ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Московский государственный университет   
технологий и управления

Кафедра информационных технологий

**Базы данных**

Методические указания по выполнению   
курсовой работы

для студентов по специальности

*230102 Автоматизированные системы обработки информации и управления*

2009

Автор: Г.А. Варнакова

Рецензент: А.Н. Баранов – к.т.н., директор представительства МГУТУ в г.Рязани

**Введение**

Курсовая работа – один из наиболее активных видов самостоятельной работы студентов высших учебных заведений.

Выполнение курсовой работы предусмотрено государственным образовательным стандартом и учебным планом.

Целью данной курсовой работы является получение студентами практического навыка самостоятельной разработки баз данных.

В курсовой работе студенты решают следующие основные задачи:

1. Построение информационной модели базы данных.
2. Создание физической модели базы данных.

В процессе разработки необходимо пройти все основные этапы создания информационной модели базы данных. Для создания физической модели базы данных можно пользоваться любыми средствами СУБД.

Самостоятельная работа над темой способствует получению глубоких и прочных знаний по базам данных.

# 1 Основные требования к курсовой работе

Задание на курсовую работу выдается студентам индивидуально преподавателем дисциплины - руководителем курсовой работы в период лабораторно-экзаменационной зимней сессии 3 курса на занятиях по дисциплине Базы данных. Курсовая работа в полном объеме должна быть выполнена самостоятельно до летней лабораторно-экзаменационной сессии 3 курса.

Примерныйперечень тематики работ приведен ниже.

Курсовую работу студент выполняет непосредственно под руководством преподавателя данной дисциплины.

*Обязанности руководителя курсовой работы:*

* выдача задания на курсовую работу;
* оказание помощи студенту в разработке плана работы, в выборе языка программирования;
* рекомендация основной литературы, справочных материалов и других источников по теме;
* проведение консультаций;
* проверка выполнения работы и пояснительной записки.

Курсовые работы должны быть ориентированы на разработку базы данных.

Курсовая работа состоит из двух частей: теоретической и практической. Теоретическая часть представляется пояснительной запиской, а практическая – разработанной базой данных.

Работа должна содержать две главы. Первая глава должна быть посвящена теории разработки баз данных. Вторая глава должна быть посвящена разработке информационной модели базы данных и структуры базы данных.

В процессе выполнения курсовой работы необходимо создать отношения, составляющие базу данных (БД), включая определение типов полей, индексирование и заполнение таблиц записями. Количество записей в таблицах не ограничивается, однако их должно быть столько, чтобы наглядно иллюстрировать правильность выполнения работы (минимум 15 записей).

Структура созданной БД, хранимые отношения и записи, типы полей, ключи, взаимосвязи должны быть описаны в пояснительной записке.

Курсовая работа должна быть выполнена и защищена в срок, установленный учебным планом.

# 2 План пояснительной записки

Пояснительная записка к курсовой работе должна содержать следующие разделы:

Введение

Глава 1. Теоретические основы разработки баз данных

Глава 2. Проектирование БД на основе концептуальных требований

2.1 Разработка информационной и физической модели базы данных

2.2 Контрольный пример

Заключение

Список литературы

Приложения

**3 Содержание пояснительной записки**

***3.1 Введение***

Во введении указывается значимость, степень новизны и характер темы. Объем введения от 1 до 3 страниц.

***3.2 Разработка информационной и физической модели базы данных***

В основу проектирования БД должны быть положены представления конечных пользователей конкретной организации — концептуальные требования к системе. Именно конечный пользователь в своей работе принимает решения с учетом получаемой в результате доступа к базе данных информации. От оперативности и качества этой информации будет зависеть эффективность работы организации. Данные, помещаемые в базу данных, также предоставляет конечный пользователь.

При рассмотрении требований конечных пользователей необходимо принимать во внимание следующее:

* + база данных должна удовлетворять актуальным информационным потребностям организации. Получаемая информация должна по структуре и содержанию соответствовать решаемым задачам;
  + база данных должна обеспечивать получение требуемых данных за приемлемое время, то есть отвечать заданным требованиям производительности;
  + база данных должна удовлетворять выявленным и вновь возникающим требованиям конечных пользователей;
  + база данных должна легко расширяться при реорганизации и расширении предметной области;
  + база данных должна не зависеть (или мало зависеть) от количества помещаемых в нее данных;
  + база данных должна легко изменяться при изменении программной и аппаратной среды;
  + загруженные в базу данных корректные данные должны оставаться корректными;
  + база данных должна содержать только достоверные данные. Достоверность данных должна обеспечиваться как при вводе новых данных, так и при редактировании уже имеющихся данных;
  + данные до включения в базу данных должны проверяться на достоверность;
  + доступ к данным, размещаемым в базе данных, должны иметь только лица с соответствующими полномочиями.

