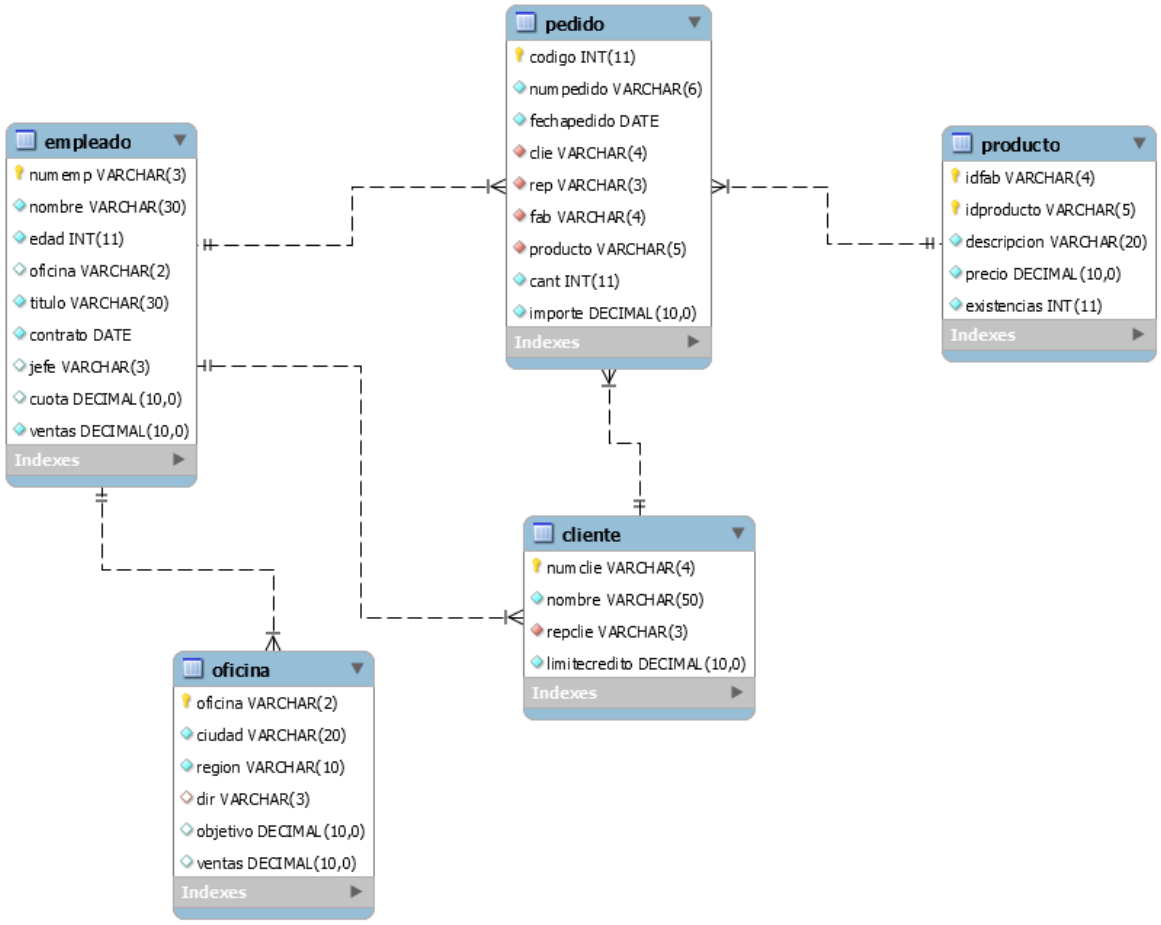
GUIA_04_Desarrollo de objetos transaccionales de las bases de datos
TALLER No. 2
PROCEDIMIENTOS ALMACENADOS Y DISPARADORES

ACTIVIDADES POR DESARROLLAR:

Para esta actividad deberá emplear el script compartido en la carpeta de Material de Apoyo denominado **05_05_SCRIPT_Empresa.sql** cuyo modelo relacional está representado por la siguiente imagen:



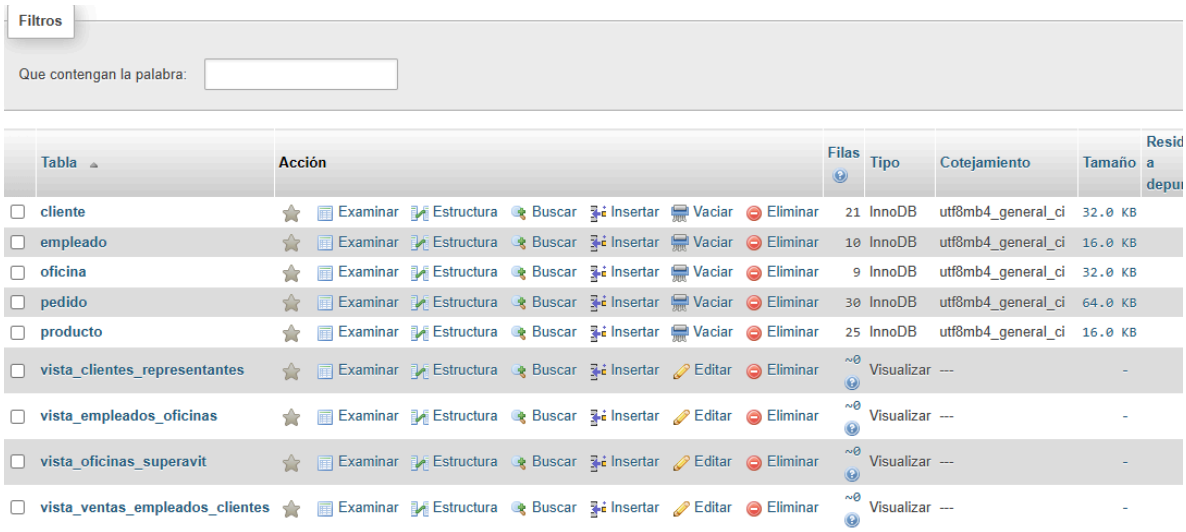
1. Descargue el script y ejecútelo en Mysql Workbench o en el phpMyAdmin según como usted trabaje. Si ya lo tiene descargado del taller anterior sólo abra el gestor de bases de datos y trabaje sobre la base ya creada.

2. El script trae todas las sentencias de creación de tablas, relaciones e inserciones que se requieren para realizar la actividad. Ejecútelas ordenadamente.
3. Cree los procedimientos almacenados para la CRUD de las tablas EMPLEADO, PRODUCTO y CLIENTE. Serán 4 procedimientos para cada tabla para Insertar, Consultar por clave, Actualizar-- Insertar empleado

EMPLEADO

Crear empleado por clave

```
DELIMITER //
CREATE PROCEDURE InsertarEmpleado(
   IN p_numemp VARCHAR(3),
   IN p_nombre VARCHAR(30),
   IN p_edad INTEGER,
   IN p_oficina VARCHAR(2),
   IN p_titulo VARCHAR(30),
   IN p_contrato DATE,
   IN p_jefe VARCHAR(3),
   IN p_cuota DECIMAL,
   IN p_ventas DECIMAL
)
BEGIN
    INSERT INTO empleado VALUES (p_numemp, p_nombre, p_edad, p_oficina, p_titulo, p_contrato, p_jefe, p_cuota, p_ventas);
END //
DELIMITER ;
```



Consultar empleado por clave

```
DELIMITER //
CREATE PROCEDURE ConsultarEmpleado(
   IN p_numemp VARCHAR(3)
)
BEGIN
   SELECT * FROM empleado WHERE numemp = p_numemp;
END //
DELIMITER ;
```

Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0004 segundos.)

```
SELECT * FROM `empleado`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

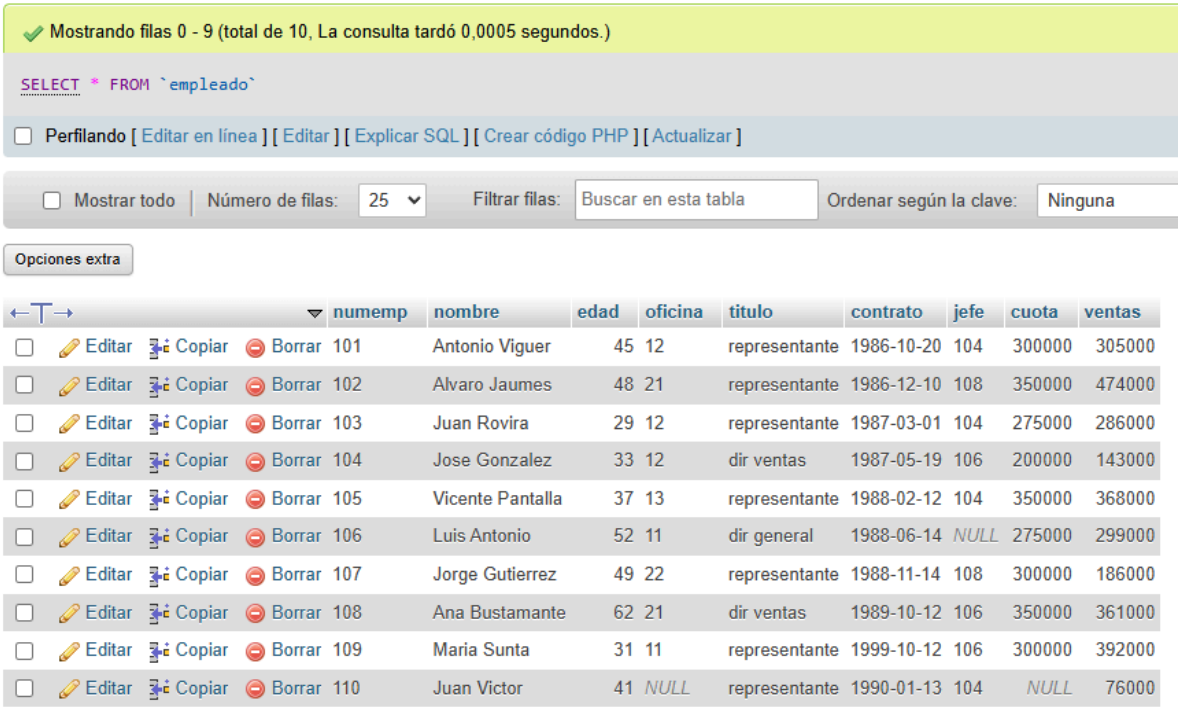
Opciones extra

	numemp	nombre	edad	oficina	titulo	contrato	jefe	cuota	ventas
Editar Copiar Borrar	101	Antonio Viguer	45	12	representante	1986-10-20	104	300000	305000
Editar Copiar Borrar	102	Alvaro Jaumes	48	21	representante	1986-12-10	108	350000	474000
Editar Copiar Borrar	103	Juan Rovira	29	12	representante	1987-03-01	104	275000	286000
Editar Copiar Borrar	104	Jose Gonzalez	33	12	dir ventas	1987-05-19	106	200000	143000
Editar Copiar Borrar	105	Vicente Pantalla	37	13	representante	1988-02-12	104	350000	368000
Editar Copiar Borrar	106	Luis Antonio	52	11	dir general	1988-06-14	NULL	275000	299000
Editar Copiar Borrar	107	Jorge Gutierrez	49	22	representante	1988-11-14	108	300000	186000
Editar Copiar Borrar	108	Ana Bustamante	62	21	dir ventas	1989-10-12	106	350000	361000
Editar Copiar Borrar	109	Maria Sunta	31	11	representante	1999-10-12	106	300000	392000
Editar Copiar Borrar	110	Juan Victor	41	NULL	representante	1990-01-13	104	NULL	76000

Actualizar empleado

```
DELIMITER //
CREATE PROCEDURE ActualizarEmpleado(
   IN p_numemp VARCHAR(3),
   IN p_nombre VARCHAR(30),
   IN p_edad INTEGER,
   IN p_oficina VARCHAR(2),
   IN p_titulo VARCHAR(30),
   IN p_contrato DATE,
   IN p_jefe VARCHAR(3),
   IN p_cuota DECIMAL,
   IN p_ventas DECIMAL
)
BEGIN
   UPDATE empleado
   SET nombre = p_nombre,
      edad = p_edad,
      oficina = p_oficina,
```

```
    titulo = p_titulo,
    contrato = p_contrato,
    jefe = p_jefe,
    cuota = p_cuota,
    ventas = p_ventas
  WHERE numemp = p_numemp;
END //
DELIMITER ;
```



Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0005 segundos.)

```
SELECT * FROM `empleado`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

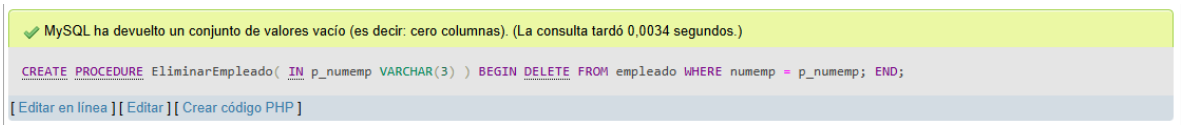
Opciones extra

	numemp	nombre	edad	oficina	titulo	contrato	jefe	cuota	ventas
Editar Copiar Borrar	101	Antonio Viguer	45	12	representante	1986-10-20	104	300000	305000
Editar Copiar Borrar	102	Alvaro Jaumes	48	21	representante	1986-12-10	108	350000	474000
Editar Copiar Borrar	103	Juan Rovira	29	12	representante	1987-03-01	104	275000	286000
Editar Copiar Borrar	104	Jose Gonzalez	33	12	dir ventas	1987-05-19	106	200000	143000
Editar Copiar Borrar	105	Vicente Pantalla	37	13	representante	1988-02-12	104	350000	368000
Editar Copiar Borrar	106	Luis Antonio	52	11	dir general	1988-06-14	NULL	275000	299000
Editar Copiar Borrar	107	Jorge Gutierrez	49	22	representante	1988-11-14	108	300000	186000
Editar Copiar Borrar	108	Ana Bustamante	62	21	dir ventas	1989-10-12	106	350000	361000
Editar Copiar Borrar	109	Maria Sunta	31	11	representante	1999-10-12	106	300000	392000
Editar Copiar Borrar	110	Juan Victor	41	NULL	representante	1990-01-13	104	NULL	76000

Eliminar empleado

```
DELIMITER //
CREATE PROCEDURE EliminarEmpleado(
  IN p_numemp VARCHAR(3)
)
BEGIN
  DELETE FROM empleado WHERE numemp = p_numemp;
END //
DELIMITER ;
```

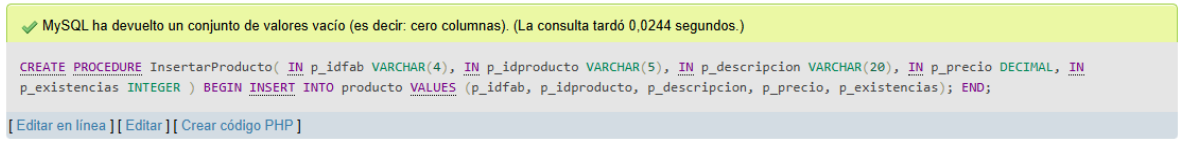
y Eliminar.



