

в кучу 1 камень или 5 камней. Например, имея кучу из 10 камней, за один ход можно получить кучу из 11 или 15 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 47. Победителем считается игрок, сделавший последний ход, т. е. первым получивший кучу, в которой будет 47 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 46$. Выполните следующие задания, в каждом случае обосновывая свой ответ.

- 1) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающие ходы.
- 2) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.
- 3) Укажите два значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.
- 4) Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани.

§ 12

База данных как модель предметной области

12.1. Общие представления об информационных системах

Способность накапливать информацию и обеспечивать эффективный доступ к ней является сегодня определяющим фактором развития и поддержания жизнеспособности человеческого общества.



Объёмы накопленных человечеством данных неуклонно растут. В 2011 году общий мировой объём сгенерированных человечеством данных составил более 1,8 зеттабайт (1,8 трлн Гбайт). Столько же «весят» 200 млн двухчасовых фильмов в формате высокой чёткости, которые можно просматривать ежедневно без перерыва в течении 47 млн лет. Ожидается, что количество данных на планете будет как минимум удваиваться каждые два года вплоть до 2020 года. (По материалам журнала «Суперкомпьютеры», № 1(17), 2014.)

Развитие средств вычислительной техники и информационных технологий открыло новые способы хранения, представления и поиска информации, сделав возможным создание информационных систем, обеспечивающих практически мгновенное получение каждым требуемой ему информации в той или иной предметной области (части реального мира). Центральной частью любой информационной системы является база данных.



База данных (БД) — совокупность данных, организованных по определённым правилам, отражающая состояние объектов и их отношений в некоторой предметной области (транспорт, медицина, образование, право и т. д.), предназначенная для хранения во внешней памяти компьютера и постоянного применения.

Информационная система — это совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств¹⁾.

В наше время информационные системы находят широкое применение в производственной, маркетинговой, финансовой, кадровой и многих других сферах деятельности.

Примерами информационных систем (рис. 3.22), доступными каждому пользователю, в том числе и с помощью мобильных устройств, являются:

- информационные системы, обеспечивающие возможность получения справочной информации о расписании самолётов, поездов дальнего следования, электричек и автобусов, а также покупке железнодорожных и авиабилетов;
- информационные системы, позволяющие узнать наличие и цены на медикаменты в аптеках, места в отелях города (региона) и др.;

¹⁾ Закон Российской Федерации «Об информации, информационных технологиях и о защите информации».

- разнообразные поисково-информационные картографические службы;
- порталы, содержащие информацию нормативно-правового характера и др.

Узнайте, для чего предназначены информационные системы, представленные на рисунке 3.22.

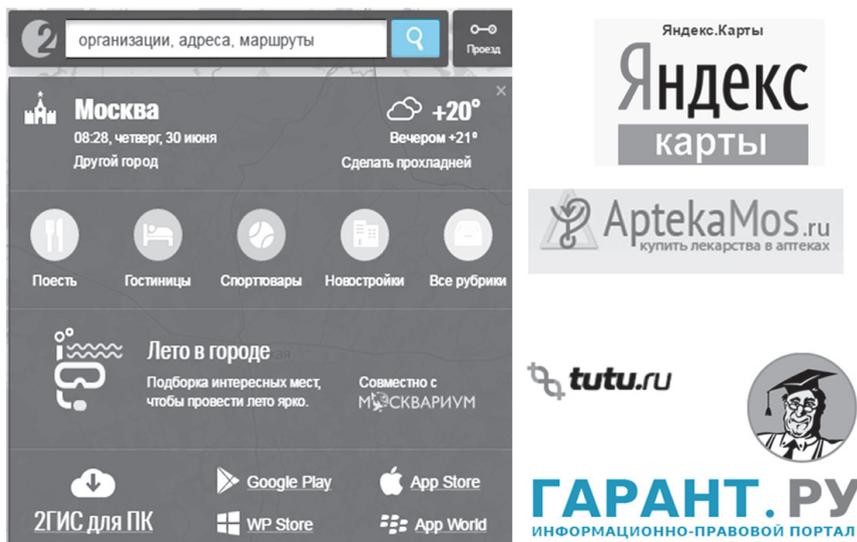


Рис. 3.22. Примеры информационных систем

12.2. Предметная область и её моделирование

Предметная область — это часть реального мира, рассматриваемая в рамках определённой деятельности. Например, можно рассматривать такие предметные области, как школа, библиотека, поликлиника, кинотеатр, склад и т. д.

