

1. Введение в технологию баз данных

Какую бы отрасль знаний мы ни взяли, для того, чтобы осветить ее всесторонне, не хватит никаких разумных рамок одной публикации. Естественно, это касается технологии баз данных (БД) и настоящей книги. Как следует из названия, она охватывает лишь малую (но не последнюю по значению) часть проблематики технологии БД. Тем не менее, для полноты картины нельзя хотя бы кратко не упомянуть об особенностях этой технологии в целом и не указать то место, которое в ней занимает собственно моделирование данных.

Прежде всего, следует сказать о назначении технологии БД. Большая Советская энциклопедия (БСЭ) так определяет понятие «технология»: «Технология (от греч. *techne* – искусство, мастерство, умение) – это совокупность приемов и способов получения, обработки или переработки сырья, материалов, полуфабрикатов или изделий, осуществляемых в различных отраслях промышленности, в строительстве и т. д.; научная дисциплина, разрабатывающая и совершенствующая такие приемы и способы». В частности, информационная технология предлагает аппаратно-программные системы для управления информацией, а соответствующие научные дисциплины обеспечивают технологическую базу для их создания. Соответственно, технология баз данных – это технология, регламентирующая процесс создания и эксплуатации информационных систем, в которых для долговременного хранения информации используется база данных, а управляет ею некоторая система управления базами данных (СУБД).

В каких же случаях имеет смысл создавать подобные системы и использовать при этом технологию БД? Это, безусловно, следует делать при выполнении всех перечисленных ниже условий.

1. Вы нуждаетесь в долговременном хранении информации о некоем фрагменте реального или идеального мира (в дальнейшем мы будем называть этот фрагмент предметной областью и использовать при этом аббревиатуру ПрО).

2. Система создается для удовлетворения информационных потребностей нескольких людей, в том числе тех, кто в силу разных причин не в состоянии воспринимать этот мир с помощью своих органов чувств.

3. Вам необходима не «бесформенная куча», извлечение информации из которой будет отнимать много времени и усилий, а хорошо структурированное хранилище, в котором каждый бит информации «лежит на своей полке» и непротиворечиво связан с остальными.

4. Моделируемый в системе мир динамичен, и поэтому специальные пользователи должны вносить изменения в его информационное описание.

5. Большая часть пользователей будет удовлетворять свои потребности в информации об этом мире, обращаясь к информационной системе, в которой должны обеспечиваться специальные механизмы, способствующие удобному извлечению и преобразованию информации.

6. Производная информация должна получаться из первичной, введенной непосредственно людьми, с помощью простых универсальных преобразований, и этот процесс не должен сопровождаться разработкой специальных алгоритмов.

7. Информация вводится, хранится и предъявляется преимущественно в алфавитно-цифровой форме.

Вообще, довольно затруднительно указать список критериев, при наличии которых следует использовать технологию БД (тем более для людей, с ней незнакомых). Ситуация осложняется тем, что часто она применяется в сочетании с другими информационными технологиями. Одно можно сказать почти наверняка – по окончании освоения данной книги скрупулезный читатель сможет без труда решить проблему применимости технологии БД.

Далее мы кратко охарактеризуем особенности систем БД. Впоследствии мы обязательно рассмотрим эти вопросы подробнее.

Информационную технологию баз данных отличает использование СУБД той или иной модели данных (МД) – концепции представления в ней объектов мира и их взаимосвязей. Развитые модели, обеспечивающие адекватное представление данных, относящихся к различным предметным областям, в конечном итоге определяют степень соответствия СУБД целям их создания.

Здесь следует сделать замечание о нетрадиционном использовании термина «модель» в технологии БД. Действительно, в большинстве отраслей знаний (в том числе в математике) «комплекс взглядов, представлений, идей, дающий целостное представление о закономерностях и существующих связях определенной области действительности» [БСЭ] принято называть теорией, методологией моделирования. А модель в таком случае это не что иное, как идеализированное в рамках соответствующей теории представление области действительности. Как видим, в технологии БД произошла замена смысла термина «модель», в ней это слово обозначает теорию моделирования, а результат моделирования есть не что иное, как БД. Этот казус сложился исторически во время появления первых моделей БД и с тех пор имеет место в большинстве публикаций. Никто из авторов не берет на себя ответственность исправить ситуацию. Последуем их примеру и мы. Но чтобы ваше сознание не столкнулось с явным противоречием, следует иметь в виду только что сказанное.

