

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

Кафедра "Информационные технологии"

Т. Л. ШИНКЕВИЧ, Т. Н. ЛИТВИНОВИЧ

**СИСТЕМЫ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ**

**Пособие по выполнению контрольной работы
для студентов ФБО**

Гомель 2005

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

Кафедра "Информационные технологии"

Т. Л. ШИНКЕВИЧ, Т. Н. ЛИТВИНОВИЧ

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

**Пособие по выполнению контрольной работы
для студентов ФБО**

*Одобрено методической комиссией факультета
безотрывного обучения*

Гомель 2005

УДК 681.3(076.5)
Ш 626

Рецензент – **Ю. А. Пшеничнов**, канд. техн. наук, доцент кафедры «Информационные технологии».

Шинкевич Т. Л., Литвинович Т. Н.

Ш 626 Системы управления базами данных: Пособие по выполнению контрольной работы для студентов заочной формы обучения, экономических специальностей факультета безотрывного обучения *Т.Л. Шинкевич, Т.Н. Литвинович* – Гомель: БелГУТ, 2005. – 70 с.

В данном пособии изложен теоретический материал с примерами по курсу «ТОХОД», охватывающий следующие разделы: проектирование баз данных, создание таблиц и форм баз данных, создание, форматирование и редактирование запросов и отчетов в СУБД MS Access. Пособие предназначено для выполнения контрольной работы студентами заочной формы обучения факультета безотрывного обучения. Пособие может быть так же использовано студентами дневной формы обучения в качестве дополнительного учебного материала при подготовке к зачетам и экзаменам по соответствующим дисциплинам.

УДК 681.3(073.5)

© Т. Л. Шинкевич, Т. Н. Литвинович 2005.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	6
1.1 Проектирование базы данных «Железнодорожное расписание»	6
1.2 Запуск MS ACCESS	8
2 ТАБЛИЦЫ.....	9
2.1 Создание таблицы.....	10
2.1.1 Создание таблицы в режиме Конструктора.....	12
2.2 Определение имен полей.....	13
2.3 Типы данных.....	13
2.4 Заполнение полей таблицы в окне конструктора.....	14
2.5 Свойства полей	14
2.5.1 Задание маски ввода.....	16
2.5.2 Задание свойства условие на значение	16
2.5.3 Задание свойства подстановка	17
2.6 Выбор ключевого поля	18
2.6.1 Свойства ключевого поля	19
2.7 Сохранение таблицы.....	19
2.8 Присвоение имени объектам	19
2.9 Межтабличные связи	19
2.9.1 Создание схемы данных.....	21
2.9.2 Задание условий целостности	22
2.10 Ввод данных в таблицу.....	23
2.11 Перемещение по таблице.....	24
2.12 Редактирование данных в таблице	25
2.13 Изменение структуры таблицы	25
2.14 Сжатие данных.....	25
2.15 Внешний вид таблицы	26
3 ФОРМЫ.....	26
3.1 Создание формы	26
3.1.1 Создание автоформы.....	27
3.2 Ввод и редактирование данных в форме.....	29
3.3 Мастер форм	29
3.3.1 Создание подчиненной формы	31
3.4 Конструктор форм	33
3.4.1 Изменение структуры формы	35
3.5 Элементы управления.....	35
3.5.1 Перемещение элементов управления	38
3.5.2 Изменение размеров элементов управления	38

3.5.3	Создание заголовка формы.....	39
3.5.4	Задание стиля оформления элементов управления.....	39
3.5.5	Изменение расположения элементов управления.....	39
3.5.6	Добавления элемента управления в форму.....	39
3.5.7	Создание элементов управления.....	40
3.6	Создание кнопочной формы.....	40
3.6.1	Создание макроса.....	43
4	ЗАПРОСЫ.....	45
4.1	Создание запросов на выборку.....	46
4.1.1	Создание запроса на выборку при помощи мастера.....	46
4.1.2	Создание запроса на выборку без помощи мастера.....	47
4.2	Структура бланка запроса.....	50
4.3	Создание запроса с параметрами, запрашивающего ввод условий отбора при каждом запуске.....	51
4.4	Создание итоговых запросов.....	53
4.5	Формирование условий отбора в запросах.....	55
4.6	Построитель выражений. Создание вычисляемых полей в запросах.....	56
5	ОТЧЕТЫ.....	57
5.1	Создание отчета при помощи мастера.....	58
5.2	Создание автоотчетов.....	61
5.3	Вычисление итоговых значений в отчетах.....	61
5.4	Создание отчета в режиме конструктора.....	62
5.5	Окно отчета в режиме конструктора. Настройка параметров окна.....	63
5.6	Редактирование и форматирование отчета.....	64
5.7	Создание вычисляемых и итоговых полей.....	65
6	ПОРЯДОК ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ.....	66
7	РАБОЧАЯ ПРОГРАММА.....	69
	КОНТРОЛЬНЫЕ ВОПРОСЫ.....	76
	СПИСОК ЛИТЕРАТУРЫ.....	78

ВВЕДЕНИЕ

База данных – это совокупность хранимых во внешней памяти компьютера большого объема данных, представляющих собой комплекс взаимосвязанных элементов (записей) для обеспечения информационных потребностей пользователя.

Кроме данных, база содержит методы и средства, позволяющие пользователю оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным пользователям, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием базы данных тесно связано понятие системы управления базой данных. Система управления базами данных является посредником между пользователем и базой данных.

Система управления базами данных - это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержанием, редактирования содержимого и визуализации информации (отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода).

Система управления базами данных Microsoft Access является реляционной СУБД для операционной системы Microsoft Windows.

В MS Access предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации. Основными объектами Access являются таблицы, запросы, формы, отчеты, макросы и модули.

Таблицы – это основные объекты базы данных. В таблицах хранятся все данные, имеющиеся в базе, и структура базы (поля, их типы и свойства).

Формы – это средства для ввода данных. С помощью форм данные можно не только вводить, но и отображать.

Запросы – это объекты, которые служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде.

Отчеты по своим свойствам и структуре во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер).

1 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Разработка любой базы данных начинается с ее проектирования.

Продуманная структура и правильно выбранный формат записей способствуют эффективной работе с базой данных и позволяют быстро получить доступ к необходимой информации.

На этапе проектирования необходимо:

- 1 Определить назначение базы данных.
- 2 Принять решение о том, какие исходные данные (таблицы) база должна содержать.
- 3 Определить поля, которые будут входить в таблицы, типы полей и их свойства.
- 4 Выбрать поля, содержащие уникальные значения.
- 5 Распределить данные по таблицам.
- 6 Назначить связи между таблицами.

1.1 Проектирование базы данных «Железнодорожное расписание»

Имеются следующие сведения о движении поездов: № поезда, пункт отправления, пункт прибытия, расстояние, коэффициент на стоимость, категория поезда, тип вагона, стоимость билета, время отправления, время прибытия на конечную станцию.

Необходимо с помощью СУБД ACCESS разработать структуру базы данных, отражающую реальную ситуацию о движении поездов. Для этого необходимо:

1 Распределить данные по таблицам и определить уникальные (ключевые) поля.

В таблице «*Маршрут*» будут содержаться сведения: номер поезда, пункт отправления, пункт прибытия, расстояние, коэффициент на стоимость. В этой таблице каждому номеру поезда соответствует только один маршрут (пункт прибытия - пункт отправления). Следовательно, поле *номер поезда* можно определить как уникальное (ключевое) поле.

Информацию о категории поездов вынесем в отдельную таблицу «*Категории поездов*», где каждой категории поезда присвоим код. Коду категории в данной таблице соответствует только одно название категории. Следовательно, поле *код* категории можно определить как уникальное (ключевое) поле.

Информацию о стоимости билетов по типам вагонов вынесем в таблицу «*Тип вагонов*». В этой таблице каждому типу вагона присвоим код, который в данной таблице будет уникален. Следовательно, поле *код* типа вагона в таблице «*Тип вагонов*» будет ключевым.

В таблице **«Железнодорожное расписание»** будут содержаться сведения о расписании поездов, категории поездов, типах вагонов: номер поезда, код категории, код типа вагона, время прибытия, время отправления. Так как в данной таблице поле **номер поезда** является уникальным, то его можно определить как ключевое.

2 Для каждого поля, включаемого в запись, установить тип данных, определяющий вид информации, которая будет храниться в поле. Данные в поле счетчика всегда уникальны по определению, поэтому во всех таблицах ключевым полям присвоим тип данных «счетчик».

3 Установить связи между таблицами. Из таблицы **«Маршрут»** ключевое поле **«номер поезда»** свяжем с соответствующим полем таблицы **«Железнодорожное расписание»** (в этой таблице номер поезда – ключевое поле) – тип связи «один – к – одному». Таблицы **«Категории поездов»** и **«Железнодорожное расписание»** свяжем по полю **«код категории»** (тип связи «один – ко – многим»). Таблицы **«Тип вагонов»** и **«Железнодорожное расписание»** свяжем по полю **код** типа вагона (тип связи «один – ко – многим»). Разработанная структура базы данных **«Железнодорожное расписание»** представлена на рисунке 1.1.

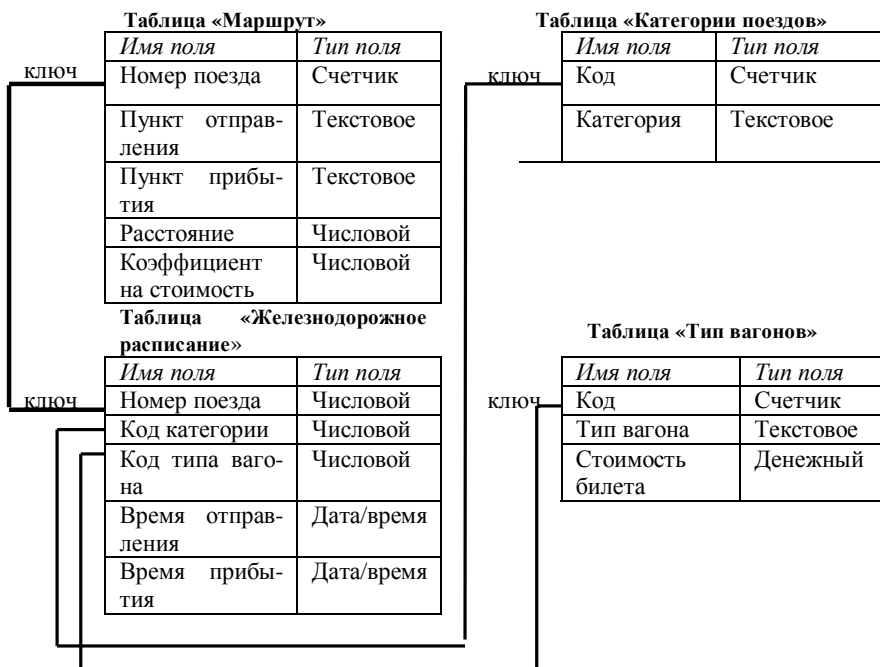


Рисунок 1.1 – Структура базы данных «Железнодорожное расписание»

При выполнении данного раздела, студентам необходимо на отдельном листе спроектировать базу данных по соответствующему варианту (распределить данные по таблицам(выделить минимум две таблицы), указать тип полей, ключевые поля, связи между таблицами). Спроектированная база данных *«Железнодорожное расписание»* представлена на рисунке 1.2.

Таблица «Маршрут»

Имя поля	Тип поля	Ключевое поле
Номер поезда	Счетчик	Ключ
Пункт отправления	Текстовое	
Пункт прибытия	Текстовое	
Расстояние	Числовой	
Коэффициент на стоимость	Числовой	

Таблица «Категории поездов»

Имя поля	Тип поля	Ключевое поле
Код	Счетчик	Ключ
Категория	Текстовое	

Таблица «Тип вагонов»

Имя поля	Тип поля	Ключевое поле
Код	Счетчик	Ключ
Тип вагона	Текстовое	
Стоимость	Денежное	

Таблица «Железнодорожное расписание»

Имя поля	Тип поля	Ключевое поле
Номер поезда	Числовой	Ключ
Код категории	Числовое	
Код типа вагона	Числовое	
Время отправления	Дата\время	
Время прибытия	Дата\время	

Рисунок 1.2 – База данных «Железнодорожное расписание»

1.2 Запуск MS ACCESS

Для запуска MS ACCESS в главном меню Windows (кнопка «Пуск») необходимо выбрать **Программы**→**MS Access**. Откроется окно (рисунок 1.3), в котором находятся две основные опции для создания новой и открытия существующей базы данных.

Для создания новой базы данных необходимо выбрать опцию «*Новая база данных*». Нажать кнопку *ОК*.

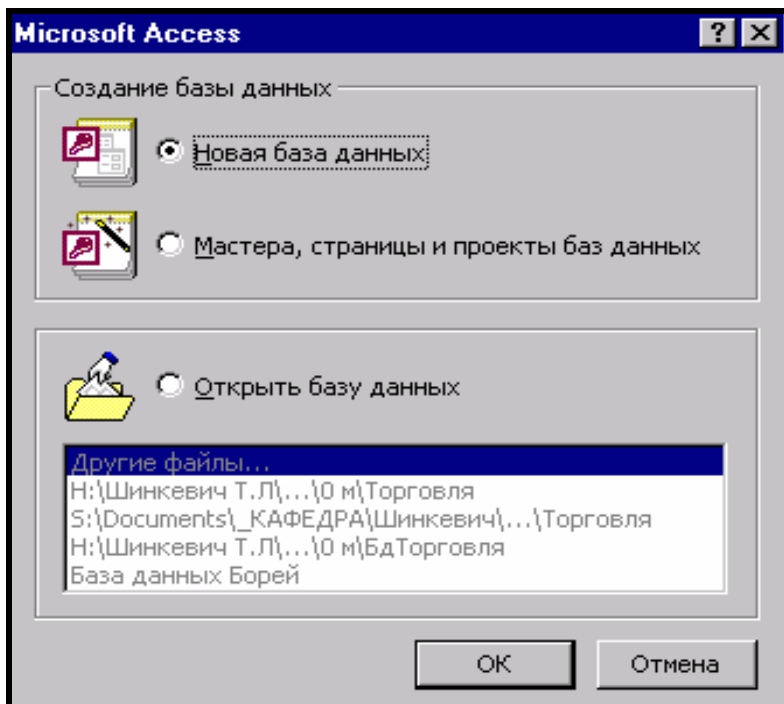


Рисунок 1.3 – Окно-приглашение MS Access

Откроется окно (рисунок 1.4), в котором необходимо указать место сохранения базы данных, имя базы данных и нажать кнопку *Создать*. Сохранить файл базы данных надо до того, как приступите к созданию таблиц. Таким образом, создается пустая база данных, в которую впоследствии можно поместить таблицы, формы, запросы, отчеты.

2 ТАБЛИЦЫ

В ACCESS вся информация хранится в виде таблиц. Таблицей является совокупность данных, объединенных общей темой. Для каждой темы отводится отдельная таблица, что позволяет избежать повторения сохраняемых данных. Каждая строка таблицы представляет собой запись, а столбец – поле. Запись содержит набор данных об одном объекте: например, номер по-

езда, маршрут, а поле – однородные данные обо всех объектах (например, маршрут всех занесенных в таблицу поездов). Количество полей в записи, и их тип определяются в процессе проектирования базы данных.

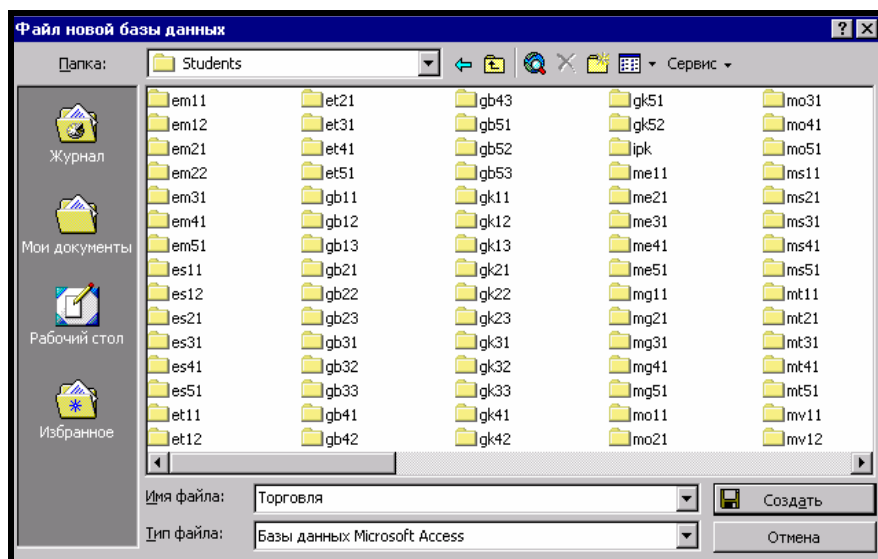


Рисунок 1.4 – Создание файла базы данных

2.1 Создание таблицы

Для создания таблицы необходимо открыть базу данных, выбрать вкладку **Таблица** и щелкнуть на кнопке **Создать** (рисунок 2.1).

Откроется диалоговое окно **Новая таблица** (рисунок 2.2).

В этом окне можно выбрать один из пяти возможных способов создания таблицы:

- Режим таблицы (Datasheet View) – таблица создается путем ввода имен полей в заголовки каждого столбца.
- Конструктор (Design View) – таблица создается путем составления списка имен полей и задания свойств каждого поля.
- Мастер таблиц (Table Wizard) – таблица создается автоматически с помощью программы-мастера, которая предлагает выбрать поля из списка и содержит заголовки для различных видов таблиц (свыше 100).

– Импорт таблиц (Import Table) – таблица создается путем импорта данных из другой базы или из другой электронной таблицы.

– Связь с таблицами (Link Table) – таблица создается путем установления связей с таблицей, существующей в другой базе данных.

Если речь идет о создании новой таблицы, то удобнее всего воспользоваться для этого **режимом Конструктора (Design View)**. Этот способ удобен тем, что позволяет одновременно задать поля будущей таблицы и назначить свойства этих полей.

