



CƠ SỞ DỮ LIỆU

Khoa Công nghệ thông tin – Đại học Sài Gòn

Chương 4

NGÔN NGỮ TRUY VẤN SQL



Nội dung chi tiết

1. Giới thiệu
2. Định nghĩa dữ liệu
3. Truy vấn dữ liệu
4. Cập nhật dữ liệu
5. Khung nhìn (view)
6. Chỉ mục (index)

1. Giới thiệu

Ngôn ngữ ĐSQH

- Cách thức truy vấn dữ liệu

→ Khó khăn cho người sử dụng

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

1. Giới thiệu

SQL gồm

- Định nghĩa dữ liệu (DDL)
- Thao tác dữ liệu (DML)
- Định nghĩa khung nhìn
- Ràng buộc toàn vẹn
- Phân quyền và bảo mật
- Điều khiển giao tác

SQL sử dụng thuật ngữ

- Bảng ~ quan hệ
- Cột ~ thuộc tính
- Dòng ~ bộ

Lý thuyết : Chuẩn SQL-92

Ví dụ : SQL Server

sinhvien(masv, tensv,.....)

1. Giới thiệu

SQL thao tác trên mô hình vật lý của Cơ sở dữ liệu.

Mô hình vật lý có cấu trúc tương tự với mô hình dữ liệu quan hệ, trong đó gồm các khái niệm:

- Bảng ~ quan hệ
- Cột ~ thuộc tính
- Dòng ~ bộ

Với mô hình vật lý, ta cần xác định cụ thể kiểu dữ liệu của từng thuộc tính.

Giới thiệu Microsoft SQL Server

Là chương trình giúp tạo, quản lý và truy xuất cơ sở dữ liệu do Microsoft phát hành.

Sử dụng mô hình CSDL quan hệ.

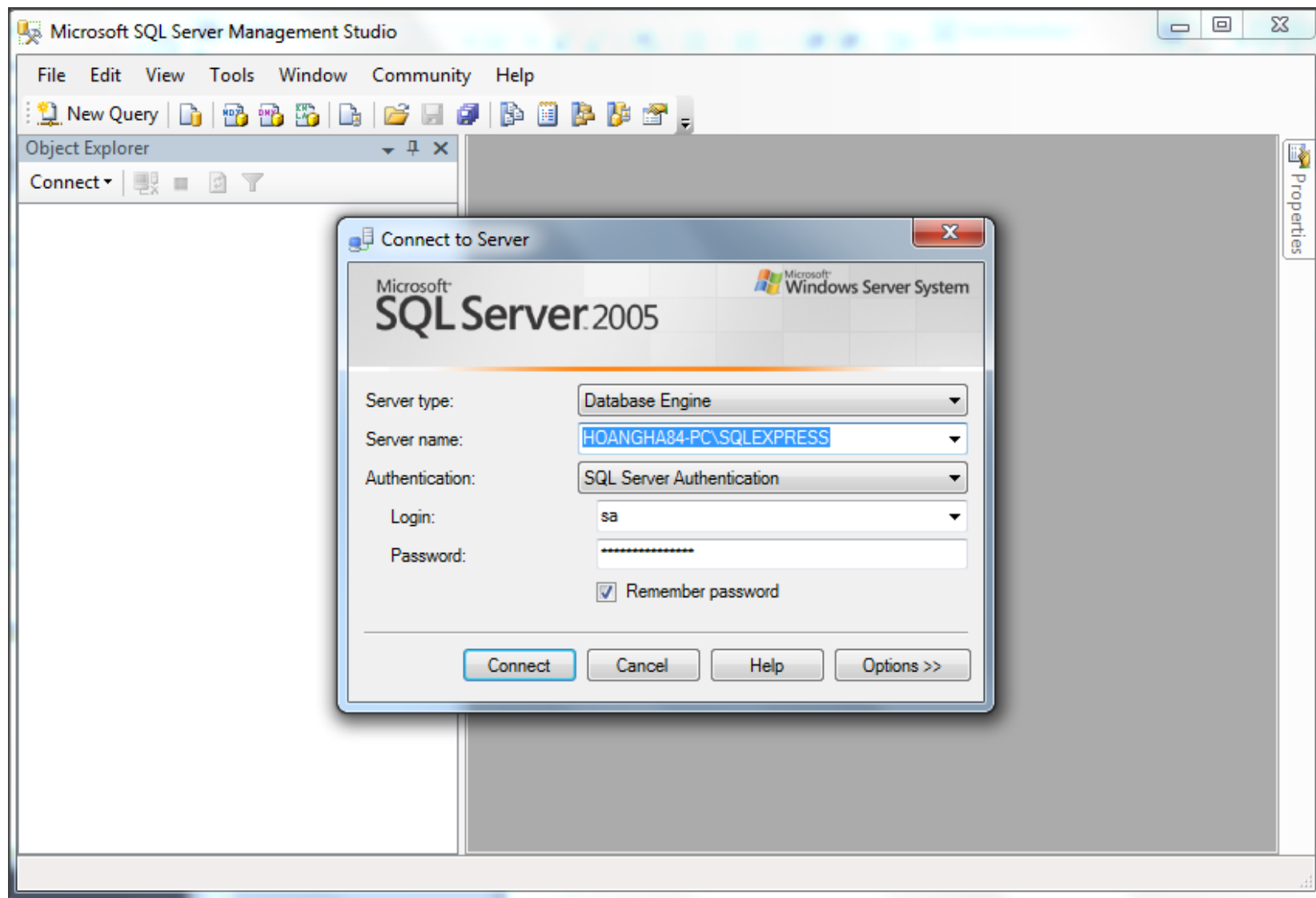
Ngôn ngữ truy xuất dữ liệu là ANSI-SQL.

Không phân biệt chữ hoa với chữ thường.

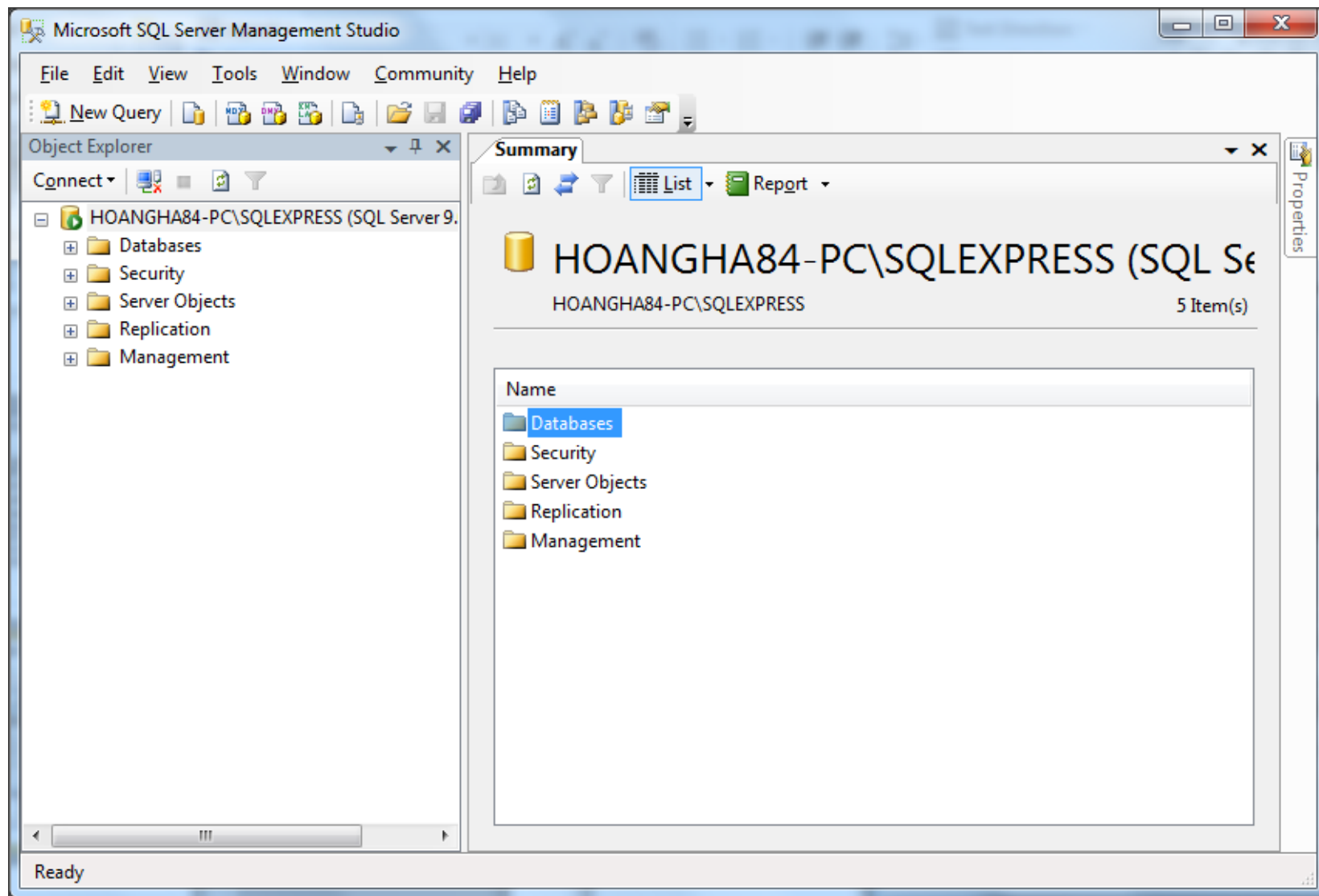
Có kèm tài liệu tham khảo với tên gọi Books Online:

- Hỗ trợ xem thông tin các lệnh.
- Tìm kiếm từ khóa.
- Đầy đủ cấu trúc các lệnh SQL.
- Có ví dụ tham khảo.

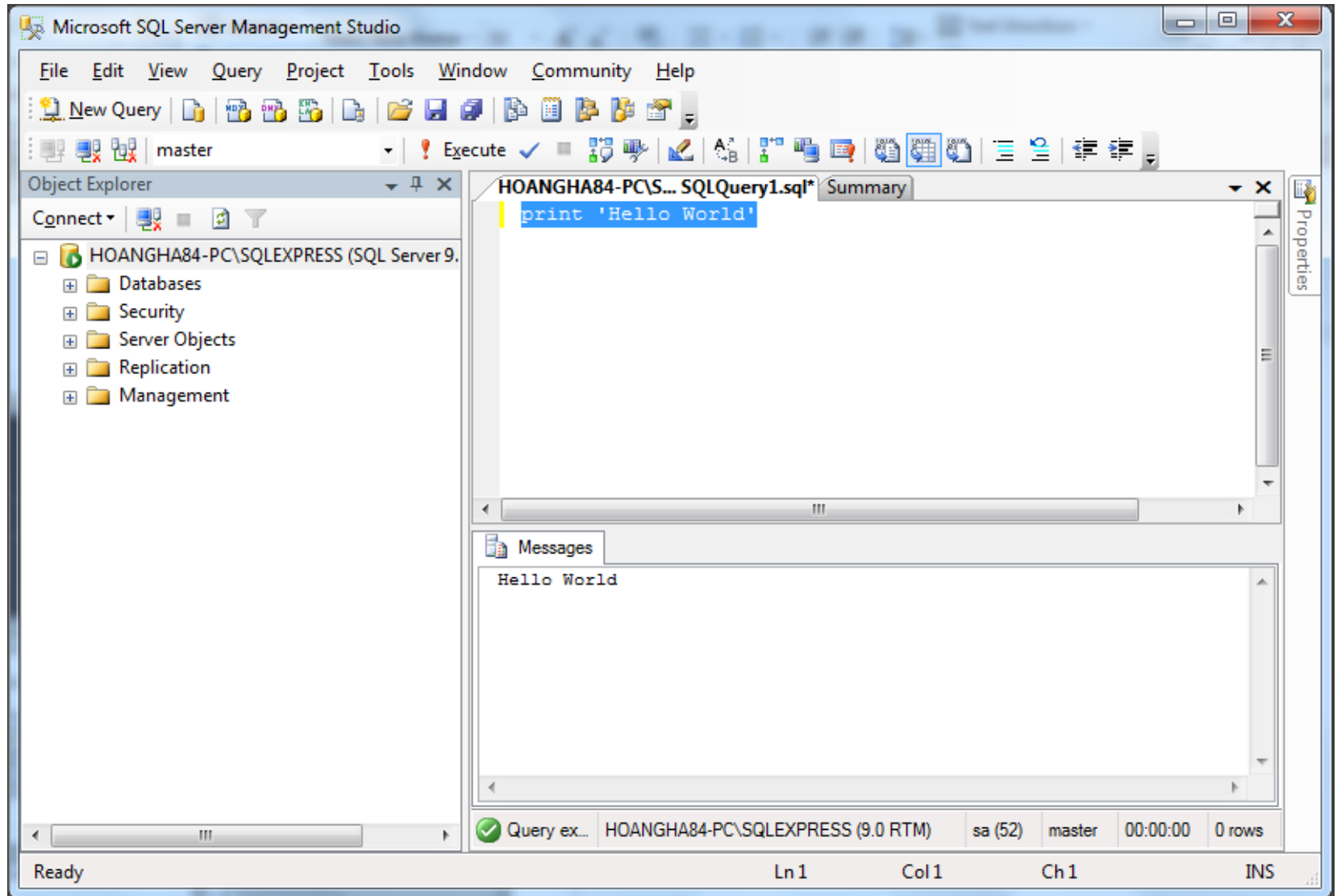
Giao diện đăng nhập



Giao diện làm việc



Giao diện truy vấn



Các thao tác cơ bản

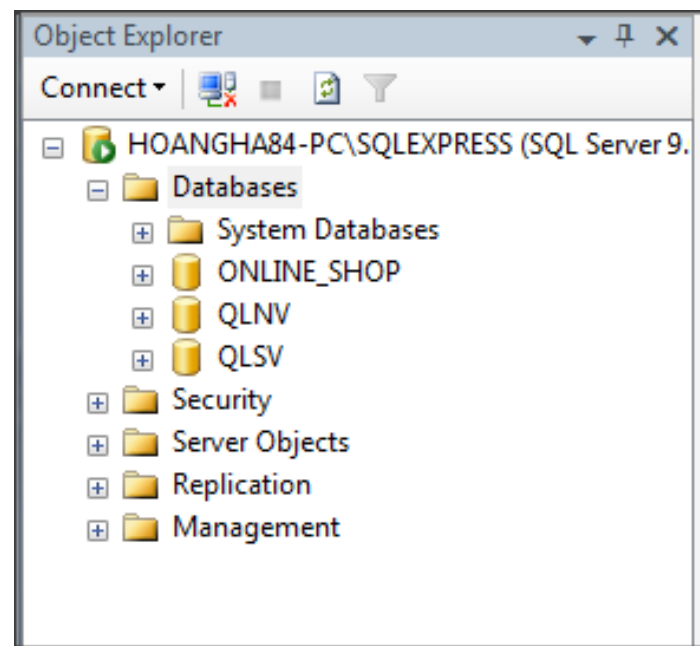
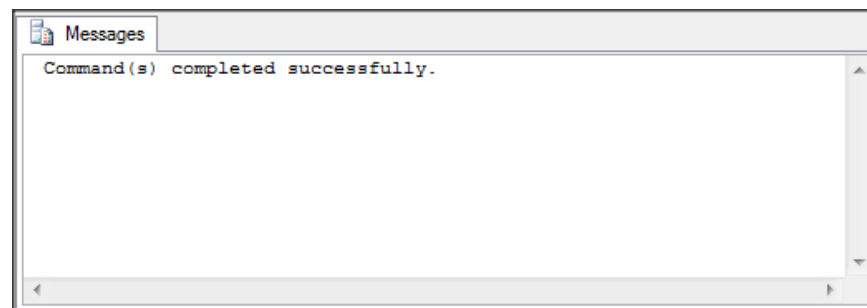
Muốn chạy một lệnh nào đó:

- Chọn lệnh cần chạy.
- Nhấn F5 hoặc nhấn nút Execute.

