Базы данных
Лекция №1
13.02.2013
Клюкин Виталий Леонидович
Основные понятия информации, данных, хранилища данных

Информация – сведения, разъяснения, ознакомления.

Информацию модно классифицировать по различным критериям:

5. По истинности
a. истинная
b. ложная
6. по способу восприятия
. визуальная
a. аудиальная
b. тактильная
c. обонятельная
d. вкусовая
7. По форме представления
   . текстовая – в виде символов, предназначенных обозначать лексемы языка
   a. числовая – в виде цифр и знаков, обозначающих математические действия
   b. графическая – в виде изображений, предметов, графики
   c. звуковая – устная или в виде записей передача лексем языка аудиальным путем
5. по назначению
   a. массовая – содержит общие простые сведения и оперирует набором понятий понятным большей части социума
   b. специальная – содержит специфический набор понятий, при использовании происходит передача сведений, которые могут быть не понятны основной массе населения, но не обходимый понятный в рамках узкой социальной группы
   c. секретная – передается узкому кругу лиц и по закрытым (защищенным каналам связи)
   d. личная (приватная) – набор сведений о какой-либо личности, определяющий его социальное положение
5. по значению
   a. актуальные – информация ценная в данный момент времени
   b. достоверная – полученная без искажений
   c. понятная – выраженная на языке понятном тому, кому она предназначена
   d. полная – достаточная для понимания или принятия правильного решения
   e. полезная – полезность информации определяется субъектом

Хранение информации осуществляется с помощью ее переноса на некоторые материальные носители. Семантическая информация (на смысл) зафиксированная на материальные носители называется *документом.* Есть признаки когда эту информацию можно считать документом. Для хранения используются различные устройства. Сформированные по назначению информации образуют базы данных, банки данных, базы знаний и т. д.

Передачей семантической информации называется процесс ее пространственного переноса от источника к получателю. Представления с их помощью семантическая информация о каком-либо объекте, явлении или процессе называется *сообщение.* Поскольку информация не материальна, ее обработка заключается в различных преобразованиях. Информация предназначенная для обработки называется *данными.*

Основным видом первичной обработки информации является преобразование в форму удобную для восприятия органами чувств человека.

Важнейшим видом обработки семантической информации является определение смысла, заключающегося в некотором сообщении. В отличии от первичной семантическая информация не имеет количественной меры. Смысл или есть или его нет. Смысл сообщения описывается на искусственном языке, отражающем смысловые связи между словами исходного текста. Словарь такого языка называется *тезаурус*.

Данные – это представление фактов в формализованном виде пригодном для передачи и обработки. При этом данные могут выступать как источник информации.

Обработка данных включает следующие операции.

Лекция 2
13.02.2013

- Ввод данных – накопление с целью обеспечения достаточной полноты для принятия решения.
- Формализация данных – приведение к одинаковой форме.
- Фильтрация данных – отсеивание лишних данные, в которых нет необходимости, для повышения их достоверности и адекватности.
- Сортировка данных – упорядочивание данных по заданному признаку с целью удобства и использования.
- Архивация – операция хранения в удобной и доступной форме.
- Защита – меры, направленные на предотвращение утраты, воспроизведения и модификации.
- Транспортировка – прием и передача данных между участниками информационного процесса.
- Преобразование – перевод данных из одной формы в другую или из одной структуры в другую.

Объемы и структуры данных зависят от конкретной предметной области, частоты обращения к данным и модификации, жизненном циклом данных и т. д.

Информация в «чистом виде» часто не представляет практического интереса для пользователя, ее необходимо обрабатывать, анализировать и делать определенные выводы. Любая практическая задача связана с манипуляцией информации и данными.

Для более удобной обработки и манипуляции данными были разработаны базы и банки данных.

Понятие Базы данных можно применить к любой связанной информации хранимой как правило в виде таблицы. Для выполнения операций с базами данных необходимо выполнять следующие действия:

1. Добавление новой информации в существующие файлы БД
2. Добавление новых пустых файлов в БД
3. Изменение информации в существующих файлах БД
4. Поиск информации в БД
5. Удаление информации и существующих файлов БД
6. Удаление файлов из БД

В БД входят состав любой ИС (информационной системы), ЭС (экспортной системы) и СППР (система поддержки принятия решений).

БД – совокупность связанных данных, правила организации которых основаны на общих принципах описания, хранения и манипулирования данными.

БД – именованная совокупность сортированных данных относящихся к определенной области и находящихся под централизованным управлением.

Предметная область – часть реального мира, представляющая интерес для данного исследования или использования в информационной системе.

Достоинства БД:

1. Компактность
2. Высокая скорость обработки информации
3. Более низкие трудозатраты
4. Оперативность поиска информации
5. Возможность неоднократного использования информации большим количеством пользователей

Codasyl – организация ассоциация

1. Установление многосторонних связей между элементами данных. Метод организации данных должен обеспечивать удобное представление этих данных.
2. Производительность. СУБД должна обеспечивать время ответа удовлетворительное для диалога «человек-терминал».

СУБД должна обеспечивать соответствующую пропускную способность.
В системах предназначенной только для пакетной обработки время ответа не так важно, и метод физической организации выдирается из условий обеспечения более эффективной пакетной обработки, большая зависимость не от СУБД а от каналов связи.

3. Минимальные затраты. Для уменьшения затрат для создания и эксплуатации БД выбираются такие методы организации, которые минимизируют требования к внешней памяти, при этом выбирается вариант удовлетворяющий затратам на преобразование и экономии памяти

Лекция №3
21.02.2013

4. Минимальная избыточность. Одной из целей организации БД является устранение избыточных данных и контроль за теми противоречиями, которые они вызывают.

5. Возможность поиска. Чаще всего в приложениях типы запросов заранее определены. Одним из требований к системам является обеспечение обработки таких запросов, которые заранее не запланированы.

