

DANIEL RODRÍGUEZ FERNÁNDEZ 1DAM

Cargamos el script almacén el cual contiene las siguientes tablas:

```
mysql> source C:\Users\danie\Desktop\Almacen.sql
Query OK, 1 row affected (0.06 sec)

Database changed
Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.08 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.05 sec)

Query OK, 8 rows affected (0.01 sec)
```

- **InventarioMov:** contendrá información de movimientos de inventarios, tanto entradas como salidas. Las entradas contendrán valor 1 en el campo E/S y 0 en caso contrario.
- **AuditoriaInventarios:** Esta tabla guardara toda la información de los **UPDATE** y **DELETE** que se realicen sobre nuestra tabla de movimientos de inventario.
- **Stock:** Esta tabla contendrá los stocks de cada producto o artículo por almacén o ubicación física.

A continuación, crea los siguientes **TRIGGERS**:

1. **TRIGGER: Stock_Insert:** Servirá para actualizar el stock al momento en que se inserta un nuevo movimiento de inventario. También verificara si es un producto nuevo, registrándolo en nuestra tabla de stock para poder llevar el control de sus movimientos.

```
1 delimiter $$
2 drop trigger if exists stock_insert; $$
3 create trigger stock_insert before insert on inventariomov for each row
4 BEGIN
5     declare CONTADOR int;
6     select count(*) into CONTADOR from stock where stock.CodProd = new.CodProd;
7     if CONTADOR = 0
8     then
9         insert into stock values (new.Almacen, new.CodProd, new.cantidad);
10    else
11        if new.inout = 'E'
12        then
13            update stock set StockActual = StockActual + new.Cantidad where CodProd = new.CodProd;
14        elseif new.'INOUT' = 'S'
15        THEN
16            update stock set StockActual = StockActual - new.Cantidad where CodProd = new.CodProd;
17        end if;
18    end if;
19 end; $$
20
```

stock_insert

Services

almacen@localhost

console 40 ms

Output

ID_MOV	Almacen	INOUT	CodProd	Cantidad
10	11	1 E	A0010	12
11	12	1 E	A0010	16

```

1  if CONTADOR = 0
2  then
3      insert into stock values (new.Almacen, new.CodProd, new.cantidad);
4  else
5      if new.inout = 'E'
6      then
7          update stock set StockActual = StockActual + new.Cantidad where CodProd = new.CodProd;
8      elseif new.`INOUT` = 'S'
9      THEN
10         update stock set StockActual = StockActual - new.Cantidad where CodProd = new.CodProd;
11     end if;
12 end if;
13 end; $$
14
15 insert into inventariomov values(NULL, 1, 'E', 'ASS02', 12.00);
16 select * from inventariomov;
17 select * from stock;
18
19 stock_insert

```

Services

almacen@localhost

console 111 ms

Output

ID_HOV	Almacen	INOUT	CodProd	Cantidad
12	1	E	30802	16
13	1	S	SS02	16
14	1	E	ASS02	16
15	1	E	ASS02	12

```

7  if CONTADOR = 0
8  then
9      insert into stock values (new.Almacen, new.CodProd, new.cantidad);
10 else
11     if new.inout = 'E'
12     then
13         update stock set StockActual = StockActual + new.Cantidad where CodProd = new.CodProd;
14     elseif new.`INOUT` = 'S'
15     THEN
16         update stock set StockActual = StockActual - new.Cantidad where CodProd = new.CodProd;
17     end if;
18 end if;
19 end; $$
20
21 insert into inventariomov values(NULL, 1, 'E', 'ASS02', 12.00);
22 select * from inventariomov;
23 select * from stock;
24
25 stock_insert

```

Services

almacen@localhost

console 111 ms

Output

Almacen	CodProd	StockActual
14 1	ASS02	16
15 1	SS02	16

delimiter \$\$

drop trigger if exists stock_insert; \$\$

create trigger stock_insert before insert on inventariomov for each row
BEGIN

```

    declare CONTADOR int;
    select count(*) into CONTADOR from stock where stock.CodProd =
new.CodProd;
    if CONTADOR = 0
    then
        insert into stock values (new.Almacen, new.CodProd,
new.cantidad);
    else
        if new.inout = 'E'
        then
            update stock set StockActual = StockActual + new.Cantidad
where CodProd = new.CodProd;
            elseif new.`INOUT` = 'S'
            THEN
                update stock set StockActual = StockActual - new.Cantidad
where CodProd = new.CodProd;
            end if;
        end if;
    end; $$
insert into inventariomov values(NULL, 1, 'E', 'ASS02', 12.00);
select * from inventariomov;
select * from stock;

```

2. **TRIGGER: Stock_Update:** Servirá para actualizar el stock al momento en que se Modifica un movimiento de inventario. Registrará también en nuestra tabla de auditorías, la cantidad que estuvo antes de ser modificado, como también la nueva cantidad, junto con el usuario que hizo la modificación. También registrara la fecha y hora de la operación.

