

Проектирование реляционных баз данных

Содержание

1. Цель работы.
 2. Последовательность выполнения задания.
 3. Краткие теоретические сведения.
 - 3.1. Этапы разработки базы данных
 - 3.2. Концептуальное моделирование данных
 - анализ предметной области. Основные понятия концептуальной модели данных.
 - построение ER-диаграмм
 - 3.3. Логическое моделирование данных
 - основные понятия реляционной модели данных
 - способы построения логической модели данных.
 4. Содержание отчёта
 - Приложение1. Варианты заданий к курсовой работе.
 - Список литературы
-

1. Цель работы

Целью выполнения курсовой работы по курсу “Базы данных” является:

- изучение этапов проектирования реляционных баз данных;
 - приобретение практических навыков в разработке и реализации информационных систем;
 - приобретение навыков работы с реляционными базами данных.
-

2. Задание:

По заданному в варианте описанию предметной области разработать и реализовать проект реляционной базы данных

Последовательность выполнения задания

1. Разработка концептуальной модели данных
 1. Анализ предметной области; выделение объектов, информация о которых должна храниться в базе данных, определение их атрибутов, связей между объектами и характеристик этих связей
 2. Построение ER – диаграмм
2. Разработка логической модели данных
 1. Создание таблиц, определение типов данных в каждом поле, ограничений на установку диапазонов допустимых значений, первичных ключей.
 2. Определение внешних ключей и связей между таблицами.
3. Разработка пользовательского интерфейса. Создание запросов и представлений, необходимых пользователю при работе с базой данных. Проектирование форм для просмотра, ввода, редактирования и поиска данных в базе данных. Разработка отчётов для вывода нужных данных на печать.

4. Реализация проекта базы данных средствами выбранной СУБД.

Примечание: пункты 3 и 4 задания выполняются на лабораторном практикуме. Для их выполнения используйте методические указания к лабораторному практикуму.

3.Краткие теоретические сведения.

3.1. Этапы разработки базы данных. Многоуровневое представление данных.

Целью разработки любой базы данных является хранение и использование информации о какой-либо предметной области.

При разработке базы данных обычно выделяется несколько уровней моделирования, при помощи которых происходит переход от предметной области к реализации базы данных средствами конкретной СУБД. Можно выделить следующие уровни:

- Сама предметная область
- Модель предметной области
- Концептуальная модель данных
- Логическая модель данных
- Физическая модель данных
- Собственно база данных и приложения

Предметная область - это часть реального мира, данные о которой необходимо отразить в базе данных. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т.д. Предметная область бесконечна и содержит как существенно важные понятия и данные, так и малозначащие или вообще не значащие данные. Так, если в качестве предметной области выбрать учет товаров на складе, то понятия "накладная" и "счет-фактура" являются существенно важными понятиями, а то, что сотрудница, принимающая накладные, имеет двоих детей - это для учета товаров неважно. Однако, с точки зрения отдела кадров данные о наличии детей являются существенно важными. Таким образом, важность данных зависит от выбора предметной области.

Модель предметной области. Модель предметной области - это наши знания о ней. Знания могут быть как в виде неформальных знаний в мозгу эксперта, так и выражены формально при помощи каких-либо средств. В качестве таких средств могут выступать текстовые описания предметной области, наборы должностных инструкций, правила ведения дел в компании и т.п. Опыт показывает, что текстовый способ представления модели предметной области неэффективен. Гораздо более информативными и полезными при разработке баз данных являются описания предметной области, выполненные при помощи специализированных графических схем. Модель предметной области описывает скорее процессы, происходящие в предметной области и данные, используемые этими процессами. От того, насколько правильно смоделирована предметная область, зависит успех дальнейшей разработки приложений.

Концептуальная модель данных. На следующем, более низком уровне находится концептуальная модель данных предметной области. Концептуальная модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Примеры понятий - "сотрудник", "отдел",

"проект", "зарплата". Примеры взаимосвязей между понятиями - "сотрудник числится ровно в одном отделе", "сотрудник может выполнять несколько проектов", "над одним проектом может работать несколько сотрудников". Примеры ограничений - "возраст сотрудника не менее 16 и не более 60 лет".

Концептуальная модель данных является начальным прототипом будущей базы данных. Эта модель строится в терминах информационных единиц, но *без привязки к конкретной СУБД*. Основным средством разработки концептуальной модели данных в настоящий момент являются различные варианты **ER-диаграмм** (**Entity-Relationship, диаграммы сущность-связь**).

