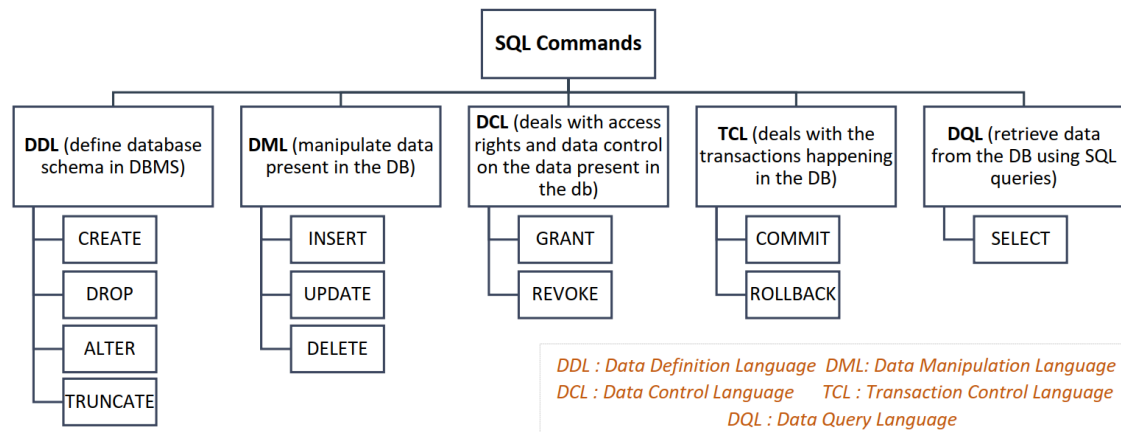


## Structured Query language (SQL)



### 1.DDL (Data Definition Language)

- Các lệnh DDL được sử dụng để **định nghĩa cấu trúc của cơ sở dữ liệu**, bao gồm việc tạo, sửa đổi và xóa các đối tượng trong cơ sở dữ liệu (bảng, chỉ mục, schema, v.v.).
- **CREATE**: Dùng để **tạo ra các đối tượng** trong cơ sở dữ liệu như bảng, chỉ mục, hoặc các schema.
- **DROP**: Dùng để **xóa các đối tượng đã được tạo ra**, ví dụ như bảng hoặc cơ sở dữ liệu.
- **ALTER**: Dùng để **thay đổi cấu trúc của một đối tượng đã tồn tại**, chẳng hạn như thêm cột mới vào bảng.
- **TRUNCATE**: Dùng để **xóa toàn bộ dữ liệu** trong bảng mà **không xóa chính bảng** đó

→ Các lệnh DDL định nghĩa cấu trúc của cơ sở dữ liệu và các đối tượng bên trong nó.

#### \* CREATE

Ví dụ: Tạo một bảng **Students** với các cột **StudentID**, **Name**, và **Age**.

```
sql Sao chép mã

CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Age INT
);
```

Lệnh này sẽ tạo ra bảng **Students** với các cột như đã nêu.

## \* DROP

Ví dụ: Xóa bảng **Students** khỏi cơ sở dữ liệu.

```
sql Sao chép mã  
  
DROP TABLE Students;
```

Lệnh này sẽ xóa hoàn toàn bảng **Students**, bao gồm cả dữ liệu bên trong nó.

## \* ALTER

Ví dụ: Thêm cột **Gender** vào bảng **Students**.

```
sql Sao chép mã  
  
ALTER TABLE Students  
ADD Gender VARCHAR(10);
```

Lệnh này sẽ thêm cột **Gender** vào bảng **Students**.

## \* TRUNCATE

Ví dụ: Xóa toàn bộ dữ liệu trong bảng **Students** nhưng không xóa bảng.

```
sql Sao chép mã  
  
TRUNCATE TABLE Students;
```

Lệnh này sẽ xóa tất cả các hàng trong bảng **Students** mà không ảnh hưởng đến cấu trúc của bảng.



	StudentID	Name	Age	Gender
	NULL	NULL	NULL	NULL

Xóa toàn bộ rows dù bảng có sẵn data.

## 2. DML (Data Manipulation Language)

Các lệnh DML được sử dụng để thao tác dữ liệu hiện có trong cơ sở dữ liệu như chèn, cập nhật và xóa dữ liệu.

- **INSERT**: Dùng để thêm dữ liệu vào bảng.
- **UPDATE**: Dùng để cập nhật dữ liệu đã có trong bảng.
- **DELETE**: Dùng để xóa dữ liệu từ bảng.

→ Các lệnh DML **thao tác trên dữ liệu hiện có** trong cơ sở dữ liệu.

## \* INSERT

Ví dụ: Thêm một sinh viên mới vào bảng **Students**.

```
sql Sao chép mã  
  
INSERT INTO Students (StudentID, Name, Age, Gender)  
VALUES (1, 'Nam', 20, 'Male');
```

Lệnh này sẽ **thêm một hàng mới** với thông tin sinh viên vào bảng **Students**.

## \* UPDATE

Ví dụ: Cập nhật tuổi của sinh viên **Nam** thành 21.

```
sql Sao chép mã  
  
UPDATE Students  
SET Age = 21  
WHERE Name = 'Nam';
```

Lệnh này sẽ cập nhật tuổi của sinh viên có tên là **Nam** thành 21.

## \* DELETE

Ví dụ: Xóa sinh viên có **StudentID** là 1 khỏi bảng **Students**.

```
sql Sao chép mã  
  
DELETE FROM Students  
WHERE StudentID = 1;
```

Lệnh này sẽ xóa sinh viên có **StudentID** là 1 khỏi bảng.