6. Целостность (согласованность). Многие СУБД позволяют работать в многопользовательском режиме. При использовании одних и тех же данных несколькими пользователями необходимо чтобы элементы данных и связи между ними не разрушались. Хранение данных, их обновление и добавление в случае различного рода сбоев должны восстанавливаться без потерь. Но в каждой СУБД можно установить время сохранения данных.

7. Безопасность и секретность. Данные в БД должны храниться в тайне и сохранности. Под безопасностью данных понимают защиту данных от несанкционированного доступа. А под секретностью данных понимается право отдельных лиц передавать и использовать соответствующую информацию.

8. Связь с прошлым (совместимость). Новые варианты информационной системы (ПО в информационной системе) должны быть совместимы с ранее используемой.

9. Связь с будущим. Одной из задач при разработке БД является необходимость планирования БД так, чтобы при ее изменении не нужно было модифицировать прикладные программы (максимальная независимость БД и приложений).

10. Простота использования. Средства используемые для представления логического описания данных должны быть простыми и понятными. Интерфейс ПО должен быть ориентирован на конечного пользователя – не профессионала в области БД. Интуитивно понятный.

Классификация БД.

1. По модели данных:
   a. Иерархическая
   b. Сетевая
   c. Реляционная
   d. Объектно ориентированная
   e. Постреляционная
   f. Многомерная

2. По типу хранимых данных:
   a. Документальная и документографическая
   b. Фактографическая
   c. Гипертекстовая
   d. Мультимедийная

3. По степени распределенности:
   a. Централизованные – БД логически или физически расположена на одном компьютере (локально или удаленно).
   b. Распределенные – БД, в которых ее составные части физически размещаются в территориально удаленных местах, соединенных компьютерной сетью.
   c. Параллельные БД.

Типы данных по типу хранимых данных.

Докуметальные и документографические БД – полнотекстовые БД, содержащие полные тексты документов. К ним можно отнести БД рефератов, газетных статей, научных сборников статей и диссертаций и т. п. Главная задача – информационная обеспечение на основе запроса.

Основными функциями ДИПС (документально-информационная поисковая система) (напр. Консультант) являются:

- Ввод и регистрация документаций.
- Хранение
- Обработка
- Поиск

Автоматизация информационного поиска требует формализации представления основного смыслового содержания информационного запроса и документа. Решение о выдаче документа в ответ на запрос принимается на основе именно смыслового соответствия содержания документа информационного запроса.

Докуметографическихе БД в отличие от документальных содержит только описание документа. Чаще всего используется библеографическое описание документа, ключевые слова, реферат и аннотация. Данные системы используются для автоматизированного поиска документов и выдачи полного текста в виде копий.

Фактографические БД – БД содержащая информацию относящуюся непосредственно к конкретной предметной области. Данные информационные системы используют фактические сведения, представленные в виде формализованных записей данных. Примеры фактографических БД, БД о продукции, о социальных данных, транспортные системы, в области культуры и искусства, лингвистические БД.

Гипертекстовые БД. Основной особенностью гипертекстовой БД является нелинейная организация информационных единиц, которые могут быть представлены текстом, аудио и видео информацией. Гипертекст можно определить как нелинейную документацию которая имеет древовидную структуру, при этом навигацию по данной структуре выбирает сам пользователь.

Основные признаки гипертекстовой БД:

- Структурный – БД должна иметь информационные единицы и структурные и семантические связи между ними.
- Функциональный – приложение должно иметь средства для работы в режимах администрирования и пользователя.

Управление функционированием гипертекстовой системы выполняется на основе принципа прямого манипулирования объектами с немедленно видимыми результатами.

При разработке дизайна используются технологии с учетом человеческой психики.

Лекция №4
27.02.2013

Мультимедийная БД. Пользователь может выполнять поиск и идентификацию мультимедийных объектов. Пока мультимедийные БД не в полной мере отвечают БД. Сейчас используются традиционные БД с упорядочивание и поиском по различным атрибутам мультимедиа данным. Это связано со сложностью обработки и распознавания образов за приемлемое время. Организация мультимедийных БД предполагает использование искусственного интеллекта. Управление данных систем выполняется на основе принципа прямого манипулирования объектами с немедленно видимыми результатами. При разработке дизайна приложения используются соответствующие технологии с учетом человеческой психики.

Один из примеров мультимедийных СУБД является Jasmine. Jasmine позволяет разрабатывать виртуальную модель предприятия, представляя бизнес данные с помощью мультимедийного интерфейса и формирует логику приложений и накапливает данные в БД. При этом объекты могут быть спроектированы один раз и затем многократно использованы. Jasmine управляет всем пространством данных от видео и звука до молекулярных структур, финансовых инструментов, и телекоммуникационных сетей. Структура интерфейса Jasmine, включает в себя прикладные программы для бизнеса и интернета, а качестве хранилища используются реляционный БД с соответствующим интерфейсом, можно использовать разные языки программирования (Java, Visual Basic, C++). Приложения Jasmine допускают распространения на рабочих станциях, клиентах, как в автономных так и использующих веб-браузеры. Бизнес логика и хранение объектов мультимедиа реализуется на серверной части.