- Nếu chạy thành công:

Xem các đối tượng đang quản lý trong hệ thống:

- Mở rộng các đối tượng tương ứng bên cửa sổ Object Explorer
- Có thể click phải chuột để hiện menu ngữ cảnh.



Lệnh GO

Lệnh GO (GO command) không phải là một từ khóa của SQL (SQL statement).

GO là một lệnh được nhận bởi trình biên dịch MS SQL code editor.

GO đánh dấu kết thúc 1 tập lệnh SQL (batch)

Mỗi tập lệnh SQL sẽ được thực thi đồng thời, nếu xảy ra lỗi ở bất kì lệnh nào sẽ hủy bỏ toàn bộ lệnh của tập lệnh đó.

```
go
select
    jfhksdhfjkdj
    hdgshd
go
```

Lệnh GO

Phân biệt:

drop database test

create database test

use test

drop database test

GO

create database test

GO

use test

GO

2. Định nghĩa dữ liệu

Là ngôn ngữ mô tả (DDL)

- Tạo CSDL
- 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
- ...

a. Kiểu dữ liệu

Số (numeric)

- INTEGER
- SMALLINT
- NUMERIC, NUMERIC(p), NUMERIC(p,s)
- DECIMAL, DECIMAL(p), DECIMAL(p,s)
- REAL
- DOUBLE PRECISION
- FLOAT, FLOAT(p)

a. Kiểu dữ liệu

Chuỗi ký tự (character string)

- CHARACTER, CHARACTER(n)
- CHARACTER VARYING(x)

Chuỗi bit (bit string)

- BIT, BIT(x)
- BIT VARYING(x)

Ngày giờ (datetime)

- DATE gồm ngày, tháng và năm
- TIME gồm giờ, phút và giây
- TIMESTAMP gồm ngày và giờ

b. Tạo CSDL

Mỗi CSDL (trong MS SQL Server) có tên duy nhất.

Mỗi CSDL gồm 2 tập tin vật lý:

- Tập tin .MDF: lưu trữ nội dung CSDL
- Tập tin .LDF: lưu trữ log

qlsv
qlsv_log

Sau khi chạy, đọc thông báo thành công hay thất bại trong cửa sổ Message.

Dựa vào thông báo lỗi để biết được nguyên nhân lỗi.

Sau khi chạy lệnh tạo CSDL, nhớ refresh Database.

b. Tạo CSDL

```
CREATE DATABASE database_name  
[  
    ON  
        [ PRIMARY ] [ <filespec> [ ,...n ]  
        [ , <filegroup> [ ,...n ] ]  
        [ LOG ON { <filespec> [ ,...n ] } ]  
    ]  
    [ COLLATE collation_name ]  
    [ WITH <external_access_option> ]  
]  
[:]
```

b. Tạo CSDL

```
<filespec> ::=  
{  
(  
    NAME = logical_file_name ,  
    FILENAME = 'os_file_name'  
        [ , SIZE = size [ KB | MB | GB | TB ] ]  
        [ , MAXSIZE = { max_size [ KB | MB | GB | TB ] |  
UNLIMITED } ]  
        [ , FILEGROWTH = growth_increment [ KB | MB | GB | TB  
| % ] ]  
)[ ,...n ]  
}
```

b. Tạo CSDL

Tạo CSDL đơn giản không tham số:

```
CREATE DATABASE TEST
```

Tạo CSDL chỉ định đường dẫn nơi chứa CSDL:

```
CREATE DATABASE QLNV  
ON (  
    NAME = 'QLNV_Data',  
    FILENAME = 'D:\QLNV_Data.mdf',  
    SIZE = 10MB,  
    MAXSIZE = UNLIMITED,  
    FILEGROWTH = 5MB)  
LOG ON (  
    NAME = 'QLNV_Log',  
    FILENAME = 'D:\QLNV_Log.ldf',  
    SIZE = 5MB,  
    MAXSIZE = UNLIMITED,  
    FILEGROWTH = 2MB)
```

c. Chỉnh sửa CSDL

Lệnh chỉnh sửa chung thường là ALTER + Loại + Tên đối tượng:

```
ALTER DATABASE database_name
{
    <add_or_modify_files>
  | <add_or_modify_filegroups>
  | <set_database_options>
  | MODIFY NAME = new_database_name
  | COLLATE collation_name
}
[;]
```

Ví dụ:

```
ALTER DATABASE AdventureWorks
MODIFY FILE
( NAME = Test1dat2,
  FILENAME = N'c:\t1dat2.ndf'
);
```

d. Xóa CSDL

Lệnh xóa chung thường là DROP + Loại + Tên đối tượng

DROP DATABASE <Database_Name>

Ví dụ:

DROP DATABASE test

e. 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  
(  
    MANV CHAR(9) not null,  
    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  
)
```


e. Lệnh tạo bảng

<RBTV>

- NOT NULL
- NULL
- UNIQUE (Khóa chỉ định)
- DEFAULT
- PRIMARY KEY
- FOREIGN KEY / REFERENCES
- CHECK

Đặt tên cho RBTV

CONSTRAINT <Ten_RBTV> <RBTV>

Ví dụ - Tạo bảng có kèm RBTV

```
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 - Tạo bảng có kèm RBTV

```
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 - RBTB có đặt tên

CREATE TABLE NHANVIEN

(

HONV VARCHAR(10) CONSTRAINT NV_HONV_NN NOT NULL,

TENLOT VARCHAR(20) NOT NULL,

TENNV VARCHAR(10) NOT NULL,

MANV CHAR(9) CONSTRAINT NV_MANV_PK PRIMARY KEY,

NGSINH DATETIME,

DCHI VARCHAR(50),

PHAI CHAR(3) CONSTRAINT NV_PHAI_CHK

CHECK (PHAI IN ('Nam', 'Nu')),

LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (10000),

MA_NQL CHAR(9),

PHG INT

)

Ví dụ - RBTB có đặt tên

CREATE TABLE PHANCONG

(

MA_NVIEN CHAR(9),

SODA INT,

THOIGIAN DECIMAL(3,1),

CONSTRAINT PC_MANVIEN_SODA_PK PRIMARY KEY

(MA_NVIEN, SODA),

CONSTRAINT PC_MANVIEN_FK FOREIGN KEY (MA_NVIEN)

REFERENCES NHANVIEN(MANV),

CONSTRAINT PC_SODA_FK FOREIGN KEY (SODA)

REFERENCES DEAN(MADA)

)

f. Lệnh 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>
```

Mở rộng cột

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

f. Lệnh sửa bảng

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 DROP COLUMN NGHENGHIEP

ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP CHAR(50)

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 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)
```

Ví dụ - Thay đổi RBTV khóa ngoại