В результате анализа поставленной заказчиком задачи и обработки требований конечных пользователей составляется концептуальная модель. Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среде хранения.

При разработке логической модели базы данных, прежде всего необходимо решить, какая модель данных наиболее подходит для отображения конкретной концептуальной модели предметной области. Коммерческие системы управления базами данных поддерживают одну из известных моделей данных или некоторую их комбинацию. Почти что все популярные системы для персональных компьютеров поддерживают реляционную модель данных.

Отображение концептуальной модели данных на реляционную модель производится относительно просто. Каждый прямоугольник концептуальной модели отображается в одно отношение, которое отражает представление пользователя в удобном для него табличном формате. Простота отображения связана с тем, что при разработке концептуальной модели использовался реляционный подход. Построение реляционной базы данных основано на нормализации отношений.

Нормализация отношений – это процесс построения оптимальной структуры таблиц и связей в реляционной базе данных. В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные.

Здесь также решается вопрос ликвидации избыточности информации. То есть концептуальные требования, используемые несколькими структурными подразделениями, сводятся в одну таблицу с одновременным добавлением ключей для перехода к другим таблицам (для других структурных подразделений). Таким образом добиваются существенного сокращения требуемых объемов памяти. На этом этапе также решается вопрос о том, какие таблицы будут справочниками, т.е. информация в этих таблицах не изменяется или изменяется очень медленно.

Следует иметь в виду, что чрезмерное увеличение количества таблиц приводит к потере общей идеи создания БД и сама БД становится трудной для понимания и управления.

Оптимальное количество таблиц для БД объема предприятия должна быть не более 50. Всего существует 5 нормальных форм. На практике, как правило, при создании приложения БД в объеме предприятия используется первые 3 нормальные формы.

Рассмотрим последовательность проектирования базы данных, которые должны обеспечить необходимую независимость данных и выполнение эксплуатационных требований (пожеланий пользователей).

*Этап 1. Определение сущностей*

*Этап 2. Определение взаимосвязей между сущностями*

Определить для включенных в модель сущностей взаимосвязи в соответствии с ранее описанными рекомендациями. Представить на этом этапе информационную модель задачи в виде диаграммы «Сущность – Связь».

*Этап* *3*. *Задание первичных и альтернативных ключей, определение атрибутов сущностей*

Для каждой сущности нужно определить атрибуты, которые будут храниться в базе данных. При этом необходимо учитывать тот факт, что при переходе от логической к физической модели данных, может произойти усечение числа объектов. На самом деле, как правило, значительное число данных, необходимых пользователю, может быть достаточно легко подсчитано в момент вывода информации. В то же время, в связи с изменением алгоритмов расчета или исходных величин, некоторые расчетные показатели приходится записывать в базу данных, чтобы гарантированно обеспечить фиксацию их значений.

Атрибуты, включаемые в состав БД для рассматриваемой модели, нужно привести в таблице под названием «Атрибуты и первичные ключи сущностей информационной модели». Информационную модель после третьего этапа проектирования привести на схеме «Взаимосвязи между атрибутами сущностей».

*Этап 4. Приведение модели к требуемому уровню нормальной формы*

В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные. Введение нормализации отношений при разработке информационной модели обеспечивает минимальный объем физической, то есть записанной на каком-либо носителе БД и ее максимальное быстродействие, что впрямую отражается на качестве функционирования информационной системы. Нормализация информационной модели выполняется в несколько этапов.

*Данные, представленные в виде двумерной таблицы, являются первой нормальной формой реляционной модели данных.*

Первый этап нормализации заключается в образовании двумерной таблицы, содержащей все необходимые атрибуты информационной модели, и в выделении ключевых атрибутов. Очевидно, что полученная весьма внушительная таблица будет содержать очень разнородную информацию. В этом случае будут наблюдаться аномалии включения, обновления и удаления данных, так как при выполнении этих действий придется уделить внимание данным (вводить или заботиться о том, чтобы они не были стерты), которые не имеют к текущим действиям никакого отношения.

Привести таблицы, в которых описаны сущности и атрибуты первой нормальной формы.

*Отношение задано во второй нормальной форме, если оно является отношением в первой нормальной форме и каждый атрибут, не являющийся первичным атрибутом в этом отношении, полностью зависит от любого возможного ключа этого отношения.*

Если все возможные ключи отношения содержат по одному атрибуту, то это отношение задано во второй нормальной форме, так как в этом случае все атрибуты, не являющиеся первичными, полностью зависят от возможных ключей. Если ключи состоят более чем из одного атрибута, отношение, заданное в первой нормальной форме, может не быть отношением во второй нормальной форме. Приведение отношений ко второй нормальной форме заключается в обеспечении полной функциональной зависимости всех атрибутов от ключа за счет разбиения таблицы на несколько, в которых все имеющиеся атрибуты будут иметь полную функциональную зависимость от ключа этой таблицы. В процессе приведения модели ко второй нормальной форме в основном исключаются аномалии дублирования данных.

