

# MySQL Advanced

The logo for CYBERLEARN is positioned behind the word 'MySQL'. It features a stylized icon of three stacked cubes to the left of the text 'CYBERLEARN'. Below the main text, there is a line of smaller text in Vietnamese: 'ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH'.

# Thao tác với nhiều Table

Customers				
customer_id	first_name	last_name	email	birthday
1	Hào	Nguyễn	nphao@gmail.com	1998-05-11
2	Thảo	Trương	ttthao@gmail.com	1999-12-10

Orders			
order_id	order_date	price	amount
1	2020-11-24	1000	70
2	2020-10-07	2000	90

# Tại sao phải cần liên kết các bảng



# Tại sao phải cần liên kết các bảng

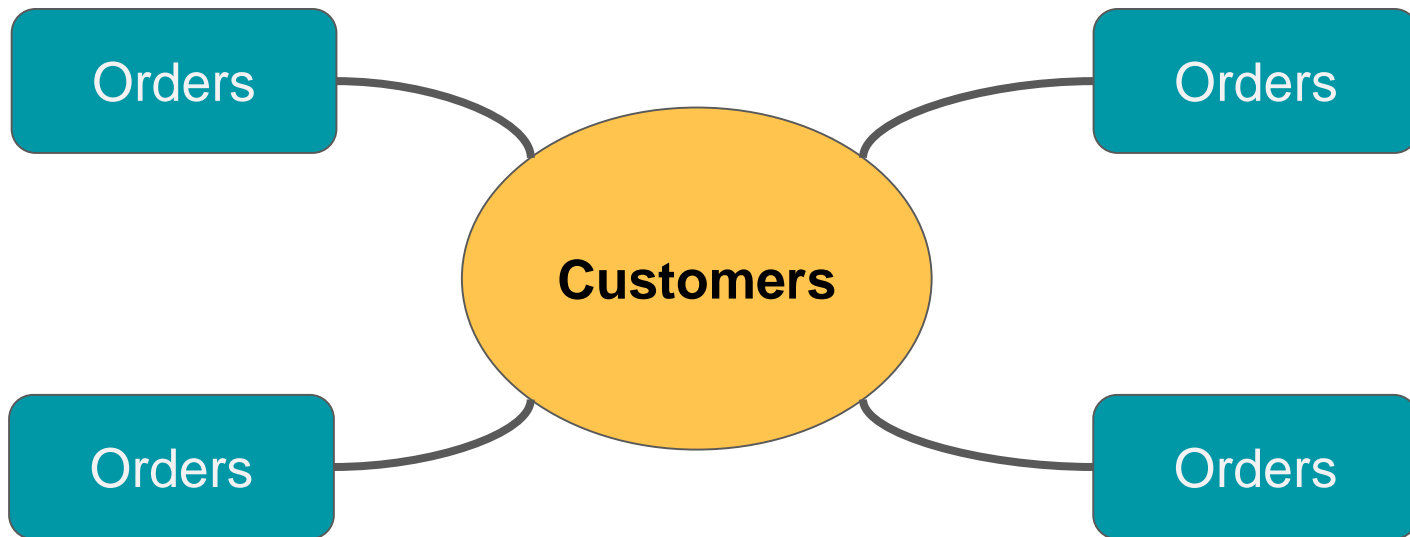
Vấn đề

- 01 làm sao lấy ra khách hàng đã đặt hóa đơn có mã là 1 ?
- 02 làm sao lấy ra các hóa đơn đặt hàng của một khách hàng
- 03 làm sao để lấy ra khách hàng mua vào ngày 2020-11-24 ?
- 04 làm sao để lấy ra các người dùng mua hàng hơn 100\$



Các câu truy vấn trên đều đề cập đến ít nhất là 2 đối tượng vì thế chúng ta phải liên kết các đối tượng này lại.

# Phân tích mối quan hệ của 2 bảng

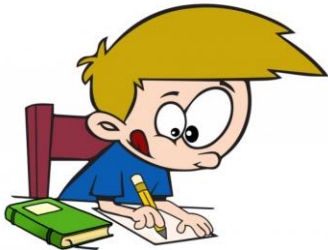


Một khách hàng có nhiều hóa đơn đặt hàng đúng không ta ?

