

Práctica de Triggers II

1. Se pide: Crear un trigger que valide que ante un insert de una o más filas en la tabla `items`, realice la siguiente validación:
 - Si la orden de compra a la que pertenecen los ítems ingresados corresponde a clientes del estado de California, se deberá validar que estas órdenes puedan tener como máximo 5 registros en la tabla `item`.
 - Si se insertan más ítems de los definidos, el resto de los ítems se deberán insertar en la tabla **`items_error`** la cual contiene la misma estructura que la tabla `items` más un atributo fecha que deberá contener la fecha del día en que se trató de insertar.

Ej. Si la Orden de Compra tiene 3 ítems y se realiza un insert masivo de 3 ítems más, el trigger deberá insertar los 2 primeros en la tabla `items` y el restante en la tabla `items_error`.

Supuesto: En el caso de un insert masivo los items son de la misma orden.

2. Triggers Dada la siguiente vista

```
CREATE VIEW ProdPorFabricante AS
    SELECT m.manu_code, m.manu_name, COUNT(*)
    FROM manufact m INNER JOIN products p
    ON (m.manu_code = p.manu_code)
    GROUP BY manu_code, manu_name;
```

Crear un trigger que permita ante un insert en la vista `ProdPorFabricante` insertar una fila en la tabla `manufact`.

Observaciones: el atributo `leadtime` deberá insertarse con un valor default 10
El trigger deberá contemplar inserts de varias filas, por ej. ante un `INSERT / SELECT`.

3. Crear un trigger que ante un `INSERT` o `UPDATE` de una o más filas de la tabla `Customer`, realice la siguiente validación.
 - La cuota de clientes correspondientes al estado de California es de 20, si se supera dicha cuota se deberán grabar el resto de los clientes en la tabla `customer_pend`.
 - Validar que si de los clientes a modificar se modifica el Estado, no se puede superar dicha cuota.

Si por ejemplo el estado de CA cuenta con 18 clientes y se realiza un update o insert masivo de 5 clientes con estado de CA, el trigger deberá modificar los 2 primeros en la tabla `customer` y los restantes grabarlos en la tabla `customer_pend`.

La tabla `customer_pend` tendrá la misma estructura que la tabla `customer` con un atributo adicional `fechaHora` que deberá actualizarse con la fecha y hora del día.

4. Dada la siguiente vista

```
CREATE VIEW ProdPorFabricanteDet AS
    SELECT m.manu_code, m.manu_name, pt.stock_num, pt.description
    FROM manufact m LEFT OUTER JOIN products p ON m.manu_code = p.manu_code
    LEFT OUTER JOIN product_types pt ON p.stock_num = pt.stock_num;
```

Se pide: Crear un trigger que permita ante un `DELETE` en la vista `ProdPorFabricante` borrar los datos en la tabla `manufact` pero sólo de los fabricantes cuyo campo `description` sea NULO (o sea que no tienen stock).

Observaciones: El trigger deberá contemplar borrado de varias filas mediante un `DELETE` masivo. En ese caso sólo borrará de la tabla los fabricantes que no tengan productos en stock, borrando los demás.

5. Se pide crear un trigger que permita ante un delete de una sola fila en la vista ordenesPendientes valide:
- Si el cliente asociado a la orden tiene sólo esa orden pendiente de pago (paid_date IS NULL), no permita realizar la Baja, informando el error.
 - Si la Orden tiene más de un ítem asociado, no permitir realizar la Baja, informando el error.
 - Ante cualquier otra condición borrar la Orden con sus ítems asociados, respetando la integridad referencial.

Estructura de la vista: customer_num, fname, lname, Company, order_num, order_date
WHERE paid_date IS NULL.