



БАЗЫ ДАННЫХ

Понятия. Функции. Язык SQL



История возникновения баз данных

В широком смысле:

- База данных как любое средство хранения и обработки данных
- Зародилось в 4000 г. до н.э. в древнем Шумере
- Размытие такого понятия БД до «архива» или даже «письменности»

История возникновения баз данных

В узком смысле:

- Традиционное понимание БД
- 1955 год – оборудование для обработки записей на основе перфокарт
- 1960 год – оперативные сетевые базы данных, понятие схемы базы данных
- 1970 год – Эдгар Ф. Кодд ввел понятие реляционной модели данных

Понятие базы данных

База данных (БД) – совокупность материалов, систематизированных для поиска и обработки с помощью вычислительных систем.

СУБД (система управления БД) – совокупность программных средств для управления созданием и использованием баз данных.

Признаки БД

- БД хранится и обрабатывается в вычислительной системе
- Данные в БД логически структурированы
- БД включает схему (метаданные), описывающие логическую структуру БД

Модели данных

- Иерархическая (древовидная)
- Сетевая
- Объектно-ориентированная
- Реляционная (табличная)
 - *MySQL, MSSQL, MS Access, Firebird, Interbase*
- Объектно-реляционная
 - *PostgreSQL, Oracle Database*

Среды хранения данных

- В оперативной памяти
- Во вторичной памяти (жесткий диск)
- В третичной памяти (отсоединяемое устройство массового хранения: магнитные ленты, оптические диски)

Способ доступа к БД

- **Файл-серверные СУБД**

- *MS Access, dBase, FoxPro*

- **Клиент-серверные СУБД**

- *MySQL, MSSQL, Oracle Database, PostgreSQL, Firebird*

- **Встраиваемые СУБД**

- *SQLite, Firebird Embedded, MSSQL Compact*

Функции СУБД

- Управление данными во внешней и оперативной памяти
- Журнализация изменений
- Управление транзакциями
- Резервное копирование
- Репликация данных
- Восстановление данных после сбоев
- Поддержка языков баз данных (SQL)

Транзакция и логическая целостность

Транзакция – последовательность операций над БД, рассматриваемых как единое целое.

Либо транзакция успешно выполняется и СУБД фиксирует все изменения БД, либо ни одно из этих изменений никак не отражается на состоянии БД.

Понятие транзакции необходимо для поддержания логической целостности БД.

Целостность БД – соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.

VLDB (Very Large DB)

- Teradata
- Netezza
- Exadata
- MapReduce
- Oracle Big Data Appliance
- Yandex ClickHouse
- Apache Hadoop

Реляционная база данных

Основные понятия:

- База данных
- Таблица
- Столбец
- Ряд (строка)
- Значение «ячейки»

Язык баз данных

SQL – Structured Query Language

- язык структурированных запросов
- формальный не процедурный информационно-логический язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных под управлением СУБД.

Язык баз данных

По своему назначению операторы SQL разделяется на:

- **DDL** – Data Definition Language
– определение схемы данных
- **DML** – Data Manipulation Language
– манипулирование данными
- **DCL** – Data Control Language
– определение доступа к данным
- **TCL** – Transaction Control Language
– управление транзакциями

Язык баз данных

- У разных производителей СУБД разные диалекты SQL, несовместимые между собой.
- Разными производителями вводятся разные расширения процедурных возможностей языка (хранимые процедуры и надстройки).

NoSQL

Отрицание SQL

- Документо-ориентированные СУБД:
 - *MongoDB, CouchDB*
- Кеширующие СУБД:
 - *MemcachedDB, Redis*

NoSQL

Not only SQL

СУБД MySQL

- Кросс-платформенное ПО
- API для различных языков программирования
- Поддержка UTF-8
- Поддержка ACID и транзакций
- Поддержка хранимых процедур и функций
- Поддержка репликации данных
- Высокая надежность и производительность
- Стандарт «де-факто» для веб-приложений и сайтов

Движки MySQL

- MyISAM (*не рекомендуется!*)
- InnoDB
- Maria (MariaDB)
- XtraDB (Percona Server)

Интерфейсы доступа

- Консольный клиент (mysql)
- API-доступ из различных ЯП
- MySQL Workbench
- HeidiSQL, DBeaver, Navicat, Sequel Pro
- Веб-приложения: phpMyAdmin, Adminer

SQL-запросы

– Совокупность операторов и вычисляемых инструкций.