Основное назначение МД – обеспечение процесса интерпретации (осмысления) данных и превращения их в информацию. Для этого в каждой модели данных имеются следующие компоненты:

- правила порождения допустимых **структур** данных;
- правила порождения **ограничений целостности** данных;
- набор **операций** над данными.

На основании правил определяется схема БД (основные понятия предметной области и их взаимосвязи), а операции необходимы, чтобы первоначально наполнить схему непротиворечивыми данными и поддерживать их в дальнейшем в адекватном предметной области состоянии, а также обеспечивать получение необходимой информации.

Так, например, правила структуризации для реляционной модели определяют БД как совокупность отношений (таблиц). Каждое отношение соответствует одному типу объектов (смысл их представляется именем отношения) и задает одинаковый для всех объектов этого типа набор атрибутов (столбцов). В схеме отношения для каждого атрибута определено его имя, представленное в таблице в заголовке столбца. Таким образом, структуру отношения может олицетворять поименованная шапка пустой таблицы.

Наряду с этим реляционную схему БД составляют условия, которым должны удовлетворять данные, заполняющие тела таблиц. Иногда их называют бизнес-правилами или, чаще, ограничениями целостности. Последний термин связан с поддержанием БД в целостном, непротиворечивом состоянии, максимально соответствующем состоянию моделируемой предметной области. Окончательно эту проблему не решат никакие средства автоматизации. Только человек в состоянии определить, какие факты об объектах предметной области являются истинными. Но частично помочь ему в этом и призваны ограничения целостности, декларированные в схеме БД. Так, ограничение целостности, говорящее о том, что значениями атрибута *Рост* отношения *ЧЕЛОВЕК* могут быть целые числа в интервале от 50 до 300, предостережет от многих ошибок ввода. Задачей человека все же остается указание в каждом случае точного, известного ему значения из этого интервала.

До тех пор, пока не определена как минимум структура данных, система не позволит вводить эти данные. Это происходит потому, что при создании, изменении и удалении данных обязательно должно быть указано, элементом какой структуры они являются. Так при определении в системе нового человека нужно с помощью

соответствующей операции создать строку именно в таблице *ЧЕЛОВЕК*, причем значения в строке должны четко соответствовать шапке таблицы.

Задачей же системы является запоминание соответствия данному его интерпретации (имени отношения и имени атрибута) и совместное их предоставление по запросам пользователей. Именно интерпретация и превращает значения данных в информацию, которая собственно и интересует пользователя. Тела таблиц без наименований и шапок – просто бессмысленная совокупность символов машинного алфавита.

Эффективность применения человеком технологии БД существенно определяется тем, насколько понятны ему сущность данных и представляемой ими информации. Модели данных служат средством достижения такого понимания.

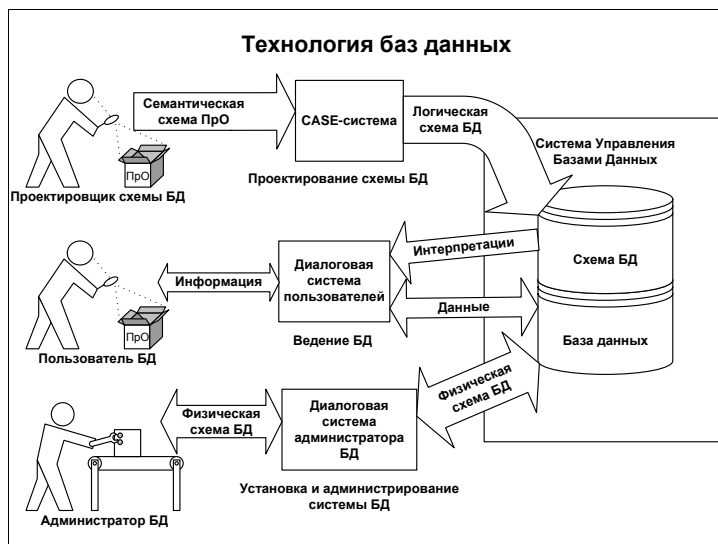


Схема иллюстрирует основные процессы технологии БД и ее участников, как одушевленных, так и неодушевленных.

