select \* from employee

select \* from employee

select \* from employee

select \* from employee

select \* from employee

select \* from employee

select \* from employee

These statements are sent to SQL Server Database Engine separately

Also the Queries are always compiled

select \* from employee

select \* from employee

select \* from employee

select \* from employee

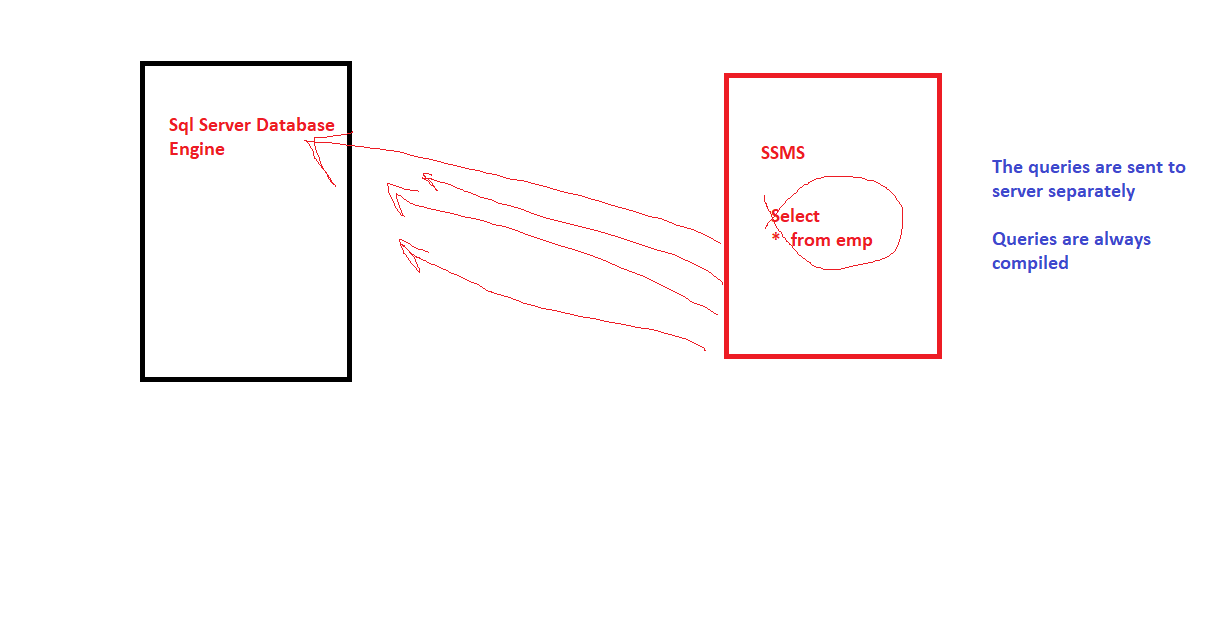
select \* from employee

select \* from employee

select \* from employee

Go

This Go keyword will create a block of these statements, they will go together to Sql Server Database Engine , but there is no name for this block



Stored Procedure > It’s a block of Sql Statements

Purpose > Reusability

It’s stored in a pre-compiled form

Procedure may or may not return a value

create proc GetEmployees

AS

Begin

select \* from employee

End

create proc GetEmployees

AS

Begin

select \* from employee

End

exec GetEmployees

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

create proc InsertEmployee

AS

Begin

insert into employee (id, name, address,deptid, perks,salary)

values (11,'Farhan','Delhi',1,2100,23000)

End

Call a Procedure

exec InsertEmployee

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

* Procedure which takes input parameters

alter proc InsertEmployee (@id int , @name varchar(20),

@address varchar(30), @deptid int, @perks int , @salary int)

AS

Begin

insert into employee (id, name, address,deptid, perks,salary)

values (@id,@name,@address,@deptid, @perks,@salary)

End

* Call a procedure which takes parameters

exec InsertEmployee 12,'Agar','Delhi' ,1 ,2100, 23000

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

* Procedure which returns a value

alter proc InsertEmployee (@id int , @name varchar(20),

@address varchar(30), @deptid int, @perks int , @salary int)

AS

Begin

if(Exists(Select \* from employee where id=@id))

return 0

else

begin

insert into employee (id, name, address,deptid, perks,salary)

values (@id,@name,@address,@deptid, @perks,@salary)

return 1

End

End

* Call a procedure which returns a value

declare @flag int

exec @flag= InsertEmployee 13,'Agar','Delhi' ,1 ,2100, 23000

if (@flag=1)

print 'Inserted'

else

print 'Record with this ID already exist'

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

* Procedure can return values through 2 ways
* 1. By using return statement > Procedure can return only 1 value and that too integer
* 2. By using output parameters, procedure can return more than 1 value

Now we want to pass Id of Employee & get his details

* Procedure which returns multiple values by using output parameters

create proc GetEmployeeDetails(@id int , @name varchar(20) output,

@address varchar(30) output , @salary int output)

AS

Begin

Select @name = name , @address = address , @salary = salary

from employee where id=@id

End

* Call a procedure which returns values using output paramters

declare @name varchar(20)

declare @address varchar(30)

declare @salary int

exec GetEmployeeDetails 1, @name output , @address output, @salary output

print @name

print @address

print @salary

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

Procedure to Update

create proc UpdateEmployee(@id int, @salary int, @address varchar(30))

AS

Begin

Update employee set address = @address, salary = @salary where id=@id

End

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

Procedure to Delete

create proc DeleteEmployee(@id int)

AS

Begin

Delete employee where id=@id

End

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

Functions are used to perform some specific task

create function GetNumber()

returns int

As

Begin

return 1

End

* To Call Function

select DBo.GetNumber()

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

create function CombineStrings (@string1 varchar(20) , @string2 varchar(20))

returns varchar(50)

As

Begin

return @string1 + ' ' + @string2

End

* To Call Function

select dbo.CombineStrings('Deepak','Kumar')

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

Functions have to always return a value

SP can have all statements

Functions can have only Select statement

SP are stored in pre-compiled form

Functions are always compiled when you call them

SP can have exception handling , functions dont have

SP can call functions, Functions can not call SP

SP can have input , output parameters

Functions can only have input parameters