

Министерство общего и профессионального образования
Российской Федерации

Государственное образовательное учреждение высшего
профессионального образования “Санкт-Петербургский
государственный архитектурно-строительный университет”

Кафедра прикладной математики и информатики

ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Методические указания

по выполнению лабораторной работы по курсу “Базы данных” для
студентов специальности 230400 “Прикладная математика”

Санкт-Петербург
2009

УДК 681.3

Проектирование реляционных баз данных: Метод. указания по выполнению лабораторной работы по курсу “Базы данных” для студентов специальности 230400 “Прикладная математика”/ СПбГАСУ; Сост.: В. В. Шацков СПб., 2009. 21 с.

Методические указания предназначены для помощи студентам при выполнении лабораторной работы по проектированию реляционных баз данных с использованием системы управления базами данных MS Access. В них дано описание основных этапов проектирования базы данных и рассмотрен пример создания базы данных для конкретной предметной области. В заключение приведены задания на лабораторную работу.

Ил. 6. Библиогр.: 2 назв.

Рецензент д-р физ.-мат.наук, профессор Б.Г. Вагер (СПбГАСУ)

Проектирование реляционных баз данных

Составитель: Шацков Владимир Валентинович

Введение

На сегодняшний день реляционные базы уверенно являются неотъемлемой частью большинства информационных систем. В этой области прикладного программирования стремительно развиваются новые технологии, платформы реализации и среды разработки приложений. Тем не менее остается классическая часть и общепризнанный подход к процессу проектирования, разработки и сопровождения реляционной базы данных, разбитый на последовательные этапы, которые обычно приходится реализовывать. Данная работа имеет целью усвоение и закрепление теоретических знаний в области проектирования реляционных баз данных и изучения особенностей конкретных систем управления базами данных. Проектирование баз данных включает в себя следующие этапы:

1. Концептуальное проектирование, включающее в себя системный анализ рассматриваемой предметной области поставленной задачи и составление ее словесного описания. Этот этап в данной работе отсутствует, то есть каждый студент получает словесное описание объектов системной области и связей, существующих между ними, в качестве исходного задания. На данном этапе необходимо представить концептуальную модель в виде ER-диаграммы
2. Логическое проектирование базы данных. На этом этапе осуществляется выбор СУБД (системы управления базами данных), с помощью которой база данных будет реализована. В лабораторной работе предполагается использование СУБД MS Access. Поскольку эта СУБД реализует реляционную модель данных, то необходимо осуществить переход от ER-модели к реляционной модели и затем нормализовать полученные отношения
3. Создание таблиц, входящих в реляционную модель с учетом существующих между ними связей. В данной лабораторной работе требуется составить для каждой создаваемой таблицы оператор CREATE TABLE языка SQL

В отчете по данной работе студенты должны представить ER-диаграмму, созданную на основе выданного им в качестве задания словесного описания некоторой конкретной предметной области, схему реляционной модели, полученную из ER-модели, тексты

операторов SQL CREATE TABLE для создания каждой таблицы реляционной модели.

Построение ER-модели

ER-модель (модель “сущность-связь”) – широко используемый инструмент разработки баз данных, представляющий собой формализованное графическое представление предметной области в виде прямоугольников, линий и специальных символов.

Преимуществом ER-модели заключается в том, что она понятна и разработчикам баз данных, и заказчикам, то есть будущим ее пользователям, и потому все возможные изменения в структуре данных согласуются между ними именно на уровне этой модели. ER-модель никак не связана ни с какой конкретной СУБД, этот выбор осуществляется впоследствии. В основе ER-модели лежат понятия “сущность”(“entity”) и “связь” (“relation”). Не существует удовлетворительного формального описания понятия “сущность”. Можно сказать, что это – совокупность значимых объектов, существующих в рамках предметной области. Например, при построении системы учета успеваемости студентов придется оперировать сущностями “студенты”, “преподаватели”, ”дисциплины”, а в случае создания банковской системы – сущностями “клиенты” и “счета” Каждая сущность включает набор атрибутов, представляющих собой именованные свойства сущности. Например, сущность “студенты” может иметь атрибуты:

1. Номер группы
2. Фамилия
3. Имя
4. Номер зачетной книжки”,
5. Дата рождения

Сущность – совокупность объектов, а каждый объект этой совокупности – экземпляр сущности. Экземпляры должны быть различимы, то есть не должно быть двух экземпляров с одинаковыми наборами атрибутов. Каждая сущность имеет набор атрибутов, уникальный для каждого экземпляра сущности, называемый ключом. Например, для сущности “студенты” ключ состоит из одного атрибута “Номер зачетной книжки”, так как точно не существует двух и более студентов с одинаковым значением этого атрибута. Между сущностями могут быть установлены связи, определяющие, как сущности соотносятся или взаимодействуют между собой. Связь

может существовать между двумя сущностями (бинарная связь) или между сущностью и ею же самой (рекурсивная связь). Наличие связи означает, что одна из них имеет атрибут или набор атрибутов, составляющий ключ другой сущности. Например, сущность “студенты” и сущность “учебные группы” связаны между собой атрибутом “номер группы”, который является ключом сущности “учебные группы” и присутствует в наборе атрибутов сущности “студенты”.