Логическая модель данных – это данные, представленные на языке описания данных конкретной СУБД.

Логическая модель данных включает в себя следующие составляющие:

- структура данных;
- ограничения, накладываемые на данные
- операции, производимые над данными.

Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную (объектно-ориентированную модель данных).

Наиболее распространённая модель данных, используемая большинством СУБД – **реляционная модель**. Поэтому можно считать, что логическая модель данных для нас формулируется в терминах реляционной модели данных.

Физическая модель данных. На еще более низком уровне находится физическая модель данных. Физическая модель данных описывает хранение данных средствами конкретной СУБД. Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД. При этом решения, принятые на уровне логического моделирования определяют некоторые границы, в пределах которых можно развивать физическую модель данных. Качество физической модели во многом зависит от выбора СУБД.

Собственно база данных и приложения. И, наконец, как результат предыдущих этапов появляется собственно сама база данных. База данных реализована на конкретной программно-аппаратной основе, и выбор этой основы позволяет существенно повысить скорость работы. Например, можно выбирать различные типы компьютеров, менять количество процессоров, объем оперативной памяти, дисковые подсистемы и т.п. Очень большое значение имеет также настройка СУБД в пределах выбранной программно-аппаратной платформы.

Многоуровневая архитектура (концептуальный, логический и физический уровни) позволяет обеспечить *независимость хранимых данных* от использующих их программ. При необходимости можно переписать хранимые данные на другие носители информации и (или) реорганизовать их физическую структуру, изменив лишь физическую модель данных. Можно подключить к системе любое число новых пользователей (новых приложений), дополнив, если надо, логическую модель. Указанные изменения физической и логической моделей не будут замечены существующими пользователями системы (окажутся "прозрачными" для них), так же как не будут замечены и новые пользователи. Следовательно, независимость данных обеспечивает возможность развития системы баз данных без разрушения существующих приложений.

Решения, принятые на каждом этапе моделирования и разработки базы данных, будут сказываться на дальнейших этапах. Поэтому особую роль играет принятие правильных решений *на ранних этапах проектирования*.

3.2. Концептуальное моделирование данных

. Одна из наиболее распространённых концептуальных моделей данных - модель "Сущность-Связь" (часто ее называют кратко ER-моделью). На использовании разновидностей ER-модели основано большинство современных подходов к проектированию реляционных баз данных. Основными понятиями ER-модели являются сущность, связь и атрибут.

Сущность – это класс однотипных объектов, информация о котором должна сохраняться и быть доступна. Каждая сущность должна иметь имя, выраженное существительным в единственном числе. Примерами сущностей могут быть такие классы объектов как "Поставщик", "Сотрудник", "Накладная". В ER-диаграммах сущность изображается в виде прямоугольника, содержащего имя сущности (рисунок 1).

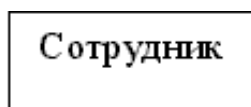


Рисунок 1 – Изображение сущности в ER-диаграмме

Экземпляр сущности – это конкретный представитель данной сущности. Например, представителем сущности "Сотрудник" может быть "Сотрудник Иванов". Экземпляры сущностей должны быть *различимы*, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

Атрибут сущности – это именованная независимая характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Примерами атрибутов сущности "Сотрудник" могут быть такие атрибуты как "Табельный номер", "Фамилия", "Имя", "Отчество", "Должность", "Зарплата" и т.п. . Атрибут изображается в виде эллипса, содержащего имя атрибута (рисунок 2).



Рисунок 2 – Изображение атрибутов сущности на ER-диаграмме

Связь – это ассоциация, поясняющая отношения между сущностями. Например, связи между сущностями могут выражаться следующими фразами – "СОТРУДНИК может *иметь* несколько ДЕТЕЙ", "каждый СОТРУДНИК обязан *числиться* ровно в одном ОТДЕЛЕ", и т.д. Графически связь изображается линией, соединяющей две сущности. Каждая связь имеет два конца и одно или два наименования, например "иметь", "принадлежать" и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Каждая связь может иметь один из следующих **типов связи по значности**:

Связь **один к одному** означает, что одному экземпляру первой сущности

соответствует один экземпляр второй сущности (рисунок 3).

Связь **один ко многим** (1:m) означает, что одному экземпляру первой сущности соответствует несколько экземпляров второй сущности, но не наоборот (рисунок 4).