Обосновать приведение своей базы данных ко второй нормальной форме. Привести таблицы, в которых описаны сущности и атрибуты второй нормальной формы.

*Отношение задано в третьей нормальной форме, если оно задано во второй нормальной форме и каждый атрибут этого отношения, не являющийся первичным, не транзитивно зависит от каждого возможного ключа этого отношения.*

Транзитивная зависимость выявляет дублирование данных в одном отношении.

Преобразование в третью нормальную форму происходит за счет разделения исходного отношения на два.

Основные правила, которым нужно следовать при проектировании базы данных:

* + Исключать повторяющиеся группы — для каждого набора связанных атрибутов создать отдельную таблицу и снабдить ее первичным ключом. Выполнение этого правила автоматически приведет ко второй нормальной форме.
  + Исключать избыточные данные — если атрибут зависит только от части составного ключа, переместить атрибут в отдельную таблицу. Это правило помогает избежать потери одних данных при удалении каких-то других. Везде, где возможно использование идентификаторов вместо описания, выносить в отдельную таблицу список идентификаторов с пояснениями к ним.
  + Исключать столбцы, которые не зависят от ключа — если атрибуты не вносят свою лепту в описание ключа, переместить их в отдельную таблицу.

В общем случае рекомендуется использовать вместо естественных атрибутов коды в следующих случаях:

* + В предметной области может наблюдаться синомия, то есть естественный атрибут отношения не обладает свойством уникальности. Например, среди сотрудников фирмы могут быть однофамильцы или даже полные тезки. В этом случае решить проблему помогает уникальный табельный номер.
  + Если отношение участвует во многих связях, то для их отображения создается несколько таблиц, в каждой из которых повторяется идентификатор отношения. Для того чтобы не использовать во всех таблицах длинный естественный атрибут объекта, можно применять более короткий код. Это также будет способствовать повышению быстродействия вашей системы.
  + Если естественный атрибут может изменяться во времени (например, фамилия), то это может вызвать очень большие сложности при эксплуатации системы. Использование неизменяемого кода (табельного номера) позволит избежать этих сложностей.

Обосновать приведение своей базы данных к третьей нормальной форме. Привести измененные таблицы, в которых описаны сущности и атрибуты третьей нормальной формы. Привести диаграмму взаимосвязи между атрибутами сущностей после нормализации модели.

*Этап 5.Физическое описание модели*

На этом этапе нужно составить проекты таблиц, которые будут в дальнейшем реализовываться в конкретной СУБД. Структуру разработанной базы данных отразить в таблице.

На этапе физического проектирования обеспечить безошибочность и точность информации, хранящейся в базе данных. Это называется обеспечением целостности базы данных.

Обеспечением целостности базы данных называется система мер, направленных на поддержание правильности данных в базе в любой момент времени.

В СУБД целостность данных обеспечивается набором специальных предложений, называемых ограничениями целостности.

Ограничения целостности — это набор определенных правил, которые устанавливают допустимость данных и связей между ними.

Ограничения целостности в большинстве случаев определяются особенностями предметной области. Ограничения целостности могут относиться к разным объектам базы данных: атрибутам (полям), записям, отношениям, связям между ними и т. п. Для полей могут использоваться следующие виды ограничений:

* Тип и формат поля автоматически допускают ввод только данных определенного типа. Выбор типа поля Date в формате ДД.ММ.ГГ позволит пользователю ввести только шесть чисел. При этом первая пара цифр не сможет превысить в лучшем случае значения 31, а вторая — 12.
* Задание диапазона значений, как правило, используется для числовых полей. Диапазон допустимых значений может быть ограничен с двух сторон (закрытый диапазон), а может с какой-то одной: верхней или нижней (открытый диапазон).
* Недопустимость пустого поля позволяет избежать появления в БД «ничейных» записей, в которых пропущены какие-либо обязательные атрибуты.
* Задание списка значений позволяет избежать излишнего разнообразия данных, если его можно ограничить. Например, для указания типа кузова мы можем ограничить фантазию пользователя только общепринятыми названиями: Седан, кабриолет и т. д.
* Проверка на уникальность значения какого-то поля позволяет избежать записей-дубликатов. Вряд ли будет удобно в справочнике клиентов иметь несколько записей для одного и того же лица.

На пятом этапе также предусматриваются меры по обеспечению ссылочной целостности, то есть установление между таблицами не противоречивых взаимосвязей. Установление не противоречивых взаимосвязей и обеспечение достоверности в данных в любой момент времени является главной и самой трудоемкойзадачей.

Для реализации ограничений целостности, имеющих отношение к записи, таблицам или связям между ними, в СУБД могут использоватся триггеры.

Пример разработки базы данных приведен в приложении А.