Связи бывают трех типов:

1. Связь “один к одному”. При этой связи один элемент первой сущности связан не более чем с одним экземпляром второй сущности. Этот тип связи встречается сравнительно редко
2. Связь “один ко многим” В этом случае один экземпляр первой сущности связан с несколькими экземплярами второй сущности, но один экземпляр второй сущности связан не более чем с одним экземпляром второй сущности. Первая сущность считается в данной связи основной, вторая – подчиненной. Примером может служить рассмотренная выше связь между сущностями “учебные группы” и “студенты”. Сущность “учебные группы” является здесь основной, сущность “студенты” – подчиненной, так как одна группа состоит из нескольких студентов, и, следовательно, одному экземпляру сущности “группы” соответствует несколько экземпляров сущности “студенты”, принадлежащих этой группе.
3. Связь “многие ко многим”. В этом случае каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, а каждый экземпляр второй сущности – с несколькими экземплярами первой. В качестве примера можно привести связь между сущностями “преподаватели” и “дисциплины” системы учета успеваемости. Эта связь имеет тип “многие ко многим”, так как один преподаватель может вести несколько учебных дисциплин, а одна дисциплина вестись несколькими преподавателями.

Важной характеристикой связи является ее обязательность или модальность. Обязательность связи означает, что любому экземпляру сущности обязательно соответствует хотя бы один экземпляр другой сущности, состоящей с первой в обязательной связи. В противном случае связь является необязательной. Связь может иметь неодинаковую модальность с разных концов. Например, каждая

учебная группа имеет куратора из числа преподавателей, однако не каждый преподаватель является куратором.

Следовательно, для построения ER-модели необходимо сначала выявить сущности, характерные для заданной предметной области и определить для каждой сущности набор атрибутов, имеющих значение исходя из поставленной задачи. Затем необходимо определить все имеющиеся связи между сущностями и характеристики этих связей. После этого строится ER-диаграмма, то есть собственно графическое изображение модели с использованием одной из существующих нотаций. Для простых случаев ER-диаграмму можно рисовать вручную, для более сложных можно воспользоваться специализированным программным обеспечением, известным, как CASE-средства. В данной работе мы будем придерживаться нотации POWER DESIGNER. Сущности в ER-диаграмме изображаются в виде прямоугольников, внутри которых перечисляются имена атрибутов. Ключевые атрибуты подчеркиваются. Имя сущности указывается в прямоугольнике первым и отделяется от списка атрибутов чертой. Связи между сущностями изображаются в виде линий, соединяющих прямоугольники соответствующих сущностей. Для сущности, находящейся со стороны “многие”, линия связи должна заканчиваться значком из трех расходящихся линий. Необязательность связи обозначается значком, имеющим на конце меньшую окружность.

В качестве примера создадим несколько упрощенную модель системы учета успеваемости студентов. Оценки, получаемые студентами по изучаемым дисциплинам, заносятся в специальную таблицу, где указывается номер зачетной книжки студента, код дисциплины, табельный номер преподавателя, дата получения оценки и собственно оценка. Студент не может в один и тот же день два раза сдавать одну и ту же дисциплину. Студенты разбиты по группам. Каждая группа имеет свой номер и куратора из числа преподавателей. Каждый преподаватель может вести несколько дисциплин, каждая учебная дисциплина может вестись несколькими преподавателями. Таким образом, можно выделить следующие сущности:

1. Студенты;
2. Преподаватели;
3. Учебные дисциплины;

4. Группы;
5. Оценки;

Атрибутами сущности “студенты” будут:

1. номер зачетной книжки;
2. фамилия,
3. имя,
4. номер группы.

Ключом здесь будет номер зачетной книжки.

Сущность “преподаватели” будет иметь атрибуты:

1. табельный номер;
2. фамилия;
3. имя;
4. отчество.

Ключом сущности, естественно, является табельный номер преподавателя.

У сущности “учебные дисциплины” будет два атрибута – код дисциплины и наименование дисциплины, при этом первый атрибут - ключевой.

Сущность “группы” следующие атрибуты:

1. номер группы;
2. табельный номер куратора группы;

Здесь номер группы – ключевой атрибут.

И, наконец, сущность “оценки” имеет следующие атрибуты:

1. номер зачетной книжки студента;
2. код дисциплины;
3. код преподавателя;
4. дату получения оценки;
5. оценка.

Напомним, что ключом сущности является атрибут или набор атрибутов, значения которых являются уникальными для каждого экземпляра сущности. В данном случае ключом сущности “оценки” будет набор из трех атрибутов – номер зачетной книжки, код дисциплины и дата получения оценки, так как предполагается, что студент в один день по некоторой дисциплине может получить только одну оценку.

Теперь рассмотрим связи, имеющиеся между сущностями.

1. Так как в одной группе числятся несколько студентов, между сущностями “группы” и “студенты” существует связь “один ко многим”. Эта связь обязательна с обеих сторон

2. Так как у каждой группы есть куратор из числа преподавателей, между сущностями “группы” и “преподаватели” существует связь “один к одному”, необязательная со стороны “преподаватели”, так как не каждый преподаватель является куратором какой-либо группы.
3. Поскольку каждый преподаватель может вести несколько дисциплин, а одна дисциплина вестись несколькими преподавателями, то между сущностями “преподаватели” и “дисциплины” имеется связь “многие ко многим”. Поскольку любую дисциплину преподается хотя бы одним преподавателем, а любой преподаватель преподает хотя бы одну дисциплину, то связь обязательна с обеих сторон
4. Между сущностью “студенты” и сущностью “оценки” существует связь “один ко многим”. Каждый студент неоднократно получает оценки по изучаемым дисциплинам. Возможно, что студент не успел получить еще ни одной оценки, следовательно, связь является необязательной со стороны “студенты”.
5. Между сущностью “преподаватели” и сущностью “оценки” существует связь “один ко многим”. Каждый преподаватель выставляет оценки по своим дисциплинам. Существует вероятность, что преподаватель не поставил ни одной оценки, потому что только что был принят на работу, следовательно, связь является необязательной со стороны “преподаватели”.
6. Между сущностью “дисциплины” и сущностью “оценки” существует связь “один ко многим”, так как по каждой дисциплине может быть выставлено много оценок. Существует вероятность, что по некоторой дисциплине не было еще выставлено ни одной оценки, следовательно, связь является необязательной со стороны “дисциплины”.

