

---

# NGÔN NGỮ TRUY VẤN SQL

---

---

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu
- Một số hàm thông dụng

# Giới thiệu

- SQL (Structured Query Language)
  - Ngôn ngữ cấp cao
  - Người sử dụng chỉ cần đưa ra nội dung cần truy vấn
  - Được phát triển bởi IBM (1970s), được gọi là SEQUEL (Structured English Query Language)
  - Được ANSI công nhận và phát triển thành chuẩn
    - SQL-86
    - SQL-92
    - SQL-99

# Giới thiệu (tt)

- SQL gồm các câu lệnh cho phép
  - Định nghĩa dữ liệu DDL (Data Definition Language)
  - Thao tác dữ liệu DML (Data Manipulation Language)
  - Ràng buộc toàn vẹn
  - Định nghĩa khung nhìn
  - Phân quyền và bảo mật
  - ...
- SQL sử dụng thuật ngữ
  - Bảng ~ quan hệ
  - Cột ~ thuộc tính
  - Dòng ~ bộ

---

# Nội dung chi tiết

- Giới thiệu
- **Định nghĩa dữ liệu**
  - Kiểu dữ liệu
  - Các lệnh định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Một số hàm thông dụng

# 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 DATABASE
  - ...

---

# Kiểu dữ liệu

- Số
  - SMALLINT
  - INT
  - NUMERIC
  - DECIMAL
  - REAL
  - FLOAT
  - ....

# Kiểu dữ liệu (tt)

- Chuỗi ký tự
  - CHAR, VARCHAR
  - NCHAR, NVARCHAR (gõ dấu tiếng Việt Unicode)
  - ...
- Chuỗi bit
  - BIT
  - BITINT
  - ...
- Ngày giờ
  - DATETIME
  - SMALLDATETIME



# Lệnh 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 (  
    HONV NVARCHAR (20),  
    TENLOT NVARCHAR (50),  
    TENNV NVARCHAR (20),  
    MANV NVARCHAR (20),  
    NGSINH          SMALLDATETIME,  
    DCHI NVARCHAR (50),  
    PHAI NVARCHAR (10),  
    LUONG INT,  
    MA_NQL NVARCHAR (20),  
    PHG INT  
)
```

# Lệnh tạo bảng (tt)

- <RBTV>
  - NOT NULL
  - UNIQUE
  - DEFAULT
  - PRIMARY KEY
  - FOREIGN KEY / REFERENCES
  - CHECK
  
- Đặt tên cho RBTV

**CONSTRAINT** <Ten\_RBTV> <RBTV>

# Ví dụ - Tạo bảng có RBTV

```
CREATE TABLE NHANVIEN (  
    HONV NVARCHAR (20) NOT NULL,  
    TENLOT NVARCHAR (50) NOT NULL,  
    TENNV NVARCHAR (20) NOT NULL,  
    MANV NVARCHAR (20) PRIMARY KEY,  
    NGSINH          SMALLDATETIME,  
    DCHI NVARCHAR(50),  
    PHAI NVARCHAR(10) CHECK (PHAI IN ('Nam',  
    'Nu')),  
    LUONG INT DEFAULT (1000000),  
    MA_NQL NVARCHAR(20),  
    PHG INT  
)
```

# Ví dụ - Tạo bảng có RBTV (tt)

```
CREATE TABLE PHONGBAN (  
    TENPHG NVARCHAR(40) UNIQUE,  
    MAPHG INT PRIMARY KEY,  
    TRPHG NVARCHAR(20),  
    NG_NHANCHUC SMALLDATETIME DEFAULT  
(GETDATE())  
)  
CREATE TABLE DIADIEM_PHG(  
    MAPHG INT NOT NULL,  
    DIADIEM NVARCHAR(50) NOT NULL,  
    CONSTRAINT PK_DIADIEM_PHG PRIMARY KEY (MAPHG ,  
DIADIEM)  
)
```

# Ví dụ - Tạo bảng có RBTV (tt)

```
CREATE TABLE DEAN (  
    TENDA NVARCHAR(40) UNIQUE,  
    MADA INT PRIMARY KEY,  
    DDIEM_DA NVARCHAR(50),  
    PHONG INT  
)
```

