



Chương 4

NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC (SQL – Structured Query Language)

NỘI DUNG CHƯƠNG 4



1. Ngôn ngữ mô tả dữ liệu
(Data Definition Language - DDL)
2. Ngôn ngữ thao tác dữ liệu
(Data Manipulation Language - DML)
3. Ngôn ngữ truy vấn dữ liệu
(Structured Query Language - SQL)
4. Một số dạng truy vấn thường gặp
5. Minh họa giữa Đại số quan hệ & SQL

0. GIỚI THIỆU

- SQL là ngôn ngữ truy vấn dựa trên đại số quan hệ và là ngôn ngữ phi thủ tục. Công dụng chính của SQL bao gồm:
 - Định nghĩa dữ liệu (data definition).
 - Truy vấn và Xử lý dữ liệu (data manipulation).
 - Điều khiển dữ liệu (data control)
- Khởi nguồn của SQL là SEQUEL - Structured English Query Language, năm 1974)
- Các chuẩn SQL
 - SQL89
 - SQL92 (SQL2)
 - SQL99 (SQL3)
- Quy định về cú pháp của SQL:
 - Có thể được viết trên nhiều dòng (cuối mỗi lệnh nên có lệnh GO)
 - không được phép viết tách xuống hàng đối với các từ khóa, tên hàm, tên thuộc tính, tên bảng, tên đối tượng.
 - Không phân biệt chữ hoa, chữ thường trong câu lệnh.

3

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL)

1.1. CÁC LỆNH CHÍNH:

- Gồm 3 lệnh chính:
 - Tạo mới đối tượng:


```
CREATE <Loại đối tượng> <Tên đối tượng>
```
 - Hiệu chỉnh đối tượng:


```
ALTER <Loại đối tượng> <Tên đối tượng>
          ADD|DROP <Thành phần> <Tên thành phần>
```
 - Xóa đối tượng:


```
DROP <Loại đối tượng> <Tên đối tượng>
```

Giải thích:

- <Loại đối tượng>: là các đối tượng như Database, Table, View, Stored Procedure, Trigger, ...
- <Thành phần>: là các thành phần như Constraint, Column, ...

4

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL)

1.2. Database:

1.2.1. Lệnh tạo Database

■ Cú pháp:

```
CREATE DATABASES Tên_CSDL
[On Primary (Name = TênFile_CSDL,
    Filename= 'Đường dẫn và tên file mdf',
    Size = Kích thước khởi tạo ,
    MaxSize = kích thước tối đa,
    FileGrowth = kích thước mỗi lần được tăng) ]
[Log On ( Name = TênFile_log,
    Filename = 'Đường dẫn và tên file ldf',
    Size = Kích thước khởi tạo ,
    MaxSize = kích thước tối đa,
    FileGrowth = kích thước mỗi lần được tăng) ]
```

Giải thích:

- Nếu không dùng 2 tham số *On Primary* và *Log On* ⇒ 2 file .mdf và .ldf thường sẽ được lưu vào folder 'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA'
- Nếu có dùng tham số *On Primary* và không dùng tham số *Log On* ⇒ File .ldf sẽ được lưu chung folder với file .mdf

5

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL)

1.2. Database:

1.2.1. Lệnh tạo Database (tt)

```
CREATE DATABASES Tên_CSDL
[On Primary (Name = TênFile_CSDL,
    Filename= 'Đường dẫn và tên file mdf',
    Size = Kích thước khởi tạo ,
    MaxSize = kích thước tối đa,
    FileGrowth = kích thước mỗi lần được tăng) ]
[Log On ( Name = TênFile_log,
    Filename = 'Đường dẫn và tên file ldf',
    Size = Kích thước khởi tạo ,
    MaxSize = kích thước tối đa,
    FileGrowth = kích thước mỗi lần được tăng) ]
```

Giải thích:

- **Size:** nếu không chỉ ra sẽ dựa vào kích thước của file model.mdf
- **MaxSize:** nếu không chỉ ra, kích thước này sẽ là không giới hạn
- **FileGrowth:** nếu không chỉ ra sẽ dựa vào kích thước của file model.mdf

6

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL)

1.2. Database:

1.2.1. Lệnh tạo Database (tt)

- Ví dụ: tạo CSDL QuanLyBanHang. File .mdf và ldf được lưu vào folder theo ý người dùng

```
USE MASTER
GO
Create Database QuanLyBanHang
    On Primary ( Name = QLBHData,
                Filename = 'D:\Project\Data\QLBH_Data.mdf',
                Size = 10MB ,
                MaxSize = 100MB,
                FileGrowth = 10MB)
    Log On ( Name = QLBHLog,
            Filename = 'D:\Project\Data\QLBH_log.ldf',
            Size = 10MB ,
            MaxSize = 100MB,
            FileGrowth = 10MB)
```

7

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.2. Database: (tt)

1.2.2. Hiệu chỉnh Database

- Thêm file vào Database

```
ALTER DATABASE QuanLyBanHang
    ADD FILE( Name = QLBH2,
             Filename='D:\Project\Data\QLBH2_Data.ndf',
             Size = 10MB ,
             MaxSize = 100MB,
             FileGrowth = 10MB)
```

- Xóa file ra khỏi Database

```
Alter Database QuanLyBanHang
    Remove File QLBH2
```

- Hiệu chỉnh lại các tùy chọn đã định trước đó:

```
Alter Database QLVT
    Modify File ( Name=QLBHData,FileGrowth = 5MB)
```

1.1.3. Xóa toàn bộ Database

```
Drop Database QLVT
```

8

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table:

Các kiểu dữ liệu trong SQL Server

Các kiểu dữ liệu	SQL Server
Chuỗi ký tự	varchar(n), char(n), nvarchar(n), nchar(n) Text
Số	tinyint, smallint, int numeric, decimal, float, real
Ngày tháng	smalldatetime, datetime

9

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table: (tt)

1.3.1. Lệnh tạo table

■ Lệnh đơn giản

```
CREATE TABLE Tên_bảng
(
    Tên_cột1    Kiểu_dữ_liệu [NOT NULL] ,
    Tên_cột2    Kiểu_dữ_liệu [NOT NULL] [, ...]
)
```

- Từ khóa NOT NULL chỉ định không cho phép dữ liệu tại cột tương ứng bị bỏ trống.

10

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table: (tt)

1.3.1. Lệnh tạo table (tt)

■ Tạo table có cột định danh

```
CREATE TABLE Tên_bảng
(
    Tên_cột1    Kiểu_dữ_liệu_số
    IDENTITY [(Số_bắt_đầu, Chỉ_số_tăng)] ,
    Tên_cột2    Kiểu_dữ_liệu [NOT NULL] [, ...]
)
```

- Kiểu dữ liệu số: dạng số nguyên (int, smallint, tinyint, numeric và decimal)
 - Với numeric và decimal: cần chỉ định không lấy số lẻ.
- Số bắt đầu: SQL Server sử dụng để cấp phát cho mẫu tin đầu tiên. Mặc định là 1.
- Chỉ số tăng: số cộng lên để cấp phát cho những mẫu tin kế tiếp. Mặc định là 1.

11

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table: (tt)

1.3.1. Lệnh tạo table (tt)

■ Tạo table có khóa chính

- Cách 1: thường dùng khi khóa chính chỉ là 1 thuộc tính

```
CREATE TABLE Tên_bảng
(
    Tên_cột1    Kiểu_dữ_liệu PRIMARY KEY,
    Tên_cột2    Kiểu_dữ_liệu [NOT NULL] [, ...]
)
```

- Cách 2: thường dùng khi khóa chính gồm nhiều thuộc tính

```
CREATE TABLE Tên_bảng
(
    Tên_cột1    Kiểu_dữ_liệu PRIMARY KEY,
    Tên_cột2    Kiểu_dữ_liệu [NOT NULL] [, ...]
    PRIMARY KEY (TênThuộcTính1, TênThuộcTính2
                [, TênThuộcTính3,...]
)
```

12

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table: (tt)

1.3.2. Thay đổi cấu trúc table

1.3.2.1. Cú pháp chung:

```
ALTER TABLE Tên_Table
ADD|DROP COLUMN|CONSTRAINT|. . .
                        Tên_Thành_Phần_Cần_Thay_Đổi
```

1.3.2.2. Thêm cột mới vào table:

■ Cú pháp

```
ALTER TABLE Tên_Table
ADD Tên_cột Kiểu_dữ_liệu [, ...]
```

■ Giải thích:

- **Tên cột:** tên của cột mới được thêm vào bảng.
- **Kiểu dữ liệu:** kiểu dữ liệu tương ứng của cột mới.

