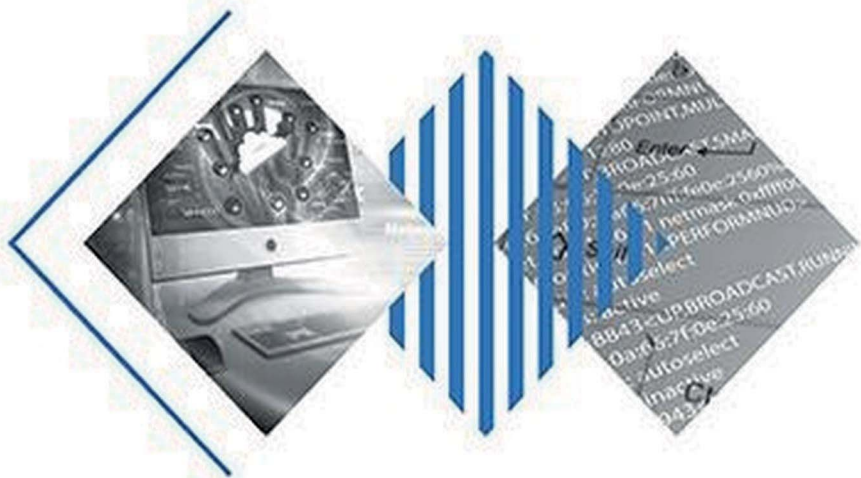


В.И. Виноградов, М.В. Виноградова

ПОСТРЕЛЯЦИОННЫЕ МОДЕЛИ ДАННЫХ И ЯЗЫКИ ЗАПРОСОВ



УДК 004.65
ББК 32.973.2
В49

Издание доступно в электронном виде на портале *ebooks.bmstu.ru*
по адресу: <http://ebooks.bmstu.ru/catalog/193/book1615.html>

Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

*Рекомендовано Редакционно-издательским советом
МГТУ им. Н.Э. Баумана в качестве учебного пособия*

Виноградов, В. И.

В49 Постреляционные модели данных и языки запросов :
учебное пособие / В. И. Виноградов, М. В. Виноградова. —
Москва : Издательство МГТУ им. Н. Э. Баумана, 2017. —
96, [4] с. : ил.

ISBN 978-5-7038-4283-6

Изложены принципы и примеры построения моделей данных: сущность–связь, объектная, реляционная, объектно-реляционная, а также модель полуструктурированных данных. Приведены основные конструкции и правила построения запросов к базам данных на языках SQL, OQL, объектном расширении SQL, Datalog, XPath и XQuery.

Для студентов, обучающихся по программам специальности «Информатика и вычислительная техника» и изучающих дисциплину «Постреляционные базы данных» в МГТУ им. Н.Э. Баумана, а также для студентов других специальностей, изучающих дисциплины, связанные с разработкой программного обеспечения и баз данных.

УДК 004.65
ББК 32.973.2

ISBN 978-5-7038-4283-6

© МГТУ им. Н.Э. Баумана, 2017
© Оформление. Издательство
МГТУ им. Н.Э. Баумана, 2017

Глава 1. СОЗДАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

1.1. Базы данных и системы управления базами данных

База данных (БД) — массив данных, используемый в прикладных программах, который сохраняется после выключения компьютера. Она предназначена для хранения больших объемов информации в структурированном виде в течение длительного времени.

Системы управления базами данных (СУБД) — компьютерные программы, предоставляющие клиентским приложениям доступ к БД.

Из истории развития баз данных

Первое поколение БД появилось в конце 60-х годов XX века. При работе с ними СУБД выполняли функции файловой системы для хранения данных и обеспечения доступа к ним. **Базы данных первого поколения** поддерживали сетевые и иерархические модели данных. Их применяли для бронирования и продажи авиабилетов, в банковских системах (ведение счетов и проводка платежей) и в корпоративных (информация о сотрудниках, товарах и продажах).

Ко *второму поколению* относят **реляционные базы данных**, появившиеся в начале 70-х годов XX века и активно использовавшиеся в течение почти 30 лет. Считается, что их теорию сформулировал Эдгар Франк Кодд. Особенность реляционных БД состоит в том, что все данные и связи между ними хранятся в таблицах. Для определения структуры данных и манипулирования их значениями используют язык SQL (Structured Query Language — структурированный язык запросов).

