



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Рубцовский индустриальный институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»
(РИИ АлтГТУ)

Е.А. Дудник

ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Методические указания о порядке выполнения курсовой работы для
студентов направления подготовки «Информатика и вычислительная
техника»

Рубцовск 2021

УДК 519.6

Дудник, Е.А. Проектирование баз данных: Методические указания о порядке выполнения курсовой работы для студентов направления подготовки «Информатика и вычислительная техника / Е.А. Дудник; Рубцовский индустриальный институт. – Рубцовск: РИИ, 2021. – 19 с. [ЭР].

В данных методических указаниях изложен порядок выполнения курсовой работы по дисциплине «Базы данных», содержатся правила оформления курсовой работы и перечень предлагаемых тем. Рассчитано на студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» для получения практических навыков в создании программного пакета по учебным информационно-экономическим задачам.

Рассмотрены и одобрены
на заседании кафедры ПМ РИИ.
Протокол № 9 от 18.03.21г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ	8
Содержание введения пояснительной записки	9
Содержание конструкторской части пояснительной записки	9
Содержание технологической части пояснительной записки	9
Содержание экспериментальной части пояснительной записки.....	10
Содержание заключительной части пояснительной записки	11
Список литературы	11
Содержание приложения пояснительной записки	11
2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	11
ЗАДАНИЯ ПО ТЕМАМ	12
СПИСОК ЛИТЕРАТУРЫ.....	14

Введение

Дисциплина "Базы данных" изучается в 6 семестре и служит для расширения знаний в области организации баз данных и получения навыков программирования, ведения базы данных в среде СУБД (системы управления базой данных) Visual Foxpro.

В результате изучения дисциплины студенты должны овладеть: основными конструкциями языка СУБД Visual Foxpro, приемами организации и ведения баз данных; составления программы в среде СУБД Visual Foxpro, навыками отладки программ.

Целью курсового проекта является усвоение и закрепление теоретического материала, приобретение практических навыков работы с СУБД при выполнении основных элементарных операций с базами данных. Курсовой проект составляется на основе лабораторных работ, охватывающих основные разделы курса. Студенты на основании исходных данных должны разработать вычислительную систему с базой данных, организованной в среде Visual Foxpro.

Проектирование структур данных

Программирование баз данных в целом процесс чисто технологический, и алгоритм ведения работ по такой разработке достаточно простой и наглядный. Но алгоритм этот не будет линейным, любая база решает несколько параллельных и достаточно независимых задач:

1. Любая база данных - это система интерфейсов: между программой и пользователем, между различными автономными базами, обменивающимися данными.

2. Любая база - это сложные конструкции данных, и нужно специально заботиться об их сохранности, компактности, защищённости от несанкционированного доступа, информационной достаточности для решаемой задачи.

Проектирование структур данных - это борьба между двумя противоположными требованиями: требованием информативности данных и их компактности. Требование информативности исходит из существа задачи, это в некотором смысле объективное требование. Требование компактности проистекает из ограниченных возможностей существующих компьютерных систем. Может показаться, что с ростом ёмкости магнитных носителей проблема компактности данных ослабевает, но нельзя забывать, что с ростом количества байт в базе затрудняется поиск нужной информации.

Необходимо определиться, насколько структура базы должна быть гибкой. Системы управления базами данных подразумевают возможность не только обработки баз с фиксированной структурой, но и изменения структуры базы в процессе работы с ней. Необходимо определиться, насколько глубокие

изменения структуры базы допустимы. Чем больше возможностей изменить структуру базы, тем сложнее логика программы.

Что лучше: одна большая общая база или несколько маленьких специализированных? Если, к примеру, мы создаем базу данных о работниках предприятия, то можем занести все данные о всех работниках в одну общую базу, а можем создать отдельные базы для каждого отдельного подразделения. Специализированные базы позволяют экономить память, так как необходимая структура от подразделения к подразделению может меняться и, чтобы всё учесть, нужно заполнять для всех сотрудников все возможные поля, что, конечно, неэкономично, но, с другой стороны, большая общая база усложняет процедуры обработки.

Структуры данных и процедуры их обработки, конечно, вещи взаимосвязанные, но, тем не менее, есть смысл задать вопрос о приоритете. Однако понятно, что проблема организации структуры данных первична.