PRODUCTO

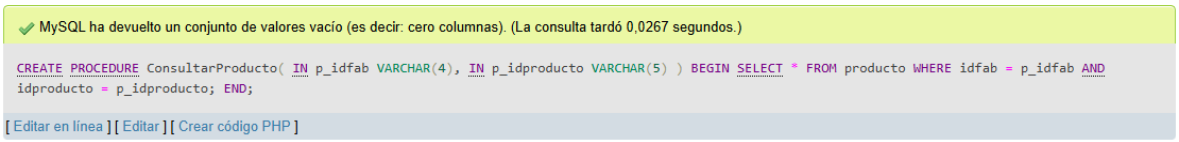
Insertar producto

```
DELIMITER //
CREATE PROCEDURE InsertarProducto(
   IN p_idfab VARCHAR(4),
   IN p_idproducto VARCHAR(5),
   IN p_descripcion VARCHAR(20),
   IN p_precio DECIMAL,
   IN p_existencias INTEGER
)
BEGIN
      INSERT INTO producto VALUES (p_idfab, p_idproducto, p_descripcion, p_precio, p_existencias);
END //
DELIMITER ;
```



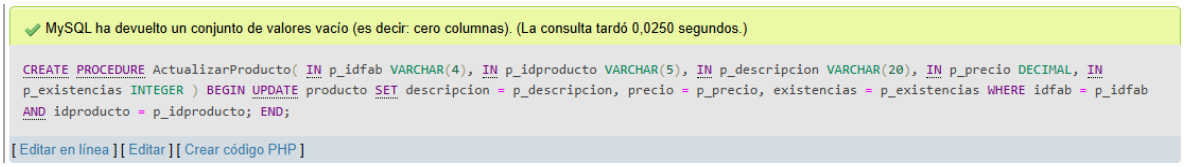
Consultar producto por clave

```
DELIMITER //
CREATE PROCEDURE ConsultarProducto(
   IN p_idfab VARCHAR(4),
   IN p_idproducto VARCHAR(5)
)
BEGIN
   SELECT * FROM producto WHERE idfab = p_idfab AND idproducto = p_idproducto;
END //
DELIMITER ;
```



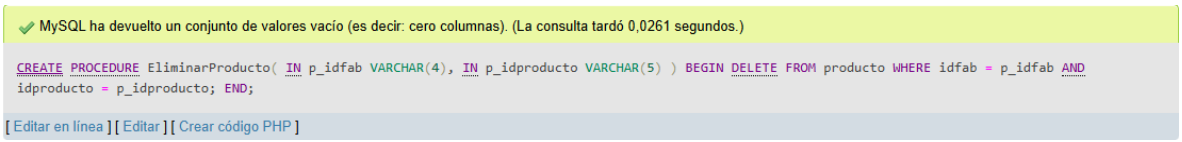
Actualizar producto

```
DELIMITER //
CREATE PROCEDURE ActualizarProducto(
   IN p_idfab VARCHAR(4),
   IN p_idproducto VARCHAR(5),
   IN p_descripcion VARCHAR(20),
   IN p_precio DECIMAL,
   IN p_existencias INTEGER
)
BEGIN
   UPDATE producto
   SET descripcion = p_descripcion,
      precio = p_precio,
      existencias = p_existencias
   WHERE idfab = p_idfab AND idproducto = p_idproducto;
END //
DELIMITER ;
```



Eliminar producto

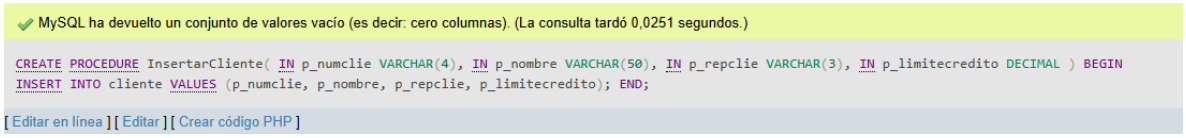
```
DELIMITER //
CREATE PROCEDURE EliminarProducto(
   IN p_idfab VARCHAR(4),
   IN p_idproducto VARCHAR(5)
)
BEGIN
   DELETE FROM producto WHERE idfab = p_idfab AND idproducto = p_idproducto;
END //
DELIMITER ;
```



CLIENTE

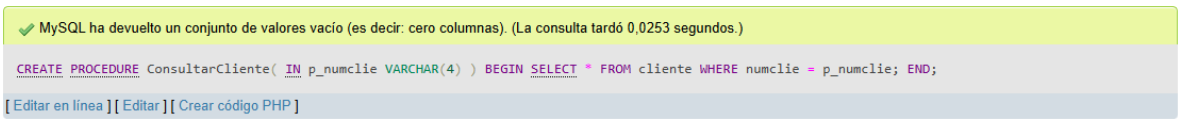
Insertar cliente

```
DELIMITER //
CREATE PROCEDURE InsertarCliente(
   IN p_numclie VARCHAR(4),
   IN p_nombre VARCHAR(50),
   IN p_repclie VARCHAR(3),
   IN p_limitecredito DECIMAL
)
BEGIN
   INSERT INTO cliente VALUES (p_numclie, p_nombre, p_repclie, p_limitecredito);
END //
DELIMITER ;
```



Consultar cliente por clave

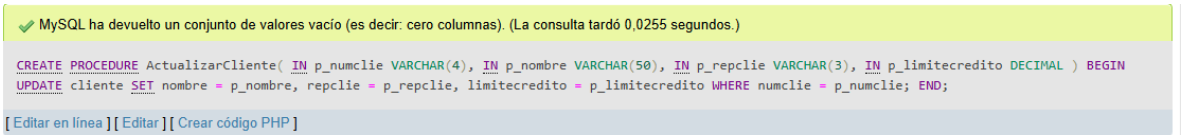
```
DELIMITER //
CREATE PROCEDURE ConsultarCliente(
   IN p_numclie VARCHAR(4)
)
BEGIN
   SELECT * FROM cliente WHERE numclie = p_numclie;
END //
DELIMITER ;
```



Actualizar cliente

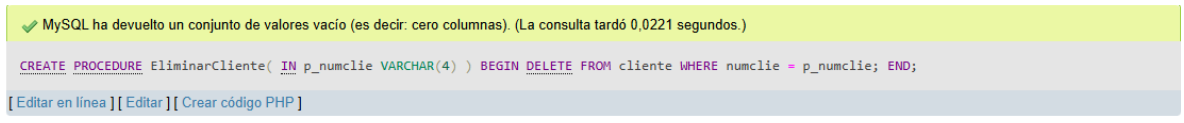
```
DELIMITER //
```

```
CREATE PROCEDURE ActualizarCliente(
   IN p_numclie VARCHAR(4),
   IN p_nombre VARCHAR(50),
   IN p_repclie VARCHAR(3),
   IN p_limitecredito DECIMAL
)
BEGIN
   UPDATE cliente
   SET nombre = p_nombre,
      repclie = p_repclie,
      limitecredito = p_limitecredito
   WHERE numclie = p_numclie;
END //
DELIMITER ;
```



Eliminar cliente

```
DELIMITER //
CREATE PROCEDURE EliminarCliente(
   IN p_numclie VARCHAR(4)
)
BEGIN
   DELETE FROM cliente WHERE numclie = p_numclie;
END //
DELIMITER ;
```

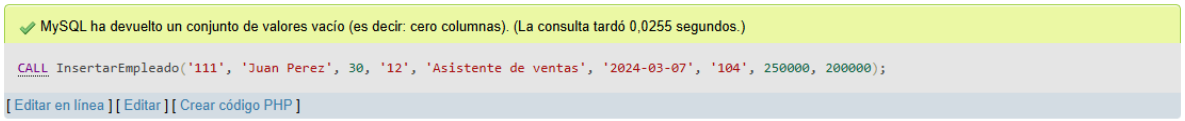


4. Realice pruebas con el comando CALL para utilizar cada uno de los procedimientos que creó para la base de datos EMPRESA

