

Структуры базы данных. Часть 1.

Наумов Д.А., доц. каф. ИТГД, доц. каф. КТ

Основы компьютерных наук (3 часть), 2019

Содержание лекции

- 1 План дисциплины
- 2 Основные понятия и определения
- 3 Типы приложений: транзакционная и аналитическая обработка
- 4 Основные модели данных

Разделы дисциплины 'Основы компьютерных наук', 3-ий семестр обучения:

- **Структуры базы данных. (24 часа)** Общие понятия. Многоуровневый подход к реализации баз данных. Реляционная модель. Объектно-ориентированные базы данных.
- **Операционные системы. (24 часа)** Однопроцессорные системы. Многопроцессорные системы. Классификация программного обеспечения. Функции операционных систем. История развития ОС.
- **Основы технологии разработки программного обеспечения. (16 часов)** Жизненный цикл ПО. Стадии жизненного цикла. Модульность. Методы проектирования. Нисходящие и восходящие методы разработки.

Рекомендуемая литература:

- ❶ Карпова Т. С. Базы данных: модели, разработка, реализация. — СПб.: Питер, 2002. — 304 с., ил.
- ❷ Базы данных : учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2015. — 463 с. — Серия : Бакалавр. Прикладной курс
- ❸ Дейт К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс 2005. — 1328 с.: ил.
- ❹ Тарасов С. В. СУБД для программиста. Базы данных изнутри. — М.: СОЛОН-Пресс, 2015. — 320 с.: ил.
- ❺ Программирование баз данных SQL. Типичные ошибки и их устранение / Б. Карвин. — М.: Рид Групп, 2012. — 336с.

- База данных (БД) — структурированное поименованное хранилище информации.

Пример: файл CSV? файл с текстом лекции?

- Система управления базами данных (СУБД) — специализированное программное обеспечение, обеспечивающее доступ к базе данных как к совокупности её структурных единиц.

Пример: текстовый редактор для CSV-файла? LibreOffice Calc для CSV-файла?

Пример базы данных

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
3	Chardonnay	Geyser Peak	2001	5	2003
6	Chardonnay	Simi	2000	4	2002
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumé Blanc	Ch. St. Jean	2001	4	2003
22	Fumé Blanc	Robt. Mondavi	2000	2	2002
30	Gewurztraminer	Ch. St. Jean	2002	3	2003
43	Cab. Sauvignon	Windsor	1995	12	2004
45	Cab. Sauvignon	Geyser Peak	1998	12	2006
48	Cab. Sauvignon	Robt. Mondavi	1997	12	2008
50	Pinot Noir	Gary Farrell	2000	3	2003
51	Pinot Noir	Fetzer	1997	3	2004
52	Pinot Noir	Dehlinger	1999	2	2002
58	Merlot	Clos du Bois	1998	9	2004
64	Zinfandel	Cline	1998	9	2007
72	Zinfandel	Rafanelli	1999	2	2007

Базу данных можно рассматривать как подобие *электронной картотеки*, т.е. хранилище или контейнер для некоторого набора файлов данных, занесенных в компьютер.

Пользователям СУБД предоставляется возможность выполнять (или передавать системе запросы на выполнение) множество различных операций над данными:

- добавлять пустые "файлы";
- вставлять новые данные в существующие "файлы";
- получать данные из существующих "файлов";
- удалять данные из существующих "файлов";
- изменять данные в существующих "файлов";
- удалять существующие "файлов" из базы данных;

Пример выборки информации из базы данных

Выборка: SELECT WINE, BIN#, PRODUCER FROM CELLAR WHERE READY = 2004 ;		
Результат (в том виде, в каком он отображается, например, на экране дисплея):		
WINE	BIN#	PRODUCER
Cab. Sauvignon	43	Windsor
Pinot Noir	51	Fetzer
Merlot	58	Clos du Bois

Язык SQL (*Structured Query Language*, язык структурированных запросов) был первоначально разработан компанией IBM, а в настоящее время поддерживается большинством коммерческих СУБД, представленных на рынке, и является официальным стандартом языка для работы с реляционными базами данных.

