#### Кафедра: Автоматика и Информационные Технологии

## основы разработки приложений баз Данных

#### ОГЛАВЛЕНИЕ

1. Системы управления базами данных

1.1 Основы баз данных

1.2 Таблицы базы данных

1.3 Средства для работы с базами данных

1.4 Инструментальные средства

1.5 Компоненты

1.6 Технология создания приложения

1.6.1 Создание таблиц базы данных

1.6.2 Создание формы приложения

1.7 Программа BDE Administrator

1.8 Работа с псевдонимами

1.9 Работа со связанными таблицами

2. Задания к лабораторным работам

# 1. Системы управления базами данных

#

# 1.1 Основы баз данных

Информационные системы позволяют автоматизировать сбор и обработку данных. Они являются банками данных, включающими:

* вычислительную систему;
* базу данных (БД);
* систему управления базами данных (СУБД);
* набор прикладных программ.

БД обеспечивает хранение информации и представляет собой совокупность данных, организованных по определенным правилам. БД позволяет структурировать, хранить и обрабатывать данные различного типа.

СУБД – это совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД. Персональная СУБД обеспечивает возможность создания локальных СУБД. К ним относятся Paradox, DBase, FoxPro, Access. Многопользовательские СУБД позволяют создавать информационные системы, функционирующие в архитектуре клиент-сервер. К ним относятся Oracle, InterBase, Microsoft SQL Server.

Языковые средства современных СУБД включают:

* язык описания данных, предназначенный для описания логической структуры данных;
* язык манипулирования данными, обеспечивающий выполнение основных операций над данными – ввод, модификацию и выборку;
* структурированный язык запросов (Structure Query Language, SQL), обеспечивающий управление стандартным средством доступа к удаленным БД;
* язык запросов по образцу (Query By Example, QBE), обеспечивающий визуальное конструирование запросов к БД.

Прикладные программы, или приложения, служат для обработки данных, содержащихся в БД. Пользователь осуществляет управление БД и работу с ее данными именно с помощью приложения.

# 1.2 Таблицы базы данных

БД состоят из взаимосвязанных таблиц. Таблицы хранятся в файлах. Они похожи на отдельные документы или электронные таблицы. В отличие от документов, таблицы БД поддерживают многопользовательский режим, т.е. с ними одновременно могут работать несколько приложений.

Таблицы Paradox являются достаточно развитыми и удобными при создании БД. Основные достоинства таблиц Paradox:

* имеется много различных типов полей для представления данных;
* поддерживается целостность данных;
* предоставляется возможность организации проверки вводимых данных;
* поддерживается защита таблиц с помощью паролей.

В табл. 1 представлен список типов полей таблиц Paradox7.

Таблица 1

Типы полей таблиц Paradox7

|  |  |  |
| --- | --- | --- |
| Тип | Обозначения | Описание обозначения |
| Alpha | A | Строка символов. Длина не более 255 символов |
| Number | N | Число с плавающей точкой. Диапазон 10-307-– 10308, точность 15 цифр мантиссы |
| Money | $ | Денежная сумма |
| Short | S | Целое число. Диапазон -32768 – 32767 |
| LongInteger | I | Целое число. Диапазон -2 0147 0483 648 – 2 147 483 647 |
| BCD | # | Число в двоично-десятичном формате |
| Date | D | Дата. Диапазон 01.01.9999 до н.э. – 31.12.2999 |
| Time | T | Время |
| TimeStamp | @ | Дата и время |
| Memo | M | Строка символов. Длина не ограничена. Первые 240 символов хранятся в файле таблицы, остальные в файле с расширением MB |
| FormattedMemo | F | Строка символов. Строка содержит форматированный текст |
| Graphic | G | Графические изображения. Форматы bmp, pcx, tif, gif и epx |
| Тип | Обозначения | Описание обозначения |
| OLE | O | Данные в формате, который поддерживается технологией OLE |
| Logical | L | Логическое значение. Допустимые значения T (Истина) и F (Ложь) |
| Autoincrement | A | Автоинкрементное поле. При добавлении к таблице новой записи в поле автоматически заносится уникальное значение. При удалении записи значение ее автоинкрементного поля больше не будет использоваться. Значение такого поля допустимо для чтения и обычно используется в качестве ключевого |
| Binary | B | Последовательность байтов. Длина не ограничена. Байты содержат произвольное двоичное значение |
| Bytes | Y  | Последовательность байтов. Длина не более  255 байт |

Имя поля в таблице Paradox должно состоять из букв и цифр и начинаться с букв. Максимальная длина имени составляет 25 символов. Ключевые поля должны быть первыми в структуре таблицы.

Для полей можно определить специальный диапазон, в котором должны находиться значения, вводимые в эти поля, а также минимальное и максимальное допустимое значение. Для каждого поля можно задать значение по умолчанию, которое автоматически заносится в поле при добавлении новой записи.

Недостатком таблиц Paradox является наличие относительно большого количества файлов, требуемых для хранения и содержащихся в таблице. Файлы таблиц имеют следующие расширения:

* DB – таблица с данными;
* MB – BLOB – данные;
* PX – главный индекс (ключ);
* XG? и YG? – вторичные индексы;
* VAL – параметры для проверки данных и целостности ссылок;
* TV и FAM – форматы вывода таблицы в программе Database Desktop.

# 1.3 Средства для работы с базами данных

К средствам Delphi, предназначенным для работы с БД, относятся следующие:

* инструментальные средства (специальные программы и пакеты, обеспечивающие обслуживание БД вне разрабатываемых приложений);
* компоненты, предназначенные для создания приложений, которые осуществляют операции с БД.

# 1.4. Инструментальные средства

Для операций с БД система Delphi предлагает такие инструментальные средства:

* Borland Database Engine (BDE) – процессор баз данных, который представляет собой набор библиотек, предназначенных для организации доступа к БД из приложений Delphi;
* BDE Administrator – утилита для настройки BDE. Позволяет настраивать различные параметры БД;
* Database Desktop – программа для создания и редактирования таблиц,
SQL- и QBE-запросов;
* SQL Explorer – проводник БД, позволяющий настраивать параметры БД.

**1.5 Компоненты**

Компоненты, связанные с БД, делятся на визуальные и невизуальные:

* невизуальные компоненты служат для организации доступа к данным, содержащимся в таблицах. Они представляют собой промежуточное звено между данными таблиц БД и визуальными компонентами;
* визуальные компоненты используются для создания интерфейсной части приложения. С их помощью пользователь может выполнять такие операции с таблицами, как просмотр или редактирование данных.

На странице Data Access (рис. 1) находятся невизуальные компоненты, с помощью которых можно организовать доступ к данным:

* Data Source – источник данных;
* Table – набор данных, основанных на таблице БД;
* Query – набор данных, основанных на SQL-запросе;
* StoredProс – набор данных, основанных на процедуре, которая хранится на сервере;
* DataBase – соединении с БД;
* Session – текущий сеанс работы с БД;
* BatchMove – выполнение операций над группой записей;
* UpdateSQL – модификация набора данных, основанного на SQL-запросе;
* NestedTable – вложенная таблица.

