



PHẦN I: TỔ CHỨC CƠ SỞ DỮ LIỆU

CHƯƠNG IV : QUẢN LÝ VÀ THAO TÁC DỮ LIỆU

Phù Khắc Anh



NGÔN NGỮ TRUY VẤN SQL

A decorative blue wavy line with a gradient, flowing from the top left towards the bottom left of the slide.

CÁC TOÁN TỬ NHÓM



MỘT SỐ KỸ THUẬT TRUY VẤN



ÔN TẬP KIỂM TRA



I. NGÔN NGỮ TRUY VẤN SQL

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu
- Truy vấn thao tác cơ sở dữ liệu
- Truy vấn điều khiển cơ sở dữ liệu

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu

❖ Là Truy vấn mô tả

- Lược đồ cho mỗi quan hệ
- Miền giá trị tương ứng của từng thuộc tính
- Ràng buộc toàn vẹn
- Chỉ mục trên mỗi quan hệ

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu :

Các câu lệnh về định nghĩa dữ liệu đã học ở các chương trước

- Các câu lệnh về tạo, sửa, xóa database
- Các câu lệnh về tạo, sửa, xóa table
- Các câu lệnh về tạo, sửa, xóa ràng buộc toàn vẹn (constraint)
-

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu

Kiểu dữ liệu :

Dữ liệu gồm 2 loại:

- System-supplied Datatype: các dữ liệu cơ bản của SQL server
- User-defined data types: Kiểu dữ liệu do người dùng định nghĩa

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu
- Các kiểu dữ liệu cơ bản của SQL Server

- **Integers (số nguyên):**

- ✓ Bigint: khoảng giá trị từ -2^{63} đến $2^{63} - 1$ tức từ -9.223.372.036.854.775.808 đến 9.223.372.036.854.775,807.
- ✓ Int: khoảng giá trị từ -2^{31} đến $2^{31} - 1$ tức từ -2.147.483.648 đến 2.147.483.647.
- ✓ Smallint: khoảng giá trị từ -2^{15} đến 2^{15} tức từ -32.768 đến 32.767.
- ✓ Tinyint: khoảng giá trị từ 0 đến 255.
- ✓ Bit: chứa giá trị 0 hoặc 1.

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu

Các kiểu dữ liệu cơ bản của SQL Server

- **Decimal và numeric (số thập phân):**

✓ Decimal: có giá trị từ $-10^{38} + 1$ đến $10^{38} - 1$.

✓ Numeric: tương đương với decimal.

- **Money và smallmoney (kiểu tiền tệ)**

✓ Money: khoảng giá trị từ -2^{63} đến 2^{63} tức từ -

922.337.203.685.477,5808 922.337.203.685.477,5807.

✓ Smallmoney: khoảng giá trị từ -214.748,3648 đến 214.748,3647

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu:

Các kiểu dữ liệu cơ bản của SQL Server

- **Approximate Numerics**

- ✓ Float: độ chính xác từ $1.79E + 308$ đến $1.79E + 308$.
- ✓ Real: độ chính xác từ $-3.40E + 38$ đến $3.40E + 38$.

- **Datetime and smalldatetime:**

- ✓ datetime: khoảng giá trị từ 1/1/1753 đến 31/12/9999.
- ✓ smalldatetime: khoảng giá trị từ 1/1/1900 đến 6/6/2079.

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu

Các kiểu dữ liệu cơ bản của SQL Server

- **Character Strings (kiểu chuỗi):**

- ✓ Char: không chứa Unicode, chiều dài tối đa là 8.000 ký tự.
- ✓ Varchar: không chứa Unicode, chiều dài tối đa là 8.000 ký tự.
- ✓ Text: không chứa Unicode, chiều dài tối đa là $2^{31} - 1$ tương đương với 2.147.483.647 ký tự

- **Unicode Character Strings (kiểu chuỗi có chứa Unicode):**

- ✓ nchar: có chứa Unicode, chiều dài tối đa là 4.000 ký tự
- ✓ nvarchar: có chứa Unicode, chiều dài tối đa là 4.000 ký tự
- ✓ ntext: có chứa Unicode, chiều dài tối đa là $2^{30} - 1$ tức 1.073.741.823 ký tự.

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn định nghĩa cơ sở dữ liệu

Một số lưu ý khi tạo bảng :

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - 1. Cập nhật dữ liệu
 - * Câu lệnh Insert
 - * Câu lệnh Delete
 - * Câu lệnh Update
 - 2. Truy vấn dữ liệu
 - * Câu lệnh Select đơn giản
 - * Truy vấn lồng, gom nhóm

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Insert
- **Chèn từng dòng dữ liệu**

INSERT INTO tên_bảng[(cột 1, cột 2, ...)]

VALUES (giatri1, giatri2, ...)

- **Ví dụ: chèn thông tin Khoa(makh, tenkh, dienthoai)**

Với thông tin sau : 'DHT10', 'Khoa Luật', '054821135')

INSERT INTO khoa VALUES ('DHT10', 'Khoa Luật', '054821135')

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Insert

- **Chèn từng dòng dữ liệu**

INSERT INTO tên_bảng[(cột 1, cột 2, ...)]

VALUES (giatri1, giatri2, ...)

- **Ví dụ: chèn thông tin Khoa(makh, tenkh, dienthoai)**

Với thông tin sau : 'DHT10', 'Khoa Luật', '054821135')

INSERT INTO khoa VALUES ('DHT10', 'Khoa Luật', '054821135')

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Insert
- **Ví dụ 2: Chèn thông tin bảng SINHVIEN**

INSERT INTO Sinhvien(masv, hodem, ten, gioitinh, malop) **VALUES** ('0241020008', N'Nguyễn Công', N'Chính', 1, 'C24102')

Cách khác :

INSERT INTO sinhvien **VALUES** ('0241020008', N'Nguyễn Công', N'Chính', NULL, 1, NULL, 'C24102')

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Insert

- **Chèn nhiều dòng dữ liệu**

INSERT INTO tên_bảng[(danh_sách_cột)]
câu_lệnh_SELECT

- Lưu ý:
 - lưu lại tất cả các thông tin của câu truy vấn
 - Trong câu lệnh SELECT được sử dụng mệnh đề **COMPUTE ... BY**

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

UPDATE tên_bảng

SET tên_cột 1 = biểu_thức | giá trị 1 [, tên_cột
2 = biểu_thức | giá trị 2]

[**FROM** danh_sách_bảng]

[**WHERE** điều_kiện]

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : Cập nhật lại số đơn vị học trình của các môn học có số đơn vị học trình nhỏ hơn hoặc bằng 2

UPDATE monhoc

SET sodvht = 3

WHERE sodvht <= 2

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

- **Hàm Case :**

```
Bí_danh = CASE Tên_cột  
    WHEN Giá_trị1 then Kết_quả1  
    WHEN Giá_trị2 then Kết_quả2  
    ...  
    ELSE Hiện_thị  
END
```

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : Sử dụng cấu trúc CASE trong câu lệnh UPDATE bổ sung tiền phòng trong bảng nhatkypdong

**Trước khi cập nhật bảng
NHATKY PHONG**

| SOPHONG | LOAIPHONG | SONGAY | TIENPHONG |
|---------|-----------|--------|-----------|
| 101 | A | 5 | |
| 202 | B | 5 | |
| 101 | A | 2 | |
| 102 | C | 3 | |

**Sau khi cập nhật bảng
NHATKY PHONG**

| SOPHONG | LOAIPHONG | SONGAY | TIENPHONG |
|---------|-----------|--------|-----------|
| 101 | A | 5 | 500 |
| 202 | B | 5 | 350 |
| 101 | A | 2 | 200 |
| 102 | C | 3 | 150 |

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : Sử dụng cấu trúc CASE trong câu lệnh UPDATE bổ sung tiền phòng trong bảng nhatkypdong