Основные модели данных для представления информации

Три основных модели:

1. Иерархическая
2. Сетевая
3. Реляционная

Иерархическая модель данных. Реализует данные в виде древовидной структуры. Деревом в информатике называют совокупность корневого элемента и множество подчиненных ему элементов в которой отношения между элементами носит подчиненный вертикальный характер. Горизонтальные связи в такой системе не допускаются. В данной модели имеется корневой узел (root) он располагается на самом высоком уровне и не имеет узлов предшественников. Остальные узлы называются порожденными, каждый из которых имеет исходный находящийся. На следующем уровне каждый узел может иметь несколько узлов-потомков. Узлы не имеющие порожденных называются листьями. Элементы иерархии в этом случае служат в основном для навигации. Между исходным узлом и порожденными узлами по условию модели существует связь «один ко многим». Иерархия должна удовлетворять следующим условиям:

1. Она имеет исходный узел, из которого строится дерево. Каждое дерево имеет один корень;
2. Узел имеет не пустое множество атрибутов, которые описывают объект;
3. Порожденные узлы могут добавляться в дерево как в вертикальном так и в горизонтальном направлении;
4. Доступ к порожденным узлам возможен только через исходный узел, поэтому существует только один путь доступа к каждому узлу;
5. Возможно наличие нескольких экземпляров каждого узла каждого уровня, при этом каждый экземпляр исходного узла начинает логическую запись.

Условия для использования иерархической модели:

1. Предприятие состоит из отделов, в которых работают сотрудники
2. В каждом отделе может работать несколько сотрудников
3. Сотрудник не может работать больше чем в одном отделе

Основные операции иерархической модели.

1. Добавление в БД новой записи, при этом для корневой записи обязательно определение ключа.
2. Изменение данных в предварительно извлеченной (считанной) записи. Ключевые данные не изменяются.
3. Удаление некоторой записи и всех подчиненных ей записей.
4. Два варианта извлечения

1. Извлечение корневой записи по ключевому значению
2. Извлечение следующей записи в порядке левостороннего хода дерева.

Основные достоинства иерархической БД:

1. Связь данной модели именовать не требуется.
2. Удобно для отображения связей «один ко многим» в предметной области

Основные недостатки иерархической БД:

1. Сложность отображения связей «многие ко многим»
2. Усложнение операций добавления новых объектов и удаление старых непосредственно в БД.

Примеры СУБД поддерживающих иерархические структуры данных — IMS

Сетевая модель данных.

Сетевая модель данных является моделью объекта связей где допускаются только бинарные связи типа «многие к одному», что позволяет использовать для представления данных модель в виде ориентированных графов. Основное отличие моделей состоит в том что в сетевой модели запись может быть членом более чем одного группового отношения. Тип группового отношения задается его именем и определяет общее для всех экземпляров данного типа свойство. Экземпляр группового отношения представляется записью владельца и множеством (возможно пустым) подчиненных записей.

CODASYL предложила свои требования к сетевым СУБД.

Лекция №5
7.03.2013

1. GO TOP – переход на первую запись
2. GO BOTTOM – переход на последнюю запись
3. SKIP [<b,N>] – переход.
   SKIP 1 – переход на следующую
   SKIP -1 – переход на предыдущую
   - BOF () – истина если начало таблицы
   - EOF() – истина если конец таблицы
   - RECNO () – номер текущей записи
   - RECCOUNT () – число записей в таблице
   - IF <b,L> [ELSE] ENDIF
   - NOT = ! – отрицание

```
IF ! BOF()
SKIP -1
ENDIF

IF ! EOF
SKIP 1
ENDIF
```

4. APPEND BLANK – добавление
5. DELETE [FOR <b.L> /ALL] – удаление. Без параметров помечает текущую запись на удаление.

DELETED – истина если запись помечена на удаление
SET DELETE ON/OFF – установка обработки удаленных записей
PACK – физическое удаление. Требует чтобы таблица находилась в (эксклюзивном) однопользовательском режиме
При физическом удалении просто создается новая таблица с неудаленными записями.
RECALL – восстановление
MESSAGEBOX(<b.c1>[,<b.N>[,b.c2]])
1 параметр – сообщение
2 параметр – кнопки и иконки
3 параметр – заголовок
Возвращает числовое значение. Значение нажатой клавиши.

6. THISFORM.[<имя контейнера>.]<имя объекта>.<свойство или метод> – обращение к объекту.
7. THISFORM.REFRESH – каждая процедура кроме выхода должна заканчиваться строкой

THISFORM.RELEASE – выход. Удаление формы и из памяти и с экрана.
FOR <имя поля> = <b.N1> <b.N2> STEP <b.N3>

Некоторые полезные функции.
VAL (<b.c>) – преобразует символ в число
STR(<b.N>) – число в строку
Стандартный размер – 10 символов.
CTOD (<b.c>) – строка в дату
DTOC (<b.D>)– дата в строку
ALLTRIM (<b.c>) – удаляет все пробелы
UPPER(<b.c>) – переводит в верхний регистр
LOWER(<b.c>) – переводит в нижний регистр

REPLACE <имя поля> WITH <выражение>[....]
REPL – допускается не все имя функции, а только 4 символа
; – команда на этой строке не закончилась

```
REPL DR WITH CTOD (ALLTRIM(STR(THISFORM.CALENDAR.DAY))+'.'+;
                   ALLTRIM(STR(THISFORM.CALENDAR.MOUNTH))+'.'+;
                   ALLTRIM(STR(THISFORM.CALENDAR.YEAR)))
                     '07'+'.'+'03'+'.'+'13'
```

Лекция №6
13.03.2013

Сетевая модель данных.

Codasyl – своя стандартная модель сетевых СУБД.

1. Элемент данных – наименьшая именованная единица данных представляемая в БД значением
2. Агрегат данных – именованный набор элементов данных внутри записи
    1. Вектор, то есть одномерная последовательность элементов данных, имеющих однотипные характеристики.
    2. Повторяющиеся группы, то есть набор данных обычно с разными характеристиками, которые могут многократно повторяться внутри записи.
    3. Запись – экземпляр именованного набора данных состоящего из нуля, одного или нескольких элементов или агрегатов данных. Он определяется статьей записи, которая определяет тип записи. В этом случае каждая запись может иметь свой тип. В БД может быть объявлено произвольное число типов записей и может существовать произвольное число записей одного типа.

