

**Министерство образования и науки Украины
Донбасская государственная машиностроительная академия**

ЛЕКЦИОННЫЙ МАТЕРИАЛ К ИЗУЧЕНИЮ КУРСА

по дисциплине:

«ОРГАНИЗАЦИЯ БАЗ ДАННЫХ И ЗНАНИЙ»

(часть 1)

для студентов специальности
7.04030302 «Системы и методы принятия решений»,
дневной и заочной форм обучения

Краматорск 2014

**Министерство образования и науки Украины
Донбасская государственная машиностроительная академия**

ЛЕКЦИОННЫЙ МАТЕРИАЛ К ИЗУЧЕНИЮ КУРСА

по дисциплине:

«ОРГАНИЗАЦИЯ БАЗ ДАННЫХ И ЗНАНИЙ»

(часть 1)

для студентов специальности
7.04030302 «Системы и методы принятия решений»,
дневной и заочной форм обучения

Утверждено
на заседании кафедры ИСПР
Протокол № 2 от 09.09.2014

Краматорск 2014

Лекционный материал к изучению курса по дисциплине «Организация баз данных и знаний» (для студентов специальности 7.04030302 «Системы и методы принятия решений», заочной и дневной форм обучения) / сост. О. Л. Ольховская. – Краматорск: ДГМА, 2014. – 26 с.

Составитель	Ольховская Оксана Леонидовна, к.э.н., доцент
Отв. за выпуск	Мельников Александр Юрьевич, к.т.н., доцент

СОДЕРЖАНИЕ

1 Проектирование баз данных	5
1.1 Основные понятия и определения теории баз данных и знаний	5
1.2 Информационное моделирование предметных областей	12
1.2.1 Инфологическое проектирование базы данных	14
1.2.2 Даталогическое проектирование базы данных (реляционная модель данных)	16
1.2.3 Модели данных	19
1.3 Проектирование реляционных баз данных с использованием нормализации	21
Список рекомендуемой литературы	25

1 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

1.1 Основные понятия и определения теории баз данных и знаний

Основные вопросы: Понятие «информация» и «данные». Роль информации в современном мире. Семантика данных. Понятие база данных, базы знаний, банка данных. Понятие системы управления базами данных (СУБД): назначение, возможности.

Основные компоненты архитектуры СУБД и соотношение запросов пользователей с БД.

Понятие «информация» и «данные». Роль информации в современном мире.

Информацией называют любые сведения о каких-либо явлениях, событиях, процессах, являющиеся объектами восприятия, передачи, преобразования, хранения и использования. На основе информации углубляются познания законов развития материального мира, взаимосвязываются и координируются работы, контролируются процессы и принимаются решения.

Реальный мир воспринимается как последовательность разных, хотя иногда и взаимосвязанных, явлений. Люди всегда пытались описать эти часто непонятные для них явления. Такое описание называют данными, т. е. информация, фиксируемая в определенной форме и пригодная для последующей обработки, хранения и передачи, называется **данными**. Данные характеризуют некоторую предметную область и являются информационной моделью некоторой части реального мира.

Предметной областью называют определенную часть реального мира, представляющую интерес для конкретного исследования или планируемых действий и соответственно для использования и отображения в информационной системе (в банке данных или знаний).

Процесс восприятия состояния системы в виде данных, описывающих состояние системы, называется **фиксацией данных**.

Обычно для фиксации данных применяется конкретное средство общения (естественный язык, изображения) на конкретном носителе (на камне, доске, бумаге, а в современных условиях – на магнитном или оптическом носителе).

Естественный язык достаточно гибок для представления данных и их интерпретации (семантики), поэтому обычно и то, и другое фиксируется совместно. Примером может служить утверждение «Предельная прочность разрушаемых пород 100». Здесь «100» – данное, а «Предельная прочность разрушаемых пород» – его семантика.

Семантика, в широком смысле слова, – анализ отношения между языковыми выражениями и миром, реальным или воображаемым, а также само это отношение и совокупность таких отношений. Данное отношение состоит в том, что языковые выражения (слова, словосочетания, предложения, тексты) обозначают то, что есть в мире.

Основные понятия и определения теории база данных и знаний.

Банки данных и банки знаний являются наиболее совершенной и прогрессивной формой организации информации и знаний в персональных компьютерах. Главная их задача – ответы на информационные запросы пользователей к банку данных или банку знаний с целью получения искомой информации, т. е. обеспечение пользователей требуемой информацией.