UPDATE nhatkypdong

SET tienphong=songay* **dongia**

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : Sử dụng cấu trúc CASE trong câu lệnh UPDATE bổ sung tiền phòng trong bảng nhatkypdong

UPDATE nhatkypdong

SET tienphong=songay*

I. NGÔN NGỮ TRUY VẤN SQL

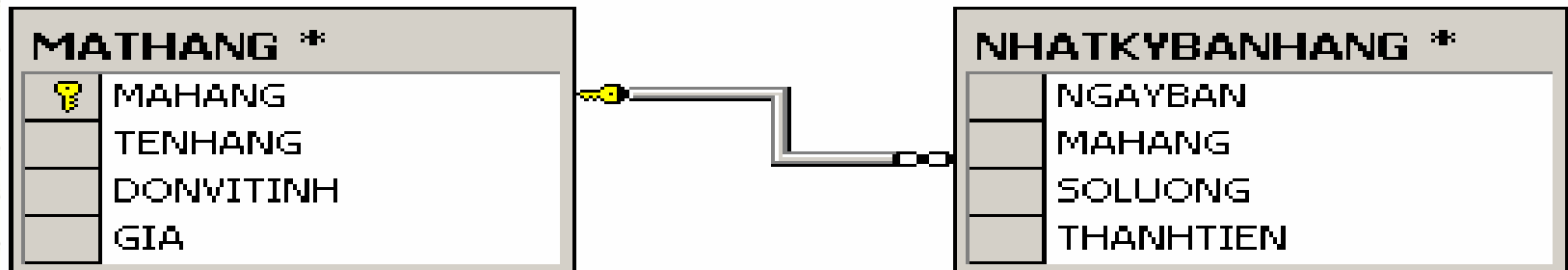
- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : cập nhật giá trị trường

THANHTIEN của bảng

NHATKYBANHANG theo công thức

THANHTIEN = SOLUONG × GIA



I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Update

Ví dụ : cập nhật giá trị trường

THANHTIEN của bảng

NHATKYBANHANG theo công thức

THANHTIEN = SOLUONG × GIA

UPDATE nhatkybanhang **SET** thanhtien =
soluong*gia **FROM** mathang **WHERE**
nhatkybanhang.mahang = mathang.mahang

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Delete

DELETE FROM tên_bảng
[**FROM** danh_sách_bảng]
[**WHERE** điều_kiện]

chỉ định điều kiện đối
với các dòng dữ liệu
cần xóa

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Delete
- Ví dụ : xoá bảng SINHVIEN có nơi sinh ở Huế

DELETE FROM sinhvien **WHERE** noisinh
LIKE '%Huế%'

- Ví dụ : xoá bảng SINHVIEN thuộc lớp 'Tin K24'

DELETE FROM sinhvien **FROM** lop **WHERE**
lop.malop=sinhvien.malop **AND** tenlop='Tin K24'

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Delete
- Ví dụ 9: xoá những lớp không có sinh viên học

DELETE FROM lop **WHERE** malop **NOT IN**
(**SELECT DISTINCT**
malop **FROM** sinhvien)

Sử dụng truy vấn con trong câu lệnh DELETE để làm điều kiện cho câu lệnh

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Câu lệnh Delete

Xoá toàn bộ dữ liệu trong bảng

TRUNCATE TABLE tên_bảng

- Ví dụ : xoá toàn bộ dữ liệu trong bảng
diemthi:

DELETE FROM diemthi

Hay TRUNCATE TABLE diemthi

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu :
 - Cập nhật dữ liệu : Một số lưu ý

I. NGÔN NGỮ TRUY VẤN SQL

- Truy vấn thao tác dữ liệu : 2. Câu lệnh Select

Cú pháp tổng quát

SELECT

[**ALL** | **DISTINCT**][**TOP n**] danh_sách_chọn

[**INTO** tên_bảng_mới]

FROM danh_sách_bảng/khung_nhìn

[**WHERE** điều_kiện]

[**GROUP BY** danh_sách_cột]

[**HAVING** điều_kiện]

[**ORDER BY** cột_sắp_xếp]

[**COMPUTE** danh_sách_hàm_gộp [**BY** danh_sách_cột]]

Phải tuân
theo đúng thứ
tự như trong
cú pháp

I. NGÔN NGỮ TRUY VẤN SQL

- Ví dụ 11: Kết quả của câu lệnh sau đây cho biết mã lớp, tên lớp và hệ đào tạo của các lớp hiện có

SELECT malop,tenlop,hedaotao
FROM lop

| MALOP | TENLOP | HEDAOTAO |
|--------|----------|-----------|
| C24101 | Toán K24 | Chính quy |
| C24102 | Tin K24 | Chính quy |
| C24103 | Lý K24 | Chính quy |
| C24301 | Sinh K24 | Chính quy |
| C25101 | Toán K25 | Chính quy |
| C25102 | Tin K25 | Chính quy |
| C25103 | Lý K25 | Chính quy |
| C25301 | Sinh K25 | Chính quy |
| C26101 | Toán K26 | Chính quy |
| C26102 | Tin K26 | Chính quy |

I. NGÔN NGỮ TRUY VẤN SQL

- 2.1. Truy vấn không điều kiện
- 2.2. Truy vấn có điều kiện
- 2.3. Tạo mới bảng dữ liệu từ kết quả của câu lệnh
SELECT
- 2.4. Sắp xếp kết quả truy vấn
- 2.5. Phép hợp
- 2.6. Phép nối
- 2.7. Thống kê dữ liệu với GROUP BY
- 2.8. Truy vấn lồng
- 2.9. Thống kê dữ liệu với COMPUTE

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

```
SELECT *|DISTINCT| TOP| DS THUOC TINH  
FROM TÊN_BẢNG
```

**Ví dụ 12: Liệt kê
danh sách KHOA**

```
SELECT *  
FROM khoa
```

| MAKHOA | TENKHOA | DIENTHOAI |
|--------|--------------------------|-----------|
| DHT01 | Khoa Toán cơ - Tin học | 054822407 |
| DHT02 | Khoa Công nghệ thông tin | 054826767 |
| DHT03 | Khoa Vật lý | 054823462 |
| DHT04 | Khoa Hoá học | 054823951 |
| DHT05 | Khoa Sinh học | 054822934 |
| DHT06 | Khoa Địa lý - Địa chất | 054823837 |
| DHT07 | Khoa Ngữ văn | 054821133 |
| DHT08 | Khoa Lịch sử | 054823833 |
| DHT09 | Khoa Mác - Lê Nin | 054825698 |
| DHT10 | Khoa Luật | 054821135 |

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Liệt kê chi tiết thuộc tính

Ví dụ 13: *cho biết mã lớp, tên lớp, năm nhập học và khoá của các lớp*

SELECT malop,tenlop,
namnhaphoc, khoa
FROM lop

| MALOP | TENLOP | NAMNHAPHOC | KHOA |
|--------|----------|------------|------|
| C24101 | Toán K24 | 2000 | 24 |
| C24102 | Tin K24 | 2000 | 24 |
| C24103 | Lý K24 | 2000 | 24 |
| C24301 | Sinh K24 | 2000 | 24 |
| C25101 | Toán K25 | 2001 | 25 |
| C25102 | Tin K25 | 2001 | 25 |
| C25103 | Lý K25 | 2001 | 25 |
| C25301 | Sinh K25 | 2001 | 25 |
| C26101 | Toán K26 | 2002 | 26 |
| C26102 | Tin K26 | 2002 | 26 |

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Đặt tên bí danh

- tiêu_đề_cột = tên_trường
- tên_trường **AS** tiêu_đề_cột
- tên_trường tiêu_đề_cột

Ví dụ 14: cho biết mã lớp, tên lớp và khoá học của các lớp trong trường

SELECT 'Mã lớp'= malop,
tenlop 'Tên lớp' , khoa **AS**
'Khoá'
FROM lop

| Mã lớp | Tên Lớp | Khoá |
|--------|----------|------|
| C24101 | Toán K24 | 24 |
| C24102 | Tin K24 | 24 |
| C24103 | Lý K24 | 24 |
| C24301 | Sinh K24 | 24 |
| C25101 | Toán K25 | 25 |
| C25102 | Tin K25 | 25 |
| C25103 | Lý K25 | 25 |
| C25301 | Sinh K25 | 25 |
| C26101 | Toán K26 | 26 |
| C26102 | Tin K26 | 26 |

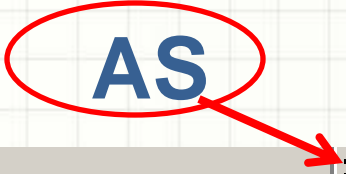
I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Mỗi một biểu thức trở thành một cột trong kết quả truy vấn.

