данных в системе «клиент–сервер».

ple), более развитые диалоговые средства, генератор приложений на основе объектного подхода, меню пользователя, которое можно настраивать, базовый язык программирования PAL (Paradox Application Language) для разработки приложений.

**Средство быстрой разработки приложений. RAD** (Rapid Application Development) – современное средство быстрой разработки приложений в информационных технологиях. Для RAD характерны:

- применение объектно-ориентированных языков программирования, что позволяет применять модульный принцип составления программ;
- визуальные средства разработки программ (вместо написания кодов команд рисуется интерфейс пользователя, и ему задаются нужные функции диалоговыми средствами);
- поддержка стандартных протоколов обмена информацией между приложениями, что позволяет разрабатывать многоуровневые приложения, независимые от источника данных;
- возможность создания приложений «клиент—сервер», что позволяет конструировать приложения неограниченной сложности и обеспечивать требования обработки информации на всем объекте.

Функции современного программиста сводятся к выбору готового программного модуля (блока, кубика) и вставке его в нужное место в создаваемой программе, т. е. RAD облегчает труд программиста.

Microsoft включает широкий диапазон средств разработки программного продукта: Access, SQL Server, Visual Basic, Visual C++, Visual FoxPro. Эти пакеты можно применять для создания приложений пользователя по обработке информации — как в среде одного пакета, так и в виде интеграции средств из различных пакетов. Перечисленные пакеты поддерживают новые технологии: OLE, ODBC, DAO, RDAO, ActiveX и др. Можно разрабатывать приложения независимо от данных.

#### **4.4 Режимы работы пользователя с системой управления базой данных**

В информатике считается, что термин «режим» определяется как определенный порядок работы или состояния компьютера или программы. Пользователь может работать в одном из следующих режимов: интерактивный, под управлением меню команд, выполнение программ (программный). Рассмотрим их подробнее.

*Интерактивный режим* (непосредственная интерпретация команд) – это способ реализации возможностей языка, т. е. непосредственное выполнение команд. Система выдает подсказку и ожидает ответа – ввода соответствующей команды. После ввода команды система осуществляет синтаксический контроль текста введенной команды и (при отсутствии ошибок) выполняет

команду. Команда в процессе ее выполнения может проводить собственный диалог с пользователем или выдавать конкретные сообщения. После выполнения текущей команды система постоянно выдает подсказку (приглашение) о готовности принять очередную команду.

*Режим под управлением меню команд.* Сеанс работы пользователя выполняется под управлением многоуровневых меню. На первом (высшем) уровне находится группа команд, которые выполняют однотипные функции (например, «Файл»), а на другом, как правило, – одна команда.

*Режим выполнения программ.* Пользователь может писать программы на языке команд, который поддерживает СУБД, производить отладку и выполнение программ. Текст программы можно вводить с помощью встроенного текстового или любого другого редактора. Программы сохраняются в виде файлов с расширением (-prg).

#### **4.5 Функциональные возможности системы управления базой данных**

Современные СУБД обеспечивают:

- интегрирование данных, которые не дублируются, в единой базе данных и их многоцелевое использование;
- набор средств управления данными и обеспечение их целостности и непротиворечивости;
- однократность ввода информации;
- развитый благоприятный интерфейс для пользователя, который позволяет вводить и модифицировать информацию, выполнять поиск и представлять информацию в текстовом и графическом виде;
- средства программирования высокого уровня, что позволяет создавать собственные приложения;
- защиту данных от сбоев и несанкционированного доступа;
- выборку информации, которой интересуется пользователь;
- распечатку информации в нужном виде (вся таблица, выбранные записи и поля);
- отображение данных таблиц в графическом виде.

В процессе подготовки отчета, выбора данных можно выполнять различные вычисления.

**Архитектура «файл–сервер» и «клиент–сервер».** Удаленный доступ к данным реализуется по одной из двух концепций.

*Концепция «файл–сервер»:* файлы базы данных с файл–сервера передаются для обработки на рабочих станциях (рисунок 4.3).

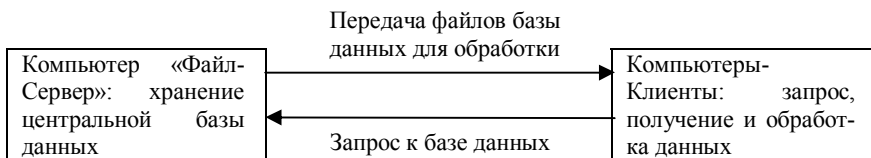


Рисунок 4.3 – Архитектура «файл-сервер»

*Концепция «клиент-сервер»:* хранение централизованной базы данных на центральном компьютере (сервер базы данных) и основная обработка данных. По запросу клиента (рабочей станции) на сервере отыскиваются и выбираются нужные данные и транспортируются по сети от сервера до рабочих станций (рисунок 4.4). Основной объем обработки данных выполняется центральной машиной.

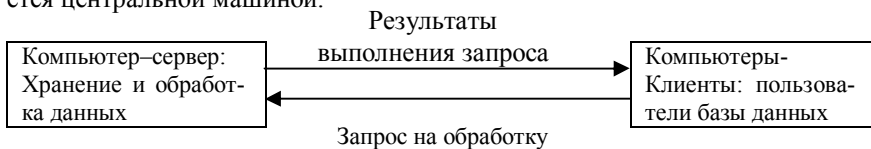


Рисунок 4.4 – Архитектура «клиент-сервер»

#### 4.6 Обзор современных систем управления базами данных

Для всех современных СУБД характерно наличие инструментария поддержки целостности данных в многопользовательской базе данных – так называемого блокирования. Сущность этого понятия в том, что когда с элементом данных работает один пользователь, другие пользователи в это время не могут изменить значение данного элемента.

Обеспечивается мониторинг базы данных (набор статистической информации) для оценки производительности функционирования системы. Приведем основные сведения о некоторых СУБД.

**SQL Server 7.0.** Реляционная СУБД в среде операционных систем (ОС) Windows NT и Windows 95/98. Основные параметры: может управлять до 32 767 базами данных, а каждая база может включать до 2 млрд таблиц. В одной таблице базы данных может быть до 1024 столбцов; количество строк не ограничивается. Для одной таблицы может быть определено до 250 индексов. SQL Server 7.0 имеет архитектуру «клиент-сервер».

Для формирования запросов применяется язык программирования запросов Transact-SQL. Имеет компоненту «координатор распределенных транзакций» (выполняются на нескольких серверах), что позволяет осуществлять более 30 тыс. транзакций в минуту.

**Oracle 8.** Объектно-реляционная СУБД, т. е. построенная на основе соединения объектно-ориентированной и реляционной теорий. Она разрабо-

тана с ориентацией на ОС Windows NT и Solaris. Система хранит большие системные объемы информации объекта управления, что позволяет не только выполнять классические, традиционные процедуры, но и поддерживать процедуры принятия управленческих решений.

Есть данные, что СУБД Oracle 8 может функционировать в среде 10 тыс. пользователей и базах данных объемом 100 терабайт<sup>1</sup> (1 TB = 1 048 576 000 000 байт). Выполняет от 40 до 110 тыс. транзакций в минуту.

СУБД Oracle 8 поддерживает технологию распределенной обработки данных, совмещенную с функционированием централизованной базы данных, и архитектуру «клиент–сервер», а также обработку данных в WWW (публикация данных в Internet).

Применяется язык запросов SQL PLUS, который кроме выполнения функций SQL обрабатывает наборы данных связанных объектов. Для программирования задач имеется процедурный язык PL/SQL.

Oracle 8 имеет высокоэффективные генераторы экранных форм и отчетов. Поддерживает совместимость баз данных других систем (Access, DB2 и др.).

**Informix.** СУБД реляционного типа в среде ОС UNIX, управляет структурированными и неструктурированными данными. Реализуется параллельная обработка данных различных типов (традиционные, трехмерные данные, звук, видео, изображение, кодированные документы). Соответственно управляет данными географических информационных систем, компьютерной полиграфией.

Используется язык доступа к данным Informix-SQL. Организовано взаимодействие Web-приложений с базами данных. Можно создавать интеллектуальные Web-приложения. Администратор базы данных осуществляет ежедневный мониторинг базы данных (статистическое измерение и отображение функционирования системы).

**Sybase.** СУБД реляционного типа. Имеет несколько серверов, каждый из которых выполняет соответствующие функции: унифицированный доступ к разнородным источникам данных различных СУБД (Informix, Oracle, DB2 и др.); массовая параллельная обработка; оптимизация предварительных запросов и запросов «налету»; одновременное обслуживание до 32 устройств загрузки и выгрузки частей базы данных. Реализован принцип параллельной обработки больших баз данных в реальном времени.

**DB2.** Семейство современных СУБД (DB2MVS, DB2 Common Server и др.) универсального типа. Серверы ИС\$2 работают под управлением OS/2, AIX, Windows NT, HP-UX, Solaris, SCO, SINIX. Обрабатывает мультиме-

---

<sup>1</sup> tera- (T) – префикс, означающий один триллион; terabyte (TB) – терабайт – мера, используемая для запоминающих устройств с высокой плотностью записи данных.

дийные данные. Можно создавать приложения на языках C, C++, Basic. Применена технология распараллеливания, что обеспечивает поддержку современных баз данных больших объемов. Для пользователей могут создаваться группы с общим доступом к дисковому массиву данных.

Размер одной таблицы может быть до 1 ТВ или 254 независимых разделов по 4 Гбайт. Поддерживает несколько методов формирования интерактивных запросов, обмен с серверами Internet.

**Ingres II.** СУБД объектно-реляционного типа. Применена Web-технология. Взаимодействует с базой данных Oracle, Sybase, xBASE, Informix, Microsoft SQL Server. Поддерживается ОС UNIX и подобными ей, Siemens, Open VMS, Tandem, Windows NT..

**xBASE.** СУБД, использующая объектно-ориентированный язык dBASE. Построена по архитектуре «клиент–сервер» с программно-изменяемой клиентской частью.

**IMBASE.** Специальная СУБД иерархического типа, ориентированная на автоматизацию проектных и конструкторских работ. Архитектура «клиент–сервер». Поддерживается ОС Windows 95/98 и Windows NT. СУБД IMBASE можно устанавливать локально.

**Access 2000.** Реляционная СУБД. Технология организации и обработки данных производится в системе «клиент–сервер» для корпоративных баз данных. Применять можно в сети и локально.

**Объектно-ориентированные СУБД.** На момент написания пособия известны следующие СУБД: SemStone, Vbase, ORION, PDM, IRIS и др. Применяются объектно-ориентированные языковые и программные средства. В этих СУБД обеспечиваются параллельный доступ к данным и контроль параллельной обработки и восстановления данных. Применяются абстрактные типы данных.

В среде этих СУБД функционируют понятия (термины) «объект», «класс», «расширяемость», «наследование», «инкапсуляция», «полиморфизм».

*Класс (class)* – это средство для описания группы объектов в рамках одного класса и полной характеристики любого объекта данного класса. Классы определяют типы содержащихся в них объектов и способы их поведения.

*Расширяемость (extensibility)* – возможность определения на основе существующих объектов новых объектов.

*Наследование (inheritance)* – передача определенных свойств от класса к его производным более низкой иерархии.

*Инкапсуляция данных (data encapsulation)* – независимость каждого объектного модуля (в пределах разумного) от других, для которых доступен только сам модуль в целом, но не его содержимое.

*Полиморфизм* (polymorphism) – возможность переопределения процедуры в производном классе. Последний наследует свои структуры данных и процедуры из другого класса. Базовый класс не определяет точно тип объектов в производном классе.

#### **4.7 Тенденция развития систем управления базами данных**

В недавнем прошлом понятие «автоматизация обработки информации» означало так называемые традиционные процедуры: сбор, регистрация, обработка, передача, хранение и выдача информации. Развитие элементной базы для компьютеров, а также программного обеспечения и его части (операционных систем) позволило начать переход к обработке нетрадиционных данных (графика, рисунки, тексты, звуковая информация, видео, страницы HTML – язык разметки гипертекста) и другой информации. Возрастают роль интеграции информации, необходимость обмена информацией между базами данных различных СУБД, работы в локальных сетях в условиях использования информации базы данных многими пользователями.

Определенное влияние на развитие СУБД оказала ОС Windows 95, которая с 1996 г. принята в качестве стандарта для персональных компьютеров. Популярными 32-разрядными СУБД являются Access, Paradox 7 for Windows 95 and Windows NT (Borland) и Lotus.

Согласно литературным данным основным направлением в развитии СУБД является разработка универсальных СУБД (традиционная и нетрадиционная интегрированная обработка). Существуют два направления: объектное и объектно-реляционное (совершенствование существующих реляционных СУБД). Нетрадиционные данные в современных СУБД недоступны для внутренней обработки. Разрабатываются объектно-реляционные СУБД DB2 для ОС AIX и OS/2, интегрированная СУБД Oracle, Universal Server. Получают дальнейшее развитие специализированные серверы для поддержания данных в разных базах.

Дальнейшее развитие СУБД предусматривает:

- улучшение многооконного графического интерфейса;
- возможности работы в локальном режиме и в режиме клиента на рабочей станции;
- расширение специальных средств (мастера, конструкторы, эксперты);
- наличие собственных языков программирования;
- применение объектной технологии типа OLE2.

Интеграция баз данных требует разработки проблем параллельной обработки информации на многопроцессорных вычислительных системах. В этом направлении предусмотрено развитие СУБД:

а) без раздела ресурсов (за каждым процессором закреплены выделенные области памяти и диски), например DB2, Navigation Server;



б) с совместным использованием дискового пространства (все процессоры делят между собой операционную память и место на диске), например СУБД AdabasD.

В целом СУБД совершенствуются и развиваются с учетом совершенствования и расширения функциональных возможностей, а также архитектуры СУБД и качества внешнего интерфейса; расширяются возможности обработки сложных запросов с одновременным увеличением скорости обработки.

Развитие баз данных, СУБД и Internet происходит в направлении расширения их «интеллекта» за счет развития и применения интеллектуальных программных и языковых средств, что приведет к созданию баз знаний по соответствующим предметным областям и глобальной базы знаний.

## 5 СТРУКТУРА MICROSOFT ACCESS

### 5.1 Общая характеристика

*Microsoft Access* (англ. access – доступ) – это популярная система управления базой данных, построенная на основе реляционных моделей и ориентированная на 32-разрядные микропроцессоры. В настоящее время в основном используется Access 2000 для Windows 95-98-2000 и MS Access XP. Первая версия Access появилась в 1992 г.

Access может выступать в качестве клиента или сервера по отношению к другим приложениям, например MS Word, MS Excel и др. MS Access содержит в себе развитую систему обработки приложений для Windows. Программа не пишется в классическом смысле этого термина – достаточно определить формы, отчеты и с помощью простых команд (инструкций) Visual Basic связать их с данными базы.

Access – полнофункциональная реляционная СУБД. В этой системе можно пользоваться практически всеми средствами операционной системы MS Windows. Применяется многодокументальный интерфейс (MDI – Multiple Dokument Interface), т. е. одновременно можно работать с несколькими таблицами, формами, отчетами, макросами, модулями.

Таблица является основой базы данных. В Access вся информация содержится в таблицах. Обеспечена связь между объектами. Таблицы можно связывать простым наглядным способом.

Access относится к функционально полной реляционной СУБД и имеет все необходимые средства для выполнения трех комплексных типов функций:

- а) определение данных (задание структуры базы данных и их описание);
- б) обработка данных;

в) управление данными (можно указать, кому данные доступны, их корректность, можно выделить правила коллективного пользования и др.).

MS Access – это современное приложение Windows. Можно использовать все возможности DDE (Dynamic Data Exchange – динамический обмен данными); реализуются также принцип связи и внедрение OLE-объектов, т. е. ссылок на конкретную информацию, которая сохраняется в своей первоначальной форме. OLE-объектом может быть Excel-таблица, Paintbrush-иллюстрация, звуковой файл и др.

Вставлять объекты можно путем копирования объекта через буфер обмена.

Access воспринимает множество различных форматов данных, включая файловые структуры других СУБД (Paradox, dBASE III, dBASE IV, FoxPro и др.).

Access работает с базами данных, которые, будучи созданными в других СУБД, поддерживают стандарт ODBC, включая Microsoft SQL Server, Oracle, DB2 и Rdb.

В Access входят специальные программы: «Конструктор», «Мастер», «Построитель выражений», «Маска ввода», что позволяет создавать экранные формы, отчеты, запросы.

*Программы «Конструктор»* содержат инструментальные средства, позволяющие довольно простым способом составлять нужные формы, запросы, отчеты.

*Программы «Мастер»* помогают проектировать форму или отчет путем ответа пользователя на ряд вопросов для получения законченного отчета (формы, запросы).

Access позволяет обращаться к базам данных, которые созданы в других системах (dBASE, Paradox и др.) без перевода их в формат Access. В Access реализован принцип коллективного представления и обработки данных.

Пользователю предоставляются средства автоматизации разработки приложений – элементов управления: в диалоге с системой пользователь может создать нужные ему элементы управления.

Введено контекстное меню с набором команд – открывается правой кнопкой мыши в любой момент (можно не обращаться к главному меню).

*Построитель выражений* (expression builder), содержащий список готовых выражений, оказывает пользователю помощь при определении (формировании) выражений в таблице, запросе, форме, отчете, макросе, модуле.

Инструментарий «Переместить и отпустить» позволяет перетягивать запросы, таблицы из окна базы данных в формы, отчеты, а также перетягивать таблицы Excel в окно базы данных Access.

Приложение «*Мастер по разработке баз данных*» (Database Wizard) содержит более 20 шаблонов, которые создают таблицы, формы, отчеты.

Программы *Мастер по созданию форм* (Form Wizard) и *Мастер отчетов* (Report Wizard) предназначены для автоматизации проектирования

форм и отчетов. С их помощью можно составлять сложные отчеты на основе более чем одной таблицы. Эти программы анализируют поля, которые включаются в отчет и решают, какую создать форму — простую или отчет.

*Конструкторы форм и отчетов* позволяют создать на основе информации базы данных формы и отчеты, соответствующие современным технологиям.

Все элементы интерфейса приняли трехмерное отображение. Можно прятать объекты от доступа других пользователей к информации. Расширены функции помощи Help.

Функции контроля правильности вводимого текста и исправления ошибок выполняет *Автокорректор* (AutoCorrect), а орфографический контроль информации – *Мастер орфографического контроля*.

MS Access – это не только классическая СУБД, но и система для разработки приложений. Можно разрабатывать приложения для работы с базой данных, не зная языка программирования.

## 5.2 Объекты MS ACCESS

Основу базы данных составляют хранящиеся в ней данные. Однако в базе данных есть и другие компоненты, которые принято называть объектами. К ним относятся:

- *Таблицы* – содержат данные. Каждая таблица содержит информацию о субъектах определенного типа (например, клиентах). Поля (столбцы) таблицы служат для хранения различных характеристик субъектов (например, имена и фамилии), а каждая запись (строка) содержит сведения о конкретном субъекте (например, Иванове Иване Ивановиче).

- *Запросы* – позволяют выбрать нужные данные из одной или нескольких таблиц. Можно создавать запросы на выборку, обновление, удаление и добавление данных. С помощью запросов можно также создавать новые таблицы, выбирая данные по одному или нескольким условиям из одной или нескольких существующих таблиц.

- *Формы* – предназначены в основном для ввода данных. Формы можно использовать для более наглядного представления данных таблиц или наборов записей запросов.

- *Страницы* – файлы в формате HTML, позволяющие просматривать данные Access с помощью Internet Explorer.

- *Отчеты* – предназначены для форматирования, вычисления итогов и печати выбранных данных.

- *Макросы* – выполняют одну или несколько операций автоматически.

Простейшая база данных должна содержать хотя бы одну таблицу. Соответственно, структура простейшей базы данных тождественно равна структуре ее таблицы.

### 5.3 Типы данных MS ACCESS

Access поддерживает девять типов данных:

- **Текстовый** – выбирается для полей с алфавитно-цифровыми данными, длиной не более 255 символов. По умолчанию устанавливается длина 50 символов.

- **Поле МЕМО** – следует использовать только в тех случаях, когда размер текста может превысить 255 символов или в тексте встречаются такие символы, как табуляция или возврат каретки. Длина этого поля составляет до 64 000 символов.

- **Числовой** – выбирается для числовых данных. Причем, в свойстве **размер поля** необходимо определить точность значений данных и объем памяти. Чаще всего используются: *длинное целое* (от  $-2\,147\,483\,648$  до  $+2\,147\,483\,647$ ), *одинарное с плавающей точкой* (числа с точностью до 7 значащих цифр в диапазоне от  $-3,410^{38}$  до  $+3,410^{38}$ ), *действительное* (позволяет хранить числа с заданной точностью от  $-10^{28}$  до  $+10^{28}$ ).

- **Дата/время** – используется для хранения календарных дат или значений времени и позволяет выполнять вычисления в единицах измерения времени: минутах, секундах, часах, днях, месяцах, годах.

- **Денежный** – используется для хранения денежных значений, обеспечивает точность до 15 знаков слева от десятичной запятой и 4 знака справа. Если нужно хранить точные десятичные числа, не являющиеся денежными значениями, лучше задать *числовой тип* данных с размером поля *действительное*.

- **Счетчик** – специально предназначен для автоматической генерации первичного ключа, т.е. полей, имеющих уникальные значения в каждой записи. Таблица не может содержать более одного поля с таким типом данных.

- **Логический** – используется для хранения значений **Истина** (True) или **Ложь** (False).

- **Поле объекта OLE** – позволяет хранить данные, содержащие графические объекты (рисунки, диаграммы), звуковые фрагменты, которые могут иметь динамические связи с другими приложениями Windows.

- **Гиперссылка** – используется для хранения простых или сложных «ссылок» на внешний файл или документ.

**Мастер подстановок** – позволяет ограничить набор значений, которые может содержать поле, т.е. можно создать раскрывающийся список допустимых значений. Значения для раскрывающегося списка могут поступать из определенной таблицы или запроса либо список создается специально для этой цели. Каждая таблица в базе данных должна иметь первичный ключ, т.е. уникальное значение для каждой записи в таблице. Ключ может состоять из нескольких полей. Он необходим для формирования первичного ин-

декса таблицы. Первичный индекс ускоряет поиск по ключевому полю (полям) и используется для определения типа связи между таблицами при объединении таблиц.

Ключевое поле помечается значком ключа в области выделения строки. Для вставки ключевого поля (ключевых полей) необходимо выделить его (их). После чего щелкнуть на кнопку **Ключевое поле** на панели инструментов или выбрать пункт **Ключевое поле** в меню **Правка**.

В процессе создания базы данных при описании свойств полей возможно использование маски ввода. Маска ввода – это способ наложения ограничения на ввод данных в поле. Маска ввода служит различным целям:

- обеспечивает соответствие данных определенному формату, а также заданному типу данных, вводимых в каждую позицию, например, 999-99-9999;
- вводит постоянные символы, такие, как дефис (–), косая черта (/), которые сохраняются вместе со значением поля и не требуется вводить эти символы самостоятельно.

Чтобы задать для поля маску ввода, надо выполнить следующие действия:

1 В окне конструктора щелчком мыши выделить поле, для которого необходимо задать маску ввода.

2 В нижней части окна щелкнуть в строке свойства **Маска ввода**.

3 Далее можно непосредственно ввести маску ввода или щелкнуть на кнопке построения (...) в конце строки и построить маску с помощью Мастера масок ввода.

Если нужно ограничить набор значений, которые может содержать поле, то можно создать раскрывающийся список допустимых значений. Для создания такого списка необходимо выполнить следующие действия:

1 В окне конструктора таблиц выбрать тип данных **Мастер подстановок**.

2 На первом шаге мастера подстановок выбрать вариант **Будет введен фиксированный набор значений** и щелкнуть по кнопке **Далее**.

3 Задать количество столбцов раскрывающегося списка (по умолчанию – 1). Затем в столбцы таблицы в диалоговом окне ввести значения раскрывающегося списка и щелкнуть на кнопке **Далее**.

4 На последнем шаге Мастера подстановок заменить надпись для поля подстановок и щелкнуть на кнопке **Готово**.

Задавать собственный список рекомендуется при небольшом наборе значений подстановки и при достаточно редком внесении в них изменений значений.

Если в таблице есть поле, которое всегда или в большинстве случаев будет содержать одно и то же значение, можно задать для него *значение по умолчанию*, назначив это значение свойству **Значение по умолчанию**.

## 5.4 Запуск MS ACCESS и завершение работы

**Способы запуска MS Access:**

а) стартовое меню «**Пуск**» Windows → команда «**Программы**» → MS Access;

б) в случае отсутствия в меню «Программы» MS Access надо в этом меню выбрать пиктограмму «**MS Office**» и в подменю выбрать «**MS Access**»;

в) выбрать пиктограмму «MS Access» с помощью **Проводника** Windows;

г) выбрать ярлык, созданный для запуска MS Access.

При любом способе запуска на экран выводится окно-приглашение MS Access (рисунок 5.1), в котором находятся две основные опции для создания новой и открытия существующей базы данных.

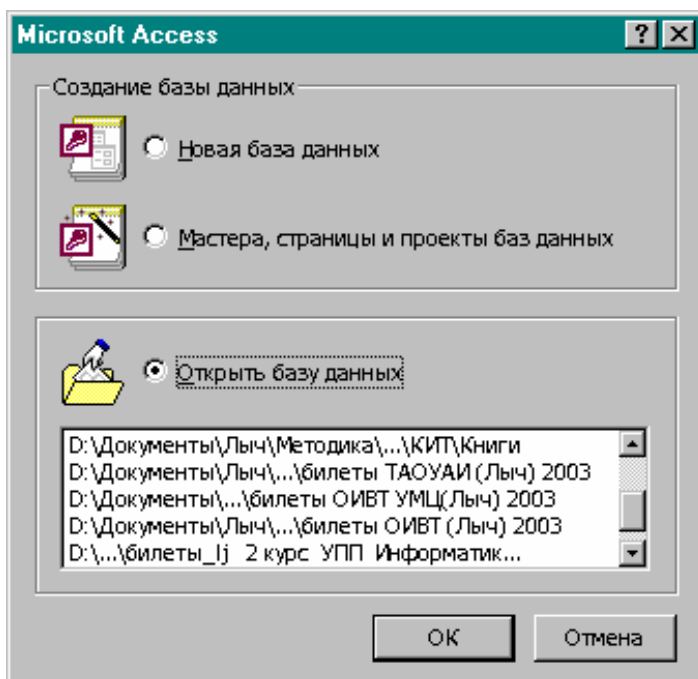


Рисунок 5.1 – Окно-приглашение MS Access

Запустить MS Access можно и через файл базы данных: дважды щелкнуть на имя файла базы данных. Файлы базы данных имеют расширения MDB. «Свежие» файлы базы данных (последние 15) хранятся в стартовом меню «**Документы**». Можно вызвать MS Access и командой «**Выполнить**» в стартовом меню Windows.

### Способы выхода из MS Access:

- а) выбрать команду «Файл» → «Выход»;
- б) выбрать команду «Закрывать» системного меню главного окна MS Access;
- в) в системном меню главного окна дважды нажать кнопку «Закрывать Access-окно».

Если при выходе из Access в каком-либо окне содержится несохраненная информация, то система в окне диалога спросит, сохранить ли ее, или выйти без сохранения изменений, или отменить выход.

**Структура окна Access.** *Интерфейс базы данных* – это комплекс программ (невидимых для пользователя), который организует диалог в процессе работы пользователя с базой данных. Он реализует сопряжение и связь пользователя с базой данных.

После загрузки MS Access на экране появляется *главное окно*, т. е. рабочее пространство, на котором размещаются другие окна. При первом запуске Access выводит на экран окно базы данных и панель инструментов базы данных. Экран включает в себя четыре области: строка меню, панель инструментов, окно базы данных и строка состояния (рисунок 5.2).

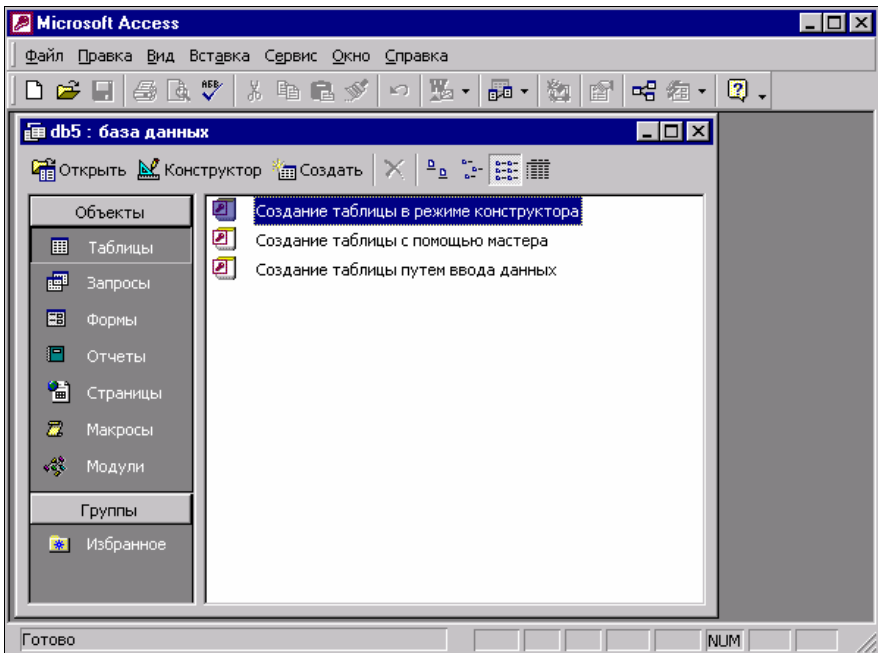


Рисунок 5.2 – Главное окно MS Access

*Строка меню* содержит меню («Файл», «Правка», «Вид» «Вставка», «Сервис», «Окно» и др.) текущего окна, которое меняется в зависимости от объекта, с которым работают. Содержание многих пунктов меню (команды и опции) известны из ОС Windows, Excel, Word. Каждый пункт меню объединяет набор команд общей функциональной направленности, которые раскрываются после вызова имени пункта меню. Начиная работать с компьютером, следует подробно изучить команды пункта меню.

*Панель инструментов* содержит набор кнопок, состав которых зависит от назначения конкретной панели инструментов. Кнопки предназначены для вызова команд и выполнения соответствующих операций. Каждому режиму работы соответствует собственная панель инструментов. Внешний вид панели инструментов можно настроить по своему желанию командой «Вид» → «Панель инструментов» → [диалоговое окно со списком всех панелей]. Выбранные панели помечаются знаком «v». Кнопкой «Создать» в диалоговом окне «Панель инструментов» выбранная панель добавляется на экран. Панель инструментов можно переместить в любое место экрана. Можно также менять размеры панелей инструментов.

*Окно базы данных* имеет шесть вкладок: «Таблицы», «Запросы», «Формы», «Отчеты», «Макросы», «Модули», каждая из которых содержит объекты текущей базы данных. Для перехода на нужную вкладку надо щелкнуть на ее имя. Одновременно можно открыть только одно окно базы данных; при открытии нового окна предыдущее окно закрывается. В правой части окна базы данных находятся три кнопки: «Открыть», «Конструктор», «Создать». Кнопка «Открыть» открывает выбранный объект (таблицу, запрос, форму, отчет). При открытии таблицы открывается окно таблицы; при переходе на вкладку «Отчеты» кнопка «Открыть» меняется на «Просмотр». Кнопка «Конструктор» предназначена для модификации выбранного объекта, кнопка «Создать» – для создания объекта.

*Строка состояния* расположена внизу главного окна. В ней выводится краткая справочная информация о текущем режиме работы или выбранном объекте. О готовности системы принять команду пользователя в этой строке свидетельствует появление сообщения «Готово» («Ready»).

