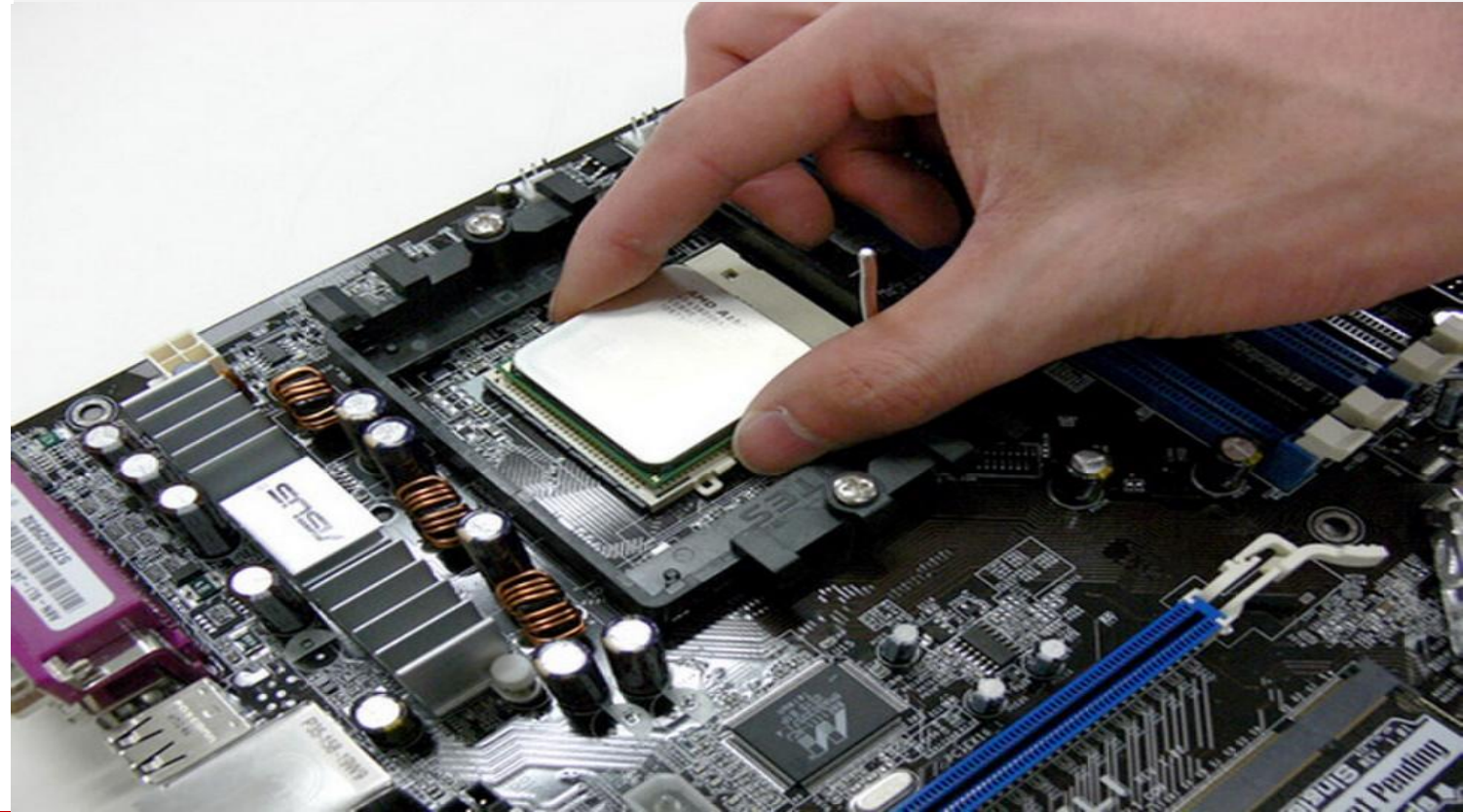




جامعة برج العرب التكنولوجية
BORG AL ARAB TECHNOLOGICAL UNIVERSITY

Database Programming

Lecture 5

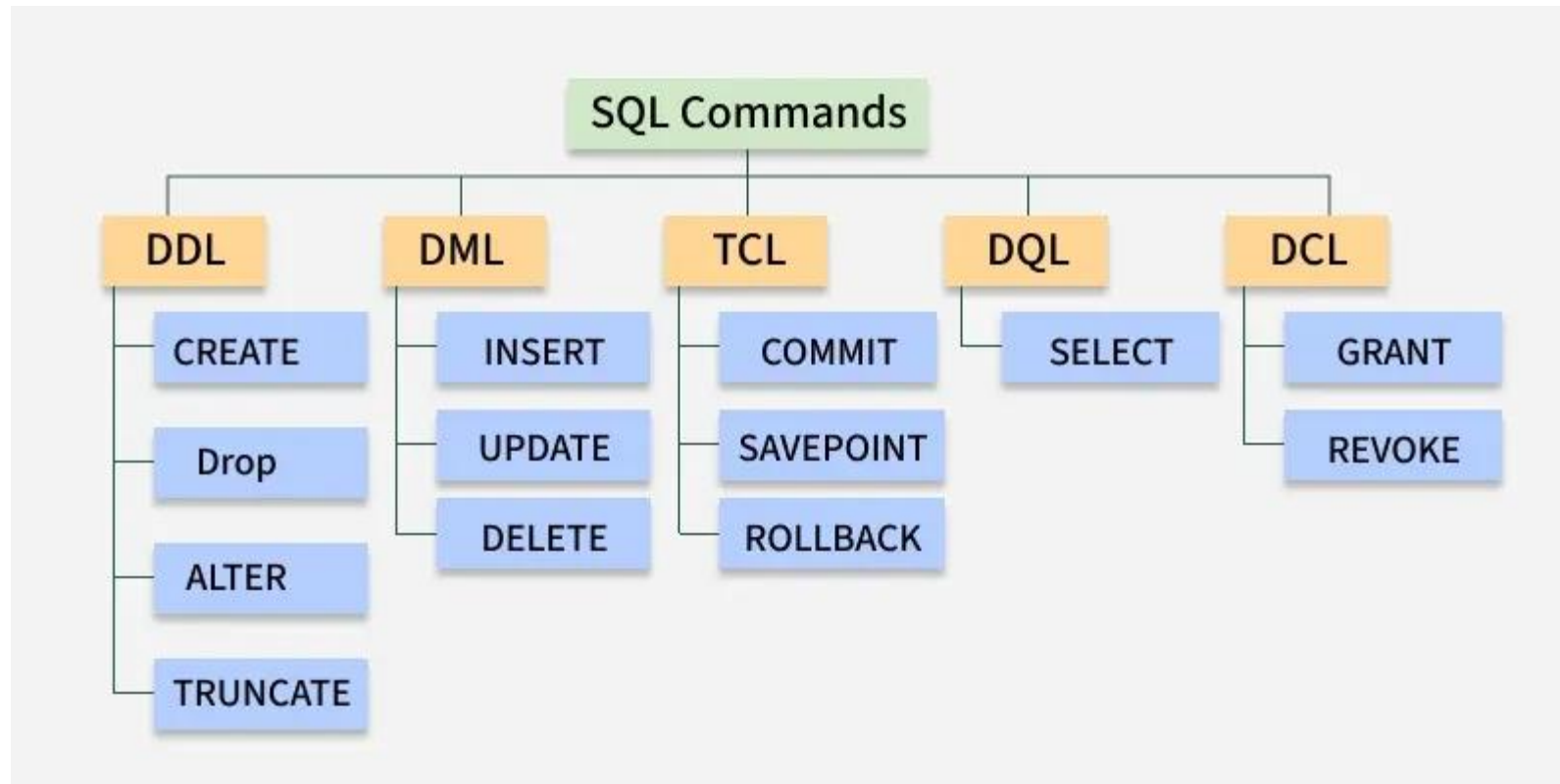


By. Mohamed Elshabka

SQL Commands

LECTURE 5

SQL Command Categories



SQL Command Categories

DDL (Data Definition Language)

Commands used to **create, modify, or delete database structure** (tables, schemas, etc.).

