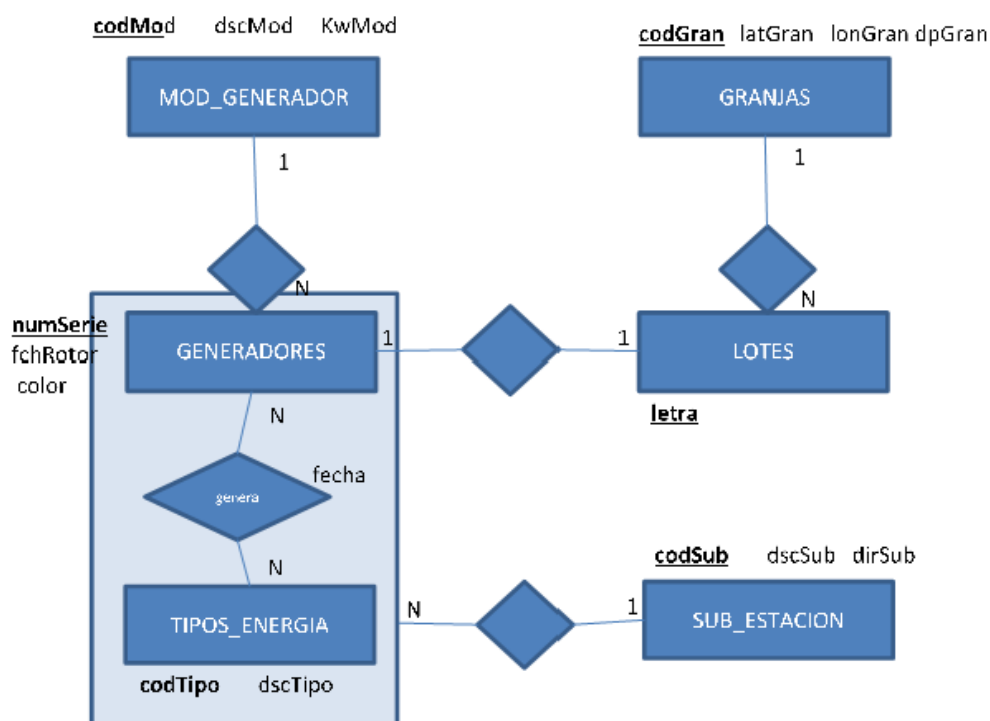


EVALUACION	Solución Parcial 2	GRUPO	N3A	FECHA	26/11/2019
MATERIA	Bases de Datos 2				
CARRERA	Analista Programador/Analista en Tecnologías de la Información				
CONDICIONES	<ul style="list-style-type: none"> - Duración: 2.5 horas - Sin material - 30 Puntos 				

Una empresa dedicada a la fabricación e instalación de aerogeneradores cuenta con las siguientes estructuras:



Modelo_Generador(idMod, dscMod, capMod, acumMod)
 Granjas(idGranja, latGranja, lonGranja, dptoGranja)
 Lotes(letraLote, idGranja)
 Generador(idMod, numSerie, fchRotor, color, letraLote, idGranja)
 Tipo_Energia(idTipo, dscTipo)
 SubEstacion(codSub, dscSub, dirSub)
 Genera(idGenera, idMod, numSerie, idTipo, fchGenera, cntGenera, codSub)
 Auditoria(idA, fecha, tabla, campo, anterior, actual, clave, observaciones)

Se pide:

1. Realizar un procedimiento almacenado que dado un Generador reste al campo *fchRotor* la cantidad de días que fue utilizado para generar energía, al finalizar la actualización verificar la cantidad de días que faltan para el mantenimiento (diferencia entre la fecha del día y la fecha del campo *fchRotor*) no sea inferior a 10, de serlo, entonces debe mostrar un mensaje por pantalla.

```
CREATE PROCEDURE Parcial_EJ1
@idMod int,@numSerie char(10)
AS
BEGIN
DECLARE
@dias_uso int
SELECT @dias_uso=COUNT(DISTINCT(genera.fchGenera))
FROM Genera
WHERE Genera.idMod=@idMod AND Genera.numSerie=@numSerie

UPDATE Generador
SET fchRotor = DATEADD(day,-@dias_uso,fchRotor)
WHERE idmod=@idMod AND numSerie=@numSerie

IF DATEDIFF(day,(SELECT fchRotor
FROM Generador
WHERE idMod=@idMod AND numSerie=@numSerie),getdate()) < 10

PRINT 'El generador '+ltrim(str(@idMod)) + ' ' + @numSerie + ' necesita mantenimiento'

END
```