## 5.5 Настройка рабочей среды

Пользователь может с помощью инструментария MS Access – набора опций, которые представляются в диалоговом окне «Опции» – создать по своему желанию рабочую среду (адаптировать Access), наиболее благоприятную для него.

Для этого при открытой базе данных в главном меню надо подать команду: «Опции» → «Инструменты» → [диалоговое окно Опции] (в Access 97 команда «Сервис» → «Параметры» → [Параметры]).



Все параметры настройки объединены в 9 разделов: вид; общие; редактирование/поиск; оформление вывода таблицы; таблицы и запросы; формы и отчеты; клавиатуры; модуль; другие.

С помощью соответствующих опций можно установить режим показа (например, статусной строки, имен колонок), порядок сортировки данных, выбрать нужный способ поиска, установить цвет и вид шрифта, визуальные эффекты. Опции для конструктора таблиц и запросов, форм и отчетов позволяют получать эти объекты в нужном виде. Пользователь может вывести на экран или снять с экрана любую панель инструментов.

Как уже упоминалось, можно перемещать панели инструментов по экрану или сделать панель плавающей.

## 5.6 Выражения. Построитель выражений

Выражение – это основное средство выполнения многих операций, а в MS Access – это любая комбинация операторов, констант, значений текстовых констант, функций, имен полей, элементов управления или особенностей, результатом которой является конкретное значение.

С помощью выражений задаются значения аргументов макрокоманд, условия отбора и вычисляемые поля в запросах. Комбинация идентификаторов, операторов и значений дает соответствующий результат. Идентификатор обеспечивает ссылку на значение поля, элемента управления или особенности.

**Построитель выражений.** Пользователь **имеет** возможность создавать (писать) выражения с помощью построителя выражений. Он содержит список выражений, применяющихся наиболее часто. Для создания выражения с помощью построителя необходимо его вызвать и с его помощью сформировать выражение (которое копируется в то поле, из которого был вызван построитель выражений).

*Способы вызова построителя выражений:*

- нажать кнопку **«Построить»** на панели инструментов;
- в окне конструктора (форм, запросов, отчетов) в строке **«Условие отбора»** в контекстном меню выбрать команду **«Построить»**;
- нажать кнопку **«Построить»** в том месте, где необходимо ввести выражение (в окне Конструктора запросов в строке **«Поле»** или **«Условие отбора»** в окнах особенностей).

*Окно построителя выражений* (рисунок 5.3) состоит из трех частей: поле для создания выражения; кнопки операторов (+, -, \* и т. д.); элементы выражений (три поля).

В верхней области окна находится поле для ввода (формирования) выражения. Ниже размещены кнопки операторов (+, -, \* и т. д.).

В левом поле зоны «Элементы выражений» выводится список папок – источников данных (таблицы, запросы, формы, отчеты) и средства построения выражений (функции, константы и др.); в среднем поле отображается состав элементов, которые входят в объект, выбранный в левом поле; в правом поле выводится список всех выбранных элементов, которые непосредственно входят в создаваемое выражение.

**Функции.** MS Access включает широкий диапазон функций, в том числе функции для предметных экономических областей: статистические, финансовые и др.

Функция возвращает значение, которое является результатом вычислений или выполнения других операций. Например, функция Sum (выражение) возвращает сумму набора значений поля, которые содержатся в заданном поле запроса, и пропускает поля со значением Null (отсутствуют данные либо значения неизвестны). Пример получения суммы произведений двух полей:

SUM (количество \* стоимость).

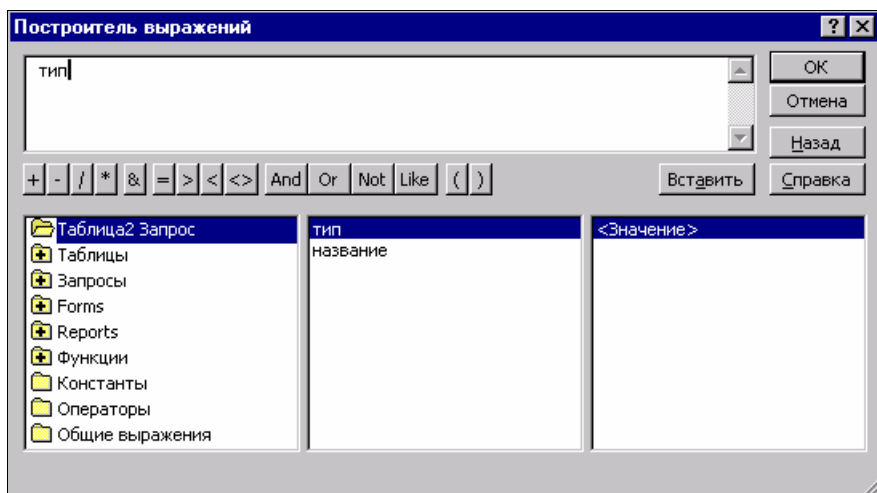


Рисунок 5.3 – Окно построителя выражений

*Статистические функции* применяются для определения статистических данных на основе выбора числовых значений. Эти функции используются в запросах и статистических выражениях. Так, функция Avg (**выражение**) вычисляет среднее арифметическое набора чисел, содержащихся в указанном поле запроса. Операндами выражения могут быть поля таблицы, константы либо функции.

### **Контрольные вопросы**

- 1 Перечислите основные компоненты базы данных.
- 2 Что такое поле базы данных?
- 3 Перечислите основные атрибуты полей базы данных.
- 4 Какие типы данных могут быть использованы в базах данных?
- 5 Что такое ключевое поле?
- 6 Маска ввода. Назначение и способы создания.
- 7 Назначение Мастера подстановок.

### **Контрольные задания**

- 1 На основании выбранного варианта спроектировать базу данных по заданной тематике.
- 2 Предусмотреть наличие раскрывающихся списков, значений по умолчанию и ввода по маске.

<b>№ варианта</b>	<b>Наименование базы данных</b>	<b>Наименование полей</b>
1	Ведомость для начисления заработной платы	1 Цех (список) 2 Ф.И.О. 3 Больничные дни (значение по умолчанию – 0) 4 Начисляемый заработок 5 Удержано 6 К выплате 7 Дата выдачи (маска ввода)
2	Библиотечный каталог	1 Шифр ББК (маска ввода) 2 Ф.И.О. автора 3 Наименование 4 Жанр (список) 5 Количество (значение по умолчанию – 1) 6 Стоимость
3	Железнодорожное расписание	1 № поезда 2 Пункт отправления (значение по умолчанию – Гомель) 3 Пункт назначения 4 Время отправления (маска ввода) 5 Время прибытия (маска ввода) 6 Категория (список) 7 Время в пути
4	Протокол соревнований	1 Место проведения (значение по умолчанию – манеж «Динамо») 2 Дата проведения (маска ввода) 3 Ф.И.О. участника 4 Год рождения 5 Вид состязания (список) 6 Результат 7 Занятое место (список)

№ варианта	Наименование базы данных	Наименование полей
5	Справочник автоинспектора	1 Марка автомобиля (список) 2 Год выпуска 3 Цвет (список) 4 Дата регистрации (маска ввода) 5 Пробег в километрах (значение по умолчанию – 0)
6	Сведения о поставках	1 Дата поставки (маска ввода) 2 Категория товара (список) 3 Наименование товара 4 Объем поставки (значение по умолчанию – 0) 5 Цена единицы 6 Общая стоимость
7	Телефонный справочник	1 Ф.И.О. абонента 2 Район (список) 3 Адрес 4 № телефона (маска ввода) 5 Дата установки (маска ввода) 6 Категория (значение по умолчанию – отдельный)
8	Дневник метеонаблюдений	1 Дата (маска ввода) 2 Пора года (список) 3 Температура (значение по умолчанию – 0) 4 Давление (значение по умолчанию – 0 мм рт.ст.) 5 Направление ветра (список) 6 Осадки (список)
9	Отдел кадров	1 Табельный номер (счетчик) 2 Ф.И.О. сотрудника 3 Отдел (список) 4 Должность (список) 5 Дата поступления на работу (маска ввода) 6 Стаж работы (значение по умолчанию – 1)
10	Карточка абитуриента	1 Ф.И.О. 2 Дата подачи документов (маска ввода) 3 Отделение (список) 4 Специальность (список) 5 Оценка на экзамене по русс. яз. (значение по умолчанию – 2) 6 Оценка на экзамене по математике (значение по умолчанию – 2) 7 Средний балл

## 6 СОЗДАНИЕ И КОРРЕКТИРОВКА БАЗЫ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

### 6.1 Способы создания таблиц

**Создание базы данных.** Содержание понятия «*файл базы данных*» в MS Access шире по сравнению с традиционным понятием. Это совокупность данных и объектов, которые относятся к конкретной задаче. Файл базы данных включает в себя объекты, т. е. таблицы, запросы, формы, отчеты, макросы, модули, процедуры; в MS Access 2000 к ним добавляется объект страницы доступа к данным.

В MS Access база данных представляет собой совокупность средств (таблицы, формы, запросы, отчеты) для ввода, хранения, просмотра, выборки и управления информацией.

Файл новой базы данных в MS Access создается в окне «Файл новой базы данных». В окно можно войти двумя способами:

1. В первом окне диалога, которое открывается при запуске MS Access, выбрать параметр «**Новая база данных**» и нажать кнопку «**ОК**». Откроется окно «**Файл новой базы данных**» (рисунок 6.1).

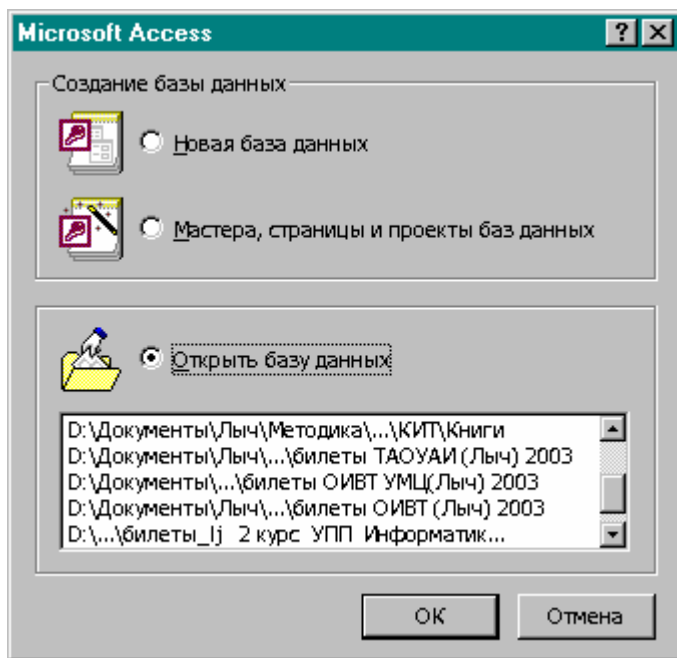


Рисунок 6.1 – Окно «Файл новой базы данных»

В этом окне представлен пустой файл базы данных с автоматически присвоенным именем.

2. В основном меню выбрать пункт «**Файл**» → «**Создать базу данных**» → [Создание] → выбрать вкладку «**Общие**» и нажать кнопку ОК. На экране появится окно «**Файл новой базы данных**».

В обоих случаях входа в окно «**Файл новой базы данных**» в поле «**Папка**» нужно выбрать имя папки, в которой будет сохраняться создаваемый файл базы данных, а в поле «**Имя файла**» ввести имя файла базы данных и нажать кнопку «**Создать**». На экране откроется окно нового файла «**База данных**» (рисунок 6.2).

Окно базы данных содержит шесть вкладок. В окне базы данных будем создавать все объекты, которые входят в базу данных. Названия объектов соответствуют ярлыкам вкладок: «**Таблицы**», «**Запросы**», «**Формы**», «**Отчеты**», «**Макросы**», «**Модули**».

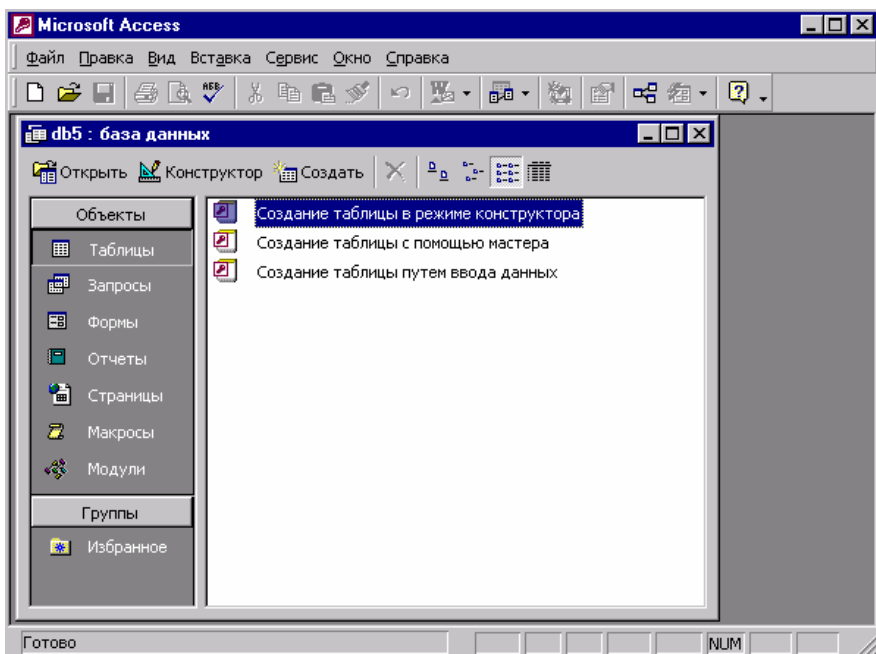


Рисунок 6.2 – Окно «База данных»

## 6.2 Создание таблиц

Начинать следует с определения структуры таблицы, соответствующей предметной области, т. е. с определения полей, которые надо включить в

таблицу, типов данных (полей), длины каждого поля, а также алгоритма обработки данных, удерживающихся в таблице.

Создание таблицы в MS Access осуществляется в окне «База данных».

Общий алгоритм создания таблицы: открыть окно создания базы данных и перейти на вкладку «Таблицы» → кнопка «Создать» → [«Новая таблица»] → выбрать один из пяти способов создания таблицы («Режим таблицы», «Конструктор», «Мастер таблиц», «Импорт таблиц», «Связь с таблицами») → «ОК» → [«Сохранение»] → задать имя созданной таблицы → «ОК».

### 6.2.1 Создание таблиц с помощью Мастера

MS Access содержит соответствующий перечень прототипов таблиц, которые можно использовать для создания собственных таблиц. При этом экономится время и обеспечиваются стандартные имена и типы данных полей таблицы.

Для создания таблицы с помощью Мастера нужно открыть окно базы данных → перейти на вкладку «Таблицы» → нажать кнопку «Создать» → «Новая таблица» → Выбрать опцию «Мастер таблиц» → «ОК» (рисунок 6.3). Мастер таблиц создает новую таблицу в диалоге с переходом от одного шага к другому (в обоих направлениях) на основе существующих шаблонов таблиц.

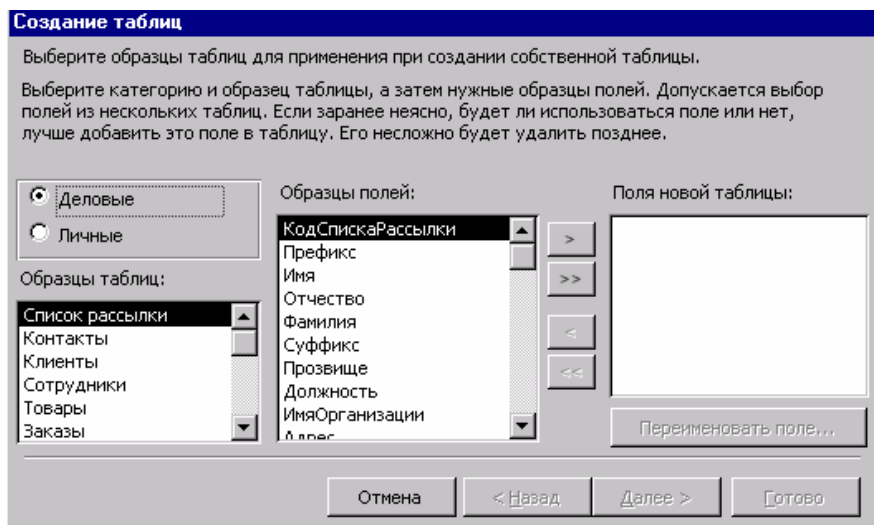


Рисунок 6.3 – Окно Мастера для формирования таблицы

Окно диалога «Создание таблиц» содержит три списка: «Образцы таблиц», «Образцы полей», «Поля новой таблицы». При выборе в окне «Образ-

цы таблиц» каждого из прототипов таблицы меняется содержание окна «Образцы полей».

Дальнейшие действия: выбрать из списка «Образцы таблиц» подходящий прототип таблицы, а из списка «Образцы полей» – нужные нам поля и перенести их в список «Поля новой таблицы». Для выбора полей предназначены кнопки навигации со стрелками, которые размещены между списками «Образцы полей» и «Поля новой таблицы» (таблица 6.1).

Т а б л и ц а 6.1 – Функции кнопок выбора полей

Кнопки	Назначение
>	Выделенное поле в списке «Образцы полей» добавляется в список «Поля новой таблицы».
>>	Все поля из списка «Образцы полей» добавляются в список «Поля новой таблицы».
<	Из списка полей созданной таблицы удаляется поле, выделенное в списке «Поля новой таблицы».
<<	Удаляются все поля из списка «Поля новой таблицы».

При необходимости можно изменить наименование любого выбранного поля из списка «Поля новой таблицы» при помощи кнопки «Переименовать поле».

После завершения формирования списка полей создаваемой таблицы нужно присвоить имя таблице, нажав кнопку «Далее»; при этом Мастер предлагает свой вариант имени таблицы, который можно принять, нажав клавишу «Tab», или отвергнуть, введя свое имя.

Можно выбрать способ обозначения ключа; для перехода к следующему окну следует нажать кнопку «Далее». Создаваемую таблицу Мастер помогает связать (если нужно) с существующими таблицами: для этого следует выбрать из списка таблицу с базой данных, с которой надо ее связать, нажать кнопку «Связь», установить один из двух возможных типов связей и нажать кнопку «ОК».

Тем самым предоставляется вся необходимая Мастеру информация для создания таблицы. Система предлагает возможные действия (изменение структуры таблицы, непосредственный ввод данных в таблицу, ввод данных в таблицу с помощью формы). После их выбора надо нажать кнопку «Готово» →осуществляется запуск Мастера на формирование и вывод таблицы.

### 6.2.2 Проектирование таблицы в окне Конструктора

Пользователь должен представлять себе (иметь на бумаге) структуру проектируемой таблицы, после чего, используя программу «Конструктор» в качестве инструментария, создать ее.

Последовательность действий при использовании программы «**Конструктор**»:



1 Сделать активным окно «База данных» (см. рисунок 6.2) при помощи команды «Файл» → «Создать» для создания новой базы данных или команды «Файл» → «Открыть» при открытии существующей базы данных.

2 Выбрать вкладку «Таблицы» в окне «База данных».

3 Нажать кнопку «Создать» → [«Создание таблиц»].

4 Выбрать способ создания таблицы: «Конструктор».

5 Нажать кнопку «ОК». На экране откроется окно Конструктора таблиц с мигающим курсором в первом ряду графы «Имя поля» (рисунок 6.4). В верхней части окна находятся в виде таблицы компоненты для описания логической структуры создаваемой таблицы: «Имя поля», «Тип данных», «Описание».

6 В первую строку верхней части окна Конструктора ввести имя первого поля, его тип и (при необходимости) описание этого поля. После набора имени поля на клавиатуре и нажатия клавиши «Enter» курсор переместится в графу «Тип данных».

**Имена полей.** Предназначены для ссылки на данные таблицы. Правила присвоения имен в Access: не более 64 символов. В одной таблице поля не могут иметь одинаковое имя; оно не должно начинаться с пробела; запрещены точка (.), восклицательный знак (!), прямые скобки [ ] и управляющие символы с кодами ASCII 1–31.

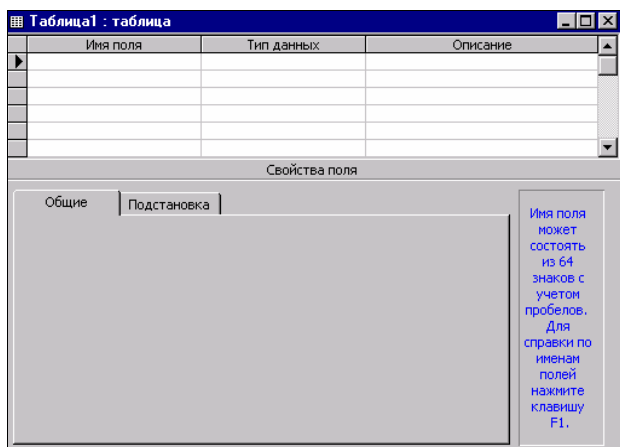


Рисунок 6.4 – Окно Конструктора таблиц

Рекомендуется при присвоении полям имен применять «говорящую» аббревиатуру и короткие названия. После ввода имени поля для перехода в графу «Тип данных» нужно нажать клавишу «Tab» или «Enter». В этой графе автоматически появятся тип «Текстовый», а также кнопка для открытия списка типов полей T, а в области «Свойства поля» (рисунок 6.5) — перечисление свойств со значениями по умолчанию.

Для смены типа данных нужно нажать кнопку раскрытия списка типов полей и выбрать необходимый тип. Ввод типов данных заканчивается нажатием клавиши «Enter». Курсор автоматически переместится в графу «Описание» (ее можно не заполнять).

**Тип данных.** В графе «Тип данных» кнопкой ▼ раскрываем список и выбираем нужное значение. В Access допускаются следующие типы данных: текстовый, числовой, денежный, счетчик, дата/время, поле MEMO, поле объекта OLE, мастер подстановок.

**Описание.** В графе «Описание» даются пояснения полям. При работе с таблицей пояснения будут появляться в нижней части экрана (строка состояния), как только мы окажемся в этом поле, и напоминать о значении поля.

7 Для связи таблиц в базе данных нужно определить поле-ключ (значения ключа в таблице не повторяются, а значит, однозначно определяют каждую запись): нажать нужное поле, затем кнопку «Ключевое поле», на панели инструментов или в контекстном меню выбрать команду «Ключевое поле». Слева от имени поля в области выделенного ряда появится символ ключа.

8 Для перехода к описанию следующего поля нажать клавишу «Tab». Описать все поля таблицы и сохранить ее командой «Файл» → «Сохранить».

9 В окне «Сохранение» ввести имя создаваемой таблицы и нажать кнопку «ОК».

10 На запрос системы о создании ключевого поля ответить нажатием кнопки «Нет».

11 Закрыть окно Конструктора таблиц и возвратиться в окно базы данных, где появится имя таблицы.

12 В нижней части окна Конструктора «Свойства поля» каждое поле таблицы характеризуется дополнительными свойствами, которые определяют условия ввода данных (рисунок 6.5).

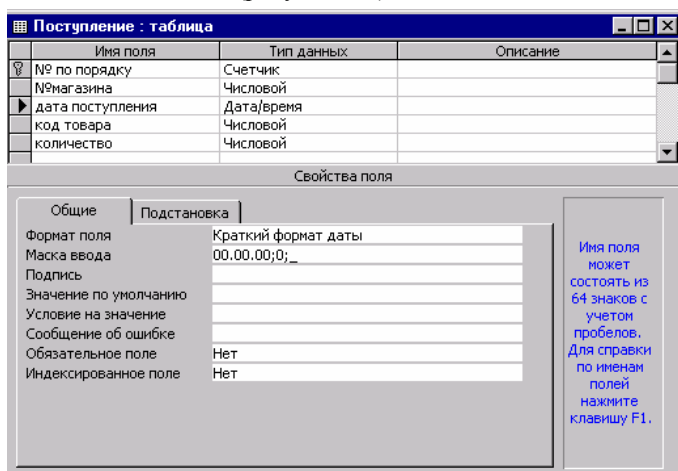


Рисунок 6.5 – Свойства полей таблицы



рейти в режим Конструктора таблиц при помощи команды «Вид» → «Конструктор таблиц» или нажатия кнопки на панели инструментов «Представление таблицы».

7 Сохранить созданную таблицу командой «Файл» → «Сохранить макет» или нажатием кнопки «Сохранить» на панели инструментов. В окне «Сохранение» ввести имя таблицы и нажать кнопку «ОК».

8 На запрос о создании ключевого поля ответить «Да» при желании, чтобы это поле автоматически задавала система с типом «Счетчик». В таблице появится первое поле с именем «Код», а его значение вводится автоматически в виде натурального ряда чисел с увеличением на единицу. Если же ключевое поле устанавливается в режиме Конструктора, то на вопрос о создании ключевого поля нужно нажать кнопку «Нет».

9 При необходимости сделать корректировку типов и свойств полей в режиме Конструктора.

10 Для ввода данных в таблицу вернуться в режим таблицы при помощи команды «Вид» → «Режим таблицы» или нажатия кнопки «Представление таблицы» на панели инструментов.

**Ввод данных в таблицу.** В окне базы данных на вкладке «Таблицы» находятся имена созданных таблиц. Для ввода в них информации нужно, прежде всего, открыть таблицу. Ее имя может находиться в нижней части меню пункта «Файл»; в противном случае подается команда «Файл» → «Открыть» → [«Открытие файла базы данных»], после чего следует выбрать нужную базу данных и открыть ее; открыть таблицу можно и в диалоговом окне базы данных: нажав вкладку «Таблицы», увидим список созданных таблиц; дважды нажать имя нужной таблицы или выбрать его и нажать кнопку «Открыть» в окне базы данных. На экране откроется таблица в режиме таблицы, после чего можно приступить к вводу информации.

Если первое поле в таблице описано с типом данных «Счетчик» или MS Access создаст его в качестве ключевого поля, то вводить информацию в это поле не надо. Его содержимое автоматически увеличивается на единицу в каждой следующей записи. Для перехода к вводу следующего поля надо нажать клавишу «Tab» или «Enter». Для возвращения в пропущенное поле предназначена клавиша «Shift», для пропуска пустых граф – клавиша «Tab».

После ввода значения последнего поля нажимается клавиша «Tab», и курсор переходит в начало следующей записи.

В области выбора записи (слева от левого поля) находятся пиктограммы, которые меняются при изменении вида записи:

- *черный треугольник* – область выбора следующей записи; показывает, на какой записи находимся в настоящий момент;
- *символ «карандаш»* свидетельствует о том, что осуществляется ввод информации или ее редактирование и что внесенные изменения еще не сохранились;
- *звездочка* появляется в конце новой записи.

Для перемещения данных внутри таблицы их предварительно выделяют (соответствующие действия приведены в таблице 6.2).

В процессе ввода информации в таблицу MS Access после нажатия клавиши «Tab» в последнем поле записи и переходе к новой записи автоматически записывается (сохраняется) предшествующая запись на диск.

**Т а б л и ц а 6.2 – Выделение полей и записей**

Выделяемая область	Место установки указателя и нажатия кнопки мыши
Все записи таблицы	Командой «Правка» →«Выделить все»
Все поле	В левую часть поля; указатель принимает вид белого крестика
Несколько символов в поле	В начале позиции выделения; при нажатой кнопки мыши перемещать указатель для расширения области
Несколько совместных полей	В левую часть поля; указатель принимает вид белого крестика
Запись	В области выделения записи
Несколько записей	В области выделения первой выделяемой записи
Столбец	В области заголовка столбца
Несколько совместных столбцов	В области заголовка первого выделяемого столбца

**Перемещение по таблице.** Осуществляется при помощи мыши. Для быстрого перемещения имеются клавишные команды (таблица 6.3).

**Т а б л и ц а 6.3 – Клавишные команды для быстрого перемещения по таблице**

Клавишная команда	Перемещение на ...
[Tab] или [Enter] или →	Следующее поле
[Shift-Tab]	Предыдущее поле
↓	Следующая запись
↓	Предыдущая запись
[Home]	Первое поле бегущей записи
[End]	Последнее поле бегущей записи
[Ctrl-Home]	Первое поле первой записи
[Ctrl-End]	Последнее поле последней записи
[Page Down]	Следующую страницу (на 26 рядов)
[Page Up]	Предыдущую страницу

### 6.3 Просмотр и редактирование содержания полей

Просмотр информации в длинном поле. Текст, который выходит за границы ширины столбца и не виден полностью, можно просмотреть целиком. Для этого следует:

- установить курсор в нужное поле;
- нажать клавиши «Shift-F2». Откроется диалоговое окно «Область ввода», в котором содержание поля приведено полностью.

**Вставка в поле значения этого же поля в предыдущей записи при вводе информации в таблицу:**

- 1) в режиме таблицы выбрать поле, в которое надо вставить значение;
- 2) клавишная команда «Ctrl-« (апостроф)». В выбранное поле будет введено значение этого же поля предыдущей записи.

Изменение отдельных символов в поле: ошибочно введенный символ в поле удалить клавишей «Backspace» или выделить текст мышью и нажать клавишу «Delete».

**Удаление всего значения поля и замена новым значением:**

- 1) курсор поставить в крайнюю левую часть поля, из которого нужно удалить информацию (курсор примет вид белого крестика);
- 2) нажать на левую кнопку мыши, после чего можно ввести новую информацию.

**Вставка нового поля в таблицу:**

- 1) выделить поле (столбец), перед которым вставляется новое поле;
- 2) пункт «Вставка» → «Столбец». Во вставленное поле ввести имя поля.

**Отмена произведенных действий:** порядок приведен в таблице 6.4.

Таблица 6.4 – Отмена произведенных действий

Что отменить	Клавиши и команды
Только что введенные данные	Нажать клавишу «Esc»
Произведенное действие	На панели инструментов нажать кнопку «Отменить» или дать команду «Правка»→ «Отменить удаление»
Всю информацию, только что введенную в запись	Команда «Правка»→«Удалить запись»
Всю информацию, только что сохраненную в записи	Клавишная команда «Ctrl-Z»

### 6.4 Редактирование, поиск, замена данных

**Перемещение данных внутри таблицы.** Применяются стандартные средства Windows: удаление в буфер, копирование в буфер и вставка из буфера.

**Выделение полей и записей.** Для перемещения данных внутри таблицы (нескольких символов в поле, всего поля, нескольких соседних полей,

столбца, нескольких соседних столбцов, запись, несколько записей, все записи в таблице) их необходимо предварительно выделить (см. таблицу 6.2). Команда «Правка» → «Выделить все записи».

### **Перемещение данных из одного поля в другое:**

1) дважды щелкнуть мышью на поле, из которого будут перенесены данные, или установить курсор в крайнюю левую часть поля и, как только он примет форму белого крестика, нажать на кнопку мыши для выделения текста всего поля;

2) команда «Правка» → «Вырезать» или нажать кнопку «Удалить в буфер» на панели инструментов;

3) нажать кнопку мыши в любом месте поля, в которое переносится информация, и дать команду «Правка» → «Вставить» или нажать на кнопку «Вставить из буфера».

Копирование информация из нескольких полей (копирование данных из нескольких рядом стоящих полей в то же самое количество полей):

1) выбрать поля для копирования, переместить указатель мыши на выбранные поля, начиная от левого края первого поля, где появится белый крестик;

2) скопировать выбранные поля в буфер одним из известных способов;

3) выделить поля, в которые следует поместить данные;

4) вставить содержание буфера одним из известных способов.