Банк данных – это автоматизированная система, включающая базу данных, лингвистические, программные, технические, организационно-методические средства, обеспечивающие централизованное накопление и коллективное многоцелевое использование информации в различных областях деятельности пользователей. В банке данных содержатся совокупности фактов о качественных и количественных характеристиках конкретных объектов предметной области.

Банк данных – это одна из форм информационных систем.

Банк данных (БнД) — это автоматизированная система, представляющая совокупность информационных, программных, технических средств и персонала, обеспечивающих хранение, накопление, обновление, поиск и выдачу данных. Главными составляющими банка данных являются база данных и программный продукт, называемый системой управления базой данных (СУБД).

База данных (БД) – совокупность данных конкретной предметной области. Они организованы по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, и не зависят от программ обработки.

Термины "банк данных" и "база данных" являются очень близкими синонимами для обозначения некоторого структурированного массива информации. Предполагается, что банки данных содержат информацию, с

которой можно производить достаточно ограниченное число манипуляций (поиск, просмотр), в то время как базы данных предоставляют возможность какой-то специальной обработки информации (с помощью специально написанных программ).

Банк данных является сложной человеко-машинной системой, включающей в свой состав различные взаимосвязанные и взаимозависимые компоненты. Ядром банка данных является база данных. Информационный компонент банка данных состоит из баз данных, схем баз данных и словарей данных. Словари данных имеют особо важное значение. **Метаинформация** включает в себя описание структуры БД (схемы и подсхемы), модель предметной области, информацию о пользователях и их правах, описание формы входных и выходных документов. Централизованное хранилище метаинформации называется **словарем данных**. Особенно большое значение имеют словари данных в системах автоматизированного проектирования ИС.

Банки данных классифицируют по следующим свойствам:

- 1) скорость доступа – определяет время реакции, т.е. получение ответа на запрос пользователя;
- 2) доступность – определяет, какие данные, содержащиеся в БД, доступны данной категории пользователей;
- 3) гибкость – определяет возможность получить ответ на сложные запросы;
- 4) целостность – отвечает за снижение избыточности данных, согласованность данных при упорядочении обновления.

База знаний — это совокупность моделей, правил и факторов (данных), порождающих анализ и выводы для нахождения решений сложных задач в некоторой предметной области.

База знаний состоит из локальных библиотек данных, содержащих определения всех объектов конкретной модели, а также общей библиотеки, в которой хранятся все «встроенные» в систему описания объектов.

База знаний, БЗ (англ. *Knowledge base, KB*) — это особого рода база данных, разработанная для управления знаниями (метаданными), то есть сбором, хранением, поиском и выдачей знаний.

База знаний — это" совокупность моделей, правил и факторов (данных), порождающих анализ и выводы для нахождения решений сложных задач в некоторой предметной области.

База знаний — организованная совокупность знаний, относящихся к какой-нибудь предметной области. Знанием является проверенный практикой результат познания действительности. Иначе говоря, знание — это накопленные человечеством истины, факты, принципы и прочие объекты познания. Поэтому в отличие от базы данных, в базе знаний располагаются познаваемые сведения, содержащиеся в документах, книгах, статьях, отчетах.

Базы данных — множество взаимосвязанных единиц данных, которые могут обрабатываться одной или несколькими прикладными системами.

База данных — это совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

База данных — это объект управления в банке данных. База данных описывает состояние объектов предметной области на определенный момент времени совокупностью предложений на некотором формализованном языке. При этом определяются значения всех факторов на данный момент в виде совокупности взаимосвязанных хранящихся вместе данных.

Особенностью базы данных является то, что она рассчитана на использование при создании различных независимых программ и приложений. В этом основное отличие базы данных от обычных файлов данных — они используются только в конкретно разработанном для них приложении.

Система управления базами данных (СУБД) состоит из языковых и программных средств, предназначенных для создания, ведения и эксплуатации баз данных.

Система управления базами знаний — это объектная БД с возможностями интеллектуального поиска и автоматического переупорядочивания структуры в зависимости от действий пользователей.

СУБД — это программное обеспечение, которое взаимодействует с прикладными программами пользователя и базой данных и обладает перечисленными ниже возможностями:

- позволяет создать базу данных, что обычно осуществляется с помощью языка определения данных (**DDL – Data Definition Language**). Язык DDL предоставляет пользователям средства указания типа данных и

их структуры, а также средства задания ограничений для информации, хранимой в базе данных;

– позволяет вставлять, обновлять, удалять и извлекать информацию из базы данных, что обычно осуществляется с помощью языка манипулирования данными (**DML – Data Manipulation Language**). Наличие централизованного хранилища всех данных и их описаний позволяет использовать язык DML как общий инструмент организации запросов, который иногда называют языком запросов (*query language*). Наличие языка запросов позволяет устранить присущие файловым системам ограничения, при которых пользователям приходится иметь дело только с фиксированным набором запросов или постоянно возрастающим количеством программ, что порождает другие, более сложные проблемы управления программным обеспечением.