Нововведения CODASYL

1. *Ключ БД*. Для того чтобы отличать одну запись от другой при ее помещении в БД назначается идентификатор (ключ БД). Ключ остается постоянным идентификатором пока запись существует в БД. Так как в записи может существовать коллекция записей (несколько записей одного типа) то становятся определенными следующие операции:

1. добавление новых записей
2. изменение и удаление существующих
3. селекция (выбор) записей
4. предоставление записи процессу

2. Указатель текущей записи. Для движения по коллекции вводится понятие текущей записи.

Логическая структура БД «собирается» из двухуровневых деревьев. Запись являющаяся корневой вершиной дерева называется *записью-владельцем*., подчиненные ей вершины – *записями-членами*. Объединение данных типов записей представляет собой набор

<ФОТО>

Основные операции сетевых БД.

1. *Добавление* записи в БД в зависимости от режима добавления, либо включить ее в групповое отношение, где она объявляется подчиненной, либо не включать ни в какое групповое отношение тогда она будет корневой.
2. *Включение* в групповое отношение, то есть связать подчиненную запись с записью-владельцем.
3. *Переключение.* Связать подчиненную запись с другой записью-владельцем в том же групповом отношении.
4. *Обновление.* То есть изменение значения элементов в предварительно извлеченной записи.
5. *Извлечение* – считывание записи последовательно по значению ключа. При этом от владельца можно перейти к подчиненным записям и наоборот.
6. *Удаление.* При этом обязательные элементы записи должны быть предварительно исключены из группового отношения, то есть в данном случае из набора, фиксированные – должны быть удалены вместе с владельцем, а необязательные останутся в БД.

Достоинства сетевой модели.

1. Можно использовать для более широкого круга задач
2. Удобны для реализации связи многие команды.

Недостатки

1. Каждую связь нужно именовать.
2. Сложность для восприятия достаточно разветвленной структуры БД.

Примером СУБД основанной на сетевой модели данных является IDMS (Integral Database Managment System).

Реляционная модель данных.

Основная модель данных. Впервые была предложена в 1970 году сотрудником компании IBM. Сейчас фактически стала стандартной. Достигается более высокий уровень абстракции данных в предметной области чем в иерархической и сетевой модели, следовательно класс задач которые можно реализовать более широкий. Для нее был разработан теоретический математический базис (матаппарат) с использованием теории множеств. Реляционная модель основывается на понятии отношений как оно вводится в математике.

Пусть даны множества D1, D2,..., Dn, тогда N – отношение на этих множествах, если R – есть множество упорядоченных кортежей d1, d2,...,dn, таких что d1 принадлежит D1, d2 in D2, dn in Dn. Множество D – домены отношения R. Величина N – степень отношений. Реляционная модель может быть представлена в виде таблиц. Эти таблицы удовлетворяют определенным ограничениям в отличие от реляционных файлов, поэтому могут рассматриваться как математическое отношение. Строки этих таблиц соответствуют кортежам, а столбцы – атрибутам отношения. Домен – набор значений из которых извлекаются значения для данного столбца.

Лекция
27.03.2013

1. Теория нормальных форм.

В реляционной БД схема содержит структурную и семантическую информацию. Структурная связана с объявлением отношений, а семантическая выражается множеством известных функциональных зависимостей между атрибутами отношений.

В основном рассматривают зависимости между составными и не составными ключами.

Некоторые функциональные зависимости могут быть нежелательными из- за побочных эффектов (аномалий), которые они вызывают при модификации БД. При этом корректной считается схема, в которой отсутствуют нежелательные функциональные зависимости. В противном случае приходится прибегать к процедуре *декомпозиции.* Цель процедуры – устранение нежелательных функциональных зависимостей.

Нормализация – это пошаговый обратимый процесс замены данного набора отношений другим, имеющим более простую и регулярную структуру.

Атрибут, входящий в ключ, называется *первичным (ключевым),* в противном случае – *не первичным*.

Первая нормальная форма – отношение находится в первой нормальной форме, если значение всех его атрибутов простые (атомарные), то есть не является множеством или повторяющейся группой.

Есть два отношения «Рейсы» и атрибуты «Пункт отправления», «Пункт назначения» и …

«Номер маршрута», «Время отправления» и «Время прибытия»

Вторая нормальная форма – функциональная зависимость A→B называется полной, если B зависит от всей группы атрибутов, а не от ее части, где A – список ключевых атрибутов, B – не ключевой атрибут, но это атрибуты одного отношения.

Пример: БД должна содержать информацию о поставках и товарах.

Получается отношение Поставщик-товар-цена

семантика:

1. поставщик может поставлять разные товары
2. один и тот же товар могут поставлять разные поставщики
3. цена товара фиксирована, не зависит от поставщика

составной ключ «поставщик и цена».

Цена зависит от товара.

Не полная функциональная зависимость атрибута Цена от ключа приводит к следующим аномалиям:

1. Аномалия включения – если у поставщика появляется новый товар, информация о товаре и его цене не может появиться в БД до тех пор, пока поставщик не начинает его поставлять
2. Аномалия удаления – если поставки некоторого товара прекращаются, из БД придется удалить сведения о товаре и его цене, даже если он есть у поставщиков.
3. Аномалия обновления – при изменении цены товара необходим просмотр всего отношения для поиска всех поставок данного товара у всех поставщиков для изменения цены.

Отношение находится во второй нормальной форме, если оно находится в первой и каждый не первичный атрибут функционально полно зависит от ключа.

Третья нормальная форма **–** для приведения в третью нормальную форму рассматривают транзитивную форму.

Если есть следующие зависимости: A → B, A не → A , B → C, то A → C

Фирма получает товары только с одного склада.

Фирма – ключ.

Если объем склада изменился, необходим просмотр всего отношения и изменение кортежей для фирм, связанных со складом.

Хранение с атрибутами «Фирма — склад» и «Склад — объем».