```
CREATE TABLE PHANCONG (  
    MA_NVIEN NVARCHAR(20) FOREIGN KEY (MA_NVIEN)  
        REFERENCES NHANVIEN(MANV),  
    SODA INT FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA),  
    THOIGIAN NUMERIC(3,1)  
)
```

# Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE NHANVIEN (  
    HONV NVARCHAR(20) CONSTRAINT NV_HONV_NN NOT NULL,  
    TENLOT NVARCHAR(50) NOT NULL,  
    TENNV NVARCHAR(20) NOT NULL,  
    MANV NVARCHAR(20) CONSTRAINT NV_MANV_PK PRIMARY  
KEY,  
    NGSINH          SMALLDATETIME,  
    DCHI NVARCHAR(50),  
    PHAI NVARCHAR(10) CONSTRAINT NV_PHAI_CHK  
        CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (10000),  
    MA_NQL NVARCHAR(20),  
    PHG INT
```

)

# Ví dụ - Đặt tên cho RBTV (tt)

```
CREATE TABLE PHANCONG (  
    MA_NVIENT NVARCHAR(20),  
    SODA INT,  
    THOIGIAN NUMERIC(3,1),  
    CONSTRAINT PC_MANVIEN_SODA_PK PRIMARY KEY (MA_NVIENT,  
SODA),  
    CONSTRAINT PC_MANVIEN_FK FOREIGN KEY (MA_NVIENT)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PC_SODA_FK FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA)  
)
```



# Lệnh sửa bảng

- Dùng để: thay đổi cấu trúc bảng, thay đổi RBTV
- Thêm cột

```
ALTER TABLE <Tên_bảng> ADD <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>]
```

- Ví dụ

```
ALTER TABLE NHANVIEN ADD NGHENGHIEP NVARCHAR(50)
```

- Xóa cột

```
ALTER TABLE <Tên_bảng> DROP COLUMN <Tên_cột>
```

- Ví dụ

```
ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP
```

---

# Lệnh sửa bảng (tt)

- Hiệu chỉnh cột

```
ALTER TABLE <Tên_bảng> ALTER COLUMN<Tên_cột> <Kiểu_dữ_liệu_mới>
```

- Ví dụ

```
ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP NVARCHAR(70)
```

# Lệnh sửa bảng (tt)

## ■ Thêm RBTV

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

- Ví dụ

```
ALTER TABLE PHONGBAN ADD  
    CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),  
    CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PB_NGNHANCHUC_DF DEFAULT (GETDATE())  
        FOR NG_NHANCHUC,  
    CONSTRAINT PB_TENPB_UNI UNIQUE (TENPB)
```

# Lệnh sửa bảng (tt)

- Xem các RBTV

```
SP_HELPCONSTRAINT <Tên_bảng>
```

- Ví dụ **SP\_HELPCONSTRAINT** PHONGBAN

- Xóa RBTV

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

- Ví dụ

```
ALTER TABLE PHONGBAN DROP PB_MAPHG_PK
```

```
ALTER TABLE PHONGBAN DROP PB_TRPHG
```

```
ALTER TABLE PHONGBAN DROP PB_NGNHANCHUC_DF
```

```
ALTER TABLE PHONGBAN DROP PB_TENPB_UNI
```

