

Base de Datos 2 - Obligatorio 2013

- El obligatorio será realizado en forma grupal.
- La entrega se hará de manera electrónica a danielgonzalezbernal@gmail.com para los alumnos del Buceo y a fer200417@gmail.com para los alumnos del LATU.
- Se deberá entregar un archivo .zip el cual debe contener los 3 archivos mencionados posteriormente con asunto Base de Datos 2 – Obligatorio 2.
En el mail poner nombre apellido y cédula de identidad de cada integrante.
- El plazo para la entrega es hasta el **Domingo 30/06/2013** hasta las 23:59 hs. Una vez culminado el plazo estipulado, **no** se aceptarán más obligatorios por parte de los docentes.

Video Club

Se pide:

PARTE 1. Uso de triggers

Objetivo: Experimentar en el uso de **triggers** como mecanismo de control del cumplimiento de restricciones de integridad.

1. Totalidades.

Un mecanismo adecuado para controlar el cumplimiento de las restricciones de totalidad existentes en el MER son los **triggers**. Estos permiten disparar la ejecución de procedimientos ante eventos sobre tablas (por ejemplo inserciones o borrados). Combinando estos procedimientos con el uso de **tablas de control** podemos detectar y almacenar aquellas tuplas que violan las restricciones de totalidad planteadas. Tomemos por ejemplo la restricción de totalidad existente en la relación idiomas_de_pelicula entre *peliculas* e *idiomas*, la cual exige que para toda película se conozca al menos un idioma. Una estrategia para detectar aquellas tuplas que no cumplen la restricción es la siguiente:

i) Definir un trigger que se ejecutará cada vez que se **inserte** una tupla en la tabla *peliculas*. Este trigger registrará en una tabla de control que la inserción se realizó.

ii) Definir un trigger que se ejecutará cada vez que se **inserte** una tupla en la tabla *idiomas_de_peliculas*. Este trigger borrará la tupla correspondiente a la película de la tabla de control.

De esta forma podemos asegurar que la tabla de control sólo contiene aquellas películas que violan la restricción de totalidad

A continuación especificamos en detalle los triggers requeridos para resolver el problema.

Nombre Trigger: tot_peliculas

Se ejecuta Cuando se inserta una tupla en peliculas

Acción Inserta una tupla en la tabla: **inconsist_tot_pel** (id_alerta,id_pelicula)

Nombre Trigger: tot_peliculas_borrar

Se ejecuta Cuando se inserta una tupla en idiomas_de_peliculas

Acción Borra la tupla correspondiente de la tabla: **inconsist_tot_pel**(id_alerta,id_pelicula)

Importante: - El atributo **id_alerta** debe ser de tipo **SERIAL**.

- El atributo **id_pelicula** debe ser del mismo tipo que su correspondiente en la tabla *idiomas_de_peliculas* (integer)

- Utilizar los nombre de los triggers mencionados anteriormente.

Se debe entregar: un archivo llamado parte1_1.sql que contenga la creación de las tablas y la creación de los triggers necesarios para esta parte

Los **triggers** también pueden ser usados para asegurar el cumplimiento de restricciones no estructurales (RNE), por ejemplo impidiendo la inserción de tuplas que las violen. Teniendo en cuenta la restricción podemos imaginarnos que el procedimiento de creación de una sucursal y asignación de su encargado podría ser el siguiente:

- Para controlar que el empleado que se indica como encargado efectivamente trabaje en dicha sucursal se puede implementar un trigger como se indica a continuación:

- Utilizar el nombre del trigger dado anteriormente.

4 - Descontar todos los pagos que el cliente haya realizado ANTES de la fecha de cálculo.

Nombre Función: `balance_clientes`

Parámetros: Entrada: fecha (timestamp)

Acción Recorre la tabla *clientes* y para cada uno de ellos invoca la función *balance_cliente*.

Por cada invocación inserta una tupla en la tabla *balances* con el cliente (`id_cliente`), la fecha de cálculo (fecha) y el saldo calculado.

Importante:

- La tabla **balance** debe tener los atributos `id_cliente`, `fecha_calculo`, `saldo`
- Utilizar los nombres de funciones dados anteriormente.

Se debe entregar: un archivo llamado `parte2.sql` que contenga la creación de las tablas y la creación de los procedimientos almacenados necesarios para resolver esta parte.

¹ Recordar que para cada película se conoce el `costo_alquiler` que corresponde a un plazo igual a `duracion_alquiler`. Si la película no fue devuelta transcurrido ese plazo (en días) se considera que está atrasado.