SQL: Structured Query Language < Ngôn ngữ truy vấn có cấu trúc>

1.Tạo Database:

Create database <name file>;

2. Comment :

-- : chú thích

3. Tạo bảng:

Create table <name file>;

Ex: Create table Student

(

ID nvarchar(10),

NAME nvarchar(100)

)

4. Tạo Database:

Ex:

Create database SQLQUERY

Go

Use SQLQUERY

Go

Create table Student

(

…….

)

Go

5. Xóa bảng trong Database:

+ Truncate table Student -- xóa dữ liệu trong bảng

+ Drop table Student -- xóa toàn bộ bảng

6. Sửa bảng

Alter table Student add Ngaysinh Date

7. Kiểu dữ liệu

A white and black text on a white background

Description automatically generated

Note: Unicode là viết tiếng việt được: Ex: nchar, nvarchar ntext, Else: char, varchar, text

A screenshot of a computer

Description automatically generated

8. Insert, delete, update table

a.Chèn dữ liệu

-- thêm dữ liệu

-- kiểu số -> số bình thường

-- kiểu kí tự or ngày -> để trong ngoặc nháy đơn

-- nếu là unicode thì cần có n phía trước cặp nháy đơn N''

INSERT INTO <Tên Table>

        ( column**1**, column**2**, column**3**, … , column**n**, )

VALUES  (

Gợi ký nhập dữ liệu, -- Tên column**1** – kiểu dữ liệu tương ứng column**1**

Gợi ký nhập dữ liệu, -- Tên column**2** – kiểu dữ liệu tương ứng column**2**

Gợi ký nhập dữ liệu, -- Tên column**3** – kiểu dữ liệu tương ứng column**3**

 …

          )

Ex:

insert dbo.Test

values (

10, -- maso int

N'Trong', -- ten nvarchar

'20040603', -- ngaysinh date yyyy/mm/dd

1, -- nam bit true

N'NA', -- diachi char

10000 -- tienluong float

)

b.Xóa dữ liệu

1.Truncate table < tên table>

2.delete < tên table>

Có điều kiện: delete dbo.<tên table> where <tên cột> <toán tử> <điều kiện>

A screenshot of a document

Description automatically generated

A white rectangular box with black text

Description automatically generated

A white rectangular box with black text

Description automatically generated

A screenshot of a chat

Description automatically generated

c.cập nhật dữ liệu

Khi cập nhật có thể thay đổi toàn bộ dữ liệu

update table dbo.<tên table> set <tên cột> <thay đổi>

Ex: update dbo.Test set TienLuong = 1, DiaChi ='HN'd

9 Khóa chính:

A white sheet with black text

Description automatically generated

* C1:
* create table Testprimarykey1
* (
* ID int unique not null,
* Name nvarchar(100) default N'Vuongductrong'
* )
* go
* -- khi đã tạo bảng, muốn sửa cột thành primary key
* alter table dbo.Testprimarykey1 add primary key (ID)
* C2:
* -- tạo primary key trong bảng không phải ngay khi khai báo
* create table TestPrimaryKey2
* (
* ID int not null,
* Name Nvarchar(100) default N'VuongDucTrong'
* primary key (ID)
* )
* Go
* C3:
* -- tạo primary key trong bảng mà đặt tên cho key đó
* create table TestPrimaryKey3
* (
* ID int not null,
* Name Nvarchar(100) default N'VuongDucTrong'
* constraint pk\_test3 -- sau này xóa key cho dễ
* primary key (ID)
* )
* Go
* C4:
* -- tạo primary key sau khi tạo bảng và đặt tên cho key đó
* create table TestPrimaryKey4
* (
* ID int not null,
* Name Nvarchar(100) N'VuongDucTrong'
* )
* go
* alter table dbo.TestPrimaryKey4
* add constraint pk\_Test4
* primary key (ID)
* C5
* -- khóa chính có 2 trường
* create table TestPrimaryKey5
* (
* ID1 int not null,
* ID2 int not null,
* Name Nvarchar(100) default N'VuongDucTrong'
* primary key (ID1, ID2)
* )
* Go

