



## **CƠ SỞ DỮ LIỆU**

### **BÀI 5: NGÔN NGỮ TRUY VẤN SQL – TRUY VẤN DỮ LIỆU TRÊN MỘT BẢNG PHẦN 1**

- Tìm hiểu câu lệnh truy vấn trên một bảng
- Mệnh đề **SELECT**
- Mệnh đề **FROM**
- Mệnh đề **WHERE**
- Mệnh đề **GROUP BY**
- Mệnh đề **HAVING**
- Mệnh đề **ORDER BY**



- Bao gồm các lệnh cho phép truy vấn dữ liệu mà không làm thay đổi dữ liệu hoặc các đối tượng trong CSDL
  - Là các truy vấn bắt đầu bằng từ khóa ***SELECT***
  - Trả về một bộ các thuộc tính hoặc một tập hợp các bộ thuộc tính

# CÚ PHÁP CÂU LỆNH SELECT

## ■ Cú pháp:

SELECT [DISTINCT] Column(s)

FROM TableName, Views

[WHERE Conditions]

[GROUP BY Row(s)]

[HAVING Conditions]

[ORDER BY Column(s)]

[asc|desc]

(Các mệnh đề trong cặp dấu ngoặc [] không bắt buộc)

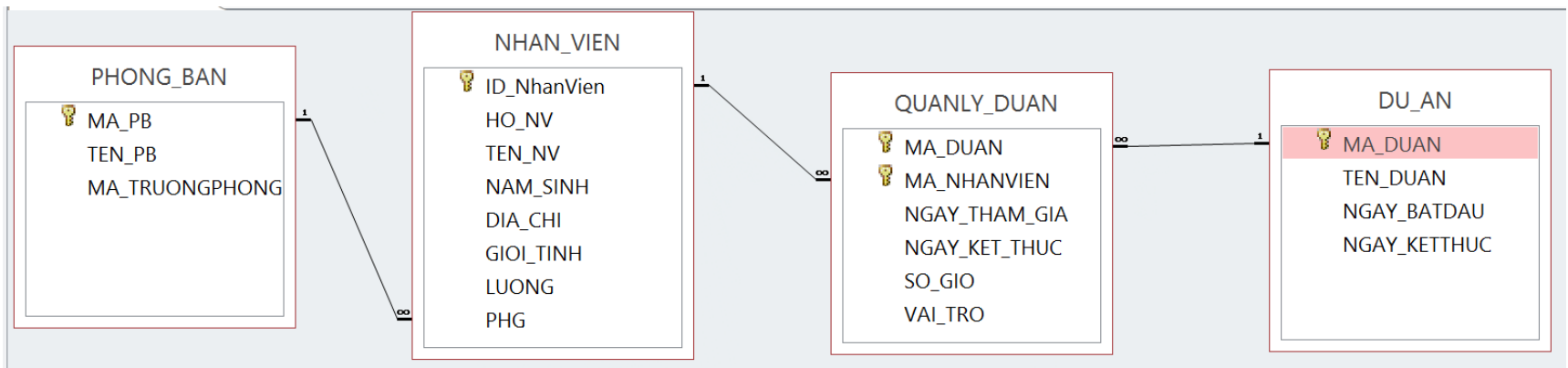
- DISTINCT có thể là các giá trị:

✓ All: trả về mọi bản ghi tìm thấy

✓ Distinct: trả về các bản ghi không trùng lặp

- Mệnh đề SELECT: các cột hiển thị
- Mệnh đề FROM: tên bảng
- Mệnh đề WHERE/HAVING cho phép truy vấn lựa chọn theo hàng
- Mệnh đề GROUP BY cho phép nhóm dữ liệu theo hàng
- Mệnh đề ORDER BY cho phép sắp xếp dữ liệu theo cột

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



- Truy vấn lựa chọn tất cả các hàng và cột

```
SELECT * FROM TableName
```

( \* đại diện tất cả các cột có trong table)

- Truy vấn lựa chọn một số cột

```
SELECT Column1, Column2 ...
```

```
FROM TableName
```

- Truy vấn lựa chọn tất cả các cột của bảng NHAN\_VIEN

```
SELECT * FROM nhan_vien ;
```