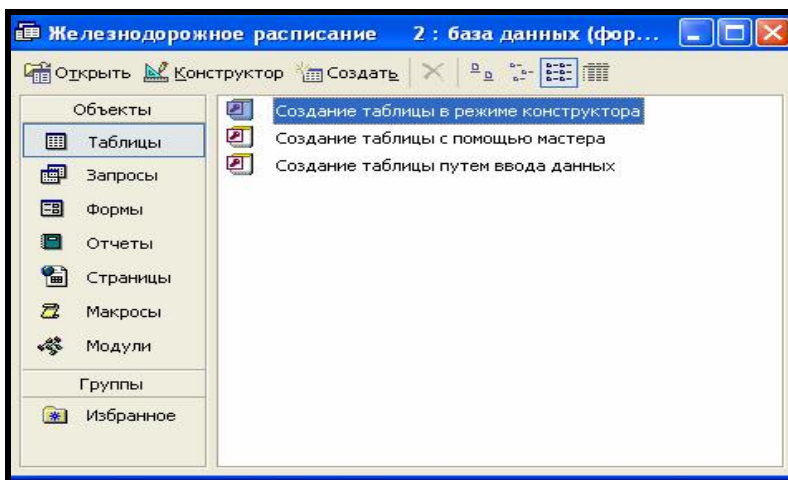


Рисунок 2.1 – Окно базы данных

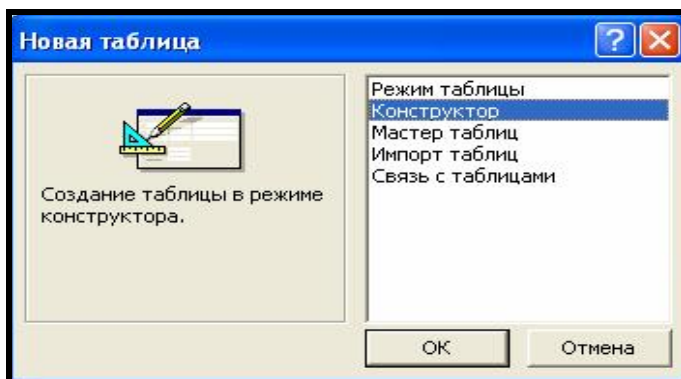


Рисунок 2.2 – Создание таблицы

2.1.1 Создание таблицы в режиме Конструктора

Для создания таблицы в режиме **Конструктора** необходимо выбрать строку **Конструктор** в окне **Новая таблица** и нажать кнопку **ОК**. Откроется окно конструктора таблицы (рисунок 2.3).

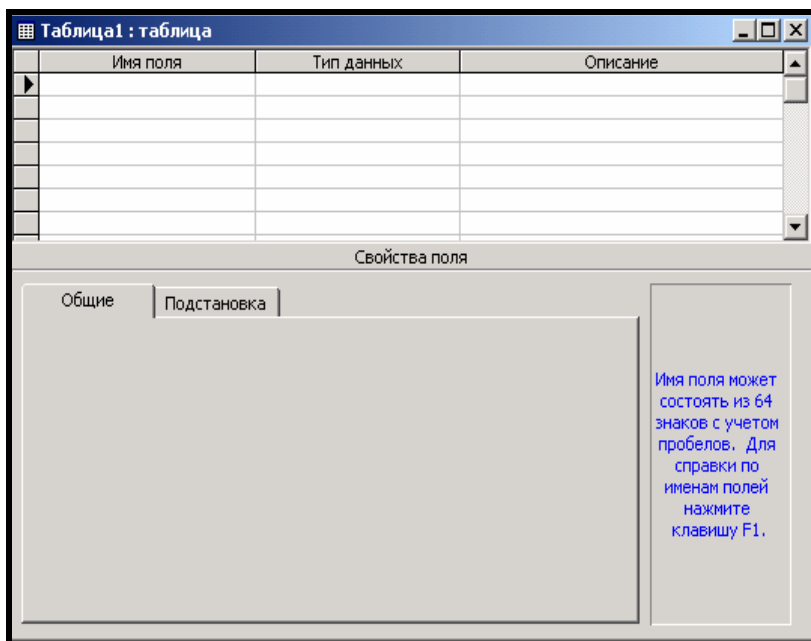


Рисунок 2.3 – Создание таблицы в режиме Конструктора


Окно Конструктора таблицы имеет следующую структуру:

- столбец маркировки (черный треугольник в этом столбце указывает на ту строку, в которой в данный момент находится курсор ввода; значок ключа – на то, что данное поле является ключевым);
- столбец – имя поля;
- столбец – тип данных;
- столбец – описание.

В нижней левой части окна Конструктора представлены свойства полей.

В нижней правой части окна Конструктора отображается справочная информация о допустимых на каждом этапе проектирования действиях.

Для начала ввода необходимо щелкнуть мышью в столбце **имя поля** и ввести название первого поля (в данном случае это номер поезда). Затем

клавишами ENTER, TAB или щелчком мыши перейти в столбец *тип данных* и из списка, который откроется при нажатии на , выбрать нужный тип данных.

2.2 Определение имен полей

Имя поля может иметь длину до 64 символов и содержать пробелы и любые специальные символы, кроме точек, восклицательных знаков и прямых скобок. В таблице не должно быть двух полей с одинаковыми именами, поскольку Access идентифицирует поле по его имени.

2.3 Типы данных

Тип данных указывает, какие данные могут быть введены. Access позволяет выбрать один из следующих типов данных:

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он хранится в другом месте базы данных, а в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда.

Числовой – тип данных для хранения действительных чисел.

Дата/время – тип данных для хранения календарных дат и текущего времени.

Денежный – тип данных для хранения денежных сумм.

Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование — для порядковой нумерации записей.

Логический – тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

Поле объекта OLE – специальный тип данных, предназначенный для хранения объектов OLE, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей MEMO, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).

Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод в данных поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

По умолчанию устанавливается тип данных Текстовый. Чтобы изменить тип данных, надо щелкнуть мышью на строке поля в столбце Тип данных (Data Type) , после чего щелкнуть на кнопке раскрывающийся список и выбрать из раскрывающегося списка нужный тип данных (рисунок 2.4).

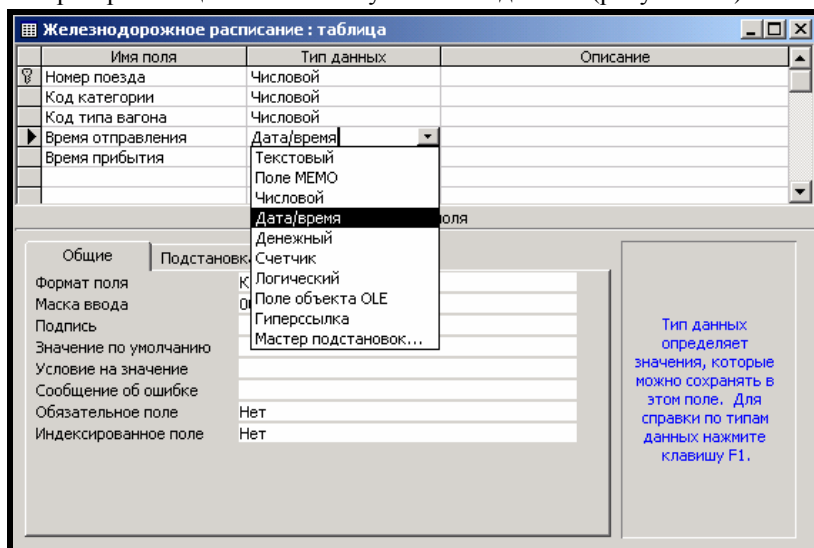


Рисунок 2.4 – Выбор типа данных

2.4 Заполнение полей таблицы в окне конструктора

Для каждого поля в окне конструктора последовательно вводятся имя, затем тип данных. Если назначение поля неочевидно по его имени, необходимо ввести произвольное описание поля в столбец **«Описание»**. Переход к следующей ячейке строки осуществляется с помощью клавиши TAB или щелчком мыши на этой ячейке. Переход к заполнению следующего поля осуществляется клавишами TAB, ENTER или щелчком мыши.

2.5 Свойства полей

Каждое поле имеет свой набор свойств, от которых зависит то, как содержимое этого поля отображается на экране, как оно хранится в базе дан-

ных, как оно проверяется и как им можно управлять. Некоторые свойства являются общими для всех типов полей, другие относятся исключительно к отдельным типам данных.

Некоторые из свойств задаются по умолчанию. Свойства полей не являются обязательными. Их можно настраивать по желанию, а можно и не трогать.

Все типы полей, кроме полей счетчика и объекта OLE, имеют следующие основные свойства:

- Формат поля (Format) – определяет, как должно отображаться содержимое поля.
- Подпись (Caption) – определяет тот текст, который будет отображаться рядом с полем или в отчете.
- Значение по умолчанию (Default Value) – определяет значение, которое вводится в поле автоматически при добавлении новых записей.
- Условие на значение (Validation Rule) – определяет область или диапазон значений данных, допустимых в поле.
- Сообщение об ошибке (Validation Text) – задает текст сообщения, которое появляется в строке состояния при попытке ввести в поле недопустимое значение.
- Обязательное поле (Required) – содержит значение Да (Yes) или Нет (No), в зависимости от того, обязательно ли должно это поле иметь какую-либо запись.

Следующие свойства относятся в основном к текстовым полям:

- Размер поля (Field Size) – максимально допустимое количество символов для данного поля.
- Маска ввода (Input Mask) – обеспечивает ввод данных в указанном формате.
- Индексированное поле (Indexed) – указывает Access, должны ли создаваться индексы для этого поля: индексирование поля ускоряет процесс сортировки, поиска и фильтрации содержимого. Поле, являющееся первичным ключом таблицы, индексировано всегда. Стоит задавать индексацию тех полей, по которым возможно проведение сортировки в формах или отчетах.
- Сжатие Unicode (Unicode Compression) – представляет собой схему кодировки всех известных символов, для достижения чего вместо одного байта каждый символ представлен двумя. Когда эта функция отключена, Access компрессирует все символы, начинающиеся с байта 0, при сохранении и распаковывает их при обращении к ним.

Поля числового и денежного типов имеют одно дополнительное свойство – число десятичных знаков, которое определяет, какое количество десятичных знаков после запятой будет отображаться на экране и храниться в поле.

Чтобы установить свойство поля, надо выделить строку поля, после чего в левой нижней части окна конструктора откроется список свойств данного поля. Щелчком мыши установить курсор на строку свойства и выбрать нужное значение.

2.5.1 Задание маски ввода

При вводе данных, например в поле дата поступления, было бы удобно, чтобы в этом поле заранее вводилась точка в качестве разделителя – тогда ее не надо было вводить вручную.

Форматы для ввода данных называются масками ввода.

В состав Access входит мастер ввода масок. Чтобы задать маску ввода, надо:

- 1 Щелкнуть по строке свойства – Маска ввода, появится кнопка с тремя точками.

- 2 Щелкнуть по кнопке с тремя точками. Пользователю будет предложено сохранить таблицу, после чего откроется окно **Создание масок ввода**.

- 3 Выбрать тип маски ввода (например, краткий формат даты для поля дата поступления) и щелкнуть по кнопке **Готово** (Finish). Диалоговое окно **Создание масок ввода** закроется, и маска ввода будет внесена в свойства поля.

Хранение данных в базе вместе с маской ввода увеличивает размер базы данных, но позволяет облегчить ввод данных.

2.5.2 Задание свойства условие на значение

Условие на значение (Validation Rule) – это свойство, с помощью которого можно проверять данные, вводимые в таблицу или форму.

Условие на значение для поля устанавливается в строке свойства **Условие на значение**. Когда после ввода данных пользователь переключается на следующее поле или щелкнет вне его, то Access проверяет, соответствуют ли введенные данные установленному условию на значение. Если условие не выполняется, выводится **Сообщение об ошибке**.

Чтобы задать условие на значение для поля, надо:

- 1 Открыть таблицу в режиме конструктора.

- 2 Выделить поле.

- 3 Щелкнуть в строке свойства **Условие на значение** и ввести условие с помощью логических операторов (например, для поля цена введем <50 000).

- 4 В строке свойства **Сообщение об ошибке** ввести текст сообщения, которое должно отображаться в строке состояния при вводе недопустимых данных.

2.5.3 Задание свойства подстановка

Поля в таблице могут содержать ограниченное число значений. Поэтому эти значения лучше не вводить, а выбирать из списка.

Чтобы создать для поля список значений необходимо:

1 Открыть таблицу в режиме *Конструктора*.

2 Щелкнуть мышью в столбце «*Тип данных*» соответствующего поля, выбрать в соответствующем списке значение *Мастер подстановок*. Будет запущен *Мастер*.

3 В диалоговом окне «*Создание подстановки*» установить опцию *будет введен фиксированный набор значений* или *объект «столбец подстановки» будет использовать значения из таблицы или запроса* и нажать кнопку *Далее*.

4 В следующем диалоговом окне *Создание подстановки*, если в п.3 установлена опция «*будет введен фиксированный набор значений*» указать число столбцов (например 1) и ввести значения данного поля и завершить формирование списка нажатием кнопки *Готово*. Если в п.3 установлена опция «*объект «столбец подстановки» будет использовать значения из таблицы или запроса*», то необходимо выбрать имя таблицы или запроса, из которого будет выполнена подстановка. Нажать кнопку *Далее*. Выбрать с помощью кнопок < или << поля, которые будут отображаться в таблице подстановке. Нажать кнопку *Далее*. Если полей выбрано более, чем одно, то отменить опцию «*Скрыть ключевой столбец*». Нажать кнопку *Готово*.

5 В области *Свойства поля* на вкладке *Подстановка* для выбранного поля появятся новые свойства (рисунок 2.5).

При заполнении таблицы появится возможность не вводить данные вручную, а выбирать из списка (рисунок 2.6).

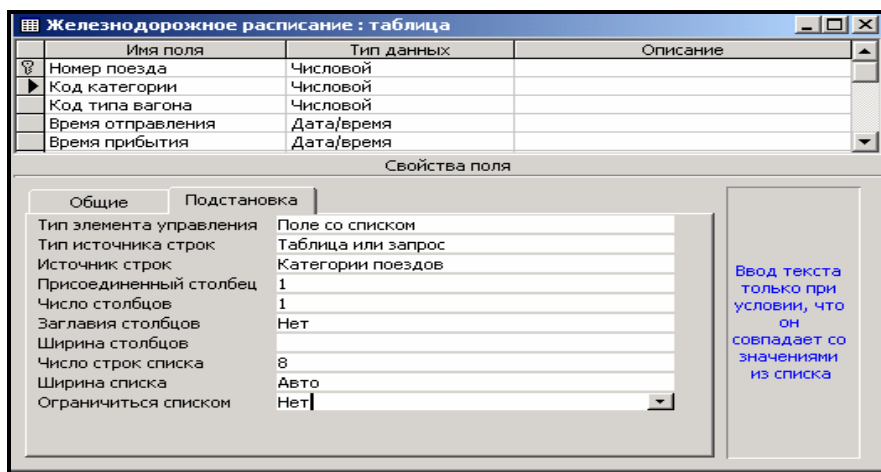


Рисунок 2.5 – Свойство поля «Подстановка»

Железнодорожное расписание : таблица						
	Номер поезда	Код категории	Код типа вагона	Время прихода	Время отправления	
	+	1	4	1	12:00	15:00
	+	53	1	3	3:00	22:10
	▶	54	1	4	12:00	23:00
	+	55	1	5	6:10	22:15
	+	56	2	3	8:55	21:40
	+	67	3	4	9:10	23:55
	+	68	4	4	5:15	22:45
	+	75	3	3	21:00	7:10
	+	76	3	4	2:20	21:30
	+	95	3	5	6:00	15:00
	+	107	1	3	12:00	23:10
	+	649	3	1	9:00	21:10
	*	0	0	0		

Запись: 3 из 12

Рисунок 2.6 – Выбор данных из списка

2.6 Выбор ключевого поля

В реляционной базе данных таблицы должны быть связаны друг с другом. Эта связь устанавливается с помощью уникальных полей. Уникальные поля – это такие поля, в которых значения не могут повторяться. Например, в базе данных **Железнодорожное расписание** в таблице **Маршрут** таким уникальным полем может быть **номер поезда**. Такое поле (или комбинация полей), которое однозначно идентифицирует запись в таблице, называется первичным ключом.

Если в таблице нет такого поля, которое могло бы служить ключом, можно объединить несколько полей. Чтобы создать сложный ключ, состоящий из нескольких полей, надо, удерживая нажатой клавишу CTRL, выделить те поля, которые необходимо включить в ключ и щелкнуть правой кнопкой мыши, в открывшемся контекстном меню выбрать пункт **Ключевое поле**.

Если никакая комбинация полей не подходит для ключа, можно ввести в таблицу дополнительное поле – например, порядковый номер записи, присвоив ему тип – счетчик.

Чтобы задать ключевое поле, надо:

- 1 Выделить поле (или поля), выбранное в качестве ключевого, щелчком мыши слева от столбца **Имя поля**.

2 Нажать кнопку **Ключевое поле** на панели инструментов или щелкнуть правой кнопкой мыши, и в открывшемся контекстном меню выбрать пункт **Ключевое поле**.

Если к моменту выхода из режима конструктора первичный ключ для создаваемой таблицы объявлен не будет, Access выдаст запрос о необходимости включения в таблицу поля первичного ключа. Необходимо выбрать нужный ответ Да или Нет.

2.6.1 Свойства ключевого поля

Для ключевого поля Access автоматически устанавливает в строке свойства **Обязательное поле** значение Да(Yes), а в строке свойства **Индексированное поле** – значение Да (совпадение не допускается). Если в качестве первичного ключа служит группа из нескольких полей, то все поля объявляются обязательными, но совпадения внутри каждого из них допускаются.

2.7 Сохранение таблицы

При закрытии окна конструктора таблицы Access напомнит, что созданную структуру таблицы надо сохранить. Для сохранения структуры выберите пункт **Да**. При выборе пункта **Нет** вы останетесь в режиме конструктора.

2.8 Присвоение имени объектам

Access позволяет называть объекты баз данных так, как вам нравится, но есть неофициальное соглашение Лежински-Реддика, следуя которому можно сделать имена объектов понятными и для других людей. По соглашению, имя объекта начинается тегом, идентифицирующим тип объекта: тбл(tbl), фрм(frm), отч(rpt), зпр qry) и др. В именах не используются пробелы и знаки препинания, первая буква имени должна быть прописной, например тблМагазины, тблПоставки.

2.9 Межтабличные связи

Access позволяет строить реляционные базы данных, отдельные таблицы которых могут быть связаны между собой. Связь между таблицами определяет тип отношения между их полями. Как правило, связывают ключевое поле одной таблицы с соответствующим ему полем другой таблицы, которое называется полем внешнего ключа (рисунок 2.7).