В предметной области можно выделить некоторые объекты (классы объектов) и зафиксировать их свойства (атрибуты).

Объект предметной области — это факт, лицо, событие, предмет, о котором могут быть собраны данные.

Информационный объект или сущность — это описание некоторого класса реальных объектов в виде совокупности свойств.



Сущность предметной области — это класс объектов предметной области; по сути, это совокупность однотипных объектов.

Примерами объектов (с точки зрения внешнего мира) или сущностей (с точки зрения БД) являются ученик, класс, кабинет, время занятий и т. д. Сущность УЧЕНИК может быть представлена в БД с помощью следующих атрибутов: номер личного дела, фамилия, имя, отчество, год рождения. Это можно записать так:

УЧЕНИК (НОМЕР ЛИЧНОГО ДЕЛА, ФАМИЛИЯ, ИМЯ,
ОТЧЕСТВО, ГОД РОЖДЕНИЯ).

Между объектами, а следовательно, и между соответствующими им сущностями могут существовать связи одного из следующих типов:

- «один к одному» (обозначается 1 : 1);
- «один ко многим» (обозначается 1 : М);
- «многие к одному» (обозначается М : 1);
- «многие ко многим» (обозначается М : М).

Связь 1 : 1 имеет место, когда одному экземпляру одной сущности соответствует один экземпляр другой сущности. Такая связь может быть установлена между сущностями ВЫПУСКНИК_ШКОЛЫ и АТТЕСТАТ: каждый выпускник школы получает аттестат о среднем образовании и каждый аттестат принадлежит одному выпускнику.

Связь 1 : М имеет место, когда одному экземпляру одной сущности может соответствовать несколько экземпляров другой сущности. Например, у матери может быть несколько детей; в одном кинотеатре может быть несколько залов; в одном классе, как правило, множество учеников.

Связь М : 1 является противоположной к связи 1 : М; она имеет место, когда нескольким экземплярам одной сущности соответствует один экземпляр другой сущности. Например, несколько учеников учатся в одном классе.

Связь М : М имеет место, когда нескольким экземплярам одной сущности соответствует несколько экземпляров другой сущности. Например, многие ученики получают много разных оценок; каждый учитель, преподающий в 11 классе, обучает многих учащихся, а каждый учащийся 11 класса обучается у нескольких учителей; один автор может написать несколько книг, и, в то же время, одна книга может быть написана несколькими авторами.

Существуют связи, которыми каждый экземпляр одной сущности обязательно связан с одним или несколькими экземплярами другой сущности. Например, связь между сущностями КЛАСС и УЧЕНИК такова, что каждый ученик принадлежит к определённому классу, и каждый класс состоит из определённой группы учеников. Возможны связи, при которых каждый экземпляр одной сущности не обязательно связан хотя бы с одним экземпляром другой сущности.

Для создания БД необходимо, прежде всего, построить модель её предметной области, определив, данные о каких объектах будут в ней храниться и какие связи между этими данными необходимо учесть.

Модель предметной области, включающую в себя сущности, их атрибуты и связи между сущностями называют моделью «сущность–связь», или ER-моделью (от англ. *Entity–Relationship* — сущность–связь).

Для большей наглядности при создании моделей «сущность–связь» пользуются условными графическими обозначениями: сущности изображаются в виде прямоугольников, атрибуты — в виде эллипсов, связи — в виде ромбов.

Построим модель «сущность–связь» для предметной области «Авиаперелёты», в которой рассмотрим две сущности: ПАССАЖИР и БИЛЕТ (рис. 3.23).

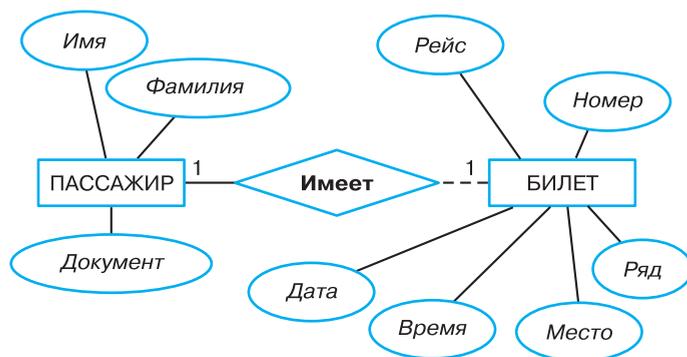


Рис. 3.23. Модель «сущность–связь» предметной области «Авиаперелёты»

