**CƠ SỞ DỮ LIỆU**

Cơ sở dữ liệu ( **Database** ) là nơi lưu trữ dữ liệu của 1 ứng dụng

Hệ quản trị cơ sở dữ liệu (CSDL) là nơi lưu trữ, thao tác với các DB

SQL ( **Structured Query Language** ) là ngôn ngữ truy vấn có cấu trúc được dùng để tạo, sửa, xoá và lấy dữ liệu từ 1 hệ quản trị CSDL

Các nhóm câu lệnh của SQL ( 4 nhóm )

### DML (***Data Manipulation Language***) : insert, update, delete, select

* **DDL** (***Data Definition Language***) : create, alter, drop
* **DCL** (***Data Control Language***): revoke, grant
* **TCL** (***Transaction Control Language***): commit, rollback, savepoint

**DDL** là ngôn ngữ định nghĩa dữ liệu. Các lệnh DDL bao gồm:

* **Lệnh CREATE**: Tạo một bảng, một View của bảng, hoặc đối tượng khác trong Database.
* **Lệnh ALTER**: Sửa đổi một đối tượng Database đang tồn tại, ví dụ như một bảng.
* **Lệnh DROP**: Xóa toàn bộ một bảng, một View của bảng hoặc đối tượng khác trong một Database.

**DML** là ngôn ngữ thao tác dữ liệu. Các lệnh DML bao gồm:

* **Lệnh SELECT**: Lấy các bản ghi cụ thể từ một hoặc nhiều bảng.
* **Lệnh INSERT**: Tạo một bản ghi.
* **Lệnh UPDATE**: Sửa đổi các bản ghi.
* **Lệnh DELETE**: Xóa các bản ghi.

**DCL** là ngôn ngữ điều khiển dữ liệu. Các lệnh DCL bao gồm:

* **Lệnh GRANT**: Trao một quyền tới người dùng.
* **Lệnh REVOKE**: Thu hồi quyền đã trao cho người dùng.

## **TCL** sử dụng các lệnh này có trong các giao dịch

* **COMMIT** : Chuyển các thay đổi xuống database.
* **ROLLBACK** : Quay lại trạng thái ban đầu trước khi bị *transaction* chỉnh sửa dữ liệu do xảy ra lỗi trong bên trong xử lý của *transaction.*
* **SAVEPOINT** : Lưu trạng thái dữ liệu ở một thời điểm bất kỳ của một transaction, chúng ta có thể roll-back về trạng thái này bất cứ khi nào

**ERD**

Sơ đồ quan hệ giữa các thực thể (**ERD**) Entity Relationship Diagram

- Vì yêu cầu của khách hàng (**SRS**) sofware Requirement Specification chỉ là các dòng chữ khô khan => cần phải thiết kế **ERD**

- Các bước thiết kế **ERD** :

+ Từ SRS xác định Entity ( thực thể )

+ Xác định các thuộc tính của Entity

+ Xác định mối quan hệ giữa các Entity ( 1-1 , 1-n , n-n )

+ Chuỗi : char, varchar, text, longtext

+ Ngày : date, datetime, time

- Constraint ( ràng buộc ) là quy tắc ( rule )

Mục đích : Hạn chế dữ liệu đưa vào bảng, đảm bảo dữ liệu chính xác, tin cậy và toàn vẹn

- 1 số rằng buộc :

+ Not null : không cho phép null ( không bỏ trống )

+ UNIQUE : duy nhất không trùng lập

+ Primary key : khoá chính ( not null + unique )

+ Forgein key : khoá ngoại

+ Check

+ Defaut

+ Index

- Các bước chuyển đổi ERD sang Table

**B1** : Biến đổi các Entity thành các table

+ Thuộc tính khoá là thuộc tính chính của table

+ Các thuộc tính thường là thuộc tính của table

**B2** : Biến đổi thuộc tính đa trị

+ Tách thuộc tính đa trị thành 1 bảng mới, xác định mối quan hệ giữa bảng hiện tại và bảng mới

**B3** : Biến đổi mối quan hệ 1 – 1

+ Đặt thuộc tính khoá ở phía tuỳ chọn sang bắt buộc

**B4** : Biến đổi mối quan hệ 1 – n

+ Đặt khoá ngoại ( FK ) ở bên nhiều ( n ) , khoá chính PK ( 1 )

