

EVALUACIÓN	Examen	GRUPO	Todos	FECHA	27/09/2019
MATERIA	Base de Datos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	- Puntos: Máximo:100 Mínimo: 70				

Ejercicio 1. MER y MR (50 puntos)

Una empresa dedicada a la fabricación e instalación de aerogeneradores desea crear un modelo de datos para su sistema de información, para ello cuenta con el siguiente análisis:

La empresa posee diversos modelos de generadores, los mismos están identificados con un código, se conoce también la descripción, y capacidad de generación expresada en Kw.

Existen varios generadores de cada modelo, se sabe que cada unidad generador está identificado con un número de serie, no pueden existir unidades de generadores si no están asociados a los modelos, dos modelos diferentes de generadores pueden tener el mismo número de serie, se sabe que además del número de serie, las unidades de cada generador tienen una fecha de cambio de rotor y un color.

La empresa posee granjas de generadores, cada granja está identificada con un código, se conoce la ubicación de cada granja (latitud y longitud) y el departamento donde se encuentra.

Cada granja posee lotes donde se instalan las unidades de generador, un lote está identificado con una letra, la misma letra puede repetirse en granjas diferentes.

Cada lote de una granja puede tener instalado una sola unidad de generador, cada unidad de generador está instalada en un solo lote de una granja.

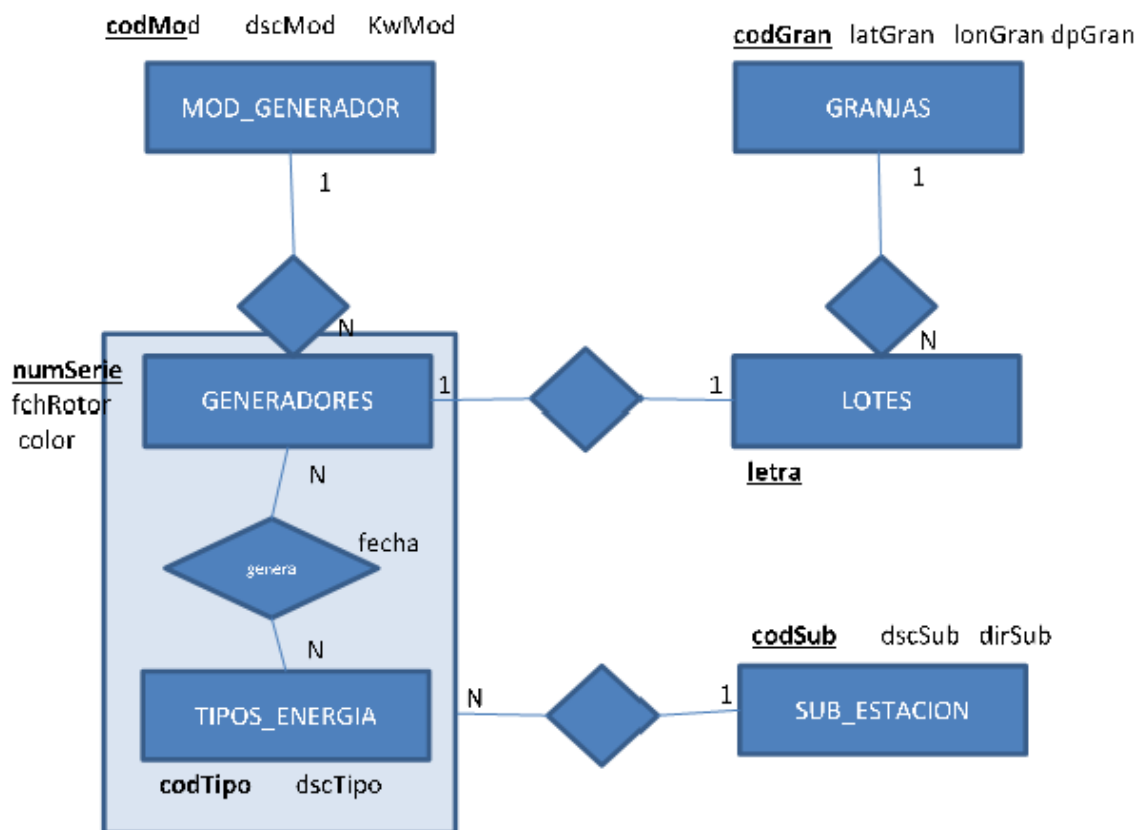
Cada unidad de generador genera muchos tipos diferentes de energía, un tipo de energía puede ser generada por muchos generadores, se sabe que los tipos de energía están identificados por un código, se conoce también una descripción general de dicho tipo de energía.