Каждый пассажир, собирающийся лететь, например, в Париж, имеет билет. Двух одинаковых билетов, как и двух одинаковых пассажиров, не существует. Сущность ПАССАЖИР в данной модели характеризуется свойствами: ФАМИЛИЯ, ИМЯ и ДОКУМЕНТ, удостоверяющий личность. Атрибуты сущности БИЛЕТ — РЕЙС, ДАТА, ВРЕМЯ, РЯД, МЕСТО и НОМЕР БИЛЕТА. Между сущностями ПАССАЖИР и БИЛЕТ существует связь **Имеет**. Это связь «один к одному»; соответствующие обозначения находятся над линиями связи возле прямоугольников сущностей. Эта связь является обязательной (сплошная линия на схеме) для сущности ПАССАЖИР (для того чтобы быть пассажиром, человек должен иметь билет) и необязательной (пунктирная линия на схеме) для сущности БИЛЕТ, поскольку не все билеты на рейс могут быть проданы.

12.3. Представление о моделях данных



Модель данных — это совокупность структур данных и операций их обработки.

С помощью модели данных могут быть представлены сущности и взаимосвязи между ними. Выделяют три основных типа моделей данных: иерархическую, сетевую и реляционную.

Иерархическая модель данных определяет организацию данных об объектах в виде дерева. В иерархической модели данных у каждого объекта есть только один объект высшего уровня, которому он подчинен (родительский), и может быть несколько подчиненных объектов (потомков). Исключение составляет только наивысший по иерархии объект — у него нет родительского объекта. В иерархической модели данных каждый родительский объект в совокупности с подчиненными объектами (потомками) можно рассматривать как отдельное дерево.

Пример иерархической организации данных представлен на рисунке 3.24. Информация БД «Школа» структурирована в виде иерархических деревьев, количество которых равно количеству подразделений (НАЧАЛЬНАЯ ШКОЛА, ОСНОВНАЯ ШКОЛА, СТАРШАЯ ШКОЛА). На первом уровне находится сущность ПОДРАЗДЕЛЕНИЕ (НОМЕР, НАЗВАНИЕ, РУКОВОДИТЕЛЬ). Сущности второго уровня — КЛАССЫ (КОД КЛАССА, КЛАССНЫЙ

РУКОВОДИТЕЛЬ), сущности третьего уровня — УЧЕНИКИ (ЛИЧНОЕ ДЕЛО, ФАМИЛИЯ). Подчёркиванием выделен атрибут, который однозначно определяет каждый экземпляр сущности.

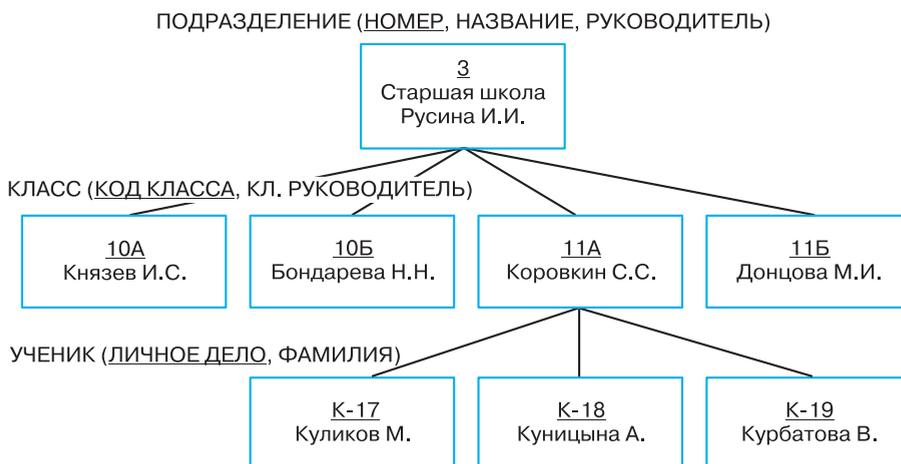


Рис. 3.24. Пример иерархической организации данных

Для обработки данных в иерархической модели данных используется следующий набор команд:

- найти указанное дерево (например, дерево 1);
- перейти от одного дерева к другому (например, от дерева 11А к дереву 10А);
- перейти от родительского объекта к объекту-потомку внутри дерева (например, от объекта 11А к объекту К-18);
- перейти от одного объекта к другому в порядке, предусмотренном иерархической структурой (например, от объекта 9А к объекту 10А);
- вставить новый объект в указанном месте;
- удалить текущий объект и др.