Ясно, что данные в базе должны быть каким-либо образом упорядочены. Предположим, что порядок должен быть алфавитным, но по какому полю упорядочивать записи? Кроме того, поля могут иметь различную природу и, например, по числовым полям записи можно упорядочивать по возрастанию. Можно создать и какой-то свой собственный порядок, просто расположив записи так или иначе. Проблема заключается в том, что для разных процедур обработки информации могут понадобиться разные порядки. Поэтому разумно создать специальную конструкцию, в которой бы хранились различные порядки записей.

Конечно, сейчас базы данных не делаются на обычном Паскале, не имеющем средств визуального программирования. Сейчас всё происходит очень технологично и в распоряжении программиста имеются очень мощные средства работы. «Недостаток» современных средств в том, что они очень удобны в работе. Очень многое, о чём надо бы заботиться программисту, берёт на себя система программирования, причём даже довольно трудные вещи. В системе программирования заложено всплывающие меню, ниспадающие меню, линейка прокрутки и т.п. Программы управления базами данных не слишком изобилуют логическими сложностями, поэтому проект такой программы можно составить достаточно подробно.

Интерфейс с пользователем

Общение пользователя с базой данных заключается в запросах пользователя на выполнение нужных ему операций над данными и выдаче базой ответов на запросы. Операции над данными можно разбить на систему классов операций:

1. Ввод новых данных.
2. Редактирование имеющихся данных.
3. Просмотр данных.
4. Удаление данных.

Каждый класс может быть достаточно сложным и состоять из целой группы операций или даже из группы классов операций. Например:

Просмотр данных:

- Печать данных.
- Вывод данных на экран монитора.

Таким образом, всё множество операций, могущих понадобиться пользователю, выстраиваются в виде древовидного меню операций, и проектирование интерфейса - это проектирование системы меню так, чтобы:

1. Путь до требуемой команды был максимально коротким.
2. Наиболее часто используемые команды были расположены на верхних уровнях меню.
3. Пользователь мог удобно отслеживать свой путь по дереву меню.
4. Команды группировались в соответствии с некоторой естественной и понятной пользователю логикой.

Ещё одна важнейшая проблема интерфейса - это различные дополнительные услуги пользователю. Например:

1. Система подсказок. Важно, чтобы пользователь мог получить необходимую подсказку в том момент и в том месте, в котором у него возникла проблема. Это можно осуществить двумя способами. Во-первых, можно в главном меню создать справочную службу и дать пользователю возможность, не прерывая текущей работы, обратиться к справочной службе. Во-вторых, можно прогнозировать типы ошибок, которые возможны в том или ином случае, и оказывать конкретную помощь без обращения к общей справочной службе.

2. Возможность отмены неверного решения.

3. Система настроек. Существует тип команд, для выполнения которых есть различные варианты. Например, выводить на экран монитора можно в самом различном виде: в виде таблиц, списков, столбцов и т.д. Если перечислять все возможные способы выполнения таких команд и записывать в виде пунктов меню, то программа получится слишком громоздкой.

Основы языка структурированных запросов (SQL)

SQL – язык запросов, основан на наборах записей. Он не имеет никаких команд для отображения или управления информацией. SQL предназначен для хранения и поиска данных.

Для создания запроса можно использовать окно Command или исходный текст программного модуля. В результате запроса получается файл, содержащий данные, удовлетворяющие условию запроса. В SQL выбирается и обрабатывается множество записей, к примеру, результатом команды SEEK является конкретная запись.

Команда SELECT языка SQL

1. Запрос данных

Select * <имя таблицы>

В результате выполнения команды создается файл, содержащий все поля данной таблицы.

2. Запрос данных с использованием арифметических выражений

Select <имя поля1...[, имя поля2]...[, арифметическое выражение] from <имя таблицы>

В результате выполнения команды создается файл, содержащий указанные поля из данной таблицы и результаты арифметических выражений.

3. Отбор запросов с помощью условия WHERE

Select * <имя таблицы> where <условие>

В условиях Where используются следующие условия поиска:

а) операторы сравнения (<, >, >=, <=, =, <>);

б) диапазона BETWEEN – предназначен для проверки на вхождения в диапазон требуемых значений:

Where <имя поля> between <значение1> and <значение2>;

в) вхождения IN – определяет, есть ли заданное значение в списке

where <поле> in (значение1, ..., значениеN);

г) сопоставление с образцом Like – так же, как в DOS, можно использовать символы маски (символы ‘_’, ‘%’ аналогичны ‘?’, ‘*’ в DOS):

where <имя поля> like “C%” - выбор записей, у которых заданное имя поля начинается с буквы “C”;

д) неопределенное значение IS NULL - проверяют неопределенное значение, которое не задавалось. Неопределенные значения означают, что имеются проблемы с записью или что данные должны быть отброшены.