Связь **многие ко многим** (m:m) означает, что одному экземпляру первой сущности соответствует несколько экземпляров второй сущности и наоборот (рисунок 5).

Каждая связь может иметь один из следующих **типов связи по членству**:

- **Обязательная** связь означает, что обе сущности зависят от наличия связи. Т.е, экземпляр одной сущности *обязан быть связан не менее чем с одним* экземпляром другой сущности, и наоборот;
- **Необязательная** связь означает, что ни одна из сущностей не зависит от наличия связи. Т.е, экземпляр одной сущности *может быть связан с одним или несколькими* экземплярами другой сущности, *а может быть и не связан* ни с одним экземпляром.
- **Возможная** связь означает, что одна из сущностей зависит от наличия связи. Т.е. один конец связи *необязательный*, а другой- *обязательный*, например: "каждый сотрудник *может* иметь несколько детей", и "каждый ребенок *обязан* принадлежать ровно одному сотруднику" (рис.6)



Рисунок 3 – Связь 1:1

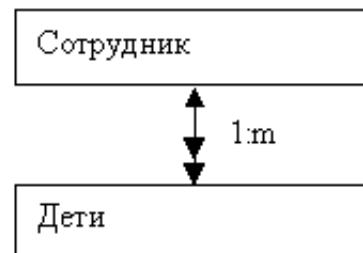


Рисунок 4 – Связь 1:m

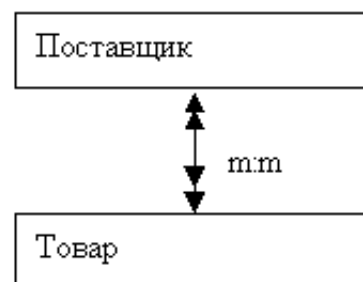


Рисунок 5 – Связь m:m

При разработке ER-моделей необходимо получить следующую информацию о предметной области:

1. Список сущностей предметной области.
2. Список атрибутов сущностей.
3. Описание взаимосвязей между сущностями.

Пример: Необходимо разработать информационную систему по заказу оптовой торговой фирмы, которая должна выполнять следующие действия:

- Хранить информацию о покупателях.
- Печатать накладные на отпущенные товары.
- Следить за наличием товаров на складе.

Анализируя эту информацию, выделим сущности:

- *Покупатель*;
- *Накладная*;
- *Товар*;
- *Склад* (если фирма имеет несколько складов).

Каждый покупатель является юридическим лицом и имеет наименование, адрес, банковские реквизиты. Каждый товар имеет наименование, цену, а также характеризуется единицами измерения. Каждая накладная имеет уникальный номер, дату выписки, список товаров с количествами и ценами, а также общую сумму накладной. Накладная выписывается с определенного склада и на определенного покупателя. Каждый склад имеет свое наименование. Исходя из этих данных, выпишем атрибуты сущностей:

- *Покупатель: Наименование покупателя, Адрес, Банковские реквизиты;*
- *Товар: Наименование товара, Цена товара, Единица измерения;*
- *Склад: Наименование склада;*
- *Накладная: Номер накладной, Дата выписки, Склад, Покупатель;*

Поскольку список товаров в накладной не может быть атрибутом, выделим его в отдельную сущность:

Список товаров в накладной: Наименование товара, Цена товара в накладной, Количество товара в накладной.

Сумма накладной равна сумме стоимостей всех товаров, входящих в накладную. Эта характеристика не является независимой, и не может считаться атрибутом

Определим связи между сущностями. Покупатели могут покупать много товаров, и товары могут продаваться многим покупателям. Если фирма имеет несколько складов, то каждый товар может храниться на нескольких складах. Покупатели покупают товары, получая при этом накладные, в которые внесены данные о

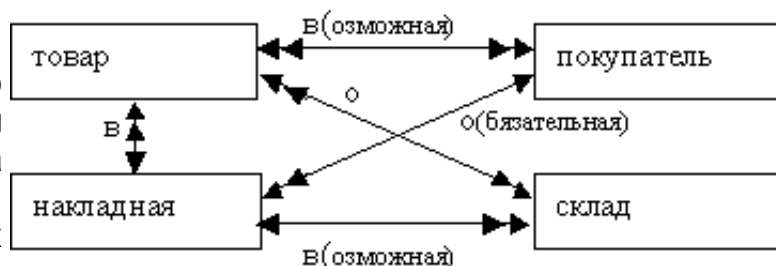


Рисунок 6 – Пример ER-диаграммы

количестве и цене купленного товара. Каждый покупатель может получить несколько накладных. Каждая накладная *обязана* выписываться на одного покупателя. Каждая накладная *обязана* содержать не менее одного товара (не бывает пустых накладных). Каждый товар может быть продан нескольким покупателям через несколько накладных. Кроме того, каждая накладная должна быть выписана с определенного склада, и с любого склада может быть выписано много накладных. Один из вариантов ER-диаграммы приведён на рисунке 6.