Наиболее распространенным типом непроцедурного языка является язык структурированных запросов (**Structured Query Language – SQL**), который в настоящее время определяется специальным стандартом и фактически является обязательным языком для любых реляционных СУБД:

1 Предоставляет контролируемый доступ к базе данных с помощью перечисленных ниже средств:

а) системы обеспечения защиты, предотвращающей несанкционированный доступ к базе данных со стороны пользователей;

б) системы поддержки целостности данных, обеспечивающей непротиворечивое состояние хранимых данных;

в) системы управления параллельной работой приложений, контролирующей процессы их совместного доступа к базе данных;

г) системы восстановления, позволяющей восстановить базу данных до предыдущего непротиворечивого состояния, нарушенного в результате сбоя аппаратного или программного обеспечения;

д) доступного пользователям каталога, содержащего описание хранимой в базе данных информации.

Другими словами, система управления базами данных (СУБД) – это программная система для централизованного управления данными, хранимыми в базе данных, и их поддержания в состоянии, соответствующем предметной области.

При выполнении основных из этих функций СУБД должна использовать различные описания данных. Естественно, что проект базы данных надо начинать с анализа предметной области и выявления

требований к ней отдельных пользователей (сотрудников организации, для которых создается база данных). Проектирование обычно поручается человеку (группе лиц) – *администратору базы данных* (АБД). Им может быть как специально выделенный сотрудник организации, так и будущий пользователь БД, достаточно хорошо знакомый с машинной обработкой данных.

Объединяя частные представления о содержимом БД, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться в будущих приложениях, АБД сначала создает обобщенное неформальное описание создаваемой базы данных. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных, называют ***инфологической моделью данных*** (рис. 1.1).

Такая человеко-ориентированная модель полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока какие-то изменения в реальном мире не потребуют изменения в ней некоторого определения, чтобы эта модель продолжала отражать предметную область. Остальные модели, показанные на рис. 1.1, являются компьютеро-ориентированными. С их помощью СУБД дает возможность программам и пользователям осуществлять доступ к хранимым данным лишь по их именам, не заботясь о физическом расположении этих данных. Нужные данные отыскиваются СУБД на внешних запоминающих устройствах по *физической* модели данных.

Так как указанный доступ осуществляется с помощью конкретной СУБД, то модели должны быть описаны на *языке описания данных* этой СУБД. Такое описание, создаваемое АБД по инфологической модели данных, называют ***даталогической*** или ***концептуальной*** моделью данных.

Трехуровневая архитектура (инфологический, даталогический и физический уровни) позволяет обеспечить *независимость хранимых данных* от использующих их программ. АБД может при необходимости переписать хранимые данные на другие носители информации и (или) реорганизовать их физическую структуру, изменив лишь физическую модель данных. АБД может подключить к системе любое число новых пользователей (новых приложений), дополнив, если надо, даталогическую модель. Указанные изменения физической и даталогической моделей не

будут замечены существующими пользователями системы (окажутся "прозрачными" для них), так же как не будут замечены и новые пользователи. Следовательно, независимость данных обеспечивает возможность развития системы баз данных без разрушения существующих приложений [Т.Ф. Лебедева, БАЗЫ ДАННЫХ, Учебное пособие, Кемерово, 2006, с. 99, с. 11-12].



Рисунок. 1.1. Уровни моделей данных

С функцией обработки данных тесно связано управление транзакциями. **Транзакция** – это последовательность операций над БД, рассматриваемых СУБД как единое целое. В процессе функционирования СУБД транзакция либо успешно выполняется и СУБД фиксирует (commit) изменения БД, произведенные ей во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД.

При помощи транзакций поддерживается логическая целостность базы данных за счет объединения элементарных операций над разными файлами в одну транзакцию.

1.2 Информационное моделирование предметных областей

Основные вопросы: сущность информационного моделирования. Этапы проектирования базы данных.

Реальным средством моделирования данных является не формальный метод нормализации отношений, а так называемое семантическое моделирование. В качестве инструмента семантического моделирования используются различные варианты диаграмм *сущность-связь* (*ER – Entity-Relationship*).

Диаграммы сущность-связь позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей.