Отношение находится в *третьей нормальной форме*, если оно находится во второй нормальной форме и в нем отсутствуют транзитивные зависимости не первичных атрибутов от ключа.

Поиск в Fox.

1. фильтрация, используется вся таблица. SETFILTER TO [<b.L>]
SETFILTER TO [FAM = Иванов]
SETFILTER TO – снятие фильтра

```
GO TOP

IF RECNO() > RECC()
MESSAGEBOX (' Not found')
ENDIF
```

2. Locate For <b.L>
Found() - истина если найдена хотя бы одна запись.

3. SEEK() – поиск в индексированных таблицах. Истина при удачном, ложь – при не удачном.

```
OB= CREATEOBJECT ('EXCEL.SHEET')
OB.APPLICATION.CELLS(1,1).VALUE='№ Компьютера'
STR=2
```

```
SCAN
OB.APPLICATION.CELLS(STR.1).VALUE='№ Компьютера'
STR=STR+1
ENDSCAN
OB.APPLICATION.VISIBLE=.T.
READ
```

Лекция
4.04.2013

Нормальная форма Бойсса-Кодда (третья усиленная).

Например: есть отношение «Проект» с атрибутами: деталь, № проекта, поставщик.

Условия:

- каждый поставщик обслуживает только один проект;
- каждый проект может обслуживаться несколькими поставщиками;
- в проекте используется несколько деталей;
- каждая деталь поставляется одним поставщиком.

В качестве ключа выбираются деталь и № проекта.

Определим функциональные зависимости:

- поставщик зависит от детали и от № проекта по определению ключа;
- № проекта зависит от поставщика.

Аномалия обновления: если меняется поставщик некоторого типа детали, необходим просмотр всего отношения для изменения всех кортежей, содержащих эти детали.

Декомпозиция:

1. № проекта, деталь;
2. поставщик, № проекта.

Отношение находится в ***НФВК***, если оно находится в третьей нормальной форме и в нем отсутствуют зависимости первичных атрибутов от непервичных.

Говорят, что B многозначно зависит от A, если каждому значению A соответствует множество значений B, никак не связанных с другими атрибутами отношения; при этом B может быть пустым множеством.

Например: есть отношение «Преподаватель» с атрибутами: шифр, дисциплина, дети, должность. Ключевыми являются шифр и дисциплина.

Условия:

- каждый преподаватель может иметь несколько детей;
- каждый ребенок должен принадлежать одному преподавателю;
- преподаватель может вести несколько дисциплин;
- одну и ту же дисциплину могут вести несколько преподавателей.

Номер кортежа	**Шифр**	**Дети**	**Дисциплина**	**Должность**
1	6501	Вася	БД	Доцент
2	6501	Петя	АСНИ	Доцент
3	6501	Вася	АСНИ	Доцент
4	6501	Петя	БД	Доцент
5	6548	Коля	БД	Профессор

В данном отношении есть две многозначные зависимости:

1. дети - шифр;

2. дисциплина - шифр.

Независимым является отношение дети — должность.

Разбиваем на три отношения:

1. дети: шифр, дети;
2. дисциплина: шифр, дисциплина;
3. должность: шифр, должность.

Отношение находится в ***четвертой нормальной форме***, если оно находится в НФВК и в нем отсутствуют многозначные зависимости, которые не являются функциональными.

Отношение находится в ***пятой нормальной форме***, если каждая его проекция содержит не менее одного возможного ключа и минимум один непервичный атрибут.

Цели нормализации:

1. освобождение набора отношений от нежелательных зависимостей при модификации;
2. стремление сделать набор отношений более нейтральным для прикладного ПО.

Процесс нормализации:

ненормализованное отношение → 1НФ → 2НФ → 3НФ → 4НФ

1. переход от произвольной структуры данных, не являющийся двумерным, к двумерным отношениям;
2. устранение всех неполных зависимостей атрибутов, не являющихся основными, от вероятных ключей;
3. устранение транзитивных зависимостей;
4. устранение нетривиальных многозначных зависимостей.

Технология, сущность, связь.

Недостатки:

1. первоначальное размещение всех атрибутов в одном отношении является неестественной операцией;
2. сложно сразу определить полный список атрибутов;
3. сложно выделить все зависимости между атрибутами.

В реальном проектировании структуры БД используется семантическое моделирование, которое опирается на смысл данных. При этом используются варианты диаграмм сущность, связь.

Лекция. Технология ERD
10.04.2013

Сущность — это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность именуется существительным в единственном числе.

Экземпляр сущности - это конкретный представитель данной сущности.

Атрибут сущности — это именованная характеристика, являющаяся некоторым свойством сущности. Атрибут выражается существительным в единственном числе, возможно, с прилагательным.

Ключ сущности — это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушает его уникальность.

Связь — это некоторая ассоциация между двумя сущностями. Она позволяет по одной сущности находить связанные с ней другие сущности. Именуется глаголом в неопределенной форме.

Три типа связи:

1. один к одному: имеется одна сущность, разделенная на две;
2. один ко многим;
3. много ко многим.

Пример разработки простой модели: при разработке необходимо получить следующую информацию о предметной области:

1. определить список сущностей предметной области;
2. определить список атрибутов сущностей;
3. взаимосвязи между сущностями.

Пример: разработать информационную систему по заказу оптовой торговой фирмы. Необходимо:

1. хранить информацию о покупателях;
2. печатать накладные на отпущенные товары;
3. следить за наличием товаров на складе.

Условия семантики:

- покупатель может купить несколько товаров;
- товар может быть продан нескольким покупателям;
- каждый покупатель может получить несколько накладных;
- каждая накладная должна выписываться на одного покупателя;
- накладная должна содержать один или более товаров;
- каждый товар может содержаться в нескольких накладных;
- накладная должна выписываться одного склада;
- со склада может быть выписано несколько накладных.

