# EXAMEN SEPTIEMBRE BASE DATOS HECTOR GARCIA GONZALEZ

1)

USE SepProfe

GO

CREATE FUNCTION DBO.ANTIGUEDAD\_CUIDADORES12315(@CODcuidador int ) RETURNS DECIMAL AS

BEGIN

DECLARE @ANTIGUEDAD INT

SET @ANTIGUEDAD = 0

SELECT @ANTIGUEDAD = DATEDIFF(YY,cuidadores.fechaingreso, GETDATE()) FROM cuidadores where cuidadores.codcuidador = @CODcuidador

RETURN @ANTIGUEDAD

END

GO

SELECT DBO.ANTIGUEDAD\_CUIDADORES12315(2),cuidadores.nombrecuidador from cuidadores

GO

**2)XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**

USE SepProfe

GO

CREATE FUNCTION DBO.FUNCION33(@DIA1 varchar(20), @DIA2 varchar(20), @centroPROCEDENCIA varchar(20)) returns table AS

RETURN

(SELECT cuidadores.nombrecuidador , animales.precio, animales.nombreanimal FROM cuidadores,animales

WHERE animales.identidad IN (SELECT procedencias.identidad FROM procedencias where procedencias.nombreentidad = @centroPROCEDENCIA)

AND animales.codanimal IN (SELECT horarios.codanimal FROM horarios WHERE horarios.codcuidador IN (SELECT cuidadores.codcuidador FROM cuidadores )

AND horarios.diasemana = @DIA1 OR horarios.diasemana = @DIA2 ))

GO

SELECT \* FROM DBO.FUNCION33('LUNES','MARTES','CARAFRE')

GO

3)

USE SepProfe

GO

CREATE FUNCTION DBO.FUNCION111(@numero int) RETURNS

TABLE

AS

return (SELECT COUNT(animales.codanimal) , procedencias.nombreentidad FROM animales, procedencias WHERE animales.identidad IN(SELECT procedencias.identidad FROM procedencias )

AND animales.estado = 'SANO' and COUNT(animales.codanimal) >@numero)

go

select \* from DBO.FUNCION111(1)

4)xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

USE SepProfe

GO

CREATE PROCEDURE DBO.procedimiento4 @codigocuidador SMALLINT, @DIAdelaSEMANA VARCHAR(20) , @NUMEROanimales INT OUTPUT AS

BEGIN

SET @NUMEROanimales= 0

SELECT @NUMEROanimales= COUNT(a.codanimal) FROM animales a ,horarios h WHERE a.codanimal IN (SELECT h.codanimal FROM horarios WHERE horarios.diasemana = @DIAdelaSEMANA

AND h.codcuidador = @codigocuidador)

RETURN @NUMEROanimales

END

GO

DECLARE @NUMEROanimales INT

EXEC DBO.procedimiento4 2,'LUNES', @NUMEROanimales OUTPUT

SELECT @NUMEROanimales

GO

**5)XXXXXXXXX**

USE SepProfe

GO

DECLARE CursorAnimales CURSOR SCROLL FOR SELECT \* FROM animales

OPEN CursorAnimales

FETCH FIRST FROM CursorAnimales

FETCH ABSOLUTE 3 FROM CursorAnimales

FETCH LAST FROM CursorAnimales

GO

CLOSE CursorAnimales

DEALLOCATE CursorAnimales

GO

6) USE SepProfe

GO

CREATE TRIGGER Trigger6 ON horarios

AFTER INSERT, DELETE AS

BEGIN

if exists (select count(\*) from inserted where inserted.horafin >13 )

declare @aumento smallint

set @aumento = 2

update cuidadores set manteniento = mantenimientos + @aumento where horarios.codcuidador in (select inserted.ccodcuidador from inserted)

if(select count(\*) from deleted )>1

update cuidadores set manteniento = mantenimientos + @aumento where

END

GO

7)XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

USE SepProfe

GO

CREATE PROCEDURE PROCEDIMIENTO7 @nombreENTIDAD varchar(20) AS

BEGIN

SELECT animales.nombreanimal AS ANIMAL\_CUIDADO, cuidadores.nombrecuidador AS NOMBRE\_CUIDADOR FROM procedencias, animales, horarios, cuidadores

