

# Por qué no uso Triggers ni Procedures en SQL

## Triggers/Procedures versus Observers

Llevo más de 30 años trabajando con SQL en sistemas UNIX, dando soporte a miles de clientes y actuando como bombero IT. Sí, bombero, porque apago fuegos técnicos que surgen como resultado de malas prácticas, o, lo que es peor, de seguir utilizando reliquias del pasado sin actualizarse.

Una de esas reliquias son los **TRIGGERS** y las **PROCEDURES**.

Qué bonitas eran en su día, qué problemáticas eran y qué problemáticas siguen siendo.

La cuestión es que un motor de bases de datos es lo que es, y en mi opinión, introducir lógicas de programación dentro del motor de bases de datos, a día de hoy, está obsoleto y es peligroso (siempre lo fue).

- Un cambio de versión del motor **SGBD**.
- Un cambio de **SGBD**.
- Un problema que obliga a restaurar desde un backup y este es incompleto por error del comando usado en la copia de seguridad.

Y seguramente habrá más.

Lo cierto es que todo eso incumple los principios SOLID, al desplazar en un nivel superior, ya no en la aplicación, sino en el motor de datos, unas lógicas que son **responsabilidad** del programa. Al menos así lo veo en el año 2024.

Por ejemplo, podemos utilizar **Observers** a nivel de aplicación.

- Fáciles de programar.
- Dentro de la app.
- Cumplen el objetivo de ser **sólidos** en nuestra aplicación, y dotan de la responsabilidad al **modelo** sobre el que se ejecutarían esos triggers.
- Fácilmente testeables (o los tests, esa cosa que se sigue odiando en muchos lugares, quizás por la falta de... visión o ganas de actualizarse).

# Anécdota

En el caso que me ocupa, un Magento 1.9, obsoleto, corriendo en un MariaDB 10.4, PHP 5.4, etc., y que está en producción con decenas de ventas al día.

No se puede trabajar en producción, y mucho menos si la aplicación se llama **Magento**, paradigma de los problemas en el comercio electrónico.

Así que cuando me enfrenté a ello, y pese a ser root, estar acostumbrado a lidiar con MySQL (y sus forks), me entró la pena negra al ver que había triggers heredados de un intento de migración con la herramienta oficial de Magento para tratar de pasar a Magento 2.

No era la primera vez. Alguna vez lo había visto en un restore de urgencia de otro cliente.

Algo que te rompe cuando estás con bases de datos de 5 a 10Gb, bajo presión, en producción,...

```
ERROR 1449 (HY000) at line 924: The user specified as a definer ('migration'@'99.99.99.99') does not exist
```

Bueno, como ya sabía de qué iba al ver la palabra **migration** y el contexto **Magento**, sabía que había más de un trigger.

## Procesos de reparación

### Obtener todos los triggers

```
mysql> SHOW TRIGGERS;
Trigger Event Table Statement Timing Created sql_mode Definer
character_set_client collation_connection Database Collation
trg_catalog_category_entity_after_insert INSERT catalog_category_entity BEGIN\nINSERT
INTO m2_cl_catalog_category_entity (`entity_id`, `operation`) VALUES (NEW.entity_id,
'INSERT')ON DUPLICATE KEY UPDATE operation = 'INSERT';\nEND AFTER NULL
migration@176.31.31.251 utf8 utf8_general_ci latin1_swedish_ci
...
...
```

Script para eliminar triggers Mejor un script, que llamaremos `drop_triggers.sh`.

```
# Reemplaza los valores de `usuario`, `password`, y `nombre_de_tu_base` con tus propios
valores.
MYSQL_USER="usuario"
MYSQL_PASS="password"
```

```
DATABASE_NAME="nombre_de_tu_base"
```

```
# Obtiene la lista de triggers y los elimina uno por uno
```

```
mysql -u"$MYSQL_USER" -p"$MYSQL_PASS" -D"$DATABASE_NAME" -e "SELECT TRIGGER_NAME FROM  
information_schema.TRIGGERS WHERE TRIGGER_SCHEMA = '$DATABASE_NAME';" | grep -v TRIGGER_NAME |  
while read TRIGGER_NAME; do  
    mysql -u"$MYSQL_USER" -p"$MYSQL_PASS" -D"$DATABASE_NAME" -e "DROP TRIGGER IF EXISTS  
\`$TRIGGER_NAME\`";  
done
```

👉 Por favor, en serio. Este es algo a tener en cuenta. Backups y backups, y de los que sabes que funcionarán. Yo no soy responsable de tus actos (ni nadie).

## Pruebas en docker

Bueno, si antes de hacer lóos, necesitas trabajar en un entorno de desarrollo, con docker, te dejo unos pasos, ya que el otro puede ser algo más completo.

### Obtener la lista

Puedes como alternativa obtener la lista de trigger de la base de datos, con [TablePlus](#) en formato CSV, y crear un fichero con la columna que define el nombre del trigger, por ejemplo `triggers.txt`.

## Docker

Obtenemos el nombre del container de mysql

```
$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS  
PORTS          NAMES  
b49e62fa5b29  web4-apache-httpd                  "httpd-foreground"                   19 hours ago  Up 19 hours  
0.0.0.0:8080->80/tcp    web4-apache-httpd-1  
bfd56c6f6c65  web4-php-fpm                        "docker-php-entrypoi..."           19 hours ago  Up 19 hours  
9000/tcp          web4-php-fpm-1  
d804a906babd  mariadb:10.1                        "docker-entrypoint.s..."           19 hours ago  Up 19 hours  
0.0.0.0:3337->3306/tcp  web4-mysql-db-1
```

Subimos el fichero

```
docker cp triggers.sh web4-mysql-db-1:/root/
```

Y ejecutarlo en el container.

```
$ docker exec -it web4-mysql-db-1 bash
root@d804a906babd:/# cd /root
# chmod 755 delete_triggers.sh
# ./delete_triggers.sh
Eliminando trigger: trg_catalog_category_entity_after_insert
Eliminando trigger: trg_catalog_category_entity_after_update
Eliminando trigger: trg_catalog_category_entity_after_delete
Eliminando trigger: trg_catalog_category_entity_datetime_after_insert
Eliminando trigger: trg_catalog_category_entity_datetime_after_update
Eliminando trigger: trg_catalog_category_entity_datetime_after_delete
Eliminando trigger: trg_catalog_category_entity_decimal_after_insert
Eliminando trigger: trg_catalog_category_entity_decimal_after_update
Eliminando trigger: trg_catalog_category_entity_decimal_after_delete
Eliminando trigger: trg_catalog_category_entity_int_after_insert
Eliminando trigger: trg_catalog_category_entity_int_after_update
Eliminando trigger: trg_catalog_category_entity_int_after_delete
Eliminando trigger: trg_catalog_category_entity_text_after_insert
Eliminando trigger: trg_catalog_category_entity_text_after_update
Eliminando trigger: trg_catalog_category_entity_text_after_delete
Eliminando trigger: trg_catalog_category_entity_varchar_after_insert
Eliminando trigger: trg_catalog_category_entity_varchar_after_update
Eliminando trigger: trg_catalog_category_entity_varchar_after_delete
Eliminando trigger: trg_catalog_category_product_after_insert
Eliminando trigger: trg_catalog_category_product_after_update
Eliminando trigger: trg_catalog_category_product_after_delete
Eliminando trigger: trg_catalog_compare_item_after_insert
Eliminando trigger: trg_catalog_compare_item_after_update
Eliminando trigger: trg_catalog_compare_item_after_delete
Eliminando trigger: trg_catalog_product_bundle_option_after_insert
Eliminando trigger: trg_catalog_product_bundle_option_after_update
Eliminando trigger: trg_catalog_product_bundle_option_after_delete
Eliminando trigger: trg_catalog_product_bundle_option_value_after_insert
Eliminando trigger: trg_catalog_product_bundle_option_value_after_update
Eliminando trigger: trg_catalog_product_bundle_option_value_after_delete
Eliminando trigger: trg_catalog_product_bundle_selection_after_insert
Eliminando trigger: trg_catalog_product_bundle_selection_after_update
Eliminando trigger: trg_catalog_product_bundle_selection_after_delete
Eliminando trigger: trg_catalog_product_bundle_selection_price_after_insert
```

```
Eliminando trigger: trg_catalog_product_bundle_selection_price_after_update
Eliminando trigger: trg_catalog_product_bundle_selection_price_after_delete
Eliminando trigger: trg_catalog_product_entity_after_insert
Eliminando trigger: trg_catalog_product_entity_after_update
Eliminando trigger: trg_catalog_product_entity_after_delete
Eliminando trigger: trg_catalog_product_entity_datetime_after_insert
Eliminando trigger: trg_catalog_product_entity_datetime_after_update
Eliminando trigger: trg_catalog_product_entity_datetime_after_delete
Eliminando trigger: trg_catalog_product_entity_decimal_after_insert
Eliminando trigger: trg_catalog_product_entity_decimal_after_update
Eliminando trigger: trg_catalog_product_entity_decimal_after_delete
Eliminando trigger: trg_catalog_product_entity_group_price_after_insert
Eliminando trigger: trg_catalog_product_entity_group_price_after_update
Eliminando trigger: trg_catalog_product_entity_group_price_after_delete
```

Toda una aventura, que en modo stress, no es buena para nadie.

# Finales

Muchas veces oigo en mi trabajo, "Es que se puede hacer así".

Con mil cosas, caracteres extendidos, desarrollos con millones de ficheros en la misma carpeta, etc. Y la misma cantinela. "Ahora el límite..." "Se puede..."

La realidad es muy distinta. Y eso se aprende en el infierno de los sistemas, cuando llaman a nuestra puerta.

Una cosa es que se pueda y otra muy distinta, es que se deba.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

---

Revision #1

Created 2024-03-21 12:40:40 UTC by Abkrim

Updated 2024-03-21 12:40:59 UTC by Abkrim