10. Khóa ngoại

-- điều kiện để tạo khóa ngoại

-- tham chiếu tới khóa chính

-- Unique, not null

-- cùng kiểu dữ liệu

-- cùng số lượng trường có sắp xếp

-- lợi ích:

-- đảm bảo toàn vẹn dữ liệu

-- không có trường hợp tham chiếu tới dữ liệu không tồn tại

A screenshot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

EX:

create database Primary\_Foreign

go

use Primary\_Foreign

go

create table BoMon

(

MaBM char(10) primary key,

Name Nvarchar(100) default N'Tên bộ môn'

)

go

create table Lop

(

MaLop char(10) primary key,

Name nvarchar(100) default N'Tên lớp'

)

go

create table GiaoVien

(

MaGV char(10) not null,

Name Nvarchar(100) default N'Tên giáo viên',

DiaChi Nvarchar(100) default N'Địa chỉ giáo viên',

NgaySinh Date,

Sex bit,

MaBM char(10),

-- tạo khóa ngoại ngay khi tạo bảng

foreign key(MaBM) references dbo.BoMon(MaBM)

)

go

alter table dbo.GiaoVien add Primary key (MaGV)

create table HocSinh

(

MaHS char(10) primary key,

Name Nvarchar(100),

MaLop char(10)

)

go

-- tạo khóa ngoại sau khi tạo bảng

alter table dbo.HocSinh add foreign key(MaLop) references dbo.Lop(MaLop)

-- hủy khóa

-- alter table dbo.HocSinh drop constraint FK\_HS

+ Khóa chính phục vụ khóa ngoại, khóa ngoại tham chiếu thông qua khóa chính

+ Tại sao cần khóa chính và khóa ngoại?

Để đảm bảo toàn vẹn dữ liệu của cơ sở dữ liệu

Sự liên kết giữa khóa chính và khóa ngoại đó là quan hệ

10. Truy vấn dữ liệu

-- cấu trúc truy vấn

select \* from dbo.BOMON -- lấy hết các dữ liệu trong bảng bộ môn

-- lấy mã đề tài + tên đề tài trong bảng bộ môn

select MABM, TENBM from dbo.BOMON

-- đổi tên cột hiển thị

select MABM as 'HowKteam.com', TENBM as N'Giáo dục' from dbo.BOMON

-- xuất ra mã giáo viên + tên + tên bộ môn giáo viên đó dạy

C1: select MAGV, HOTEN, TENBM from dbo.GIAOVIEN, dbo.BOMON

C2: select GV.MAGV, GV.HOTEN, BM.TENBM from dbo.GIAOVIEN as GV, dbo.BOMON as BM

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Ex:

-- lấy ra giáo viên lương > 2000

select \* from dbo.GIAOVIEN

where LUONG > 2000

-- lấy ra giáo viên là nữ và lương > 2000

select \* from dbo.GIAOVIEN as GV

where GV.PHAI = N'Nữ' and GV.LUONG > 2000

-- lấy ra giáo viên nhỏ hơn 40 tuổi

-- year -> lấy ra năm của ngày

-- getdate -> lấy ngày hiện tại

select \* from dbo.GIAOVIEN as GV

where year(getdate()) - year(GV.NGSINH) > 40

-- lấy ra họ tên giáo viên, năm sinh và tuổi của giáo viên ngỏ hơn 40 tuổi

select GV.HOTEN, GV.NGSINH, year(getdate()) - year(GV.NGSINH) from dbo.GIAOVIEN as GV

where year(getdate()) - year(GV.NGSINH) > 40

-- lấy ra mã Gv, tên Gv và tên khoa của giáo viên đó làm việc

select gv.MAGV, gv.HOTEN, k.TENKHOA from dbo.GIAOVIEN as gv, dbo.BOMON as bm, dbo.KHOA as k

where bm.MAKHOA = k.MAKHOA and gv.MABM = bm.MABM

select \* from dbo.BOMON

-- lấy ra giáo viên là trưởng bộ môn

select gv.\* from dbo.GIAOVIEN as gv, dbo.BOMON as bm

where bm.TRUONGBM = gv.MAGV

-- count(\*) -> đếm số lượng của tất cả các record

-- count(tên trường) -> đếm số lượng tên trường đó

select count(\*) as N'Số Lượng giáo viên' from dbo.GIAOVIEN

-- đếm số lượng người thân của giáo viên có mã gv là 007

select count(nt.TEN) as N'Số Lượng người thân' from dbo.GIAOVIEN as gv, dbo.NGUOITHAN as nt