На рисунке 1 показана ER-диаграмма, построенная для данной задачи, с учетом существующих связей. В кружках при каждой линии, обозначающей связь между сущностями, указан номер этой связи в соответствии с приведенной выше нумерацией. Имя каждой сущности отделено от имен атрибутов жирной чертой, ключевые атрибуты каждой сущности подчеркнуты.



Рис.1

Переход к реляционной модели.

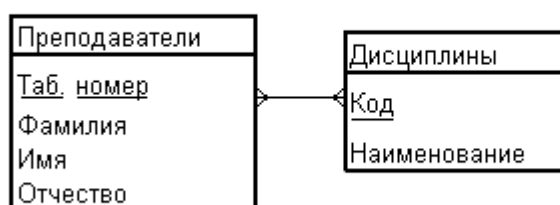
На следующем этапе проектирования необходимо преобразовать ER-модель в реляционную. Основной структурой реляционной модели является отношение (relation), графической интерпретацией которого служит таблица. Каждое отношение состоит из некоторого ограниченного числа кортежей, а содержание каждого кортежа определяется набором атрибутов отношения. Каждый атрибут имеет определенный тип, значение которого берется из определенного домена. Кортежам отношения соответствуют строки таблицы, количество столбцов таблицы равно количеству атрибутов отношения, а тип величины, находящейся в соответствующем столбце определяется типом соответствующего атрибута. Отношения могут быть связаны между собой посредством набора атрибутов, одинаково содержащихся в обоих отношениях. Связи между отношениями в реляционной модели, в отличие от ER-модели, могут иметь только тип “один ко многим”, то есть одно отношение всегда будет основным, а второе – подчиненным и одному кортежу основного отношения могут соответствовать несколько кортежей подчиненного отношения. Данное соответствие означает, что у обоих кортежей значения набора атрибутов, по которому связаны отношения, совпадают. Обычно у основного отношения данный набор атрибутов является первичным ключом, и, следовательно, уникальным для каждого

кортежа. У второго отношения данный набор атрибутов называется внешним ключом для данной связи. Кортежей с одинаковым значением внешнего ключа может быть сколько угодно, но не может существовать кортежей со значением внешнего ключа, которому не соответствовал какой-либо первичный ключ основного отношения. Связь “один к одному” рассматривается, как частный случай связи “один ко многим”. Связи “многие ко многим” в реляционной модели быть не может. Чтобы преобразовать ER-модель в реляционную модель необходимо выполнить следующие действия.

1. Поставить каждой сущности ER-модели в соответствие отношение реляционной модели, при этом каждому атрибуту сущности ставится в соответствие атрибут отношения реляционной модели. Ключ сущности становится первичным ключом соответствующего отношения (PRIMARY KEY). При этом имена сущностей и отношений, равно, как и атрибутов, могут не совпадать. Желательно при указании имен отношений и атрибутов реляционной модели использовать латиницу, поскольку эти имена чаще всего являются идентификаторами в некотором языке программирования.
2. В каждое отношение, соответствующее подчиненной сущности, добавляется набор атрибутов, соответствующий ключу основной сущности, если, конечно, он там уже не присутствовал. В любом случае этот набор атрибутов становится внешним ключом в подчиненном отношении (FOREIGN KEY).
3. При обязательном характере связи у атрибутов, соответствующих внешнему ключу, устанавливается свойство отсутствия неопределенных значений (NOT NULL)
4. Если в ER-модели имеются связи “многие ко многим”, то их надо преобразовать в связи “один ко многим”, поскольку связи “многие ко многим” в реляционной модели не допускаются. Для этого в реляционную модель добавляется связующее отношение, атрибуты которого соответствуют атрибутам первичных ключей обоих отношений, участвующих в связи “многие ко многим”. Связующее отношение будет находиться в связи “один ко многим” с каждым из этих отношений. В рассматриваемом примере связь “один ко многим” имеют сущности “преподаватели” и “дисциплины”. В реляционной

модели вводится связующее отношение R, атрибутами которого будут “ID_Subject” и “Tab_num”. Первый атрибут соответствует первичному ключу сущности “дисциплины”, а второй – первичному ключу сущности “преподаватели”. Это отношение будет иметь связь “один ко многим” к отношениям, соответствующим сущностям “преподаватели” и “студенты”.

ER-модель



Добавление
связующего
отношения R

Реляционная модель

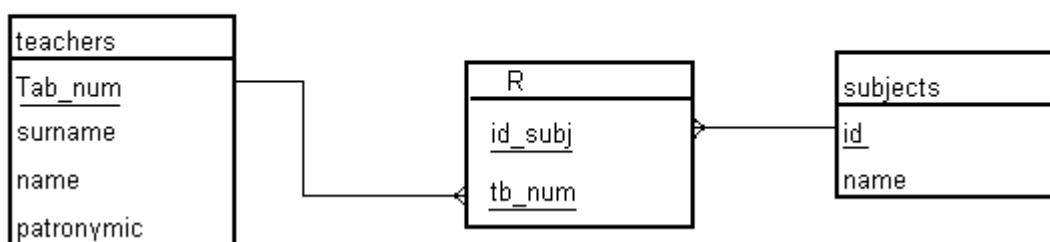


Рис.2

В принципе, уже на уровне ER-модели можно провести ее нормализацию. В данной работе требуется, чтобы модель удовлетворяла условиям третьей нормальной формы.