**Изменение последовательности полей в таблице** (без выхода из режима таблицы):

1) поместить указатель мыши на заголовок столбца, который будет перемещаться; после появления черной стрелки нажать кнопку мыши для выделения столбца;

2) установить указатель в середину имени поля (не приближаться близко к границе); нажать кнопку мыши и сохранять ее в таком положении; появится белая стрелка с небольшим прямоугольником;

3) переместить поле в нужное место – появится широкая черная вертикальная черта, которая движется между полями при перемещении столбца и указывает на левое место границы поля после отпускания кнопки мыши.

**Изменение вида таблицы.** Можно сделать столбцы и строки шире или уже, сделать столбцы прозрачными или непрозрачными.

### **Расширение столбцов.** *Первый способ:*

1) установить указатель мыши на правую границу расширяемого столбца (примет вид двунаправленной стрелки);

2) нажать кнопку мыши и сохранять ее в таком положении — появится вертикальная черта, которая показывает границу поля после того, как будет отпущена кнопка мыши;

3) перемещать линию вправо до нужной ширины столбца.

Аналогичным образом можно сузить столбец.

*Второй способ:*

- 1) выделить столбец;
- 2) команда **«Формат»** → **«Ширина столбца»**;
- 3) в появившемся окне нажать кнопку **«По ширине данных»** и кнопку **«ОК»**.

**Изменение количества отображаемых столбцов.** Таблица по ширине может не уместиться на экране. В такой ситуации можно оставить на экране лишь необходимые столбцы, а остальные скрыть: выделить столбец, который хотим скрыть; команда **«Формат»** → **«Скрыть столбцы»** (или в контекстном меню **«Ширина столбца»**, **«Скрыть столбцы»**). Скрытые столбцы не удаляются, и информация в них сохраняется.

*Вывод на экран скрытых столбцов:*

- 1) команда **«Формат»** → **«Показать столбцы»**. В диалоговом окне **«Отображение столбцов»** флажками помечены столбцы, которые в данный момент видны; скрытые столбцы не помечены;
- 2) установить флажок около скрытых столбцов;
- 3) закрыть окно.

**Фиксирование столбцов таблицы на экране** (если по ширине таблица не вмещается на экране) при переходе в правую часть таблицы:

- 1) выделить столбец таблицы, который должен оставаться на экране неподвижным при перемещении в правую часть таблицы;
- 2) команда **«Формат»** → **«Закрепить столбцы»**. Закрепленный столбец займет крайнее левое положение в таблице. После этого при перемещении по таблице вправо закрепленные столбцы останутся на экране;
- 3) для отмены закрепления столбцов предназначена команда **«Формат»** → **«Освободить все столбцы»**.

**Изменение параметров шрифта:**

- 1) команда **«Формат»** → **«Шрифт»**; откроется окно диалога **«Выбор шрифтов»**;
- 2) выбрать нужный шрифт и нажать кнопку **«ОК»**.

**Удаление записей из таблицы.** Можно удалить одну, несколько и все записи из таблицы. Для удаления одной или нескольких записей нужно:

- 1) выделить одну или несколько записей, которые нужно удалить;
  - 2) нажать клавишу **«Delete»** или дать команду **«Правка»** → **«Удалить»**;
- на запрос о подтверждении удаления ответить нажатием клавиши **«ОК»**.

Для удаления всей информации из таблицы: **«Правка»** → **«Удалить все записи»** → **«Delete»** → **«ОК»**.

## 6.5 Сортировка данных

Данные можно сортировать по любому полю.

*Первый способ:*

- 1) выделить столбец (поле) для сортировки;



2) дать команду «**Записи**» → «**Сортировка**», выбрать порядок сортирования: «**По возрастанию**» или «**По убыванию**».

*Второй способ сортировки данных* – с помощью кнопок панели инструментов «Сортировка по возрастанию» и «Сортировка по убыванию».

*Третий способ сортировки:* в контекстном меню выбрать команду «Сортировка по возрастанию» или «Сортировка по убыванию».

**Сортировка двух полей одновременно.** Поля, по которым будем сортировать данные, должны размещаться справа налево по возрастанию ключа; необходимо выполнять условие подчиненности, например:

« <b>Группа</b> »	« <b>Фамилия</b> »
<b>ПГ</b>	<b>ЧГ</b>

Если поля не размещены в таком порядке (например, ЧГ – «**Фамилия**», ПГ – «**Группа**»), то нужно:

1) снять фиксацию со всех полей командой «**Формат**»→«**Освободить все столбцы**»;

2) переместить (мышь на область маркировки поля) поле высшей группы (например, «**Группа**») налево от поля низшей группы («**Фамилия**»);

3) перемещаемое поле оставить выделенным (оба поля-ключа должны быть выделены одновременно); для выделения второго поля нажать и сохранять в таком положении клавишу «Shift»; нажать мышью на область выбора второго поля;

4) нажать кнопку, соответствующую сортировке, – MS Access рассортирует записи по двум полям;

5) закрыть таблицу и на запрос, нужно ли сохранить изменения, ответить «**Нет**».

## 6.6 Фильтры

Предназначены для отбора только тех записей, которые соответствуют заданному значению критерия. Фильтр можно создать (с помощью кнопок панели инструментов и команд меню) в любом режиме работы с таблицей, которая должна быть открытой.

**Фильтрация данных по выделенной информации.** Критерий фильтрации задается путем выделения информации в таблице данных. Результатом этой процедуры является таблица с записями, отобранными по выделенным значениям поля из исходной таблицы. Последовательность действий:

1) открыть таблицу-источник;

2) в соответствующем поле, например «**Код товара**», выбрать нужный код (его значение);

3) дать команду «**Запись**» → «**Фильтр**» → «**Фильтр по выделенному**» или нажать кнопку «**Фильтр по выделенному**».

Выбранная по заданному критерию информация выводится на экран.

**Фильтрация путем исключения записей по критерию:** на экран выводятся только те записи, которые не соответствуют заданному критерию. Для этого нужно открыть таблицу данных; задать критерий выбора значений; открыть контекстное меню на выбранном месте; выбрать команду **«Исключить выделенное»**.

Команда **«Изменить фильтр»** открывает диалоговое окно с вкладками для задания критерия фильтрации и подается кнопкой **«Изменить фильтр»**; можно также дать команду **«Запись»**→ **«Фильтр»**→ **«Изменить фильтр»** →[Фильтр].

В диалоговом окне установки фильтра размещены все поля таблицы, а также ярлыки (в нижней части окна) **«Найти»** и **«Или»**.

При установке указателя мыши на любое поле таблицы появляется кнопка раскрытия списка, при помощи которой открываются все значения этого поля.

Последовательность действий:

- 1) курсор мыши – в поле выбора;
- 2) раскрыть список кнопкой и выбрать из него нужное значение;
- 3) на панели инструментов нажать кнопку **«Применить фильтр»**.

В команде **«Изменить фильтр»** можно устанавливать критерии **«И»** и **«Или»** и применить выражение.

**Расширенный фильтр.** Позволяет выполнять не только фильтрацию, но и одновременную сортировку по нескольким полям. В качестве критерия отбора можно задавать любые шаблоны. Последовательность действий:

- 1) открыть таблицу-источник, которую следует профильтровать;
- 2) дать команду **«Запись»**→ **«Фильтр»** → **«Расширенный фильтр»** → открывается окно диалога **«Фильтр»**, в верхней части которого находится список полей таблицы, а в нижней — бланк для записи критерия отбора (выражение для фильтра);
- 3) из верхней части окна перетянуть мышью в нижнюю часть окна в строку **«Поле»** имя поля, по которому будет задан критерий отбора;
- 4) в строке **«Сортировка»** из раскрытого списка выбрать тип сортировки (при необходимости производить эту операцию);
- 5) в строке **«Условие отбора»** ввести критерий отбора;
- 6) если фильтрация производится по нескольким полям, то надо повторить для каждого поля действия 3–5;
- 7) после задания значения фильтра на панели инструментов нажать кнопку **«Применить фильтр»**.

**Сохранение фильтра.** По команде **«Файл»** →**«Сохранить макет»** в режиме просмотра созданный фильтр сохраняется вместе с таблицей.

При открытии таблицы сохраненный фильтр будет действовать в процессе работы с таблицей.

## 6.7 Поиск информации

**Поиск.** Для поиска конкретной записи в базе данных предназначена команда **«Правка»** → **«Найти»**. Последовательность действий:

- 1) открыть таблицу, в которой будет происходить поиск записи;
- 2) активизировать поле поиска;
- 3) ввести команду **«Правка»** → **«Найти»** → [Найти];
- 4) в поле **«Что»** ввести объект поиска;
- 5) задать способы поиска (любая часть поля; все поле; начало строки;

искать только по бегущему полю и др.).

**Поиск и замена.** Время от времени возникает необходимость найти записи в базе данных по заданному значению и заменить его новым. Предположим, в базе данных «Города и поселки Беларуси» нужно во всех населенных пунктах изменить название улицы Ленина на улицу Независимости. Это достигается автоматически по команде **«Правка»** → **«Заменить»** → [Замена в поле]. В строку ввода **«Что»** следует ввести то, что нужно заменить (в данном случае — «Ленина»). В поле **«Заменить на»** ввести новое значение («Независимости»). Затем нажать кнопку **«Поиск далее»**. В результате этих действий Access найдет и маркирует эти записи для замены, но сама замена осуществится лишь после нажатия кнопки **«Заменить»**.

## 6.8 Изменение структуры и вида таблицы

Созданную структуру таблицы можно модифицировать: исправить ошибки или внести изменения. Для модификации структуры таблицы базы данных надо установить в окне базы данных указатель мыши на модифицируемую таблицу и нажать кнопку **«Конструктор»**. Открывается окно Конструктора с отражением структуры выбранной таблицы.

В процессе модификации можно:

- 1) изменить имя поля (удалить старое и ввести новое);
- 2) изменить тип данных поля (в окне со списком «Тип поля» выбрать нужный тип);
- 3) изменить размер поля (в нижней части экрана в строке ввода **«Размер поля»** удалить бегущий размер поля и ввести с помощью клавиатуры новый);
- 4) добавить новое поле (имена полей перечисляются в столбце); нажать имя поля, над которым будет вставлено новое поле; затем нажать клавишу **«Insert»** или дать команду **«Правка»** → **«Вставка строки»**; описать новое поле: имя, тип и т. д. (можно через контекстное меню);
- 5) удалить поле (нажать на имя этого поля; команда **«Правка»** → **«Удалить строку»**);
- 6) изменить последовательность полей в структуре (перенести поле на другое место — выше или ниже). Для этого:

- выделить поле нажатием на область выбора поля (слева от имени поля) — вся строка будет выделена;
- еще раз нажать на кнопку мыши и не отпускать ее; над курсором появится маленький пунктирный прямоугольник;
- перенести всю строку, чтобы она оказалась прямо под тем полем, перед которым следует ее поместить;
- отпустить кнопку мыши. Поле будет находиться на новом месте.

#### **Удаление ненужной таблицы из базы данных:**

- 1) в поле базы данных выбрать имя таблицы;
- 2) дать команду меню «Правка»→«Удалить» →[.] →«ОК». MS Access удалит таблицу.

**Индексы** – средство для ускорения поиска и сортировки записей в базе данных. Задаются ключевые поля, которые индексируются автоматически. Например, для поиска записей по *полю* «Табельный номер» в соответствующей таблице нужно создать индекс для этого поля.

**Ключ**, как уже упоминалось, однозначно определяет каждую запись в таблице. Повтор значения ключа в одной таблице недопустим. Ключ подтверждает полномочия на доступ к данным, используется для идентификации элементов множества.

В межтабличных связях ключевые поля применяются для ссылок на указанные записи в обеих таблицах. При ссылке на ключевое поле с другой таблицы оно называется *полем внешнего ключа*.

Ключевое **поле** – это одно или несколько полей, комбинация значений которых однозначно идентифицирует каждую запись в таблице. Ключевые поля применяются для ссылок в межтабличных связях.

**Первичный ключ** удерживает информацию, которая однозначно идентифицирует запись уникальным значением индексного поля. Для первичного ключа выбирается поле таблицы, в котором для каждой записи удерживается уникальная информация.

#### **Создание ключевых полей:**

- 1) открыть таблицу в режиме Конструктора;
- 2) выделить одно или несколько полей, которые определяются в качестве ключей. Для выделения одного поля выбрать область выделения строки нужного поля (слева от имени поля);
- 3) нажать кнопку «**Ключевое поле**», находящуюся на панели инструментов. В области выбора поля появится пиктограмма с изображением ключа.

*Замечание.* Первичный ключ можно установить и другим способом: выделить поле для первичного ключа; в контекстном меню выбрать команду «Первоначальный ключ».

#### **Удаление ключевых полей:**

- 1) удалить межтабличную связь, если связано ключевое поле;
- 2) открыть таблицу в режиме Конструктора;

3) для текущего ключевого поля выбрать область выделения строки и на панели управления нажать кнопку «Ключевое поле».

## 6.9 Связи между таблицами

MS Access устанавливает постоянные взаимосвязи между таблицами, участвующими при создании форм, запросов и отчетов (см. п. 2.6). Поля, с помощью которых устанавливаются постоянные взаимосвязи между таблицами, могут иметь разные имена, однако лучше использовать совпадающие имена. Связь между таблицами осуществляется на основе значений совпадающих полей (поле-ключ, например «КодТовара»).

**Определение отношений «много-ко-многим» между таблицами:**

1) исходная позиция: две таблицы с отношениями «много-ко-многим». Например, информация оптовой базы удерживается в двух таблицах — в таблице **«Поставщики»** (имеет следующую структуру: КодПоставщика, КодТовара, Стоимость, СпособДоставки, СрокПоставки) и таблице **«Потребители»** (имеет следующую структуру: КодПотребителя, КодТовара, Стоимость, Количество, СрокПоставки);

2) создать третью, связывающую, таблицу, добавив в нее поля, описание которых совпадает с описанием ключевых полей (для ссылок на указанные записи в одной таблице из другой таблицы) в обеих таблицах. В нашем примере это «КодТовара». В третьей таблице ключевые поля являются внешними ключами, потому что определяют способ связывания таблиц. Другие поля в таблицу связи добавляются без ограничений;

3) в таблице связей определить ключ, который удерживает все ключевые поля двух связываемых таблиц. Связь между таблицами устанавливается на основе значений в совпадающих полях (в нашем примере «КодТовара»).

Между связывающей таблицей и каждой из двух таблиц определяются отношения «один-ко-многим».

**Алгоритм определения связей между таблицами:**

1) все открытые таблицы закрыть;  
2) переключиться в окно базы данных;  
3) на панели инструментов нажать кнопку **«Схема данных»**;  
4) если в базе данных не определены связи, то на экране появится окно **«Добавление таблицы»**. Если на экране этого окна нет, а связи нужно определить, то на панели инструментов следует нажать кнопку **«Добавить таблицу»**. Если связываемые таблицы отображены на экране, то переходим к следующим пунктам;

5) дважды щелкнуть мышью по именам таблиц, для которых необходимо установить связи. Закрыть окно диалога **«Добавление таблиц»**;

6) установить связь между полями путем выбора поля в одной таблице и «перетягивания» его на соответствующее поле в другую таблицу. Связыва-

ваемые поля должны быть одинакового типа данных и иметь содержание одного типа (имена могут не совпадать);

7) в окне диалога **«Схема данных»** проверить имена, которые представлены в двух колонках. При необходимости внести изменения;

8) для создания связей нажать кнопку **«Создать»**;

9) для каждой пары таблиц, которые необходимо связать, выполнить действия, указанные в п. 5-8.

*Просмотр всех связей в базе данных:* на панели инструментов нажать кнопку **«Показать все связи»**.

*Просмотр связей только для определенной таблицы:* на панели инструментов нажать кнопку **«Показать прямые связи»**.

*Внесение изменений в структуру таблицы:* в контекстном меню выбрать команду **«Конструктор таблиц»**.

*Связывание таблицы по принципу «сама с собой» или полей таблицы с другим полем этой же таблицы:* дважды добавить таблицы.

#### **Удаление связей между таблицами:**

1) закрыть все таблицы (в открытых таблицах удаление невозможно);

2) перейти в окно базы данных. Для переключения с одного окна в окно базы данных нажать клавишу **«F11»**;

3) на панели инструментов нажать кнопку **«Схема данных»**;

4) если на экране не отображены таблицы со связями для удаления, то на панели инструментов нажать кнопку **«Добавить таблицу»** и дважды щелкнуть мышью по имени каждой добавляемой таблицы; нажать кнопку **«Закрыть»**.

5) выбрать линию связи для удаления и нажать клавишу **«Del»**.

**Пример связывания таблиц.** Предположим, даны две таблицы: таблица с именем **«Ценник»** (структура: ГруппаТоваров, КодТовара, Наименование, Стоимость) и таблица с именем **«Поставки»** (структура: Поставщик; КодПоставщика, ГруппаТовара, КодТовара, Дата, Количество).

Поле **«КодТовара»** в таблице **«Ценник»** является первичным ключом, а в таблице **«Поставки»** – внешним ключом.

Тип взаимосвязей между приведенными таблицами – **«один-ко-многим»**. В этих взаимосвязях таблица, которая удерживает первичный ключ и составляет часть **«один»** в отношениях **«один-ко-многим»**, называется *главной таблицей*. Таблица, которая удерживает такой же тип информации со стороны **«много»** в отношениях **«один-ко-многим»**, называется подчиненной.

Каждой записи в таблице **«Ценник»** может соответствовать несколько записей в таблице **«Поставки»**. В свою очередь, запись в таблице **«Поставки»** не может иметь больше одной соответствующей ей записи в таблице **«Ценник»**. Каждый товар имеет одну стоимость, а поступлений (количества) может быть много.

Обязательное условие связи двух таблиц – наличие совпадающих полей. Ключевое поле главной таблицы должно соответствовать аналогичному полю подчиненной (связанной) таблицы.

При наличии отношений между таблицами MS Access будет автоматически выбирать связанные данные из таблиц в запросах, отчетах и формах.

Связи между таблицами создаются в окне диалога «Схема данных» в такой последовательности:

1) в окне «База данных» командой «Сервис» → «Схема данных» или нажатием кнопки на панели инструментов «Схема данных» открываем окно диалога «Схема данных» (рисунок 6.7);

2) в окно «Схема данных» помещены таблицы, между которыми надо установить связи. Для этого следует ввести команду «Связи» → «Добавить таблицу»; откроется окно «Добавление таблицы». В этом окне в списке таблиц выделить первую добавляемую таблицу (в нашем примере это «Ценник») и нажать кнопку «Добавить». Рядом с окном «Добавление таблицы» появится окно со структурой таблицы «Ценник». Затем выбрать другую добавляемую таблицу (в нашем примере «Поставки») и нажать кнопку «Добавить»;

3) после этого закрыть окно диалога «Добавление таблицы» кнопкой «Закрыть». В окне диалога «Схема данных» появятся две таблицы, которые нужно связать (см. рисунок 6.7);

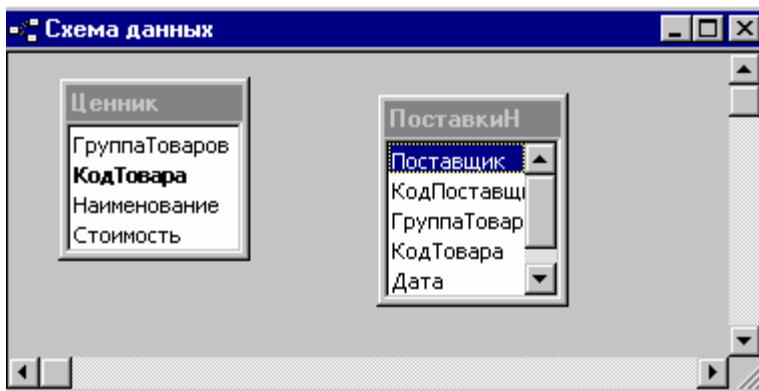


Рисунок 6.7 – Окно «Схема данных» с таблицами для связей

4) создать связи между таблицами: выбрать поле в первой, связываемой таблице «Ценник» (например, «КодТовара»), и переместить его мышью на соответствующее поле («КодТовара») второй таблицы «Поставки». Откроется окно «Связи» (рисунок 6.8).

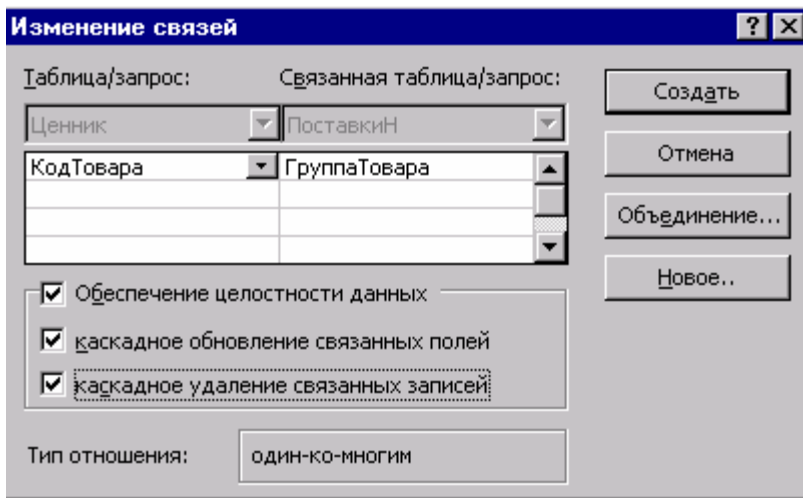


Рисунок 6.8 – Окно «Связи»

*Замечание.* Связываемые поля необязательно должны иметь одинаковые **имена**, но должны быть одного типа данных, а поля типа **«Числовой»** должны иметь одинаковый **«Размер поля»**; для связи нескольких полей таблиц нужно нажать клавишу **«Ctrl»**;

5) в окне диалога **«Связи»** проверить правильность имен связываемых полей. При необходимости можно выбрать другие имена полей;

6) установить флажок **«Обеспечение целостности данных»** и флажок **«Каскадное обновление связанных полей»**.

**Целостность данных** обеспечивается правилами, которые гарантируют поддержание устойчивых межтабличных связей при вводе или удалении записей. Для поддержания связей между записями в связанных таблицах и для обеспечения защиты данных от случайного удаления или изменения MS Access имеет систему правил под названием **«Целостность данных»**.

Для обеспечения целостности данных должны выполняться *следующие условия*:

- связанные поля **должны быть одного** типа данных;
- связанное поле главной таблицы является ключевым полем или **имеет** уникальный **индекс** (для ключевых полей создается автоматически);
- обе таблицы принадлежат одной базе данных.

При работе с данными после установки их целостности необходимо пользоваться *следующими правилами*:

- в поле внешнего ключа связанной таблицы нельзя ввести значение, которое не удерживается в ключевом поле главной таблицы;



- из главной таблицы не допускается удаление записи, если в подчиненной таблице существуют связанные с ней записи;
- невозможно изменить значение ключевого поля в главной таблице, если существуют записи, которые с ней связаны.

Установка флажка **«Обеспечение целостности данных»** обеспечивает вывод на экран предупреждения и невыполнение действий, которые нарушают одно из перечисленных правил.

**Условия целостности данных.** Целостность данных (data integrity) — это правильность данных и их соответствие ожидаемым значениям после перемещения или обработки; проверка точности содержимого отдельных полей в таблицах, файлах; контроль за успешным и правильным обновлением базы данных при каждой транзакции. *Транзакция* — это запрос на изменение базы данных; входящее сообщение, которое переводит базу данных из одного непротиворечивого состояния в другое;

7) после проверки правильности имен или выборки других имен полей нажимается кнопка **«Создать»**. Происходит возвращение в окно **«Схема данных»**, где линией показана установленная связь между таблицами (рисунок 6.9).

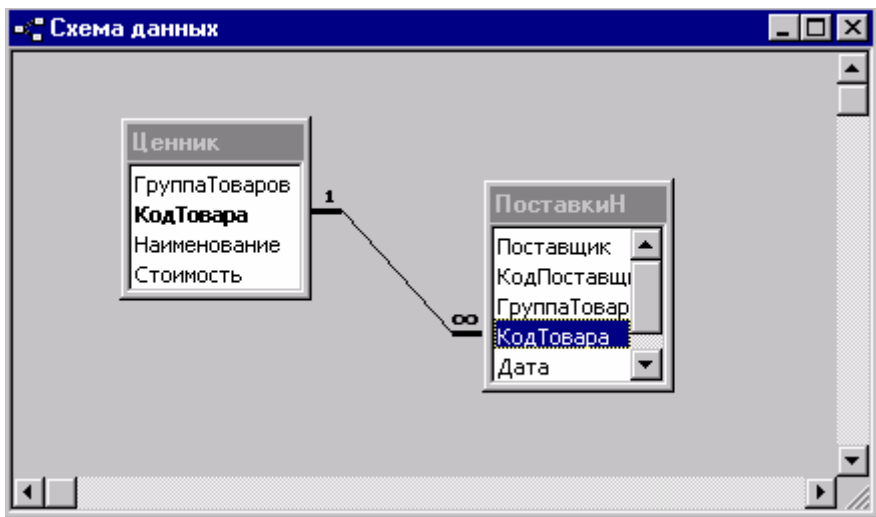


Рисунок 6.9 – Отображение установленной связи между таблицами

Для установки связей между другими полями нужно выполнить действия, указанные в пп. 4–7, и закрыть окно **«Схема данных»**.

В окне диалога **«Схема данных»** создаются связи между таблицами; можно также изменить структуру таблицы, изменить существующую связь, удалить связь, удалить таблицу с окна диалога **«Схема данных»**.

### **Изменение структуры таблицы в окне диалога «Схема данных»:**

- 1) в окне диалога «Схема данных» выбрать имя модифицируемой таблицы;
- 2) в контекстном меню выбрать команду «**Конструктор таблицы**»;
- 3) внести необходимые изменения в структуру таблицы;
- 4) закрыть окно кнопкой «**x**» и ответить на запрос «Да».

### **Изменение существующей связи:**

- 1) все открытые таблицы закрыть;
- 2) находясь в окне базы данных, нажать кнопку «Схема данных» на панели инструментов;
- 3) если таблицы, связь между которыми надо изменить, не отображаются в окне «Схема данных», нужно нажать кнопку «Добавить» на панели инструментов. Перевести указатель мыши на имя нужной таблицы и дважды щелкнуть по имени. После этого нажать кнопку «Закрывать»;
- 4) поставить указатель мыши на линию связи, которую надо изменить, и дважды щелкнуть;
- 5) в открывшемся окне «Связи» внести нужные изменения и нажать кнопку «ОК».

### **Удаление связей:**

- 1) закрыть окно «Схема данных»;
- 2) находясь в окне базы данных, на панели инструментов нажать кнопку «Схема данных»;
- 3) поставить указатель мыши на линию связи, которую следует удалить; выделить ее;
- 4) нажать клавишу «**Del**» и на запрос ответить «Да».

### **Удаление таблицы из макета схемы данных:**

- 1) открыть окно диалога «Схема данных»;
- 2) выбрать таблицу, которую следует удалить из этого окна, и нажать клавишу «**Del**»; таблица вместе с определенными для нее связями будет удалена. В базе данных таблица и ее связи сохраняются.

**Каскадные операции.** Целостность данных в связанных таблицах обеспечивают каскадные операции двух видов:

- операции каскадного обновления;
- операции каскадного удаления.

Эти операции можно включать и выключать путем установки соответствующих флажков: «Каскадное обновление связанных полей» и «Каскадное удаление связанных полей».

Если установлен флажок «Каскадное обновление связанных полей», то любые изменения в значении ключевого поля в главной таблице, которая стоит на стороне «один» в отношениях 1:М, ведут к автоматическому обновлению соответствующих значений во всех связанных записях.

Например, при изменении кода студента в главной таблице автоматически произойдет обновление во всех записях других связанных таблиц, в которых присутствуют записи со значением кода этого студента.

При установке флажка «Каскадное удаление связанных таблиц» при удалении записи из главной таблицы обеспечивается автоматическое удаление связанных записей в подчиненных таблицах.

### **Контрольные задания**

#### **Задание № 1**

Для созданной в предыдущей работе базы данных создать копию и осуществить с ней (копией) следующие действия:

№ варианта	Наименование базы данных	Выполняемые операции редактирования
1	Ведомость для начисления зарплаты	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «Ф.И.О.»</li> <li>2 Удалить поле «К выплате»</li> <li>3 Переместить поле «Дата выдачи» между полями «Ф.И.О.» и «Больничные дни»</li> <li>4 Для поля «Цех» изменить свойства, придав ему статус поля</li> <li>5 Вставить в структуру базы данных поле «Вид выплат» (список)</li> </ol>
2	Библиотечный каталог	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «Наименование»</li> <li>2 Удалить поле «Стоимость»</li> <li>3 Переместить поле «Жанр» между полями «Шифр ББК» и «Ф.И.О. автора»</li> <li>4 Для поля «Количество» изменить значение по умолчанию</li> <li>5 Вставить в структуру базы данных поле «Место издания» (список) и «Год издания»</li> </ol>
3	Железнодорожное расписание	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «№ поезда»</li> <li>2 Удалить поле «Время в пути»</li> <li>3 Поменять местами поля «Время отправления» и «Время прибытия»</li> <li>4 Для полей «Время отправления» и «Время прибытия» изменить маску ввода</li> <li>5 Вставить в структуру базы данных поле «Расстояние» (значение по умолчанию – 0)</li> </ol>
4	Протокол соревнований	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «Ф.И.О. участника»</li> <li>2 Удалить поле «Место проведения»</li> <li>3 Переместить поле «Вид состязания» между полями «Дата проведения» и «Ф.И.О. участника»</li> <li>4 Для поля «Вид состязания» изменить свойства, придав ему статус поля</li> <li>5 Вставить в структуру базы данных поле «Наименование команды» (список)</li> </ol>

№ варианта	Наименование базы данных	Выполняемые операции редактирования
5	Справочник автоинспектора	<ol style="list-style-type: none"> <li>1 Добавить в структуру базы данных поле «№ п/п» (счетчик)</li> <li>2 Сделать поле «№ п/п» ключевым</li> <li>3 Удалить поле «Пробег в километрах»</li> <li>4 Поменять местами поля «Марка автомобиля» и «Год выпуска»</li> <li>5 Для поля «Дата регистрации» изменить маску ввода</li> </ol>
6	Сведения о поставках	<ol style="list-style-type: none"> <li>1 Добавить в структуру базы данных поле «№ поставки» (счетчик)</li> <li>2 Сделать поле «№ поставки» ключевым</li> <li>3 Удалить поле «Общая стоимость»</li> <li>4 Поменять местами поля «Наименование товара» и «Категория товара»</li> <li>5 Для поля «Категория» изменить свойства, придав ему статус поля</li> </ol>
7	Телефонный справочник	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «№ телефона»</li> <li>2 Удалить поле «Категория»</li> <li>3 Переместить поле «№ телефона» перед полем «Ф.И.О. абонента»</li> <li>4 Для поля «Район» изменить свойства, придав ему статус поля</li> <li>5 Вставить в структуру базы данных поле «Льгота» (список)</li> </ol>
8	Дневник метеонаблюдений	<ol style="list-style-type: none"> <li>1 Добавить в структуру базы данных поле «№ п/п» (счетчик)</li> <li>2 Сделать ключевым поле «№ п/п»</li> <li>3 Удалить поле «Давление»</li> <li>4 Переместить поле «Осадки» между полями «Температура» и «Направление ветра»</li> <li>5 Добавить в структуру базы данных поле «Время снятия показаний» (маска ввода)</li> </ol>
9	Отдел кадров	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «Табельный номер»</li> <li>2 Удалить поле «Стаж работы»</li> <li>3 Переместить поле «Дата поступления на работу» между полями «Ф.И.О. сотрудника» и «Отдел»</li> <li>4 Для поля «Отдел» изменить свойства, придав ему статус поля</li> <li>5 Вставить в структуру базы данных поле «Дата рождения» (маска ввода) между полями «Ф.И.О. сотрудника» и «Дата поступления на работу»</li> </ol>
10	Карточка абитуриента	<ol style="list-style-type: none"> <li>1 Сделать ключевым поле «Ф.И.О.»</li> <li>2 Удалить поле «Отделение»</li> <li>3 Поменять местами поля «Оценка на экзамене по русс. яз.» и «Оценка на экзамене по математике»</li> <li>4 Для полей «Оценка на экзамене по ...» изменить значения по умолчанию</li> <li>5 Вставить в структуру базы данных поле «Образование» (список)</li> </ol>

## Задание № 2

Для базы данных, созданной в разд. 5, выполнить следующие действия:

1. Заполнить базу данных 25-30 записями (данные, вводимые в базу данных, должны соответствовать всем требованиям, предъявляемым к информации: целостность, полнота, достоверность и т.д.).
2. Добавить в базу данных 5 записей.
3. Удалить 5 записей: одну – в начале базы данных, одну в конце базы данных, три – в промежутках между ранее введенными записями.
4. Закрепить последний столбец.
5. Скрыть второй и третий столбцы.
6. Снять закрепление и отобразить скрытые столбцы.
7. С помощью функции *Поиск и замена* осуществить редактирование информации в базе данных.

