Chương 4

Ngôn ngữ truy vấn SQL

GV: Ths. Bùi Hữu Đông

Nội dung

1. Giới thiệu
2. Ngôn ngữ định nghĩa dữ liệu
3. Ngôn ngữ thao tác dữ liệu
4. Ngôn ngữ truy vấn dữ liệu

1. Giới thiệu

SQL (Structured Query Language)

- Là ngôn ngữ chuẩn cho truy vấn và thao tác trên CSDL quan hệ
- Ngôn ngữ truy vấn khai báo không thủ tục (non-procedure). Một truy vấn SQL mô tả điều mình cần (what), không cần nêu cách thức có được nó (how)
- Được phát triển bởi IBM (1970s)
- Ban đầu được gọi là SEQUEL
- Được ANSI công nhận và phát triển thành chuẩn
  - SQL-86
  - SQL-89
  - SQL-92
  - SQL-99
  - SQL-2003
  - SQL-2006
  - SQL-2008
  - SQL-2011

1. Giới thiệu

- **SQL gồm**
  - Định nghĩa dữ liệu (DDL)
  - Thao tác dữ liệu (DML)
  - Truy vấn dữ liệu
- **SQL sử dụng thuật ngữ**
  - Bảng ~ quan hệ
  - Cột ~ thuộc tính
  - Dòng ~ bộ
- **SQL mang ngữ nghĩa *bag***
  - Các biểu thức SQL có thể trả về các bộ trùng, nếu ta không loại bỏ chúng một cách tường minh

Nội dung

1. **Giới thiệu**
2. **Ngôn ngữ định nghĩa dữ liệu**
   - Tạo bảng
   - Khai báo RBTV
   - Sửa cấu trúc bảng
   - Xóa bảng
3. **Ngôn ngữ thao tác dữ liệu**
4. **Ngôn ngữ truy vấn dữ liệu**

2. Ngôn ngữ định nghĩa dữ liệu

- **Là ngôn ngữ 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

- **Gồm**
  - CREATE TABLE (tạo bảng)
  - DROP TABLE (xóa bảng)
  - ALTER TABLE (sửa bảng)
  - CREATE DOMAIN (tạo miền giá trị)
  - CREATE DATABASE (tạo cơ sở dữ liệu)
  - ...

Kiểu dữ liệu

Số	tinyint, smallint, int, numeric(m,n), decimal(m,n), float, real, smallmoney, money bit
Chuỗi ký tự	varchar(n), char(n), nvarchar(n), nchar(n)
Ngày tháng	smalldatetime, datetime
Chuỗi nhị phân	binary, varbinary
…	

2.1 Tạo bảng

- **Để định nghĩa một bảng**
  - Tên bảng
  - Các thuộc tính
    - Tên thuộc tính
    - Kiểu dữ liệu
    - Các RBTV trên thuộc tính
- **Cú pháp:**

```
CREATE TABLE <Tên_bảng>
(
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
    …
    [<RBTV>]
)
```

Ví dụ tạo bảng

```
CREATE TABLE NHANVIEN
(
    MANV CHAR(9),
    HONV VARCHAR(10),
    TENLOT VARCHAR(20),
    TENNV VARCHAR(10),
    NGSINH DATETIME,
    DCHI VARCHAR(50),
    PHAI CHAR(3),
    LUONG INT,
    MA_NQL CHAR(9),
    PHG INT
)
```

2.2 Khai báo RBTV

- **<RBTV>**
  - NOT NULL
  - NULL
  - UNIQUE tên thuộc tính
  - DEFAULT (giá trị mặc định)
  - PRIMARY KEY (các thuộc tính khóa chính)
  - FOREIGN KEY / REFERENCES
  - CHECK (tên thuộc tính điều kiện)

- **Đặt tên cho RBTV**
  - **CONSTRAINT** <Ten_RBTV> <RBTV>

Ví dụ

```
CREATE TABLE NHANVIEN (
        HONV VARCHAR(10) NOT NULL,
        TENLOT VARCHAR(20) NOT NULL,
        TENNV VARCHAR(10) NOT NULL,
        MANV CHAR(9) PRIMARY KEY,
        NGSINH DATETIME,
        DCHI VARCHAR(50),
        PHAI CHAR(3) CHECK (PHAI IN ('Nam', 'Nu')),
        LUONG INT DEFAULT (10000),
        MA_NQL CHAR(9),
        PHG INT
)
```

Ví dụ

```
CREATE TABLE PHONGBAN (
    TENPB VARCHAR(20) UNIQUE,
    MAPHG INT NOT NULL,
    TRPHG CHAR(9),
    NG_NHANCHUC DATETIME DEFAULT(GETDATE())
)

CREATE TABLE PHANCONG (
    MA_NVIEN CHAR(9) FOREIGN KEY (MA_NVIEN)
            REFERENCES NHANVIEN(MANV),
    SODA INT REFERENCES DEAN(MADA),
    THOIGIAN DECIMAL(3,1)
)
```