Where <имя поля> not IS NULL - данное поле определено;

е) составные условия (not, and, or).

4. Упорядочение результатов запроса ORDER BY

Можно упорядочить результаты по заданному ключу:

Select * from <имя таблицы> order by <ключ>

5. Операция объединения

а) внутреннее объединение – двух и более таблиц по заданному условию:

select <псевдоним1>.<имя поля1 > [, ... < псевдоним2>.<имя поляN>... <арифметические выражения>] **where** <условия объединения

т.е. <псевдоним1>.<имя поля1> = <псевдоним2>.<имя поля1 >>;

б) внешнее объединение – когда для связи между двумя и более таблицами используется команда WHERE:

where <псевдоним1>.<имя поля1> = <псевдоним2>.<имя поля1 >

6. Выполнение операций добавление, изменение и удаление

а) добавление записи Inset:

Insert into <имя таблицы> (<имя поле1>[, имя поле2, ...]) values (<значения1>[, значения2, ...])

Добавляет запись с заданным значением для каждого поля.

б) Изменение записи update:

```
update < имя таблицы > set <имя поле1> = <значение>
```

в) Удаление записи delete:

```
delete from < имя таблицы > where <условия>
```

В результате действия этой команды записи, удовлетворяющие данному условию, будут помечены на удаление.

Просмотр – это способ визуализации данных.

1. Создание определения просмотра:

```
Create SQL view <имя просмотра> as select * from <имя таблицы>
```

2. Использование просмотра:

```
Use < имя просмотра >
```

```
Browse
```

Использование команды SQL и сохранение команды в виде просмотров избавляет от повторной операции набора команды.

1. СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

В предлагаемых заданиях представленная структура данных соответствует оперативной информации по заданной теме или информации, изменяющейся после определенного периода расчетов. В оперативную информацию не входят поля, имеющие большую длину, такие данные хранятся в базе для справочной информации. Связь оперативной и справочной информации обычно реализуется через коды или ключевые поля. Необходимо нормализовать базу данных, создать отдельные таблицы базы данных, в которых все поля связаны с ключевым полем. Таблиц с одинаковыми ключевыми полями быть не должно. Необходимо также отделить справочную информацию (расшифровки кодов, содержащиеся в исходных данных) и оперативную информацию.

В состав курсовой работы входят: комплекс программ, расчетно-пояснительная записка и приложение (листинг программы).

Комплекс программ представляет собой законченный программный продукт, который может настраиваться на конкретную программно-техническую среду ЭВМ. Взаимодействие пользователя осуществляется на основе простого общепринятого меню. Комплекс программ должен содержать необходимые подсказки и помощь как по эксплуатации, так и для интерпретации полученных результатов.

Расчетно-пояснительная записка должна содержать следующие разделы:

1. ВВЕДЕНИЕ

2. КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1. Задание

2.2. Построение функциональной схемы

2.3. Выделение справочной информации

- 2.4. Постановка задачи
- 2.5. Схема структур таблиц базы данных со связями
- 2.6. Формы таблиц и отчетов

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

- 3.1. Назначение и условия применения программы
- 3.2. Руководство системного программиста
- 3.3. Руководство пользователя

4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

- 4.1. Экранные формы (контрольный пример)
- 4.2. Отчеты (контрольный пример)

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

Кроме того, пояснительная записка должна иметь оглавление и приложение.

Содержание **введения** пояснительной записки

Во введении даются общие сведения о теме курсовой работы, назначении и цели работы. Указывается необходимость и актуальность поставленной задачи. Описывается содержание пояснительной записки по параграфам. Представляется заказчик и исполнитель. Определяются основные задачи и реализуемые общие функции в курсовой работе.

Содержание **конструкторской части** пояснительной записки

2.1. Первый параграф конструкторской части содержит задание.

2.2. Второй параграф конструкторской части содержит постановку задачи. Краткую характеристику задачи, представление входной и выходной информации. Описывается структура таблиц базы данных. Выделяется справочная информация. Указываются ключевые поля индексных файлов. Строится схема структур таблиц базы данных со связями.