Напомним, что отношение находится в первой нормальной форме, если значения всех его атрибутов атомарны, то есть представляют собой простые, а не составные типы данных. В нашем примере это условие выполняется для всех сущностей. Значениями атрибутов всех сущностей являются только числа, строки или даты. Массивов, структур или подтаблиц среди них нет.

Сущность удовлетворяет условиям второй нормальной формы, если каждый неключевой атрибут зависит от всего первичного ключа, но не зависит от какой-либо части первичного ключа. Из определения следует, что условие второй нормальной формы может быть нарушено только у составного первичного ключа. В приведенном примере составной первичный ключ имеется у сущности “оценки”. Не ключевым атрибутом здесь является только один атрибут “оценка”. Его значение определяется всей

совокупностью ключевых атрибутов, поэтому сущность условиям второй нормальной формы отношение “оценки” удовлетворяет. Сущность находится в третьей нормальной форме, если оно находится во второй нормальной форме и каждый не ключевой атрибут зависит только от первичного ключа, но не зависит какого-либо не ключевого атрибута. Все сущности нашего примера удовлетворяют условиям третьей нормальной формы. Полученную реляционную модель также изобразим в графической форме. Каждому отношению будет соответствовать прямоугольник, в который будут вписаны имена атрибутов отношения и их типы. Набор возможных типов определяется официальным стандартом языка SQL, однако у конкретных СУБД обычно имеются расхождения со стандартом. В нашей модели атрибуты могут принимать текстовые, числовые значения, или представляют собой какую-либо дату. Рассмотрим эти типы подробнее.

1. Тип даты вообще не определен в стандарте SQL и в каждой СУБД определяется по-своему. В настоящей работе для создания таблиц используется СУБД MS ACCESS, в которой для указания даты и времени имеется тип Date.
2. Для представления строчных атрибутов используется стандартный тип Character(n) или, сокращенно, CHAR(n), поддерживаемый любой СУБД. Здесь n – максимальная длина атрибута в символах. Длину каждого строчного атрибута необходимо согласовывать с заказчиком, но в данной работе ее можно выбирать произвольно, по усмотрению разработчика.
3. Для числовых атрибутов можно использовать тип integer, если атрибут может принимать только целочисленные значения, или тип float, если атрибут может иметь дробную часть. Типы decimal(m,n) и numeric(m,n) в MS ACCESS не поддерживаются.

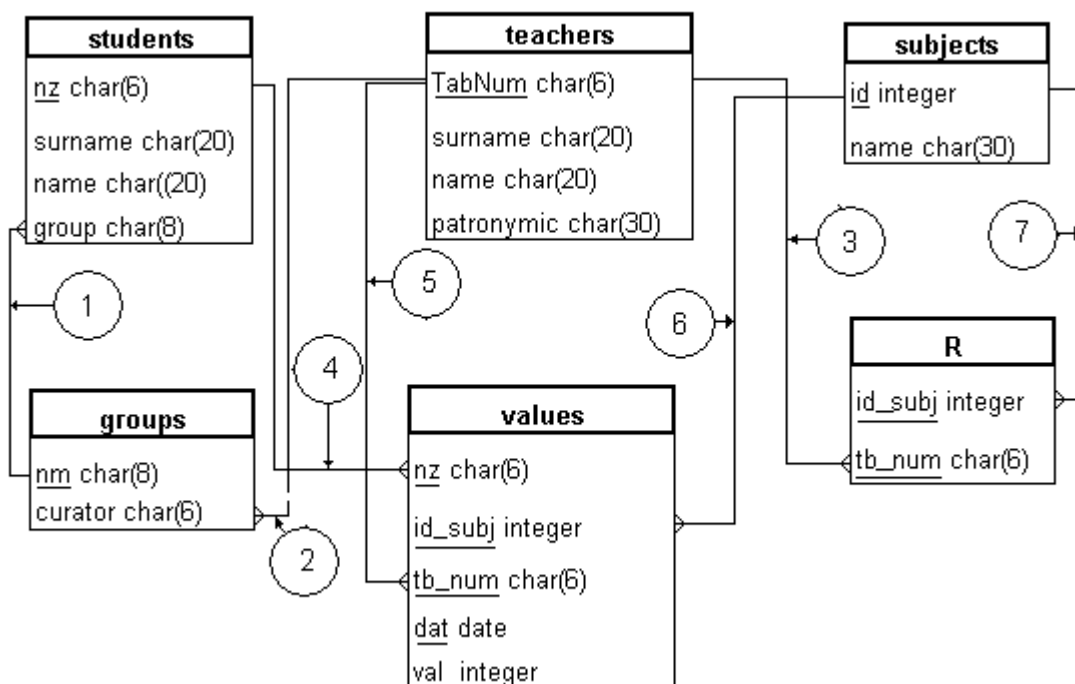


Рис.3

На рисунке 3 приведена реляционная модель для нашего примера. В ней восемь отношений, причем семь из них соответствуют семи сущностям ER-модели, а восьмое отношение R появилось в результате преобразования связи “многие ко многим” между сущностями “преподаватели” и “дисциплины” к двум связям “один ко многим” между отношениями subjects и R, а также teachers и R. На рисунке 3 линии, соответствующие связям “один ко многим” четко соединяют именно те атрибуты двух отношений, которые участвуют в этой связи. Отношение, участвующее в связи со стороны “многие”, является подчиненным, а это означает, что атрибут подчиненного отношения может брать значения только из списка значений соответствующего ему по связи атрибута главного отношения. По сравнению с ER-моделью изменились связи, которые имели в ней номера 1 и 4 (см. рис.1).

1. Связь между отношениями teachers и groups является связью один к одному, поскольку только один из преподавателей может быть куратором группы. Однако эта связь рассматривается, как частный случай связи один ко многим, причем отношение groups является подчиненным, так как значение атрибута curator (табельный номер преподавателя) берется из табельных номеров отношения teachers. Таким образом, атрибут curator подчиненного отношения groups является внешним ключом для первичного ключа TabNum главного отношения teachers.