3.2. Логическое моделирование данных

Наиболее распространённой логической моделью на сегодняшний день является **реляционная модель**. Основы реляционной модели данных были впервые изложены в статье Е. Кодда [43] в 1970 г. Наиболее распространенная трактовка реляционной модели данных принадлежит К. Дейту. []. Согласно Дейту, реляционная модель состоит из трех частей:

- Структуры данных. (структурная часть)
- Ограничений, накладываемых на данные (целостная часть).
- Набора допустимых операций над данными (манипуляционная часть).

3.2.2. Структура реляционных данных

Единственной структурой данных, используемой в реляционной модели, является **отношение** (relation). Отношение представляет собой связь между элементами нескольких множеств атомарных однотипных значений, именуемых **доменами**. Говорят, что значения принадлежат к одному и тому же домену, если имеет смысл их

сравнение. Например, фамилию сотрудника можно сравнить с фамилией другого человека, но нет смысла сравнивать её с наименованием товара. **Атрибутом отношения** называют набор значений, принадлежащих к одному и тому же домену. Например: атрибут “Фамилия сотрудника” взят из домена фамилий.

Сотрудник

Фамилия сотрудника	Имя сотрудника	Отчество сотрудника	Должность	Оклад
Иванов	Иван	Иванович	Слесарь	1000
Петров	Петр	Петрович	Электрик	2000

Рисунок 7 – Пример реализации отношения “Сотрудник”

1. Отношения реализуются в виде двумерных таблиц (рисунок 7), обладающих следующими свойствами:
2. Каждая таблица состоит из однотипных строк и имеет уникальное имя.
3. В каждой позиции таблицы на пересечении строки и столбца может содержаться только одно значение.
4. Строки таблицы обязательно отличаются друг от друга хотя бы одним значением.
5. Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (например: даты, фамилии, наименования, целые числа или денежные суммы).
6. При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке, независимо от содержания.

Реляционная база данных представляет собой совокупность взаимосвязанных таблиц, содержащий всю информацию, которую необходимо хранить и обрабатывать. **Схемой базы данных** называют список, содержащий имена таблиц, имена атрибутов таблиц, ключевые атрибуты и внешние ключи.

3.2.3. Ограничения целостности

Во второй части реляционной модели данных определяются два ограничения, которые должны выполняться в *любой* реляционной базе данных. Это:

- Целостность сущностей.
- Целостность внешних ключей.

Определение 1. Пусть дано отношение R . Подмножество атрибутов K отношения R будем называть **потенциальным ключом**, если K обладает следующими свойствами:

1. **Свойством уникальности** - в отношении не может быть двух различных кортежей, с одинаковым значением K .
2. **Свойством минимальности** - никакое подмножество в K не обладает свойством уникальности.

Любое отношение имеет по крайней мере один потенциальный ключ. Действительно, если никакой атрибут или группа атрибутов не являются потенциальным ключом, то, в силу уникальности кортежей, все атрибуты вместе образуют потенциальный ключ.

Потенциальный ключ, состоящий из одного атрибута, называется **простым**, а состоящий из нескольких атрибутов - **составным**. Отношение может иметь несколько потенциальных ключей. Традиционно, один из потенциальных ключей объявляется **первичным**, а остальные - **альтернативными**. Т.к. потенциальные ключи служат идентификаторами объектов предметной области (т.е. предназначены для различения объектов), то значения этих идентификаторов не могут содержать неизвестные значения. Действительно, если бы идентификаторы могли содержать null-значения, то мы не могли бы дать ответ "да" или "нет" на вопрос, совпадают или нет два идентификатора.

Это определяет следующее **правило целостности сущностей**:

Правило целостности сущностей. Атрибуты, входящие в состав некоторого потенциального ключа не могут принимать неопределённых (null)-значений.