where gv.MAGV = 007 and nt.MAGV = gv.MAGV

-- lấy ra tên giáo viên và tên đề tài người đó tham gia

select gv.HOTEN, dt.TENDT from dbo.GIAOVIEN as gv, dbo.THAMGIADT as tg, dbo.DETAI as dt

where gv.MAGV = tg.MAGV and tg.MADT = dt.MADT

-- lấy ra tên giáo viên và tên đề tài người đó tham gia khi mà người đó tham gia nhiều hơn 1 lần

-- truy vấn lồng

-- bài tập

-- 1. xuất ra thông tin giáo viên và giáo viên quản lý chủ nhiệm của người đó

-- C1

select \* from dbo.GIAOVIEN

where GVQLCM != MAGV

-- C2

select gv1.HOTEN, gv2.HOTEN from dbo.GIAOVIEN as gv1, dbo.GIAOVIEN as gv2

where gv1.GVQLCM = gv2.MAGV

-- 2. xuất ra số lượng giáo viên của khoa CNTT

select count(\*) from dbo.GIAOVIEN as gv, dbo.BOMON as bm, dbo.KHOA as k

where gv.MABM = bm.MABM and k.MAKHOA = bm.MAKHOA and k.MAKHOA = 'CNTT'

-- 3. xuất ra thông tin giáo viên và đề tài người đó tham gia khi mà kết quả là đạt

select \* from dbo.GIAOVIEN as gv, dbo.THAMGIADT as tg

where tg.KETQUA = N'Đạt' and gv.MAGV = tg.MAGV

11. Tìm kiếm gần đúng

A screenshot of a computer

Description automatically generated

Ex:

-- xuất ra thông tin giáo viên mà tên bắt đầu bằng chữ l

select \* from dbo.GIAOVIEN

where HOTEN like 'l%'

-- xuất ra thông in giáo viên kết thúc bằng chữ n

select \* from dbo.GIAOVIEN

where HOTEN like '%n'

-- xuất ra thông tin giáo viên có tồn tại chữ n

select \* from dbo.GIAOVIEN

where HOTEN like '%n%'

-- xuất ra thông tin giáo viên có tồn tại chữ ế

select \* from dbo.GIAOVIEN

where HOTEN like N'%ế%'

12. Inner join

A white background with black text

Description automatically generated

A diagram of a table and table

Description automatically generated

Ex:

-- inner join -> kiểu cũ -> có thể sau này không còn hỗ trợ

-- mọi join đều cần điều kiện

select \* from dbo.GIAOVIEN, dbo.BOMON

where BOMON.MABM = GIAOVIEN.MABM

-- inner join kiểu mới

select \* from dbo.GIAOVIEN inner join dbo.BOMON

on BOMON.MABM = GIAOVIEN.MABM and BOMON.TRUONGBM = ''

-- có thể viết tắt inner join -> join

select \* from dbo.KHOA join BOMON

on BOMON.MAKHOA = KHOA.MAKHOA

13. Outer join

A white text with red text

Description automatically generated

A blue circles with black text

Description automatically generated

Ex:

-- gom 2 bảng lại theo điều kiện bên nào không có dữ liệu thì để null

select \* from dbo.GIAOVIEN full outer join dbo.BOMON

on BOMON.MABM = GIAOVIEN.MABM

-- Cros join là tổ hợp mỗi record của bảng A với all record của bảng B

select \* from dbo.GIAOVIEN cross join dbo.BOMON

A screenshot of a white background with black text

Description automatically generated

Ex:

--left join

-- bảng bên phải nhập vào bảng bên trái

-- record nào bảng phải không có thì để null

-- record nào bảng bên trái không có thì không điền vào

select \* from dbo.GIAOVIEN left join dbo.BOMON

on BOMON.MABM = GIAOVIEN.MABM

A diagram of a table

Description automatically generated

-- right join

select \* from dbo.GIAOVIEN right join dbo.BOMON

on BOMON.MABM = GIAOVIEN.MABM

14. Union

Chức năng: - Kết hợp các tập kết quả

- Loại bỏ các bản ghi giống nhau

- Số lượng và các kiểu dữ liệu của các cột phải khớp

- Thứ tự các cột giống nhau

Ex:

select MAGV from dbo.GIAOVIEN

where LUONG >= 2500

union

select MAGV from dbo.NGUOITHAN

where PHAI = N'Nữ'