#### 3.3 Контрольный пример

Контрольный пример должен содержать экспериментальную часть. Привести содержимое базы данных.

***3.4 Заключение***

Содержит краткую оценку полученных результатов, сжато оценивается уровень разработки, ее гибкость, широта, модульность.

***3.5 Список литературы***

Содержит перечень книг, журнальных статей, справочников, методических указаний и пособий и других печатных материалов, используемых при выполнении курсовой работы.

***3.6 Приложения***

Включает спецификации к схемам, распечатки программ, образцы входных и выходных документов, распечатки экранных форм, иллюстрации и т.д.

4 Оформление пояснительной записки

Пояснительная записка к курсовой работе должна содержать не менее 40 страниц машинописного текста формата А4 с рисунками и схемами.

Пояснительная записка должна содержать в определенной последовательности следующие разделы: титульный лист, задание на выполнение курсовой работы. Далее в пояснительную записку входит: содержание (оглавление), основная текстовая часть, список использованной литературы, приложение в виде графиков, таблиц, листингов программ и т.п., включение которых в основную текстовую часть записки нецелесообразно.

Записку можно подготовить на компьютере и распечатать на принтере на одной стороне стандартного листа формата А4.

Оформлять пояснительную записку следует в соответствии с ГОСТ

В список литературы заносят все использованные источники. Литература приводится в алфавитном порядке фамилий авторов или в порядке цитирования.

В пояснительной записке недопустимо использование произвольных сокращений слов и не общепринятых слов, иностранного происхождения. В случае большого количества используемых в тексте сокращений перед текстовой частью работы помещается список сокращений с их расшифровкой.

###### 5 Защита курсовой работы

Курсовая работа выполняется студентом в сроки, установленные учебным планом и заданием. Во время экзаменационной сессии преподаватель не принимает курсовые работы. В процессе выполнения работы студент получает консультации у преподавателя, устраняет допущенные ошибки. Наиболее типичными ошибками являются:

* отклонение от требований к структуре работы;
* отсутствие логики в изложении материала;
* отсутствие убедительных доказательств, обоснований и выводов;
* нерациональный алгоритм решения задачи;
* несамостоятельно выполненная работа;
* отклонение от плана при оформлении пояснительной записки;
* отсутствие заголовка, вводной части, заключения, списка литературы, оглавления, приложения, а также нечеткие формулировки, грамматические ошибки, небрежность оформления;
* нарушение установленного порядка и сроков сдачи курсовой работы.

Все перечисленные недостатки снижают ценность курсовой работы и могут служить основанием для снижения оценки, или для не допуска к защите, или неудовлетворительной оценке при защите работы.

Не позднее, чем за неделю до защиты пояснительная записка в прошнурованном виде предъявляется на проверку руководителю курсовой работы.

Преподаватель проверяет соответствие содержания пояснительной записки заданию, отмечает недостатки работы, пишет рецензию на курсовую работу и принимает решение о направлении курсовой работы на защиту или доработку.

На защите студенту предоставляется для доклада и демонстрации базы данных 10 – 15 минут.

В докладе следует изложить постановку задачи,ее актуальность и новизну, главные этапы и результаты работы и четко сформулировать выводы. Далее студенту задаются вопросы. Вопросы по содержанию и выполнению работы могут задавать преподаватель и студенты группы, присутствующие на защите курсовой работы. Ответы на вопросы студентов также учитываются при выставлении итоговой оценки.

Оценка "отлично" выставляется, если грамотно описана разработка базы данных, грамотно и полно изложен материал в пояснительной записке, имеются элементы творчества, поиска, делаются самостоятельно обоснованные выводы.

Оценка "хорошо" выставляется за работу, которая выполнена на высоком теоретическом и практическом уровне, полностью и всесторонне освещены все вопросы темы, но отсутствует творческий подход в разрешении рассматриваемой проблемы, недостаточно сформированы собственные предложения для разрешения проблемы.

Оценка "удовлетворительно" выставляется за те работы, в которых правильно освещены основные вопросы темы, но выбран для реализации нерациональный алгоритм решения задачи и вместе с тем автор не проявил умения логически излагать и обосновать содержание работы, не выполнил самостоятельный анализ разработки. При защите даны неполные ответы по содержанию работы.

Оценка "неудовлетворительно" выставляется в тех случаях, когда информационная модель базы данных не разработана, программный продукт не работает или в нем имеются ошибки, и студент не может ответить на вопросы по содержанию работы. В этом случае студенту выдается новая тема, и назначаются новые сроки выполнения курсовой работы.

###### 6 Задание на выполнение курсовой работы

Тема курсовой работы должна звучать так:

***Разработка базы данных «…».***

В кавычках должно быть приведено название базы данных в соответствии с вариантом.

Например:

***Разработка базы данных «Туристическое агентство»***.

Пояснительная записка должна содержать две части:

1. Провести анализ теоретической части разрабатываемой темы (глава 1).