К необязательным, но часто важным процессам создания информационных систем (ИС) по этой технологии относятся процессы разработки специализированного программного обеспечения (ПО), реализующего специфическую для предметной области (ПрО) бизнес-логику. Эти процессы не рассматриваются в данном курсе по двум причинам. Во-первых, как уже отмечалось, они являются необязательными, если ведение БД осуществляется пользователями с помощью универсальных диалоговых инструментов, представляющих БД как совокупность таблиц. Все манипуляции с данными в них сводятся к созданию, изменению, удалению и просмотру строк этих таблиц. Второй причиной, по которой мы не рассматриваем эти процессы, является тот факт, что они по большому счету относятся к технологии создания ИС на базе универсальных языков программирования, таких как, Visual Basic, Java, C++, C#, Delphi и т.д. Именно с применением этой технологии в случае надобности создается специализированная диалоговая система пользователей, показанная на схеме.

Стилизованные человечки обозначают роли, которые играют люди в системе БД. В реальной системе возможны ситуации, когда одну и ту же роль исполняет несколько человек, образуя соответствующую функциональную группу (например, отдел администрирования БД). С другой стороны, в небольшой автономной системе для личных нужд все роли играет один человек.

Прямоугольниками на схеме представлены программно-аппаратные средства, управляющие данными в оперативной и долговременной памяти компьютера. Эти же инструменты обеспечивают взаимодействие всех участников технологии БД (как одушевленных, так и неодушевленных).

Дуги с одной или двумя стрелками на концах олицетворяют как раз информационные потоки между людьми и программами, а также между программами. Стрелка указывает направление, в котором осуществляется передача данных.

И, наконец, цилиндр внутри прямоугольника «Система Управления Базами Данных» обозначает все, что хранится в долговременной памяти компьютера (как правило, это один или несколько жестких дисков – винчестеров) и управляется СУБД. Пометки «Схема БД» и «База данных» характеризуют основные обязательные компоненты этого хранилища.

На рисунке отражены самые важные процессы технологии БД. Прежде всего, это самый первый, самый важный и обязательный процесс – проектирование схемы БД. Как

правило, это дело поручают профессионалам в этой технологии. От того, насколько качественной будет схема БД, существенно зависят простота и легкость осуществления всех остальных процессов и успех всей системы в целом. Именно поэтому мы сконцентрируем свое внимание прежде всего на изложении теории и практики проектирования схем БД.

В двух словах этот процесс состоит в выполнении следующих действий. Проектировщик схемы БД изучает самостоятельно и с помощью экспертов процессы ПрО, структуры и закономерности ее информационного описания и формализует полученные знания в виде семантической схемы ПрО. Эта схема строится с использованием той или иной семантической модели данных, концепция которой максимально близка человеческому восприятию мира.

Правильный выбор подходящей семантической модели может существенно облегчить задачу проектировщика. Решающее значение при этом имеют два фактора:

- мощность языка, обеспечивающая богатые выразительные способности, которые позволят в полной мере описать семантику ПрО;
- поддержка модели CASE-средством (CASE – Computer Aided Software/System Engineering – программная система для проектирования и реализации других программных систем), обеспечивающим автоматическую трансляцию схемы ПрО на язык СУБД-ориентированной модели данных или даже на язык логической модели конкретной СУБД.

Чем выразительней модель, тем больше особенностей ПрО в ней можно отразить. В идеале семантическая схема должна вобрать в себя все закономерности информационного описания ПрО. При этом не будет необходимости прибегать к помощи других моделей и языков, в том числе, естественного.

При наличии специализированного программного обеспечения в лице CASE-системы можно, во-первых, просто и естественно для проектировщика осуществить ввод семантической схемы, а, во-вторых, автоматически перевести ее на язык СУБД. В идеале этот перевод не должен сопровождаться потерей информации, и каждый элемент семантической схемы должен найти свое отражение в логической модели СУБД (в крайнем случае – в процедурной форме).

