

# Join

- Ý nghĩa: Là câu lệnh dùng để kết hợp dữ liệu ở các bảng dựa trên một cột liên quan giữa chúng (Primary key – Foreign key).
- Các loại Join: Inner Join, Right Join, Left Join, Full outer join

# Inner Join

- Ý nghĩa: Dùng để trả về các bản ghi mà trường được join giữa 2 bảng có cùng giá trị.
- Cú pháp:  

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```
- NOTE: Có thể thay từ khóa 'Inner Join' bằng 'Join', kết quả trả về như nhau

## Ví dụ: Inner Join - LOL

```
SELECT f.name, cl.name, cl.create_at, cl.day_buy  
      FROM figure f  
      JOIN costume_line_figure cf ON f.id= cf.figure_id  
      JOIN costume_line cl ON cl.id = cf.costume_line_id;
```

```
ALTER TABLE skin  
ADD type_id int,  
ADD CONSTRAINT skin_type_id_fk  
FOREIGN key(type_id) REFERENCES type(id);
```

# BTVN

Dựa vào bài thiết kế database LOL, hãy thực hiện:

- + Tìm kiếm tất cả các skin limited
- + Hiển thị thông tin về các vị tướng và các dòng trang phục của các vị tướng ấy
- + Hiển thị tất cả thông tin liên quan đến skin limited
- + Tìm kiếm tất cả các trang phục loại huyền thoại có giá từ 290 đến 560

# Left Join

- Ý nghĩa: Trả về tất cả records(dữ liệu) ở bảng bên trái và dữ liệu ở bảng bên phải tương ứng và ***chấp nhận giá trị null*** khi không có dữ liệu tương ứng phù hợp.
- Cú pháp:  
SELECT column\_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column\_name = table2.column\_name;

# Right Join

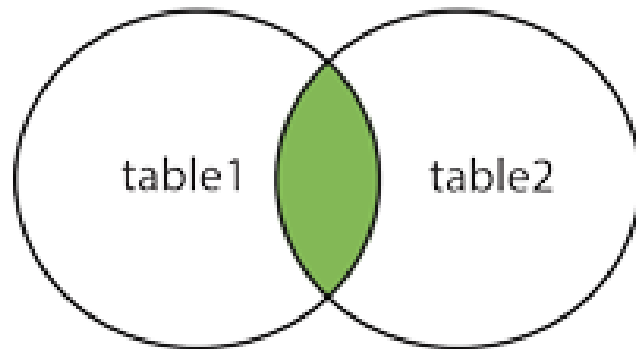
- Ý nghĩa: Trả về tất cả records(dữ liệu) ở bảng bên phải và dữ liệu ở bảng bên trái tương ứng và ***chấp nhận giá trị null*** khi k có dữ liệu tương ứng phù hợp
- Cú pháp:  
SELECT column\_name(s)  
FROM table1  
Right JOIN table2  
ON table1.column\_name = table2.column\_name;

# Full Join

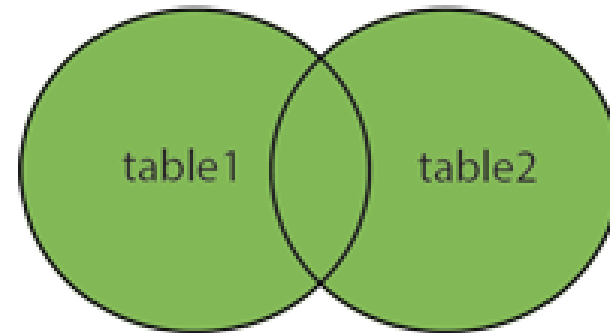
MySQL không hỗ trợ Full Join

# Summary

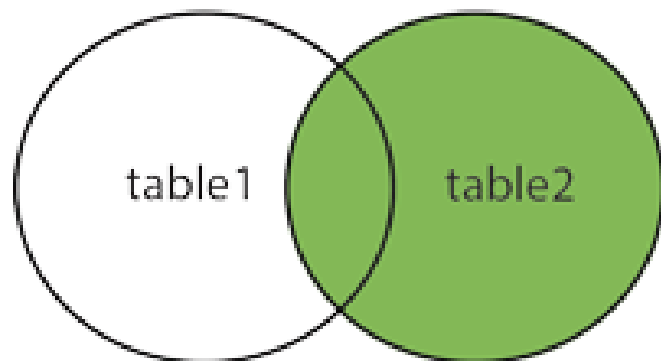
INNER JOIN



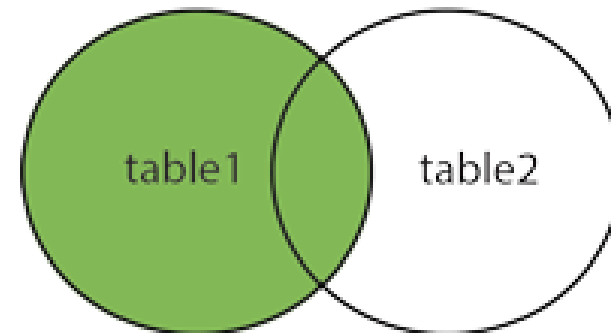
FULL OUTER JOIN



RIGHT JOIN



LEFT JOIN





# Bài tập

- Thầy Cường đang muốn xây dựng lại hệ thống trang web của techmaster và có đưa ra một vài yêu cầu nhỏ để thiết kế DB:
- Tên DB: TechmasterSystem
- Các bảng dữ liệu:

blog(id, title, description, create\_at)

user(id, name, phone, email)

role(id, name)

course(id, name, time\_line, description, price, type\_course, vote)

openingSchedule(id, start\_date, address, study\_day, study\_time)

transactionHistory(id, registration\_date, status)

accountBanking(id, name\_bank, account\_num)

## Lưu ý:

- Mỗi user chỉ có thể có 1 role nhưng mỗi role có thể là của nhiều user
- Mỗi user chỉ có thể dung tối đa 1 account banking và mỗi account chỉ thuộc về 1 user
- Mỗi user có thể tham gia nhiều khóa học và mỗi khóa học cũng sẽ có nhiều user tham gia
- Các khóa học có nhiều lịch khai giảng (openSchedule) và ngược lại
- Lịch sử giao dịch dùng để lưu thông tin đăng ký mua “*những*” khóa học của “*các*” học viên (mỗi học viên có thể thực hiện nhiều giao dịch)

# Yêu cầu:

- Thiết kế khóa chính và khóa ngoại cho các bảng dựa trên mối quan hệ giữa chúng.
- Insert ít nhất 5 records vào mỗi bảng.
- Hiển thị Hồ sơ cá nhân của các user (id, name, phone, email, role, thông tin ngân hàng)
- Tìm kiếm tất cả các học viên đăng ký học online
- Hiển thị lịch khai giảng và thông tin các khóa học trong buổi khai giảng đó
- Hiển thị chi tiết lịch sử gd của các học viên: mã giao dịch (id), tên học viên (student name), tên khóa học (course name), ngày đăng ký (registration\_date), trạng thái (status)