

# Cơ sở dữ liệu quan hệ - SQL Basic

---

Ths. Vũ Duy Khương



# Nội dung

- 1 **Giới thiệu về SQL**
- 2 **Truy vấn cơ bản**
- 3 **Một số function trong SQL**
- 4 **Toán tử**
- 5 **Bài tập thực hành**

## ❖ Cú pháp

**SELECT** <Danh sách các cột>

**FROM** <Danh sách các bảng>

**WHERE** <Điều kiện>

<Điều kiện>

- Biểu thức boolean xác định dòng nào sẽ được rút trích
- Nối các biểu thức: AND, OR, và NOT
- Phép toán:  $<$  ,  $>$  ,  $\leq$  ,  $\geq$  ,  $\neq$  ,  $=$  , LIKE và BETWEEN



# Mệnh đề SELECT

Truy vấn toàn bộ các cột của bảng

❖ Cú pháp:

```
SELECT * from ten_bang;
```

❖ Ví dụ:

Lấy tất cả các cột của bảng film

```
select * from sakila.film;
```



# Mệnh đề SELECT

**Truy vấn một số cột của bảng:**

❖ **Cú pháp:**

Select [cot\_1], [cot\_2],...[cot\_n] from table

❖ **Ví dụ:** Truy vấn cột film\_id, title,length, rental\_reate từ bảng film

```
select film_id,title,length,rental_rate  
from sakila.film;
```



# Mệnh đề SELECT

Tùy biến giá trị trả về của câu truy vấn:

❖ **Cú pháp:**

Sử dụng một số function, toán tử để custom dữ liệu trả về của lệnh truy vấn

❖ **Ví dụ:**

Cộng 2 cột title, **release\_year** từ bảng film

```
select concat(title,'_',release_year) new_col,length f  
from sakila.film;
```



# Mệnh đề FROM

## Truy vấn từ bảng đơn

### ❖ Cú pháp:

**SELECT** [\*],[danh\_sach\_cot] **FROM** ten\_bang;

### ❖ Ví dụ:

Truy vấn dữ liệu từ bảng category

**select** \* **from** sakila.category

### ❖ Sử dụng alias cho bảng:

**select** c.\* **from** sakila.category c

**select** c.category\_id,c.name,c.last\_update  
**from** sakila.category c;



# Mệnh đề FROM

**select** c.\* **from** sakila.category c

category				
select c.category_id   Enter a SQL expression to				
Grid	123	category_id	ABC name	last_update
	1	1	Action	2006-02-15 04:46:27.0
	2	2	Animation	2006-02-15 04:46:27.0
	3	3	Children	2006-02-15 04:46:27.0
	4	4	Classics	2006-02-15 04:46:27.0
	5	5	Comedy	2006-02-15 04:46:27.0
	6	6	Documentary	2006-02-15 04:46:27.0
	7	7	Drama	2006-02-15 04:46:27.0
	8	8	Family	2006-02-15 04:46:27.0



