CptS 415 Big Data

Structured Query Language

Srini Badri

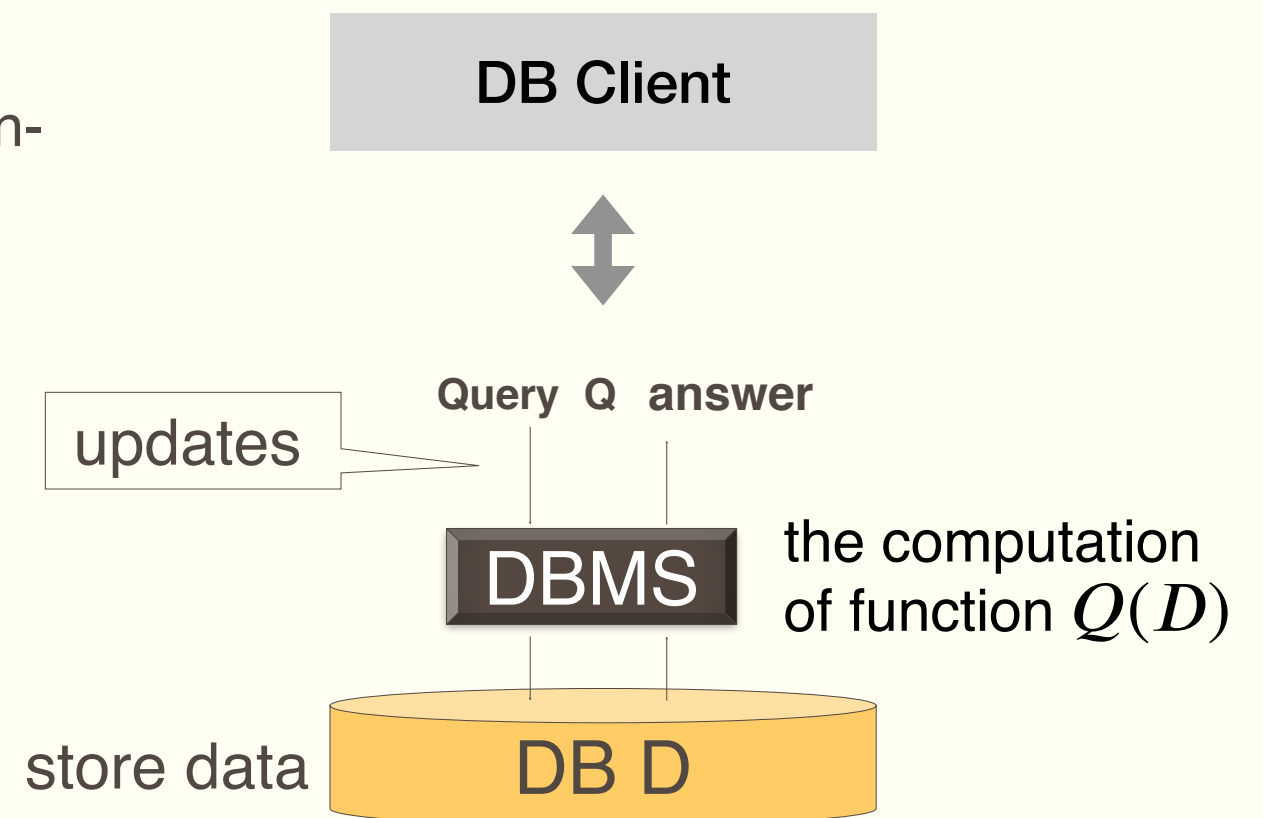# Structured Query Language (SQL)

- ## SQL
  - Simple language for perform "query" on relational database
  - Query can contain statements for creating, deleting, updating databases and tables (relations)
  - Supported by both relational and non-relational database servers

DB Client

Query Q   answer

updates

DBMS   the computation of function $Q(D)$

store data   DB D

SQL Statement

- ## SQL Statement Syntax
  - Command <database or table name> <clause> ;

- ## Types of SQL Statements:

  - Database administration (user management, backup and restore, etc.)