Модель данных должна обеспечивать целостность данных, иначе говоря, в представленных с её помощью данных не должно быть противоречий. Свойство целостности должно сохраняться при любых действиях с данными.

Основное правило обеспечения целостности в иерархической модели данных состоит в том, что ни один подчиненный объект (потомок) не может существовать без родительского объекта, за исключением одного основного родительского объекта.

При значительном количестве данных в БД, построенных по иерархической модели, поиск нужных данных может занять много времени. Например, поиск файла, содержащего определённый фрагмент текста, на всех жёстких дисках персонального компьютера может длиться несколько минут.



В Интернете подобный поиск будет длиться максимум несколько секунд, при этом будут обработаны значительно большие объёмы данных.

Постарайтесь вспомнить, за счёт чего так быстро происходит поиск в Интернете. Как это связано с индексацией данных? Используются ли аналогичные возможности в современных операционных системах?

Иерархическую модель данных удобно использовать для предметной области, объекты которой также имеют между собой иерархическую зависимость. Для предметной области, в которой объекты связаны между собой более сложной зависимостью, чем иерархия, может быть использована сетевая модель данных.

Сетевая структура данных предусматривает, что у каждого объекта может быть как несколько объектов-потомков, так и несколько родительских объектов. Пример связей между объектами при использовании сетевой модели данных изображен на рисунке 3.25.

Для обработки данных в сетевой модели данных используются команды:

- найти указанный объект среди однотипных объектов, например объект с данными об ученике Кучеренко М.;

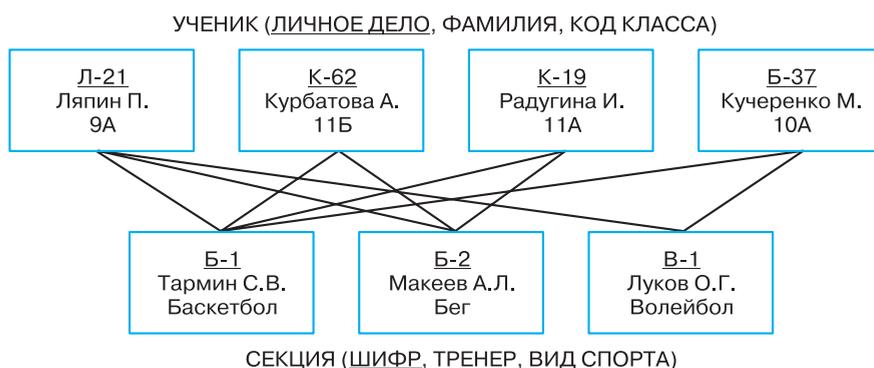
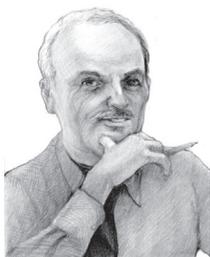


Рис. 3.25. Сетевая структура данных

- перейти от родительского объекта к первому потомку, используя определённую связь;
- перейти от объекта-потомка к родительскому объекту, используя определённую связь;
- вставить новый объект в указанном месте;
- удалить текущий объект;
- изменить объект;
- включить объект в определённую связь;
- разорвать связь и др.

Использование сетевой модели данных осложняется при значительном увеличении количества объектов предметной области и усложнении связей между ними.

Основой структуры реляционной модели данных является таблица, каждая строка которой содержит набор значений свойств одного из объектов предметной области, а каждый столбец — набор значений определённого свойства объектов предметной области. Таблица реляционной БД состоит из элементов определённых множеств, что позволяет для обработки данных этой таблицы использовать операции над множествами.



Эдгар Франк Кодд (1923–2003) — британский учёный, внёсший существенный вклад во многие области информатики. Кодд является создателем реляционной модели данных и основоположником теории реляционных БД. Математик по образованию, он ввёл в теорию БД математический подход, основывающийся на теории множеств.