Primero añadimos la columna usuario a la tabla auditoria

```
1 ALTER TABLE auditoriainventarios ADD COLUMN usuari VARCHAR(15);
```

```
1 delimiter $$
2 drop trigger if exists stock_update;$$
3 create trigger stock_update after update on inventariomov for each row
4 BEGIN
5     declare cantid int;
6     IF old.cantidad < new.cantidad then
7         set cantid = new.cantidad - old.cantidad;
8         IF old.INOUT='E' then
9             update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
10        ELSE
11            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
12        END IF;
13    ELSE
14        set cantid = old.cantidad - new.cantidad;
15        IF old.INOUT='E' then
16            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
17        ELSE
18            update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
19        END IF;
20    END IF;
21    insert into auditoriainventarios values(null, New.Almacen, New.INOUT, New.CodProd, Old.Cantidad, New.Cantidad, now(), 'Actualizacion', SESSION_USER());
22 end;$$
23 update inventariomov set cantidad = 4 where ID_MOV = 23;
24 select * from inventariomov;
25 select * from stock;
26 select * from auditoriainventarios;
```

The screenshot shows a database IDE with the trigger code from the previous block. Below the code editor, there is a table view for the 'almacen.stock' table. The table has three columns: 'Almacen', 'CodProd', and 'StockActual'. The data is as follows:

Almacen	CodProd	StockActual
14	1	ASS02
15	1	SS02
16	2	NIKE
17	6	NJORDAN

```

1 delimiter $$
2 drop trigger if exists stock_update;$$
3 create trigger stock_update after update on inventariomov for each row
4 BEGIN
5     declare cantid int;
6     IF old.cantidad<new.cantidad then
7         set cantid = new.cantidad - old.cantidad;
8         IF old.INOUT='E' then
9             update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
10        ELSE
11            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
12        END IF;
13    ELSE
14        set cantid = old.cantidad - new.cantidad;
15        IF old.INOUT='E' then
16            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
17        ELSE
18            update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
19        END IF;
20    END IF;
21 END;
22 stock_update

```

Services

Output

ID_MOV	Almacen	INOUT	CodProd	Cantidad
11	12	1 E	AO010	16
12	23	2 E	NIKE	4

```

1 delimiter $$
2 drop trigger if exists stock_update;$$
3 create trigger stock_update after update on inventariomov for each row
4 BEGIN
5     declare cantid int;
6     IF old.cantidad<new.cantidad then
7         set cantid = new.cantidad - old.cantidad;
8         IF old.INOUT='E' then
9             update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
10        ELSE
11            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
12        END IF;
13    ELSE
14        set cantid = old.cantidad - new.cantidad;
15        IF old.INOUT='E' then
16            update stock set StockActual= StockActual - cantid where stock.codProd = New.codProd;
17        ELSE
18            update stock set StockActual= StockActual + cantid where stock.codProd = New.codProd;
19        END IF;
20    END IF;
21 END;
22 stock_update