Различные объекты предметной области, информация о которых хранится в базе данных, всегда взаимосвязаны друг с другом. Такие взаимосвязи отражаются в реляционных базах данных при помощи **внешних ключей**, связывающих несколько отношений.

Т.к. внешние ключи фактически служат ссылками на строки в другом (или в том же самом) отношении, то эти ссылки не должны указывать на несуществующие объекты. Это определяет следующее **правило целостности внешних ключей**:

Правило целостности внешних ключей. Внешние ключи не должны быть несогласованными, т.е. для каждого значения внешнего ключа таблицы должно существовать соответствующее значение первичного ключа в связанной с ней таблице.

3.2.4. Способы построения логической модели данных

При разработке логической модели данных возникают вопросы: хорошо ли спроектированы отношения? Правильно ли они отражают модель предметной области, а следовательно и саму предметную область?

Рассмотрим некоторые из таких критериев, которые являются безусловно важными с точки зрения получения качественной базы данных:

- Адекватность базы данных предметной области
- Легкость разработки и сопровождения базы данных
- Скорость выполнения операций обновления данных (вставка, обновление, удаление кортежей)
- Скорость выполнения операций выборки данных

При разработке схемы базы данных необходимо выполнить следующие условия:

1. Информация в таблицах не должна повторяться (не должно быть избыточности). Избыточность приводит к проблемам при поиске и обработке данных. Поэтому желательно, чтобы каждый факт хранился только в одном месте.
2. Поля таблиц по возможности не должны принимать неопределённых (пустых) значений. Неопределённые значения могут привести к ошибкам при выполнении вычислений над теми полями таблиц, в которых они встречаются.

Очевидно, что эти условия не будут выполняться, если разместить в одной таблице разнородную информацию о нескольких объектах сразу. На рисунке 8 приведён пример универсального отношения "Сотрудник".

Сотрудник

Код	Фамилия	Возраст	Название	Должность	Оклад
-----	---------	---------	----------	-----------	-------

сотрудника	сотрудника	детей сотрудников	Отдела		
1	Иванов	10	ОГЭ	Слесарь	1000
1	Иванов	12	ОГЭ	Слесарь	1000
2	Петров	3	ОГЭ	Слесарь	1000
3	Сидоров	8	ОГМ	Электрик	3000
4	Васильев		ОГМ	Электрик	3000

Рисунок 8– Пример универсального отношения, содержащего аномалии

Поскольку в одном отделе может числиться несколько сотрудников, наименование отдела приходится повторить несколько раз. Кроме того, повторяются фамилии сотрудников, имеющих более одного ребёнка. Не все сотрудники имеют детей, поэтому атрибут “Возраст детей сотрудников” может принимать неопределённые значения. Избыточность и неопределённость данных приводят к проблемам при их вводе, обновлении и удалении. Например, при переводе сотрудника Иванова на другую должность необходимо изменить значения атрибутов “Должность” и “Оклад” дважды. При удалении данных о последнем сотруднике отдела ОГЭ будет удалена и информация о том, что такой отдел существует. Ошибки в структуре данных, приводящие к проблемам при их обработке, называют **аномалиями**. Решением проблемы будет разделение таблицы “Сотрудник” на несколько таблиц. При этом атрибуты, содержащие повторяющиеся или неопределённые значения, выносятся в отдельные таблицы, связанные с исходной таблицей при помощи внешних ключей (рисунок 10).

Сотрудник

Дети

Код сотрудника	Фамилия	Код должности	Код отдела		Код сотрудника	Возраст детей
1	Иванов	1	1		1	10
2	Петров	1	1		1	12
3	Сидоров	2	2		2	3
4	Васильев	2	2		4	8

Отдел

Должность

Код отдела	Название отдела		Код должности	Должность	Оклад
1	ОГЭ		1	Слесарь	1000
2	ОГМ		2	Электрик	3000

Рисунок 9 – Примеры нормализованных таблиц

Процесс разделения таблиц с целью улучшения их свойств называется декомпозицией, или **нормализацией**, а полученные в результате таблицы – **нормализованными**.

Заметим, что после завершения нормализации большинство таблиц содержит информацию только об одном объекте или явлении (сущности). Поэтому наиболее простым способом создания системы нормализованных таблиц является получение её из диаграммы “Сущность – связь”. Перехода от ER-диаграммы к таблицам состоит из следующих шагов:

1. Преобразование сущностей.

