

### **PRÁCTICO Nº 3 – SQL Segunda Parte**

#### **Ejercicio 1**

Utilizando SQL (Motor de Base de datos PostgreSQL), crear las bases de Datos e Identificar las claves primarias, secundarias (cláusula UNIQUE), Claves foráneas; correspondientes a los ejercicios:

- a) Ejercicio 1 de la práctica 2(Base de datos sobre artículos). Considerar que el Stock mínimo no puede ser mayor que el Stock máximo; el precio no puede ser 0 ni negativo.
- b) Ejercicio 3 de la práctica 2(Base de datos sobre prestación de Pami). Considerar que el número de Pami es un entero positivo, el dominio del atributo nombre del articulo es {Cristales, Armazón, Audífono, Prótesis, Marcapaso}. Agregar a la tabla prestación, NroPestacion (entero positivo) y precio\_prestacion (numeric(10,2)).
- c) Considerando las tablas:  
Sala (nrosala, piso, capacidad)  
Película (nropelicula, titulo, estilo)  
Proyeccion (nrosala, nropelicula, fecha\_proyeccion)

Crear un dominio para el titulo de una película que es un varchar de 45. Considerar que el nro de sala no puede ser negativo, el dominio del atributo estilo de un película es {drama,comedia,clasica}.

#### **Ejercicio 2**

Utilizando SQL, agregar datos coherentes a las bases de datos creadas en los incisos a), b) y c) del ejercicio anterior.

#### **Ejercicio 3**

- a) Realice el ejercicio 1 b) con el motor de base de datos MySQL y agregue datos coherentes a la base de datos creada.
- b) Modifique la restricción en el nombre del Artículo del inciso b) del ejercicio 1(Base de datos sobre prestación de Pami), de manera de utilizar un dominio NombreArticulo y no la cláusula Check, de la misma manera, cree un dominio para Número de Pami. Utilice las bases de datos creadas con MYSQL y PostgreSQL.

#### **Ejercicio 4**

Realice el ejercicio 1 c) con el motor de base de datos MySQL. Agregar datos coherentes a las bases de datos creada.

#### **Ejercicio 5**

Modifique el Inciso b) del ejercicio 1(Base de datos sobre prestación de Pami), agregando cláusulas de integridad referencial (Foreign key) teniendo en cuenta que: Al borrar un Afiliado debe eliminarse toda la información de prestaciones realizadas al mismo; y al eliminar un articulo, no permitir hacerlo si hay alguna prestación en la que fue utilizado.

### Ejercicio 6

Sobre la Base de datos del ejercicio 1 b) (Base de datos sobre prestación de Pami) realizar en SQL las siguientes consultas:

- a) Obtener el promedio de cada artículo en las prestaciones que estuvo involucrado. Para que un artículo este en el resultado, su promedio debe ser mayor que 100.
- b) Obtener el máximo y mínimo precio de una prestación pagado por cada afiliado. Listar Nropami, Nombre y máximo y mínimo precio.
- c) Obtener la cantidad de prestaciones realizadas por cada afiliado.

### Ejercicio 7

- a) Utilizando un trigger, modifique la base de datos creada en el ejercicio 1a), para que luego de cada inserción de un artículo se guarde la descripción en mayúsculas.
- b) Utilizando un trigger, modifique la base de datos creada en el ejercicio 1a), para que luego de cada modificación en la cantidad de un artículo se guarde información de auditoría, esta información debe quedar almacenada en la tabla auditoriaArticulo (crearla antes de definir el trigger), la información a almacenar es nroarticulo, movimiento (diferencia de cantidades) y la fecha de realización de la actualización.
- c) Modifique el inciso b) del ejercicio 1 (Base de datos sobre prestaciones de Pami) considerando que el número de pami es autonumerado.
- d) Utilizando un trigger, modifique la base de datos creada en el ejercicio 4, considerando que el número de sala es autonumerado.

**NOTA:** Utilice las bases de datos creadas con MySQL y PostgreSQL. En caso de no tener la base de datos creada deberá realizarlo antes.

### Ejercicio 8

Dada las siguientes relaciones:

Canción (nombreCD, canción)

CDs (interprete, nombreCD, NombreCia, anio\_edicion)

Grabadora ( NombreCia, direccion)

Cree la base de datos del ejercicio y agregue la tabla canciones\_desactualizadas, con la misma estructura de la tabla canciones, luego de esto cree un trigger para que cada vez que se elimina un canción se inserte automáticamente esta canción en la tabla canciones\_desactualizadas . Cree la base de datos utilizando el motor de base de datos POSTGRES.

