



MODELOS DE DATOS

PL/SQL - TRIGGERS

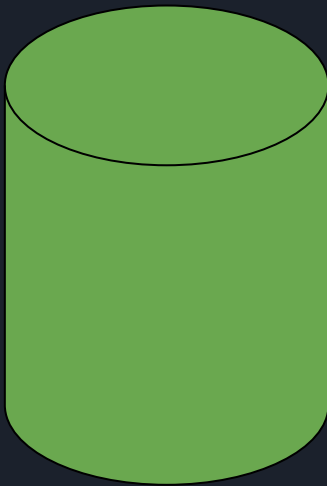


RECORDERIS...





¿QUÉ APRENDEREMOS HOY?



- Qué son los triggers (o disparadores) y para qué se usan
- Tipos de triggers
- Cómo éstos pueden ayudar a automatizar tareas operativas



Pregunta: ¿Qué es un trigger?





DEFINICIÓN: TRIGGER

Código pl/sql que se ejecuta de forma automática cuando ocurre una acción de modificación de datos (insert, update, delete).

NOTA: No aplica para operaciones de consulta (select)



TRIGGERS: Ejemplos

Ejemplo 1:

En una papelería se realiza una venta, la cual implica un *insert* en la tabla de ventas. Este insert ejecutará automáticamente un código pl/sql (también puede ser una función o un procedimiento almacenado) que hará un *update* en la tabla de inventario



TRIGGERS: Ejemplos

Ejemplo 2:

En un sistema de logística se actualiza el estado de los pedidos, por ejemplo "abierto", "en proceso", "en despacho", "finalizado". Cuando el estado cambie a "finalizado" mediante un *update*, realizar un *insert* en una tabla de encuestas de satisfacción de modo que posteriormente se pueda enviar al cliente para su calificación



TRIGGERS: Ejemplos

Ejemplo 3:

En un sistema de gestión académica un estudiante abandona un curso, lo cual implica un *delete* sobre la tabla que relaciona al estudiante con el curso. Esta operación puede generar un *insert* en una tabla de históricos, de modo que se pueda tener referencia de qué cursos han sido abandonados por cuáles estudiantes



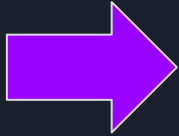
¿Para qué crear triggers?

- Automatizar algunas operaciones (como se mencionó en los ejemplos anteriores)
- Auditoría de quién modifica qué datos, cuándo y desde donde
- Tener historia de los estados de un proceso (similar a un diagrama de estados)



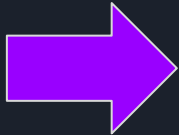
Tiempo de los triggers

BEFORE



El código pl/sql se ejecuta primero, y luego se hace la operación de update, insert o delete que disparó el trigger

AFTER

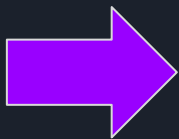


Primero se hace la operación de update, insert o delete y luego se ejecuta el código pl/sql del trigger



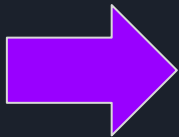
¿Cuándo usar cual?

BEFORE



Si se necesita realizar alguna validación en los datos que se están insertando. O si se debe llenar un campo con un valor que depende de otro

AFTER



Es el más común. Cuando se desea auditar un cambio o crear un histórico de cambio de estados.

Estructura de un trigger

Trigger

Encabezado

- Nombre del trigger (obligatorio)
- Tabla a la cual se asocia (obligatorio)
- Tiempo [before / after] (obligatorio)
- NO TIENE parámetros de entrada o salida

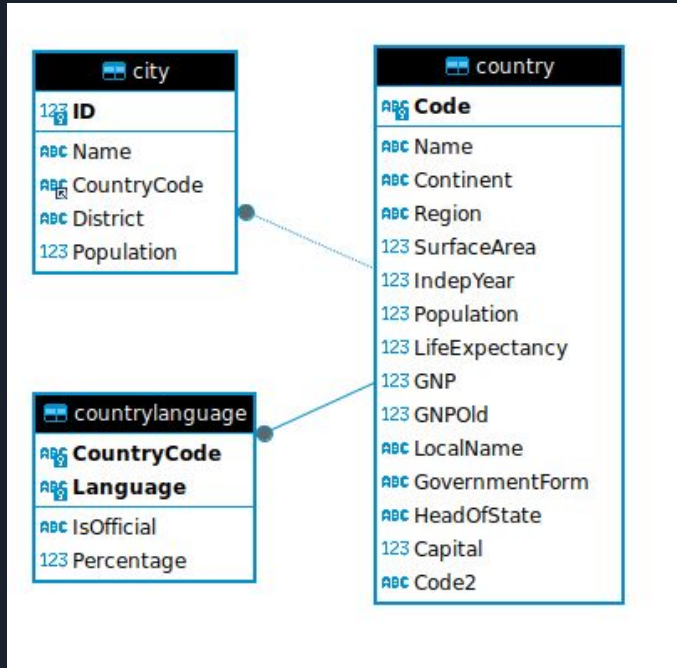
Cuerpo

Código pl/sql, incluyendo llamados a funciones o procedimientos almacenados. No es posible llamar directamente a otros triggers.

Documentación

Realmente no hace parte de la estructura. Pero... si ayuda a entender para qué sirve!

Ejemplo: Auditar cambios en "world"



Para monitorear los *cambios de población* en las ciudades, se creará una tabla para guardar la historia de cada cambio.

```
create table city_history(  
    id int(11) not null auto_increment,  
    city_id int(11) not null,  
    event_date datetime not null,  
    event_user varchar(100) not null,  
    event_action varchar(30) not null,  
    value_before varchar(255) not null,  
    value_after varchar(255) not null,  
    primary key(id)  
);
```

Ejemplo: Auditar cambios en "world"

```
delimiter $$
```

Nombre del trigger

```
create trigger tr_audit_city_population
```

```
after update on city
```

```
for each row
```

```
begin
```

```
insert into city_history(city_id, event_date, event_user,  
event_action, value_before, value_after)
```

```
values (old.id, now(), user(),
```

```
'UPDATE POPULATION', old.Population, new.Population);
```

```
end;
```

```
$$
```

```
delimiter ;
```

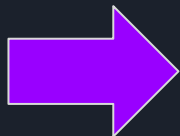
"old" y "new" son estructuras que permiten obtener los valores antes y después de actualizar

Tiempo y tipo de operación: este código se ejecuta DESPUÉS de actualizar la tabla

La tabla a monitorear es "city"

Ejemplo: Auditar cambios en "world"

Antes de
actualizar



```
MariaDB [world]> select * from city where id = 2259;
```

ID	Name	CountryCode	District	Population
2259	Medellín	COL	Antioquia	1861265

```
1 row in set (0,001 sec)
```

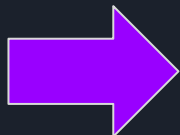
```
MariaDB [world]> select * from city_history;
```

Empty set (0,000 sec)

```
MariaDB [world]> update city set Population = 2590000 where id = 2259;
```

Query OK, 1 row affected (0,006 sec)
Rows matched: 1 Changed: 1 Warnings: 0

Después de
actualizar



```
MariaDB [world]> select * from city where id = 2259;
```

ID	Name	CountryCode	District	Population
2259	Medellín	COL	Antioquia	2590000

```
1 row in set (0,000 sec)
```

```
MariaDB [world]> select * from city_history;
```

id	city_id	event_date	event_user	event_action	value_before	value_after
1	2259	2024-03-30 19:28:25	root@172.19.0.1	UPDATE POPULATION	1861265	2590000

```
1 row in set (0,000 sec)
```



EJERCICIO (1 / 4)

NOTA

En esta presentación se proporcionó un ejemplo para triggers tipo "after update", que son eventos que se ejecutan después de actualizar. Sin embargo, y a propósito, no se proporcionaron ejemplos para después de borrar o después de insertar, ni tampoco ejemplos tipo "before", ya que quedarán como ejercicio de consulta.



EJERCICIO (2 / 4)

Ejercicio 1:

Crear un trigger que monitoree registros borrados sobre la tabla "city". En la tabla de auditoría debe registrarse así:

- En el campo "event_action" debe quedar el texto "DELETE CITY"
- En el campo "value_before" debe quedar el nombre de la ciudad, código del país, nombre del distrito y población que estaban antes de borrar el registro, como un texto separado por coma
- El campo "value_after" debe quedar con valor "null"
- En el campo "city_id" debe quedar el id de la ciudad que se borra



EJERCICIO (3 / 4)

Ejercicio 2:

Crear un trigger que monitoree registros insertados sobre la tabla "city". En la tabla de auditoría debe registrarse así:

- En el campo "event_action" debe quedar el texto "CREATE CITY"
- En el campo "value_after" debe quedar el nombre de la ciudad, código del país, nombre del distrito y población que se están insertando, como un texto separado por coma
- El campo "value_before" debe quedar con valor "null"
- En el campo "city_id" debe quedar el id de la ciudad que se inserta



EJERCICIO (4 / 4)

Ejercicio 3:

Crear un trigger que, al momento de actualizar la población, sume 1 al valor que se especifica en el update. La tabla de auditoría debe quedar igual que en el ejemplo dado en esta presentación.

- NOTA: Este trigger es tipo "before update".



¿PREGUNTAS?