2. В ER-модели связь 3 между сущностями “преподаватели” и “дисциплины” имела тип “многие ко многим”. Поскольку в реляционной модели могут существовать только связи “один ко многим”, было введено новое отношение R и вместо одной связи тип “многие ко многим” появились две связи “один ко многим” с номерами 3 и 7.

Создание таблиц базы данных

Та часть языка SQL, которая служит для создания объектов базы данных, называется DDL (Data Definition Language). Основными объектами любой базы данных являются таблицы, соответствующие отношениям реляционной модели. Для создания таблицы употребляется оператор CREATE TABLE.

Синтаксис команды имеет следующий вид:

```
CREATE TABLE <имя таблицы> (<имя столбца> <тип данных столбца>[<ограничения столбца>] [,(<определение столбца> <тип данных>[<ограничения столбца>]..)[<определения ограничений таблицы>])
```

Имена столбцов и их типы для каждой таблицы приведены на рисунке 3. На столбцы в данной работе могут быть наложены следующие ограничения:

1. Ограничение NOT NULL, означающее, что значение столбца должно быть обязательно задано при вводе новой записи. По умолчанию обычно действует ограничение NULL, по которому задавать значение для данного столбца не обязательно.
2. Ограничение UNIQUE, означающее, что в данной таблице не должно быть двух и более записей с одинаковым значением этого столбца.
3. Ограничение первичного ключа PRIMARY KEY. Это ограничение для столбца возможно в том случае, если первичный ключ таблицы состоит из одного данного столбца. В противном случае столбцы, входящие в первичный ключ, указываются в конце всего оператора CREATE TABLE в ограничениях для всей таблицы. То, что столбец является первичным ключом, означает также и то, что его значение должно быть обязательно задано (ограничение NOT NULL) и

это значение должно быть уникальным для данной таблицы (ограничение UNIQUE).

4. Ограничение внешнего ключа REFERENCES <ИМЯ ГЛАВНОЙ ТАБЛИЦЫ>. накладывается на столбец в том случае, если он является внешним ключом в связи “один ко многим” с другой, главной в данной связи таблицей. В этом случае любое значение данного столбца должно соответствовать некоторому значению первичного ключа главной таблицы.

Последние два ограничения называются ограничениями ссылочной целостности базы данных и служат для обеспечения корректности данных, содержащихся в таблицах.

Ограничения всей таблицы конце оператора CREATE TABLE предназначены для определения ограничений, накладываемых на несколько столбцов сразу. Например, это происходит в случае наличия составных первичного или внешнего ключа.

Ниже приведены операторы CREATE TABLE для создания каждой таблицы рассматриваемого нашего примера.

1. Создание таблицы Subjects.

```
Create table subjects(id integer primary key, name char(30) );
```

2. Создание таблицы teachers.

```
Create table teachers(tabnum char(6) primary key, surname char(20) not null, name char(20), patronymic char(30));
```

3. Создание таблицы R, связывающей таблицы subjects и teachers

```
Create table R(id_subj integer references subjects, tb_num char(6) references teachers, constraint pk primary key(id_subj, tb_num));
```

Здесь столбцы id_subj и tb_num заданы, как внешние ключи для главных таблиц subjects и teachers соответственно, а совокупность столбцов id_subj и tb_num объявлена, как первичный ключ таблицы, причем ограничение первичного ключа имеет имя pk.

4. Создание таблицы groups ().

Create table groups(nm char(8) primary key, curator char(6) references teachers);

Здесь первый столбец таблицы является первичным ключом, второй – внешним ключом, ссылающимся на таблицу students, то есть значения этого столбца берутся из значений первичного ключа таблицы students(столбец nz). Третий столбец является внешним ключом, ссылающимся на таблицу teachers.

5. Создание таблицы students ()

Create table students(nz char(6) primary key, surname char(20) not null, name char(20), group char(8) references groups)

При создании таблицы для столбца surname (фамилия) указано ограничение not null. Это означает, что при вводе новой записи значение для этого столбца должно быть указано обязательно.

6. Создание таблицы values ().

Create table values(nz char(6) references students, id_subj integer references subjects, tb_num char(6) references teachers, dat date, val integer not null, constraint pmk primary key (nz, id_subj, tb_num, dat));

Здесь первые три столбца являются внешними ключами, ссылающимися на первичные ключи родительских таблиц students, subjects и teachers, а первые четыре столбца в совокупности образуют составной первичный ключ, имеющий имя pmk.

Для дальнейшей работы необходимо в программе MS ACCESS создать новую базу данных. Назовем ее LAB_VD.mdb. Для этого необходимо выполнить пункт основного меню Файл/Создать/Новая база данных. Для создания запроса в окне базы данных необходимо во вкладке “Запросы”, предназначенной для создания запросов к данным, выбрать пункт “Создать”, затем пункт “Создание запроса в режиме конструктора” и в появившемся окне “Новый запрос” выбрать первый пункт “Конструктор”(рис.4).