Рис. 1. Страница Data Access

На странице Data Controls (рис. 2) находятся следующие визуальные компоненты, предназначенные для управления данными:

* DBGrid – сетка (таблица);
* DBNavigator – навигационный интерфейс;
* DBText – надпись;
* DBEdit – однострочный редактор;
* DBMemo – многострочный редактор;
* DBImage – графический образ;
* DBListBox – простой список;
* DBComboBox – комбинированный список;
* DBCheckBox – независимый переключатель;
* DBRadioGroup – группа независимых переключателей;
* DBLookupListBox – простой список, формируемый по полю другого набора данных;
* DBLookupComboBox – комбинированный список, формируемый по полю другого набора данных;
* DBRichEdit – полнофункциональный текстовый редактор;
* DBCtrlGrid – модифицированная сетка;
* DBChart – диаграмма.

Рис. 2. Страница Data Controls

#

# 1.6 Технология создания приложения

В качестве примера использования возможностей Delphi для работы с БД рассмотрим технологию создания простого приложения. Основные этапы простого приложения:

* создание таблиц БД;
* создание формы приложения.

# 1.6.1 Создание таблиц базы данных

Для работы с таблицами БД при проектировании приложения удобно использовать программу Database Desktop (рис. 3), которая позволяет:

* создавать и изменять структуру таблицы;
* редактировать записи;
* создавать, редактировать и выполнять SQL-запросы;
* создавать, изменять и выполнять запросы по образцу.

Рис. 3. Окно утилиты Database Desktop и диалоговое окно Create Table

Эту программу можно вызвать из среды Delphi по команде Tool/Database Desktop или путем запуска файла dbd32.exe, находящегося в одном каталоге с файлами программы Database Desktop. По умолчанию этот каталог называется Database Desktop.

Процесс создания новой таблицы начинается по команде New/Table (Новая/Таблица) и происходит в интерактивном режиме, при этом разработчик должен:

* выбрать тип таблицы;
* задать структуру таблицы;
* указать ключевые поля;
* определить индексы;
* определить ограничения на значения полей;
* определить пароль;
* задать ссылочную целостность (связи) между таблицами.

Обязательны два первых действия, перечисленные в этом списке. Часть действий, например задание ключевых полей, применяется только для таблиц определенных типов, например Paradox.

Сначала в окне Create Table (см. рис. 3) выбирается форма таблицы. По умолчанию предлагается формат Paradox 7, который используется в качестве примера.

После выбора типа таблицы появляется окно определения ее структуры (рис. 4), в котором выполняются все остальные действия, при этом нужно задать как минимум одно поле.

Для каждого поля задается имя (в столбце Field Name), тип (в столбце Type) и при необходимости размерность (в столбце Size). Можно задать тип поля, непосредственно указав соответствующий символ или выбрав из списка (см. рис. 4), раскрываемого после нажатия клавиши <Пробел> или щелчка мыши на столбце с указанием типов. Список содержит все типы полей, допустимые для заданного формата таблицы.

Для указания ключевых полей в столбце ключа Key нужно ввести символ "\*" следующим способом: установить курсор в эту позицию и нажать любую алфавитно-цифровую клавишу. При повторном нажатии клавиши снимается метка о принадлежности поля ключу. Ключевые поля должны быть в списке первыми, то есть располагаться в окне определения структуры таблицы наверху.

Рис. 4. Окно для определения структуры таблицы

Для выполнения дополнительных действий по определению структуры таблицы используется комбинированный список Table properties (свойства таблицы), содержащий следующие пункты:

* Secondary Index – задание индекса;
* Validity Checks – ограничения на ввод значений полей;
* Password Security – определение пароля;
* Referential Integrity – определение ссылочной целостности между таблицами;
* Table Language – задание языка;
* Table Lookup – задание полей просмотра.

После выбора этого списка в окне определения структуры таблицы появляются соответствующие элементы, с помощью которых выполняются дальнейшие действия. Как наиболее часто используемое рассмотрим задание индексов.

При выборе пункта Secondary Index комбинированного списка становится доступной кнопка Define (определить). После ее нажатия проявляется окно Define Secondary Index (задание вторичного индекса) (рис. 5). В этом окне задаются индексные поля, отображаемые в правом списке, и параметры индекса, основные из которых изменяются с помощью следующих флажков:

* Unique – индекс допускает уникальные значения для составляющих его полей;
* Case sensitive – для полей строкового типа учитывается регистр символов;
* Descending – сортировка выполняется в порядке убывания значений.

После задания состава индексных полей и нажатия кнопки ОК появляется окно Save Index As, в котором указывается имя индекса.

Рис. 5. Окно задания индекса

После определения структуры таблицы ее необходимо сохранить, нажав кнопку Save as… (сохранить как) и указав расположение таблицы и ее имя на диске.

В последующем структуру таблицы можно изменить, вызвав команду Table/Restructure... (таблица/изменить структуру), которая доступна только для открытой таблицы.

# 1.6.2 Создание формы приложения

В качестве примера рассмотрим форму приложения, с помощью которого можно перемещаться по записям таблицы БД, просматривать и редактировать поля записей, вставлять в таблицу новые записи, а также удалять из таблицы записи.

Вид формы приложения представлен на рис. 6. На форме расположены следующие компоненты: Table1, DataSource1, DBGrid1 и DBNavigator1.

Рис. 6. Форма приложения для работы с БД

Компонент Table1 обеспечивает взаимодействия с БД. Для связи с требуемой таблицей необходимо установить соответствующие значения свойств DataBaseName, которое указывает путь к БД, и TableName, которое задает имя таблицы. После задания таблицы БД свойству Active должно быть установлено значение True.

При смене таблицы перед значением свойств DataBaseName и TableName нужно установить значение False свойству Active.

Имя таблицы лучше выбирать из раскрывающегося списка в поле значения свойства TableName. Если путь к БД (свойство DataBaseName) задан правильно, в этом списке отображаются все доступные файлы.

Компонент DataSourse1 является промежуточным звеном между компонентом Table1, который соединен с реальной таблицей БД, и управляющими компонентами DBGrid1 и DBNavigator1, с помощью которых пользователь взаимодействует с этой таблицей. На компонент Table1, с которым связан компонент DataSourse1, указывает свойство DataSet последнего.

Компонент DBGrid1 отображает содержимое таблицы БД в виде сетки, в которой столбцы соответствуют полям, а строки – записям. По умолчанию пользователь может просматривать и редактировать данные. Компонент DBNavigator1 предоставляет возможность перемещаться по таблице, редактировать, вставлять и удалять записи. Компоненты DBGrid1 и DBNavigator1 связываются со своим источником данных – компонентом DataSourse1 через свои свойства DataSource.