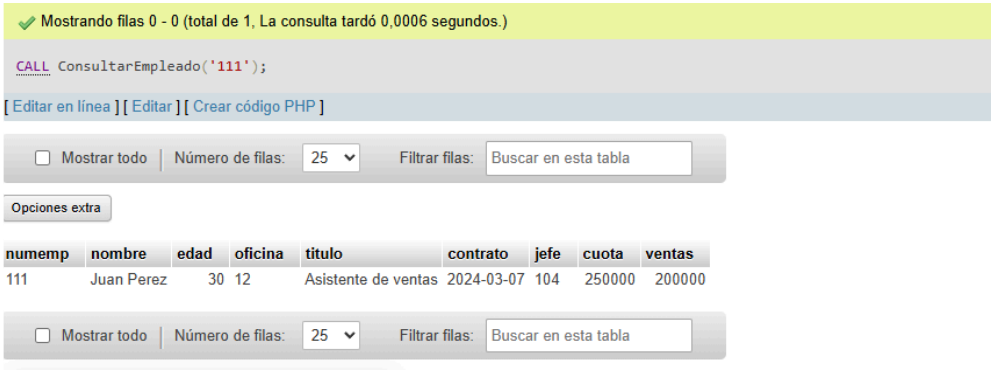
   EMPLEADO

   -- Insertar empleado

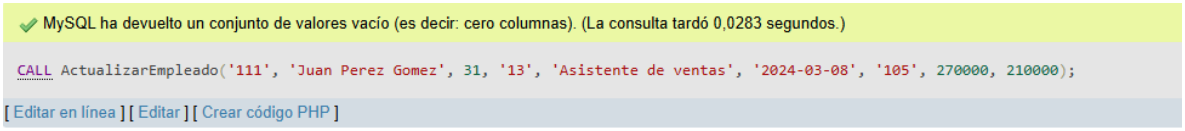
CALL InsertarEmpleado('111', 'Juan Perez', 30, '12', 'Asistente de ventas', '2024-03-07', '104', 250000, 200000);



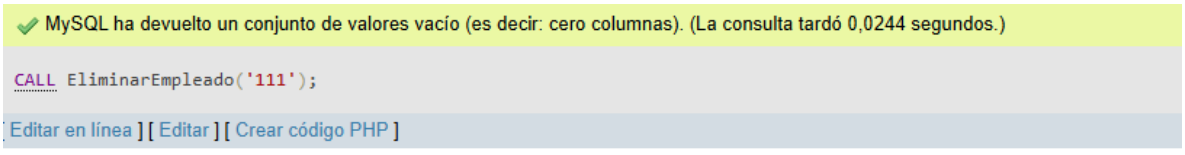
-- Consultar empleado por clave
CALL ConsultarEmpleado('111');



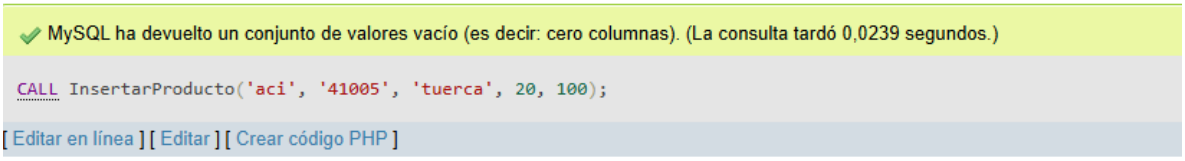
-- Actualizar empleado
CALL ActualizarEmpleado('111', 'Juan Perez Gomez', 31, '13', 'Asistente de ventas', '2024-03-08', '105', 270000, 210000);



-- Eliminar empleado
CALL EliminarEmpleado('111');



PRODUCTO
-- Insertar producto
CALL InsertarProducto('aci', '41005', 'tuerca', 20, 100);



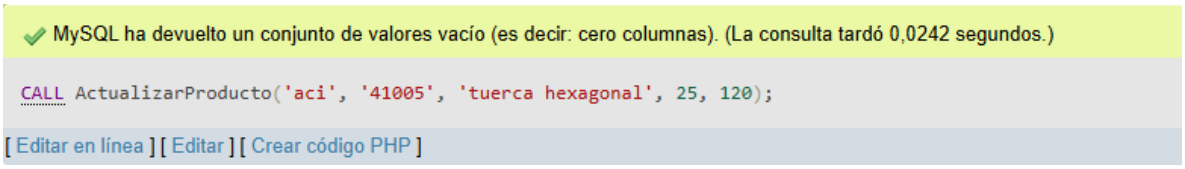
-- Consultar producto por clave

CALL ConsultarProducto('aci', '41005');



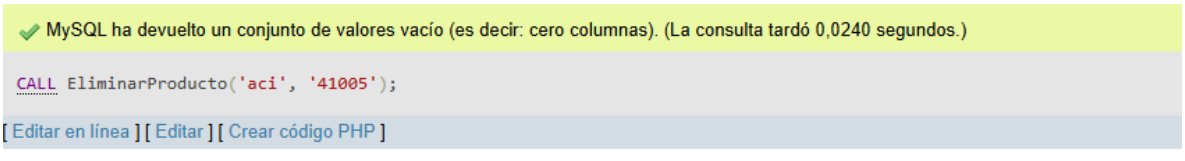
-- Actualizar producto

CALL ActualizarProducto('aci', '41005', 'tuerca hexagonal', 25, 120);



-- Eliminar producto

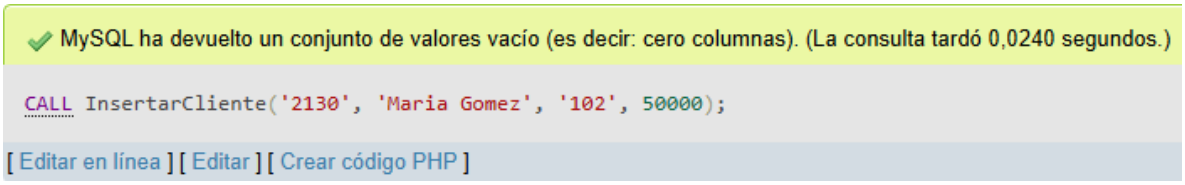
CALL EliminarProducto('aci', '41005');



CLIENTE

-- Insertar cliente

CALL InsertarCliente('2130', 'Maria Gomez', '102', 50000);



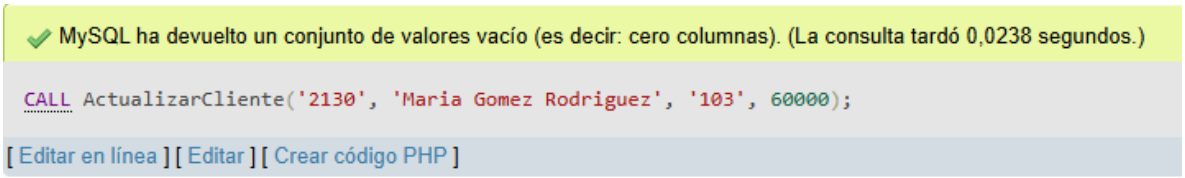
-- Consultar cliente por clave

CALL ConsultarCliente('2130');



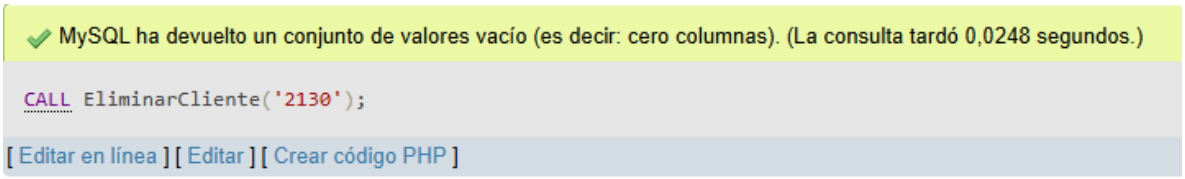
-- Actualizar cliente

CALL ActualizarCliente('2130', 'Maria Gomez Rodriguez', '103', 60000);