При отсутствии подходящего CASE-инструмента необходимую трансляцию схемы ПрО проектировщик осуществляет вручную, на бумаге. Для этого он должен знать методику этого перевода – набор правил преобразования структур и ограничений целостности из семантической модели в логическую модель СУБД. Часто на практике лучше вообще не использовать CASE-систем (вооружившись при этом мощной семантической моделью), чем применять CASE-средства со слабой моделью и примитивными правилами трансформации.

В дальнейшем мы рассмотрим ряд семантических моделей и познакомимся с несколькими методиками трансляции схем из них в реляционную модель, поддерживаемую большинством современных СУБД.

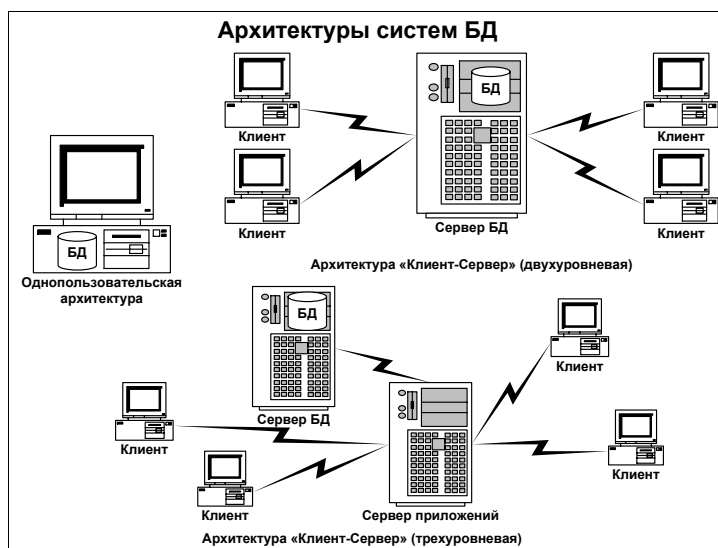
Если рассматривать жизненный цикл системы БД в хронологическом порядке, после разработки схемы БД и диалоговой системы пользователей (напомним, что эта задача остается за бортом нашего рассмотрения) администратор системы осуществляет ее установку на оборудовании организации-заказчика. В дальнейшем он с помощью специальных инструментов, предоставляемых разработчиками СУБД, администрирует систему на уровне физической схемы БД – схемы хранения данных и их интерпретаций на диске. Основной обязанностью администратора БД является обеспечение эффективной бесперебойной работы БД, не взирая на возможный выход из строя аппаратуры и программ. Рамки этой книги не позволяют нам обсудить процесс администрирования БД.

И, наконец, мы подошли к описанию основного процесса системы БД (ради которого она и создавалась) – ее применения пользователями. Мы уже говорили о том, что БД – это хранилище актуальной информации о ПрО. Поэтому ее пользователи могут

выступать в двух ролях – «писателей» и «читателей» (аналогично ролям, сопровождаемым бумажное общение на естественном языке). Первые обязаны отражать в БД изменения ПрО, причем максимально оперативно и точно. Вторые удовлетворяют свои информационные потребности, обращаясь к системе БД.

И так этот процесс продолжается все время эксплуатации системы. Естественно, что он занимает основную часть жизненного цикла системы БД и существенно зависит от качества выполнения остальных задач, упоминавшихся ранее.

Следует отметить, что вся сложность системы скрыта от пользователей за простыми и естественными для них диалоговыми инструментами. Поэтому пользовательский процесс не нуждается в серьезном рассмотрении, хотя мы познакомимся с некоторыми языками манипулирования данными, но они рассчитаны скорее на подготовленных пользователей.



