# CHƯƠNG I) TỔNG QUAN SQL SEVER

1. **Các câu lệnh:**
2. **Cú pháp tạo Database:**

**CREATE DATABASE** *database\_name (Tên cơ sở dữ liệu)*

[ON

[PRIMARY] [<filespec>[,… n]

[, <file group> [,… n]

[LOG ON{<filespec>[,… n] } ] ***=> Lịch sử giao dịch***

]

]

[;]

<filespace>::=

{

NAME = logical\_file\_name ***=> Tên Lo-gic***

FILE NAME = ‘os\_file\_name’ , ***=> Tên file đặt***

**Đường dẫn tới file cơ sở dữ liệu**

}

**VD 1.1:**

* Tạo Database “My DBT” với tập tin dữ liệu chính là “ My DBT\_Data.mdf ”.
* Dung lượng khởi tạo là 10 MB; tối đa là 100 MB.
* Độ gia tăng kích thước là 10%.
* Tập tin bản ghi giao dịch là “ My DBT ”\_log.ldf
* Dung lượng ban đầu là 20 MB và kích thước tối đa không giới hạn.
* Độ gia tăng dung lượng là 10 MB.

**LG:**

**CREATE DATABASE** My DBT

[ON

**PRIMARY**

(

Name = My DBT\_Data.mdf ,

File name = ‘ C:\HeQuanTriCSDL\My DBT\_Data.mdf ’,

Size = 10 MB,

MaxSize = 100 MB,

File Growth = 10%

**)**

**LOG ON**

**(**

Name = My DBT\_log,

File Name =‘ C:\HeQuanTriCSDL\My DBT\_Data.ldf ’,

Size = 20 MB,

MaxSize = UNLIMITED

File Growth = 10%

**)**

**VD 1.2:**

* Tạo một Database K59 HTTT với tập tin dữ liệu chính là K59 HTTT.mdf:
* Dung lượng khởi tạo là 10 MB; dung lượng tối đa là 100 MB;
* Độ gia tăng kích thước là 10 MB
* Tập tin bản ghi giao dịch là K59 HTTT.log.ldf:
* Dung lượng ban đầu là 100 MB và kích thước tối đa là 200 MB;
* Độ gia tăng dung lượng là 10%.

**LG:**

**CREATE DATABASE K59 HTTT**

**[ON**

**PRIMARY**

**(**

Name = K59 HTTT.mdf,

File Name = ‘D:\DanhSachSinhVien\K59 HTTT.mdf’,

Size = 10 MB,

MaxSize = 100 MB,

FileGrowth = 10 MB

**)**

**LOG ON**

**(**

Name = K59 HTTT\_log.ldf,

FileName = ‘D:\DanhSachSinhVien\K59 HTTT\_log.ldf’,

Size = 100 MB,

MaxSize = 200 MB,

FileGrowth = 10%

**)**

1. **Cú pháp xóa một Database:**

|  |
| --- |
| **DROP DATABASE *database\_name[,…n]*** |

**VD1.3: Xóa Database My DBT?**

* **LG:**

DROP DATABASE My DBT

1. **Cú pháp đổi tên Database:**

|  |
| --- |
| **EXEC SP\_RENAMEDB ‘ <Tên cũ>’ , ‘<Tên mới>’** |

**VD: Đổi tên Database “HTTT” thành “HE THONG THONG TIN”?**

* **LG:**

EXEC SP\_RENAME DB ‘HTTT’,‘HE THONG THONG TIN’

1. **Cú pháp sửa đổi Database:**

* **Một số cú pháp sửa đổi Database:**
* ALTER DATABASE <Tên Database>.
* ADD FILE <Thông tin file dữ liệu> [,… n].
* ADD LOG FILE <Thông tin file log> [,…n].
* REMOVE FILE <Tên lo-gic>. (Xóa)
* MODIFY FILE <Thông tin file>. (Sửa đổi)

1. **Bảng – Table**
2. **Tạo bảng:**

* **Cú pháp tạo bảng:**

|  |
| --- |
| **CREATE TABLE** Table\_name  **(** { <column\_definition> | <table\_constraint>}  [,…n]  **)** |

**VD1: Tạo bảng LOP:**

USE QuanLyDiemSinhVien

GO

CREATE TABLE LOP

(

MaLop char(10) NOT NULL,

TenLop nvarchar(50) NOT NULL,

Khoa char(10)

CONSTRAINT RB\_KHOA C PRIMARY KEY(MaLop)

)

**VD2:**

**CREATE TABLE My Custorness**

**(**

Cust ID int

INDENTITY (100,1)

PRIMARY KEY.Company Name(50)

**)**

**VD3:** Cho một cơ sở dữ liệu gồm các bảng sau:

Diem(**MaSV**, ***MaMon***, Diem)

MonHoc(**MaMon**, TenMon, SoTC)

Lop(**MaLop,** TenLop, SiSo)

|  |
| --- |
| **\*Note: Từ khóa cần viết đúng:**   * **Primary Key:** Khóa chính. * **Foreign Key:** Khóa ngoại. * **Constraint**: Ràng buộc * **RBKC:** Ràng buộc khóa chính. * **RBKN:** Ràng buộc khóa ngoại. * **Check:** Điều kiện cần kiểm tra. |

1. **Tạo bảng Diem (MaSV, MaMon, Diem)**

CREATE TABLE Diem

(

MaSV int, MaMon int,

Diem int Constraint RBKC3 Primary Key

(MaSV, MaMon), Constraint RBKN3 Foreign Key

(MaSV) REFERENCES

SinhVien (MaSV), Constraint RBKN4 Foreign Key

(MaMon) REFERENCES MonHoc (MaMon)

)