Различают концептуальные и физические ER-диаграммы. Концептуальные диаграммы не учитывают особенностей конкретных СУБД. Физические диаграммы строятся по концептуальным диаграммам и представляют собой прообраз конкретной базы данных. Сущности, определенные в концептуальной диаграмме становятся таблицами, атрибуты становятся колонками таблиц (при этом учитываются допустимые для данной СУБД типы данных и наименования столбцов), связи реализуются путем миграции ключевых атрибутов родительских сущностей и создания внешних ключей.

При правильном определении сущностей, полученные таблицы будут сразу находиться в ЗНФ.

Наименование **ER-диаграмм** происходит от английского «Entity-Relationship» («сущность-связь»).

Диаграммы **ER-экземпляров** иллюстрируют степень связи и класс принадлежности сущностей.

Классическая форма представления сущностей и их атрибутов использует прямоугольники для обозначения самих сущностей и овалы – для обозначения их атрибутов. Связи между сущностями принято обозначать ромбом (рис. 2.2).

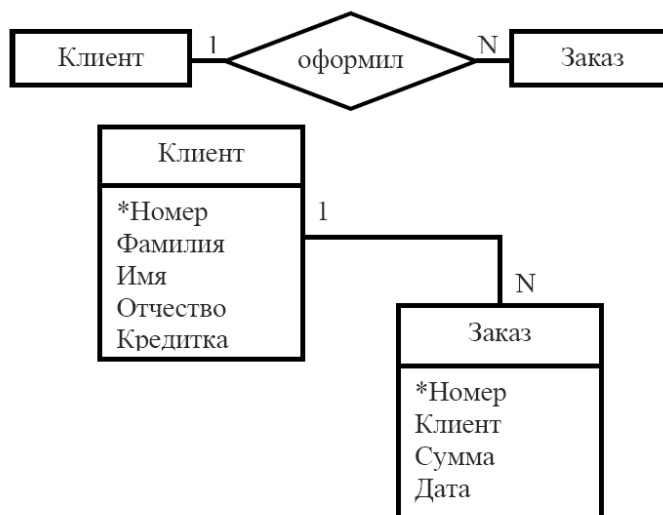


Рис. 2.2. Связи между сущностями

Сущности, для которых связи являются обязательными, называют **слабыми**. В случае если при проектировании необходимо отметить слабую сущность, ее принято обозначать прямоугольником со sdвоенными границами (рис. 2.3).



Рис. 2.3. Обозначение слабой сущности

Преобразование связи «многие-ко-многим». Связь вида «многие-ко-многим» (рис. 2.4) нереализуема в реляционной модели данных.



Рис. 2.4. Связь «многие-ко многим»

Такую связь требуется преобразовать в две связи вида «один ко многим» введением некоторой вспомогательной сущности.

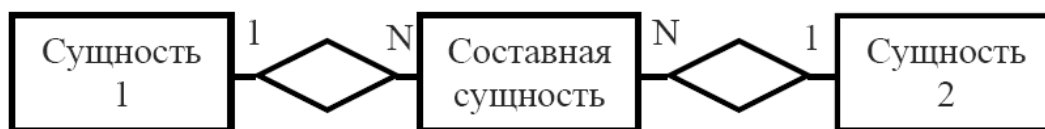


Рис. 2.5. Преобразованная связь

Проектирование базы данных делится на два этапа: построение **инфологической** (информационно-логической, или семантической) модели (рис. 1.1) и **дatalogической** модели (рис. 1.3).

1.2.1 Инфологическое проектирование базы данных

Основные вопросы: сущность и цель инфологического проектирования баз данных. Основные определения инфологического моделирования.

Инфологическое проектирование основано на анализе **предметной области** и возможных информационных запросов к будущей базе данных, которая должна представлять собой информационную модель этой предметной области. На этом этапе проектирования необходимо выявить основные информационные объекты с их свойствами, а также связями между ними, т. е. инфологическое проектирование связано с определенным структурированием информации. Инфологическое проектирование позволяет задать структуру будущей базы данных на бумаге, еще до работы с компьютером.

Предметная область (ПО) – часть реального мира, подлежащая изучению с целью организации управления и последующей автоматизации.

ПО определена, если известны существующие в ней объекты, их свойства и отношения.

Структура данных – способ объединения нескольких элементов данных в один.