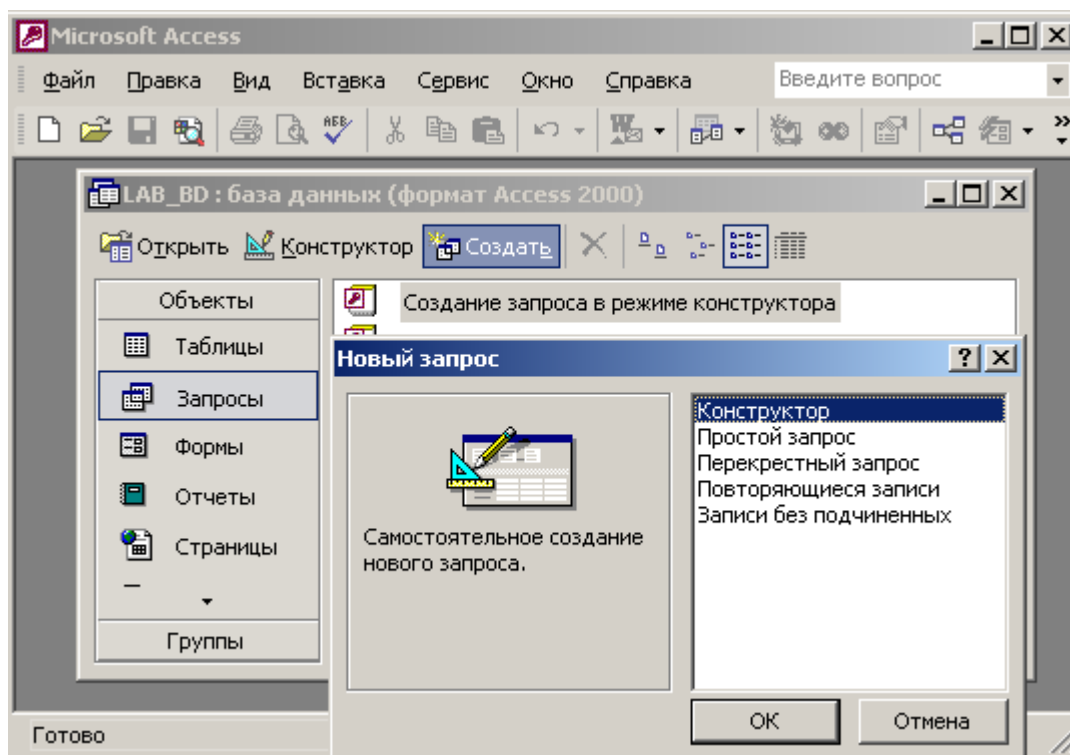


Рис. 4

Затем надо закрыть появившееся окно “Добавление таблицы” и, выполнив пункт основного меню “Вид/Режим SQL”, перейти к режиму непосредственного ввода текста запроса на языке SQL(Рис.5)

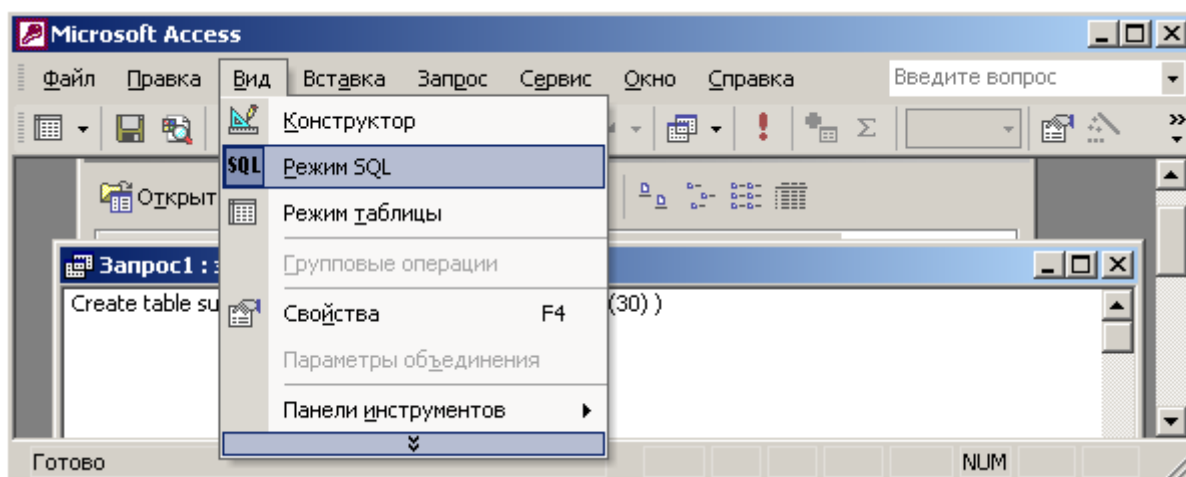


Рис.5

Далее в окне запроса необходимо ввести его текст и затем выполнить, выбрав пункт основного меню “Запрос/Запуск”. Запросы (их в данной работе будет шесть) желательно сохранять, давая им имена соответственно таблицам, которые они создают. Например, первый запрос, создающий таблицу subjects, желательно сохранить под именем create_subjects.

В результате выполнения шести запросов в базе данных появятся шесть таблиц, между которыми будут установлены связи, соответствующие ограничениям внешнего ключа, указанных в запросах. В завершение отчета по лабораторной работе необходимо привести схему данных, полученную выполнением пункта основного меню Сервис/Схема данных (рис.6)