1. **Tạo bảng MonHoc (MaMon, TenMon, SoTC)**

CREATE TABLE MonHoc

(

MaMon char(10), TenMon nchar(50), SoTC int

Constraint RB1 Primary Key (MaMon),

Constraint RB2 Check(SoTC >=1 and SoTC <= 15)

)

1. **Tạo bảng Lop (MaLop, TenLop, SiSo)**

**CREATE TABLE Lop**

**(**

MaLop int, TenLop nchar(50), SiSo int,

Constraint RB1 Primary Key (MaLop),

Constraint RB2 Check (SiSo > 0 and SiSo < 100)

**)**

**VD4: Thêm một file chứa dữ liệu cho Database HTTT?**

**LG:**

**ALTER DATABASE** HTTT

**ADD FILE**

**(**

Name = HTTT2,

File Name = ‘D:\HTTT.ndf’,

Size = 5 MB,

MaxSize = 100 MB,

FileGrowth = 5 MB

**)**

**VD5: Thêm hai file Log kích thước 5 MB cho Database HTTT?**

**LG:**

**ALTER Database** HTTT

**ADD** Log File

**(**

Name = HTTTLogFile2,

FileName = ‘D:\HTTTLogFile2.ldf’,

Size = 5 MB,

MaxSize = 100 MB,

FileGrowth = 5 MB

**)**

**(**

Name = HTTTLogFile3,

FileName = ‘D:\HTTTLogFile3.ldf’,

Size = 5 MB,

MaxSize = 100 MB,

FileGrowth = 5 MB

**)**

**VD6: Xóa File Log HTTTLogfile2 vừa tạo ra cho Database HTTT**

**LG:**

**ALTER DATABASE** HTTT

**Remove File** HTTTLogfile2

1. **Thay đổi cấu trúc bảng**
2. **Thêm một cột vào bảng có sẵn**

* **Cú pháp:**

|  |
| --- |
| ALTER TABLE <Tên bảng> ADD <Tên cột><Kiểu dữ liệu 1>[,<Tên cột n><Kiểu dữ liệu n>,…] |

**VD:** NhanVien (MaNV, TenNV, GioiTinh).

Thêm cột ngày sinh vào bảng trên?

LG:

ALTER TABLE NhanVien ADD NgaySinh DateTime

1. **Chỉnh sửa cột trong bảng**

* **Cú pháp:**

|  |
| --- |
| ALTER TABLE <Tên bảng> ALTER COLUMN <Tên cột 1><Kiểu dữ liệu 1>, <Tên cột n><Kiểu dữ liệu n> |

**VD:** Sửa cột TenNV trong bảng NhanVien thành cột có kiểu dữ liệu là nvarchar(20)?

**LG:**

ALTER TABLE NhanVien ALTER COLUMN TenNV nvarchar(20)

1. **Xóa cột trong bảng**

* **Cú pháp:**

|  |
| --- |
| **ALTER TABLE <Tên bảng> DROP COLUMN <Tên cột 1>, <Tên cột n>,…** |

* **VD:** PhongBan(MaPhong, TenPhong, DiaDiem)

Xóa cột DiaDiem trong bảng PhongBan?

**LG:**

ALTER TABLE PhongBan DROP COLUMN DiaDiem

1. **Thêm ràng buộc khóa chính**

* **Cú pháp:**

|  |
| --- |
| ALTER TABLE <Tên Bảng> ADD CONTRAINT <Tên ràng buộc>  PRIMARY KEY (<Danh sách các cột làm khóa chính>) |

* **Ví dụ:** KetQua (MaSV, MaMH, KetQua).

Thêm khóa chính cho bảng KetQua?

**LG:**

ALTER TABLE KetQua ADD CONTRAINT RBKC

PRIMARY KEY (MaSV, MaMH)

1. **Xóa bảng**

* **Cú pháp:**

|  |
| --- |
| DROP TABLE <Tên bảng cần xóa> |

* **Chú ý khi xóa bảng:**
* Nếu không có tham chiếu vòng thì tiến hành xóa bảng chứa khóa ngoại trước sau đó xóa bảng còn lại, hoặc xóa khóa ngoại rồi sau đó tiến hành xóa các bảng.
* Nếu có tham chiếu vòng thì xóa một khóa để làm mất tham chiếu vòng rồi tiến hành như trường hợp 1.
* Có thể sử dụng câu lệnh làm vô hiệu hóa các RB trên bảng trước khi xóa bảng.

ALTER TABLE<Tên Bảng> NoCheck Constraint All

1. **Cập nhật dữ liệu cho bảng**
2. **Thêm dữ liệu**

* **Cú pháp:**

|  |
| --- |
| Insert Into <Tên Bảng> [<Danh sách các cột>]  Values (<Danh sách các giá trị tương ứng>) |

* **Ví dụ:** Cho bảng: MonHoc (MaMH, TenMH, SoTC).

Thêm vào bảng trên với MaMH là TDC01, TenMH là “Tin học Đại cương”, SoTC = 3

LG:

* Cách 1: Insert Into MonHoc

Values (‘TDC01’, ‘Tin hoc Dai cuong’, 3)

* Cách 2: Insert Into MonHoc

Values (‘TDC01’, ‘N Tin học Đại cương, 3)

**🟋 Chú ý:**

* Để viết Tiếng Việt trong SQL, ta thêm kí tự “N” trước chuỗi.
* Nếu thuộc tính được khai báo là “NOT NULL” thì bắt buộc phải có giá trị khi nhập một bộ dữ liệu vào bảng.

1. **Sửa dữ liệu bảng**

* **Cú pháp:**

**UPDATE** <Tên bảng>

**SET** <Tên cột 1> = <Giá trị 1>,…, <Tên cột n> = <Giá trị n>

[ **Where** <Điều kiện> ]

* **VD:** Cập nhật SoTC cho môn TDC01 tăng 1?

