TSQL ASSIGNMENT DAY-01

Puneeth Puligundla 20249

1. What is the significance of ‘GO’?

A. The ‘GO’ clause is used as a batch separator i.e it separates a set of SQL statements into batches. All statements after the previous GO and the ones, before the present GO are treated as a batch and are sent to the SQL Server engine for execution. The variable declared in a batch are not visible outside the batch. It can also be used to execute a T-SQL statement multiple times.

Syntax:

T-sql statements  
 ------------- BATCH 1  
GO

T-sql statements  
 --------------BATCH 2  
GO

RECOGNIZED IN: Sql Server Management Studio (SSMS), SQLCMD, OSQL.

2. Numeric Functions in T-SQL

A.

SELECT CEILING(30.4)

SELECT FLOOR(19.4)

SELECT Rand(7)

SELECT ROUND(18.4354,2)

SELECT sqrt(225)

3. String Functions in TSQL

select ascii(‘a’)

SELECT CHAR(98)

SELECT CONCAT('Puneeth ', 'Puligundla');

SELECT LEFT('SQL Training', 5);

select len('Practice')

select lower('MYNAME')

SELECT LTRIM(' Hai Hello!')

SELECT REPLACE('Practice', 'P', ' ')

SELECT REPLICATE('Practice', 7);

SELECT reverse('Palindrome')

SELECT RIGHT('SQL Training', 5);

SELECT RTRIM('Hai Hello! ')

SELECT SUBSTRING('SQL Training', 1, 3)

select upper('myname')

SELECT TRIM(' Practice ')

4. Date Functions

DECLARE @date date= '99-07-19';

DECLARE @datetime datetime= @date;

SELECT @date AS '@date', @datetime AS '@datetime';

DECLARE @date date = '1999-07-19';

DECLARE @datetimeoffset datetimeoffset(3) = @date;

SELECT @date AS '@date', @datetimeoffset AS '@datetimeoffset';

DECLARE @date date = '1999-07-19'

DECLARE @datetime2 datetime2(3) = @date;

SELECT @date AS '@date', @datetime2 AS '@datetime2(3)';

6. Transact SQL Queries

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

that deptno

CREATE PROCEDURE GetAllEmpDetailsWithDno (@dno int)

AS

BEGIN

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

FROM emp

WHERE deptno=@dno

END

EXEC GetAllEmpDetailsWithDno 20

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

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

CREATE PROCEDURE DispEmpSal(@empno int)

AS

BEGIN

SELECT sal\*12 'ANNUAL SAL'

FROM emp

WHERE empno=@empno

END

EXEC DispEmpSal 7499

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

3. Create a SP which accepts deptno and displayhow many employees working in

that deptno

CREATE PROCEDURE DispCountEmp(@deptno int)

AS

BEGIN

SELECT COUNT(empno) 'COUNT OF EMPS'

FROM emp

WHERE deptno=@deptno

END

EXEC DispCountEmp 20

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

4. Create a SP which accepts deptno and display what is the min and max

sal taken in that deptno

CREATE PROCEDURE DispMinNMaxSal(@deptno int)

AS

BEGIN

SELECT MIN(sal) 'MIN SAL',MAX(sal) 'MAX SAL'

FROM emp

WHERE deptno=@deptno

END

EXEC DispMinNMaxSal 20

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

5. Create function which accepts empno and return his/her annual sal

CREATE FUNCTION GetAnnualSal(@empno int)

returns int

BEGIN

DECLARE @sal int

SELECT @sal=sal

FROM emp WHERE empno=@empno

return @sal\*12

END

SELECT dbo.GetAnnualSal(7369)

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

6. Create a function which accepts deptno and returns no of employee available

in that deptno

CREATE FUNCTION GetCountEmps(@deptno int)

returns int

BEGIN

DECLARE @count int

SELECT @count=count(empno)

FROM emp WHERE deptno=@deptno

return @count

END

SELECT dbo.GetCountEmps(20)

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

7. Create a function which accepts mgrid and display all emps who report to

that person

CREATE FUNCTION GetDetailsByMgrId(@mgr int)

returns TABLE

AS

return

(

SELECT empno,ename,mgr

FROM emp

WHERE mgr=@mgr

);

SELECT \* FROM GetDetailsByMgrId(7698)

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

8. Create a function which accepts num as a salary and display all emps who get

sal more than given sal

CREATE FUNCTION GetDetailsBySal(@sal int)

returns TABLE

AS

return

(

SELECT empno,ename,sal

FROM emp

WHERE sal>@sal

);

SELECT \* FROM GetDetailsBySal(1000)

7. STORED PROCEDURE USING OUTPUT PARAMETER

A. CREATE 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