

Select data part 2

Mục tiêu bài học

- ❖ Lấy dữ liệu có giới hạn số lượng (Select Top)
- ❖ Các hàm tính gộp (Aggregate functions)
- ❖ Nhóm gộp dữ liệu (Group by)
- ❖ Truy vấn con (Sub-query)

Select TOP

- ❖ Từ khóa TOP được sử dụng để lấy ra một số lượng bản ghi đầu tiên từ tập kết quả.
- ❖ Cú pháp lấy ra với số lượng bản ghi xác định:

```
SELECT TOP <number>
```

- ❖ Ví dụ 1: Lấy ra 3 sinh viên đầu tiên trong bảng SINH_VIEN

```
SELECT TOP 3 *  
FROM SINH_VIEN
```

Select TOP

- ❖ Từ khóa TOP còn hỗ trợ lấy ra một số lượng bản ghi theo tỷ lệ phần trăm trên tổng số lượng từ tập kết quả
- ❖ Cú pháp lấy ra với số lượng bản ghi theo tỷ lệ xác định:

```
SELECT TOP <number> PERCENT
```

- ❖ Ví dụ 2: Lấy ra 10% sinh viên đầu tiên trong bảng SINH_VIEN

```
SELECT TOP 10 PERCENT *  
FROM SINH_VIEN
```

Tính toán các giá trị trong tập kết quả

- ❖ Câu lệnh **SELECT** có thể chứa các biểu thức toán học bằng cách áp dụng các toán tử cho một hoặc nhiều cột.
- ❖ Cho phép tạo ra các tập kết quả không tồn tại trong bảng, và được tính toán từ các cột giá trị được lưu trữ trong bảng.

Tính toán các giá trị trong tập kết quả

- ❖ Ví dụ: Thực hiện lấy ra danh sách họ và tên trong bảng SINH_VIEN

```
SELECT ho + ten AS 'Ho va ten'  
FROM SINH_VIEN
```

Truy vấn sử dụng MIN

❖ Lấy giá trị nhỏ nhất trong tập kết quả theo logic

➤ Số nhỏ nhất

➤ Ngày/giờ cũ nhất

➤ Ký tự đầu tiên theo alphabet A - Z

❖ Cú pháp:

```
SELECT MIN (column_name)  
FROM <table_name>
```

Truy vấn sử dụng MIN

- ❖ Ví dụ: Lấy ra giá bán rẻ nhất trong bảng SAN_PHAM

```
SELECT MIN (gia_ban)  
FROM SAN_PHAM
```

- ❖ Kết quả từ câu lệnh truy vấn với MIN chỉ trả về 1 giá trị

Truy vấn sử dụng MAX

❖ Lấy giá trị lớn nhất trong tập kết quả theo logic

➤ Số lớn nhất

➤ Ngày/giờ mới nhất

➤ Ký tự đầu tiên theo alphabet Z - A

❖ Cú pháp:

```
SELECT MAX (column_name)  
FROM <table_name>
```

Truy vấn sử dụng MIN

- ❖ Ví dụ: Lấy ra giá bán cao nhất trong bảng SAN_PHAM

```
SELECT MAX (gia_ban)  
FROM SAN_PHAM
```

- ❖ Tương tự như MIN, kết quả từ câu lệnh truy vấn với MAX chỉ trả về 1 giá trị

Truy vấn sử dụng SUM

❖ Tính tổng của tất cả các giá trị số khác NULL trong một cột.

❖ Cú pháp:

```
SELECT SUM(column_name)  
FROM <table_name>
```

❖ Ví dụ: Tính tổng số lượng mua của khách hàng có mã 'KH01' trong bảng CHI_TIET

```
SELECT SUM(SoLuong) FROM CHI_TIET  
WHERE MaKH = 'KH01'
```

Truy vấn sử dụng AVG

- ❖ Tính giá trị trung bình của tất cả các giá trị số khác NULL trong một cột
- ❖ Cú pháp:

```
SELECT AVG(column_name)  
FROM <table_name>
```
- ❖ Ví dụ: Tính trung bình số lượng bán của sản phẩm có mã 'SP01' trong bảng CHI_TIET

```
SELECT AVG(SoLuong) FROM CHI_TIET  
WHERE MaSP= 'SP01'
```

Truy vấn sử dụng COUNT(*)

❖ Đếm số lượng record trong tập kết quả

❖ Cú pháp:

```
SELECT COUNT(*)  
FROM <table_name>
```

❖ Ví dụ: Cho biết có bao nhiêu khách hàng địa chỉ ở HCM

```
SELECT COUNT(*) FROM KHACH_HANG  
WHERE DiaChi = 'HCM'
```

Truy vấn sử dụng COUNT(tên cột)