Третье поколение баз данных, называемых **постреляционные**, начало развиваться с 90-годов XX века. Тогда появились объектные, объектно-реляционные и полуструктурированные БД, которые

расширяют возможности реляционных БД и позволяют хранить и обрабатывать как атомарные значения, так и объекты со сложной структурой.

Объектные базы данных, основанные на объектно-ориентированной парадигме, — альтернатива реляционному подходу. *Объектно-реляционные* БД поддерживают обратную совместимость с реляционными базами и расширяют их возможности. *Полуструктурированные* БД развиваются параллельно на основе сетевых и иерархических БД и позволяют работать с частично структурированными данными.

Проектирование баз данных

Процесс создания информационной системы, содержащей БД, можно разделить на два этапа:

- проектирование и реализация БД;
- разработка приложения для работы с БД.

Основные задачи проектирования БД:

- выбор модели для описания структуры данных;
- определение элементов данных, их типов и связей между ними;
- формирование структур данных в терминах выбранной модели.

При проектировании БД сначала формируют концептуальную модель, а затем на ее основе — логическую. Концептуальная модель не зависит от средств реализации. Для ее построения обычно используют подход *сущность—связь*, который позволяет формализовать объекты предметной области и описать их как наборы сущностей, их атрибутов и связей.

Логическая модель строится с учетом последующей реализации БД в конкретной СУБД. Выделяют четыре основные модели построения логических моделей данных:

- реляционная;
- объектная;
- объектно-реляционная;
- модель полуструктурированных данных.

Перечисленные модели, языки их описания и принципы построения определяются стандартами. Поскольку стандарты обуславливают возможности и ограничения моделей, для физической

реализации в конкретной СУБД логическая модель БД должна быть адаптирована к ее особенностям.

Требования к системе управления базами данных

Программу считают полноценной СУБД, если она обладает следующими возможностями:

- поддерживает язык определения данных (Data Definition Language (DDL)) для создания баз данных и описания их структуры;
- поддерживает язык манипулирования данными (Data Manipulation Language (DML)) для построения запросов к данным (добавление, удаление, изменение и чтение данных в БД);
- длительно (долговременно и постоянно) может хранить большие объемы структурированных данных;
- поддерживает работу в многопользовательской среде (параллельная работа нескольких пользователей);
- восстанавливает информацию после сбоев (как программных, так и аппаратных);
- выполняет транзакции — набор действий, которые осуществляются либо все, либо ни одно;
- защищает информацию от несанкционированного доступа;
- оптимизирует запросы и осуществляет эффективный поиск информации в огромных объемах данных.

Далее будут подробно рассмотрены модели данных, принципы их построения и языки описания.

Вопросы для самопроверки

1. Что называют базой данных и системой управления базами данных?
2. Что такое модель данных?
3. Перечислите поколения баз данных.
4. Назовите известные модели данных. Какие из них относятся к третьему поколению?
5. Чем отличаются логические и концептуальные модели данных?
6. В какой последовательности проектируют базы данных?
7. Сформулируйте определение системы управления базами данных. Какие требования предъявляют к системам управления базами данных?

1.2. Модель сущность–связь




Модель сущность–связь, называемая также ER-моделью (entity-relationship model) позволяет описывать структуру БД на концептуальном уровне. Для построения такой модели применяют графическую нотацию, задающую наборы графических символов, позволяющих описывать объекты предметной области, их атрибуты и связи между ними. Существует несколько нотаций описания модели сущность–связь, например нотации Чена, Мартина, IDEF1X, Баркера и др.

Основные элементы модели

При построении модели сущность–связь с помощью нотации Чена базовыми элементами являются сущности, их атрибуты, ключи и связи между сущностями (табл. 1.1).

Таблица 1.1

Базовые элементы модели сущность–связь в нотации Чена

Элемент модели	Графическое обозначение	Описание элемента
Сущность (набор сущностей)		Определяет объект предметной области или набор подобных объектов
Атрибут		Характеризует сущность. Может относиться к связи
Ключ (первичный ключ)		Соответствует атрибуту(ам), идентифицирующему(им) сущность
Связь		Соединяет две или более сущности. Связи могут быть 1–1, 1–М, М–1 и М–М

Сущность или *набор сущностей* соответствует объекту предметной области или набору подобных объектов с одинаковыми характеристиками (например, сущность Персона). Сущность в нотации Чена имеет имя, уникальное в пределах модели, и обозначается прямоугольником (см. табл. 1.1).

