

SQL Store Procedure

1. Creating a Store procedure

```
Create proc [StoreProcName]
    @para1 datatype, @para2 datatype...
as
begin
    print 'Hello'
    print 'You can code here'
end
go
```

2. Datatype:

Number

Bit: 0 and 1 (1 bit)

Tinyint: 0 to 255 (1 byte)

Smallint: -32,768 to 32,767 (2 bytes)

Int: -2^{31} to $2^{31}-1$ (4 bytes)

Bigint: -2^{63} to $2^{63}-1$ (8 bytes)

Real: -3.4^{38} to -1.18^{38} , 0, and 1.18^{38} to 3.4^{38} (4 bytes)

Smallmoney: -214,748.3648 to 214,748.3647 (4 bytes)

Money: -922,337,203,685,477.5808 to 922,337,203,685,477.5807 (8 bytes)

String

char(n): 1 byte per character, defined by n up to a maximum of 8000 bytes

varchar(n): 1 byte per character, stored up to a maximum of 8000 bytes

text: 1 byte per character, stored up to a maximum of 2 GB

nchar(n): 2 bytes per character, defined by n up to a maximum of 4000 bytes

nvarchar(n): 2 bytes per character, stored up to a maximum of 4000 bytes

ntext: 2 bytes per character stored up to a maximum of 1 GB

Datetime

Datetime: 01/01/1753 → 12/31/9999

Smalldatetime: 01/01/1900 → 06/06/2079

Date: 01/01/0001 to 12/31/9999

Time: 00:00:00.0000000 → 23:59:59.9999999 nanoseconds

User-defined datatype:

sp_addtype

[@typename =] type,

[@phystype =] system_data_type,

[[@nulltype =] 'null_type']

EXEC sp_addtype CMND, 'varchar(11)', 'NOT NULL'

EXEC sp_addtype NgaySinh, datetime, 'NULL'

EXEC sp_addtype SoDT, 'varchar(24)', 'NOT NULL'

EXEC sp_addtype SoFax, 'varchar(24)', 'NULL'

3. Variable Declaration:

Declare @VariableName datatype

Variable name must begin with @

Declare @n int

Declare @s varchar(10)

Declare @p datetime

Declare @Sum float, @Count int

4. If:

if (a condition use '>' '<' '!= ' '=' '>=' '<=')

begin

.....

....

end

else

begin

...

....

end

5. While

While (Condition)

begin

.....

.....

end

6. Note:

Return, break, continue

7. Operation =:

```
declare @i int
set @i=1
declare @v nvarchar(20)
set @v=N'Nguyễn Văn A'
declare @g datetime
set @g='10/22/2015'
```

It is also possible to assign value to a variable by query instead of set instruction

```
Declare @var1 int, @var2 nvarchar(50)
select @Var2 = HoTen, @Var1 = Tuoi
from SV
where MaSV = 1
```

8. Changing datatype:

cast (@VariableName as DataType)

Ex:

```
Declare @i int
Set @i=123
Declare @u varchar(10)
set @u=cast(@i as varchar(10))
Declare @k varchar(10)
Set @k='123'
Declare @j int
Set @j= cast(@k as int)
```

9. case

CASE [input_expression]

WHEN when_expression THEN result_expression

[...n]

[ELSE else_result_expression]

END

Select * From NHAN_VIEN

Where datediff(yy, NgaySinh, getdate())

>= Case Phai

when 'Nam' then 60

when 'Nu' then 55

End

Select MaNV, HoTen, 'Loai' = Case

when CapBac<=3 then 'Binh Thuong'

when CapBac is null then 'Chua xep loai'

else 'Cap Cao' End

From NhanVien

10. Call a Store Procedure

Exec StoreName [value of para1],[value of para2]

11. Example

Check the number to be a prime or not?

create proc isprime

@n int

as

begin

declare @i int

set @i=2

while (@i<=@n-1)

begin

```

        if(@n% @i=0)
        begin
            print cast(@n as varchar(10)) + ' is not prime'
            return
        end
        set @i=@i+1
    end
    print cast(@n as varchar(10)) + ' is prime'
end
go
Exec isprime 10
--Compute the factorial of n
create proc factorial
    @n int
as
begin
    declare @i int
    declare @s int
    set @i=1
    set @s=1
    while(@i<@n)
    begin
        set @s=@s*@i
        set @i=@i+1
    end
    print cast(@N as varchar(5)) +'!=' +cast(@s as varchar(5))
end

```

go

Exec factorial 6

12. Returning a value from a store procedure by a output para.

-CALCULATE THE SUM OF THE PRIME NUMBERS FROM 1 TO N

create proc isprime

 @n int, @kq int output

as

begin

 declare @i int

 set @i=2

 while (@i<=@n-1)

 begin

 if(@n% @i=0)

 begin

 set @kq=0

 return

 end

 set @i=@i+1

 end

 set @kq=1

end

go

create proc primesum

 @n int

as

begin

```

declare @i int
declare @s int
set @i=1
set @s=0
while(@i<=@n)
begin
    declare @kq int
    exec isprime @i,@kq output
    if (@kq=1)
        set @s=@s+@i
    set @i=@i+1
end
print 'Result: '+cast(@s as varchar(10))
end
go
exec primesum 10

```

13. Load a value from DB to assign it to a variable

```

declare @VariableName1 datatype
declare @VariableName2 datatype
select @VariableName1=column1, @VariableName2 =column2
from tables
where conditions

```

Ex: Print the teacher name of @teacherID


```

create proc GetTeacherName
    @TeacherID nchar(10)
as
begin
    declare @TeacherName nvarchar(50)
    select @TeacherName=t.name
    from teacher t
    where t.ID=@TeacherID
    print @TeacherName
end
go
exec GetTeacherName 'GV00001'

```

Exercise

- A. Compute $1.2.3 \dots @n$. Return the result by output para
 Create proc Factorial @n int, @f int output
- B. Show $1! + 2! + \dots + @n!$. Print the result on the screen
 Create proc FactorialSum @n int.....
- C. List the students who have passed @SubjectName.
 Create proc StudentList @SubjectName nvarchar(50).....
- D. Show the average grade of @StudentName. Print the result on the screen and
 return the result by output para
 Create proc GetAverGrade @StudentName nvarchar(50), @Grade float output....