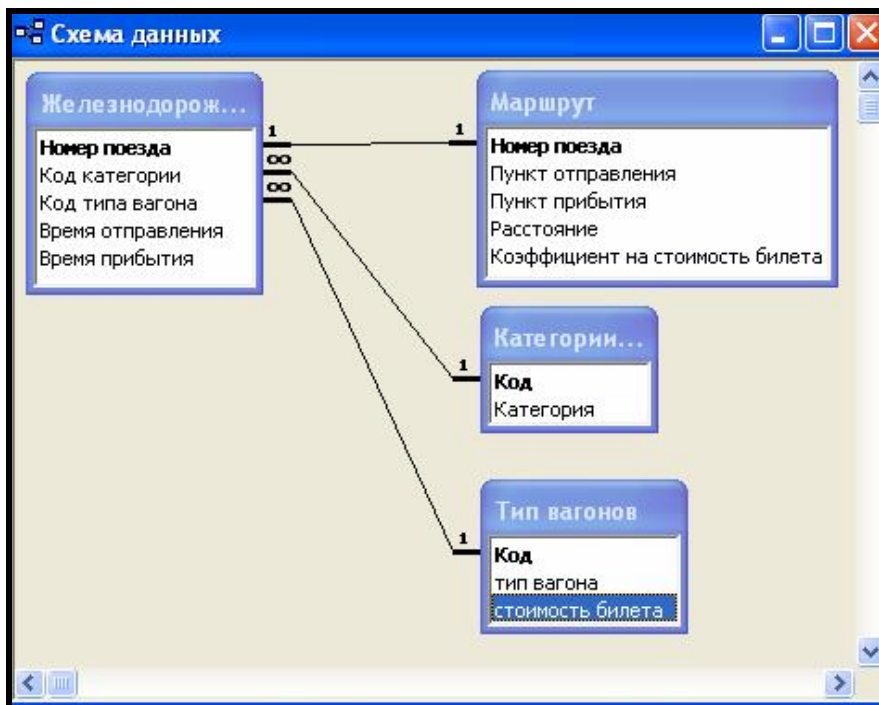


Рисунок 2.7 – Схема данных

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

Существуют три типа связей:

– один-к-одному – каждая запись таблицы А не может быть связана более чем с одной записью таблицы Б;

один-ко-многим – одна запись в таблице А может быть связана со многими записями таблицы Б;

многие-ко-многим – каждая запись в таблице А может быть связана со многими записями в таблице Б, а каждая запись в таблице Б – со многими записями в таблице А.

В Access существует возможность отображения связанных таблиц в окне главной таблицы. Для этого необходимо активизировать главную таблицу в режиме просмотра. Между областью выделения и первым столбцом таблицы имеется узкий столбец, в котором против каждой записи проставлен знак плюс (+) (рисунок 2.8). При щелчке на нём левой кнопкой мыши отображается часть подчинённой таблицы, где содержатся только те записи, которые связаны с текущей записью главной таблицы. При этом плюс (+) меняется на минус (-).

	Идентификатор	Код категории	Код типа здания	Формат площади	Формат планировки
	1		1	1200	300
	Маршрут				
	Имя: ...				
	←				
	53		3	300	200
	Маршрут				
	▶ Имя-С. Тэтэрбург				
	✕				
-	АА	1	4	100	100
-	АБ	1	4	100	100
-	АВ	2	4	100	100
-	АГ	2	4	100	100
-	АД	2	4	100	100
-	АЕ	3	4	100	100
-	АЖ	3	4	100	100
-	АИ	3	4	100	100
-	АК	1	4	100	100
-	АМ	3	1	100	100
*	И	И	1		

Рисунок 2.8 – Связанные таблицы в режиме просмотра

2.9.1 Создание схемы данных

Чтобы создать связи между таблицами, надо:

1 В меню **Сервис** выбрать команду **Схема данных**. На экране появится окно **Схема данных**, в котором можно просмотреть и определить связи между таблицами базы данных, если они установлены.

2 Если связи между таблицами не установлены, окно **Схема данных** будет пустым и необходимо добавить в него таблицы, для которых устанавливаются связи.

3 Чтобы добавить в окно **Схема данных** таблицы, можно щелкнуть правой кнопкой мыши на свободном месте окна и в контекстном меню выбрать пункт **Добавление таблицы**. Откроется окно **Добавление таблицы** (рисунок 2.9). Или щелкнуть по кнопке **Отобразить связи** на панели инструментов и выбрать из списка нужные таблицы.

4 В окне **Добавление таблицы** выделить название таблицы и нажать кнопку **Добавить**. Так необходимо добавить все таблицы, между которыми надо установить связи.

5 Перетащить с помощью мыши используемое для связи поле первичного ключа главной таблицы к соответствующему полю связанной таблицы. На экране появится окно **Изменение связей** (рисунок 2.10). В этом окне

главная таблица и первичный ключ изображаются в левом столбце, а подчиненная таблица и внешний ключ – в правом столбце. Установить флажки: Обеспечение целостности данных, Каскадное обновление связанных полей, Каскадное удаление связанных записей.

6 В нижней части окна будет предложен тип связи между таблицами.

7 Нажать кнопку **ОК**.

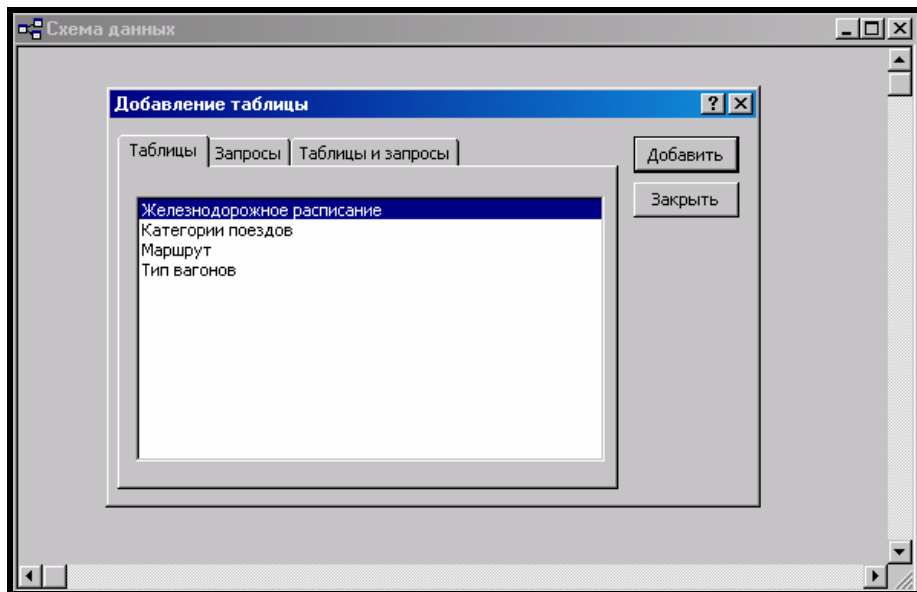


Рисунок 2.9 – Добавление таблиц

2.9.2 Задание условий целостности

Условие целостности предназначено для того, чтобы обеспечить соответствие записей в подчиненной таблице записям в главной таблице. Оно предотвратит непреднамеренное удаление пользователем тех записей в главной таблице, которые связаны с записями в подчиненной таблице.

Если допускается каскадное обновление связанных полей, то при изменении значений первичного ключа в главной таблице происходит автоматическое обновление значений внешнего ключа в подчиненной таблице.

Если разрешено каскадное удаление связанных записей, то удаление записи в главной таблице приводит к удалению всех связанных записей в подчиненной таблице.

Для задания условий целостности необходимо:



– Выбрать в пункте меню **Сервис** пункт **Схема данных**. Откроется окно **Схема данных**.

- Двойным щелчком мыши на существующих линиях связи открыть диалоговое окно **Изменение связей** (рисунок 2.10.) и в нем установить флажки: Обеспечение целостности данных, Каскадное обновление связанных полей, Каскадное удаление связанных записей.
- Щелкнуть по кнопке **ОК**.
- Закрывать окно **Схема данных**. Сохранить внесенные изменения.

2.10 Ввод данных в таблицу

После задания всех полей и определения их свойств выполняется ввод данных в таблицу.

Чтобы заполнить таблицу, следует:

1 Выделить имя таблицы в окне базы данных и нажать кнопку **Открыть**. Если в данный момент открыто окно конструктора таблицы, необходимо перейти в режим отображения таблицы, нажав кнопку **Вид** на панели инструментов . При этом вид кнопки изменится на , и ее повторное нажатие приведет к возврату в окно конструктора таблицы.

2 Ввести информацию в первое поле, щелкнув в нем мышью. Для перехода к следующему полю нажмите клавишу TAB или ENTER. Затем можно продолжить ввод информации с нажатием клавиши TAB после заполнения каждой ячейки, пока не дойдете до конца записи. Для перехода на следующую запись необходимо нажать клавиши TAB или ENTER в конце строки.

3 Чтобы изменить ширину столбца, установите указатель мыши на разделительной линии между заголовками столбцов. Когда указатель мыши изменит свою форму и примет вид двунаправленной стрелки, перетащите его в нужном направлении.

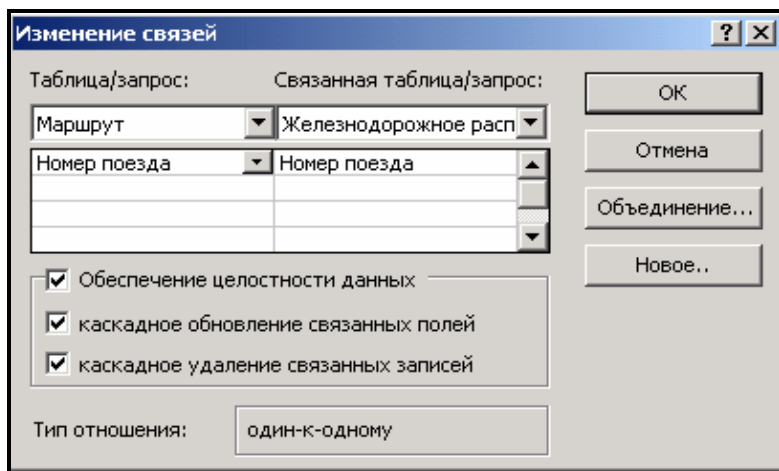


Рисунок 2.10 – Задание условий целостности

На листе данных активная запись обозначается треугольным маркером, а пустая запись – звездочкой. Для обозначения записи, в которой выполняется ввод, используется изображение карандаша. Все маркеры появляются в столбце маркировки, расположенном в левой части листа данных (рисунок 2.11).

Поле счетчика (Счетчик) заполняется автоматически. Access самостоятельно увеличивает значение переменной в этом поле, присваивая каждой записи новый номер.

	Код	тип вагона	стоимость билета
+	1	общий	8 200р.
+	2	межобластной	10 500р.
+	3	плацкартный	15 600р.
+	4	купейный	25 000р.
+	5	мягкий	56 000р.
		(Счетчик)	0р.

Запись: 6 из 6

Рисунок 2.11 – Окно таблицы в режиме заполнения

2.11 Перемещение по таблице

В программе Access существуют различные способы перемещения по таблице.

Для активизации одного поля записи достаточно выполнить щелчок на нем. Поле, которое находится в невидимой части таблицы, можно отобразить на экране с помощью полос прокрутки.

Перемещаться по таблице в любом направлении позволяют клавиши управления курсором. С помощью клавиш TAB и ENTER можно перемещаться по полям слева направо, а посредством комбинации клавиш SHIFT+TAB – обратном направлении.

Наиболее удобным средством перемещения по таблице являются кнопки из области **Запись**, расположенной в левом нижнем углу листа данных. С их помощью можно перейти к любому блоку данных. Эти кнопки соответствуют командам из подменю **Перейти** меню **Правка**. Для активизации записей с известным номером достаточно ввести этот номер в специальное **Поле номера записи** и нажать клавишу ENTER.

В программе Access по умолчанию содержимое поля полностью выделяется, если переход произведен с помощью клавиш управления курсором или клавиши TAB. Если же поле активизировано путем выполнения на нем щелчка, то в нем появится мерцающий курсор ввода, свидетельствующий о том, что можно начинать ввод.

Программа Access автоматически сохраняет каждую запись по завершении ее обработки.

2.12 Редактирование данных в таблице

Для редактирования данных в таблице необходимо открыть таблицу, установить указатель мыши щелчком на той ячейке, содержимое которой надо изменить. Указатель примет форму “I”. Внести изменения, нажать ENTER.

Двойным щелчком можно выделить слово и затем набрать другое слово, которое его заменит.

Для того чтобы выделить все содержимое поля, установите указатель мыши на левом краю поля. При этом указатель изменит форму. Выделить все поле в этот момент можно одним щелчком.

В Access существуют два режима обработки данных таблицы: режим замены и режим вставки.

В режиме замены содержимое активизированного поля полностью маркируется и удаляется в момент ввода новых данных.

В режиме вставки внутри активизированного поля появляется курсор ввода. При вводе новых данных старые не удаляются, а смещаются вправо. Ввод осуществляется в той позиции, в которой был выполнен щелчок.

Переход из режима вставки в режим замены и наоборот позволяет клавиша INS. Режим замены обозначается индикатором ЗМЩ в строке состояния окна Access.

2.13 Изменение структуры таблицы

Изменение структуры таблицы (типа данных полей, порядка их следования и т.д.) выполняется в режиме конструктора таблицы. Изменения в структуре записи заполненной таблицы чреваты потерей данных.

2.14 Сжатие данных

Access может автоматически сжимать базу данных при ее закрытии.

- Для активизации функции сжатия данных:
- в пункте меню **Сервис** выбрать команду **Параметры** и вкладку **Общие**;
 - проверить, установлен ли флажок «*Сжимать*» при закрытии;
 - щелкнуть на кнопку **ОК**.

2.15 Внешний вид таблицы

Access позволяет изменить макет таблицы.

Макет таблицы – это вид, в котором данные представлены в окне режима таблицы.

Макет таблицы можно изменить для всех таблиц данного приложения – меню **Сервис** команда **Параметры** или только для конкретной таблицы.

Для установки режима макета таблицы в пункте меню **Формат** надо выбрать команду **Режим таблицы**. Откроется окно **Формат таблицы**. Здесь можно сделать необходимые установки.

Чтобы установить вид, начертание и размер шрифта для записей таблицы, надо выбрать в пункте меню **Формат** команду **Шрифт**.

3 ФОРМЫ

3.1 Создание формы

Форма – объект базы данных, предназначенный для ввода данных в таблицу; **кнопочная форма** – для открытия других форм.

Перед созданием формы необходимо создать структуру таблицы или запроса, для которых будет создаваться форма.

Формы обеспечивают пользователям возможность ввода данных без необходимости знать, как устроена таблица. Одна форма может содержать данные из многих таблиц, благодаря чему обеспечивается возможность ввода данных в различные таблицы из одного места. Макет формы можно выполнить таким образом, что форма на экране будет выглядеть точно так же, как и первоисточник: заявление о приеме, квитанция, заполняемая клиентами, и прочие документы, используемые для сбора данных, подлежащих вводу в базу данных.

Для создания формы необходимо:

- 1 На вкладке **Формы** окна базы данных нажать кнопку **Создать**.

2 На экране появится диалоговое окно **Новая форма**, в котором программа предлагает выбрать способ создания формы (рисунок 3.1).

Существует несколько способов создания форм:

1 Автоформа (Auto Form) – автоматическое создание форм.

2 Мастер форм (Form Wizard) – создание форм с помощью мастера.

3 Конструктор (Form Design View) – создание формы вручную в режиме конструктора.

4 Мастер диаграмм – создание формы с диаграммой на основе выбранных полей таблицы.

5 Мастер сводных таблиц – создание сводной таблицы Microsoft Excel на основе таблиц и запросов Access.

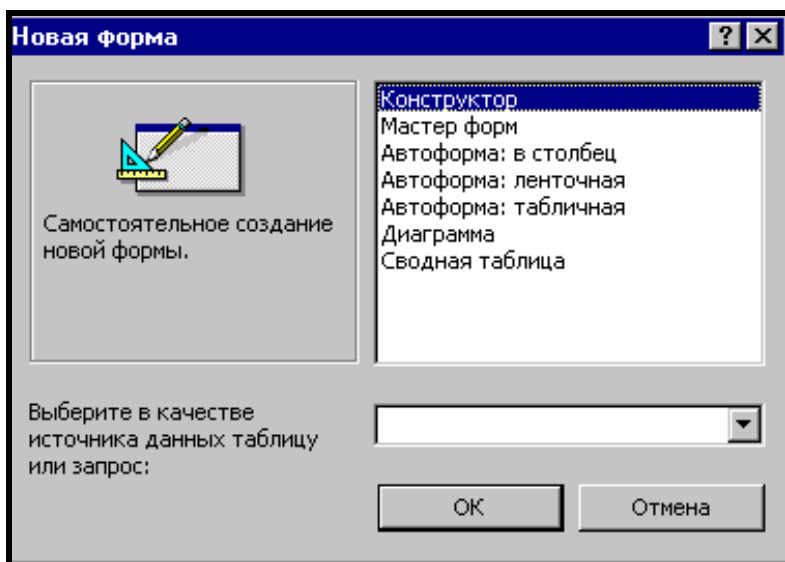


Рисунок 3.1 – Создание формы

3.1.1 Создание автоформы

Создание автоформы используется в том случае, когда пользователю необходимы формы, которые содержат всю информацию из отдельной таблицы или запроса. В автоформе каждое поле расположено на отдельной строке, с левой стороны которой отображается надпись к данному полю.

Автоформы бывают трех типов:

1 Автоформа: в столбец – для каждой записи отводится отдельная страница формы. Данная форма подходит для записей с большим числом полей. Для перехода от одной записи к другой необходимо щелкнуть на поле номера записи.

2 Автоформа: ленточная – каждая запись размещается в одной строке таблицы. Данная форма позволяет наглядно представить несколько записей.

3 Автоформа: табличная – имеет вид таблицы и выступает в качестве подчиненной формы в составных формах.

Для создания автоформы необходимо:

1 На вкладке **Формы** окна базы данных нажать кнопку **Создать**.

2 Выбрать способ создания формы – автоформа.

3 В нижней части окна **Новая форма** из списка выбрать таблицу или запрос, на основе которого будет создана форма; нажать кнопку **OK** (рисунок 3.2).

4 На экране откроется готовая форма (рисунок 3.3). При ее закрытии Access предложит сохранить форму, указав ее имя.

