



Triggers en SQL

Explorando la definición, importancia y ejemplos de uso en SQL.

ADRIAN STIVEN MURILLO PALACIOS





Introducción


Los triggers en SQL son procedimientos almacenados que se ejecutan automáticamente en respuesta a ciertos eventos en las tablas de una base de datos. Son fundamentales para mantener la integridad de los datos y automatizar acciones dentro de una base de datos. En esta presentación, exploraremos la definición de triggers y su importancia en el contexto de SQL.





Definición de Triggers


Un trigger es un conjunto de instrucciones SQL que se ejecuta en respuesta a eventos específicos en una tabla, como insertar, actualizar o eliminar datos. Pueden activarse antes o después de que un evento ocurra, y son útiles para realizar validaciones, auditorías y automatizaciones dentro de la base de datos.





Importancia de los Triggers en SQL

Los triggers son esenciales para garantizar la integridad de los datos al proporcionar un mecanismo para aplicar automáticamente reglas o restricciones. Ayudan a mantener la consistencia de la base de datos sin la intervención manual, reduciendo así el riesgo de errores. Además, permiten realizar auditorías de cambios, logueo de eventos y mantener registros históricos de modificaciones.





Tipos de Triggers

Existen varios tipos de triggers en SQL, clasificados generalmente en:

1. ****BEFORE Trigger****: Se activa antes de que se ejecute la operación que provoca su ejecución, útil para validar datos antes de su inserción o modificación.
2. ****AFTER Trigger****: Se activa después de que la operación se completa, ideal para realizar tareas como el registro en auditorías o la actualización de otras tablas.
3. ****INSTEAD OF Trigger****: Se utiliza principalmente en vistas. Este tipo de trigger reemplaza la acción de la operación original, permitiendo realizar acciones distintas al interactuar con vistas.





Ejemplo de Trigger de Insert

Un trigger de tipo INSERT se activa automáticamente cuando se inserta una nueva fila en una tabla. Por ejemplo, si tenemos una tabla de `empleados`, podemos crear un trigger que registre automáticamente la fecha de inserción de cada nuevo empleado en una tabla de `auditoría`.

```
```sql
CREATE TRIGGER trg_AuditorialInsert
AFTER INSERT ON empleados
FOR EACH ROW
BEGIN
INSERT INTO auditoria (tabla, accion, fecha)
VALUES ('empleados', 'INSERT', NOW());
END;
```

```





Ejemplo de Trigger de Update

Los triggers de tipo UPDATE se activan cuando se actualizan los registros en la tabla. Por ejemplo, podemos crear un trigger que notifique cuando un salario ha sido actualizado. Esto es útil para mantener un registro de cambios en los salarios.

```
```sql
CREATE TRIGGER trg_SalarioUpdate
AFTER UPDATE ON empleados
FOR EACH ROW
BEGIN
 IF OLD.salario != NEW.salario THEN
INSERT INTO auditoria (tabla, accion, fecha)
VALUES ('empleados', 'UPDATE', NOW());
 END IF;
END;
```

...







# Ejemplo de Trigger de Delete

Los triggers de tipo DELETE se ejecutan cuando se elimina una fila de una tabla. Se pueden usar para asegurar que hay un registro de lo que se ha eliminado. Por ejemplo, al eliminar un empleado, podemos registrar esta acción.

```
```sql
CREATE TRIGGER trg_EmpleadoDelete
AFTER DELETE ON empleados
FOR EACH ROW
BEGIN
INSERT INTO auditoria (tabla, accion, fecha)
VALUES ('empleados', 'DELETE', NOW());
END;
```

```







# Conclusión

Los triggers son herramientas poderosas en SQL que permiten la automatización de tareas y el mantenimiento de la integridad de los datos.







**GRACIAS.!**