Базы данных изначально использовались в больших корпорациях и крупных организациях как основа для больших систем обработки транзакций. (Ранее термин «транзакция» применяли для обозначения банковской операции перевода денег, теперь он используется в более широком смысле – любая передача данных в информационных системах. В технологии БД у этого слова есть свой особый смысл, но об этом мы поговорим чуть позже.) Этими системами управляли большие ЭВМ (мэйнфреймы), представлявшие собой единый компьютер, к которому могли подключаться многочисленные терминалы пользователей. На этом компьютере запускалась система БД, которая одновременно вела диалог со всеми пользователями через терминалы. Эта архитектура систем БД в настоящее время встречается очень редко (как и сами мэйнфреймы). Поскольку в ней используется один компьютер, она напоминает однопользовательскую архитектуру.

По мере того, как популярность завоевывали микрокомпьютеры, технология БД также мигрировала в этом направлении и стала использоваться для однопользовательских, персональных приложений. В **однопользовательской архитектуре** вся система БД управляется одним компьютером, на котором расположены и БД, и СУБД, и диалоговая система пользователя.

Затем, когда микрокомпьютеры начали объединять в рабочие группы, технология БД была модифицирована с учетом этой тенденции. Многопользовательская архитектура, применяемая в локальных сетях, значительно отличается от многопользовательской архитектуры, применявшейся на мэйнфреймах. В случае последних в обработке приложения БД участвовал только один процессор, а в локальных сетях для этого могут использоваться несколько процессоров. Поскольку эта ситуация, помимо очевидной выгоды (большая производительность), влечет за собой и новые трудности (координация действий независимых процессоров), возник новый стиль многопользовательской обработки БД, называемый **клиент-серверной архитектурой БД**.

В отличие от однопользовательской архитектуры (часто называемой еще автономной), в которой имеется только один компьютер, клиент-серверная система состоит из множества компьютеров, объединенных в сеть. Одни компьютеры, называемые клиентами, занимаются обработкой прикладных программ, которые ведут диалоги с пользователями. Другие компьютеры, называемые серверами, занимаются обработкой БД.

Тип компьютеров может быть совершенно разным. Теоретически, в роли клиентских компьютеров могут выступать мэйнфреймы или микрокомпьютеры. Однако из соображений стоимости функции клиентов почти всегда выполняют

микрокомпьютеры. Точно так же в роли сервера может выступать компьютер любого типа, но по экономическим причинам функции сервера чаще всего также выполняет микрокомпьютер. Клиенты и серверы соединяются в локальную или глобальную сеть.

Часто система БД имеет один сервер, хотя это не всегда так. Различные серверы могут управлять разными БД. Если обработкой БД занимаются несколько серверов, то чтобы такая система могла считаться клиент-серверной, каждый из серверов должен управлять своей БД. Если же два сервера обрабатывают одну БД, такая система называется системой распределенной обработки БД.

Наконец, в настоящее время БД используются в приложениях для Интернета и интрасетей. Во многих приложениях БД, применяющих интернет-технологии, используется **трехуровневая архитектура**, включающая сервер БД, web-сервер и браузер. Каждый из этих уровней может работать под управлением своей операционной системы на отдельном компьютере. Так же как и двухуровневую, трехуровневую архитектуру программных продуктов технологии БД можно реализовать и на одном компьютере.

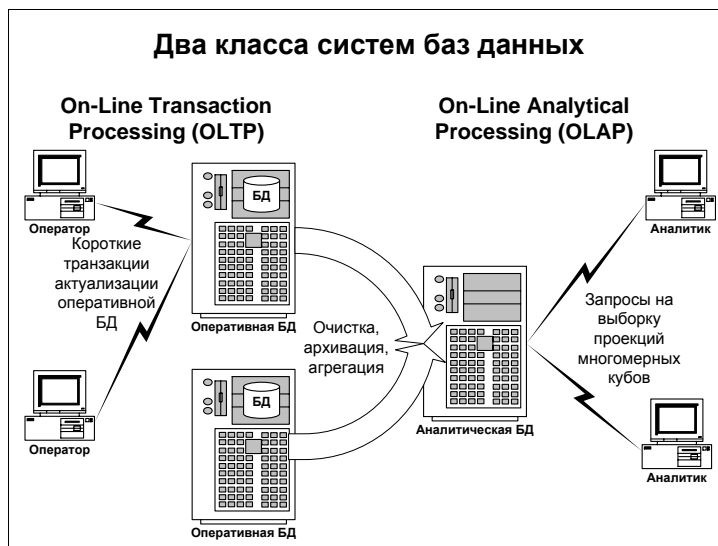
Через интерфейс между web-сервером и сервером БД передаются SQL-операторы и реляционные данные. Через интерфейс между web-сервером и браузером передаются web-страницы, клиентский код и данные.