```
CREATE TABLE NHANVIEN
```

```
(
```

```
    MANV VARCHAR(20) PRIMARY KEY,
```

```
    HOTEN NVARCHAR(50)
```

```
)
```

```
ALTER TABLE PHONGBAN
```

```
    ADD TRGPHG VARCHAR(20)
```

```
ALTER TABLE PHONGBAN
```

```
    ADD CONSTRAINT PB_TRGPHG_FK FOREIGN KEY  
(TRGPHG) REFERENCES NHANVIEN(MANV) ON DELETE CASCADE
```

g. 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

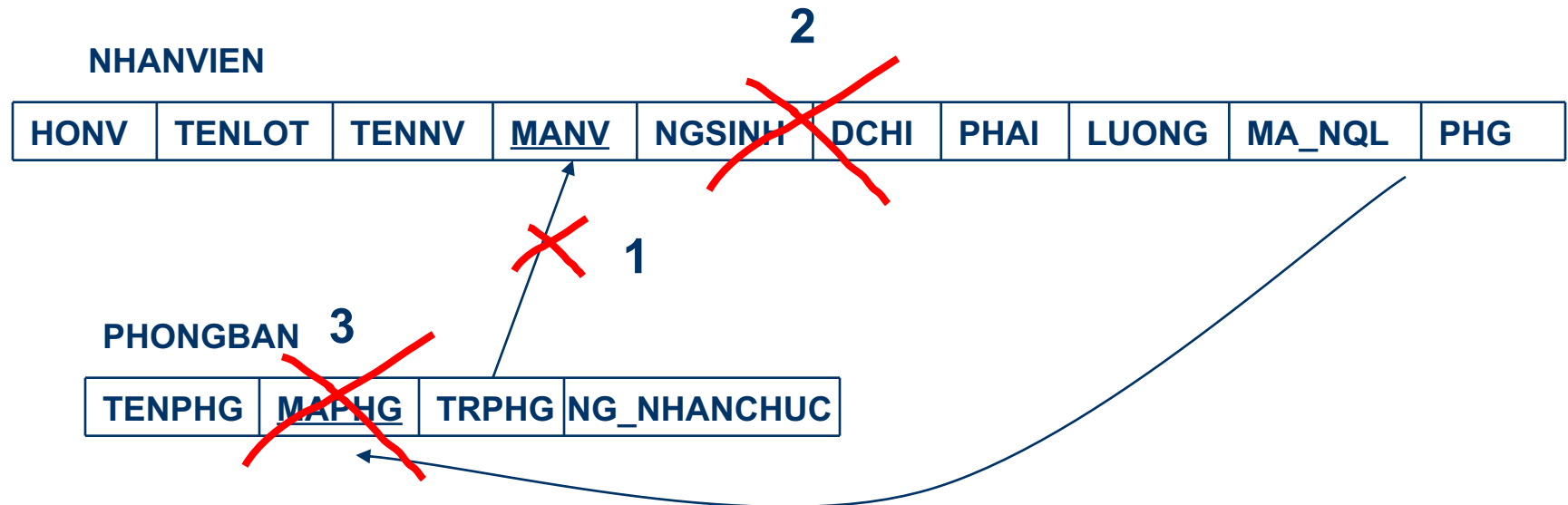
```
DROP TABLE NHANVIEN
```

```
DROP TABLE PHONGBAN
```

```
DROP TABLE PHANCONG
```

g. Lệnh xóa bảng

- Trình tự xóa bảng như sau:



1. ALTER TABLE PHONGBAN DROP COLUMN TRPHG

Hay ALTER TABLE PHONGBAN DROP fk_PHONGBAN

2. DROP TABLE NHANVIEN

3. DROP TABLE PHONGBAN

k. Lệnh 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>
```

Ví dụ

```
CREATE DOMAIN Kieu_Ten AS VARCHAR(30)
```

3. 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

a. 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>
 - Biểu thức boolean xác định dòng nào sẽ được rút trích
 - Nối các biểu thức: AND, OR, và NOT
 - Phép toán: < , > , ≤ , ≥ , ≠ , =, LIKE và BETWEEN

a. Truy vấn cơ bản

SQL và ĐSQH

π

\times

σ

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

**SELECT L
FROM R
WHERE C** $\longrightarrow \pi_L(\sigma_C(R))$

Ví dụ

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

Lấy tất cả các cột
của quan hệ kết quả

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

$\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)$

Mệnh đề SELECT

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

MANV	HONV	TENLOT	TENNV
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

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

Mệnh đề SELECT

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

Mở rộng

```
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

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

Mệnh đề SELECT

Mở rộng

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

MANV	LUONG10%
333445555	33000
987987987	27500

$$\rho_{\text{MANV,LUONG10\%}}(\pi_{\text{MANV,LUONG*1.1}}(\sigma_{\text{PHG=5} \wedge \text{PHAI='Nam'}}(\text{NHANVIEN})))$$

Mệnh đề SELECT

Loại bỏ các dòng trùng nhau

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

LUONG

30000

25000

28000

38000

- Tồn chi phí
- Người dùng muốn thấy

Ví dụ

Cho biết MANV và TENNV làm việc ở phòng 'Nghien cuu'

$R1 \leftarrow \text{NHANVIEN} \bowtie_{\text{PHG}=\text{MAPHG}} \text{PHONGBAN}$

$KQ \leftarrow \pi_{\text{MANV}, \text{TENNV}} (\sigma_{\text{TENPHG}=\text{'Nghien cuu'}}(R1))$

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

Mệnh đề WHERE

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

↓ ↓

TRUE TRUE

Biểu thức luận lý

Mệnh đề WHERE

Độ ưu tiên

SELECT MANV, TENNV

FROM NHANVIEN, PHONGBAN

WHERE (TENPHG='Nghien cuu' OR TENPHG='Quan ly') AND

PHG=MAPHG

Mệnh đề WHERE

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

NOT BETWEEN

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

Mệnh đề WHERE

LIKE

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

Ký tự bất kỳ

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

Chuỗi bất kỳ

Mệnh đề WHERE

NOT LIKE

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

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

Mệnh đề WHERE

Escape

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

↓
'Nguyễn%'

