



CƠ SỞ DỮ LIỆU

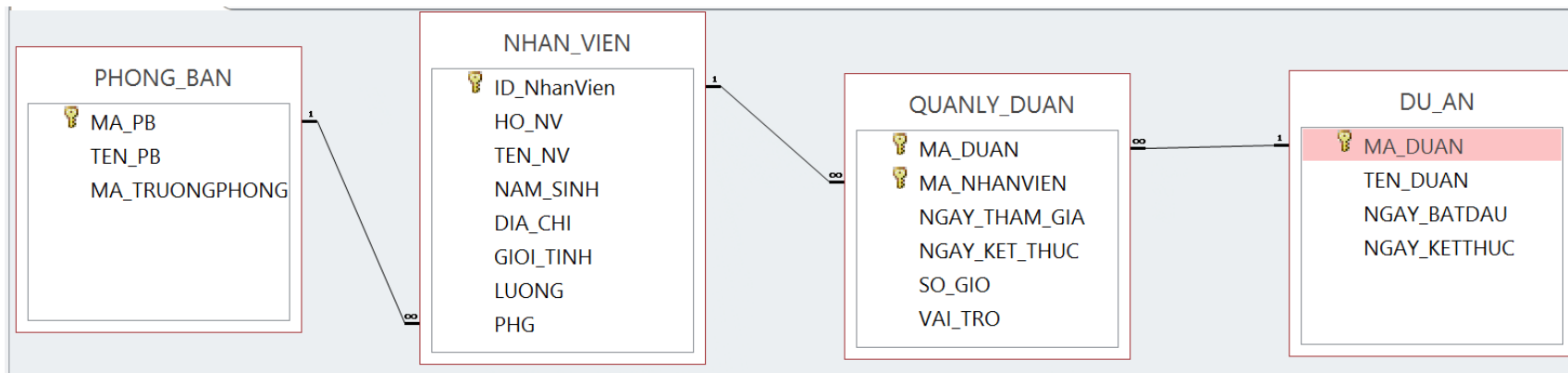
BÀI 6: NGÔN NGỮ TRUY VẤN SQL- TRUY VẤN DỮ LIỆU TRÊN NHIỀU BẢNG PHẦN 1

- Tìm hiểu câu lệnh truy vấn Select trên nhiều bảng với mệnh đề JOIN
- Phân loại **INNER JOIN, LEFT/RIGHT OUTER JOIN**
- Câu truy vấn lồng nhau - **SUB QUERY**



- Các cách truy vấn dữ liệu trên nhiều bảng
 - Sử dụng phép tích
 - Sử dụng mệnh đề JOIN
 - Câu truy vấn con

- Sử dụng cơ sở dữ liệu quản lý nhân viên (đã giới thiệu ở bài 2)



- Sử dụng điều kiện kết bằng trong mệnh đề **WHERE**
- Nếu xuất hiện tên cột trùng nhau trong nhiều bảng thì bắt buộc phải sử dụng tên bảng hoặc bí danh bảng trước tên cột
- Cú pháp:

```
SELECT [table.]column1, [table.]column2, ...  
FROM table1, table2  
WHERE table1.column = table2.column ;
```

- Thực hiện truy vấn để hiển thị dữ liệu của 2 bảng nhan_vien, và phong_ban gồm: id_nhanvien, ten_nv, phg, ma_pb, ten_pb

```
SELECT nhan_vien.id_nhanvien, nhan_vien.ten_nv, nhan_vien.phg,  
phong_ban.ma_pb, phong_ban.ten_pb  
FROM phong_ban, nhan_vien  
WHERE phong_ban.ma_pb = nhan_vien.phg ;
```

- Hoặc được viết tắt gọn khi không có các cột trùng tên

```
SELECT id_nhanvien, ten_nv, phg, ma_pb, ten_pb  
FROM phong_ban, nhan_vien  
WHERE phong_ban.ma_pb = nhan_vien.phg ;
```

- Thực hiện truy vấn để hiển thị dữ liệu của 3 bảng nhan_vien, du_an và quanly_duan gồm: ma_duan, ten_duan, ho_nv, ten_nv, ngay_tham_gia, ngay_ket_thuc

```
SELECT du_an.ma_duan, ten_duan, ho_nv, ten_nv, ngay_tham_gia,  
ngay_ket_thuc
```

```
FROM nhan_vien, du_an , quanly_duan
```