Примеры операций вставки, удаления и обновления

Вставка новых данных:

```
INSERT  
INTO CELLAR ( BIN#, WINE, PRODUCER, YEAR, BOTTLES, READY )  
VALUES ( 53, 'Pinot Noir', 'Saintsbury', 2001, 6, 2005 ) ;
```

Удаление существующих данных:

```
DELETE  
FROM CELLAR  
WHERE BIN# = 2 ;
```

Модификация существующих данных:

```
UPDATE CELLAR  
SET BOTTLES = 4  
WHERE BIN# = 3 ;
```

Используемые термины:

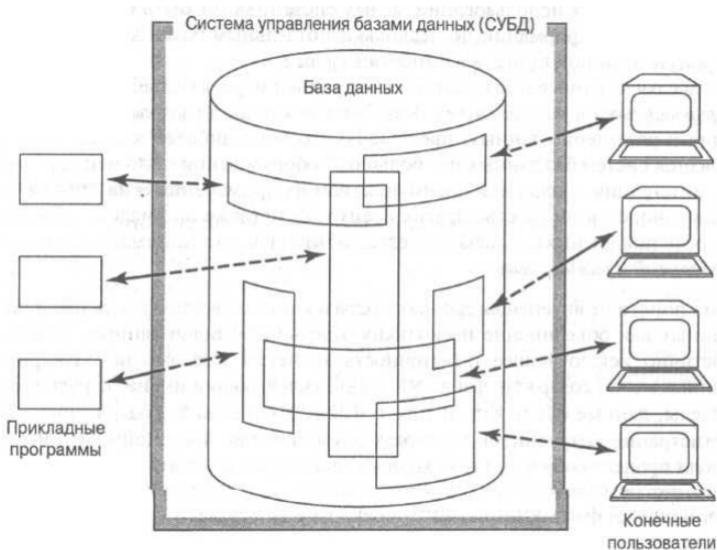
- файлы, записи.
- таблицы, строки, столбцы.
- отношения, кортежи и атрибуты.

В примере предполагается, что в столбцах WINE и PRODUCER содержатся строковые данные, а во всех остальных столбцах — целочисленные данные. Но, как правило, столбцы могут содержать данные произвольной сложности:

- LABEL. Фотография этикетки винной бутылки.
- REVIEW. Текст отзыва о качестве вина, полученный из определенного винного магазина.
- MAP. Карта той местности, где было изготовлено это вино.
- NOTES. Звукозапись, содержащая комментарии о вкусе вина.

Столбец BIN# является **первичным ключом** (primary key) таблицы CELLAR (подразумевается, что любые две строки этой таблицы никогда не будут содержать одно и то же значение поля BIN#)

Упрощенная схема СУБД



Четыре главных компонента СУБД:

- данные;
- аппаратное обеспечение;
- программное обеспечение;
- пользователи.

Однопользовательская система (single-user system) — это система, в которой к базе данных может получить доступ одновременно только один пользователь.

Многопользовательская система (multi-user system) — это такая система, в которой к базе данных могут получить доступ сразу несколько пользователей.

В общем случае данные в базе данных (по крайней мере, в больших системах) являются:

- интегрированными - подразумевается возможность представить базу данных как объединение нескольких отдельных файлов данных, полностью или частично исключая избыточность хранения информации.
- разделяемыми - подразумевается возможность использования

К аппаратному обеспечению системы относится:

- тома вторичной (внешней) памяти (обычно это магнитные диски), используемые для хранения информации, а также соответствующие устройства ввода—вывода (дисководы и т.п.), контроллеры устройств, каналы ввода—вывода и т.д.;
- аппаратный процессор (или процессоры) вместе с оперативной (первичной) памятью, предназначенные для поддержки работы программного обеспечения системы баз данных.

- Между физической базой данных (т.е. данными, которые реально хранятся на компьютере) и пользователями системы располагается уровень программного обеспечения, который можно называть по-разному: диспетчер базы данных (database manager), сервер базы данных (database server) или, что более привычно, система управления базами данных, СУБД (DataBase Management System — DBMS).
- Основная задача СУБД — дать пользователю базы данных возможность работать с ней, не вникая во все подробности работы на уровне аппаратного обеспечения.
- СУБД позволяет конечному пользователю рассматривать базу данных как объект более высокого уровня по сравнению с аппаратным обеспечением, а также предоставляет в его распоряжение набор операций, выражаемых в терминах языка высокого уровня.