Назначение сервера БД – нести на себе СУБД, которая выполняет SQL-операторы и осуществляет управление БД. Web-сервер выполняет три основные функции. Во-первых, он принимает запросы и генерирует отклики в формате HTTP. Во-вторых, на web-сервере располагается и выполняется код, реализующий бизнес-логику приложения. В-третьих, по мере необходимости он генерирует SQL-запросы к БД и принимает отклики на них от сервера БД. Браузер в приложениях БД также выполняет три функции. Во-первых, браузер является HTTP-клиентом: он генерирует запросы на страницы и на различные действия. Во-вторых, он содержит сценарный интерпретатор, позволяющий запускать сценарии на клиентской машине. Наконец, браузер материализует представления, преобразуя HTML или другой язык разметки в содержимое окна на экране.

На схеме для промежуточного уровня использован термин «сервер приложений». Это понятие является более общим по отношению к понятию «web-сервер». В действительности трехуровневая архитектура систем БД может быть реализована не только по web-технологии (хотя чаще в последнее время встречается именно она). Кроме web-сервера роль базового программного обеспечения для сервера приложений могут играть другие специально предназначенные для этого программные продукты – мониторы обработки транзакций (Transaction Processing Monitor – TPM).

Мониторы транзакций – программные системы, обеспечивающие эффективное управление информационно-вычислительными ресурсами в распределенной системе. Они представляют собой гибкую, открытую среду для разработки и управления мобильными приложениями, ориентированными на оперативную обработку распределенных транзакций. В числе важнейших характеристик TPM – масштабируемость, поддержка функциональной полноты и целостности приложений, достижение максимальной производительности обработки данных при невысоких стоимостных показателях, поддержка целостности данных в гетерогенной среде. TPM опираются на трехуровневую модель «клиент-сервер».

На современном рынке мониторов транзакций основными «действующими лицами» являются такие системы, как ACMS (DEC), CICS (IBM), TOP END (NCR), PATHWAY (Tandem), ENCINA (Transarc), TUXEDO System (USL).



Последующий материал требует пояснения термина «транзакция». Транзакция (англ. transaction) – это логически неделимая единица работы с данными (обычно включающая несколько операций над БД), которая должна выполняться атомарным образом. Система БД гарантирует, что либо в БД будут произведены изменения, продуцируемые всеми операциями транзакции, либо не будет выполнено ни одно из них, если до завершения транзакции в системе произойдет сбой. Таким образом, транзакция или полностью выполняется, или полностью отменяется (как будто она вообще не выполнялась.)

Первоначально СУБД проектировались для управления большим потоком транзакций, каждая из которых сопровождалась внесением незначительных изменений в оперативные данные предприятия – т.е. в данные, которые предприятие обрабатывало в процессе своей повседневной деятельности. Системы подобного типа называются системами **оперативной обработки транзакций**, или OLTP-системами (англ. Online Transaction Processing – **OLTP**). Размер БД для OLTP-систем может варьироваться от совсем малого, всего в несколько мегабайт, до среднего, порядка нескольких гигабайт, и дальше вплоть до очень большого, на уровне нескольких терабайт или даже петабайт.

Все дальнейшее изложение будет посвящено применению технологии БД именно для этого класса систем, поэтому второй класс охарактеризуем здесь более подробно.

Лицам, ответственным за принятие корпоративных решений, необходимо иметь доступ ко всем данным организации независимо от их расположения. Для выполнения полного анализа деятельности организации, определения ее бизнес-показателей, выяснения характеристик существующего спроса и тенденций его изменения необходимо иметь доступ не только к текущим данным, но и к ранее накопленным (историческим) данным.