**LG:**

**UPDATE** MonHoc

**SET** SoTC = SoTC + 1

**Where** MaMH = ‘TDC01’

* **VD:** Cập nhật SoTC môn TDC01 thành 4?

**LG:**

**UPDATE** MonHoc

**SET** SoTC = 4

**Where** MaMH = ‘TDC01’

* **VD:** Cho bảng: Diem (MaMon, MaSV, Diem).

Sửa điểm môn học có mã là “MH01” của sinh viên SV01 thành điểm 10?

LG:

**UPDATE** Diem

**Set** Diem = 10

**Where** MaMon = ‘MH01’ and MaSV = ‘SV01’

1. **Xóa dữ liệu bảng**

* **Cú pháp:**

|  |
| --- |
| Delete From <Tên bảng>  [ Where <Điều kiện chọn> ] |

* **Ví dụ:** Cho hai bảng:

MonHoc (MaMon, TenMon, SoTC)

Diem (MaMon, MaSV, Diem)

Xóa các môn có SoTC < 2?

LG:

Delete From MonHoc

Where SoTC < 2

* **Ví dụ:** Xóa bảng Diem của sinh viên có MaSV là SV01?

**LG:**

**Delete From** Diem

**Where** MaSV = ‘SV01’

* **Ví dụ:** Xóa bảng Diem của sinh viên tên là “Nguyen Thanh Lam” ?

**LG:**

Delete From Diem

Where MaSV =

**(**Select MaSV

From SinhVien

Where TenSV = ‘Nguyen Thanh Lam’ **)**

1. **Thêm ràng buộc khóa chính**

* **Cú pháp**

|  |
| --- |
| ALTER Table <Tên bảng>  Add Constraint <Tên ràng buộc>  Primary Key ( <Danh sách các cột làm khóa chính> ) |

* **Ví dụ: Cho bảng:** KetQua (MaSV, MaMH, KetQua)

Thêm khóa chính cho bảng KetQua?

**LG:**

Alter Table KetQua

Add Constraint RBKC

Primary Key (MaSV, MaMH)

1. **View**
2. **Khái niệm:**

* View là một bảng ảo được định nghĩa bởi một truy vấn phát biểu Select
* View được hình thành dữ liệu từ một hoặc nhiều bảng thật.

1. **Lệnh tạo View**

Create View <Tên View> [ <Danh sách các cột> ]

As <Câu lệnh Select>

**VD:** Tìm sinh viên có tên là “Nguyen”?

**LG:**

Create View DanhSachSV (MaSinhVien, HoTenSV)

As

Select MaSV, HoTen

From SinhVien

Where HoTen like ‘\*Nguyen’

***\*Chú ý:*** *“like” là toán tử so khớp, thường đi kèm với những kí tự đặc biệt.*

**VD:** Tìm ra các sinh viên có điểm thi của các môn đăng kí học?

**LG:**

Create View DSSVDKHoc

As

Select MaSV As ‘MaSinhVien’

From Diem

Group by MaSV

**VD: Cho Cơ sở dữ liệu sau:**

SV (***MaSV***, HoTenSV, GTinh, NgaySinh)

MonHoc (***MaMon***, TenMon, SoTC)

Diem (***MaMon***, ***MaSV***, Diem)

1. **Tạo View chứa Họ tên sinh viên của những sinh viên nữ có điểm môn Hệ quản trị cơ sở dữ liệu trên 8?**

**LG:**

* **Cách 1:**

Select HoTenSV

From SV

Where GTinh = ‘Nu’ and MaSV in

(Select MaSV

From DIEM

Where MaMon = ‘He Quan Tri CSDL ‘ and Diem > 8)

* **Cách 2:**

Select HoTenSV

From SV, MonHoc, Diem

Where SV.MaSV = Diem.MaSV and

MonHoc.MaMon = Diem.MaMon and

TenMon = ‘He Quan Tri CSDL’and GTinh = ‘Nu’ and

Diem > 8

1. **Tạo View chứa MaSV, HoTenSV, của những sinh viên có học môn MH01 nhưng không học môn MH02?**

**LG:**

Select HoTenSV

From SV

Where MaSV in

(

Select MaSV

From DIEM

Where MaMon = ‘MH01’

)

And MaSV Not in

(

Select MaSV

From DIEM

Where MaMon = ‘MH02’

)

1. **Tạo View chứa MaSV, HoTenSV, của những sinh viên có học môn MH01 và MH02?**

**LG:**

Select HoTenSV

From SV

Where MaSV in

(

Select MaSV

From DIEM

Where MaMon = ‘MH01’

)

And MaSV in

(

Select MaSV

From DIEM

Where MaMon = ‘MH02’

)

# CHƯƠNG II) LẬP TRÌNH VỚI SQL SEVER

1. **Biến và cách sử dụng biến**

* Biến là một đối tượng có thể chứa giá trị thuộc một kiểu dữ liệu nhất định.
* Tên biến bắt đầu bằng kí tự “@”.