A screenshot of a computer

Description automatically generated

Union all: giữ lại các dòng, kể cả dòng trùng lặp

Ex:

select MAGV from dbo.GIAOVIEN

where LUONG >= 2500

union all

select MAGV from dbo.NGUOITHAN

where PHAI = N'Nữ'

A screenshot of a white sheet with numbers

Description automatically generated

15. SELECT INTO

Tạo ra bảng mới:

Ex:

-- Lấy hết dữ liệu của bảng gv đưa vào bảng mới tên là TableA

-- Bảng này có các record tương ứng như bảng gv

select \* into TableA from dbo.GIAOVIEN

-- Tạo ra 1 bảng TableB mới. Có 1 cột dữ liệu là Hoten tương ứng như bảng gv

-- Dữ liệu của bảng TableB lấy từ gv ra

select Hoten into TableB from dbo.GIAOVIEN

-- Tạo ra 1 bảng TableB mới.Có 1 cột dữ liệu là Hoten tương ứng như bảng gv

-- với điều kiện lương > 2000

-- Dữ liệu của bảng TableB lấy từ gv ra

select HoTen into TableC from dbo.GIAOVIEN

where LUONG > 2000

-- Tạo 1 bảng mới từ nhiều bảng

select MAGV, HOTEN, TENBM into GVBackup from dbo.GIAOVIEN, dbo.BOMON

where BOMON.MABM = GIAOVIEN.MABM

-- Tạo ra 1 bảng GVBK y chang nhưng không có dữ liệu

select \* into GVBK from dbo.GIAOVIEN

where 6 > 9

16. INSERT INTO SELECT

Ex:

-- insert into select -> copy dữ liệu vào bảng đã tồn tại

select \* into CloneGV from dbo.GIAOVIEN

where 1 = 0

insert into CloneGV

select \* from dbo.GIAOVIEN

select \* from CloneGV

17. Truy vấn lồng

Ex:

-- Truy vấn lồng trong from

select \* from dbo.GIAOVIEN, (select \* from dbo.DETAI) as dt

-- 1. Xuất ra danh sách giáo viên tham gia nhiều hơn 1 đề tài

-- Lấy ra tất cả thông tin của giáo viên khi mà

select \* from dbo.GIAOVIEN as gv

-- Số lượng đề tài giáo viên đó tham gia > 1

where 1 <

(

select count(\*) from dbo.THAMGIADT

where MAGV = gv.MAGV

)

-- 2. Xuất ra thông tin của khoa mà có nhiều hơn 2 giáo viên

select \* from dbo.KHOA as k

where 2 <

(

select count(\*) from dbo.GIAOVIEN as gv, dbo.BOMON as bm

where gv.MABM = bm.MABM and bm.MAKHOA = k.MAKHOA

)

-- sắp xếp giảm dần descent

select \* from dbo.GIAOVIEN

order by MAGV desc

-- sắp xếp tăng dần ascend

select \* from dbo.GIAOVIEN

order by MAGV asc

-- lấy ra top 5 phần tử

select top(5) \* from dbo.GIAOVIEN

18. Group by -- Xuất ra danh sách tên bộ môn và số lượng giáo viên của bộ môn đó.

select TENBM, count(\*) from dbo.BOMON as bm, dbo.GIAOVIEN as gv

where bm.MABM = gv.MABM

group by bm.TENBM

-- cột hiển thị phải là thuộc tính nằm trong khối group by hoặc là agreegate function

/\*

Agreeate Function

avg() : trả về giá trị trung bình

count(): trả về số lượng dòng

first(): trả về giá trị đầu tiên

last() : trả về giá trị cuối cùng

max() : trả về giá trị lớn nhất

min() : trả về giá trị nhỏ nhất

round(): trả về giá trị đượ làm tròn

sum () : trả về tổng

\*/

19. Group by having

Ex:

-- Xuất ra số lượng giáo viên trong từng bộ môn mà số gv > 2

-- having -> như where của select nhưng dành cho group by

-- having là where của group by

select bm.MABM, count(\*) from dbo.GIAOVIEN as gv, dbo.BOMON as bm

where bm.MABM = gv.MABM

group by bm.MABM

having count(\*) > 1

20. Auto increament

Ex:

create table TestAuto

(

ID int primary key identity,

-- tự tăng trường này, phải là số.