Для упрощения подобного анализа была разработана концепция хранилища данных. Предполагается, что такое хранилище (своеобразная аналитическая БД) содержит сведения, поступающие из самых разных источников данных, в том числе, оперативных БД, а также различные накопительные и сводные данные. Лицам, ответственным за принятие корпоративных решений, необходимо иметь мощные инструменты анализа накопленных в хранилище данных. Основными средствами анализа в последние годы стали инструменты **оперативной аналитической обработки** (англ. Online Analytical Processing – **OLAP**). Термин OLAP определяет архитектуру, которая поддерживает сложные аналитические приложения. Большинство OLAP-приложений создается на

основе специализированных многомерных СУБД с ограниченным по структуре набором данных и настраиваемым пользовательским интерфейсом.

В OLAP-технологии серверы БД для хранения данных и связей между ними используют многомерные структуры. Многомерные структуры лучше всего представлять как многомерные кубы данных. В диалоговых средствах OLAP многомерный куб визуально представляется в виде динамично изменяемых таблиц. Данные откладываются по двум осям, которыми являются строки и столбцы. Иногда используется третья ось, значениям которой соответствуют закладки. В таком случае на каждой закладке представлена своя таблица значений, иногда называемая слоем.

На каждой оси может быть показано одно или несколько измерений. Когда на одной оси показываются два или более измерений куба, то приводятся все возможные комбинации данных одного измерения с данными другого. Ячейки куба представляют меры куба – значения какого-то арифметического показателя (измеряемого в деньгах, килограммах, метрах и т.д.). Возможные значения данных в измерении называются членами, они представляют собой различные значения этого измерения. Члены одного измерения могут образовывать иерархию (например, иерархия временных периодов – день, неделя, месяц и т.д.).

Диалоговые OLAP-инструменты позволяют динамично строить всевозможные слои и проекции куба с использованием техник «drag and drop» (перетаскивание элементов диалога с одного места экрана на другое) и «drill-down and drill-up» (перемещение по уровням иерархии членов измерения с агрегированного на более детальный и наоборот). Тем самым легко и быстро без помощи программистов пользователь может получать различные интересующие его взгляды на объект анализа.

Специфика решаемых в OLTP- и OLAP-системах задач настолько существенна, что попытка решить их с использованием одной и той же БД и СУБД (а такая возможность, как правило, предоставляется) скорее всего окажется ужасно неэффективной. Поэтому имеет смысл в случае необходимости применять в каждом случае специальные БД и СУБД.



Целью данного курса лекций является глубокое усвоение знаний по основам моделирования данных в технологии БД, изучение основных широко применяемых на практике моделей данных и освоение методик проектирования схем БД.

Логика и последовательность изложения материала вытекает из поставленной цели и той цепочки преобразований, которые проходят представления о ПрО на пути от объективной реальности к БД, хранящейся в памяти компьютера. Этот многослойный «пирог» представлений изображен в левой части приведенной схемы.

Справа от него в горизонтальных слоях указаны соответствующие каждому уровню представлений исследующая его область науки и техники, подходящее программное обеспечение и конкретные модели данных. Если для уровня представлений имеется формальная система – высшая форма теории, она указана в правой колонке.

ПрО представляет собой некоторый интересующий заказчика системы БД фрагмент реального или идеального мира. Первоначальные представления о ПрО возникают **в голове у проектировщика БД** на основании его личного знакомства с ней, а также на основании сведений, полученных у экспертов по ПрО или из других источников. Эти исключительно неформализованные представления составляют зрительные, слуховые, обонятельные, осязательные образы.

Знакомясь с различными моделями данных, проектировщик в каждом случае вооружается набором формальных понятий модели. Его задачей является перевод своего неформального восприятия ПрО на язык конкретной модели данных, используя ее формальный понятийный набор.

Ближе всего к человеческим принципам миропонимания находятся так называемые семантические модели данных. Поэтому указанная задача в них реализуется проще всего. В результате получается **семантическая схема** ПрО – ее первое формальное представление. Но поскольку не существует программных систем для создания БД (СУБД), поддерживающих непосредственно семантические модели, а БД – это конечная цель в технологии БД, мы не можем этим ограничиться.