1. **Biến cục bộ:**

* ***Cú pháp khai báo biến cục bộ:***

Declare @<Tên biến> <Kiểu dữ liệu>, …,<Tên biến n> <Kiểu dữ liệu n>

* **Ví dụ:**

Declare @HoTen nvarchar(30)

* Chú ý: Kiểu “text; ntext; image” không chấp nhận khi khai báo biến.
* ***Gán giá trị cho biến:***

|  |
| --- |
| Set @<Tên biến> = <Giá trị>  hoặc  Select @<Tên biến> = <Giá trị> |

* Giá trị gán cho biến phải phù hợp với kiểu dữ liệu của biến.
* Giá trị ở đây có thể là giá trị cụ thể, tên biến, biểu thức hoặc kết quả truy vấn.
* ***Hiển thị ra màn hình:***
* ***Hiển thị giá trị hiện hành của biến:***

Print @<Tên biến> hoặc Select @<Tên biến>

* ***Hiển thị chuỗi kí tự hoặc số***

Print ‘<Chuỗi kí tự>’ / <Số>

***Ví dụ 1: Hiển thị ra giá trị của biến x:***

Print ‘Gia tri cua x la:’

Print @x

***Ví dụ 2: Cho hai biến x = 10 và y = 12, tính tổng z = x + y***

Declare @x int, @y int, @z int

Set @x = 10

Set @y = 12

Set @z = @x + @y

Select @z

***Ví dụ 3: Tính Z = YX***

Declare @X int, @Y int, @Z int

Set @X = 5

Set @Y = 2

Set @Z = Power(@Y, @X)

Print @Z

***Ví dụ 4: Gán giá trị từ một câu truy vấn:***

***(AVG là hàm tính trung bình)***

Declare @X int, @Y int

Set @X = 10

Select @Y = (Select AVG (DIEM) From DIEM)

Select @Y

***Ví dụ 5: Khai báo một biến var 1, gán giá trị cho biến là “Thanh” và đưa ra giá trị của var 1***

* **Cách 1:**

Declare @var 1 nvarchar(30)

Select @var 1 = ‘Thanh’

Select @var 1 = HoTen

From SV

Where MaSV = ‘SV2’

* **Cách 2:**

Declare @var 1 nvarchar(30)

Select @var 1 = ‘Thanh’

Set @var 1 =

(Select HoTen From SV

Where MaSV = ‘SV2’ )

***Ví dụ 6:***

Declare @ a nchar(10), @b nvarchar(30)

Select @ a = MaSV, @ b = HoTen

From SV ***=> Lệnh truy vấn***

Where MaSV = ‘SV01’

Select @a As ‘MaSinhVien’, @b As ‘HoTen’ ***=> Đưa kết quả của lệnh truy vấn ra màn hình***

* **Chú ý:**
* Với câu lệnh gán Set: Nếu kết quả trả về của câu truy vấn có nhiều hàng thì nó không trả về kết quả.
* Với câu lệnh gán: Nếu kết quả trả về của câu truy vấn có nhiều hàng, thì nó sẽ trả về một giá trị cuối cùng được lấy về.

1. **Biến toàn cục**

* Biến toàn cục là các biến do SQL Sever cung cấp.
* Tên biến bắt đầu bằng kí tự “@@”.
* Một số các toán tử:
* Toán tử đại số: Cộng (+), trừ (-); nhân (\*); chia lấy phần nguyên (/); chia lấy phần dư (%).
* Toán tử so sánh: >; <; >=; <=; !=
* Toán tử lo-gic: Kiểm tra điều kiện của hai biểu thức (Trả về giá trị True hoặc False). Một số toán tử lo-gic:

|  |  |
| --- | --- |
| ***Tên toán tử*** | ***Ví dụ*** |
| All | 5 > All (Select \* From Sales) |
| And | 5 > 7 and 6 > 15 |
| Or | 5 > 2 or 10 > 3 |
| Any | 5 > Any (Select Qty From Sales) |
| Between… And | 5 between and 10 |
| Exists | Exists(Select \* From Test) |
| In | 5 in (1, 2, 4, 5, 8) |

1. **Cấu trúc lệnh**
2. **Cấu trúc lệnh rẽ nhánh If – else**

*If <expression>*

*Begin*

*SQL – Statoments*

*End*

*[ else*

*Begin*

*SQL – Statoments*

*End ]*

***Ví dụ: Cho X = 5; Y = 2. Tính tổng Z = X + Y và so sánh giá trị X và Y?***

Declare @X int, @Y int, @Z int

Set @X = 5

Set @Y= 2

Set @Z = @X + @Y

If @Z > 0

Print STR(@X) + ‘LON HON’ + STR(@Y)

Else

Print STR(@Y) + ‘LON HON’ + STR(@X)

**Ví dụ: Viết hàm xét thành tích theo điểm trung bình. Nếu điểm trung bình lớn hơn 7 thì khá, còn lại thì là điểm thấp**

Declare @X int

Set @X = (Select AVG (DIEM) From (DIEM)

If @X > 7

Print N ‘Điểm từ khá trở lên’

Else

Print N ‘Điểm hơi thấp’

**Ví dụ: Viết hàm kiểm tra xem năm 2016 có phải là năm nhuận không (Nam nhuận chia hết cho 4 và không chia hết cho 100)?**

Declare @Nam int

Set @Nam = 2016

If (@Nam % 4 = 0) and (@Nam % 100 != 0)

Print N ‘Day la nam nhuan.’

Else

Print N ‘Day khong phai nam nhuan.’

**Ví dụ: Cho bảng SinhVien (MaSV, HoTen, MaLop)**

**Viết lệnh để thêm dữ liệu mới (SV1, Hồ Thị Lan, CT11) vào bảng. Biết rằng mỗi lớp chỉ tối đa 50 sinh viên?**

* **Phân tích đề:**
* **Cho biết lớp CT11 có bao nhiêu sinh viên:**

Select Count (\*) = @SiSo

From SinhVien

Where MaLop = ‘CT11’

* **So sánh số lượng sinh viên với 50:**

If (@SiSo < 50)

Insert Into SinhVien (MaSV, HoTen, MaLop)

Values (‘SV1’, ‘Ho Thi Lan’, ‘CT11’)

Else

Print N ‘Lop da du sinh vien’

* **Đáp án đầy đủ:**

Select Count (\*) = @SiSo

From SinhVien

Where MaLop = ‘CT11’

If (@SiSo < 50)