■ Ví dụ:

```
ALTER TABLE DonDatHang
ADD NgayDatHang DATETIME
```

13

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)

1.3. Table: (tt)

1.3.2. Thay đổi cấu trúc table (tt)

1.2.2.3. Xóa cột ra khỏi table:

■ Cú pháp

```
ALTER TABLE Tên_Table
DROP Tên_cột [, ...]
```

■ Giải thích:

- **Tên cột:** tên của cột sẽ bị xóa.

■ Ví dụ:

```
ALTER TABLE DonDatHang
DROP NgayDatHang
```

1.2.2.4. Sửa đổi kiểu dữ liệu của cột :

■ Cú pháp

```
ALTER TABLE Tên_Table
ALTER COLUMN Tên_cột Kiểu_dữ_liệu_mới
```

■ Ví dụ:

```
ALTER TABLE VatTu
ALTER COLUMN Dvtinh VARCHAR(20)
```

14

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)
 1.3. Table: (tt)
 1.3.2. Thay đổi cấu trúc table (tt)

Thêm ràng buộc toàn vẹn

ALTER TABLE <tên table>
ADD CONSTRAINT <tên
 ràng buộc>

- UNIQUE** tên thuộc tính
- PRIMARY KEY** tên thuộc tính
- FOREIGN KEY** tên thuộc tính
REFERENCES tên_table
 (t_tính khóa chính) **[ON
 DELETE CASCADE] [ON
 UPDATE CASCADE]**
- CHECK** (tên t_tính điều_kiện)

15

1. NGÔN NGỮ MÔ TẢ DỮ LIỆU (Data Definition Language-DDL) (tt)
 1.3. Table: (tt)
 1.3.2. Thay đổi cấu trúc table (tt)

Một số ví dụ về lệnh ALTER

i. Tạo khóa chính cho table nhân viên

```
ALTER TABLE NhanVien ADD CONSTRAINT pk_NV
PRIMARY KEY (MaNV)
```

ii. Tạo khóa ngoại: nhân viên trực thuộc phòng ban

```
ALTER TABLE NhanVien ADD CONSTRAINT
fk_NV_PB FOREIGN KEY (MaPB) REFERENCES
PhongBan (MaPB)
```

iii. Tạo ràng buộc lương phải > 0

```
ALTER TABLE NhanVien ADD CONSTRAINT
chk_Luong CHECK (Luong>0)
```

16

2. LỆNH THAO TÁC DỮ LIỆU (Data Manipulation Language - DML)

2.1. Lệnh INSERT INTO

```
INSERT INTO Tên_bảng [ (Danh_sách_cột) ]
VALUES (Danh_sách_giá_trị)
```

- Có thể không cần chỉ định ra tên của các cột
 - Số lượng các giá trị bằng số các cột trong bảng
 - Danh sách các giá trị mà chúng ta đưa vào phải theo đúng thứ tự của các cột bên trong bảng

```
INSERT INTO VATTU (MAVTU, TENVTU, DVTINH, PHANTRAM)
VALUES ('LO01', 'Loa Panasonic 1000W', 'Bộ', 10)
```

17

2. CÁC LỆNH THAO TÁC DỮ LIỆU (Data Manipulation Language - DML)

2.2. Lệnh INSERT INTO SELECT

```
INSERT [INTO] Tên_bảng [ (Danh_sách_cột) ]
SELECT Danh_sách_cột
FROM Tên_bảng_dl_nguồn
WHERE Điều_kiện_lọc
```

- Danh sách các cột của câu SELECT phải tương ứng với các cột của mệnh đề INSERT

18

2. CÁC LỆNH THAO TÁC DỮ LIỆU (Data Manipulation Language - DML) (tt)

2.3. Lệnh DELETE FROM

```
DELETE [FROM] Tên_bảng
      [FROM Tên_bảng1
        INNER|LEFT|RIGHT JOIN Tên_bảng2
        ON Biểu_thức_liên_kết]
      [WHERE Điều_kiện_xóa]
```

- Tên bảng: bảng có các dòng dữ liệu muốn hủy bỏ
- Tên bảng1, tên bảng2: các bảng có quan hệ dữ liệu, được dùng để kết nối các quan hệ nhằm tra cứu các thông tin trong khi xóa dữ liệu
- Nếu không sử dụng mệnh đề WHERE thì tất cả các dòng dữ liệu sẽ bị hủy

19

2. CÁC LỆNH THAO TÁC DỮ LIỆU (Data Manipulation Language - DML) (tt)

2.4. Lệnh UPDATE

```
UPDATE Tên_bảng
      SET Tên_cột = Biểu_thức [ , ...]
      [FROM Tên_bảng1
        INNER|LEFT|RIGHT JOIN Tên_bảng2
        ON Biểu_thức_liên_kết]
      [WHERE Điều_kiện_sửa_đổi]
```

UPDATE PNHAP

```
SET TGNHAP = ( SELECT SUM(SLNHAP*DGNHAP)
                FROM CTPNHAP CTPN
                WHERE PNHAP.SOPN=CTPN.SOPN )
```

20

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL)

■ Tương quan giữa SQL và ĐSQH

- Phép chọn: $\delta_{\langle \text{điều kiện} \rangle}$ ($\langle \text{tên quan hệ} \rangle$) tương ứng mệnh đề WHERE trong SQL
- Phép chiếu: $\Pi_{\langle \text{thuộc tính} \rangle}$ ($\langle \text{tên quan hệ} \rangle$) tương ứng mệnh đề SELECT trong SQL
- Các phép toán tập hợp: Hợp tương ứng mệnh đề UNION trong SQL
- Tích Descartes ($R \times S$) tương ứng mệnh đề FROM trong SQL
- Kết (θ): là các điều kiện trong mệnh đề WHERE
 - Equi-join: Tất cả đều là phép so sánh bằng
 - Inner join, left join, right join: cùng tên

21

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

■ Các toán tử sử dụng trong truy vấn

- Toán tử so sánh: $=, >, <, >=, <=, <>$
- Toán tử logic: AND, OR, NOT
- Phép toán số học: $+, -, *, /$
- Khác:
 - BETWEEN
 - IS NULL
 - LIKE: So sánh chuỗi tương đối (hay tìm chuỗi con trong toàn bộ chuỗi)
 - Cú pháp: $s \text{ LIKE } 'p'$, p có thể chứa $\%$ hoặc $_$
 - $\%$: thay thế bất kỳ chuỗi ký tự nào có chiều dài ≥ 0
 - $_$: thay thế **một** ký tự bất kỳ
 - IN
 - EXISTS

22

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

■ Ví dụ sử dụng các toán tử sử dụng trong truy vấn

■ Toán tử LIKE

```
SELECT      MaNV, HoNV, TenLot, TenNV
FROM      NhanVien
WHERE HoNV LIKE 'Ng%'
```

■ Toán tử IN

```
SELECT      TenNV, NgaySinh
FROM      NhanVien
WHERE month(NgaySinh) in (7,8,9)
hay
SELECT      TenNV
FROM      NhanVien
WHERE MaNV NOT IN (
                SELECT      MaNV
                FROM          PhanCong )
```

23

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

SELECT [**ALL** | **DISTINCT** | **TOP n** [**PERCENT**] [**WITH TIES**]

* | <thuộc tính> | <biểu thức> | <hàm tập hợp>

[**INTO** <tên table sẽ được tạo mới>]

FROM <tên table>

[**WHERE** <biểu thức điều kiện1>]

[**GROUP BY** <thuộc tính>]

[**HAVING** <biểu thức điều kiện2>]

[**ORDER BY** <thuộc tính> [**ASC** | **DESC**], ...]

[**COMPUTE** <hàm tập hợp> [**BY** <biểu thức2>]

24

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.1. SELECT: thực hiện phép chọn trên các thuộc tính, biểu thức tương ứng.