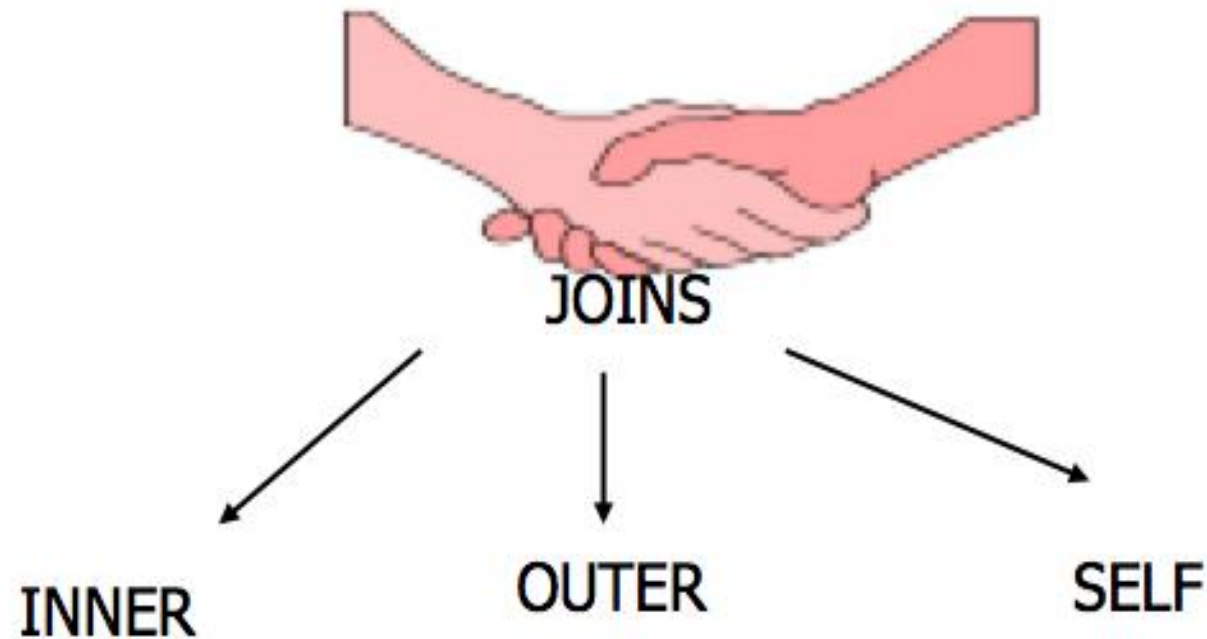
```
WHERE (du_an.ma_duan = quanly_duan.ma_duan) AND  
(nhan_vien.id_nhanvien = quanly_duan.ma_nhanvien) ;
```

- Đơn giản hóa các câu truy vấn khi cần sử dụng tên bảng cho việc truy xuất các cột