Insert Into SinhVien (MaSV, HoTen, MaLop)

Values (‘SV1’, ‘Ho Thi Lan’, ‘CT11’)

Else

Print N ‘Lop da du sinh vien’

1. **Cấu trúc Case:**
2. **Simple Case:**

* Một biểu thức sẽ được dùng để so sánh với một tập các giá trị để xác định kết quả.
* Cú pháp:

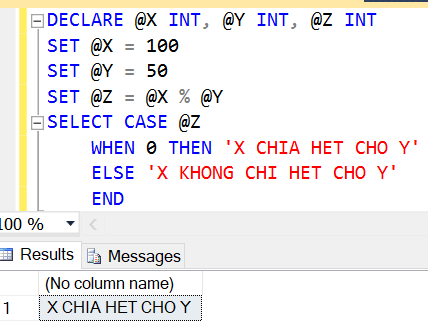
|  |
| --- |
| Case <Biểu thức>  When <Gia\_Tri1> Then <Ket\_Qua\_1>  When <Gia\_Tri2> Then <Ket\_Qua\_2>  [Else <Ket\_Qua\_Khac>  End |

1. **Searched Case:**

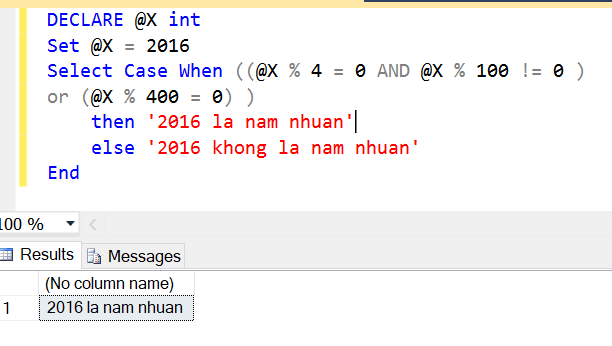
* Một biểu thức sẽ được dùng để so sánh với một tập các giá trị để xác định kết quả.

|  |
| --- |
| Case  When <Điều kiện 1> then <Kết quả 1>  [When<ĐIều kiện 2> then <Kết quả 2>  [...n]  [Else <Kết quả khác>]  End |

* **VÍ DỤ 1: Cho X = 100; Y = 50, Z = X/Y. Kiểm tra xem X chia hết cho Y không?**



* **VÍ DỤ 2: Kiểm tra xem năm 2016 phải năm nhuận không?**

****

* **VÍ DỤ 3:** Cho lược đố sau: NHANVIEN(MANV, HOTEN, GTINH, NAMSINH).

Liệt kê danh sách các nhân viên đến tuổi về hưu. Biết tuổi về hưu của nam là 60 tuổi, của nữ là 55 tuổi?

**LG:**

Select \* From NHANVIEN

WHERE (2021 - NAMSINH) >= Case GTINH

WHEN 'NAM' Then 60

WHEN 'NU' Then 55

End

* **VÍ DỤ 4:**

Cho lược đồ sau: DIEM(MASV, MAMON, DIEM)

Lập danh sách bảng điểm và xếp loại tương ứng. Biết rằng: Xếp loại trung bình nếu điểm nhỏ hơn 7, xếp loại khá nếu điểm lớn hơn 7 và nhỏ hơn 8, xếp loại giỏi nếu điểm lớn hơn hoặc bằng 8?

**LG:**

Select MASV, MAMON, DIEM

CASE

WHEN (DIEM < 7) THEN 'Trung binh'

WHEN (DIEM >=7 AND DIEM < 8) THEN 'Kha'

WHEN (DIEM >=8) THEN 'Gioi'

END

FROM DIEM

1. **CẤU TRÚC WHILE**

* **Cú pháp:**

While <Điều kiện>

<Lệnh/ Khối lệnh>

[Break]

[Continue]

* Có thể thêm Break và Continue trong khối lệnh của while.
* Break: Thoát khỏi vòng lặp While hiện hành.
* Continue: Trở lại đầu vòng While, bỏ qua các câu lệnh sau đó.

**VD1:** Cho lược đồ sau: SINHVIEN(MaSV int, HoTen nvarchar(30)).

Viết chương trình chèn thêm một mã sinh viên mới và bảng SINHVIEN theo quy định:

MaSV phải tăng dần, nếu có chỗ trống thì mã mới xác định sẽ chèn vào chỗ trống đó.

**LG:**

Declare @MaSV int

Set @MaSV = 1

While Exits (Select \*From SINHVIEN Where MaSV = @MaSV)

Set @MaSV = @MaSV + 1

Insert Into SINHVIEN(MaSV) Values (@MaSV)

**VD2:** Chèn thêm vào bảng SINHVIEN các mã sinh viên từ 49 tới 99

**LG:**

Declare @MaSV int

Set @MaSV = 49

While (@MaSV >= 49 and @MaSV <= 99)

Begin

While

Exits (Select \*From SINHVIEN Where MaSV = @MaSV)

Set @MaSV = @MaSV + 1

Insert Into SINHVIEN(MaSV) Values (@MaSV).