- Каждая простая сущность становится таблицей.
- Каждый атрибут сущности становится атрибутом (столбцом) таблицы.
- Уникальный идентификатор сущности становится ключом таблицы.
- Если в ER-диаграмме присутствовали подтипы сущности, они выносятся в отдельные таблицы.

2. Преобразование связей

- Сущности, связанные обязательной связью типа 1:1, можно объединить в одну таблицу.
- Связи типа 1:1 (возможные) и связи типа 1:m реализуются путем переноса ключевых атрибутов таблиц, соответствующих сущностям, стоящим со стороны “один” или с обязательного конца связи, в таблицы, соответствующие сущностям, стоящим со стороны “много” или с необязательного конца связи, в качестве внешних ключей (рисунок 10).



Рисунок 10

Сотрудник			Дети		
Код сотрудника	Фамилия сотрудника		Код сотрудника	Имя ребёнка	Возраст ребёнка

Рисунок 10 – Реализация связи типа 1:m

- Связи типа m:m и необязательные связи реализуются при помощи промежуточной таблицы, содержащей ключевые атрибуты связываемых таблиц в качестве внешних ключей (рисунок 11).

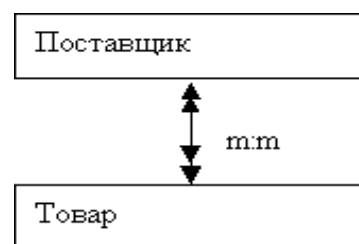


Рисунок 11

Поставщик			Товар		
Код поставщика	Наименование организации	№ счета	Код товара	Наименование товара	Единицы Измерения

Поставки

Код поставщика	Код товара	Дата поставки	Оптовая цена

Рисунок 11 – Реализация связи типа m:m

4. Содержание отчёта.

В пояснительную записку к курсовой работе необходимо включить:

1. Краткое описание предметной области
2. Изображение сущностей и их атрибутов
3. ER-диаграмму
4. Таблицы базы данных с указанием первичных и внешних ключей

Приложение 1. Варианты заданий

Вариант 1. Универмаг

База данных должна содержать сведения о следующих объектах:

- Сотрудники - фамилия, имя, отчество, адрес, дата рождения, должность, отдел, оклад.
- Отделы - наименование, зав.отделом, работники.
- Товар - наименование, группа, поставщик, наличие на складе, отдел, страховой запас, цена.
- Поставщики - название, адрес, телефон, банковский счёт, товар.

Выходные документы: Сводка о наличии товаров в отделах и на складе на конец рабочего дня.

Вариант. 2. Бюро по найму

База данных должна содержать сведения о следующих объектах:

- Клиенты - регистрационный номер, адрес, телефон, пол, образование, номер квитанции об уплате.
- Заявка - клиент, профессия(должность), минимальная зарплата, регион.
- Работодатели - Название фирмы, организационно-правовая форма, форма собственности, адрес, телефон, инспектор по кадрам, номер договора.
- Предложение - работодатель, профессия(должность), ограничения на пол, возраст, образование, регион.

Выходные документы: Для клиента и работодателя(выборочно и для всех) - список согласованных предложений-заявок.

Вариант 3. Агенство недвижимости

База данных должна содержать сведения о следующих объектах:

- Клиенты - регистрационный номер, фамилия, адрес, телефон, заявка.
- Заявки - купля/продажа, регистрационный номер, описание заявки.
- Описание заявки на покупку - регион, этажность, материал дома, количество комнат, площадь (общая и жилая), цена, условия купли.
- Описание заявки на продажу - регион, этажность, материал дома, количество

комнат , адрес, цена, условия продажи

Выходные документы: список предложений, удовлетворяющих желаниям клиента.

Вариант 4. Торгово-посредническое предприятие

База данных должна содержать сведения о следующих объектах:

- Поставщики - Наименование, организационно-правовая форма, форма собственности, адрес, банковские реквизиты, ассортимент товаров по группам.
- Товар - группа товара, наименование, срок реализации.
- Поставка - товар, объем, дата, цена, условия поставки.
- Заказ - товар, цена, объем, дата, клиент, условия выполнения.
- Клиенты - Наименование, организационно-правовая форма, форма собственности, адрес, банковские реквизиты.

Выходные документы: счет-фактура.