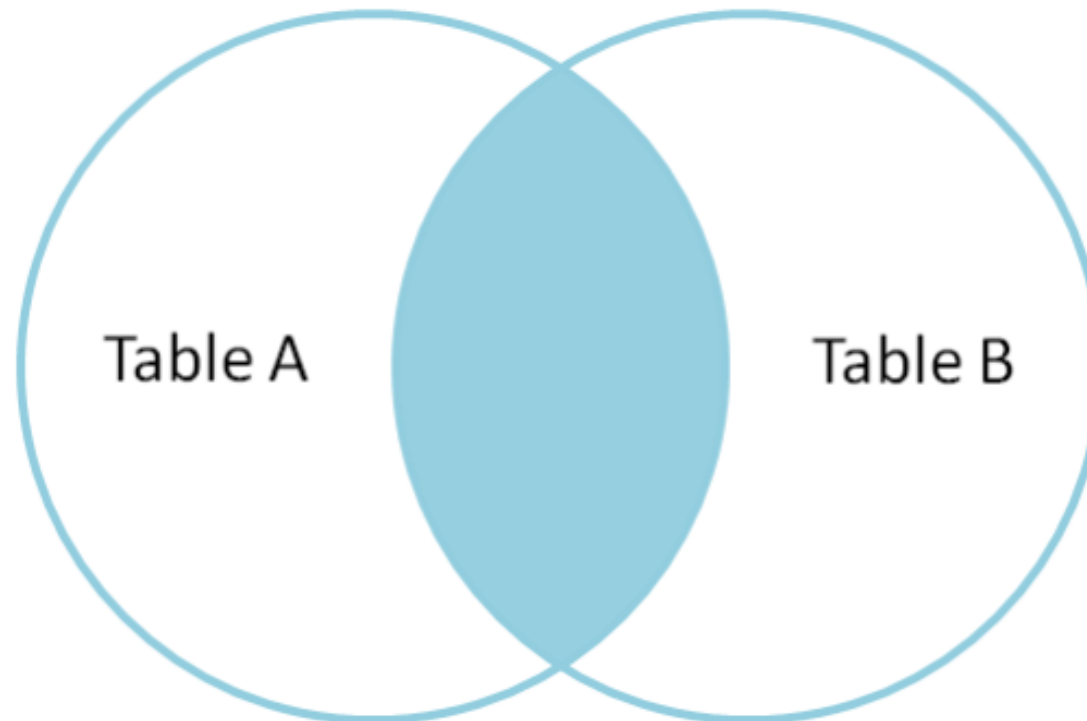
```
SELECT B.ma_duan, B.ten_duan, A.ho_nv, A.ten_nv,  
C.ngay_tham_gia, C.ngay_ket_thuc  
FROM nhan_vien A, du_an B, quanly_duan C  
WHERE (B.ma_duan = C.ma_duan) AND (A.id_nhanvien =  
C.ma_nhanvien) ;
```

Lưu ý: Khi đã đặt bí danh thì phải dùng qua bí danh để đại diện cho tên bảng chứ không thể dùng tên bảng.

- JOIN là phép kết nối dữ liệu từ nhiều bảng lại với nhau
- Có 3 loại JOIN



- **INNER JOIN** trả về kết quả là các bản ghi mà trường được join ở hai bảng khớp nhau, các bản ghi chỉ xuất hiện ở một trong hai bảng sẽ bị loại.



- Cú pháp:

```
SELECT column1, column2,...
```

```
FROM table1
```

```
    [INNER] JOIN table2 ON <điều kiện join 1>
```

```
    [ [INNER] JOIN table3 ON <điều kiện join 2> ]
```

```
... ;
```

- Sử dụng câu lệnh JOIN để hiển thị dữ liệu của 2 bảng Nhân viên và Phòng ban

```
SELECT ID_NhanVien, ho_nv, ten_nv, ten_pb FROM phong_ban PB  
INNER JOIN nhan_vien NV ON PB.ma_pb = NV.phg ;
```

- Sử dụng mệnh đề JOIN để hiển thị dữ liệu trong 3 bảng: NHAN_VIEN, DU_AN và QUANLY_DUAN

```
SELECT DA.ma_duan, DA.ten_duan, NV.ho_nv, NV.ten_nv,  
       QL.ngay_tham_gia, QL.ngay_ket_thuc  
FROM ((nhan_vien NV INNER JOIN quanly_duan QL  
      ON NV.id_nhanvien = QL.ma_nhanvien)  
      INNER JOIN du_an DA ON DA.ma_duan = QL.ma_duan);
```

- Phép tích sẽ nhân số lượng bản ghi 2 bảng, sau đó loại bỏ các bản ghi không thỏa mãn điều kiện
 - Ví dụ: Bảng PB có 3 bản ghi, bảng NV có 4 bản ghi -> tích sẽ cho ra 12 bản ghi, sau đó sẽ loại bỏ các bản ghi không thỏa mãn điều kiện: **PB.MA_PB = NV.PHG**
- Với **INNER JOIN** thì trong quá trình thực hiện tích 2 bảng nó sẽ kiểm tra điều kiện ở **ON** luôn, nếu đúng thì chọn, sai thì bỏ qua.
- Như vậy xét về tốc độ truy vấn thì trường hợp sử dụng **INNER JOIN** sẽ nhanh hơn rất nhiều so với sử dụng phép tích