End

**VD3:** Xóa mã sinh viên bảng SINHVIEN các mã sinh viên từ 49 tới 99

**LG:**

Declare @MaSV int

Set @MaSV = 49

While (@MaSV >= 49 and @MaSV <= 99)

Begin

While

Exits (Select \*From SINHVIEN Where MaSV = @MaSV)

Set @MaSV = @MaSV + 1

Delete From SINHVIEN

Where ((@MaSV >= 49 and @MaSV <= 99)

End

## **THỦ TỤC VÀ HÀM (PROCEDURE AND FUNCTION)**

1. **Thủ tục (Procedure)**
2. **Khái niệm:**

* Tập các phát biểu T – SQL và SQL Sever biên dịch thành một kế hoạch thực thi đơn.
* T – SQL Store Procedure tương tự các ngôn ngữ lập trình khác:
* Chấp nhận biến đầu vào và trả lại kết quả khi thực hiện.
* Chứa những câu lệnh lập trình có thể thao tác được với CSDL và có thể gọi đến các thủ tục khác.
* Trả lại giá trị trạng thái khi thủ tục được gọi để xác định việc thực hiện thủ tục thành công hay thất bại.
* **Lệnh tạo một Thủ tục – Procedure:**

Create Proc <Tên thủ tục> [Danh sách các tham số]

[With Ecomple | Encryption | Recompile, Encryption]

As

[Declare <Biến cục bộ>] <Các câu lệnh của thủ tục>

* **Lệnh thực thi thủ tục - Procedure:**

<Tên thủ tục> [<Danh sách các đối số>]

* **Nếu gọi thủ tục bên trong một thủ tục khác:**

Exce <Tên thủ tục> [<Danh sách các đối số>]

* **Ví dụ: Cho Cơ sở dữ liệu sau:**
* SV(MaSV, HoDem, Ten, NgaySinh, GTinh, NoiSinh, MaLop)
* Lop (MaLop, TenLop, Khoa, HeDaoTao, NamNhapHoc, SiSo, MaKhoa)
* Khoa (MaKhoa, TenKhoa, DienThoai)
* MonHoc (MaMonHoc, TenMonHoc, SoDVHT)
* DiemThi (MaSV, MaMonHoc, DiemLan1, DiemLan2)
* **VD1: Thủ tục không tham số:** Tạo thủ tục liệt kê danh sách các sinh viên:

Create Procedure DSSV

AS Select \* From SV

* ***Thực thi thủ tục:*** Exce DSSV
* **VD2: Thủ tục có một tham số:** Tạo thủ tục lập danh sách sinh viên của một lớp nào đó?

Create Proc DSSV\_Lop (@MaLop nvarchar(5))

As Select \* From SV Where MaLop = @MaLop

* **Thực thi thủ tục:** Exce DSSV\_Lop ‘L01’
* **VD3:** Thủ tục nhiều tham số: Tạo thủ tục để nhập dữ liệu cho bảng MonHoc?

Create Proc Nhap\_MonHoc (@MaMonHoc nchar(10), @TenMonHoc nvarchar(30), @SoDVHT int)

As

Insert Into MonHoc Values (@MaMonHoc, @TenMonHoc, @SoDVHT)

* Thực thi thủ tục: Exec Nhap\_MonHoc ‘MH1’, ‘Toan’, 3
* **VD4:** Thủ tục nhiều tham số: Tạo thủ tục kiểm tra xem hai môn học có cùng số Đơn vị học trình (Số tín chỉ) hay không?

Create Proc SoTC (@MaMon1 int, @MaMon2 int)

As

Declare @SoTC1 int, @SoTC2 int

Select @SoTC1 = SoTC From MonHoc Where MaMon = @MaMon1

Select @SoTC2 = SoTC From MonHoc Where MaMon = @MaMon2

If (@SoTC1 = @SoTC2)

Print ‘Hai mon hoc co cung so tin chi.’

Else

Print ‘Hai mon hoc khac so tin chi.’

* **VD5:** Kiểm tra xem hai lớp có nhập học cùng năm không?

Create Proc KiemTra (@MaLop1 nchar(10), @MaLop2 nchar(10))

As

Declare @Nam1 BigInt, @Nam2 BigInt

Select @Nam1 = NamNhapHoc From Lop Where MaLop = @MaLop1

Select @Nam2 = NamNhapHoc From Lop Where MaLop = @MaLop2

If (@Nam1 = @Nam2) print ‘Nhap hoc cung nam.’

Else print ‘Nhap hoc khac nam.’

* Giá trị trả về của tham số trong thủ tục:

@<Tên tham số><Kiểu dữ liệu> Output

* **VD1: Tạo thủ tục tính tổng hai số a và b?**

Create Proc SP\_Cong (@a int, @b int, @c int output)

As Select @c = @a + @b

* ***Thực thi thủ tục:***

Declare @Tong int

Select @Tong = 0

Execute SP\_Cong 100, 200 @Tong output

Select @Tong

* **VD2: Tạo thủ tục trả về điểm trung bình của một sinh viên nào đó?**

Alter Proc DiemTB (@MaSV int, @Diem int output)

As

Select @Diem = AVG (Diem) From Diem Where MaSV = @MaSV

* **Thực thi thủ tục:**

Declare @DiemTb int

Exce DiemTB1, @DiemTB output Select DiemTB

1. **Tham số có giá trị mặc định:**

* **Khai báo:**

**@<Tên tham số><Kiểu dữ liệu> = <Giá trị mặc định>**

* **VD1: Tìm kiếm sinh viên nào có nơi sinh đã được gán mặc định là “Huế”. Nếu không có thì in ra Nơi sinh và tên lớp học?**
* **Tạo thủ tục:**

Create Proc SP\_Test(@TenLop nvarchar(30) = NULL, @NoiSinh nvarchar(100) = ‘Hue’)

As

Begin

If @TenLop is NULL

Select HoDem, Ten From SV Inner Join Lop

On SV.MaLop = Lop.MaLop

Where NoiSinh = @NoiSinh

Else

Select HoDem, Ten From SV Inner Join Lop

On SV.MaLop = Lop.MaLop

Where NoiSinh = @NoiSinh and TenLop = @TenLop

* **Thực thi thủ tục:**

1. **Cho biết họ và tên của các sinh viên sinh tại Huế?**

* SP\_Test

1. **Cho biết họ tên của các sinh viên lớp Tin3A sinh tại Huế?**

* SP\_Test @TenLop = ‘Tin3A’

