MySQL - RDBMS

Trainer: Mr. Nilesh Ghule

Database logical layout

- Database/schema is like a namespace/container that stores all db objects related to a project. e.g. classwork, sales, ...
- It contains tables, constraints, relations, stored procedures, functions, triggers, ...
- There are some system databases e.g. mysql, performance_schema, information_schema, sys, ... They contains db internal/system information.
  - e.g. SELECT user, host FROM mysql.user;
- A database contains one or more tables.
- Tables have multiple columns.
- Each column is associated with a data-type.
- Columns may have zero or more constraints. → restrictions on data values.
- The data in table is in multiple rows.
- Each row have multiple values (as per columns).

Database physical layout

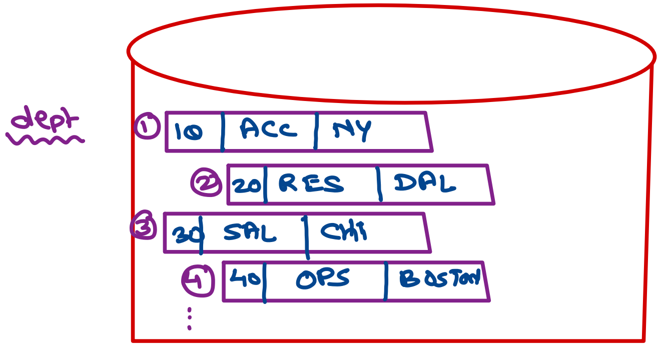
linux terminal> stat filepath
- shows metadata of a file (from inode).
- IO Block (4096). Blocks: 8+ (num of sectors = 512B).

Ubuntu

- In MySQL, the data is stored on disk in its data directory i.e. /var/lib/mysql
- Each database/schema is a separate sub-directory in data dir. e.g. classwork sales
- Each table in the db, is a file on disk.
- e.g. student table in current db is stored in file /var/lib/mysql/db/student.ibd.

  Assuming InnoDb engine
- Data is stored in binary format.
- A file may not be contiguously stored on hard disk.
- Data rows are not contiguous. They are scattered in the hard disk.
- In one row, all fields are consecutive.
- When records are selected, they are selected in any order.

select * from tablename;



① 10 | ACC | NY
② 20 | RES | DAL
③ 30 | SAL | CHI
④ 40 | OPS | BOSTON
⋮

MySQL data types

SMALL INT - 2 bytes - short → 1 bit ± 15 bit data } ± 2^{15} ± 32768

0-65535 2^{16} - 16 bits data ← unsigned short

MySQL → INT

Oracle → NUMBER

Derby → Integer

- RDBMS have similar data types (but not same).
- MySQL data types can be categorised as follows
  - Numeric types (Integers)
    - TINYINT (1 byte), SMALLINT (2 byte), MEDIUMINT (3 byte), INT (4 byte), BIGINT (8 byte), BIT(n bits)
    - integer types can signed (default) or unsigned.
  - Numeric types (Floating point)
    - approx. precision – FLOAT (4 byte), DOUBLE (8 byte) I DECIMAL(m, n) – exact precision
  - Date/Time types
    - DATE, TIME, DATETIME, TIMESTAMP, YEAR
  - String types – size = number of chars * size of char
    - CHAR(1-255) – Fixed length, Very fast access.
    - VARCHAR(1-65535) – Variable length, Stores length + chars.
    - TINYTEXT (255), TEXT (64K), MEDIUMTEXT (16M), LONGTEXT (4G) – Variable length, Slower access.
  - Binary types – size = number of bytes
    - BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
  - Miscellaneous types
    - ENUM, SET

