MySQL - RDBMS

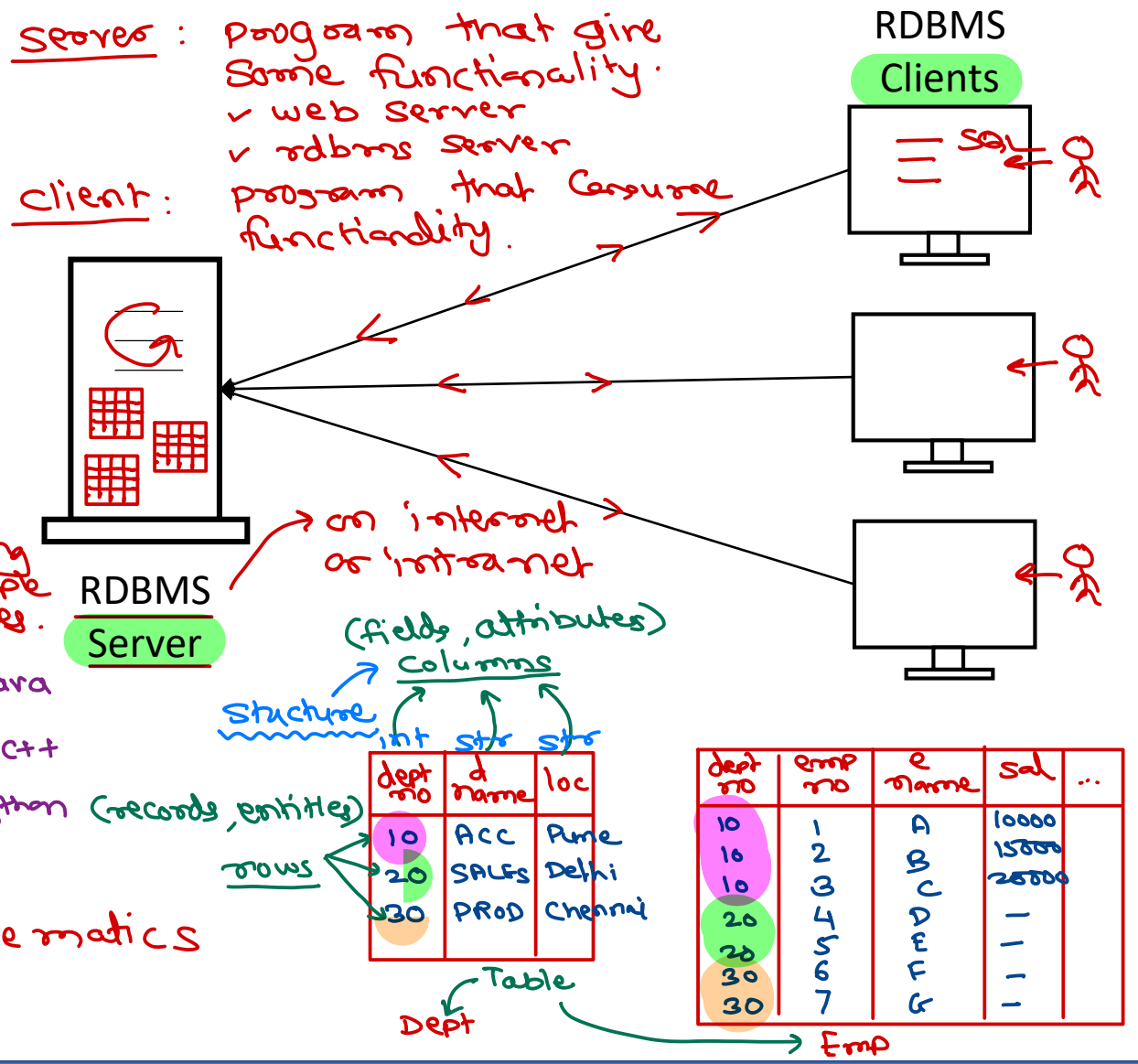
Trainer: Mr. Nilesh Ghule

DBMS

- Any enterprise application need to manage data.
- In early days of software development, programmers store data into files and does operation on it. However data is highly application specific.
- Even today many software manage their data in custom formats e.g. Tally, Address book, etc.
- As data management became more common, DBMS systems were developed to handle the data. This enabled developers to focus on the business logic e.g. FoxPro, DBase, Excel, etc.
- At least CRUD (Create, Retrieve, Update and Delete) operations are supported by all databases.
- Traditional databases are file based, less secure, single-user, non-distributed, manage less amount of data (MB), complicated relation management, file-locking and need number of lines of code to use in applications.

RDBMS

- RDBMS is relational DBMS.
- It organizes data into Tables, rows and columns. The tables are related to each other.
- RDBMS follow table structure, more secure, multi-user, server-client architecture, server side processing, clustering support, manage huge data (TB), built-in relational capabilities, table-locking or row-locking and can be easily integrated with applications.
- e.g. DB2, Oracle, MS-SQL, MySQL, MS-Access, SQLite, …
- RDBMS design is based on Codd's rules developed at IBM (in 1970).