# Lệnh xóa bảng

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

- Cú pháp

```
DROP TABLE <Tên_bảng>
```

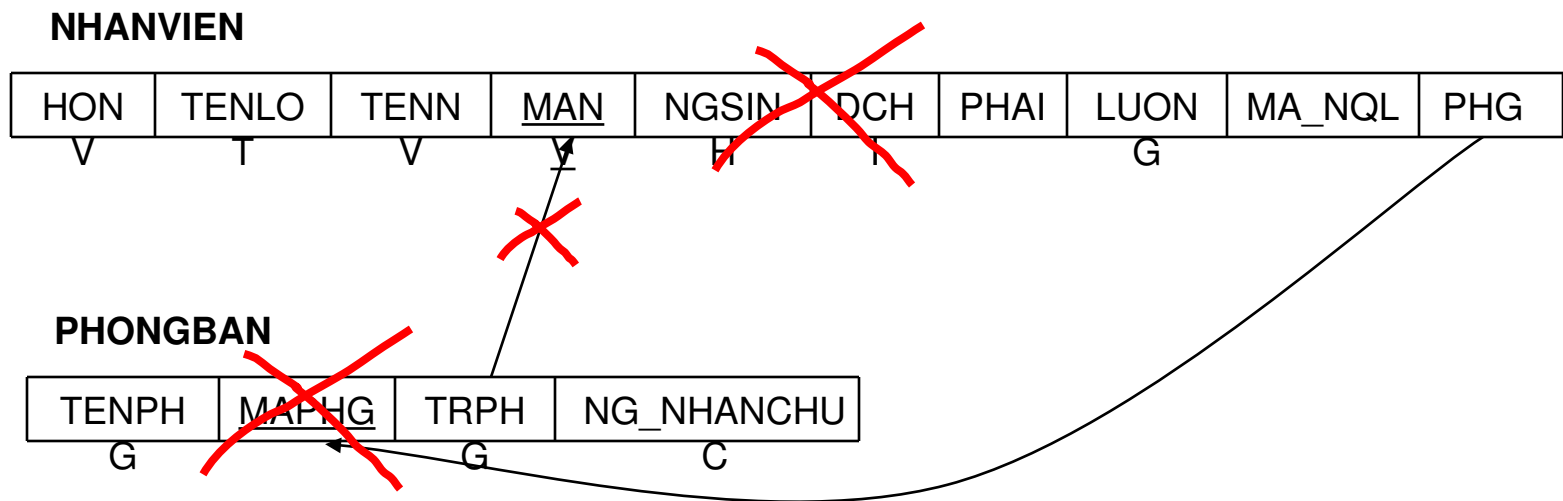
- Ví dụ

```
DROP TABLE NHANVIEN
```

```
DROP TABLE PHONGBAN
```

```
DROP TABLE PHANCONG
```

# Lệnh xóa bảng (tt)



# Lệnh tạo kiểu dữ liệu mới

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

```
CREATE TYPE <Tên_kdl_mới> FROM <KDL_Có_sẵn>
```

- Ví dụ

```
CREATE TYPE Kieu_Ten FROM NVARCHAR(30)
```

- Xóa kiểu dữ liệu tự định nghĩa

```
DROP TYPE <Tên_kdl_mới>
```

- Ví dụ

```
DROP TYPE Kieu_Ten
```

# Sử dụng DEFAULT và RULE

- Default và Rule là đối tượng có thể kết với một hoặc nhiều cột

- Tạo Default

```
CREATE DEFAULT <Tên_Default> AS <BT_giá_trị>
```

- Ràng buộc Default vào cột

```
sp_bindefault <Tên_Default>, '<Tên_bảng.Tên_cột>'
```

- Gỡ bỏ Default khỏi cột

```
sp_unbindefault '<Tên_bảng.Tên_cột>'
```

- Ví dụ

```
CREATE DEFAULT NV_LUONG_DF AS 10000
```

```
sp_bindefault NV_LUONG_DF, 'NHANVIEN.LUONG'
```

```
sp_unbindefault 'NHANVIEN.LUONG'
```



# Sử dụng DEFAULT và RULE (tt)

- Tạo Rule

```
CREATE RULE <Tên_rule> AS <BT_điều_kiện>
```

- Ràng buộc Rule vào cột

```
sp_bindrule < Tên_rule >, '<Tên_bảng.Tên_cột>'
```

- Gỡ bỏ Rule khỏi cột

```
sp_unbindrule '<Tên_bảng.Tên_cột>'
```

- Ví dụ

```
CREATE RULE R_LUONG AS @LUONG>=10000
```

```
sp_bindrule R_LUONG , 'NHANVIEN.LUONG'
```

```
sp_unbindrule 'NHANVIEN.LUONG'
```