Модели данных конкретных СУБД, хотя и отличаются друг от друга, часто имеют много общих особенностей, которые образуют так называемые СУБД-ориентированные модели данных. В частности, к ним относится самая популярная в настоящее время реляционная модель. Если в качестве СУБД вы выбрали одну из реляционных СУБД, вам предстоит построить реляционную схему своей ПрО, которая является одной из разновидностей **СУБД-ориентированных схем**.

Каждая СУБД предлагает свой диалект СУБД-ориентированной модели данных, и поэтому в конечном итоге именно на нем необходимо представить схему ПрО. Здесь следует заметить, что каждая СУБД имеет не одну, а две модели данных – логическую и физическую. Первая из них обращена в сторону человека (проектировщика или пользователя БД) и предлагает языковые и диалоговые инструменты общения именно для него. Собственно, вторую модель СУБД (физическую) редко кто, кроме разработчиков этой СУБД, и знает, поскольку обращена она к средствам хранения данных в оперативной и внешней памяти.

Естественно, что последние модели определяют самые низкоуровневые представления ПрО – **логическую и физическую схемы БД**. От структурных понятий этих моделей не следует ожидать простоты восприятий (это доступно лишь профессионалам), в них все ориентировано на эффективное (по памяти и по времени отклика на запросы) хранение данных.

Структура курса

1. Введение в технологию баз данных
2. Основные понятия
 - 2.1. Данные и модели данных
 - 2.2. Структуры
 - 2.3. Ограничения целостности
 - 2.4. Операции
 - 2.5. Демонстрационная предметная область
3. Семантические модели данных
 - 3.1. Принципы семантического моделирования
 - 3.2. Модель данных «Сущность-Связь» (ER-модель)
 - 3.3. Модель данных «Сущность-Связь-Отображение» (ERM-модель)
4. СУБД-ориентированные модели данных
 - 4.1. Обзор СУБД-ориентированных моделей данных
 - 4.2. Реляционная модель данных
 - ...
 - 4.2.5. Теория реляционных БД и классическая методика проектирования реляционных схем БД
5. Семантическая методика проектирования реляционных схем БД

В заключительной части введения анонсируем предлагаемый вниманию читателя материал. Структура курса отражает его логику.

В главе «Основные понятия» мы введем основные термины моделирования данных, определим понятие «модель данных» и ее составные части – правила структуризации данных и задания ограничений целостности, а также множество операций над данными. В последующих главах анализ конкретных моделей будет происходить по схеме «структуры – ограничения целостности – операции».

В главе «Семантические модели данных» мы познакомимся с ER-моделью и ERM-моделью, а также методикой семантического моделирования для ERM-модели.

Глава «СУБД-ориентированные модели данных» почти полностью посвящена замечательной во многих аспектах, широко распространенной в настоящее время реляционной модели. В заключительном параграфе главы мы познакомимся с классической методикой проектирования реляционных схем БД. Она отличается тем, что в ней в явном виде отсутствует промежуточное представление ПрО в виде семантической схемы.

И, наконец, квинтэссенция курса – глава «Семантическая методика проектирования реляционных схем БД», повествующая о том, как на практике следует решать задачу проектирования схем БД. Для ее глубокого усвоения понадобится практически весь предыдущий материал книги, поэтому к этому моменту чтения лучше подойти с надежной непротиворечивой системой знаний по моделям данных в голове.

Вопросы и задания к главе 1

1. Укажите отличительные особенности технологии БД, выделяющие ее среди технологий вообще и информационных технологий, в частности.
2. Каковы предпосылки использования технологии БД?
3. В чем особенности термина «модель» в технологии БД?
4. Каково основное назначение моделей данных?
5. Перечислите составные части любой модели данных.
6. Каково назначение структур данных, ограничений целостности и операций над данными?
7. Укажите основные процессы в системах БД, кто и какие задачи в них реализует.
8. Для чего предназначены CASE-системы?
9. Перечислите и кратко охарактеризуйте архитектуры систем БД.
10. В чем основные различия OLTP- и OLAP-систем БД?

11. Какие формы представлений о ПрО выделяют в технологии БД? Охарактеризуйте каждую из них.