Целостность¹⁾ в реляционной модели данных обеспечивается соблюдением двух принципов:

- 1) обязательная возможность идентификации объекта (экземпляра сущности) за счёт уникальности набора значений его свойств, указанных в строке реляционной таблицы;
- 2) обязательная корректность связей между таблицами.

В соответствии с моделью данных, лежащей в основе БД, различают иерархические, сетевые и реляционные БД. Последние мы рассмотрим более подробно.

¹⁾ Более подробно этот вопрос будет раскрыт в следующем параграфе.

12.4. Реляционные базы данных

Итак, основным объектом реляционной БД является таблица. Каждая такая таблица, называемая реляционной таблицей или отношением, обладает следующими свойствами:

- все столбцы в таблице однородные, т. е. все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись (рис. 3.26).

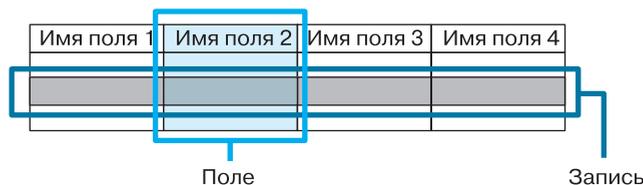


Рис. 3.26. Основные структурные элементы реляционной таблицы

Поле (столбец реляционной таблицы) — элементарная единица логической организации данных, которая соответствует конкретному атрибуту сущности.

Запись (строка реляционной таблицы) — совокупность логически связанных полей, соответствующая конкретному экземпляру сущности. Например, информация о крупнейших озёрах мира в виде реляционной таблицы представлена на рисунке 3.27.

Для наглядности представления связей между таблицами переходят к представлению структур таблиц, указывая только имена полей:

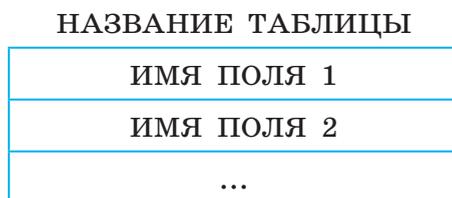




Рис. 3.27. Таблица реляционной БД

Например, структура таблицы, представленной на рисунке 3.27, будет иметь вид:

КРУПНЕЙШИЕ ОЗЕРА

НАЗВАНИЕ
МАТЕРИК
МАКСИМАЛЬНАЯ ГЛУБИНА
ПЛОЩАДЬ

Первичный ключ (идентификатор) реляционной таблицы — это поле или совокупность полей, которые однозначно определяют каждую строку (запись) в таблице.

Основные свойства первичного ключа:

- 1) однозначная идентификация записи (запись должна однозначно определяться значением ключа);
- 2) отсутствие избыточности (удаление любого поля первичного ключа приведёт к нарушению свойства однозначной идентификации записи).

Ключ, состоящий из одного поля, называется простым ключом (ключевым полем). Ключ называется составным, если он включает в себя несколько полей.

В таблице БД, представленной на рисунке 3.27, в качестве ключевого можно использовать поле НАЗВАНИЕ: значения в этом поле являются уникальными для каждой записи, потому что крупных озёр с одинаковыми названиями не существует.

Для хранения данных о сущностях некоторой предметной области может использоваться несколько связанных между собой таблиц. Связь между таблицами устанавливается с помощью ключевых полей.

Можно связать две реляционные таблицы, если ключ одной связываемой таблицы ввести в состав ключа другой таблицы (возможно совпадение ключей). Ключевое поле одной связываемой таблицы можно ввести в структуру другой таблицы, при этом оно уже не будет ключевым; такое поле называется внешним ключом.



Между таблицами *A* и *B* установлена связь «один к одному», если каждая запись в таблице *A* может иметь не более одной связанной с ней записи в таблице *B*, и наоборот — каждая запись в таблице *B* может иметь не более одной связанной с ней записи в таблице *A*.

Связь между таблицами имеет тип «один к одному», если она установлена по совпадающим первичным ключам.

Таблицы **УЧЕНИК** и **УЧЕБНЫЙ ГОД** связаны по типу «один к одному» (рис. 3.28). В этом случае имеет место совпадение ключей.