# Mối quan hệ một - nhiều



Một <==> Nhiều



tóm lại : một khách hàng có nhiều đơn đặt hàng

# Liên kết 2 bảng theo mỗi quan hệ 1 - N

Customers				
customer_id	first_name	last_name	email	birthday
1	Hào	Nguyễn	nphao@gmail.com	1998-05-11
2	Thảo	Trương	ttthao@gmail.com	1999-12-10

Orders				
order_id	order_date	price	amount	customer_id
1	2020-11-24	1000	70	1
2	2020-10-07	2000	90	1

# Code Tạo quan hệ Một - Nhiều

```
• CREATE TABLE customers(  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    email VARCHAR(150) NOT NULL,  
    birthday DATE NOT NULL  
);  
  
• CREATE TABLE orders(  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_date DATE NOT NULL,  
    price FLOAT NOT NULL,  
    amount FLOAT NOT NULL,  
    customer_id INT,  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

- tạo thêm column **customer\_id** ở table **orders** ( tạo khóa phụ ở bảng nhiều )
- ràng buộc khóa phụ cho **customer\_id** vừa tạo ra , tham chiếu tới khóa chính của table **customers** là **customer\_id**.



# Thêm dữ liệu vào bảng

```
INSERT INTO customers(first_name,last_name,email,birthday) VALUES  
("Hào" , "Nguyễn" , "nphao@gmail.com" , "1998-05-11"),  
("Thảo" , "Trương" , "ttthao@gmail.com" , "1998-12-06");
```

```
INSERT INTO orders(order_date,price,amount,customer_id) VALUES  
("2020-01-01" , 2000 , 90 , 1),  
("2020-06-15" , 2000 , 90 , 2),  
("2020-06-15" , 2000 , 90 , 2),  
("2020-01-01" , 2000 , 90 , 1),  
("2020-06-15" , 2000 , 90 , 2);
```

# Truy vấn dữ liệu theo yêu cầu

01 Tìm đơn Đặt Hàng do khách hàng Hào đặt ?

02 Tổng số lượng sản phẩm đã mua của các khách hàng ?

03 ...



**Bắt tay vào làm thôi nào !**

# Cross Join ( Cách 1 )

- Tìm đơn Đặt Hàng do khách hàng Hào đặt ?

```
SELECT customer_id FROM customers WHERE first_name='Hào';  
SELECT * FROM orders WHERE customer_id = 1;
```



- ưu điểm : rõ ràng , dễ học.
- nhược điểm : phải thao tác 2 bước và phải có sự nhúng tay của con người



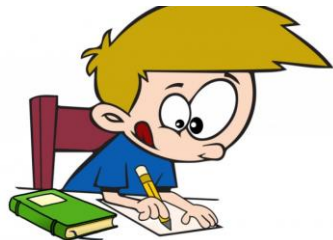
# Cross Join ( Cách 2 )

- Tìm đơn Đặt Hàng do khách hàng Hào đặt ?



- SOFTONE -

```
• SELECT * FROM orders
  WHERE customer_id = (
    SELECT customer_id FROM customers
    WHERE first_name='Hào'
  );
```



- ưu điểm : rõ ràng , dễ học.
- nhược điểm : phải thao tác 2 bước select

# Inner Join ( Cách 1 )

- Tìm đơn Đặt Hàng do khách hàng Hào đặt ?

```
-- c1
```

```
SELECT * FROM customers, orders
```

```
WHERE customers.customer_id = orders.customer_id && first_name='Hào';
```



- ưu điểm : rõ ràng , dễ học.
- nhược điểm : phải truy vấn tới 2 table , và điều kiện where rất khó mở rộng.

# Inner Join ( Cách 2 )

- Tìm đơn Đặt Hàng do khách hàng Hào đặt ?
- **SELECT \* FROM** customers  
inner join orders  
ON customers.customer\_id = orders.customer\_id  
where first\_name='Hào';



- Giải quyết được vấn đề truy vấn 2 bảng
- Các dự án thực tế thường dùng cách này

# Left Join

- Tổng số lượng sản phẩm đã mua của các khách hàng?

**SELECT**

first\_name,

last\_name,

IFNULL(SUM(amount), 0) AS total\_spent

**FROM** customers

**LEFT JOIN** orders

ON customers.customer\_id = orders.customer\_id

**GROUP BY** customers.customer\_id

**ORDER BY** total\_spent;



# Right Join

- Tổng số lượng sản phẩm đã mua của các khách hàng?

**SELECT**