Вариант 5. ГИБДД (Государственная автоинспекция)

База данных должна содержать сведения о следующих объектах:

- Владельцы автотранспортных средств : Физические лица - фамилия, имя, отчество, адрес, телефон, дата регистрации, автотранспортные средства..
- Водители : Фамилия, имя, отчество, адрес, номер водительского удостоверения, дата выдачи удостоверения, категория, дата регистрации.
- Автотранспортные средства : Модель(марка), цвет, номер гос.регистрации, номер двигателя, номер кузова, номер шасси, номер технического талона, дата выдачи техталона, дата постановки на учет, владелец.
- Дорожно-транспортное происшествие : Дата, место, участники, тяжесть, описание, лицо, проводившее расследование.

Выходные документы: Список ДТП по сезонам, по месту происшествия

Вариант 6. Библиотека

База данных должна содержать сведения о следующих объектах:

- Книжный фонд - название, автор(ы), год и место издания.
- Читатели - фамилия, телефон, адрес, номер читательского билета, дата регистрации, дата перерегистрации.
- Выдача - книга(и), дата, срок сдачи, читатель.

Выходные документы: гистограмма читательского спроса.

Вариант 7. Автохозяйство

База данных должна содержать сведения о следующих объектах:

- Автомобиль - марка, номер гос.регистрации, грузоподъемность, назначение(вид перевозимого груза).
- Водители - Фамилия, классность, стаж, закрепление за автомобилем.
- Клиенты - Фамилия, адрес, телефон, заказ.
- Заказ - дата заказа, груз, адрес отправления, адрес назначения, дата и время выполнения заказа, стоимость, автомобиль, водитель.

Выходные документы: квитанция в приеме заказа.

Вариант 8. Журналистский архив

База данных должна содержать сведения о следующих объектах:

- Печатные издания (газеты, журналы) - название, место издания, издатель, дата первого выпуска,
- Содержание номеров (заголовки статей, № страницы)

Выходные документы: Подборки по темам - автор(ы), издание, дата, страницы.

Вариант 9. Реестр предприятий

База данных должна содержать сведения о следующих объектах:

- Предприятия - Наименование, организационно-правовая форма, форма собственности, адрес, банковские реквизиты,
- Ассортимент товаров: название, группа, предприятие-поставщик.
- Товар - группа товара, наименование, срок реализации, условия поставки, цена .

Выходные документы: список поставщиков, удовлетворяющих заданным требованиям.

Вариант 10. Отдел кадров

База данных должна содержать сведения о следующих объектах:

- Сотрудники - Фамилия, отдел, должность, оклад, семейное положение, отношение к военной службе,
- Продвижение по службе сотрудников (трудовая книжка) – сотрудник, должность, дата поступления, дата увольнения, причина, № приказа
- Отделы - название, штатное расписание (список должностей, количество сотрудников на каждой должности), список сотрудников.

Выходные документы:

1. Статистическая отчетность по командировкам.
2. Статистическая отчетность по фонду заработной платы.

Вариант 11. Телеателье

База данных должна содержать сведения о следующих объектах:

1. Сотрудники - фамилия, имя, отчество, должность, оклад, адрес, дата рождения.
2. Мастера - фамилия, имя, отчество, разряд, специализация, район обслуживания.
3. Абонементное обслуживание - фио клиента, адрес, телевизор, дата регистрации, номер квитанции об уплате абонементной платы за текущий период.
4. Заявки - фио клиента, адрес, телевизор, неисправность, дата регистрации, дата выполнения, мастер, номер квитанции.
5. Склад - наименование детали, остаток на складе, распределение по мастерам.

Выходные документы: Гистограммы распределения неисправностей по типам телевизоров

Вариант 12. Абоненты ГТС

База данных должна содержать сведения о следующих объектах:

1. Абоненты:
 - Физические лица - фамилия, имя, отчество, адрес, номер телефона, абонементная плата, дата уплаты, номер квитанции, льготы.
 - Юридические лица - наименование, адрес, руководитель, номер телефона, абонементная плата, дата уплаты, номер квитанции.

2. Телефоны - номер, владелец, категория, состояние.
3. Заявки на установку - дата регистрации, фамилия, адрес, льготы.
4. Заявки на ремонт - дата регистрации, фамилия, адрес, дата выполнения.

Выходные документы:

Извещение об уплате абонементной платы, извещение об оплате междугородного разговора, повторные извещения.