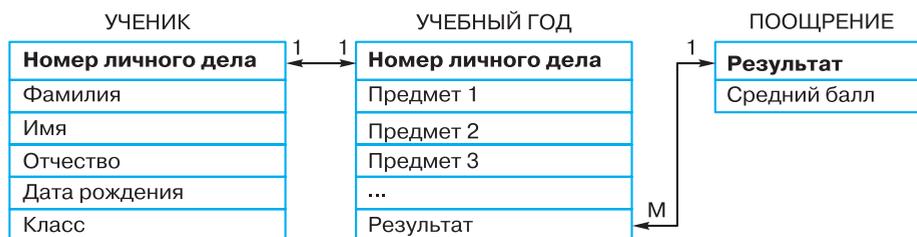


Рис. 3.28. Пример связей «один к одному» и «один ко многим» между таблицами

Между таблицами *A* и *B* установлена связь «один ко многим», если каждая запись в таблице *A* может быть связана с несколькими записями таблицы *B*, но каждая запись в таблице *B* не может быть связана более чем с одной записью таблицы *A*.



Между таблицами **ПООЩРЕНИЕ** и **УЧЕБНЫЙ ГОД** связь «один ко многим». В этом случае ключевое поле одной связываемой таблицы (**ПООЩРЕНИЕ**) введено в структуру другой таблицы (**УЧЕБНЫЙ ГОД**) так, что там оно уже не является ключевым и рассматривается как внешний ключ. В рамках этой связи таблица, содержащая первичный ключ, считается главной, а таблица, содержащая внешний ключ, считается подчинённой.

Между таблицами *A* и *B* установлена связь «многие ко многим», если каждой записи таблицы *A* может соответствовать несколько записей в таблице *B*, и наоборот — каждой записи таблицы *B* может соответствовать несколько записей в таблице *A*.



Такая связь всегда реализуется с помощью третьей связующей таблицы *C*. Связь «многие ко многим» представляет собой комбинацию двух связей типа «один ко многим»: между таблицами *A* и *C* и между таблицами *B* и *C*.

Например, связь между таблицами **ЧИТАТЕЛЬ** и **КНИГА** базы данных **БИБЛИОТЕКА** может быть реализована с помощью таблицы **АБОНЕМЕНТ**¹⁾ (рис. 3.29).

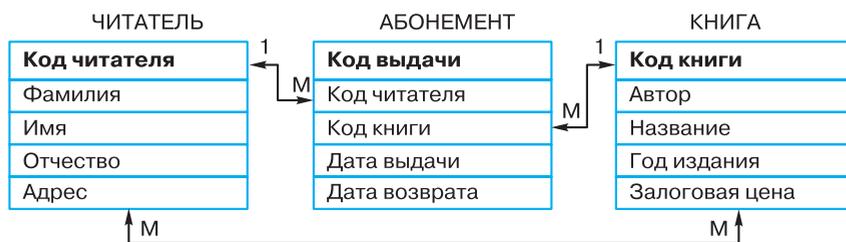


Рис. 3.29. Пример связей «один ко многим» и «многие ко многим» между таблицами

¹⁾ Самостоятельно уточните значение слова «абонемент» в дополнительных источниках информации.

САМОЕ ГЛАВНОЕ

База данных (БД) — совокупность данных, организованных по определённым правилам, отражающая состояние объектов и их отношений в некоторой предметной области, предназначенная для хранения во внешней памяти компьютера и постоянного применения.

Информационная система — это совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств.

Модель данных — это совокупность структур данных и операций их обработки. Выделяют три основных типа моделей данных: иерархическую, сетевую и реляционную.

Основным объектом реляционной БД является таблица, обладающая следующими свойствами: все столбцы в таблице однородные, т. е. все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер; каждый столбец имеет уникальное имя; одинаковые строки в таблице отсутствуют; порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись.

Первичный ключ (ключевое поле, идентификатор) реляционной таблицы — это поле или совокупность полей, которые однозначно определяют каждую строку (запись) в таблице. Ключ, состоящий из одного поля, называется простым ключом. Ключ называется составным, если он включает в себя несколько полей.

Для хранения данных о сущностях некоторой предметной области может использоваться несколько связанных между собой таблиц. Связь между таблицами устанавливается с помощью ключевых полей.

Между таблицами A и B установлена связь «один к одному», если каждая запись в таблице A может иметь не более одной связанной с ней записи в таблице B , и наоборот — каждая запись в таблице B может иметь не более одной связанной с ней записи в таблице A .