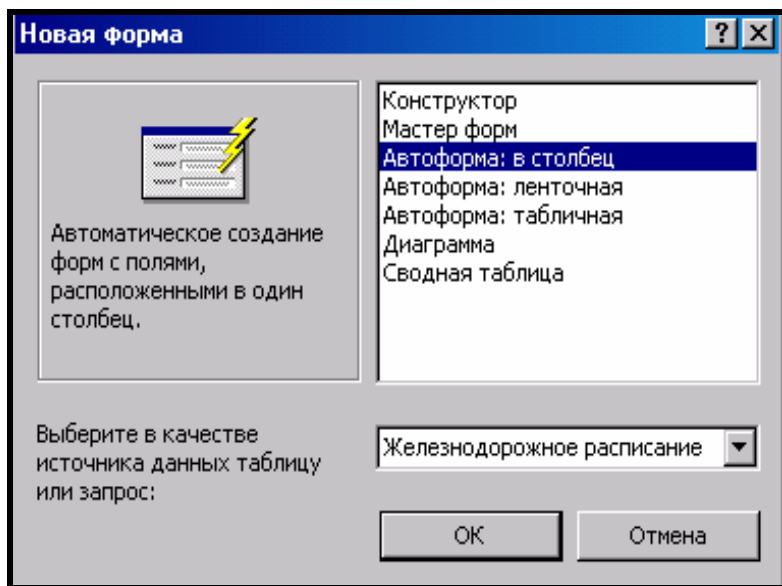


Рисунок 3.2 – Создание формы с помощью автоформы в столбец

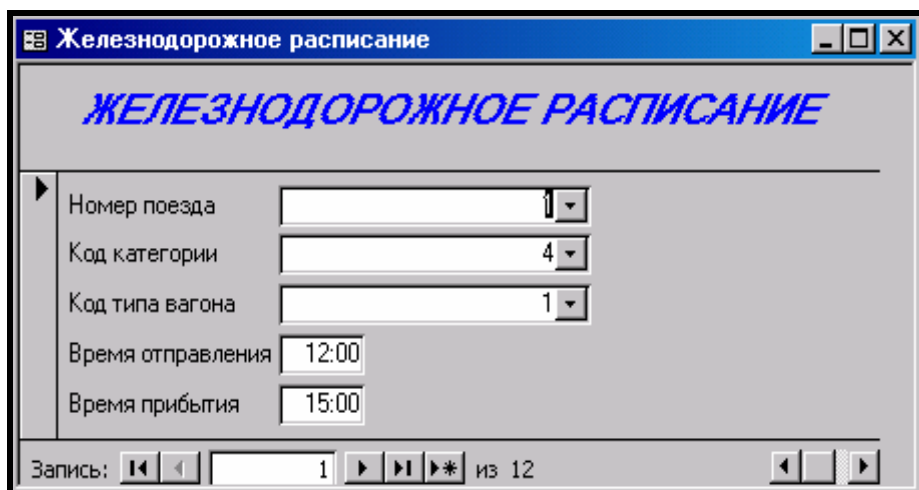


Рисунок 3.3 – Автоформа в столбец

3.2 Ввод и редактирование данных в форме

В табличных формах информация вводится и редактируется так же, как это выполняется в таблицах в режиме просмотра. Поле для редактирования выбирается установкой указателя мыши на левом краю поля с последующим щелчком.

В ленточных формах и в формах в столбец данные, подлежащие редактированию, выделяются перетаскиванием мыши, после чего вводятся правильные данные.

Переход из поля в поле в форме осуществляется клавишей **Tab** или клавишами управления курсором.

Для перехода к следующей записи надо нажать клавишу **Tab**, находясь на последнем элементе управления формы.

Во всех формах для перехода между записями можно использовать **кнопки перехода**.

Для удаления записи из формы ее необходимо маркировать, щелкнув в столбце маркировки (выделение записи отменяется при выполнении щелчка вне столбца маркировки), и нажать клавишу **Del**.

3.3 Мастер форм

Использование мастера форм позволяет включать в форму поля из различных таблиц.

Для создания формы с помощью мастера необходимо:

1 Открыть вкладку **Формы** окна базы данных и нажать кнопку **Создать**.

2 В появившемся окне **Новая форма** задать создание формы с помощью мастера и нажать кнопку **ОК** (рисунок 3.4).

3 В **первом** диалоговом окне мастера (рисунок 3.5) выбрать в списке **Таблицы\запросы** таблицу, из которой в форму будут включены поля, а в списке **Допустимые поля** выбрать поля для подготавливаемой формы. С помощью кнопок > или >> перенести выбранные поля в список **Выбранные поля**.

4 В списке **Таблицы\запросы** последовательно можно выбирать таблицы, из которых необходимо включать поля в форму. При включении в форму полей из различных таблиц будет создана многотабличная форма.

5 После отбора всех полей из таблиц в форму нажать кнопку **Далее** для перехода в следующее окно мастера форм.

6 Во **втором** окне мастера форм необходимо выбрать **вид** формы. Для этого активизируйте один из переключателей (в один столбец; ленточный; табличный; выровненный) и нажмите кнопку **Далее**.

7 В **третьем** окне мастера задать **стиль** оформления и нажать кнопку **Далее**.

8 В последнем окне мастера ввести **имя формы** и выбрать одну из опций: открыть форму для просмотра и ввода данных или изменить макет формы. Нажать кнопку **Готово**.

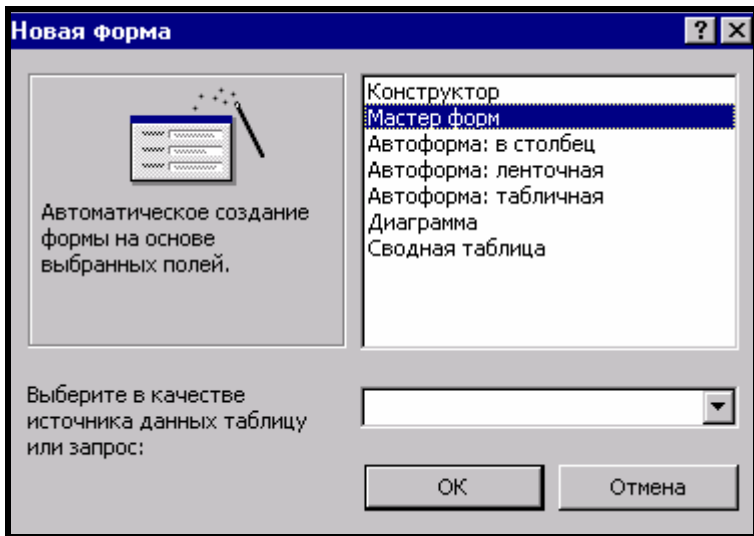


Рисунок 3.4 – Создание формы с помощью мастера

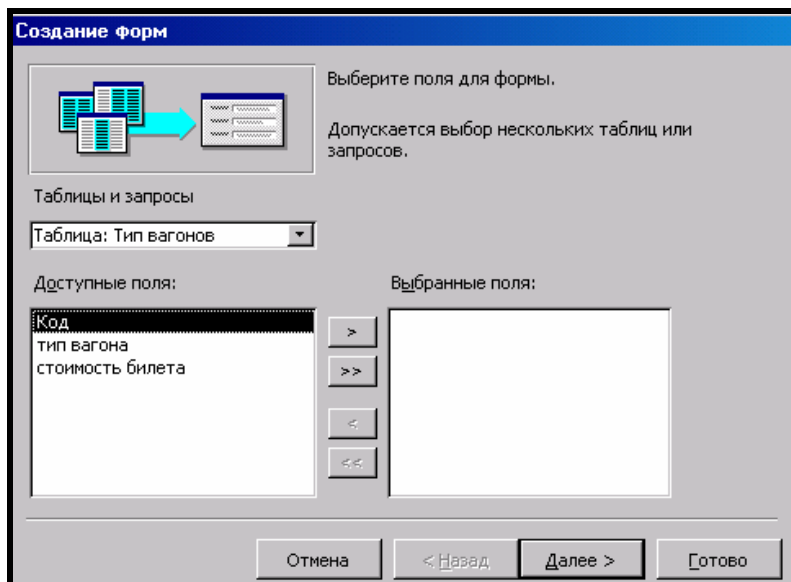


Рисунок 3.5 – Выбор полей

3.3.1 Создание подчиненной формы

Подчиненная форма – это вложенная форма, позволяющая отображать данные из связанной таблицы в тех случаях, когда одной записи из главной таблицы может соответствовать несколько записей подчиненной таблицы (связь один-ко-многим). Подчиненная форма внедряется в главную связь (один-ко-многим). Если главная форма уже достаточно сложна, то можно выбрать отображение подчиненной формы не как вложенной, а как связанной. В этом случае в главной форме содержится кнопка, щелчком на которой открывается подчиненная форма.

Чтобы для таблиц, связанных один-ко-многим, создать форму, необходимо:

- 1 Открыть вкладку **Формы** окна базы данных и нажать кнопку **Создать**.
- 2 Запустить мастер форм.
- 3 В списке **Таблицы\запросы** выбрать основную таблицу и скопировать нужные поля в список **Выбранные поля**.

4 В списке **Таблицы|запросы** выбрать подчиненную таблицу и скопировать нужные поля в список **Выбранные поля**; нажать кнопку **Дале**.

5 Выбрать вид представления данных: хотите ли вы, чтобы форма основывалась на главной или на связанной форме, при этом в качестве исходной можно выбрать главную или подчиненную таблицу (рисунок 3.6).

6 Опции **Подчиненные формы** и **Связанные формы** определяют отношения между таблицами. Выбрав опцию **Подчиненные формы**, создают главную форму, содержащую подчиненную форму. Если выбрать параметр **Связанные формы**, то в главной форме будет содержаться кнопка, щелчком на которой открывается подчиненная форма. Нажать кнопку **Далее**.

7 Выбрать **внешний вид** подчиненной формы. Нажать кнопку **Далее**.

8 Выбрать **стиль оформления** формы. Нажать кнопку **Далее**.

9 Ввести **заголовок главной формы**. Для подчиненной формы оставить предложенное программой имя.

10 Установить одну из опций: открыть форму для просмотра или ввода данных (рисунок 3.7) или изменить макет формы. При выборе второй опции и нажатии кнопки **Готово** на экране появится проект формы.

Перемещение от одной записи к другой в главной и подчиненной форме осуществляется с помощью кнопок перехода между записями:

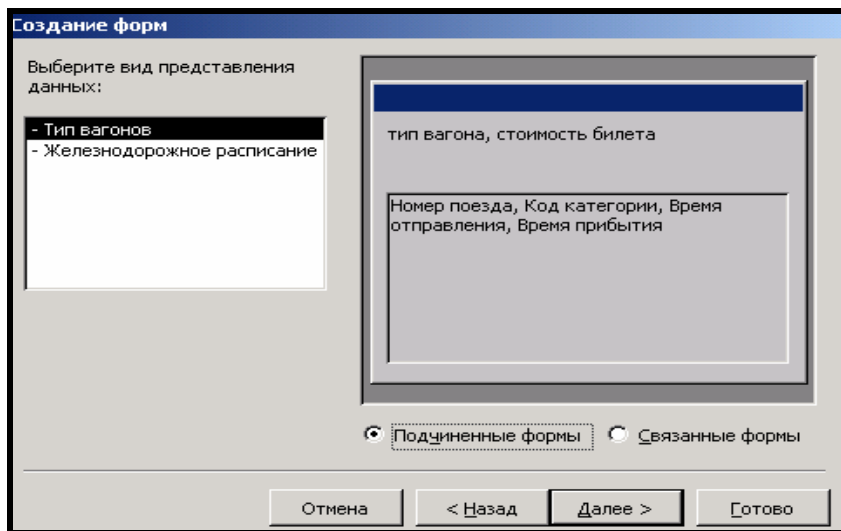


Рисунок 3.6 – Создание подчинённой формы

Тип вагонов1

ДАННЫЕ О ПОЕЗДАХ

тип вагона:

стоимость билета:

Железнодорожное расписание

	Номер поезда	Код категории	Время отправления	Время прибытия
	1	4	12:00	15:00
	649	3	9:00	21:00
▶	0	0		

Запись: из 3

Запись: из 5

Рисунок 3.7 – Макет подчиненной формы для ввода данных

3.4 Конструктор форм

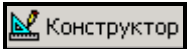
Режим конструктора используется для создания форм вручную.


Для создания формы в режиме конструктора необходимо выполнить следующие действия:

- 1 В окне базы данных выбрать вкладку **Формы**.
- 2 Нажать кнопку **Создать**.
- 3 В диалоговом окне **Новая форма** выбрать пункт **Конструктор**.
- 4 Выбрать имя таблицы или запроса, содержащих данные, на основе которых будет создана форма. Если форма не будет содержать данных (например, если форма будет использоваться как кнопочная для открытия других форм или отчетов или как специальное диалоговое окно), ничего не выбирайте в списке. **Примечание.** Форма, использующая данные из нескольких таблиц, должна быть основана на запросе, включающем эти таблицы.
- 5 Перетащить поля из таблицы в область данных.

Сохранить таблицу, указав имя.

6 Изменения в форму можно внести в режиме Конструктора.

Чтобы открыть форму в режиме конструктора, необходимо нажать соответствующую кнопку на панели инструментов .

Окно конструктора содержит горизонтальные и вертикальные линейки, панель форматирования и панель инструментов. Дополнительно можно открыть панель элементов (кнопка  на панели инструментов), с помощью которой в форму можно ввести новые объекты (рисунок 3.8).

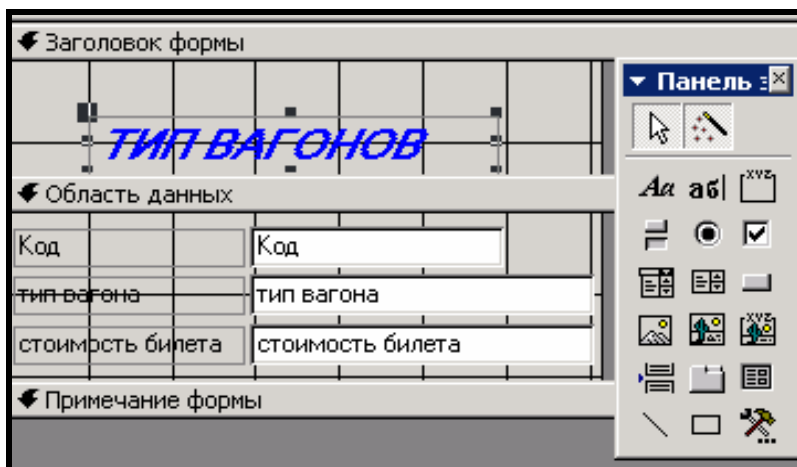


Рисунок 3.8 – Вид формы в конструкторе

Форма в режиме конструктора имеет три раздела и несколько различных элементов управления (рисунок 3.8):

1 **Заголовок формы** – этот раздел находится в начале первой страницы формы и используется для размещения заголовка формы.

2 **Примечание формы** – этот раздел расположен в конце последней страницы формы и используется для размещения полезных советов, адресованных пользователю, а также другой полезной информации.

3 **Область данных** – это основная область, в которой располагаются записи данных.

Если разделы заголовка и примечания в форме не отображаются, необходимо выбрать команду **Вид – Заголовок\примечание** формы.

3.4.1 Изменение структуры формы

Для изменения размера области данных надо переместить указатель мыши в конец области данных и установить его там, где начинается раздел примечаний. Указатель изменит свой вид (примет вид двунаправленной стрелки). Перетащить разделительную полосу раздела **«Примечание формы»** вниз, увеличивая тем самым размер раздела **«Область данных»**.

Для того чтобы увеличить размер раздела заголовка, необходимо установить указатель мыши между разделом заголовка и областью данных (указатель мыши примет вид двунаправленной стрелки), после чего перетащить вниз разделительную полосу **«Области данных»**.

3.5 Элементы управления

Элементами управления в формах называются все объекты, с которыми можно выполнять какие-то действия, в том числе и текстовые поля.

Для выделения объекта достаточно щелкнуть на нем мышью – вокруг объекта появятся маркеры выделения.

Для выделения нескольких элементов управления необходимо установить указатель на одной из линеек (указатель изменит свой вид на жирную стрелку) и нажать кнопку мыши. Линия пройдет через форму. После того как кнопка мыши будет отпущена, все элементы управления, через которые прошла линия, будут выделены.

Если элементы управления, которые необходимо выделить, не располагаются рядом, можно выделить один элемент, а затем выделять другие, удерживая клавишу SHIFT.

Удаление выделенных элементов управления производится клавишей Delete.

Элементы управления могут быть **связанными**, **свободными** или **вычисляемыми**.

Связанный элемент управления присоединен к полю базовой таблицы или запроса. Такие элементы управления используются для отображения, ввода или обновления значений из полей базы данных.

Для **вычисляемого элемента** управления в качестве источника данных используется выражение. В выражении могут быть использованы данные из

поля базовой таблицы или запроса для формы или отчета, а также данные другого элемента управления формы или отчета. Для свободного элемента управления источника данных не существует.

Свободные элементы управления используются для вывода на экран данных, линий, прямоугольников и рисунков.

Элементы управления, которыми может пользоваться разработчик, представлены на Панели элементов. Ее открывают щелчком на соответствующей кнопке



панели инструментов Microsoft Access или командой **Вид > Панель элементов** (рисунок 3.9).

Надписи предназначены для отображения в форме или отчете описательных текстов, таких как заголовки, подписи или краткие инструкции. В надписях не выводятся значения полей или выражений; они всегда являются свободными и не меняются при переходе от записи к записи.

Поля используются в форме или отчете для отображения данных из таблицы, запроса или инструкции SQL. Поле такого типа называют присоединенным, потому что оно связано с данными в поле. Кроме того, существуют свободные поля. Например, можно создать свободное поле для отображения результатов вычислений или для приема данных, вводимых пользователем. Содержимое свободного поля нигде не сохраняется.

В форме или отчете **выключатель** может быть использован как отдельный элемент управления, в котором отображаются значения логического поля из базовой таблицы, запроса.

В форме или отчете **переключатель** может быть использован как элемент управления, в котором отображаются значения логического поля из базовой таблицы, запроса.

В форме или отчете **флажок** может быть использован как отдельный элемент управления, в котором отображаются значения логического поля из базовой таблицы, запроса.

Во многих случаях удобнее выбрать значение из списка, чем вводить конкретное значение с клавиатуры по памяти. **Поле со списком** позволяет выбрать любой из этих способов ввода значения, не требуя при этом значительного места в форме. Поле со списком является комбинацией двух элементов: поля и раскрывающегося списка.

Во многих случаях удобнее выбрать значение из **списка**, чем вводить конкретное значение по памяти. Кроме того, выбор из списка позволяет быть уверенным, что введенное значение является допустимым.

Кнопки используются в формах для выполнения определенного действия или ряда действий. Например, можно создать в форме кнопку, открывающую другую форму.

Подчиненная форма – это форма, находящаяся внутри другой формы.

Первичная форма называется главной формой, а форма внутри формы называется подчиненной формой.

Подчиненным отчетом называют отчет, вставленный в другой отчет. При комбинировании отчетов один из отчетов является главным.



Рисунок 3.9 – Панель элементов

Набор вкладок используется для создания многостраничной формы.