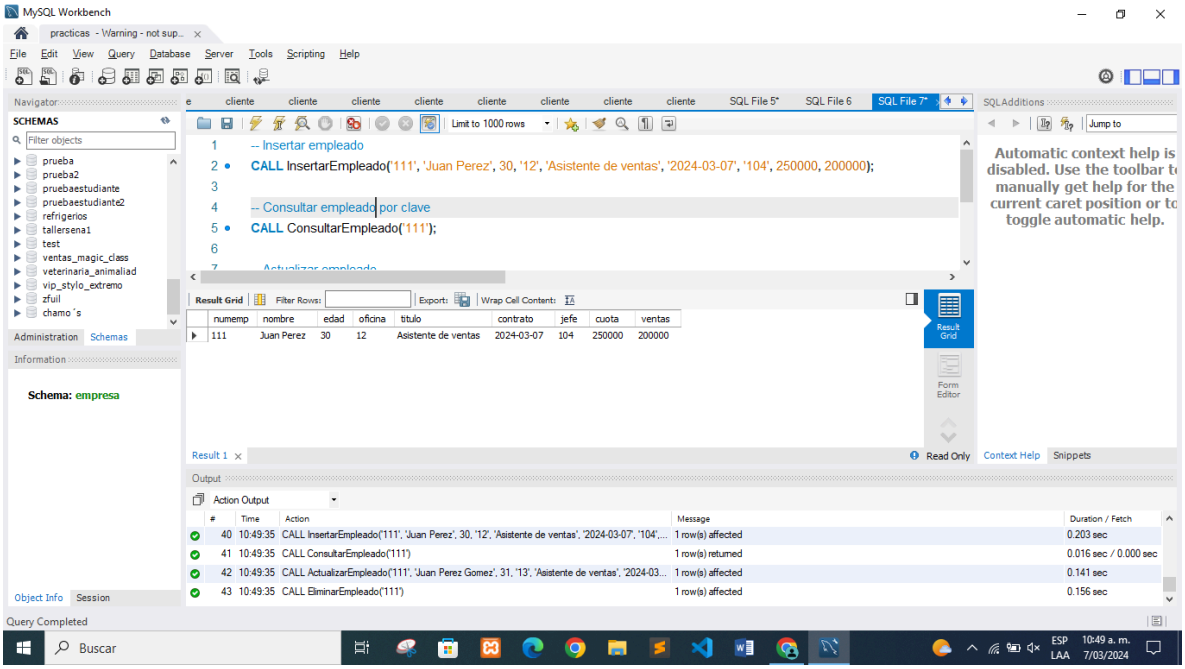
-- Eliminar cliente

CALL EliminarCliente('2130');

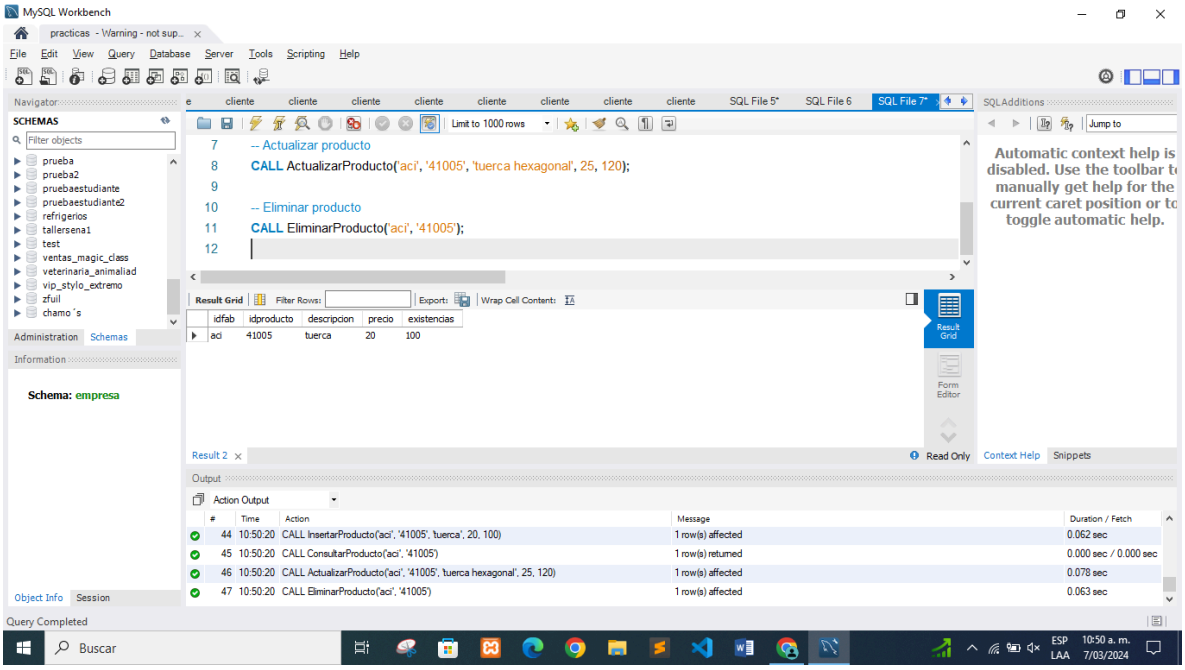


5. Tome pantallazo del código de creación de cada procedimiento almacenado y de por lo menos dos llamados (uso del comando CALL) para usar cada uno.