- Xóa Default, Rule

```
DROP DEFAULT <Tên_Default>
```

```
DROP RULE <Tên_rule>
```

```
DROP DEFAULT NV_LUONG_DF
```

```
DROP RULE R_LUONG
```

---

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- **Cập nhật dữ liệu**
  - Thêm (insert)
  - Xóa (delete)
  - Sửa (update)
- Truy vấn dữ liệu
- Một số hàm thông dụng

# Lệnh INSERT

- Dùng để thêm 1 hay nhiều dòng vào bảng
- Để thêm dữ liệu
  - Tên bảng
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng
- Cú pháp (thêm 1 dòng)

```
INSERT INTO <tên bảng>[(<danh sách các thuộc tính>)]  
VALUES (<danh sách các giá trị>)
```

# Ví dụ

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV)  
VALUES (N'Nguyễn', N'Trọng', N'Hòa', '123')
```

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV, DCHI)  
VALUES ( N'Nguyễn', N'Thanh', N'Tùng', '333', NULL)
```

```
INSERT INTO NHANVIEN  
VALUES (N'Trần', N'Thanh', N'Tâm', '453', '7/31/1962', N'Mai Thị Lựu', 'Nam',  
25000,'333',5)
```

# Lệnh INSERT (tt)

## ■ Nhận xét

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và cho phép NULL
- Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
  - Khóa chính
  - Tham chiếu
  - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

---

# Lệnh INSERT (tt)

- Cú pháp (thêm nhiều dòng)

```
INSERT INTO <tên bảng>[(<danh sách các thuộc tính>)]  
    <câu truy vấn con>
```

# Ví dụ

```
CREATE TABLE THONGKE_PB (  
    TENPHG NVARCHAR(20),  
    SL_NV INT,  
    LUONG_TC INT  
)
```

```
INSERT INTO THONGKE_PB(TENPHG, SL_NV, LUONG_TC)  
SELECT TENPHG, COUNT(MANV), SUM(LUONG)  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG
```

---

# Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

```
DELETE FROM <tên bảng>  
[WHERE <điều kiện>]
```



# Ví dụ

```
DELETE FROM NHANVIEN  
WHERE HONV='Tran'
```

```
DELETE FROM NHANVIEN  
WHERE MANV='333'
```

```
DELETE FROM NHANVIEN
```

# Lệnh DELETE (tt)

## ■ Nhận xét

- Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
- Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
  - Không cho xóa
  - Xóa luôn những dòng có giá trị đang tham chiếu đến
    - \* CASCADE
  - Đặt NULL cho những giá trị tham chiếu

# Lệnh DELETE (tt)

MANV	HONV	TENLO	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
33344555	Nguyen	Thanh	Tung	12/08/1955	638 NVC	Nam	40000	88866555	5
98798798	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	33344555	5
45345345	Tran	Thanh	Tam	07/31/1972	543 MTL	Nu	25000	33344555	5
99988777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	98765432	4
98765432	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	88866555	4
98798798	Tran	Hong	Quang	04/08/1969	980 LHP	Nam	25000	98765432	4
88866555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

MA_NVIE	SODA	THOIGIAN
33344555	10	10.0
88866555	20	20.0
98798798	10	35.0
98798798	30	5.0
98765432	30	20.0
45345345	1	20.0

# Lệnh DELETE (tt)

TENPHG	MAPHG	MA_NVLEN	NG_NHANCHU
Nghien cuu	5	333445555	05/22/1988
Dieu hanh	4	987987987	01/01/1995
Quan ly	1	888665555	06/19/1981

MANV	HONV	TENLO	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
33344555	Nguyen	Thanh	Tung	12/08/1955	638 NVC	Nam	40000	88866555	NULL
98798798	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	33344555	NULL
45345345	Tran	Thanh	Tam	07/31/1972	543 MTL	Nu	25000	33344555	NULL
99988777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	98765432	4
98765432	Le	Quynh	Nhu	07/20/1951	219 TD Q3	Nu	43000	88866555	4
98798798	Tran	Hong	Quang	04/08/1969	980 LHP	Nam	25000	98765432	4
88866555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

# Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

```
UPDATE <tên bảng>  
SET <tên thuộc tính>=<giá trị mới>,  
      <tên thuộc tính>=<giá trị mới>,  
      ...  
[WHERE <điều kiện>]
```

# Ví dụ

```
UPDATE NHANVIEN  
SET NGSINH='08/12/1965'  
WHERE MANV='333445555'
```

```
UPDATE NHANVIEN  
SET LUONG=LUONG*1.1
```

# Ví dụ

- Với đề án có mã số 10, hãy thay đổi nơi thực hiện đề án thành 'Vung Tau' và phòng ban phụ trách là phòng 5

```
UPDATE DEAN
```

```
SET DIADIEM_DA='Vung Tau', PHONG=5
```

```
WHERE MADA=10
```

# Lệnh UPDATE

## ■ Nhận xét

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
- Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
  - Không cho sửa
  - Sửa luôn những dòng có giá trị đang tham chiếu đến
    - \* CASCADE



# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - Một số kiểu truy vấn khác
- Một số hàm thông dụng

# Truy vấn dữ liệu

- Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó
- Dựa trên

Phép toán ĐSQH

+

Một số bổ sung

- Cho phép 1 bảng có nhiều dòng trùng nhau
- Bảng là *bag*  $\neq$  quan hệ là *set*

# Truy vấn cơ bản

- 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>
  - Phép toán: +, -, \*, /, %
  - Phép so sánh: =, <, <=, >, >=, <>, !=
  - Phép toán logic: and, or, not
  - Phép toán tập hợp: all, any, in, like, between, exists

# Ví dụ

Lấy tất cả các cột  
của quan hệ kết quả

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

MANV	HONV	TENLO	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
33344555	Nguyen	Thanh	Tung	12/08/1955	638 NVC	Nam	40000	88866555	5
98798798	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	33344555	5

# Mệnh đề SELECT

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

MANV	HONV	TENLO	TENNV
33344555	Nguyen	Thanh	Tung
98798798	Nguyen	Manh	Hung

# 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
33344555	Nguyen	Thanh	Tung
9879 <sup>5</sup> 8798	Nguyen	Manh	Hung
7			

# Mệnh đề SELECT (tt)

Mở rộng

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

MANV	HO TEN
33344555	Nguyen Thanh Tung
9879 <sup>5</sup> 8798	Nguyen Manh Hung
7	

# Mệnh đề SELECT (tt)

Mở rộng

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

MANV	LUONG10%
33344555	33000
9879 <sup>5</sup> 8798	27500
7	



# Mệnh đề SELECT (tt)

Loại bỏ các dòng trùng nhau

```
SELECT DISTINCT LUONG
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

LUONG

---

30000

25000

28000

38000

# Mệnh đề WHERE

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

Biểu thức luận lý

↓ TRUE                      ↓ TRUE

# 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)

BETWEEN

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG >= 20000 AND LUONG <= 30000
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG BETWEEN 20000 AND 30000
```

# Mệnh đề WHERE (tt)

NOT BETWEEN

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG NOT BETWEEN 20000 AND  
30000
```

# Mệnh đề WHERE (tt)

LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen _ _ _ _ '
```

Ký tự bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen %'
```

Chuỗi bất kỳ

# Mệnh đề WHERE (tt)

NOT LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV LIKE 'Nguyen'
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV NOT LIKE 'Nguyen'
```

# Mệnh đề WHERE (tt)

Ngày giờ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGSINH BETWEEN '12/8/1955' AND '7/19/1966'
```



# Mệnh đề WHERE (tt)

NULL

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

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

# Mệnh đề FROM

Không sử dụng mệnh đề WHERE

```
SELECT MANV, MAPHG
FROM NHANVIEN, PHONGBAN
WHERE TRUE
```

MANV	MAPHG
33344555	1
<sup>5</sup> 33344555	4
<sup>5</sup> 33344555	5
<sup>5</sup> 98798798	1
<sup>7</sup> 98798798	4
<sup>7</sup> 98798798	5
<sup>7</sup> ...	...

# Mệnh đề FROM (tt)

Tên bí danh

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