```

Services

Output

N_Auditoria	Almacen	INOUT	CodProd	Cantidad_Anterior	Cantidad_Nueva	Fecha_Hora	Accion	usuario
37	38 2	E	NIKE	2	3	2022-05-06 10:57:01	Actualizacion	root@localhost
38	39 2	E	NIKE	3	4	2022-05-06 11:09:43	Actualizacion	root@localhost

```

delimiter $$
drop trigger if exists stock_update;$$
create trigger stock_update after update on inventariomov for each row
BEGIN
    declare cantid int;
    IF old.cantidad<new.cantidad then
        set cantid = new.cantidad - old.cantidad;
        IF old.INOUT='E' then
            update stock set StockActual= StockActual + cantid where stock.codProd =
New.codProd;
        ELSE
            update stock set StockActual= StockActual - cantid where stock.codProd =
New.codProd;
        END IF;
    ELSE
        set cantid = old.cantidad - new.cantidad;
        IF old.INOUT='E' then
            update stock set StockActual= StockActual - cantid where stock.codProd =
New.codProd;
        ELSE

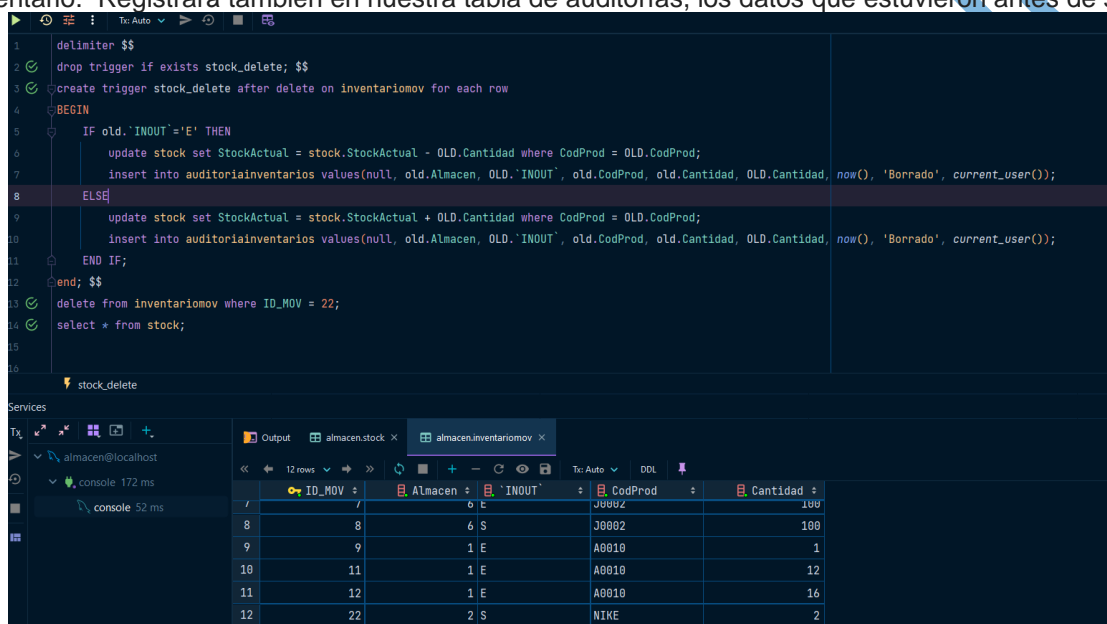
```

```

        update stock set StockActual= StockActual + cantid where stock.codProd =
New.codProd;
    END IF;
END IF;
insert                                into                                auditoriainventarios
values(null,New.Almacen,New.INOUT,NEW.CodProd,Old.Cantidad,New.Cantidad,now(),'Actua
lizacion',SESSION_USER());
end;$$
update inventariomov set cantidad = 4 where ID_MOV = 23;
select * from inventariomov;
select * from stock;
select * from auditoriainventarios;

```

14. **TRIGGER: Stock_Delete:** Servirá para actualizar el stock al momento en que se elimina un movimiento de inventario. Registrará también en nuestra tabla de auditorías, los datos que estuvieron antes de ser eliminado.



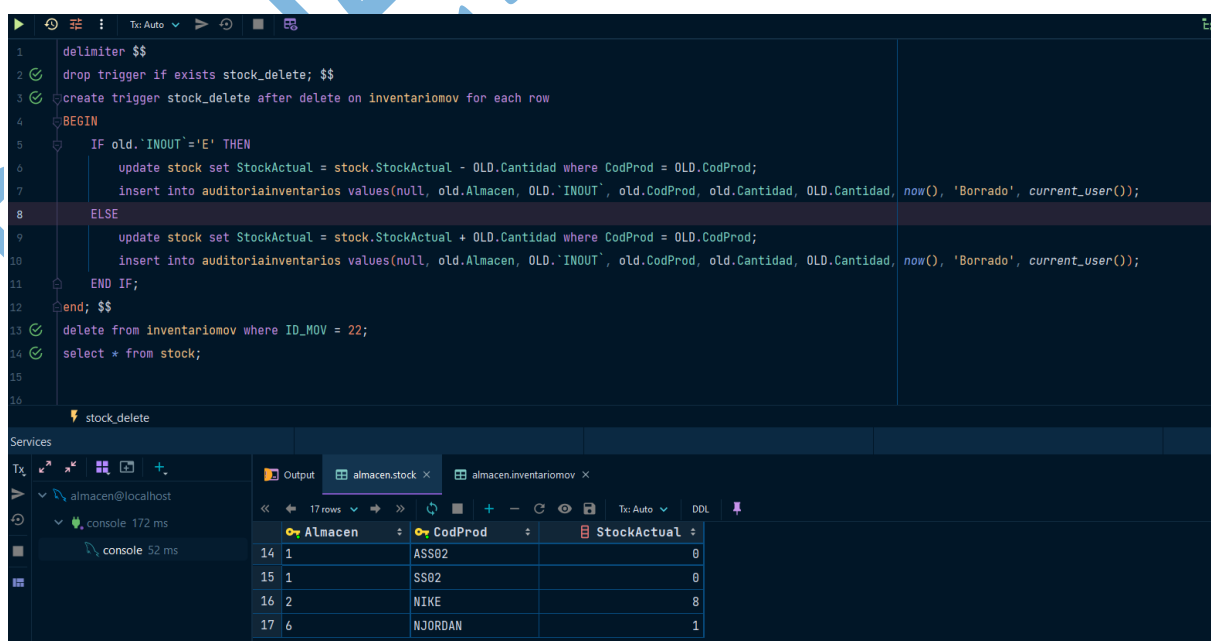
```

1 delimiter $$
2 drop trigger if exists stock_delete; $$
3 create trigger stock_delete after delete on inventariomov for each row
4 BEGIN
5     IF old.'INOUT'='E' THEN
6         update stock set StockActual = stock.StockActual - OLD.Cantidad where CodProd = OLD.CodProd;
7         insert into auditoriainventarios values(null, old.Almacen, OLD.'INOUT', old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
8     ELSE
9         update stock set StockActual = stock.StockActual + OLD.Cantidad where CodProd = OLD.CodProd;
10        insert into auditoriainventarios values(null, old.Almacen, OLD.'INOUT', old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
11    END IF;
12 end; $$
13 delete from inventariomov where ID_MOV = 22;
14 select * from stock;
15

