MySQL & SQL

Lecture 2

Contents

- Introduction to SQL
- Create Database, Drop Database
- Create, drop, alter table
- Insert data
- Update data
- Delete data
- Data types
- Constraints

SQL

- SQL stands for **Structured Query Language**
- SQL lets you access and manipulate databases
- It allows all operations (such as Create, Insert, Update, Delete and others) in DBMS
- It is being used in ***MySQL, Oracle, SQL Server, Sybase, PostgreSQL***
- ***MySQL vs SQL***

SQL (more…)

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Most Common Command in SQL

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

SQL Syntax

- **A statement is end with semicolon (;)**
- **It is not case sensitive**
-
- *CREATE DATABASE **test**;*

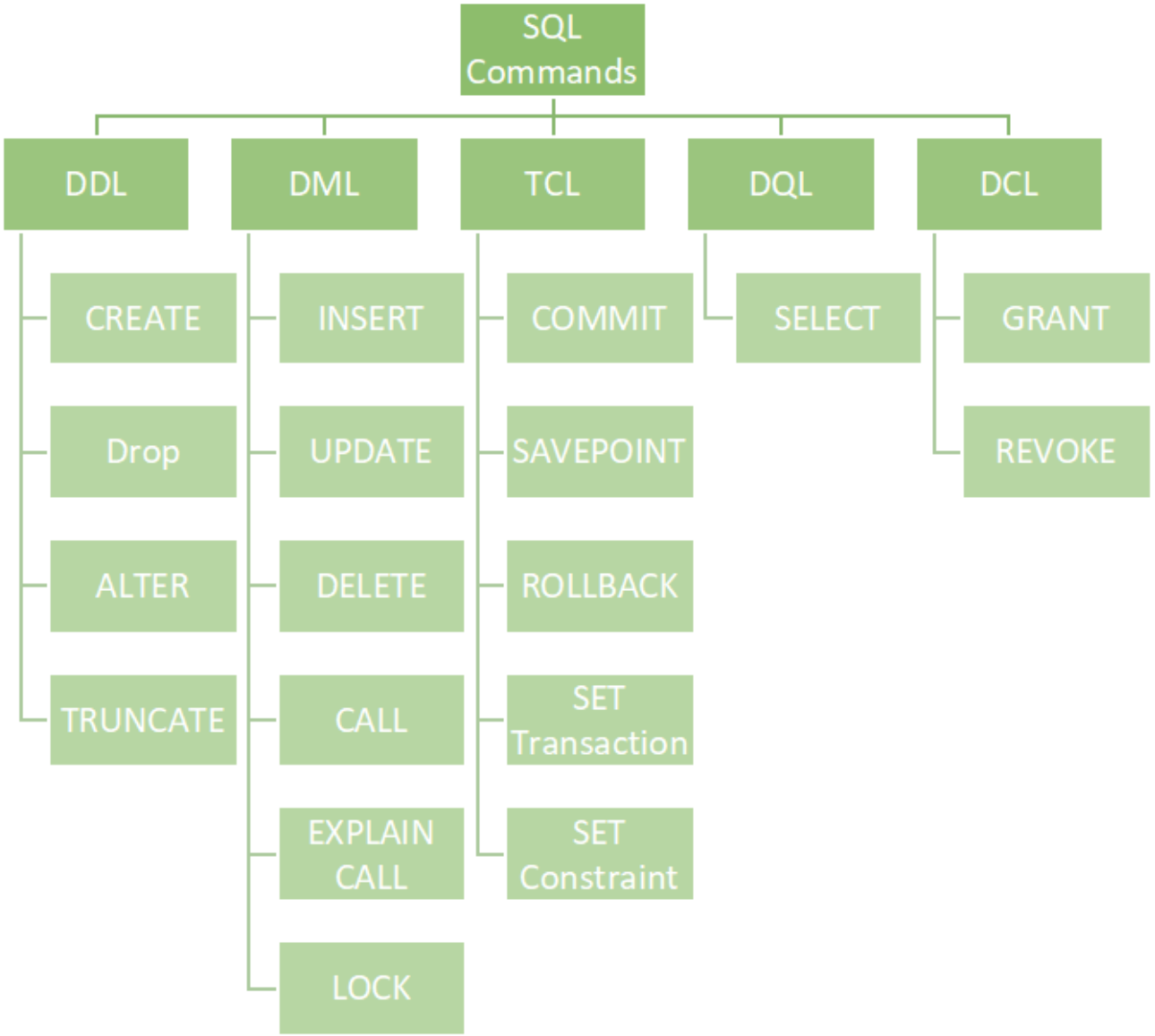
- *CREATE TABLE **students**(*

id INT NOT NULL AUTO_INCREMENT,
name VARCHAR(200),
phone VARCHAR(20),
dob DATE,
PRIMARY KEY(id)
);

SQL Commands

SQL Commands are mainly categorized into five categories:

- Data Definition Language (DDL)
  - *CREATE, ALTER, DROP*
- Data Manipulation Language (DML)
  - *SELECT, INSERT, UPADTE, DELETE*
- Data Control Language (DCL)
  - *GRANT, REVOKE*
- Data Query Language (DQL)-SELECT
- Transaction Control Language (TCL)
  - COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION


SQL Commands
DDL
DML
TCL
DQL
DCL
CREATE
Drop
ALTER
TRUNCATE
INSERT
UPDATE
DELETE
CALL
EXPLAIN CALL
LOCK
COMMIT
SAVEPOINT
ROLLBACK
SET Transaction
SET Constraint
SELECT
GRANT
REVOKE

Database Operations

- *CREATE DATABASE test;*
- *DROP DATABASE test;*

CREATE Table -Syntax

The CREATE TABLE statement is used to create a new table in a database.