# Mệnh đề FROM

## Truy vấn từ nhiều bảng

### ❖ Cú pháp:

**SELECT** [...], [danh\_sach\_cot] **from** bang\_1, bang\_2, ..., bang\_n

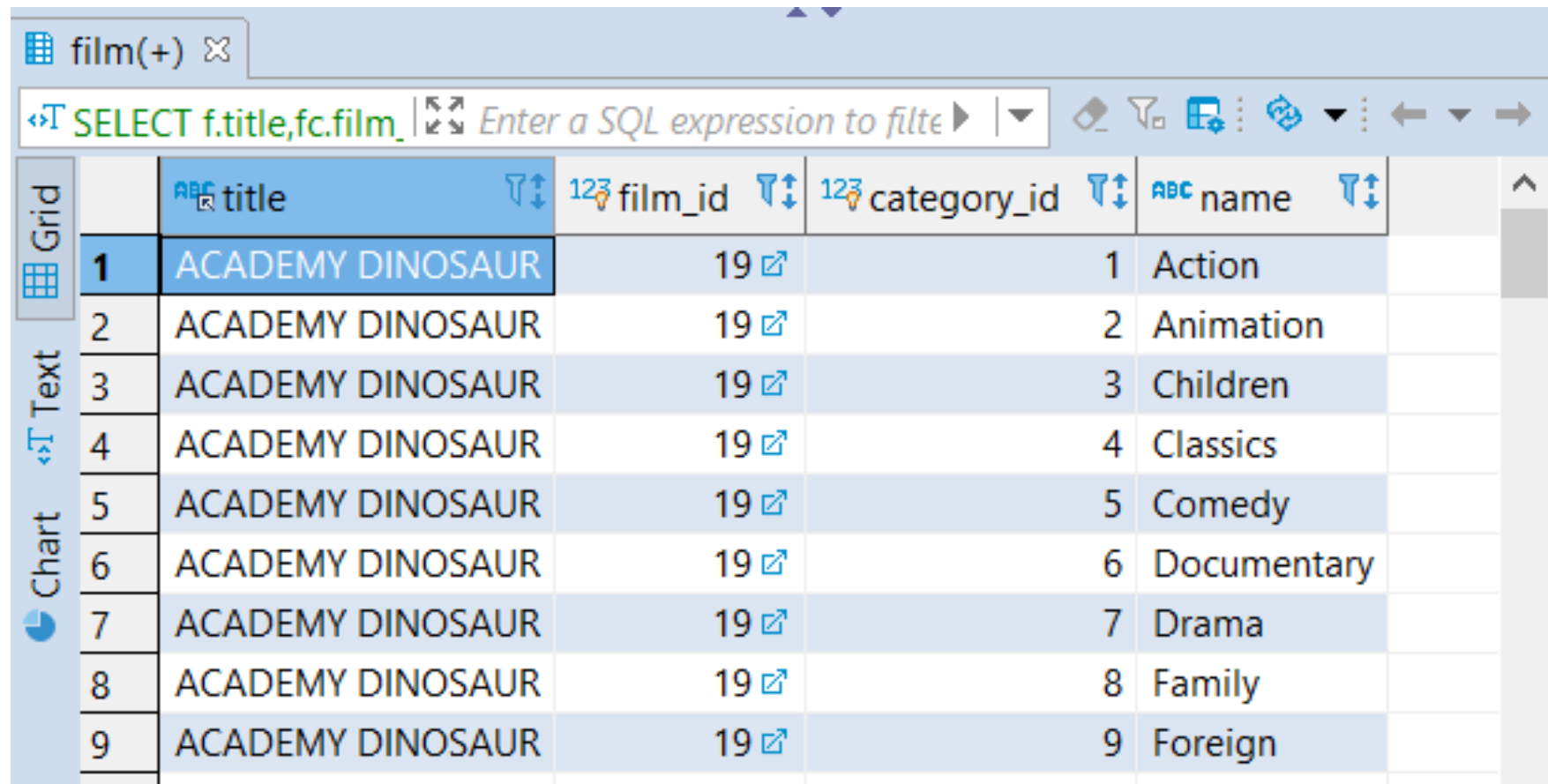
### ❖ Ví dụ: Truy vấn dữ liệu từ bảng film, film\_category, category

SELECT f.title, fc.film\_id, c.\*

from film f, film\_category fc, category c

# Mệnh đề FROM

Kết quả:



	title	film_id	category_id	name
1	ACADEMY DINOSAUR	19	1	Action
2	ACADEMY DINOSAUR	19	2	Animation
3	ACADEMY DINOSAUR	19	3	Children
4	ACADEMY DINOSAUR	19	4	Classics
5	ACADEMY DINOSAUR	19	5	Comedy
6	ACADEMY DINOSAUR	19	6	Documentary
7	ACADEMY DINOSAUR	19	7	Drama
8	ACADEMY DINOSAUR	19	8	Family
9	ACADEMY DINOSAUR	19	9	Foreign



# Mệnh đề WHERE

**WHERE:** Sử dụng để giới hạn phạm vi tìm kiếm

❖ **Cú pháp:**

```
SELECT [*],[danh_sach_cot] FROM ten_bang  
WHERE [cac_bieu_thuc_dk]
```

➤ **Chú ý:**

Mệnh đề where có thể được dùng trong mệnh đề update, delete

❖ **Các biểu thức điều kiện:**

Phép toán: <, >, <=, >=, <>, =, like, between, in, not in

Nối các biểu thức: and, or



# Mệnh đề WHERE

**Ví dụ:** Tìm kiếm những bộ phim có title = 'ANGELS LIFE'  
**select \* from sakila.film f where title = 'ANGELS LIFE';**

film				
select * from sakila.film f where title = 'ANGELS LIFE'   Enter a SQL expression to filter results				
Grid		123 film_id	ABC title	ABC description
	1	25	ANGELS LIFE	A Thoughtful Display of a Woman And a Ast