Ví dụ đặt tên RBTV

```
CREATE TABLE NHANVIEN (
        HONV VARCHAR(10) CONSTRAINT NN_NV_HONV NOT NULL,
        TENLOT VARCHAR(20) NOT NULL,
        TENNV VARCHAR(10) NOT NULL,
        MANV CHAR(9) CONSTRAINT PK_NV_MANV PRIMARY KEY,
        NGSINH DATETIME,
        DCHI VARCHAR(50),
        PHAI CHAR(3) CONSTRAINT CK_NV_PHAI
              CHECK (PHAI IN ('Nam', 'Nu')),
        LUONG INT CONSTRAINT DF_NV_LUONG DEFAULT (10000),
        MA_NQL CHAR(9),
        PHG INT
)
```

Ví dụ đặt tên RBTV

```
CREATE TABLE PHANCONG (
      MA_NVIEN CHAR(9),
      SODA INT,
      THOIGIAN DECIMAL(3,1),
      CONSTRAINT PK_PC_MANVIEN_SODA PRIMARY KEY
            (MA_NVIEN, SODA),
      CONSTRAINT FK_PC_MANVIEN FOREIGN KEY (
            MA_NVIEN) REFERENCES NHANVIEN(MANV),
      CONSTRAINT FK_PC_SODA FOREIGN KEY (SODA)
            REFERENCES DEAN(MADA)
)
```

2.3 Sửa bảng

- **Được dùng để**
  - Thay đổi cấu trúc bảng
  - Thay đổi RBTV
- **Thêm cột**

ALTER TABLE <Tên_bảng>

ADD <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>]

- **Xóa cột**

ALTER TABLE <Tên_bảng>

DROP COLUMN <Tên_cột>

- **Sửa cột**

ALTER TABLE <Tên_bảng>

ALTER COLUMN <Tên_cột> <Kiểu_dữ_liệu_mới>

2.3 Sửa bảng (tt)

◆ Thêm RBTV

```
ALTER TABLE <Tên_bảng> ADD
        CONSTRAINT <Ten_RBTV> <RBTV>,
        CONSTRAINT <Ten_RBTV> <RBTV>,
    …
```

◆ Xóa RBTV

```
ALTER TABLE <Tên_bảng> DROP <Tên_RBTV>
```

Ví dụ thay đổi cấu trúc bảng

- ```
  ALTER TABLE NHANVIEN ADD NGHENGHIEP CHAR(20)
  ```
- ```
  ALTER TABLE NHANVIEN ALTER COLUMN
      NGHENGHIEP CHAR(50)
  ```
- ```
  ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP
  ```
  ```

Ví dụ thay đổi RBTV

