

Tóm tắt các câu lệnh TSQL đã học

1) Trên cơ sở dữ liệu (database)

Các phiên bản của SQL Server: Enterprise, Personal, Desktop, Developer,...

Các bộ phận của SQL Server: Enterprise Manager, Query Analyzer, Books Online,...

Các loại tập tin của SQL Server: data file (primary, secondary), log file

Các loại cơ sở dữ liệu của SQL Server: system database, user database

create database: tạo cơ sở dữ liệu

```
create database HoSoSinhVien
```

```
create database HoSoSinhVien
on (name='HSSV_data', filename='c:\HSSV_data.mdf')
log on (name='HSSV_log', filename='c:\HSSV_log.ldf')
```

drop database: xóa cơ sở dữ liệu

```
drop database HoSoSinhVien
```

alter database: sửa các thông tin của cơ sở dữ liệu

```
alter database HoSoSinhVien
modify name = HSSV
```

exec sp_dboption: sửa các thông tin của cơ sở dữ liệu

```
exec sp_dboption HoSoSinhVien, 'read only', 'true'
exec sp_dboption HoSoSinhVien, 'autoshrink', 'true'
exec sp_dboption HoSoSinhVien, 'single_user'
```

dbcc: điều khiển cơ sở dữ liệu

```
dbcc shrinkdatabase(HoSoSinhVien, 10)
```

2) Trên bảng (table)

Nhớ các kiểu số liệu: text, binary, numeric, money, datetime, bit, variant

Nhớ các ràng buộc: default, check, unique, foreign, primary

Nhớ các kiểu toàn vẹn: entity, domain, referential, user

Nhớ các thuộc tính bổ trợ: identity, null

create table: tạo bảng

Tạo bảng với các cột

```
create table SinhVien (
    MaSV int,
    TenSV nchar(50)
)
```

Tạo với các ràng buộc

```
create table SinhVien (  
    MaSV int primary key,  
    TenSV nvarchar(50) not null,  
    QueQuan int references DiaPhuong(MaDP)  
)
```

drop table: xóa bảng

```
drop table SinhVien
```

alter table..add: thêm cột

Thêm cột bằng cách dùng lệnh alter table

```
alter table SinhVien  
    add QueQuan int
```

alter table..drop column: xóa cột

```
alter table SinhVien  
    drop column QueQuan
```

alter table..alter column: thêm thuộc tính not null

```
alter table SinhVien  
    alter column TenSV nchar(50) not null
```

alter table..add primary key: thêm khóa chính

Thêm ràng buộc khóa chính cần phải biến một cột null thành not null trước khi thêm. Nếu để lệnh chuyển đổi thuộc tính not null ngay cạnh lệnh thêm khóa chính thì phải chèn từ khóa go vào vì có thể lệnh trên chưa được thực hiện do đó lệnh dưới báo lỗi

```
alter table SinhVien  
    alter column MaSV int not null  
  
go  
  
alter table SinhVien  
    add primary key (MaSV)
```

alter table..add foreign key: thêm ràng buộc khóa ngoài

```
alter table SinhVien  
    add foreign key (QueQuan) references DiaPhuong(MaDP)
```

alter table..add default: thêm ràng buộc mặc định

```
alter table SinhVien  
add default 'khong ten' for TenSV
```

exec sp_help: xem thông tin bảng

```
exec sp_help SinhVien
```

3) Trên bản ghi (record)

Nhớ thêm phần biểu thức điều kiện

Nhớ thêm phân ký tự thay thế

insert..values: thêm các bản ghi vào bảng

```
insert into SinhVien (MaSV, TenSV, QueQuan)  
values (1, N'Nguyễn Văn A', 1)
```

insert..select: thêm các bản ghi từ bảng khác vào bảng

```
insert into DocGia  
select MaSV, TenSV, QueQuan from SinhVien  
insert into DocGia  
select MaGV, TenGV, QueQuan from GiaoVien
```

select..into: đưa kết quả lựa chọn vào bảng mới

```
select MaSV, TenSV, Diem  
into SinhVienKha  
from SinhVien  
where Diem > 7.0
```

delete: xóa các bản ghi từ bảng

```
delete from SinhVien  
delete from SinhVien where MaSV=1234
```

truncate: xóa toàn bộ bản ghi của bảng

```
truncate table SinhVien
```

update: sửa các bản ghi trong bảng

```
update SinhVien  
set NhómTruong = 1  
where MaSV < 4
```

```
update SinhVien
    set NhomTruong = 4
    where (MaSV > 3 and MaSV < 8)
```