- Sử dụng JOIN hoặc phép tích để hiển thị thông tin của 4 bảng gồm: họ và tên nhân viên, lương, tên phòng ban mà nhân viên thuộc về, tên dự án, ngày bắt đầu tham gia vào dự án và số giờ đã làm

```
SELECT Concat(NV.ho_nv, ' ', NV.ten_nv) AS 'Họ và tên', NV.luong,  
PB.ten_pb, DA.ten_duan, QL.ngay_tham_gia, QL.so_gio  
FROM (((phong_ban PB INNER JOIN nhan_vien NV  
ON PB.Ma_pb = NV.pgh) INNER JOIN quanly_duan QL ON  
NV.id_nhanvien = QL.ma_nhanvien) INNER JOIN du_an DA ON  
DA.ma_duan = QL.ma_duan);
```

- Viết câu truy vấn hiển thị các thông tin bao gồm họ, tên, lương của nhân viên, tên dự án với điều kiện nhân viên thuộc phòng ban có tên 'Thiết kế', tham gia vào các dự án có ngày bắt đầu '1/1/2016'

```
SELECT NV.ho_nv, NV.ten_nv, NV.luong, DA.ten_duan, FROM
(((phong_ban PB INNER JOIN nhan_vien NV
ON PB.Ma_pb = NV.pgh) INNER JOIN quanly_duan QL ON
NV.id_nhanvien = QL.ma_nhanvien) INNER JOIN du_an DA ON
DA.ma_duan = QL.ma_duan)
WHERE PB.Ten_pb = 'Thiết kế' AND DA.ngay_batdau = '2016-01-01' ;
```

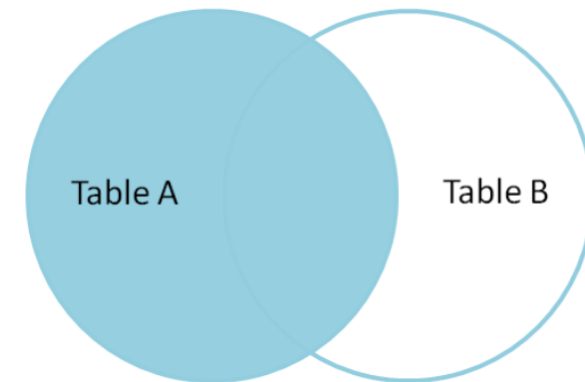
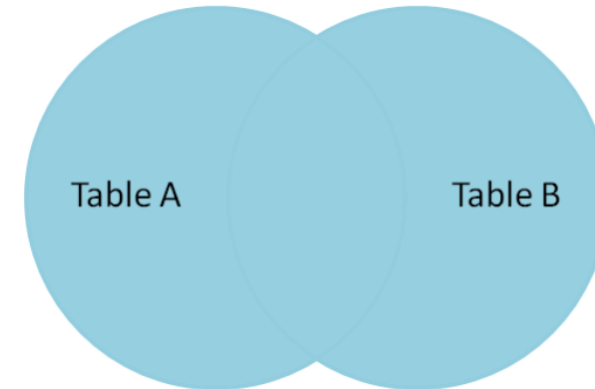



CƠ SỞ DỮ LIỆU

BÀI 6: NGÔN NGỮ TRUY VẤN SQL- TRUY VẤN DỮ LIỆU TRÊN NHIỀU BẢNG PHẦN 2

■ Có 2 loại:

- **FULL OUTER JOIN**: kết quả gồm tất cả các bản ghi của cả hai bảng. Với các bản ghi chỉ xuất hiện trong một bảng thì các cột dữ liệu từ bảng kia được điền giá trị NULL
- **HALF OUTER JOIN** (**LEFT** hoặc **RIGHT**): nếu bảng A LEFT OUTER JOIN với bảng B thì kết quả gồm các bản ghi có trong bảng A, với các bản ghi không có mặt trong bảng B thì các cột từ B được điền NULL. Các bản ghi chỉ có trong B mà không có trong A sẽ không được trả về



- FULL OUTER JOIN

```
SELECT column1, column2,...  
FROM table1 FULL [OUTER] JOIN table2  
ON <điều kiện join>;
```