Ví dụ 15: tên và số tiết của các môn học

SELECT tenmonhoc, sodvht*15 **AS**
sotiet
FROM monhoc



| TENMONHOC | SOTIET |
|-------------------------|--------|
| Hoá đại cương | 45 |
| Tin học đại cương | 60 |
| Ngôn ngữ C | 75 |
| Lý thuyết hệ điều hành | 60 |
| Cấu trúc dữ liệu và ... | 60 |
| Đại số tuyến tính | 60 |
| Giải tích 1 | 60 |
| Bài tập Đại số | 30 |
| Bài tập Giải tích 1 | 30 |
| Vật lý đại cương | 45 |

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Hiển thị giá trị trong câu Select

Ví dụ 16:

SELECT tenmonhoc, 'Số tiết: ', sodvht*15 **AS**
sotiet

FROM monho

| TENMONHOC | (No column name) | SOTIET |
|--------------------------------|------------------|--------|
| Hoá đại cương | Số tiết: | 45 |
| Tin học đại cương | Số tiết: | 60 |
| Ngôn ngữ C | Số tiết: | 75 |
| Lý thuyết hệ điều hành | Số tiết: | 60 |
| Cấu trúc dữ liệu và giải thuật | Số tiết: | 60 |
| Đại số tuyến tính | Số tiết: | 60 |
| Giải tích 1 | Số tiết: | 60 |
| Bài tập Đại số | Số tiết: | 30 |
| Bài tập Giải tích 1 | Số tiết: | 30 |
| Vật lý đại cương | Số tiết: | 45 |

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Loại bỏ các dòng dữ liệu trùng nhau trong kết quả truy vấn dùng **DISTINCT**

VD 17: liệt kê danh sách khoa trong LOP

SELECT khoa **FROM** lop và:

SELECT DISTINCT khoa **FROM** lop

| KHOA | KHOA |
|------|------|
| 24 | 24 |
| 24 | 25 |
| 24 | 26 |
| 24 | |
| 25 | |
| 25 | |
| 25 | |
| 25 | |
| 26 | |
| 26 | |

I. NGÔN NGỮ TRUY VẤN SQL

2.1. Truy vấn không có điều kiện

Giới hạn số lượng dòng trong kết quả truy vấn

Ví dụ 18: hiển thị họ tên và ngày sinh của 5 sinh viên đầu tiên trong danh sách

```
SELECT TOP 5 hodem,ten,ngaysinh  
FROM sinhvien
```

Ví dụ 19: hiển thị họ tên và ngày sinh của 10% số lượng sinh viên hiện có trong bảng SINHVIÊN

```
SELECT TOP 10 PERCENT  
hodem,ten,ngaysinh FROM sinhvien
```

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

- ❖ Xác định các điều kiện đối với việc truy xuất dữ liệu.

WHERE một biểu thức logic

Ví dụ 20: danh sách các môn học có số đơn vị học trình lớn hơn 3

SELECT *
FROM monhoc
WHERE sodvht>3

| MA MONHOC | TEN MONHOC | SODVHT |
|-----------|--------------------------------|--------|
| TI-001 | Tin học đại cương | 4 |
| TI-002 | Ngôn ngữ C | 5 |
| TI-003 | Lý thuyết hệ điều hành | 4 |
| TI-004 | Cấu trúc dữ liệu và giải thuật | 4 |
| TO-001 | Đại số tuyến tính | 4 |
| TO-002 | Giải tích 1 | 4 |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

Trong mệnh đề **WHERE** thường sử dụng:

- a. Các toán tử so sánh
- b. Kiểm tra giới hạn của dữ liệu (**BETWEEN/ NOT BETWEEN**)
- c. Danh sách **IN/NOT IN**
- d. Các toán tử kết hợp điều kiện (**AND, OR**)
- e. Kiểm tra khuôn dạng dữ liệu.
- f. Các giá trị **NULL**

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

a. Toán tử so sánh

| Toán tử | ý nghĩa |
|---------|-------------------|
| = | Bằng |
| > | Lớn hơn |
| < | Nhỏ hơn |
| >= | Lớn hơn hoặc bằng |
| <= | Nhỏ hơn hoặc bằng |
| <> | Khác |
| !> | Không lớn hơn |
| !< | Không nhỏ hơn |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

a. Toán tử so sánh

Ví dụ 21 : Cho biết *mã, họ tên và ngày sinh* của các sinh viên có *tên là Anh* và có *tuổi nhỏ hơn hoặc bằng 20*.

SELECT masv,hodem,ten,ngaysinh **FROM** sinhvien **WHERE** (ten='Anh') **AND** (YEAR(GETDATE()) - YEAR(ngaysinh) <=20)

| MASV | HODEM | TEN | NGAYSINH |
|------------|-----------------|-----|---------------------|
| 0261010001 | Lê Hoàng Phương | Anh | 1984-03-04 00:00:00 |
| 0261010002 | Lê Thị Vân | Anh | 1984-10-14 00:00:00 |
| 0261020002 | Lê Thúc Quốc | Anh | 1984-12-04 00:00:00 |
| 0261020004 | Nguyễn Thị Lan | Anh | 1984-08-23 00:00:00 |
| 0261020005 | Nguyễn Thị Lan | Anh | 1984-07-25 00:00:00 |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

b. Kiểm tra giới hạn của dữ liệu

| Cách sử dụng | Ý nghĩa |
|---|---|
| Thuộc_Tính BETWEEN a AND b | Thuộc_tính \geq a AND Thuộc_tính \leq b |
| Thuộc_Tính NOT BETWEEN a AND b | Thuộc_tính $<$ a OR Thuộc_tính $>$ b |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

b. Kiểm tra giới hạn của dữ liệu

Ví dụ 22: Họ tên và tuổi của các sinh viên có tên là Bình và có tuổi nằm trong khoảng từ 20 đến 22

```
SELECT hodem,ten,year(getdate())-  
year(ngaysinh) AS tuoi
```

```
FROM sinhvien
```

```
WHERE ten='Bình' AND
```

```
year(getdate())-year(ngaysinh) BETWEEN 20  
AND 22
```

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

c. danh sách các giá trị (IN hoặc NOT IN)

Ví dụ 23: Danh sách các môn học có số đơn vị học trình là 2, 4 hoặc 5

Dùng toán tử luận lý (OR)

```
SELECT * FROM monhoc
```

```
WHERE sodvht=2 OR sodvht=4 OR sodvht=5
```

Dùng IN

```
SELECT * FROM monhoc
```

```
WHERE sodvht IN (2,4,5)
```


I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

d. Toán tử LIKE (NOT LIKE)

Chúng thường được kết hợp với các ký tự đại diện sau đây:

| Ký tự đại diện | ý nghĩa |
|----------------|--|
| % | Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự |
| _ | Ký tự đơn bất kỳ |
| [] | Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f]) hay một tập (ví dụ [abcdef]) |
| [^] | Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định (ví dụ [^a-f] hay một tập (ví dụ [^abcdef])). |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

d. Toán tử LIKE (NOT LIKE)

Ví dụ 24: Liệt kê danh sách sinh viên có họ đệm chứa từ 'Lê'