2. Разработать соответствующую базу данных, имеющую минимум заданных полей и занести в БД не менее 15 записей, приближенного к достоверному содержания. В задании указан минимум концептуальных требований, который может быть расширен по усмотрению студента.

**Вариант задания должен соответствовать порядковому номеру студента в списке группы.**

**Варианты заданий**

**Задания к главе 1:**

1. Основные этапы развития машинной обработки данных
2. Модели и структуры данных
3. Физические модели баз данных
4. Модели и этапы проектирования баз данных
5. Проектирование реляционной базы данных
6. Распределенная обработка данных
7. Транзакции и целостность баз данных
8. Управление базами данных в СУБД
9. Направления развития концепций и систем обработки данных
10. Теорико-графофые модели данных
11. Реляционная модель данных
12. Инфологическое моделирование
13. Принципы поддержки целостности в реляционной модели данных
14. Модели транзакций
15. Перспективы развития БД и СУБД
16. Роль баз данных в электронном бизнесе
17. Принципы управления распределенной информацией
18. Технология клиент-сервер в базах данных и программное обеспечение промежуточного слоя
19. Неоднородные федеративные базы данных и мультибазы данных
20. Перспективы управления распределенной информацией
21. Объектно-ориентированные базы данных
22. Языки баз данных и управления информацией
23. Перспективные модели баз данных и управления информацией
24. Защита информации в БД
25. Программирование в СУБД
26. Создание базы данных и управление ею
27. Работа с формами в СУБД
28. Работа с отчетами в СУБД
29. Управление проектом и создание приложений в СУБД
30. Запросы к базе данных
31. Меню приложения
32. Создание справочной системы приложения
33. Защита баз данных
34. Администрирование баз данных
35. Средства автоматизации проектирования БД
36. Физические модели баз данных