### 3. DCL (Data Control Language)

Các lệnh DCL được sử dụng để kiểm soát quyền truy cập và quyền điều khiển trên cơ sở dữ liệu.

- **GRANT**: Dùng để **cấp quyền truy cập và quyền hạn** cho người dùng.
- **REVOKE**: Dùng để **thu hồi quyền truy cập hoặc quyền hạn đã cấp** cho người dùng.

→ Các lệnh DCL quản lý quyền truy cập và quyền điều khiển trên dữ liệu trong cơ sở dữ liệu.

#### \* GRANT

Ví dụ: Cấp quyền SELECT cho người dùng **user1** trên bảng **Students**.

```
sql Sao chép mã  
  
GRANT SELECT ON Students TO user1;
```

Lệnh này sẽ cấp quyền truy vấn dữ liệu (SELECT) từ bảng **Students** cho người dùng **user1**.

#### \* REVOKE:

Ví dụ: Thu hồi quyền SELECT từ người dùng **user1** trên bảng **Students**.

```
sql Sao chép mã  
  
REVOKE SELECT ON Students FROM user1;
```

Lệnh này sẽ **thu hồi quyền truy vấn dữ liệu (SELECT)** từ người dùng **user1**.

### 4. TCL (Transaction Control Language)

Các lệnh TCL được sử dụng để quản lý các giao dịch trong cơ sở dữ liệu.

- **COMMIT**: Dùng để xác nhận các thay đổi đã thực hiện trong một giao dịch và lưu lại vĩnh viễn.
- **ROLLBACK**: Dùng để hoàn tác các thay đổi đã thực hiện trong một giao dịch nếu có lỗi xảy ra.

→ Các lệnh TCL xử lý các giao dịch và điều khiển việc ghi nhận hoặc hoàn tác các thay đổi trong cơ sở dữ liệu.

## \* COMMIT

Ví dụ: Sau khi **thêm sinh viên mới** vào bảng **Students**, bạn có thể sử dụng **COMMIT** để lưu lại thay đổi.

```
sql Sao chép mã  
  
INSERT INTO Students (StudentID, Name, Age, Gender)  
VALUES (2, 'Hoa', 22, 'Female');  
COMMIT;
```

Lệnh **COMMIT** này sẽ xác nhận và lưu các thay đổi vào cơ sở dữ liệu.

### Tác dụng của **COMMIT**:

- Xác nhận thay đổi: Khi bạn thực hiện một số thay đổi trong cơ sở dữ liệu (ví dụ như chèn một bản ghi mới bằng lệnh **INSERT**), các thay đổi này chưa được lưu lại ngay lập tức. Chúng chỉ tạm thời tồn tại trong phiên làm việc của bạn.
- Lưu lại thay đổi: Lệnh **COMMIT** được sử dụng để lưu lại các thay đổi này vào cơ sở dữ liệu một cách vĩnh viễn. Sau khi **COMMIT**, các thay đổi sẽ không thể bị hoàn tác thông qua lệnh **ROLLBACK**.
- Kết thúc giao dịch: **COMMIT** kết thúc giao dịch hiện tại, nghĩa là tất cả các lệnh SQL được thực hiện trước đó trong giao dịch sẽ được xác nhận và lưu trữ.

### Ví dụ minh họa:

- **Trước COMMIT:** Bạn chèn một bản ghi mới vào bảng **Students**, nhưng chưa thực hiện **COMMIT**. Nếu bạn kiểm tra bảng bằng lệnh **SELECT**, có thể bạn vẫn thấy bản ghi đó (tùy vào cách quản lý giao dịch của hệ thống cơ sở dữ liệu), nhưng bản ghi này chưa thực sự được lưu trữ vĩnh viễn.
- **Sau COMMIT:** Sau khi bạn thực hiện lệnh **COMMIT**, bản ghi mới sẽ được lưu trữ vĩnh viễn vào cơ sở dữ liệu. Ngay cả khi bạn kết thúc phiên làm việc hoặc hệ thống gặp sự cố, bản ghi này vẫn tồn tại.

## \* ROLLBACK:

Ví dụ: Nếu bạn thêm dữ liệu sai vào bảng **Students**, bạn có thể sử dụng **ROLLBACK** để hoàn tác.

```
sql Sao chép mã  
  
INSERT INTO Students (StudentID, Name, Age, Gender)  
VALUES (3, 'Linh', 19, 'Female');  
ROLLBACK;
```

Lệnh **ROLLBACK** sẽ hoàn tác lệnh **INSERT** trước đó và hủy bỏ việc thêm dữ liệu.

## 5. DQL (Data Query Language)

- **SELECT:** Dùng để truy vấn và lấy dữ liệu từ cơ sở dữ liệu.

→ Lệnh DQL được sử dụng để truy xuất dữ liệu từ cơ sở dữ liệu.

DQL chỉ có một lệnh chính là **SELECT**, dùng để truy vấn dữ liệu từ cơ sở dữ liệu.

## \* SELECT

Ví dụ: Truy vấn tất cả các sinh viên trong bảng **Students**.

```
sql Sao chép mã  
SELECT * FROM Students;
```

Lệnh này sẽ lấy tất cả các hàng từ bảng **Students**.

Một ví dụ khác, bạn chỉ muốn lấy tên và tuổi của các sinh viên:

```
sql Sao chép mã  
SELECT Name, Age FROM Students;
```

Lệnh này sẽ chỉ lấy các cột **Name** và **Age** từ bảng **Students**.