```
SELECT TENNV, NSINH, TENN, TN.NGSINH
NGSINH
FROM NHANVIEN NV, THANNHAN TN
FROM NHANVIEN, THANNHAN
WHERE MANV=MA_NVIAN
WHERE MANV=MA_NVIAN
```

# 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>
```

- ASC: tăng (mặc định)
- DESC: giảm

# Mệnh đề ORDER BY (tt)

- Ví dụ

```
SELECT MA_NVIENT, SODA  
FROM PHANCONG  
ORDER BY MA_NVIENT DESC, SODA
```

MA_NVIENT	SODA
99988777	10
99988777	30
98798798	10
98798798	30
98765432	10
98765432	20
98765432	30

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - **Tập hợp, so sánh tập hợp và truy vấn lồng**
  - Hàm kết hợp và gom nhóm
  - Một số dạng truy vấn khác
- Một số hàm thông dụng

# Phép toán tập hợp trong SQL

- SQL có cài đặt các phép toán
  - Hội (UNION)
  - Giao (INTERSECT)
  - Trừ (EXCEPT)
- Kết quả trả về là tập hợp
  - Loại bỏ các bộ trùng nhau
  - Để giữ lại các bộ trùng nhau
    - UNION ALL
    - INTERSECT ALL
    - EXCEPT ALL

# Phép toán tập hợp trong SQL (tt)

## ■ Cú pháp

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

### **UNION [ALL]**

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

### **INTERSECT [ALL]**

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

### **EXCEPT [ALL]**

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>



# Phép toán tập hợp trong SQL (tt)

```
SELECT *  
FROM NHANVIEN  
WHERE PHG = 4 AND PHAI='Nu'  
  
UNION  
  
SELECT *  
FROM NHANVIEN  
WHERE PHG = 5 AND PHAI='Nam'
```

```
SELECT *  
FROM NHANVIEN  
WHERE PHG = 4  
  
INTERSECT  
  
SELECT *  
FROM NHANVIEN  
WHERE PHAI='Nam'
```

```
SELECT * FROM NHANVIEN  
  
EXCEPT  
  
SELECT * FROM NHANVIEN  
WHERE PHAI='Nam' AND PHG = 4
```

# 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)

# Truy vấn lồng (tt)

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Câu truy vấn con thường trả về một tập các giá trị
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- 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

# Truy vấn lồng (tt)

- Có 2 loại 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

# Ví dụ - 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'
```

)

# Ví dụ - 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 )
```

# 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

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - **Hàm kết hợp và gom nhóm**
  - Một số dạng truy vấn khác
- Một số hàm thông dụng



# Hàm kết hợp

- COUNT

- COUNT(\*) đếm số dòng
- COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
- COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính

- MIN

- MAX

- SUM

- AVG

- Các hàm kết hợp được đặt ở mệnh đề SELECT

# Ví dụ

- Cho biết số lượng nhân viên của từng phòng ban

PHG	SL_NV
5	3
4	3
1	1

MANV	HONV	TENLO	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
33344555	Nguyen	Thanh	Tung	12/08/1955	638 NVC	Nam	40000	88866555	5
98798798	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	33344555	5
45345345	Tran	Thanh	Tam	07/31/1972	543 MTL	Nu	25000	33344555	5
99988777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	98765432	4
98765432	Le	Quynh	Nhu	07/20/1951	219 TD Q3	Nu	43000	88866555	4
98798798	Tran	Hong	Quang	04/08/1969	980 LHP	Nam	25000	98765432	4
88866555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