- HALF OUTER JOIN

```
SELECT column1, column2,...  
FROM table1 LEFT|RIGHT [OUTER] JOIN table2  
ON <điều kiện join>;
```

■ Left outer join

```
SELECT NV.ID_nhanvien, NV.ho_nv, NV.ten_nv, PB.ten_pb
FROM phong_ban PB LEFT JOIN nhan_vien NV
ON PB.ma_pb = NV.phg;
```

ID_NhanVien	HO_NV	TEN_NV	TEN_PB
1	Nguyen	An	San xuất 1
3	Tran	Cuong	San xuất 1
9	Tran	Thuy	San xuất 1
2	Le	Bao	San xuất 2
4	Hoang	Lan	San xuất 2
5	Van	Toan	Quan Ly Chat Luong
6	Nguyen	Thi	Quan Ly Chat Luong
7	Le	Loi	Thiet ke
10	Hoang	Chinh	Thiet ke
8	Hoang	Nga	Nghien cuu cong nghe
			Kinh Doanh

■ Right outer join

```
SELECT NV.ID_nhanvien, NV.ho_nv, NV.ten_nv, PB.ten_pb
FROM nhan_vien NV RIGHT JOIN phong_ban PB
ON PB.ma_pb = NV.phg;
```

ID_NhanVien	HO_NV	TEN_NV	TEN_PB
1	Nguyen	An	San xuất 1
3	Tran	Cuong	San xuất 1
9	Tran	Thuy	San xuất 1
2	Le	Bao	San xuất 2
4	Hoang	Lan	San xuất 2
5	Van	Toan	Quan Ly Chat Luong
6	Nguyen	Thi	Quan Ly Chat Luong
7	Le	Loi	Thiet ke
10	Hoang	Chinh	Thiet ke
8	Hoang	Nga	Nghien cuu cong nghe
			Kinh Doanh

- Viết câu truy vấn hiển thị thông tin bao gồm: Mã, họ, tên nhân viên, và tên phòng ban mà nhân viên trực thuộc. Nếu nhân viên chưa được phân bổ vào phòng nào thì cột tên phòng để trống

```
SELECT NV.ID_NhanVien, NV.ho_nv, NV.ten_nv, PB.ten_pb  
FROM phong_ban PB RIGHT JOIN nhan_vien NV  
ON PB.ma_pb = NV.phg;
```

ID_NhanVien	HO_NV	TEN_NV	TEN_PB
11	Le	Hoang	
1	Nguyen	An	San xuất 1
3	Tran	Cuong	San xuất 1
9	Tran	Thuy	San xuất 1
2	Le	Bao	San xuất 2
4	Hoang	Lan	San xuất 2
5	Van	Toan	Quan Ly Chat Luong

- Một bảng kết nối với chính nó

- Ví dụ:

MA_NV	TEN_NV	MA_QUANLY
1	Nga	
2	An	1
3	Van	2
4	Hoang	2

- Hiển thị tên nhân viên và tên người Quản Lý của anh ấy:

```
SELECT NV.ma_nv, NV.ten_nv, QL.ten_nv AS 'Nguoi quan ly'  
FROM nhan_vien NV INNER JOIN nhan_vien QL  
ON QL.ma_nv = NV.ma_quanly;
```

- Là câu truy vấn SELECT nằm lồng bên trong một câu truy vấn khác
- Câu truy vấn con có thể được sử dụng:
 - Trong mệnh đề WHERE như một điều kiện tìm kiếm
 - Trong mệnh đề HAVING như một điều kiện tìm kiếm
 - Trong mệnh đề FROM như một đặc tả bảng
 - Trong mệnh đề SELECT như một đặc tả cột
- Nội dung câu truy vấn con được đặt trong cặp dấu ngoặc ()

- Sử dụng câu truy vấn con để hiển thị thông tin các nhân viên có lương lớn hơn mức lương trung bình toàn công ty

```
SELECT * FROM nhan_vien  
WHERE luong >  
    (SELECT AVG(luong) FROM nhan_vien);
```