Выбор элемента управления выполняется одним щелчком на его значке в Панели элементов, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его присоединенная надпись. По умолчанию эта надпись стандартная, например для переключателей это Переключатель1, Переключатель2 и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Для создания флажков, переключателей и выключателей выполнить следующие действия: на панели элементов выбрать необходимый элемент управления; в списке полей выбрать соответствующее поле и, не отпуская мышью, перетащить его в область данных. Для выключателя надо создать надпись.

3.5.1 Перемещение элементов управления

Для перемещения элементов управления необходимо:

1 Выделить элементы управления. Передвинуть указатель мыши к краю выделенного элемента, стараясь не попасть на маркеры изменения размера. Указатель должен принять вид открытой ладони.

2 Нажать левую клавишу мыши и перетащить элемент управления в новое положение.

3 При попытке вывести элемент управления за пределы текущей области формы область формы увеличивается.

Если установить указатель мыши на левом верхнем маркере выделения элемента управления или связанной с ним надписи, то указатель примет вид указательного пальца. В таком случае перетаскивание элемента управления приводит к раздельному перемещению элемента управления и связанной с ним надписи.

3.5.2 Изменение размеров элементов управления

Изменение размеров элементов управления выполняется так же, как это принято для графических объектов, перетаскиванием маркеров изменения размера, которые расположены по углам и сторонам выделенных элементов управления.

3.5.3 Создание заголовка формы

Для создания заголовков и других свободных надписей в форме служит элемент управления **Надпись (Aa)**.

Для создания заголовка необходимо:

Щелкнуть на панели элементов по кнопке **Надпись**.

1 Передвинуть указатель мыши в область заголовка, он примет вид прописной буквы «А» с перекрестием.

2 Щелкнуть на том месте, где необходимо начать ввод текста.

3 Когда текст введен, щелкнуть на любом месте формы, чтобы закрыть элемент управления.

Выделить созданный элемент и использовать панель форматирования для форматирования надписи.

3.5.4 Задание стиля оформления элементов управления

Для задания стиля оформления элементов управления необходимо:

1 Выделить элементы управления.

2 Для оформления границ выделенных элементов используются кнопки панели форматирования: **Цвет** линии\границы, **Толщина** линии\границы, **Оформление**.

3.5.5 Изменение расположения элементов управления

Фон формы образуется сеткой горизонтальных и вертикальных линий и точек. При размещении элемента управления в форме он автоматически выравнивается относительно линий сетки как по горизонтали, так и по вертикали.

Чтобы выровнять несколько элементов управления, надо их выделить и в пункте меню **Формат** выбрать команду **Выровнять**, после чего в раскрываемом списке выбрать тип выравнивания.

3.5.6 Добавления элемента управления в форму

Если в таблицу добавлено новое поле, то в существующей форме, основанной на данной таблице, это поле будет отсутствовать. Чтобы добавить в форму новое поле, необходимо:

– выбрать команду **Вид-Список полей**;

– перетащить нужное поле из списка полей в форму и поместить его в выбранное место;

отформатировать созданный элемент управления.

3.5.7 Создание элементов управления

С помощью кнопок панели элементов можно создавать различные элементы управления: новые поля, переключатели, флажки, поля со списком и т.д.

Для создания элемента управления *поле со списком* необходимо:

- 1 Открыть форму в режиме конструктора.
- 2 Щелкнуть по кнопке *поле со списком* на панели элементов.
- 3 Перетащить из списка полей в область данных поле, которое будет полем со списком. Откроется окно мастера «Создание полей со списком».

4 В *1-м шаге* мастера – выбрать способ, каким поле со списком будет получать значения: объект «поле со списком» будет использовать значения из таблицы или запроса; будет введен фиксированный набор данных; поиск записи в форме на основе значения, которое содержит поле со списком; Нажать кнопку *Далее*.

5 Во *2-м шаге* мастера – если выбран способ «а», то необходимо выбрать таблицу или запрос со значениями, которые будет содержать поле со списком. Нажать кнопку *Далее*.

6 В *3-м шаге* мастера – выбрать поле, которое содержит значения, которые следует включить в поле со списком. Отобранные поля (или поле) станут столбцами в объекте «поле со списком». Нажать кнопку *«Далее»*.

7 В *4-м шаге* мастера задать ширину столбца. Нажать кнопку *«Далее»*.

8 В *5-м шаге* мастера установить опцию «Сохранить в поле» и выбрать в раскрывающемся списке имя поля. Нажать кнопку *«Далее»*.

9 В *6-м шаге* мастера задать подпись, которую будет содержать поле со списком. Нажать кнопку *«Готово»*. При закрытии формы подтвердить изменения макета или структуры формы. Если в 1-м шаге мастера выбрать «фиксированный набор», то необходимо задать набор значений поля, которые будут использованы.

3.6 Создание кнопочной формы

Кнопки в форме используются для выполнения действия или набора действий. Например, можно создать кнопку, открывающую другую форму, запрос, отчет и т.п.

Для создания кнопочной формы необходимо:

1 В окне базы данных на вкладке **Формы** нажать кнопку **Создать**



2 В окне **Новая форма** выбрать режим создания формы **Конструктор**. Проверить наличие в окне формы **Панели элементов**. Для вывода панели элементов на экран выбрать команду **Вид – Панель элементов**

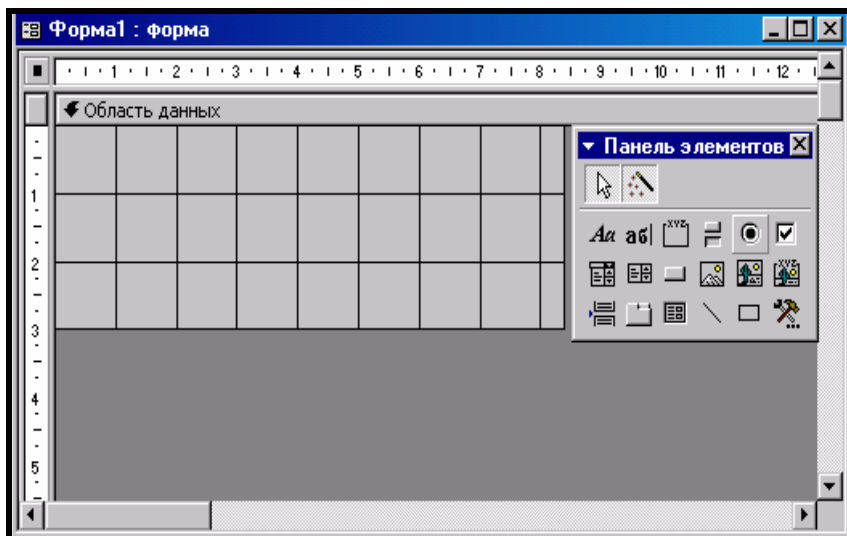


Рисунок 3.10 – Создание кнопочной формы

3 Активизировать на **Панели элементов** команду **Мастера**



4 Выбрать элемент управления **Кнопка** (кликнуть по значку кнопки ле-

вой кнопкой мыши)



5 Переместить кнопку в область данных формы (кликнуть левой кнопкой мыши в свободном месте области данных).

6 В появившемся окне (рисунок 3.11) из списка **категории** выбрать нужную категорию, из списка **действия** выбрать действие, которое должна выполнять кнопка. Нажать кнопку **Далее**.

7 В следующем окне (рисунок 3.12) выбрать объект, для которого создается кнопка. Нажать кнопку **Далее**.

8 В следующем окне выбрать опцию **Открыть форму и показать все записи**. Нажать кнопку **Далее**.

9 В следующем окне (рисунок 3.13) возможен выбор одного из двух режимов:

- переключатель установлен в положение **‘ТЕКСТ’**, тогда на кнопке будет выведен поясняющий текст, например **Открыть форму**
- переключатель установлен в положение **‘РИСУНОК’**, тогда на кнопке будет выведен рисунок. Нажать кнопку **Далее**.

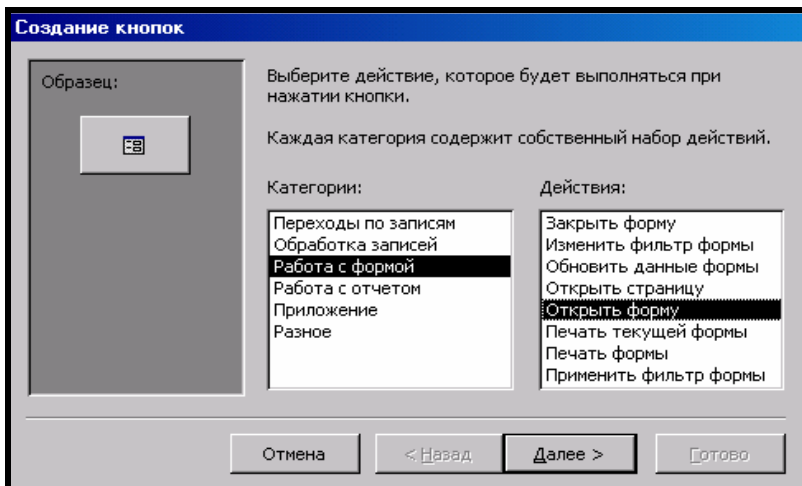


Рисунок 3.11 – Создание кнопок

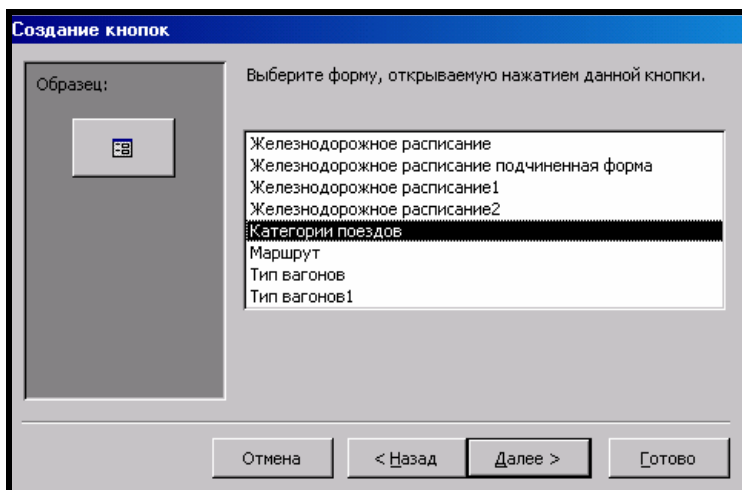


Рисунок 3.12 – Выбор объекта

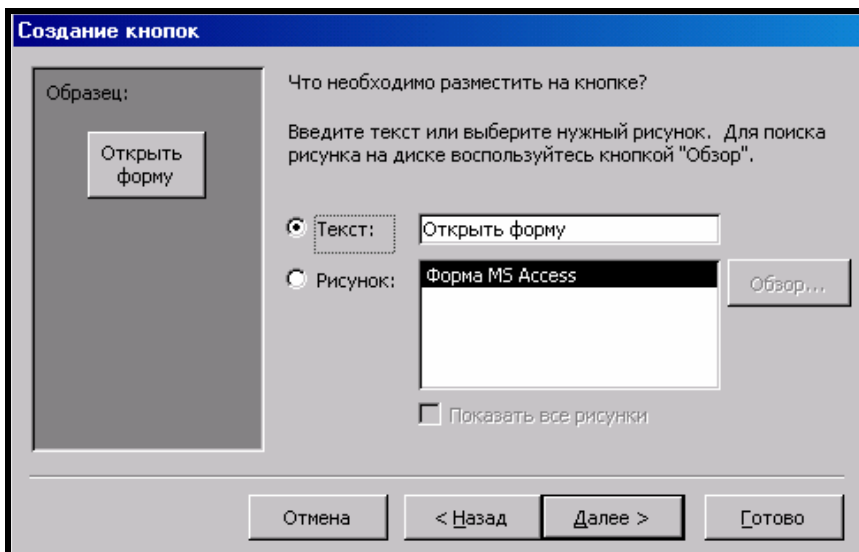


Рисунок 3.13 – Выбор надписи на кнопке

10 В следующем окне нажать кнопку *Готово*.

ПРИМЕЧАНИЕ 1 Для создания кнопок, открывающих запросы и макросы, из списка *КАТЕГОРИИ* необходимо выбрать *РАЗНОЕ*.

ПРИМЕЧАНИЕ 2 Для создания кнопок, открывающих таблицы и модули сначала надо создать модуль.

3.6.1 Создание макроса

1 В окне базы данных выбрать из списка объектов *МАКРОСЫ* и команду *СОЗДАТЬ*.

2 Из списка *МАКРОКОМАНДА* выбрать действие, которое должен выполнить макрос (рисунок 3.14) например *Открыть Таблицу*.

3 В поле *Аргументы макрокоманды* (рисунок 3.15) выбрать имя таблицы, для которой создается макрос.

4 Закрыть макрос, сохранив его.

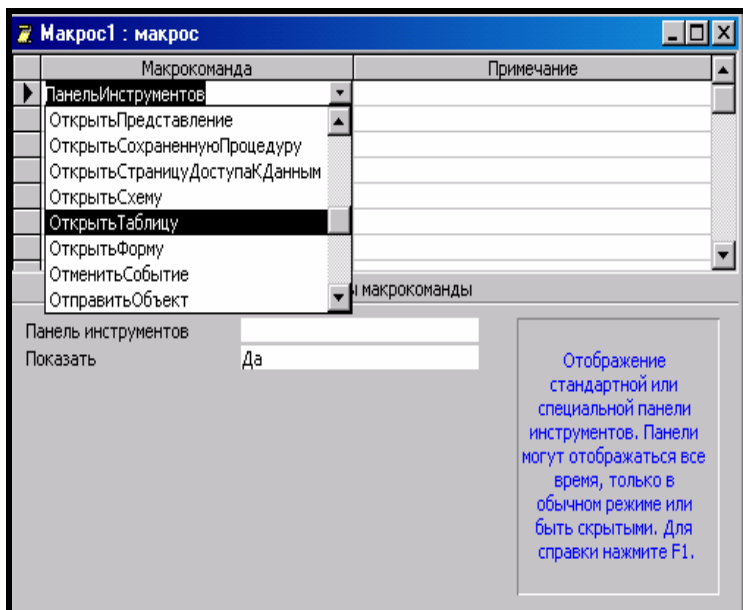


Рисунок 3.14 – Выбор макрокоманды

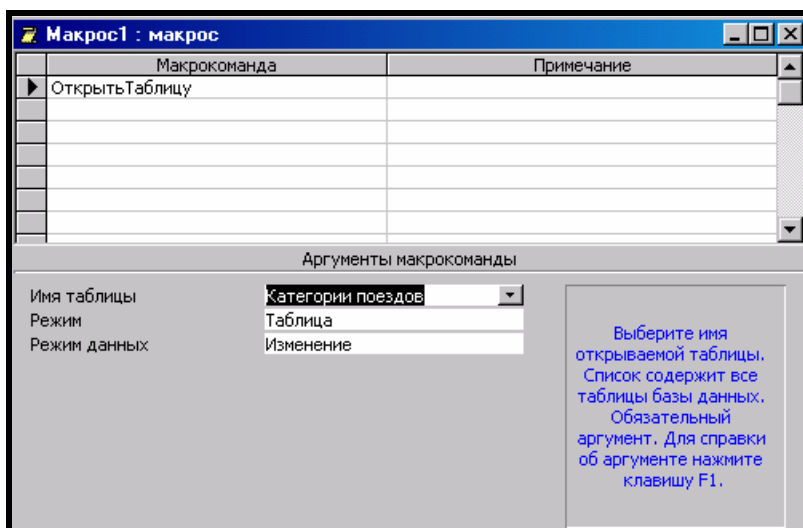


Рисунок 3.15 – Выбор аргумента макрокоманды

4 ЗАПРОСЫ

Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным. В Microsoft Access есть несколько типов запросов:

1 **Запросы на выборку.** Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

2 **Запросы с параметрами** Запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например двух дат. Затем Microsoft Access может вернуть все записи, приходящиеся на интервал времени между этими датами. Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным.

3 **Перекрестные запросы.** Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

4 **Запросы на изменение.** Запросом на изменение называют запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение:

– **Запрос на удаление записей.** Данный запрос удаляет группу записей из одной или нескольких таблиц.

– **Запрос на обновление записей.** Данный запрос вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 5 процентов увеличивается зарплата сотрудников определенной категории.

– **Запрос на добавление записей.** Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них.

– **Запрос на создание таблицы.** Данный запрос создает новую таблицу на основе всех или части данных из одной или нескольких таблиц.

5 *Запросы SQL*. Запрос SQL — это запрос, создаваемый при помощи инструкций SQL.

4.1 Создание запросов на выборку

Запрос на выборку является наиболее часто используемым типом запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы.

4.1.1 Создание запроса на выборку при помощи мастера



Пример 1. Создайте запрос на выборку, выводящий данные из полей *№ поезда*, *Время отправления*, *Время прибытия*. Имя запроса '*Запрос 1*'.