# 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>
```

- Sau khi gom nhóm

- Mỗi nhóm các bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm

# Ví dụ

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

MA_NVIE	SODA	THOIGIAN
12345678	1	32.5
12345678	2	7.5
33344555	2	10.0
33344555	3	10.0
33344555	10	10.0
88866555	20	20.0
98798798	10	35.0
98798798	30	5.0
98765432	30	20.0
98765432	20	15.0
45345345	1	20.0
45345345	2	20.0

3

# Ví dụ

- Cho biết những nhân viên tham gia từ 2 đề án trở lên

MA_NVIE	SODA	THOIGIAN
12345678	1	32.5
12345678	2	7.5
33344555	2	10.0
33344555	3	10.0
33344555	10	10.0
88866555	20	20.0
98798798	10	35.0
98798798	30	5.0
98765432	30	20.0
98765432	20	15.0
45345345	1	20.0
45345345	2	20.0

3

bị loại ra

# Điều kiện trên 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>

# Nhận xét

## ■ Mệnh đề GROUP BY

- Các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm kết hợp) phải xuất hiện trong mệnh đề GROUP BY

## ■ Mệnh đề HAVING

- Sử dụng các hàm kết hợp trong mệnh đề SELECT để kiểm tra một số điều kiện nào đó
- Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện

# Nhận xét (tt)

- Thứ tự thực hiện câu truy vấn có mệnh đề GROUP BY và HAVING
  - (1) Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
  - (2) Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
  - (3) Áp dụng các hàm kết hợp cho mỗi nhóm
  - (4) Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
  - (5) Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT



---

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - **Một số dạng truy vấn khác**
- Một số hàm thông dụng

---

# Một số dạng truy vấn khác

- Truy vấn con ở mệnh đề FROM
- Điều kiện kết ở mệnh đề FROM
  - Phép kết tự nhiên
  - Phép kết ngoài
- Cấu trúc CASE
- SELECT ... INTO

# Truy vấn con ở mệnh đề FROM

- Kết quả trả về của một câu truy vấn phụ là một bảng

- Bảng trung gian trong quá trình truy vấn
- Không có lưu trữ thật sự

- Cú pháp

```
SELECT <đanh sách các cột>  
FROM R1, R2, (<truy vấn con>) AS tên_bảng  
WHERE <điều kiện>
```

- Ví dụ  
SELECT MANV, TENNV  
FROM NHANVIEN, (SELECT MAPHG  
FROM PHONGBAN  
WHERE TENPHG= 'Nghien cuu') AS B  
WHERE PHG = MAPHG

# Điều kiện kết ở mệnh đề FROM

- Kết bằng

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

- Ví dụ

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

# Điều kiện kết ở mệnh đề FROM

- Kết ngoài

```
SELECT <danh sách các cột>
```

```
FROM R1 LEFT|RIGHT|FULL JOIN R2 ON <biểu thức>
```

```
WHERE <điều kiện>
```

- Ví dụ

```
SELECT MANV, TENNV
```

```
FROM NHANVIEN LEFT JOIN PHONGBAN ON PHG = MAPHG
```

# Cấu trúc CASE

- Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- Cú pháp

```
CASE <tên cột>  
    WHEN <giá trị> THEN <biểu thức>  
    WHEN <giá trị> THEN <biểu thức>  
    ...  
    [ELSE <biểu thức>]  
END
```

# Cấu trúc CASE (tt)

```
SELECT MANV, TENNV , PHONG=  
    CASE PHG  
        WHEN 1 THEN N'MỘT'  
        WHEN 2 THEN N'HAI'  
        WHEN 3 THEN N'BA'  
        WHEN 4 THEN N'BỐN'  
        WHEN 5 THEN N'NHĂM'  
    END  
FROM NHANVIEN
```

# Select ... into...

- Dùng để tạo ra một bảng mới và đưa dữ liệu vào bảng mới

```
SELECT <danh sách các cột>  
[INTO <tên bảng>]  
FROM <danh sách các bảng>  
[WHERE <điều kiện>]
```

- Ví dụ: tạo bảng gồm những nhân viên nam

