Database Technologies

1. Test speakers from Zoom settings
2. Keep your notebook ready for quick notes
3. Check code sharing utility

Agenda

0. RDBMS
1. Database logical vs physical layout
2. Getting started
3. MySQL data-types
4. CREATE table
5. SQL scripts
6. ~~INSERT query~~
7. ~~SELECT query~~

RDBMS Software

- Oracle: Most popular RDBMS. Powerful, Efficient. Many features SQL + PL-SQL. Enterprise (Commercial).
- MS-SQL: Microsoft. Popular RDBMS. Powerful, Efficient. Many features. Enterprise (Commercial).
- MySQL: Most Popular Open-Source RDBMS. Efficient. Community/Enterprise.
- Maria: Open-Source clone of MySQL. Community.
- Postgre-SQL: Open-Source RDBMS. Popular on UNIX/Linux. High performance.
- Informix: Low-level, High Performance database.
- Cybase: RDBMS.
- DB2, Telon: RDBMS on Mainframe.
- Derby: Open-source. Single user database. By default available with JDK-8.0.
- SQLite: Open-source. File-based. Light-weight. Single user database. Android mobile. Data transfer/prototype.
- MS-Access: Open-source. File-based. Single user database.

MySQL installation

- MySQL server (mysqld.exe)
  - Listen on port 3306
  - Accept SQL queries, process data, produce result, and send back to the client.
- MySQL clients
  - MySQL CLI (mysql.exe)
    - cmd> mysql -u root -h localhost -p
      - -h --> server host (server name, ip address, or localhost)
      - -p --> accept password.
    - cmd> mysql -u root -h localhost -pmanager
      - -ppassword --> no space after -p
      - insecure (password is visible).
    - cmd> mysql -u root -p

        - -h if not given, default is localhost (server running on same computer as of client)
    - MySQL Shell (mysqlsh.exe)
        - New in MySQL 8.0 +
        - JS/Python interface(syntax) to MySQL server
        - Advanced features of MySQL 8.0+ (e.g. JSON processing, ...)
    - MySQL Workbench
        - GUI client
            - See all tables, execute SQL queries, ...
- MySQL Utilities -- used Db admin (DBA)
    - mysqldump -- take backup
    - mysqlrestore -- restore backup
    - ...

MySQL Logical vs Physical Layout

- Where data is stored?
    - MySQL Data is stored -- Data directory.
    - Windows: C:\ProgramData\MySQL\MySQL Server 8.0\Data
    - Linux (Ubuntu): /var/lib/mysql
- CREATE DATABASE classwork; OR CREATE SCHEMA classwork;
    - Database/Schema --> Logical Container for all the data tables, constraints, procedures, triggers, ...
    - SCHEMA --> Project
    - Schema --> Directory/Folder in Data directory
- CREATE TABLE students ...
    - Table --> Logical collection of rows/columns.
        - Column -- Fields/Attributes -- Data type
            - DESCRIBE students;
        - Row -- Record
            - SELECT * FROM students;
    - Table --> Binary file in Schema directory (inside data directory).
        - MySQL file extension -- .ibd (Innodb storage).
        - Table structure (column info) + data (rows/records)
        - Row is internally stored contiguous. [1, Superman, 95.0]
        - Multiple rows are scattered in the disk.

Getting Started

cmd> mysql -u root -p

- SQL queries always executed on mysql> prompt not on basic command prompt.
- Comments in SQL start with --.
- User creation, database creation and assigning permissions can be done with "root" login only.

```
-- Create a database (done yesterday).
CREATE DATABASE classwork;
```

```
SHOW DATABASES;

-- Create a new user (more details in DCL lecture).
--  User name "sunbeam" can access from "localhost".
--  Password "sunbeam"
CREATE USER sunbeam@localhost IDENTIFIED BY 'sunbeam';

-- Give permissions to user on database.
--  give all permissions on classwork database to sunbeam user.
GRANT ALL PRIVILEGES ON classwork.* TO sunbeam@localhost;
-- activate permissions
FLUSH PRIVILEGES;

EXIT;
```

cmd> mysql -u sunbeam -psunbeam

```
-- print current user and current database
SELECT USER(), DATABASE();

SHOW DATABASES;

-- activate classwork database
USE classwork;

SELECT USER(), DATABASE();

SHOW TABLES;

DESCRIBE students;

SELECT * FROM students;
-- "*" means all columns
```

Login into MySQL CLI

- cmd> mysql -u sunbeam -psunbeam

```
SELECT USER(), DATABASE();

USE classwork;

SELECT USER(), DATABASE();

EXIT;
```

- cmd> mysql -u sunbeam -psunbeam classwork
- Login with sunbeam user and sunbeam password and also select classwork database.

```
SELECT USER(), DATABASE();
```

SQL - Structured Query Language

- Structured -- Fixed Syntax.
- Categories:
  - DDL, DML, DQL, DCL, TCL

CREATE TABLE

- DDL
- Syntax: CREATE TABLE table_name(col1 DATATYPE1, col2 DATATYPE1, col3 DATATYPE1, ...);
- Example (yesterday):
  - CREATE TABLE students(roll INT, name CHAR(20), marks DOUBLE);
    - students --> table_name
    - roll --> col1 name
    - name --> col2 name
    - marks --> col3 name
    - INT, CHAR(), DOUBLE --> data types
- Table & Database names are case sensitive on Linux. Not case sensitive on Windows.
- Rest all keywords and names are case insensitive (on all OS).

Data Types

- May vary from RDBMS to RDBMS.
- MySQL data types
  - Numerical types (Integer or Floating points)
  - Date/Time types
  - String types
  - Binary types
  - Misc types.

CHAR vs VARCHAR vs TEXT

```
DROP TABLE temp;

CREATE TABLE temp(c1 CHAR(4), c2 VARCHAR(4), c3 TEXT(4));
-- TEXT(4) --> 4 is ignored --> TINYTEXT max size is 255 chars.

SHOW TABLES;
DESCRIBE temp;

SELECT * FROM temp;

INSERT INTO temp VALUES('A', 'A', 'A');
INSERT INTO temp VALUES('ABCD', 'ABCD', 'ABCD');
```

```
SELECT * FROM temp;

INSERT INTO temp VALUES('ABCDEF', 'ABCD', 'ABCDEF'); -- error
INSERT INTO temp VALUES('ABCD', 'ABCDEF', 'ABCDEF'); -- error
INSERT INTO temp VALUES('ABC', 'ABC', 'ABCDEFGHI'); -- allowed (TINYTEXT -- max 255 chars)

SELECT * FROM temp;
```

INSERT Query

- We will continue this in NEXT session.
- Syntax: INSERT INTO tablename VALUES(col1_value, col2_value, col3_value, ...);