Вариант 13. Сеть магазинов

База данных должна содержать сведения о следующих объектах:

1. Магазины - название, адрес, персонал, наличие товара.
2. Продавцы - фамилия, адресные данные.
3. Выручка - дата, магазин, товар, продавец.
4. Товар - название

Выходные документы: распределение выручки по магазинам и продавцам по месяцам

Вариант 14. Питомник

База данных должна содержать сведения о следующих объектах:

1. Сотрудники - фамилия, имя, отчество, адрес, должность, оклад.
2. Животные - кличка, возраст, порода, описание экстерьера, родословная, участие в выставках и соревнованиях, сведения о спаривании, хозяин.
3. Щенки - возраст, родители, врач-ветеринар.
4. Клиенты - фио, адрес, животное.
5. Заявки - клиент, животное, дата заявки.

Выходные документы: Сертификат на животное, акт о выбраковке.

Вариант 15. Поликлиника

База данных должна содержать сведения о следующих объектах:

1. Медицинская карта пациента (номер страхового полиса, ФИО, адрес, телефон, участок, сведения о посещениях врачей, диагноз, назначения)
2. Врачи: ФИО, специальность, домашний адрес, телефон, расписание работы

Выходные документы:

1. Талон пациента для предъявления в страховую компанию
2. Больничный лист

Вариант 16. Справочная аптеки

База данных должна содержать сведения о следующих объектах:

1. Список лекарств и их синонимов с указанием расфасовки(упаковки), распределения по аптекам города и частоты запросов
2. Адреса аптек с указанием транспортных маршрутов и графика работы

Выходные документы: Гистограмма запросов на конкретное лекарство по месяцам

Вариант 17. Гостиница

База данных должна содержать сведения о следующих объектах:

1. Распределение номеров по этажам, с указанием общего количества мест в номере, количества свободных мест и проживающих

2. Паспортные данные гостей и распределение их по номерам с указанием времени(срока) проживания
3. Адресные данные коридорных и горничных и расписание их дежурств
4. Услуги, оказанные гостям

Выходные документы : Счет, предъявляемый при выписке гостя

Вариант 18. Пароходство

База данных должна содержать сведения о следующих объектах:

1. Судно - название, грузоподъемность, порт приписки, судовая роль (список экипажа), назначение, координаты, дата постановки на капитальный ремонт.
2. Рейс - судно, генеральный груз, порт отправления, порт назначения, порты захода, даты выхода-план, даты выхода-факт, даты прибытия-план, даты прибытия-факт, причина задержки, фрахтователь.

Выходные документы: Сводка о задержке судов в портах захода, сводка о распределении координат судов на текущий час.

Вариант 19. Журнал успеваемости

База данных должна содержать сведения о следующих объектах:

1. Студенты: ФИО, группа, специальность, факультет, курс, группа, год поступления, домашний адрес
2. Занятия: вид занятия, дисциплина, максимальное и минимальное количество баллов за занятие, преподаватель
3. Сведения об оценках каждого студента по всем дисциплинам

Выходные документы:

1. Общее количество набранных студентами баллов по дисциплинам за семестр
2. Средние баллы групп по дисциплинам

Вариант 20. Строительная фирма

База данных должна содержать сведения о следующих объектах:

1. Строящиеся объекты: адрес, название и количество требуемых стройматериалов;
2. Стройматериалы: название, остаток на складе, поставщик
3. Строительные бригады: бригадир, состав (работники), объект, за которым закреплена бригада;
4. Строители: ФИО, специальности, которыми владеет, домашний адрес

Выходные документы:.. Список необходимых материалов, отсутствующих на складе

Возможны и другие варианты тем на выбор студентов, согласованные с преподавателем, ведущим занятия.

Список литературы

1. Кузнецов С.Д. Введение в системы управления базами данных //СУБД. - 1995. - №1,2,3,4, 1996. - №1,2,3,4,5.
2. Хансен Г., Хансен Д.. Базы данных: разработка и управление: Пер. с англ. – М.: ЗАО “Издательство БИНОМ”, 1999.- 704 с.
3. Дейт К. Введение в системы баз данных //6-издание. - Киев: Диалектика, 1998. -

784 с.

4. Пушников А.Ю. Введение в системы управления базами данных. Часть 1. Реляционная модель данных: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 108 с..
 5. Пушников А.Ю. Введение в системы управления базами данных. Часть 2. Нормальные формы отношений и транзакции: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 138 с
-

[назад](#)