4) Truy vấn (query)

Hỗ trợ truy vấn: distinct, top, as, identity

Phép toán tập hợp: in, like, between

Các hàm tổng nhóm: sum, max, min, avg

4.1) Truy vấn đơn giản

select *: Hiện tất cả bảng

```
select *
from SinhVien
```

select: Hiện một số cột

```
select TenSV, DiemTB
from SinhVien
```

select..where: Hiện một số dòng / bản ghi

```
select TenSV, DiemTB
from SinhVien
where DiemTB > 6.0
```

select..order by: Hiện và sắp xếp theo điểm rồi theo tên

```
select TenSV, DiemTB
from SinhVien
order by DiemTB desc, TenSV asc
```

select..distinct: Hiện danh sách giá trị không trùng lặp

```
select distinct QueQuan
from SinhVien
```

select..top: Hiện các dòng đầu tiên trong bảng

```
select top 3 TenSV, DiemTB
from SinhVien
order by DiemTB desc, TenSV asc
```

4.2) Truy vấn lồng nhau (nested query)

select..where (select)

Hiện tất cả những người trong bảng nhân viên có lương bằng lương lớn nhất của những người có trong công ty:

```
select TenNV, Luong
from NhanVien
where Luong = (select max(Luong) from NhanVien)
```

select..where (in)

Hiện tất cả những người trong bảng nhân viên có lương lớn nhất hoặc lớn nhì của những người có trong công ty:.

```
select TenNV, Luong
from NhanVien
where Luong in (select top 2 Luong from NhanVien order by Luong)
```

Câu lệnh select trong sẽ tạo ra một tập hai giá trị (top 2) đó là lương lớn nhất và lương lớn nhì. Và câu lệnh select thứ nhất sẽ chọn ra những người mà lương nằm trong tập lớn nhất và lớn nhì

select..where (in sub)

Hiện ra tất cả những người có lương lớn nhất phòng của anh ta (không phải lớn nhất trong công ty mà lớn nhất trong phòng hoặc đơn vị mà anh ta thuộc về)

```
select nv1.TenNV, nv1.Luong
from NhanVien as nv1
where nv1.Luong = (select max(Luong) from NhanVien where
Phong=nv1.Phong)
```

Câu lệnh select trong sẽ trả về giá trị lương lớn nhất nhưng không phải lớn nhất trong toàn công ty mà lớn nhất trong phòng của nv1. Sau đó câu lệnh select ngoài cùng sẽ xác định xem nv1 có được chọn không bằng cách kiểm tra lương anh ta với lương lớn nhất của phòng anh ta.

4.3) Truy vấn tổng nhóm (subtotal query / grouping query)

select..group by: Thống kê theo tiêu chí

Hiện ra số lượng các nhân viên ứng với từng quê

```
select QueQuan, count(*)
from NhanVien
group by QueQuan
```

Đếm số nam và số nữ trong công ty

```
select GioiTinh, count(*)
from NhanVien
group by GioiTinh
```

Tính tổng thu nhập theo từng phòng

```
select Phong, sum(Luong)
from NhanVien
group by Phong
```

select..having: Hiện ra một số nhóm phù hợp

Chỉ đếm số lượng người ở Hải Phòng và số lượng người ở Hà nội

```
select QueQuan, count(*)
from NhanVien
group by QueQuan
having (QueQuan = 'HP', QueQuan = 'HN')
```

Chỉ hiện ra những phòng nào có tổng thu nhập lớn hơn 500000

```
select Phong, sum(Luong)
from NhanVien
group by Phong
having sum(Luong) > 5000000
```

Chỉ hiện ra những tỉnh nào có số lượng người lớn hơn 10

```
select QueQuan, count(*)
from NhanVien
group by QueQuan
having count(*) > 10
```