■ Xác định số dòng có trong tập kết quả:

- **ALL:** (mặc định) hiển thị tất cả các dòng có trong lệnh SELECT.
- **DISTINCT:** khi có nhiều dòng trùng nhau trên thuộc tính được chỉ ra, chỉ giữ lại 1 dòng trong số các dòng trùng nhau.
- **TOP n [PERCENT] [WITH TIES]:**
 - Hiển thị n dòng đầu tiên có trong tập kết quả.
 - Nếu có thêm từ khóa PERCENT, n sẽ tính theo phần trăm.
 - Mệnh đề TOP có thể được dùng trong các phát biểu SELECT, INSERT, UPDATE, và DELETE. Khi dùng với các phát biểu INSERT, UPDATE và DELETE, giá trị n phải được đặt trong cặp ngoặc đơn.
 - **WITH TIES:** nếu được dùng, những dòng có số thứ tự dòng trong kết quả lớn hơn giá trị n nhưng có giá trị của thuộc tính được chọn bằng (hoặc giống) với dòng cuối cùng (trong tập n dòng quy định) cũng được hiện ra trong tập kết quả. Lúc này số dòng hiện ra có thể lớn hơn n.

25

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.1. SELECT:

■ Xác định số cột có trong tập kết quả:

- *****: dùng để chỉ tất cả các thuộc tính có trong <bảng>
- **<thuộc tính>**: tên các thuộc tính cần hiển thị trong kết quả
- **<biểu thức>**: là sự kết hợp một cách hợp lệ giữa các thuộc tính, các toán tử và các hàm.

3.2. INTO: kết quả truy vấn sẽ được đưa vào bảng mới

- **<tên bảng mới>**: bảng mới sẽ được tạo để lưu giữ kết quả của lệnh truy vấn.
- Các cột có trong bảng <tên bảng mới> chính là các cột được chỉ định ngay sau từ khóa **SELECT**, với kiểu của chúng cũng chính là kiểu của các cột có trong bảng gốc.
- Nếu không có mẫu tin nào thỏa điều kiện **WHERE**, bảng mới vẫn tạo ra nhưng không có mẫu tin nào.

26

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.3. FROM:

- Cho biết nguồn dữ liệu cung cấp cho lệnh *SELECT* được lấy từ bảng nào.
- Khi lấy dữ liệu từ nhiều bảng: sử dụng 1 trong các cách kết hợp sau: INNER|LEFT|RIGHT|FULL JOIN
- Có thể đặt bí danh cho các tên bảng.

3.4. WHERE: áp dụng phép chọn trên table

- *<biểu thức điều kiện>*: là 1 biểu thức có giá trị kết quả thuộc kiểu Boolean.
- Khi lấy dữ liệu từ nhiều table nhưng không dùng các cách kết hợp: INNER|LEFT|RIGHT|FULL JOIN, khi đó có thể dùng phép kết (>=, =, <>, <=, =, ...). Tuy nhiên phổ biến nhất là phép kết bằng dưới dạng:

<Tên_Table1>.<TênField> = <Tên_Table2>.<TênField>

27

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

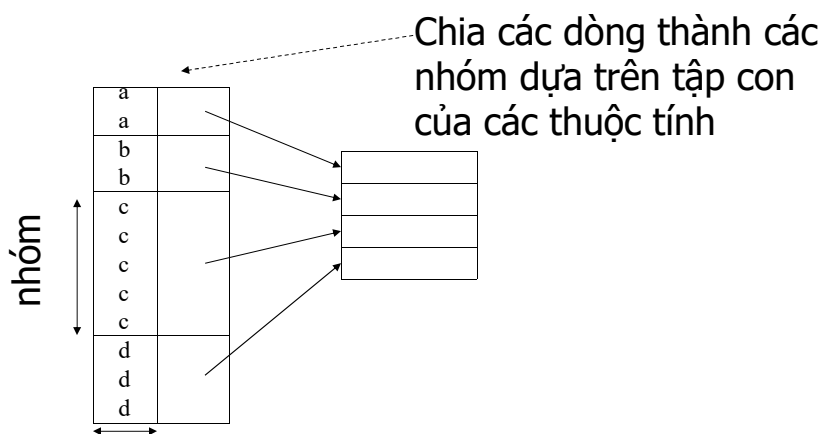
3.5. GROUP BY:

- Dùng để tạo ra các giá trị tổng (aggregate values) cho từng dòng trong kết quả của phát biểu SELECT.
- *<thuộc tính>*: các thuộc tính tham gia việc phân nhóm dữ liệu.
- Những thuộc tính có tham gia trong thành phần GROUP BY để phân nhóm phải được liệt kê trong danh s
- Mỗi nhóm được mô tả bằng một dòng các thuộc tính, được giới hạn bởi:
 - Các thuộc tính chung của tất cả thành viên thuộc nhóm (được liệt kê trong mệnh đề GROUP BY).
 - Các phép toán trên nhóm.

28

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.5. GROUP BY:



Các thuộc tính GROUP BY

29

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.5. GROUP BY:

- Ví dụ: cho biết tổng lương từng phòng ban

```
SELECT pb.TenPB, 'TL'=SUM(nv.Luong)
FROM NhanVien nv, PhongBan pb
WHERE nv.MaPB = pb.MaPB
GROUP BY pb.TenPB
```

NHANVIEN			
MANV	TENNV	MAPB	LUONG
001	Quyen	QL	3.000.000
002	Tung	NC	2.500.000
003	Nhan	DH	2.500.000
004	Tien	NC	2.200.000
005	Vu	DH	2.200.000
006	Hung	NC	2.000.000
007	Tam	NC	2.200.000
008	Van	NC	1.800.000

nv.MaPB=pb.MaPB

TenPB	Luong
Quan Ly	3000000
Nghien Cuu	2500000
Dieu Hanh	2500000
Nghien Cuu	2200000
Dieu Hanh	2200000
Nghien Cuu	2000000
Nghien Cuu	2200000
Nghien Cuu	1800000

PHONGBAN			
MAPHG	TENPHG	TRPHG	NGNC
QL	Quan Ly	001	22/05/2000
DH	Dieu Hanh	003	10/10/2002
NC	Nghien Cuu	002	15/03/2002

30

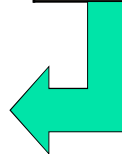
3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.5. GROUP BY:

TenPB	Luong
Quan Ly	3000000
Nghien Cuu	2500000
Dieu Hanh	2500000
Nghien Cuu	2200000
Dieu Hanh	2200000
Nghien Cuu	2000000
Nghien Cuu	2200000
Nghien Cuu	1800000

Gom (sắp
xếp) nhóm

TenPB	Luong
Quan Ly	3000000
Dieu Hanh	2500000
Dieu Hanh	2200000
Nghien Cuu	2500000
Nghien Cuu	2200000
Nghien Cuu	2000000
Nghien Cuu	2200000
Nghien Cuu	1800000

Tính toán
trên từng
nhóm

TenPB	TL
Quan Ly	3000000
Dieu Hanh	4700000
Nghien Cuu	10700000

31

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.6. HAVING:

- Đóng vai trò như một filter dùng để lọc lại các giá trị cần select cho một nhóm hoặc toàn bộ.
- HAVING luôn được sử dụng với mệnh đề GROUP BY.
- <biểu thức điều kiện 2>: là 1 biểu thức so sánh có giá trị kết quả thuộc kiểu Boolean. Trong biểu thức điều kiện này phải có dùng 1 trong các hàm kết hợp.

32

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.6. HAVING: (tt)

- Ví dụ: cho biết tên phòng ban có tổng lương > 3000000

```
SELECT    pb.TenPB, 'TL'=SUM(nv.Luong)
FROM      NhanVien nv, PhongBan pb
WHERE     nv.MaPB = pb.MaPB
GROUP BY  pb.TenPB
HAVING    SUM(nv.Luong) > 3000000
```

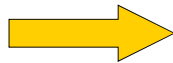
33

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.5. GROUP BY:

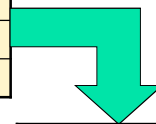
TenPB	Luong
Quan Ly	3000000
Nghien Cuu	2500000
Dieu Hanh	2500000
Nghien Cuu	2200000
Dieu Hanh	2200000
Nghien Cuu	2000000
Nghien Cuu	2200000
Nghien Cuu	1800000

Gom
(sắp xếp)
nhóm



TenPB	Luong
Quan Ly	3000000
Dieu Hanh	2500000
Dieu Hanh	2200000
Nghien Cuu	2500000
Nghien Cuu	2200000
Nghien Cuu	2000000
Nghien Cuu	2200000
Nghien Cuu	1800000

Tính toán trên
từng nhóm



TenPB	TL
Dieu Hanh	4700000
Nghien Cuu	10700000

Lọc bỏ các
nhóm không
thỏa



TenPB	TL
Quan Ly	3000000
Dieu Hanh	4700000
Nghien Cuu	10700000

34

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.7. ORDER BY <thuộc tính> [DESC] :

- Sắp xếp kết quả truy vấn bởi 1 hay nhiều cột (độ ưu tiên giảm dần từ trái sang phải).
- Mặc định là sắp xếp giảm dần, từ khóa DESC dùng để sắp xếp giảm dần.
- Chú ý:
 - Mệnh đề ORDER BY không thể được sử dụng trong các truy vấn con và View.
 - Không thể sử dụng mệnh đề ORDER BY đối với các column có kiểu dữ liệu dạng ntext, text hoặc image

35



3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.8. COMPUTE BY <thuộc tính> [DESC] : Được sử dụng như là một phát sinh kết quả tóm lược cho các column và xuất hiện sau phần trình bày của tập kết quả. Khi được sử dụng kèm với từ khóa BY, mệnh đề COMPUTE kiểm soát các ngắt đoạn và tổng từng phần trong tập kết quả.