Для этого, выполните следующие действия:

- 1 В окне базы данных выберите вкладку *Запросы* и команду *Создать*;
- 2 В появившемся диалоговом окне (рисунок 4.1) выберите команду

Простой запрос;

3 В появившемся диалоговом окне (рисунок 4.2) выполните следующие действия:

- выберите из списка *Таблицы и Запросы* таблицу или запрос, поля из которого будут использованы в запросе;
- перенесите поля для запроса из списка *Доступные поля* в список *Выбранные поля*, при помощи кнопок  
- нажмите кнопку *Далее*;

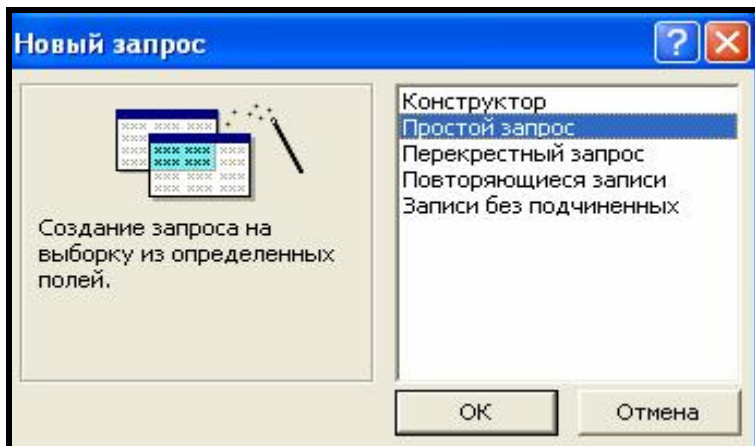


Рисунок 4.1 – Выбор режима создания запроса

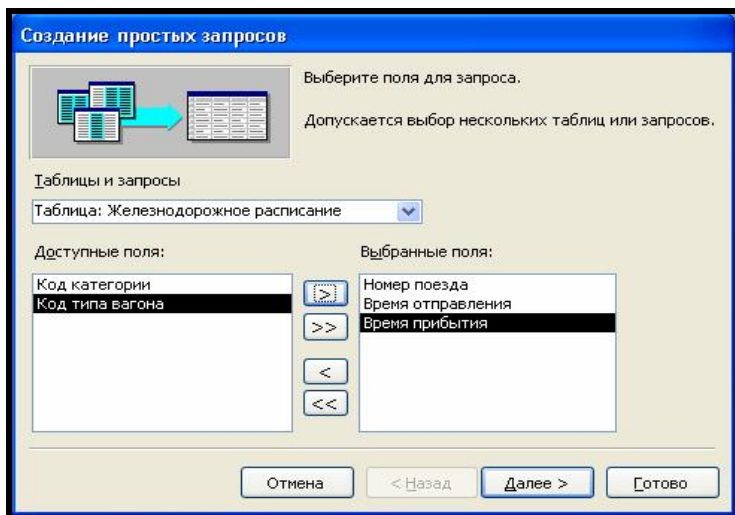
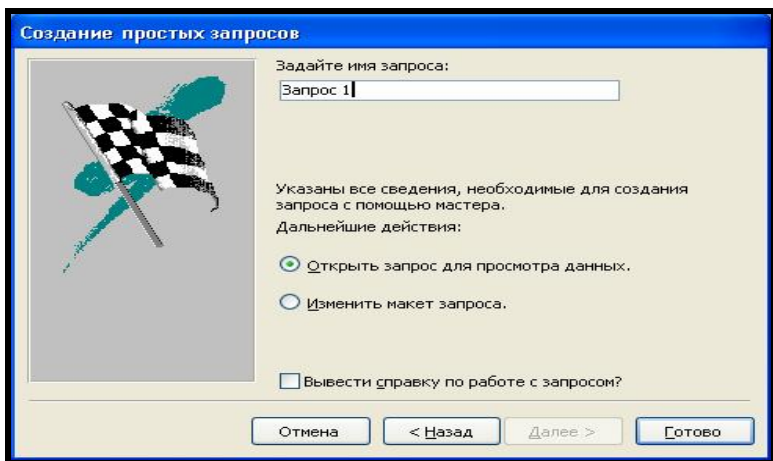


Рисунок 4.2 – Выбор полей для запроса

4 В последнем диалоговом окне дайте имя запросу и нажмите кнопку **Готово**:



4.1.2 Создание запроса на выборку без помощи мастера

Пример 2. Создайте запрос на выборку, выводящий данные из полей *Тип вагона*, *Стоимость билета* из таблицы *Тип вагонов*. Отсортируйте записи по полю «Стоимость билета». Имя запроса '*Запрос 2*'.

Для этого, выполните следующие действия:

- 1) В окне базы данных выберите вкладку *Запросы* и команду *Создать*;
- 2) В появившемся диалоговом окне (рисунок 4.1) выберите команду *Конструктор*;
- 3) В окне диалога *Добавление таблицы* (рисунок 4.3) перейдите на вкладку, содержащую объекты, данные из которых будут использованы в запросе. Двойным щелчком мыши выберите объекты, которые нужно добавить в окно запроса, и нажмите кнопку *Закреть*;
- 4) В бланке запроса (рисунок 4.4) в строке *Поле* первого столбца выведите название первого поля, данные из которого будут использоваться в запросе. Для этого курсор установите в строку *Поле* первого столбца. В появившемся раскрывающемся списке (стрелка справа) выберите название поля, записи из которого будут использоваться в запросе;

Выполните вышеуказанные действия для всех полей, записи из которых будут использоваться в запросе. Каждое поле выводите в последующем столбце;

5) Просмотрите результат вы-

полнения запроса, нажав кнопку **Вид**  на панели инструментов (ПИ);

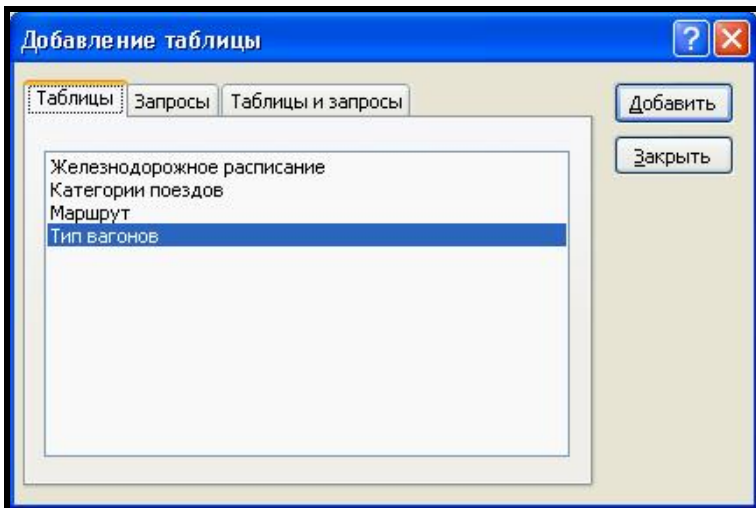


Рисунок 4.3 – Добавление таблицы

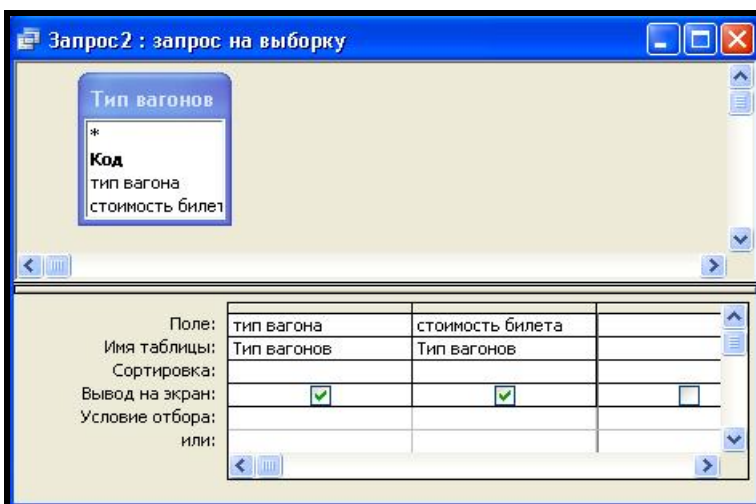


Рисунок 4.4 – Вывод полей в бланк запроса

7) Чтобы отсортировать записи по полю (в данном случае по полю «Стоимость билета») необходимо для поля «стоимость билета» в строке **Сортировка** в раскрываемом списке выбрать **по возрастанию** (рисунок 4.5).

8) Закройте запрос **X**, выполняя его сохранение.

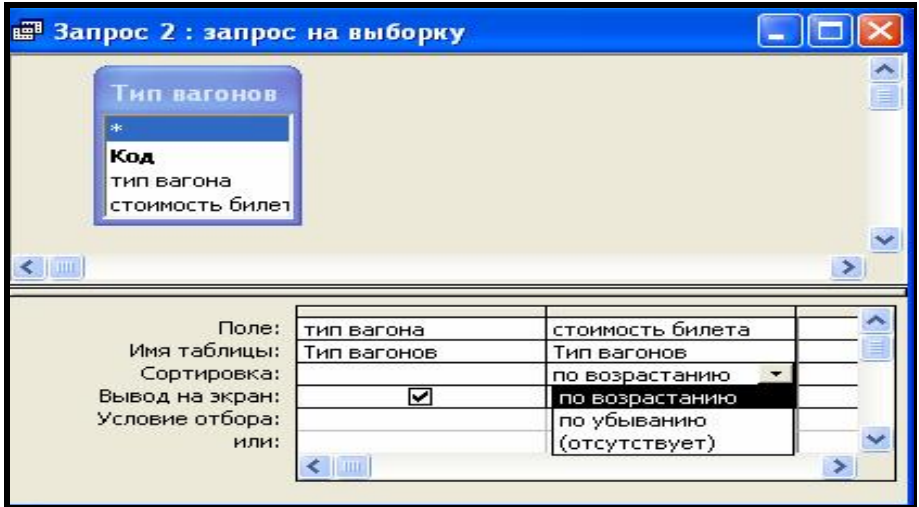
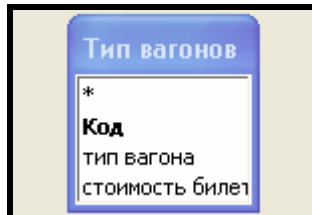


Рисунок 4.5 – Сортировка записей по полю

4.2 Структура бланка запроса

Запрос в режиме конструктора состоит из следующих основных частей, представленных на рисунке 4.4:

- списка полей объектов БД, на основе которых создается запрос:



- бланка запроса:

Поле:	тип вагона	стоимость билета	
Имя таблицы:	Тип вагонов	Тип вагонов	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			
или:			

Бланк запроса представляет собой таблицу, столбцы которой являются полями запроса, а строки свойствами полей запроса. Рассмотрим свойства полей запроса:

- **поле** – в данной строке выводится название поля запроса;
- **имя таблицы** – в данной строке выводится имя таблицы, в которой находится соответствующее поле запроса;
- **сортировка** – в данной строке выбирается тип сортировки записей соответствующего поля запроса. Возможные варианты сортировки выбираются из списка:

Поле:	тип вагона	стоимость билета	
Имя таблицы:	Тип вагонов	Тип вагонов	
Сортировка:			
Вывод на экран:	по возрастанию	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	по убыванию		
или:	(отсутствует)		

- **вывод на экран** – данное свойство регулирует вывод на экран записей соответствующего поля: – не выводить, - выводить;
- **условие отбора...или** – в данной строке устанавливаются условия отбора записей соответствующего поля в запросе.

4.3 Создание запроса с параметрами, запрашивающего ввод условий отбора при каждом запуске

Запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например, условие для возвращения записей или значение, которое требуется вставить в поле.

Пример 3. На основе запроса 2, создайте запрос с параметром по полю *Тип вагона*. Имя запроса '*Запрос 3*'.

Для этого выполните следующие действия:

- 1 Создайте новый запрос на выборку или откройте существующий запрос в режиме конструктора;
- 2 Для каждого поля, которое предполагается использовать как параметр, введите в ячейку строки **Условие отбора** бланка запроса (рисунок 4.6) текст - приглашение, заключенный в квадратные скобки. Это приглашение будет выводиться при запуске запроса. Текст - приглашение должен отличаться от имени поля, но может включать его;

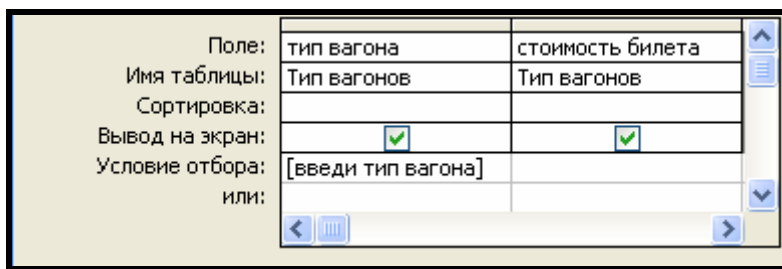



Рисунок 4.6 – Создание параметрического запроса

- 3 Для просмотра результатов нажмите кнопку **Вид**  на ПИ;
- 4 В появившемся диалоговом окне (рисунок 4.7) введите конкретное значение параметра;

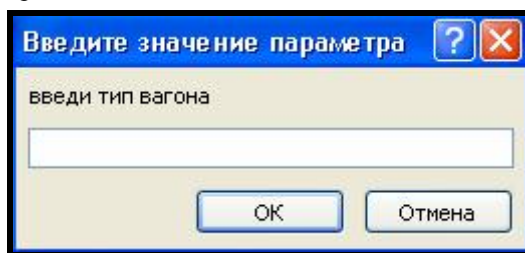



Рисунок 4.7 – Выполнение параметрического запроса

- 5 Вернитесь в режим конструктора запроса, снова нажав кнопку **Вид**  на ПИ;
- 6 Закройте запрос, выполняя его сохранение.

Примечание 1:

- чтобы открыть существующий запрос, выделите его в окне БД и выберите команду **Открыть**;
- при сохранении существующего запроса с новым именем используется команда **Файл – Сохранить как**;
- для поля, которое выводит даты, можно ввести приглашения следующего вида: "Введите начальную дату:" и "Введите конечную дату:", чтобы задать границы диапазона значений. Для этого в ячейке **Условие отбора** соответствующего поля наберите:
 - **Between [Введите начальную дату:] And [Введите конечную дату:]**;
 - чтобы запросить у пользователя один или несколько символов для поиска записей, которые начинаются с этих символов или содержат их, создайте запрос с параметрами, использующий оператор *Like* и подстановочный знак “звездочка” (*).

Например,

- выражение *Like* [Введите первый символ для поиска:] & * выполняет поиск слов, начинающихся с указанного символа;
- выражение *Like* * & [Введите любой символ для поиска:] & * выполняет поиск слов, которые содержат указанный символ.

4.4 Создание итоговых запросов

Пример 4. Создайте итоговый запрос, вычисляющий минимальную и максимальную стоимость билета на каждый поезд. Имя запроса **‘Запрос 4’**.

Задачу будем решать в два этапа:

- Создадим запрос на выборку **‘Запрос 4(1)’**, в котором выведем данные из полей *Тип вагона*, *Стоимость билета*, *Номер поезда* и создадим поле *Приведенная стоимость*.
- Создадим итоговый запрос **‘Запрос 4’** по заданию.

Для решения **1 этапа** выполните следующие действия:

1 В окне базы данных выберите вкладку **Запросы** и команду **Создать**;

2 В появившемся диалоговом окне (рисунок 4.1) выберите команду **Конструктор**;

3 Выполните действия 3,4,5 из п.4.1.2., переместив в бланк запроса требуемые поля;

- 4 Сохраните запрос;
- 5 Установите курсор в строку **Поле** нового столбца и выберите команду **Построить** из контекстного меню;
- 6 Появится окно построителя выражений (рисунок 4.10);
- 7 Используя методику, описанную в п. 4.5, создайте выражение, вычисляющее приведенную стоимость, и назовите полученное в запросе поле **Приведенная стоимость**;

Приведенная стоимость: [Тип вагонов];[стоимость билета]*[Маршрут];[Козффициент на стоимость билета]
<input checked="" type="checkbox"/>

- 8 Закройте запрос, сохранив его.
- Для решения **этапа 2** выполните следующие действия:
- 1 В окне базы данных выберите вкладку **Запросы** и команду **Создать**;
 - 2 В появившемся диалоговом окне выберите команду **Простой запрос** (рисунок 4.1);
 - 3 В появившемся диалоговом окне (рисунок 4.2) выполните следующие действия:
 - выберите таблицу или запрос, поля из которого будут использованы в запросе из списка **Таблицы и Запросы**;
 - поля для запроса перенесите из списка **Доступные поля** в список **Выбранные поля**, при помощи кнопок **>** **>>**. В данном примере выбираем поля **Номер поезда**, **Приведенная стоимость** из запроса ‘Запрос 4(1)’;
 - после выбора полей нажмите кнопку **Далее**;
 - 4 В появившемся диалоговом окне (рисунок 4.8) установите переключатель в положение **Итоговый** и нажмите кнопку **Итоги**, для вычисления итоговых значений числовых полей, использующихся в запросе;

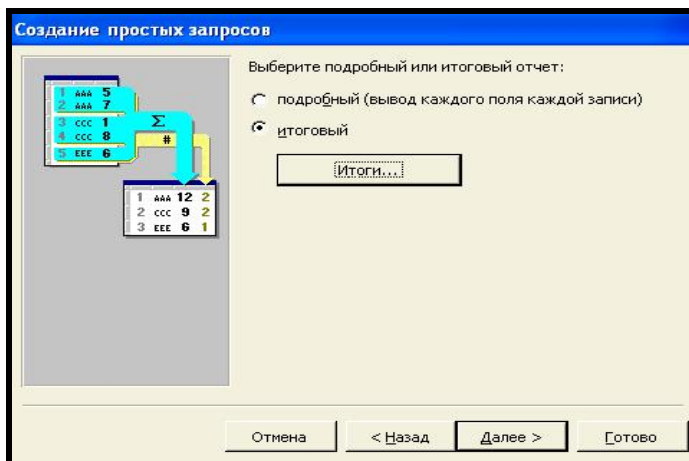


Рисунок 4.8 – Создание итогового запроса

5 Выберите одну или несколько расчетных функций(рисунок 4.9) и нажмите кнопку **OK**. В данном примере выбираем функции **Min**, **Max**:

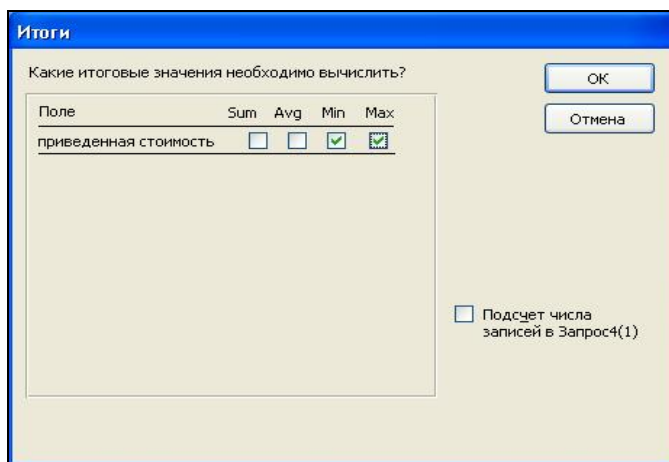


Рисунок 4.9 – Выбор итоговой функции

Примечание 2 - В итоговых запросах рассчитываются итоги только по числовым полям БД.

4.5 Формирование условий отбора в запросах

Условия отбора для различных типов данных формируются при помощи: логических операций и операций отношения, представленных в таблице 4.1.

Таблица 4.1

Логические операции		
And	Логическое умножение	Необходимо одновременное выполнения заданных условий
Or	Логическое сложение	Достаточно выполнение одного из заданных условий
Not	Логическое отрицание	Инверсия условия
Like	Совпадение	Совпадение по маске
Операции сравнения		
>	Больше	
<	Меньше	
>=	Больше или равно	
<=	Меньше или равно	
<>	Неравно	

Примеры формирования различных условий отбора представлены в таблице 4.2.

Таблица 4.2

Тип данных	Постановка задачи	Пример формирования условия отбора
Текстовый	Вывод записей, с названием города 'Киев'	"Киев"
Дата/время	Вывод записей, с датой 23.03.01	#23.03.01#
Текстовый	Вывод записей, с названием городов Киев или Минск	"Киев" <i>or</i> "Минск"
Дата/время	Вывод записей, с датой 23.03.01 или 24.03.01	#23.03.01# <i>or</i> #24.03.01#
Текстовый	Вывод записей, с названием городов Киев и Минск	"Киев" <i>and</i> "Минск"
Текстовый	Вывод записей, с названиями городов между Минском и Киевом	<i>Between</i> "Киев" <i>and</i> "Минск"
Целый	Вывод записей в интервале (5;15)	>5 <i>and</i> <15


Примеры использования масок при формировании условий отбора записей в запросах представлены в таблице 4.3.

