TSQL ASSIGNMENT DAY-01

Puneeth Puligundla 20249

1. TRIGGERS

1. -- Cursor Creation and example

UPDATE emp SET job='TRAINEE' WHERE empno=7499

DISABLE TRIGGER TrgCantDelPres ON emp

ENABLE TRIGGER TrgCantDelPres ON emp

DECLARE empcursor CURSOR FOR

SELECT empno,ename,sal

FROM emp ORDER BY sal desc

DECLARE @name varchar(50),@sal int,@empno int

OPEN empcursor

FETCH NEXT FROM empcursor INTO @empno,@name,@sal

WHILE @@FETCH\_STATUS=0

BEGIN

PRINT @name + 'IS EARNING ' + CAST(@sal AS varchar)

FETCH NEXT FROM empcursor INTO @empno,@name,@sal

END

CLOSE empcursor

DEALLOCATE empcursor

GO

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

2. -- INSTEAD OF TRIGGER

CREATE TRIGGER TrgInsOfDel ON emp

INSTEAD OF DELETE

AS

DECLARE @empno int;

DECLARE @job varchar(50);

SELECT @job=d.job FROM deleted d;

SELECT @empno=d.empno FROM deleted d;

BEGIN

IF(@job='PRESIDENT')

BEGIN

RAISERROR('Cannot delete President',16,1);

END

ELSE

BEGIN

DELETE FROM emp where empno=@empno;

PRINT('Record Deleted From Table');

END

END

GO

DELETE FROM emp WHERE empno=7839  
---------------------------------------------------------------------------------------------------

3. --Incorrect Update Trigger

CREATE TRIGGER TrgIncorrectUpd ON emp

INSTEAD OF UPDATE

AS

DECLARE @empno int;

DECLARE @sal int;

DECLARE @salBefore int;

SELECT @sal=i.sal FROM inserted i;

SELECT @empno=i.empno FROM inserted i;

SELECT @salBefore=emp.sal FROM emp WHERE emp.empno=@empno;

BEGIN

IF(@sal<=@salBefore)

BEGIN

RAISERROR('Cannot decrease hike in salary',16,1);

END

ELSE

BEGIN

UPDATE emp SET sal=@sal where empno=@empno;

PRINT('Record Updated From Table');

END

END

GO

SELECT \* FROM emp

UPDATE emp

SET sal=900

WHERE empno=7369

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

4. --Cannot Delete on Tuesdays

CREATE TRIGGER TrgIncDelOnTues ON emp

INSTEAD OF DELETE

AS

DECLARE @empno int;

DECLARE @dt DATETIME;

SELECT @empno=d.empno FROM deleted d;

SELECT @dt=GETDATE();

BEGIN

IF (CAST(DATENAME(dw,@dt) AS varchar))='TUESDAY'

BEGIN

RAISERROR('Cannot delete on tuesdays',16,1);

END

ELSE

BEGIN

DELETE FROM emp WHERE empno=@empno;

END

END

GO

DELETE FROM emp WHERE empno=7369

2. DROP A TRIGGER

A. DROP TRIGGER IF EXISTS TrgIncDelOnTues

3. ENABLE AND DISABLE TRIGGERS

A.

CREATE TRIGGER TrgCantDelPres ON emp

INSTEAD OF UPDATE

AS

DECLARE @empid int;

DECLARE @job varchar(50);

SELECT @empid=i.empno FROM inserted i;

SELECT @job=i.job FROM inserted i;

BEGIN

IF(@job='PRESIDENT')

BEGIN

RAISERROR('Cannot Update PRESIDENT',16,1);

END

ELSE

BEGIN

UPDATE emp SET job=@job WHERE empno=@empid;

PRINT ('TRIGGER CALLED')

END

END

GO

UPDATE emp SET job='PRESIDENT' WHERE empno=7499

DISABLE TRIGGER TrgCantDelPres ON emp

UPDATE emp SET job='PRESIDENT' WHERE empno=7499

ENABLE TRIGGER TrgCantDelPres ON emp

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. The SET NOCOUNT in T-SQL

A. The SET NOCOUNT { ON | OFF } when set to ON, stops the message that shows the count of the number of rows affected by a T-SQL statement or a Stored Procedure from being returned as a part of the result set.

SET NOCOUNT ON ---------------- Count is NOT Returned

SET NOCOUNT OFF ---------------- Count is Returned

The main purpose of this is to provide a performance boost by reducing the network traffic greatly. It is in effect at execution at the run time and not at parse time.

Permission requires membership in the public role.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_