SELECT hodem,ten

FROM sinhvien

WHERE hodem **LIKE** 'Lê%'

| HODEM | TEN |
|-----------------|------|
| Lê Thị Thanh | Châu |
| Lê Thị | Anh |
| Lê Văn Khoa | Bảo |
| Lê Thị | Dí |
| Lê Tất Uyên | Châu |
| Lê Hoàng Phương | Anh |
| Lê Thị Vân | Anh |
| Lê Đăng | Ánh |
| Lê Huy | Đan |
| Lê Thúc Quốc | Anh |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

d. Toán tử LIKE (NOT LIKE)

Ví dụ 25: Liệt kê danh sách sinh viên có họ đệm chứa từ 'Lê' và tên có ký tự đầu nằm trong 2 ký tự A hoặc B

```
SELECT hodem,ten  
FROM sinhvien  
WHERE hodem LIKE 'Lê%'  
AND ten LIKE '[AB]%'
```

| HODEM | TEN |
|-----------------|-----|
| Lê Thị | Ành |
| Lê Văn Khoa | Bảo |
| Lê Hoàng Phương | Ành |
| Lê Thị Vân | Ành |
| Lê Thúc Quốc | Ành |

I. NGÔN NGỮ TRUY VẤN SQL

2.2. Truy vấn có điều kiện (WHERE)

e. Giá trị NULL (NOT NULL)

- Nếu không có dữ liệu được nhập cho cột và không có mặc định cho cột hay kiểu dữ liệu trên cột đó. Kiểu số sẽ chứa giá trị NULL nếu giá trị được chỉ định gây tràn số.
- Để kiểm tra giá trị của một cột có giá trị NULL hay không, ta sử dụng cách viết:

WHERE tên_cột **IS NULL**

hoặc:

WHERE tên_cột **IS NOT NULL**

I. NGÔN NGỮ TRUY VẤN SQL

2.3. Tạo mới bảng dữ liệu từ kết quả của câu lệnh SELECT

Câu lệnh **SELECT ... INTO** có tác dụng tạo một bảng mới có cấu trúc và dữ liệu được xác định từ kết quả của truy vấn

Ví dụ 26: tạo một bảng TUOISV bao gồm các thuộc tính HODEM, TEN và TUOI

```
SELECT hodem, ten, year(getdate())-year(ngaysinh)  
AS tuoi INTO tuoisv FROM sinhvien
```

I. NGÔN NGỮ TRUY VẤN SQL

2.4. Sắp xếp kết quả câu truy vấn (ORDER BY)

Ví dụ 27: danh sách các môn học và sắp xếp theo chiều giảm dần của số đơn vị học trình

SELECT top 5 mamh,tenmh

FROM monhoc

ORDER BY sodvht **DESC**

I. NGÔN NGỮ TRUY VẤN SQL

2.4. Sắp xếp kết quả câu truy vấn (ORDER BY)

Ví dụ 28 : danh sách SINH VIEN bao gồm các thông tin **hodem,ten,gioitinh, tuoi** xếp hai cột 3 và 4

```
SELECT hodem,ten,gioitinh,  
YEAR(GETDATE())- YEAR(ngaysinh) AS tuoi
```

```
FROM sinhvien
```

```
WHERE ten='Bình'
```

```
ORDER BY 3, 4
```

| HODEM | TEN | GIOITINH | TUOI |
|---------------|------|----------|------|
| Nguyễn Thị | Bình | 0 | 23 |
| Hoàng Văn | Bình | 1 | 21 |
| Châu Văn Quốc | Bình | 1 | 21 |
| Nguyễn Thanh | Bình | 1 | 22 |
| Nguyễn Đình | Bình | 1 | 22 |
| Nguyễn Công | Bình | 1 | 25 |

I. NGÔN NGỮ TRUY VẤN SQL

2.4. Sắp xếp kết quả câu truy vấn (ORDER BY)


Ví dụ 29: danh sách các môn học và sắp xếp theo chiều giảm dần của số đơn vị học trình

```
SELECT * FROM monhoc ORDER BY sodvht DESC
```

Ví dụ 30: Liệt kê danh sách sinh viên

```
SELECT hodem,ten,gioitinh,  
year(getdate())-  
year(ngaysinh) AS tuoi
```

```
FROM sinhvien ORDER BY gioitinh,tuoi
```



Đặt tên cho
cột công thức

I. NGÔN NGỮ TRUY VẤN SQL

2.5. Phép hợp

- Gộp kết quả của hai hay nhiều truy vấn thành một tập kết quả duy nhất.
- Phép hợp chỉ giữ lại một dòng dữ liệu giống nhau .
- Nếu muốn giữ lại các dòng này, ta phải sử dụng thêm từ khoá ALL trong truy vấn thành phần.

I. NGÔN NGỮ TRUY VẤN SQL

2.5. Phép hợp

Câu_lệnh_1

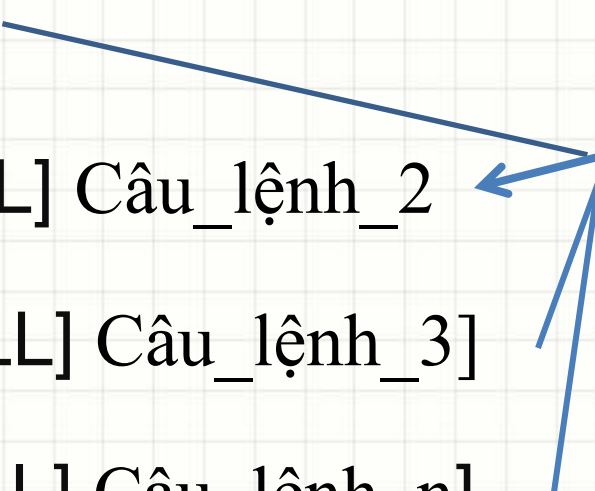
UNION [ALL] Câu_lệnh_2

[**UNION** [ALL] Câu_lệnh_3]

[**UNION** [ALL] Câu_lệnh_n]

[**ORDER BY** cột_sắp_xếp]

[**COMPUTE** danh_sách_hàm_gộp [BY
danh_sách_cột]]



```
SELECT  danh_sách_cột
[FROM
danh_sách_bảng|khung_nhìn]
[WHERE điều_kiện]
[GROUP BY danh_sách_cột]
[HAVING điều_kiện]
```

I. NGÔN NGỮ TRUY VẤN SQL

2.5. Phép hợp

- Ví dụ 31:

SELECT A,B **FROM** Table1
UNION ALL
SELECT D,E **FROM** table2

| A | B |
|---|---|
| a | 1 |
| b | 2 |
| d | 4 |
| b | 6 |
| a | 1 |
| b | 2 |

| D | E |
|---|---|
| d | 1 |
| e | 2 |
| f | 3 |
| g | 4 |

| A | B |
|---|---|
| a | 1 |
| b | 2 |
| d | 4 |
| b | 6 |
| a | 1 |
| b | 2 |
| d | 1 |
| e | 2 |
| f | 3 |
| g | 4 |

- Ví dụ 32:

SELECT A,B **FROM**
Table1
UNION
SELECT D,E **FROM** table2

| A | B |
|---|---|
| a | 1 |
| b | 2 |
| b | 6 |
| d | 1 |
| d | 4 |
| e | 2 |
| f | 3 |
| g | 4 |

I. NGÔN NGỮ TRUY VẤN SQL

2.5. Phép hợp

- Số cột , Kiểu dữ liệu, Thứ tự cột, Tiêu đề cột
- Truy vấn thành phần đầu tiên có thể có **INTO**
- Mệnh đề **ORDER BY** và **COMPUTE**
- Mệnh đề **GROUP BY** và **HAVING**
- Phép toán **UNION** có thể được sử dụng bên trong câu lệnh **INSERT**.
- Phép toán **UNION** không được sử dụng trong câu lệnh **CREATE VIEW**.