Пользователей можно разделить на три большие и отчасти перекрывающиеся группы:

- прикладные программисты, которые отвечают за написание прикладных программ, использующих базу данных.
- конечные пользователи, которые работают с системой баз данных в интерактивном режиме.
- администраторы базы данных.

Исторически, в устройстве СУБД выделяли три уровня, предложенных ещё в 1975 году в отчёте ANSI/X3/SPARC:

- внешний уровень наиболее близок к приложениям и пользователям, он связан со специфичными для них способами представления данных.
- логический уровень, также называемый концептуальным, для описаний данных, не зависящих от физической реализации;
- логический уровень, также называемый концептуальным, для описаний данных, не зависящих от физической реализации.

Внутренний уровень не всегда связан с файловой системой. Наиболее развитые современные СУБД представляют собой по сути специализированную операционную систему, способную управлять физическими устройствами хранения, кешем данных, процессами и потоками, оперативной памятью, асинхронным запуском и внутренним планировщиком задач минуя собственно операционную систему компьютера.

Транзакцией в СУБД называется совокупность операций над данными, являющаяся неделимой (атомарной).

Отличительные особенности транзакционных приложений:

- Обработка идёт в режиме реального или приближенного к реальному времени. Время отклика системы при запросе оператора не превышает единиц секунд.
- Запросы представляют собой интенсивный поток коротких операций по вставке, изменению и удалению небольшого числа записей в БД. Эти операции могут быть как одиночными транзакциями, так и объединяться в более крупные транзакции.

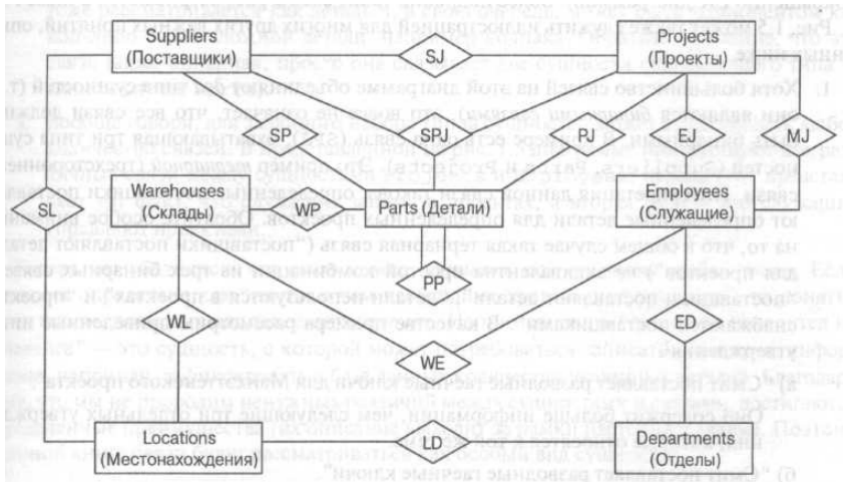
В реляционной СУБД любой оператор SQL является одиночной транзакцией по умолчанию. OLTP (On-Line Transaction Processing) — собственно интерактивная транзакционная обработка. ACID (Atomicity-Consistency-Isolation-Durability) — принципы неделимости, целостности, изолированности и надёжности.

аналитическая обработка данных имеет следующие отличительные признаки:

- Данные находятся в **режиме чтения**, за исключением моментов их обновления.
- Выборки представляют собой **одиночные тяжёлые запросы**: поиски и расчёты по множеству произвольных критериев могут охватывать значительную часть данных в базе.
- **Время отклика системы не регламентировано**, нередко пользователь имеет возможность прервать слишком долго.
- Размеры базы данных, как правило, на порядок и больше превышают таковые для транзакционной.

Аббревиатурой для интерактивной аналитической обработки является OLAP (On-Line Analytical Processing). Соответственно, **интерактивное** приложение, работающее с СУБД в режиме OLAP, относится к аналитическим.