При разработке приложения значения всех свойств компонентов можно задать с помощью Инспектора объектов. При этом требуемые значения можно набрать в поле значений или выбрать из раскрывающихся списков. В табл. 2 приведены компоненты для работы с БД, а также основные свойства и их значения. В данном примере используется таблица из файла с названием spispe1.db, которая лежит в папке, указанной пользователем. Путь к каталогу указывает псевдоним pimDB.

Таблица 2

Значения свойств компонентов

|  |  |  |
| --- | --- | --- |
| Компонент | Свойства | Значения |
| Table1 | Active | spispe1.db |
| DataBaseName TableName | primDB | true |
| DataSourse1 | DataSet | Table1 |
| DBGrid1 | DataSource | DataSource1 |
| DBNavigator1 | DataSource | DataSource1 |

Для автоматизации процесса создания формы, использующей компоненты для операции с БД, можно вызвать Database Form Wizard (Мастер форм баз данных). Появится окно, представленное на рис. 7. Мастер форм баз данных расположен на странице Business хранилища объектов, доступ к которому осуществляется по команде File/New… (файл/новый).

Рис. 7. Окно Мастера форм баз данных

Мастер форм баз данных позволяет создавать формы для работы с отдельной таблицей и с взаимосвязанными таблицами, при этом можно использовать наборы данных Table и Query.

# 1.7 Программа BDE Administrator

Программа BDE Administrator представляет собой администратор процессора баз данных BDE. Для вызова администратора BDE запускается файл bdadmin.exe. Администратор позволяет настраивать параметры БД и системные параметры (операционной системы). Параметры псевдонима: название, тип, путь. Параметры драйвера: тип, язык. Системные установки: установки по умолчанию, форматы даты, времени и числовые параметры.

Для настройки некоторого параметра в левой части окна администратора BDE выбирается нужный объект, после чего в правой части окна становится доступным список параметров этого объекта. Добавить новый объект можно, выбрав в окне администратора пункт меню Object/New (Объект/Новый).

# 1.8 Работа с псевдонимами

Псевдоним (alias) указывает на местоположение БД и представляет собой специальное имя для обозначения каталога.

Для добавления нового псевдонима перед вызовом пункта меню Object/New администратора нужно выбрать вкладку Database в левой части окна. В диалоговом окне нужно выбрать тип драйвера. Для локальных таблиц Paradox выбирается тип STANDARD.

После нажатия кнопки ОК создается новый псевдоним, и его данные отображаются в окне администратора BDE (рис. 8). Новый псевдоним автоматически получает имя STANDARD1 и параметры по умолчанию.

Псевдоним для работы с локальными БД имеет следующие три параметра:

* DEFAULT DRIVER – указывает формат таблицы БД;
* ENABLE BCD – указывает на необходимость перевода чисел в формат BCD, что позволяет более точно выполнять числовые значения, но уменьшает скорость их выполнения;
* PATH – указывает расположение (каталог) БД.

Рис. 8. Установка параметров псевдонимов

# 1.9 Работа со связанными таблицами

Между отдельными таблицами БД может существовать связь, которая организуется через поля таблиц. Поля связи обязательно должны быть индексированными. Связь между таблицами определяет отношение подчиненности, при котором одна таблица является главной (родительской), а другая – подчиненной (дочерней или детальной). Обычно используется связь "один ко многим", когда одной записи в главной таблице может соответствовать несколько записей в подчиненной таблице.

Для организации связи между таблицами в подчиненной таблице используются свойства (в Object Inspector):

* MasterSource – источник данных главной таблицы;
* IndexName – текущий индекс;
* IndexFieldNames – поле или поля связи текущего индекса подчиненной таблицы;
* MasterFields – поле или поля индекса главной таблицы.

Для таблиц Paradox в качестве полей связи должны использоваться поля главного индекса, а для подчиненной таблицы – поля вторичного индекса.

Рассмотрим пример вывода данных из главной и подчиненной таблиц (рис. 9). Для организации связи в качестве поля связи главной таблицы берется автоинкрементное поле Kod уникального кода. По этому полю построен главный ключ, значение которого автоматически формируется при добавлении новой записи и в пределах таблицы является уникальным. В подчиненной таблице полем связи является целочисленное поле KodPe, по которому построен вторичный индекс.

Рис. 9. Форма связи между таблицами "один ко многим"

В верхней части формы (см. рис. 9) выводится список всех подвижных единиц (ПЕ), в нижней – сведения о ремонтах выбранной ПЕ. Для наглядности в наборы данных включены все поля таблиц, которые отображаются в компонентах DBGrid. При этом названия заголовков столбцов совпадают с названиями полей.

В рассмотренном примере связь между таблицами устанавливается при выполнении приложения. Обычно таблицы связывают на этапе разработки через Инспектор объектов. При этом для установки свойства MasterFields можно использовать редактор полей связи (Field link Designer). В списке Detail Fields выбирается поле подчиненной таблицы, а в списке Master Fields – поле главной таблицы (рис. 10).

Рис. 10. Редактор полей связи

После нажатия кнопки Add выбранные поля связываются, что отображается в списке Joined Fields. Заполнение свойства MasterFields происходит после закрытия окна при нажатии кнопки OK.

В Delphi для работы с наборами данных таблиц применяются компоненты Table, Query, Decision Query и StoredProc.

Компонент Table представляет набор данных, который в некоторый момент времени может быть связан с одной таблицей БД. Расположение БД, с таблицами которой выполняются операции, указывает свойство DataBaseName типа String (рис. 11). Значением свойства является имя каталога, в котором расположены БД, или псевдоним, ссылающийся на этот каталог.

Рис. 11. Свойства компонента Table в инспекторе объектов

Связь между таблицей и компонентом Table устанавливается через его свойство TableName типа TFileName, которое определяет имя таблицы (и имя файла, содержащего данные). По умолчанию в набор данных Table попадают все записи связанной с ним таблицы. Для отбора данных можно использовать фильтрацию.

Для того чтобы запретить пользователю изменять записи, можно использовать свойство ReadOnly типа Boolean. По умолчанию это свойство имеет значение False. Это означает, что пользователь может редактировать записи.

Установить текущий индекс можно с помощью свойства IndexName типа String. Текущий индекс выбирается из списка индексов, которые были заданы при создании таблицы. На этапе разработки приложения все возможные значения свойств IndexName содержат раскрывающиеся списки, доступные в Инспекторе объектов.

Наборы данных могут находиться в открытом или закрытом состоянии, на что указывает свойство Active типа Boolean. Если свойству Active установлено значение True, то набор данных открыт. Если значение свойства Active равно False (по умолчанию), то набор данных закрыт и его связь с БД разорвана.

Для вывода содержимого набора данных в табличном виде удобно использовать сетку, представленную компонентом DBGrid. Внешний вид сетки соответствует внутренней структуре таблицы БД и набора данных, при этом строке сетки соответствует запись, а столбцу – поле.