- Ví dụ 1:


```
SELECT MaKhoaHoc, MSSV, Diem
FROM KetQua
WHERE MaKhoaHoc IN ('K1','K2')
ORDER BY MaKhoaHoc
COMPUTE AVG(Diem)
```

Ý nghĩa	Kết quả hiển thị
Thông tin chi tiết kết quả của 2 MaKhoaHoc='K1' và 'K2'	
Điểm trung bình của cả 2 MaKhoaHoc='K1' và 'K2'	

36

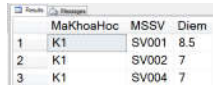

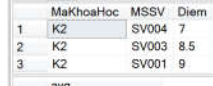

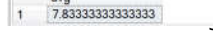
3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

GIẢI THÍCH

3.8. COMPUTE BY <thuộc tính> [DESC] :

- Ví dụ 1:

```
SELECT MaKhoaHoc, MSSV, Diem
FROM KetQua
WHERE MaKhoaHoc IN ('K1','K2')
ORDER BY MaKhoaHoc
COMPUTE AVG(Diem)
COMPUTE AVG(Diem) by MaKhoaHoc
```

Ý nghĩa	Kết quả hiển thị
Thông tin chi tiết kết quả của MaKhoaHoc='K1'	
Điểm trung bình của MaKhoaHoc='K1'	
Thông tin chi tiết kết quả của MaKhoaHoc='K2'	
Điểm trung bình của MaKhoaHoc='K2'	
Điểm trung bình của cả 2 MaKhoaHoc='K1' và 'K2'	

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL)

3.1. Lệnh SELECT FROM (tt)

Hàm tập hợp (Aggregate Function) có thể sử dụng trong quá trình tạo biểu thức (hay điều kiện):

- SUM** (<thuộc tính>): tính tổng giá trị của <thuộc tính> của các bộ trong table.
- MAX** (<thuộc tính>): tính giá trị lớn nhất của <thuộc tính> của các bộ trong table.
- MIN** (<thuộc tính>): tính giá trị nhỏ nhất của <thuộc tính> của các bộ trong table.
- AVG** (<thuộc tính>): tính giá trị trung bình của <thuộc tính> của các bộ trong table.
- COUNT** (*) <thuộc tính> | **DISTINCT** (<thuộc tính>) : đếm các bộ trong table.
 - COUNT** (*) : đếm tất cả các bộ.
 - COUNT** (<thuộc tính>) : chỉ đếm những bộ mà giá trị của <thuộc tính> là khác NULL.
 - COUNT (DISTINCT (<thuộc tính>))**: chỉ đếm những bộ mà giá trị của <thuộc tính> là khác NULL. Đồng thời những bộ mà giá trị trùng nhau trên <thuộc tính> chỉ được đếm 1 (đại diện) cho cả nhóm.

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.2. Truy vấn con

- Các mệnh đề SELECT được lồng vào nhau
- Kết quả của câu SELECT bên trong sẽ được chuyển cho truy vấn bên ngoài.
- Truy vấn ngoài cùng sẽ cho kết quả cuối cùng
- Các mệnh đề SELECT được nối với nhau bằng các phép so sánh <, >, <>, =, >=, <=, IN, EXISTS, ALL, ANY, SOME,... trong mệnh đề **WHERE** hoặc **HAVING**

39

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.2. Truy vấn con (tt)

Ví dụ:

- Sử dụng với WHERE:

```
SELECT TENNHACC, DIENTHOAI
FROM NHACC
WHERE MANHACC IN (SELECT MANHACC FROM DONDH
                  WHERE CONVERT(CHAR(7),NGAYDH,21)='2002-01')
```

- Sử dụng với HAVING:

```
SELECT MaKH
FROM DONDH
GROUP BY MaKH
HAVING COUNT(*) >= ALL ( SELECT COUNT(*)
                        FROM DONDH
                        GROUP BY MaKH
                        HAVING >=5 )
```

40

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.2. Truy vấn con (tt)

- **Quy ước:**

- Truy vấn con phải được đặt trong ngoặc ()
- Chỉ được phép dùng một cột hoặc một biểu thức sẽ trả về giá trị trong mệnh đề SELECT của truy vấn con
- Có thể trả về là một giá trị đơn lẻ hoặc một danh sách các giá trị
- Cấp độ lồng nhau không giới hạn

```
SELECT TENNHACC, DIENTHOAI
FROM NHACC
WHERE MANHACC IN
    (SELECT MANHACC FROM DONDH
     WHERE CONVERT(CHAR(7), NGAYDH, 21)='2002-01')
```

41

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.2. Truy vấn con (tt)

- Ví dụ: Mã số và tên các nhân viên có số giờ làm việc nhiều nhất

```
SELECT nv.MaNV, nv.TenNV, SUM(pc.ThoiGian)
FROM NhanVien nv, PhanCong pc
WHERE nv.MaNV = pc.MaNV
GROUP BY nv.MaNV, nv.TenNV
HAVING SUM(pc.ThoiGian) >= ALL ( SELECT SUM(ThoiGian)
                                FROM PhanCong
                                GROUP BY MaNV )
```

42

3. LỆNH TRUY VẤN DỮ LIỆU (Structured Query Language - SQL) (tt)

3.2. Truy vấn con (tt)

```
SELECT nv.TenNV, SUM(pc.ThoiGian)
FROM NhanVien nv, PhanCong pc
WHERE nv.MaNV = pc.MaNV
GROUP BY nv.TenNV
```

MaNV	TenNV	Tong TG
001	Quyên	42.5
002	Tùng	40
003	Nhan	37.5
004	Tien	32.5
006	Hung	30.5
007	Tam	30
008	Van	22.5

```
SELECT SUM(ThoiGian)
FROM PhanCong
GROUP BY MaNV
```

MaNV	Tong TG
001	42.5
002	40
003	37.5
004	32.5
006	30.5
007	30
008	22.5

HAVING SUM(pc.ThoiGian) >= ALL



MaNV	TenNV	Tong TG
001	Quyên	42.5

43

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP

4.1. Dạng LỚN/NHỎ nhất

4.1.1. Tìm giá trị lớn/nhỏ nhất trong 1 field của table (không quan tâm đến giá trị của các field khác); yêu cầu kết quả trả về của SELECT gồm 1 thông tin về giá trị

Ví dụ:

- Cho biết điểm lớn nhất có trong table kết quả

```
SELECT MAX(Diem)
FROM KetQua
```

- Cho biết điểm lớn nhất của môn học có mã môn học là 'CSDL'

```
SELECT MAX(Diem)
FROM KetQua
WHERE MaMH='CSDL'
```

44

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.1. Dạng LỚN/NHỎ nhất (tt)

4.1.2. Tìm giá trị lớn/nhỏ nhất trong 1 field của table (có quan tâm đến giá trị của các field khác)

- Cho biết điểm lớn nhất của từng sinh viên

```
SELECT MaSV, MAX(Diem)
FROM KetQua
GROUP BY MaSV
```

45

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.1. Dạng LỚN/NHỎ nhất (tt)

4.1.3. Tìm giá trị lớn/nhỏ nhất trong 1 field của table, yêu cầu kết quả trả về của SELECT gồm nhiều thông tin:

- Nhận xét:
 - Các table tham gia trong SubSelect là tập con của các table trong MainSelect
 - Nếu có sử dụng điều kiện chọn trong 2 SELECT (Main và Sub) thì các điều kiện này thường giống nhau
- Dạng câu truy vấn:
 - Sử dụng **MAX** (tìm lớn nhất) hoặc **MIN** (tìm nhỏ nhất)

```
SELECT *
FROM XXX, [các table khác]
WHERE <Tên field>= (SELECT MAX(<Tên field>)
                     FROM XXX)
```
 - Sử dụng **>=ALL**(tìm lớn nhất) hoặc **<=ALL** (tìm nhỏ nhất)

```
SELECT *
FROM XXX, [các table khác]
WHERE <Tên field>=ALL (SELECT <Tên field>
                      FROM XXX)
```

Trong đó: XXX: tên của (1 hoặc nhiều) table sử dụng cho câu truy vấn ⇒ Main và Sub SELECT dùng chung cùng 1 table.