```
CREATE TABLE PHONGBAN (
    TENPB VARCHAR(20),
    MAPHG INT NOT NULL,
    TRPHG CHAR(9),
    NG_NHANCHUC DATETIME
)
ALTER TABLE PHONGBAN ADD
    CONSTRAINT PK_PB_MAPHG PRIMARY KEY (MAPHG),
    CONSTRAINT FK_PB_TRPHG FOREIGN KEY (TRPHG)
              REFERENCES NHANVIEN(MANV),
    CONSTRAINT DF_PB_NGNHANCHUC DEFAULT (GETDATE())
              FOR (NG_NHANCHUC),
    CONSTRAINT UNI_PB_TENPB UNIQUE (TENPB)
```

2.4 Xóa bảng

- **Được dùng để xóa cấu trúc bảng**
  - Tất cả dữ liệu của bảng cũng bị xóa

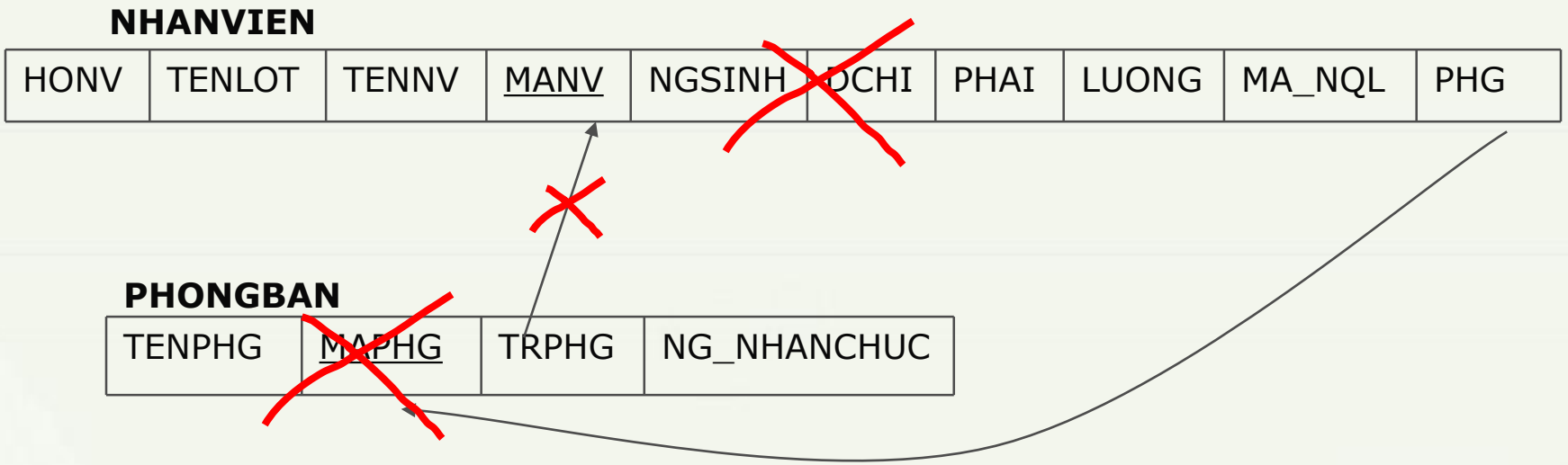
- **Cú pháp**

  DROP TABLE <Tên_bảng>

- **Ví dụ**
  - `DROP TABLE NHANVIEN`
  - `DROP TABLE PHONGBAN`
  - `DROP TABLE PHANCONG`

2.4 Xóa bảng

- Khi muốn xóa một bảng phải xóa tất cả những khóa ngoại tham chiếu tới bảng đó trước.



Tạo miền giá trị

- Tạo ra một kiểu dữ liệu mới kế thừa những kiểu dữ liệu có sẵn
- Cú pháp

CREATE DOMAIN <Tên_kdl_mới> AS <Kiểu_dữ_liệu>

[**DEFAULT** defaultValue]

[**CHECK** (condition)

- Ví dụ
  - `CREATE DOMAIN Kieu_Ten AS VARCHAR(30)`
  - `CREATE DOMAIN t_cvu AS CHAR(2)`
    `DEFAULT 'KT'`
    `CHECK(VALUE IN(NULL,'TK','LT','PT','KT'))`

Nội dung

1. **Giới thiệu**
2. **Ngôn ngữ định nghĩa dữ liệu**
3. **Ngôn ngữ thao tác dữ liệu**
   - Thêm dữ liệu
   - Sửa dữ liệu
   - Xóa dữ liệu
4. **Ngôn ngữ truy vấn dữ liệu**

3.1 Thêm dữ liệu

- **Cú pháp**

  INSERT INTO <tên bảng>(<danh sách các thuộc tính>)

  VALUES (<danh sách các giá trị>)

- **Ví dụ:**
  - `insert into SANPHAM values('BC01','But chi', 'cay', 'Singapore', 3000)`
  - `insert into SANPHAM(masp,tensp,dvt,nuocsx,gia) values ('BC01','But chi', 'cay', 'Singapore', 3000)`

3.2 Xóa dữ liệu

◆ **Cú pháp**

DELETE FROM <tên bảng>

[**WHERE** <điều kiện>]

◆ **Ví dụ:**

- Xóa toàn bộ nhân viên

```
DELETE FROM NHANVIEN
```

- Xóa những sản phẩm do Trung Quốc sản xuất có giá thấp hơn 10000

```
DELETE FROM SANPHAM
WHERE (Gia<10000) and (Nuocsx='Trung Quoc')
```

3.3 Sửa dữ liệu

- **Cú pháp**

  UPDATE <tên bảng>

  SET <tên thuộc tính>=<giá trị mới>,

  <tên thuộc tính>=<giá trị mới>,

  …

  [**WHERE** <điều kiện>]

- **Ví dụ:** Tăng giá 10% đối với những sản phẩm do "Trung Quoc" sản xuất

```
UPDATE SANPHAM
SET Gia = Gia*1.1
WHERE Nuocsx='Trung Quoc'
```

Nội dung

1. Giới thiệu
2. Ngôn ngữ định nghĩa dữ liệu
3. Ngôn ngữ thao tác dữ liệu
4. Ngôn ngữ truy vấn dữ liệu

Truy vấn cơ bản

- **Dạng tổng quát của một câu truy vấn đơn giản gồm 3 mệnh đề**

  SELECT <danh sách các cột>

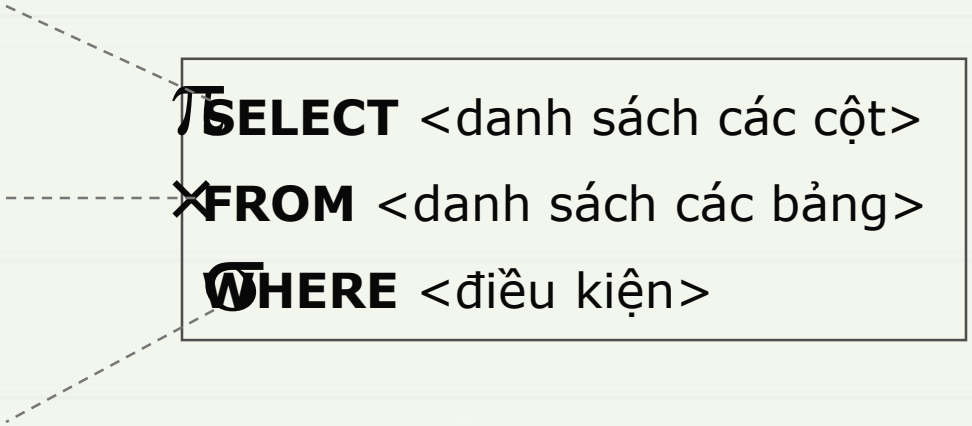
  FROM <danh sách các bảng>

  [**WHERE** <điều kiện>]

  - <danh sách các cột>
    - Tên các cột cần được hiển thị trong kết quả truy vấn
  - <danh sách các bảng>
    - Tên các bảng liên quan đến câu truy vấn
  - <điều kiện>
    - Biểu thức boolean xác định dòng nào sẽ được rút trích
    - Nối các biểu thức: AND, OR, và NOT
    - Phép toán: < , > , ≤ , ≥ , ≠ , =, LIKE và BETWEEN

Truy vấn cơ bản

- **SQL và ĐSQH**



SELECT L
FROM R
WHERE C

$\pi_L(\sigma_C(R))$

Ví dụ

Lấy tất cả các cột của quan hệ kết quả