## Задание № 3

В созданной и заполненной базе данных произвести сортировку и отбор информации по следующим критериям:

№ варианта	Наименование базы данных	Сортировка	Фильтрация
1	Ведомость для начисления зарплаты	<ol style="list-style-type: none"><li>1. В алфавитном порядке по полю «Ф.И.О.»</li><li>2. В порядке убывания по полям «Цех» и «К выплате»</li></ol>	<ol style="list-style-type: none"><li>1. Фамилия начинается с буквы «Н»</li><li>2. Цех – «Инструментальный» и количество больничных дней более 3</li></ol>
2	Библиотечный каталог	<ol style="list-style-type: none"><li>1. В алфавитном порядке по полю «Наименование»</li><li>2. В порядке возрастания по полям «Шифр ББК» и «Ф.И.О. автора»</li></ol>	<ol style="list-style-type: none"><li>1. Наименование включает слово «экономика»</li><li>2. Жанр – роман или повесть и стоимость превышает 1500 р.</li></ol>
3	Железнодорожное расписание	<ol style="list-style-type: none"><li>1. В порядке убывания по полю «№ поезда»</li><li>2. В порядке возрастания по полям «Пункт назначения» и «Время в пути»</li></ol>	<ol style="list-style-type: none"><li>1. Пункт назначения – все, кроме Москвы</li><li>2. Время отправления – не позднее 16:00 и время в пути не превышает 8 часов</li></ol>
4	Протокол соревнований	<ol style="list-style-type: none"><li>1. В порядке возрастания по полю «Дата проведения»</li><li>2. В порядке убывания «Вид состязания» и «Ф.И.О. участника»</li></ol>	<ol style="list-style-type: none"><li>1. Фамилия участника начинается с букв «Но»</li><li>2. Вид состязаний – бег, год рождения – с 1971 по 1976</li></ol>

№ варианта	Наименование базы данных	Сортировка	Фильтрация
5	Справочник автоинспектора	1. В алфавитном порядке по полю «Марка автомобиля» 2. В порядке убывания по полям «Год выпуска» и «Пробег в километрах»	1. Марка автомобиля – Вольво 2. Дата регистрации – не позднее 07.07.1996, цвет – красный и синий
6	Сведения о поставках	1. В алфавитном порядке по полю «Наименование товара» 2. В порядке убывания по полям «Категория товара» и «Общая стоимость»	1. Категория товара – продовольственные 2. Дата поставки – первый квартал 2000 года
7	Телефонный справочник	1. В алфавитном порядке по полю «Ф.И.О. абонента» 2. В порядке возрастания по полям «Район» и «№ телефона»	1. № телефона начинается с цифры 5 и заканчивается на цифру 2 2. Дата установки – третий квартал 1999 года
8	Дневник метеонаблюдений	1. В порядке возрастания по полю «Дата» 2. В порядке убывания по полям «Осадки» и «Температура»	1. Осадки – снег и дождь 2. Температура – не ниже 0 и не выше 1
9	Отдел кадров	1. В алфавитном порядке по полю «Ф.И.О. сотрудника» 2. В порядке возрастания по полям «Отдел» и «Дата поступления»	1. Фамилия заканчивается на «ов» или «ова» 2. Должность – инженер со стажем работы от 3 до 5 лет
10	Карточка абитуриента	1. В порядке возрастания среднего балла 2. В порядке возрастания по полям «Специальность» и «Ф.И.О.»	1. Фамилия начинается с буквы «К» 2. Специальность – менеджмент и средний балл не ниже 4.3

#### Задание № 4

№ варианта	Наименование базы данных	Задание
1	Ведомость для начисления зарплаты	1. Создать таблицу «Карточка отдела кадров», содержащую поля «Ф.И.О.», «Домашний адрес», «Домашний телефон» и «Паспортные данные» 2. Установить связь между исходной и созданной таблицами по полю «Ф.И.О.». Тип связи выбрать самостоятельно

№ варианта	Наименование базы данных	Задание
2	Библиотечный каталог	<p>1. Создать таблицу «Сопроводительная карточка книги», содержащую поля «Наименование», «Год издания». «Место издания», «Инвентарный номер»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «Наименование». Тип связи выбрать самостоятельно.</p>
3	Железнодорожное расписание	<p>1. Создать таблицу «Маршрутный лист», содержащую поля «№ поезда», «Ф.И.О. начальника поезда». «Количество вагонов в составе», «Наличие вагона-ресторана»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «№ поезда». Тип связи выбрать самостоятельно</p>
4	Протокол соревнований	<p>1. Создать таблицу «Карточка участника», содержащую поля «Ф.И.О. участника», «Дача рождения». «Рост», «Вес», «Домашний адрес»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «Ф.И.О. участника». Тип связи выбрать самостоятельно</p>
5	Справочник автоинспектора	<p>1. Создать таблицу «Владелец автомобиля», содержащую поля «Ф.И.О. владельца», «Домашний адрес», «Домашний телефон», «№ автомобиля»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «№ автомобиля». Тип связи выбрать самостоятельно</p>
6	Сведения о поставках	<p>1. Создать таблицу «Сопроводительный лист товара», содержащую поля «Наименование товара», «Предприятие-производитель», «Дата выпуска», «Срок реализации»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «Наименование товара». Тип связи выбрать самостоятельно</p>
7	Телефонный справочник	<p>1. Создать таблицу «Дополнительная информация», содержащую поля «№ телефона», «Тип аппарата», «Наличие АОН», «Наличие модема»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «№ телефона». Тип связи выбрать самостоятельно</p>
8	Дневник метеонаблюдений	<p>1. Создать таблицу «Дополнительная информация», содержащую поля «Дата», «Ф.И.О. наблюдателя», «Должность наблюдателя», «Время снятия показаний»</p> <p>2. Установить связь между исходной и созданной таблицами по полю «Дата». Тип связи выбрать самостоятельно</p>

№ варианта	Наименование базы данных	Задание
9	Отдел кадров	1. Создать таблицу «Карточка сотрудника», содержащую поля «Ф.И.О. сотрудника», «Домашний адрес», «Домашний телефон», «Паспортные данные», «Табельный номер» 2. Установить связь между исходной и созданной таблицами по полю «Табельный номер». Тип связи выбрать самостоятельно
10	Карточка абитуриента	1. Создать таблицу «Данные абитуриента», содержащую поля «Ф.И.О.», «Домашний адрес», «Домашний телефон», «Паспортные данные», «Средний балл аттестата» 2. Установить связь между исходной и созданной таблицами по полю «Ф.И.О.». Тип связи выбрать самостоятельно

## 7 ФОРМИРОВАНИЕ ЗАПРОСОВ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

### 7.1 Понятие запроса к базе данных

*Запрос* (query) есть механизм выбора и представления информации из базы данных. Запрос направляется пользователем или программой в СУБД для поиска отдельных записей в базе данных. Вопрос, сформированный по отношению к базе данных, и есть запрос.

Запросы отличаются от фильтров следующими свойствами:

- Запрос можно сохранить в виде объекта базы данных.
- Запрос может содержать поля из нескольких таблиц.
- В качестве основы для отчета или формы можно использовать запрос.
- Специальные запросы можно использовать для вычислений, для обобщения информации, для добавления записей из одной таблицы в другую, для удаления записей, а также для глобального редактирования.

Метод создания запроса изобретен фирмой ИВМ в 70-х гг. XX в. Для выборки информации из базы данных предназначены Мастера запросов и Конструкторы запросов. С их помощью можно указывать поля, которые должны быть изображены для выбранных записей, создавать сложные критерии для выбора записей с одной или нескольких таблиц, делать вычисления с выбранными данными.

Механизм запросов реализует одно из основных назначений базы данных – быстрый поиск информации и получение ответов на поставленные вопросы.

Применяются два типа запросов: запросы по образцу (QBE) и структурированный язык запросов (SQL).



*Запрос по образцу* (QBE) – средство для отыскания необходимой информации в базе данных (из одной или нескольких таблиц), обычно путем занесения известных значений в таблицу. Система запроса по образцу выводит данные, имеющие отношение к вводимым. Запрос формируется не на специальном языке, а *путем заполнения бланка запроса в окне Конструктора запросов*. Указываются критерии выборки записей в исходной таблице. С точки зрения пользователя, конструирование запроса – это просто пометка нужных элементов формы в окне Конструктора запросов.

*SQL-запросы*. С помощью структурированного языка формируются запросы любой сложности. Запрос составляется из последовательности SQL-инструкций, которые задают, что надо сделать с вводным набором данных для генерации выходного набора. Выбранную информацию с базы данных можно обработать с помощью специальных анализирующих функций (минимальные и максимальные значения, средние величины, суммы).

SQL-запрос представляет собой последовательность инструкций, в которую можно включить выражения и вызов агрегатных функций.

Работа с SQL-запросом рассчитана на специалиста-программиста. Пользователю, как правило, не приходится обращаться к языку SQL напрямую.

Все запросы Access строит на основе соответствующего SQL-запроса. Чтобы просмотреть его, необходимо в активном окне проектирования запроса дать команду «Вид»→ «SQL».

**Создание запросов выбора.** MS Access из основной таблицы базы данных выбирает те запросы, которые соответствуют критериям запроса. Результатом обработки запроса является временный динамичный набор данных.

## 7.2 Создание запроса с помощью Мастера

При создании запроса необходимо определить:

- поля в базе данных, по которым будет идти поиск информации;
- предмет поиска в базе данных;
- вид (перечень полей) в результате выполнения запроса.

Простой запрос обеспечивает выборку всех записей по заданным пользователем полям и не выполняет выборку записей по критериям отбора, а также сортирование записей.

*Последовательность создания простого запроса с помощью Мастера:*

- 1) в окне базы данных перейти на вкладку «Запрос»;
- 2) нажать кнопку «Создать» →[«Новый запрос»] →выбрать опцию создания запросов с помощью Конструктора запроса или один из Мастеров создания запроса;
- 3) выбрать опцию «Простой запрос», нажать кнопку «ОК»; открывается окно диалога «Создание простых запросов» (рисунок 7.1);

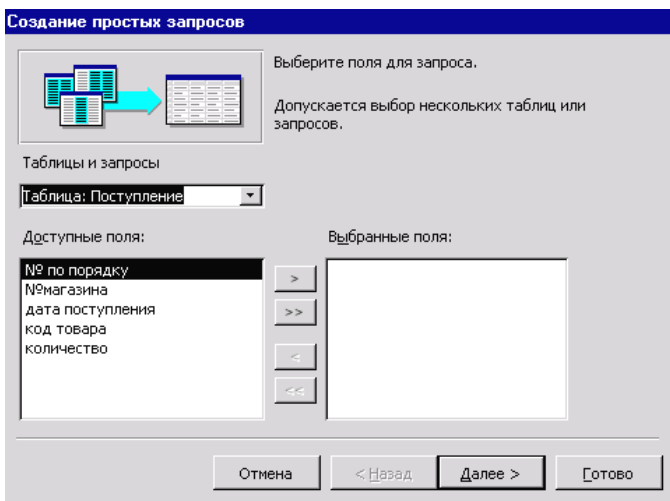


Рисунок 7.1 – Окно Мастера простого запроса

4) открыть список «Таблицы/Запросы», щелкнув мышью кнопку списка, и выбрать таблицу-источник или запрос, а из списка «Доступные поля» — список полей таблицы (таблиц) и включить их в зону «Выбранные поля» для создания запроса, дважды щелкнув мышью или нажав одну из кнопок навигации.

Если запрос формируется на основе нескольких таблиц или запросов-источников, необходимо повторить действия, указанные в п. 4, для каждого источника;

5) после выбора всех полей для формирования запроса щелкнуть кнопку «Далее»; на экране появится следующее окно диалога Мастера запросов, где предлагается: «**Выберите подробный или итоговый отчет**»;

6) подсчитать результаты, для чего следует выбрать опцию «**Итоговый**»;

7) для определения суммы по указанным полям щелкнуть кнопку «**Итоги...**»; открывается окно диалога «**Итоги**» – выбираем (путем установки флажка) вид выполняемого значения, которое следует получить, например «**Sum**» – подсчет значений поля по всем записям. Расчеты выполняются над значениями полей типа «числовой» и «денежный»;

8) щелкнуть кнопку «**ОК**» и возвратиться в предыдущее окно;

9) щелкнуть кнопку «**Далее**» для продолжения формирования запроса. В окне диалога задать (ввести) имя созданного запроса и выбрать один из вариантов дальнейших действий. Если выбрана опция «Открытие результатов выполнения запроса», то на экран выводится таблица с результатами работы созданного запроса; если выбрана опция «Изменение структуры

запроса», то необходимо перейти в окно Конструктора запросов, в котором при необходимости можно дать новые параметры запроса;

10) щелкнуть кнопку **«Готово»**; на экране появится таблица с результатами запроса;

11) закрыть окно запроса; возвратиться в окно базы данных, в котором указано имя созданного запроса.

### 7.3 Создание запроса с помощью Конструктора

С помощью Конструктора можно создать следующие виды запросов: простые, по условию, параметрические, итоговые, с вычисляемыми полями.

Чтобы вызвать Конструктор запросов, необходимо перейти в окно базы данных.

*Последовательность действий:*

1) в окне базы данных выбрать вкладку **«Запросы»** и щелкнуть кнопку **«Создать»**; откроется окно диалога **«Новый запрос»**;

2) выбрать опцию **«Конструктор»** и нажать кнопку **«ОК»**; открывается окно диалога **«Добавление таблицы»**. За ним появляется неактивное окно **«Запрос на выборку»**. В нем из списка следует выбрать имя таблицы-источника, на основе которой будет проводиться выбор данных, и щелкнуть кнопку **«Добавить»**; в окне **«Запрос на выборку»** появится список выбранных полей для запросов. Если запрос формируется на основе нескольких таблиц, то действия, указанные в п. 2, повторяются для каждой таблицы-источника;

3) закрыть окно диалога кнопкой **«Закрыть»**; окно **«Добавление таблицы»** закроется, а окно **«Запрос на выборку»** станет активным. На экране появится окно Конструктора запросов (рисунок 7.2), а в главном меню – команда **«Запрос»**.

С помощью инструментария окна Конструктора формируем запрос:

- выбираем таблицу-источник, из которой проводится выборка записей;
- выбираем поля результатов запроса;
- указываем критерии для выборки, группирования и упорядочения данных.

Команда меню **«Запрос»** позволяет добавлять таблицы в окно Конструктора запроса, удалять таблицы и другие операции.

Меню **«Вид»** содержит команды управления запросом.

Окно Конструктора условно состоит из двух частей – верхней и нижней (см. рисунок 7.2).

*В верхней части* окна Конструктора запроса размещается схема данных запроса, которая содержит список таблиц-источников и отражает связь между ними.

*В нижней части* окна находится Бланк запроса, в котором каждая строка выполняет конкретную функцию:

«Поле» – указываются имена полей, которые участвуют в запросе;

- «Имя таблицы» – имя таблицы, с которой выбрано это поле;
- «Сортировка» – указывает тип сортировки;
- «Вывод на экран» – устанавливается флажок просмотра поля на экране;
- «Условие отбора» – задаются критерии поиска.

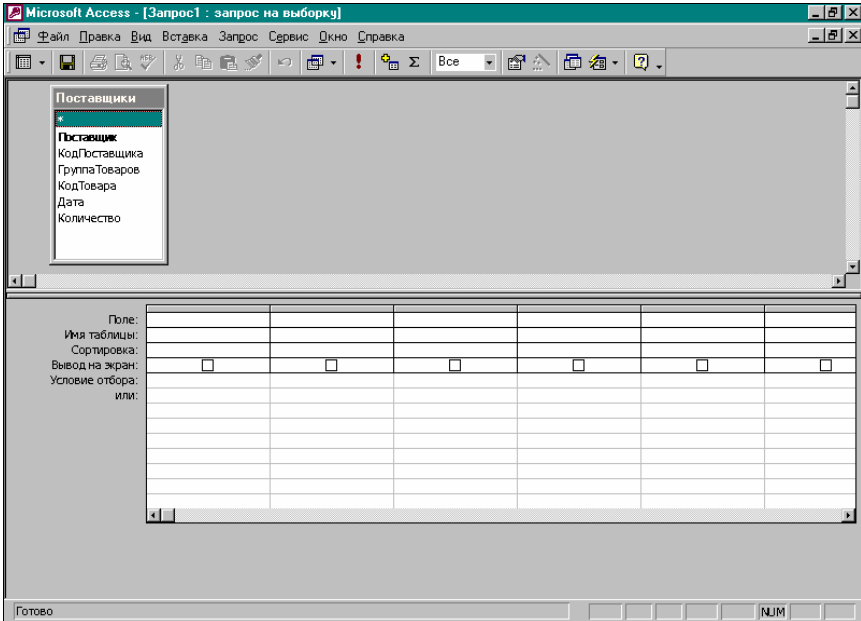


Рисунок 7.2.– Окно Конструктора запросов для нового запроса

Техника заполнения Бланка запроса:

1) переместить имена полей с источника (верхняя часть окна) в Бланк запроса. В верхней части окна помещены таблицы и их поля, из которых нужные переносятся в нижнюю часть бланка запроса. Для примера используем таблицу «Поставки».

В нижней части окна в ячейке «Поле» кнопкой раскрываем структуру базы данных, выбираем (щелкаем) необходимое имя поля из таблицы, например «КодТовара». После этого в ячейке «Поле» появится имя выбранного поля, а ниже, в строке «Имя таблицы», — имя базы данных (рисунок 7.3).

Переносим другое имя «Количество» из таблицы в запрос. Это можно сделать перетягиванием мышью выбранного поля с верхней части окна запроса в Бланк запроса в следующую ячейку «Поле». После этого нижняя часть окна «Запрос на выборку» будет содержать определенную информацию (см. рисунок 7.3);

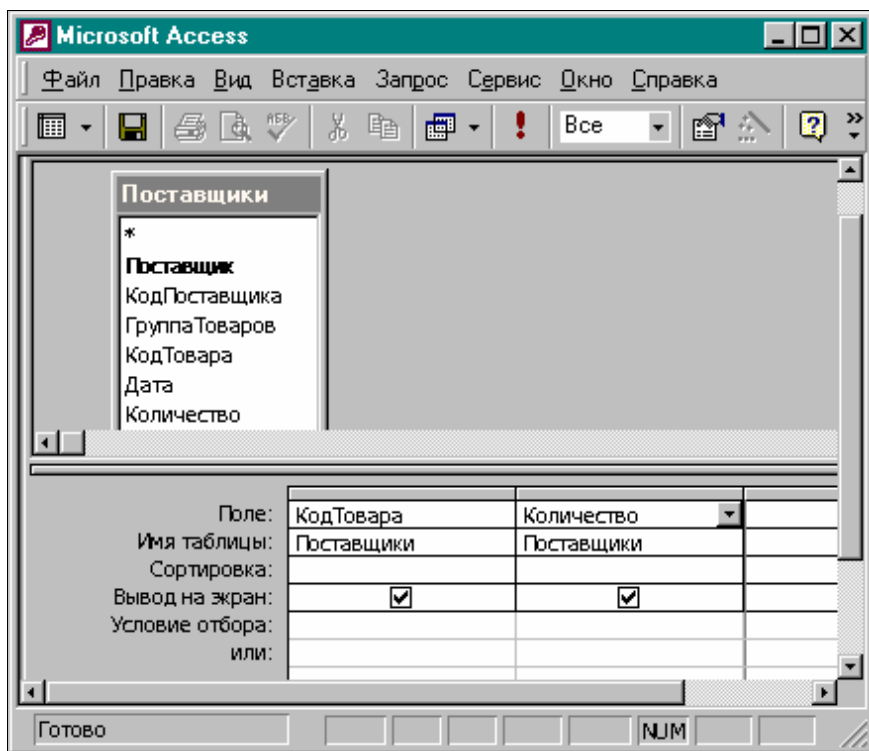


Рисунок 7.3 – Нижняя часть окна Конструктора запросов

2) задать принцип сортировки. Курсор мыши перевести в строку «**Сортировка**», в соответствующем имени поля появится кнопка открытия списка, щелкнуть ею и в открывшемся списке выбрать способ сортировки (например, «По возрастанию»);

3) в строке «**Вывод на экран**» автоматически устанавливается флажок просмотра найденной информации в поле (для отмены вывода на экран следует снять флажок);

4) в строке «**Условия отбора**» и размещенного ниже «**ИЛИ**» ввести условия ограниченного поиска – критерии поиска. В этом ряду напротив поля-ключа (например, «КодТовара») ввести с клавиатуры условие выбора записей для формирования запроса, например «4060», или в поле «Количество» > 500. Для выбора записей КодаТовара 4060 и 4055 условия в строке «Условия отбора» записываются так: «4060 OR 4055». Это же условие можно записать в две строки: в строке «Условия отбора» записать 4060, а в строке «ИЛИ» — 4055;

5) после завершения формирования запроса закрыть окно **«Запрос на выборку»**; открывается окно диалога **«Сохранить»** → **«Да»** → (**«Сохранение»**) → ввести имя созданного запроса → **«ОК»** → вернуться в окно базы данных, в котором указано имя созданного запроса.

Чтобы открыть запрос из окна базы данных, надо выделить имя запроса и щелкнуть кнопку **«Открыть»**; на экране появится окно **«Запрос по выбору»** с присвоенным именем.

**Добавление полей в Бланк запроса.** Выбор полей с таблицы-источника (верхняя часть окна **«Запрос на выборку»**) в Бланк запроса (нижняя часть окна) происходит или путем «перенести-и-бросить», или путем выбора поля таблицы в верхней части окна двойным щелчком мыши. В строке **«Поле»** появится имя поля, а в строке **«Имя таблицы»** автоматически появится имя таблицы-источника; в строке **«Вывод на экран»** будет установлен флажок.

**Удаление полей с Бланка запроса.** Бланк запроса можно очистить (удалить все поля) командой **«Правка»** → **«Очистить бланк»**.

**Изменение порядка полей в Бланке запроса.** В Бланке запроса поля размещаются в той последовательности, в которой мы хотим их видеть в итоговой таблице. Установленный порядок можно изменить. Для этого необходимо:

1) установить курсор мыши на область выбора столбца (маленький прямоугольник, размещенный сразу над названием поля); как только указатель мыши примет вид стрелки, нажать кнопку для выделения столбца (это единственное место для «захвата» столбца для его переноса);

2) нажать и не отпускать кнопку мыши; на конце указателя появится прямоугольник;

3) переместить столбец в нужном направлении (толстая вертикальная черта покажет его бегущее положение);

4) отпустить кнопку мыши, как только толстая вертикальная черта будет находиться в нужном месте; поле будет перенесено на новое место.

**Сортировка результатов выборки.** Для сортировки результатов выборки по одному полю необходимо перейти на строку **«Сортировка»** нужного поля и в раскрывающемся списке выбрать способ сортировки (по возрастанию или по убыванию). Для отказа от сортировки по выбранному ранее полю следует выбрать значение **«Отсутствует»**.

Для сортировки по нескольким полям необходимо последовательно перейти на поля сортировки и установить для них критерии (флажки).

**Запуск запроса.** Для запуска запроса на исполнение из окна Конструктора **«Запрос на выборку»** надо на панели инструментов нажать кнопку **«Запуск» (!)** или подать команду **«Запрос»** → **«Запуск»**. Результаты выборки данных по запросу выводятся на экран в режиме таблицы.

**Сохранение запроса.** Перед непосредственным сохранением необходимо присвоить имя запросу. Дать команду **«Файл»** → **«Сохранить»**

как/экспорт»→ [«Сохранение объекта»]; в поле «Новое имя» Access во умолчанию присвоит имя запросу. Можно ввести свое имя и нажать на кнопку «ОК».

**Применение простого критерия выборки записей для ограниченного поиска.** *Последовательность:*

- 1) определить поле-ключ;
- 2) в строке «Условия отбора» бланка запроса ввести критерии выборки записей. Например, по полю «Номенклатурный номер» необходимо выбрать все записи номенклатурного номера 2540: надо записать = 2540;
- 3) по желанию можно снять флажок вывода на экран поля-ключа;
- 4) запустить запрос кнопкой «Запуск» или командой «Запрос» →«Запуск»;
- 5) можно сохранить этот запрос под своим именем.

**Изменение внешнего вида итоговой таблицы запроса.** Применяются те же средства, что и для обычных таблиц. Можно сделать поле невидимым, зафиксировать поля, изменить шрифт, изменить параметры столбцов и строк. Для этого необходимо перейти в режим таблиц, подать команду «Формат» и выбрать необходимую команду.

#### 7.4 Вычисляемые поля в запросах. Построитель выражений

В итоговую таблицу запросов можно включить вычисляемые поля (графы), т.е. получить данные, которые отсутствуют в исходной таблице.

Новое вычисляемое поле создается прямо в Бланке запросов. Выражение записывается в пустую ячейку бланка с помощью «Построителя выражений», который предназначен для составления математических, логических и других выражений.

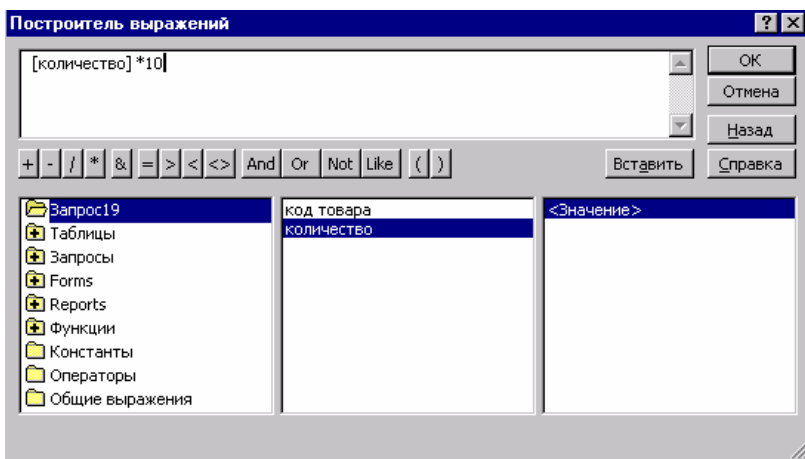


Рисунок 7.4 – Окно «Построитель выражений» со сформированным запросом

Окно «Построитель выражений» (рисунок 7.4) вызывается одним из следующих способов:

- нажать кнопку «**Построить**» на панели инструментов;
  - в открытом окне Конструктора (форм, запросов, отчетов) щелкнуть правой кнопкой мыши строку «Условие отбора» и выбрать команду «Построить»;
- в строке свойств «**Данные**» нажать кнопку «**Построитель**».

Окно «Построитель выражений» включает в себя четыре области с собственными полосами прокрутки. Создаваемое выражение формируется в верхней области – поле ввода выражения. Там же размещаются и кнопки со знаками операторов.

Три области поиска элементов для записи выражения размещены в нижней части окна. *Левая область* содержит список источников данных (таблицы и др.) и средства для записи выражений (функции и др.); *средняя область*—элементы, которые входят в выбранный объект в левом окне; *правая область* предназначена для формирования нужного объекта (элемента).

Поиск нужных элементов начинается с левой области. Списки объектов в левой, средней и правой областях имеют иерархическое подчинение.

Последовательность вставки элемента выражения в верхнее поле ввода:

- 1) в левой области выбрать необходимую папку и открыть ее для выбора нужного объекта (элемента);
- 2) выбрать объект из той папки, где он находится;
- 3) последовательно в средней и правой областях выбрать нужный элемент и перенести его в поле ввода выражения (верхняя часть окна) кнопкой «Вставить» или дважды щелкнуть элемент мышью;
- 4) в зоне кнопки операторов выбрать нужный оператор и щелкнуть мышью для переноса его в создаваемое выражение;
- 5) после формирования нужного выражения нажать кнопку «ОК». Рассмотрим это на примере включения вычисляемого поля и формирования выражения с помощью «Построителя выражений».

Таблица «Приход» включает следующие поля: **КодТовара, Товар, Количество, Цена**. Необходимо получить следующую структуру: **КодТовара, Количество, Цена, Сумма**. Для записи выражения надо:

- 1) открыть окно Конструктора запроса для таблицы;
- 2) перейти на строку «**Поле**» пустого столбца Бланка запроса;
- 3) нажать кнопку «**Построить**» на панели инструментов или выбрать одноименную команду с контекстного меню; на экране откроется окно построения выражений (см. рисунок 7.4).

Дважды щелкнуть мышью вкладку «**Таблицы**» — откроется перечень таблиц; выбрать имя нужной таблицы (щелкнуть один раз). В соседнем окне появится перечисление полей выбранной таблицы. В нашей примере для **получения вычисляемого поля «Сумма»** надо поле «**Количество**» умножить на поле «**Цена**».



Для записи (добавления) в выражение поля «**Количество**» с таблицы «**Приход**» надо открыть ее и выбрать это поле из списка.

В правом от окна «**Таблицы**» списке «**Значение**» перечисляются все поля таблицы «**Приход**» (выбранной таблицы). Выбрать поле «**Количество**» и дважды щелкнуть его мышью или нажать кнопку «**Вставить**» в окне «**Построитель выражений**». Выбранное поле (в нашем примере — «**Количество**») будет записано в окне ввода выражения в качестве операнда (вверху над знаками операций): **[Приход]! [Количество]** – обратите внимание на синтаксис.

Далее записать (ввести) знак операции (в примере операция умножения (\*)). Достаточно нажать кнопку со знаком умножения (кнопки операций размещены под областью ввода выражений) или ввести знак операций с клавиатуры:

**[Приход]! [Количество] \***

Добавить в выражение другой операнд – поле «**Цена**» тем же способом, что и поле «**Количество**». В результате получаем выражение:

**[Приход]! [Количество] \* [Приход] [Цена]**

После формирования выражения нажать клавишу «ОК»; выражение, которое записано с помощью программы «Построитель выражений», появляется в строке «**Поле**» Бланка запросов. Каждое поле **итоговой** таблицы **должно** иметь имя. Access автоматически присвоит имя «**Выражение1**», которое отделяется от выражения двоеточием. В законченном виде в бланке-запросе выражение выглядит следующим образом:

**Поле: Выражение1 : [Приход]! [Количество] \* [Приход]! [Цена]**

Можно заменить имя поля итога на «говорящее», например «**Сумма**». Для этого необходимо дважды щелкнуть на «**Выражение1**» и ввести имя поля (в нашем примере – «**Сумма**»). В результате получим:

**Поле: Сумма: [Приход]! [Количество] \* [Приход]! [Цена]**

**Выбор записей с таблицы, значения которых не соответствуют определенному условию одного из полей.** Для этого предназначен оператор **Not** (Нет), который записывается перед сравниваемым значением. Например, для того чтобы выбрать все записи с таблицы «**Приход**», за исключением записей со значением «**Мыла**» в поле «**Товар**», надо:

- 1) открыть запрос;
- 2) в строке «**Условие отбора**» поля «**Товар**» записать: **Not Мыла** или **<> мыла**;
- 3) установить флажок «**Вывод на экран**»;
- 4) для выполнения запроса нажать кнопку «**Запуск**», на экране появится итоговая таблица.