Как правило, это «*предложение на человеческом языке*», которое всегда начинается с глагола в повелительном наклонении:

```
SELECT name FROM clients  
WHERE money > 100 LIMIT 10;
```

```
ВЫБЕРИ столбец имя ИЗ таблицы клиентов  
ГДЕ денег > 100 С ЛИМИТОМ В 10;
```

SQL-запросы

Запросы могут записаны на нескольких строчках.

Каждый SQL-запрос заканчивается точкой с запятой: ;

-- два дефиса (а также символ решетки #)
обозначают начало однострочного комментария,
содержимое игнорируется до конца строки

/* многострочные комментарии
тоже существуют 😊
однако, любые кавычки здесь лучше не
использовать ☹ */

Запись чисел

Числа записываются «как есть»:

10

-5

1.23

Математические действия вполне привычны:

1 + 2

4 * (5 - 2)

25 / 10

Запись строк

Строки записываются в одинарных кавычках:

`'Hello world'`

MySQL допускает строки в двойных кавычках:

`"Hello world"`, если НЕ включен режим ANSI_QUOTES.

По стандарту SQL в двойных же кавычках записывается идентификатор (имена БД, таблиц, столбцов и пр.).

Идентификаторы

Идентификаторы (имена БД, таблиц, столбцов и пр.). помещаются в одинарные обратные кавычки («бэктики»):

``name`` (например, имя столбца name).

Квотирование идентификаторов может быть опущено, если слово не является оператором или зарезервированным ключевым словом.

Сравнения

Операторы сравнения вполне традиционны:

a > 2

b < 5

a >= b

b <= c

d = 10

f != 3

f <> 3

Сравнения

Операторы сравнения по диапазону:

a **BETWEEN 1 AND 10**

b **NOT BETWEEN 10 AND 20**

Операторы сравнения по списку:

a **IN (1, 3, 5, 9)**

b **NOT IN ("John", "Mary")**

Логические операторы

a = 1 **AND** b = 2

a = 1 **OR** a = 2

Файлы SQL-запросов

- Набор запросов может быть сохранен в текстовый файл с расширением ***.sql**.
- Запросы из файла могут быть отправлены в СУБД для исполнения.
- Обычно если запросы содержат инструкции по созданию таблиц и вставки данных в них, то такой SQL-файл называют «резервная копия» («дамп», от англ. *dump*).
- Большие дампы хранят в сжатом виде, чаще в gzip (реже в zip).

DDL-операторы SQL

- CREATE
- DROP
- ALTER

DML-операторы SQL

- SELECT
- INSERT
- UPDATE
- DELETE

DCL-операторы SQL

- GRANT
- REVOKE

TCL-операторы SQL

- BEGIN
- COMMIT
- ROLLBACK

Запросы на выборку

Получение данных из таблиц осуществляется с помощью оператора **SELECT**:

```
SELECT столбец1, столбец2, столбец3  
[ FROM таблица ]  
[ WHERE условие ]  
[ LIMIT количество ] ;
```

Запросы на выборку

Например:

```
SELECT `name`, `lastname` FROM `users` LIMIT 5;
```

```
SELECT `name`, `email` FROM `users`;
```

```
SELECT `name` FROM `users` WHERE `type` = 'ADMIN';
```

```
SELECT 'Hello world';
```

Запросы на выборку

Если нужно получить все столбцы, то можно* использовать «звёздочку»:

```
SELECT * FROM `users` LIMIT 5;
```

Здесь «звёздочка» (*) – это все столбцы указанной таблицы.

* Не стоит использовать * в выборке без необходимости, т.к. столбцов в таблице может быть значительно больше, чем нужно для запроса!

Кроме того, не стоит опускать и оператор **LIMIT**, без него вы получите слишком много строк (*все строки*) из таблицы!

Ограничение выборки

Оператор `LIMIT [n,] m`

где **n** – кол-во строк для пропуска, **m** – кол-во строк для выборки.

```
SELECT * FROM `products` LIMIT 10;
```

```
SELECT * FROM `products` LIMIT 0, 10;
```

```
SELECT * FROM `products` LIMIT 10, 10;
```

Получение числа записей выборки

Функция COUNT(*)

```
SELECT COUNT(*) FROM `products`;
```

```
SELECT COUNT(*) FROM `users`  
WHERE `type` = 'ADMIN';
```