```

stock_delete

ID_MOV	Almacen	INOUT	CodProd	Cantidad
7	6	E	J0002	100
8	8	S	J0002	100
9	9	E	A0010	1
10	11	E	A0010	12
11	12	E	A0010	16
12	22	S	NIKE	2



```

1 delimiter $$
2 drop trigger if exists stock_delete; $$
3 create trigger stock_delete after delete on inventariomov for each row
4 BEGIN
5     IF old.'INOUT'='E' THEN
6         update stock set StockActual = stock.StockActual - OLD.Cantidad where CodProd = OLD.CodProd;
7         insert into auditoriainventarios values(null, old.Almacen, OLD.'INOUT', old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
8     ELSE
9         update stock set StockActual = stock.StockActual + OLD.Cantidad where CodProd = OLD.CodProd;
10        insert into auditoriainventarios values(null, old.Almacen, OLD.'INOUT', old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
11    END IF;
12 end; $$
13 delete from inventariomov where ID_MOV = 22;
14 select * from stock;
15

```

stock_delete

Almacen	CodProd	StockActual
14 1	ASS02	0
15 1	SS02	0
16 2	NIKE	8
17 6	NJORDAN	1

```

1 delimiter $$
2 drop trigger if exists stock_delete; $$
3 create trigger stock_delete after delete on inventariomov for each row
4 BEGIN
5     IF old.`INOUT`='E' THEN
6         update stock set StockActual = stock.StockActual - OLD.Cantidad where CodProd = OLD.CodProd;
7         insert into auditoriainventarios values(null, old.Almacen, OLD.`INOUT`, old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
8     ELSE
9         update stock set StockActual = stock.StockActual + OLD.Cantidad where CodProd = OLD.CodProd;
10        insert into auditoriainventarios values(null, old.Almacen, OLD.`INOUT`, old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado', current_user());
11    END IF;
12 end; $$
13 delete from inventariomov where ID_MOV = 22;
14 select * from stock;
15 select * from auditoriainventarios;

```

N_auditoria	Almacen	INOUT	CodProd	Cantidad_Anterior	Cantidad_Nueva	Fecha_Hora	Accion	usuari
31	32 2	E	NIKE	12	5	2022-05-04 11:56:24	Actualizacion	root@localhost
32	33 2	E	NIKE	5	5	2022-05-04 11:59:02	Borrado	root@localhost
33	34 3	E	NIKE	2	2	2022-05-04 22:33:19	Borrado	root@localhost
34	35 3	S	NIKE	2	2	2022-05-04 22:34:44	Borrado	root@localhost
35	36 2	S	NIKE	2	2	2022-05-04 22:47:53	Borrado	root@localhost

```

delimiter $$
drop trigger if exists stock_delete; $$
create trigger stock_delete after delete on inventariomov for each row
BEGIN
    IF old.`INOUT`='E' THEN
        update stock set StockActual = stock.StockActual - OLD.Cantidad
where CodProd = OLD.CodProd;
        insert into auditoriainventarios values(null, old.Almacen,
OLD.`INOUT`, old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado',
current_user());
    ELSE
        update stock set StockActual = stock.StockActual + OLD.Cantidad
where CodProd = OLD.CodProd;
        insert into auditoriainventarios values(null, old.Almacen,
OLD.`INOUT`, old.CodProd, old.Cantidad, OLD.Cantidad, now(), 'Borrado',
current_user());
    END IF;
end; $$
delete from inventariomov where ID_MOV = 22;
select * from inventariomov;
select * from stock;
select * from auditoriainventarios;

```