```
SELECT *
FROM NHANVIEN
WHERE PHG=5
```

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	5
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	5

$$\sigma_{PHG=5}(NHANVIEN)$$

Mệnh đề SELECT

◆ Loại bỏ các dòng trùng nhau

- Sử dụng **DISTINCT** trong mệnh đề SELECT

```
SELECT DISTINCT LUONG
FROM NHANVIEN
WHERE PHG=5
```

LUONG
30000
25000
38000

Mệnh đề SELECT(tt)

◆ Tên bí danh

```
SELECT MANV, HONV AS HO, TENLOT AS 'TEN LOT', TENNV AS TEN
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	HO	TEN LOT	TEN
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

$$\rho_{MANV,HO,TEN\ LOT,TEN}(\pi_{MANV,HONV,TENLOT,TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)))$$

Mệnh đề SELECT (tt)

- Có thể chứa các biểu thức toán học liên quan tới +, -, *, / và thực hiện trên các hằng số hay thuộc tính của bộ

```
SELECT MANV, HONV + ' ' + TENLOT + ' ' + TENNV AS 'HO TEN'
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	HO TEN
333445555	Nguyen Thanh Tung
987987987	Nguyen Manh Hung

```
SELECT MANV, LUONG*1.1 AS 'LUONG10%'
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

Mệnh đề WHERE

- **Mô tả các điều kiện mà kết quả phải thỏa**
  - Tương ứng với điều kiện chọn của ĐSQH
- **Điều kiện kết hợp với nhau bởi AND, OR hay NOT**
- **Các so sánh có thể được áp dụng trên kết quả của các biểu thức toán học**

Ví dụ

- Tìm các nhân viên có chức vụ LT thuộc phòng P3

```
SELECT *
FROM NHANVIEN
WHERE CVU='LT' AND PHG='P3'
```

Mệnh đề WHERE (tt)

- **BETWEEN … AND…**
  - Ví dụ: Tìm tất cả nhân viên có lương tối thiểu là 20000 và lương tối đa là 45000

```
SELECT *
FROM NHANVIEN
WHERE LUONG>=20000 AND LUONG<=45000
```

```
SELECT *
FROM NHANVIEN
WHERE LUONG BETWEEN 20000 AND 45000
```

- **NOT BETWEEN… AND…**

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE LUONG NOT BETWEEN 20000 AND 45000
```

Mệnh đề WHERE(tt)

- Độ ưu tiên

```
SELECT MANV, TENNV
FROM NHANVIEN, PHONGBAN
WHERE (TENPHG='Nghien cuu' OR TENPHG='Quan ly')
      AND PHG=MAPHG
```

Mệnh đề WHERE (tt)

So sánh chuỗi

- Sử dụng bất kỳ phép so sánh nào chẳng hạn <, >, =, …
- Để so sánh gần đúng sử dụng **LIKE**
  - '%' thay thế chuỗi ký tự bất kỳ
  - '_' thay thế 1 ký tự bất kỳ

- Ví dụ:
  - Cho biết tên các nhân viên bắt đầu bằng chữ 'A'

```
SELECT TENNV
FROM NHANVIEN
WHERE TENNV LIKE 'A%'
```

  - Cho biết tên các nhân viên bắt đầu bằng chữ 'N' và có bốn ký tự

```
SELECT TENNV
FROM NHANVIEN
WHERE TENNV LIKE 'N_ _ _'
```

Mệnh đề WHERE (tt)

- **So sánh chuỗi**
  - Các ký tự đặc biệt sử dụng **ESCAPE** hoặc **[ ]**
  - Ví dụ:

```
SELECT MASP, TENSP
FROM SANPHAM
WHERE MASP LIKE 'B/_%'  ESCAPE '/'
```

'B_'