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

- Truy vấn dữ liệu từ hai hay nhiều bảng.
- Phép nối kết hợp các dòng dữ liệu trong các bảng khác nhau lại theo một hoặc nhiều điều kiện nào đó và hiển thị chúng trong kết quả truy vấn.

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Mã lớp và tên lớp của các lớp thuộc Khoa Công nghệ Thông tin

Thông tin

SELECT malop,tenlop

FROM khoa,lop

WHERE khoa.makhoa =

lop.makhoa **AND**

tenkhoa='Khoa Công nghệ

Thông tin'

| MAKHOA | TENKHOA | DIENTHOAI |
|--------|--------------------------|-----------|
| DHT01 | Khoa Toán cơ - Tin học | 054822407 |
| DHT02 | Khoa Công nghệ thông tin | 054826767 |
| DHT03 | Khoa Vật lý | 054823462 |
| DHT04 | Khoa Hóa học | 054823951 |
| DHT05 | Khoa Sinh học | 054822934 |
| DHT06 | Khoa Địa lý - Địa chất | 054823837 |

| MALOP | TENLOP | KHOA | HEDAOTAO | NAMN |
|--------|----------|------|-----------|------|
| C24101 | Toán K24 | 24 | Chính quy | 2000 |
| C24102 | Tin K24 | 24 | Chính quy | 2000 |
| C24103 | Lý K24 | 24 | Chính quy | 2000 |
| C24301 | Sinh K24 | 24 | Chính quy | 2000 |
| C25101 | Toán K25 | 25 | Chính quy | 2001 |
| C25102 | Tin K25 | 25 | Chính quy | 2001 |
| C25103 | Lý K25 | 25 | Chính quy | 2001 |
| C25301 | Sinh K25 | 25 | Chính quy | 2001 |
| C26101 | Toán K26 | 26 | Chính quy | 2002 |
| C26102 | Tin K26 | 26 | Chính quy | 2002 |

| |
|-------|
| DHT03 |
| DHT05 |
| DHT01 |
| DHT02 |
| DHT03 |
| DHT05 |
| DHT01 |
| DHT02 |

| MALOP | TENLOP |
|--------|---------|
| C24102 | Tin K24 |
| C25102 | Tin K25 |
| C26102 | Tin K26 |

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Để thực hiện được một phép nối, cần phải xác định được những yếu tố sau:

- Cột hiển thị
- Bảng tham gia vào truy vấn.
- Điều kiện để thực hiện

Danh sách chọn trong phép nối

- trùng tên nhau thì tên cột phải được viết dưới dạng : **tên_bảng.tên_cột**
- Dấu sao (*) : **tên_bảng.***

Mệnh đề **FROM** trong phép nối

Mệnh đề **WHERE** trong phép nối

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Ví dụ 33: mã SV, họ và tên, mã lớp, tên lớp và tên khoa

SELECT masv, hodem, ten, sinhvien.malop, tenlop, tenkhoa

FROM sinhvien, lop, khoa

WHERE sinhvien.malop = lop.malop **AND**

lop.makhoa=khoa.makhoa

Phép nối tự nhiên

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Phép tự nối và các bí danh

Ví dụ 34: họ tên và ngày sinh của các sinh viên có cùng ngày sinh với sinh viên Trần Thị Kim Anh

SELECT b.hodem,b.ten,b.ngaysinh

FROM sinhvien a, sinhvien b

WHERE a.hodem='Trần Thị Kim' **AND** a.ten='Anh' **AND**

a.ngaysinh=b.ngaysinh **AND** a.masv<>b.masv



Tự nối

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Phép nối không dựa trên tiêu chuẩn bằng

- Phép nối ngoài (**outer-join**): cho phép những dòng không thoả mãn điều kiện nối có mặt trong kết quả của phép nối.
- **SQL cung cấp các loại phép nối ngoài sau đây:**
- **Phép nối ngoài trái (ký hiệu: *=)**
- **Phép nối ngoài phải (ký hiệu: =*)**

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Phép nối không dựa trên tiêu chuẩn bằng

Ví dụ 35:

```
SELECT * FROM nhanvien,donvi  
WHERE nhanvien.madv=donvi.madv
```

Bảng DONVI

| MADV | TENDV |
|------|------------|
| 1 | Doi ngoai |
| 2 | Hanh chinh |
| 3 | Ke toan |
| 4 | Kinh doanh |

Bảng NHANVIEN

| HOTEN | MADV |
|--------|------|
| Thanh | 1 |
| Hoa | 2 |
| Nam | 2 |
| Vinh | 1 |
| Hung | 5 |
| Phuong | NULL |

Bảng kết quả

| HOTEN | MADV | MADV | TENDV |
|-------|------|------|------------|
| Thanh | 1 | 1 | Doi ngoai |
| Hoa | 2 | 2 | Hanh chinh |
| Nam | 2 | 2 | Hanh chinh |
| Vinh | 1 | 1 | Doi ngoai |

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Phép nối không dựa trên tiêu chuẩn bằng

Ví dụ 35: **SELECT** *

FROM nhanvien,donvi

WHERE nhanvien.madv*=donvi.madv

Bảng DONVI

| MADV | TENDV |
|------|------------|
| 1 | Doi ngoai |
| 2 | Hanh chinh |
| 3 | Ke toan |
| 4 | Kinh doanh |

Bảng NHANVIEN

| HOTEN | MADV |
|--------|------|
| Thanh | 1 |
| Hoa | 2 |
| Nam | 2 |
| Vinh | 1 |
| Hung | 5 |
| Phuong | NULL |

Bảng kết quả

| HOTEN | MADV | MADV | TENDV |
|--------|------|------|------------|
| Thanh | 1 | 1 | Doi ngoai |
| Hoa | 2 | 2 | Hanh chinh |
| Nam | 2 | 2 | Hanh chinh |
| Vinh | 1 | 1 | Doi ngoai |
| Hung | 5 | NULL | NULL |
| Phuong | NULL | NULL | NULL |

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

- Phép nối trong

tên_bảng_1 **[INNER] JOIN** tên_bảng_2 **ON**
điều_kiện_nối

Ví dụ 36: họ tên và ngày sinh của các sinh viên lớp
Tin K24

```
SELECT hodem,ten,nsinh FROM sinhvien s,lop l  
WHERE tenlop='Tin K24' AND s.malop=l.malop
```

I. NGÔN NGỮ TRUY VẤN SQL

- Hay

```
SELECT hodem,ten,nsinh FROM sinhvien  
INNER JOIN lop ON  
sinhvien.malop=lop.malop  
WHERE tenlop='Tin K24'
```

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Thực hiện phép nối trong trên nhiều bảng

- Ví dụ 38: hiển thị họ tên và ngày sinh của các sinh viên thuộc Khoa Công nghệ Thông tin

SELECT hodem,ten,ngaysinh

FROM

(sinhvien **INNER JOIN** lop ON
sinhvien.malop=lop.malop)

INNER JOIN khoa ON lop.makhoa=khoa.makhoa

WHERE tenkhoa=N'Khoa công nghệ thông tin'

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

Phép nối ngoài

- Phép nối ngoài trái (LEFT OUTER JOIN)
- Phép nối ngoài phải (RIGHT OUTER JOIN)
- Phép nối ngoài đầy đủ (FULL OUTER JOIN)

tên_bảng_1

LEFT|RIGHT|FULL [OUTER] JOIN

tên_bảng_2 **ON** điều_kiện_nối

Bảng DONVI

| MADV | TENDV |
|------|------------|
| 1 | Doi ngoai |
| 2 | Hanh chinh |
| 3 | Ke toan |
| 4 | Kinh doanh |

Bảng NHANVIEN

| HOTEN | MADV |
|--------|------|
| Thanh | 1 |
| Hoa | 2 |
| Nam | 2 |
| Vinh | 1 |
| Hung | 5 |
| Phuong | NULL |