**Задания к главе 2:**

1. Туристическое агентство.  
   Концептуальные требования:  
   Клиент: ФИО, паспортные данные, дата рождения, место рождения, адрес проживания, гражданство, телефон, место работы, адрес работы, гражданство, номер загранпаспорта; страна пребывания, гостиница, даты тура, наименование тура, стоимость путевки дата продажи путевки, ФИО менеджера, дополнительные сведения.
2. Кадровое агентство.  
   Концептуальные требования:  
   ФИО клиента, паспортные данные, дата рождения, адрес клиента, телефон клиента, образование, специальность, стаж работы, предыдущее место работы, дополнительные навыки, ФИО менеджера, наименование заказчика, телефон заказчика, адрес заказчика.
3. Касса продажи авиабилетов.  
   Концептуальные требования:  
   Рейс самолета, конечный пункт, тип самолета, вместимость самолета, дата и время вылета, номер места, стоимость билета, ФИО пассажира, документ пассажира, адрес пассажира, информация о рейсе.
4. Учет кадров на предприятии.  
   Концептуальные требования:  
   ФИО работника, дата рождения, место рождения, паспортные данные, страховое свидетельство, ИНН, адрес, общий стаж работы, дата оформления на предприятие, подразделение, должность, оклад, сведения о детях.
5. Учет призывников в военкомате.  
   Концептуальные требования:  
   ФИО призывника, дата рождения, место рождения, паспортные данные, семейное положение, место учебы, место работы, специальность, состояние здоровья, ФИО родных, сведения о близких родственниках.
6. Учет товаров в хозяйственном магазине.  
   Концептуальные требования:  
   код товара, наименование товара, цена товара, единица измерения количества поступившего товара, количество товара, дата изготовления, срок годности, дата поступления товара в торговый зал, дата продажи, количества проданного товара, остаток, наименование поставщика товара, адрес поставщика товара.
7. Учет видеофильмов в пунктах проката.  
   Концептуальные требования:  
   инвентарный номер кассеты или диска, название фильма, режиссер, жанр, дата выдачи, срок проката, цена, дата возврата, ФИО клиента, номер паспорта, адрес, телефон.
8. Услуги в парикмахерской.  
   Концептуальные требования:  
   ФИО парикмахера, дата, время, наименование услуги, стоимость услуги.
9. Служба проката бытовой техники.  
   Концептуальные требования:  
   наименование техники, тип техники, дата выдачи в прокат, стоимость проката, дата выдачи, срок проката, цена, дата возврата, ФИО клиента, номер паспорта, адрес, телефон.
10. Библиотечный фонд.  
    Концептуальные требования:  
    тематика, название книги, автор, издательство, год издания, цена, количество, инвентарный номер книги, отдел, содержание.
11. Услуги в автосервисе.   
    Концептуальные требования:  
    ФИО клиента, номер паспорта, адрес, телефон, марка автомобиля, номер автомобиля, услуги, которые необходимо выполнить, стоимость услуг.
12. Касса продажи билетов на междугородные автобусные рейсы.  
    Концептуальные требования:  
    наименование маршрута, номер маршрута, дата отправления, время отправления, номер места, стоимость билета, дата продажи.
13. Купля-продажа недвижимости  
    Концептуальные требования:  
    сведения о клиенте, сведения о продаваемом объекте недвижимости, сведения о заказе на покупку недвижимости.
14. Аптека. Учет медикаментов.  
    Концептуальные требования:  
    наименование медикамента, дата выпуска, фирма, тип, поставщик, цена, количество (в наличии, продано, остаток), дата продажи.
15. Учет офисной техники в фирме.  
    Концептуальные требования:  
    наименование техники, тип, технические характеристики, дата поступления, дата списания, помещение.
16. Автосалон. Продажа автомобилей.  
    Концептуальные требования:  
    сведения об автомобиле, сведения о покупателе, сведения о продавце.
17. Общежитие.  
    Концептуальные требования:  
    сведения о проживающих, сведения о комнатах, сведения об оборудовании комнат, сведения об оплате.
18. Гостиница.  
    Концептуальные требования:  
    сведения о проживающих, сведения о комнатах, сведения об оборудовании комнат, сведения об оплате.
19. Молочный склад.  
    Концептуальные требования:  
    номер накладной, поставщик, потребитель, наименование товара, количество, цена, сумма.
20. Автострахование.  
    Концептуальные требования:  
    страхователь, собственник, допущенные к управлению лица, водительское удостоверение, срок страхования, сведения об автомобиле.
21. Коммерческая палатка.  
    Концептуальные требования:  
    тип товара, наименование товара, закупочная цена, количество закупленного товара, количество проданного товара, дата продажи, остаток товара, отпускная цена, поставщик, дата поступления товара.
22. Ателье.  
    Концептуальные требования:  
    ФИО мастера, дата приема заказа, дата выполнения заказа,, наименование услуги, стоимость услуги.
23. Мастерская по ремонту обуви.  
    Концептуальные требования:  
    ФИО мастера, дата приема заказа, дата выполнения заказа, наименование услуги, стоимость услуги.
24. Мебельный магазин.  
    Концептуальные требования:  
    изготовитель, адрес, телефон, наименование товара, цена, количество, дата продажи выручка.
25. Учет телефонных переговоров.  
    Концептуальные требования:  
    ФИО абонента, № телефона абонента, куда звонил, длительность, тариф, № тел.адресата, дата разговора, счет.
26. Учет результатов участников соревнования.  
    Концептуальные требования:  
    ФИО участника, вид соревнования, место участника, ср.балл, организация, город.
27. Автозаправочная станция.  
    Концептуальные требования:  
    ФИО оператора, дата заправки, время заправки, № колонки, вид топлива, кол-во топлива, цена за 1л, сумма оплаты.
28. Мастерская металлоремонта.  
    Концептуальные требования:  
    ФИО мастера, наим.работы, стоимость, дата приема, №квитанции, ФИО клиента, дата возврата.
29. Страхование.  
    Концептуальные требования:  
    страхователь, вид страхования, срок страхования, сумма страхования.
30. Хранение сыров на складе готовой продукции.  
    Концептуальные требования:  
    *н*аименование, категория, способ хранения, срок хранения, форма выпуска, производитель, поступление, отгрузка, остаток.
31. Учет членов партии в организации.  
    Концептуальные требования:  
    ФИО, дата вступления в организацию, общественная нагрузка, выполняемые поручения, дата рождения, место рождения, паспортные данные, адрес, место работы, общий стаж работы.
32. Поэты России.  
    Концептуальные требования:  
    ФИО поэта, дата рождения, крупные произведения, дата смерти.
33. Композиторы России.  
    Концептуальные требования:  
    ФИО композит, дата рождения, оперы, балеты, прочие произвед., дата смерти, примечание.
34. Великие ученые.  
    Концептуальные требования:  
    ФИО, дата рождения, специализация, страна работы, дата смерти, премии.
35. Кинорежиссеры.  
    Концептуальные требования:  
    ФИО, дата рождения, наим.фильма, студия, дата выпуска, тематика.
36. Страны мира.  
     Концептуальные требования:  
    наименование страны, наименование материка, столица, площадь, население, реки, озера, моря.

**ПРИЛОЖЕНИЕ А**

**Задание:**

Разработка базы данных «Учет заказов в фотоцентре».

Концептуальные требования:

* ФИО клиента, адрес, вид работ, формат, вид бумаги, особые отметки, количество.

**1 Разработка базы данных**

Структура таблиц определяет эффективность программ, обрабатывающих эти таблицы, и всего приложения в целом. Реляционная модель базы данных основывается на математических принципах теории реляционных наборов. Для простоты манипулирования данными при создании таких наборов рекомендуется нормализовать эти данные.