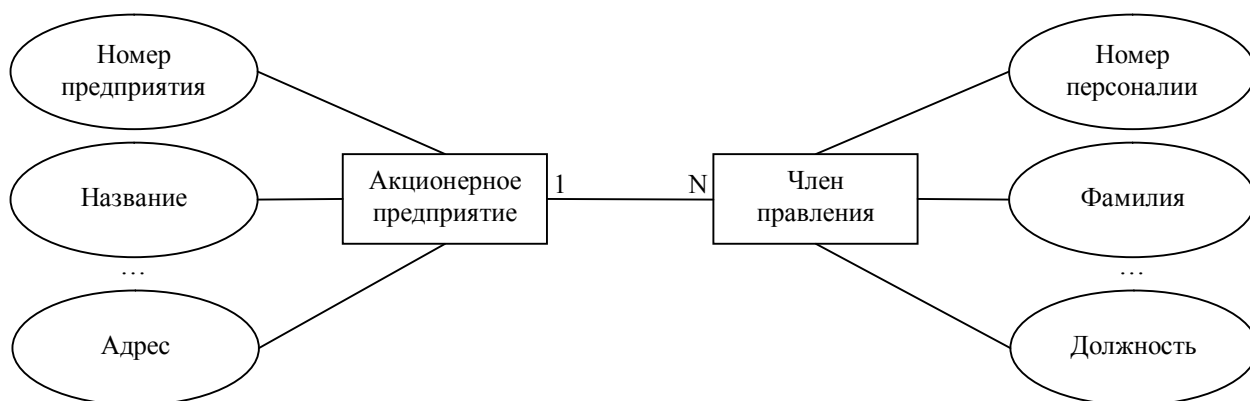


Рис. 1.1. Инфологическая модель

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность – некоторый объект, явление из рассматриваемой предметной области (рис. 1.2). Примеры сущностей: человек, автомобиль, сделка, визит к стоматологу.

Атрибуты – данные, описывающие свойства сущности. Примеры атрибутов: фамилия, цвет, стоимость, дата.

Тип сущности – признак принадлежности к некоторому классу (множеству) объектов (явлений). Тип сущности характеризуется множеством ее атрибутов.

Экземпляр сущности – конкретный объект, принадлежащий определенному классу объектов. Экземпляр сущности определяется значениями ее атрибутов (рис. 1.2).

Связь – ассоциирование двух или более сущностей.



Рис. 1.2. Пример сущности и экземпляра сущности

Для описания инфологической модели используются ER-диаграммы.

1.2.2 Даталогическое проектирование базы данных (реляционная модель данных)

Основные вопросы: сущность даталогического проектирования. Реляционной модели данных: понятие, свойства, основные определения теории реляционного моделирования. Связи между сущностями. Виды связей. Понятие ключа.

Даталогическое проектирование сводится к представлению инфологической модели в терминах выбранной **системы управления базами данных (СУБД)**, т. е. даталогическая модель описывает собственно **данные** (информацию, которая будет записана в памяти компьютера) и связи между данными. Каждая СУБД опирается на определенную **модель данных**. Наиболее популярной в настоящее время является **реляционная модель данных**.

Понятие реляционный (англ. *relation* – отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

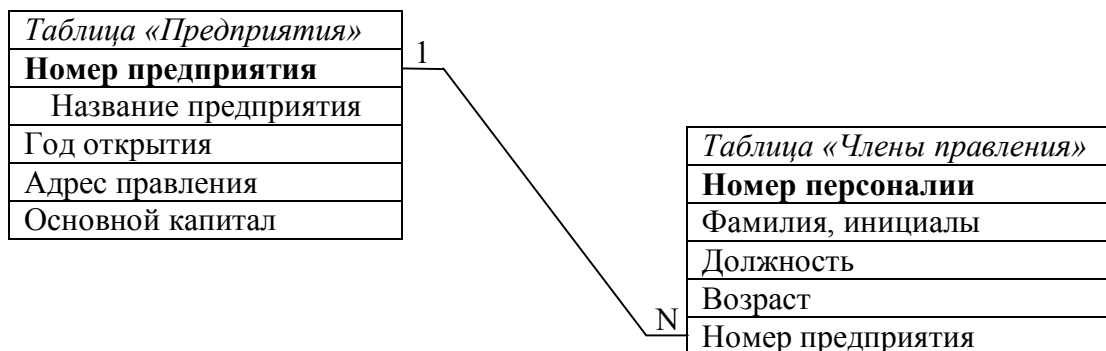


Рис. 1.3. Даталогическая реляционная модель

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы – один элемент данных;
- все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде *таблиц*, строки которых соответствуют *записям*, а столбцы – *полям*.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет *составной ключ*.

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы *внешний ключ* – ключ второй таблицы.

Ключ обеспечивает:

- однозначную идентификацию записей таблицы;
- ускорение выполнения запросов к БД;
- установление связи между отдельными таблицами БД;
- использование ограничений ссылочной целостности.