Компонент DataSource – источник данных. Используется как промежуточное звено между набором данных и визуальными компонентами, с помощью которых пользователь управляет этим набором данных. Для указания набора данных, с которым связан источник данных, используется свойство DataSet типа TDataSet последнего. Визуальные компоненты связаны с источником данных через свои свойства DataSource при проектировании в Инспекторе объектов.

# 2. Задания к лабораторным работам

**Вариант 1**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для ежедневного учета работы водителей в трамвайном депо. Учет ведется на основании документа – путевой лист. В путевом листе содержится следующая информация: дата, маршрут, график, смена, табельный номер водителя, фамилия, время начала работы, время окончания работы, номер вагона, простои на линии. Простои содержат следующую информацию: код простоя, начало простоя, окончание простоя. За водителем закреплен номер вагона и номер маршрута. Водитель может работать и на другом вагоне, и на другом маршруте. График содержит: идеальное (по графику) время начала работы, время окончания работы, номер маршрута, номер смены.

2. Создать запросы, в которых необходимо:

* вывести коды простоев за последний месяц, которые произошли в первую смену;
* вывести список водителей, работающих на “своем” и на “другом” маршруте;
* вывести табельные номера водителей, работающих по две смены в день.

3. С помощью запроса создать таблицу “Опасные маршруты”, содержащую список маршрутов, у которых за последний месяц были простои по вине дорожно-транспортных происшествий.

4. В путевом листе измените поле “Дата”, проставив текущую дату для маршрута № 2.

5. Определить, сколько простоев было по кодам простоя за последний месяц на каждом маршруте.

6. Вывести информацию о всех графиках для любого маршрута, заданного пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о водителе, маршруте, вагонах, простоях, графиках для каждого маршрута), а также текущей информации для путевого листа.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – путевой лист, подчиненная – информация о простоях.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Путевой лист”, сгруппировав данные по табельному номеру водителя. Вывести дату, номер маршрута, номер графика, номер смены, номер вагона, реальное время начала и окончания работы. Отсортировать данные по дате.

**Вариант 2**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для организации компьютерного учета перемещения узлов трамвая в ходе ремонтов. В накладной необходимо учитывать следующую информацию: номер трамвая, номер узла, дату установки узла на трамвай, дату снятия узла с трамвая, причину снятия узла, фамилию слесаря, получившего задание. Для слесаря определенной квалификации оформляются отдельно две накладные (для снятия или установки узлов вагона).

2. Создать запросы, в которых необходимо:

* вывести номера вагонов, в которых за последний месяц устанавливали один тип узла, например колесную пару;
* вывести номера узлов, которые были установлены и сняты в течение последнего квартала.

3. С помощью запроса создать таблицу ”Часто заменяемые узлы”, в которой вывести список названий узлов, снятых с вагона по причине, например, перегорания предохранителей в течение последнего квартала.

4. Обновить информацию о слесаре, который снимал детали по причине перегорания предохранителей, например: 01.01.2006.

5. Определить, сколько было замен узлов за определенный период на каждом вагоне.

6. Вывести информацию о всех вагонах, в которых произошли замены любого узла, заданного пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о вагоне, узле, слесаре, причинах замены узлов), а также текущей информации по накладной для установки узла на вагон и накладной для снятия узла с вагона.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – накладная для установки узла на вагон. Подчиненная – накладная для снятия узла с вагона.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Накладная для установки узла на вагон”, сгруппировав данные по номеру вагона. Вывести номер узла, дату установки узла. Отсортировать данные по номеру узла.

**Вариант 3**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета расхода электроэнергии в трамвайном управлении. На предприятии имеется три подстанции. На каждой подстанции стоит по три электросчетчика. Каждый счетчик фиксирует показания соответствующего участка. Ежемесячно работник предприятия должен отчитаться в показаниях счетчика и рассчитать расход электроэнергии в денежном выражении. С 23.00 до 6.00 действует ночной тариф, также существуют выходной (суббота, воскресенье) и праздничный тарифы. Показания всех счетчиков фиксируются при смене тарифа.

2. Создать запросы, в которых необходимо:

* вывести количество электроэнергии, которое потребила подстанция “Южная” в ночное время за последний месяц;
* вывести дату, когда показания счетчика не менялись, например в случае аварии на подстанции.

3. С помощью запроса создать таблицу “Наиболее потребляемые участки”, в которой будет список участков, где потребление электроэнергии за последний месяц в дневное время суток составило более 1000 кВт.

4. Увеличить цену тарифа на 20 % для дневного тарифа.

5. Определить количество потраченной электроэнергии на каждом участке за последний месяц. Сгруппировать по тарифам.

6. Вывести информацию о тарифах (название, время действия, цена), заданных пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о подстанции, счетчиках, тарифах), а также текущей информации для учета электроэнергии.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – учетная карточка электроэнергии, подчиненная – информация о тарифах.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Учетная карточка”, сгруппировав данные по номеру счетчика, тарифу. Вывести дату, время, показания счетчика. Вычислить количество электроэнергии, потребляемой по каждому тарифу.

**Вариант 4**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета перемещения комплектующих деталей компьютеров на предприятии: инвентарный номер компьютера, название и номер комплектующей детали, дата установки, рабочее место. Предприятие имеет несколько отделов, в каждом отделе несколько рабочих мест. Должности работников в разных отделах могут быть одинаковыми.

2. Создать запросы, в которых необходимо:

* вывести инвентарный номер и наименование ПК, на которые за последний месяц установили один тип комплектующих деталей, например CDROM;
* вывести список работников, на ПК которых устанавливались комплектующие детали в последний месяц.

3. С помощью запроса создать таблицу ‘‘Комплектующие детали для директора’’, в которой был список комплектующих деталей, установленных на ПК директора за последний месяц.

4. Для комплектующих деталей типа CDROM обновить наименование ’’Устройство для чтения компакт-дисков’’.

5. Определить, сколько рабочих мест в каждом отделе было модернизировано (установлены комплектующие детали) за последний год.

6. Вывести информацию об установках комплектующих деталей на ПК с указанием рабочего места для любого сотрудника, ФИО работника задается в режиме диалога.

7. Создать формы для ввода справочной информации (об отделах предприятия, рабочих местах, компьютерах, комплектующих деталях), а также текущей информации для накладной по установке деталей на ПК.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – накладная для установки деталей на ПК. Подчиненная форма – наличие комплектующих деталей на данном ПК.

9. С помощью Мастера отчетов создать отчет на основе таблицы ‘‘Накладная для установки КД’’, сгруппировав данные по инвентарному номеру ПК. Вывести комплектующие детали, дату установки КД, Отсортировать данные по номеру комплектующих деталей.

