

# Rutinas



- Funciones (Functions)
- Procedimientos Almacenados (Stored Procedures)
- Disparadores (Triggers)

# Disparadores

Triggers

# Triggers

Un Trigger en MySQL es un programa almacenado (stored program), creado para ejecutarse automáticamente cuando ocurra un evento en la base de datos.

- Pueden ser activados antes (before) o después (after) de la ejecución de un evento
- Los eventos pueden ser: INSERT, UPDATE, DELETE

# Triggers

Sintaxis:

```
CREATE [DEFINER={usuario|CURRENT_USER}]  
TRIGGER nombre_del_trigger {BEFORE|AFTER}  
{UPDATE|INSERT|DELETE}  
ON nombre_de_la_tabla  
FOR EACH ROW  
<bloque_de_instrucciones>
```

**DEFINER**= {usuario|CURRENT\_USER}: Indica al gestor de bases de datos qué usuario tiene privilegios en su cuenta, para la invocación de los triggers cuando surjan los eventos DML.

Por defecto esta característica tiene el valor CURRENT\_USER que hace referencia al usuario actual que está creando el Trigger.

# Triggers

**nombre\_del\_trigger:** Existe una nomenclatura muy práctica para nombrar un trigger, la cual da una mejor legibilidad en la administración de la base de datos.

Primero va el nombre de tabla, luego se usa la inicial del momento de ejecución (AFTER o BEFORE) y seguido de la inicial de la operación DML. Por ejemplo:

`autos_AU_trigger`

**BEFORE | AFTER:** Especifica si el Trigger se ejecuta antes o después del evento DML.

**UPDATE | INSERT | DELETE:** Indica la sentencia que se usará para que se ejecute el Trigger.

# Triggers

**ON nombre\_de\_la\_tabla:** En esta sección se establece el nombre de la tabla asociada.

**FOR EACH ROW:** Establece que el Trigger se ejecute por cada fila en la tabla asociada.

**<bloque\_de\_instrucciones>:** Define el bloque de sentencias que el Trigger ejecutará al ser invocado.

# Triggers

## Identificadores NEW y OLD en Triggers

Para relacionar el trigger con columnas específicas de una tabla se deben usar los identificadores OLD y NEW.

OLD indica el valor antiguo de la columna y NEW el valor nuevo que pudiese tomar.

Por ejemplo: **OLD**.id\_disco ó **NEW**.id\_disco.

Si se usa la sentencia UPDATE se puede usar un valor OLD o NEW, ya que se modifican registros existentes por nuevos valores.

En cambio si se usa INSERT sólo se usa NEW, ya que su naturaleza es únicamente de insertar nuevos valores a las columnas.

Con DELETE se usa OLD debido a que se borran valores que existen con anterioridad.

# Triggers

- ¿Qué utilidades tienen los Triggers?
  - Con los Triggers se pueden implementar varios casos de uso que mantengan la integridad de la base de datos, como validar información, calcular atributos derivados, seguimientos de movimientos en la base de datos, etc.
  - Cuando surge el caso en donde se necesita que se ejecute una acción implícitamente (sin que se ejecute manualmente) sobre los registros de una tabla, entonces se puede considerar el uso de un Trigger.



# Triggers

Ejemplo: (usar la base de datos collectionhw)

## 1. Crear una tabla para almacenar los datos actualizados

```
CREATE TABLE preciosActualizados (  
  id int NOT NULL AUTO_INCREMENT,  
  sku char(5) NOT NULL,  
  modelo smallint(6) NOT NULL,  
  precioViejo decimal(10,2) NOT NULL,  
  fechaRegistro timestamp NOT NULL,  
  PRIMARY KEY (id)) ;
```

# Triggers

2. Crear un trigger para registrar los autos que actualizaron sus precios

```
Create trigger autos_AU_trigger after update on carritos  
for each row  
insert into  
preciosActualizados(sku,modelo,precioViejo,fechaRegistro)  
values(OLD.sku,old.modelo,old.precio,now());
```

3. Mostrar el contenido de la tabla preciosActualizados:

```
select * from preciosActualizados;
```

4. Llamar al procedimiento almacenado actualizar\_precios:

```
call actualizar_precios(2006,10);
```

5. Mostrar el contenido de la tabla preciosActualizados:

```
select * from preciosActualizados;
```

# Consulta de rutinas

1. Ver el estado de funciones, triggers o procedures

```
show procedure status\G
```

```
show function status\G
```

```
show triggers\G
```

2. Ver el código:

```
show create procedure actualiza_precios\G
```

```
show create function saludo\G
```

```
show create trigger autos_AU_trigger\G
```

3. Borrar

```
drop procedure actualiza_precios;
```

```
drop function saludo;
```

```
drop trigger autos_AU_trigger;
```

4. Consultar rutinas de una sola base

```
SELECT ROUTINE_TYPE, ROUTINE_NAME, ROUTINE_DEFINITION FROM  
INFORMATION_SCHEMA.ROUTINES WHERE ROUTINE_SCHEMA='pixup'\G
```

¿Preguntas?