# WHERE với từ khóa LIKE

Được sử dụng để tìm kiếm dữ liệu theo một mẫu định sẵn

Có 2 dạng mẫu thường sử dụng:

- Mẫu %: Bất kỳ ký tự nào
- Mẫu \_: Dấu gạch dưới đại diện cho 1 ký tự

Ví dụ `Select * from employee where name like 'Ab_m'`

- Nghĩa là tìm tất cả các nhân viên có tên gồm 4 ký tự
- Bắt đầu là ký tự Ab và kết thúc là ký tự m

# WHERE với từ khóa LIKE

Mẫu sử dụng	Ý nghĩa
Where name like 'a%'	Tìm name bắt đầu bằng ký tự a
Where name like '%a'	Tìm name kết thúc bằng ký tự a
Where name like '%a%'	Tìm name có chứa ký tự a
Where name like 'a_%'	Tìm name có ít nhất 2 ký tự và bắt đầu là a
Where name like '_a%'	Tìm name có ký tự thứ 2 là a
Where name like 'a%b'	Tìm name bắt đầu là a và kết thúc là b



# WHERE với từ khóa LIKE

**Ví dụ 1:** Tìm các nước có tên bắt đầu là chuỗi ký tự 'Au'

**Ví dụ 2:** Tìm các nước có tên chứa chuỗi ký tự '%Au%'

**Ví dụ 3:** Tìm các nước mà tên có 2 ký tự bắt đầu là chuỗi Bn

**Ví dụ 4:** Tìm các nước mà tên bắt đầu là chuỗi 'Au' kết thúc là 'ia'



# WHERE với từ khóa LIKE

**Ví dụ 1:** Tìm các nước có tên bắt đầu là chuỗi ký tự 'Au'  
*select \* from sakila.country where country like 'Au%';*

**Ví dụ 2:** Tìm các nước có tên chứa chuỗi ký tự '%Au%'  
*select \* from sakila.country where country like '%Au%';*

**Ví dụ 3:** Tìm các nước mà tên có 2 ký tự bắt đầu là chuỗi Bn  
*select \* from sakila.country where country like 'Bn\_';*

**Ví dụ 4:** Tìm các nước mà tên bắt đầu là chuỗi 'Au' kết thúc là 'ia'  
*select \* from sakila.country where country like 'Au%ia';*





# Toán tử AND, OR, NOT

## Toán tử AND và OR:

Được sử dụng để lọc các dữ liệu dựa trên nhiều hơn một điều kiện:

- Toán tử AND hiển thị một dữ liệu nếu tất cả các điều kiện được phân tách bằng AND đều ĐÚNG.
- Toán tử OR hiển thị một dữ liệu nếu bất kỳ điều kiện nào được phân tách bởi OR là TRUE.

## Toán tử NOT:

Hiển thị bản ghi nếu (các) điều kiện KHÔNG ĐÚNG.



# Toán tử AND,OR,NOT

## Toán tử AND

### Cú pháp:

**SELECT** [danh\_sach\_cot] **FROM**

ten\_bang

**WHERE** condition1 **AND** condition2 **AND** ... conditionN;

### Ví dụ:

Tìm những bộ film sản xuất năm 2006 và độ dài film lớn hơn 60 phút

*select \* from sakila.film where release\_year='2006' and length>60;*



# Toán tử AND,OR,NOT

## Toán tử OR

### Cú pháp:

**SELECT** [danh\_sach\_cot] **FROM**

ten\_bang

**WHERE** condition1 **OR** condition2 **OR**... conditionN;

### Ví dụ:

Tìm những bộ film có tỉ lệ rating là G hoặc độ dài hơn 60 phút

*select \* from sakila.film where rating='G' or length>60;*



# Toán tử AND,OR,NOT

## Toán tử NOT

### Cú pháp:

```
SELECT [danh_sach_cot] FROM  
ten_bang  
WHERE NOT condition;
```

### Ví dụ:

Tìm tất cả các quốc gia mà tên không là ‘Australia’

```
select * from sakila.county where not country= 'Australia'
```