DML (Data Manipulation Language)

Commands used to **insert, update, or delete data inside tables**.

DQL (Data Query Language)

Commands used to **retrieve (query) data from the database**.

DCL (Data Control Language)

Commands used to **control user permissions and access to the database**.

TCL (Transaction Control Language)

Commands used to **manage database transactions and changes** (save, commit, or undo operations).

CREATE TABLE

Usage: Used to create new tables or database objects.

Structure:

```
CREATE TABLE table_name (  
    column_name datatype [constrain t],  
    column_name datatype [constrain t],  
    ...  
);
```

Example (Departments and Projects tables):

```
CREATE TABLE Departments (  
    DeptID INT PRIMARY KEY,  
    DeptName VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE Projects (  
    ProjectID INT PRIMARY KEY,  
    ProjectName VARCHAR(50)  
);
```

CREATE TABLE

Example (Employees table with FK reference to the Departments table and some constraints):

```
CREATE TABLE Employees (  
    EmpID INT PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL,  
    Email VARCHAR(100) UNIQUE,  
    Age INT CHECK (Age >= 18),  
    DepartmentID INT,  
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DeptID)  
);
```

DROP TABLE

Usage: Delete tables permanently including all data.

Structure:

```
DROP TABLE table_name;
```

Example:

```
DROP TABLE Projects;
```

ALTER TABLE

Usage: Used to modify the structure of an existing table: - Add/drop columns - Add/change constraints - Add primary keys or foreign keys

Structure:

```
ALTER TABLE table_name  
ADD|DROP|MODIFY column_name datatype|constraint;
```

Examples:

Add Column (DOB)

```
ALTER TABLE Employees  
ADD DOB DATE;
```

Adds Date of Birth column to Employees.

Add CHECK Constraint

```
ALTER TABLE Employees  
ADD CONSTRAINT email_check  
CHECK (Email LIKE '%@%');
```

Ensures emails contain @, improving data integrity.

Add Foreign Key (Project Assignment)

```
ALTER TABLE Employees  
ADD CurrentProjectID INT;  
ALTER TABLE Employees  
ADD FOREIGN KEY (CurrentProjectID) REFERENCES Projects(ProjectID);
```

TRUNCATE TABLE

Usage: Remove all rows but keep the table structure.

Structure:

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE Employees;
```

RENAME TABLE

Usage: Change table name without affecting data.

Structure:

```
RENAME TABLE old_table_name TO new_table_name;
```

Example:

```
RENAME TABLE Employees TO EmployeeRecords;
```

INSERT

Usage: Add new rows to a table.

Structure:

```
INSERT INTO table_name (column1, column2, ...)
```

```
VALUES (value1, value2, ...);
```

Example:

```
INSERT INTO Employees (EmpID, Name, Age, DepartmentID)  
VALUES (1, 'Alice', 25, 1);
```

UPDATE TABLE

Usage: Modify existing data in a table.

Structure:

```
UPDATE table_name  
SET column1 = value1, column2 = value2,  
WHERE condition;
```

Example:

```
UPDATE Employees  
SET Age = 26  
WHERE EmpID = 1;
```

DELETE TABLE

Usage: Remove rows from a table.

Structure:

```
DELETE FROM table_name WHERE condition;
```

Example:

```
DELETE FROM Employees WHERE E  
mpID = 1;
```

SELECT TABLE

Usage: Retrieve data from one or more tables. **Example:**

Structure:

SELECT column1, column2, ...

FROM table_name

WHERE condition;

SELECT Name, Email

FROM Employees

WHERE Age >= 18;

Thanks