

BÀI THỰC HÀNH SỐ 2

I. Mục tiêu.

- Hiện thực được ngôn ngữ thao tác dữ liệu và truy vấn đơn giản SQL trên hệ quản trị SQL Server
- Các lệnh thao tác dữ liệu: thêm, xóa, sửa
- Cú pháp lệnh truy vấn (select ... from... where)
- Giới thiệu Select ... into ..., view

II. Hướng dẫn chung

Câu truy vấn tổng quát

```
SELECT [DISTINCT] danh_sách_cột  
FROM danh sách các quan hệ (hay bảng, table)  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột_gom_nhóm]  
[HAVING điều_kiện_trên_nhóm]  
[ORDER BY cột1 ASC | DESC, cột2 ASC | DESC,... ]
```

Dạng truy vấn đơn giản

```
SELECT <Danh sách các cột> FROM <Danh sách Bảng>  
SELECT * FROM <Tên Bảng>
```

Mệnh đề where: để xác định điều kiện lấy dữ liệu

```
SELECT <Danh sách các cột> FROM <Tên bảng> WHERE ...
```

Ví dụ:

```
SELECT MaSV, HoDem, Ten FROM SINHVIEN WHERE GioiTinh = "Nữ";  
  
SELECT MaSV, HoDem, Ten  
FROM SINHVIEN  
WHERE GioiTinh = "Nam" and MaLop = "CSDL-IT004";
```

1. LIKE

Mệnh đề LIKE trong SQL được sử dụng để so sánh một giá trị với các giá trị tương tự sử dụng toán tử ký tự đại diện (wildcard). Có hai ký tự đại diện được sử dụng kết hợp với toán tử LIKE.

- Phần trăm (%)
- Dấu gạch dưới (_)

Dấu phần trăm thể hiện không, một hoặc nhiều ký tự. Dấu gạch dưới đại diện cho một số hoặc một ký tự. Những ký hiệu này có thể được sử dụng trong sự kết hợp.

```
SELECT FROM table_name
WHERE column LIKE 'XXXX%' or SELECT FROM table_name
WHERE column LIKE '%XXXX%' or SELECT FROM table_name
WHERE column LIKE 'XXXX_' or SELECT FROM table_name
WHERE column LIKE '_XXXX' or SELECT FROM table_name
WHERE column LIKE '_XXXX_'
```

Ví dụ:

WHERE SALARY LIKE '20%' --Tìm trong cột lương bất kỳ giá trị nào bắt đầu bằng 20.

WHERE SALARY LIKE '%2' --Tìm trong cột lương bất kỳ giá trị nào kết thúc bằng 2.

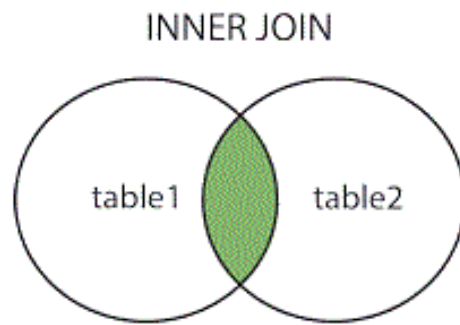
2. JOIN

Mệnh đề **JOIN** trong SQL được sử dụng để kết hợp các bản ghi từ hai hay nhiều bảng trong cơ sở dữ liệu. JOIN là một phương tiện để kết hợp các trường từ hai bảng bằng cách sử dụng các giá trị chung cho mỗi bảng.

```
SELECT A.column1, A.column2, B.column1, B.column3
FROM table1 A, table2 B
WHERE A.column1 = B.column2;
```

a. Inner Join

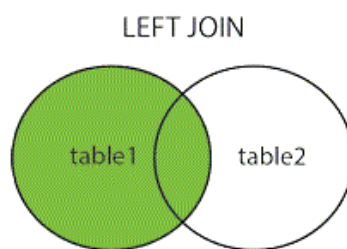
Lệnh **Inner Join** trong SQL tạo một bảng kết quả mới bằng cách kết hợp các giá trị cột của hai bảng (table1 và table2) dựa trên điều kiện nối. Truy vấn so sánh từng hàng của table1 với từng hàng của table2 để tìm tất cả các cặp hàng thỏa mãn điều kiện nối. Khi điều kiện nối được thỏa mãn, các giá trị cột cho mỗi cặp hàng A và B phù hợp sẽ được kết hợp thành một hàng kết quả.



```
SELECT table1.column1, table2.column2...
FROM table1
INNER JOIN table2
ON table1.common_field = table2.common_field;
```

