### Có mấy cách để viết câu lệnh SELECT? Cách nào tốt hơn? Tại sao?

\*\*Có hai cách chính để viết câu lệnh **SELECT**:

**- Cách 1: SELECT toàn bộ cột**: Sử dụng SELECT \* để chọn tất cả các cột của bảng.

+ SELECT \* FROM TableName;

- Cách 2: **SELECT cụ thể các cột**: Chỉ chọn các cột cụ thể mà mình cần.

+ SELECT column1, column2 FROM TableName;

- **Cách tốt hơn**: Viết cụ thể các cột bạn cần lấy ra. Vì nó:

* Giúp **tăng hiệu suất**: Chỉ lấy dữ liệu cần thiết, giảm tải dữ liệu trả về.
* **Tối ưu hóa truy vấn**: Giảm dung lượng dữ liệu cần xử lý và truyền tải, đặc biệt quan trọng khi làm việc với bảng lớn.
* **Dễ bảo trì**: Bạn biết rõ những gì mình cần lấy ra, tránh sự phụ thuộc vào thay đổi cấu trúc bảng trong tương lai.

### Làm sao để SELECT danh sách các giá trị?

- Để **SELECT** một danh sách các giá trị từ bảng, ta sử dụng cú pháp:

+ SELECT column\_name FROM table\_name WHERE column\_name IN (value1, value2, value3);

### Sự khác nhau giữa 3 toán tử AND, OR, NOT?

**Sự khác nhau**:

* **AND** yêu cầu **tất cả** điều kiện đều đúng.
* **OR** chỉ yêu cầu **một trong số** các điều kiện đúng.
* **NOT** phủ định điều kiện, nghĩa là nếu điều kiện đúng thì nó sẽ trả kết quả sai và ngược lại.

### Sự khác nhau giữa 2 toán tử IN và LIKE?

- **IN**: Kiểm tra giá trị có thuộc trong danh sách giá trị cụ thể hay không.

Vd: SELECT \* FROM employees WHERE department IN ('PO', 'Sales', 'PM', 'DEV' );

- **LIKE**: Tìm kiếm giá trị khớp với một mẫu cụ thể (có thể sử dụng với ký tự đại diện).

Vd: SELECT \* FROM employees WHERE name LIKE 'T%'; -- Tên bắt đầu bằng chữ T

- **Sự khác nhau**:

* **IN**: Dùng khi bạn biết **cụ thể** các giá trị cần kiểm tra.
* **LIKE**: Dùng khi bạn muốn tìm kiếm theo **mẫu**, thường với chuỗi ký tự.

### 5. Công dụng của câu lệnh JOIN là gì?

* **JOIN** được sử dụng để **kết hợp** các hàng từ hai hay nhiều bảng dựa trên một điều kiện liên kết ( cột liên quan) giữa chúng (thường là qua khóa chính và khóa ngoại).

### 6. Có mấy loại câu lệnh JOIN? Sự khác nhau giữa các loại JOIN?

-  JOIN gồm 3 loại cơ bản INNER JOIN, LEFT JOIN, SELF JOIN.

### -**Sự khác nhau giữa INNER JOIN, LEFT JOIN và SELF JOIN**

**+ INNER JOIN**:

* 1. Trả về các hàng chỉ khi có giá trị tương ứng trong cả hai bảng.
  2. **Sử dụng khi** bạn chỉ cần các kết quả có quan hệ hợp lệ ở cả hai bảng.

**+ LEFT JOIN:**

* 1. Trả về tất cả các hàng từ bảng bên trái, kể cả khi không có giá trị tương ứng từ bảng bên phải (giá trị không khớp sẽ là NULL).
  2. **Sử dụng khi** bạn muốn lấy tất cả các giá trị từ bảng bên trái, bất kể có khớp với bảng bên phải hay không.

**+SELF JOIN**:

* 1. Liên kết một bảng với chính nó, thường được sử dụng khi bạn cần so sánh dữ liệu từ cùng một bảng.
  2. **Sử dụng khi** bạn cần so sánh hoặc lấy dữ liệu giữa các hàng trong cùng một bảng, ví dụ như danh sách nhân viên và quản lý.

### 7. Công dụng của câu lệnh ORDER BY là gì?

- Mệnh đề ORDER BY trong SQL được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc giảm dần, dựa trên một hoặc nhiều cột. ASC được sử dụng để sắp xếp tăng dần và DESC được sử dụng để sắp xếp giảm dần.

### 8. Công dụng của câu lệnh GROUP BY là gì?

Mệnh đề GROUP BY trong SQL được sử dụng hợp tác với câu lệnh SELECT để sắp xếp dữ liệu giống nhau thành các nhóm. Mệnh đề GROUP BY này tuân theo mệnh đề WHERE trong câu lệnh SELECT và đứng trước mệnh đề ORDER BY.

### 9. Công dụng của câu lệnh HAVING là gì?

* **HAVING** dùng để **lọc các nhóm** sau khi đã sử dụng GROUP BY, thường dùng với các hàm tổng hợp.

### 10. Công dụng của câu lệnh DISTINCT là gì?

* **DISTINCT** dùng để loại bỏ các giá trị trùng lặp trong kết quả trả về.