**Вариант 5**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета поступающих заявок от линейных работников на устранение неисправности в трамвае на транспортном предприятии. В базу данных входят: дата и время подачи заявки, характер неисправности, кто передал заявку, время получения заявки, кому передана заявка на исполнение. Если неисправность не устранена, то указать причину. Заявку подает водитель или кондуктор трамвая. Принимает заявку слесарь, ремонтник, электрик.

2. Создать запросы, в которых необходимо:

* вывести номера вагонов, в которых за последний месяц устранял неисправности один и тот же работник, например Петров;
* вывести список исполнителей с указанием их работы (номер вагона, характер неисправности, причина устранения неисправности) за последний месяц;
* вывести номера заявок, даты заявок, когда исполнили (сразу получили заявку, по мере ее поступления).

3. С помощью запроса создать таблицу ‘‘Неисправность – сломанные двери’’, в которой будет список вагонов и наименование причины неустранения заявки, если в течение последнего квартала была неисправность “сломанная дверь”.

4. Обновить в заявке поле “дата” на текущую дату для исполнителя “Сидоров”.

5. Определить, сколько было заявок каждого типа неисправности за определенный период для каждого вагона.

6. Вывести информацию о всех исполнителях, которые устраняют определенные неисправности, заданные пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о характере неисправности, работниках, делающих заявки, исполнителях, причинах неустранения неисправностей, вагонах), а также текущей информации для заявки на устранение неисправности.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заявка на устранение неисправности, подчиненная – информация о характере неисправности на данном вагоне (устраненной или нет) в течение последнего года.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заявка на устранение неисправности”, сгруппировав данные по номеру вагона. Вывести дату заявки, характер неисправности, исполнителя, причину неустранения заявки. Отсортировать данные по типу заявки.

**Вариант 6**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета рабочего времени сотрудников, т.е. организовать табель рабочего времени. Необходимо организовать учет следующих данных: табельный номер сотрудника, фамилия, имя, отчество, дата и время работы, причина невыхода на работу, адресные данные сотрудников, телефон, возраст, стаж работы. Также необходимо проводить поверку соответствия режима работы каждого сотрудника его индивидуальному графику работы.

2. Создать запросы, в которых необходимо:

* вывести список сотрудников, которые не вышли на работу по причине отгулов за последний месяц;
* вывести список сотрудников, указав, в каком режиме они работали

10-го числа последнего месяца;

* вывести список сотрудников, у которых поменялся режим работы.

3. С помощью запроса создать таблицу “Болеющие сотрудники”, содержащую список сотрудников, которые в последний месяц не выходили на работу по причине болезни.

4. Для ночных смен обновить время режима работы с 12 на 10 часов.

5. Определить, сколько времени работал каждый сотрудник в последнем месяце.

6. Вывести информацию о сотрудниках (табельный номер, ФИО, адрес, режим работы), которые за последний год, например, не выходили на работу. Причина невыхода на работу задается в режиме диалога.

7. Создать формы для ввода справочной информации (о сотрудниках, режимах работы, причинах невыхода на работу), а также текущей информации для заполнения табельного листа.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – табельный лист, подчиненная – сведения о сотруднике.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Табельный лист”, сгруппировав данные по табельному номеру водителя. Вывести дату, режим работы, время режима работы, время работы (реальное), причину невыхода на работу (если таковая есть). Отсортировать данные по дате.

**Вариант 7**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для ежедневного учета движения библиотечного фонда: автор и название книги, год издания, фамилия читателя, причина отсутствия книги в библиотеке. Книги необходимо учитывать по категориям. Читательская карточка должна содержать: фамилию, адрес, место работы, телефон рабочий, телефон домашний. В случае задержки книги читателем принимаются меры, которые фиксируются в читательской карточке.

2. Создать запросы, в которых необходимо:

* вывести список книг (авторы, название, год издания) категории “детективы”, которые заказывали читатели за последний месяц;
* вывести список читателей, заказавших книги в прошлом месяце. Указать, возвратили или нет книгу;
* вывести список книг, которые запрашивал читальный зал (т.е. дата возврата соответствует дате заказа) в последний месяц.

3. С помощью запроса создать таблицу “Злостные должники”, в которой указать список читателей (ФИО, адрес, телефон), которые не сдали книги после второго телефонного звонка.

4. Присвоить читателю имя “должник“, если книга не возвращена на следующий день после установленной даты возврата.

5. Определить, сколько должников не вернули книги каждой категории.

6. Вывести список возвращенных книг (авторы, название книги, год издания). Дата возврата книги задается пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о категориях книг, книгах, читателе, мерах, принимаемых к должникам), а также текущей информации “Заказ читателя”, “Возврат книги”.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заказ читателя, подчиненная – возврат книги.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заказ читателя”, сгруппировав категории книг. Вывести дату заказа, ФИО читателя, автора, название книги, год издания. Отсортировать данные по ФИО читателя.

**Вариант 8**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для ежедневного учета посещений врачей любой квалификации в поликлинике. В больничной карточке указывается: дата приема, фамилия, имя и отчество больного, возраст, адрес, ФИО врача, поставленный диагноз, назначенная процедура, номер больничного, дата открытия и закрытия больничного.

2. Создать запросы, в которых необходимо:

* вывести список больных, посетивших педиатра за последний месяц с диагнозом ОРЗ;
* вывести список больных и процедур, назначаемых им или нет за последний месяц;
* вывести номера больничных листов, которые были закрыты в день их открытия.

3. С помощь запроса создать таблицу “Возможная эпидемия”, содержащую список больных, которые за последнюю неделю заболели гриппом.

4. Для больных с диагнозом ОРЗ и датой открытия больничного листа недельной давности закрыть больничный лист.

5. Определить, сколько больных в возрасте от 40 до 50 лет были на больничном в течение года с одинаковыми диагнозами.

6. Вывести информацию о всех больных, которых принимал врач. ФИО врача задается пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о враче, квалификации врача, больных, диагнозах, процедурах), а также текущей информации для больничной карточки.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – больничная карточка, подчиненная – регистрация больничного листа.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Больничная карточка”, сгруппировав данные по ФИО врача. Вывести дату приема, ФИО больного, процедуры. Отсортировать данные по ФИО больного.

**Вариант 9**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета движения компьютерной техники на предприятии. Закрепить за каждым рабочим местом необходимое количество компьютерной техники. Присвоить каждой единице компьютерной техники инвентарный номер и учитывать ее перемещение по рабочим местам с указанием даты и фамилии электронщика, осуществившего перестановку или списание на склад.

2. Создать запросы, в которых необходимо:

* вывести список рабочих мест, с которых в течение последнего года списали компьютерную технику одного вида, например принтеры;
* вывести список электронщиков с указанием выполненной ими работы в течение последнего месяца.

3. С помощью запроса создать таблицу “Работники с новой компьютерной техникой”, где необходимо вывести список сотрудников, которым в последний месяц установили новые мониторы.

4. Переименовать вид компьютерной техники “Ксерокс” на “Копировальный аппарат”.

5. Определить, сколько техники по видам было списано каждым электронщиком в течение последнего года.