ID_NhanVien	HO_NV	TEN_NV	NAM_SINH	DIA_CHI	GIOI_TINH	LUONG	PHG
1	Nguyen	A	1/1/1987	15 Quang Trung Da Nang	<input checked="" type="checkbox"/>	\$1,000.00	PB001
2	Le	B	2/3/1982	16 Le Loi	<input type="checkbox"/>	\$800.00	PB002
3	Tran	C	5/6/1988	24 Tran Cao Van	<input checked="" type="checkbox"/>	\$1,200.00	PB001
(New)					<input type="checkbox"/>	\$0.00	

- Truy vấn lựa chọn cột họ, tên và lương của bảng NHAN\_VIEN

```
SELECT ho_nv, ten_nv, luong FROM nhan_vien ;
```

HO_NV	TEN_NV	LUONG
Nguyen	A	\$1,000.00
Le	B	\$800.00
Tran	C	\$1,200.00

- Viết câu truy vấn hiển thị tất cả các thông tin các phòng ban trong công ty

```
SELECT * FROM phong_ban ;
```

- Viết câu truy vấn hiển thị thông tin các cột: tên dự án, ngày bắt đầu và ngày kết thúc dự án trong bảng DU\_AN

```
SELECT Ten_duan, ngay_batdau, ngay_ketthuc  
FROM du_an ;
```



- Mệnh đề AS giúp thay đổi tiêu đề cột hiển thị trong tập kết quả

```
SELECT Column1 AS 'alias', ... FROM TableName
```

- Ví dụ:

```
SELECT ho_nv AS 'Ho Nhan Vien', ten_nv AS 'Ten  
Nhan Vien', luong  
FROM nhan_vien ;
```

'Ho Nhan Vien' ▼	'Ten Nhan Vien' ▼	LUONG ▼
Nguyen	A	\$1,000.00
Le	B	\$800.00
Tran	C	\$1,200.00
		\$0.00

- Ngoài danh sách các cột, chúng ta có thể chứa các biểu thức trong mệnh đề SELECT.
- Sử dụng hàm CONCAT() để nối chuỗi trong MySQL.
- Ví dụ: Hiển thị Họ và tên nhân viên

```
SELECT Concat(ho_nv, ' ', ten_nv) as 'Ho va Ten'  
FROM nhan_vien ;
```

'HO va TEN'
Nguyen A
Le B
Tran C

- Từ khóa **DISTINCT** loại bỏ các hàng trùng nhau trong tập kết quả
- Ví dụ: Table nhân viên có nhiều người họ Le, Nguyen, Tran

```
SELECT DISTINCT ho_nv  
FROM nhan_vien ;
```

HO_NV
Le
Nguyen
Tran

- Mệnh đề **TOP n** (dùng với SQL Server) dùng để hiển thị **N** hàng hoặc **N%** hàng đầu tiên trong bảng. MySQL thay bằng **LIMIT n** (*cuối truy vấn*)
- Ví dụ:
  - Hiển thị 5 hàng đầu tiên trong bảng nhân viên
 

```
SELECT TOP 5 * FROM nhan_vien
```

(SQL Server)

```
SELECT * FROM nhan_vien LIMIT 5
```

(MySQL)
  - Hiển thị 10% hàng đầu tiên trong bảng nhân viên
 

```
SELECT TOP 10 PERCENT * FROM nhan_vien
```

- Giúp loại bỏ các hàng không thỏa mãn điều kiện trong tập kết quả

- Cú pháp:

SELECT [DISTINCT] Column(s)

FROM TableName

[WHERE Conditions ]

- Biểu thức Condition:

Column/Expression    Operator    Value

- Một số toán tử (Operator) sử dụng trong biểu thức **Condition**:
  - Toán tử so sánh:  $>$  ,  $<$  ,  $>=$  ,  $<=$  ,  $=$  ,  $<>$
  - Toán tử logic: **AND**, **OR**, **NOT**
  - So sánh xâu gần đúng: **LIKE** (kết hợp ký tự đại diện)
  - Miền giá trị: **BETWEEN ... AND ...**
  - Trong tập hợp: **IN**
  - Giá trị NULL: **IS**

