**Primer Trabajo – Bases de Datos II**

Elizabeth Alzate Zapata

Cristian Mejía Martínez

**Nota:** El código de cada punto se encuentra en el archivo *trabajo.sql* separados por comentarios de SQL con un nombre específico*,* cada punto se encuentra numerado de la siguiente manera:

* **Punto 1 =** *Trigger de inserción 1*
* **Punto 2 =** *Trigger de inserción 2*
* **Punto 3 =** *Trigger de inserción 3*
* **Punto 4 =** *Trigger de actualización sobre cooperativa*
* **Punto 5 =** *Trigger de borrado sobre cooperativa*
* **Punto 6 =** *Trigger de borrado sobre socio*
* **Punto 7 =** *Programa 1 PL/SQL*
* **Punto 8 =** *Programa 2 PL/SQL*

**TRIGGERS**

**Punto 1. Trigger de Inserción en tabla cooperativa (3%)**

Este trigger se encarga de asignar el valor de 0 al atributo c\_acumulado en el registro a ingresar a la tabla cooperativa antes de realizar el ingreso de este, asegurándose así de que, sin importar el valor que se le asigne al atributo c\_acumulado a la hora de realizar una inserción en la tabla cooperativa, este siempre sea ingresado como 0.

**Punto 2. Trigger de Inserción en tabla socio (3%)**

Este trigger, similar al anterior, se encarga de asignar el valor de 0 en el atributo s\_acumulado antes de realizar una inserción a la tabla socio, garantizando así que, sin importar qué valor se le asigne a la hora de insertar un nuevo registro, este siempre será añadido como 0.

**Punto 3. Trigger de inserción en tabla coopexsocio (3%)**

Similar a los triggers anteriores, este se encarga de asignar el valor de 0 en el atributo sc\_acumulado antes de realizar una inserción a la tabla coopexsocio, garantizando así que, sin importar qué valor se le asigne a la hora de insertar un nuevo registro, este siempre será añadido como 0.

**Punto 4. Trigger de actualización en tabla cooperativa (25%)**

Este trigger distribuye el incremento añadido equitativamente por cada socio a la hora de actualizar la tabla cooperativa, definiendo el incremento como la resta entre el valor nuevo añadido en la actualización con el valor anterior (se permiten incrementos negativos), para luego actualizar las demás tablas con los valores correspondientes, en este caso, las tablas coopexsocio y socio deben actualizarse sumando el valor acumulado almacenado en cada registro de los socios vinculados a la cooperativa, añadiéndole el incremento determinado.

**Punto 5. Trigger de borrado sobre tabla cooperativa (15%)**

Se optó por realizar un trigger AFTER DELETE que, una vez eliminado el registro en la tabla cooperativa, consulte a los registros en la tabla coopexsocio que correspondan con la cooperativa recién eliminada mediante un cursor para iterar por cada uno de los socios vinculados a dicha cooperativa y actualizar el atributo s\_acumulado en la tabla socio para cada uno de los socios vinculados, eliminando el valor que les aportaba dicha cooperativa de su acumulado total. Por último, una vez se hayan actualizado todos los registros afectados de la tabla socio, se procede a eliminar todos los registros asociados a la cooperativa eliminada en la tabla coopexsocio.

**Punto 6. Trigger de borrado sobre tabla socio (10%)**

Se optó por realizar un trigger AFTER DELETE que, una vez eliminado el registro en la tabla socio, consulte todos los registros asociados al socio eliminado en la tabla coopexsocio y los elimine.

**PROGRAMAS**

**Punto 7. Primer programa (20%)**

Para resolver este punto se optó por realizar un procedimiento llamado consulta\_cooperativa con un parámetro de entrada llamado cod\_coope que corresponderá al código ingresado de la cooperativa a consultar, en éste procedimiento se obtienen los datos de la cooperativa solicitada en donde el código corresponda y se cuentan la cantidad de socios vinculados con la cooperativa en la tabla coopexsocio para mostrarlos por pantalla, luego, se itera por cada uno de los socios vinculados a la cooperativa a través de un cursor para mostrar su información y, finalmente, se suma el valor acumulado por todos los socios vinculados a la cooperativa para mostrarlo por pantalla.

**Punto 8. Segundo programa (21%)**

Para resolver este punto se realizó un procedimiento similar al anterior, en donde se tiene un parámetro de entrada correspondiente al id del socio a consultar, se consulta la información del socio y se cuenta el número de cooperativas en las que participa para mostrarlas por pantalla, luego, se itera por cada una de estas cooperativas a través de un cursor para mostrar su información. Una vez iteradas todas las cooperativas a las que pertenece el socio en cuestión, se itera a través de las cooperativas en las que no participa a través de un right join entre las cooperativas en las que participa el socio y la tabla de cooperativas, obteniendo únicamente aquellas que figuran en la tabla de cooperativas, pero no en las que participa el socio consultado, con el fin de mostrar su nombre.