```
select *  
from nhanvien  
where tennv like 'H!_' escape '!'
```

Hi

Ha

He

H_

Mệnh đề WHERE

Ngày giờ

SELECT MANV, TENNV

FROM NHANVIEN

WHERE NGSINH BETWEEN '1955-12-08' AND '1966-07-19'

'1955-12-08' YYYY-MM-DD

'17:30:00' HH:MI:SS

'12/08/1955' MM/DD/YYYY

'05:30 PM'

'December 8, 1955'

'1955-12-08 17:30:00'

Mệnh đề WHERE

NULL

- Sử dụng trong trường hợp
 - Không biết (value unknown)
 - Không thể áp dụng (value inapplicable)
 - Không tồn tại (value withheld)
- Những biểu thức tính toán có liên quan đến giá trị NULL sẽ cho ra kết quả là NULL
 - x có giá trị là NULL
 - $x + 3$ cho ra kết quả là NULL
 - $x + 3$ là một biểu thức không hợp lệ trong SQL
- Những biểu thức so sánh có liên quan đến giá trị NULL sẽ cho ra kết quả là UNKNOWN
 - $x = 3$ cho ra kết quả là UNKNOWN
 - $x = 3$ là một so sánh không hợp lệ trong SQL

Mệnh đề WHERE

NULL

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

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

Mệnh đề WHERE

Khi không có mệnh đề WHERE → Tương đương với WHERE
(mệnh đề luôn đúng)

SELECT *

FROM PHONGBAN

WHERE 1=1

Mệnh đề WHERE

SELECT MANV, MAPHG

FROM NHANVIEN, PHONGBAN

WHERE TRUE

MANV	MAPHG
333445555	1
333445555	4
333445555	5
987987987	1
987987987	4
987987987	5
...	...

Mệnh đề FROM

Điều kiện kết của phép kết cũng được thể hiện thông qua mệnh đề WHERE còn các quan hệ trong phép kết được thể hiện trong mệnh đề FROM.

Ví dụ: Với mỗi phòng ban hãy cho biết các địa điểm của phòng ban đó

PHONGBAN(TENPHG, MAPHG, TRPHG, NG_NHANCHUC)

DIADIEMPHG(MAPHG, DIADIEM)

KQ ← PHONGBAN ⋈_{MAPHG=MAPHG} DIADIEMPHG

SELECT *

FROM PHONGBAN, DIADIEMPHG

WHERE PHONGBAN.MAPHG=DIADIEMPHG.MAPHG

Mệnh đề FROM

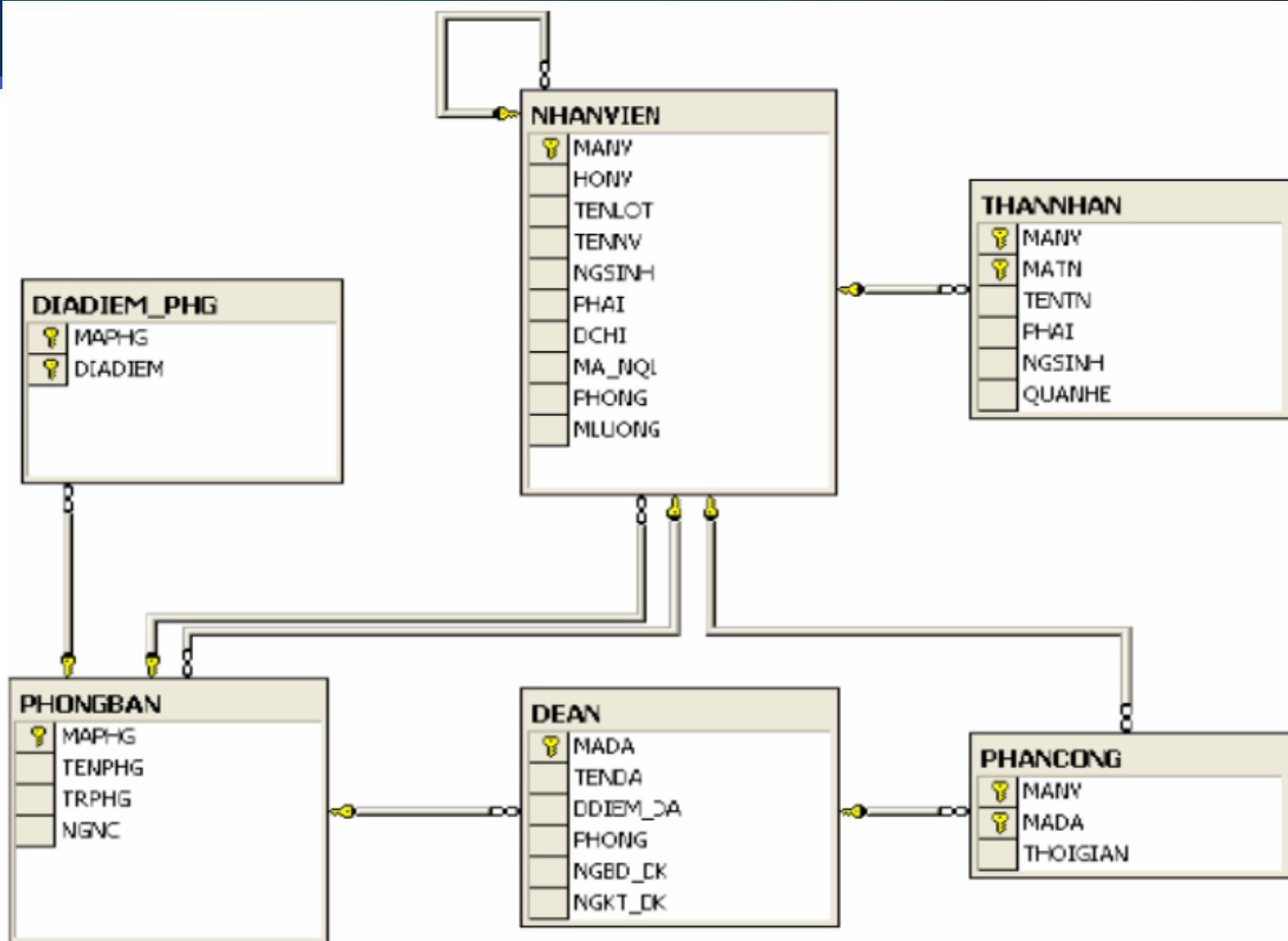
Tên bí danh

~~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_NVIE~~

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



Ví dụ

1. 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
2. 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
3. Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó
4. Tìm họ tên của những nhân viên được "Nguyen Thanh Tung" phụ trách trực tiếp

Ví dụ 1

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

Ví dụ 2

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

Ví dụ 3

Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó

Ví dụ 4

Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp

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

Ví dụ

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

MA_NVIEN	SODA
999887777	10
999887777	30
987987987	10
987987987	30
987654321	10
987654321	20
987654321	30

b. Phép toán tập hợp trong SQL

SQL có cài đặt các phép toán

- Hợp (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

a. Phép toán tập hợp trong SQL

Cú pháp

Hợp

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
UNION [ALL]
```

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

Giao

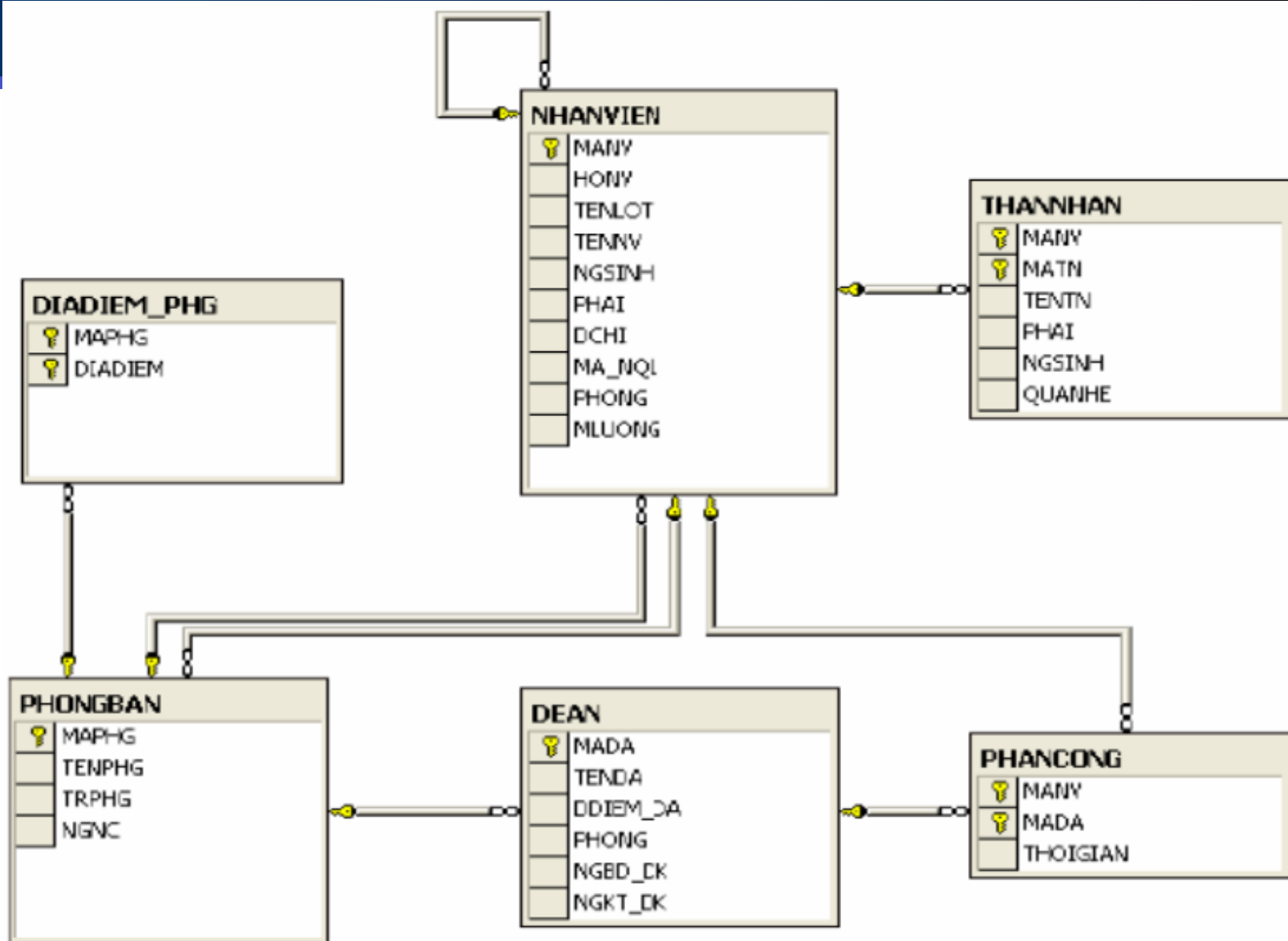
```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
INTERSECT [ALL]
```