Связи между сущностями. После определения сущностей следует описать связи (взаимоотношения) между ними.

Связь устанавливается между **экземплярами** сущностей, а не их типами. Для связывания экземпляров сущностей используются их уникальные идентификаторы экземпляров сущности (первичные ключи).

Виды связей:

- один-к-одному;
- один-ко-многим;
- многие-ко-многим.

Вид связи определяется количеством экземпляров сущностей, участвующих в связи с каждой из сторон.

При связи вида «один-к-одному», экземпляр каждой из сущностей может быть связан не более чем с одним экземпляром другой сущности.

При обнаружении связи вида «один-к-одному» следует проанализировать причину ее возникновения. Часто такую связь лучше преобразовать в «один-ко-многим» или уничтожить, объединив две сущности в одну (новая сущность будет содержать атрибуты обеих первоначальных сущностей).

При связи между сущностями вида «многие-ко-многим» каждому экземпляру первой сущности могут быть поставлены в соответствие несколько экземпляров второй сущности (и наоборот).

Реляционная модель баз данных не позволяет реализовать связь «многие-ко-многим». Эта проблема может быть решена введением дополнительной сущности).

Домен атрибута – множество значений, которые атрибут может принимать. Доменом может быть, например, множество допустимых значений даты, диапазон целых чисел или множество текстовых строк. Также это может быть множество цветов и оттенков или список компаний – поставщиков.

При реализации модели на языке конкретной СУБД домены приводятся в соответствие с имеющимися типами данных.

В системах баз данных даталогическая модель предметной области представляет собой *схему базы данных*.

1.2.3 Модели данных

Основные вопросы: понятие модели данных. Виды моделей данных. Структура данных.

Понятие модели данных, является одним из фундаментальных в области баз данных.

Модель данных – представление о предметной области в виде данных и связей между ними.

То есть, **модель данных** – это совокупность взаимосвязанных структур данных и операций над этими структурами.

Понятие “Модель данных” включает три компонента:

- 1) организацию данных (количество и типы объектов модели данных, ограничения на структуру данных);
- 2) множество допустимых операций над данными:
 - операции выборки (поиск);
 - операции модификации (включить, удалить, изменить данные);
- 3) средства обеспечения логической целостности и достоверности данных (ограничения на значения данных и связи), с помощью которых достигается непротиворечивость хранимой информации.

Выбор модели данных зависит от объема информации, сложности решаемых задач и имеющегося технического и программного обеспечения.

Целью логического проектирования является построение логической модели избыточной и непротиворечивой базы данных. При построении логической модели на основе инфологической модели выбирают ту или иную модель данных. Различают иерархические, сетевые и реляционные модели данных.

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево). **Граф** – это совокупность объектов со связями между ними.

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. **Узел** – это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне.

Схематично иерархическая модель данных представлена на рис 4.2.

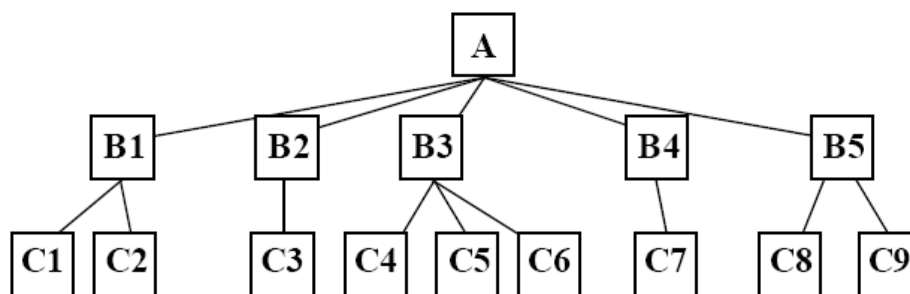


Рис. 4.2. Схема иерархической модели базы данных

В **сетевой структуре** при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом. Сетевую модель можно представить как граф с записями в виде узлов графа и наборами в виде его ребер (рис. 4.3).

Граф является более общей структурой по сравнению с деревом (иерархия), поскольку отдельный узел может иметь произвольное количество родительских узлов, а также произвольное количество подчиненных узлов.