Атрибут характеризует сущность (например, атрибут возраст). Ему присваивают имя, уникальное среди всех атрибутов сущности. Атрибут может иметь атомарное значение (например,

число или строка), множество атомарных значений или структуру, состоящую из атомарных значений. Если атрибут описывает более сложную структуру, возможно, его следует выделить в отдельную сущность. Атрибут может относиться не к сущности, а к связи. Каждый атрибут принадлежит только одной сущности, что в нотации Чена обозначается линией между сущностью и ее атрибутом (рис. 1.1).

Ключ (первичный ключ) (см. табл. 1.1) представляет собой атрибут(ы), идентифицирующий(ие) сущность. Его значение позволяет выделить один конкретный объект предметной области из множества подобных (например, номер ИНН). Составной ключ определяет несколько атрибутов (например, серия и номер паспорта).

Каждая сущность должна иметь ключ. В нотации Чена все атрибуты, входящие в ключ (простой или составной), подчеркиваются. Альтернативные ключи на модели сущность–связь не изображаются.

Связь (см. табл. 1.1) соединяет две сущности и более между собой. Связи могут быть 1–1 (один к одному), 1–М (один ко многим), М–1 (многие к одному) и М–М (многие ко многим). Связь между двумя сущностями называют бинарной, между тремя — тернарной (от лат. *ternarius* — «тройной»), между N сущностями — N -арной. Ей присваивают имя, уникальное в пределах модели.

В нотации Чена конец связи, имеющий множественность 1 (один), обозначают линией со стрелкой, а конец связи, имеющий множественность М (много), — линией без стрелки. Множественность можно указывать явно (1 или М).

Построение простых моделей

Рассмотрим примеры построения простых моделей сущность–связь.

Пример 1.1. *Построение модели для описания фильмов, их актеров и студий.*

Определим предметную область, связанную с фильмами. Для начала выделим основные сущности, их атрибуты, ключи и связи между сущностями. Указываем только информацию, которая будет храниться в БД и представлять интерес для пользователя информационной системы.

Каждый фильм имеет название, год выпуска и длительность. Идентификацию фильма проведем на основе его названия и года.

В фильме играет множество актеров. При этом каждый из них имеет Ф.И.О., ИНН и образование. Идентификатором актера считаем его ИНН. Актер может играть в нескольких фильмах. Полагаем, что фильм снят одной студией, которая имеет название и адрес, идентификатором выберем название.

Проанализировав предметную область, выделим сущности Фильм, Актер, Студия со следующими атрибутами:

Фильм (Название, Год выпуска, Жанр и Длительность),
 составной ключ Название и Год

Актер (Ф.И.О., ИНН, Образование), ключ ИНН

Студия (Название и Адрес), ключ Название

Кроме того, определены следующие связи между сущностями:

- связь бинарная Играл, объединяющая сущности Фильм и Актер, типа М–М (в фильме много актеров, актер играл во многих фильмах);

- связь бинарная Снят, объединяющая сущности Фильм и Студия, типа М–1. Фильм снят одной студией, на студии выпущено много фильмов.

По итогам построения модели сущность–связь получим схему, представленную на рис.1.1.

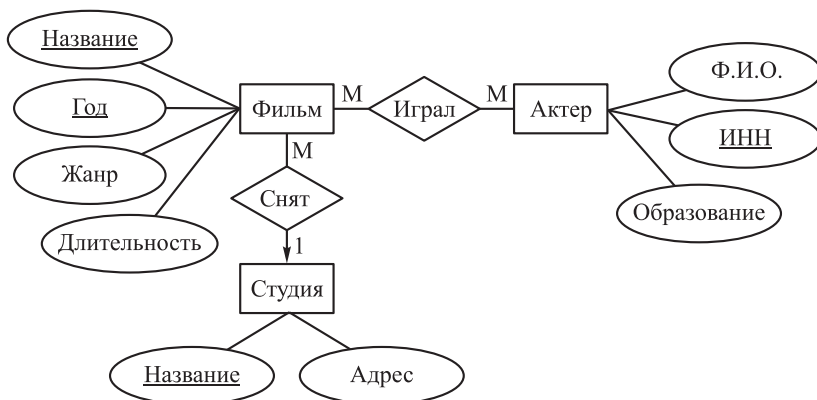


Рис. 1.1. ER-модель БД фильмов

Пример 1.2. Построение модели БД учебных занятий.