46

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.1. Dạng LỚN/NHỎ nhất (tt)

4.1.3. Tìm giá trị lớn/nhỏ nhất trong 1 field của table, yêu cầu kết quả trả về của SELECT gồm nhiều thông tin: (tt)

i. Chỉ sử dụng 1 table trong Sub Select

1/- Cho biết tên những môn học có số tín chỉ nhiều nhất

Cách 1:

```
SELECT TenMH
FROM MonHoc
WHERE SoTC = (SELECT Max(SoTC)
              FROM MonHoc
              )
```

Cách 2:

```
SELECT TenMH
FROM MonHoc
WHERE SoTC >= ALL ( SELECT SoTC
                    FROM MonHoc
                    )
```

2/- Cho biết mã số, tên, địa chỉ của sinh viên có điểm nhỏ nhất trong table kết quả (không quan tâm điểm của môn học (hay khóa học) nào)

Cách 1:

```
SELECT MaSV, Ten, DiaChi
FROM KetQua K, SinhVien S
WHERE K.MaSV=S.MaSV
AND Diem=( SELECT Min(Diem)
            FROM KetQua
            )
```

Cách 2:

```
SELECT MaSV, Ten, DiaChi
FROM KetQua K, SinhVien S
WHERE K.MaSV=S.MaSV
AND Diem<=ALL( SELECT Diem
                FROM KetQua
                )
```

47

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.1. Dạng LỚN/NHỎ nhất (tt)

4.1.3. Tìm giá trị lớn/nhỏ nhất trong 1 field của table, yêu cầu kết quả trả về của SELECT gồm nhiều thông tin: (tt)

ii. Sử dụng nhiều table trong Sub SELECT

3/- Cho biết mã số, tên, địa chỉ của sinh viên có điểm lớn nhất trong môn học có mã là CSDL

Cách 1:

```
SELECT MaSV, Ten, DiaChi
FROM SinhVien S, Ket Qua KQ,
      GiangDay GD
WHERE S.MaSV= KQ.MaSV
AND KQ.MaKhoaHoc=GD.MaKhoaHoc
AND MaMH= 'CSDL'
AND Diem=(SELECT Max(Diem)
           FROM KetQua K, GiangDay G
           WHERE K.MaKhoaHoc=G.MaKhoaHoc
           AND MaMH= 'CSDL'
           )
```

Cách 2:

```
SELECT MaSV, Ten, DiaChi
FROM SinhVien S, Ket Qua KQ
WHERE S.MaSV= KQ.MaSV
AND MaMH= 'CSDL'
AND Diem >= ALL(SELECT Diem
                FROM KetQua K, GiangDay G
                WHERE MaMH= 'CSDL'
                AND K.MaKhoaHoc=G.MaKhoaHoc
                )
```

48

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.1. Dạng LỚN/NHỎ nhất (tt)

4.1.3. Tìm giá trị lớn/nhỏ nhất trong 1 field của table, yêu cầu kết quả trả về của SELECT gồm nhiều thông tin: (tt)

- iii. Tìm giá trị lớn/nhỏ nhất dựa trên Aggregate Function (điều kiện so sánh được đặt trong mệnh đề HAVING)

Ví dụ: Cho biết mã số và tên của sinh viên có điểm trung bình lớn nhất

```
SELECT S.MaSV, Ten
FROM KetQua, SinhVien S
WHERE KetQua.MaSV= SinhVien.MaSV
GROUP BY S.MaSV, Ten
HAVING AVG(Diem)>=ALL (SELECT AVG(Diem)
                        FROM KetQua
                        GROUP BY MaSV
                        )
```

49

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.2. Dạng KHÔNG/CHƯA có

4.2.1. **Loại 1:** tìm 1 thể hiện ở table đầu 1 sao cho thể hiện này không (hay chưa) có xuất hiện ở table đầu α .

■ **Sử dụng NOT IN**

□ *Dạng thể hiện đơn giản:*

```
SELECT <TênField>
FROM <Table1>                --<Tên_Table phía
đầu 1>
WHERE Table1.PrimaryKey
      NOT IN (SELECT (Table $\alpha$ .ForeignKey
                     FROM <Table $\alpha$ > --<Tên_Table phía
đầu  $\alpha$ >
                     [WHERE <BiểuThứcĐiềuKiện>]
                     )
```

50

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

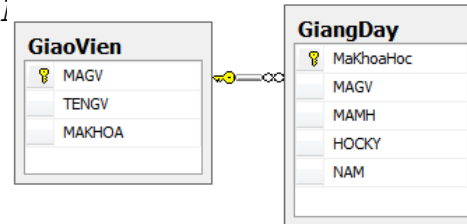
4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.1. **Loại 1**: tìm 1 thể hiện ở table đầu 1 sao cho thể hiện này không (hay chưa) có xuất hiện ở table đầu α . (tt)

■ **Sử dụng NOT IN (tt)**

□ *Ví dụ: Cho biết Tên giáo viên không tham gia giảng dạy trong học kỳ 1 năm 2002*

```
SELECT TenGiaoVien
FROM GiaoVien
WHERE MaGV NOT IN (SELECT MaGV
                    FROM GiangDay
                    WHERE
                        NamHoc=2002
                    )
```



4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.1. **Loại 1**: tìm 1 thể hiện ở table đầu 1 sao cho thể hiện này không (hay chưa) có xuất hiện ở table đầu α . (tt)

■ **Sử dụng LEFT JOIN**

□ *Dạng thể hiện đơn giản:*

```
SELECT <TênField>
FROM <Table1> LEFT JOIN <Table $\alpha$ >
ON <Table1>.PrimaryKey= <Table $\alpha$ >.ForeignKey
WHERE <Table $\alpha$ >.ForeignKey IS NULL
-- Table1 : Tên_Table phía đầu 1
-- Table $\alpha$  : Tên_Table phía đầu
```

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

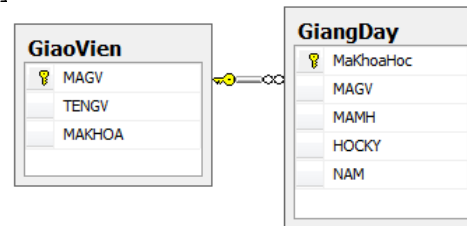
4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.1. **Loại 1**: tìm 1 thể hiện ở table đầu 1 sao cho thể hiện này không (hay chưa) có xuất hiện ở table đầu α . (tt)

- **Sử dụng LEFT JOIN (tt)**

□ *Ví dụ: Cho biết Tên giáo viên không tham gia giảng dạy trong học kỳ 1 năm 2002*

```
SELECT TenGiaoVien
FROM GiaoVien GV
      LEFT JOIN GiangDay GD ON GV.MaGV=GD.MaGV
WHERE HocKy=1 AND NamHoc=2002
      AND GD.MaGV IS NULL
```



4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.2. **Loại 2**: tìm 1 thể hiện ở table đầu 1 khi thể hiện này đã có xuất hiện ở table đầu α , nhưng những thể hiện ở table đầu α lại không thỏa với điều kiện đưa ra.