# SO SÁNH DÙNG TOÁN TỬ LIKE

Kí tự đại diện	Mô tả	Example
<b>_</b>	Đại diện cho 1 kí tự	SELECT * FROM nhan_vien WHERE ho_nv LIKE 'H_'
<b>%</b>	Đại diện cho một chuỗi kí tự có độ dài bất kì	SELECT * FROM nhan_vien WHERE ten_nv LIKE 'B%'
<b>[ ]</b>	Đại diện cho 1 kí tự đơn được liệt kê trong khoảng.	SELECT * FROM nhan_vien WHERE ten_nv LIKE '[HT]%'
<b>[^]</b>	Đại diện cho 1 kí tự đơn không được liệt kê trong khoảng.	SELECT * FROM nhan_vien WHERE ten_nv LIKE '^ [HT]%'

- Hiển thị họ, tên nhân viên có lương trên 800\$
- Hiển thị họ, tên nhân viên có lương nằm trong khoảng 800\$ đến 1000\$
- Hiển thị tất cả các thông tin dự án có ngày bắt đầu từ ngày 01/01/2017
- Hiển thị thông tin của các phòng ban có chứa chuỗi 'Sản xuất'
- Hiển thị thông tin họ, tên, lương của nhân viên có lương thấp hơn 800\$ và mã phòng ban là 'PB002'



- Hiển thị họ, tên nhân viên có lương trên 800\$

```
SELECT ho_nv, ten_nv FROM nhan_vien  
WHERE lương > 800 ;
```

- Hiển thị họ, tên nhân viên có lương nằm trong khoảng 800\$ đến 1000\$

```
SELECT ho_nv, ten_nv FROM nhan_vien  
WHERE lương >= 800 AND lương <= 1000 ;
```

Hoặc

```
SELECT ho_nv, ten_nv FROM nhan_vien  
WHERE lương BETWEEN 800 AND 1000 ;
```

- Hiển thị tất cả các thông tin dự án có ngày bắt đầu từ ngày 01/01/2017

```
SELECT * FROM du_an  
WHERE ngay_batdau >= '2017-01-01' ;
```

- Hiển thị thông tin của các phòng ban có chứa chuỗi 'Sản xuất'

```
SELECT * FROM phong_ban  
WHERE ten_pb LIKE '% Sản xuất %' ;
```

- Hiển thị thông tin họ, tên, lương của nhân viên có lương thấp hơn 800\$ và mã phòng ban là 'PB002'

```
SELECT ho_nv, ten_nv, luong  
FROM nhan_vien  
WHERE luong < 800 AND phg = 'PB002' ;
```



## **CƠ SỞ DỮ LIỆU**

### **BÀI 5: NGÔN NGỮ TRUY VẤN SQL – TRUY VẤN DỮ LIỆU TRÊN MỘT BẢNG PHẦN 2**

# CÁC HÀM TỔNG HỢP (AGGREGATE FUNCTIONS)

- Một số hàm nhóm như:
  - Max(column) - Tìm giá trị lớn nhất trong cột column
  - Min(column) - Tìm giá trị nhỏ nhất trong cột column
  - Avg(column) - Tìm giá trị trung bình của cột column
  - Sum(column) - Tìm giá trị tổng của cột column
  - Count – Hàm đếm số bộ
- Ví dụ hiển thị mức lương cao nhất trong bảng nhân viên  

```
SELECT Max(Luong) AS 'Lương cao nhất'  
FROM nhan_vien
```

- Hiển thị mức lương trung bình của các nhân viên thuộc mã phòng 'PB001'

```
SELECT Avg(luong) AS 'Lương trung bình'
FROM nhan_vien
WHERE phg='PB001' ;
```

- Hiển thị số lượng dự án có ngày kết thúc trước ngày 31/12/2016.

```
SELECT Count(*) AS 'Số lượng dự án'
FROM du_an
WHERE ngay_ket_thuc<'2016-12-31' ;
```