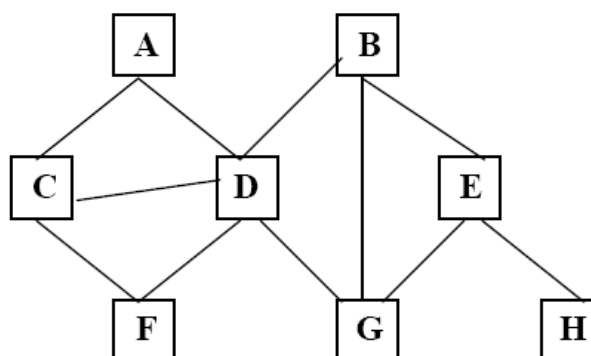


Рис. 4.3. Схема сетевой модели базы данных

Структура [structure] – фиксированное упорядоченное множество объектов и связей между ними.

С понятие структура связаны следующие термины:

– **структура данных [data structure]** – множество элементов данных, объединенных и упорядоченных определенным образом;

– **структура информационной базы [information support structure]** – упорядоченная по определенным правилам совокупность подмножеств записей **информационных элементов**, образующих **информационную базу** и необходимых для реализации функций автоматизированной системы,

– **структура базы данных [DB structure]** – принцип или порядок организации записей в **базе данных** и связей между ними. Структуру БД принято рассматривать на разных уровнях **абстракции** (представления) и, в частности: концептуальном (с позиции администратора предприятия), реализации или внешнем (с позиций конечного пользователя и прикладного программиста) и физическом или внутреннем (с позиций системного аналитика и системного программиста). Соответственно этим уровням различают **концептуальную, внешнюю и физическую** модели и/или схемы организации данных,

– **абстракция [abstraction]** – использование для описания или представления общих свойств объекта без конкретной его реализации (например, **типов объектов** – "читатель", "фирма", "автомобиль", но не конкретно названных читателей, фирм, марок автомобилей и т.п.);

– **абстрактная структура данных [abstract logic design]** – структура данных, определенная функционально посредством выполняемых на ней операций. Такая структура не связана с поименованными типами объектов.

1.3 Проектирование реляционных баз данных с использованием нормализации

Исходной точкой является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих лучшими свойствами. Процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером набора ограничений является ограничение первой нормальной формы – значения всех атрибутов отношения атомарны.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);

- нормальная форма Бойса-Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Основными свойствами нормальных форм можно назвать следующие:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

В основе процесса проектирования лежит метод нормализации, декомпозиция отношения, находящегося в предыдущей нормальной форме, в два или более отношений, удовлетворяющих требованиям следующей нормальной формы.

Наиболее важные на практике нормальные формы отношений основываются на фундаментальном в теории реляционных баз данных понятии функциональной зависимости.

Функциональная зависимость.

Определение: Если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X , если в любой момент времени каждому значению X соответствует ровно одно значение Y .

Функциональная зависимость обозначается $X \rightarrow Y$. Отметим, что X и Y могут представлять собой не только единичные атрибуты, но и группы, составленные из нескольких атрибутов одного отношения.

Функциональные зависимости представляют собой связи типа «один ко многим», существующие внутри отношения.

Некоторые функциональные зависимости могут быть нежелательны.

Определение:

Избыточная функциональная зависимость - зависимость, заключающая в себе такую информацию, которая может быть получена на основе других зависимостей, имеющихся в базе данных.

Корректной считается такая схема базы данных, в которой отсутствуют избыточные функциональные зависимости. В противном случае приходится прибегать к процедуре **декомпозиции (разложения)** имеющегося множества отношений. При этом порождаемое множество содержит большее число отношений, которые являются проекциями отношений исходного множества. *Обратимый* пошаговый процесс замены

данной совокупности отношений другой схемой с устранением избыточных функциональных зависимостей называется **нормализацией**.

Нормализация – это процесс приведения таблиц к неизбыточной форме посредством операции проекции (расщепление одной таблицы на несколько).

Отношение называется нормализованным или приведенным к **первой нормальной форме (1НФ)**, если все его атрибуты простые или атомарные (неделимые). Отношение, находящееся в первой нормальной форме, будет иметь следующие свойства:

- в отношении нет одинаковых кортежей;
- кортежи не упорядочены;
- атрибуты не упорядочены и различаются по наименованиям;
- все значения атрибутов атомарные.

Простой атрибут – атрибут, значения которого атомарны (неделимы).

Сложный атрибут – получается соединением нескольких атомарных атрибутов, которые могут быть определены на одном или разных доменах.

Взаимно независимые атрибуты - два или более атрибута взаимно независимы, если ни один из этих атрибутов не является функционально зависимым от других.

Первичный ключ – атрибут или набор атрибутов, который используется в данном отношении для идентификации (различения) кортежей (записей). Значение ключа, следовательно, уникально для каждой записи.