**Выбор записей при условии неточного совпадения условий поиска.** Если вы не помните точного написания условия (ключ) поиска, следует ис-

пользовать оператор **Like** (похожий). Образ задается сразу за оператором Like. Разумеется, образ может содержать и точное значение, например

**like мыла**

Образ поиска может содержать символы шаблона для **поиска** диапазона значений. Так, в строке «**Условие отбора**» может быть указано:

**Like «мы\*»**

Символы шаблонов приведены в таблице 7.1.

Таблица 7.1 – Символы шаблонов

Символ в образе	Соответствие в выражении
?	Любой один текстовый символ
*	Соответствует любой цифре или любому символу
#	Любая одна цифра
[ список знаков ]	Любой один знак в «списке знаков»
[!список знаков ]	Любой один знак, который не входит в «список знаков»

*Применение оператора Like.* Предположим, надо выбрать товары, название которых начинается на букву «Б» или «Д». В строке «**Условие отбора**» в поле «**Товар**» введем условие **Like' [БД]\*'**.

По этой команде будут выбраны все товары, название которых начинается на букву «Б», а потом на букву «Д».

**Выбор записей по диапазону значений.** Границы диапазона задаются точно. В окне Конструктора применяются следующие операторы:

> (больше) >= (не меньше) < (меньше) <= (не больше) Between (между)

*Пример.* Выбрать список товаров, цена которых больше 20000:

- 1) открыть окно Конструктора запроса для таблицы «Приход»;
- 2) перенести поля, которые хотим видеть в Бланке запроса;
- 3) в строке «Условие отбора» в поле «Цена» записать > 20000;
- 4) нажать кнопку «Запуск».

*Пример.* Вывести список товаров ценой от 20000 до 50000;

- 1) открыть окно запроса для таблицы «Приход»;
- 2) выбрать поле «Товар» в таблице «Приход». Перейти на строку «Условие отбора» и нажать кнопку «Построить» на панели инструментов. В списке операторов сравнения выбираем Between. В области ввода выражения появится шаблон для ввода параметров этого оператора:

**Between «Выражение» And «Выражение»;**

- 3) ввести значение первого параметра (нижнюю границу) 20000 и второго параметра 50000;

- 4) нажать кнопку «Запуск»; на экране появится итоговая форма.

*Пример задания диапазона символьных данных:* вывести список товаров с названиями от буквы «М» до буквы «С».

Действия: в строке условия отбора ввести Like '[M-C]\*'

**Сложные критерии выборки.** Часто приходится выбирать записи по условию, которое задается для нескольких полей таблицы или по нескольким условиям для одного поля. В этом случае применяются «И-запросы» (выбор записей только при условии выполнения всех условий) и «ИЛИ-запросы» (выбор записей при выполнении хотя бы одного из условий).

*Пример.* Выбрать записи о товарах с названиями от буквы «А» до буквы «Ж», «Цена» которых 300000:

«Поле» Товар. Количество. Цена.

«Условие отбора» Like'[А-Ж]\*' 300000

При задании «ИЛИ-запроса» каждое условие выборки должно размещаться на отдельной строке Бланка заказа.

*Пример.* Выбрать товары с названием от буквы «А» до буквы «Ж» и ценой 300000 и 500000:

«Поле» Товар. Количество. Цена.

«Условие отбора» Like'[А-Ж]\*' 300000

«или» 500000

Операторы **OR** и **AND** позволяют формировать в одной строке сложное условие выборки. Предшествующий пример, в частности, записывается следующим образом:

«Поле» Товар. Количество. Цена.

«Условие отбора» 300000 OR 500000

## 7.5 Технология создания запросов

**Объединение нескольких таблиц для проектирования запросов.** Довольно часто применяются многотабличные запросы – информация берется не с одной, а с нескольких взаимосвязанных таблиц. Для формирования многотабличного запроса в окне Конструктора запросов надо добавить все таблицы, которые участвуют в выборке, и задать условия их объединения.

Для добавления таблицы подается команда «Запрос» → «Добавить таблицу» → [ ] **выбрать таблицу** → «Добавить» → «Заккрыть» или на панели инструментов нажать кнопку «Добавить таблицу».

Образ таблицы появляется в схеме запросов. Связь между таблицами, которые участвуют в запросе, показывается линиями. Можно установить нужную связь способом «перенести и бросить» (выбрать поле в одной из таблиц, нажать кнопку мыши и перенести выбранное поле на связываемое поле в другой таблице).

Для запуска запроса предназначена кнопка «Запуск» на панели инструментов.

**Итоговые запросы.** Очень часто необходимо обрабатывать данные и в запросах выдавать результаты обработки. Запросы, которые выполняют вычисления в группах записей, называются *итоговыми*. Можно вычислить сумму, среднее, минимальное и максимальное значения поля.

Для создания итогового запроса надо, находясь в окне Конструктора запросов, подать команду «Вид»→ «Групповые операции» или нажать на кнопку «Групповые операции» на панели инструментов. После этого в Бланке запроса (внизу окна Конструктора запросов) появится новая строка под названием «Групповые операции». В ней в соответствующей графе указываем тип выполняемого вычисления, раскрывая список типов операций. Список типов операций в строке «Групповые операции» приведен в таблице 7.2.

Кроме рассматриваемых запросов выбора применяются и другие: запрос действия, параметрические и перекрестные запросы.

**Запрос действия.** Позволяет сменять и переносить данные в таблицах, добавлять и выделять записи, создавать новые таблицы с динамического набора. Существуют четыре типа запросов действия: запрос добавления, запрос удаления, запрос обновления и запрос создания таблиц.

Таблица 7.2 – Типы операций

Значение	Выполняемые операции
Sum	Сложение
Avg	Среднее значение
Min	Минимальное значение
Max	Максимальное значение
Count	Количество записей
StDev	Стандартное отклонение
Var	Дисперсия
First	Значение в первой записи
Last	Значение в последней записи

*Замечание.* В окне базы данных Access перед именем запроса действия устанавливает восклицательный знак (!).

**Запрос добавления.** Записи одной таблицы (отобранные запросом или все) можно присоединять в конец другой таблицы.

**Запрос удаления.** Можно удалить группу записей, которые отобраны по некоторому критерию.

**Запрос обновления.** Можно изменить группу записей, которые отобраны на основе определенных критериев. Для изменения группы записей с помощью запросов обновления надо:

- 1) в режиме Конструктора запроса на панели инструментов кнопкой «Тип запроса» выбрать команду «Обновление»; в нижнем окне появится строка «Обновление»;
- 2) в Бланке запроса со списка полей мышью перенести поля, которые необходимо обновить, или определить для них условия отбора;
- 3) при необходимости в ячейке «Условие отбора» указывается условие отбора;

4) для полей, которые необходимо обновить, в ячейку «**Обновление**» ввести выражение или значение, которое будет использовано для изменения:

**Поле:** Цена.

**Имя таблицы:** Поставки

**Обновление:** Цена \* 102;

5) для просмотра обновленных записей предназначена кнопка «**Вид**» на панели инструментов. Выводимый список не будет содержать новых значений. Для возвращения в режим Конструктора запросов снова нажать кнопку «**Вид**» на панели инструментов. Любые изменения надо вносить в режиме Конструктора;

6) для создания новой таблицы нажать кнопку «**Запуск**» (!) на панели инструментов.

**Запрос создания таблицы.** Используя динамичный набор данных, сформированных при выполнении запроса, можно создать таблицу с помощью запроса создания таблицы. Для этого подается команда «**Запрос**»→ «**Создание таблицы**»→«**Создание таблицы**»→ ввести имя новой таблицы «**ОК**».

## 7.6 Параметрические запросы

Некоторые запросы используются периодически без изменения критериев, другие запросы получаются путем незначительных видоизменений основных запросов.

В параметрическом запросе указывается критерий, который может изменяться по заказу пользователя. Такой запрос выгодно применять как основу для форм и отчетов. Например, в отчете появляется приглашение ввести месяц, за который он составлен. Условие отбора записей задается непосредственно при вызове запроса. При этом для внесения изменений не требуется открывать запрос в окне Конструктора (строка «Условие отбора»).

### **Последовательность создания параметрического запроса:**

- 1) создать запрос в режиме Конструктора;
- 2) в Бланке запроса (нижняя часть окна Конструктора) в строке «**Условие отбора**» вводится условие отбора в виде приглашения в квадратных скобках, например [Введите код товара];
- 3) закрыть окно «**Запрос на выборку**» и на запрос, сохранить ли изменения, ответить «**Да**»; вернуться в окно базы данных, где представлен созданный запрос (выделенный);
- 4) нажать кнопку «**Открыть**»; откроется окно диалога «**Введите значение параметра**» с текстом-приглашением, например «Введите код товара»;
- 5) ввести значение параметра отбора и нажать кнопку «**ОК**». На экране «**Запрос на выборку**» выводится запрос с записями, которые отображены по определенным параметрам. Если параметр введен неправильно или нет записей с заданным значением, то на экране выводится запрос только с названиями полей без их значений или сообщение об ошибке.

### *Другой вариант создания параметрического запроса:*

1) в окне Конструктора на свободное место щелкнуть правой кнопкой мыши и в контекстном меню выбрать команду **«Параметры»**; открывается окно **«Параметр запроса»** с двумя графами — **«Параметр»** и **«Тип данных»**;

2) в первой строке графы **«Параметр»** вводится текст-приглашение, например **«Введите код»**, а в графе **«Тип данных»** щелкнуть мышью; появится открывающаяся кнопка списка типа данных;

3) после выбора типа данных нажать клавишу **«ОК»**.

## 7.7 Перекрестные запросы

В перекрестном запросе отражаются результаты расчетов (средние величины, суммы и т. д.) по **значению одного поля** таблицы. Результаты вычислений группируются *по двум наборам данных*: один находится в левом столбце таблицы, другой – в верхней строке. Перекрестный запрос представляет информацию в более компактном виде, чем обычные запросы. Данные сгруппированы как по горизонтали, так и по вертикали и размещены компактно в виде двумерной таблицы (сравните данные, представленные в таблице 7.3, и те же данные, отраженные в перекрестном запросе).

Т а б л и ц а 7.3 – Сведения о работе фермеров

Фермер	Продукция	Валовой сбор
Астрожский	Рожь	100
Астрожский	Вика	50
Астрожский	Рапс	300
Калиновский	Рожь	200
Калиновский	Вика	100
Калиновский	Рапс	500

Те же данные, отраженные в перекрестном запросе:

Фермер	Рожь	Вика	Рапс
Астрожский	100	50	300
Калиновский	200	100	500

Для названия заголовков колонок можно использовать значения нескольких полей или **выражений**, например **«Всего»**.

Access имеет Мастер по созданию перекрестного запроса.

*Последовательность действий при создании перекрестного запроса с помощью Мастера:*

1) открыть окно базы данных;

2) выбрать вкладку **«Запросы»**;

3) нажать кнопку **«Создать»**, откроется окно диалога **«Новый запрос»**;

4) выбрать Мастер **«Перекрестный запрос»** и нажать кнопку **«ОК»**; откроется окно **«Создание перекрестных таблиц»**;

5) выбрать из списка таблицу-источник и дальше выполнять инструкции в диалоговых окнах Мастера.

**Последовательность действий при создании перекрестного запроса без Мастера:**

1) в окне базы данных выбрать вкладку **«Запросы»** и нажать кнопку **«Создать»**; в открывшемся окне **«Новый запрос»** выбрать **«Режим Конструктора»** и нажать кнопку **«ОК»**.

2) в открывшемся окне **«Добавление таблицы»** выбрать вкладку, где находятся объекты с информацией для формирования запроса; выбрать источник для участия в запросе;

3) нажать кнопку **«Закрыть»** на панели инструментов; в Бланке запроса в строке **«Поле»** добавить поле и условие отбора;

4) на панели инструментов нажать кнопку **«Тип запроса»** и выбрать опцию **«Перекрестный»**;

5) для полей, значения которых будут заголовками строк, выбрать ячейку в строке **«Перекрестная таблица»**, выбрать **«Заголовки строк»** и оставить в строке **«Групповая операция»** значение **«Группировка»** для полей, значения которых будут представлены в запросе в качестве заголовков столбцов. Выбрать ячейку в строке **«Перекрестная таблица»** и выбрать **«Заголовки строк»**. Значение **«Заголовки столбцов»** можно выбрать только для одного поля и оставить — **«Группировка»**.

*Замечание.* По умолчанию заголовки столбцов сортируются по алфавиту или числовому значению.

## 7.8 Обработка запросов

Запрос для базы данных – один из элементов управления базой данных и средств манипулирования с ней.

**Сохранение запросов.** Для многократного использования разработанного запроса его надо сохранить:

1) в меню **«Файл»** → **«Сохранить»** → [**«Сохранить как/экспорт»**];

2) дать имя запросу, под которым он будет включен в список запросов в окне базы данных.

Для записи старой версии запроса под новым именем (запись без внесения изменений во время работы): **«Файл»** → **«Сохранить как»** → дать новое имя.

**Открытие запросов в режиме проектирования:**

1) в окне базы данных **щелкнуть** мышью на вкладку **«Запросы»**;

2) выделить (маркировать) необходимый запрос в списке;

3) щелкнуть мышью на кнопку **«Конструктор»**.

Открытие запроса выбора или перекрестного запроса в табличной форме:

- 1) в окне базы данных щелкнуть мышью на пиктограмму запроса;
- 2) нажать кнопку «Открыть».

**Модификация запроса.** В режиме проектирования можно вносить в ранее созданный запрос следующие изменения: переносить поля на другое место; добавлять и удалять поля; изменять критерии; изменять способ сортировки данных; подавлять (сохранять) или показывать поля; изменять, добавлять, удалять вычисляемые выражения.

### *Контрольные задания*

#### **Задание № 1**

Для созданной базы данных выполнить следующие запросы:

№ варианта	Наименование базы данных	Запросы к базе данных
1	Ведомость для начисления заработной платы	<p><u>Простой запрос:</u> вывести на экран список всех работников инструментального цеха с указанием Ф.И.О. сотрудника и суммы «К выплате»</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список работников сборочного цеха, которые были на больничном более трех дней и у которых сумма удержания превышает 2000 рублей;</p> <p>б) вывести на экран список работников, с указанием Ф.И.О. и суммы «К выплате», которые получали деньги в период с 07.07.2002 по 11.07.2002 (при этом дату получения денег на экране не отображать)</p>
2	Библиотечный каталог	<p><u>Простой запрос:</u> вывести на экран список всех книг, являющихся романами, с указанием Ф.И.О. автора и наименования книги</p> <p><u>Составные запросы:</u></p> <p>а) получить на экране список всех книг по экономике (на основании шифра ББК), стоимость которых колеблется от 1500 до 5000 рублей;</p> <p>б) вывести список книг, находящихся в библиотеке в единичном экземпляре и представляющих жанры романа и фантастики</p>
3	Железнодорожное расписание	<p><u>Простой запрос:</u> вывести на экран список всех скорых поездов (№ поезда от 1 до 99)</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список пассажирских поездов (№ поезда от 100), прибывающих в Гомель в промежутке времени от 12:00 до 00:00;</p> <p>б) вывести на экран список поездов, находящихся в пути менее пяти часов и более 12. При этом на экране отобразить поля «№ поезда», «Пункт отправления» и «Пункт назначения»</p>



№ варианта	Наименование базы данных	Запросы к базе данных
4	Протокол соревнований	<p><u>Простой запрос:</u> вывести на экран список всех победителей с указанием Ф.И.О. участника и вида состязаний</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список всех состязаний, проводившихся во втором квартале 2000 года в манеже «Динамо»;</p> <p>б) вывести на экран список участников состязаний молодежи 18 лет, занявших призовые места</p>
5	Справочник автоинспектора	<p><u>Простой запрос:</u> вывести на экран список всех автомобилей красного цвета. На экране должны быть отображены марка автомобиля и год выпуска</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список всех автомобилей, зарегистрированных во втором полугодии 1998 года с пробегом не более 7500 км;</p> <p>б) вывести на экран список всех автомобилей марки BMW, год выпуска которых лежит в промежутке от 1980 до 1990</p>
6	Сведения о поставках	<p><u>Простой запрос:</u> вывести на экран список всех промышленных товаров с указанием наименования товара и объема поставки</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список всех товаров продовольственной категории, поставленных в первом и третьем кварталах 2000 года;</p> <p>б) вывести на экран список товаров, наименование которых начинаются с буквы «С» и ценой, лежащей в диапазоне от 150 до 1500 рублей</p>
7	Телефонный справочник	<p><u>Простой запрос:</u> вывести на экран список абонентов с указанием Ф.И.О. и адреса, номера телефонов которых начинаются с цифры 5</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список всех абонентов железнодорожного района, получивших номер телефона во втором или третьем кварталах 1995 года;</p> <p>б) вывести на экран список всех абонентов Советского района, установленных на блокираторе</p>
8	Дневник метеонаблюдений	<p><u>Простой запрос:</u> вывести на экран список дней, в которые температура превышала нулевую отметку</p> <p><u>Составные запросы:</u></p> <p>а) вывести на экран список записей из метеодневника, которые были произведены с 1991 по 2002 гг.;</p> <p>б) вывести на экран все записи из дневника метеонаблюдений, когда при ветрах, дующих с северных направлений, наблюдались осадки любого вида</p>

№ варианта	Наименование базы данных	Запросы к базе данных
9	Отдел кадров	<u>Простой запрос:</u> вывести на экран список сотрудников, чей табельный номер превышает 100 <u>Составные запросы:</u> а) вывести на экран список сотрудников, поступивших на работу с 01.07.1990 по 01.07.1995 в должности инженера; б) вывести на экран список сотрудников с именами, начинающимися с буквы «А» и стажем работы от 5 до 10 лет
10	Карточка абитуриента	<u>Простой запрос:</u> вывести список абитуриентов с указанием Ф.И.О. и оценок по экзаменам, подавших заявления до июля <u>Составные запросы:</u> а) вывести список абитуриентов, получивших неудовлетворительные оценки хотя бы по одному из предметов; б) вывести на экран список абитуриентов, подавших заявления на дневное отделение, чей средний балл превышает 4,5

## Задание № 2

Для копии созданной базы данных выполнить следующие запросы:

№ варианта	Наименование базы данных	Запросы к базе данных
1	Ведомость для начисления зарплаты	<u>Запрос удаления:</u> удалить список всех работников инструментального цеха <u>Запрос обновления:</u> в связи с увеличением минимальной заработной платы сумму удержания увеличить в 1,5 раз, а начисляемый заработок – в 2 раза <u>Запрос добавления:</u> в копию таблицы добавить список работников инструментального цеха из главной таблицы
2	Библиотечный каталог	<u>Запрос удаления:</u> удалить список всех книг по экономике <u>Запрос обновления:</u> в связи с переоценкой стоимости книг увеличить стоимость книг в 1,8 раз <u>Запрос добавления:</u> в копию таблицы добавить список книг по экономике из главной таблицы
3	Железнодорожное расписание	<u>Запрос удаления:</u> удалить список поездов, прибывающих в город Минск до 6 утра <u>Запрос обновления:</u> в связи с ремонтными работами на железной дороге время отправления из города Гомеля переносится на 2 часа позже <u>Запрос добавления:</u> в копию таблицы добавить список поездов, прибывающих в город Минск до 6 утра, из главной таблицы

№ варианта	Наименование базы данных	Запросы к базе данных
4	Протокол соревнований	<p><u>Запрос удаления:</u> удалить список всех участников соревнований по плаванию</p> <p><u>Запрос обновления:</u> изменить название команды «Спартак» на «Динамо»</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список всех участников соревнований по плаванию из главной таблицы</p>
5	Справочник автоинспектора	<p><u>Запрос удаления:</u> удалить список всех автомобилей марки Ауди</p> <p><u>Запрос обновления:</u> изменить цвет машин с черного на синий, которые были выпущены в 1999 году</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список автомобилей марки Ауди из главной таблицы</p>
6	Сведения о поставках	<p><u>Запрос удаления:</u> удалить список всех непродовольственных товаров</p> <p><u>Запрос обновления:</u> увеличить цену единицы товара в 1,7 раз</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список непродовольственных товаров из главной таблицы</p>
7	Телефонный справочник	<p><u>Запрос удаления:</u> удалить список всех абонентов, номера телефонов которых начинаются с 52</p> <p><u>Запрос обновления:</u> изменить название Советского района на Железнодорожный</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список абонентов, номера телефонов которых начинаются с 52, из главной таблицы</p>
8	Дневник метеонаблюдений	<p><u>Запрос удаления:</u> удалить все записи, в которых время снятия показаний с 22:00 до 6:00</p> <p><u>Запрос обновления:</u> изменить все записи, в которых время снятия показаний 21:00 на 22:00</p> <p><u>Запрос добавления:</u> в копию таблицы добавить записи из главной таблицы, в которых снятие показаний производилось с 22:00 до 6:00</p>
9	Отдел кадров	<p><u>Запрос удаления:</u> удалить список всех сотрудников бухгалтерии</p> <p><u>Запрос обновления:</u> увеличить стаж работы на 1 год</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список сотрудников бухгалтерии из главной таблицы</p>

№ варианта	Наименование базы данных	Запросы к базе данных
10	Карточка абитуриента	<p><u>Запрос удаления:</u> удалить список всех абитуриентов, получивших на экзамене два балла</p> <p><u>Запрос обновления:</u> изменить название специальности «Коммерческая деятельность» на «Коммерческая деятельность на транспорте»</p> <p><u>Запрос добавления:</u> в копию таблицы добавить список абитуриентов из главной таблицы, получивших на экзамене два балла</p>

### Задание № 3

Создать многотабличные запросы:

№ варианта	Наименование базы данных	Многотабличные запросы
1	Ведомость для начисления зарплаты	<p><u>Простой запрос:</u> вывести на экран список всех работников инструментального цеха с указанием Ф.И.О. сотрудника и его паспортных данных</p> <p><u>Составной запрос:</u> вывести на экран список работников сборочного цеха, которые были на больничном более трех дней и домашний телефон которых начинается на 52</p>
2	Библиотечный каталог	<p><u>Простой запрос:</u> вывести на экран список всех книг, являющихся романами, с указанием Ф.И.О. автора, года и места издания</p> <p><u>Составной запрос:</u> получить на экране список всех книг по экономике (на основании шифра ББК), стоимость которых колеблется от 1500 до 5000 рублей и место издания Минск</p>
3	Железнодорожное расписание	<p><u>Простой запрос:</u> вывести на экран список всех скорых поездов (№ поезда от 1 до 99) с указанием Ф.И.О. начальника поезда</p> <p><u>Составной запрос:</u> вывести на экран список пассажирских поездов (№ поезда от 100), прибывающих в Гомель в промежутке времени от 12:00 до 24:00 с количеством вагонов более 10</p>
4	Протокол соревнований	<p><u>Простой запрос:</u> вывести на экран список всех победителей с указанием Ф.И.О. участника, вида состязаний и даты рождения</p> <p><u>Составной запрос:</u> вывести на экран список всех участников состязаний, проводившихся в манеже «Динамо», вес которых не превышает 60 кг</p>

№ варианта	Наименование базы данных	Многотабличные запросы
5	Справочник автоинспектора	<p><u>Простой запрос:</u> вывести на экран список всех автомобилей красного цвета. На экране должны быть отображены марка автомобиля, № автомобиля и Ф.И.О. владельца</p> <p><u>Составной запрос:</u> вывести на экран список всех автомобилей, зарегистрированных во втором полугодии 1998 года с пробегом не более 7500 км, владельцы которых проживают на пр. Ленина</p>
6	Сведения о поставках	<p><u>Простой запрос:</u> вывести на экран список всех промышленных товаров с указанием наименования товара, объема поставки и предприятия-производителя</p> <p><u>Составной запрос:</u> вывести на экран список всех товаров продовольственной категории, поставленных в первом квартале 2000 года и срок реализации которых заканчивается 1 июля 2000 года</p>
7	Телефонный справочник	<p><u>Простой запрос:</u> вывести на экран список абонентов с указанием Ф.И.О., адреса, номера телефонов, которых начинаются с цифры 7 и имеют функцию АОН</p> <p><u>Составной запрос:</u> вывести на экран список всех абонентов Центрального района, получивших номер телефона во втором квартале 1997 года, подключивших модем</p>
8	Дневник метеонаблюдений	<p><u>Простой запрос:</u> вывести на экран список дней, в которые температура превышала нулевую отметку с указанием времени снятия показания</p> <p><u>Составной запрос:</u> вывести на экран список записей из метеодневника, которые были произведены в первой половине дня нечетных дней</p>
9	Отдел кадров	<p><u>Простой запрос:</u> вывести на экран список сотрудников, чей табельный номер превышает 100 с указанием паспортных данных</p> <p><u>Составной запрос:</u> вывести на экран список сотрудников, поступивших на работу с 01.07.1990 по 01.07.1995 в должности инженера, телефон которых начинается с цифры 4</p>
10	Карточка абитуриента	<p><u>Простой запрос:</u> вывести список абитуриентов с указанием Ф.И.О. и оценок по экзаменам, подавших заявления до июля, с указанием среднего балла аттестата</p> <p><u>Составной запрос:</u> вывести список абитуриентов, получивших неудовлетворительные оценки хотя бы по одному из предметов, со средним баллом аттестата &gt;4</p>

## 8 ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА БАЗЫ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

### 8.1 Проектирование форм и работа с ними

Понятие **«форма»** в базе данных означает структурированное окно, экранное поле либо независимый элемент интерфейса с заранее установленными областями для ввода либо изменения информации; визуальный шаблон, который упорядочивает предоставляемые формой данные, что позволяет лучше их организовывать и просматривать. Форма дает возможность вводить и просматривать информацию базы данных.

Как объект базы данных форма используется для ввода и корректировки данных в таблицу в качестве специального окна диалога для выбора; в виде кнопочной формы для открытия других форм либо отчетов. В форме может быть представлена информация из нескольких связанных таблиц. Представление данных в формах может быть различным: поля располагаются столбцом или таблицей.

Источником данных для формы являются записи таблицы либо запроса. Связь между формой и записями таблицы осуществляют элементы управления (графические объекты). Информация, которая не связана с таблицей или запросом, сохраняется в макете формы.

Форма облегчает восприятие информации, это гибкий способ представления информации.

### 8.2 Простейший способ создания формы

*Последовательность действий:*

- 1) открыть окно базы данных (источник данных);
- 2) в окне базы данных выбрать вкладку **«Таблицы»**;
- 3) выбрать таблицу – источник данных, для которой следует построить форму;
- 4) дать команду **«Вставка»** → **«Автоформа»** либо нажать кнопку **«Новый объект»** на панели инструментов;

5) выбрать с раскрытого списка опцию **«Автоформа»**. В результате на экране появится готовая форма (рисунок 8.1), в которую включены все поля таблицы. Имена полей расположены столбцом, а справа от названия каждого поля отражаются значения каждого поля первой записи таблицы-источника.

Для каждой записи базы данных отводится отдельная страница. Для перемещения по записям внизу формы есть поле **«Запись»**, в которое вводится номер той записи, которую желательно просмотреть. Внизу формы расположены кнопки перемещения по записям: переход на первую запись; переход назад на одну запись; переход вперед на одну запись; переход на последнюю запись; переход на чистую страницу, куда можно ввести новую запись.

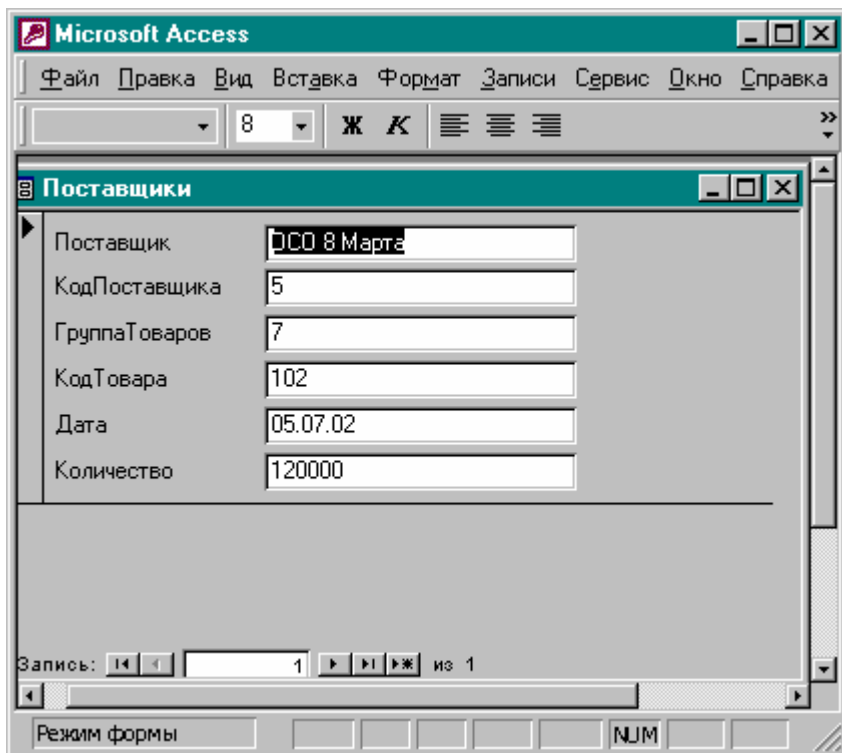


Рисунок 8.1 – Форма, созданная в режиме «Автоформа»

*Переход от формы в режим таблицы.* Для перехода с режима формы в режим таблицы следует подать команду «**Вид**» → «**Режим таблицы**» либо на панели инструментов нажать кнопку «Новый объект» и из списка выбрать пункт «Режим таблицы».

Для добавления в таблицу очертаний сетки подать команду «**Формат**» → «**Ячейки**» → [«Вид сетки»] → выбрать нужные компоненты диалогового окна.

**Способы создания форм.** Для манипуляций с объектами таблицы можно эффективно применять в меню команду «Формат» и ее опции. Кроме простейшего способа Создания формы MS Access реализует и такие способы (виды форм):

- а) Конструктор формы – с его помощью самостоятельно можно создать собственные формы с заданными параметрами;

б) Мастер формы – создаем форму на основе выбора полей, предложенных Мастером;

в) автоформа: в столбец – автоматически создается форма в один или несколько столбцов;

г) автоформа: ленточная – автоматически создается ленточная форма (поля расположены столбцами);

д) автоформа: табличная – автоматически создается табличная форма;

е) диаграмма – позволяет создать форму с диаграммой;

ж) сводная таблица – позволяет создать форму со сводной таблицей Excel.

