

## Acciones Procedurales en Bases de Datos

### Triggers, funciones y stored procedures

#### Ejercicio 1

- a) Implemente mediante triggers en PostgreSQL los controles de los ejercicios 3a) y 3d) del [Práctico 2](#). Previamente complete una tabla como la siguiente con los eventos críticos a controlar en cada caso.

A modo de guía, dada la siguiente solución declarativa para el chequeo: *Cada proveedor sólo puede proveer productos a sucursales de su localidad*

```
create assertion chk_localidades check not exists (  
  select 1 from proveedor p  
  join provee pr on (pr.nro_prov = p.nro_prov)  
  join sucursal s on (s.cod_suc = pr.cod_suc)  
  where p.localidad <> s.localidad );
```

se deberían analizar cuáles son los eventos críticos que se requieren controlar y completar la grilla que se incluye a continuación con los datos necesarios para completar la cabecera de el/los trigger/s requerido/s:

Tabla / Evento	Insert	Update	Delete	Granularidad
PROVEEDOR	no / si	no / si (atributo/s)	no / si	f. e. row / statement
PROVEE	no / si	no / si (atributo/s)	no / si	f. e. row / statement
SUCURSAL	no / si	no / si (atributo/s)	no / si	f. e. row / statement

- b) Provea al menos 2 sentencias disparadoras para el/los trigger/s definidos (sobre distintas tablas y/o eventos) y chequee la respuesta del DBMS.

#### Ejercicio 2

Para las restricciones solicitadas en el ejercicio 4 del [Práctico 2](#), determine los eventos críticos (puede orientarse mediante la tabla anterior) e implemente con triggers aquellos chequeos que no sea posible incorporar declarativamente en PostgreSQL.

#### Ejercicio 3

Considere dos tablas EMPLEADO\_1 y EMPLEADO\_2 con atributos (id\_empleado, nombre, apellido, sueldo) y que contienen las siguientes tuplas (*Nota: se indican los valores de id\_empleado y sueldo respectivamente*) :

EMPLEADO\_1: <1,...,500> y EMPLEADO\_2: <2,...,700>, <3,...,300>, <4,...,700>

Suponga que se define el siguiente trigger en PostgreSQL:

```
CREATE OR REPLACE FUNCTION autoIncremento()  
RETURNS TRIGGER AS $$  
BEGIN  
  UPDATE empleado_1 SET  
    sueldo = sueldo + (SELECT min(sueldo)*0.05 FROM empleado_1)  
  RETURN;  
END; $$ LANGUAGE plpgsql;  
  
CREATE TRIGGER trigger_autoincremento  
BEFORE INSERT ON Empleado_1  
  [ granularidad ]  
EXECUTE FUNCTION autoIncremento();
```

y que luego, por medio de la sentencia **INSERT INTO EMPLEADO\_1 SELECT \* FROM EMPLEADO\_2**; se insertan en EMPLEADO\_1 las tuplas que contiene EMPLEADO\_2. Indique el estado final de la tabla EMPLEADO\_1 si: a) [ **granularidad** ] = for each row ; b) [ **granularidad** ] = for each statement.

## **Acciones Procedurales en Bases de Datos**

### **Triggers, funciones y stored procedures**

#### **Ejercicio 4**

En el esquema de Películas, considere que se quiere mantener un registro de quién y cuándo realizó actualizaciones sobre las entregas de películas. Cree una tabla HIS\_ENTREGA que tenga por lo menos las siguientes columnas: nro\_registro, fecha, operación, cant\_reg\_afectados, usuario.

- Provea el/los trigger/s necesario/s para mantener actualizada en forma automática la tabla HIS\_ENTREGA cuando se realizan actualizaciones (insert, update o delete) en la tabla ENTREGA o RENGLON\_ENTREGA
- Determine el resultado en las tablas si se ejecuta la operación:  
`DELETE FROM ENTREGA WHERE id_video = 3582 ;`  
según el o los triggers definidos sean FOR EACH ROW o FOR EACH STATEMENT, evalúe la diferencia a partir de ambos tipos de granularidad.

#### **Ejercicio 5**

En el esquema de Películas, se desea llevar otra tabla histórica en la cual conste, para cada EMPLEADO, el tiempo que ha permanecido en el sistema y el tiempo promedio en cada departamento.

Plantee los cambios necesarios en el esquema para contemplar ambos datos (mediante la/s sentencia/s SQL necesaria/s) y luego:

- Implemente este control mediante trigger/s
- Plantee un stored procedure que realice esta actualización
- ¿Ambos enfoques garantizan tener actualizada en todo momento la información?
- ¿Qué ocurriría con la información pre-existente en la base de datos al momento de incorporar el trigger o procedimiento?
- Analice si se podría implementar lo anterior mediante chequeos declarativos (de tabla o generales)

#### **Ejercicio 6**

Para el esquema Voluntarios, implemente un trigger que:

- ante un cambio de tarea de un voluntario, coloque en 0 la cantidad de horas aportadas, independientemente de si el valor de horas\_aportadas de la sentencia update es diferente.
- cuando se actualice la cantidad de horas aportadas, el nuevo valor no puede ser ni menor, ni superior al 10 % del valor que tenía anteriormente.

#### **Ejercicio 7**

Para el esquema del ejercicio 2 del Práctico 2, cree la siguiente tabla en la que se requiere llevar registro de la cantidad de textos publicados y la fecha de la última publicación, por cada autor:

TextosporAutor (autor, cant\_textos, fecha\_ultima\_public)

- Analice cuál sería el recurso procedural más apropiado para completar la información de esta tabla a partir de los datos ya existentes en la/s tabla/s, incluya la implementación completa y plantee un ejemplo de su utilización.
- Defina el/los trigger/s que estime necesarios para mantener la base consistente cuando se produzcan actualizaciones sobre Artículo.