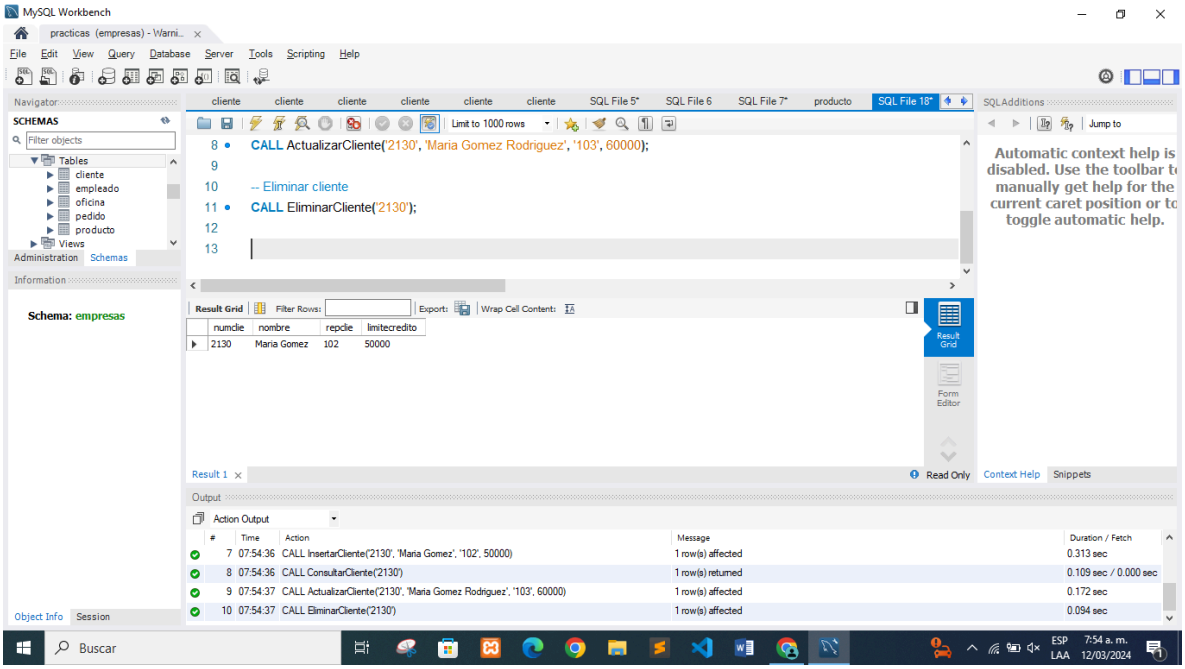
EMPLEADO



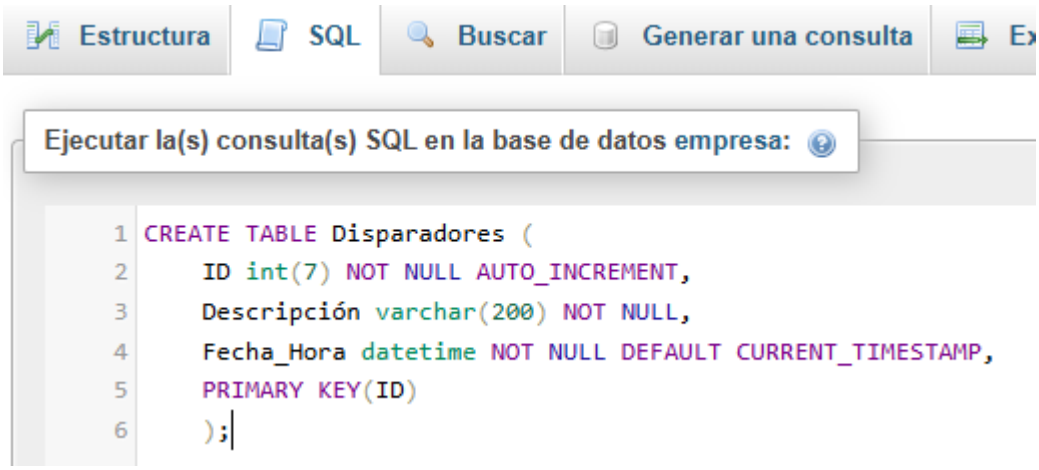
PRODUCTO



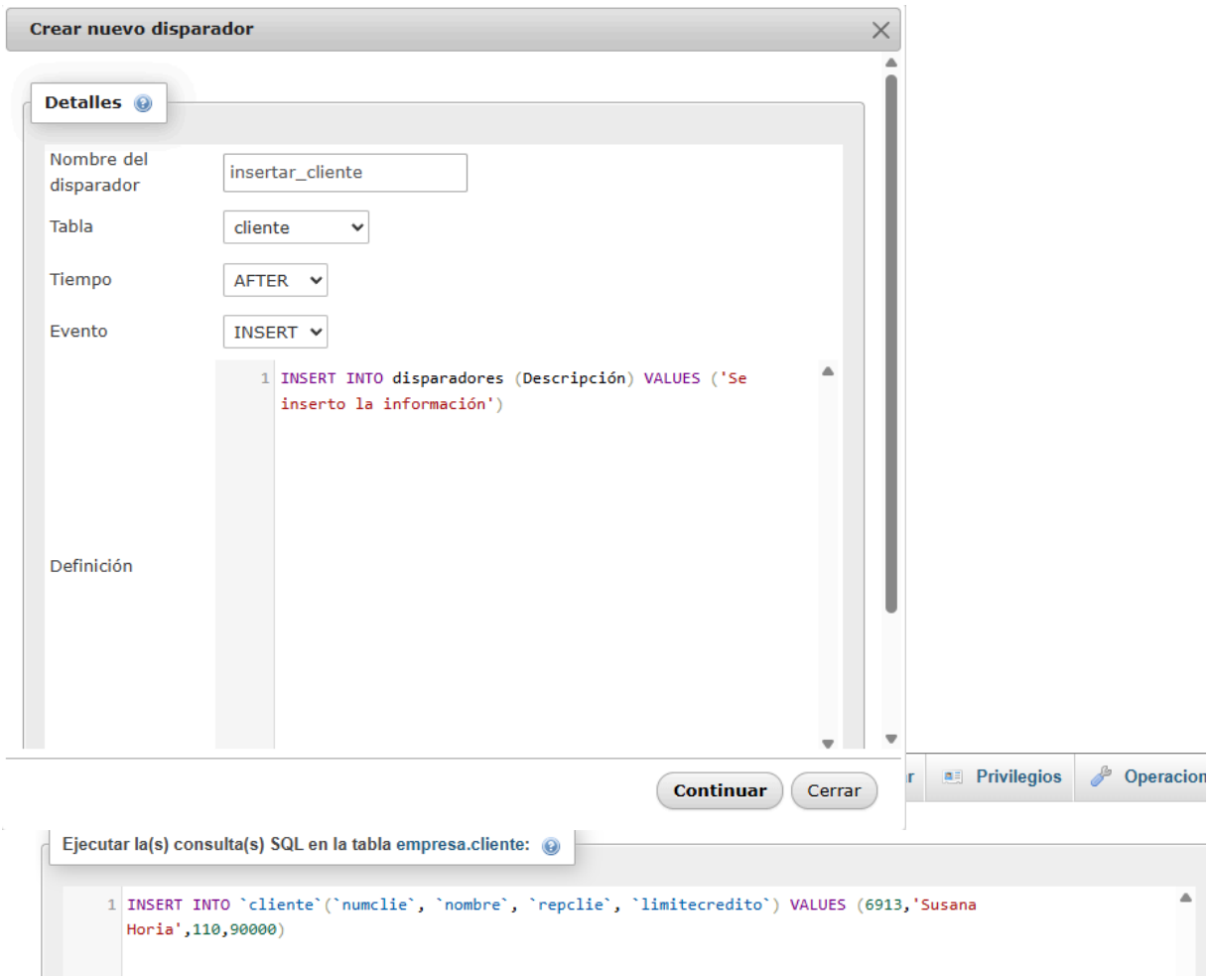
CLIENTE



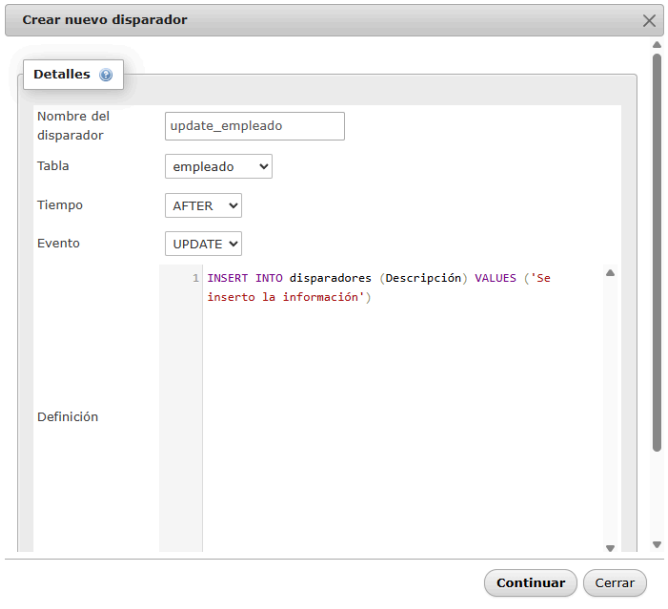
6. Pegue los pantallazos en un documento de Word.
7. Tome pantallazos al código de creación de cada disparador y péguelos en el documento que venía trabajando.

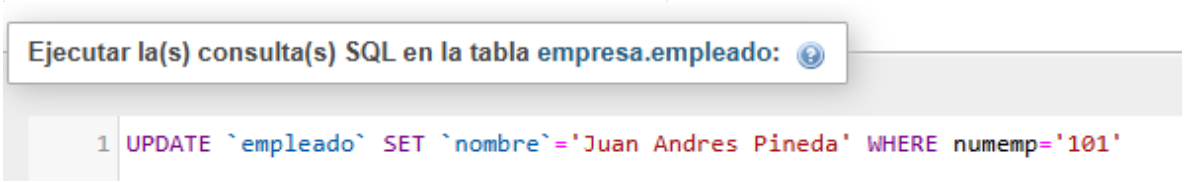

Estructura
SQL
Buscar
Generar una consulta
Ejecutar la(s) consulta(s) SQL en la base de datos empresa:
CREATE TABLE Disparadores (
ID int(7) NOT NULL AUTO_INCREMENT,
Descripción varchar(200) NOT NULL,
Fecha_Hora datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(ID)
);