2.3. В третьем параграфе конструкторской части приводится функциональная схема работы по заданной теме, алгоритм. Приводятся применяемые расчетные формулы.

2.4. В четвертом параграфе конструкторской части привести схемы форм видеogramм и отчетов.

Содержание **технологической части** пояснительной записки

3.1. Назначение и условия применения программы

В разделе "Назначение и условия применения программы" должны быть указаны назначение и функции, выполняемые программой, условия, необходимые для выполнения программы (объем оперативной памяти,

требования к составу и параметрам периферийных устройств, требования к программному обеспечению).

3.2. Руководство системного программиста

Эта часть оформляется в соответствии с ГОСТ 19.504-79 и должна содержать следующие разделы:

- характеристики программы;
- обращение к программе;
- входные и выходные данные;
- сообщения.

В разделе "Характеристики программы" приводится описание основных процедур и функций программы (временные, режим работы, средства контроля правильности выполнения и самовосстановления программы). Блок-схема связей процедур программы прилагается. Излагаются требования к аппаратным средствам и программному обеспечению ЭВМ.

3.3. Руководство пользователя

Данный раздел должен заканчиваться изложением "Руководства программиста", где описывается порядок работы пользователя с комплексом программ. Эта часть оформляется в соответствии с ГОСТ 19.504-79 и должна содержать следующие разделы:

- характеристики программы;
- обращение к программе;
- входные и выходные данные;
- сообщения.

В разделе "Характеристики программы" приводится описание основных характеристик и особенностей программы для пользователя. Описывается перечень реализации функциональных задач программы.

В разделе "Обращение к программе" должно быть приведено описание вызова программы. Подробное описание возможностей меню для пользователя по реализации функциональных задач программы.

В разделе "Сообщения" должны быть указаны тексты сообщений, выдаваемых программой в ходе ее выполнения, описаны их содержание и действия, которые необходимо предпринять по этим сообщениям.

Содержание экспериментальной части пояснительной записки

Экспериментальная часть должна содержать разработанные тесты для проверки правильности работы комплекса программ. Результаты тестирования на контрольных примерах оформляются в виде форм и отчетов.

В этой части работы должны быть представлены примеры использования комплекса программ с изложением постановки конкретной решаемой функциональной задачи, описанием конкретных вводимых исходных данных и полученных результатов с указанием значений характеристик требуемых ресурсов ЭВМ (затраты памяти, время счета и т.д.).

Содержание **заключительной части** пояснительной записки

В заключении должны содержаться краткие выводы по результатам выполненной работы и предложения по их использованию, дальнейшему развитию или модификации разработанного комплекса программ.

Список литературы

Прилагается список литературы, используемой при выполнении курсовой работы. Список литературы должен быть пронумерован и оформлен по правилам библиографии (ГОСТ 7.1 - 2003).

Содержание **приложения** пояснительной записки

Приложение содержит листинги программного пакета.

2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

Оформление расчетно-пояснительной записки осуществляется в соответствии с ГОСТ 7.32-81 ЕСКД.

Название разделов и их возможное содержание уже рассмотрены, возможные названия подразделов приведены выше.

Все разделы записки расположены на отдельных листах, имеют сквозную нумерацию. Номер ставится в правом верхнем углу. Первым листом является титульный лист (он не нумеруется). Далее размещается оглавление расчетно-пояснительной записки, которое включает наименования всех разделов, подразделов и пунктов (если они имеют названия) с указанием номеров страниц, где они начинаются.

Затем следует перечень условных обозначений, символов, единиц и терминов. Они размещаются в алфавитном порядке, причем слева приводится сокращение, а справа его расшифровка. Если сокращение используется менее трех раз, то расшифровка может быть дана в тексте.

Далее содержание курсового проекта.

ЗАДАНИЯ ПО ТЕМАМ

1 вариант. Проектирование базы данных

Разработайте структуру базы данных "Телефонные переговоры" для хранения необходимой информации.

Создайте таблицы и связи между ними. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество абонента.
 2. Телефонный номер абонента.
 3. Домашний адрес абонента. (Мемо)
 4. Телефонный код и название города, куда звонил абонент.
 5. Тариф за 1 минуту разговора с указанным городом.
 6. Дата разговора, время разговора.
 7. Продолжительность разговора.
 8. Телефон, по которому звонил абонент
-

2 вариант. Проектирование базы данных

Разработайте структуру базы данных "Библиотека" для хранения необходимой информации.