**B5** : Biến đổi mối quan hệ n – n

+ Tách thêm 1 bảng mới table 3 : có khoá chính là tập hợp 2 khoá của 2 thực thể tham gia vào mối quan hệ => gọi là khoá phức hợp ( composite key )

**CHUẨN HOÁ CƠ SỞ DỮ LIỆU**

**Mục đích** : tránh dư thừa dữ liệu, dễ cập nhật dữ liệu

1/ 1 NF

2/ 2 NF

3/ 3 NF

*1 số câu truy vấn đầy đủ* :

SELECT , FROM , WHERE , GROUP BY , HAVING , ORDER BY , LIMIT

Join dùng để lấy thông tin dữ liệu trên 2 hoặc nhiều bảng.

**Các loại JOIN** :

- Inter join : lấy toàn bộ phần chung của 2 bảng

- Onter join :

+ Left join : chung + bên trái

+ Right join : chung + bên phải

+ Full join : lấy toàn bộ phần chung

- Where : là mệnh đề điều khiển => dùng để truy xuất dữ liệu theo 1 điều kiện nào đó

1 số toán tử sau where ( < , > , < > , = , or , and , between , like , in , not )

- Group by, having : nhóm dữ liệu trên 1 cột sử dụng để thống kê hay đi cùng với các hàm count() , sun() , min() , max() , avg()

- Order by : sử dụng để sắp xếp theo 1 cột hoặc nhiều cột

asc : tăng dần

desc : giảm dần

**Hàm gộp** ( aggregate function )

- 1 số loại hàm gộp sử dụng với group by

+ count ( )

+ min ( )

+ max ( )

+ avg ( )

+ sum ( )

Sử dụng **UNION** & **UNION ALL**

- Union dùng để nối kết quả 2 hay nhiều cây truy vấn

- ĐK : + 2 câu truy vấn ( query ) phải cùng số lượng cột

+ Cùng kiểu dữ liệu của từng cột

ð Lưu ý : ở MSQL mới thì không cần phải bắt buộc các cột cùng dữ liệu

+ Union : dùng để gộp các record trùng nhau

+ Union all : không gộp các record trùng nhau

Ý ngĩa của **LIMIT**

- Limit giới hạn số lượng các record được hiển thị ( lấy ra )

- Limit A,B : A vị trí cần lấy , B là số lượng cần lấy

- Limit A : A là số lượng cần lấy và vị trí cần cần lấy là 0

**Truy vấn con** ( sub query ) là 1 câu truy vấn lồng vào 1 câu truy vấn khác => hạn chế sử sụng truy vấn con ( làm giảm tốc độ )

**INDEX , VIEW , SP , FUNCTION , TRIGGER**

**INDEX** ( chỉ mục ) là bảng tra cứu đặc biệt có thể sử dụng để tăng hiệu quả tìm kiếm dữ liệu

- Cú pháp tạo :

C1 : CREATE INDEX index\_name ON table\_name (column\_name)

C2 : AITER TABLE table\_name ADD INDEX index\_name (column\_name)

- Cú pháp xoá :

C1 : DROP INDEX index\_name ON table\_name

C2 : AITER TABLE table\_name DROP INDEX index\_name

Cần lưu ý :

- Không tạo index cho các bản nhỏ, các bản thường xuyên update, insert, có nhiều giá trị null

- Chỉ những cột hay tìm kiếm thì mới tạo index

**VIEW** ( bảng ảo )

- Là các truy vấn select được lưu lại với dạng bảng và sau đó có thể truy vấn dữ liệu từ view như tới table

- Mục đích : tái sử dụng code, bảo mật thông tin cho bảng chính

**SP** tương tự như hàm void trong Java là tập hợp các câu lệnh query để thực hiện 1 nghiệp vụ nào đó.

* Cú pháp :

Delimiter $$

CREATE PROCEDURE sp\_name (param)

BEGIN

// code sql

END $$

Delimiter ;

=> Param là tham số ( 3 kiểu )

+ IN : truyền tham trị

+ ON : truyền tham chiếu

+ INOUT : kết hợp IN và OUT

**FUNCTION**

Tập hợp các câu lệnh query thực hiện 1 nghiệp vụ nào đó nhưng có trả về giá trị => Cho phép tạo ra các function cho riêng mình

=> Param trong function mặc định là **IN**