```
SELECT MASP, TENSP
FROM SANPHAM
WHERE MASP LIKE 'B[_]%'
```

Mệnh đề WHERE (tt)

Giá trị NULL

- Các bộ có thể có giá trị NULL ở một số thuộc tính
- **IS NULL** (**IS NOT NULL**) được dùng để kiểm tra giá trị null
- Ví dụ:

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NULL

SELECT MANV, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NOT NULL
```

- Kết quả của biểu thức toán học nào liên quan tới null là null
  - Ví dụ: 9 + null trả về null
- Các hàm tính toán bỏ qua các giá trị null

Mệnh đề WHERE (tt)

- **Sử dụng IN để kiểm tra một giá trị thuộc tính có thuộc một tập giá trị cho trước không**

- **Ví dụ:**
  - Cho biết tên các nhân viên thuộc một trong các phòng P1, P2, P3

```
SELECT TENNV
FROM NHANVIEN
WHERE PHG IN ('P1', 'P2', 'P3')
```

Mệnh đề FROM

- **Nhiều bảng có thể được truy vấn trong một lệnh SQL bằng cách liệt kê chúng trong mệnh đề FROM**
  - Chỉ ra điều kiện kết trên các thuộc tính tương ứng
  - Nếu không chỉ ra điều kiện kết liên quan giữa các bảng trong mệnh đề WHERE thì câu truy vấn tương đương phép tích trong ĐSQH

- **Ví dụ**

```
SELECT *
FROM PHONGBAN, DEAN
WHERE PHONGBAN.MAPHG = DEAN.PHONG
```

Mệnh đề FROM (tt)

- **Tên bí danh**

```
SELECT TENPHG, DIADIEM
FROM PHONGBAN, DDIEM_PHG
WHERE MAPHG=MAPHG
```

```
SELECT TENPHG, DIADIEM
FROM PHONGBAN AS PB, DDIEM_PHG AS DD
WHERE PB.MAPHG=DD.MAPHG
```

```
SELECT TENNV, NGSINH, TENTN, NGSINH
FROM NHANVIEN, THANNHAN
WHERE MANV=MA_NVIEN
```

```
SELECT TENNV, NV.NGSINH, TENTN, TN.NGSINH
FROM NHANVIEN NV, THANNHAN TN
WHERE MANV=MA_NVIEN
```

Ví dụ

1. Cho biết tên các dự án có ngân sách lớn hơn 250000.
2. Danh sách các chức vụ và lương, không liệt kê các dòng trùng
3. Tên các nhân viên sinh sau ngày 1/7/1970 có lương lớn hơn 35000 và có chức vụ là ‘LT’ hoặc ‘PT’.
4. Danh sách tên phòng ban, tên dự án do phòng ban đó chủ trì tương ứng và tên người quản lý phòng ban đó.
5. Cho biết tên nhân viên và tên dự án mà nhân viên tham gia. Chỉ liệt kê các dự án được chủ trì bởi phòng ban khác với phòng nhân viên trực thuộc.
6. Với những đề án ở ‘Ha Noi’, cho biết mã đề án, mã phòng ban chủ trì đề án, họ tên trưởng phòng cùng với ngày sinh và địa chỉ của người ấy
7. Tìm họ tên của nhân viên phòng số 5 có tham gia vào đề án “Sản phẩm X” với số giờ làm việc trên 10 giờ
8. Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó
9. Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp

Các phép toán tập hợp

- **Các phép toán tập hợp liên quan tới hai truy vấn riêng biệt được biểu diễn như sau**

  SQL1 **UNION** [ALL] SQL2

  SQL1 **INTERSECT** [ALL] SQL2

  SQL1 **EXCEPT** [ALL] SQL2

- **ALL: yêu cần giữ lại các dòng trùng.**
  - Mặc định là loại bỏ các dòng trùng

Ví dụ

- Cho biết mã nhân viên tham gia dự án với nhiệm vụ là Quan ly hay Tu van

```
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Quan ly'
   OR NVU='Tu van'
```

```
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Quan ly'
UNION
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Tu van'
```

Ví dụ (tt)

- Cho biết mã nhân viên tham gia dự án với nhiệm vụ là Quan ly và Tu van

```
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Quan ly'
INTERSECT
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Tu van'
```

```
SELECT DISTINCT PC1.MANV
FROM PHANCONG PC1,
PHANCONG PC2
WHERE PC1.MANV=PC2.MANV
  AND PC1.NVU='Quan ly'
  AND PC2.NVU='Tu van'
```

Ví dụ (tt)

- Cho biết mã nhân viên không tham gia dự án

```
SELECT DISTINCT MANV
FROM NHANVIEN
EXCEPT
SELECT DISTINCT MANV
FROM PHANCONG
```

Mệnh đề ORDER BY

- **Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó**
- **Cú pháp**

  SELECT <danh sách các cột>

  FROM <danh sách các bảng>

  WHERE <điều kiện>

  ORDER BY <danh sách các cột>

- **Nếu muốn thứ tự tăng dần, sử dụng ASC (mặc định)**
- **Nếu muốn thứ tự giảm dần, sử dụng DESC**