Атрибуты сущностей:

- для покупателя: наименование, адрес, банковский депозит;
- для накладной: уникальный номер, дата выписки, с какого склада выписана, для какого покупателя, наименование товара, единицы измерения, качество, цена, сумма;
- для товара: наименование, цена, единица измерения.

Проектирование физической модели:

Выполняется миграция…

Ключи, которые мигрируют, будут внешними ключами.

На экзамене будет задачка, для предметной области по условиям построить ERD, привести в нормальную форму и сделать выборку на SQL.

08.05.2013

Транзакции

транзакция - операция по взаимодействию с данными, переводящая данные из одного целостного состояния в другое целостное состояние. Одно из определений, и заисано не полностью верно. Транзакция с точки зрения воздействия на СУБД - неделимая последовательность манипулирования данными. Каждое обращение - отдельная транзакция. Вставки, удаление, выборки - всё это отдельная транзакция. С точки зрения пользователя транзакция выполняется по принципу "всё или ничего", то есть либо транзакция выполняется целиком (и переводит БД из одного целостного состояния в другое), либо, при

каком-либо типе сбоя, БД возвращается в исходное состояние и происходит откат транзакции.

Например, в виде транзакции можно оформить добавление или вставку данных в таблицу и сделать либо подтверждение, либо отказ. При подтвержднии транзакция фиксируется, либо полностью откатывается.

С точки зрения надёжности транзакция является единицей восстановления данных после сбоев. Практически у любого сервера БД есть журнал выполнения транзакций, где сохраняюся все действия по обращению к БД, иногда размеры этих журналов намного больше, чем сама БД.

Устанавливается время, через которое мы бэкапим базу, а между бэкапом и сбоем восстановление делается из лога транзакций. Основная работа адимна БД - восстановление тех серверов, которые упали.

У транзакций есть свойства. **АСИД**.

А - атомарность. Либо выполняется вся целиком, либо вся целиком не выполняется.

с - Согласованность. Внутри БД согласованность может нарушаться.

И - изоляция. Транзакции разных приложений не должны мешать друг другу, как если бы они выполнялись по очереди.

Д - долговечность. Если транзакция выполнена, то результаты её работы должны сохраниться в БД даже если в следующий момент произойдёт сбой системы.

Обычно транзакция начинается автоматически с момента присоединения пользователя к СУБД и продолжается до тех пор, пока не произойдёт одно из следующих событий:
1) Подана команда commit work - зафиксировать тразакцию
2) rollback work - откатить тразакцию. (есть понятия такие, как откат и накат, немножко другое)
3) произошло отсоединение пользователя от СУБД.
4) Произошёл сбой.

Команда Commit Work завершает транзакцию (то есть данные фиксируются в БД) и автоматически начинает новую. При этом гарантируется, что результаты работы фиксируются, то есть сохраняются в БД.

Отсюда выводы: не нужно делать транзакции слишком большими, чтобы не откатывать слишком далеко. Создание таблицы иногда делают в двух транзакциях: в первой создают структуру, во второй задают ключи, связи.

Применение команды rollback work откатывают все изменения транзакции так, как будто их вообще не было.
Автоматическое фиксирование транзакций, чтобы некоторые из них не были зафиксированы. Для этого ставится UPS, чтобы завершить все транзакции.

Самый сложный случай - когда произошёл сбой.

Суть сводится к тому, что при следующем после сбоя запуске системы выполняется анализ происходивших до момента сбоя транзакций.
Транзакции, для которых была подана команда commit work, но результаты которых не были занесены БД, выполняются снова.
Но могут начинаться не с начала, а с какого-то момента транзакции (это называется накат, по аналогии с докачкой большого файла из инета после дисконнекта).
Команды, поданные после commit work не возвращаются.

Но свойства АСИД не всегда выполняются, особенно что касается атомарности и изоляции.
На фзическом уровне транзакции будут выполняться не одновременно, а всё равно по очереди.

Но по долговечности есть ещё одно замечание.
Есть внешняя транзакция А, а внутри неё транзакция В. Подана команда commit work, внутренняя транзакция завершена, внешняя нет.
Внутренние транзакции будут окончательно зафиксированы только если будет зафиксирована внешняя транзакция.
Если внешняя транзакция не успела нормально завершиться, то и внутренняя отменяется.

В следующем семестре будет предмет про сайты и веб-технологии.
Это будет либо Клюкин вести, либо кто-то другой.
Если Клюкин, то трёхзвенная архитектура, то (L)AMP и CMSки (Joomla, Wordpress)

Ограничение целостности.
Согласованность -некоторое утверждение, котрое может быть истинным ило ложным в зависимости от согласованности БД.
Возраст сотрудников должен быть от 18 до 65 лет.
Сумма накладной должна равняться сумме произведений цен товаров на их количество.
Когда сотрудник должен быть в одном отделе - пример ссылочной целостности.
Любое ограничение целостности является семантическим понятием данной предметной области. То есть одно и то же ограничение может быть целостным в одной предметной области, но не являться целостным для другой.

База данных находится в согласованном (целостном) состоянии, если выполнены все ограничения целостности, определёные для БД.

Имеется два типа реакции на попытки нарушения целостности:
1) Отказ выполнить незаконную операцию. Это не значит, что система при всяком неправильном действии должна зависать, а должна на каждое действие выдавать какую-то реакцию. Считается, что 60% исходного кода - защита от дурака.
2) Выполнение компенсирующих действий. (каждое действие это транзакция, но внутри транзакции целостность может быть нарушена)

Внутри транзакции до её завершения должны быть выполнены компенсирующие действия.