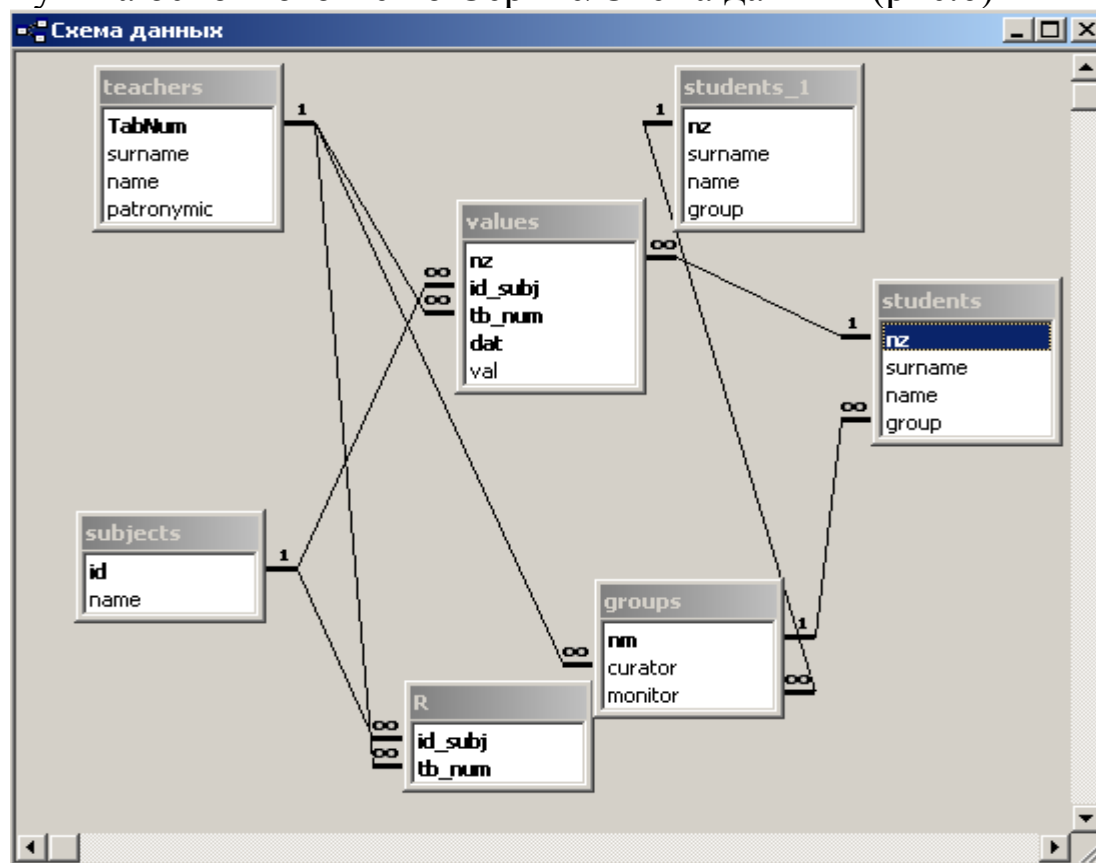


Рис.6

Здесь наглядно указаны структура всех шести полученных таблиц и связи между ними.

Задания по лабораторной работе “Разработка реляционных баз данных”

Задание 1

В качестве предметной области рассматривается автотранспортное предприятие (гараж), предоставляющее услуги автомобильных перевозок топлива. База данных должна хранить данные об автомобилях, водителях, марках перевозимого топлива, данные о перевозках - автомобиль, водитель, перевезенное топливо, пробег, дата. Водители закреплены за автомобилями, причем за одним автомобилем может быть закреплено несколько водителей и каждый водитель может ездить на нескольких автомобилях.

Задание 2.

Агентство занимается продажей авиабилетов на различные рейсы, ведет учет проданных билетов и учет пассажиров, купивших билеты. Поэтому возникает потребность в хранении и обработке данных, сгруппированных следующим образом:

1. информация о расписании рейсов (номер рейса, тип самолета, пункт отправления, пункт назначения, дата вылета, время вылета, время полета, цена билета);
2. информация о свободных местах на рейс (номер рейса, дата вылета, общее количество мест, количество свободных мест);
3. информация о пассажирах, купивших билеты на рейсы (номер паспорта, фамилия, имя, отчество, номер рейса, дата вылета)
4. Архив, в который помещается информация о выполненном рейсе (Номер рейса, дата вылета, общее количество мест, количество проданных мест).

Задание 3.

Спроектировать базу данных для агентства по трудоустройству. Агентство по трудоустройству ведет списки лиц, ищущих работу, и списки вакансий. Вакансии поступают от организаций с указанием должности и оклада. В заявках претендентов, кроме анкетных данных, указывается желаемая должность и оклад, каждая вакансия заполняется несколькими претендентами, согласно их анкетным данным и передается агентством работодателю. Работодатель, независимо от агентства, отбирает одного из претендентов, который должен занять вакансию в базе данных агентства, после этого вакансии других претендентов аннулируются.

Задание 4

Спроектировать базу данных, предназначенную для хранения информации о торгах на товарно-сырьевой бирже. На торги могут быть представлены разные товары одной и той же фирмы и одни и те же товары разных фирм. Каждый товар имеет свой уникальный код, произведен определенной фирмой в определенное время. Товар имеет гарантийный срок хранения, единицу измерения. Товары поставляются партиями. Партия характеризуется: номером, условиями поставки (предоплата или нет). Партии

товаров выставляют брокеры. В одну партию товаров включаются разнообразные товары от разных производителей определенного количества и цены.. Считается, что партии товаров, выставленные на продажу, покупает сама биржа, и она же расплачивается с брокером и производителями товара. Если условием поставки указана предоплата, то биржа перечисляет деньги в день заключения договора, а если нет — то в день отгрузки. Брокеры работают за фиксированный процент прибыли от суммы заключенных сделок и перечисляют конторе, в которой они работают, фиксированную сумму денег, а все остальные заработанные ими деньги составляют их чистый доход (зарплату).

Задание 5.

Рассматривается предприятие, которое занимается поставкой и вводом в эксплуатацию оборудования для предприятий. БД должна хранить данные о заказчиках, о ценах и сроках выполнения заказов. На один вид оборудования может поступать несколько заявок от различных предприятий. На поставку оборудования или комплектующих на предприятие-заказчик оформляется заказ. На одно предприятие может быть поставлено оборудование разных наименований и в нескольких экземплярах по разным заказам. Необходимо вести учет поставок. С заказчиком по каждой поставке составляется договор купли-продажи.