Таблица 4.3.

Использование масок в условиях отбора		
Текстовый	Вывод записей, в которых встречается сочетание 'ров'	Like "*"ров*"
Текстовый	Вывод записей, начинающиеся на 'К' и 'М'	Like "К*" or Like "М*"

4.6 Построитель выражений. Создание вычисляемых полей в запросах

Построитель выражений используется для удобства создания выражений и вычисляемых полей в БД.

Вызвать окно построителя выражения можно при помощи команды **Построить** в контекстном меню или кнопкой  на панели инструментов.

Пример 5. Создать запрос, переводящий стоимость билета в ЕВРО. Имя запроса **'Запрос 5'**.

Для этого выполните следующие действия:

- 1 Создайте новый или откройте существующий запрос в режиме конструктора. Для данного примера откройте *Запрос 2* в режиме конструктора;
- 2 В бланке запроса установите курсор в строке **Поле** нового столбца;
- 3 Вызовите контекстное меню и выберите команду **Построить**. Появится окно построителя выражений, представленное на рисунке 4.10;

Окно **Построителя выражений** состоит из двух основных частей:

- в верхнюю часть окна (окно ввода формулы) вводится выражение;
- нижняя часть окна разбита на три области. В левой области выводится список объектов базы данных. В центральной области выводится список полей объектов базы данных;



Рисунок 4.10 – Окно построителя выражений

4 Введите выражение (рисунок 4.11) и нажмите кнопку **OK**. Выбор полей выполняется двойным нажатием “мышью” по названию поля. Набор основных арифметических и логических операций осуществляется при помощи соответствующих кнопок или клавиатуры;

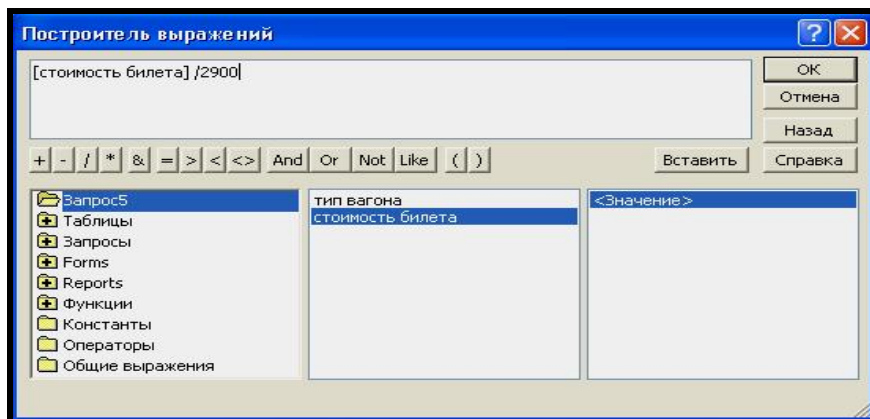
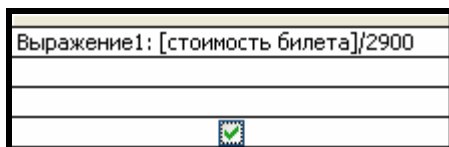
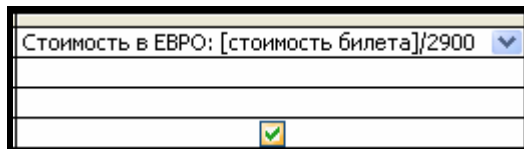


Рисунок 4.11 – Создание выражений

5 В бланке запроса появится новое поле:



6 Вместо фразы ‘Выражение1’ введите новое название поля:



7 Сохраните запрос с именем ‘Запрос 5’, выбрав команду **Файл – Сохранить как**.

5 ОТЧЕТЫ

Отчет – объект БД, предназначенный для форматирования, вычисления итогов и печати выбранных данных. Отчеты можно создавать на основе таблиц и запросов.

5.1 Создание отчета при помощи мастера

Пример 6. Создать отчет по всей базе данных, в котором выводятся данные из полей *Номер поезда*, *Пункт отправления*, *Пункт прибытия*, *Время отправления*, *Время прибытия*. В отчете выполнить сортировку записей поля *Номер поезда*. Имя отчета '*Отчет 1*'.

Для этого, выполните следующие действия:

- 1 В окне базы данных выберите вкладку **Отчеты** и команду **Создать**;
- 2 В появившемся диалоговом окне (рисунок 4.10) выберите команду **Мастер отчетов**:

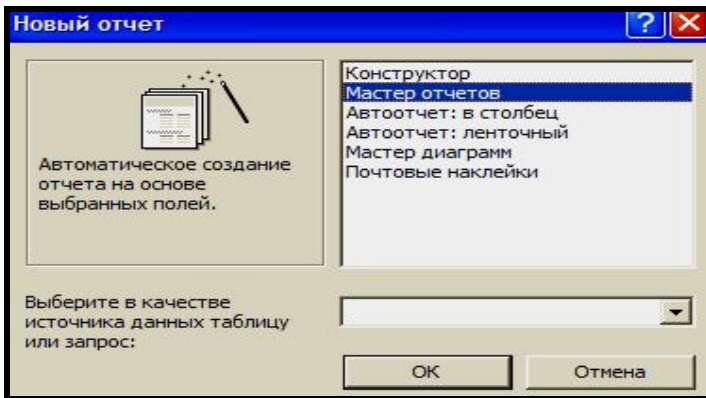
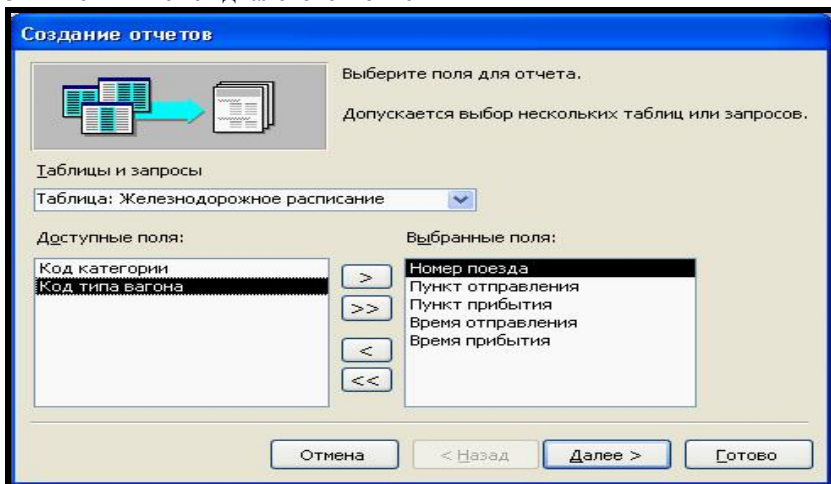


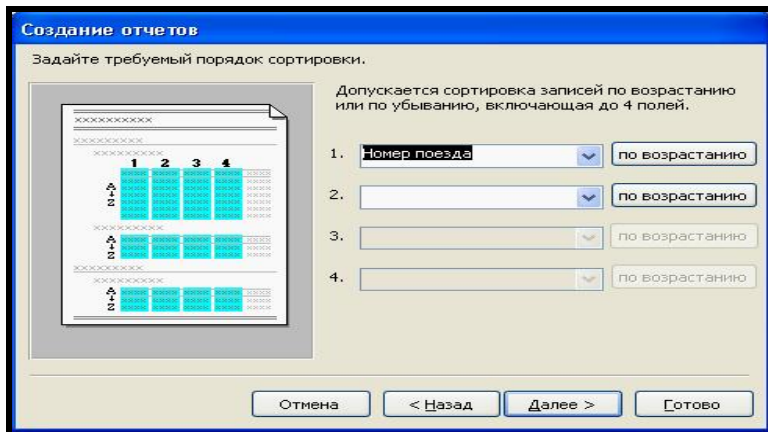
Рисунок 4.10 – Выбор режима создания отчета

- 3 В появившемся диалоговом окне

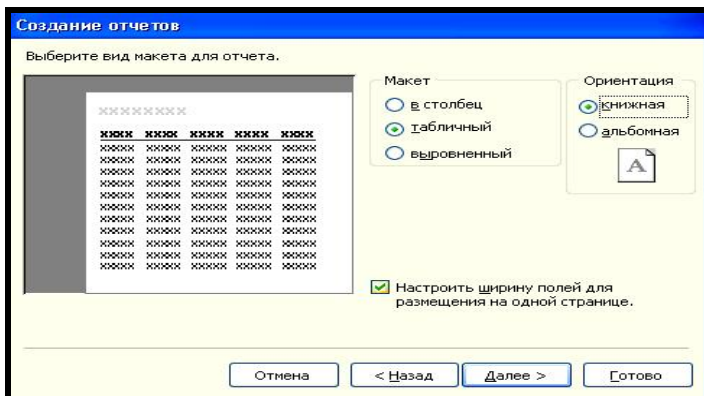


выполните следующие действия:

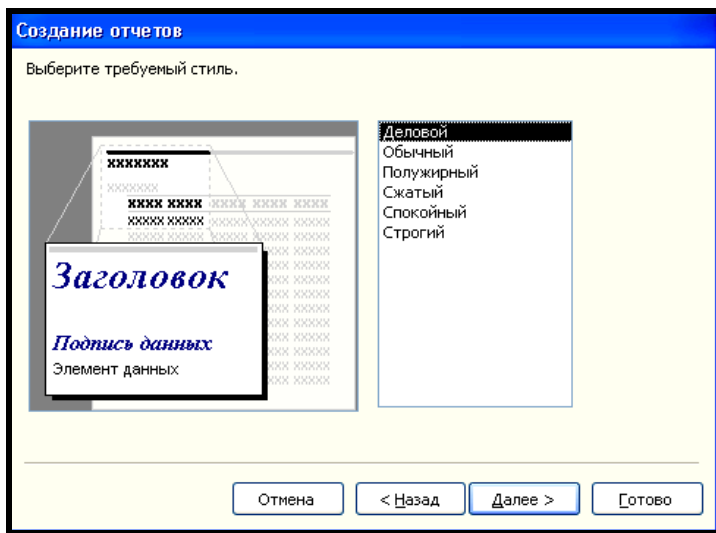
- Выберите таблицу или запрос, поля из которого будут использованы в отчете из списка **Таблицы и Запросы**;
 - Перенесите поля для отчета из списка **Доступные поля** в список **Выбранные поля**, при помощи кнопок > >>;
 - Повторите вышеуказанные действия, если выбираете поля из разных таблиц или запросов;
 - Нажмите кнопку **Далее**;
- 4 В появившемся диалоговом окне выберите при необходимости поля для группировки при помощи кнопки > и нажмите кнопку **Далее**;
- 5 В появившемся диалоговом окне при необходимости выполните сортировку записей и нажмите кнопку **Далее**:



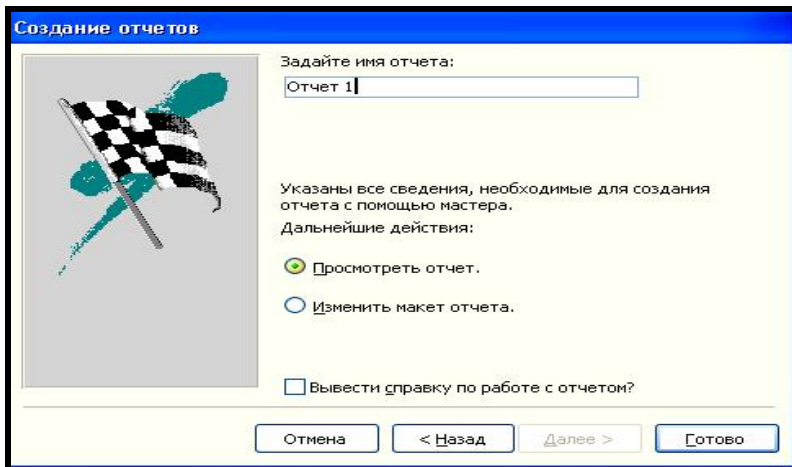
- 6 В появившемся диалоговом окне выберите **Макет отчета**, **Ориентацию** страницы и нажмите кнопку **Далее**:



7 В появившемся диалоговом окне выберите стиль оформления отчета и нажмите кнопку *Далее*:



8 В последнем диалоговом окне дайте имя отчету и нажмите кнопку *Готово*:



5.2 Создание автоотчетов

Пример 7. Создайте ленточный автоотчет для таблицы *Маршрут*. Имя отчета '*Отчет 2*'.

Для этого, выполните следующие действия:

- 1 В окне базы данных выберите вкладку *Отчеты* и команду *Создать*;
- 2 В появившемся диалоговом окне (см. рисунок 4.10) выберите:
 - команду *Автоотчет ленточный (Автоотчет в столбец)*;
 - источник данных для отчета. В данном примере таблицу *Маршрут*;
 - нажмите кнопку *ОК*;
- 3 Закройте отчет, предварительно выполнив его сохранение командой *Файл – Сохранить как*.

5.3 Вычисление итоговых значений в отчетах

Пример 8. Создайте отчет на основе запроса 4(1). Сгруппируйте данные по полю *Номер поезда*. Вычислите среднее значение приведенной стоимости для каждого поезда. Имя отчета '*Отчет 3*'.

Для этого выполните следующие действия:

- 1 Выполните пункты 1-3 из п. 5.1, перенося в отчет необходимые поля из запроса 4;
- 2 Выполните п.4 из п 5.1, создавая необходимые уровни группировки(рисунок 4.11);
- 3 В появившемся диалоговом окне нажмите кнопку *Итоги*, для вычисления итоговых значений и выберите необходимые для расчета функции: *Sum* – сумма, *Avg* – среднее, *Min* – минимум, *Max* – максимум (рисунок 4.12). Нажмите кнопку *ОК*, затем – кнопку *Далее*;
- 4 Закончите создание отчета, выполнив п.6 - 8 из п. 5.1.

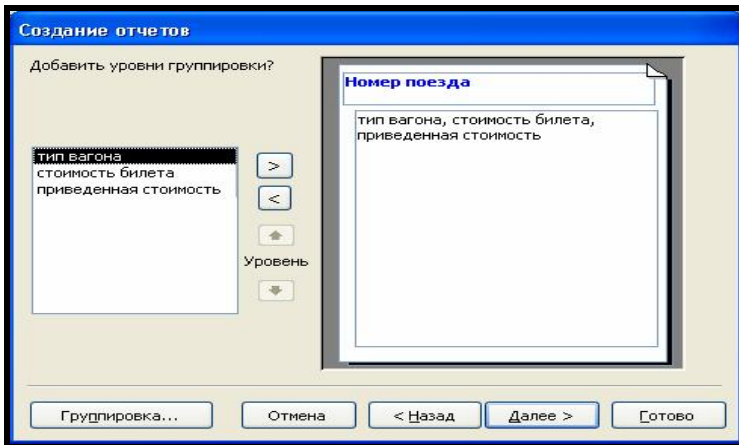


Рисунок 4.11 – Выбор уровня группировки

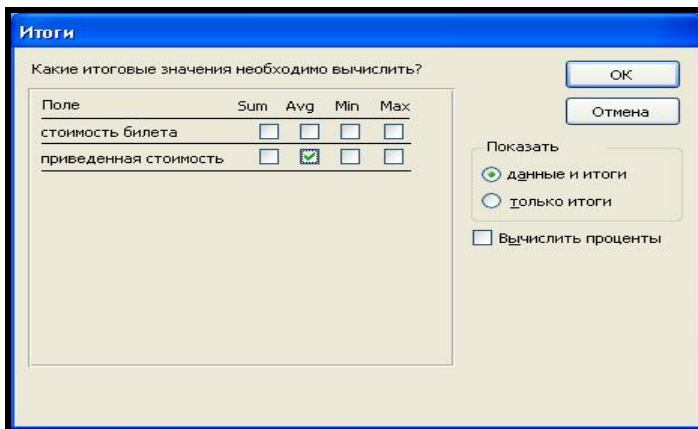


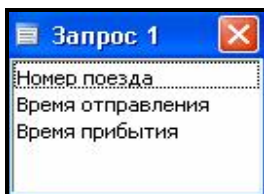
Рисунок 4.12 – Вычисление итоговых значений

5.4 Создание отчета в режиме конструктора

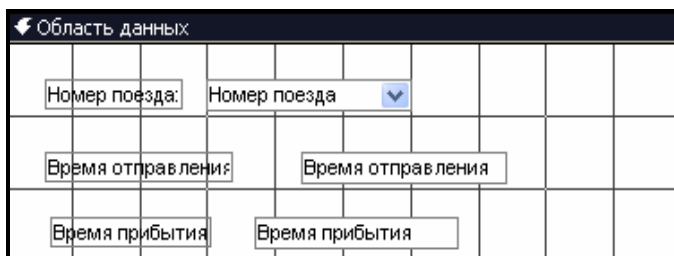
Пример 9. Создайте отчет в режиме конструктора на основе запроса 1. Имя отчета 'Отчет 4'.

Для этого выполните следующие действия:

- 1 В окне базы данных выберите вкладку *Отчеты* и команду *Создать*;
- 2 В появившемся диалоговом окне (рисунок 4.10) выберите команду *Конструктор* и источник данных для отчета. Для данного примера *Запрос 1*.
- 3 Нажмите кнопку *ОК*;
- 3 Перетащите поочередно поля из списка полей



в область данных отчета;



4 Закройте отчет, выполняя его сохранение.

5.5 Окно отчета в режиме конструктора. Настройка параметров окна.

Чтобы открыть существующий отчет в режиме конструктора выделите отчет в окне БД и нажмите кнопку **Конструктор**. Рассмотрим основные элементы структуры отчета, представленные на рисунке 4.13:

1 **Заголовок отчета** расположен на первой странице отчета и содержит название отчета. В многостраничном отчете раздел заголовка отображается только на первой странице. В заголовок отчета при помощи команды **Дата и время** из меню **Вставка** можно вставлять системные дату и время создания отчета;

2 **Верхний и нижний колонтитулы** используются для вывода дополнительной информации. В верхний или нижний колонтитул при помощи команды **Номера страниц** из меню **Вставка** можно добавлять нумерацию страниц в отчет;

3 **В области данных** находятся поля (непосредственно данные) из таблиц и запросов, на основе которых строится отчет;

4 **Примечание** располагается в конце отчета и может содержать поля с рассчитанными итоговыми значениями.

