

[Página Principal](#) / [Mis cursos](#) / [Base de Datos](#) / [UNIDAD 4](#)
/ [Conceptos avanzados del SQL, Seguridad y Backups](#)

Conceptos avanzados del SQL, Seguridad y Backups

Evento BEFORE o AFTER



Si deseamos **crear un trigger** que se dispare después de un insert en la tabla productos, hay que hacerlo de esta manera:

```
</> CREATE TRIGGER after_productos_insert AFTER INSERT ON productos
```

Recordá que debés establecer el **DELIMITER** antes y después del código del trigger al igual que con los procedimientos.

```
</> SQL
```

```
DELIMITER //  
CREATE TRIGGER after_productos_insert AFTER INSERT ON pedidos_productos  
FOR EACH ROW  
BEGIN  
    UPDATE productos  
    SET stock=stock - NEW.cant,  
        Fechamod=CURDATE()  
    WHERE codigo=NEW.codproducto;  
END //  
  
DELIMITER ;
```

NEW y OLD representan el registro de la tabla que es objeto del trigger

NEW es el registro que está siendo insertado y **OLD** es el viejo registro.

Se utiliza **OLD** cuando se está haciendo un *update* para representar la fila que es modificada (la vieja fila) y **NEW** para representar los nuevos valores que están siendo utilizados para realizar la modificación.

Por lo tanto podés ver que en este trigger, lo que estamos haciendo con los datos que se han insertado en la tabla pedidos_productos, es tomar la cantidad y restársela al stock de la tabla productos para el producto específico del renglón que se está tratando.

De esta manera, cada vez que se ingrese un renglón de pedido, se actualizará el stock del producto correspondiente.

Ahora bien, seguramente te preguntarás qué sucede si el pedido tiene muchos renglones



¿Cómo funciona en esos casos?

Hacé clic en el botón para ver la respuesta.

Pues bien, lo que sucede es que cuando se haga el insert de cada renglón de pedido (o sea un registro en la tabla pedidos_productos) se estará haciendo además de ese insert, un update de la tabla productos para el producto afectado por la fila. Es decir, un insert se ha transformado en realidad en dos operaciones: un insert y un update.

Entonces, podés darte cuenta que no es muy bueno que usemos tantos triggers, porque pueden afectar el comportamiento de las aplicaciones y éstas se pueden hacer más lentas debido a tantas operaciones en la base de datos.

Ésta es precisamente una de las razones por las cuales no se recomienda usar mucho los triggers.

Una **aplicación común de los triggers** es para auditar cambios en las tablas, o la actividad de los usuarios o, por ejemplo, para casos de sensibilidad de datos (o sea datos que son muy importantes para las empresas), en los que se quiera saber quienes fueron los usuarios que borraron los datos de una tabla, específicamente.

Para hacer eso, lo que se necesita primero es una tabla donde ir guardando los datos que se auditarán. La tabla de auditoría generalmente tiene la misma estructura de la tabla original pero con el agregado de las siguientes columnas:



```
FechaMov  
TipoMov  
Usuario
```

Para obtener el usuario, se utiliza la función **USER()**

Los demás datos provendrán de las tablas NEW, OLD o del código SQL que pongamos en el trigger.

[< Anterior](#)

[Siguiente >](#)



Buenos Aires Ciudad

Dirección General de Gestión Digital
Ministerio de Modernización

[Español - Internacional \(es\)](#)

[English \(en\)](#)

[Español - Internacional \(es\)](#)