# WHERE in và not in

❖ **Where in:** Được sử dụng để tìm các giá trị có trong một tập

**Cú pháp:**

*select [ds\_cot] from ten\_table where cot\_tim in (v1,v2);*

❖ **Where not in:** Được sử dụng để tìm các giá trị không có trong 1 tập

**Cú pháp:**

*select [ds\_cot] from ten\_table  
where cot\_tim not in (v1,v2);*

# WHERE in và not in

**Ví dụ where in:** Tìm các bộ film có tỉ lệ rating là 'R','PG-13','NC-17','A'

*Select \* from film where rating in ('R','PG-13','NC-17','A')*

**Ví dụ where not in:** Tìm các bộ film có tỉ lệ rating không là 'R','PG-13','NC-17','A'

*Select \* from film where rating not in ('R','PG-13','NC-17','A')*



# Từ khóa limit và offset

Được sử dụng để giới hạn bản ghi trả về của câu lệnh select

**Cú pháp:**

```
select [ds_cot] from ten_bang limit size offset index
```

size: giới hạn số lượng bản ghi trả về

index: Vị trí bắt đầu của khối size trả về

**Ví dụ:** limit 10 offset 2:

Trả về 10 bản ghi, bắt đầu từ bản ghi thứ 3 tới bản ghi 12



# Từ khóa limit và offset

**Ví dụ:** Trả về giới hạn 10 bản ghi từ bảng film, bắt đầu từ bản ghi thứ 20

➤ `Select * from film limit 10 offset 20;`





# Từ khóa limit và offset

Kết quả:

country				
select * from sakila.country limit 10 offset 20				
	country_id	country	last_update	
1	21	Chad	2006-02-15 04:44:00.0	
2	22	Chile	2006-02-15 04:44:00.0	
3	23	China	2006-02-15 04:44:00.0	
4	24	Colombia	2006-02-15 04:44:00.0	
5	25	Congo, The Demc	2006-02-15 04:44:00.0	
6	26	Czech Republic	2006-02-15 04:44:00.0	
7	27	Dominican Repub	2006-02-15 04:44:00.0	
8	28	Ecuador	2006-02-15 04:44:00.0	
9	29	Egypt	2006-02-15 04:44:00.0	
10	30	Estonia	2006-02-15 04:44:00.0	



# Một số function trong SQL

- ❖ Hàm **Min()** và **Max**
- ❖ **COUNT()**, **AVG()** và **SUM()**
- ❖ **IFNULL()**, **ISNULL()**, **COALESCE()**, and **NVL()**



# Hàm Min(), Max()

**Hàm Min():** Trả về giá trị nhỏ nhất của cột đã chọn

❖ **Cú pháp:**

**SELECT** MIN(column\_name)

**FROM** table\_name

**WHERE** condition;

**Ví dụ:** Tìm thời lượng ngắn nhất trong các bộ film

*select min(length) from sakila.film f;*



# Hàm Max()

**Hàm Max():** Trả về giá trị nhỏ nhất của cột đã chọn

❖ **Cú pháp:**

**SELECT** MAX(column\_name)

**FROM** table\_name

**WHERE** condition;

❖ **Ví dụ:** Tìm thời lượng dài nhất trong các bộ film

*select max(length) from sakila.film f;*



# Hàm COUNT(),AVG(),SUM()

## Hàm COUNT():

Trả về số hàng phù hợp với tiêu chí được chỉ định

### ❖ Cú pháp:

```
SELECT COUNT(*)  
FROM table_name  
WHERE condition;
```

### ❖ Ví dụ:

Đếm số lượng tất cả các bộ film có rating='G'

```
select count(*) from sakila.film f where rating='G';
```



# Hàm COUNT(),AVG(),SUM()

## Hàm AVG():

trả về giá trị trung bình của một cột số

### ❖ Cú pháp:

```
SELECT AVG(ten_cot)
FROM table_name
WHERE condition;
```

### ❖ Ví dụ:

Tính thời gian trung bình của các bộ film có rating='G'

```
select avg(length) from sakila.film f where rating='G';
```