Классификация ограничения целостности:
1) По способам реализации
2) По времени проверки
3) По области действия

По способам реализации различают два варианта:
1) Декларативная поддержка ограничений целостности. Выполняется средствами языка определения данных.
2) процедурная поддержка ограничений целостности

У многих СУБД есть элементы поддержки ссылочной целостности типа Каскадирование. Когда мы удаляем записи из родительской таблицы, то каскадом все связанные записи из дочерней таблицы так же удаляются.
Процедурная поддержка заключается в использовании триггеров и хранимых процедур.
Хранимая процедура выполняется пользователем, триггер срабатывает без участия пользователя по какому-либо событию.

Декларативная поддержка ограничения целостности заключается в определении ограничений средствами языка определения данных. Обычно декларативные ограничения устанавливаются для доменов и аттрибутов на целостность сущностей и ссылочную целостность. *Обычно они устанавливаются средствами самой СУБД (как раз таки языком определения данных). В фоксе были отдельные поля, где можно задать варинты ограничения целостности. Целостности сущности определяются, когда например мы создаём первичный ключ. Или когда создаём с помощью команды Create, ограничение команы not nul или когда используем каскадирование (когда удаляем записи из родительской таблицы, удаляются все соответствующие ей записи из дочерней таблицы). А бывает и что так нельзя., и бывает что можем из родительской удалить, а из дочерней не удаляются.*
Процедурная поддержка обеспечения целостности. Заключается в использовании три

Ограничение целостности, как декларативное, так и процедурное, всегда приводит к созданию или использованию некоторого программного кода, реализующего это ограничение.
Разница заключается в том, как он создаётся и где хранится.
Если СУБД не поддерживает ни декларативное, ни процедурное ограничение целостности (например, методами триггеров), то это ограничение целостности всё равно нужно писать, но в приложении.

Ограничение целостностности по времени проверки

Есть два типа:

Немедленно проверяемые

Проверяемые непосредственно в момент операции, которая может нарушить ограничения. *Например, проверка уникальности того же первичного кюча выполняестся в момент выполнения операции.* Если ограничение нарушается, операция обычно откатывается (если операция выполняется в виде транзакции).

С отложенной проверкой

По окончанию выполнения транзакции, когда в момент окончания транзакции целостность может быть нарушена *(например, commit wapr)*. Если целостность нарушается - транзакция откатывается.

Область, на которую накладываются ограничения:

Область домена
Ограничение базы данных, которое накладывается на всю строку.

Область атрибута
Ограничения атрибута проверяются, ограничения домена - нет. Являются немедленно проверяемыми. Больше ни от чего не зависят и ни что не может их изменить.

Ограничение кортежа на всю строку
Может быть наложено дополнительное ограничение. *Например, если атрибут "должность" в отношении "сотрудники"принимает значение "директор" содержит значение не менее $1000.*
*Другой пример. В отношении "накладная" есть такой пример, что сумма=стоимость*количество. Вопрос в том, считать один раз сразу, или каждый раз считать. А ещё нужно учитывать, что в результате округлений накапливается.*

Ограничение отношения
Накладываетя на всё отношение и с точки зрения времени проверки оно может быть как немедленно проверяемым, так и с отложенной проверкой. Это зависит от того, что проверяется. Елси на ключ, то немедленно.
Накладывается на значение двух или более связанных между собой ограничений.
Оно может быть тоже как немедленно проверяемым, так и с отложенной проверкй.

Контрольная.
ЕРД, довести до норм. вида, сделать выборку по условию.
Пример. Есть список типов компов (офисные, для геймеров и так далее). Есть список комплектующих. Условие1: каждый комп должен содержать нескольок типов комплектующих. Условие2: каждое комплектующее может входить в состав нескольких компов.
Условие для выборки: подсчитать стоимость комплектующих для каждого типа компа.
То есть две колонки должно быть: 1 название типа компа, и 2 его стоимость.

Вариант2
Есть список продавцов, есть список товаров. Условие1: Каждый продавец может продать несколько типов товаров. Условие2: каждый товар может быть продан несколькими разными продавцами.
Условие для выборки: посчитать, сумму продаж за месяц каждым продавцом. (то есть добавляется период времени)

На следующей лекции будет доклад Распределённые БД, а Облачные на последней. На следующей уточниться.-

22.05.2013
Параллельные? Базы данных
Data Warehouse
Специально разработ анная БД для подготовки отчётов и бизнес-анализа. Данные, поступающие в хранилище, как правило, доступны только для чтения и загружаются с определённой периодичностью. Другими словами, это некоторые выборки (это как выборка, сохранённая в представлении).

Принципы организации бизнес-хранилища:
1. Проблемно-предметная ориентация. Данные разделяются по категориям и хранятся в определённом виде.
2. Интегрированность. В хранилище хранятся объединённые данные, а не по какой-то одной проблеме.
3. Некорректируемость, доступ в режиме только чтения.
4. Зависимость от времени. Данные актуальны только в том случае, если они привязаны к некоторому промежутку или моменту времени.

Есть два архитектурных направления реализации хранилищ:
1. Нормализованные хранилища. Данные хранятся в нормализованном виде. Есть недостаток, присущий обычным нормализованным БД. Но есть и недостаток, присущий обычным БД: если используется много таблиц, то это приводит к увеличению времени реакции и ухудшению производительности системы.
2. Хранилище с измерениями. Используется многомерная модель данных, а не реляционная. С данной моделью можно намного быстрее выполнять разнообразные выборки. С этой моделью работает технология OLAP.

В зависимости от объёмов и назначения хранилища данных имеют разные названия, например:
Витрина данных, киоск данных, рынок данных, магазин данных и т.п.)))

Витрина данных (data markt) - срез хранилища данных, предсталяющий собой массив тематической узконаправленной информации.
Концепция витрин данных - это множество тематических БД, содержащих различные типы информации.
Достоинства:
1. Пользователи (аналитики) работают только с теми данными, которые им реально нужны.