Между таблицами A и B установлена связь «один ко многим», если каждая запись в таблице A может быть связана с несколькими записями таблицы B , но каждая запись в таблице B не может быть связана более чем с одной записью таблицы A .

Между таблицами A и B установлена связь «многие ко многим», если каждой записи таблицы A может соответствовать несколько записей в таблице B , и наоборот — каждой записи таб-

лицы *B* может соответствовать несколько записей в таблице *A*. Такая связь всегда реализуется с помощью третьей связующей таблицы *C*. Связь «многие ко многим» представляет собой комбинацию двух связей типа «один ко многим»: между таблицами *A* и *C* и между таблицами *B* и *C*.

Можно связать две реляционные таблицы, если ключ одной связываемой таблицы ввести в состав ключа другой таблицы (возможно совпадение ключей). Ключевое поле одной связываемой таблицы можно ввести в структуру другой таблицы, при этом оно уже не будет ключевым; такое поле называется внешним ключом.

Вопросы и задания



1. Для чего нужно упорядоченное хранение данных?
2. Что такое информационная система? Каково основное назначение информационных систем?
3. Имеете ли вы опыт использования каких-либо информационных систем?
4. Что такое база данных? Как связаны информационная система и база данных?
5. Что такое предметная область? Как представляются объекты предметной области и их свойства в информационной модели предметной области?
6. Что такое сущность? Что такое экземпляр сущности? Приведите примеры.
7. Что называют моделью «сущность–связь»?
8. Постройте модель «сущность–связь» для предметной области «Концертный зал».
9. Назовите типы связей между сущностями предметной области.
10. Определите тип связей между сущностями:
 - 1) КЛИЕНТ и ЗАКАЗ в интернет-магазине;
 - 2) МАШИНА и ЧАСТИ МАШИНЫ;
 - 3) УЧИТЕЛЬ и УЧЕНИК в школе;
 - 4) КОМНАТА и ГОСТЬ в отеле;
 - 5) ГРАЖДАНИН и ПАСПОРТ.
11. Что такое модель данных? Для чего она создаётся?
12. Опишите иерархическую модель данных.
13. Опишите сетевую модель данных.
14. Опишите реляционную модель данных.



15. Опишите таблицу реляционной БД.
16. Что такое ключевое поле? Каковы требования к ключевому полю?
17. Какого типа связи могут быть установлены между таблицами реляционной БД? Охарактеризуйте каждый тип связи.
18. Во фрагменте БД представлены сведения об участниках выставки:

Таблица 1

Страна	Участник
Великобритания	Стив
Германия	Мейер
США	Кинкейд
Россия	Сафронов
Канада	Селби
Германия	Рихард
Великобритания	Дейв
Германия	Гюнтер
Россия	Глазунов
Германия	Зив

Таблица 2

Участник	Жанр
Глазунов	Натюрморт
Селби	Пейзаж
Сафронов	Портрет
Мейер	Пейзаж
Кинкейд	Пейзаж
Дейв	Портрет
Рихард	Натюрморт
Гюнтер	Пейзаж
Зив	Натюрморт
Стив	Портрет

- 1) Охарактеризуйте связь между представленными таблицами БД.
- 2) Художники из скольких стран представили на выставке пейзажи?
- 3) Представьте всю имеющуюся информацию о выставке в одной таблице.
- 4) Представьте всю имеющуюся информацию о выставке в форме графа.

19. Во фрагменте БД представлены сведения о родственных отношениях:



Таблица 1

ID	Фамилия И. О.	Пол
2272	Диковец А. Б.	Ж
2228	Диковец Б. Ф.	М
2299	Диковец И. Б.	М
2378	Диковец П. И.	М
2356	Диковец Т. И.	Ж
2331	Тесла А. П.	М
1217	Тесла П. А.	М
1202	Ландау М. А.	Ж
2227	Решко Д. А.	Ж
2240	Решко В. А.	Ж
2322	Друк Г. Р.	Ж

Таблица 2

ID Родителя	ID Ребёнка
2227	2272
2227	2299
2228	2272
2228	2299
2272	2240
2272	1202
2272	1217
2299	2356
2299	2378
2322	2356
2322	2378

Представьте имеющуюся информацию в форме графа и ответьте на следующие вопросы.

- 1) Сколько внуков у Решко Д. А.?
- 2) Информация о скольких супружеских парах представлена в таблицах?
- 3) Какой идентификационный номер (ID) у дяди Решко В. А.?