SQL - Structured Query Language (wiki)

- Clients send SQL queries to RDBMS server and operations are performed accordingly.
- Originally it was named as RQBE (Relational Query By Example).
- SQL is ANSI standardised in 1987 and then revised multiple times adding new features. Recent revision in 2016.
- SQL is case insensitive. → db name & table name are case sensitive on Linux. (MySQL)
  select
  SELECT
- There are five major categories: → table structure
  - DDL: Data Definition Language e.g. CREATE, ALTER, DROP, RENAME.
  - DML: Data Manipulation Language e.g. INSERT, UPDATE, DELETE. ← rows/records
  - DQL: Data Query Language e.g. SELECT. → rows/records
  - DCL: Data Control Language e.g. CREATE USER, GRANT, REVOKE. ← security
  - TCL: Transaction Control Language e.g. SAVEPOINT, COMMIT, ROLLBACK. ← concurrent users
- Table & column names allows alphabets, digits & few special symbols.
- If name contains special symbols then it should be back-quotes.
- e.g. Tbl1, `T1#`, `T2$` etc. Names can be ~~max~~ 30 chars long.

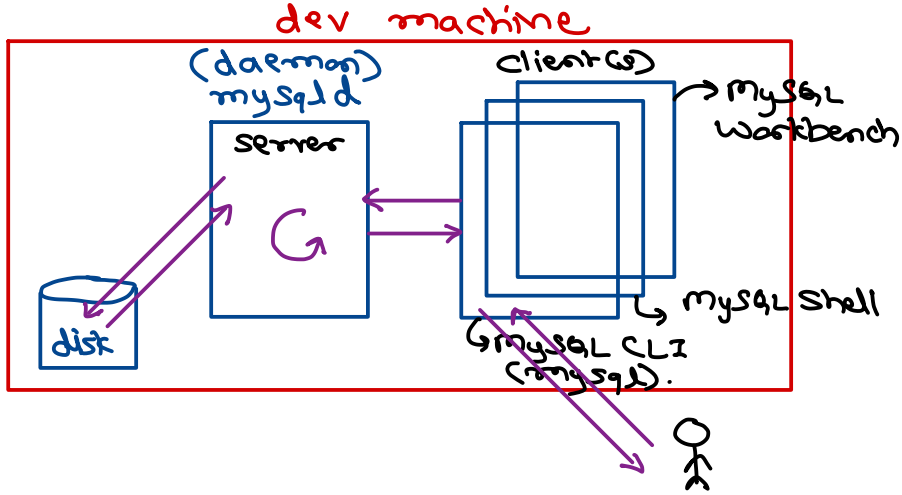
MySQL → developed C/C++.

- Developed by Michael Widenius in 1995. It is named after his daughter name Myia.
- Sun Microsystems acquired MySQL in 2008.
- Oracle acquired Sun Microsystem in 2010.
- MySQL is free and open-source database under GPL. However some enterprise modules are close sourced and available only under commercial version of MySQL.
- MariaDB is completely open-source clone of MySQL.
- MySQL support multiple database storage and processing engines.
- MySQL versions:
  - < 5.5: MyISAM storage engine
  - 5.5: InnoDb storage engine
  - 5.6: SQL Query optimizer improved, memcached style NoSQL
  - 5.7: Windowing functions, JSON data type added for flexible schema
  - 8.0: CTE, NoSQL document store.
- MySQL is database of year 2019 (in database engine ranking).

MySQL → Community (Free)
MySQL → Enterprise (paid)

MySQL installation on Ubuntu/Linux

- terminal> sudo apt-get install mysql-community-server mysql-community-client
- This installs MySQL server (mysqld) and MySQL client (mysql).
- MySQL Server (mysqld)
  - Run as background process.
  - Implemented in C/C++.
  - Process SQL queries and generate results.
  - By default run on port 3306.
    - Controlled via systemctl.
      - terminal> sudo systemctl start|stop|status|enable|disable mysql
- MySQL client (mysql)
  - Command line interface
  - Send SQL queries to server and display its results.
  - terminal> mysql –u root -p
- Additional MySQL clients
  - MySQL workbench
  - PHPMyAdmin



Getting started

- root login can be used to perform CRUD as well as admin operations.
- It is recommended to create users for performing non-admin tasks.
  - mysql> CREATE DATABASE db;
  - mysql> SHOW DATABASES;
  - mysql> CREATE USER dbuser@localhost IDENTIFIED BY 'dbpass';
  - mysql> SELECT user, host FROM mysql.user;
  - mysql> GRANT ALL PRIVILEGES ON db.* TO dbuser@localhost;
  - mysql> FLUSH PRIVILEGES;
  - mysql> EXIT;
- terminal> mysql –u dbuser –pdbpass
  - mysql> SHOW DATABASES;
  - mysql> SELECT USER(), DATABASE();
  - mysql> USE db;
  - mysql> SHOW TABLES;
  - mysql> CREATE TABLE student(id INT, name VARCHAR(20), marks DOUBLE);
  - mysql> INSERT INTO student VALUES(1, 'Abc', 89.5);
  - mysql> SELECT * FROM student;

Database physical layout

- 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.
- 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.
- 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.

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;

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>