Mệnh đề ORDER BY (tt)

- Ví dụ: Tìm tất cả mã nhân viên và tên nhân viên chức vụ LT, sắp xếp theo phòng giảm dần và sau đó là ngày sinh tăng dần

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE NVU='LT'
ORDER BY PHG DESC, NGSINH ASC
```

Hàm kết hợp

- **Có 5 hàm kết hợp cơ bản:**
  - COUNT
    - COUNT(*) đếm số dòng
    - COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
  - SUM(A)
  - AVG(A)
  - MIN(A)
  - MAX(A)
- **COUNT, MAX, MIN được áp dụng cho một kiểu dữ liệu bất kỳ, SUM và AVG chỉ áp dụng cho kiểu số**
- **Sử dụng DISTINCT để loại bỏ các dòng trùng**
- **Ngoại trừ COUNT(*), giá trị NULL không được tính đến trong các hàm khác**

Ví dụ

- Cho biết số lượng nhân viên trong công ty

```
SELECT COUNT(MANV)
FROM NHANVIEN
```

- Cho biết tổng lương, lương cao nhất, lương thấp nhất, lương trung bình của các nhân viên thuộc P4

```
SELECT SUM(LUONG),MAX(LUONG),MIN(LUONG),AVG(LUONG)
FROM NHANVIEN
WHERE PHG='P4'
```

Ví dụ

- Có bao nhiêu dự án có nhân viên tham gia?
- Cho biết số lượng nhân viên của phòng nghiên cứu
- Cho biết lương trung bình của nhân viên tham gia dự án hơn 10 giờ

Mệnh đề Group By (Gom nhóm)

- **Muốn áp dụng hàm kết hợp cho từng nhóm**
- **Mệnh đề GROUP BY được dùng để gom nhóm các tập hợp bộ theo (các) thuộc tính cần gom nhóm**
  - Các hàm kết hợp tính toán riêng cho từng nhóm
  - Có một dòng output ứng với mỗi nhóm
  - Sau khi gom nhóm, mỗi nhóm bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm
- **Cú pháp**

  SELECT <danh sách các cột>

  FROM <danh sách các bảng>

  [WHERE <điều kiện>]

  GROUP BY <danh sách các cột gom nhóm>

  - <danh sách các cột> chỉ có thể là các cột trong <danh sách các cột gom nhóm>

Ví dụ

- Cho biết số lượng nhân viên của từng phòng

```
SELECT PHG, COUNT(*) AS SL_NV
FROM NHANVIEN
GROUP BY PHG
```

- Với mỗi phòng ban, cho biết tên phòng và lương cao nhất của phòng

```
SELECT TENPHG, MAX(LUONG) AS MAXLUONG
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG
GROUP BY MAPHG,TENPHG
```

- Với mỗi nhân viên cho biết mã số, họ tên, số lượng đề án và tổng thời gian mà họ tham gia

Mệnh đề HAVING

- **Mục đích: chọn ra các nhóm thỏa điều kiện nào đó**
- **Mệnh đề HAVING: điều kiện được áp dụng trên mỗi nhóm**
- **Cú pháp**

SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <danh sách các cột gom nhóm>
HAVING <điều kiện trên nhóm>

Ví dụ

- Cho biết mã nhân viên tham gia nhiều hơn 3 dự án

```
SELECT MA_NVIEN
FROM PHANCONG
GROUP BY MA_NVIEN
HAVING COUNT(*) > 3
```

- Cho biết tên các phòng ban có lương trung bình của nhân viên lớn hơn 45000

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG
GROUP BY MAPHG,TENPHG
HAVING AVG(LUONG) > 45000
```

Truy vấn với GROUP BY và HAVING

- Bước 1: Chọn các dòng thỏa điều kiện trong mệnh đề WHERE
- Bước 2: Các dòng này được gom nhóm với mệnh đề GROUP BY
- Bước 3: Tính hàm kết hợp cho mỗi nhóm
- Bước 4: Lọc các nhóm theo mệnh đề HAVING
- Bước 5: Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

Ví dụ

- Cho biết tên các phòng ban có lương trung bình của nhân viên lớn hơn 45000, chỉ tính các nhân viên sinh sau ngày 1/1/1970.

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG AND NGSINH > '1/1/1970'
GROUP BY MAPHG, TENPHG
HAVING AVG(LUONG) > 45000
```

- Cho biết tên phòng có số lượng nhân viên nữ trên 5 người.
- Với mỗi dự án, cho biết tên phòng ban chủ trì, tên dự án, và số lượng nhân viên tham gia. Chỉ tính các dự án có tổng số giờ >=30.

Câu truy vấn con

- Để diễn đạt một câu truy vấn phức tạp, một truy vấn có thể có một/nhiều câu truy vấn con (subquery) lồng bên trong.
- Subquery có thể được đặt ở mệnh đề SELECT, FROM, WHERE hay HAVING.
- Ví dụ: Cho biết tên các nhân viên có lương trên mức lương trung bình.

