GO statement is used to group the commands into batches and the advantage is that even if there’s an error in one group of commands the other statements get executed.

1. create a SP which accepts deptno and display all emp details who belong to that deptno

CREATE PROCEDURE GetEmpDetailsByDeptNo(@deptno int)

AS

Begin

SELECT empno,ename,sal,job,deptno,comm,mgr from emp

WHERE deptno=@deptno

END

EXEC GetEmpDetailsByDeptNo 20

2. create a SP which accepts empno and display his/her annual sal

CREATE PROCEDURE GetAnnualSalary(@empno int)

AS

BEGIN

SELECT sal\*12 AS 'Annual Salary' FROM emp

WHERE empno=@empno

END

EXEC GetAnnualSalary 7369

3. create a SP which accepts deptno and display  
how many employees working in that deptno

CREATE PROCEDURE NoOfEmpByDeptNo(@deptno int)

AS

BEGIN

SELECT COUNT(empno) FROM emp

WHERE deptno=@deptno

END

EXEC NoOfEmpByDeptNo 30

4. create a SP which accepts deptno and display what is the min and max sal taken in that deptno

CREATE PROCEDURE GetMin\_MaxSal(@deptno int)

AS

BEGIN

SELECT MIN(sal) AS 'MinSal',Max(sal) AS 'MaxSal' FROM emp

WHERE deptno=@deptno

END

EXEC GetMin\_MaxSal 10

1. create function which accepts empno and return his/her annual sal

CREATE FUNCTION GetAnnualSal(@empno int)

returns int

BEGIN

DECLARE @AnnualSal int

SELECT @AnnualSal=sal\*12 FROM emp WHERE empno=@empno

return @AnnualSal

END

SELECT dbo.GetAnnualSal(7369)

1. create a function which accepts deptno and returns no of employee available in that deptno

CREATE FUNCTION NoOfEmpByDepartmentNo(@deptno int)

returns int

AS

BEGIN

DECLARE @Count int

SELECT @Count=COUNT(empno) FROM emp

WHERE deptno=@deptno

return @Count

END

SELECT dbo.NoOfEmpByDepartmentNo(10)

1. Create a function which accepts mgrid and display all emps who report to that person

CREATE FUNCTION GetEmpByMgrID(@MgrID int)

RETURNS TABLE

AS

RETURN

(

SELECT \* FROM emp WHERE mgr=@MgrID

);

SELECT \* FROM GetEmpByMgrID(7839)

1. Create a function which accepts num as a salary and display all emps who get sal more than given sal

CREATE FUNCTION GetSalgreaterthanGiven(@sal int)

RETURNS TABLE

AS

RETURN

(

SELECT \* FROM emp WHERE sal>@sal

);

SELECT \* FROM GetSalgreaterthanGiven(1250)

9.SP to delete a row in a table

CREATE PROCEDURE DelDept(@deptno int)

AS

BEGIN

DELETE FROM dept

WHERE deptno=@deptno

END

Exec DelDept 50

10.SP to change a particular field in a row

CREATE PROCEDURE ChangeDeptLoc(@deptno int,@loc VARCHAR(50))

AS

BEGIN

UPDATE dept

SET loc=@loc

WHERE deptno=@deptno

END

Exec ChangeDeptLoc 50,'Bengaluru'

11.SP with Output Parameter

ALTER PROCEDURE GetEmployeeCountByDeptNoOutputParameter

@deptno int,

@EmployeeCount int Output

AS

BEGIN

SELECT @EmployeeCount=COUNT(empno) FROM emp

WHERE deptno=@deptno

END

Declare @EmployeeCount int

EXEC GetEmployeeCountByDeptNoOutputParameter 30,@EmployeeCount OUTPUT

Print @EmployeeCount

Numeric Functions

1. SELECT CEILING(1.70)
2. SELECT CEILING(1.30)
3. SELECT RAND(1)
4. SELECT RAND()
5. SELECT ROUND(235.415, 2) AS RoundValue;
6. SELECT SQRT(100)

STRING Functions

1. SELECT ASCII('A') AS A
2. SELECT CHAR(65) AS '65'
3. SELECT CONCAT('Hi! ','Good morning');
4. SELECT LEFT('Hi! ',1);
5. SELECT LEN('Good morning');
6. SELECT LOWER('GOOD');
7. SELECT LTRIM(' Hi')
8. SELECT REPLACE('Hi Good morning','Hi','Hello')
9. SELECT REPLICATE('0', 4)
10. SELECT REVERSE('REVERSE')
11. SELECT RIGHT('Hi! Good morning ',9);
12. SELECT RTRIM(' Hi ')
13. SELECT SUBSTRING('Hi Good morning',3,13)
14. SELECT TRIM(' Hi ')
15. SELECT UPPER('Good morning');

Date and Time Functions

1. DECLARE @time time(4) = '12:15:04.1237';
2. DECLARE @datetime2 datetime2(3) = @time;
3. SELECT @datetime2 AS '@datetime2', @time AS '@time';
4. SELECT
5. CAST('2007-05-08 12:35:29. 1234567 +12:15' AS time(7)) AS 'time'

,CAST('2007-05-08 12:35:29. 1234567 +12:15' AS date) AS 'date'

1. SELECT CAST( GETDATE() AS Date ) ;
2. SELECT DATENAME(year,'2007-10-30 12:15:32.1234567 +05:10');
3. SELECT DATEPART(week, '2007-04-21 ')
4. SELECT DATEPART(year, '2007-04-21 ')
5. SELECT DAY('2015-04-30')
6. SELECT MONTH('2015-04-30')
7. SELECT YEAR('2015-04-30')
8. SELECT DATEDIFF(day,'2036-02-28','2036-03-01');
9. SELECT DATEADD(year,15, '20060731');
10. SELECT DATEADD(month, 12, '20060731');
11. IF ISDATE('2009-05-12 ') = 1

PRINT 'VALID'

ELSE

PRINT 'INVALID';

1. SELECT

ISDATE('2020-12-05 11:20:30')