Es necesidad de la empresa que se registre la fecha que se genera cada tipo de energía.

Cada generación es enviada a una sub estación de control, una sub estación de control puede recibir muchas generaciones de energía, de las sub estaciones se conoce su código que la identifica, se sabe su descripción y su dirección.

SE PIDE:

- Diagrama Entidad Relación, explicitando supuestos y restricciones no estructurales en lenguaje natural. (35 puntos)
- Pasaje a Modelo Relacional de la solución de la parte anterior, el modelo debe quedar por lo menos en tercera forma normal 3NF (15 puntos)
- Explique brevemente por que es importante que el modelo sea llevado a la 3NF



```
MOD_GENERADOR(codMod, dscMod, KwMod)
GENERADORES(numSerie, codMod, fchRotor, color)
GRANJAS(codGran, latGran, lonGran)
LOTES(letra, codGran, numSerie, codMod)
TIPOS_ENERGIA(codTipo, dscTipo)
SUB_ESTACION(codSub, dscSub, dirSub)
GENERA(numSerie, codMod, codTipo, fecha, codSub)
```

Ejercicio 2. Consultas (50 puntos)

Un depósito en las afueras de la ciudad posee un sistema para el manejo de sus productos, se cuenta con las siguientes estructuras:

productos(sku,descrip,precio,stock)
cajas(cajanum,cajalargo,cajaancho,cajaalto)
pallet(numpallet,fchpallet)
entrega(numEntrega,sku,cajanum,fecha,cantidad,valor,numpallet)

1. Mediante el lenguaje de consultas SQL resolver los siguientes puntos:

a. Mostrar todos los datos de las cajas que están contenidos en la última entrega

```
SELECT cajas.*  
FROM cajas,entrega  
WHERE cajas.cajanum=entrega.cajanum and  
entrega.fecha = (SELECT MAX(fecha)  
FROM entrega)
```

b. Mostrar de los productos su sku y descripción, de las cajas su número y medidas y la cantidad total entregada de cada uno de esos productos y cajas, filtrar los resultados de tal manera que solo muestre aquellos registros con cantidades totales mayores a 10

```
SELECT productos.sku,productos.descrip,cajas.cajanum,cajas.cajaalto,  
cajas.cajaancho,cajas.cajalargo,SUM(entrega.cantidad) as cant  
FROM cajas,productos,entrega  
WHERE cajas.cajanum=entrega.cajanum and  
productos.sku=entrega.sku  
GROUP BY productos.sku,productos.descrip,cajas.cajanum,cajas.cajaalto,  
cajas.cajaancho,cajas.cajalargo  
HAVING SUM(entrega.cantidad) > 10
```

c. Para cada producto mostrar su sku, su descripción, el total entregado y la fecha de la entrega más vieja del año 2019.

- d. Del total de entregas por producto, mostrar el promedio, la cantidad menor y la cantidad mayor

```
SELECT AVG(miTabla.cantidad) as promedio,  
       MIN(miTabla.cantidad) as Menor, MAX(miTabla.cantidad) as Mayor  
FROM (SELECT sku, count(numEntrega) as cantidad  
      FROM entrega  
      GROUP BY sku) miTabla
```

- e. Mostrar los datos de los pallet, cuantas cajas y productos diferentes se entregaron en ellos, solo tener en cuenta aquellos registros cuyo valor total de entregas supere el promedio de precios de todos los productos

```
SELECT pallet.numpallet, pallet.fchpallet, COUNT(DISTINCT(sku)) as cntProd,  
       COUNT(DISTINCT(cajanum)) as cntCajas  
FROM pallet, entrega  
WHERE pallet.numpallet=entrega.numpallet  
GROUP BY pallet.numpallet, pallet.fchpallet  
HAVING SUM(valor) > (SELECT AVG(precio)  
                     FROM productos)
```