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

Trừ

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
EXCEPT [ALL]
```

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

Ví dụ 5

Cho biết các mã đề án có

- Nhân viên với họ là 'Nguyen' tham gia hoặc,
- Trưởng phòng chủ trì đề án đó với họ là 'Nguyen'

SELECT SODA

FROM NHANVIEN NV, PHANCONG PC

WHERE NV.MANV=PC.MA_NVIAN AND HONV='Nguyen'

UNION

SELECT MADA

FROM NHANVIEN NV, PHONGBAN PB, DEAN

WHERE NV.MANV=PB.TRPHG AND MAPHG=PHONG

AND HONV='Nguyen'

Ví dụ 6

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

~~SELECT TENNV, PHAI FROM NHANVIEN
INTERSECT
SELECT TENTN, PHAI FROM THANNHAN~~

SELECT NV.*
FROM NHANVIEN NV, THANNHAN TN
WHERE NV.MANV=TN.MA_NVIAN
AND NV.TENNV=TN.TENTN AND NV.PHAI=TN.PHAI

Ví dụ 7

Tìm những nhân viên không có thân nhân nào

SELECT MANV FROM NHANVIEN

~~**EXCEPT**~~

SELECT MA_NVIEN AS MANV FROM THANNHAN

**select MANV
from NHANVIEN
where MANV not in (select MA_NVIEN
from THANNHAN)**

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

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

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, PHONGBAN  
WHERE TENPHG='Nghien cuu' AND PHG=MAPHG
```

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

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE PHG IN ( 1, 4, 5)
```


Ví dụ 5

SELECT SODA

FROM NHANVIEN NV, PHANCONG PC

WHERE NV.MANV=PC.MA_NVIENT AND NV.HONV='Nguyễn'

UNION

SELECT MADA

FROM NHANVIEN NV, PHONGBAN PB, DEAN DA

WHERE NV.MANV=PB.TRPHG AND PB.MAPHG=DA.PHONG

AND NV.HONV='Nguyễn'

Ví dụ 5

**SELECT DISTINCT TENDA
FROM DEAN
WHERE MADA IN**

**(
SELECT SODA
FROM NHANVIEN, PHANCONG
WHERE MANV=MA_NVIN AND HONV='Nguyễn'
)**

OR MADA IN

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

Ví dụ 7

Tìm những nhân viên không có thân nhân nào

C1 **SELECT ***
 FROM NHANVIEN
 WHERE MANV NOT IN (
 SELECT MA_NVIAN
 FROM THANNHAN)

C2 **SELECT ***
 FROM NHANVIEN
 WHERE MANV <> ALL (
 SELECT MA_NVIAN
 FROM THANNHAN)

Ví dụ 8

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

```
SELECT *
```

```
FROM NHANVIEN
```

```
WHERE LUONG > ANY (
```

```
    SELECT LUONG
```

```
    FROM NHANVIEN
```

```
    WHERE PHG=4 )
```

```
SELECT NV1.*
```

```
FROM NHANVIEN NV1, NHANVIEN NV2
```

```
WHERE NV1.LUONG > NV2.LUONG AND NV2.PHG=4
```

Ví dụ 9

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

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

Ví dụ 10

Tìm những trưởng phòng có tối thiểu một thân nhân

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

Ví dụ - Lồng tương quan

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

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

Ví dụ 6

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

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


Ví dụ 7

Tìm những nhân viên không có thân nhân nào

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

Ví dụ 8

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

```
SELECT *  
FROM NHANVIEN NV1  
WHERE EXISTS (  
    SELECT *  
    FROM NHANVIEN NV2  
    WHERE NV2PHG=4  
    AND NV1.LUONG>NV2.LUONG)
```

Ví dụ 10

Tìm những trưởng phòng có tối thiểu một thân nhân

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

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

Phép chia trong SQL

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

S	D	E
b_i	a	1
	b	1

$R \div S$	A	B	C
a_i	α	a	γ
	γ	a	γ

$R \div S$ là tập các giá trị a_i trong R sao cho không có giá trị b_i nào trong S làm cho bộ (a_i, b_i) không tồn tại trong R

Phép chia trong SQL

Sử dụng NOT EXISTS để biểu diễn

```
SELECT R1.A, R1.B, R1.C
FROM R R1
WHERE NOT EXISTS (
    SELECT *
    FROM S
    WHERE NOT EXISTS (
        SELECT *
        FROM R R2
        WHERE R2.D=S.D AND R2.E=S.E
        AND R1.A=R2.A AND R1.B=R2.B AND
R1.C=R2.C ))
```

Ví dụ 12

Tìm tên các nhân viên được phân công làm tất cả các đồ án

- Tìm tên các nhân viên mà không có đề án nào là không được phân công làm
- Tập bị chia: PHANCONG(MA_NVIEN, SODA)
- Tập chia: DEAN(MADA)
- Tập kết quả: KQ(MA_NVIEN)
- Kết KQ với NHANVIEN để lấy ra TENNV

Ví dụ 12

```
SELECT NV.TENNV
FROM NHANVIEN NV, PHANCONG PC1
WHERE NV.MANV=PC1.MA_NVIENT
AND NOT EXISTS (
    SELECT *
    FROM DEAN DA
    WHERE NOT EXISTS (
        SELECT *
        FROM PHANCONG PC2
        WHERE PC2.SODA=DA.MADA
        AND PC1.MA_NVIENT=PC2.MA_NVIENT ))
```


c. 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 13

Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên

```
SELECT SUM(LUONG), MAX(LUONG), MIN(LUONG),  
AVG(LUONG)  
FROM NHANVIEN
```

Ví dụ 14

Cho biết số lượng nhân viên của phòng 'Nghien cuu'