TRIGGER 1

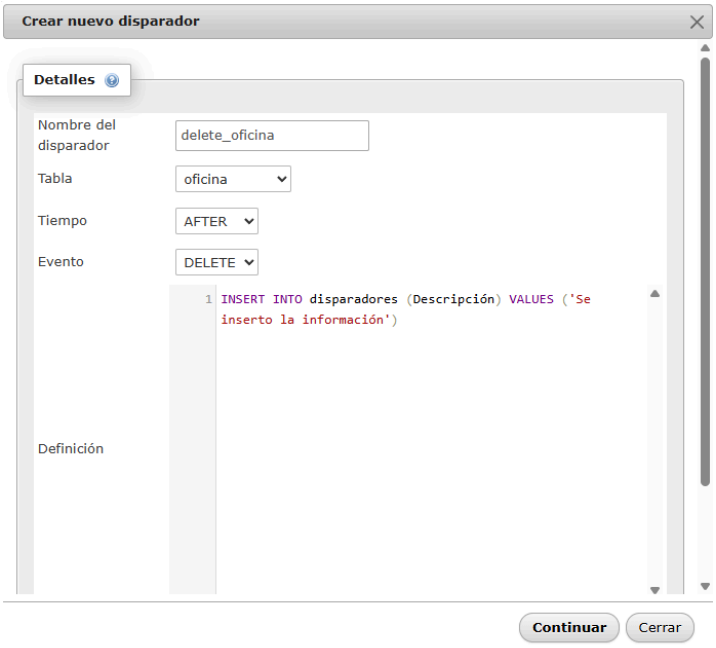

Crear nuevo disparador
Detalles
Nombre del disparador
insertar_cliente
Tabla
cliente
Tiempo
AFTER
Evento
INSERT
INSERT INTO disparadores (Descripción) VALUES ('Se inserto la información')
Definición
Continuar
Cerrar
Privilegios
Operacion
Ejecutar la(s) consulta(s) SQL en la tabla empresa.cliente:
INSERT INTO `cliente`(`numclie`, `nombre`, `repclie`, `limitecredito`) VALUES (6913,'Susana Horia',110,90000)

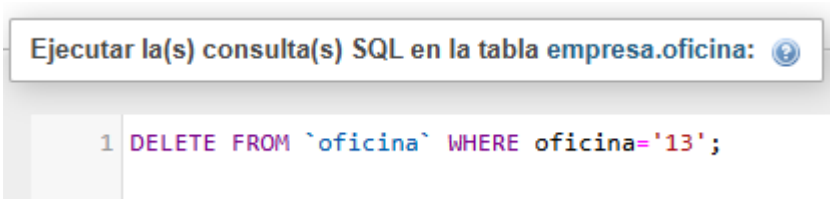
TRIGGER 2



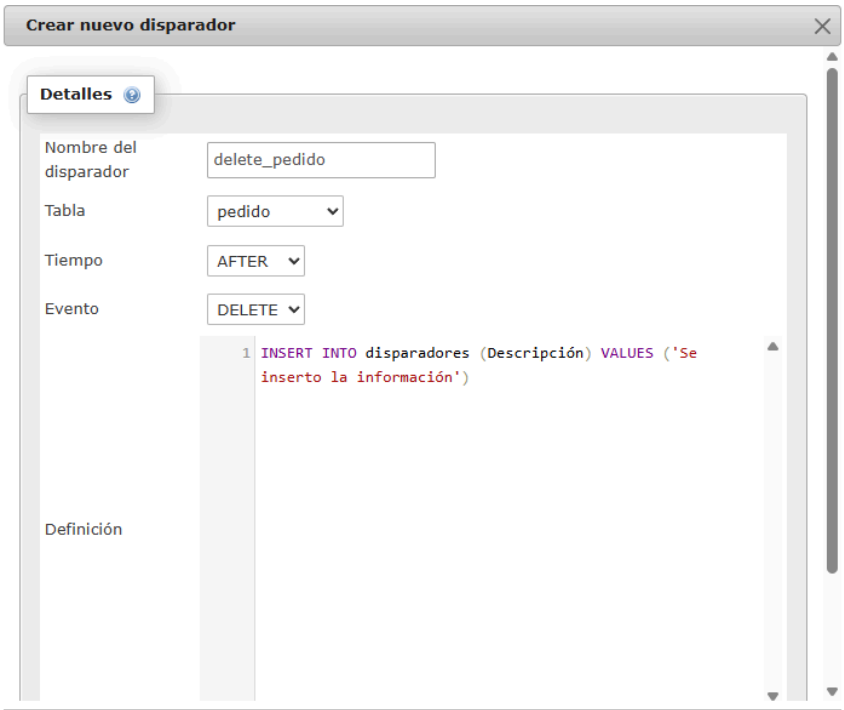


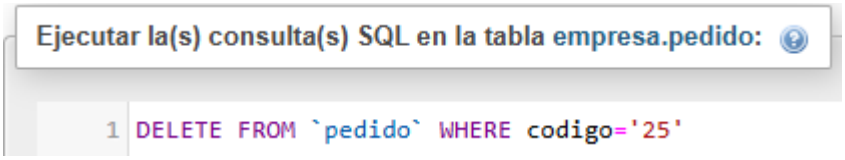
TRIGGER 3



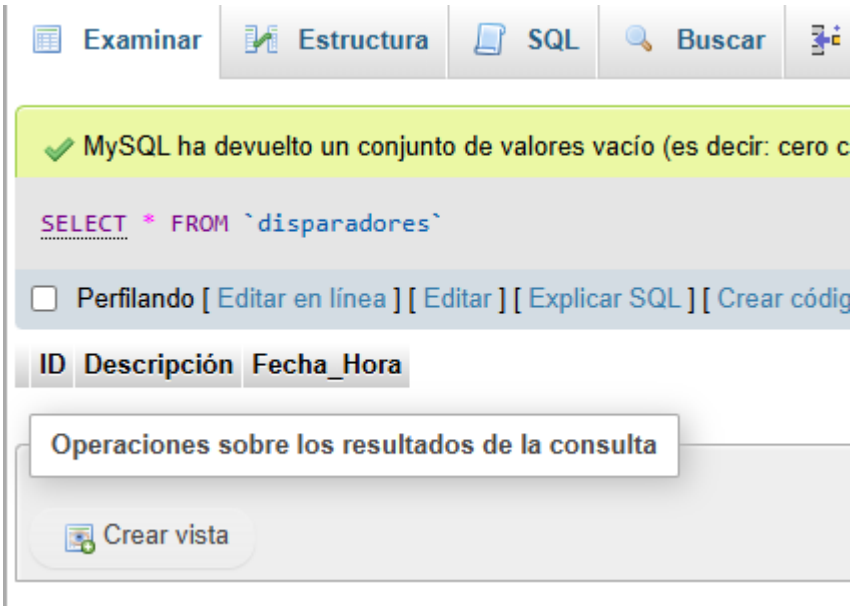


TRIGGER 4

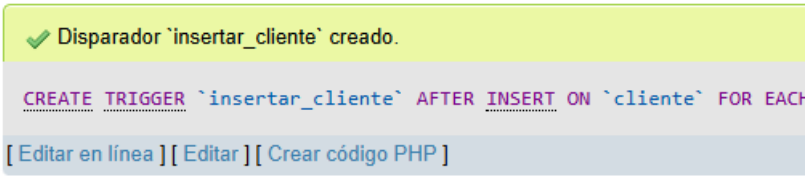




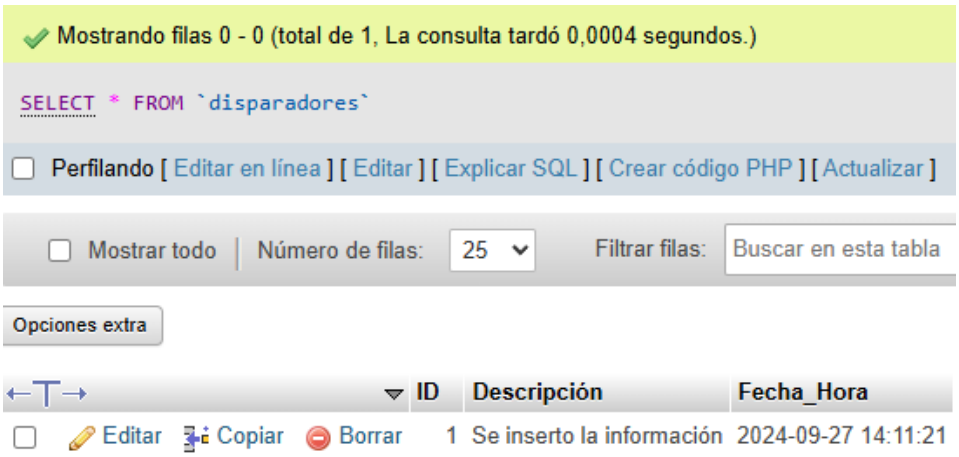
8. Tome Pantallazo del resultado de cada trigger