База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество читателя.
2. Домашний адрес читателя. (Мемо)
3. Телефон читателя.
4. Дата рождения читателя.
5. Номер читательского билета.
6. Автор книги, которую взял читатель.
7. Название книги, жанр, год издания и издательство.
8. Цена книги.
9. Дата выдачи книги.
10. Дата возврата книги плановая и фактическая.

3 вариант. Проектирование базы данных

Разработайте структуру базы данных "Резервирование билетов" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество пассажира.
2. Домашний адрес пассажира. (Мемо)
3. Телефон пассажира
4. Номер поезда и название маршрута.
5. Номер вагона.
6. Дата отправления.
7. Время отправления/прибытия.
8. Пункт назначения.
9. Стоимость проезда.

5 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Фотоуслуги**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
 2. Домашний адрес клиента. (Мемо)
 3. Телефон клиента
 4. Наименование фотоуслуги.
 5. Описание фотоуслуги (характеристика)
 6. Количество единиц заказа.
 7. Цена за единицу.
 8. Дата приемки заказа.
 9. Дата выдачи заказа.
-

6 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Коммунальные услуги**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество квартиросъемщика.
2. Домашний адрес квартиросъемщика. (Мемо)
3. Номер лицевого счета.
4. Количество жильцов.
5. Площадь квартиры, кв.м.
6. Вид услуги (название платежа).
7. Стоимость услуги на квадратный метр площади.
8. Стоимость услуги на 1 жильца.
9. Дата оплаты.
10. Скидка, % (льготы при оплате).

7 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Прокат товаров**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
 2. Серия и номер паспорта клиента.
 3. Домашний адрес клиента. (Мемо)
 4. Телефон клиента
 5. Наименование товара.
 6. Описание товара.
 7. Стоимость товара.
 8. Дата выдачи.
 9. Дата возврата плановая и фактическая.
 10. Стоимость проката за сутки.
-

8 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Кино**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Название фильма.
2. Жанр фильма.
3. Страна-производитель фильма.
4. Название кинотеатра.
5. Адрес кинотеатра. (Мемо)
6. Время показа.
7. Цена билета.

9 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Классный журнал**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество ученика.
2. Домашний адрес ученика. (Мемо)
3. Домашний телефон ученика.
4. Наименование предмета.
5. ФИО учителя, ведущего предмет.
6. Дата получения оценки.
7. Вид работы (контрольная, самостоятельная, ответ у доски, тест).
8. Оценка.

10* вариант. Проектирование базы данных

Разработайте структуру базы данных "**Шахматный турнир**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя шахматиста.
2. Клуб, от которого выступает шахматист.
3. Тренер шахматиста.
4. Дата проведения игры.
5. Фамилия и имя соперника.
6. Клуб, от которого выступает соперник.
7. Тренер соперника.
8. Итог игры.
9. Доп. информация о соревновании. (Мемо)

11. Разработайте структуру базы данных "Инвесторы"

Разработайте структуру базы данных "**Инвесторы**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества инвестора.
2. Банк реквизиты инвестора.
3. Сумма
4. Дата вложений.
5. Телефон.
6. Цель инвестиции. (Мемо)

7. Банковские реквизиты предприятия.
- 8.

12. Разработайте структуру базы данных "**Зарплата**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества.
2. Банк реквизиты работника.
3. Оклад
4. Дата выплаты зарплаты
5. Телефон
6. Должность
7. Расчетный месяц.
8. Домашний адрес. (Мемо)

13. Разработайте структуру базы данных "**Книги**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества автора.
2. Название книги.
3. Кол-во страниц
4. Дата издания
5. Издательство
6. Количество экземпляров
7. Цена книги
8. Адрес издательства (Мемо)

14. Разработайте структуру базы данных "**Врачи**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества врача.
2. Номер кабинета.
3. Специальность
4. Дни приема пациентов
5. Время приема
6. Домашний адрес врача (Мемо)

15. Разработайте структуру базы данных "**Поликлиника**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества пациента.
2. Адрес.
3. Телефон
4. Дата рождения
5. Шифр заболевания
6. Дата открытия больничного
7. Количество дней на больничном
8. Доп. информация о заболевании. (Мемо)

16. Разработайте структуру базы данных "**Школа**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества учителя.
2. Предмет.
3. Номер кабинета
4. Количество уроков в учебном году
5. Вид контроля (опрос, контрольная работа)
6. Количество интерактивных занятий
7. Дата рождения
8. Доп. информация об учителе. (Мемо)

17 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Интернет**" для хранения необходимой информации. Создайте таблицы и связи между ними. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество абонента.
 2. Телефонный номер абонента.
 3. Домашний адрес абонента.
 4. Телефонный код.
 5. Тариф за скорость соединения .
 6. Дата оплаты.
 7. Скорость соединения.
-

18 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Каталог в Библиотеке**" для хранения необходимой информации.