Пример диаграммы "Сущность-связь"



Модель данных

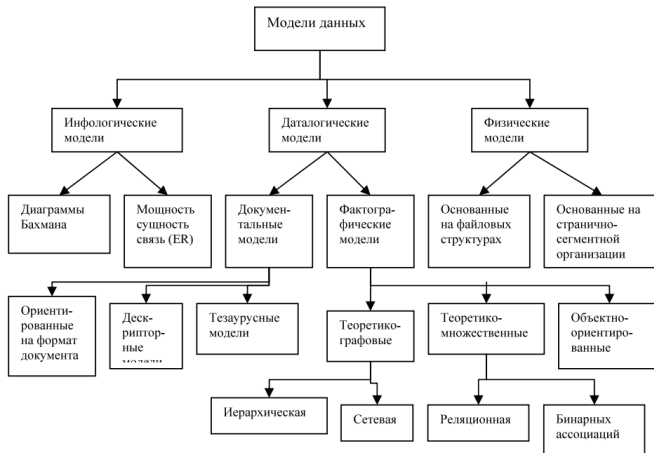
это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Упомянутые объекты позволяют моделировать структуру данных, а операторы — поведение данных.

Реализация (implementation) заданной модели данных

это физическое воплощение на реальной машине компонентов абстрактной машины, которые в совокупности составляют эту модель.

Термин модель данных (иногда, некорректно) встречается в литературе другом толковании: модель данных представляет собой модель перманентных данных некоторого конкретного предприятия.

Классификация моделей данных [Карпова Т., Базы данных]



Иерархическая модель

- исторически, первая моделью данных, (т.е. способ их организации, структурирования, доступа и манипуляции данными).

IMS (Information Management System) фирмы IBM Разработанная в 1966 году до сих пор эксплуатируется на новейших мэйнфреймах серии Z, обеспечивая высокую производительность обработки порядка сотни тысяч транзакций в секунду.

Современной массово доступной каждому программисту реализацией иерархической модели данных является XML, точнее, разнообразные «движки» и API для манипуляции со структурами, созданными на основе этой технологии с обязательным применением схем определения данных.

Пример XML-документа

```
<?xml version="1.0"?>
<sales>
  <orders>
    <order>
      <num>S01</num>
      <date>2014-02-20</date>
      <client>
        <name>Пирожки ООО</name>
        <address>ул. Благодатная 235</address>
        <phone>322-223-322</phone>
      </client>
      <items>
        <product>
          <name>Мука</name>
          <quantity>50</quantity>
          <units>кг</units>
          <price>45</price>
        </product>
        <product>
          <name>Дрожжи</name>
          <quantity>300</quantity>
```

Пример запроса к данным XML-документа

```
var nodes = xmlDoc.SelectNodes(  
    "/sales/orders/order[client/name='Пирожки ООО' and  
    items/product[name = 'Мука']/quantity > 50]");  
  
foreach(node in nodes) {  
    ...  
}
```

Если при использовании входного языка и/или API наиболее низкого уровня из доступных программисту для доступа к данным (значениям узлов, переменных, полей и т. д.) **требуется указывать некоторый путь**, то лежащая в основе СУБД модель базируется на **иерархиях**.

Преимущества иерархической модели

- отнести относительную простоту восприятия логической структуры базы данных человеком.
- высокое быстродействие при транзакционной обработке, когда номенклатура типов запросов фиксирована.

Недостатки иерархической модели

- медленный доступ к данным нижних уровней иерархии;
- чёткая ориентация структур на определённые типы запросов, что исключает универсальность использования одной и той же БД разными типами приложений;
- модель графов ограничена деревьями, что сужает область её применения.

Сетевая модель

Сетевую модель можно представить в виде графа с узлами в виде записей, и рёбрами, отображающими наборы.

До наших дней дожило только небольшое число СУБД, реализующих сетевую модель, например американская Raima (бывшая dbVista) и отечественная КроносПро.

Запись

соответствует аналогичному понятию структурного типа в традиционных языках программирования: `record` в Паскаль-подобных или `struct` в наследниках Си.

Набор данных

служит для связывания двух типов записей отношением «один-ко-многим».

Организация связей в сетевой модели



Преимущества сетевой модели

- стандартизация. Стандарт CODASYL определяет базовые понятия модели и формальный язык описания.
- быстроедействие сетевых БД данных сравнимо с таковым для иерархических БД.
- Полное использование теоретико-графовых моделей предоставляет высокий уровень абстракции описания предметных областей, не ограниченных иерархиями.
- гибкость доступа к данным через любую последовательность связанных записей, а не через их иерархию.

Недостатки сетевой модели

- жёсткость задаваемых структур и сложность реструктуризации схем БД.
- сложная структура управления памятью в транзакционных приложениях с частыми изменениями связей.