Examinar
Estructura
SQL
Buscar
MySQL ha devuelto un conjunto de valores vacío (es decir: cero c
SELECT * FROM `disparadores`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig
ID Descripción Fecha_Hora
Operaciones sobre los resultados de la consulta
Crear vista

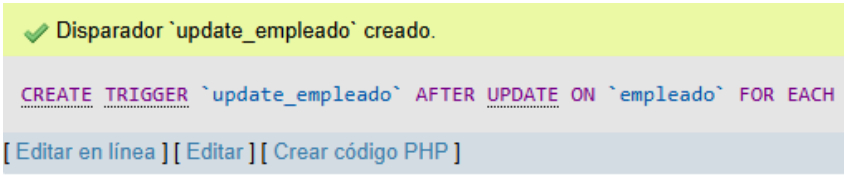
TRIGGER 1


Disparador `insertar_cliente` creado.
CREATE TRIGGER `insertar_cliente` AFTER INSERT ON `cliente` FOR EACH
[Editar en línea] [Editar] [Crear código PHP]


Disparadores
Seleccionar todo
Exportar
Eliminar
Nombre Tiempo Evento
insertar_cliente AFTER INSERT Editar Exportar Eliminar

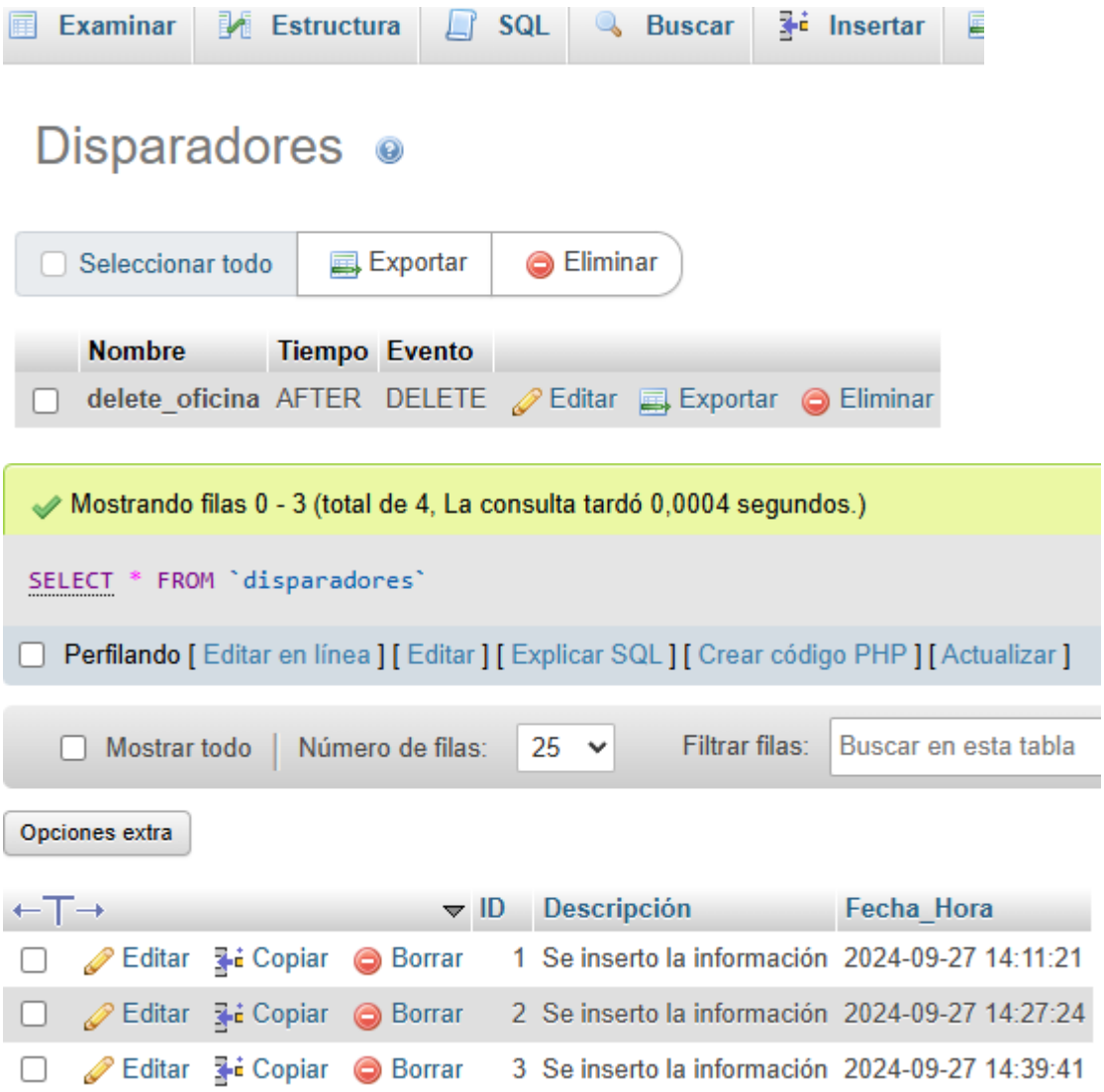

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0004 segundos.)
SELECT * FROM `disparadores`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Opciones extra
ID
Descripción
Fecha_Hora
Editar
Copiar
Borrar
1
Se inserto la información
2024-09-27 14:11:21

TRIGGER 2


Disparador `update_empleado` creado.
CREATE TRIGGER `update_empleado` AFTER UPDATE ON `empleado` FOR EACH
[Editar en línea] [Editar] [Crear código PHP]


Disparadores
Seleccionar todo
Exportar
Eliminar
Nombre
Tiempo
Evento
update_empleado
AFTER
UPDATE
Editar
Exportar
Eliminar
Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0004 segundos.)
SELECT * FROM `disparadores`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Opciones extra
ID
Descripción
Fecha_Hora
Editar
Copiar
Borrar
1
Se inserto la información
2024-09-27 14:11:21
Editar
Copiar
Borrar
2
Se inserto la información
2024-09-27 14:27:24

TRIGGER 3



TRIGGER 4

Disparador `delete_pedido` creado.

```
CREATE TRIGGER `delete_pedido` AFTER DELETE ON `pedido` FOR EACH R
```

[Editar en línea] [Editar] [Crear código PHP]

Disparadores

Seleccionar todo | Exportar | Eliminar

	Nombre	Tiempo	Evento			
☐	delete_pedido	AFTER	DELETE	Editar	Exportar	Eliminar

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0004 segundos.)

```
SELECT * FROM `disparadores`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

← T →		ID	Descripción	Fecha_Hora
☐	Editar Copiar Borrar	1	Se inserto la información	2024-09-27 14:11:21
☐	Editar Copiar Borrar	2	Se inserto la información	2024-09-27 14:27:24
☐	Editar Copiar Borrar	3	Se inserto la información	2024-09-27 14:39:41
☐	Editar Copiar Borrar	4	Se inserto la información	2024-09-27 14:46:02