- **Sử dụng NOT IN**

□ *Dạng thể hiện đơn giản:*

```
SELECT <TênField>
FROM <Table1>
WHERE Table1.PrimaryKey
      NOT IN (SELECT Table $\alpha$ .ForeignKey
              FROM <Table $\alpha$ >
              [WHERE <Biểu thức điều kiện>])
```

-- Table1 : Tên_Table phía đầu 1

-- Table α : Tên_Table phía đầu α

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

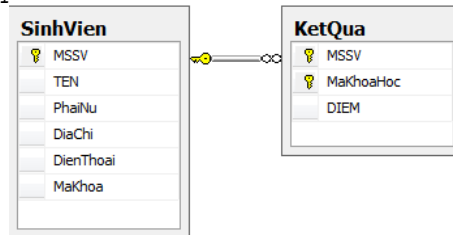
4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.2. **Loại 2:** tìm 1 thể hiện ở table đầu 1 khi thể hiện này đã có xuất hiện ở table đầu α , nhưng những thể hiện ở table đầu α lại không thỏa với điều kiện đưa ra. (tt)

■ Sử dụng NOT IN (tt)

□ **Ví dụ:** Cho biết Mã SV có kết quả các môn học đều trên 8 điểm

```
SELECT MaSV
FROM SinhVien
WHERE MaSV NOT IN (SELECT MASV
                    FROM KetQua
                    WHERE Diem < 8)
```



4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.2. **Loại 2:** tìm 1 thể hiện ở table đầu 1 khi thể hiện này đã có xuất hiện ở table đầu α , nhưng những thể hiện ở table đầu α lại không thỏa với điều kiện đưa ra. (tt)

■ Sử dụng LEFT JOIN

□ **Dạng thể hiện đơn giản:**

```
SELECT <TênField>
FROM <Table1> LEFT JOIN (SELECT *
                        FROM <Table $\alpha$ >
                        WHERE <BiểuThứcĐiềuKiện>)
ON Table1.PrimaryKey= Table $\alpha$ .ForeignKey
WHERE Table $\alpha$ .ForeignKey IS NULL

-- Table1 : Tên_Table phía đầu 1
-- Table $\alpha$  : Tên_Table phía đầu
```

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.2. Dạng KHÔNG/CHƯA có (tt)

4.2.2. **Loại 2:** tìm 1 thể hiện ở table đầu 1 khi thể hiện này đã có xuất hiện ở table đầu α , nhưng những thể hiện ở table đầu α lại không thỏa với điều kiện đưa ra. (tt)

■ Sử dụng LEFT JOIN (tt)

□ **Ví dụ:** Cho biết Tên giáo viên không tham gia giảng dạy trong học kỳ 1 năm 2002

```
SELECT TenGiaoVien
FROM GiaoVien GV LEFT JOIN (SELECT *
                             FROM Giangday
                             WHERE HocKy=1 AND NamHoc=2002) As GD
ON GV.MaGV=GD.MaGV
WHERE GD.MaGV IS NULL
```

57

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)

4.3. Dạng TẤT CẢ (phép chia trong đại số quan hệ)

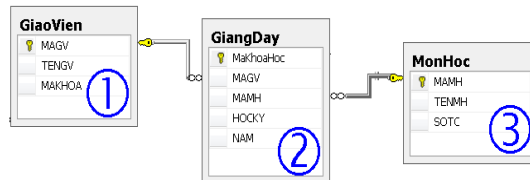
4.3.1. **Loại 1:** gồm 3 nhóm table

■ Nhận xét: Câu truy vấn sử dụng 1 bộ gồm 3 nhóm table:

- Nhóm 1: là table chứa dữ liệu kết quả của câu truy vấn. Nhóm này thường chỉ gồm 1 table và thường là table dạng danh mục.
- Nhóm 3: là table chứa dữ liệu cần kiểm tra của câu truy vấn (Tên giáo viên đã dạy đủ tất cả các môn học, tên học sinh đã học đủ tất cả các môn học, ...) Nhóm này thường cũng chỉ gồm 1 table và thuộc dạng danh mục.
- Nhóm 2: Nhóm này gồm những table nằm "kẹp" giữa nhóm 1 và nhóm 3, chứa khóa ngoại của 2 table nhóm 1 và nhóm 2. Là table chứa dữ liệu liên kết giữa 2 nhóm 1 và 3.

■ Dạng câu truy vấn:

```
SELECT *
FROM ①
WHERE NOT EXISTS (SELECT *
                  FROM ③
                  WHERE NOT EXISTS (SELECT *
                                    FROM ②
                                    WHERE
                                      ②.ForeignKey2=③.PrimaryKey
                                      AND ②.ForeignKey1=①.PrimaryKey ) )
```

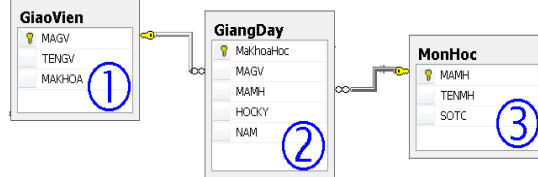


58

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)
 4.3. Dạng TẮT CẮ (phép chia trong đại số quan hệ) (tt)
 4.3.1. **Loại 1:** gồm 3 nhóm table

- **VD 1: Nhóm 2 gồm 1 table:** Giả sử mỗi giáo viên đều có khả năng dạy tất cả các môn học (có trong table môn học). Cho biết mã số và tên của những giáo viên đã dạy đủ tất cả các môn học

```
SELECT TenGV
FROM GiaoVien V
WHERE NOT EXISTS (SELECT *
                  FROM MonHoc M
                  WHERE NOT EXISTS (SELECT *
                                   FROM GiangDay G
                                   WHERE G.MaGV=V.MaGV
                                       AND M.MaMH=G.MaMH
                                   )
                  )
```



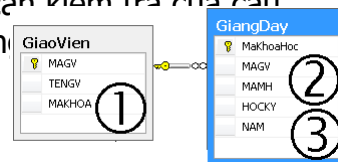
59

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)
 4.3. Dạng TẮT CẮ (phép chia trong đại số quan hệ) (tt)
 4.3.2. **Loại 2:** gồm 2 nhóm table

- **Nhân xét:** Thực ra dạng này cũng gồm 3 nhóm table, nhưng trong đó nhóm 2 và nhóm 3 sử dụng table trùng nhau:
 - Nhóm 1: là table chứa dữ liệu kết quả của câu truy vấn (Tên giáo viên, tên học sinh, ...). Nhóm này thường chỉ gồm 1 table và là table dạng danh mục.
 - Nhóm 3: là table chứa dữ liệu cần kiểm tra của câu truy vấn (chứa điều kiện chính của phép chọn).
 - Nhóm 2: là table chứa dữ liệu cần kiểm tra của câu truy vấn. Nhóm này thường cũng trong nhóm 3.

- **Dạng câu truy vấn:**

```
SELECT *
FROM ①
WHERE NOT EXISTS (SELECT *
                  FROM ②a
                  WHERE NOT EXISTS (SELECT *
                                   FROM ②
```



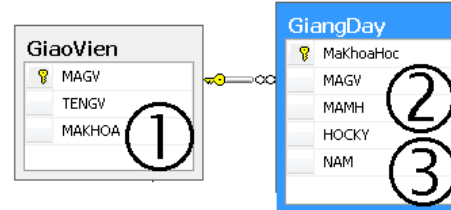
60

4. MỘT SỐ DẠNG TRUY VẤN THƯỜNG GẶP (tt)
 4.3. Dạng TẤT CẢ (phép chia trong đại số quan hệ) (tt)
 4.3.2. **Loại 2:** gồm 2 nhóm table

- **Ví dụ:** Cho biết mã số và tên những giáo viên đã dạy đủ tất cả các môn học mà giáo viên 'GV01' đã dạy (kết quả không hiển thị giáo viên GV01)

```
SELECT MaGV, TenGV
FROM GiaoVien V
WHERE MaGV <> 'GV01'
```

```
AND NOT EXISTS (SELECT *
                  FROM GiangDay G
                  WHERE MaGV='GV01'
                  AND NOT EXISTS (SELECT *
                                  FROM GiangDay GD
                                  WHERE GD.MaMH=G.MaMH
                                  AND GD.MAGV= V.MaGV
                                  )
                  )
```



61

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL

- 5.1. Cho biết tên, địa chỉ, điện thoại của tất cả các sinh viên

■ $KQ \leftarrow \Pi_{Ten, DiaChi, DienThoai} (SinhVien)$

```
SELECT Ten, DiaChi, DienThoai
FROM SinhVien
```

- 5.2. Cho biết tên và số tín chỉ của các môn học

■ $KQ \leftarrow \Pi_{TenMH, SoTC} (MonHoc)$

```
SELECT TenMH, SoTC
FROM MonHoc
```

- 5.3. Cho biết kết quả học tập của sinh viên có mã số SV03

■ $KQ \leftarrow \Pi_{MaKhoaHoc, Diem} (\delta_{MSSV='SV03'}(KetQua))$

```
SELECT MaKhoaHoc, Diem
FROM KetQua
WHERE MSSV='SV03'
```

62

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.4. Cho biết tên và số tín chỉ của các môn học có mã có dạng: ký tự thứ 1 là C, ký tự thứ 3 là D