❖ Đếm và trả về số lượng các giá trị khác NULL trong cột

❖ Cú pháp:

```
SELECT COUNT(<tên cột>)  
FROM <tên bảng>
```

❖ Ví dụ: Cho biết có bao nhiêu khách hàng có thông tin địa chỉ

```
SELECT COUNT(DiaChi)  
FROM KHACH_HANG
```

MỆNH ĐỀ GROUP BY

- ❖ **GROUP BY** sử dụng để phân vùng kết quả theo một hoặc nhiều cột dữ liệu được nhóm lại với nhau, mỗi nhóm tạo ra một tập giá trị duy nhất

id	fruit
1	Apple
2	Orange
3	Apple
4	Banana
5	Orange



fruit
Apple
Banana
Orange

MỆNH ĐỀ GROUP BY

❖ Cú pháp:

```
SELECT <tên cột>
```

```
FROM <tên table>
```

```
GROUP BY <tên cột 1>,... <tên cột n>
```


MỆNH ĐỀ GROUP BY

- ❖ Danh sách các cột theo sau từ khóa **GROUP BY** được gọi là cột được nhóm.
- ❖ Mỗi nhóm có thể kết hợp từ nhiều cột
- ❖ Đối với mỗi nhóm chỉ có duy nhất 1 bản ghi.

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT country, **COUNT(*)** AS number
FROM Customers
GROUP BY country;

country	number
UAE	1
UK	2
USA	2

GROUP BY Với WHERE

❖ Cú pháp:

```
SELECT <tên cột> ..n
```

```
FROM <tên table>
```

```
WHERE <biểu thức điều kiện> ..n
```

```
GROUP BY <tên cột 1>,... <tên cột n>
```

GROUP BY Với WHERE

- ❖ Mệnh đề **WHERE** cũng có thể sử dụng với mệnh đề **GROUP BY** để hạn chế về các hàng gộp nhóm.
- ❖ Các record đáp ứng điều kiện sẽ được đưa vào xử lý gộp nhóm
- ❖ Các record không thỏa mãn điều kiện sẽ bị loại bỏ trước khi gộp nhóm.

GROUP BY Với WHERE

Ví dụ: Đếm số lượng sinh viên nam và sinh nữ có quê ở HCM

```
SELECT giới_tinh, COUNT (ma_sv) AS 'Số lượng sinh viên'  
FROM SINH_VIEN  
WHERE que_quan = 'HCM'  
GROUP BY giới_tinh
```

GROUP BY Với HAVING

❖ Cú pháp:

SELECT <tên cột>, ..n

FROM <tên table>

WHERE <biểu thức điều kiện>, ..n

GROUP BY <tên cột>, ..n

HAVING <biểu thức điều kiện>, ..n

GROUP BY Với HAVING

- ❖ Mệnh đề **HAVING** chỉ được sử dụng để chỉ ra một điều kiện áp dụng cho các nhóm gộp
- ❖ Hoạt động như mệnh đề **WHERE**, nhưng là bộ lọc trên các nhóm, sau khi đã được gộp nhóm bởi **GROUP BY**.
- ❖ Thường sử dụng với các hàm tính gộp.

GROUP BY Với HAVING

Ví dụ: Cho biết những mã khoa có nhiều hơn 3 chuyên ngành trong bảng Chuyên ngành

```
SELECT ma_khoa  
FROM CHUYEN_NGANH  
GROUP BY ma_khoa  
HAVING count(ma_khoa) > 3
```


Truy vấn nhiều bảng

- ❖ Vì không thể lưu toàn bộ dữ liệu trong một bảng đơn. Chúng ta thường cần phải kết hợp và lấy dữ liệu từ nhiều bảng khác nhau.
- ❖ Có các cách kết hợp dữ liệu từ nhiều bảng:
 - **SUB-QUERY**: Đặt một truy vấn bên trong một truy vấn khác
 - **UNION**: Kết hợp các hàng dữ liệu từ nhiều bảng dữ liệu
 - **JOIN**: Phép nối bảng, kết hợp các cột dữ liệu từ nhiều bảng dữ liệu

Subquery

❖ Cú pháp:

```
SELECT <tên cột>
```

```
FROM <tên bảng>
```

```
WHERE <tên cột> [toán tử]
```

```
(
```

```
    SELECT <tên cột> FROM <tên bảng>
```

```
    WHERE <Biểu thức điều kiện>
```

```
)
```

Subquery

- ❖ Truy vấn con (Subquery) là truy vấn được lồng bên trong một truy vấn khác. Truy vấn ngoài được gọi là truy vấn cha (Parent query), truy vấn bên trong là truy vấn con.
- ❖ Mục đích của một truy vấn con là để trả về các kết quả cho truy vấn bên ngoài.
- ❖ Có thể có nhiều truy vấn con được viết lồng nhau.