xx xxxxx x.x xxx

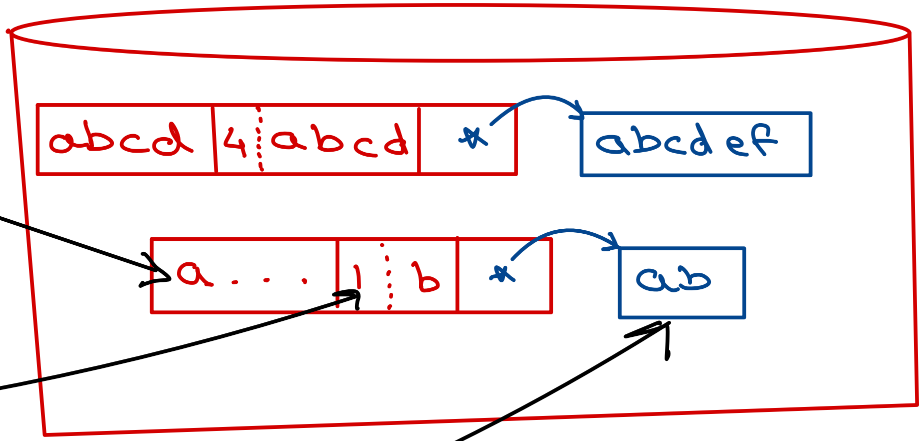
← n →

← m →

CHAR vs VARCHAR vs TEXT

- CHAR
  gender CHAR(6)
  mobile CHAR(14)
  answer CHAR(1) → Y/N
  - Fixed inline storage. (max 255 chars)
  - If smaller data is given, rest of space is unused.
  - Very fast access.
- VARCHAR
  addr VARCHAR(100)
  review VARCHAR(200)
  - Variable inline storage. (max 65536 chars)
  - Stores length and characters.
  - Slower access than CHAR.
- TEXT
  - Variable external storage.
    blog TEXT
    article TEXT
  - Very slow access.
  - Not ideal for indexing.
- CREATE TABLE temp(c1 CHAR(4), c2 VARCHAR(4), c3 TEXT(4));
  x
- DESC temp;
  (4)
  TINYTEXT (max 255 chars)
- INSERT INTO temp VALUES('abcd', 'abcd', 'abcdef');

INSERT INTO temp VALUES('a', 'b', 'ab');



INSERT – DML

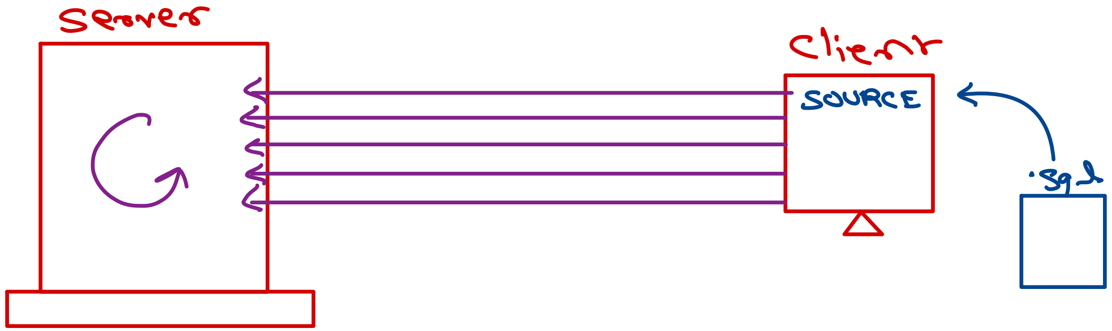
- Insert a new row (all columns, fixed order).
  - INSERT INTO table VALUES (v1, v2, v3);
- Insert a new row (specific columns, arbitrary order).
  - INSERT INTO table(c3, c1, c2) VALUES (v3, v1, v2);
  - INSERT INTO table(c1, c2) VALUES (v1, v2);
  - Missing columns data is NULL.
  - NULL is special value and it is not stored in database.
- Insert multiple rows.
  - INSERT INTO table VALUES (av1, av2, av3), (bv1, bv2, bv3), (cv1, cv2, cv3);
- Insert rows from another table.
  - INSERT INTO table SELECT c1, c2, c3 FROM another-table;
  - INSERT INTO table (c1,c2) SELECT c1, c2 FROM another-table;

SQL scripts

- SQL script is multiple SQL queries written into a .sql file.
- SQL scripts are mainly used while database backup and restore operations.
- SQL scripts can be executed from terminal as:
  - terminal> mysql –u user –ppassword db < /path/to/sqlfile

input redirection

- SQL scripts can be executed from command line as:
  - mysql> SOURCE /path/to/sqlfile ✓
- Note that SOURCE is MySQL CLI client command.
- It reads commands one by one from the script and execute them on server.



SELECT – DQL

- Select all columns (in fixed order).
  - SELECT * FROM table;
- Select specific columns / in arbitrary order.
  - SELECT c1, c2, c3 FROM table;
- Column alias
  - SELECT c1 AS col1, c2 col2 FROM table;
- Computed columns.
  - SELECT c1, c2, c3, expr1, expr2 FROM table;

```
SELECT c1,
CASE WHEN condition1 THEN value1,
WHEN condition2 THEN value2,
…
ELSE valuen
END
FROM table;
```

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>