WHERE procedencias.nombreentidad = @nombreENTIDAD and procedencias.identidad IN (SELECT animales.identidad FROM animales WHERE animales.codanimal IN

(SELECT horarios.codanimal FROM horarios WHERE horarios.codcuidador IN (SELECT cuidadores.codcuidador FROM cuidadores)))

--GROUP BY

END

GO

EXEC PROCEDIMIENTO7 'ASOFRA'

GO

**8) xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

USE SepProfe

GO

DECLARE @CUIDADOR VARCHAR(20) ,@ANIMALcuidado VARCHAR(20), @DIAsemana VARCHAR(20)

DECLARE MIcursorANIMALES CURSOR FOR SELECT c.nombrecuidador,a.nombreanimal,h.diasemana FROM horarios h

JOIN animales a ON h.codanimal = a.codanimal

JOIN cuidadores c ON c.codcuidador = h.codcuidador

OPEN MIcursorANIMALES

FETCH MIcursorANIMALES INTO @CUIDADOR ,@ANIMALcuidado , @DIAsemana

PRINT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

PRINT 'DIA DE LA SEMANA ' + @DIAsemana

PRINT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

WHILE (@@FETCH\_STATUS=0)

if @DIAsemana = 'LUNES'

BEGIN

PRINT 'CUIDADOR= ' + @CUIDADOR + ' ANIMAL= ' + @ANIMALcuidado

FETCH MIcursorANIMALES INTO @CUIDADOR ,@ANIMALcuidado , @DIAsemana

end

if @DIAsemana = 'MARTES'

BEGIN

PRINT 'CUIDADOR= ' + @CUIDADOR + ' ANIMAL= ' + @ANIMALcuidado

FETCH MIcursorANIMALES INTO @CUIDADOR ,@ANIMALcuidado , @DIAsemana

end

if @DIAsemana = 'MIERCOLES'

BEGIN

PRINT 'CUIDADOR= ' + @CUIDADOR + ' ANIMAL= ' + @ANIMALcuidado

FETCH MIcursorANIMALES INTO @CUIDADOR ,@ANIMALcuidado , @DIAsemana

end

GO

CLOSE MIcursorANIMALES

DEALLOCATE MIcursorANIMALES

GO

**9)xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

\*RECOVERY- LA AFIRMACION ES FALSA. La Recovery es la última instrucción del RESTORE. La base de datos esta activa y ya no permite posteriores procesos de restauración. Si se desea comprobar los resultados habrá que empezar de nuevo y realizar una copia completa.

\*NONRECOVERY-LA AFIRMACIÓN ES VERDADERA. La base de datos no es accesible y solo permite operaciones de restauración. Se usa cuando aparte de copias completas también hay copias diferenciales y del registro de transacción.

\*STANDBY- LA AFIRMACIÓN ES FALSA. La base de datos queda en forma de modo lectura y se usa para cuando después de una restauración se quiere comprobar cómo han quedado los resultados.

**10)xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

Un proceso por lotes es un conjunto de instrucciones T-SQL que se ejecutan en lote o dicho de otra forma, se ejecutan todas juntas. Si alguna de estas instrucciones falla, el resto de instrucciones de este lote también fallara.

Las diferencias entre un proceso por lotes y una transacción es que el proceso por lotes controla desde el lado cliente cuantas instrucciones se van a ejecutar y en una transacción se controla desde el lado del servidor cuanto trabajo va a tener que realizar SQL SERVER con esas operaciones hasta que hayan sido confirmadas las transacciones. Estas transacciones o bien son confirmadas con LA INSTRUCCIÓN COMMIT o bien deshechas con la instrucción ROLLBACK.