4.4) Truy vấn liên bảng (cross table query / joining query)

select..inner join: ghép các cặp bản ghi thỏa mãn điều kiện

Ghép bảng nhân viên và hiện ra tên nhân viên và tên địa phương

```
select NhanVien.TenNV, DiaPhuong.TenDP
from NhanVien inner join DiaPhuong on NhanVien.QueQuan =
DiaPhuong.MaDP
```

select..left outer join: lấy tất cả phía trái và ghép (nếu có) với phải

Lấy tất cả những nhân viên kể cả những nhân viên có quê quán không hợp lệ (nghĩa là mã quê quán không có trong bảng địa phương)

```
select NhanVien.TenNV, DiaPhuong.TenDP
from NhanVien left outer join DiaPhuong on NhanVien.QueQuan =
DiaPhuong.MaDP
```

select..right outer join: lấy tất cả phía phải và ghép (nếu có) với phía trái

Lấy tất cả những địa phương ghép với nhân viên, các địa phương không hợp lệ sẽ được ghép với bộ dữ liệu rỗng. Không hiện ra các nhân viên không có mã quê quán phù hợp

```
select NhanVien.TenNV, DiaPhuong.TenDP
from NhanVien right outer join DiaPhuong on NhanVien.QueQuan =
DiaPhuong.MaDP
```

select..full outer join: lấy từ hai phía và ghép nếu có

Lấy tất cả những nhân viên (nếu không có quê quán phù hợp thì ghép với bộ dữ liệu rỗng) và tất cả những địa phương kể cả không có nhân viên.

```
select NhanVien.TenNV, DiaPhuong.TenDP
from NhanVien right outer join DiaPhuong on NhanVien.QueQuan =
DiaPhuong.MaDP
```

select..cross join: trả về tất cả các cặp có thể ghép

Ghép từng nhân viên với tất cả các địa phương. Như vậy nếu có m nhân viên và có n địa phương thì bảng đích sẽ có m*n dòng. n dòng đầu cho nhân viên thứ nhất ghép với các địa phương. n dòng sau cho nhân viên thứ hai ghép với các địa phương. và tiếp tục như thế tới nhân viên thứ m.

```
select NhanVien.TenNV, DiaPhuong.TenDP
from NhanVien cross join DiaPhuong
```

ref: Tham khảo thêm một số lệnh

Xem tất cả các CƠ SỞ DỮ LIỆU người dùng trong MÁY CHỦ

Mỗi khi một cơ sở dữ liệu được tạo ra, bảng sysdatabases trong cơ sở dữ liệu master sẽ chứa thông tin về bảng mới tạo ra đó. Do vậy chúng ta có thể liệt kê tất cả các cơ sở dữ liệu trong một máy chủ bằng cách liệt kê các bản ghi trong bảng sysdatabases

```
select *
from master.dbo.sysdatabases
where sid <> 1
```

Xem tất cả các BẢNG người dùng trong CƠ SỞ DỮ LIỆU

Mỗi khi một bảng được tạo ra trong một cơ sở dữ liệu, thông tin về bảng đó sẽ được chứa trong bảng sysobjects của chính cơ sở dữ liệu chứa bảng vừa tạo. Vì vậy chúng ta có thể xem danh sách các bảng trong cơ sở dữ liệu bằng cách truy vấn bảng sysobjects. Các bảng do người dùng tạo có kiểu là 'U'

```
select *
from Northwind.dbo.sysobjects
where xtype='U'
```

Trong ví dụ này, chúng ta lấy tất cả các bản ghi bảng sysobjects của cơ sở dữ liệu Northwind nhưng chỉ lấy những bảng người dùng, nghĩa là những bảng có xtype là 'U'

Xem tất cả các CỘT thuộc về một BẢNG

Mỗi khi một cột trong bảng được tạo, bảng syscolumns trong cơ sở dữ liệu sẽ chứa thông tin về cột vừa tạo. Thuộc tính id của cột sẽ chứa định danh của bảng mà cột đó

thuộc về. Vì vậy để xem các cột trong một bảng dữ liệu, chúng ta có thể truy vấn bảng syscolumns

```
declare @x int
set @x = (select id from sysobjects where name='Employees')
select * from syscolumns where id = @x
```