```
SELECT *  
INTO NHANVIEN_NAM  
FROM NHANVIEN  
WHERE PHAI='NAM'
```



# Kết luận

**SELECT** <danh sách các cột>

[**INTO** <tên bảng>]

**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>]

# Nội dung chi tiết

- Giới thiệu
- Định nghĩa dữ liệu
  - Kiểu dữ liệu
  - Các lệnh định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- **Một số hàm thông dụng**
  - Hàm toán học
  - Hàm chuỗi
  - Hàm ngày tháng
  - Hàm chuyển đổi kiểu

# Hàm toán học

Hàm	Diễn giải	Ví dụ	Kết quả
Power(X,Y)	Tính X lũy thừa Y	Select Power(2,5)	32
Round(X,n)	Làm tròn X còn n số lẻ	Round(123.4567,2)	123.46
Square(X)	Tính bình phương của X	Square(5)	25
SQRT(X)	Tính căn bậc 2 của X	SQRT(16)	4
Sum(cột)	Tính tổng cột	Sum(luong)	
Count(cột)	Đếm số phần tử khác Null	Count(MANV)	
Count(*)	Đếm số dòng	Count(*)	
Max(cột)	Cho giá trị lớn nhất	Max(luong)	
Min(cột)	Cho giá trị nhỏ nhất	Min(luong)	
Avg(cột)	Tính trung bình cột	Avg(luong)	

# Hàm chuỗi

Hàm	Diễn giải	Ví dụ	Kết quả
Ascii(C)	Trả về mã Ascii của ký tự C	select ASCII('A')	65
Char(N)	Trả về ký tự có mã Ascii là N	Char(66)	B
Len(S)	Trả về chiều dài S	Len('abc xyz')	7
Lower(S)	Chuyển S sang chữ thường	Lower('abcXYZ')	abcxyz
Upper(S)	Chuyển S sang chữ hoa	Upper('abcXYZ')	ABCXYZ
LTrim(S)	Cắt khoảng trắng bên trái	LTrim(' abc')	abc
Rtrim(S)	Cắt khoảng trắng bên phải	RTrim('abc ')	abc
Left(S,n)	Trích n ký tự bên trái S	Left('abcxyz',4)	abcx
Right(S,n)	Trích n ký tự bên phải S	Right('abcxyz',4)	cxyz

# Hàm chuỗi (tt)

Hàm	Diễn giải	Ví dụ	Kết quả
S1+S2	Nối S1 với S2	Select 'abc'+'XYZ'	abcXYZ
CharIndex (S1,S)	Trả về vị trí đầu tiên của S1 xuất hiện trong S	CharIndex ('bc','abcxyzabc')	2
SubString (S,p,n)	Trích n ký tự từ vị trí p của S	SubString ('abcxyz',2,4)	bcxy
Replace (S1,S2,S3)	Thay tất cả S2 trong S1 bằng S3	Replace ('ababc','ab','xy')	xyxyc
Reverse(S)	Trả về chuỗi đảo ngược S	Reverse('abcxyz')	zyxcba

# Hàm ngày tháng

Các hàm có sử dụng tham số DatePart, giá trị của DatePart được cho như bảng sau

<b>DatePart</b>	<b>Giá trị</b>	<b>Diễn giải</b>
DD	1-31	Ngày trong tháng
MM	1-12	Tháng trong năm
QQ	1-4	Quý trong năm
DW	1-7 (Sun-Sat)	Thứ trong tuần
YY	1753-9999	Năm

# Hàm ngày tháng (tt)

Hàm	Diễn giải	Ví dụ	Kết quả
GetDate()	Trả về ngày giờ hiện hành	Select GetDate()	2007-06-18 08:34:44.107
Day(date)	Trích phần ngày	Day(GetDate())	18
Month(date)	Trích phần tháng	Month(Getdate())	6
Year(date)	Trích phần năm	Year(GetDate())	2007
DatePart (DatePart,date)	Trích DatePart của date	DatePart(mm, GetDate())	6
DateAdd (DatePart, n,date)	Thêm n DatePart vào date	DateAdd (dd,-2, Getdate())	2007-06-16
DateDiff(DatePart, date1,date2)	Trả về số DatePart giữa 2 ngày	DateDiff(mm, '2007-02-16', Getdate())	4

# Hàm chuyển đổi kiểu

Hàm	Diễn giải	Ví dụ	Kết quả
Cast(exp AS data_type)	Chuyển giá trị exp sang kiểu data_type	Cast('123' AS Int) Cast(123 AS Varchar(10))	123 '123'
Convert(data_type, exp [,style])	Chuyển giá trị exp sang kiểu data_type theo style	Convert(Varchar(20), GetDate(),103) Convert(Varchar(20), GetDate(),101)	18/06/2007 06/18/2007