Определим предметную область, связанную с проведением лекций. Их читают в аудитории, по дисциплине, для группы. Груп-

па имеет шифр, который ее идентифицирует, и относится к кафедре. Дисциплина имеет название, и ее читает лектор. Идентификатором дисциплины выберем ее название. Аудитория имеет номер и расположена в учебном корпусе. Номер аудитории примем за ее идентификатор.

В аудитории читается много дисциплин для многих групп. Группа слушает много дисциплин во многих аудиториях. Дисциплину читают для разных групп во многих аудиториях.

Проанализировав предметную область, выделим сущности со следующими атрибутами:

Дисциплина (Название, Лектор), ключ Название

Группа (Шифр, Кафедра), ключ Шифр

Аудитория (Номер, Корпус), ключ Номер

Кроме того, определим тернарную связь Изучает, которая соединяет все сущности между собой (рис. 1.2).

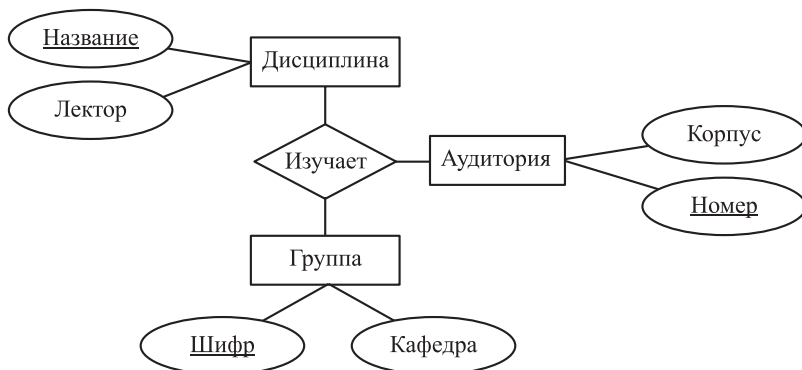


Рис. 1.2. ER-модель БД учебных занятий

Дополнительные элементы модели

Наряду с базовыми элементами в модели сущность–связь допустимо использовать дополнительные элементы, приведенные в табл. 1.2.

Слабая (зависимая) сущность не может функционировать без другой сущности, называемой в данном случае поддерживающей, или независимой (см. табл. 1.2). Например, студент не мо-

Оглавление

Предисловие	3
Глава 1. СОЗДАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ	5
1.1. Базы данных и системы управления базами данных	5
Вопросы для самопроверки	7
1.2. Модель сущность–связь	8
Вопросы для самопроверки	15
1.3. Реляционная модель	15
Вопросы для самопроверки	19
1.4. Описание схемы реляционной БД на языке SQL	20
Вопросы для самопроверки	24
Глава 2. ПОСТРЕЛЯЦИОННЫЕ МОДЕЛИ ДАННЫХ	25
2.1. Объектно-реляционная модель	25
Вопросы для самопроверки	31
2.2. Описание объектно-реляционной модели на языке SQL с объектным расширением	32
Вопросы для самопроверки	39
2.3. Объектная модель	40
Вопросы для самопроверки	49
2.4. Модель полуструктурированных данных	49
Вопросы для самопроверки	57
Глава 3. ФОРМИРОВАНИЕ ЗАПРОСОВ К БАЗАМ ДАННЫХ	58
3.1. Запросы на языке SQL	58
Вопросы для самопроверки	66
3.2. Запросы на языке SQL с объектным расширением	66
Вопросы для самопроверки	71
3.3. Запросы на языке OQL	72
Вопросы для самопроверки	78
3.4. Запросы на языках XPath и XQuery	78
Вопросы для самопроверки	88
3.5. Запросы на языке Datalog	88
Вопросы для самопроверки	95
Литература	96