Taller Triggers y Vistas

Camilo Esteban Paez Neuta cpaezn@unal.edu.co

1. Creacion de base de datos

```
create table Productos (
 nombre varchar(30) not null,
  precio float(2) not null,
  stock int default 0,
  producto_id char(36),
 primary key (producto_id)
);

create table Ventas (
 producto_id char(36) not null,
  cantidad int not null,
  fecha datetime default current_timestamp,
  venta_id int auto_increment,
  primary key(venta_id),
  foreign key (producto_id) references Productos(producto_id)
);

create table Auditoria (
 tipo enum('insert', 'update', 'delete') not null,
  fecha datetime default current_timestamp,
  info varchar(100) not null,
  auditoria_id int auto_increment,
  primary key (auditoria_id)
);
```

2. Insercion de datos

```
// Productos

INSERT INTO Productos (nombre, precio, stock, producto_id)
VALUES ('Laptop', 1200.50, 50, 'd3b07384d113edec49eaa6238ad5ff00');

INSERT INTO Productos (nombre, precio, stock, producto_id)
VALUES ('Smartphone', 799.99, 100, '9a3c73a4207e7f8d5fba9e6c1d2345a1');

INSERT INTO Productos (nombre, precio, stock, producto_id)
VALUES ('Auriculares', 150.00, 200, 'e4c204c9f803d87ac6e0ae5a6543b2e9');

INSERT INTO Productos (nombre, precio, stock, producto_id)
VALUES ('Teclado Mecánico', 99.95, 150, '15a75837223fb375842b156fa04b163b');

INSERT INTO Productos (nombre, precio, stock, producto_id)
```

```
VALUES ('Monitor 4K', 500.00, 25, '37289f52b1c67e0e9b773aff556ca1e4');

// Ventas

INSERT INTO Ventas (producto_id, cantidad)
VALUES ('d3b07384d113edec49eaa6238ad5ff00', 2);

INSERT INTO Ventas (producto_id, cantidad)
VALUES ('9a3c73a4207e7f8d5fba9e6c1d2345a1', 1);

INSERT INTO Ventas (producto_id, cantidad)
VALUES ('e4c204c9f803d87ac6e0ae5a6543b2e9', 5);

INSERT INTO Ventas (producto_id, cantidad)
VALUES ('15a75837223fb375842b156fa04b163b', 3);

INSERT INTO Ventas (producto_id, cantidad)
VALUES ('37289f52b1c67e0e9b773aff556ca1e4', 1);
```

3. Funcion calular_total

La palabra clave `deterministic` señala que la funcion es pura, siempre que se ejecute con las misma variables su resultado sera siempre igual

```
create function calcular_total(precio float(2), cantidad int)
returns FLOAT(2) deterministic
begin
  return precio * cantidad;
end;
```

4. Procedimiento almacenado registrar_venta

Las palabras claves `start transaction` permite realizar multiples consultas en un mismo cuerpo

```
create procedure registrar_venta(
    IN in_producto_id CHAR(36),
    IN in_cantidad INT)

 begin
    start transaction;
    insert into Ventas (producto_id, cantidad)
    values (in_producto_id, in_cantidad);

    update Productos
    set stock = stock - in_cantidad
    where producto_id = in_producto_id;

    commit;
end;
```

5. Trigger after_venta

```
create trigger after_venta
after insert on Ventas
for each row
begin
    insert into Auditoria (tipo, info)
    values ('insert',
        CONCAT('Venta ID ', NEW.venta_id,
            ', Producto ', NEW.producto_id,
            ', Cantidad ', NEW.cantidad));
end;
```

6. Vista resumen_ventas

```
create view resumen_ventas as
select
    p.nombre as producto,
    v.cantidad,
    calcular_total(p.precio, v.cantidad) as total
from Ventas v
join Productos p on v.producto_id = p.producto_id;
```

7. Trigger validar_stock

`signal sqlstate '45000'` lanza un error que cancela la insercion

```
create trigger validar_stock
before insert on Ventas
for each row
begin
    declare current_stock INT;

    select stock into current_stock
    from Productos
    where producto_id = NEW.producto_id;

    if current_stock < NEW.cantidad then
        signal sqlstate '45000'
        set message_text = 'Stock insuficiente para realizar la venta';
    end if;
end;
```

Preguntas

- ¿Que sucede si intentas vender mas productos de los que hay en stock? R// Por la implementacion de la funcion `validar_stock` se enviara un mensaje 'Stock insuficiente para realizar la venta' y se lanzara una señal de error 45000 que no permitira que la operacion se realize protegiendo la base de datos
- Como modificarias la funcion `calcular_total` para incluir un descuento del 10% en productos cuyo precio sea mayor a 500? R//

```
create function calcular_total(precio float(2), cantidad int)
returns FLOAT(2) deterministic
begin
  declare total FLOAT(2);
  set total = precio * cantidad;

  if total > 500 then
    return total * 0.9;
  else
    return  total
end;
```

- Que otras validaciones podrias agregar al trigger `validar_stock`? R// Las cantidad en la tabla venta es un entero firmado, por lo que podria llegar a ser ser negativo si alguien manipula el valor, esto se corrige dejandolo sin firmar desde la definicion de la tabla, o en este caso agregandola al trigger quedaria asi:

```
create trigger validar_stock
before insert on Ventas
for each row
begin
    declare current_stock INT;

    select stock into current_stock
    from Productos
    where producto_id = NEW.producto_id;

    if NEW.cantidad <= 0 then
        signal sqlstate '45000'
        set message_text = 'Cantidad no puede ser negativa o 0';
    if current_stock < NEW.cantidad then
        signal sqlstate '45000'
        set message_text = 'Stock insuficiente para realizar la venta';
    end if;
end;
```