b. Left Join

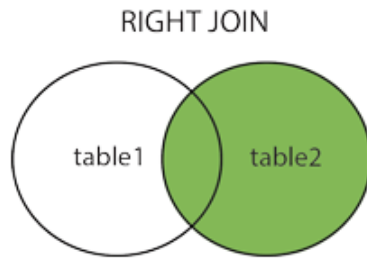
Lệnh **Left Join** trong SQL trả về tất cả các hàng từ bảng bên trái, ngay cả khi không có kết quả khớp nào trong bảng bên phải. Điều này có nghĩa là nếu mệnh đề ON không khớp với các bản ghi của bảng bên phải. Lệnh Left Join vẫn sẽ trả về một hàng trong kết quả, nhưng với NULL cho mỗi cột của bảng bên phải.



```
SELECT table1.column1, table2.column2...
FROM table1
LEFT JOIN table2
ON table1.common_field = table2.common_field;
```

c. Right Join

Lệnh **Right Join** trong SQL trả về tất cả các hàng từ bảng bên phải, ngay cả khi không có kết quả khớp nào trong bảng bên trái. Điều này có nghĩa là nếu mệnh đề ON không khớp với các bản ghi của bảng bên trái. Lệnh Right Join vẫn sẽ trả về một hàng trong kết quả, nhưng với NULL cho mỗi cột của bảng bên trái.

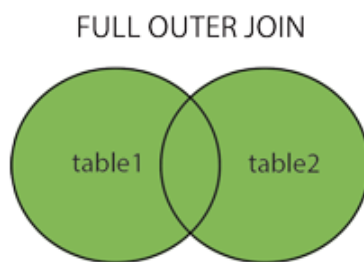


```
SELECT table1.column1, table2.column2...
FROM table1
RIGHT JOIN table2
ON table1.common_field = table2.common_field;
```

d. Full Join

Lệnh **Full Join** trong SQL là sự kết hợp các kết quả của cả hai phép nối Left Join và Right Join.

Kết quả join sẽ chứa tất cả các bản ghi từ cả hai bảng và điền vào NULL cho các kết quả bị thiếu ở hai bên.



```
SELECT table1.column1, table2.column2...
FROM table1
FULL JOIN table2
ON table1.common_field = table2.common_field;
```

e. Self Join

Lệnh **Self Join** trong SQL được sử dụng để nối một bảng với chính nó.

```
SELECT a.column_name, b.column_name...
FROM table1 a, table1 b
WHERE a.common_field = b.common_field;
```

III. Bài tập thực hành

A. Hoàn thành bài Lab 1 (nếu chưa làm xong)

B. Xem file BT 2 “VIII.QUẢN LÝ CỬA HÀNG HOA “

- Tạo các quan hệ và khai báo các khóa chính, khóa ngoại của quan hệ.
- Tạo các dữ liệu mẫu cho bài (tối thiểu 5 bộ cho mỗi bảng)
- **Với mỗi đề, thực hiện câu 1 (a&b) và câu 2 (a,b,c,d)**
- So sánh cách viết Đại số quan hệ (trong lớp lý thuyết) và cách viết SQL của 6 câu (tổng 2 đề là 12 câu) nói trên. Sinh viên tạo file Word, có bảng so sánh.

File nộp:

- MSSV_BTTH2_QuanLyCuaHangHoa.sql
- MSSV_BTTH2_QuanLyCuaHangHoa.pdf (hoặc doc/docx)

C. Làm các bài tập sau:

Câu 1: Bài QuanLyBanHang (19 câu):

I.2 --> I.10 (9 câu)

II.2 --> II.4 (3 câu)

III.1 --> III.7 (7 câu)

File nộp: MSSV_BTTH2_QuanLyBanHang.sql

Câu 2: Bài QuanLyGiaoVu (13 câu)

I.3 → I.8 (chưa làm I.9 & I.10), I.11 → I.14 (10 câu)

III.1 → III.3 (3 câu)

File nộp: MSSV_BTTH2_QuanLyGiaoVu.sql

Bài tập làm theo cá nhân

Không nộp bài hoặc các bài làm giống nhau sẽ bị 0 điểm.

Ở đầu các file nộp có ghi chú Họ tên và MSSV.

Ví dụ cho ghi chú đầu file: -- 2412456 -- Lê Thị ABCDEFGHIJK