• Ví dụ 37:

SELECT *

FROM nhanvien

LEFT OUTER JOIN donvi

ON nhanvien.madv=donvi.madv

SELECT *

FROM Nhanvien

RIGHT OUTER JOIN

donvi **ON**

nhanvien.madv=donvi.madv

| HOTEN | MADV | MADV | TENDV |
|--------|------|------|------------|
| Thanh | 1 | 1 | Doi ngoai |
| Hoa | 2 | 2 | Hanh chinh |
| Nam | 2 | 2 | Hanh chinh |
| Vinh | 1 | 1 | Doi ngoai |
| Hung | 5 | NULL | NULL |
| Phuong | NULL | NULL | NULL |

| HOTEN | MADV | MADV | TENDV |
|-------|------|------|------------|
| Thanh | 1 | 1 | Doi ngoai |
| Vinh | 1 | 1 | Doi ngoai |
| Hoa | 2 | 2 | Hanh chinh |
| Nam | 2 | 2 | Hanh chinh |
| NULL | NULL | 3 | Ke toan |
| NULL | NULL | 4 | Kinh doanh |

I. NGÔN NGỮ TRUY VẤN SQL

2.6. Phép nối (phép kết)

• Phép nối ngoài đây đủ hiển thị trong kết quả truy vấn cả những dòng dữ liệu không thoả điều kiện nối của cả hai bảng tham gia vào phép nối.

• Ví dụ 37:

SELECT *

FROM nhanvien

FULL OUTER JOIN

donvi **ON**

nhanvien.madv=donvi.madv

Bảng DONVI

| MADV | TENDV |
|------|------------|
| 1 | Doi ngoai |
| 2 | Hanh chinh |
| 3 | Ke toan |
| 4 | Kinh doanh |

Bảng NHANVIEN

| HOTEN | MADV |
|--------|------|
| Thanh | 1 |
| Hoa | 2 |
| Nam | 2 |
| Vinh | 1 |
| Hung | 5 |
| Phuong | NULL |

| HOTEN | MADV | MADV | TENDV |
|--------|------|------|------------|
| Thanh | 1 | 1 | Doi ngoai |
| Hoa | 2 | 2 | Hanh chinh |
| Nam | 2 | 2 | Hanh chinh |
| Vinh | 1 | 1 | Doi ngoai |
| Hung | 5 | NULL | NULL |
| Phuong | NULL | NULL | NULL |
| NULL | NULL | 4 | Kinh doanh |
| NULL | NULL | 3 | Ke toan |

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

- Phân hoạch các dòng dữ liệu trong bảng thành các nhóm dữ liệu để thực hiện tính toán các giá trị thống kê như tính tổng, tính giá trị trung bình,...
- **Các hàm gộp**
 - Được sử dụng để tính giá trị thống kê cho toàn bảng hoặc trên mỗi nhóm dữ liệu.
 - Chúng có thể được sử dụng như là các cột trong danh sách chọn của câu lệnh SELECT hoặc xuất hiện trong mệnh đề HAVING
 - Không được phép xuất hiện trong mệnh đề WHERE

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

Các hàm gộp

Hàm gộp

SUM([ALL | DISTINCT] *biểu_thức*)

AVG([ALL | DISTINCT] *biểu_thức*)

COUNT([ALL | DISTINCT] *biểu_thức*)

COUNT(*)

MAX(*biểu_thức*)

MIN(*biểu_thức*)

Chức năng

Tính tổng các giá trị.

Tính trung bình của các giá trị

Đếm số các giá trị trong biểu thức.

Đếm số các dòng được chọn.

Tính giá trị lớn nhất

Tính giá trị nhỏ nhất

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với **GROUP BY**

Trong đó:

- Hàm **SUM** và **AVG** chỉ làm việc với các biểu thức số.
- Hàm **SUM**, **AVG**, **COUNT**, **MIN** và **MAX** bỏ qua các giá trị **NULL** khi tính toán.
- Hàm **COUNT(*)** không bỏ qua các giá trị **NULL**.
- Mặc định, các hàm gộp thực hiện tính toán thống kê trên toàn bộ dữ liệu.
- Cần loại bỏ bớt các giá trị trùng nhau (chỉ giữ lại một giá trị), ta chỉ định thêm từ khoá **DISTINCT** ở trước biểu thức là đối số của hàm.

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

Thống kê trên toàn bộ dữ liệu

- Ví dụ 38: Để thống kê trung bình điểm lần 1 của tất cả các môn học

```
SELECT AVG(diemlan1)
```

```
FROM diemthi
```

- Ví dụ 39: Tuổi lớn nhất, tuổi nhỏ nhất và độ tuổi trung bình của tất cả các sinh viên sinh tại Huế:

```
SELECT MAX(YEAR(GETDATE())-  
YEAR(ngaysinh)),  
MIN(YEAR(GETDATE())-YEAR(ngaysinh)),  
AVG(YEAR(GETDATE())-YEAR(ngaysinh))  
FROM sinhvien WHERE noisinh='Huế'
```


I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

Thống kê theo nhóm dữ liệu

- Tính toán các giá trị thống kê trên các nhóm dữ liệu
- GROUP BY để phân hoạch dữ liệu vào trong các nhóm.

Ví dụ 40: cho biết sĩ số (số lượng sinh viên) của mỗi lớp

SELECT lop.malop,tenlop, COUNT(masv) **AS** siso
FROM lop,sinhvien
WHERE lop.malop=sinhvien.malop
GROUP BY lop.malop,tenlop

| MALOP | TENLOP | SISO |
|--------|----------|------|
| C24101 | Toán K24 | 5 |
| C24102 | Tin K24 | 8 |
| C24103 | Lý K24 | 7 |
| C24301 | Sinh K24 | 5 |
| C25101 | Toán K25 | 5 |
| C25102 | Tin K25 | 6 |
| C25103 | Lý K25 | 6 |
| C25301 | Sinh K25 | 8 |
| C26101 | Toán K26 | 5 |
| C26102 | Tin K26 | 5 |

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

Thống kê theo nhóm dữ liệu

Ví dụ 41:

SELECT s.masv,hodem,ten,

sum(diemlan1*sodvht)/**sum**(sodvht)

FROM sinhvien s, diemthi d,monhoc m

WHERE s.masv=d.masv AND d.mamh=m.mamh

GROUP BY s.masv,hodem,ten

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với **GROUP BY**

Chỉ định điều kiện đối với hàm gộp

- **HAVING** được sử dụng nhằm chỉ định điều kiện đối với các giá trị thống kê được sản sinh từ các hàm gộp

I. NGÔN NGỮ TRUY VẤN SQL

2.7. Thống kê dữ liệu với GROUP BY

Chỉ định điều kiện đối với hàm gộp

Ví dụ 42: trung bình điểm thi lần 1 của các sinh viên có điểm trung bình lớn hơn hoặc bằng 5

```
SELECT s.masv,hodem,ten,  
        SUM (diemlan1*sodvht)/sum(sodvht) dtb  
FROM sinhvien s, diemthi d,monhoc m  
WHERE s.masv=d.masv AND  
        d.mamonhoc=m.mamonhoc  
GROUP BY s.masv,hodem,ten  
HAVING sum(diemlan1*sodvht)/sum(sodvht)>=5
```

I. NGÔN NGỮ TRUY VẤN SQL

2.8. Thống kê dữ liệu với COMPUTE

GROUP BY biết được giá trị thống kê trên mỗi nhóm dữ liệu nhưng không biết được chi tiết dữ liệu trên mỗi nhóm

Ví dụ 42: Cho ta biết được số lượng lớp của mỗi khoa

```
SELECT k.makhoa, tenkhoa,  
        COUNT(malop) AS solop  
FROM khoa k, lop l  
WHERE k.makhoa=l.makhoa  
GROUP BY k.makhoa,tenkhoa
```

| MAKHOA | TENKHOA | SOLOP |
|--------|--------------------------|-------|
| DHT01 | Khoa Toán cơ - Tin học | 3 |
| DHT02 | Khoa Công nghệ thông tin | 3 |
| DHT03 | Khoa Vật lý | 2 |
| DHT05 | Khoa Sinh học | 2 |

