



PRACTICO Nº 2

INTEGRIDAD Y SEGURIDAD DE LOS DATOS

- 1) Teniendo en cuenta el esquema de relaciones del TP 1 defina en SQL las reglas de integridad siguientes:
 - I. Crear un tipo de datos para las horas de vuelo de los pilotos y de los aviones. Las horas de vuelo no pueden ser un nro. Negativo ni mayores a 100000. Asociar los atributos mencionados al nuevo tipo de datos.
 - II. Crear una nueva relacion Jefe que tenga un atributo DNI que sea clave primaria y clave foranea de Trabajador y que cuando se elimine el trabajador asociado acarree la eliminacion de la tupla asociada en JEFE, lo mismo con las actualizaciones del DNI en trabajador
 - III. No deben permitirse pilotos con horas de vuelo mayores a las horas de vuelo del piloto que tiene la menor fecha de inicio
 - IV. No puede existir en ningun momento en la base de datos un DNI de un Piloto que a su vez sea un DNI de un TRABAJADOR que repara o reparó algún Avion

- 2) Considérese la siguiente base de datos relacional:
empleado (nombre-empleado, calle, ciudad)
trabaja (nombre-empleado, nombre-empresa, sueldo)
empresa (nombre-empresa, ciudad)
jefe (nombre-empleado, nombre-jefe)
Dese una definición en el DDL de SQL de esta base de datos. Identifíquense las restricciones de integridad referencial que deban cumplirse e inclúyanse en la definición del DDL.

- 3) Las restricciones de integridad referencial implican exactamente a dos relaciones. Considérese una base de datos que incluye las relaciones siguientes:
trabajador-fijo (nombre, despacho, teléfono, sueldo)
trabajador-tiempo-parcial (nombre, sueldo-por-hora)
dirección (nombre, calle, ciudad)

Supóngase que se desea exigir que cada nombre que aparece en dirección aparezca en trabajador-fijo o en trabajador-tiempo-parcial, pero no necesariamente en ambos.
 - a. Propóngase una sintaxis para expresar esta restricción.
 - b. Discútanse las acciones que debe realizar el sistema para aplicar una restricción de este tipo.



- 4) Indique si las siguientes operaciones serían aceptadas o rechazadas por el DBMS, justificando en cada caso

1) U1: CREATE TABLE Stocks_peli... ;	8) U2: GRANT INSERT ON Stocks_peli TO U4;
2) U1: CREATE TABLE Peli_en_video... ;	9) U1: REVOKE UPDATE ON Stocks_peli TO U2, U3 CASCADE;
3) U1: GRANT ALL PRIVILEGES ON Stocks_peli, Peli_en_video TO U5 WITH GRANT OPTION;	10) U4: GRANT UPDATE ON Stocks_peli TO U6 ;
4) U1: GRANT UPDATE ON Stocks_peli, T2 TO U2, U3 WITH GRANT OPTION;	11) U2: UPDATE Peli_en_video ...;
5) U2: UPDATE Peli_en_video ...;	12) U3: UPDATE Stocks_peli ...;
6) U5: GRANT INSERT , UPDATE ON Stocks_peli TO U3 ;	13) U3: GRANT UPDATE ON Stocks_peli TO U4;
7) U2: GRANT UPDATE ON Stocks_peli TO U4 WITH GRANT OPTION;	14) U2: UPDATE Peli_en_video ...;
	15) U4: UPDATE Stocks_peli ... ;

- 5) El usuario A ha creado la tabla Usuario(#U, nombre, tarea), y luego ejecuta los siguientes comandos SQL:

```
GRANT INSERT ON Usuario TO B WITH GRANT OPTION;
GRANT SELECT ON Usuario TO B WITH GRANT OPTION;
GRANT SELECT ON Usuario TO C;
CREATE ROLE GrupoUsers;
GRANT GrupoUsers TO E, F;
GRANT DELETE ON Usuario TO GrupoUsers;
```

- a) Indique quienes pueden ejecutar exitosamente los siguientes comandos:

1. SELECT * FROM Usuario WHERE #U='C';
2. INSERT INTO Usuario VALUES ('C','Gerente');
3. GRANT SELECT ON Usuario TO D;
4. REVOKE GrupoUsers FROM F;

- b) Establezca la habilitación (o no) para realizar los siguientes comandos:

```
B: GRANT INSERT ON Usuario TO D;
A: REVOKE INSERT ON Usuario FROM B CASCADE;
A: REVOKE GrupoUsers FROM E;
E: DELETE FROM Usuario;
F: DELETE FROM Usuario;
```

- c) Cuáles permisos conservan los usuarios después de lo anterior?



LABORATORIO Nº 2

POSTGRES, PL/PGSQL, INTEGRIDAD Y SEGURIDAD DE LOS DATOS

Pautas: Se deberá entregar un informe con la resolución de los ejercicios

Sobre la base de datos tp1-Aviones

- 1) Crear la tabla "audit" con los siguientes atributos
 - Tabla (investigar si se puede crear una restriccion para que solamente permita en este campo un nombre de tabla que exista en la base de datos)
 - Tipo Operacion (crear una restriccion para que solamente permita el siguiente domino 'SELECT', 'INSERT', 'DELETE', 'UPDATE')
 - Usuario
 - Fecha/Hora
- 2) Crear los siguientes TRIGGERS
 - para INSERCIÓN sobre Tabla avion que inserte en la tabla audit la operacion realizada
 - para ELIMINACIÓN sobre Tabla avion que inserte en la tabla audit , la operación realizada
- 3) Crear dos usuarios en la base de datos TP1-Aviones, denominados userA y userB
- 4) Investigar y documentar la gramática y diferentes modificadores de las sentencias GRANT y REVOKE en Postgres
- 5) Otorgar los siguientes permisos:
 - de seleccion al userA sobre todas las tablas y permisos de INSERT sobre la tabla audit con la opción WITH GRANT OPTION
 - Otorgar permisos de seleccion, insert y update al userB sobre las siguientes tablas: avion, piloto, pilotoAvion
- 6) Conectarse con el usuario userB a la base de datos y hacer una insercion sobre la tabla avion, documentar lo que arroja el motor. Si no fue posible realizar la operación, solucionar el inconveniente conectándose con el usuario userA