2. Витрины обычно содержат заранее агрегированные данные. (заранее могут быть использованы выражения, функции, данные заранее подготовлены, обработаны, поэтому время на все агрегации меньше)

Источником данных для витрин является общее хранилище данных.
Система реализуется на трёх основных уровнях:

1. Общекорпоративная БД с нормализованной схемой (детализированные данные). *(многомерная модель строится на оснвое обычной реляционной модели)*
2. БД уровня подразделения реализуется на основе многомерных СУБД (агрегированные данные)
3. Приложение для аналитики на рабочих местах пользователей.

Эти реализации позволяют использовать достоинства при работе с агрегированными данными. Быстро, удобно, эффективно. На каждом уровне используются преимущества каждого типа баз данных.

Начнём с промежуточного типа:

Постреляционная модель

Должна быть атомарной. В некоторых условиях может уменьшать эффективность.
Постреляционная модель, снимающая ограничения релцяионной модели данных. Она допускает многозначные поля, значения которых состоят из отзначений.. При этом набор значений считается самостоятельной таблицей, встроенной в основную таблицу. Допускается, что поле может быть множеством, а может быть группой. Данная модель поддерживает ассоциированные многозначные поля (ассоциации). И первое значение одного столбца ассоциации соответствует первым значениям всех других столбцов. Таблицы, содержащие многозначные поля, находятся в **непервой нормальной форме**. (NF2 потому что NFNF).
Данная форма не нарушает принципов реляционной алгебры (без проблем можно использовать операции реляционной алгебры). (но могут быть использованы расширенные операторы, которые извлекают данные из таблиц NF2, и рассматривать их как данные, поступившие из таблиц в первой нормальной форме). Одним из недостатков реляционной модели является необходимость поддержания связи между таблицами. Здесь же когда одна таблица встроена внутрь другой, дополнительно связь поддерживать не нужно.
Отсюда и упрощается логическое представление данных, и это же позволяет более быстро выполнять различные операции.

В данной модели нет необходимости хранить внешние ключи потому что одна таблица является частью другой. Аналогично со ссылочными ограничениями целостности. Здесь в явном виде выполняется каскадирование. Например, при удалении записей из основной таблицы, естественно, автоматиески удаляются все вложенные в эту запись таблицы. Основное достоинство данной модели- возможность представления множества связанных реляционных таблиц одной пост-реляционной таблицей. Это повышает наглядность представления информации и эффективность её обработки. Основной недостаток это сложность обеспечения целостности и непротиворечивости хранимых данных.
В конце, пример СУБД, которая основана на постреляционной модели: CACHE.

Плавно переходим к многомерным модлям.

Многомерные модели БД

Считается, что многомерный подход к представлению данных в БД появился почти одновременно с реляционным. Особенно часто испольуются данные модели в технологиях или системах проведения анализа и принятия управленческих решений (СППР).
Многомерность моделей означает многомерное логическое представление информации при описании и в операциях манипулирования данными, а не визуализация.

Показывает презенташку с OLAP-кубиками.

Измерение - это множество однотипных данных, образующих одну из граней гиперкуба. Например, если взять измерение времени (дни, месяцы, годы) или какие-то географические параметры (город, район, регион, страна и т.д.)
Мера (ячейка, которая расположена на гранях) - это поле, значение которого определеяется фиксированным набором измерений. Чаще всего тип поля числовой. Значения могут быть переменными или вычисляться по формулам.

Из куба может быть составлен обычный плоский отчёт. Данный куб содержит всю информацию, которая может понадобиться для ответа на любые запросы. Обычно куб создаётся из соединения таблиц с применением схемы Звезда или Снежинка.

Для работы с данными системами используются специальныме механизмы. На переднюю грань мы вывели данные, которые нужны в данный момент. Можно менять грани и крутить OLAP как кубик-рубика.

При работе с многомерными моделями используется операция вращения. Суть её заключается в изменении порядка измерений при визуальном представлении данных (как при игре с кубиком рубика). Кроме того используются операции агрегации и детализации. Они означают соответственно переход к более общему или более детальному представлению информации пользователю из гиперкуба.

Архитектура многоуровневого хранилища с использованием многомерной модели данных для оперативного анализа данных.

И хоть и называется многомерная структура, но все они собираются с помощью обычных реляционных двумерных таблиц.
Используют два варианта: гиперкуб (много измерений в кубике) и поликуб (много гиперкубов в поликубе).

Основные достоинства многомерной модели:
1. Удобство и эффективность аналитической обработки.

30 мая 2013
Первая задачка из контрольной.
Построить ERD по двум спискам, задачка контрольной.
Первый список. Компьютеры:
В нём свойство ID компьютера, название компьютера. Его значения бывают: Офисный, производительный, игровой.
Вторая таблица Комплект. Поля: ID железки, название железки, цена.
Третья табличка Сборка. В ней поля: ключи родительских таблиц (правило такое есть) и коле Количество.
Связь многие ко многим заменяем на один ко многим.

select тип.название, Sum(сборка.колич*комплект.цена) FROM тип,сборка, комплект
WHERE тип.idтипа = сборка.idтипа AND сборка.idкомпл. = комплект.idкомпл.
group by тип.название;

Вторая задачка.
Условие. Есть список врачей и список пациентов.
Условие: каждый врач может принять нескольких пациентов. Один пациент может быть принят несколькими врачами. Условие для выборки: подсчитать количество приёмов для каждого врача за период от даты 1 до даты 2.

Моё. Первая таблица Врачи. Поля: ФИОврача.
Вторая таблица Пациенты. Поля:ID пациента, ФИО, дата рождения, адрес.
Третья средняя таблица Приёмы: ФиоВрача, ID пациента, дата приёма.

```
select врач.фио, count(приёмы.IDврача)
from врача, пациент, приём
where врач.IDврача = приём.IDврача AND
between(приём.
group
```