

Diplomado en Minería de Datos

Módulo 2: SQL

Proyecto Final

M. en C. Odín Miguel Ecorza Soria

24 de agosto de 2018

Fecha límite de entrega: 14 de septiembre de 2018, 9:00hrs

Elabora un script SQL que contenga las instrucciones para resolver los problemas que se plantean considerando última versión de la base de datos “dipl” que se ha usado durante las clases.

Problema 1

Eliminar de la base de datos todos los asentamientos donde no hay domicilios.

Problema 2

Mostrar los datos de la tabla **persona** para las personas que viven en un domicilio donde no habita nadie más.

Problema 3

Mostrar los siguientes datos:

1. Id del domicilio
2. Calle y número
3. Nombre del asentamiento
4. Nombre del municipio
5. Nombre del estado

Para los domicilios donde no vive nadie que están ubicados dentro de alguno de los 5 estados relacionados con más personas.

Problema 4

(2pts) Agregar una restricción sobre la columna **idcliente** de las tablas **persona** y **empresa** para evitar que dos tuplas de estas tablas puedan hacer referencia al mismo registro en **cliente**

Problema 5

Mostrar todos los datos de la tabla **persona** para los tocayos que viven en los mismos domicilios. Ordenar por identificador del domicilio, apellido paterno, apellido materno y nombre.

Problema 6

Insertar a 3000 registros en la tabla **pedido** tomando como nombres de tienda cadenas aleatorias de otras tablas de la base de datos y fechas aleatorias, se debe garantizar que los nuevos registros no violen ninguna de las restricciones sobre la tabla.

Problema 7

Insertar 12000 registros en la tabla `artspedido` garantizando que no se viole ninguna de las restricciones de la tabla.

Problema 8

Crear una vista que muestre los siguientes campos:

1. Clave del estado.
2. Número de personas registradas en el estado según la BD.
3. Número promedio de personas registradas por municipio en el estado.
4. Número promedio de personas registradas por asentamiento en el estado.
5. Número promedio de personas registradas por domicilio en el estado.

Problema 9

Crear un procedimiento almacenado `signo(date) -> text` que dada una fecha regrese el nombre del signo correspondiente¹, por ejemplo:

```
dipl=# select signo('2016-09-25');
+-----+
| signo |
+-----+
| libra |
+-----+
(1 row)
```

Problema 10

(2pts) Crear un procedimiento almacenado que, dada la clave de un estado, muestre la siguiente información:

1. Clave del estado.
2. Número de clientes (tanto personas como empresas) cuyo domicilio está en ese estado.
3. Número de pedidos realizados por dichos clientes.
4. Id del artículo más comprado en el estado.
5. Monto del pedido más costoso en el estado.

Problema 11

Crear una vista materializada (`persona_signo`) que muestre los siguientes datos:

1. idpersona
2. iddomicilio
3. idasentamiento
4. idmunicipio
5. idestado
6. signo zodiacal

Problema 12

Crear un disparador que refresque la vista materializada del ejercicio anterior cada vez que:

1. Se elimine una persona
2. Se agregue una persona
3. Se cambie la fecha de nacimiento de una persona

¹puedes crear tablas de apoyo para facilitar la implementación de esta función

Problema 13

Considerando la siguiente implementación:

```
create or replace function valida(y int, m int, d int) returns boolean as $$
select true;
$$ language sql;
create or replace function randate() returns date as $$
declare
    r_y int;
    r_m int;
    r_d int;
begin
    r_y := curandom(1910,100);
    r_m := curandom(1,11);
    r_d := curandom(1,30);
    while not valida(r_y,r_m,r_d) loop
        raise info 'Hubo fecha inválida';
        r_y := curandom(1910,100);
        r_m := curandom(1,11);
        r_d := curandom(1,30);
    end loop;
    return (r_y::text || '-' || r_m::text || '-' || r_d::text)::date;
end;
$$ language plpgsql;
```

Modifica la función `valida(int,int,int)->boolean` para que verifique que efectivamente los números que recibe como parámetro generan una fecha válida.