```
SELECT COUNT(*) AS SL_NV  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG AND TENPHG='Nghien cuu'
```

Ví dụ 15

Cho biết số lượng nhân viên của từng phòng ban

PHG	SL_NV
5	3
4	3
1	1

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
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	5
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

d. Gom nhóm

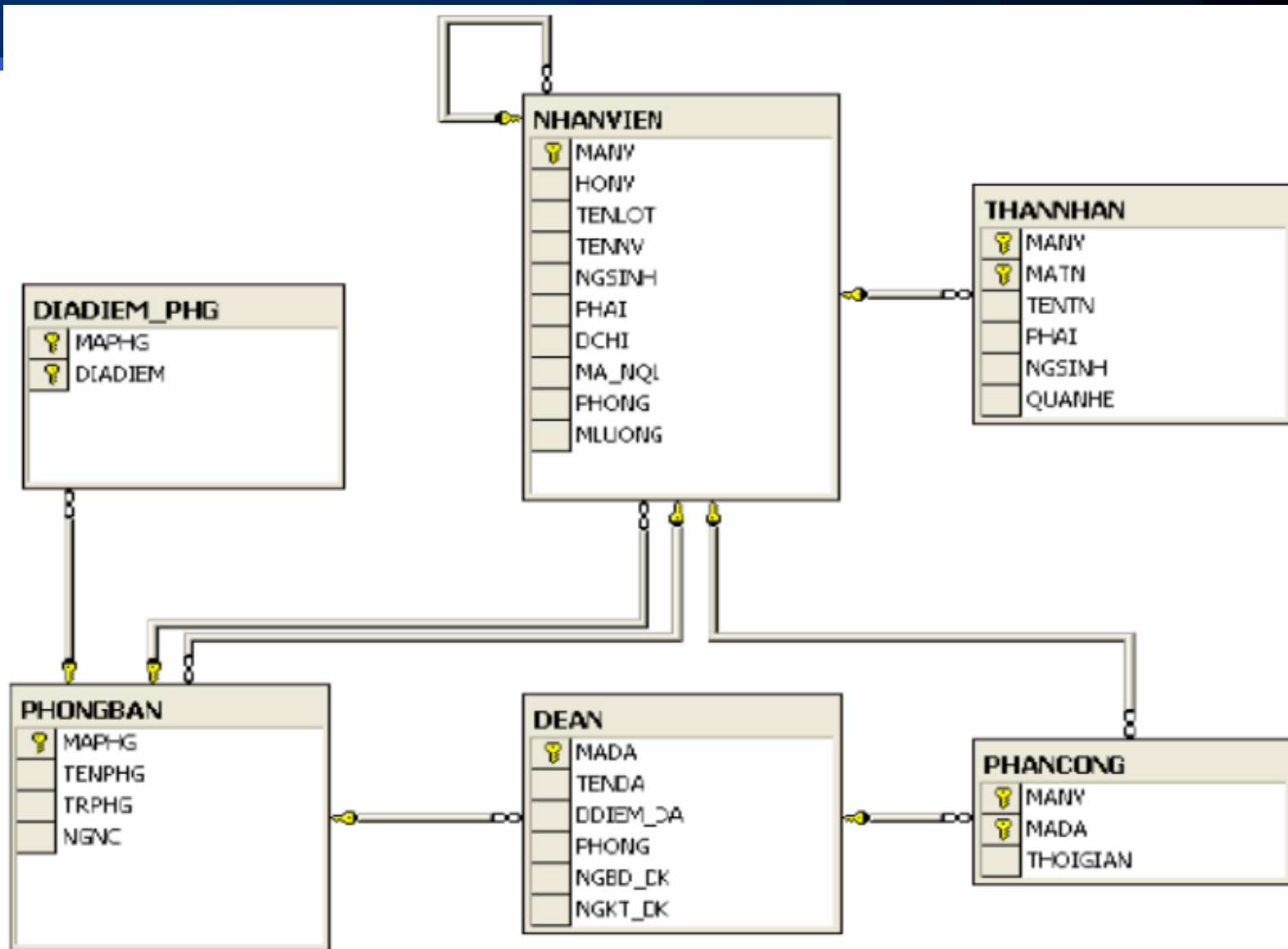
Cú pháp

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các cột gom nhóm>



Ví dụ 16

Cho biết số lượng nhân viên của từng phòng ban

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

```
SELECT TENPHG, COUNT(*) AS SL_NV  
FROM NHANVIEN, PHONGBAN  
WHERE PHONG=MAPHG  
GROUP BY TENPHG
```

Ví dụ 17

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

MA_NVIENT	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

```
SELECT MA_NVIENT, COUNT(*) AS SL_DA,  
       SUM(THOIGIAN) AS TONG_TG  
FROM PHANCONG  
GROUP BY MA_NVIENT
```

```
SELECT HONV, TENNV, COUNT(*) AS SL_DA,  
       SUM(THOIGIAN) AS TONG_TG  
FROM PHANCONG, NHANVIEN  
WHERE MA_NVIENT=MANV  
GROUP BY MA_NVIENT, HONV, TENNV
```


Ví dụ 18

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

MA_NVIENT	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

bị loại ra

e. 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>

Ví dụ 18

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

```
SELECT MA_NVIEN  
FROM PHANCONG  
GROUP BY MA_NVIEN  
HAVING COUNT(*) >= 2
```

Ví dụ 19

Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT PHONG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN  
GROUP BY PHONG  
HAVING AVG(LUONG) > 20000
```

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN, PHONGBAN  
WHERE PHONG=MAPHG  
GROUP BY TENPHG  
HAVING AVG(LUONG) > 20000
```

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

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

Ví dụ 20

Tìm những phòng ban có lương trung bình cao nhất

```
SELECT PHG, AVG(LUONG) AS LUONG_TB
```

```
FROM NHANVIEN
```

```
GROUP BY PHG
```

```
HAVING AVG(LUONG) >= ALL (
```

```
SELECT AVG(LUONG)
```

```
FROM NHANVIEN
```

```
GROUP BY PHG)
```

Ví dụ 21

Tìm tên các nhân viên được phân công làm tất cả các đồ án

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


f. 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

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

Ví dụ 22

Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT PHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN  
GROUP BY PHG  
HAVING AVG(LUONG) > 20000
```

```
SELECT PHG, TENPHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY PHG, TENPHG  
HAVING AVG(LUONG) > 20000
```

Ví dụ 22

Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT TENPHG, TEMP.LUONG_TB
FROM PHONGBAN, (SELECT PHG, AVG(LUONG) AS LUONG_TB
                FROM NHANVIEN
                GROUP BY PHG
                HAVING AVG(LUONG)> 20000 ) AS TEMP
WHERE MAPHG=TEMP.PHG
```

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>
```

Kết ngoài

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

Ví dụ 23

Tìm mã và tên các nhân viên làm việc tại phòng 'Nghien cuu'

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

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

Ví dụ 24

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

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

PHANCONG join DEAN

NHANVIEN

MA_NVIEN=MANV

←
mở rộng

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
```


Ví dụ 25

Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
SELECT HONV, TENNV  
FROM NHANVIEN  
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= ( CASE PHAI  
                                           WHEN 'Nam' THEN 60  
                                           WHEN 'Nu' THEN 55  
                                           END )
```

Ví dụ 26

Cho biết họ tên các nhân viên và năm về hưu

```
SELECT HONV, TENNV
```

```
(CASE PHAI
```

```
    WHEN 'Nam' THEN YEAR(NGSINH) + 60
```

```
    WHEN 'Nu' THEN YEAR(NGSINH) + 55
```

```
END ) AS NAMVEHUU
```

```
FROM NHANVIEN
```

Kết luận

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>]

4. Cập nhật dữ liệu

Cập nhật dữ liệu gồm các lệnh:

- Thêm (insert)
- Xóa (delete)
- Sửa (update)

a. Lệnh INSERT

Dùng để thêm 1 hay nhiều dòng vào bảng

Để thêm dữ liệu

- Tên quan hệ
- 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 ('Le', 'Van', 'Tuyen', '635635635')
```

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV, DCHI)  
VALUES ('Le', 'Van', 'Tuyen', '635635635', NULL)
```

```
INSERT INTO NHANVIEN  
VALUES ('Le', 'Van', 'Tuyen', '635635635', '12/30/1952', '98 HV', 'Nam',  
'37000', 4)
```

a. Lệnh INSERT

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 NOT 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

a. Lệnh INSERT

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 VARCHAR(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
```

1. Tạo một bảng thống kê thông tin như sau (**KHOA_TK**)

- Mã khoa
- Tên khoa
- Số Sv của khoa

b. 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='345345345'
```

```
DELETE FROM NHANVIEN
```

Xóa đi những nhân viên ở phòng 'Nghien cuu'

```
DELETE FROM NHANVIEN
WHERE PHG IN (
    SELECT MAPHG
    FROM PHONGBAN
    WHERE TENPHG='Nghien cuu')
```

b. Lệnh DELETE

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
 - Đặt NULL cho những giá trị tham chiếu

Xóa trong bảng KHOA_TK những
khoa có số sv < 5

c. 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'
```

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
```


c. 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

5. Khung nhìn

Bảng là một quan hệ được tổ chức lưu trữ vật lý trong CSDL

Khung nhìn cũng là một quan hệ

- Không được lưu trữ vật lý (bảng ảo)
- Không chứa dữ liệu
- Được định nghĩa từ những bảng khác
- Có thể truy vấn hay cập nhật thông qua khung nhìn

5. Khung nhìn

Tại sao phải sử dụng khung nhìn?