1. **Cho biết họ tên của các sinh viên sinh tại Hà Nội?**

* SP\_Test @NoiSinh = ‘Ha Noi’

1. **Cho biết họ tên của các sinh viên lớp Tin3A sinh tại Đà Nẵng?**

* SP\_Test @TenLop = ‘Tin3A’, @NoiSinh = ‘Da Nang’
* **VD2:** Tạo thủ tục liệt kê danh sách sinh viên có học một môn học nào đó?
* **Tạo thủ tục:**

Create Proc DSSV\_DKHOC (@MaMon int = NULL)

As

If (@MaMon is NULL)

Select Distinct MaSV From Diem

Else

Select MaSV From Diem Where MaMon = @MaMon

* **Thực thi thủ tục:**

Exec DSSV\_DKHOC

Exec DSSV\_DKHOC @MaMon = 1

1. **Thay đổi thủ tục:**

* Sử dụng lệnh Alter Procedure.
* Sự thay đổi vẫn giữ nguyên lại các quyền người dùng.

Alter Procedure<Tên thủ tục> [(Danh sách các tham số)]

[With Recompile | Encryption | Recompile, Encryption]

As <Các câu lệnh của thủ tục>

Drop Procedure <Tên thủ tục>

* Xóa thủ tục:
* Xóa các nội dung của thủ tục: Sử dụng hệ thống sp\_helptext

Execute sp\_helptext sp\_DSSV

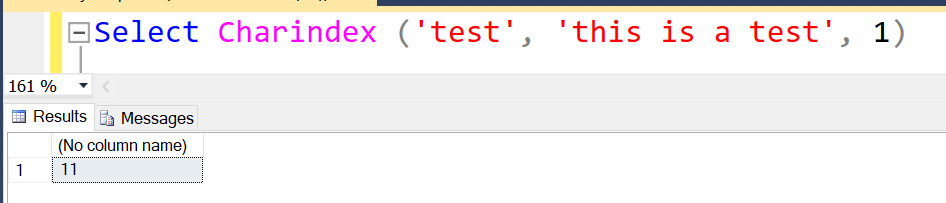
1. **Hàm (Function)**
2. **Định nghĩa:**

* Hàm là đối tượng cơ sở dữ liệu tương tự như thủ tục.
* Điểm khác biệt giữa hàm và thủ tục: Hàm trả về giá trị thông qua tên hàm, còn thủ tục thì không.
* Gồm: Hàm do hệ thống tự định nghĩa và Hàm do người dùng định nghĩa.

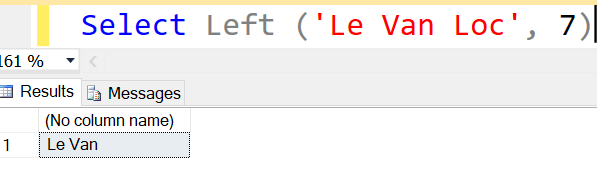
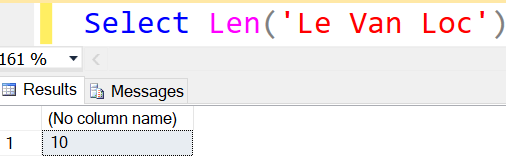
1. **Hàm so hệ thống định nghĩa**
2. **Hàm chuỗi kí tự: Tìm vị trí bắt đầu của chuỗi kí tự chỉ định.**

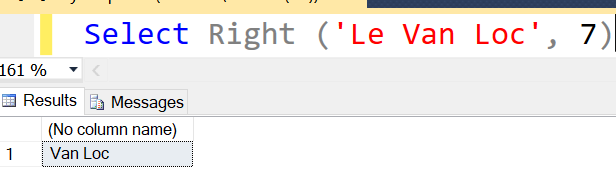
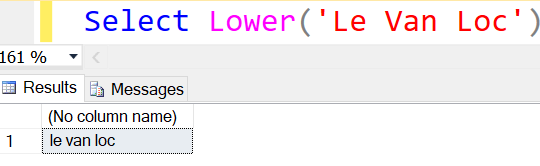
**Select Charindex (string 1, string 2, start\_prostion)**

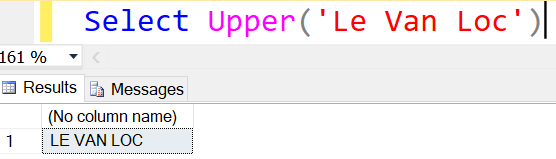
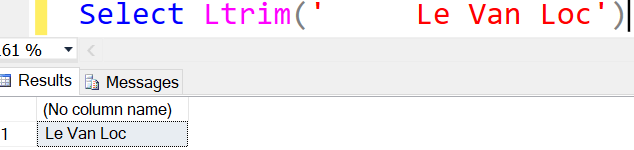
* **VD: Select Charindex (‘test’, ‘this is a test’, 1)**
* Tìm vị chí chữ t trong từ “test”

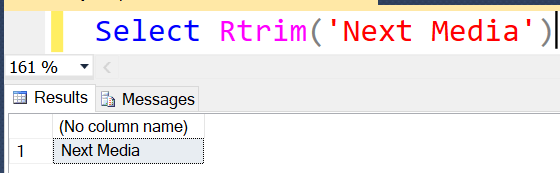
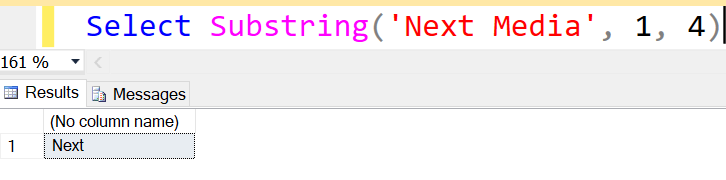
****