```
SELECT TENNV
FROM NHANVIEN
WHERE LUONG > (SELECT AVG(LUONG)
               FROM NHANVIEN)
```

Truy vấn con trong mệnh đề WHERE

- **Câu truy vấn con thường trả về một tập các giá trị**

- **Mệnh đề WHERE trong truy vấn cha có dạng**

<Biểu thức> <so sánh điều kiện> <truy vấn con>

  - Các phép toán so sánh thông thường: =, <>, <, >, <=, >=
  - Các toán tử
    - IN, NOT IN
    - ALL
    - ANY hoặc SOME
  - Kiểm tra sự tồn tại: EXISTS, NOT EXISTS
    - Kiểm tra một subquery có tạo ra dòng nào trong kết quả không.

Ví dụ

- Cho biết nhân viên tham gia dự án với thời gian nhiều hơn thời gian trung bình của cac nhân viên tham gia dự án

```
SELECT MANV
FROM PHANCONG
WHERE TGIAN > (SELECT AVG(TGIAN)
               FROM PHANCONG)
```

Ví dụ (tt)

- Cho biết mã số và tên các nhân viên có lương lớn hơn lương của tất cả nhân viên chức vụ 'LT'

```
SELECT MANV,TENNV
FROM NHANVIEN
WHERE LUONG > ALL(SELECT LUONG
                  FROM NHANVIEN
                  WHERE CVU='LT')
```

- Cho biết mã số và tên các nhân viên có lương lớn hơn ít nhất lương của một nhân viên chức vụ 'LT'

```
SELECT MANV,TENNV
FROM NHANVIEN
WHERE LUONG > ANY(SELECT LUONG
                  FROM NHANVIEN
                  WHERE CVU='LT')
```

Ví dụ (tt)

- Cho biết mã số các trưởng phòng không có thân nhân

```
SELECT TRPHG
FROM PHONGBAN
WHERE TRPHG NOT IN(SELECT MANV
                   FROM THANNHAN)
```

```
SELECT TRPHG
FROM PHONGBAN
WHERE TRPHG <> ALL(SELECT MANV
                   FROM THANNHAN)
```

Ví dụ (tt)

- Cho biết tên các dự án có nhân viên tên ‘Minh’ tham gia, hoặc các dự án được chủ trì bởi phòng ban mà nhân viên tên ‘Minh’ trực thuộc

```
SELECT DISTINCT TENDA
FROM DEAN
WHERE MADA IN (SELECT SODA
               FROM NHANVIEN, PHANCONG
               WHERE MANV=MA_NVIEN AND TENNV='Minh')
   OR MADA IN (SELECT MADA
               FROM NHANVIEN, DEAN
               WHERE PHG=PHONG AND TENNV='Minh')
```

Truy vấn con (tt)

- Cho biết tên các nhân viên có cùng tên với nhân viên khác trong công ty

```
SELECT TENNV
FROM NHANVIEN AS N1
WHERE EXISTS (SELECT *
              FROM NHANVIEN AS N2
              WHERE N1.TENNV=N2.TENNV
                AND N1.MANV<>N2.MANV)
```

- Một truy vấn con được gọi là ***tương quan*** với truy vấn ngoài nếu nó tham chiếu tới thuộc tính trong truy vấn ngoài
- Ngược lại, truy vấn được gọi là ***không tương quan***

Truy vấn con (tt)

- Truy vấn lồng có thể được chuyển thành truy vấn không lồng bằng cách sử dụng phép kết
- Ví dụ: Cho biết thông tin các dự án có nhân viên phòng ‘nghien cuu’ tham gia

```
SELECT *
FROM PHANCONG
WHERE MA_NVIEN IN (SELECT MANV
                   FROM NHANVIEN NV, PHONGBAN PB
                   WHERE NV.PHG=PB.MAPHG
                     AND PB.TENPHG='nghien cuu')

SELECT *
FROM PHANCONG PC, NHANVIEN NV, PHONGBAN PB
WHERE NV.PHG=PB.MAPHG AND PC.MA_NVIEN=NV.MANV
  AND PB.TENPHG='nghien cuu')
```

Ví dụ (tt)

- Với mỗi nhân viên, liệt kê thông tin tham gia dự án mà nhân viên đó tham gia nhiều thời gian nhất.