-- Mặc định bắt đầu từ 1 và tăng 1 đơn vị

Name Nvarchar(100)

)

go

insert dbo.TestAuto(Name) values(N'')

……………………………………………………………………………………………..

create table TestAuto2

(

ID int primary key identity (5, 10),

-- Tự tăng. bắt đầu từ 5 môi lần tăng 10 đơn v

Name Nvarchar(100)

)

go

A screenshot of a computer

Description automatically generated

21. View

-- View là bảng ảo

-- Cập nhật dữ liệu theo bảng mà view truy vấn tới mỗi khi lấy view ra dùng

Ex:

-- Tạo ra view TestView từ câu truy vấn

create view TestView as

select \* from dbo.GIAOVIEN

-- xóa view

drop view TestView

-- update view

alter view TestView as

select HOTEN from dbo.GIAOVIEN

-- tạo view có dấu

create view [Giáo dục miễn phí] as

select \* from dbo.KHOA

select \* from [Giáo dục miễn phí]

22. Check

-- check: cho nó nằm trong phạm vi mình mong muốn

-- Có thể tạo check y như khóa chính

-- C1:

create table TestCheck

(

ID int primary key identity,

LUONG int check(LUONG > 3000 and LUONG < 9000)

)

Go

--

insert dbo.TestCheck (LUONG)

values (3001)

--

-- C2

create table TestCheck

(

ID int primary key identity,

LUONG int,

check(LUONG > 3000 and LUONG < 9000)

)

go

-- C3

create table TestCheck

(

ID int primary key identity,

LUONG int,

constraint CK\_LUONG check(LUONG > 3000 and LUONG < 9000)

)

go

-- C4: sửa bảng

alter table TestCheck add constraint CK\_LUONG

check(LUONG > 3000 and LUONG < 9000)

23.Index:

-- Tạo index trên bảng người thân

-- Tăng tốc độ tìm kiếm <> chậm tốc độ thêm, xóa, sửa

-- Cho phép các trường trùng nhau

create index IndexName on dbo.NGUOITHAN(MaGV)

-- Không cho phép các trường trùng nhau

create unique index IndexNameUnique on dbo.GIAOVIEN(MaGV)

select \* from dbo.GIAOVIEN

24. Kiểu dữ liệu tự định nghĩa:

use HowKteam

go

-- Kiểu dữ liệu tự định nghĩa

-- EXEC sp\_addtype 'Tên kiểu dữ liệu', 'Kiểu dữ liệu thực tế', 'Not null' (có hay không đều được)

EXEC sp\_addtype 'NName', 'nvarchar(100)', 'Not null'

create table TestType1

(

Name NNAME,

address nvarchar(500)

)

go

EXEC sp\_addtype 'NADDRESS', 'nvarchar(500)'

-- Xóa Type

EXEC sp\_droptype 'NADDRESS'

25.Declare

-- lưu vào 1 giá trị để đơn giản hóa truy vấn

-- Tìm ra mã giáo viên có lương cao nhất

select MAGV from dbo.GIAOVIEN

where LUONG = (select min(LUONG) from dbo.GIAOVIEN)

-- C1:

-- Lấy ra tuổi của giáo viên với mã GV 001

select year(getdate()) - year(NGSINH) from dbo.GIAOVIEN

where MAGV = (select MAGV from dbo.GIAOVIEN where LUONG = (select min(LUONG) from dbo.GIAOVIEN))

-- C2:

-- Tạo ra 1 biến kiểu char lưu mã giáo viên lương thấp nhất

declare @MinSalaryMAGV char(10)

select @MinSalaryMAGV = MAGV from dbo.GIAOVIEN where LUONG = (select min(LUONG) from dbo.GIAOVIEN)

select year(getdate()) - year(NGSINH) from dbo.GIAOVIEN

where MAGV = @MinSalaryMAGV

26. If else

-- Ví dụ 1: Kiểm tra xem lương của giáo viên nhập vào có > lương TB hay không

declare @LuongTB int

declare @SoLuongGV int

select @SoLuongGV = count(\*) from dbo.GIAOVIEN

select @LuongTB = sum(LUONG) / @SoLuongGV from dbo.GIAOVIEN

declare @MaGV char(10) = '006'

declare @LuongMaGV int = 0

select @LuongMaGV = LUONG from dbo.GIAOVIEN

where MAGV = @MAGV

-- Nếu lương của @MaGV > @LuongTB

-- Xuất ra lớn hơn

-- ngược lại

-- Xuât ra nhỏ hơn

if @LuongMaGV > @LuongTB

begin

print @LuongMaGV

print @LuongTB

print N'Lớn hơn'

end

else

begin

print @LuongMaGV

print @LuongTB

print N'Nhỏ hơn'

end

27. While

declare @i int = 0

declare @n int = 10000

-- While(điều kiện thực hiện) -> khối lệnh thực hiện

while (@i < @n)

begin

insert dbo.TestCheck (Luong)

values (@i -- Luong - int

)

set @i += 1

end

select \* from dbo.TestCheck

-- Insert 10000 record vào bảng BoMon

-- Mà không trùng ID

-- Tên bộ môn tăng dần

28. Cursor <con trỏ>

-- Khi có nhu cầu duyệt từng record của bảng. Với mỗi record có kết quả xử lý riêng thì dùng Cursor

-- declare <Tên con trỏ> cursor for <câu select>

--open <Tên con trỏ>

-- fetch next from <Tên con trỏ> into <Danh sách các biến tương ứng kết quả truy vấn>

-- while @@fetch\_status = 0

-- begin

-- Câu lệnh thực hiện

-- fetch next lại lần nữa

-- fetch next from <Tên con trỏ> into <Danh sách các biến tương ứng kết quả truy vấn>

-- end

-- close <Tên con trỏ>

-- deallocate <Tên con trỏ>

-------------------------------------------------------------------------

-- Từ tuổi của giáo viên

-- Nếu > 50 thì lương = 2500

-- Nếu > 40 & < 50 thì lương = 2000

-- Ngược lại lương = 1500

select \* from dbo.GIAOVIEN

-- Lấy ra danh sách MAGV kèm tuổi đưa vào con trỏ có tên là GiaoVienCursor

declare GiaoVienCursor cursor for select MAGV, year(getdate()) - year(NGSINH) from dbo.GIAOVIEN

open GiaoVienCursor

declare @MaGV char(10)

declare @Tuoi int

fetch next from GiaoVienCursor into @MaGV, @Tuoi

while @@FETCH\_STATUS = 0

begin

if @Tuoi > 50

begin

update dbo.GIAOVIEN set LUONG = 2500

where @MaGV = MAGV

end

else if @Tuoi > 40 and @Tuoi < 50

begin

update dbo.GIAOVIEN set LUONG = 2000

where @MaGV = MAGV

end

else

begin

update dbo.GIAOVIEN set LUONG = 1500

where @MaGV = MAGV

end

fetch next from GiaoVienCursor into @MaGV, @Tuoi

end

close GiaoVienCursor

deallocate GiaoVienCursor

29.Store

/\*

- Store PROCEDURE:

Stored procedures thường được sử dụng trong các hệ thống lớn,

nơi mà việc quản lý hiệu suất, bảo mật và tính toàn vẹn dữ liệu

là rất quan trọng.

- Lưu trữ chương trình hay thủ tục để tái sử dụng đoạn code

- Thực hiện lệnh Transact-SQL EXECUTE (EXEC) thực thi các stored procedure.

- Store procedure khác với các hàm xử lý là giá trị trả về của chúng

- Không chứa trong tên và chúng không được sử dụng trực tiếp trong biểu thức

\*/

/\*

- Động: Có thể chỉnh sửa khối lệnh, tái sử dụng nhiều lần

- Nhanh hơn: Tự phân tích cú pháp cho tối ưu. Và tạo bản sao để lần chạy sau không cần thực thi lại từ đầu.