■ $KQ \leftarrow \Pi_{TenMH, SoTC} (\delta_{MaMH \text{ LIKE 'C?D*'}} (MonHoc))$

```
SELECT      TenMH, SoTC
FROM      MonHoc
WHERE     MaMH LIKE 'C?D*'
```

5.5. Cho biết danh sách các môn học được dạy trong năm 2002

■ $KQ \leftarrow \Pi_{MaMH} (\delta_{Nam=2002} (GiangDay))$

```
SELECT      MaMH
FROM      GiangDay
WHERE     Nam=2002
```

5.6. Cho biết mã, tên, địa chỉ của các sv theo từng khoa, sắp A-Z theo tên sv

■ $KQ \leftarrow \Pi_{MaKhoa, MSSV, Ten, DiaChi} (SinhVien)$

```
SELECT      MaKhoa, MSSV, Ten, DiaChi
FROM      SinhVien
ORDER BY    MaKhoa ASC, Ten ASC
```

63
63

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.7. Cho biết điểm của các sv theo từng môn học

■ $KQ \leftarrow \Pi_{MSSV, MaMH, Diem} (KetQua \bowtie GiangDay)$

```
SELECT      MSSV, MaMH, Diem
FROM      KetQua kq, GiangDay gd
WHERE     kq.MaKhoaHoc = gd.MaKhoaHoc
ORDER BY    MaMH
```

5.8. Cho biết các sv học môn CSDL có điểm từ 8 đến 10

■ $R1 \leftarrow \delta_{MaMH='CSDL'} (GiangDay)$

■ $R2 \leftarrow R1 \bowtie_{MaKhoaHoc} KetQua$

■ $KQ \leftarrow \Pi_{MSSV} (\delta_{8 \leq Diem \leq 10} (R2))$

```
SELECT      MSSV
FROM      KetQua kq, MyView v
WHERE     kq.MaKhoaHoc=v.MaKhoaHoc AND
          (Diem BETWEEN 8 AND 10)
```

```
CREATE VIEW MyView AS
SELECT      *
FROM      GiangDay
WHERE     MaMH='CSDL'
```

64
64

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.8. Cho biết các sv học môn CSDL có điểm từ 8 đến 10

- $R1 \leftarrow \sigma_{MaMH='CSDL'}(GiangDay)$
 - $R2 \leftarrow R1 \bowtie_{MaKhoaHoc} KetQua$
 - $KQ \leftarrow \Pi_{MSSV}(\sigma_{8 \leq Diem \leq 10}(R2))$
- ```
SELECT MSSV
FROM KetQua kq, GiangDay gd
WHERE kq.MaKhoaHoc=gd.MaKhoaHoc AND
 MaMH='CSDL' AND Diem BETWEEN 8 AND 10)
```

#### 5.9. Cho biết tên khoa, tên môn học mà các sv trong khoa đã học

- $R1 \leftarrow \Pi_{TenKhoa, MSSV}(Khoa \bowtie_{MaKhoa} SinhVien)$
  - $R2 \leftarrow \Pi_{TenKhoa, MaKhoaHoc}(R1 \bowtie_{MSSV} KetQua)$
  - $R3 \leftarrow \Pi_{TenKhoa, MaMH}(R2 \bowtie_{MaKhoaHoc} GiangDay)$
  - $KQ \leftarrow \Pi_{TenKhoa, TenMH}(R3 \bowtie_{MaMH} MonHoc)$
- ```
SELECT      TenKhoa, TenMH
FROM        Khoa k, SinhVien sv, KetQua kq,
            GiangDay gd, MonHoc mh
WHERE k.MaKhoa=sv.MaKhoa AND sv.MSSV=kq.MSSV AND
      kq.MaKhoaHoc=gd.MaKhoaHoc AND gd.MaMH=mh.MaMH
```

65

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.10. Cho biết mã, tên các sv có kết quả một môn học nào đó trên 8

- $R1 \leftarrow \Pi_{MSSV}(\sigma_{Diem > 8}(KetQua))$
 - $KQ \leftarrow \Pi_{MSSV, Ten}(R1 \bowtie_{MSSV} SinhVien)$
- ```
SELECT sv.MSSV, Ten
FROM KetQua kq, SinhVien sv
WHERE kq.MSSV=sv.MSSV AND Diem > 8
```

#### 5.11. Cho biết mã, tên các sv có kết quả các môn học đều trên 8

- $R1 \leftarrow \Pi_{MSSV}(\sigma_{Diem \leq 8}(KetQua))$
  - $R2 \leftarrow (\Pi_{MSSV}(SinhVien)) - R1$
  - $KQ \leftarrow \Pi_{MSSV, Ten}(R2 \bowtie_{MSSV} SinhVien)$
- ```
SELECT      MSSV, Ten
FROM        SinhVien
WHERE MSSV NOT IN ( SELECT      MSSV
                    FROM        KetQua
                    WHERE        Diem <= 8
                    )
```

66

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.12. Cho biết có bao nhiêu sv

- $KQ(SLSV) \leftarrow \mathfrak{S}_{COUNT(MSSV)}(SinhVien)$

```
SELECT 'SLSV'=COUNT(MSSV)
FROM SinhVien
```

5.13. Cho biết có bao nhiêu sv nữ khoa CNTT

- $R1 \leftarrow \delta_{PhaiNu='Yes' \wedge MaKhoa='CNTT'}(SinhVien)$

- $KQ(SLSV) \leftarrow \mathfrak{S}_{COUNT(MSSV)}(R1)$

```
SELECT 'SLSV'=COUNT(MSSV)
FROM SinhVien
WHERE PhaiNu='Yes' and MaKhoa='CNTT'
```

5.14. Cho biết mã, tên, địa chỉ và điểm trung bình của từng sv

- $R1(MSSV, DTB) \leftarrow MSSV \mathfrak{S}_{AVG(Diem)}(KetQua)$

- $KQ \leftarrow \Pi_{MSSV, Ten, DiaChi, DTB}(R1 \bowtie_{MSSV} SinhVien)$

```
SELECT sv.MSSV, Ten, DiaChi, 'DTB'=AVG(Diem)
FROM KetQua kq, SinhVien sv
WHERE kq.MSSV=sv.MSSV
GROUP BY sv.MSSV, Ten, DiaChi
```

67
67

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.15. Cho biết số lượng điểm ≥ 8 của từng sv

- $R1 \leftarrow \delta_{Diem \geq 8}(KetQua)$

- $R2(MSSV, SLDT8) \leftarrow MSSV \mathfrak{S}_{COUNT(Diem)}(R1)$

- $KQ \leftarrow \Pi_{MSSV, SLDT8}(SinhVien \bowtie_{MSSV} R2)$

```
CREATE VIEW view2 AS
SELECT MSSV, 'SLDT8'=COUNT(Diem)
FROM KetQua
WHERE Diem >= 8
GROUP BY MSSV
```

```
SELECT SinhVien.MSSV, Ten, SLDT8
FROM SinhVien LEFT JOIN view2 on
SinhVien.MSSV=view2.MSSV
```

5.16. Cho biết số lượng tín chỉ của từng sv

- $R1 \leftarrow \Pi_{MSSV, MaMH}(KetQua \bowtie_{MaKhoaHoc} GiangDay)$

- $R2 \leftarrow \Pi_{MSSV, SoTC}(R1 \bowtie_{MaMH} MonHoc)$

- $R3(MSSV, TSTC) \leftarrow MSSV \mathfrak{S}_{SUM(SoTC)}(R2)$

- $KQ \leftarrow \Pi_{MSSV, Ten, TSTC}(R3 \bowtie_{MSSV} SinhVien)$

```
SELECT sv.MSSV, Ten, 'TSTC'=SUM(SoTC)
FROM SinhVien sv, KetQua kq, GiangDay gd, MonHoc mh
WHERE sv.MSSV=kq.MSSV AND kq.MaKhoaHoc=gd.MaKhoaHoc
AND gd.MaMH=mh.MaMH
GROUP BY sv.MSSV, Ten
```

68
68

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.17. Cho biết tên những sv chỉ mới thi đúng 1 môn học

- $R1 \leftarrow \Pi_{MSSV, MaMH} (KetQua \bowtie_{MaKhoaHoc} GiangDay)$
- $R2(MSSV, SLMH) \leftarrow MSSV \Join_{COUNT(MaMH)}(R1)$
- $KQ \leftarrow \Pi_{Ten} (\delta_{SLMH=1} (R2 \bowtie_{MSSV} SinhVien))$

```
CREATE VIEW view3 AS
SELECT      Distinct MSSV, MaMH
FROM      KetQua kq, GiangDay gd
WHERE      kq.MaKhoaHoc = gd.MaKhoaHoc
```

```
SELECT      Ten
FROM      SinhVien sv, view3 v
WHERE      sv.MSSV=v.MSSV
GROUP BY      Ten
HAVING      COUNT (MaMH)=1
```