  - Data management (adding, updating, deleting tables and records)

  - Transforming raw data to useful information

- ## Variation among DBMS Products:

  - Statements and Data Types vary slightly from vendor to vendor

Data Types

- Integer data:

  - TINYINT, SMALLINT, INTEGER (INT), BIGINT

- Floating-point data:

  - DECIMAL, NUMERIC

  - FLOAT, DOUBLE

- Text data:

  - CHAR, VARCHAR, TEXT

- Binary data:

  - BIT

  - BINARY, VARBINARY, BLOB

- Date and time data:

  - DATE, TIME, DATETIME, TIMESTAMP

- Json data:

  - JSON

SQL - Database Statements

- Create Database:

  - CREATE DATABASE <database_name>;

- Switch Database:

  - USE <database_name>;

- List Databases:

  - SHOW DATABASES;

- Delete Database:

  - DROP DATABASE <database_name>

- Examples
  - CREATE DATABASE DemoDB;
  - SHOW DATABASES;
  - USE DemoDB;
  - DROP DATABASE DemoDB;

SQL - Table (Relation) Statements

- Create Table:

    - CREATE TABLE <table_name> ([<attribute> <data_type> <attribute_constraint>]);

- Update/Alter Table:

    - ALTER TABLE <table_name> <constraint>;

- Drop Table:

    - DROP TABLE <table_name>;

- Table Schema:

    - DESCRIBE TABLE <table_name>;

- Table Constraints:

    - SHOW CREATE TABLE <table_name>;

SQL - Table (Relation) Statements

- Examples:

  - CREATE TABLE Job (

    JOB_CODE SMALLINT PRIMARY KEY,

    JOB_CLASS VARCHAR(30) NOT NULL,

    CHG_HOUR DECIMAL(5, 2) NOT NULL

    );

  - DESCRIBE Job;

  - DROP TABLE Job;

  - SHOW CREATE TABLE job;

SQL - INSERT Statement

- Syntax
  - INSERT INTO <table_name> ([<attribute_names>])
    VALUES ([<attribute_values>]);

- Examples

  - INSERT INTO JOB VALUES (500, 'Programmer', 35.75);

  - INSERT INTO JOB (JOB_CODE, JOB_CLASS, CHG_HOUR) VALUES (500, 'Programmer', 35.75);

SQL - SELECT Statement

- Syntax
  - SELECT <Attributes>

    FROM <Relation>

    WHERE <Condition>;

- Examples
  - SELECT * FROM PROJECT;

  - SELECT * FROM EMPLOYEE WHERE JOB_CODE = 501;

  - SELECT EMP_LNAME, EMP_FNAME FROM EMPLOYEE WHERE JOB_CODE = 501;

SQL - SELECT Statement (Multiple Conditions)

- ## Syntax
  - SELECT <Attributes>

    FROM <Relation>

    WHERE <Condition 1> **AND** <Condition 2>;

- ## Examples
  - SELECT EMP_LNAME, EMP_FNAME FROM EMPLOYEE WHERE JOB_CODE = 501 **AND** EMP_HIREDATE > "2000-01-01";

SQL - UPDATE Statement

- Syntax
  - UPDATE <table_name>
    SET <assignment>
    WHERE <condition>;

- Examples

  - UPDATE JOB SET CHG_HOUR = 55.00 WHERE JOB_CODE=500;

SQL - DELETE Statement

- Syntax
  - DELETE <Attributes>
    FROM <Relation>
    WHERE <Condition>;

- Examples

  - DELETE FROM JOB WHERE JOB_CODE = 500;

SQL - Reference

- Documents:

  - Refer to SQL statements documentation for the specific DBMS

- MySQL:

  - https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html

- PostgreSQL:

  - https://www.postgresql.org/docs/9.5/sql.html

MySQL and SQL Demo

MySQL - Community Server

- Download and Installation:

    - https://dev.mysql.com/downloads/

    - Choose appropriate version and download based on your OS

    - Follow installation step - choose appropriate security level

- Setup:

    - update PATH variable - eg. /usr/local/mysql/bin

- Verify Install:

    - Start MySQL Server

    - Launch Terminal or Command Prompt

    - Type 'mysql --version'

MySQL - Community Server

- Connect:

  - *mysql -h host -u user -p*

  - *mysql -u user -p*

- Disconnect:

  - *quit*

  - c*trl-d* in Mac / c*trl-z* in Windows

MySQL - Community Server

- Create Users:

  - CREATE USER 'demouser' IDENTIFIED BY 'demopwd';

- List Users:

  - SELECT USER, HOST FROM mysql.user;