- Bảo mật: Giới hạn quyền cho user nào sử dụng user nào không

- Giảm bandwidth: với các gói transaction lớn. Vài ngàn dòng lệnh một lúc thì dùng sotre sẽ đảm bảo.

\*/

/\*

- CREATE PROC <Tên\_store>

- [Parameter nếu có]

as

begin

<code xử lý>

end

Nếu chỉ là câu truy vấn thì có thể không cần begin và end

\*/

create proc usp\_Test

@MAGV nvarchar(10), @Luong int

as

begin

select \* from dbo.GIAOVIEN

where MAGV = @MAGV and LUONG = @Luong

end

go

-- C1: không cần thứ tự

exec dbo.usp\_Test @MaGV = N'', @Luong = 0

-- C2: phải tuần tự

exec dbo.usp\_Test N'', 0

-- execute dbo.usp\_Test @MaGV = N'', @Luong = 0

-- execute dbo.usp\_Test N'', 0

go

create proc usp\_SelectAllGiaoVien

as select \* from dbo.GIAOVIEN

go

exec dbo.usp\_SelectAllGiaoVien

30. Function

Tổ chức và thực thi mã lệnh một cách hiệu quả, có cấu trúc, và có thể tái sử dụng

-- tao function co parameter

create function uf\_LUONGGV(@MAGV char(10))

returns int

as

begin

declare @Luong int

select @Luong = LUONG from dbo.GIAOVIEN where MAGV = @MAGV

return @Luong

end

31. Trigger

Triggers giúp tự động hóa các công việc kiểm tra ràng buộc, duy trì tính toàn vẹn của dữ liệu, hoặc ghi lại các thay đổi trong cơ sở dữ liệu.

-- Trigger sẽ được gọi mỗi khi có thao tác thay đổi thông tin bảng

-- Inserted: Chứa những trường đã insert | update vào bảng

-- Deleted: Chứa những trường đã bị xóa khỏi bảng

create trigger utg\_InsertGiaoVien on dbo.GIAOVIEN

for insert, update

as

begin

-- rollback tran -- hủy bỏ thay đổi cập nhật bảng

print 'Trigger2'

end

go

-- Không cho phép xóa thông tin của giáo viên có tuổi < 1

create trigger utg\_AbortOlderThan40 on dbo.GIAOVIEN

for delete

as

begin

declare @Count int = 0

select @count = count(\*) from deleted

where (year(getdate()) - year(deleted.NGSINH)) < 1

if (@count > 0)

begin

print N'Không được xóa người > 40 tuổi'

rollback tran

end

end

32. Trainsaction

Transaction trong SQL là một nhóm các thao tác SQL được thực thi như một đơn vị duy nhất, đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong cơ sở dữ liệu. Một transaction có thể bao gồm một hoặc nhiều câu lệnh SQL, và tất cả các câu lệnh này phải thành công hoặc thất bại cùng nhau.

select \* from dbo.NGUOITHAN where TEN = 'Vy'

begin transaction

delete dbo.NGUOITHAN where TEN = 'Hùng'

-- chuỗi thao tác loằn ngoằn phức tạp

rollback -- hủy bỏ trans

--------------------------------------------------------------------

begin transaction

delete dbo.NGUOITHAN where TEN = 'Hùng'

-- chuỗi thao tác loằn ngoằn phức tạp

commit -- chấp nhận trans

declare @Trans varchar(20)

select @Trans = 'Trans1'

---------------------------------------------------------------------

begin transaction @trans

delete dbo.NGUOITHAN where TEN = 'Hùng'

-- chuỗi thao tác loằn ngoằn phức tạp

commit transaction @Trans -- chấp nhận trans

---------------------------------------------------------------------

begin transaction

save transaction Trans1

delete dbo.NGUOITHAN where TEN = 'Hùng'

save transaction Trans2

delete dbo.NGUOITHAN where TEN = 'Vy'

rollback transaction Trans1

33. EXISTS được sử dụng để kiểm tra sự tồn tại của các hàng trong một tập kết quả con (subquery). Nó trả về giá trị TRUE nếu tập kết quả con có ít nhất một hàng, và FALSE nếu không có hàng nào.