MS Access создает формы любым перечисленным способом *в такой начальной последовательности:*

1) открыть окно базы данных;

2) перейти на вкладку «Формы» в окне базы данных;

3) в окне диалога нажать кнопку «Создать» (рисунок 8.2);

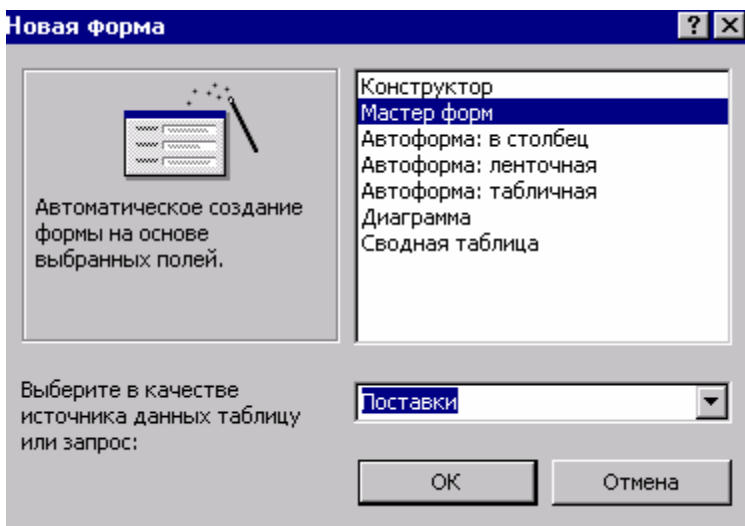


Рисунок 8.2 – Окно диалога «Новая форма»

4) выбрать способ создания формы;

5) нажать кнопку «ОК».

*Замечание.* Окно диалога «Новая форма» можно также открыть командой «Вставка» → «Форма» либо кнопкой «Новый объект» на панели инструментов с выбором опций «Новая форма».



### 8.3 Создание формы с помощью Мастера

Можно создавать формы, различные как по стилю оформления, так и по содержанию полей; форму можно создать на основе нескольких таблиц. Имена полей задаются пользователем.

Для подключения Мастера к работе нужно в окне диалога «Новая форма» выбрать «Мастер форм» и нажать «ОК». После запуска Мастера форм пользователь должен отвечать на вопросы каждого текущего экрана Мастера и нажать кнопку «Далее» для перехода к следующему окну. С текущего окна Мастера при необходимости можно вернуться к предыдущему при помощи кнопки «Назад».

После запуска Мастера форм на экране появляется окно диалога «Создание форм», в котором определяем поля создаваемой формы (рисунок 8.3).

Форма создается для таблицы базы данных. Эту таблицу нужно выбрать со списка «Таблицы/ Запросы». После этого в списке «Доступные поля» появится перечень всех полей выбранной таблицы, из которого выбираем и при помощи мыши и кнопок переноса переносим в список «Выбранные поля» имена полей, которые необходимы в форме.

Создание форм

Выберите поля для формы.  
Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы  
Таблица: Поставщики

Доступные поля:

- Поставщик
- КодПоставщика
- ГруппаТоваров
- КодТовара
- Дата
- Количество

Выбранные поля:

Отмена < Назад Далее > Готово

Рисунок 8.3 – Создание формы с помощью Мастера

Мастер форм может создавать формы на основе полей нескольких таблиц: выбрать нужные поля сначала со списка первой, а потом второй таблицы и перенести в список **«Выбранное поле»**.

После завершения формирования списка полей формы переходим в следующее окно путем нажатия кнопки **«Далее»**.

Если для создания формы использовались поля нескольких связанных таблиц, то необходимо выбрать тип представления данных, указать тип создаваемой формы: подчиненные или связанные (установить необходимые опции) — нажать кнопку **«Далее»** для перехода в следующее окно.

В следующем окне диалога Мастера задаем внешний вид формы, которая создается для одной таблицы (выбрать опцию «ленточный» либо «табличный»), и нажимаем кнопку **«Далее»**.

Следующее окно содержит инструментарий выбора стиля формы (глобус, ель, камень, облако, обычный, сумерки и т. д.). После выбора стиля переходим (кнопка **«Далее»**) в последнее окно диалога: вводим имя созданной формы, а если форма создавалась для связанных таблиц, то вводим имя и подчиненной формы, затем нужно указать дальнейшие действия (открытие формы для просмотра и ввода данных или изменение макета формы). Для завершения создания формы нажать кнопку **«Готово»**; на экране появится форма, созданная Мастером с вашим участием. По мере необходимости ее можно модифицировать с помощью Конструктора форм.

#### **8.4 Создание формы с помощью Конструктора**

Конструктор форм предназначен для создания форм любой сложности, позволяя располагать их в необходимом виде.

*Последовательность действий при создании новой формы в окне Конструктора форм:*

- 1) в окне базы данных выбрать вкладку **«Формы»**;
- 2) нажать кнопку **«Создать»** → [«Новая форма»];
- 3) выбрать из списка способы создания формы **«Конструктор»**;
- 4) выбрать таблицы – источник данных из списка **«Выберите в качестве источника данных таблицу»**. Источником данных для построения формы может быть только одна таблица. Если же необходимо использовать данные не одной, а нескольких таблиц, то сначала следует создать запрос на основе этих таблиц, и этот запрос будет источником данных для формы;
- 5) нажать **«ОК»**. Откроется окно Конструктора форм с отображением **«Область данных»**, в которой располагаются имена полей данных (рисунк 8.4).

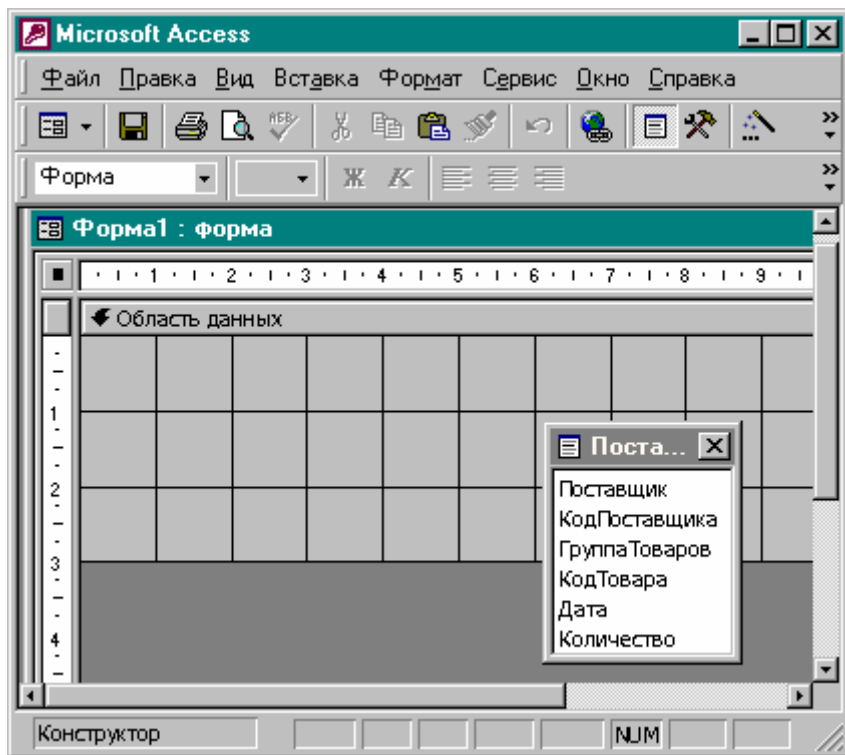


Рисунок 8.4 – Окно Конструктора формы

**Области формы.** После открытия окно Конструктора формы содержит, как минимум, одну область – **«Область данных»** (см. рисунок 8.4). Всего Конструктор формы может включать несколько областей (рисунок 8.5), которые добавляются командой **«Вид» → «Заголовок/Примечание формы»** и **«Вид» → «Колонтитулы»**.

*Что располагается, записывается в каждой области Конструктора форм?*

В области **«Заголовок формы»** можно записывать название формы, названия (подписи) столбцов, кнопки для открытия подчиненных форм и др. Все перечисленное отражается в верхней части экрана, а при печати — внизу первой страницы.

Область **«Верхний колонтитул»** может удерживать заголовок формы, заголовки столбцов, графические объекты и другую информацию, которая располагается в верхней части каждой страницы только при печати формы и в режиме предварительного просмотра.

**«Область данных»** удерживает записи имен полей (данные) с таблиц.

В области «**Нижний колонтитул**» можно вставлять дату, номер страницы и другую информацию для расположения в нижней части каждой страницы.

Область «**Примечание формы**» может Удерживать кнопки и инструкции по применению формы. Содержит элементы управления для подсчета итогов в каждом столбце. Информация отражается в нижней части экрана, а при печати выводится на последней странице после «**Области данных**».

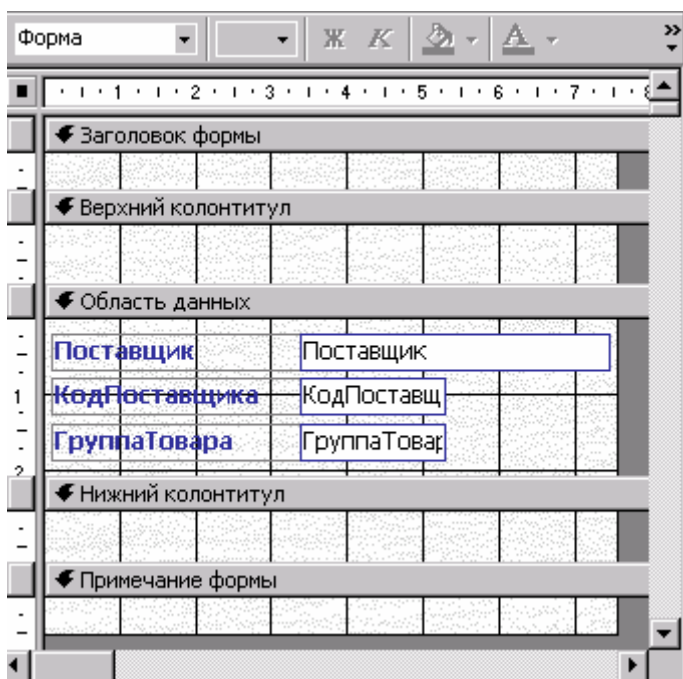


Рисунок 8.5 – Окно Конструктора формы со всеми областями

Для изменения размера любой области в окне Конструктора указатель мыши нужно установить на верхнюю часть границы между областями; курсор должен принять форму двунаправленной стрелки с прямоугольником посередине; не отпуская кнопку мыши, переместить границу области;

б) подать команду «**Формат**» → «**Автоформат**» либо кнопкой «Автоформат» открыть окно диалога; выбрать из списка стиль (глобус, ель, сумерки...) и нажать кнопку «**ОК**»;

7) установить размер формы: указатель мыши перевести в правый нижний угол формы (курсор примет вид двунаправленной стрелки), нажать кнопку мыши и перемещать до получения нужного размера формы;

- 8) командой **«Вид»** → **«Свойства»** открыть окно свойств формы;
- 9) выбрать вкладку **«Макет»** и задать размеры окна форм (редактированные свойства формы);
- 10) для установки связей формы с таблицей выбрать вкладку **«Данные»** → **«Источник записей»** → [ ], а также источник записей, для которых создается форма (нажать кнопку раскрытия списка таблиц). Закрыть окно клавишей **«X»**;
- 11) на стандартной панели инструментов после ввода источника записей стала доступной кнопка **«Список полей»**; нажатие на нее открывает панель **«Список полей»**, откуда можно брать нужные поля для расположения их в форме.

Самый простой способ добавления текстового окна с полями таблицы — дать команду **«Вид»** → **«Список полей»**, после чего следует перетянуть нужное поле в нужное место;

12) при желании запретить пользователю менять данные в форме, т. е. предусмотреть только просмотр, а не редактирование формы, установив в окне свойств формы **«Разрешить изменение»** значение **«Нет»**;

13) при желании запретить добавлять новые записи в таблицу либо удалять уже существующие. Для этого установить в окне свойства форм **«Разрешить добавление»** значение **«Нет»**. После этого становятся недоступными команды **«Вставка»** → **«Запись»**; **«Запись»** → **«Ввод данных»**; **«Правка»** → **«Удалить запись»**.

**Панель элементов.** Предназначена для расположения объектов (элементов управления) в форме при помощи кнопок, которые входят в состав панели. Назначение и функции каждой кнопки объясняются после установки указателя мыши на кнопку. Размещение текстовой информации на экранной форме осуществляется с помощью кнопки **«Надпись»** на панели элементов. Функция этой кнопки — введение наименования в форму нового поля. В панели элементов каждая пиктограмма представляет собой стилизованное отображение управляющего элемента (оформление), который можно встроить в форму (рисунок 8.6).

#### **Расположение текста в форме. Ввод заголовка формы:**

- 1) открыть форму в режиме Конструктора;
- 2) на панели элементов выбрать указателем мыши кнопку **«Надпись»**;
- 3) перенести курсор мыши в область **«Заголовок формы»** на место расположения текста; нажатой левой кнопкой мыши создать поле (прямоугольник) для ввода текста; отпустить кнопку, ввести текст и нажать клавишу **«Enter»**. Поле с текстом будет выделенным; теперь его можно форматировать (выбрать шрифт, цвет и т. д.).

**Вычисление итогов и применение выражений.** Для представления в форме результатов вычисления по соответствующему выражению нужно

создать вычисляемый элемент управления (поле). Это можно сделать без применения программы «**Построитель выражений**» и с ее помощью.

Подключение итогового поля без применения программы «**Построитель выражений**»:

- 1) в режиме Конструктора открыть форму;
- 2) на панели инструментов выбрать кнопку «**Поле**»;
- 3) в форме выбрать место для расположения нового элемента управления (поля) и щелкнуть мышью. В форме появится свободный элемент управления (два прямоугольника: «поле» и «свободный»);

4) курсор поставить в поле (прямоугольник «свободный») и ввести выражение для вычисления итога, например:

= [Количество] \* [Цена],

а в левый прямоугольник (прямоугольник «поле») — текст, например: «Всего». После этого закрыть форму.

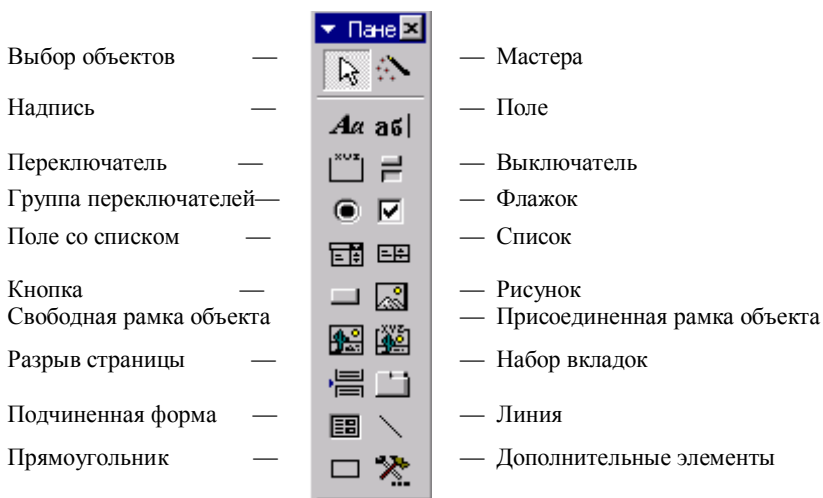


Рисунок 8.6 – Панель элементов

**Подключение итогового поля с помощью программы «Построитель выражения»** (см. подразд. 7.4 и рисунок 7.4):

- 1) запустить «**Построитель выражений**»;
- 2) в левой нижней области Построителя выражений выбрать папку с нужным элементом;
- 3) в нижней средней области дважды щелкнуть мышью элемент для его вставки в поле выражения либо выбрать тип элементов;

4) при выборе типа в нижней левой области значения будут отражаться в нижней левой области. Для вставки значения в поле выражения дважды щелкнуть мышью по нему;

5) вставить нужный оператор в выражение: указатель мыши поставить в необходимую позицию поля выражения и выбрать кнопку со знаком операции;

6) после записи выражения нажать кнопку **«ОК»**. Созданное выражение будет скопировано в ту позицию, с которой был вызван **«Построитель выражений»**.

Для ввода пояснительного текста перед итогом (в левом прямоугольнике свободного элемента управления) нужно выделить левый прямоугольник: в контекстном меню выбрать пункт «Свойства», перейти на «Макет» и в строчку «Подпись» ввести текст подписи к выражению, например «Всего», закрыть окно свойств. В форме появятся пояснительный текст и выражение.

**Расположение полей ввода в форме.** В форму, как правило, нужно добавлять поля разных типов. Поле ввода считается наиболее простым типом. Для его расположения в форме нужно:

1) выбрать кнопку **«Поле»** на панели элементов;

2) щелкнуть мышью на то место, где желательно видеть поле ввода; в форме появится связанный объект, который состоит из поля ввода и его надписи;

3) выделить поле ввода и открыть для него окно свойств;

4) для связывания созданного поля с полем таблицы выбрать во вкладке **«Данные»** свойство **«Данные»**; кнопкой раскрыть список и выбрать из списка полей открытой таблицы то поле, которое нужно добавить в форму. Для связывания поля с выражением нажать кнопку **«Построить»**; в окне диалога **«Построитель выражений»** записывается выражение;

5) задать с помощью панели инструментов либо в окне свойств поля тип, шрифт, размер, цвет шрифта, цвет рамки, тип и цвет фона и другие нужные параметры;

6) если информацию с создаваемого поля нужно сделать доступной только для чтения, то значение свойств **«Доступ»** установить в **«Нет»**;

7) для создания краткого пояснения полю во вкладке **«Другие»** выбрать **«Всплывающая подсказка»**;

8) для определения значения поля по умолчанию нужно задать свойство **«Значение по умолчанию»**;

9) выделить название поля и открыть для него окно свойств;

10) чтобы задать надпись текста, выбрать во вкладке **«Макет»** свойство **«Подпись»** и ввести в поле ввода свойств текст надписи к полю;

11) с помощью панели **«Форматирование»** либо окна свойств задать для надписи тип шрифта, размер, цвет шрифта, цвет рамки, тип, цвет фона и другие параметры.

**Скрытые поля.** Используя окно свойств поля, можно сделать непросматриваемыми и поле ввода, и надпись к нему (не просматриваемыми в режиме просмотра формы): для свойства «**Вывод на экран**» вкладки «**Макет**» задать значение «Нет».

**Отображение сообщения в строке состояния.** Как уже отмечалось, информация в окне Конструктора таблицы столбца «**Описание**» в режиме просмотра формы выводится в строке состояния того поля, в котором находится курсор. Можно заменить сообщение в форме командой: «Другие»→ «Текст строки состояния» →[]→ в окно свойств поля ввести текст, который будет появляться в строке состояния.

**Создание кнопок управления.** Для выполнения определенных действий в формах можно использовать кнопки, например кнопку для открытия другой формы. Всего в MS Access предусмотрена возможность создания более 30 кнопок. Для создания кнопок предназначен Мастер создания кнопок.

*Последовательность создания кнопки:*

- 1) на панели элементов установить режим Мастера кнопок;
- 2) на панели выбрать элемент «Кнопка»;
- 3) указателем мыши выбрать место в форме, где должна быть кнопка, и щелкнуть мышью—откроется диалоговое окно «Создание кнопок»;
- 4) в первом окне диалога в двух списках – «Категории» и «Действия» – выбрать из набора действий нужное и нажать кнопку «Далее»;
- 5) во втором окне диалога выбрать форму, которая будет открываться при нажатии кнопки, и нажать кнопку «Далее»;
- 6) в третьем окне диалога выбрать тип отображаемой информации на кнопке (текст или рисунок). Если выберем текстовую информацию, то в поле ввода рядом с опцией «Текст» ввести текст, отображаемый на кнопке. Для расположения графической информации выбрать опцию «Рисунок», нажать кнопку «Обзор» и выбрать графическое изображение; нажать кнопку «Далее» для перехода к следующему шагу;
- 7) ввести имя кнопки и нажать кнопку «Готово» – кнопка создана.

**Добавление рисунка или другого объекта в форму.** Объекты, созданные в других приложениях, можно добавить к форме для лучшего восприятия информации. Присоединенный объект сохраняется в таблице. При переходе к новой записи в форме отображается другой объект (например, фотография сотрудника).

**Вставка свободного объекта.** При переходе к новой записи объект не меняется. Свободный объект является частью структуры формы (например, герб «Погоня», созданный в Power Point).

Для вставки в экранные формы графических отображений следует:

- 1) щелкнуть мышью на панели элементов кнопку «**Рисунок**»;
- 2) указателем мыши открыть окно диалога «**Выбор рисунка**», где в списке «**Тип файла**» выбрать тип изображения, а в строке «**Имя файла**» – имя вставляемого рисунка и нажать на кнопку «**ОК**».



**Размещение в форме объекта типа OLE.** Классическим примером является размещение в форме фотографий сотрудников.

*Последовательность действий:*

1) на панели элементов выбрать кнопку **«Присоединенная рамка объекта»;**

2) щелкнуть мышью место, где желательно добавить поле; не отпуская кнопки мыши, переместить указатель так, чтобы по диагонали получилась рамка необходимого размера;

3) открыть окно свойств созданного объекта (**«Вид»** → **«Свойства»** → [Свойства]);

4) выбрать вкладку **«Данные»** для связи созданного поля с полем таблицы; в поле ввода свойств раскрыть список и из списка полей раскрытой таблицы выбрать поле типа OLE-фото;

5) в режиме формы просмотреть форму, при необходимости вернуться в режим Конструктора и увеличить или уменьшить размер поля.

**Добавление в форму текущей даты и времени:**

1) открыть форму в режиме Конструктора форм;

2) подать команду **«Вставка»** → **«Дата и время»;**

3) установить соответственно флажки **«Формат даты»** и **«Формат времени»** и выбрать нужный формат.

## 8.5 Главные и подчиненные формы

Главная и подчиненная формы имеют одинаковые поля связей. *Главной* называется форма, с которой связываются другие формы. Форма, которая вставляется в другую форму, называется *подчиненной*. Она позволяет выводить в нее данные нескольких записей с таблиц или запросов, которые связаны с одной текущей записью в главной форме. Например, в главной форме отображена запись группы товаров **«Краски»**, а в подчиненной форме — все записи, которые входят в эту группу товаров. Подчиненная форма обеспечивает вывод данных с таблиц либо запросов, которые находятся в отношении **«один-ко-многим»**. Главная форма удерживает данные в отношении со стороны **«один»**, а подчиненные — со стороны **«многих»**.

Рекомендуется сначала создать подчиненную форму, а затем главную и разместить в нее подчиненную. Подчиненные формы рекомендуется создавать на основе не данных таблиц, а запросов. Для ускорения выполнения подчиненных форм в них нужно включать минимально необходимые поля. Главная форма может удерживать несколько подчиненных форм.

Для обеспечения связи с главной формой все поля подчиненной формы рекомендуется индексировать.

Соотношения главной формы и подчиненных форм позволяют отображать отношения между таблицами и получать доступ к связям с разных таблиц.

**Типы подчиненных форм.** Подчиненная форма может иметь вид таблиц или **форм** (простых либо ленточных),

**Создание подчиненных форм.** Прежде всего нужно проверить наличие необходимых связей между таблицами.

***Способы создания подчиненных форм:***

- одновременно с главной формой;
- создание подчиненной формы, а затем ее добавление в существующую (главную) форму;
- добавление одной существующей **формы** в другую существующую для создания подчиненной формы.

*Одновременное создание главной и подчиненной форм:*

1) на панели инструментов нажать **кнопку «Новый объект: автоформа»** и выбрать опцию **«Форма»** →[«Новая форма»];

2) выбрать **«Мастер форм»** и нажать клавишу **«ОК»**;

3) в первом окне выбрать из списка таблицу либо запрос, которые включаются в создаваемую форму;

4) выбрать из этой таблицы поля, которые включаются в создаваемую форму;

5) в этом же окне диалога Мастера выбрать из списка другую таблицу (запрос), которая в отношении «один-ко-многим» находится со стороны «многие»;

6) выбрать поля из другой таблицы для включения в форму;

7) нажать кнопку **«Далее»** и выбрать параметр **«Подчиненные формы»**.

В диалоге выполнить Инструкции Мастера и нажать кнопку **«Готово»**. Будут созданы две формы: главная с элементом управления и подчиненная.

*Создание подчиненной формы, а затем ее добавление в существующую (главную) форму:*

1) открыть главную (существующую форму) в режиме Конструктора форм;

2) убедиться в том, что на панели инструментов нажата Кнопка **«Мастера»**;

3) на панели элементов нажать кнопку **«Подчиненная форма/отчет»** и перевести указатель мыши на основную форму (он примет форму нажатой кнопки со знаком «плюс»);

4) в основной форме указателем мыши выбрать место, куда будет помещена подчиненная форма. Мышью выбрать левый верхний угол подчиненной формы и при нажатой кнопке мыши создать Прямоугольник — место для подчиненной формы. Отпустить кнопку мыши; в прямоугольнике формы появится свободный элемент управления;

5) щелкнуть правой кнопкой мыши по созданному прямоугольнику и выбрать параметр **«Свойства»**; откроется окно **«Подчиненная форма/отчет: Внедренный»**;

6) выбрать параметр «**Данные**» и в столбце «**Объект-источник**» раскрыть список имен форм; выбрать подчиненную форму; закрыть окно «**Подчиненная форма**». В прямоугольнике появится имя подчиненной формы;

7) правой кнопкой мыши отметить элемент (квадратик с именем «**Внедренный**») и выбрать пункт «**Свойства**» – [«Надпись»];

8) перейти на вкладку «**Макет**» и в столбце «**Подпись**» ввести название подчиненной формы. Закрыть окно. Возвратиться в главную форму, в которой находится подчиненная форма с присвоенным именем;

9) командой «**Вид**» – «**Режим формы**» перейти в режим Просмотра созданной формы;

10) закрыть форму.

*Добавление одной существующей формы в другую существующую для создания подчиненной формы:*

1) в режиме Конструктора открыть существующую форму, которую будем использовать как главную (со стороны «один» в отношении «один-многим»);

2) проверить, нажата ли на панели элементов кнопка «**Мастера**».

3) клавишей «**F11**» переключиться в окно базы данных. Если окно базы данных закрывает окно главной формы, его нужно сместить для получения к нему доступа;

4) из окна базы данных мышью переместить форму либо таблицу, которая будет подчиненной, в главную форму. В форме добавится элемент управления подчиненной формой (прямоугольник с именем подчиненной формы);

5) перейти в режим просмотра созданной формы командой «**Вид**»→ «**Режим формы**» либо кнопкой на панели инструментов.

## 8.6 Списки и поля со списками в форме

В ряде случаев удобнее выбрать значение из списка, чем вводить конкретное значение с клавиатуры по памяти.

Форму можно рассматривать как окно доступа к базе данных. В формах можно организовать поиск записи по значению, которое выбрано из списка. Для поиска в формах записи по выбранному из списка значению применяются элементы управления «окно списка» и «поле со списком».

*Список* (окно списка) – это элемент управления по выбору значений из списка. Это может быть только список или список и надпись (пояснение). Список состоит из строк, включающих один или более столбцов, которые могут иметь заголовки. В окне списка отображается список значений, которые ранее определены для отбора. Значения для списка можно внести в таблицу либо, наоборот, взять их из нее.

*Поле со списком* – это элемент управления, состоящий из двух элементов – поля и раскрывающегося списка. В это поле можно вводить значения с клавиатуры либо выбирать значения в раскрывающемся списке. На рисунке 8.7 приведен список «Стоимость» и значения списка 900000, 800000, 30000.

**Последовательность создания списка и поля со списком.** В качестве примера создадим форму под именем «Со списком», в которой будет отражаться основная информация из таблицы «**Поставки**», а для каждого кода товара его цена будет выбираться из списка цен. Последний «берется» из таблицы «**Ценник**». Связь между таблицами «**Ценник**» и «**Поставки**» установлена «один-ко-многим».

*Последовательность действий:*

- 1) открыть форму в режиме Конструктора;
- 2) на панели элементов проверить, нажата ли кнопка «Мастера» (если нет, то нажать);
- 3) на панели элементов выбрать кнопку «Список» или «Поле со списком» (в нашем примере – «Список»);
- 4) указатель мыши перевести в то место формы, где будут расположены список либо поле со списком, и щелкнуть кнопкой мыши; откроется окно Мастера «Создание списков»;
- 5) в первом окне диалога Мастера выбрать таблицу со значениями, которые будут содержать список либо поле со списком (в нашем примере это таблица «**Ценник**»). Нажать кнопку «Далее», перейти во второе окно диалога с Мастером;
- 6) выбрать поле со значениями, которое будет содержать список (в нашем примере это поле «**Цена**»). Перейти в третье окно диалога с Мастером и выполнить инструкции по выбору нужных параметров. В последнем окне диалога нажать кнопку «**Готово**» для вывода списка либо поля со списком в режиме Конструктора;
- 7) для обеспечения работы списка на панели инструментов нажать кнопку «Свойства», появится окно диалога, в нем выбрать вкладку «**Данные**» и в строке «**Данные**» указать информацию, которая включается в список (в нашем примере это поле «**Код Товара**»);
- 8) закрыть окно «Список», закрыть окно Конструктора формы, вернуться в окно базы данных;
- 9) открыть форму «Со списком» в окне базы данных (см. рисунок 8.7)

**Синхронизация записей в двух формах.** Можно создать с помощью Мастера форм две связанные формы и одновременно просматривать связанные записи. Главная и подчиненная формы связаны между собой таким образом, что в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме. При переходе по записям в первой форме автоматически выводятся связанные с ней записи в другой форме, например, просмотр списка клиентов в одной форме автоматически может сопровождаться просмотром видов товаров, которые купил клиент.

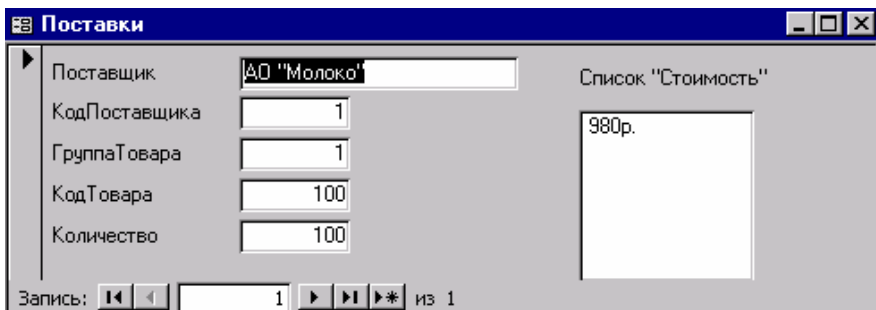


Рисунок 8.7 – Форма со списком «Стоимость»

Для синхронизации записей в двух формах Мастером форм автоматически создается кнопка для открытия второй формы.

Для создания двух связанных таблиц Мастером форм на втором шаге (окно диалога) нужно выбрать поля из двух таблиц. На третьем шаге установить переключатель «Связанные формы». В одной форме автоматически создается кнопка для вызова другой формы и синхронизации записей между связанными формами.

## 8.7 Редактирование записей в форме

Как уже отмечалось, формы создаются для автоматизации процесса заполнения базы данных и их редактирования. Перед началом редактирования нужно открыть сохраненную форму (в окне базы данных дважды щелкнуть мышью по пиктограмме формы либо выбрать форму и нажать кнопку «Открыть»). В форме отображается значение одной записи. С помощью формы выполняются операции добавления записей в таблицу, переноса и копирования, удаления и сортировки записей.

*Добавление записей в таблицу с помощью формы:*

- 1) в окне базы данных открыть форму (выбрать форму и нажать кнопку «Открыть»);
- 2) перейти к пустой записи (обозначается «\*»), которая всегда находится в конце таблицы;
- 3) активизировать команду «Запись» → «Ввод данных» (либо «Ctrl +»);
- 4) ввести данные в пустые поля формы. Клавишей «Tab» осуществляется переход к следующей записи.

*Перенос и копирование записей:*

- 1) маркировать записи, предназначенные для переноса;
- 2) дать команду меню «Правка» – «Копировать».

*Удаление записей:*

- 1) маркировать запись в форме;

2) дать команду «Правка» → «Удалить запись» → [] → подтвердить удаление→«ОК».