```
IFNULL(first_name, 'MISSING') AS first,  
IFNULL(last_name, 'USER') as last,  
order_date,  
amount,  
SUM(amount)
```

**FROM** customers

**RIGHT JOIN** orders

```
ON customers.customer_id = orders.customer_id
```

**GROUP BY** first\_name, last\_name;

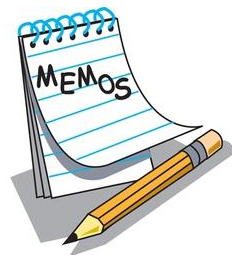




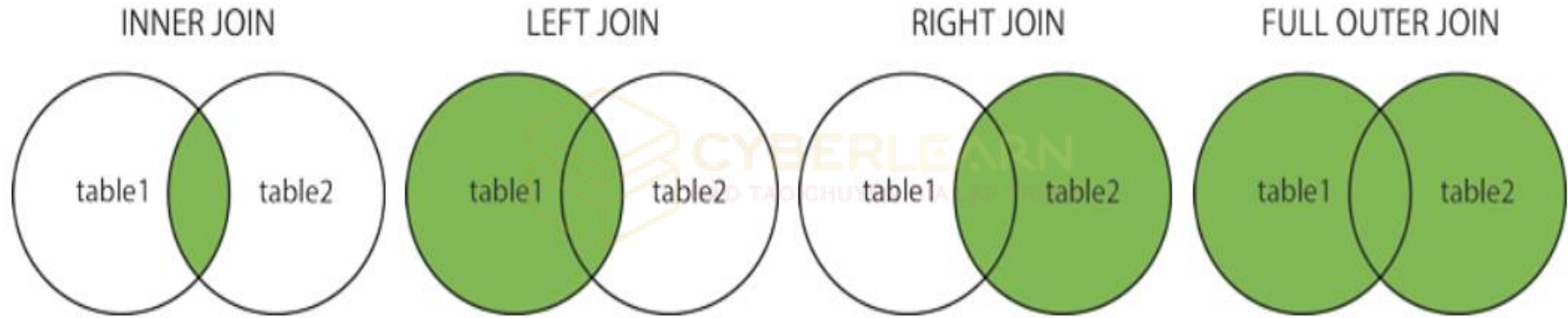
# Full Join

- sự kết hợp của left join và right join :

