

Базы данных

Севрюков С.Ю. (<http://bit.ly/ask4me>)

Определение

База данных — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия[1].

База данных — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации[2].

Классификация

Классификация БД по модели данных:

- Иерархические,
- Сетевые,
- Реляционные,
- Объектные,
- Объектно-ориентированные,
- Объектно-реляционные,
- Документ-ориентированные.

Классификация БД по среде физического хранения:

- БД во вторичной памяти (традиционные): средой постоянного хранения является периферийная энергонезависимая память (вторичная память) — как правило, жёсткий диск. В оперативную память СУБД помещает лишь кэш и данные для текущей обработки;
- БД в оперативной памяти (in-memory databases): все данные находятся в оперативной памяти;
- БД в третичной памяти (tertiary databases): средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило, на основе магнитных лент или оптических дисков. Во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кэш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры;

Классификация БД по содержанию:

- Географические;
- Исторические;
- Научные;
- Мультимедийные.

Классификация БД по степени распределённости:

- Централизованные (сосредоточенные);
- Распределённые.

Реляционная алгебра

Реляционная алгебра — формальная система манипулирования отношениями в реляционной модели данных. Существует в двух несколько различающихся вариантах:

- Алгебра Кодда (Э. Кодд, 1970)
- Алгебра А (К. Дейт, Х. Дарвен)

Наряду с реляционным исчислением является способом получения результирующего отношения в реляционной модели данных.

Реляционная алгебра представляет собой набор операторов, использующих отношения в качестве аргументов и возвращающих отношения в качестве результата. Таким образом, реляционный оператор f выглядит как функция с отношениями в качестве аргументов:

$$R = f(R_1, R_2, \dots, R_n)$$

Реляционная алгебра является замкнутой, так как в качестве аргументов в реляционные операторы можно подставлять другие реляционные операторы, подходящие по типу:

$$R = f(f_1(R_{11}, R_{12}, \dots), f_2(R_{21}, R_{22}, \dots), \dots)$$

В реляционных выражениях можно использовать вложенные выражения сколь угодно сложной структуры.

Нормальные формы

Нормализация может применяться к таблице, которая представляет собой правильное отношение.

Первая нормальная форма (1NF)

Отношение находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

Вторая нормальная форма (2NF)

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме, и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого возможного ключа. Функционально полная зависимость означает, что атрибут функционально зависит от всего составного потенциального ключа, но при этом не находится в функциональной зависимости от какой-либо из входящих в него частей. Или другими словами: в 2NF нет неключевых атрибутов, зависящих от части составного ключа.

Третья нормальная форма (3NF)

Согласно определению Кодда, таблица находится в 3NF тогда и только тогда, когда выполняются следующие условия:

- Отношение R (таблица) находится во второй нормальной форме;
- Каждый непервичный атрибут R находится в нетранзитивной (то есть прямой) зависимости от каждого ключа R.

Таким образом, отношение находится в 3NF тогда и только тогда, когда оно находится во 2NF и отсутствуют транзитивные зависимости неключевых атрибутов от ключевых. Транзитивной зависимостью неключевых атрибутов от ключевых называется следующая: $A \rightarrow B$ и $B \rightarrow C$, где A — набор ключевых атрибутов (ключ), B и C — различные множества неключевых атрибутов.

Нормальная форма Бойса — Кодда (BCNF)

Это более строгая версия третьей нормальной формы.

Отношение находится в BCNF тогда и только тогда, когда каждая ее нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

Четвёртая нормальная форма (4NF)

Таблица находится в 4NF, если она находится в BCNF и не содержит нетривиальных многозначных зависимостей.

Пятая нормальная форма (5NF)

Таблица находится в 5NF, если она находится в 4NF и любая многозначная зависимость соединения в ней является тривиальной. Пятая нормальная форма в большей степени является теоретическим исследованием и практически не применяется при реальном проектировании баз данных.

Так же существуют:

- Доменно-ключевая нормальная форма (DKNF);
- Шестая нормальная форма (6NF).

SQL

SQL — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. SQL основывается на реляционной алгебре.

Операторы SQL делятся на:

- операторы определения данных (Data Definition Language, DDL);
- операторы манипуляции данными (Data Manipulation Language, DML);
- операторы определения доступа к данным (Data Control Language, DCL);
- операторы управления транзакциями (Transaction Control Language, TCL).

Обеспечение целостности данных

Транзакции

Транзакция — группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта. Транзакции обрабатываются транзакционными системами, в процессе работы которых создаётся история транзакций.

Различают последовательные (обычные), параллельные и распределённые транзакции. Распределённые транзакции подразумевают использование больше чем одной транзакционной системы и требуют намного более сложной логики (например, two-phase commit — двухфазный протокол фиксации транзакции). Также, в некоторых системах реализованы автономные транзакции, или под-транзакции, которые являются автономной частью родительской транзакции.

Одним из наиболее распространённых наборов требований к транзакциям и транзакционным системам является набор ACID (Atomicity, Consistency, Isolation, Durability). Вместе с тем, существуют специализированные системы с ослабленными транзакционными свойствами.

Уровень изоляции

0 — **Неподтверждённое чтение** (Read Uncommitted, Dirty Read, грязное чтение) — чтение незафиксированных изменений своей транзакции и конкурирующих транзакций, возможны нечистые, неповторяемые чтения и фантомы.

1 — **Подтверждённое чтение** (Read Committed) — чтение всех изменений своей транзакции и зафиксированных изменений конкурирующих транзакций, нечистые чтения невозможны, возможны неповторяемые чтения и фантомы.

2 — **Повторяемое чтение** (Repeatable Read, Snapshot) — чтение всех изменений своей транзакции, любые изменения, внесённые конкурирующими транзакциями после начала своей недоступны, нечистые и неповторяемые чтения невозможны, возможны фантомы.

3 — **Упорядоченный** — (Serializable, сериализуемый) — упорядоченные (сериализуемые) транзакции. Идентичен ситуации при которой транзакции выполняются строго последовательно одна после другой. То есть транзакции, результат действия которых не зависит от порядка выполнения шагов транзакции (запрещено чтение всех данных изменённых с начала транзакции, в том числе и своей транзакцией). Фантомы невозможны.

Чем выше уровень изоляции, тем больше требуется ресурсов, чтобы их поддерживать.

В СУБД уровень изоляции транзакций можно выбрать как для всех транзакций сразу, так и для одной конкретной транзакции. По умолчанию в большинстве баз данных используется уровень 1 (Read Committed). Уровень 0 используется в основном для отслеживания изменений длительных транзакций или для чтения редко изменяемых данных. Уровни 2 и 3 используются при повышенных требованиях к изолированности транзакций.

Современные СУБД

Промышленные коммерческие СУБД:

- Microsoft SQL Server [5];
- Oracle[6];
- DB2[7].

Свободно-распространяемые (в частности, с открытым исходным кодом) СУБД:

- PostgreSQL [8]
- MySQL [9]
- Firebird [10]

Документо-ориентированные СУБД:

- CouchDB [11]
- MongoDB [12]
- Redis [13]

Рекомендуемые источники и литература

1. Дейт К. Дж. Введение в системы баз данных. — 8-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2005
2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — 3-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2003
3. Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007
4. [Раздел курсов на сайте Интернет университета информационных технологий, посвященный БД](#)
5. <http://microsoft.com/sql>
6. <http://www.oracle.com/database/>
7. [Раздел DB2 на сайте IBM](#)
8. www.postgresql.org
9. www.mysql.com
10. www.firebirdsql.org
11. couchdb.apache.org
12. www.mongodb.org
13. redis.io