```
SELECT *
FROM PHANCONG
WHERE TGIAN = (SELECT MAX(TGIAN)
               FROM PHANCONG PC
               WHERE MANV=PC.MANV)
```

- Cho biết tên các nhân viên không tham gia dự án

```
SELECT TENNV
FROM NHANVIEN NV
WHERE NOT EXISTS (SELECT *
                  FROM PHANCONG PC
                  WHERE NV.MANV=PC.MANV)
```

Ví dụ (tt)

- Cho biết các nhân viên có người thân có cùng tên và cùng giới tính

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

Các quy tắc cú pháp

- **Số lượng thuộc tính trong mệnh đề select của subquery phải bằng với số lượng thuộc tính được so sánh**
- **Phải sử dụng alias nếu muốn truy vấn đến một bảng xuất hiện ở cả truy vấn trong và ngoài**
- **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

Truy vấn tất cả

- Thực hiện bằng một phép chia
- Cách 1: sử dụng NOT EXISTS, NOT EXISTS

R

A	B
a	1
a	2
b	1
c	1
c	2

S

B
1
2

R÷S

A
a
c

```
SELECT x.A
FROM R x
WHERE NOT EXISTS (
            SELECT *
            FROM S y
            WHERE NOT EXISTS (
                        SELECT *
                        FROM R z
                        WHERE z.A=x.A AND z.B=y.B))
```

- Cách 2: đếm

Ví dụ

◆ Cho biết tên các nhân viên tham gia tất cả các dự án

```
SELECT TENNV
FROM NHANVIEN NV                    1
WHERE NOT EXISTS (
                SELECT *
                FROM DEAN D                 2
                WHERE NOT EXISTS (                  3
                                SELECT *
                                FROM PHANCONG PC
                                WHERE PC.MA_NVIEN = NV.MANV
                                    AND PC.SODA = D.MADA ))
```

1: Nhân viên NV sao cho

2: không có đề án D nào mà không có

3: một bộ phân công PC chứng tỏ NV có tham gia D

Ví dụ (tt)

- **Cho biết tên các nhân viên tham gia tất cả các dự án**
  - Kiểm tra số lượng các dự án mà nhân viên tham gia có bằng với số lượng dự án không

```
SELECT TENNV
FROM NHANVIEN, PHANCONG
WHERE MA_NVIEN=MANV
GROUP BY TENNV
HAVING COUNT(SODA) = (SELECT COUNT(*)
                      FROM DEAN)
```

Các loại phép kết

- **Kết bằng**

```
SELECT <danh sách các cột>
FROM R1 [INNER] JOIN R2 ON  <biểu thức>
WHERE <điều kiện>
```

- **Kết ngoài**

```
SELECT <danh sách các cột>
FROM R1 LEFT|RIGHT [OUTER] JOIN R2 ON  <biểu thức>
WHERE <điều kiện>
```

Ví dụ

- Cho biết mã và tên nhân viên làm việc ở phòng 'Thiet ke'

```
SELECT MANV, TENNV
FROM NHANVIEN INNER JOIN PHONGBAN ON PHG=MAPHG
WHERE TENPHG='Thiet ke'

SELECT MANV, TENNV
FROM NHANVIEN, PHONGBAN
WHERE TENPHG='Thiet ke' AND PHG=MAPHG
```

- Có thể mô tả trực tiếp điều kiện kết trong mệnh đề FROM thay vì WHERE
- Chữ INNER là tùy chọn có thể bỏ

Ví dụ (tt)

- Cho biết họ tên nhân viên và tên phòng ban mà họ là trưởng phòng nếu có

```
SELECT TENNV, HONV, TENPHG
FROM NHANVIEN LEFT JOIN PHONGBAN ON MANV=TRPHG

SELECT TENNV, HONV, TENPHG
FROM PHONGBAN RIGHT JOIN NHANVIEN ON MANV=TRPHG
```

- Tìm tên các nhân viên và tên các đề án nhân viên tham gia nếu có

```
SELECT NV.TENNV,  DA.TENDA
FROM (PHANCONG PC JOIN DEAN DA ON SODA=MADA)
     RIGHT JOIN NHANVIEN NV ON PC.MA_NVIEN=NV.MANV
```

Tóm tắt truy vấn SQL

- **Cú pháp tổng quát của một câu truy vấn**

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
[WHERE <điều kiện>]
[GROUP BY <các thuộc tính gom nhóm>]
[HAVING <điều kiện trên nhóm>]
[ORDER BY <các thuộc tính sắp thứ tự>]
```

Bài tập

1. Cho biết TenNV có lương thấp hơn lương của quản lý trực tiếp và cho biết thấp hơn bao nhiêu.
2. Cho biết tên các nhân viên trong đó có chữ ‘A’. Sắp xếp kết quả tăng dần theo tên.
3. Với mỗi phòng ban, liệt kê tên phòng ban và lương trung bình của những nhân viên làm việc cho phòng ban đó.
4. Liệt kê chức vụ và số lượng nhân viên ứng với mỗi chức vụ có ít nhất 2 nhân viên.
5. Cho biết tên phòng ban chủ trì dự án có ngân sách lớn nhất.
6. Cho biết tên các phòng ban chủ trì dự án có ngân sách không phải lớn nhất.

Bài tập (tt)

7. Cho biết tên phòng ban chủ trì nhiều dự án nhất.
8. Danh sách những nhân viên có trên 2 thân nhân.
9. Danh sách những nhân viên (HONV, TENLOT, TENNV) được phân công *tất cả* đề án do phòng số 4 chủ trì.
10. Tìm những nhân viên (HONV, TENLOT, TENNV) được phân công *tất cả* đề án mà nhân viên Nguyen Van A làm việc.