* **Một vài hàm chuỗi khác:**
* Select Left (string, number of charactess): Trả về chuỗi kí tự tính trừ trái.
* Select Right (string, number of charactess): Trả về chuỗi kí tự tính trừ phải.
* Select Len(string): Trả về độ dại của chuỗi kí tự nhập vào.
* Select Lower (string): Trả về chuỗi kí tự viết thường.
* Select Upper(string): Trả về chuỗi kí tự viết hoa.
* Select Ltrim(string): Cắt bỏ các kí tự bên trái.
* Select Rstrim(string): Cắt bỏ các kí tự bên trái.
* Select Substring(expression, start, length): Trả về chuỗi con bao gồm length kí tự của dãy tính từ “start”.
* **Ví dụ:**

** **

**** 

****  ****

****  ****

1. **Hàm Toán học:**

* Abs(number): Giá trị tuyệt đối.
* Ceiling(number): Số nguyên nhỏ nhất >= n
* Floor(number): Số nguyên lớn nhất <= n
* Round(number, precision): Làm tròn số, lấy precision chữ số thập phân sau dấu phảy.
* Sqare(number): Trả về giá trị n2
* Sqrt(number): Trả về giá trị căn bậc hai của số

1. **Hàm thống kê**

* AVG ([All | Distinct] expression).
* Count: Trả về kiểu int số các phần tử của một nhóm.
* Max, Min: Trả về giá trị lớn nhât, nhỏ nhất.
* Sum: Tính tổng.
* Count\_Big: Trả về số các phần tử trong một nhóm.

1. **Hàm xử lí thời gian**



1. **Hàm do người dùng tự định nghĩa:**
2. **Hàm trả về là kiểu dữ liệu cơ sở:**

Create Function <Tên hàm ([Danh sách các tham số])>

Returns <Kiểu dữ liệu của hàm>

As

Begin

<Tập khối câu Lệnh SQL Sever>

End

**VD1:**

Create Function Tong (@a int, @b int)

Returns int As

Begin

Declare @c int

Set @c = @a + @b

Return @c

End

* Gọi hàm: Tính tổng hai số 1 và 2: Select dbo.Tong(1,2)

**VD2:** Cho lược đồ: MonHoc (MaMon, TenMon, SoTC). Xây dựng hàm cho biết số tín chỉ của một môn học bất kì?

Create Function TraVeSoTC (@MaMon nchar(10))

Returns int

Begin

Returns (Select SoTC From Mon Where MaMon = @MaMon)

End

* **Gọi hàm: Cho biết môn học MH1 có bao nhiêu tín?**

Select dbo.TraVeSoTC(‘MH1’)

1. **Kiểu trả về là bảng dữ liệu có được từ câu truy vấn:**

Create Function <Tên hàm> ([ Danh sách tham số ])

Returns Table

As

Return [(] Câu lệnh Select [)]

**VD1: Cho bảng: SV (MaSV, HoTen, GTinh, NgaySinh, MaLop)**

1. **Viết hàm cho biết mã lớp học của một sinh viên bất kì nào đó?**
2. **Viết hàm cho biết mỗi lớp có bao nhiêu sinh viên?**

**LG:**

Create Function Lop\_Hoc (@MaSV nchar(10))

Returns Table As

Return

(Select MaLop From SV

Where MaSV = @MaSV)

* **Gọi hàm:**

Select \* From dbo.Lop\_Hoc (‘SV01’)

Create Function TongSoSV ()

Returns Table As

Return

(Select MaLop, Count(\*) As SoSV

From SinhVien

Group By MaLop)

1. **Kiểu trả về của hàm là bảng dữ liệu có được sau một chuỗi các câu lệnh**

Create Function <Tên hàm> ([Danh sách các tham số])

Returns <@Biến bảng> Table <Định nghĩa bảng>

As

Begin

<Các câu lệnh của hàm>

Return

End

**VD: Cho cơ sở dữ liệu sau:**

* LOP (MaLop, TenLop, NamNhapHoc)
* SV (MaSV, HoTen, GTinh, DiaChi, MaLop)
* MON (MaMon, TenMon, SoTC)
* Diem (MaSV, MaMon, Diem)

1. **Viết hàm trả về bảng điểm của một sinh viên nào đó?**
2. **Viết hàm trả về bảng điểm của một lớp nào đó?**

**LG:**

1. Create Function BANG\_DIEM (@MaSV int)

Return @BangDiem Table (MaMon int, Diem int)

As

Begin

Insert Into @BangDiem

Select MaMon, Diem From Diem

Where MaSV = @MaSV Return

End

* Gọi hàm: Lập bảng điểm của SV1:

Select \* From dbo.BANG\_DIEM1



Create Function BangDiemLop (Lop char(5))

Returns @BangDiem Table (MaSV int, MaMon int, Diem int)

As

Begin

Insert Into @BangDiem

Select SinhVien.MaSV, Diem.MaMon, Diem.Diem

From SinhVien, Diem

Where (SinhVien.MaSV = Diem.MaSV) And

(SinhVien.MaLop = @Lop)

Return

End

* **Gọi hàm: Lập bảng điểm của lớp L01:**

Select \* From dbo. BangDiemLop (‘L01’)

**VD:**

1. **Tạo hàm cho biết một khoa có bao nhiêu lớp?**
2. **Tạo hàm cho biết một tháng có bao nhiêu ngày?**
3. **Tạo hàm xác định thứ trong tuần của một ngày?**

**LG:**



Create Function SoLopMotKhoa ()

Returns Table As

Return

(Select MaKhoa, Count (\*) As

SoLop

From Khoa

Group By MaKhoa)

GO

SELECT \* FROM TongSo()