Trong ví dụ này, ta khai báo biến @x kiểu int, sau đó lấy id của bảng Employees vào biến @x, rồi ta lấy tất cả các cột trong bảng syscolumns mà có id bằng @x, cũng có nghĩa là id của bảng Employees. Do vậy ta sẽ liệt kê được tất cả các cột trong bảng Employees

Sử dụng bảng tạm

Khai báo bảng, cập nhật bảng, và hiện bảng

```
declare @x table(MaSV int, TenSV nvarchar(20))

insert into @x values(1, N'Thường')
insert into @x values(2, N'Yến')

select * from @x
```

Khai báo bảng, tải bảng khác từ trong cơ sở dữ liệu lên bảng vừa khai báo, và hiện bảng

```
declare @x table(MaSV int, TenSV nvarchar(20))

insert into @x
select EmployeeID, firstname
from Northwind.dbo.Employees

select * from @x
```

Sử dụng biến tạm

```
declare @t money
set @t = (select max(UnitPrice) from Northwind.dbo.products)
select * from Northwind.dbo.products where UnitPrice >= @t
```

ref: Các khái niệm cơ bản của cơ sở dữ liệu

Các mô hình dữ liệu (data model)

- + Mô hình thứ bậc (hierarchical)
- + Mô hình mạng lưới (network)
- + Mô hình quan hệ (relational)

Các vấn đề của lưu trữ

- + Sự dư thừa (redundance)
- + Sự nhất quán (consistence)
- + Tính toàn vẹn (integrity)

- + Sự an toàn (security)
- + Sự chia sẻ (sharing)

Các khái niệm cơ bản

- + Cơ sở dữ liệu (database)
- + Bảng (table) hoặc quan hệ (relation)
- + Bản ghi (record) hoặc dòng (row) hoặc bộ (tuple)
- + Trường (field) hoặc cột (column)
- + Mỗi quan hệ (relationship) được biểu hiện bằng bảng (table) hoặc khóa (key)

Mối quan hệ giữa các

- + Tại sao phải lưu mối quan hệ
- + Phương pháp lưu các mối quan hệ (dùng khóa, dùng bảng)
- + Khóa chính (primary key) và khóa ngoài (foreign key)
- + Vấn đề toàn vẹn tham chiếu (referential integrity)

Ngôn ngữ định nghĩa và xử lý dữ liệu

- + Thao tác trên các đối tượng như cơ sở dữ liệu, bảng, cột
- + Phần định nghĩa cơ sở dữ liệu và bảng: create, drop, alter
- + Phần cập nhật dữ liệu: insert, delete, update
- + Phần truy vấn dữ liệu: join, union, projection, selection, sort, group
- + Phần lập trình và thao tác: declare, set, use, go

ref: Liên quan tới định nghĩa bảng

Các vấn đề về toàn vẹn dữ liệu

- + Toàn vẹn thực thể (entity integrity): primary key, unique, identity
- + Toàn vẹn tham chiếu (referential integrity): foreign key, check
- + Toàn vẹn miền dữ liệu (domain integrity): default, foreign key, check, not null
- + Toàn vẹn người dùng (user integrity): rules, stored procedures, triggers

Các hỗ trợ với toàn vẹn

- + Ràng buộc kiểm tra (check constraint): phải thỏa mãn điều kiện nào đó
- + Ràng buộc mặc định (default constraint): phải có một giá trị mặc định
- + Ràng buộc duy nhất (unique constraint): các giá trị trong cột không được trùng lặp
- + Ràng buộc khóa ngoài (foreign key constraint): các giá trị phải hợp lệ với cột khóa chính tương ứng
- + Ràng buộc khóa chính (primary key constraint): phải duy nhất và không trống
- + Thuộc tính định danh (identity property): có thể được tăng tự động
- + Thuộc tính không trống (not null property): không được phép để trống

Các kiểu dữ liệu

Giá trị nhị phân: bit

Các số nguyên: bigint, int, smallint, tinyint

Các số thực xấp xỉ: float, real

Giá trị số thực chính xác: decimal, numeric

Giá trị tiền tệ: money, smallmoney

Giá trị thời gian: datetime, smalldatetime

Các chuỗi: char, varchar, text, nchar, nvarchar, ntext,

Các chuỗi nhị phân: binary, varbinary, image

Các kiểu khác: cursor, table, variant, timestamp, uniqueidentifier