```
-- full join
select * from customers
left join orders
on orders.customer_id = customers.id
union
select * from customers
right join orders
on orders.customer_id = customers.id;
```

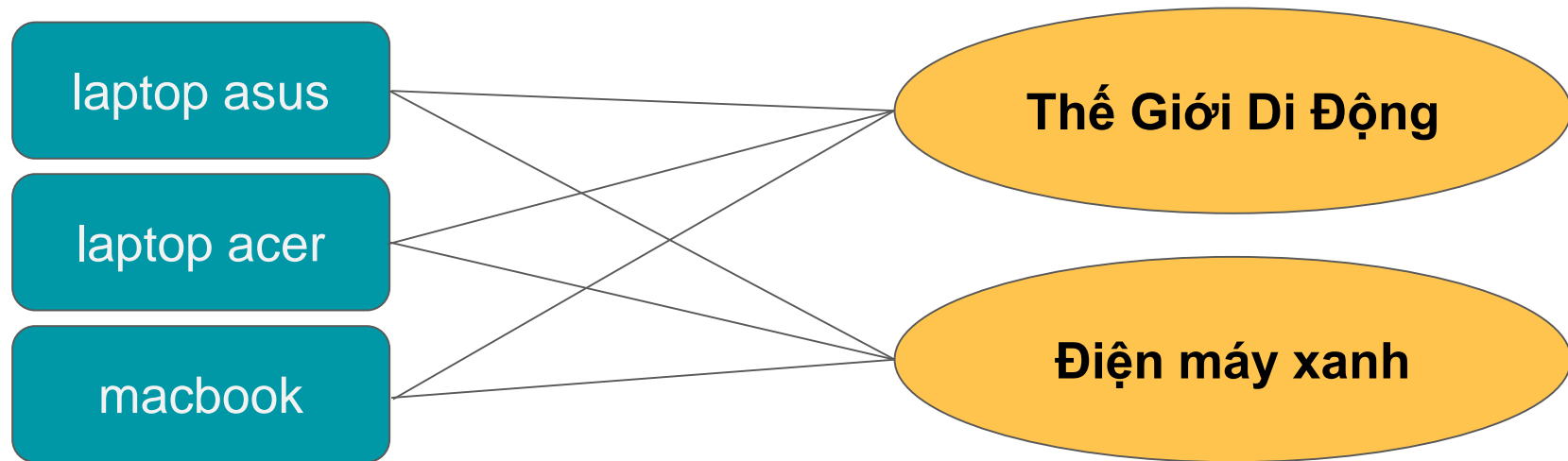


# Các kiểu JOIN



À thì ra là **JOIN** , **LEFT JOIN** và **RIGHT JOIN** dùng truy xuất liên kết các bảng với nhau .

# Phân tích mối quan hệ của 2 bảng

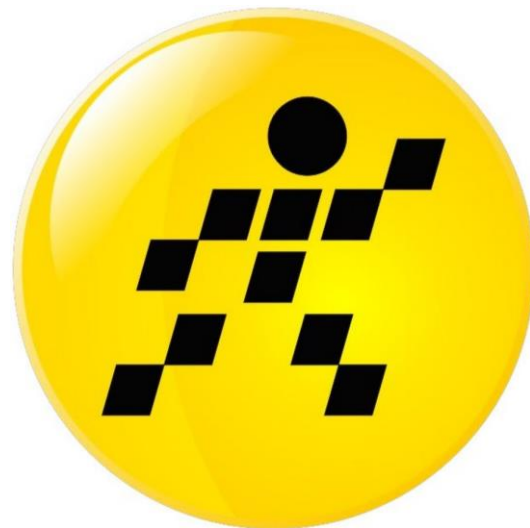


Một chiếc laptop được bán ở nhiều cửa hàng , còn một cửa hàng có thể bán nhiều laptop

# Mối quan hệ Nhiều - nhiều



Nhiều  $\Longleftrightarrow$  Nhiều



**tóm lại : một laptop và cửa hàng có mối quan hệ  
nhiều nhiều**

# Liên kết 2 bảng theo mỗi quan hệ N - N

laptops			
id	name	description	price
1	Macbook M1	Máy siêu xịn	25000
2	Razer Blade 15	Máy chơi game cực sướng	3000

laptops_stores		
id	laptop_id	store_id
1	1	2
2	2	2
3	2	1

stores		
id	name	address
1	Thế Giới Di Động	11 điện biên phủ quận 1 thành phố Hồ Chí Minh
2	Điện Máy Xanh	20 xô viết nghệ tĩnh quận bình thạnh thành phố Hồ Chí Minh

# Code tạo table laptops

```
-- tạo table laptops
create table laptops(
  id int primary key auto_increment,
  name varchar(255) not null,
  description varchar(255) not null,
  price float not null
);
-- chèn data vào table laptops
insert into laptops(name , description , price)
values ("Macbook M1" , "Máy siêu xịn" , 2500),
("Razer blade 15" , "Máy chơi game cực sướng" , 3000);
```

# Code tạo table stores

```
-- tạo table stores
create table stores(
  id int primary key auto_increment,
  name varchar(255) not null,
  address varchar(255) not null
);
-- chèn data vào table stores
insert into stores(name , address)
values ("Thế Giới Di Động" , "11 điện biên phủ quận 1 thành phố Hồ Chí Minh"),
("Điện Máy Xanh" , "20 xô viết nghệ tĩnh quận bình thạnh thành phố Hồ Chí Minh");
```

# Code tạo table laptops\_stores

```
-- tạo table laptops_stores
create table laptops_stores(
  id int primary key auto_increment,
  laptop_id int,
  store_id int,
  foreign key (laptop_id) references laptops(id),
  foreign key (store_id) references stores(id)
);
-- chèn data vào table laptops_stores
insert into laptops_stores(laptop_id , store_id)
values (1,2),(2,2),(2,1);
```



# Truy vấn dữ liệu theo yêu cầu

01 Tìm xem điện máy xanh đang bán những mẫu laptop nào và có giá bao nhiêu?

02 Tìm xem macbook đang được bán tại những cửa hàng nào?

03 Tìm xem các laptops chưa được bán tại bất kì cửa hàng nào?



**Bắt tay vào làm thôi nào !**

# Inner Join

- Tìm xem điện máy xanh đang bán những mẫu laptop nào và có giá bao nhiêu?