*Поиск записи по ее номеру:*

1) открыть форму в режиме формы;  
2) указатель мыши установить на номер в поле номера записи и дважды щелкнуть мышью;

3) ввести номер необходимой записи и нажать **«Enter»**.

**Сортировка записей в форме.** Для сортировки по одному полю:

1) установить курсор на поле-ключ;  
2) нажать одну из кнопок на панели инструментов для сортировки по убыванию либо по возрастанию значения ключа (можно также дать команду в меню «Запись» – «Сортировать».

*Для сортировки по нескольким полям:*

1) в меню дать команду «Запись» → **«Фильтр»** → **«Расширенная фильтрация / Сортировка»**→ []→ и в строке «Поле» указать поле, по содержанию которого будет идти сортировка, а в поле «Сортировка» — тип сортировки.

Во время работы с формами следует использовать команды меню «Вид», «Правка», «Запись». Команда **«Вид»** позволяет выбрать режим просмотра: форма, конструктор, таблица, таблица подчиненной формы.

**Клавиши, которые используются в формах:**

Home – курсор установится в первое поле текущей записи;

End – курсор установится в последнее поле текущей записи;

Ctrl – курсор установится в первое поле первой записи;

Tab→Enter – курсор установится в следующее поле;

Shift-Tab – курсор установится в предыдущее поле;

Ctrl-Tab – в многотабличных формах: выход из подчиненной формы и переход в следующее поле главной формы;

Ctrl-Shift-Home – переход в первое поле главной формы.

## 8.8 Представление формы

На экране форма может быть создана в проектном, рабочем или табличном виде.

Переход от представления одного вида к другому происходит по команде **«Вид»** либо кнопкой выбора представления на стандартной панели управления.

*Проектный вид:* команда **«Вид»** → **«Конструктор форм»**. Форма на экране будет представлена в режиме проектирования.

*Рабочий вид:* команда **«Вид»** → **«Форма»**. В этом режиме можно выполнять операции редактирования: добавлять, удалять, активизировать данные в таблице, в которой разработана форма.

*Табличный вид:* команда **«Вид»** → **«Таблица»**. На экран одновременно выводится не одна, а несколько записей.

**Предварительный просмотр формы.** Перед печатью рекомендуется произвести предварительный просмотр формы: перейти в рабочее представление («Вид» → «Форма»), а затем дать команду «Файл» → «Предварительный просмотр». Форму в этом режиме нужно увеличить: указатель мыши поставить на форму (примет вид Луны) и щелкнуть — форма будет увеличена; повторное нажатие возвращает отображение формы.

**Открытие формы.** В окне базы данных выбрать «Формы», имя формы и нажать кнопку «Открыть».

**Распечатка формы.** Распечатать форму можно из окна базы данных или из любого режима работы с формой.

Предварительное выравнивание страниц и установка соответствующих параметров выполняются по команде «Файл» → «Параметры страницы» → в окне диалога установить параметры печати и нажать кнопку «ОК». Установленные параметры будут сохранены вместе с формой; при следующей ее распечатке устанавливать параметры не нужно.

Для распечатки формы дать команду «Файл» → «Печать» → [Печать] → выбрать нужную опцию («Выбранная запись» либо «Все») → «ОК».

### *Контрольные задания*

#### Задание № 1

Для копии базы данных, созданной в разделе 5, выполнить следующие формы:

№ варианта	Наименование базы данных	Задания на создание форм
1	Ведомость для начисления заработной платы	1 С помощью мастера форм создать форму, содержащую поля «Цех», «Ф.И.О.» и «К выплате» 2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных 3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Фонд заработной платы», в котором будет выведена общая сумма выплаченных денег. В создаваемой форме должны использоваться различные виды оформления и графические объекты
2	Библиотечный каталог	1 С помощью мастера форм создать форму, содержащую поля «Шифр ББК», «Ф.И.О. автора» и «Наименование» 2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей 3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Общая стоимость книг библиотечного фонда», в котором будет выведено общее количество книг. В создаваемой форме должны использоваться различные виды оформления и графические объекты

№ варианта	Наименование базы данных	Задания на создание форм
3	Железнодорожное расписание	<p>1 С помощью мастера форм создать форму, содержащую поля «№ поезда», «Пункт отправления» и «Пункт назначения»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Самый продолжительный рейс», в котором будет выведено максимальное время следования поезда. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
4	Протокол соревнований	<p>1 С помощью мастера форм создать форму, содержащую поля «Вид состязаний», «Ф.И.О. участника» и «Результат»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Самый молодой участник», в котором будет выведен год рождения самого молодого участника состязаний. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
5	Справочник автотранспортного инспектора	<p>1 С помощью мастера форм создать форму, содержащую поля «Марка автомобиля», «Год выпуска» и «Пробег в км»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Максимальный пробег», в котором будет выведена цифра максимального расстояния, пройденного автомобилем. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
6	Сведения о поставках	<p>1 С помощью мастера форм создать форму, содержащую поля «Категория товара», «Наименование товара» и «Объем поставки»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Средняя стоимость товаров», в котором будет выведено среднее значение стоимости товаров. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>



№ варианта	Наименование базы данных	Задания на создание форм
7	Телефонный справочник	<p>1 С помощью мастера форм создать форму, содержащую поля «Ф.И.О. абонента», «№ телефона» и «Категория»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Количество льготников», в котором будет выведено общее число льготников. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
8	Дневник метеонаблюдений	<p>1 С помощью мастера форм создать форму, содержащую поля «Дата», «Температура» и «Осадки»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора создать форму, содержащую все поля базы данных, предусмотрев наличие вычисляемых полей «Максимальная зимняя температура» и «Минимальная летняя температура», в которых будут просчитаны и выведены соответствующие данные. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
9	Отдел кадров	<p>1 С помощью мастера форм создать форму, содержащую поля «Отдел», «Ф.И.О. сотрудника» и «Дата поступления на работу»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Штат», в котором будет выведено общее количество сотрудников. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
10	Карточка абитуриента	<p>1 С помощью мастера форм создать форму, содержащую поля «Ф.И.О.», «Оценка на экзамене по русс. яз.» и «Оценка на экзамене по математике»</p> <p>2 Создать три автоформы (табличную, ленточную, в столбец), предусмотрев отображение на экране всех полей базы данных</p> <p>3 С помощью конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемого поля «Наивысший средний балл», в котором будет выведено максимальное значение по полю «Средний балл». В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>

## Задание № 2

На основании копии базы данных выполнить следующие задания:

№ варианта	Наименование базы данных	Задания на создание форм
1	Ведомость для начисления заработной платы	С помощью конструктора форм создать две формы, содержащие все поля базы данных. Поле «Цех» в одной из форм должно заполняться с помощью переключателей, а в другой с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
2	Библиотечный каталог	С помощью конструктора форм создать две формы, содержащие все поля базы данных. В одной из форм поле «Жанр» должно заполняться с помощью переключателей, а в другой с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
3	Железнодорожное расписание	С помощью конструктора форм создать две формы, содержащие все поля базы данных. В одной из форм поле «Категория» должно заполняться с помощью флажков, а в другой с помощью переключателей. В создаваемой форме должны использоваться различные виды оформления и графические объекты
4	Протокол соревнований	С помощью конструктора форм создать две формы, содержащие все поля базы данных. В одной форме предусмотреть возможность заполнения поля «Вид состязаний» с помощью переключателей, а в другой форме поле «Результат» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
5	Справочник автоинспектора	С помощью конструктора форм создать две формы, содержащие все поля базы данных. В одной форме предусмотреть возможность заполнения поля «Цвет» с помощью переключателей, а в другой форме поле «Марка автомобиля» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
6	Сведения о поставках	С помощью конструктора форм создать две формы, содержащие все поля базы данных. Поле «Категория товара» в одной из форм должно заполняться с помощью переключателей, а в другой с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты

№ варианта	Наименование базы данных	Задания на создание форм
7	Телефонный справочник	С помощью конструктора форм создать две формы, содержащие все поля базы данных. В одной форме предусмотреть возможность заполнения поля «Район» с помощью переключателей, а в другой форме поле «Категория» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
8	Дневник метеонаблюдений	С помощью конструктора форм создать две формы, содержащие все поля базы данных, в одной форме предусмотреть возможность заполнения поля «Пора года» с помощью переключателей, а в другой форме поле «Направление ветра» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
9	Отдел кадров	С помощью конструктора форм создать две формы, содержащие все поля базы данных, в одной форме предусмотреть возможность заполнения поля «Отдел» с помощью переключателей, а в другой форме поле «Должность» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты
10	Карточка абитуриента	С помощью конструктора форм создать форму, содержащую все поля базы данных. В ней предусмотреть возможность заполнения поля «Отделение» с помощью переключателей, поле «Специальность» должно заполняться с помощью флажков. В создаваемой форме должны использоваться различные виды оформления и графические объекты

### Задание № 3

На основании копии базы данных выполнить следующие задания:

№ варианта	Наименование базы данных	Задания по созданию форм
1	Ведомость для начисления зарплаты	С помощью конструктора форм создать форму, содержащую поля из таблицы «Ведомость для начисления зарплаты», и подчиненную форму, содержащую поля из связанной таблицы «Карточка отдела кадров»
2	Библиотечный каталог	С помощью конструктора форм создать форму, содержащую поля из таблицы «Библиотечный каталог», и подчиненную форму, содержащую поля из связанной таблицы «Сопроводительная карточка книги»

№ варианта	Наименование базы данных	Задания по созданию форм
3	Железнодорожное расписание	С помощью конструктора форм создать форму, содержащую поля из таблицы «Железнодорожное расписание», и подчиненную форму, содержащую поля из связанной таблицы «Маршрутный лист»
4	Протокол соревнований	С помощью конструктора форм создать форму, содержащую поля из таблицы «Протокол соревнований», и подчиненную форму, содержащую поля из связанной таблицы «Карточка участника»
5	Справочник автоинспектора	С помощью конструктора форм создать форму, содержащую поля из таблицы «Справочник автоинспектора», и подчиненную форму, содержащую поля из связанной таблицы «Владелец автомобиля»
6	Сведения о поставках	С помощью конструктора форм создать форму, содержащую поля из таблицы «Сведения о поставках», и подчиненную форму, содержащую поля из связанной таблицы «Сопроводительный лист товара»
7	Телефонный справочник	С помощью конструктора форм создать форму, содержащую поля из таблицы «Телефонный справочник», и подчиненную форму, содержащую поля из связанной таблицы «Дополнительная информация»
8	Дневник метеонаблюдений	С помощью конструктора форм создать форму, содержащую поля из таблицы «Дневник метеонаблюдений», и подчиненную форму, содержащую поля из связанной таблицы «Дополнительная информация»
9	Отдел кадров	С помощью конструктора форм создать форму, содержащую поля из таблицы «Отдел кадров», и подчиненную форму, содержащую поля из связанной таблицы «Карточка сотрудника»
10	Карточка абитуриента	С помощью конструктора форм создать форму, содержащую поля из таблицы «Карточка абитуриента», и подчиненную форму, содержащую поля из связанной таблицы «Данные абитуриента»

## 8.9 Понятие отчета как объекта базы данных

Отчеты представляют собой наилучшее средство представления информации из базы данных в виде печатного документа. Отчеты представляют широкие возможности для группировки и вычисления промежуточных и общих итогов для большого набора данных. Каждая группа данных и итоги могут быть представлены отдельно. MS Access для создания отчета предоставляет следующие возможности:

- Для иерархического представления данных можно задать до 10 уровней группировки.

- Для каждой из групп можно задать отдельные заголовки и примечания.
- Можно производить сложные вычисления не только внутри группы или набора строк, но также использовать в вычислениях данные нескольких групп.
- В дополнение к верхнему и нижнему колонтитулам, которые печатаются на каждой странице, можно задать заголовок и примечание для всего отчета.
- Так же, как и в форме, можно вставлять в любой раздел отчета рисунки или диаграммы, внедрять подчиненные отчеты и формы.

*Отчет* — это форматированное представление данных, которые выводятся на экран, в печать или файл. Отчет может создаваться как в табличной, так и в свободной форме.

Перед непосредственным созданием отчета необходимо определить:

- цель отчета;
- информацию, которая будет включена в отчет, и из какой таблицы-источника она взята;
- форму отчета (табличная, свободная, почтовые наклейки);
- участвуют ли сортировка и группировка данных и по каким полям-ключам; какие вычислительные поля присутствуют в отчете.

Необходимо также спроектировать отчет на бумаге перед компьютерным проектированием.

Простейший способ создания отчета:

- 1) открыть окно базы данных;
- 2) выбрать вкладку «**Отчеты**»;
- 3) выбрать имя таблицы-источника данных;
- 4) дать команду «**Вставка**» → «**Авто отчет**» или на панели инструментов нажать кнопку «**Новый объект**».

В результате на экране появится отчет, в который будут включены все поля таблицы, расположенные вертикально.

Кроме того, отчет можно создать, используя:

- 1) «**Конструктор отчетов**» — самостоятельно создается отчет с заданными свойствами;
- 2) «**Мастер отчетов**» — на основе выбранных полей в диалоге;
- 3) «**Автоотчет: в столбец**» — поля в отчете располагаются в один или несколько столбцов;
- 4) «**Автоотчет: ленточный**» — создается автоматически ленточный отчет;
- 5) «**Диаграмма**» — информация отображается в виде диаграммы;
- 6) «**Почтовые наклейки**» — отчеты для почтовых наклеек.

## 8.10 Создание отчета с помощью Мастера

Создание отчета с помощью Мастера отчетов выполняется в диалоге пользователя с системой и сводится к выбору таблиц-источников, определения списка полей отчета и порядка их расположения. Мастер не создает дополнительных полей, которые не удерживаются в таблицах-источниках.

*Последовательность создания отчета:*

1) открыть окно базы данных;  
2) перейти на вкладку **«Отчеты»**;  
3) нажать кнопку **«Создать»**; откроется окно диалога **«Новый отчет»** со списком способов создания отчетов;

4) выбрать **«Мастер отчетов»**, а в строке **«Выберите в качестве источника данных»** — имя таблицы-источника и нажать кнопку **«ОК»**; открывается окно **«Создание отчетов»**;

5) определить поле создаваемого отчета:

- вызвать кнопку раскрытия списка **«Таблицы/Запросы»**;
- выбрать из списка таблиц ту базу данных, на основе которой создается отчет;
- из списка **«Доступные поля»**, т. е. из перечня всех полей выбранной таблицы, с помощью клавиш навигации перенести нужные поля в зону **«Выбранные поля»**.

*Замечание.* Если отчет создается на основе информации нескольких таблиц-источников, то действия, указанные в п. 5, повторяются для каждого источника;

6) нажать кнопку **«Далее»** для перехода к следующему технологическому шагу; определить, нужно ли сортировать сведения по какому-нибудь полю. Выбрать поле-ключ сортирования. Поля, по которым будет сортироваться информация, располагаются в верхней части правого списка в отдельной рамке (выделенной синим цветом). Access предлагает вариант группировки сведений, который можно принять или отвергнуть, создав при помощи кнопок окна диалога собственный вариант. Для изменения интервала сортировки нажать кнопку **«Группировка»**;

7) для перехода к следующему окну диалога нажать кнопку **«Далее»**; задается порядок сортировки записей отчета;

8) при наличии в отчете итоговых полей щелкнуть кнопку **«Итоги...»**; откроется окно диалога **«Итоги»**;

9) установить нужные флажки, которые определяют вид вычислений (Sum — подсчет значений поля во всех записях; Avg — расчет среднего значения; Min — расчет минимального значения; Max — расчет максимального значения) итоговых полей для итогов по группам записей и общего итога по всему отчету;

10) установить переключатель режима отражения информации в отчете в одно из положений: **«Данные и итоги»** или **«Только итоги»**;

11) нажать кнопку **«ОК»** и вернуться в окно сортировки;

12) нажать кнопку **«Далее»** и выбрать в окне диалога вид макета отчета;

13) нажав кнопку **«Далее»**, перейти в окно для выбора стиля оформления отчета;

14) нажав кнопку **«Далее»**, перейти в последнее окно; задать имя созданного отчета и выбрать один из двух вариантов дальнейшей работы (просмотр отчета или изменение структуры отчета);

15) нажать кнопку **«Готово»**. Мастер отчетов с вашим участием спроектировал отчет в нужном для вас виде.

**Просмотр и распечатка отчета.** Три способа *просмотра* созданного на экране отчета:

- 1) команда основного меню «**Файл**» → «**Предварительный просмотр**»;
- 2) в контекстном меню команда «**Просмотр**»;
- 3) на стандартной панели инструментов окно базы данных нажать кнопку «**Предварительный просмотр**».

Два способа распечатки отчета:

- а) командой «**Файл**» → «**Печать**» → в окне диалога задать параметры печати;
- б) нажать кнопку «**Печать**» на панели инструментов.

### 8.11 Создание отчета с помощью Конструктора

Мастер отчетов создает отчеты в пределах своих возможностей, которые заложены в этой программе, и конечный вид отчета в некоторых деталях может не удовлетворять пользователя. Конструктор отчета позволяет пользователю спроектировать эскиз отчета с учетом всех запланированных особенностей. В эскизе указываются принципы сортировки и группировки сведений и другие компоненты оформления отчета. В отчет могут добавляться OLE – объекты, рисунки, линии и прямоугольники. Поля отчета могут непосредственно соответствовать полям исходной таблицы или являются результатом вычислений, произведенных над значениями полей.

**Открытие окна Конструктора отчета.** Для создания нового отчета необходимо:

- 1) выбрать в окне базы данных вкладку «**Отчеты**»;
- 2) щелкнуть кнопку «**Создать**»; откроется окно «**Новый отчет**» с перечнем способов создания отчетов;
- 3) выбрать способ создания отчетов «**Конструктор**», а также источник сведений (таблицу-источник) в развернутом списке. Источником сведений для отчета может быть только одна таблица. Если данные для отчета находятся в нескольких таблицах, нужно сначала создать запрос на основе этих таблиц, который будет являться источником данных;
- 4) щелкнуть кнопку «**ОК**». На экране откроется окно Конструктора отчетов (пустой отчет-эскиз) – область для рисования нужной структуры отчета. Окно Конструктора отчетов имеет ту же структуру и области, что и окно Конструктора форм (рисунок 8.8).

При открытии Конструктора отчетов в его окне отражены три области: «**Верхний колонтитул**», «**Область данных**» и «**Нижний колонтитул**».

При необходимости в окно Конструктора можно вывести и другие области: команда «**Вид**» → «**Колонтитул**» и «**Вид**» → «**Заголовок / Примечание отчета**».

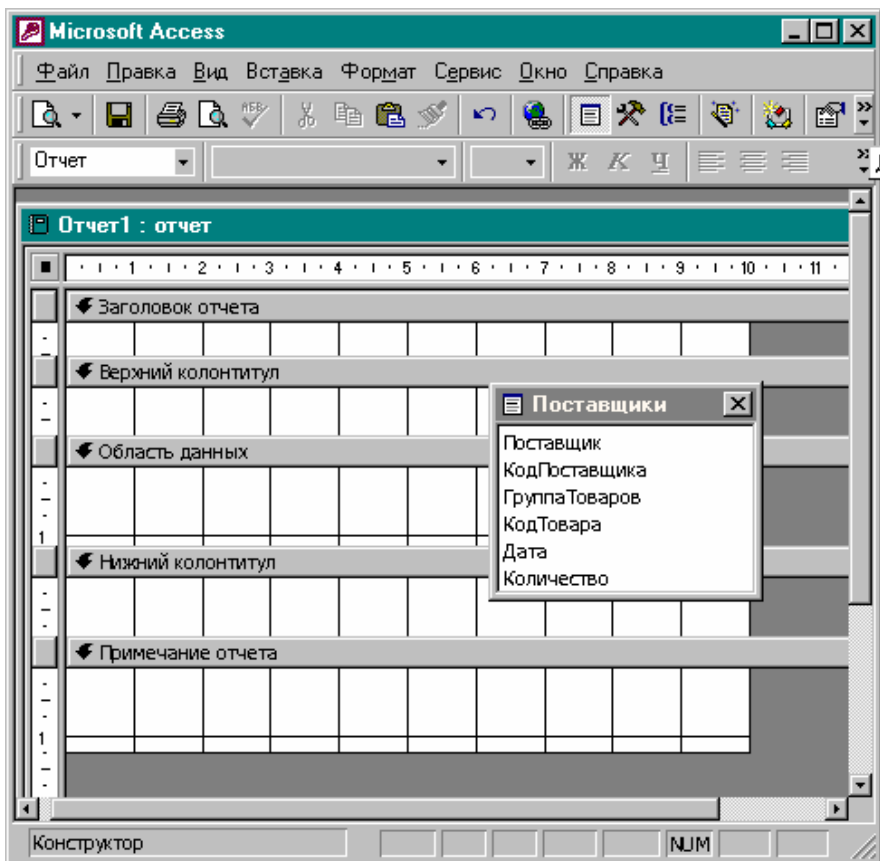


Рисунок 8.8 – Окно Конструктора отчетов

**Характеристика разделов окна Конструктора отчетов.** Структурно вся информация в отчете (окне Конструктора) разбивается на разделы, предназначенные для отражения конкретной информации. При печати отчетов разделы размещаются на страницах в определенном порядке.

Разделы отчетов отражаются в окне Конструктора и могут содержать соответствующую информацию. Приведем перечень разделов отчетов и их назначение.

Область «*Заголовок отчета*» печатается один раз в начале отчетов на первой странице перед верхним колонтитулом и может удерживать герб фирмы, название отчетов, дату распечатки отчетов и другую подобную информацию.

Область «*Верхний колонтитул*» печатается в верхней части каждой страницы отчета и может удерживать заголовки столбцов.



Область «*Область данных*» удерживает данные для отчетов (указываются имена полей). Источником данных является значение записей.

Область «*Нижний колонтитул*» печатается в нижней части каждой страницы отчета и может удерживать номера страниц.

Область «*Примечание отчета*» печатается один раз в конце отчета и может удерживать подсчеты над всеми записями отчета.

В отчете могут вычисляться итоги по группам (признакам группировки) записей. В этой ситуации в окне Конструктора появляются дополнительные области «*Заголовок группы*» (печатается в начале каждой новой группы записей), и область «*Примечание группы*» — после последней записи каждой группы выводится результат вычисления. Эта область может удерживать имя группы и другие данные конкретной группы;

5) добавить нужные разделы для установки сортировки и группировки информации внутри отчета;

6) для сохранения созданного отчета дать команду «**Файл**» → «**Сохранить**».

*Для модификации ранее созданного отчета нужно:*

- в окне базы данных перейти на вкладку «**Отчеты**»;
- выбрать имя отчета, предназначенного для модификации;
- в окне базы данных щелкнуть кнопку «**Конструктор**»; на экране откроется окно Конструктора отчетов с отчетом.

Любые действия при проектировании отчета можно отменить командой «**Правка**» → «**Отменить**».

На любой стадии проектирования отчета можно увидеть отчет с фактическими данными, нажав кнопку «Предварительный просмотр» на панели инструментов.

## 8.12 Пример создания отчета

Предположим, имеется база данных в таблице с именем «**Цех**» следующей структуры:

Цех	План	Факт
2	345	300
2	123	130
3	500	450
3	700	750
4	432	430
4	550	560
4	3	4

Требуется спроектировать отчет следующей структуры на основе сведений таблицы «Цех»:

Цех	План	Файл	Отклонение
2	345	300	-45
2	123	130	7
По цеху:	468	480	36
3	500	450	-50
3	700	780	50
По цеху:	1200	1200	0
4	432	430	-2
4	550	560	10
4	3	4	1
По цеху:	965	994	-9
Всего:	2653	2624	29

*Последовательность действий* (один из вариантов):

1) в окне базы данных перейти на вкладку «Отчет» → «Создать [«Новый отчет»] → из списка выбрать имя таблицы-источника данных и вызвать «Конструктор» → «ОК»; на экране появится окно Конструктора отчетов «Отчет: отчет 1» (пустое окно);

2) командой «Вид» → «Заголовок/ Примечание отчета» добавить в окно Конструктора области «Заголовок» и «Примечание»;

3) командой «Вид» → «Список полей» открыть маленькое окошко с отражением списка полей таблицы-источника;

4) ввод имени полей в «Область данных»: выбрать в этом окошке нужные поля и перетянуть их имена в «Область данных» Конструктора отчетов (рисунков 8.9). Для этого нужно щелкнуть мышью имя поля и тянуть его; появятся два квадратика с маркером; перетянув объект на нужное место в области данных, щелкнуть его мышью для закрепления. Имя поля как связанный элемент управления привязан к исходной таблице, поэтому представляется двумя квадратами; левый квадрат нужно удалить (щелкнуть его и нажать на клавишу «Del»).

Активизировать правый квадрат и поставить его на нужное место: привести курсор на маркер перемещения элемента (большой черный квадрат в правом верхнем углу), курсор должен принять форму указательного пальца.

Поставить курсор в позицию двунаправленной стрелки ↔ и уменьшить квадрат до нужного размера.

Курсор в виде открытой ладони перемещает сразу оба квадрата, а в форме указательного пальца – только один квадрат.

Таким же способом переместить в «Область данных» остальные нужные имена полей. Получим:



Рисунок 8.9— Область данных окна Конструктора отчетов

5) ввод названия граф «шапки» отчета: в область «Верхний колонтитул» ввести название граф «шапки» отчета. Для этого вызвать на экран (если этого не сделано раньше) «Панель элементов» командой «Вид» -> «Панель элементов»; щелкнуть на панели элементов кнопку «Надпись» [A] и отпустить ее. Перевести курсор в область «Верхний колонтитул». Курсор примет вид [+A], переместить его в нужное место и щелкнуть по квадрату, появится курсор — место для ввода данных, куда и ввести название графы (в нашем примере – «Цех»). Ввод заканчивается нажатием клавиши «Enter».

Квадратик в левом верхнем углу имеет маркер перемещения элемента, а маленькие квадратики посередине сторон – это маркеры изменения размеров элементов. С помощью маркеров редактируем элемент.

После записи всех граф отчета закрыть окно Конструктора отчетов кнопкой «X» и вернуться в окно базы данных. Результат работы можно посмотреть, нажав кнопку «Просмотр»;

6) **ввод названия отчета:** в область «Заголовок отчета» ввести название отчета, нажав кнопку «Надпись» [A] (см. рисунок 8.6), как описано в п. 5;

7) **итоговая строчка отчета:** оформить итоговую строку отчета в области «Примечание отчета». В нашем примере это слово «Всего» и подсчеты по графам. Текст «Всего» записывается с помощью элемента «Надпись».

**Ввод формул для подсчета по графам.** Для определения общей суммы по графе «План» нужно записать формулу  

$$= \text{Sum}(\text{План})$$

*Последовательность действий:* на панели элементов нажать кнопку «Поле» [aB/], отпустить мышь и ввести курсор в область «Примечание отчета». Там появится квадратик кнопки «Поле» со знаком «плюс»; щелкнуть мышью; появится свободный элемент управления (два квадратика с маркерами), например:

#### Поле 10      Свободный

Удалить левый квадратик, а в правый записать формулу подсчета:  $= \text{Sum}(\text{План})$ .

Таким же способом записать формулы для граф «Факт» и «Отклонение»:  
 $= \text{Sum}(\text{Факт})$  и  $= \text{Sum}(\text{План}) - [\text{Факт}]$ ;

8) **сохранение эскиза отчета для просмотра и модификации:** команда «Файл» → «Сохранить» → [«Сохранение объекта»] → установить фла-

жок либо в «текущей базе данных», либо «во внешнем файле или базе данных», ввести имя отчета и нажать кнопку «ОК». Вернуться в окно Конструктора отчета. Закрывать его. Вернуться в окно базы данных, откуда можно посмотреть отчет (кнопка «Просмотр»), либо в окно Конструктора отчета (кнопка «Конструктор»);

9) **промежуточные итоги. Группировка данных.** Алгоритм для подсчета по группам — нужно вести подсчет (итог) по каждому цеху и вывести его после смены значения ключевого поля («цех»):

- открыть отчет в режиме Конструктора;
- объединить записи в группы по ключевому признаку командой: «Вид» → «Сортировка и группировка» → [«Сортировка и группировка»]. Можно задавать до 10 уровней групп. Каждую группу можно распечатывать с новой страницы; можно распечатывать текст конкретной страницы;
- в столбик «Поле/выражение» ввести имя поля-ключа для группы записей с одинаковым значением поля-ключа. Щелкнуть кнопку раскрытия списка и выбрать поле-ключ группирования, например «Цех» (выделить и щелкнуть); в области «Поле» появится имя поля «Цех», в поле «Порядок сортировки» появится значение «По возрастанию» (рисунок 8.10);
- перейти в нижнюю часть окна «Свойства группы» и в строчке «Заголовок группы» нажать в правой части строчки (там, где слово «Нет»): появится кнопка раскрытия списка; раскрыть список и выбрать «Да»; в окне Конструктора отчета появится дополнительная область «Заголовок группы», а в окне «Сортировка и группировка» в области выбора поля появится значок группировки [>] Цех. В строчке «Примечание группы» установить значение «Да». В окне Конструктора появится область «Примечание группы»:

«По цеху» Цех = Sum (План);

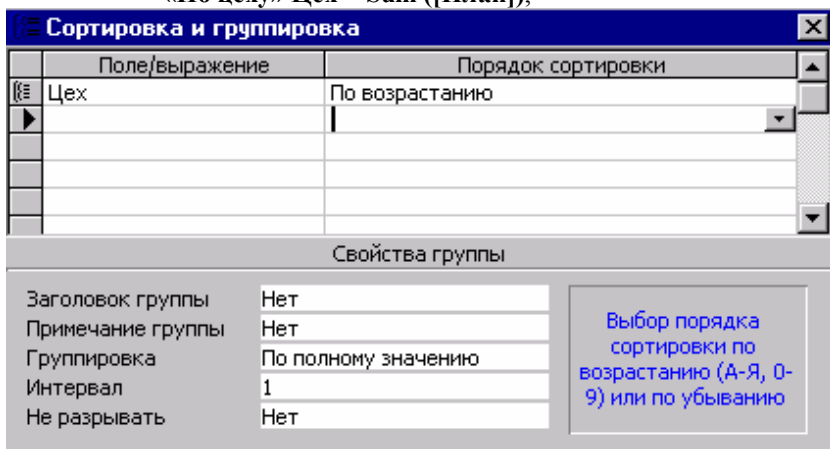


Рисунок 8.10 – Окно задания параметров сортировки и группировки в отчете

- закрыть окно диалога «Сортировка и группировка»; вернуться в окно Конструктора отчетов. В структуре отчета появятся две дополнительные области: «Заголовок группы» и «Примечание группы» (рисунок 8. 11);

The screenshot shows a report designer window titled "Отчет за месяц : отчет". The report structure is as follows:

Заголовок отчета															
Отчет за месяц															
Верхний колонтитул															
цех				план	факт										отклонение
Заголовок группы 'цех'															
цех															
Область данных															
				план	факт										отклонение
Примечание группы 'цех'															
="Итого для " & "цех" = " & " & {цех} & " (" & Count(*) & " " & ИИ(Count(*)=1,"запись","записей") & ")"															
Sum				=Sum({план})	=Sum({факт})										=Sum({план}-{факт})
Нижний колонтитул															
=Nоч()															
="Страница " & {Page} & " из " & {Pages}															
Примечание отчета															
ИТОГО				=Sum({план})	=Sum({факт})										=Sum({план}-{факт})

Рисунок 8.11 – Спроектированный отчет с помощью Конструктора

- в область «Заголовок группы» ввести название группы, например «По цеху», и имя поля «Цех».