6. Вывести информацию о компьютерной технике, которая принадлежит определенному работнику, ФИО работника задается в режиме диалога.

7. Создать формы для ввода справочной информации (о рабочих местах, компьютерной технике, виде компьютерной техники, электронщиках), а также текущей информации для заявки на перемещение компьютерной техники.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заявка на перемещение компьютерной техники, подчиненная – информация о рабочих местах (откуда и куда будет перемещаться компьютерная техника).

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заявка на перемещение компьютерной техники”, сгруппировав данные по новому рабочему месту, куда перемещена компьютерная техника. Вывести дату заявки, перемещаемую компьютерную технику, электронщика. Отсортировать данные по дате.

**Вариант 10**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета ДТП в городе для ГАИ. Рассмотреть ДТП, когда автомобиль сбивает пешехода. При организации учета ДТП необходимо заносить данные: дата, время, ФИО шофера, ФИО пешехода, ФИО свидетелей, тяжесть последствий (без травм, легкая, тяжелая травма, смертельный случай), виновность участников ДТП, адреса участников ДТП.

2. Создать запросы, в которых необходимо:

* вывести информацию о машинах, которые за последний месяц сбили пешехода на перекрестке Ленина-Луначарского;
* вывести список свидетелей аварий, которые происходили накануне и закончились тяжелой травмой для пешехода;
* вывести номера аварий, у которых не менее двух свидетелей.

3. С помощью запроса создать таблицу “Опасные перекрестки”, содержащую список перекрестков, на которых за последний месяц произошли аварии одного типа, например сбит пешеход по вине неработающего светофора.

4. Изменить адрес водителей: Свердловск на Екатеринбург.

5. Определить, сколько аварий было по видам виновности участников ДТП за последний год.

6. Вывести информацию о пострадавших в зависимости от тяжести последствия, заданной пользователем в режиме диалога.

7. Создать формы для ввода справочной информации с помощью Мастера форм по справочникам: о машине, пострадавших, виновности, тяжести последствия, свидетелях. Создать форму “Учет ДТП”.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – учетная карточка ДТП, подчиненная форма – информация о свидетелях.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Учет ДТП”, сгруппировав данные по виновности участников ДТП. Вывести дату аварии, номер машины, ФИО водителя, ФИО пострадавшего, место аварии, тяжести последствия ДТП. Отсортировать данные по степени последствия ДТП.

**Вариант 11**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для диспетчера сети аптек. Должны быть указаны адрес аптеки, телефон, специализация аптеки. Информация о лекарствах: наименование лекарств, их характеристика – от каких заболеваний, доза в упаковке, доза применения, расфасовка. Информация о наличии лекарств должна содержать: дату поступления, количество упаковок лекарства, цену, название завода-изготовителя, дату продажи последней упаковки лекарства.

2. Создать запросы, в которых необходимо:

* вывести информацию об аптеках, которые закупают лекарства в Венгрии;
* вывести информацию о том, в каких аптеках имеется в продаже анальгин и в каких количествах;
* вывести лекарства, на которые не устанавливается НДС.

3. С помощью запроса создать таблицу “Наиболее дорогие лекарства”, содержащую список лекарств от эпилепсии, которые закупали аптеки в последний год при цене за упаковку более 1000 р.

4. Увеличить на 5 % цену продажи лекарств, поступивших в аптеку за последнюю неделю.

5. Определить сумму, которую затратила каждая аптека при покупке лекарств от гриппа за последний год.

6. Вывести информацию о лекарствах, которые покупали за последний год в зависимости от заболевания, заданного пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (об аптеках, лекарствах, заболеваниях), а также текущей информации о поступлении лекарств в аптеки.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – поступление лекарств в аптеки, подчиненная форма – розничная продажа лекарств.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Поступление лекарств в аптеки”, сгруппировав данные по номерам аптек. Вывести дату поступления лекарств, количество упаковок, цену упаковки, цену продажи. Отсортировать данные по наименованию лекарств.

**Вариант 12**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для диспетчера телефонной сети в городе. Должны быть указаны: номер телефона, список абонентов, адрес абонента. Информация об АТС: номер АТС, ее адрес, телефон ремонтной службы, характеристика АТС. Диспетчер должен знать, какой номер неисправен, дату и время заявки абонента, дату и время устранения неисправности. Для оформления заявки также необходимы следующие сведения: какой мастер будет выполнять ремонт, характер жалобы абонента, характер неисправности.

2. Создать запросы, в которых необходимо:

* вывести номера телефонов, в которых за последний месяц устранили одну и ту же неисправность, например заменили номеронабиратель;
* вывести номера телефонов, не исправных и отремонтированных в течение последнего месяца;
* вывести номера телефонов, которые в течение последнего месяца неоднократно подавали заявки на ремонт.

3. С помощью запроса создать таблицу ”Возможный обрыв кабеля”, в которой указать номера телефонов и адреса абонентов, от которых поступили за последние сутки заявки с жалобой: отсутствие зуммера.

4. При установке цифрового оборудования необходимо для части номеров, а именно с 0001 до 4999, перевести номер АТС с 51 на 72.

5. Определить, сколько неисправностей каждого типа было устранено на каждой АТС за последний месяц.

6. Вывести информацию об абонентах, которых за последний год обслуживал мастер, фамилия которого задана в режиме диалога.

7. Создать формы для ввода справочной информации (об АТС, абонентах, Мастерах, характере жалоб абонента, характере неисправностей), а также текущей информации о заявках на устранение неисправностей.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заявка на устранение неисправностей, подчиненная форма – информация об абоненте.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заявка на устранение неисправностей”, сгруппировав данные по АТС. Вывести номер телефона, дату и время заявки от абонента, характер жалобы, ФИО мастера. Отсортировать данные по характеру жалобы.

**Вариант 13**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для диспетчера технического обслуживания сети кафе. Информация о кафе следующая: название кафе, адреса кафе, ФИО директора, телефон кафе, наличие технического оборудования (холодильник, печь, миксер, кофеварка и др.). В случае поломки оборудования оформляется заявка, в которой указывается дата поломки оборудования, дата ремонта, его характер (быстрый ремонт, ремонт со сменой деталей, замена оборудования на новое). Ремонтник может иметь определенную квалификацию и соответственно ремонтирует только определенный тип оборудования (холодильные установки, электрооборудование и др.).

2. Создать запросы, в которых необходимо:

* вывести информацию о кафе, в которых за последний год ремонтировалось электрооборудование и какое именно;
* вывести информацию о кафе, в которых установлены миксеры и были или нет заявки на их ремонт;
* вывести информацию о работниках, которые ремонтировали оборудование в день заявки, указать характер ремонта.

3. С помощью запроса создать таблицу “Новое оборудование”, где указать, в каких кафе за последний год заменили оборудование в связи с его поломкой.

4. Для всех вчерашних заявок установить сегодняшнюю дату ремонта.