```
-- Tìm xem điện máy xanh đang bán những mẫu laptop nào?  
select laptops.name , price from stores  
join laptops_stores on stores.id = laptops_stores.store_id  
join laptops on laptops.id = laptops_stores.laptop_id  
where stores.name = "Điện Máy Xanh";
```



- join 2 bảng vào để có đầy đủ data
- where để xác định điều kiện

# Inner Join

- Tìm xem macbook đang được bán tại những cửa hàng nào?

-- Tìm xem macbook đang được bán tại những cửa hàng nào?

```
select * from stores
join laptops_stores on stores.id = laptops_stores.store_id
join laptops on laptops.id = laptops_stores.laptop_id
where laptops.name = "Macbook M1";
```



- join 2 bảng vào để có đầy đủ data
- where để xác định điều kiện



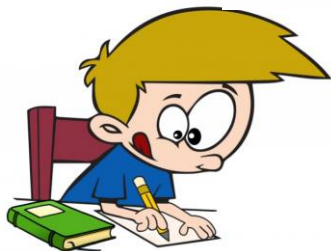
# Left Join

- Tìm xem các laptop chưa được bán tại những cửa hàng?

-- tìm có bao nhiêu lap

```
select * from laptops
left join laptops_stores
on laptops.id = laptops_stores.laptop_id
left join stores
on stores.id = laptops_stores.store_id
where stores.id is null;
```

RN  
TRÌNH



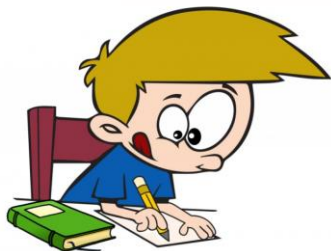
- join 2 bảng vào để có đầy đủ data
- where để xác định điều kiện

# Right Join

- Tìm xem các Macbook M1 có được bán tại thế giới di động hay không ?

