**DOCUMENTACIÓN**

**Generar un registro en caso de que se elimina algo de las siguientes tablas: productlines, product, customers**

----------------------------------------------------------------

Primero se crean las tablas a donde se van a redirigir los datos.

|  |
| --- |
| DROP TABLE IF EXISTS productlinesEliminados;  CREATE TABLE productlinesEliminados  (  id INT PRIMARY KEY AUTO\_INCREMENT,  productLine VARCHAR(50) not null,  textDescription VARCHAR(4000),  htmlDescription MEDIUMTEXT,  image MEDIUMBLOB,  fecha\_de\_eliminacion TIMESTAMP DEFAULT NOW()  );  DROP TABLE IF EXISTS productsEliminados;  CREATE TABLE productsEliminados  (  id INT PRIMARY KEY AUTO\_INCREMENT,  productCode varchar(15) not null,  productName varchar(70) not null,  productLine VARCHAR(50) not null,  productScale varchar(10) not null,  productVendor varchar(50) not null,  productDescription text not null,  quantityInStock smallint(6) not null,  buyPrice decimal(10,2) not null,  MSRP decimal(10,2) not null,  fecha\_de\_eliminacion TIMESTAMP DEFAULT NOW()  );  DROP TABLE IF EXISTS customersEliminados;  CREATE TABLE customersEliminados  (  id INT PRIMARY KEY AUTO\_INCREMENT,  customerNumber int(11) not null,  customerName varchar(50) not null,  contactLastName varchar(50) not null,  contactFirstName varchar(50) not null,  phone varchar(50) not null,  addressLine1 varchar(50) not null,  addressLine2 varchar(50) ,  city varchar(50) not null,  state varchar(50),  postalCode varchar(15),  country varchar(50),  salesRepEmployeeNumber int(11),  creditLimit decimal(10,2),  fecha\_de\_eliminacion TIMESTAMP DEFAULT NOW()  ); |

Luego se generarán los disparadores, que serán del caso before delete.

La consulta es insertar los valores que se están eliminando en otra tabla, por lo que se utiliza old antes de los campos para decirle a la consulta que tome los viejos registros para ingresarlos.

|  |
| --- |
| DELIMITER $$  DROP TRIGGER IF EXISTS before\_productline\_eliminar;  CREATE TRIGGER before\_productline\_eliminar  BEFORE DELETE  ON productlines  FOR EACH ROW  BEGIN  INSERT INTO productlinesEliminados(productLine, textDescription, htmlDescription, image)  VALUES (OLD.productLine, OLD.textDescription, OLD.htmlDescription, OLD.image);  END$$  DELIMITER ;  DELIMITER aa  DROP TRIGGER IF EXISTS before\_product\_eliminar;  CREATE TRIGGER before\_product\_eliminar  BEFORE DELETE  ON products  FOR EACH ROW  BEGIN  INSERT INTO productsEliminados(productCode, productName, productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP)  VALUES (OLD.productCode, OLD.productName, OLD.productLine, OLD.productScale, OLD.productVendor, OLD.productDescription, OLD.quantityInStock, OLD.buyPrice, OLD.MSRP);  ENDaa  DELIMITER ;  DELIMITER rr  DROP TRIGGER IF EXISTS before\_customer\_eliminar;  CREATE TRIGGER before\_customer\_eliminar  BEFORE DELETE  ON customers  FOR EACH ROW  BEGIN  INSERT INTO customersEliminados( customerNumber, customerName, contactLastName, contactFirstName, phone,  addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit)  VALUES (OLD.customerNumber, OLD.customerName, OLD.contactLastName, OLD.contactFirstName, OLD.phone,  OLD.addressLine1, OLD.addressLine2, OLD.city, OLD.state, OLD.postalCode, OLD.country, OLD.salesRepEmployeeNumber, OLD.creditLimit);  ENDrr  DELIMITER ; |

**Registros que se elimina en la tabla employees tiene que pasar a una tabla exEmployees al borrarlos**

Primero creamos la tabla exEmployees

|  |
| --- |
| DROP TABLE IF EXISTS exEmployees;  CREATE TABLE exEmployees  (  id int primary key auto\_increment,  employeeNumber int(11) not null,  lastName varchar(50) not null,  firstName varchar(50) not null,  extension varchar(10) not null,  email varchar(100) not null,  officeCode varchar(10) not null,  reportsTo int(11),  jobTitle varchar(50) not null,  fecha\_de\_eliminacion timestamp default now()  ); |

Después aplicamos la misma lógica del ejercicio anterior para insertar los valores en la nueva tabla.

|  |
| --- |
| DELIMITER $$  DROP TRIGGER IF EXISTS before\_employee\_eliminar;  CREATE TRIGGER before\_employee\_eliminar  BEFORE DELETE  ON employees  FOR EACH ROW  BEGIN  INSERT INTO exEmployees(employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle)  VALUES (OLD.employeeNumber, OLD.lastName, OLD.firstName, OLD.extension, OLD.email, OLD.officeCode, OLD.reportsTo, OLD.jobTitle);  END$$  DELIMITER ; |

**Impedir que se puede borrar algo de las tablas orders y payments**

Se utiliza lo mismo que los anteriores ejercicios el before delete, donde se realiza una condición donde no permita eliminar registros si no son nulos.

|  |
| --- |
| DELIMITER $$  DROP TRIGGER IF EXISTS Before\_order\_no\_eliminar;  CREATE TRIGGER Before\_order\_no\_eliminar  BEFORE DELETE ON orders  FOR EACH ROW  BEGIN  If Old.orderNumber is not null then  SIGNAL SQLSTATE '45000'  SET MESSAGE\_TEXT = 'No puede eliminar este registro de la tabla orders';  End If;  END$$  DELIMITER ;  DELIMITER aa  DROP TRIGGER IF EXISTS Before\_payments\_no\_eliminar;  CREATE TRIGGER Before\_payments\_no\_eliminar  BEFORE DELETE ON payments  FOR EACH ROW  BEGIN  If Old.checkNumber is not null then  SIGNAL SQLSTATE '45000'  SET MESSAGE\_TEXT = 'No puede eliminar este registro de la tabla payments';  End If;  END aa  DELIMITER ; |

**Actualizaciones o inserts a la tabla products tiene que registrarse en una tabla logProducto**

Primero se genera la tabla logProducto.

|  |
| --- |
| DROP TABLE IF EXISTS logProducto;  CREATE TABLE logProducto(  id int auto\_increment,  productCode varchar(15),  dateLog timestamp default now(),  descripcion varchar(255) not null,  primary key (id, productCode)  ); |

Después se crean dos delimitadores, para el ingreso y para actualizar

|  |
| --- |
| DELIMITER $$  DROP TRIGGER IF EXISTS after\_insertar\_products;  create trigger after\_insertar\_products  after insert on products  for each row  Begin  INSERT INTO logProducto(productCode, descripcion)  VALUES (new.productCode, CONCAT('Se ha insertado un nuevo producto: ', NEW.productName));  End$$  DELIMITER ;  DELIMITER dd  DROP TRIGGER IF EXISTS after\_actualizar\_products;  create trigger after\_actualizar\_products  after update on products  for each row  Begin  INSERT INTO logProducto(productCode, descripcion)  VALUES (old.productCode, CONCAT('Se ha actualizado un producto: ', old.productName));  End dd  DELIMITER ; |

Repositorio: <https://github.com/Jaky-Maiky/Base-de-datos/tree/master/Disparadores>