Для суммирования по группам (по цеху) в области «Заголовок группы» создать свободный элемент-формулу

$$= \text{Sum} (\{\text{План}\});$$

- в области «Примечание группы» создать свободный элемент и ввести значение «По цеху: «Цех» и выражения для подсчета итогов:

$$= \text{Sum} (\{\text{План}\}) = \text{Sum} (\{\text{Факт}\}) = \text{Sum}[\{\text{План}\}] - \{\text{Факт}\}.$$

В целом итоги работы Конструктора отчетов показаны на рисунке 8.11.

В какой области окна Конструктора записывать выражения для вычисления результатов?

- 1) вычисления для каждой отдельной записи: вычисляемое поле располагается в разделе «Область данных»;
- 2) вычисления над группой записей: вычисляемое поле размещается в дополнительном разделе «Примечание группы»;
- 3) вычисления внизу каждой страницы: вычисляемое поле размещается в области «Нижний колонтитул»;
- 4) вычисления в конце всех записей отчета: вычисляемое поле размещается в области «Примечание отчета».

Объекты, которые помещаются в отчете (имя поля, текст, рисунок), называются *элементами управления*, которые бывают:

а) *связанными* – присоединяются к конкретному полю таблицы данных (запроса);

б) *несвязанными* – не присоединяются к таблице данных (запроса);

в) *вычисляемыми* – отражают результаты расчетов на основе данных исходной таблицы (запроса).

### 8.13 Изменение вида отчета

**Размещение даты распечатки отчета.** Дату можно разместить в заголовке отчета или в нижнем колонтитуле. Для этого необходимо создать свободный элемент управления:

1) на панели элементов (см. рисунок 6.6) выбрать инструмент «Поле» [ab/];

2) установить указатель мыши на место в области «**Верхний** колонтитул», где должна быть дата, и щелкнуть мышью. В отчете появится связанный объект – поле ввода и надпись к нему;

3) выделить надпись связанного поля и удалить его клавишей «Del»;

4) выделить поле ввода, открыть контекстное меню и выбрать «**Свойство**»; открыть окно свойств;

5) перейти на вкладку «**Данные**» и в строке «**Данные**» нажать кнопку «**Построитель**». Если ее нет, щелкнуть в конце строки, и она появится; откроется окно диалога «**Построитель выражений**»;

6) открыть раздел «**Встроенные функции**» в левом списке нижней части окна (папка «**Функции**»). В среднем списке выбрать позицию «**Дата/время**», а в правом списке дважды щелкнуть мышью функцию Now. В верхней части окна Построителя выражений появится выражение для значения поля:

= Now();

7) нажать кнопку «ОК», вернуться в окно свойств, где в строке «**Данные**» будет содержаться составленное выражение;

8) перейти на вкладку «**Макет**» и из раскрытого списка выбрать подходящий **формат** отражения даты и времени;

9) закрыть окно свойств; в области заголовка будет отражаться **выражение**

Now [ ];

10) для ввода **надписи**, например «Текущая дата», выделить элемент «**надпись**» (левый квадратик) и в контекстном меню выбрать команду «**Свойства**». В открывшемся окне **перейти** на вкладку «**Макет**» и в строке «**Подпись**» ввести надпись. Закрыть окно свойств.

При распечатке отчета в нем будет печататься дата.

**Размещение номера страницы отчета:**

1) в окне Конструктора отчета дать команду **«Вставка»** → **«Номер страницы»**; откроется **окно** диалога **«Номер страницы»**;

2) в окне диалога **«Новая страница»** задать формат и размещение **номера** страницы на верхнем или нижнем колонтитуле, а также характер выравнивания (по центру и т. д.);

4) нажать кнопку **«ОК»**.

**Удаление сортировки или группы:**

1) **из окна** Конструктора отчетов дать команду **«Вид»** → **«Сортировка и группировка»** → [ ];

2) выделить поля **сортировки** (групп) путем нажатия на область выбора поля и на клавишу **«Del»** → подтвердить **«Да»**.

**Нумерация записей в области данных отчета** (в группе или по всему отчету):

1) в область данных нужно добавить несвязанное поле;

2) открыть окно свойств этого элемента управления;

3) в поле ввода свойств **«Данные»** ввести **«=1»**;

4) для свойства **«Сумма с накоплением»** установить значение **«Для всего»** или **«Для группы»**.

**Расположение информации в области данных в две колонки, а для остальных разделов – в одну:**

1) сделать отчет в один столбец;

2) для превращения информации отчета в два столбца дать команду **«Файл»** → **«Макет страницы»** → [«Макет страницы»] → выбрать вкладку **«Макет»**;

3) в поле ввода **«Число элементов по горизонтали»** указать количество столбцов – 2;

4) в рамке **«Размер элементов»** уменьшить значение показателя **«Ширина»**;

5) выбрать альбомную или книжную ориентацию элементов (вставить нужную опцию);

6) нажать кнопку **«ОК»** – в окне просмотра можно просмотреть отчет.

**Создание почтовых наклеек** на основе данных таблиц при помощи специального Мастера:

1) в окне базы данных перейти на вкладку **«Отчет»** → [«Новый отчет»];

2) нажать на кнопку **«Создать»** → [«Новый отчет»];

3) из списка способов создания отчета выбрать Мастер **«Почтовые наклейки»**;

4) в поле **«Выбор таблицы/Запроса»** выбрать таблицу или запрос с данными, которые будут использоваться при заполнении почтовой наклейки;

5) нажать кнопку **«ОК»** → [ ]. В окнах диалога необходимо выполнять инструкции Мастера создания наклеек;

6) в первом окне диалога выбрать размер наклейки и нажать кнопку **«Далее»**;

7) во втором окне установить шрифт и цвет текста наклейки и нажать кнопку **«Далее»**;

8) в третьем окне описать внешний вид почтовой наклейки и нажать кнопку **«Далее»**;

9) в четвертом окне выделить поля сортирования; перенести в поле сортировки нужные поля и нажать кнопку **«Далее»**;

10) в пятом окне ввести имя отчета и выбрать дальнейший вариант работы. Вводим имя отчета и нажимаем кнопку **«Готово»**: на экране появится отчет-наклейка. В окне базы данных на вкладке **«Отчеты»** появится имя сделанной наклейки.

**Распечатка наклеек.** Выполнить дополнительную настройку размера страниц с учетом типа принтера. **Перед** созданием отчета с наклейками нужно выделить по умолчанию принтер и специальные размеры страницы.

#### **Определение стандартного принтера и размера страницы:**

1) в меню дать команду **«Пуск»** → **«Настройка»** → **«Панель управления»** → [];

2) два раза щелкнуть мышью по значку **«Принтер»** и выбрать значок нужного принтера → пункт **«Файл»** → **«Использовать по умолчанию»**;

3) пункт **«Файл»** → **«Свойства»** выбрать вкладку **«Бумага»**;

4) в поле **«Размер бумаги»** выбрать значок **«Другие»** → [**«Определяемые пользователем размеры»**];

5) выбрать единицу измерения и ввести значение ширины и высоты страницы наклеек;

6) нажать кнопку **«ОК»**; произойдет сохранение настройки в окне **«Определяемые пользователем размеры»**;

7) нажать кнопку **«ОК»**. Настройка в окне **«Свойства: <принтер>»**. После определения размера бумаги можно приступить к созданию отчета наклеек для конкретного типа принтера.

### **8.14 Главный и подчиненные отчеты**

В созданный отчет можно вставить (добавить) другие отчеты (подчиненные). Можно связать два отчета или больше в один отчет (главный) или вставить некоторую итоговую информацию подчиненного отчета в главный отчет. Главный отчет может быть как присоединенным, так и свободным. Свободный отчет может служить контейнером нескольких несвязанных между собой отчетов, которые необходимо объединять.

Для связывания главного отчета с подотчетом необходимо открыть главный отчет в режиме проектирования и нажать клавишу **«F11»** для ото-



бражения на экране окна базы данных. Из окна базы данных следует перетянуть пиктограмму подотчета в главный отчет, разместив ее там в нужной секции.

После встраивания подчиненного отчета в главный можно выполнить соответствующие манипуляции (сменить размер отчета и др.).

Подчиненный отчет может создаваться в существующем отчете. Между таблицами должна быть установлена связь.

*Последовательность действий:*

- 1) открыть главный отчет в режиме Конструктора;
- 2) нажать на панели элементов кнопку «Мастер»;
- 3) нажать кнопку «Подчиненная форма / отчет» на панели элементов;
- 4) установить курсор в том месте главного отчета, где должен расположиться подчиненный отчет, и щелкнуть мышью;
- 5) выполнить инструкцию в диалоговых окнах Мастера до пункта «Готово».

## 8.15 Обработка отчетов

**Способы сохранения отчета:**

- 1) командой «Файл» → «Сохранить»;
- 2) кнопкой «Сохранить» на панели инструментов;
- 3) путем закрытия окна Конструктора отчета;
- 4) клавишной командой «Ctrl-F4» → «Да». Если отчет сохраняется впервые, необходимо на запрос ввести его имя.

**Способы открытия отчета:**

- 1) в окне базы данных щелкнуть вкладку «Отчет» → выбрать со списка нужный отчет;
- 2) для внесения изменений в проект отчета перейти в режим Конструктора.

**Способы просмотра отчетов:**

- 1) команда «Файл» → «Предварительный просмотр»;
- 2) в контекстном меню команда «Просмотр»;
- 3) на стандартной панели инструментов нажать кнопку «Предварительный просмотр».

Для перемещения по страницам отчета следует использовать кнопки перемещения.

**Способы распечатки отчетов из окна базы данных или окна Конструктора:**

- 1) команда «Файл» → «Печать» → {«Печать»} → задать необходимые параметры печати;

2) на панели инструментов нажать кнопку **«Печать»**; этим способом отчет печатается с текущими установками.

В окне диалога **«Печать»** можно указать номер страницы, которая будет распечатана, количество копий; выбрать режим печати в файл, а также сменить размеры полей страницы — нажать кнопку **«Настройка...»**.

#### **Включение в отчет рисунков (герба, знака фирмы):**

1) открыть отчет в режиме Конструктора. Запустить графический редактор, открыть документ с рисунком, который нужно вставить в отчет;

2) выбрать нужную часть рисунка и дать команду **«Правка»** → **«Копировать»** с меню графического редактора;

3) для возврата в Access нажать клавиши **«Alt-Tab»**. Щелкнуть мышью на любое пустое место и дать команду **«Правка»** → **«Вставить»**; рисунок появится в отчете;

4) поместить рисунок в нужное место отчета. Рисунок можно отредактировать при помощи графического редактора (дважды щелкнуть мышью по рисунку).

Можно вставить рисунок и посредством буфера Обмена Windows — автоматически добавится кадр несвязанного объекта, содержащий этот рисунок.

*В чем отличие между формой и отчетом?*

*При предварительном просмотре* данные с базы выбираются полностью. Например, если отчет формировался на основе запроса, то перед появлением на экране информации выполняется соответствующий запрос.

*При просмотре образа* происходит механический выбор информации с выходных полей (критерии выборки и сортировки игнорируются); дается общее представление о виде отчета.

### **Контрольные задания**

#### **Задание № 1**

№ варианта	Наименование базы данных	Задания на создание отчетов
1	Ведомость для начисления заработной платы	1 С помощью Мастера отчетов создать отчет, содержащий поля «Цех», «Ф.И.О.» и «К выплата» 2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей 3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Фонд заработной платы», в котором будет выведена общая сумма выплаченных денег. В создаваемом отчете должны использоваться различные виды оформления и графические объекты

№ варианта	Наименование базы данных	Задания на создание отчетов
2	Библиотечный каталог	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Шифр ББК», «Ф.И.О. автора» и «Наименование»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Библиотечный фонд», в котором будет выведено общее количество книг. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>
3	Железнодорожное расписание	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «№ поезда», «Пункт отправления» и «Пункт назначения»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Самый продолжительный рейс», в котором будет выведено максимальное время следования поезда. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>
4	Протокол соревнований	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Ф.И.О. участника», «Вид состязания» и «Результат»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Самый молодой участник», в котором будет выведен год рождения самого молодого участника состязаний. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>
5	Справочник автоинспектора	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Марка автомобиля», «Год выпуска» и «Пробег в км»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Максимальный пробег», в котором будет выведен максимальный пробег, пройденный автомобилем. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>

№ варианта	Наименование базы данных	Задания на создание отчетов
6	Сведения о поставках	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Категория товара», «Наименование товара» и «Объем поставки»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора создать отчет, содержащий все поля, добавив в отчет вычисляемое поле «Средняя стоимость», в котором будет выведено среднее значение стоимости товаров. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>
7	Телефонный справочник	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Ф.И.О. абонента», «№ телефона» и «Категория»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Общее количество установок телефонов», в котором будет выведено общее число установок телефонов, содержащихся в базе данных. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>
8	Дневник метеонаблюдений	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Дата», «Температура» и «Осадки»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора форм создать форму, содержащую все поля базы данных, предусмотрев наличие в форме вычисляемых полей «Максимальная зимняя температура» и «Минимальная летняя температура», в которых будут просчитаны и выведены соответствующие данные. В создаваемой форме должны использоваться различные виды оформления и графические объекты</p>
9	Отдел кадров	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Отдел», «Ф.И.О. сотрудника» и «Дата поступления на работу»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Штат», в котором будет выведено общее количество сотрудников. В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>

№ варианта	Наименование базы данных	Задания на создание отчетов
10	Карточка абитуриента	<p>1 С помощью Мастера отчетов создать отчет, содержащий поля «Ф.И.О.», «Оценка на экзамене по русс. яз.» и «Оценка на экзамене по математике»</p> <p>2 Создать два автоотчета (ленточный и в столбец), предусмотрев вывод на печать всех полей</p> <p>3 С помощью Конструктора отчетов создать отчет, содержащий все поля базы данных, предусмотрев наличие в отчете вычисляемого поля «Наивысший средний балл», в котором будет выведено максимальное значение по полю «Средний балл». В создаваемом отчете должны использоваться различные виды оформления и графические объекты</p>

### Задание № 2

№ варианта	Наименование базы данных	Задание на создание отчетов
1	Ведомость для начисления заработной платы	<p>1 Создать сложный отчет, включающий в себя поля «Цех», «Ф.И.О.», «Начисляемый заработок» из основной таблицы и поле «Паспортные данные» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Ведомость для начисления заработной платы», включив в него подчиненный отчет на основе таблицы «Карточка отдела кадров»</p>
2	Библиотечный каталог	<p>1 Создать сложный отчет, включающий в себя поля «Наименование», «Ф.И.О. автора», «Жанр» из основной таблицы и поля «Год издания» и «Место издания» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Библиотечный каталог», включив в него подчиненный отчет на основе таблицы «Сопроводительная карточка книги»</p>
3	Железнодорожное расписание	<p>1 Создать сложный отчет, включающий в себя поля «Номер поезда», «Пункт назначения» и «Время отправления» из основной таблицы и поле «Ф.И.О. начальника поезда» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Железнодорожное расписание», включив в него подчиненный отчет на основе таблицы «Маршрутный лист»</p>
4	Протокол соревнований	<p>1 Создать сложный отчет, включающий в себя поля «Вид состязаний», «Ф.И.О. участника» и «Результат» из основной таблицы и поля «Рост» и «Вес» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Протокол соревнований», включив в него подчиненный отчет на основе таблицы «Карточка участника»</p>

№ варианта	Наименование базы данных	Задание на создание отчетов
5	Справочник автоинспектора	<p>1 Создать сложный отчет, включающий в себя поля «Марка автомобиля», «Год выпуска», «Дата регистрации» из основной таблицы и поля «Ф.И.О. владельца» и «Номер автомобиля» из связанной таблицы</p> <p>2 Создать отчет на основе «Справочник автоинспектора», включив в него подчиненный отчет на основе таблицы «Владелец автомобиля»</p>
6	Сведения о поставках	<p>1 Создать сложный отчет, включающий в себя поля «Наименование товара», «Объем поставки», «Цена единицы» и «Общая стоимость» из основной таблицы и поля «Предприятие-производитель» и «Дата выпуска» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Сведения о поставках», включив в него подчиненный отчет на основе «Сопроводительный лист товара»</p>
7	Телефонный справочник	<p>1 Создать сложный отчет, включающий в себя поля «Ф.И.О. абонента», «Адрес», «№ телефона», «Дата установки» из основной таблицы и поля «Наличие АОН», «Наличие модема» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Телефонный справочник», включив в него подчиненный отчет на основе таблицы «Дополнительная информация»</p>
8	Дневник метеонаблюдений	<p>1 Создать сложный отчет, включающий в себя поля «Дата», «Температура», «Осадки» из основной таблицы и поля «Ф.И.О. наблюдателя» и «Время снятия показаний» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Дневник метеонаблюдений», включив в него подчиненный отчет на основе таблицы «Дополнительная информация»</p>
9	Отдел кадров	<p>1 Создать сложный отчет, включающий в себя поля «Табельный номер», «Ф.И.О. сотрудника», «Отдел», «Должность» из основной таблицы и поля «Домашний адрес» и «Паспортные данные» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Отдел кадров», включив в него подчиненный отчет на основе таблицы «Карточка сотрудника»</p>
10	Карточка абитуриента	<p>1 Создать сложный отчет, включающий в себя поля «Ф.И.О.», «Отделение» и «Специальность» из основной таблицы и поля «Паспортные данные» и «Домашний адрес» из связанной таблицы</p> <p>2 Создать отчет на основе таблицы «Карточка абитуриента», включив в него подчиненный отчет на основе таблицы «Дополнительная информация»</p>

## 8.16 Создание кнопочной формы

Обычно последней создаваемой формой является главная кнопочная форма, которая предоставляет пользователю прямой доступ к основным задачам приложения. Построение всех кнопочных форм, необходимых для управления сложным приложением, может быть очень трудоемким делом. Специальная надстройка Access – диспетчер кнопочных форм – поможет выполнить эту работу. Эта надстройка применяет достаточно сложную технику для управления всеми кнопочными формами с помощью одной формы и использует специальную таблицу-драйвер *Элементы кнопочной формы*, что позволяет определить любое число кнопочных форм и издавать до восьми кнопок в каждой из них.

Чтобы запустить Диспетчер кнопочных форм, необходимо выполнить следующие процедуры:

1. Выбрать команду **Сервис/Служебные программы/Диспетчер кнопочных форм**.

2. В ответ на предложение создать кнопочную форму нужно щелкнуть на кнопке Да. Откроется диалоговое окно. Каждая кнопочная форма в приложении называется *страницей главной кнопочной формы*. Когда на шаге 2 предыдущей процедуры щелкаете на кнопке Да, диспетчер добавляет в диалоговое окно страницу *Main Switchboard (Default)* (главная кнопочная форма по умолчанию). Чтобы перед добавлением увидеть эту кнопочную форму, необходимо выполнить следующие действия:

- 1 Закрыть диалоговое окно Диспетчер кнопочных форм.

- 2 В окне базы данных щелкнуть на кнопке **Формы**.

- 3 Выбрать форму *Switchboard* и открыть ее (либо дважды щелкнуть на ней, либо нажать кнопку **Открыть**).

Оформление кнопочной формы зависит от того, какой шаблон выбран в диалоговом окне **Параметры**. Если шаблон не задан, то оформление будет обычным.

Для добавления новых элементов в главную кнопочную форму необходимо выполнить следующие процедуры:

- 1 Если в окне Диспетчер кнопочных форм имеется несколько страниц, необходимо выделить главную – страницу *Main Switchboard (Default)*.

- 2 Затем нужно щелкнуть на кнопке **Изменить**. Откроется следующее диалоговое окно.

- 3 Чтобы добавить элемент, нужно щелкнуть на кнопке **Создать**. Откроется диалоговое окно **Изменение элемента кнопочной формы**.

- 4 В поле *Текст* надо ввести имя нового элемента *Члены клуба*.

- 5 В раскрывающемся списке *Команда* нужно выбрать команду **Открыть форму для изменения**.

- 6 В раскрывающемся списке *Форма* выбрать форму *Члены клуба*.
- 7 Щелкнуть по кнопке **ОК**, чтобы вернуться к предыдущему окну.
- 8 Повторить шаги 3–7 для каждого элемента кнопочной формы.

После того как элементы страницы созданы, можно разместить их требуемым образом на странице. В окне **Диспетчер кнопочных форм** нужно выбрать страницу, которая должна быть отредактирована. Чтобы переместить элемент вверх или вниз страницы, надо столько раз щелкнуть по кнопке **Вверх** или **Вниз** соответственно, сколько необходимо, чтобы он занял нужную позицию.

Если необходимо, чтобы при открытии базы данных выводилась главная кнопочная форма, нужно изменить параметры запуска базы данных. Для этого в Access имеется диалоговое окно **Параметры запуска**, в котором можно изменить режим открытия приложения. Данное окно вызывается командой **Сервис/Параметры запуска**.

Чтобы отобразить при открытии базы данных главную кнопочную форму, достаточно выбрать имя этой формы в раскрывающемся списке **Вывод формы/страницы**.

### *Контрольные задания*

Создать главную кнопочную форму, которая позволит автоматизировать выполнение основных функций:

- 1 Просмотр содержимого базы данных с возможностью их редактирования в режиме таблицы.
- 2 Просмотр содержимого базы данных путем открытия созданных форм.
- 3 Просмотр и печать созданных отчетов.

Настроить базу данных таким образом, чтобы при ее открытии автоматически выводилась на экран главная кнопочная форма.

## **8.17 Импорт, экспорт данных**

Большинство систем управления базами данных и электронных таблиц используют для хранения данных собственные форматы. Команды **Экспорт, Импорт и Связь с таблицами** позволяют осуществить обмен данными между MS Access и различными внешними приложениями (наиболее популярными электронными таблицами, другими системами управления базами данных и текстовыми файлами).

Для экспорта таблиц используется команда **Файл/Экспорт**. С помощью этой команды можно скопировать таблицу в другую базу данных MS Access или в другое приложение:

- 1 В окне базы данных нужно выбрать из списка таблицу, которую предполагается экспортировать.



2 Надо выбрать команду Файл/Экспорт. Откроется окно диалога **Экспорт объекта**, по своему виду схожее с окном сохранения документа. После выбора папки и имени файла с расширением .mdb нажать кнопку **Сохранить**.

3 MS Access откроет окно Экспорт.

4 В верхней части окна находится поле, в котором можно выбрать текущее имя таблицы или ввести новое. В разделе **Экспорт таблиц** содержатся две опции **Структура и данные** и **Только структура** (опция **Только структура** используется обычно для создания новых таблиц на основании уже существующих таблиц с аналогичной структурой). Нужно убедиться, что выбрана опция **Структура и данные**.

5 В заключение надо нажать кнопку ОК.

Если требуется скопировать объект из текущей таблицы в другую базу данных, для этого необязательно прибегать к экспорту. Можно воспользоваться буфером обмена, т.е. командами **Правка/Копировать** и **Правка/Вставить**. Также эти команды используются при копировании записей из одной базы данных в другую.

При копировании данных в результирующую таблицу MS Access руководствуется следующими правилами:

- Копирование данных осуществляется в соответствии с расположением столбцов исходной таблицы. Столбец 1 таблицы А будет скопирован в столбец 1 таблицы В. Имена полей значений не имеют.

- Значение столбца 1 таблицы А может не совпадать с первичным ключом столбца 1 таблицы В.

- Если в исходной таблице количество полей больше, чем в результирующей таблице, лишние поля не копируются. Если в исходной таблице полей меньше, чем в результирующей таблице, лишние поля в результирующей таблице остаются пустыми.

- Типы данных должны совпадать. Нельзя скопировать текст в числовое поле.

- Если данные не удовлетворяют свойствам в целевой таблице (маске ввода, условиям проверки достоверности данных и т. п.), копирование не выполняется.

- Свойство **Размер поля** определяет максимальный размер значения поля, т.е. копируемый текст не может быть длиннее, чем размер результирующего поля.

Для экспортирования данных в другие приложения (например, MS Word или MS Excel) в диалоговом окне **Экспорт таблиц** в поле **Тип файла** из раскрывающегося списка необходимо выбрать **Текстовые файлы** или **Microsoft Excel 97-2000** соответственно.

При импорте данных MS Access считывает данные из файла вне текущей базы данных и создает для них новую таблицу. В MS Access можно импортировать объекты из других баз Access, электронные таблицы Excel, тексто-

вые файлы (с разделителями или с фиксированной длиной записей), документы HTML и т. д.

Процедура импорта для всех типов файлов единообразна, поэтому ограничимся наиболее распространенными.

Для того чтобы импортировать данные из другой базы данных, нужно выполнить следующие действия:

1. Выбрать команду **Файл/Внешние данные/Импорт**. Откроется диалоговое окно **Импорт**.

2. Выбрать базу данных и щелкнуть по кнопке **Импорт**. На экране появится диалоговое окно **Импорт объектов**.

3. Нужно раскрыть вкладку с названием типа импортируемого объекта. Щелчком выделить объект.

4. Повторить шаг 3 для каждого импортируемого объекта.

5. Щелкнуть по кнопке **ОК**, чтобы создать новый объект (новые объекты) в открытой базе данных.

Имеется возможность импортировать данные Excel в новую таблицу или в уже существующую таблицу Access в том случае, если данные соответствующим образом разбиты на столбцы и строки. Первая строка данных может содержать заголовки столбцов или же значения.

В диалоговом окне **Импорт** в поле **Тип файла** выбрать **Microsoft Excel 97-2000**. Выбрать электронную таблицу с необходимыми данными и нажать кнопку **Импорт**. Откроется Мастер импорта электронной таблицы. Следуя шагам Мастера, можно импортировать электронную таблицу в Access.

### ***Контрольные задания***

1 Экспортировать исходную таблицу базы данных в документ MS Word, создав предварительно документ факсимильного сообщения.

2 Создать средствами MS Excel электронную таблицу и импортировать ее в исходную базу данных.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

- 1 Microsoft Access 2000: Справочник / Под ред. *Ю. Колесникова*. – СПб.: Питер, 1999. – 416 с.
- 2 *Вейскас Дж.* Эффективная работа с Microsoft Access 2000. – СПб.: Питер, 2001. – 1040 с.
- 3 *С.М. Диго.* Проектирование и использование баз данных. – М.: Финансы и статистика, 1995. – 324 с.
- 4 Информатика для юристов и экономистов / *С.В. Симонович* и др. – СПб.: Питер, 2001. – 688 с.
- 5 Информатика: Учебник / Под ред. проф. *Н.В. Макаровой*. 2-е изд. – М.: Финансы и статистика, 1998. – 458 с.
- 6 Новейший самоучитель работы на компьютере / Под ред. *С.В. Симоновича*. – М.: Десс; Инфорком-Пресс, 1999. – 448 с.
- 7 *Савицкий Н.И.* Технологии организации, хранения и обработки данных: Учеб пособие. – М.: ИНФРА-М, 2001. – 232 с.
- 8 *Хансен Г., Хансен Дж.* Базы данных: разработка и использование: Пер. с англ. – М.: БИНОМ, 1999. – 224 с.

## ОГЛАВЛЕНИЕ

1 ВВЕДЕНИЕ В ТЕХНОЛОГИИ ОРГАНИЗАЦИИ, ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ .....	3
1.1 Предмет и содержание курса .....	3
1.2 Экономическая информация .....	4
1.3 Структурные единицы экономической информации .....	7
1.4 Внемашинная организация экономической информации .....	11
2 ОРГАНИЗАЦИЯ ДАННЫХ В БАЗАХ .....	12
2.1 Причины, обусловившие появление баз данных .....	12
2.2 Понятие базы данных .....	14
2.3 Историческая справка о базах данных .....	16
2.4 Модели данных .....	17
2.5 Средства представления связей данных .....	23
2.6 Файловая модель .....	28
2.7 Централизованные и распределенные базы данных .....	29
3 ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ .....	30
3.1 Понятие процесса проектирования .....	30
3.2 Этапы проектирования реляционной базы данных .....	31
3.3 Нормализация отношений .....	35
4 СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ .....	38
4.1 Понятие системы управления базами данных .....	38
4.2 Классификация систем управления базами данных .....	40
4.3 Краткий исторический обзор систем управления базами данных для персональных компьютеров .....	42
4.4 Режимы работы пользователя с системой управления базой данных .....	43
4.5 Функциональные возможности системы управления базой данных .....	44
4.6 Обзор современных систем управления базами данных .....	45
4.7 Тенденция развития систем управления базами данных .....	48
5 СТРУКТУРА MICROSOFT ACCESS .....	49
5.1 Общая характеристика .....	49
5.2 Объекты MS ACCESS .....	51
5.3 Типы данных MS ACCESS .....	52
5.4 Запуск MS ACCESS и завершение работы .....	54
5.5 Настройка рабочей среды .....	56
5.6 Выражения. Построитель выражений .....	57
6 СОЗДАНИЕ И КОРРЕКТИРОВКА БАЗЫ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ .....	61
6.1 Способы создания таблиц .....	61
6.2 Создание таблиц .....	62
6.2.1 Создание таблиц с помощью Мастера .....	63
6.2.2 Проектирование таблицы в окне Конструктора .....	64
6.2.3 Создание таблицы в режиме таблицы .....	67
6.3 Просмотр и редактирование содержания полей .....	70
6.4 Редактирование, поиск, замена данных .....	70
6.5 Сортировка данных .....	72

6.6 Фильтры .....	73
6.7 Поиск информации.....	75
6.8 Изменение структуры и вида таблицы.....	75
6.9 Связи между таблицами.....	77
<b>7 ФОРМИРОВАНИЕ ЗАПРОСОВ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ.....</b>	<b>88</b>
7.1 Понятие запроса к базе данных.....	88
7.2 Создание запроса с помощью Мастера.....	89
7.3 Создание запроса с помощью Конструктора.....	91
7.4 Вычисляемые поля в запросах. Построитель выражений.....	95
7.5 Технология создания запросов.....	99
7.6 Параметрические запросы.....	101
7.7 Перекрестные запросы.....	102
7.8 Обработка запросов.....	103
<b>8 ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА БАЗЫ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ .....</b>	<b>110</b>
8.1 Проектирование форм и работа с ними.....	110
8.2 Простейший способ создания формы.....	110
8.3 Создание формы с помощью Мастера.....	113
8.4 Создание формы с помощью Конструктора.....	114
8.5 Главные и подчиненные формы.....	121
8.6 Списки и поля со списками в форме.....	123
8.7 Редактирование записей в форме.....	125
8.8 Представление формы.....	126
8.9 Понятие отчета как объекта базы данных.....	132
8.10 Создание отчета с помощью Мастера.....	133
8.11 Создание отчета с помощью Конструктора.....	135
8.12 Пример создания отчета.....	137
8.13 Изменение вида отчета.....	142
8.14 Главный и подчиненные отчеты.....	144
8.15 Обработка отчетов.....	145
8.16 Создание кнопочной формы.....	151
8.17 Импорт, экспорт данных.....	152
<b>СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....</b>	<b>155</b>

Учебное издание

*ЛЫЧ Юрий Павлович*  
*ШИНКЕВИЧ Татьяна Леонидовна*

**Технологии организации, хранения и обработки данных**

**Часть 1 Основы создания баз данных**

Пособие для самостоятельной работы

Редактор Н.А. Д а ш к е в и ч  
Технический редактор В.Н. К у ч е р о в а

Подписано в печать 12.06.2003 г. Формат бумаги 60x84 1/16.

Бумага газетная. Гарнитура Таймс. Печать офсетная.

Усл. печ. л. 9,30. Уч.-изд. л. 9,64. Тираж 300 экз.

Зак. № . Изд. № 3982.

Редакционно-издательский отдел БелГУТа, 246653, г. Гомель, ул. Кирова, 34.

Лицензия ЛВ № 57 от 22.10.2002 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.

Лицензия ЛП № 360 от 25.07.1999 г.