5.18. Cho biết tên môn học có số tín chỉ nhiều nhất

- $R1(SoTCNN) \leftarrow \Join_{MAX(SoTC)} (MonHoc)$
- $KQ \leftarrow \Pi_{TenMH} (R1 \bowtie_{SoTCNN=SoTC} MonHoc)$

```
SELECT      MaMH, TenMH, SoTC
FROM      MonHoc
WHERE      SoTC = (SELECT      MAX (SoTC)
                  FROM      MonHoc)
```

69
69

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.19. Cho biết mã, tên, địa chỉ của các sv có điểm CSDL lớn nhất

- $R1 \leftarrow \delta_{MaMH='CSDL'} (KetQua \bowtie_{MaKhoaHoc} GiangDay)$
- $R2 \leftarrow \Pi_{MSSV, Diem} (R1)$
- $R3(DCSDLN) \leftarrow \Join_{MAX(Diem)} (R2)$
- $R4 \leftarrow \Pi_{MSSV} (R3 \bowtie_{DCSDLN=Diem} R2)$
- $KQ \leftarrow \Pi_{MSSV, Ten, DiaChi} (R4 \bowtie_{MSSV} SinhVien)$

```
SELECT      sv.MSSV, Ten, DiaChi
FROM      KetQua kq, GiangDay gd, SinhVien sv
WHERE      kq.MaKhoaHoc=gd.MaKhoaHoc AND
            kq.MSSV=sv.MSSV AND gd.MaMH='CSDL' AND
            Diem = (SELECT      MAX(Diem)
                  FROM      KetQua kq, GiangDay gd
                  WHERE      kq.MaKhoaHoc=gd.MaKhoaHoc AND gd.MaMH='CSDL')
```

70
70

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.20. Với mỗi môn học, cho biết tên, điểm của các sv có điểm lớn nhất

- $R1 \leftarrow \Pi_{MaMH, Diem, MSSV} (KetQua \bowtie_{MaKhoaHoc} GiangDay)$
- $R2(MaMH, DLN) \leftarrow MaMH \Join_{MAX(Diem)} (R1)$
- $R3 \leftarrow \Pi_{MaMH, MSSV, Diem} (R2 \bowtie_{DLN=Diem \wedge R1.MaMH=R2.MaMH} R1)$
- $KQ \leftarrow \Pi_{MaMH, Ten, Diem} (R3 \bowtie_{MSSV} SinhVien)$

```
Create View view4 as
SELECT      MaMH, 'DLN'=MAX(Diem)
FROM      KetQua kq, GiangDay gd
WHERE      kq.MaKhoaHoc=gd.MaKhoaHoc
GROUP BY  MaMH
```

```
SELECT      v.MaMH, Ten, Diem
FROM      KetQua kq, GiangDay gd,
          view4 v, SinhVien sv
WHERE      kq.MaKhoaHoc=gd.MaKhoaHoc AND
          sv.MSSV=kq.MSSV AND
          gd.MaMH=v.MaMH AND
          kq.Diem=v.DLN
```

71
71

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.20. Với mỗi môn học, cho biết tên, điểm của các sv có điểm lớn nhất

- $R1 \leftarrow \Pi_{MaMH, Diem, MSSV} (KetQua \bowtie_{MaKhoaHoc} GiangDay)$
- $R2(MaMH, DLN) \leftarrow MaMH \Join_{MAX(Diem)} (R1)$
- $R3 \leftarrow \Pi_{MaMH, MSSV, Diem} (R2 \bowtie_{DLN=Diem \wedge R1.MaMH=R2.MaMH} R1)$
- $KQ \leftarrow \Pi_{MaMH, Ten, Diem} (R3 \bowtie_{MSSV} SinhVien)$

```
SELECT gd.MaMH, Ten, Diem
FROM KetQua kq, GiangDay gd, SinhVien sv
WHERE kq.MaKhoaHoc=gd.MaKhoaHoc AND
      sv.MSSV=kq.MSSV AND
      Diem = ( SELECT      MAX(Diem)
                FROM      KetQua kq, GiangDay gd1
                WHERE      kq.MaKhoaHoc=gd1.MaKhoaHoc AND
                          gd1.MaMH=gd.MaMH
              )
```

72
72

5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

5.21. Tên các giáo viên không tham gia giảng dạy năm 2001

- $R1 \leftarrow \Pi_{MaGV} (\delta_{Nam=2001} GiangDay)$
 - $R2 \leftarrow (\Pi_{MaGV} (GiaoVien)) - R1$
 - $KQ \leftarrow \Pi_{TenGV} (R2 \bowtie_{MaGV} GiaoVien)$
- ```
SELECT TenGV
FROM GiaoVien
WHERE MaGV NOT IN (SELECT MaGV
 FROM GiangDay
 WHERE Nam=2001)
```

### 5.22. Tên sv, tên môn học mà sv chưa học

- $R1 \leftarrow \Pi_{MSSV, MaMH} (SinhVien \times MonHoc)$
- $R2 \leftarrow \Pi_{MSSV, MaMH} (KetQua \bowtie_{MaKhoaHoc} GiangDay)$
- $R3 \leftarrow (R1 - R2)$
- $KQ \leftarrow \Pi_{Ten, TenMH} (R3 \bowtie_{MSSV} SinhVien) \bowtie_{MaGV} GiaoVien)$

```
SELECT Ten, TenMH
FROM SinhVien sv, MonHoc
WHERE MaMH NOT IN (SELECT MaMH
 FROM KetQua kq, GiangDay gd
 WHERE kq.MaKhoaHoc=gd.MaKhoaHoc
 AND kq.MSSV=sv.MSSV)
```

73  
73

## 5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

### 5.22. Tên các giáo viên tham gia dạy tất cả môn học

- $R \leftarrow \Pi_{MaGV, MaMH} (GiangDay)$
- $S \leftarrow \Pi_{MaMH} (MonHoc)$
- $R3 \leftarrow R \div S$
- $KQ \leftarrow \Pi_{TenGV} (R3 \bowtie_{MaGV} GiaoVien)$

```
SELECT TenGV
FROM GiaoVien gv
WHERE NOT EXISTS (SELECT *
 FROM MonHoc mh
 WHERE NOT EXISTS (SELECT *
 FROM GiangDay gd
 WHERE gd.MaMH=mh.MaMH
 AND gd.MaGV=gv.MaGV))
```

74  
74

## 5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

## 5.23. Tên các sv đã học tất cả môn học

- $R \leftarrow \Pi_{MSSV, MaMH} (GiangDay \bowtie_{MaKhoaHoc} KetQua)$
- $S \leftarrow \Pi_{MaMH} (MonHoc)$
- $R3 \leftarrow R \div S$
- $KQ \leftarrow \Pi_{Ten} (R3 \bowtie_{MSSV} SinhVien)$

```

SELECT Ten
FROM SinhVien sv
WHERE NOT EXISTS (SELECT *
 FROM MonHoc mh
 WHERE NOT EXISTS (SELECT *
 FROM KetQua kq, GiangDay gd
 WHERE kq.MaKhoaHoc=gd.MaKhoaHoc
 AND gd.MaMH=mh.MaMH
 AND kq.MSSV=sv.MSSV))

```

75  
75

## 5. MINH HỌA VỀ ĐẠI SỐ QUAN HỆ VÀ SQL (tt)

## 5.24. Tên giáo viên đã dạy tất cả môn do giáo viên GV03 dạy

- $R \leftarrow \Pi_{MaGV, MaMH} (GiangDay)$
- $S \leftarrow \Pi_{MaMH} (\delta_{MaGV='GV03'} GiangDay)$
- $R3 \leftarrow R \div S$
- $KQ \leftarrow \Pi_{TenGV} (R3 \bowtie_{MaGV} GiaoVien)$

```

SELECT Ten
FROM SinhVien sv
WHERE NOT EXISTS (SELECT *
 FROM GiangDay gd
 WHERE MaGV='GV03' AND NOT EXISTS (
 SELECT *
 FROM KetQua kq, GiangDay gd1
 WHERE kq.MaKhoaHoc=gd1.MaKhoaHoc
 AND gd1.MaMH=gd.MaMH
 AND kq.MSSV=sv.MSSV
)
)

```

76  
76