- Che dấu tính phức tạp của dữ liệu
- Đơn giản hóa các câu truy vấn
- Hiển thị dữ liệu dưới dạng tiện dụng nhất
- An toàn dữ liệu

Định nghĩa khung nhìn

Cú pháp

```
CREATE VIEW <tên khung nhìn> AS  
    <câu truy vấn>
```

```
DROP VIEW <tên khung nhìn>
```

Bảng ảo này có

- Danh sách thuộc tính trùng với các thuộc tính trong mệnh đề **SELECT**
- Số dòng phụ thuộc vào điều kiện ở mệnh đề **WHERE**
- Dữ liệu được lấy từ các bảng ở mệnh đề **FROM**

Ví dụ

```
CREATE VIEW NV_P5 AS
  SELECT MANV, HONV, TENLOT, TENVN
  FROM NHANVIEN
  WHERE PHG=5
```

```
CREATE VIEW TONGLNG_SLVN_PB AS
  SELECT MAPHG, TENPB, COUNT(*) AS SLNV,
    SUM(LUONG) AS TONGLNG
  FROM NHANVIEN, PHONGBAN
  WHERE PHG=MAPHG
  GROUP BY TENPHG
```

Truy vấn trên khung nhìn

Tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn

```
SELECT TENNV  
FROM NV_P5  
WHERE HONV LIKE  
‘Nguyen’
```

$$NV_P5 \leftarrow \pi_{MANV, HONV, TENLOT, TENNV} (\sigma_{PHG=5} (NHANVIEN))$$
$$\pi_{TENNV} (\sigma_{HONV='Nguyen'} (NV_P5))$$

Truy vấn trên khung nhìn

Có thể viết câu truy vấn dữ liệu từ khung nhìn và bảng

```
SELECT HONV, TENVN, TENDA, THOIGIAN  
FROM NV_P5, PHANCONG, DEAN  
WHERE MANV=MA_NVIENT AND SODA=MADA
```

Cập nhật trên khung nhìn

Có thể dùng các câu lệnh INSERT, DELETE và UPDATE cho các khung nhìn đơn giản

- Khung nhìn được xây dựng trên 1 bảng và có khóa chính của bảng

Không thể cập nhật dữ liệu nếu

- Khung nhìn có dùng từ khóa DISTINCT
- Khung nhìn có sử dụng các hàm kết hợp
- Khung nhìn có mệnh đề SELECT mở rộng
- Khung nhìn được xây dựng từ bảng có RB trên cột
- Khung nhìn được xây dựng từ nhiều bảng

Cập nhật trên khung nhìn

Sửa lại họ cho nhân viên mã '123456789' ở phòng 5 là 'Pham'

```
UPDATE NV_P5
```

```
SET HONV='Pham'
```

```
WHERE MANV= '123456789'
```

6. Chỉ mục

Chỉ mục trên thuộc tính A là một cấu trúc dữ liệu làm cho việc tìm kiếm mẫu tin có chứa A hiệu quả hơn

SELECT *

FROM NHANVIEN

WHERE PHG=5 AND PHAI='Nu'

Đọc 10.000 bộ

Đọc 200 bộ

Bảng NHANVIEN có 10.000 bộ

Có 200 nhân viên làm việc cho phòng 5

Đọc 70 bộ

6. Chỉ mục

Cú pháp

```
CREATE INDEX <tên chỉ mục> ON <tên bảng>(<tên cột>)
```

```
DROP INDEX <tên chỉ mục>
```

Ví dụ

```
CREATE INDEX PHG_IND ON NHANVIEN(PHG)
```

```
CREATE INDEX PHG_PHAU_IND ON NHANVIEN(PHG, PHAU)
```

6. Chỉ mục

Nhận xét

- Tìm kiếm nhanh trong trường hợp so sánh với hằng số và phép kết
- Làm chậm đi các thao tác thêm, xóa và sửa
- Tổn chi phí
 - Lưu trữ chỉ mục
 - Truy xuất đĩa nhiều

Chọn lựa cài đặt chỉ mục hợp lý???

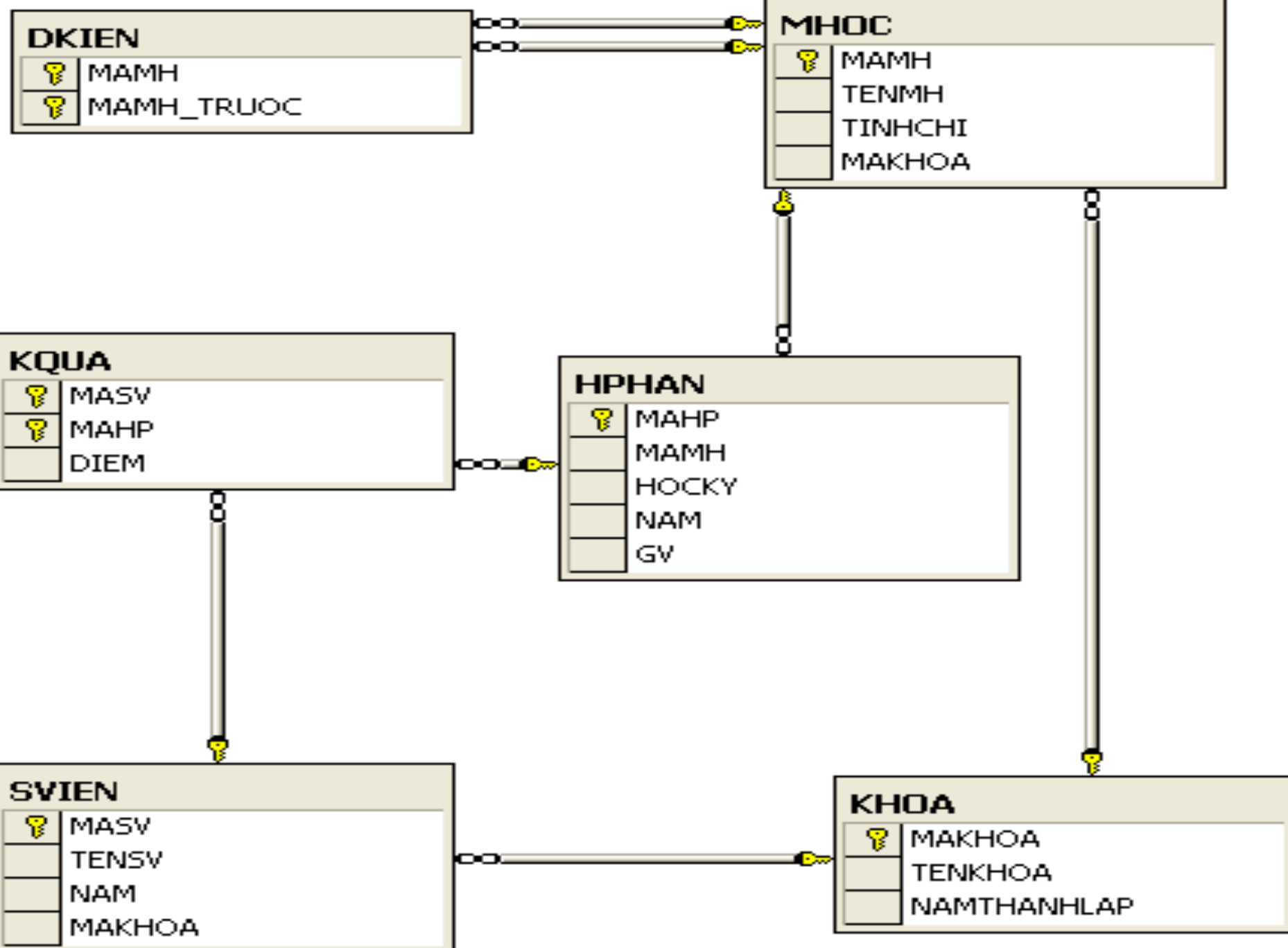
Mở rộng

MỘT SỐ LỆNH SQL THAM KHẢO



TOP

```
SELECT TOP (giatri_dau) [PERCENT] [WITH TIES]  
bieu_thuc  
FROM bang  
[WHERE dieu_kien]  
[ORDER BY bieu_thuc [ ASC | DESC ]];
```



???

Làm gì để rang buộc toàn vẹn khoá ngoài tạm thời bị vô hiệu hoá?

Thank you!