SO SÁNH JOIN VÀ CÂU TRUY VẤN CON

JOIN	CÂU TRUY VẤN CON
Kết quả có thể bao gồm các cột của cả 2 bảng	Không thể bao gồm các cột của câu truy vấn con
Sử dụng mối quan hệ giữa 2 bảng	
Chạy nhanh hơn	
	Có thể chuyển 1 giá trị tính toán ra câu truy vấn bên ngoài
	Dễ viết code và dễ hiểu

Điều kiện	Kết quả tương đương	Mô tả
$X > \text{ANY}(1,2)$	$X > 1$	X phải lớn hơn ít nhất 1 giá trị trả về từ câu truy vấn con
$X < \text{ANY}(1,2)$	$X < 2$	X phải nhỏ hơn ít nhất 1 giá trị trả về từ câu truy vấn con
$X = \text{ANY}(1,2)$	$(X = 1) \text{ OR } (X = 2)$	X IN (1,2)
$X \geq \text{ALL} (1,2)$	$X \geq 2$	X lớn hơn hoặc bằng giá trị lớn nhất trong tập giá trị trả về từ câu truy vấn con
$X \leq \text{ALL} (1,2)$	$X \leq 1$	X nhỏ hơn hoặc bằng giá trị nhỏ nhất trong tập giá trị trả về từ câu truy vấn con

- Ví dụ: Hiển thị danh sách nhân viên có lương lớn hơn lương thấp nhất của phòng 'PB002'

```
SELECT * FROM nhan_vien
```

```
WHERE lương > ANY
```

```
(SELECT lương FROM nhan_vien WHERE phg = 'PB002');
```

Hoặc

```
SELECT * FROM nhan_vien
```

```
WHERE lương >
```

```
(SELECT Min(lương) FROM nhan_vien WHERE phg = 'PB002');
```

- Ví dụ: Hiển thị danh sách nhân viên có lương lớn hơn lương của tất cả các nhân viên phòng 'PB002'

```
SELECT * FROM nhan_vien
```

```
WHERE lương > ALL
```

```
(SELECT lương FROM nhan_vien WHERE phg = 'PB002');
```

Hoặc

```
SELECT * FROM nhan_vien
```

```
WHERE lương >
```

```
(SELECT Max(lương) FROM nhan_vien WHERE phg = 'PB002');
```

- Cú pháp:

WHERE biểu thức [NOT] IN (câu truy vấn con) ;

WHERE [NOT] EXISTS (câu truy vấn con) ;

- Ví dụ: Hiển thị danh sách các phòng ban chưa có nhân viên.

```
SELECT * FROM phong_ban
```

```
WHERE ma_pb NOT IN (SELECT phg FROM nhan_vien);
```

```
SELECT * FROM phong_ban A
```

```
WHERE NOT EXISTS (SELECT * FROM nhan_vien B  
                  WHERE A.ma_pb = B.phg);
```

- Ví dụ: Hiển thị danh sách các phòng ban không có nhân viên tham gia dự án.

```
SELECT * FROM phong_ban
WHERE ma_pb NOT IN (SELECT phg FROM nhan_vien INNER JOIN
quanly_duan ON
    nhan_vien.id_nhanvien = quanly_duan.ma_nv);
```

```
SELECT * FROM phong_ban A
WHERE NOT EXISTS (SELECT *
    FROM quanly_duan C INNER JOIN nhan_vien B
        ON B.ma_duan = C.ma_duan    WHERE A.ma_pb =
    B.phg);
```

- Viết câu truy vấn để hiển thị thông tin gồm mã nhân viên, họ tên, lương của nhân viên đã tham gia nhiều hơn 5 dự án
- Viết câu truy vấn để hiển thị tổng số giờ đã làm trong các dự án của mỗi nhân viên

- ❑ Có thể truy vấn dữ liệu trên nhiều bảng bằng các cách sau:
 - ❖ Thực hiện phép tích 2 bảng
 - ❖ Sử dụng mệnh đề JOIN
 - ❖ Câu truy vấn lồng nhau
- ❑ Phép tích sử dụng điều kiện kết bằng trong mệnh đề WHERE
- ❑ Mệnh JOIN có 3 loại
 - ❖ INNER JOIN
 - ❖ SELF JOIN
 - ❖ OUTER JOIN





Cảm ơn