2. Hacer una función almacenada que dado un código de Sub Estación, muestre el promedio del total de energía generada por cada Tipo de Energía en dicha Sub Estación.
Escriba una consulta SQL que muestre los datos de la Sub Estación y dicho promedio.

```
CREATE FUNCTION Parcial_EJ2(@codSub char(5))
    RETURNS int
AS
BEGIN
    DECLARE @retorno int

    SELECT @retorno=AVG(miTabla.Total)
    FROM
        (SELECT idTipo,SUM(cntGenera) as Total
         FROM Genera
         WHERE codSub=@codSub
         GROUP BY idTipo) miTabla

    RETURN @retorno
END

/* Uso de la funcion */
SELECT codSub,dscSub,dirSub,dbo.Parcial_EJ2(codSub) as Promedio
FROM SubEstacion
```

3. Escribir un disparador que luego de ingresada una Generación de Energía acumule la cantidad de energía generada por cada Modelo de Generador, se deben tener en cuenta que pueden existir inserciones múltiples.

```
CREATE TRIGGER Parcial_EJ3
ON Genera
AFTER INSERT
AS
BEGIN

    UPDATE Modelo_Generador
    SET acumMod = acumMod + (SELECT SUM(inserted.cntGenera)
                           FROM inserted
                           WHERE inserted.idMod=Modelo_Generador.idMod)

    WHERE idMod IN (SELECT idMod
                   FROM inserted)

END
```

4. Escribir un disparador que no permita el ingreso de un nuevo Generador si su modelo correspondiente tiene un total acumulado mayor al promedio de energía generada por todos los generadores en lo que va del año, se deben tener en cuenta la posibilidad de inserciones múltiples.

```
CREATE TRIGGER Parcial_EJ4
ON Generador
INSTEAD OF INSERT
AS
BEGIN

INSERT INTO Generador SELECT *
                        FROM inserted
                        WHERE idMod NOT IN (SELECT G.idMod
                                           FROM Genera G,inserted I
                                           WHERE G.idMod=I.idMod
                                           GROUP BY G.idMod
                                           HAVING SUM(G.cntGenera) > (SELECT AVG(cntGenera)
                                                                    FROM Genera))

END
```

5. Definir un disparador que registre en la tabla Auditoria si alguien realiza una modificación en la cantidad generada o en la fecha de una Generación, el registro de auditoría debe indicar claramente cada caso, tener en cuenta que es posible que existan modificaciones múltiples.

```
CREATE TRIGGER Parcial_EJ5
ON Genera
AFTER UPDATE
AS
BEGIN
    IF UPDATE(cntGenera)
        INSERT INTO Auditoria
            SELECT getdate(), 'Genera', 'cntGenera', d.cntGenera, i.cntGenera,
                i.idGenera, 'Se modifica cantidad'
            FROM inserted i, deleted d
            WHERE i.idGenera=d.idGenera

    IF UPDATE(fchGenera)
        INSERT INTO Auditoria
            SELECT getdate(), 'Genera', 'fchGenera', d.fchGenera, i.fchGenera,
                i.idGenera, 'Se modifica la fecha'
            FROM inserted i, deleted d
            WHERE i.idGenera=d.idGenera

END
```

6. Realizar un disparador que permita borrar uno o más Modelos de Generador.

```
CREATE TRIGGER Parcial_EJ6
ON Modelo_Generador
INSTEAD OF DELETE
AS
BEGIN

DELETE
FROM Genera
WHERE idMod IN (SELECT idMod
                FROM deleted)

DELETE
FROM Generador
WHERE idMod IN (SELECT idMod
                FROM deleted)

DELETE
FROM Modelo_Generador
WHERE idMod IN (SELECT idMod
                FROM deleted)

END
```