I. NGÔN NGỮ TRUY VẤN SQL

2.8. Thống kê dữ liệu với COMPUTE

- Câu lệnh SELECT với COMPUTE cho chúng ta cả chi tiết dữ liệu và giá trị thống kê trên mỗi nhóm.
- COMPUTE hàm_gộp(tên_cột) [,..., hàm_gộp(tên_cột)] BY danh_sách_cột
- Trong đó:
 - Các hàm gộp có thể sử dụng bao gồm SUM, AVG, MIN, MAX và COUNT.
 - danh_sách_cột: là danh sách cột sử dụng để phân nhóm dữ liệu

Ví dụ 43: danh sách các lớp của mỗi khoa và tổng số các lớp của mỗi khoa

```
SELECT k.makhoa,
tenkhoa, malop,
tenlop
FROM khoa k,lop l
WHERE k.makhoa
      =l.makhoa
ORDER BY k.makhoa
COMPUTE
COUNT(malop) BY
k.makhoa
```

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|------------------------|------------|----------|
| DHT01 | Khoa Toán cơ - Tin học | C24101 | Toán K24 |
| DHT01 | Khoa Toán cơ - Tin học | C25101 | Toán K25 |
| DHT01 | Khoa Toán cơ - Tin học | C26101 | Toán K26 |
| | | <u>CNT</u> | |
| | | 3 | |

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|--------------------------|------------|---------|
| DHT02 | Khoa Công nghệ thông tin | C26102 | Tin K26 |
| DHT02 | Khoa Công nghệ thông tin | C25102 | Tin K25 |
| DHT02 | Khoa Công nghệ thông tin | C24102 | Tin K24 |
| | | <u>CNT</u> | |
| | | 3 | |

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|-------------|------------|--------|
| DHT03 | Khoa Vật lý | C24103 | Lý K24 |
| DHT03 | Khoa Vật lý | C25103 | Lý K25 |
| | | <u>CNT</u> | |
| | | 2 | |

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|---------------|------------|----------|
| DHT05 | Khoa Sinh học | C25301 | Sinh K25 |
| DHT05 | Khoa Sinh học | C24301 | Sinh K24 |
| | | <u>CNT</u> | |
| | | 2 | |

I. NGÔN NGỮ TRUY VẤN SQL

2.8. Thống kê dữ liệu với **COMPUTE BY**

- Từ khóa **DISTINCT** không cho phép sử dụng với các hàm gộp dòng
- Hàm **COUNT(*)** không được sử dụng trong **COMPUTE**.
- Sau **COMPUTE** có thể sử dụng nhiều hàm gộp, khi đó các hàm phải phân cách nhau bởi dấu phẩy.
- Các cột sử dụng trong các hàm gộp xuất hiện trong mệnh đề **COMPUTE** phải có mặt trong danh sách chọn.
- Không sử dụng **SELECT INTO** trong một câu lệnh **SELECT** có **COMPUTE**.

I. NGÔN NGỮ TRUY VẤN SQL

2.8. Thống kê dữ liệu với **COMPUTE BY**

- **COMPUTE ... BY** thì cũng phải sử dụng mệnh đề **ORDER BY**.
- cột trong **COMPUTE ... BY** là một tập con của những gì được liệt kê sau **ORDER BY**. cùng thứ tự từ trái qua phải, bắt đầu với cùng một biểu thức và không bỏ qua bất kỳ một biểu thức nào.
- **ORDER BY a, b, c**
- **COMPUTE F(X) BY a, b, c** **COMPUTE F(X) BY a, b**
COMPUTE F(X) BY a

I. NGÔN NGỮ TRUY VẤN SQL

2.8. Thống kê dữ liệu với COMPUTE BY

- Trong trường hợp sử dụng **COMPUTE** mà không có **BY** thì có thể không cần sử dụng **ORDER BY**, khi đó phạm vi tính toán của hàm gộp là trên toàn bộ dữ liệu.

Ví dụ 44: Hiển thị danh sách các lớp và tổng số lớp hiện có:

```
SELECT malop,tenlop,  
          hedaotao  
FROM lop  
ORDER BY makhoa  
COMPUTE COUNT(malop)
```

| <u>MALOP</u> | <u>TENLOP</u> | <u>HEDAOTAO</u> |
|--------------|---------------|-----------------|
| C24101 | Toán K24 | Chính quy |
| C25101 | Toán K25 | Chính quy |
| C26101 | Toán K26 | Chính quy |
| C26102 | Tin K26 | Chính quy |
| C25102 | Tin K25 | Chính quy |
| C24102 | Tin K24 | Chính quy |
| C24103 | Lý K24 | Chính quy |
| C25103 | Lý K25 | Chính quy |
| C25301 | Sinh K25 | Chính quy |
| C24301 | Sinh K24 | Chính quy |
| <u>CNT</u> | | |
| 10 | | |

Có thể thực hiện việc tính toán hàm gộp dòng trên các nhóm lồng nhau bằng cách sử dụng nhiều mệnh đề COMPUTE ... BY trong cùng một câu lệnh SELECT

- Ví dụ 45: Cho biết danh sách các lớp của mỗi khoa, tổng số lớp theo mỗi khoa và tổng số lớp hiện có

```
SELECT khoa.makhoa,  
       tenkhoa,malop,tenlop FROM khoa,lop WHERE  
       khoa.makhoa=lop.makhoa  
  
ORDER BY khoa.makhoa  
  
COMPUTE COUNT(malop) BY khoa.makhoa  
COMPUTE COUNT(malop)
```


| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|------------------------|--------|----------|
| DHT01 | Khoa Toán cơ - Tin học | C24101 | Toán K24 |
| DHT01 | Khoa Toán cơ - Tin học | C25101 | Toán K25 |
| DHT01 | Khoa Toán cơ - Tin học | C26101 | Toán K26 |

CNT

3

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|--------------------------|--------|---------|
| DHT02 | Khoa Công nghệ thông tin | C26102 | Tin K26 |
| DHT02 | Khoa Công nghệ thông tin | C25102 | Tin K25 |
| DHT02 | Khoa Công nghệ thông tin | C24102 | Tin K24 |

CNT

3

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|-------------|--------|--------|
| DHT03 | Khoa Vật lý | C24103 | Lý K24 |
| DHT03 | Khoa Vật lý | C25103 | Lý K25 |

CNT

2

| MAKHOA | TENKHOA | MALOP | TENLOP |
|--------|---------------|--------|----------|
| DHT05 | Khoa Sinh học | C25301 | Sinh K25 |
| DHT05 | Khoa Sinh học | C24301 | Sinh K24 |

CNT

2

CNT

10

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng

SELECT MANV, TENNV

FROM NHANVIEN, PHONGBAN

WHERE TENPHG='Nghien cuu' **AND** PHG=MAPHG

*Câu truy vấn cha
(Outer query)*

SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <so sánh tập hợp> (

SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>)

*Câu truy vấn con
(Subquery)*

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng

- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Câu truy vấn con thường trả về một tập các giá trị
- Mệnh đề WHERE của câu truy vấn cha
 - **<biểu thức> <so sánh tập hợp> <truy vấn con>**
 - *So sánh tập hợp thường đi cùng với một số toán tử*
 - IN, NOT IN
 - ALL
 - ANY hoặc SOME
 - *Kiểm tra sự tồn tại*
 - EXISTS, NOT EXISTS

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng

– Lồng phân cấp

- Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha.
- Khi thực hiện, câu truy vấn con sẽ được thực hiện trước.

– Lồng tương quan

- Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha.
- Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha.