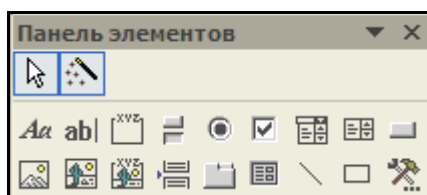
Ввод и вывод элементов отчета на экран осуществляется соответствующими командами, сгруппированными в пункте главного меню **Вид**.

☛ Заголовок отчета									
☛ Верхний колонтитул									
☛ Область данных									
☛ Нижний колонтитул									
☛ Примечание отчета									

Рисунок 4.13 – Окно отчета в режиме конструктора


5.6 Редактирование и форматирование отчета

К командам редактирования отчета относятся: копирование, перемещение и удаление объектов, а также изменение размеров элементов управления и структурных частей самого отчета. Изменение размеров элементов управления осуществляется при помощи размерных маркеров, расположенных по их периметру (см. рисунок 2.2). Изменение размеров структурных элементов осуществляется путем перемещения их верхней границы левой кнопкой мыши. Копирование, перемещение и удаление объектов осуществляется использованием стандартных команд главного меню *Правка*. Редактирование отчета заключается также в создании новых и изменении старых элементов управления. Элементы управления находятся на панели элементов, вызываемой командой *Панель элементов* главного меню *Вид*:



К основным элементам управления относятся:

1 **Надпись** – предназначена для ввода поясняющего текста в любую часть отчета. Для создания надписи выполните следующие действия:

– выберите левой кнопкой “мыши” команду надпись  на панели элементов;

– зафиксируйте курсор в месте вставки и введите текст;

2 **Поле** – предназначено для вывода значения записей из полей БД. Поле состоит из связанной с ним надписи и области вывода данных (рисунок 3.14).

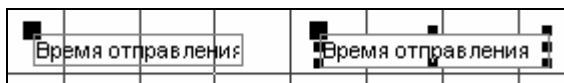




Рисунок 3.14 – Элемент управления ‘поле’

Для создания поля, выводящего данные из таблицы или запроса достаточно перетащить их из списка полей в область данных отчета.

3 **Переключатели и флажки** – предназначены для выбора одного из двух вариантов: Истина или Ложь. Используются данные элементы управления для логических полей. Для создания переключателя (флажка) выполните следующие действия:



- выберите левой кнопкой “мыши” команду переключатель  или флажок  на панели элементов;
- наведите указатель мыши на необходимое поле в списке полей и перетащите его левой кнопкой “мыши” в область данных отчета.
- форматирование отдельных элементов отчета осуществляется командами панели инструментов *Форматирование* или главного меню *Формат*.

5.7 Создание вычисляемых и итоговых полей

Вычисляемые поля предназначены для вывода результатов расчета по одному или нескольким полям БД.

Итоговые поля предназначены для вывода итоговых значений по числовым полям БД. Основные итоговые функции перечислены в разделе 1.2.

Для создания вычисляемых (итоговых) полей выполните следующие действия:

- 1 Выберите левой кнопкой “мыши” команду поле  на панели элементов;
- Зафиксируйте поле в области данных (при создании вычисляемого поля) или в примечании отчета (при создании итогового поля);
- 2 Выберите команду *Свойства* в контекстном меню. Перейдите на вкладку *Все*. В строке *Данные* нажмите кнопку вызова построителя выражений ;
- 3 Введите расчетное выражение;
- 4 Закройте окно свойств поля;
- 5 Введите текст надписи, связанной с полем.

Примечание – При создании вычисляемых и итоговых полей расчетные формулы можно вводить с клавиатуры. При вводе необходимо соблюдать регистр букв в названиях полей.

Пример 10. Отредактируйте отчет 2 следующим образом:

- создайте в области данных вычисляемое поле, вычисляющее расстояние в метрах:
- =[Расстояние]*1000;
- создайте в примечании отчета итоговое поле, вычисляющее минимальное и максимальное расстояние по всей базе данных:
- =Min([Расстояние]);
- =Max([Расстояние]);
- создайте в примечании отчета итоговое поле, вычисляющее среднее значение расстояния по всей базе данных:
- =Avg([Расстояние]);
- создайте надпись *Отчет 5* в заголовке отчета;
- в нижний колонтитул вставьте нумерацию страниц ***Вставка - Номер страницы;***
- сохраните отчет с именем 'Отчет 5'.

6 ПОРЯДОК ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ

Выполнение контрольной работы по курсу ТОХОД включает три этапа:

1) Проектирование базы данных по заданной в варианте тематике на листе бумаги. Номер варианта контрольной работы определяется по ***сумме трёх последних цифр шифра студента***. Проектирование включает:

- разработку структуры баз данных (**Обязательно** создание в базе данных нескольких связанных таблиц);
- определение полей, типов данных, ключевых полей и связей между таблицами. База данных должна содержать все поля, указанные в задании, можно добавлять другие поля кроме тех, которые заданы в задании.

2) Создание в Microsoft Access спроектированную Вами базу данных. Этот пункт задания выполняется на компьютере в среде Microsoft Access. Необходимо создать следующие элементы базы данных:

- таблицы;
- схему данных;
- формы для каждой таблицы. Форма должна содержать хотя бы один элемент управления (список, флажок, кнопку, переключатель и т.д.). С помощью форм заполнить базу данных информацией (минимум 30 записей).
- запросы;
- отчеты. В отчете создать итоговое поле по всему отчету.

3) Оформление работы согласно следующим правилам:

Контрольная работа должна включать:

- 1 Титульный лист;

- 2 Содержание;
- 3 Введение;
- 4 Раздел "1 Краткая характеристика Microsoft Access»;
- 5 Раздел "2 Постановка задачи";
- 6 Раздел "3 Результаты выполнения работы":
 - 6.1. Спроектированная на отдельном листе база данных;
 - 6.2. Распечатка:
 - 6.2.1. Таблиц в режиме отображения данных;
 - 6.2.2. Схемы данных;
 - 6.2.3. Форм;
 - 6.2.4. Запросов в режиме конструктора и в режиме отображения данных;
 - 6.2.5. Отчета.
- 7 Заключение;
- 8 Список использованной литературы.

Все разделы работы должны начинаться с заголовка. Объем контрольной работы – 10...12 страниц. Страницы должны быть пронумерованы. Номер проставляется в центре нижнего поля страницы. Титульный лист включается в общую нумерацию страниц, но номер на нем не проставляется.

Файл созданной базы данных Microsoft Access с формами, запросами и отчетами сохранить на дискете и представить преподавателю при защите контрольной работы.

7 РАБОЧАЯ ПРОГРАММА
Министерство образования Республики Беларусь
Белорусский государственный университет транспорта

«УТВЕРЖДАЮ»
Декан факультета безотрывного обучения
_____ В.В. Пигунов
«___» _____ 2004 г.

РАБОЧАЯ ПРОГРАММА
заочная форма обучения

по курсу **“Технология организации, хранения и обработки данных”**

для специальности – 1–25 01 08 «Бухгалтерский учет, анализ и аудит»
специализация 1–25 01 08-06
«Бухгалтерский учет, анализ и аудит на предприятии транспорта»

Факультет - УПП
кафедра “Информационные технологии”
курс 1
семестр 2,3

Лекции 4 час.

Экзамен – нет

Зачет – 3 семестр

Лабораторные занятия - 6час.

Контрольная работа – 3 семестр

Самостоятельная работа 44 час.

(в том числе 10 часов СУРС)

Всего часов по дисциплине 54 час.

2004

Рабочая программа составлена на основе государственного образовательного стандарта РДРБ02100.5.122-98 (специальность "Бухгалтерский учет, анализ и аудит"), утверждённого Учебно-методическим управлением по высшему образованию 9.01.1999 года.

Рассмотрена на заседании кафедры "Информационные технологии" и рекомендована к утверждению

30 июня 2004 г.

И.о. заведующего кафедрой

_____Ю.А. Пшеничнов

Рекомендована методической комиссией факультета безотрывного обучения к утверждению

200 г.

Председатель

1 Цели и задачи дисциплины, её место в учебном процессе

1.1 Цель преподавания дисциплины

Курс “Технология организации, хранения и обработки данных” предлагается в качестве общеобразовательной дисциплины для студентов I курса специальности “Бухгалтерский учет, анализ и аудит” факультета безотрывного обучения в соответствии с принципом непрерывной компьютерной подготовки студентов экономических специальностей вузов. Курс содержит основные понятия и модели баз данных, технологию создания баз данных, а также методы и средства обеспечения защиты информации.

Преподавание дисциплины обусловлено проблемами повышения уровня знаний, предъявляемых к образованию современных экономистов, их использованием в практической работе.

Цели преподавания дисциплины следующие:

- познакомить студентов с формами представления экономической информации.
- приобретение необходимых навыков работы с системами управления базами данных (на примере MS ACCESS);
- приобретение необходимых навыков работы с программами резервирования данных;
- приобретение необходимых навыков работы со средствами обеспечения защиты информации.

1.2 Задачи изучения дисциплины

Данный курс предназначен для реализации 3-го этапа в технологической цепочке изучения основ информатики и вычислительной техники – приобретение необходимых знаний и навыков по проектированию, созданию и ведению баз данных (в том числе распределенных). Это определило круг решаемых в нем *задач*:

- раскрыть функциональное назначение систем хранения и обработки данных;
- научить студентов пользоваться современными системами управления базами данных.

- научить студентов пользоваться современными методами и программными средствами защиты данных.

В результате изучения дисциплины студент должен:

Знать:

- Основные определения, свойства, характеристики, структуры и формы представления экономической информации;
- Основные понятия и модели баз данных;
- Возможности СУБД по созданию и ведению баз данных;
- Основы программирования на Visual Basic for Applications (VBA)
- Методы и средства защиты информации

Уметь:

- Работать с СУБД ACCESS: создавать базы данных, в том числе многотабличные; вводить данные в базу с помощью форм; составлять запросы и отчеты по базе данных;
- Пользоваться программами резервирования данных;
- Пользоваться программами защиты данных.

2 Содержание дисциплины

Курс «Технология организации, хранения и обработки данных» читается во 2, 3 семестрах факультета безотрывного обучения (зимняя и летняя сессии) для студентов специальности 1–25 01 08 «Бухгалтерский учет, анализ и аудит» специализация 1–25 01 08-06 «Бухгалтерский учет, анализ и аудит»

Ниже в таблице приведено почасовое распределение курса по семестрам и видам выполняемой работы:

Курс	Семестр	Лекций, час.	Лаб. занятий, час	Количество контрольных работ	Форма отчётности	СУРС	Самост. работа, час
1	2	2	2	-	-	-	28
1	3	2	4	1	зачет		16
Итого:		8	6	1		10	44

2.1 Курс лекционных занятий – 4 часа

<p>1. Определение, свойства, характеристики, структура и формы представления экономической информации. Структурные единицы экономической информации, классификация и кодирование экономической информации</p>	0,5 часа
<p>2. Основные понятия и модели баз данных. Организация данных: логическая организация данных, файловая модель. Сетевые и иерархические модели данных: структуры данных в моделях, связи объектов в моделях, особенности моделей, сравнение моделей</p>	0,5 часа
<p>3. Основы работы с СУБД Microsoft ACCESS 2000. Структура базы данных, свойства полей базы данных, типы данных. Проектирование баз данных: режимы работы с базами данных, объекты базы данных (таблицы, запросы, формы, отчеты, страницы, макросы и модули). Работа с таблицами: структура таблицы, данные таблицы. Создание таблицы: создание новой пустой таблицы, создание таблицы в режиме конструктора. Выбор для поля таблицы типа данного, размер поля, поле типа "счётчик", формат поля и маска ввода данных. Ключи и индексы: типы ключевых полей (ключевые поля счётчика, простой ключ, составной ключ), индексы, создание и изменение ключевых полей, типы ограничений в базе данных, поддержание целостности данных, создание межтабличных связей</p>	1 час
<p>4. Работа с формами: автоформы, создание форм с помощью мастера, структура формы, элементы управления формы, дизайн формы, управление последовательностью перехода.</p>	0,5 часа
<p>5. Работа с запросами: упорядочение записей в результирующей таблице, управление отображением данных в результирующей таблице, использование условия отбора, другие виды запросов. Понятие технологии клиент-сервер. Понятие SQL-языка и SQL-запроса</p>	0,5 часа
<p>6. Создание отчетов. Формирование и вывод отчетов: окно конструктора отчетов, разделы отчета, создание отчета, группировка данных отчета, размещение полей из таблиц, включение вычисляемого поля в отчет, добавление текущей даты и номера страницы, просмотр и печать отчета</p>	0,5 часа

7. Основы программирования на Visual Basic for Applications для Microsoft Access. Технология ODBC (Open Data-base Connectivity).	0,5 часа
--	----------

2.2. Курс лабораторных занятий – 6 часов

<p>1. Начало работы в СУБД ACCESS. Таблицы. Формы. Запуск и работа в окне ACCESS. Открытие базы данных. Создание базы данных: создание файла базы данных, создание таблицы базы данных. Режим Конструктора. Описание полей базы данных. Непосредственный ввод данных в таблицу. Взаимосвязи таблиц. Связи-объединения. Обеспечение целостности данных. Создание схемы данных. Понятие формы. Структура формы. Мастер и Конструктор форм. Разработка многотабличной формы. Создание многотабличной формы с помощью Мастера. Загрузка взаимосвязанных таблиц</p>	2 часа
<p>2. Основы разработки запросов. Назначение и виды запросов. Окно запроса. Условия отбора записей. Вычисляемые поля. Параметры запроса. Мастера создания запросов. Однотабличный запрос на выборку. Многотабличный запрос на выборку. Ввод параметров в запрос. Основы SQL-языка. Создание SQL-запросов</p>	2 часа
<p>3. Формирование и вывод отчетов. Создание отчетов. Окно конструктора отчетов. Группировка данных отчета. Размещение полей из таблиц. Включение вычисляемого поля в отчет. Добавление текущей даты и номера страницы. Завершение формирования отчета. Просмотр и печать отчета</p>	2 часа

2.3 Наименование тем контрольных работ

Контрольная работа № 1 – Технология создания и обработки данных в среде Microsoft Access.

При выполнении контрольной работы " Технология создания и обработки данных в среде Microsoft Access " студент с помощью СУБД ACCESS разрабатывает структуру базы данных, компоует данные в таблицы, задает в каждой таблице ключевые поля, устанавливает необходимые связи между таблицами. Состав таблиц и типы полей выбирает самостоятельно. Для каждой таблицы студент создает формы для ввода данных, проектирует и соз-

дает форму, с помощью которой будет обеспечиваться заполнение всей базы данных (в этой форме обязательно должны присутствовать элементы управления – переключатели, списки, кнопки перехода по записям и т.п.). Заполняет базу данными. Для указанных полей студент создает запросы к базе данных и отчет.

3 Учебно-методические материалы по дисциплине

3.1 Основная литература

- 1 Информатика для юристов и экономистов / Симонович С.В. и др. – СПб: Питер, 2001.
- 2 Основы экономической информатики: Учеб. пособие/ А.Н. Морозевич, Н.Н. Говядинова, В.Г. Левашенко, Б.А. Железко и др.; Под ред. проф. А.Н. Морозевича. - Мн.: Новое знание, 2001.
- 3 Савицкий Н.И. Технологии организации, хранения и обработки данных: Учеб пособие. – М.: ИНФРА-М, 2001.

3.2 Дополнительная литература

- 1 Лыч Ю. П., Шинкевич Т. Л. Технологии организации, хранения и обработки данных. Ч.1: Основы создания баз данных: Пособие для самостоятельной работы. – Гомель: БелГУТ, 2003. – 157 с.
- 2 Лыч Ю. П. Технологии организации, хранения и обработки данных. Ч.2. Основы программирования в СУБД Access: Пособие для самостоятельной работы. – Гомель: БелГУТ, 2003. – 68 с.

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1 Проектирование базы данных. Создание базы данных.
- 2 Создание таблицы. Режим Конструктора.
- 3 Описание полей базы данных.
- 4 Тип данных.
- 5 Свойства полей.
- 6 Задание маски ввода.
- 7 Выбор ключевого поля.
- 8 Непосредственный ввод данных в таблицу.
- 9 Редактирование данных в таблице.
- 10 Удаление записи из таблицы.
- 11 Взаимосвязи таблиц. Создание схемы данных.
- 12 Обеспечение целостности данных .
- 13 Понятие формы.
- 14 Создание автоформы. Типы автоформ.
- 15 Создание формы с помощью мастера форм.
- 16 Создание формы в режиме Конструктора.
- 17 Создание многотабличной формы (связь один-ко-многим) с помощью мастера форм.
- 18 Панель элементов.
- 19 Создание заголовка формы.
- 20 Создание элемента управления «поле со списком».
- 21 Изменение структуры формы.
- 22 Перемещение элементов управления.
- 23 Что такое запрос?
- 24 Назовите основные режимы создания запросов на выборку?
- 25 Что такое параметрический запрос?
- 26 Как формируются условия отбора в запросах?
- 27 Для чего используется Построитель выражений. Как его вызвать?
- 28 Как создать вычисляемое поле в запросе?
- 29 Для чего используются запросы на обновление?
- 30 Для чего используются итоговые запросы?
- 31 Другие виды запросов и их использование?
- 32 Структура бланка запроса.
- 33 Какая команда используется для открытия запроса в режиме редактирования?
- 34 Какая команда используется для просмотра результатов выполнения запроса в окне БД?
- 35 Как сохранить запрос с новым именем?

- 36 Что такое отчет?
- 37 Основные режимы создания отчетов.
- 38 Структура отчета. Настройка структурных элементов отчета.
- 39 Элемент управления – надпись. Назначение. Создание.
- 40 Элемент управления – поле. Назначение.
- 41 Вычисляемые и итоговые поля. Назначение. Создание.
- 42 Функции вычисления итогов.

СПИСОК ЛИТЕРАТУРЫ

- 1 Вейскас Дж. Эффективная работа с Microsoft Access 2000. – СПб.: Питер, 2000. – 1040с.
- 2 Microsoft Access 2000: Справочник / Под ред. Ю. Колесникова. – СПб.: Питер, 1999. – 416с.

Учебное издание

ШИНКЕВИЧ Татьяна Леонидовна
ЛИТВИНОВИЧ Татьяна Николаевна

Система управления базами данных MS ACCESS

**Пособие по выполнению контрольной работы
для студентов ФБО**

по курсу «ТОХОД»

Редактор Н.А. Дашкевич

Технический редактор В.Н. Кучерова

Подписано в печать 6.05.2005 г. Формат бумаги 60x84 1/16.

Бумага газетная. Гарнитура Таймс. Печать офсетная.

Усл. печ. л. 2,09. Уч.-изд. л. 2,24. Тираж 300 экз.

Зак. № . Изд. № 4038 .

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.