# Hàm COUNT(),AVG(),SUM()

## Hàm SUM():

Trả về tổng của một cột số

### ❖ Cú pháp:

```
SELECT SUM(ten_cot)
FROM table_name
WHERE condition;
```

❖ Lưu ý: giá trị null không được tính

### ❖ Ví dụ:

Tính tổng thời gian của các bộ film có rating='G'

```
select sum(length) from sakila.film f where rating='G';
```



# Hàm IFNULL(),COALESCE()

## Giá trị null là gì?

- ❖ Trường có giá trị NULL là trường không có giá trị.
- ❖ Nếu một trường trong bảng là tùy chọn, có thể chèn dữ liệu mới hoặc cập nhật dữ liệu mà không cần thêm giá trị vào trường này. Sau đó, trường sẽ được lưu với giá trị NULL.
- ❖ Giá trị NULL khác với giá trị 0 hoặc trường chứa khoảng trắng





# Hàm IFNULL(),COALESCE()

## ❖ Kiểm tra giá trị null là gì?

Không thể kiểm tra các giá trị NULL bằng các toán tử so sánh, chẳng hạn như =, <>, .....

## ❖ Thay vào đó sử dụng các toán tử: IS NULL và IS NOT NULL

## ❖ Cú pháp:

is null: **select** [colum] **from** name\_table **where** col\_name is null

is not null: **select** [colum] **from** name\_table **where** col\_name is not null



# Hàm IFNULL(),COALESCE()

❖ Hàm ifnull(a,b), coalsece(a,b): Trả về các giá trị thay thế b nếu a null

❖ Cú pháp:

**select** ifnull(col,value) **from** table\_name;

**select** coalsece(col,value) **from** table\_name

**ifull(a,b), coalsece(a,b):** Được dùng trong cả các lệnh insert hoặc update



# Toán tử trong SQL

- ❖ Toán tử toán học
- ❖ Toán tử so sánh
- ❖ Toán tử logic



# Toán tử số học

Được sử dụng trên các cột là số

Toán tử	Mô tả
+	$10+20=30$ : Cộng
-	$10-20=-10$ : Trừ
*	$10*20=200$ : Nhân
%	$15\% 10=5$ : Chia lấy dư

# Toán tử so sánh

Toán tử	Mô tả
=	'a'='b' So sánh bằng
<	10<20: So sánh nhỏ hơn
>	20>10: So sánh lớn hơn
>=	20>=10: So sánh lớn hơn hoặc bằng
<=	10<=10: So sánh nhỏ hơn hoặc bằng
<>	10<>20: So sánh khác

# Toán tử logic

Toán tử	Mô tả
and	True nếu tất cả điều kiện đều đúng
or	True nếu có 1 điều kiện đúng
in	True nếu tồn tại 1 giá trị trong tập
not in	True nếu không tồn tại giá trị trong tập
like	True nếu khớp với mẫu tìm kiếm
not	True nếu không tồn tại



# Bài tập thực hành

Với csdl mẫu là sakila:

**Bài 1:** Tìm tất cả các bộ phim sản xuất năm 2006 có rental\_rate lớn hơn 1

**Bài 2:** Tìm tất cả các nước mà tên có 10 ký tự

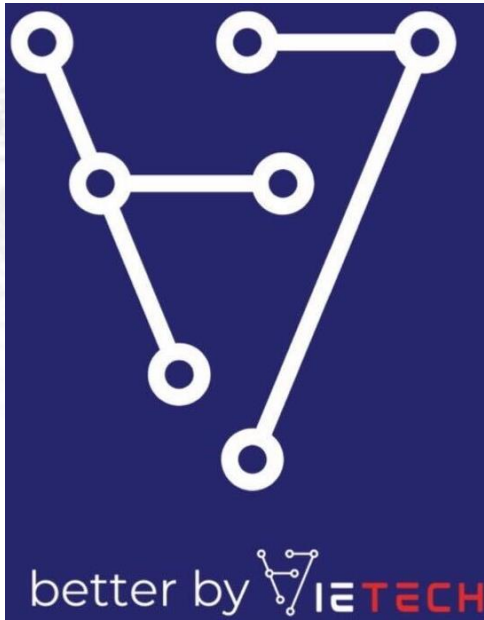
**Bài 3:** Tìm tất cả các nước mà tên có chứa chuỗi 'tv'

**Bài 4:** Tìm tất cả các địa chỉ có district='California' và address có chứa chuỗi 'La'

# Q&A







**THANK YOU**