Nombre de la práctica	Triggers			No.	3
Asignatura:	Taller de bases de datos	Carrera:	ISIC	Duración de la práctica (Hrs)	

I. Nombres:
Jocelin Reyes Rodriguez
Shania Kinnereth Diaz Moya
Jesus Silvestre Santiago Cruz

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado:
Computadora Navicat

IV. Desarrollo de la práctica:

Creación de la base de datos:

```
-- 1.- Creacion y uso de la base de datos
CREATE DATABASE IF NOT EXISTS punto_venta;
USE punto_venta;
```

```
-- Tabla de Clientes
CREATE TABLE clientes (
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100) UNIQUE,
    fecha_registro DATE
);

-- Tabla de Productos
CREATE TABLE productos (
    id_producto INT AUTO_INCREMENT PRIMARY KEY,
    nombre_producto VARCHAR(100) NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    stock INT DEFAULT 0
);

-- Tabla de Ventas
CREATE TABLE ventas (
    id_venta INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    fecha_venta DATE,
    total DECIMAL(10,2),
    FOREIGN KEY (id_cliente) REFERENCES clientes(
id_cliente)
);
```

```sql
-- Tabla de Tarjeta de Puntos
CREATE TABLE tarjeta_puntos (
    id_tarjeta INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    puntos_acumulados INT DEFAULT 0,
    fecha_ultima_actualizacion DATE,
    FOREIGN KEY (id_cliente) REFERENCES clientes(
id_cliente)
);

-- Tabla de Historial de Cambios
CREATE TABLE historial_cambios (
    id_cambio INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    columna_modificada VARCHAR(50),
    valor_anterior VARCHAR(100),
    valor_nuevo VARCHAR(100),
    fecha_cambio DATE,
    FOREIGN KEY (id_cliente) REFERENCES clientes(
id_cliente)
);
```

```
-- 1. Trigger para Actualizar Nombre de Cliente
DELIMITER //
CREATE TRIGGER TriggerActualizarClienteNombre
AFTER UPDATE ON clientes
FOR EACH ROW
BEGIN
    -- Registro del cambio de nombre del cliente
    IF OLD.nombre <> NEW.nombre THEN
        INSERT INTO historial_cambios (id_cliente
, columna_modificada, valor_anterior, valor_nuevo
, fecha_cambio)
        VALUES (NEW.id_cliente, 'nombre', OLD.
nombre, NEW.nombre, NOW());
    END IF;
END //
DELIMITER ;
```

id_cambio	id_cliente	columna_modificada	valor_anterior	valor_nuevo	fecha_cambio
37	1	0xbSingpu3	7xE9L2fbwP	cFm2NrlWDN	2003-03-28
201	1	nombre	Ethel Lopez	NuevoNombre	2025-01-06

```
DELIMITER //
CREATE TRIGGER TriggerActualizarClienteCorreo
AFTER UPDATE ON clientes
FOR EACH ROW
BEGIN
    -- Registro del cambio de correo del cliente
    IF OLD.correo <> NEW.correo THEN
        INSERT INTO historial_cambios (id_cliente, columna_modificada, valor_anterior, valor_nuevo, fecha_cambio)
        VALUES (NEW.id_cliente, 'correo', OLD.correo, NEW.correo, NOW());
    END IF;
END //
DELIMITER ;
```

id_cambio	id_cliente	columna_modificada	valor_anterior	valor_nuevo	fecha_cambio
37	1	0xbSingpu3	7xE9L2fbwP	cFm2NrlWDN	2003-03-28
201	1	nombre	Ethel Lopez	NuevoNombre	2025-01-06
202	1	correo	ethellopez@gmail.	nuevo.correo@ex	2025-01-06

```
-- 3. Trigger para Validar Precio de Producto (Insert)
DELIMITER //
CREATE TRIGGER TriggerValidarPrecioProducto
BEFORE INSERT ON productos
FOR EACH ROW
BEGIN
    IF NEW.precio < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El precio del producto no puede ser negativo.';
    END IF;
END //
DELIMITER ;
```

```
INSERT INTO productos (nombre_producto, precio, stock)
VALUES ('Producto Ejemplo', -10, 9);
```

Query	Message	Query Time	Fetch Time
INSERT INTO productos (nombre_producto, precio, stock) VALUES ('Producto Ejemplo', -10, 9)	1644 - El precio del producto no puede ser negativo.	0.000s	0.000s

```
-- 4. Trigger para Validar Stock de Producto (Insert)
DELIMITER //
CREATE TRIGGER TriggerValidarStockProducto
BEFORE INSERT ON productos
FOR EACH ROW
BEGIN
    IF NEW.stock < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El stock del producto no puede ser negativo.';
    END IF;
END //
DELIMITER ;
```

```
INSERT INTO productos (nombre_producto, precio, stock)
VALUES ('Producto Ejemplo', 10, -89);
```

Query	Message	Query Time	Fetch Time
INSERT INTO productos (nombre_producto, precio, stock) VALUES ('Producto', 10, -89)	1644 - El stock del producto no puede ser negativo.	0.000s	0.000s

```
-- 5. Trigger para Eliminar Cliente (Delete)
DELIMITER //
CREATE TRIGGER TriggerEliminarCliente
AFTER DELETE ON clientes
FOR EACH ROW
BEGIN
    INSERT INTO historial_cambios (id_cliente, columna_modificada, valor_anterior, valor_nuevo, fecha_cambio)
    VALUES (OLD.id_cliente, 'cliente', OLD.nombre, 'Cliente eliminado', NOW());
END //
DELIMITER ;
```

```
DELETE FROM clientes WHERE id_cliente = 8;

SELECT * FROM historial_cambios WHERE id_cliente = 8;
```

id_cambio	id_cliente	columna_modificada	valor_anterior	valor_nuevo	fecha_cambio
26	8	zth7ErSpQZ	bb2e7SAfau	OeisdNzLWd	2008-01-09
204	8	cliente	Dennis Wilson	Cliente eliminado	2025-01-06

```
DELIMITER //
CREATE TRIGGER TriggerEliminarProducto
AFTER DELETE ON productos
FOR EACH ROW
BEGIN
    INSERT INTO historial_cambios (id_cliente, columna_modificada, valor_anterior, valor_nuevo, fecha_cambio)
    VALUES (NULL, 'producto', OLD.nombre_producto, 'Producto eliminado', NOW());
END //
DELIMITER ;
```

```
DELETE FROM productos WHERE id_producto = 1;
```

id_cambio	id_cliente	columna_modificada	valor_anterior	valor_nuevo	fecha_cambio
205	(Null)	producto	Giape	Producto eliminac	2025-01-06

V. Conclusiones:

Los triggers automatizan acciones específicas en la base de datos en respuesta a ciertos eventos (inserciones, actualizaciones o eliminaciones), lo que reduce la necesidad de intervención manual y asegura consistencia, ayudan a mantener la integridad referencial y de datos al garantizar que ciertas reglas o restricciones se apliquen automáticamente cuando se realizan operaciones en la base de datos, son útiles para auditar y registrar cambios en los datos, lo que permite mantener un historial de modificaciones, crucial para el control de versiones y recuperación de errores. Permiten encapsular la lógica de negocio directamente en la base de datos, lo que puede simplificar la aplicación cliente al delegar responsabilidades a la base de datos, el uso excesivo o ineficiente de triggers puede afectar el rendimiento de la base de datos, ya que se ejecutan automáticamente y pueden introducir sobrecarga si no se gestionan adecuadamente.