5. Определить, сколько ремонтов по каждому типу оборудования было произведено за последний год в каждом кафе.

6. Вывести информацию о ремонтниках, которые ремонтировали оборудование за последний год в кафе, название которого было задано в режиме диалога.

7. Создать формы для ввода справочной информации (о кафе, оборудовании, типах оборудования, характере ремонта), а также текущей информации для заявки на ремонт оборудования.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заявка на ремонт оборудования, подчиненная форма – информация о кафе и оборудовании в нем.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заявка на ремонт оборудования”, сгруппировав данные по кафе. Вывести дату заявки, дату ремонта, ремонтируемое оборудование. Отсортировать данные по характеру ремонта.

**Вариант 14**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета наличия корма в зоопарке. Информация о животных следующая: название, вид, пол, количество. Рацион питания составляется в зависимости от вида и пола животного. Пусть каждое животное кормят один раз в день, животному полагается как минимум два-три вида пищи за раз. Время кормления каждого вида животного различное.

2. Создать запросы, в которых необходимо:

* вывести список животных (вид, пол, количество), у которых в последнем месяце было усиленное питание;
* вывести список продуктов с указанием количества и даты приема.

3. С помощью запроса создать таблицу “Самки-хищницы на витаминном питании”, содержащую список животных (самок-хищников), которые в последний месяц находятся на витаминном питании.

4. Увеличить на 20 % количество выдаваемых продуктов, калорийность которых выше 500 ккал, для животных, находящихся на усиленном питании.

5. Определить, сколько килограммов каждого продукта за последний месяц было съедено каждый животным.

6. Вывести информацию о животных и выданных им продуктах в указанное в режиме диалога время.

7. Создать формы для ввода справочной информации (о видах животных, поле животных, рационе, продуктах питания), а также текущей информации о приеме пищи животными.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – прием пищи животными, подчиненная форма – информация о животных.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Прием пищи животными”, сгруппировав данные по рациону. Вывести дату, время кормления животного, название животного, название продуктов, количество продуктов. Отсортировать данные по животному.

**Вариант 15**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для ежедневного учета изготовленных деталей и отправки готовых деталей на складе. Определенная деталь выпускается слесарем конкретной квалификации и разряда. Деталь имеет определенную сложность и может изготавливаться не один день. Слесарь может изготавливать деталь при наличии квалификации не ниже требуемой.

2. Создать запросы, в которых необходимо:

* вывести список деталей высокой сложности, изготовленных в течение последнего месяца, если работу выполняли слесари с разрядом с 7 по 9;
* вывести список слесарей 3 разряда, которые в течение последнего месяца изготовляли или нет детали пониженной сложности;
* вывести список табельных номеров слесарей, которые в течение месяца изготовляли детали за один день.

3. С помощью запроса создать таблицу “Ученики”, в которой детали повышенной сложности выполняли слесари, чья квалификация ниже требуемой для изготовления данных деталей.

4. В заявке на изготовление сложных деталей, которые были заказаны сегодня, обновить дату окончания изготовления деталей на завтрашнюю.

5. Определить, сколько деталей каждой сложности было изготовлено каждым слесарем в течение последнего квартала.

6. Вывести информацию о слесарях, в течение последнего месяца изготовивших детали, название которых задано пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о слесаре, его квалификации, деталях, их сложности), а также текущей информации о заявке на изготовление детали.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заявка на изготовление детали, подчиненная форма – информация о слесаре.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заявка на изготовление детали”, сгруппировав данные по сложности детали. Вывести дату заявки, дату изготовления детали, информацию о слесаре, который ее изготовил. Отсортировать данные по табельному номеру слесаря.

**Вариант 16**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для заказа билета в самолете, в которой указать дату и время заказа билета, рейс, место, дату и время отправления, дату и время прилета, ФИО пассажира, номер паспорта, цену билета. Рейс может быть международным, местным, чартерным, регулярным. Число мест в самолете зависит от модели самолета.

2. Создать запросы, в которых необходимо:

* вывести список пассажиров, заказавших билеты вчера на завтрашние рейсы в Москву;
* вывести список пассажиров, заказавших билеты в день отлета.

3. С помощью запроса создать таблицу “Чартерные рейсы”, в которой указать список пассажиров, отлетающих сегодня на чартерных рейсах на ТУ-154.

4. Обновить цену билетов, увеличив ее на 20 % с 1 числа следующего месяца.

5. Определить, сколько было заказано билетов по категориям рейсов за последний месяц.

6. Вывести информацию о рейсах, отправляющихся в ближайший месяц в город, который задан пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о номерах самолетов, моделях самолетов, рейсах, категории рейсов, пассажирах), а также текущей информации о заказах билетов.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – заказ билетов, подчиненная форма – информация о рейсе.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Заказ билетов”, сгруппировав данные по категории рейсов. Вывести дату и время заказа, дату и время отлета, дату и время прилета, ФИО пассажира. Отсортировать данные по категориям рейсам.

**Вариант 17**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для определения рейтинга канала с помощью анкетирования. В анкете указывается номер канала, название канала, характеристика (метровый, дециметровый), название передачи, вид передачи (новости, телесериал, спортивная, развлекательная или познавательная передача), рейтинг (с 1 до 10), время показа, день недели показа. Рейтинг передачи определяется по семибальной шкале.

2. Создать запросы, в которых необходимо:

* вывести список познавательных передач, транслирующихся по вторникам, с рейтингом не ниже 3, с указанием номера канала, времени показа по вторникам;
* вывести список познавательных передач, вошедших или нет в анкеты опрашиваемых, с указанием рейтинга;
* вывести список номеров каналов, которые смотрят в зависимости от дня недели, т.е. передачи с наивысшим рейтингом данного дня недели.

3. С помощью запроса создать таблицу “Наивысший рейтинг I канала”, где вывести все передачи первого канала, которые по анкетам заняли наивысший рейтинг, с указанием дня недели и времени показа.

4. Обновить время начала и время окончания для передачи “Вести” по каналу РТР с 20.00 на 20.30.

5. Определить рейтинг каналов по видам передач.

6. Вывести список передач, занимающих последнее место, которые показывают в диапазоне частот (метровом или дециметровом), заданном пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о каналах, характеристиках каналов, передачах, видах передач, рейтинге), а также текущей информации для анкеты.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – анкета, подчиненная форма – информация о передачах анкетируемого канала.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Анкета”, сгруппировав данные по каналам. Вывести название передач, время показа, день недели показа, рейтинг. Отсортировать передачи по рейтингу и определить рейтинг каждого канала.

**Вариант 18**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета фондов в музее. Экспонаты музея классифицируются по видам (мебель, скульптура, книги, сувениры). Каждый зал музея имеет название, номер. Каждый экспонат имеет свой инвентарный номер, известна дата поступления в музей, кто передал экспонат (организация или частное лицо), как (на благотворительной основе или музей приобретал за деньги), историческая ценность экспоната. Указать, выставлен ли экспонат в залах или находится в запасниках. Регистрируется дата последней реставрации, срок выставочной экспозиции до следующей реставрации.