Syntax :

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

CREATE Table (Example)

- *user(id, name, phone, password, date_of_birth)*

- ```
  CREATE TABLE user(
      id INT PRIMARY KEY AUTO_INCREMENT,
      name VARCHAR(100) NOT NULL,
      phone VARCHAR(15) NOT NULL UNIQUE,
      password VARCHAR(100) NOT NULL,
      Date_of_birth DATE
      );
  ```
  ```

DROP Table -Syntax

The **DROP TABLE** statement is used to drop an existing table in a database.

```
DROP TABLE table_name;
```

The **TRUNCATE TABLE** statement is used to delete the data inside a table, but not the table itself.

```
TRUNCATE TABLE table_name;
```

Data Types

Data Types	Example
INT(size)	Id INT
DECIMAL(size, d)	Salary DECIMAL(10,2)
VARCHAR(size)	Name VARCHAR(100)
ENUM(val1, val2, …..)	Gender ENUM('Male', 'Female')
SET(val1, val2,…..)	Skills SET('Python', 'database', 'Web')
DATE	Date_of_birth DATE
DATETIME	Order_date DATETIME

Constraints

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

Constraints (Examples)

Constraints	Example
PRIMARY KEY	Id INT PRIMARY KEY
NOT NULL	Salary DECIMAL(10,2) NOT NULL
UNIQUE	phone VARCHAR(15) UNIQUE
DEFAULT	Customer_name VARCHAR(100) DEFAULT 'Walking Customer'
CHECK	*age INT, CHECK (age>=18)*
FOREIGN KEY	*FOREIGN KEY(dept_id) REFERENCES department(id)*
INDEX	CREATE INDEX idx_phone ON user (phone);

Constraints Example

```
CREATE TABLE teacher(
      id INT PRIMARY KEY AUTO_INCREMENT,
      name varchar(100) NOT NULL,
      phone varchar(100) NOT NULL UNIQUE,
      joining_date datetime DEFAULT CURRENT_TIMESTAMP,
      dept_name varchar(100) DEFAULT ‘CSE’,
      age INT,
      CHECK (age>=20)

);
```

CREATE Tables
(including data types & Constraints)

- *departments (id, name, location, hod, phone)*
- *students (id, name, roll, phone, address, dob, **dept_id**)*

- *CREATE TABLE department(*

```
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(200) NOT NULL,
location VARCHAR(200) NULL,
hod VARCHAR(100) NULL,
phone VARCHAR(50) UNIQUE,
);
```

- *departments (id, name, location, hod, phone)*
- *students (id, name, roll, phone, address, dob, **dept_id**)*

- ```
  CREATE TABLE student(
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(200) NOT NULL,
  roll VARCHAR(100) NOT NULL UNIQUE,
  phone VARCHAR(50) UNIQUE,
  address VARCHAR(100) NULL,
  dob DATE,
  dept_id INT,
  FOREIGN KEY(dept_id) REFERENCES department(id)
  );
  DROP TABLE student;
  ```
  ```

INSERT DATA

- The INSERT INTO statement is used to insert new records in a table.
- With column name
  - ```
    INSERT INTO table_name (column1, column2, column3, ...)
    VALUES (value1, value2, value3, ...);
    ```
- Without column Name
  - ```
    INSERT INTO table_name
    VALUES (value1, value2, value3, ...);
    ```

INSERT DATA

```
INSERT INTO department(id, name,
location, hod, phone)
VALUES (NULL, 'CSE', '4th floor',
'ME','+098765432');
```

- Multiple values in single query

```
INSERT INTO department(id, name, location, hod,
phone)
VALUES (NULL, 'CSE', '4th floor','ME','+098765432'),
(NULL, 'Math', '2nd floor','MSA','+098235432');
```

Homework

(Create the following databases)

Database 1: department management system (dms)

- departments (id, name, location, hod, phone)
- courses(id, title, credit_hours, type)
- students (id, name, roll, phone, address, dob, dept_id)
- teachers(id, name, designation, salary, joining_date, department)

Database 2: ecommerce

- customers (id, name, phone, address)
- products(id, name, unit_price, quantity)
- order (id, customer_id, pro_id, order_date, total_price)

Resources

- https://www.w3schools.com/sql/default.asp

- https://www.youtube.com/watch?v=mhDJYm4cLzU&list=PLTydW-y9HsbQ2ztoaLBJTd4wwjc_oqWx4&index=1&ab_channel=TrainingwithLiveProject

Any Questions??