-- tìm xem macbook có được bán tại thế giới di động không ?

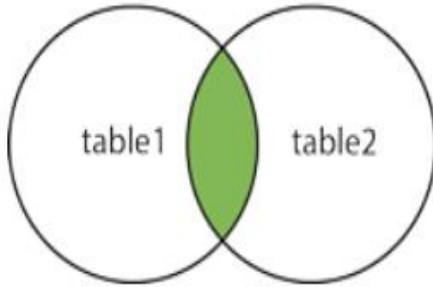
```
select * from laptops
right join laptops_stores
on laptops.id = laptops_stores.laptop_id
right join stores
on stores.id = laptops_stores.store_id
where laptops.name = "Macbook M1" && stores.name = "Thế Giới Di Động";
```



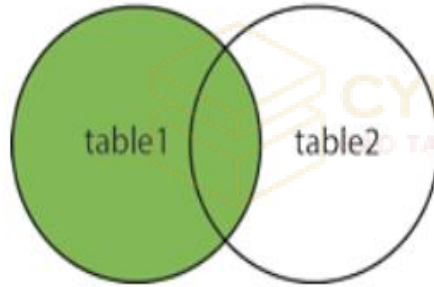
- join 2 bảng vào để có đầy đủ data
- where để xác định điều kiện

# Các kiểu JOIN

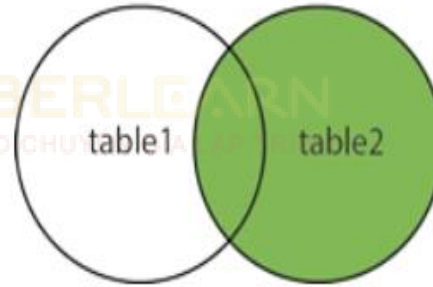
INNER JOIN



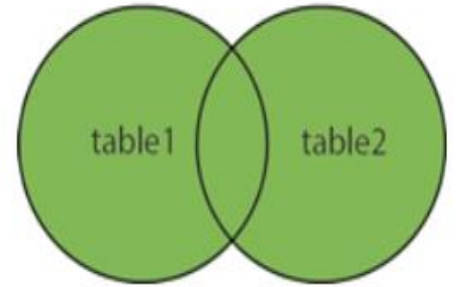
LEFT JOIN



RIGHT JOIN

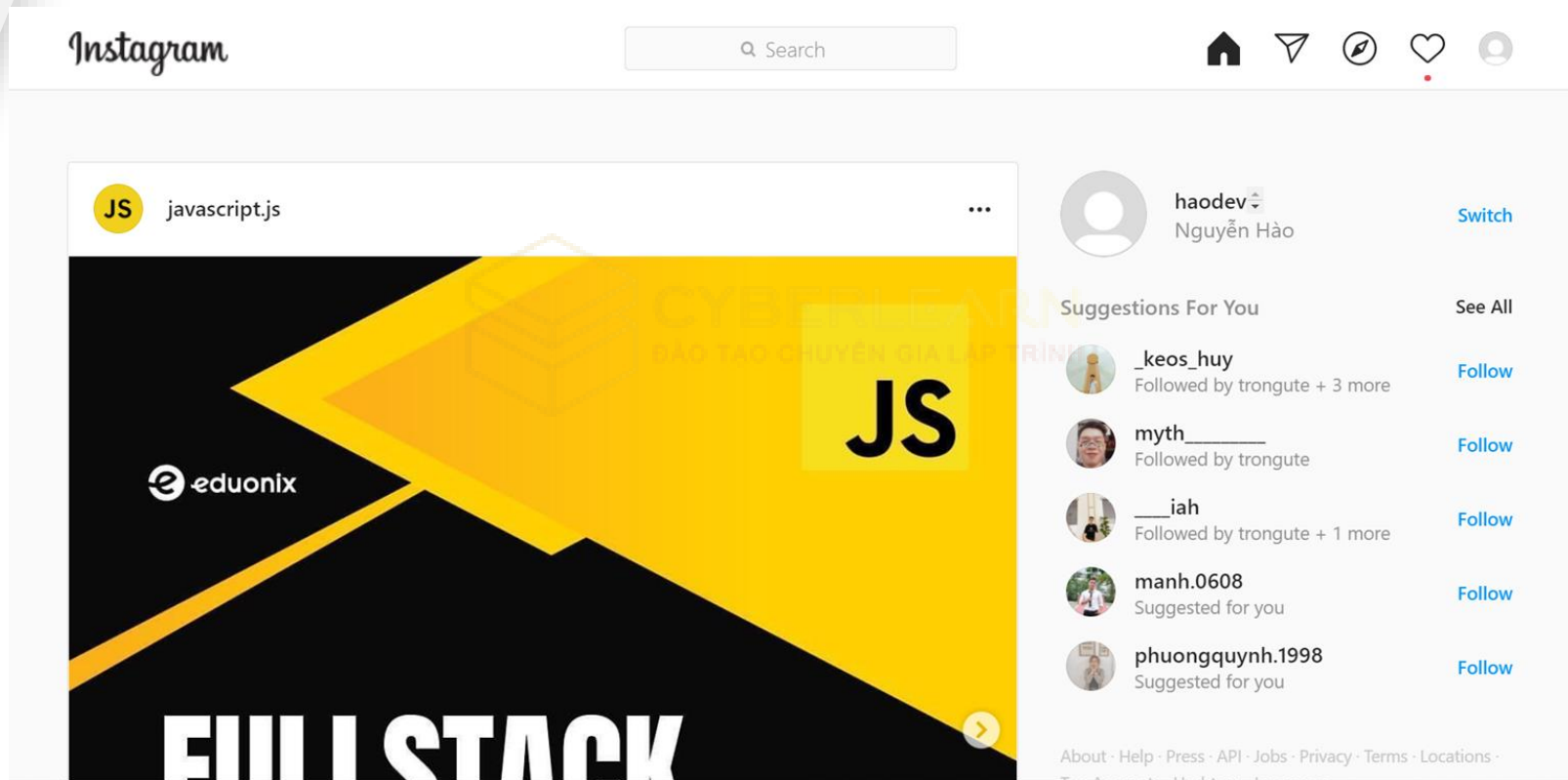


FULL OUTER JOIN

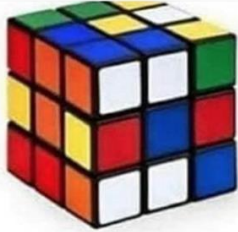










À thì ra là JOIN , LEFT JOIN và RIGHT JOIN dùng truy xuất liên kết các bảng với nhau .

# Dự án instagram



# Phân Tích Database

My programming experience		<div>JS javascript.js • Following ...</div> <div>JS javascript.js 🤔 . . . #programming #coding #programmer #developer #python #technology #javascript #code #coder #java #computerscience #html #webdeveloper #tech #css #software #webdevelopment #codinglife #linux #programmingmemes #softwaredeveloper #programmers #webdesign #programminglife #hacking #machinelearning #php #computer #pythonprogramming #bhfyp 1d</div> <div> lloud have mercy OFFER</div> <div></div> <div>Liked by karen.bergan.35 and others</div> <div>1 DAY AGO</div> <div> Add a comment... <span>Post</span></div>
The experience job recruiters want		
The salary they give		



# Phân Tích Database



javascript.js

Message



2,035 posts

609k followers

8 following

JavaScript

Community

📌 DM for credit/removal/promo 📧

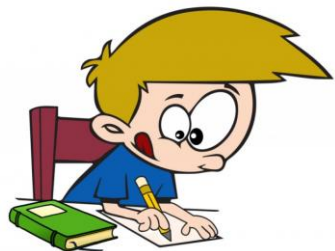
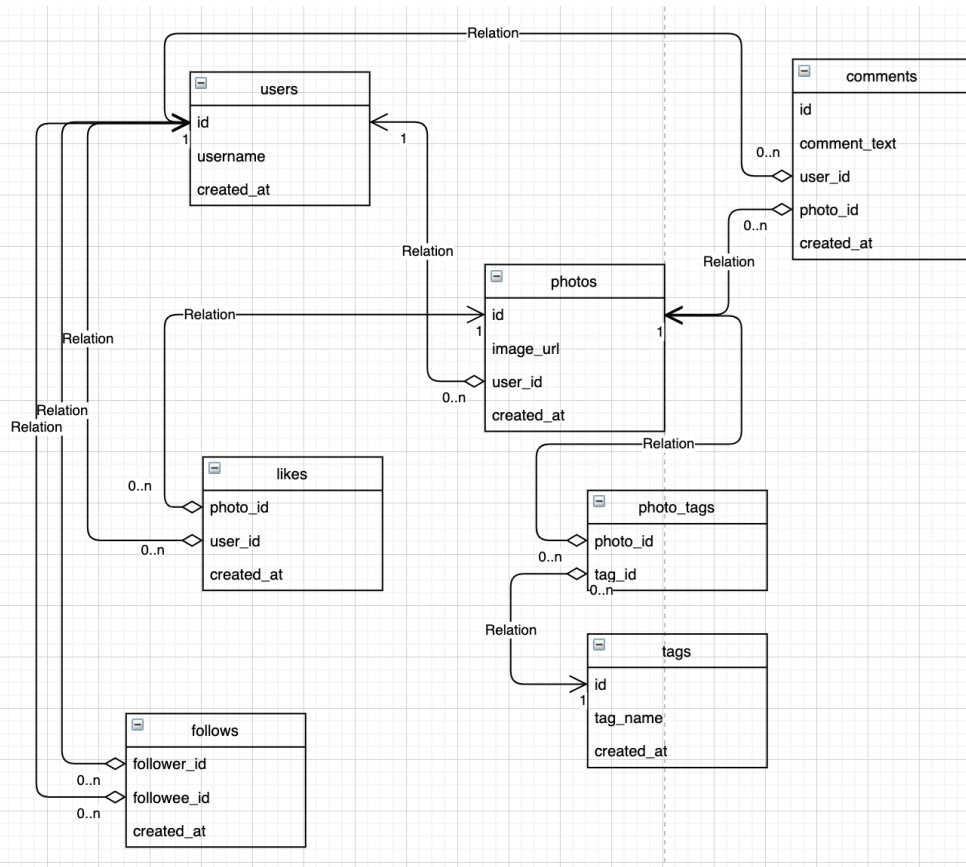
•

•

[www.docsie.io/try\\_docsie/inst-1](http://www.docsie.io/try_docsie/inst-1)



# Phân Tích Database



# Code Tạo Database



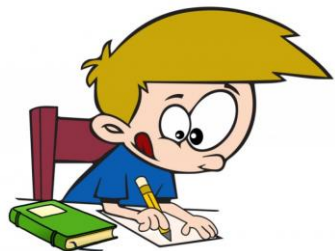
 CYBERSOFT  
Live On Video



# Thử Thách Query



- Tìm năm người dùng sử dụng ứng dụng lâu nhất .
- Tìm 2 ngày trong tuần có lượt đăng ký nhiều nhất.
- xác định người dùng không hoạt động ( người dùng ko đăng ảnh )
- xác định ảnh có nhiều like nhất và người dùng tạo ra nó
- tìm số lượng ảnh trung bình cho mỗi người .
- tìm 5 người thường được hashtags
- tình những người dùng đã thích một bức ảnh



còn nhiều câu truy vấn khác



# Happy Coding

CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH