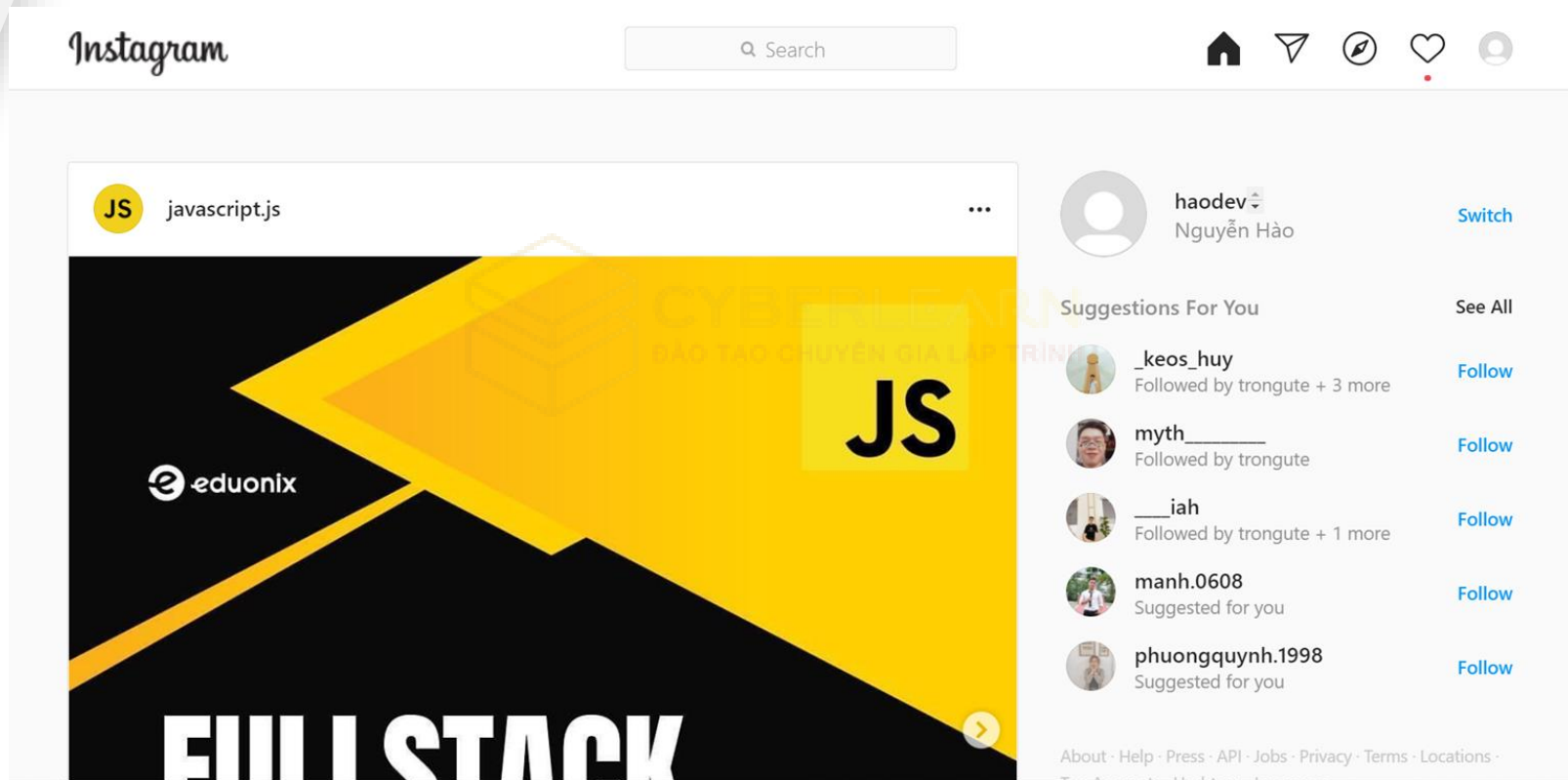


# MySQL

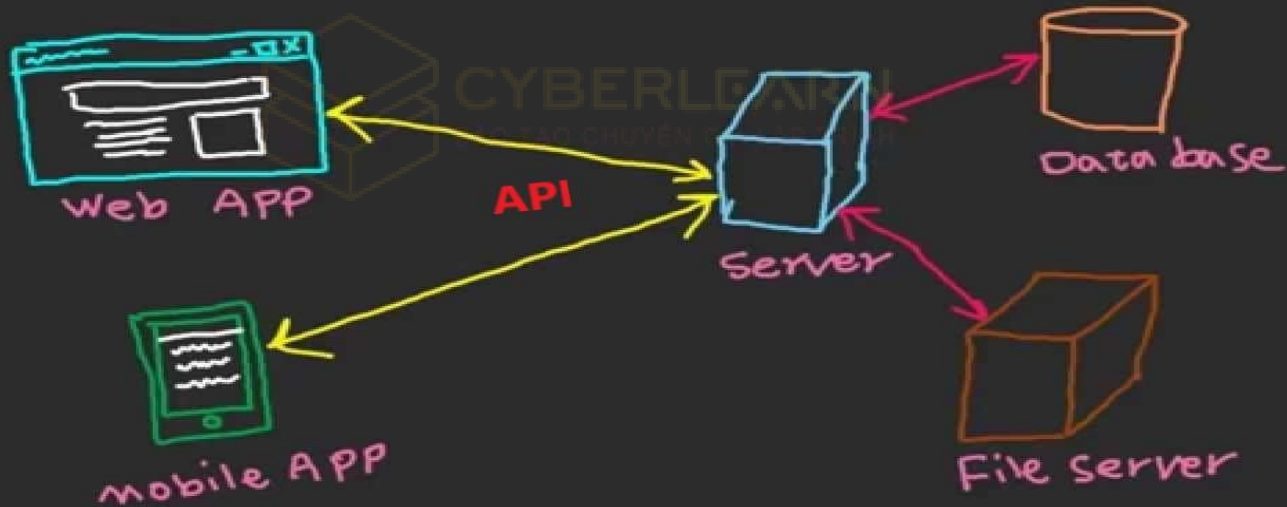
The logo for CYBERLEARN is positioned behind the MySQL text. It features a stylized 3D cube icon to the left of the word "CYBERLEARN" in a bold, sans-serif font. Below "CYBERLEARN" is the Vietnamese text "ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH" in a smaller, lighter font.

# Dự án instagram



# Mô hình website

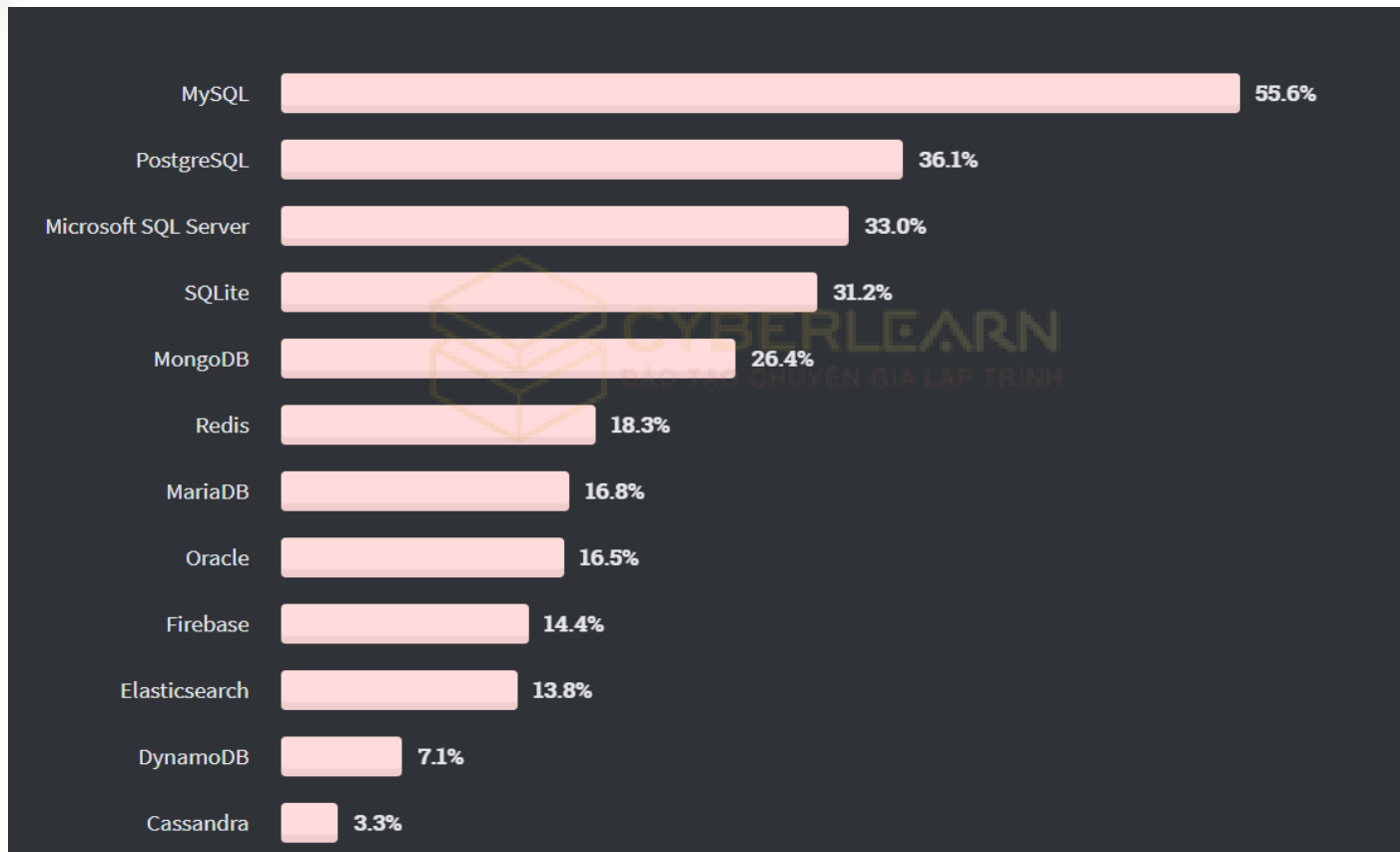
Front-End / Back-End



# Các loại SQL



# Tại Sao Lại Dùng MySQL ?

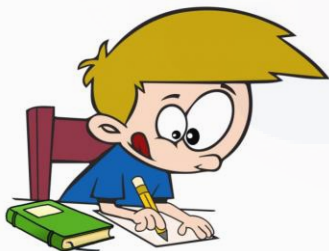


# MySQL



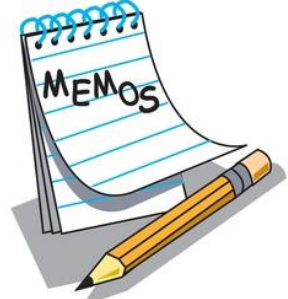
## Tóm Lại , MySQL Là

- 01 Hệ quản trị cơ sở dữ liệu
- 02 dùng MySQL vì nó phổ biến trên toàn cầu
- 03 MySQL chịu trách nhiệm xây dựng ra các cơ sở dữ liệu để lưu dữ liệu cho các trang web

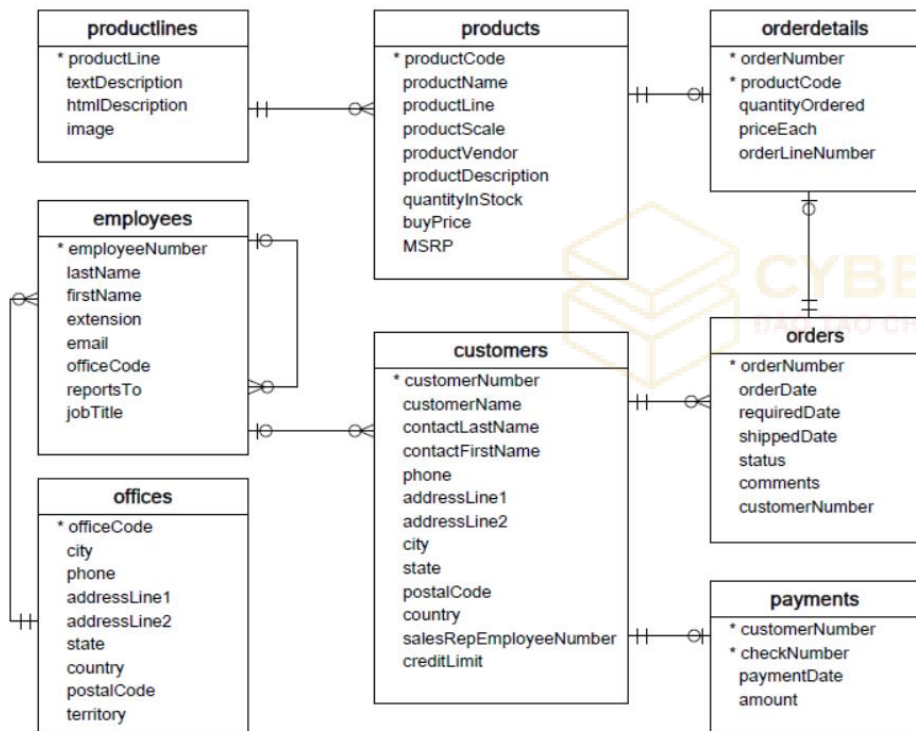


MySQL hay quá , chỉ mình đi cài đặt với

# Thao tác Database và Table



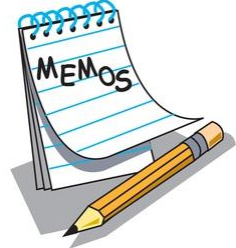
# Database là gì ?



- Bạn có thể hiểu đơn giản **DATABASE** là nơi lưu toàn bộ dữ liệu của một website.
- Bên trong database thường có các **TABLE**



# Table là gì ?



id	first_name	last_name	age	birthday
1	Hào	Nguyễn	19	1998-07-15
2	Như	Võ	18	1999-05-18
3	Hồng	Trần	20	1997-11-02

- Bạn có thể hiểu đơn giản **TABLE** là một cái bảng dùng để lưu trữ dữ liệu cho một loại đối tượng cụ thể.
- **COLUMN** là thuộc tính của đối tượng.
- **ROW** là thực thể từ đối tượng .



## Vấn đề

- 01 Làm sao để tạo và xóa một database ?
- 02 Làm sao để tạo và xóa table trong database ?
- 03 Làm sao để thực hiện các chức năng CRUD ?
- 04 Làm sao để lấy data từ table trong database ?
- 05 Làm sao để kết nối các table trong database ?
- 06 Làm sao để kiểm tra data trước khi lưu vào database?



MySQL sẽ giải quyết  
các vấn đề trên nhé



**Tạo , xóa và chọn database**

01

**Tạo và xóa table , kiểu dữ liệu cơ bản**

02

**Chèn data và ràng buộc thuộc tính**

03

**CRUD**

04

**Phương thức cho chuỗi**

05

**Tùy chỉnh lựa chọn**

06

**Hàm tổng hợp**

07

**Các toán tử logic**

08

**Các Quan hệ Table**

09

**Viết các câu truy vấn**

10



## Các tính năng cơ bản trong MySQL



# Ứng dụng quản lý người dùng trong instagram

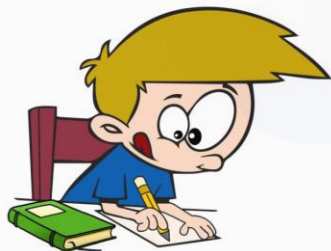


# Project Instagram



## Các tính năng của ứng dụng

- 01 Tạo một database ,và dùng database đó.
- 02 Tạo một table theo yêu cầu.
- 03 CRUD một đối tượng.
- 04 Viết các câu query theo yêu cầu



# tạo , xóa và dùng database

```
1  -- xóa database
2• DROP DATABASE IF EXISTS ig_clone;
3  -- tạo database
4• CREATE DATABASE ig_clone;
5  -- sử dụng database
6• USE ig_clone;
```

# Tạo bảng dữ liệu như sau

id	first_name	last_name	age	birthday
1	Hào	Nguyễn	19	1998-07-15
2	Như	Võ	18	1999-05-18
3	Hồng	Trần	20	1997-11-02



# Phân tích loại dữ liệu

- SOBTONE -

	id	first_name	last_name	age	birthday
Tên loại dữ liệu	số	chuỗi	chuỗi	số	ngày tháng năm
loại dữ liệu trong MySQL	INT	VARCHAR(255)	VARCHAR(255)	INT	DATE



Link Data type : [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)

# Tạo bảng

```
10  -- tạo table
11  • CREATE TABLE users(
12      id INT,
13      first_name VARCHAR(255),
14      last_name VARCHAR(255),
15      age INT,
16      birthday DATE
17  );
```



# Chỉnh sửa các cột trong bảng

Sửa độ dài của first\_name , last\_name chỉ trong 100 ký tự



```
19  -- chỉnh sửa các cột
20 • ALTER TABLE users
21   MODIFY COLUMN first_name VARCHAR(100),
22   MODIFY COLUMN last_name  VARCHAR(100);
```

# Thêm cột trong bảng



thêm cột type vào bảng :



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

```
19  -- thêm cột
20 • ALTER TABLE users
21  ADD COLUMN type VARCHAR(100);|
```

# Code xóa bảng

```
20  -- xóa table  
21 • DROP TABLE IF EXISTS users;
```



# Thêm một vài thành viên vào bảng

```
24  -- insert data
25  • INSERT INTO users(id,first_name,last_name,age,birthday)
26  VALUES
27      (1,"Hào", "Nguyễn", 19, "1998-07-15" ),
28      (2,"Như", "Võ", 18, "1999-05-18" ),
29      (3,"Hồng", "Trần", 20, "1997-11-02" );
30
```



Lúc trước học Front End thì id tự động tăng một đơn vị mà ta , sao bây giờ lại phải tạo tay vậy ?



# Constraints ( ràng buộc thuộc tính )



# Khóa chính và tự động tăng

```
24  -- AUTO_  
25 • ALTER TABLE users  
26  MODIFY COLUMN id INT AUTO_INCREMENT PRIMARY KEY;  
27  -- insert data  
28 • INSERT INTO users(first_name,last_name,age,birthday)  
29  VALUES  
30      ("Hào", "Nguyễn", 19, "1998-07-15" ),  
31      ("Nhu", "Võ", 18, "1999-05-18" ),  
32      ("Hồng", "Trần", 20, "1997-11-02" );
```



Giúp chúng ta định nghĩa ra các id tự động tăng và không bao giờ bị trùng nhau.



# giá trị không được null

first\_name , last\_name không được phép null



20 • ALTER TABLE users  
21 MODIFY COLUMN first\_name VARCHAR(100) NOT NULL,  
22 MODIFY COLUMN last\_name VARCHAR(100) NOT NULL;



Not Null bắt buộc chúng ta phải insert data vào table của database. chống dữ liệu rác.

# giá trị mặc định

giá trị mặc định cho type là “CUSTOMER”

- SOBTONE -

```
23  -- chỉnh sửa các cột
24 • ALTER TABLE users
25   MODIFY COLUMN first_name VARCHAR(100) NOT NULL,
26   MODIFY COLUMN last_name VARCHAR(100) NOT NULL,
27   MODIFY COLUMN type VARCHAR(100) NOT NULL DEFAULT "CUSTOMER";
```

CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



giúp chúng ta cài đặt giá trị mặc định cho một thuộc tính ( column )

# Bài tập : tạo bảng comments như bên dưới

id	comment_text	created_at
1	Hào đẹp trai	2020-12-05 01:30:59
2	Code tại CyberSoft thật đỉnh	2020-07-14 16:02:47
3	No Comment :))	2021-01-27 23:59:32



CREATE



READ



UPDATE



DELETE

---

C

R

U

D

# Create ( thêm )

```
24  -- insert data
25 •  INSERT INTO users(id,first_name,last_name,age,birthday)
26  VALUES
27      (1,"Hào", "Nguyễn", 19, "1998-07-15" ),
28      (2,"Như", "Võ", 18, "1999-05-18" ),
29      (3,"Hong", "Trần", 20, "1997-11-02" );
30
```



Insert và create là một câu lệnh nhé !

# Read ( đọc data từ table )

Lấy tất cả user và toàn bộ thuộc tính :



```
SELECT * FROM users;
```



dấu \* giúp chúng ta lấy toàn bộ thuộc tính của người dùng



# Read ( đọc data từ table )

Lấy tất cả user nhưng chỉ lấy first\_name, last\_name (

first\_name -> ten

last\_name -> ho

) :

```
SELECT first_name AS ten , last_name  
FROM users;
```



Dùng As để đặt lại tên thuộc tính theo yêu cầu của người dùng.



# Read ( đọc data từ table )

Lấy tất cả người dùng có độ tuổi lớn hơn hoặc bằng 19 :



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

```
54 • SELECT * FROM users  
55   where age >= 19;
```



mệnh đề where giúp lọc các lựa chọn theo một điều kiện c  
age >= 19





# Read ( đọc data từ table )

Lấy thuộc tính last\_name của tất cả người dùng có độ tuổi bé hơn hoặc bằng 20 :



```
54 • SELECT last_name FROM users
55   where age >= 20;
```



thay dấu \* bằng một thuộc tính cụ thể trong bảng thì nó chỉ lấy ra thuộc tính đó và không quan tâm các thuộc tính khác

# Update ( cập nhật data từ table )

cập nhật type thành “ADMIN” cho người dùng có id là 3 :



```
UPDATE users SET type = "ADMIN"  
WHERE id=3;
```



Update là lệnh giúp chúng ta cập nhật lại một đối tượng trong table của database



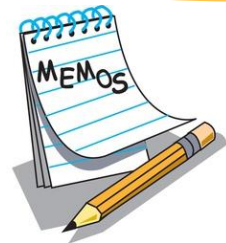
# Update ( cập nhật data từ table )

cập nhật các user có age = 19 lên thành 20

**SET** SQL\_SAFE\_UPDATES = 0;

**update** users **SET** age = 20  
**where** age = 19;

RLEARN  
TRUYỀN GIA LẬP TRÌNH



Update là lệnh giúp chúng ta cập nhật lại một đối tượng trong table của database



# Delete ( xóa data từ table )

xóa người dùng có id là 2:

```
55 • DELETE FROM users
56   WHERE id=2;
```



CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



DELETE là lệnh giúp chúng ta xóa một đối tượng trong Table của database



# Delete ( xóa data từ table )

Xóa tất cả user:

**DELETE FROM** users;



**DELETE** mà không có điều kiện thì chúng ta sẽ xóa tất cả đối tượng trong một table



## Bài tập : CRUD bảng comments

id	comment_text	created_at
1	Hào đẹp trai	2020-12-05 01:30:59
2	Code tại CyberSoft thật đỉnh	2020-07-14 16:02:47
3	No Comment :))	2021-01-27 23:59:32

# Các hàm xử lý chuỗi



# Concat

- SOBTONE -

Hãy lấy ra thuộc tính username được kết từ first\_name và last\_name theo cú pháp sau : first\_name last\_name

```
66  -- nối chuỗi
67 • SELECT CONCAT(first_name , " " , last_name) AS username
68  FROM users;
```



CONCAT là hàm nối các tham số thành một chuỗi.  
AS là định nghĩa lại tên của thuộc tính.



# SubString

Hãy lấy ra năm sinh của tất cả người dùng :

```
70  -- cắt chuỗi
71 • SELECT SUBSTR(birthday,1,4) as year_of_birth
72  from users;
```

SUBSTR là hàm cắt chuỗi . tham số :

- 1/ chuỗi cần cắt
- 2/ vị trí bắt đầu cắt
- 3/ số lượng ký tự cần cắt



# Replace

Hãy thay thế các type là “ADMIN” thành “SUPER\_ADMIN”

74 -- thay thế

75 • `SELECT REPLACE(type , "ADMIN" , "SUPER_ADMIN") as type_edit`

76 `FROM users;`



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



Ủa , vậy REPLACE khác UPDATE chỗ nào mấy cha ?

# Reverse

Hãy đảo ngược last\_name:

```
79  -- đã ngược
80 • SELECT REVERSE(last_name) AS last_name_reverse
81  FROM users;
```



REVERSE là hàm đảo ngược một cái chuỗi .

# Upper , Lower

Hãy in hoa first\_name và in thường last\_name :

86 -- in thường và in hoa  
87 • **SELECT** UPPER(first\_name) , LOWER(last\_name)  
88 **FROM** users;



**UPPER** là hàm in hoa các ký tự.  
**LOWER** là hàm in thường các ký tự.

# Bài tập : xử lý nội dung comment

yêu cầu : nếu comment dài hơn 8 ký tự thì cắt bỏ các ký tự phía sau thay thế bằng dấu “...”

ví dụ : Anh Hào rất đẹp trai

=> Anh Hào ...



# Tinh chỉnh các câu lệnh Select



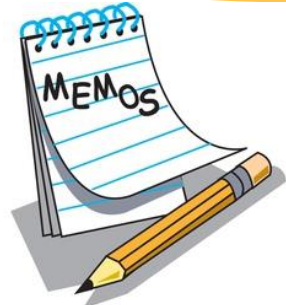
# Distinct

Hãy lấy ra những last\_name khác nhau :

```
93 • SELECT DISTINCT last_name
94   FROM users;
```




DISTINCT lấy các giá trị khác nhau từ cột của bảng

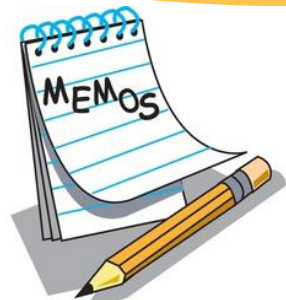


# Order By

Hãy sắp xếp người dùng theo độ tuổi có thứ tự tăng dần :

96 -- sắp xếp tăng dần  
97 • `SELECT * FROM users`  
98 `ORDER BY age ASC;`

 **ORDER BY** là sắp xếp theo một thuộc tính nào đó trong bảng  
1/ **ASC** là tăng dần ( 0-9 , a-z , A-Z )  
2/ **DESC** giảm dần , thử test cái này nhé :))





# Limit

Hãy lấy ra 2 người dùng :

```
06 • SELECT * FROM users
07   LIMIT 2;
```

LIMIT là giới hạn số phần tử lấy ra, cụ thể ở đây là 2 phần



# Like

Hãy tìm kiếm các last\_name gần giống với chuỗi “ần”

```
11  -- search like -- tìm kiếm gần giống
12 • SELECT * FROM users
13  WHERE last_name LIKE '%ần%';
```



À thì ra like là search gần giống ( không cần chính xác ) , % là ký tự gì cũng được .các chuỗi “trần” , “dần” , “vàng” đều phù hợp nhé.

# wildcards

## Wildcard Characters in MS Access

Symbol	Description	Example
*	Represents zero or more characters	bl* finds bl, black, blue, and blob
?	Represents a single character	h?t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
!	Represents any character not in the brackets	h[!oa]t finds hit, but not hot and hat
-	Represents a range of characters	c[a-b]t finds cat and cbt
#	Represents any single numeric character	2#5 finds 205, 215, 225, 235, 245, 255, 265, 275, 285, and 295



có rất nhiều ký tự đặc biệt khi dùng search like với sql , link tham khảo :  
[https://www.w3schools.com/sql/sql\\_wildcards.asp](https://www.w3schools.com/sql/sql_wildcards.asp)

# Bài tập : xử lý số lượng comment được lấy ra và sắp xếp

yêu cầu :

- lấy ra 3 comment
- sắp xếp theo thời gian tạo ra ( comment nào tạo ra trước thì để ở dưới , comment nào tạo ra sau thì để ở trên )



# Các hàm tổng hợp

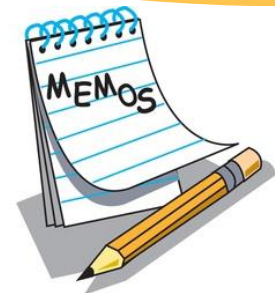


# Count

Đếm xem có bao nhiêu user trong table users ?

```
SELECT COUNT(id)
FROM users;
```

Giúp chúng ta tính toán xem trong table users có bao nhiêu user ?



# Count

Đếm xem có bao nhiêu last\_name khác nhau trong tất cả các user ?



```
SELECT COUNT(DISTINCT last_name)
FROM users;
```



Vận dụng Count kết hợp với Distinct để tính toán xem có bao nhiêu last\_name khác nhau



# Max và Min



Tìm age lớn nhất và age nhỏ nhất ?

**SELECT**

MIN(age) **as** age\_min,

MAX(age) **as** age\_max

**FROM** users;



CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



Max và Min giúp chúng ta tìm giá trị lớn nhất và nhỏ nhất có thể tìm theo date luôn nhé





# Sub Query

Tìm các user có age lớn nhất và age nhỏ nhất ?

-- các user có age nhỏ nhất

**SELECT \* FROM** users

**WHERE** age = (**SELECT** MIN(age) **FROM** users);

-- các user có age lớn nhất

**SELECT \* FROM** users

**WHERE** age = (**SELECT** MAX(age) **FROM** users);

câu lệnh query phụ giúp chúng ta tìm độ tuổi nhỏ nhất sau đó mới tìm tới user có độ tuổi được tìm thấy trong câu truy vấn phụ



# Group BY

Lập nhóm các người dùng có last\_name giống nhau :

```
-- LẬP NHÓM  
SELECT * FROM users  
GROUP BY last_name;
```

CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

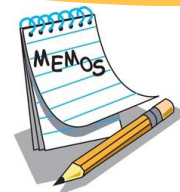


cách này hay hơn việc đếm last\_name khác nhau , ngoài ra lập nhóm ra để gì ?



# Sum

Tính tổng tiền của các user ?



-- tổng tiền của các user

```
SELECT SUM(money) FROM users;
```



SUM cho phép chúng ta tính tổng tiền của tất cả user



# Sum

Tính tổng tiền của các user có cùng last\_name ?



```
-- tổng tiền của các user có cùng last_name  
SELECT last_name , SUM(money) FROM users  
GROUP BY last_name;
```



Kết hợp **SUM** và **GROUP BY** cho phép chúng ta tính tổng tiền của tất cả user có cùng last\_name



# AVG

Tính trung bình tiền của các user có cùng last\_name ?

```
-- tính trung bình tiền của các user có cùng last_name  
SELECT last_name , AVG(money) FROM users  
GROUP BY last_name;
```



AVG cho phép chúng ta tính trung bình số tiền của tất cả user có cùng last\_name



# Bài tập

yêu cầu :

- tạo bảng books chứa các thuộc tính sau : id , name , price , page, released\_year , author\_name.
- tạo một vài cuốn sách tùy ý.
- đếm xem có bao nhiêu cuốn sách ?
- đếm xem có bao nhiêu cuốn sách của 1 tác giả.
- tổng số tiền của các cuốn sách có cùng 1 tác giả.
- tìm các cuốn sách phát hành mới nhất và cũ nhất



# Data Type



# Char

cài đặt cho first name chỉ chứa 20 ký tự :



```
-- Table
DROP TABLE IF EXISTS users;
CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(255),
    age INT,
    money DECIMAL,
    birthday DATE
);
```

- CHAR(size) : size nằm trong khoảng 0-255
- size là số ký tự có thể chứa của column
- thường dùng để lưu các chuỗi



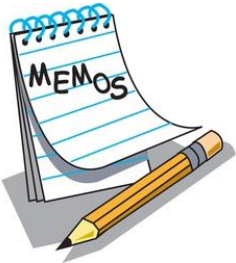
# Varchar

cài đặt cho last name chỉ chứa 30 ký tự :

-- Table

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users(  
    id INT primary key auto_increment,  
    first_name CHAR(20),  
    last_name VARCHAR(30),  
    age INT,  
    money DECIMAL,  
    birthday DATE  
);
```



- VARCHAR(size) : size nằm trong khoảng 0 - 65535
- size là số ký tự có thể chứa của column
- thường dùng để lưu các chuỗi

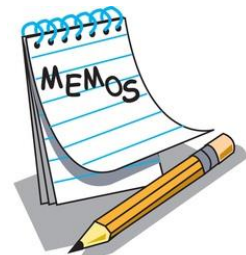
# Decimal

Cài đặt money là số thập phân có 4 chữ số và sau dấu “.” là 2 chữ số

-- Table

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users(  
    id INT primary key auto_increment,  
    first_name CHAR(20),  
    last_name VARCHAR(30),  
    age INT,  
    money DECIMAL(4,2),  
    birthday DATE  
);
```



- DECIMAL(size , d) : size nằm trong khoảng 0 - 65 , d nằm trong khoảng 0 - 10.
- size là tổng các chữ số có trong một con số
- d là các số sau dấu “.”
- thường dùng để lưu các số thập phân

# Float

Cài đặt money là số thực:

```
-- Table
DROP TABLE IF EXISTS users;

CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(30),
    age INT,
    money FLOAT,
    birthday DATE
);
```

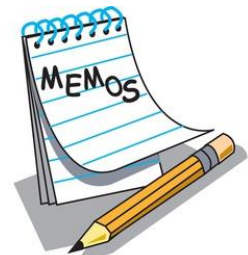


- Float : có thể lưu một con số thực khoảng 7 chữ số.
- thường dùng để lưu các số thực

# Double

Cài đặt money là số thực:

```
-- Table
DROP TABLE IF EXISTS users;
CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(30),
    age INT,
    money DOUBLE,
    birthday DATE
);
```



- Double: có thể lưu một con số thực khoảng 15 chữ số.
- thường dùng để lưu các số thực

# Date

## Cài đặt birthday lưu ngày tháng năm

```
-- Table
DROP TABLE IF EXISTS users;
CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(30),
    age INT,
    money DOUBLE,
    birthday DATE
);
```



- Date giúp chúng ta lưu được ngày tháng năm
- định dạng như sau : YYYY-MM-DD
- ví dụ : “1998-11-05”

# Time

Cài đặt birthday lưu giờ phút giây

```
-- Table
DROP TABLE IF EXISTS users;

CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(30),
    age INT,
    money DOUBLE,
    birthday Time
);
```



- Time giúp chúng ta lưu được ngày tháng năm
- định dạng như sau : hh:mm:ss
- ví dụ : “11:30:05”

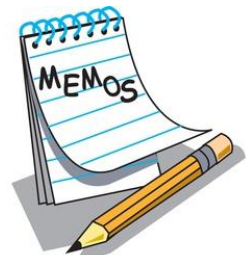
# DateTime

Cài đặt birthday lưu ngày tháng năm giờ phút giây

```
-- Table
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users(  
    id INT primary key auto_increment,  
    first_name CHAR(20),  
    last_name VARCHAR(30),  
    age INT,  
    money DOUBLE,  
    birthday DATETIME  
);
```

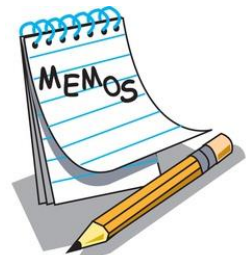


- DateTime giúp chúng ta lưu được ngày tháng năm giờ phút giây
- định dạng như sau : YYYY-MM-DD hh:mm:ss
- ví dụ : “2038-01-09 03:14:07”

# TimeStamp

Cài đặt createAt lưu mốc thời gian

```
-- Table
DROP TABLE IF EXISTS users;
CREATE TABLE users(
    id INT primary key auto_increment,
    first_name CHAR(20),
    last_name VARCHAR(30),
    age INT,
    money DOUBLE,
    birthday DATETIME,
    createAt TIMESTAMP DEFAULT NOW()
);
```



- TimeStamp giúp chúng ta lưu được mốc thời gian
- thường lưu lại thời gian tạo ra hoặc thời gian cập nhật



# Toán tử



# OPERATOR



Trong MySQL có các toán tử thông dụng sau :

1/ khác ( != )

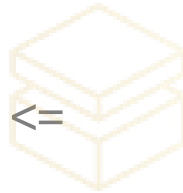
2/ NOT

3/ > , < , >= , <=

4/ And , Or

4/ Between

5/ In



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



# Toán Tử Khác

Hãy lấy ra những người dùng có age khác 18

```
-- not equal  
SELECT * FROM users  
WHERE age != 18;
```

À thì ra nó ngược lại với dấu =



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



# Toán NOT

Hãy lấy ra những người dùng có last\_name không gần giống với “ần”

```
-- not like  
SELECT * FROM users  
WHERE last_name NOT LIKE '%rầ%';
```



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



À thì ra nó ngược lại với LIKE



# Toán logic

Hãy lấy ra những người dùng có age lớn hơn bằng 16 và bé hơn bằng 18 hoặc lớn hơn bằng 19 và bé hơn bằng 20

-- nhỏ hơn, lớn hơn, ||, &&

SELECT \* FROM users

WHERE (age >= 16 && age <= 18) || (age >= 19 && age <= 20);



À thì ra “và” là && còn “hoặc” là ||

# Between

Hãy lấy ra những người dùng có age từ 17 -> 20

-- BETWEEN

SELECT \* FROM users

WHERE age BETWEEN 17 AND 20



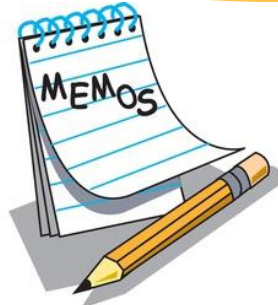
À thì ra là trong một khoảng cụ thể ở đây là : 17 -> 20

# In

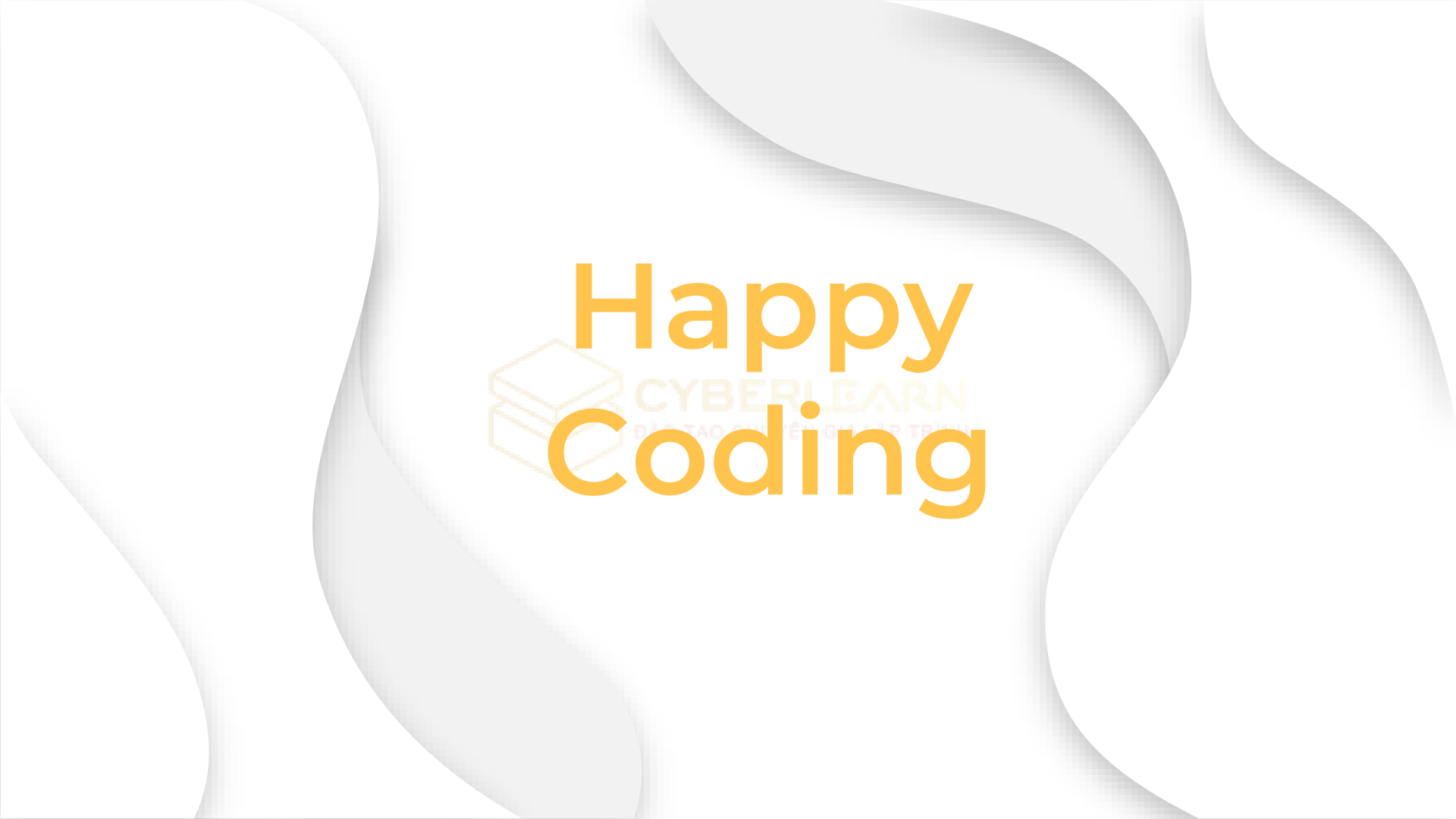
Hãy lấy ra những người dùng có age như sau : 16 , 18 , 20

-- In


```
SELECT * FROM users  
WHERE age IN (16,18,20);
```



À thì ra là lấy ra những kết quả mà mình đã đề xuất



# Happy Coding



CYBERLEARN  
ĐÀO TẠO CHUYÊN NGHIỆP