Возможный ключ – атрибут или набор атрибутов, который может быть использован для данного отношения в качестве первичного ключа.

Детерминант – любой атрибут, от которого полностью функционально зависит некоторый другой атрибут.

Определение 2НФ.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Полная функциональная зависимость присутствует, когда неключевой атрибут функционально полно зависит от составного ключа если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

Алгоритм приведения ко 2НФ:

Исходное отношение: $R(K_1, K_2, A_1, \dots, A_n, B_1, \dots, B_m)$.

Ключ: $\{K_1, K_2\}$.

Функциональные зависимости:

$\{K_1, K_2\} \rightarrow \{A_1, \dots, A_n, B_1, \dots, B_m\}$ – зависимость всех атрибутов от ключа отношения.

$\{K_1\} \rightarrow \{A_1, \dots, A_n\}$ – зависимость некоторых атрибутов от части сложного ключа.

Декомпозированные отношения:

$R_1(K_1, K_2, B_1, \dots, B_m) \rightarrow$ остаток от исходного отношения. Ключ $\{K_1, K_2\}$.

$R_2(K_1, A_1, \dots, A_n) \rightarrow$ атрибуты, вынесенные из исходного отношения вместе с частью сложного ключа. Ключ K_1 .

Определение 3НФ.

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Алгоритм приведения к 3НФ:

Исходное отношение: $R(K, A_1, \dots, A_n, B_1, \dots, B_m)$.

Ключ: K .

Функциональные зависимости:

$K \rightarrow \{A_1, \dots, A_n, B_1, \dots, B_m\}$ – зависимость всех атрибутов от ключа отношения.

$\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_m\}$ – зависимость некоторых неключевых атрибутов от других неключевых атрибутов.

Декомпозированные отношения:

$R_1(K, A_1, \dots, A_n) \rightarrow$ остаток от исходного отношения. Ключ K .

$R_2(A_1, \dots, A_n, B_1, \dots, B_m) \rightarrow$ атрибуты, вынесенные из исходного отношения вместе с детерминантом функциональной зависимости. Ключ $\{A_1, \dots, A_n\}$.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных: Учеб. для ВУЗов. – М.: Высш. шк., 1987. – 248 с.
2. Наумов А.Н. и др. Системы управления базами данных и знаний. – М.: Финансы и статистика, 1991. – 315 с.
3. Ульман Дж. Основы систем баз данных. - М.: Финансы и статистика, 1983. - 334 с.
4. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. – М.: Мир, 1991. – 276 с.
5. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх. – М.: Нолидж, 2000. – 351 с.
6. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998. – 176 с.
7. Конноли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. Теория и практика. – 2-е изд. – М.: Изд. дом Вильямс, 2000. – 1120 с.
8. Базы данных: Учебник для вузов / Под ред. А.Д. Хомоненко. – СПб.: Корона принт, 2000. – 416 с.
9. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: учебное пособие. – СПб.: Питер, 2000. – 384 с.
10. Хансен Г., Хансен Дж. Базы данных: разработка и управление. – М.: Бином, 1999. – 560 с.
11. Александровский А. Delphi 5.0. Разработка корпоративных приложений. – М.: ДМК, 2000. – 512 с.
12. Реализация баз данных Microsoft SQL Server 7.0. Учебный курс: официальное пособие для самостоятельной подготовки. – М.: Русская редакция, 2000. – 528 с.
13. Канту Марко и др. Delphi. Руководство разработчика. – К.: Век; М.: ЭНТРОП; М.: ДЕСС, 1999. – 752 с.
14. Попов А.А. Создание приложений для FoxPro 2.5/2.6 в DOS и Windows. – М.: ДЕСС, 1999. – 672 с.
15. Макаширпов С. Программирование баз данных на Visual Basic 5.0 в примерах. – СПб.: Питер, 1997. – 256 с.
16. Баженова И.Ю. Visual FoxPro 5.0. Объектно-ориентированные средства программирования. – М.: Бином, 1997. – 320 с.

17. Каратыгин С.А., Тихонов А.Ф., Тихонова Л.Н. Работа в Visual FoxPro на примерах. - М.: БИНОМ, 1995. - 512 с.
18. Баженова И.Ю. ORACLE 8/8i. Уроки программирования. – М.: Диалог-МИФИ, 2000. – 624 с.
19. Фридман А.Л. Основы объектно-ориентированной разработки программных систем. – М.: Финансы и статистика, 2000. – 192 с.
20. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – 672 с.
21. Ван Хейк Бернард. JDBC: Java и базы знаний. – М.: ЛОРИ, 1999. – 320 с.