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

```
SELECT MANV, TENNV  
FROM NHANVIEN, DIADIEM_PHG  
WHERE DIADIEM='TP HCM' AND PHG=MAPHG
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE PHG IN
```

(1, 5)

```
SELECT MAPHG  
FROM DIADIEM_PHG  
WHERE DIADIEM='TP HCM'
```

)

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

```
SELECT SODA
FROM NHANVIEN NV, PHANCONG PC
WHERE NV.MANV=PC.MA_NVIENT AND NV.HONV='Nguyễn'
UNION
SELECT MADA
FROM NHANVIEN NV, PHONGBAN PB, DEAN DA
WHERE NV.MANV=PB.TRPHG AND PB.MAPHG=DA.PHONG
AND NV.HONV='Nguyễn'
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

```
SELECT DISTINCT TENDA  
FROM DEAN  
WHERE MADA IN (
```

```
    SELECT SODA  
    FROM NHANVIEN, PHANCONG  
    WHERE MANV=MA_NVNIEN AND HONV='Nguyễn'
```

```
)  
OR MADA IN (
```

```
    SELECT MADA  
    FROM NHANVIEN, PHONGBAN, DEAN  
    WHERE MANV=TRPHG AND MAPHG=PHONG  
    AND HONV='Nguyễn')
```


I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

- *Tìm những nhân viên không có thân nhân nào*

```
SELECT *  
FROM NHANVIEN  
WHERE MANV NOT IN (  
    SELECT MA_NV  
    FROM THANNHAN )
```

```
SELECT *  
FROM NHANVIEN  
WHERE MANV <> ALL (  
    SELECT MA_NV  
    FROM THANNHAN )
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

- *Lưu những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4*

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ANY (  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG=4 )
```

```
SELECT NV1.*  
FROM NHANVIEN NV1, NHANVIEN NV2  
WHERE NV1.LUONG > NV2.LUONG AND NV2.PHG=4
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

- *Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên phòng 4*

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ALL (  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG=4 )
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng phân cấp

- *Tìm những trưởng phòng có tối thiểu một thân nhân*

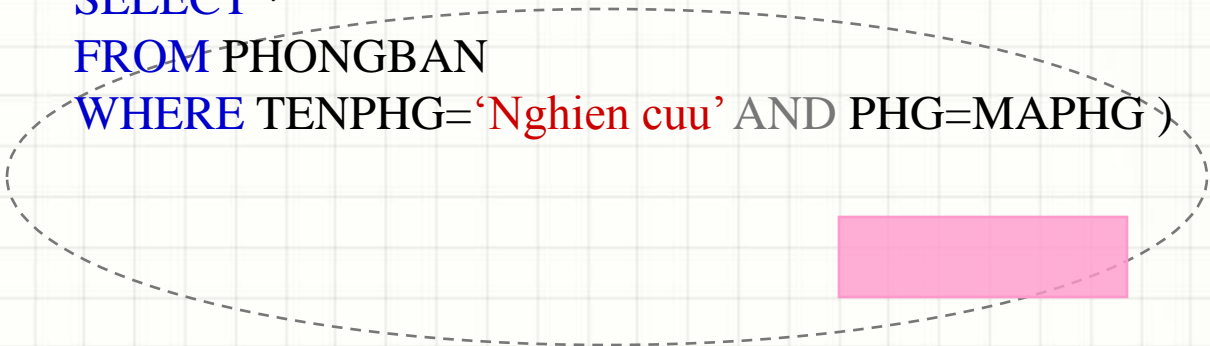
```
SELECT *  
FROM NHANVIEN  
WHERE MANV IN (SELECT MA_NVIENT FROM  
THANNHAN)  
AND MANV IN (SELECT TRPHG FROM PHONGBAN)
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng tương quan

```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE TENPHG='Nghien cuu' AND PHG=MAPHG
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE EXISTS (  
    SELECT *  
    FROM PHONGBAN  
    WHERE TENPHG='Nghien cuu' AND PHG=MAPHG )
```



I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng tương quan

- Tìm nhân viên có người thân cùng tên và cùng giới tính

```
SELECT *  
FROM NHANVIEN NV  
WHERE EXISTS (  
    SELECT *  
    FROM THANNHAN TN  
    WHERE NV.MANV=TN.MA_NVIENT  
    AND NV.TENNV=TN.TENTN  
    AND NV.PHAI=TN.PHAI )
```


I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng tương quan

- Tìm những nhân viên không có thân nhân nào

```
SELECT *  
FROM NHANVIEN  
WHERE NOT EXISTS (  
    SELECT *  
    FROM THANNHAN  
    WHERE MANV=MA_NVIEN)
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng tương quan

- *Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4*

```
SELECT *  
FROM NHANVIEN NV1  
WHERE EXISTS (  
    SELECT *  
    FROM NHANVIEN NV2  
    WHERE NV2.PHG=4  
    AND NV1.LUONG>NV2.LUONG)
```

I. NGÔN NGỮ TRUY VẤN SQL

2.9. Truy vấn lồng – lồng tương quan

- *Tìm những trưởng phòng có tôi thiểu một thân nhân*

```
SELECT *  
FROM NHANVIEN  
WHERE EXISTS (  
    SELECT *  
    FROM THANNHAN  
    WHERE MANV=MA_NVIEN )  
AND EXISTS (  
    SELECT *  
    FROM PHONGBAN  
    WHERE MANV=TRPHG )
```

Nhận xét *IN* và *EXISTS*

- **IN**

- <tên cột> IN <câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

- **EXISTS**

- Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS


Cho CSDL Địa lý có cấu trúc như sau :

- TINH_TP (MA_T_TP, DT, DS, TEN_T_TP)
- MIEN (MA_T_TP, MIEN)
- BIENGIOI (NUOC, MA_T_TP)
- LANGGIENG (MA_T_TP, LG)

1. .Xuất ra tên tỉnh, TP cùng với dân số của tỉnh,TP
2. a/ - Có diện tích $\geq 5,000 \text{ Km}^2$
b/ - Có diện tích $\geq [\text{input}]$
3. Xuất ra tên tỉnh, TP cùng với diện tích của tỉnh,TP :
 - a/ - Thuộc miền bắc
 - b/ - Thuộc miền $[\text{input}]$
4. Xuất ra các Tên tỉnh, TP biên giới thuộc miền $[\text{input}]$

1. Cho biết diện tích trung bình của các tỉnh, TP (Tổng DT/Tổng số tỉnh_TP).
2. Cho biết dân số cùng với tên tỉnh của các tỉnh, TP có diện tích $>7000 \text{ Km}^2$.
3. Cho biết dân số cùng với tên tỉnh của các tỉnh miền Bắc.
4. Cho biết mã các nước biên giới của các tỉnh miền Nam.

1. Cho biết diện tích trung bình của các tỉnh, TP. (sử dụng hàm)
2. Cho biết mật độ dân số (DS/DT) cùng với tên tỉnh, TP của tất cả các tỉnh, TP.
3. Cho biết tên các tỉnh, TP láng giềng của tỉnh Long An.
4. Cho biết số lượng các tỉnh, TP giáp với CPC.

- 
1. Cho biết tên những tỉnh, TP có diện tích lớn nhất.
 2. Cho biết tỉnh, TP có mật độ DS đông nhất.
 3. Cho biết tên những tỉnh, TP giáp với hai nước biên giới khác nhau.
 4. Cho biết danh sách các miền cùng với các tỉnh, TP trong các miền đó.

1. Cho biết những tỉnh, TP có diện tích nhỏ hơn diện tích trung bình của tất cả tỉnh, TP.
2. Cho biết tên những tỉnh, TP giáp với các tỉnh, TP ở miền Trung và không phải là miền Trung.
3. Cho biết tên những tỉnh, TP có một diện tích lớn hơn tất cả các tỉnh, TP láng giềng của nó.
4. Cho biết tên những tỉnh, TP mà ta có thể đến bằng cách đi từ TP.HCM