Задание 6.

Сфера услуг отеля включает: обслуживание и бронирование номеров, ресторан, спортивный и тренажерный залы, сауну. Основной услугой является гостиничный номер, все остальные входят в разряд прочих услуг. Все заказы и их оплата записываются в базе данных отеля. Платежи за гостиничные номера и другие услуги учитываются отдельно друг от друга. Список клиентов содержит: код клиента, имя и фамилию клиента, страну, город, номер телефона. Список номеров включает: номер комнаты, класс номера, оплата номера в день. В информацию о платежах за номера входит: дата, код клиента, номер комнаты, число дней. Описание прочих услуг включает дату, код клиента, вид услуги (ресторан, спортивные тренажеры, досуг), размер оплаты.

Задание 7.

Издательство — предприятие, занимающееся выпуском разнообразной печатной продукции. Издательство заключает договор с заказчиком (клиентом) на выполнение заказа.

Заказчиком может выступать частное лицо или организация.

Частное лицо может быть автором издания (или одним из авторов, если их несколько) или представителем автора. Организация для контактов с издательством также имеет своего представителя — контактную персону. Заказ может быть книгой, брошюрой, рекламным проспектом, буклетом, бюллетенем для голосования или каким-либо другим видом издательской продукции.

Подготовленные издательством материалы заказчика печатаются в типографиях, где издательство размещает свои заказы.

Информацию о работе издательства можно сгруппировать следующим образом:

- сведения о заказчиках (частное лицо или организация, личные данные контактной персоны, адрес, телефон, факс);
- сведения о заказах (номер заказа, заказчик, вид печатной продукции, издание, типография, дата приема заказа, отметка о выполнении, дата выполнения заказа);
- сведения об изданиях (код издания, автор и название, объем в печатных листах, тираж, номер заказа);
- сведения об авторах (личные данные: Ф.И.О., домашний адрес, телефон; дополнительные сведения);
- сведения о типографиях (название, адрес, телефон).

Задание 8.

Хозрасчетная поликлиника ведет прием и учет пациентов, учет их посещений (визитов) и учет обслуживания пациентов специалистами (врачами) поликлиники. Существует необходимость в хранении информации обо всех посещениях поликлиники пациентами и о том, на приеме у каких специалистов они находились.

Хранимую информацию о деятельности хозрасчетной поликлиники и ее пациентах можно сгруппировать так:

- пациент (номер истории болезни, Ф. И. О. пациента, домашний адрес пациента, телефон);
- специалист (личный номер специалиста, Ф. И. О. специалиста, специальность, домашний адрес, телефон);

- визиты (пациент, специалист, визит первый или повторный, дата визита, анамнез, диагноз, лечение, стоимость лекарств, стоимость услуг).
- архив, куда переносится информация о пациенте, если после его последнего визита прошел определенный срок (например, 3 года),

Задание 9.

Задание: Спроектировать базу данных и создать приложение для автоматизации работы фирмы по производству обуви. База данных должна хранить данные о каждом сотруднике, список поставщиков необходимой продукции или комплектующих и данные о каждом поставщике, список поставляемой продукции или комплектующих, список выполняемых сотрудниками работ. Каждый поставщик может поставлять несколько видов продукции. Каждый сотрудник может выполнять несколько видов работ, и каждый вид работ может выполняться несколькими сотрудниками.

Задание 10.

В базе данных фирмы “Мебель” хранится информация об изделиях, мастерах, их изготавливающих, клиентах фирмы, а также о накладных, составляемых при отгрузке изделий клиентам. Каждый мастер может изготавливать несколько изделий, однако каждое изделие изготавливается только одним мастером. В накладной указывается отпускная цена изделия, которая может отличаться от его себестоимости вследствие торговых наценок, индивидуальных скидок клиентам, в частности — при оптовых закупках. Руководству фирмы желательно знать, какие изделия выпускаются каждым из мастеров, сколько изделий и на какую сумму

Задание 11.

База данных должна хранить данные о сотрудниках предприятия, перечень отделов и подразделений предприятия. В справочнике содержатся сведения о сотрудниках. Должен учитываться отдел, в котором работает сотрудник. Каждый отдел относится к какому-либо подразделению предприятия, причем несколько отделов относятся к одному подразделению. В базе данных должна храниться информация об отделах и подразделениях.

Задание 12.

Рассмотрим специализированную библиотеку, которая располагает книжным фондом определенной тематической направленности. Предполагается, что каждая книга фонда может быть как в одном экземпляре, так и в нескольких. Поэтому каждому экземпляру книги соответствует уникальный инвентарный номер и библиотечный код книги. Данные о книге содержатся в библиографической карточке, где указывается библиотечный код книги, автор, издательство и год издания. Библиотека выдает книги читателям во временное пользование. При записи в библиотеку каждому читателю присваивается порядковый номер, ему выдается читательский билет и для него заводится учетная карточка. Учетная карточка содержит информацию о выданных и возвращенных книгах.

Рекомендуемая литература.

1. Карпова Т.С.. Базы данных: модели, разработка, реализация. СПб, Питер, 2001. 304с.
2. Бекаревич Ю.Б., Пушкина Н.В. MS ACCESS 2000 за 30 занятий. СПб, 2000. 512с.

Содержание

| | |
|---|--------------------|
| Введение..... | 1 |
| Построение ER-модели..... | 2 |
| Переход к реляционной модели..... | 7 |
| Создание таблиц базы данных..... | 12 |
| Задания по лабораторной работе “Разработка реляционных баз данных”..... | 16 |
| Рекомендуемая литература..... | 21 |
| Содержание..... | 21 |