

PRÁCTICA 28|

Script MySQL

Jared Adrián Juárez Bernal

BASE DE DATOS

3°D - DSM

```

10     email varchar(255) not null,
11     telefono integer not null,
12
13     primary key (id_cliente)
14 );
15
16 • create table pedidos (
17     id_pedidos integer auto_increment,
18     fk_cliente integer not null,
19     fecha date,
20     total integer not null,
21
22     primary key(id_pedidos),
23     foreign key(fk_cliente) references clientes(id_cliente)
24 );
25
26 • INSERT INTO CLIENTES VALUES (1,'Jared Juarez','jaredjuarez@gmail.com',4997726);
27 • INSERT INTO CLIENTES VALUES (4,'Arturo','arturo@gmail.com',8888888);
28
29 • INSERT INTO PEDIDOS VALUES (1,1,("2023-06-15"),1200);
30 • INSERT INTO PEDIDOS VALUES (2,1,("2023-07-25"),2400);

```

Output


 Action Output

| # | Time | Action | Message |
|-----|----------|--|-------------------|
| ✓ 1 | 23:05:46 | create database practica | 1 row(s) affected |
| ✓ 2 | 23:05:51 | use practica | 0 row(s) affected |
| ✓ 3 | 23:05:55 | create table clientes (id_cliente integer auto_increment, nombre varchar(255) not null, email varchar(255) not nu... | 0 row(s) affected |
| ✓ 4 | 23:06:01 | create table pedidos (id_pedidos integer auto_increment, fk_cliente integer not null, fecha date, total integer not ... | 0 row(s) affected |
| ✓ 5 | 23:06:13 | INSERT INTO CLIENTES VALUES (1,'Jared Juarez',jaredjuarez@gmail.com',4997726) | 1 row(s) affected |
| ✓ 6 | 23:06:19 | INSERT INTO CLIENTES VALUES (4,'Arturo','arturo@gmail.com',8888888) | 1 row(s) affected |
| ✓ 7 | 23:06:24 | INSERT INTO PEDIDOS VALUES (1,1,("2023-06-15"),1200) | 1 row(s) affected |
| ✓ 8 | 23:06:28 | INSERT INTO PEDIDOS VALUES (2,1,("2023-07-25"),2400) | 1 row(s) affected |

```

33  /*Paso 2:
34  Crear un disparador 'BEFORE DELETE' para la tabla 'clientes' Crearemos un
35  disparador que verifique si un cliente tiene pedidos asociados antes de eliminarlo.*/
36  DELIMITER $$
37  CREATE TRIGGER validar_antes_eliminar BEFORE DELETE ON clientes
38  FOR EACH ROW
39  BEGIN
40      DECLARE pedidos_cont integer;
41
42      SELECT count(*) into pedidos_cont from pedidos where fk_cliente = OLD.id_cliente;
43      IF pedidos_cont > 0 THEN
44          SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '=== NO SE PUEDE ELIMINAR EL CLIENTE, TIENE PEDIDOS ASOCIADOS ===';
45      END IF;
46  END;$$

```

Output

Action Output

| # | Time | Action | Message |
|---|----------|--|-------------------|
| 3 | 23:05:55 | create table clientes (id_cliente integer auto_increment, nombre varchar(255) not null, email varchar(255) not nu... | 0 row(s) affected |
| 4 | 23:06:01 | create table pedidos (id_pedidos integer auto_increment, fk_cliente integer not null, fecha date, total integer not ... | 0 row(s) affected |
| 5 | 23:06:13 | INSERT INTO CLIENTES VALUES (1,'Jared Juarez','jaredjuarez@gmail.com',4997726) | 1 row(s) affected |
| 6 | 23:06:19 | INSERT INTO CLIENTES VALUES (4,'Arturo','arturo@gmail.com',8888888) | 1 row(s) affected |
| 7 | 23:06:24 | INSERT INTO PEDIDOS VALUES (1,1,("2023-06-15"),1200) | 1 row(s) affected |
| 8 | 23:06:28 | INSERT INTO PEDIDOS VALUES (2,1,("2023-07-25"),2400) | 1 row(s) affected |
| 9 | 23:07:28 | CREATE TRIGGER validar_antes_eliminar BEFORE DELETE ON clientes FOR EACH ROW BEGIN DECLA... | 0 row(s) affected |


```
47
48  /*PASO 3:
49  Probar el disparador 'BEFORE DELETE' Intente eliminar un cliente que tenga
50  pedidos asociados y verifique si el disparador funciona correctamente:*/
51  DELETE FROM clientes WHERE id_cliente=1;
52
```

[Context Help](#) [Snippets](#)

Output

 Action Output

| # | Time | Action | Message |
|---|----------|---|--|
|  1 | 23:09:14 | DELETE FROM clientes WHERE id_cliente=1 | Error Code: 1644. === NO SE PUEDE ELIMINAR EL CLIENTE, TIENE PEDIDOS ASOCIADOS === |

```

53  /*Paso 4:
54  Crear un disparador 'AFTER DELETE' para la tabla 'clientes' Crearemos un
55  disparador que registre la eliminación de un cliente en una tabla de auditoría
56  llamada 'clientes_eliminados'.*/
57  CREATE TABLE clientes_eliminados (
58      id_cliente_eliminado integer auto_increment,
59      id_cliente integer not null,
60      nombre varchar(255) not null,
61      email varchar(255) not null,
62      telefono integer not null,
63      fecha_eliminacion date,
64
65      primary key(id_cliente_eliminado)
66  );
67  DELIMITER $$
68
69  CREATE TRIGGER registrar_eliminados AFTER DELETE ON clientes
70  FOR EACH ROW
71  BEGIN
72      insert into clientes_eliminados(id_cliente,nombre,email,telefono,fecha_eliminacion)
73      value (old.id_cliente , old.nombre , old.email , old.telefono , sysdate());
74  END;$$

```

Output

Action Output









| # | Time | Action | Message |
|-----|----------|---|-------------------|
| ✓ 1 | 23:10:33 | CREATE TABLE clientes_eliminados (id_cliente_eliminado integer auto_increment, id_cliente integer not null, ... | 0 row(s) affected |
| ✓ 2 | 23:10:37 | CREATE TRIGGER registrar_eliminados AFTER DELETE ON clientes FOR EACH ROW BEGIN insert into cliente... | 0 row(s) affected |

```

76 • select * from clientes_eliminados;
77
78 /*Paso 5:
79 Probar el disparador 'AFTER DELETE' Elimine un cliente que no tenga pedidos
80 asociados y verifique si se registra en la tabla 'clientes_eliminados'.*/
81 DELETE FROM clientes WHERE id_cliente=4;

```

<

Result Grid   Filter Rows: Edit:    Export/Import:   Wrap Cell Content: 

| | id_cliente_eliminado | id_cliente | nombre | email | telefono | fecha_eliminacion |
|---|----------------------|------------|--------|------------------|----------|-------------------|
| ▶ | 1 | 4 | Arturo | arturo@gmail.com | 8888888 | 2023-06-14 |
| * | NULL | NULL | NULL | NULL | NULL | NULL |

s_eliminados1 x

Output

 Action Output ▼

| # | Time | Action | Message |
|-----|----------|---|-------------------|
| ✓ 1 | 23:12:14 | DELETE FROM clientes WHERE id_cliente=4 | 1 row(s) affected |
| ✓ 2 | 23:12:33 | select * from clientes_eliminados LIMIT 0, 1000 | 1 row(s) returned |