Нормализация – это процесс исключения избыточной информации: сложные данные разбиваются на отдельные таблицы, между которыми могут быть установлены отношения. Для определения структуры каждой таблицы необходимо выполнить анализ функциональных зависимостей. В результате количество необходимых таблиц определяется числом функциональных зависимостей. Формально нормализация данных обеспечена, если набор таблиц удовлетворяет первым трем правилам, которые называют нормальными формами. Для приведения модели базы данных к требуемому уровню нормальной формы, а это является основой построения реляционной базы данных, процесс проектирования должен пройти несколько этапов.

Применимо к нашей задаче на первом этапе проектирования базы данных выделим следующие сущности (объект, информация о котором хранится в базе данных):

* клиент;
* заказ.

Второй этап заключается в определении взаимосвязей между сущностями согласно требованиям к базе данных. В соответствии с этим диаграмма «Сущность-связь» будет выглядеть следующим образом:

клиент

заказ

Рисунок 1.1 – Диаграмма «Сущность-связь»

В данной диаграмме используется отношение «одна запись со многими». Так как в нашем случае клиент может за одно посещение сделать несколько заказов, т.е. отдать не одну, а несколько пленок, заказав для них различный вид работы.

С третьего этапа начинается приведение модели к требуемому уровню нормальной формы.

Отношение находится в первой нормальной форме, если все его атрибуты являются простыми, т.е. имеют единственное значение.

Условия первой нормальной формы:

* должны отсутствовать повторяющиеся записи;
* должны отсутствовать повторяющиеся атрибуты;
* каждый атрибут должен быть неделим.

Для каждой сущности определим атрибуты, которые будут храниться в базе данных.

Сущность «Клиент» имеет следующие атрибуты:

* фамилия;
* имя;
* отчество;
* адрес.

Сущность «Заказ»:

* дата приема;
* дата выдачи;
* вид работы;
* вид бумаги;
* формат;
* количество;
* уникальный ключ ответственного лица;
* цена.

Отношение находится во второй нормальной форме, если оно удовлетворяет следующим условиям:

* выполняется условие первой нормальной формы;
* первичный ключ однозначно определяет запись;
* все поля записи зависят от первичного ключа;
* первичный ключ имеет минимальную форму (отсутствует избыточность).

В соответствии с этим приведем таблицу отношений атрибутов и первичных ключей.

Таблица 1.1 - Атрибуты и первичные ключи

|  |  |  |
| --- | --- | --- |
| Сущность | Первичный ключ | Атрибуты |
| Клиент | Уникальный ключ клиента | Уникальный ключ клиента  Фамилия  Имя  Отчество  Адрес |
| Заказ | Уникальный ключ заказа | Уникальный ключ заказа  Уникальный ключ клиента  Дата приема  Дата выдачи  Вид работы  Вид бумаги  Формат  Количество комплектов  Счет |

Информационная модель после данного этапа проектирования будет иметь следующий вид:

Клиент

Ун\_кл\_кл

Фамилия

Имя

Отчество

Адрес

Заказ

Ун\_кл\_зак

Ун\_кл\_кл

Дата\_пр

Дата\_выд

Вид\_раб

Вид\_бум

Формат

Колич\_ком

Счет

Рисунок 1.2 – Информационная модель

Четвертый этап заключается в приведении модели к требуемому уровню нормальной формы.

Условия третей нормальной формы:

* должны выполняться условия второй нормальной формы;
* внутри каждой сущности должны отсутствовать транзитивные связи.

С учетом изложенного в нашей модели необходимо изменить список атрибутов сущности «Заказ» и добавить новые сущности: вид работы, счет, формат, вид бумаги.

Приведем таблицу распределения сущностей и атрибутов по новым образовавшимся сущностям.

Таблица 1.2 - Сущности и атрибуты

|  |  |  |
| --- | --- | --- |
| Сущность | Первичный ключ | Атрибуты |
| Клиент | Ун\_кл\_кл | Ун\_кл\_кл  Фамилия  Имя  Отчество  Адрес |
| Заказ | Ун\_кл\_зак | Ун\_кл\_зак  Уникальный ключ клиента  Дата приема  Дата выдачи  Вид работы  Вид бумаги  Формат  Количество комплектов |
| Счет | Ун\_кл\_сч | Ун\_кл\_сч  Уникальный ключ клиента  Сумма |
| Формат | Ун\_кл\_форм | Уникальный ключ формата  Наименование  Цена |
| Вид работы | Ун\_кл\_раб | Уникальный ключ работы  Наименование  Цена |
| Вид бумаги | Ун\_кл\_бум | Уникальный ключ бумаги  Наименование  Цена |

Информационная модель после четвертого этапа проектирования будет иметь 3 НФ (нормальную форму) и выглядеть следующим образом:

Клиент

Ун\_кл\_кл

Фамилия

Имя

Отчество

Адрес

Счет

Ун\_кл\_сч

Ун\_кл\_кл

Сумма

Заказ

Ун\_кл\_зак

Ун\_кл\_кл

Дата\_пр

Дата\_выд

Ун\_кл\_раб

Ун\_кл\_бум

Ун\_кл\_форм

Колич\_ком

Формат

Ун\_кл\_форм

Наименов

Цена

Вид работы

Ун\_кл\_раб

Наименов

Цена

Вид бумаги

Ун\_кл\_бум

Наименов

Цена

Рисунок 1.3 – Диаграмма «Сущность-связь»

Пятый этап состоит в физическом описании модели. На этом этапе создаются проекты таблиц (структуры), которые будут в дальнейшем реализоваться в конкретной системе управления базами данных на машинных носителях информации.

База данных состоит из 5 таблиц. Структура базы данных приведена ниже.

Таблица 1.3 - Таблица «Клиенты» (Klient.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_кл | N | 5 | Уникальный ключ клиента |
| Фамилия | С | 15 | Фамилия |
| Имя | С | 15 | Имя |
| Отчество | С | 15 | Отчество |

Таблица 1.4 - Таблица «Заказ» (Zakaz.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_кл | N | 5 | Уникальный ключ клиента |
| Ун\_кл\_зак | С | 5 | Уникальный ключ заказа |
| Дата\_пр | D | 8 | Дата приема заказа |
| Дата\_выд | D | 8 | Дата выдачи заказа |
| Ун\_кл\_раб | С | 3 | Уникальный ключ вида работы |
| Ун\_кл\_форм | С | 3 | Уникальный ключ формата |
| Ун\_кл\_бум | С | 3 | Уникальный ключ бумаги |
| Колич | N | 2 | Количество фото |

Таблица 1.5 - Таблица «Вид работы» (Vid\_rab.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_раб | С | 3 | Уникальный ключ вида работы |
| Наимен | С | 8 | Наименование вида работы |
| Цена | N | 10,2 | Цена |

Таблица 1.6 - Таблица «Вид бумаги» (Vid\_bum.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_бум | С | 3 | Уникальный ключ вида бумаги |
| Наимен | С | 8 | Наименование вида бумаги |
| Цена | N | 10,2 | Цена |

Таблица 1.7 - Таблица «Формат» (Format.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_форм | С | 3 | Уникальный ключ формата |
| Наимен | С | 8 | Наименование формата |
| Цена | N | 10,2 | Цена |

Таблица 1.8 - Таблица «Счет» (Schet.dbf)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Содержание |
| Ун\_кл\_сч | N | 5 | Уникальный ключ счета |
| Ун\_кл\_кл | С | 8 | Уникальный ключ клиента |
| Сумма | N | 10,2 | Сумма |

Формирование счета и занесение его в таблицу Schet.dbf.

производится по формуле:

 (1)

где  - сумма заказа одного клиента;

 - количество фотографий;

 - цена одной фотографии с учетом формата и вида бумаги.

Общая сумма заказов вычисляется по формуле:

 (2)

###### Рекомендуемая литература

1. Агальцов В.П. Базы данных. М: «МИР»,2002
2. Алан Р. Саймон. Стратегические технологии баз данных. - М.: Финансы и статистика", 2001
3. Ахаян. Р.Эффективная работа с СУБД. - С-Пб.: ПИТЕР, 1997
4. Базы данных. Под ред. Проф. А.Д. Хомонена. - С-Пб.: КОРОНАпринт,2000
5. Гаврилова Т.А., Хорошевский В.Ф.. Базы знаний интеллектуальных систем. – ПИТЕР, 2001
6. Голицина О.Л. и др. Базы данных. – «ФОРУМ», 2007
7. Гэри Хансен. Базы данных. Разработка и управление. - М.: БИНОМ,1999
8. Информатика. Учебник для ВУЗов. Под ред. Проф. Н.В.Макаровой, М: «Финансы и статистка»,2003
9. Канноли, Томас, Бегг и др.Базы данных: проектирование реализация и сопровождение. Теория и практика. Пер. с англ.– М.: Издательский дом «Вильямс», 2001
10. Карпова Т. Базы данных: модели, разработка, реализация. – СПб. ПИТЕР, 2001
11. Лебедев А.Н. Visual FoxPro 9. – М.:НТ Пресс, 2005
12. Манахем Базиян и др. Использование  
     Visual FoxPro 6.0 .Издательский дом «Вильямс», 2001
13. Омельченко Л. Самоучитель Visual FoxPro 9.0. – «БХВ – Петербург», 2005
14. Попов А.А. Программирование в среде СУБД FoxPro 2.0 в DOS и 2.5 и в Windows. - М.: Радио и связь, 2000
15. Фрост Р. Проектирование и разработка баз данных. –М.:НТ Пресс, 2007