База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество читателя.
2. жанр.
3. Автор книги.
4. Название книги, год издания и издательство.
5. Тип каталога (алфавитный, предметный) .
6. Тип издания (справочник, сборник, многотомник и т.д.).
7. Количество.
8. Дата издания книги
9. О содержании книги (Мемо)

19 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Продажа билетов**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество пассажира.
2. Номер поезда и название маршрута.
3. Номер вагона.
4. Дата отправления.
5. Время отправления/прибытия.
6. Пункт назначения.
7. Стоимость проезда.
8. Доп. информация о поезде. (Мемо)

20 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Видеопрокат**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
2. Домашний адрес клиента. (Мемо)
3. Телефон клиента
4. Наименование видеофильма.
5. Количество единиц заказа.
6. Цена за единицу.
7. Дата приемки заказа.

8. Дата выдачи заказа.
-

21 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Расчет потребления электроэнергии**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество квартиросъемщика.
2. Домашний адрес квартиросъемщика. (Мемо)
3. Номер лицевого счета.
4. Показания предыдущее.
5. Показания текущее.
6. Тариф.
7. Дата оплаты.
8. Скидка, % (льготы при оплате).

22 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Лыжная база**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
 2. Серия и номер паспорта клиента. (Мемо)
 3. Домашний адрес клиента.
 4. Телефон клиента
 5. Описание товара.
 6. Стоимость за час.
 7. Дата выдачи.
 8. Время выдачи.
 9. Время возврата
 10. Стоимость
-

23 вариант. Проектирование базы данных

Разработайте структуру базы данных "**Киностудия**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Название фильма.
2. Жанр фильма.
3. Страна-производитель фильма.
4. Название киностудии.
5. Режиссёр.
6. Звукооператор.
7. Стоимость.
8. Дата выпуска фильма.
9. О содержании книги (Мемо)

24 вариант. Проектирование базы данных

24. Разработайте структуру базы данных "**Банковские расчеты**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества клиент.
2. Банк реквизиты .
3. Сумма дебета
4. Дата вложений.
5. Сумма кредита.

6. Цель платежа. (Мемо)
7. Дата операции.

25 вариант. Проектирование базы данных

25. Разработайте структуру базы данных "**Расчет премии**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества.
2. Оклад
3. Дата выплаты
4. Процент премии
5. Должность
6. Расчетный месяц.
7. Основание выплаты (Мемо)

26 вариант. Проектирование базы данных

26. Разработайте структуру базы данных "**Книжный магазин**" для хранения необходимой информации. База данных должна хранить следующую информацию:

1. Фамилия и имя отчества автора.
2. Название книги.
3. Дата издания
4. Издательство
5. Количество экземпляров в упаковке
6. Цена книги оптовая
7. Цена книги розничная
8. Описание книги. (Мемо)

Список литературы

1. Попов А.А. Программирование в среде СУБД FoxPro2.0. Построение систем обработки данных. - М.: Радио и связь, 1993. - 352 с.
2. Лебедев А.Г. Лабораторные работы по базам данных: Методические указания и задания для студентов специальности "Прикладная математика" - Рубцовск, 1995.
3. Трушин О.В. Методические указания к лабораторным работам по курсу "Организация баз данных" для подготовки бакалавров по направлению 552800 "Информатика и ВТ" - Уфа, 1996.
4. Крамм О. Системы управления базами данных dBASE II bBASE III для персональных компьютеров. - М.: Финансы и статистика, 1988. - 366 с.
5. Каррабис Д. Программирование в dBASE III PLUS. - М: Финансы и статистика, 1991. - 238 с.
6. Шнейдер Р. Access для Windows 95 (серия «Без проблем!»): Пер. с англ. - М.: БИНОМ, 1996. - 336 с.: ил.