- Mệnh đề GROUP BY cho phép nhóm các hàng dữ liệu có giá trị giống nhau thành một nhóm
- Các tính toán (thường sử dụng các hàm tổng hợp) sẽ được tính trên mỗi nhóm.
- Ví dụ: Đếm số lượng nhân viên trong **mỗi phòng**

```
SELECT Count(*) AS 'So luong',  
       phg AS 'Ma Phong ban'  
FROM nhan_vien  
GROUP BY phg;
```

- Mệnh đề HAVING đi kèm với GROUP BY giúp loại bỏ các nhóm không thỏa mãn điều kiện
- Ví dụ: Cho biết lương cao nhất của mỗi phòng ban, chỉ hiển thị các phòng có lương cao nhất > 1000\$

```
SELECT Max(luong) AS 'Lương cao nhất ',  
       phg AS 'Mã phòng ban'  
FROM nhan_vien  
GROUP BY phg  
HAVING Max(luong) > 1000 ;
```



- Tính lương trung bình của từng Phòng Ban
- Đếm số lượng nhân viên của từng Phòng
- Tính tổng lương công ty phải trả cho mỗi phòng ban, chỉ hiển thị nhóm nào có tổng > 10000\$
- Tính tổng lương công ty phải trả cho mỗi phòng ban, chỉ nhóm hàng nào có cột lương > 700\$, chỉ hiển thị nhóm nào có tổng > 10000\$

- Tính lương trung bình của từng Phòng Ban
- Đếm số lượng nhân viên của từng Phòng
- Tính tổng lương công ty phải trả cho mỗi phòng ban, chỉ hiển thị nhóm nào có tổng > 10000\$
- Tính tổng lương công ty phải trả cho mỗi phòng ban, chỉ nhóm hàng nào có cột lương > 700\$, chỉ hiển thị nhóm nào có tổng > 10000\$

```
SELECT SUM(LUONG) AS TONGLUONG, PHG
FROM NHAN_VIEN
WHERE LUONG > 700
GROUP BY PHG
HAVING SUM(LUONG) > 10000;
```

- Cho phép sắp xếp kết quả truy vấn theo cột
- Có thể sắp xếp kết quả theo chiều:
  - Tăng dần (*asc*)
  - Giảm dần (*desc*)
- Thứ tự các cột sắp xếp ưu tiên từ trái sang phải.
- Ví dụ:

```
SELECT id_nhanvien AS 'Ma nhan vien ',  
       Concat(ho_nv, ' ', ten_nv) AS 'Ho va ten'  
       phg as 'Ma phong ban'  
FROM nhan_vien  
ORDER BY phg, id_nhanvien DESC ;
```

- Hiển thị danh sách các nhân viên có trong bảng NHAN\_VIEN theo thứ tự tăng dần của trường TEN\_NV
- Hiển thị tên dự án, ngày bắt đầu, ngày kết thúc của bảng DUAN theo thứ giảm dần của trường ngày kết thúc

- Viết câu truy vấn hiển thị mã phòng ban (PHG) và lương trung bình tương ứng từng phòng, chỉ lấy các hàng có mức lương trên 500\$, chỉ hiển thị các phòng có lương trung bình >700\$ và sắp xếp theo thứ tự giảm dần theo cột PHG

```
SELECT phg AS 'Ma pb', AVG(luong) AS Luongtb  
FROM nhan_vien  
WHERE luong > 500  
GROUP BY phg  
HAVING AVG(luong) > 700  
ORDER BY phg DESC;
```

- ❑ Ngôn ngữ truy vấn SQL được sử dụng để thao tác với hệ quản trị cơ sở dữ liệu
- ❑ Phân loại ngôn ngữ SQL: DML, DDL, DCL và DQL
- ❑ Ngôn ngữ truy vấn dữ liệu là các lệnh cho phép truy vấn dữ liệu mà không làm thay đổi dữ liệu hoặc các đối tượng trong CSDL



## ❑ Các mệnh đề trong câu truy vấn:

- ❖ SELECT
- ❖ TOP (trong SQL Server)
- ❖ DISTINCT
- ❖ FROM
- ❖ WHERE
- ❖ GROUP BY
- ❖ HAVING
- ❖ ORDER BY
- ❖ LIMIT



**Cảm ơn**