Subquery

- ❖ Ví dụ: Cho biết các kết quả có điểm thi cuối kỳ môn SQL cao nhất trong bảng Kết quả.

```
SELECT *  
FROM KET_QUA  
WHERE ma_mon = 'SQL'  
AND cuoi_ky = ( SELECT MAX(cuoi_ky)  
                FROM KET_QUA  
                WHERE ma_mon = 'SQL')
```

Subquery với EXISTS

❖ Cú pháp:

```
SELECT <tên cột>  
FROM <tên bảng>  
WHERE EXISTS  
(  
    SELECT <tên cột> FROM <tên bảng>  
    WHERE <Biểu thức điều kiện>  
)
```

Subquery với EXISTS

- ❖ Toán tử **EXISTS** được sử dụng để kiểm tra sự tồn tại của bất kỳ bản ghi nào trong truy vấn con.
- ❖ Toán tử **EXISTS** trả về **TRUE** nếu truy vấn con trả về một hoặc nhiều bản ghi.

Subquery với EXISTS

- ❖ Ví dụ: Cho biết tên các sinh viên đã từng thi với kết quả trên 8 điểm

```
SELECT TenSV
FROM SINH_VIEN
WHERE EXISTS (
    SELECT MaSV FROM KET_QUA
    WHERE SINH_VIEN.MaSV = KET_QUA.MaSV
    AND DiemThi > 8 )
```

Subquery với ANY, ALL

- ❖ Toán tử **ANY** và **ALL** cho phép bạn thực hiện so sánh giữa một giá trị trong cột với một loạt các giá trị khác
- ❖ Toán tử **ANY** trả về TRUE nếu BẤT KỲ giá trị nào của truy vấn con đáp ứng điều kiện
- ❖ Toán tử **ALL** trả về TRUE nếu TẤT CẢ các giá trị truy vấn con đáp ứng điều kiện. Được sử dụng với các câu lệnh SELECT, WHERE và HAVING

Subquery với ANY

❖ Cú pháp:

```
SELECT <tên cột>  
FROM <tên bảng>  
WHERE <tên cột> <toán tử> ANY  
( SELECT <tên cột> FROM <tên bảng>  
  WHERE <Biểu thức điều kiện>)
```

❖ Toán tử có thể là: =, <>, !=, >, >=, <, <=

Subquery với ANY

- ❖ Ví dụ: Liệt kê tên các sản phẩm nếu tìm thấy BẤT KỲ bản ghi nào trong bảng chi tiết mua hàng có số lượng lớn hơn 10

```
SELECT TenSP
FROM SAN_PHAM
WHERE MaSP = ANY
( SELECT MaSP FROM CHI_TIET
  WHERE SoLuong > 10)
```

Subquery với ALL

❖ Cú pháp:

```
SELECT <tên cột>  
FROM <tên bảng>  
WHERE <tên cột> <toán tử> ALL  
( SELECT <tên cột> FROM <tên bảng>  
  WHERE <Biểu thức điều kiện>)
```

❖ Toán tử có thể là: =, <>, !=, >, >=, <, <=

Subquery với ALL

- ❖ Ví dụ: Liệt kê tên các sản phẩm nếu TẤT CẢ bản ghi trong bảng chi tiết mua hàng có số lượng lớn hơn 10

```
SELECT TenSP
FROM SAN_PHAM
WHERE MaSP = ALL
( SELECT MaSP FROM CHI_TIET
  WHERE SoLuong > 10)
```

Tổng kết

- ❖ Select Top
- ❖ Min, Max, Count, Avg, Sum
- ❖ Group by
- ❖ Having
- ❖ Subquery, Exists, Any, All

Tổng kết: Các hàm tính gộp

Tên hàm	Mô tả
SUM	Tính tổng của tất cả các giá trị số không phải là NULL trong một cột.
AVG	Tính giá trị trung bình của tất cả các giá trị số không phải là NULL trong một cột
COUNT	Đếm và trả về số lượng các giá trị khác NULL trong biểu thức. Hỗ trợ cho cả các cột kiểu số và kiểu ký tự. Có thể sử dụng COUNT (*) để trên tất cả các hàng (bản ghi) thay vì đếm trên cột xác định.
MAX	Trả về giá trị lớn nhất (số lớn nhất, ngày/giờ mới nhất, chuỗi ký tự thứ tự alphabet cuối cùng)
MIN	Trả về giá trị nhỏ nhất (số nhỏ nhất, ngày/giờ cũ nhất, chuỗi ký tự thứ tự alphabet đầu tiên)