2. Создать запросы, в которых необходимо:

* вывести список и определить ценность скульптур, находящихся в запаснике, которые необходимо будет реставрировать в следующем году;
* вывести данные о физических лицах, которые продали или передали музею экспонаты;
* вывести список экспонатов, которые в последний год музей прибрел за деньги, причем цена приобретения соответствовала ценности экспоната.

3. С помощью запроса создать таблицу “Меценаты”, содержащую список лиц (физических или юридических), которые безвозмездно передали музею экспонаты, ценность которых более 10 тыс. рублей.

4. Обновить срок следующей реставрации, заменив его на 2008 год, для мебели, которая реставрировалась в 1998 году.

5. Определить, сколько экспонатов по видам находится в запасниках и на какую сумму ценности.

6. Вывести список экспонатов ценностью выше 5 тыс. рублей, находящихся в зале, который задан пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о залах, видах экспонатов, физических и юридических лицах, которые передавали экспонаты музею, о форме приобретения экспоната), а также текущей информации о фонде музея.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – фонд музея, подчиненная форма – информация о лице, передавшем экспонат.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Фонд музея”, сгруппировав экспонаты по залам. Вывести название экспоната, ценность, дату последней реставрации. Отсортировать данные по ценности экспоната. Определить ценность экспонатов в каждом зале и общую ценность всего фонда музея.

**Вариант 19**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета больных в санатории. Карточка больного содержит следующие данные: ФИО, возраст, пол, основной диагноз, откуда приехал, на какой срок, категория путевки (одно- или двухместный номер), ФИО лечащего врача, список и даты посещения назначенных процедур.

2. Создать запросы, в которых необходимо:

* вывести список больных пенсионного возраста с диагнозом ожирение, посетивших бассейн на последней неделе;
* вывести список больных женщин с диагнозом гипертония, приехавших в санаторий неделю назад и выполняющих или нет ЛФК;
* вывести список больных, которые отдыхали в санатории один день.

3. С помощью запроса создать таблицу “Приехавшие вчера”, содержащую список больных, которые заехали в санаторий вчера в одноместные номера, лечащий врач которых невропатолог.

4. Продлить время отдыха больных, проживающих в № 321 на одну неделю.

5. Определить, сколько больных по каждому диагнозу было принято в санатории за последний год.

6. Вывести информацию о посещении процедур больным, фамилия которого задана пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о врачах, их квалификации, диагнозе больных, процедурах, номерах санатория), а также текущей информации для больничной карты о посещении процедур больными.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – больничная карта, подчиненная форма – посещение больным назначенных ему процедур.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Больничная карта”, сгруппировав данные по диагнозу. Вывести дату заезда, дату отъезда, ФИО больного, возраст, пол, адрес, ФИО лечащего врача. Отсортировать данные по лечащему врачу.

**Вариант 20**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для организации компьютерного учета дисконтных карт покупателей в магазине. Необходимо учитывать следующую информацию: дата покупки, номер дисконтной карты, ФИО покупателя, товар, фирма, скидка. Скидка на товар дается последовательно при каждой следующей покупке: 2, 4,
6, 8, 10 %. Затем выдается новая дисконтная карта с теми же процентами. Первоначально дисконтная карта выдается покупателю, если он сделал покупку на сумму более 10 тыс. рублей.

2. Создать запросы, в которых необходимо:

* вывести список и цену товара, проданного в последний месяц со скидкой

6 %;

* вывести список товара ценой более 20 тыс. рублей, проданного или нет со скидкой (указать какой);
* вывести список покупателей, которые еще не меняли дисконтную карту.

3. С помощью запроса создать таблицу “Активные покупатели года”, содержащую список покупателей, которые в течение года поменяли дисконтную карту.

4. Увеличить цену товара фирмы “Sony” на 2 %.

5. Определить, сколько товара каждой фирмы за последний месяц было продано со скидкой и на какую сумму.

6. Вывести список товара определенной фирмы, проданного за последний месяц со скидкой 10 %. Фирма товара задана пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о товарах, фирмах товара, дисконтных картах, скидках, покупателях), а также текущей информации о покупке товара со скидкой.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – покупка товара со скидкой, подчиненная форма – информация о покупателях этого товара со скидкой.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Покупка товара со скидкой”, сгруппировав данные по номеру дисконтной карты. Вывести дату покупки, товар, покупателя, скидку. Отсортировать данные по величине скидки и вывести сумму товара со скидкой и общую сумму товара.

**Вариант 21**

1. Спроектировать базу данных в первой, второй и третьей нормальных формах для учета прихода и расхода денег на счету клиента по пластиковым картам. Банк имеет 5 видов пластиковых карт с разными процентами. Проценты начисляются по пластиковой карте в первый день квартала. Информация о клиенте: номер пластиковой карты, ФИО клиента, адрес, место работы, дата открытия пластиковой карты, дата закрытия пластиковой карты, количество денег на счету в данный момент. Деньги на счет могут поступать с места работы, с другого счета, либо счет может пополнять сам клиент.

2. Создать запросы, в которых необходимо:

* вывести информацию о клиентах пластиковой карты “Юнион-карт”, которые в последний год сами пополняли свой счет;
* вывести информацию о количестве денег на счету клиентов пластиковой карты “Visa-electron”.

3. С помощью запроса создать таблицу “Богатые клиенты”, где вывести список клиентов пластиковой карты “Visa-Gold”, у которой пополнялся счет в течение последнего квартала на сумму более 60 тыс. руб.

4. Установить новую дату закрытия счета клиентов “Юнион-карт”, которые работают в магазине № 20.

5. Определить, сколько денег по видам пластиковых карт расходовали клиенты в последний месяц.

6. Вывести информацию о поступивших деньгах за последний квартал на счет, номер которого задан пользователем в режиме диалога.

7. Создать формы для ввода справочной информации (о клиентах, виде пластиковых карт, видах поступления денег), а также текущей информации о приходе денег на счет и расходе денег со счета.

8. Создать многотабличную форму с помощью Мастера форм. Главная форма – расход денег со счета, подчиненная форма – приход денег на счет.

9. С помощью Мастера отчетов создать отчет на основе таблицы “Приход денег на счет”, сгруппировав данные по видам пластиковых карт. Вывести дату прихода денег, номер счета, вид поступления денег, сумму прихода. Отсортировать данные по номеру счета. Определить общие суммы прихода денег по номеру счета и по видам пластиковых карт.

**Список литературы**

1. Гофман В.Э. Delphi 5 / В.Э. Гофман, А.Д. Хомоненко. СПб. : БХВ Санкт-Петербург, 2000. 800 с.

2. Фаронов В.В. Delphi 5: учеб. курс / В.В. Фаронов. М. : Нолидж, 2001. 608 с.