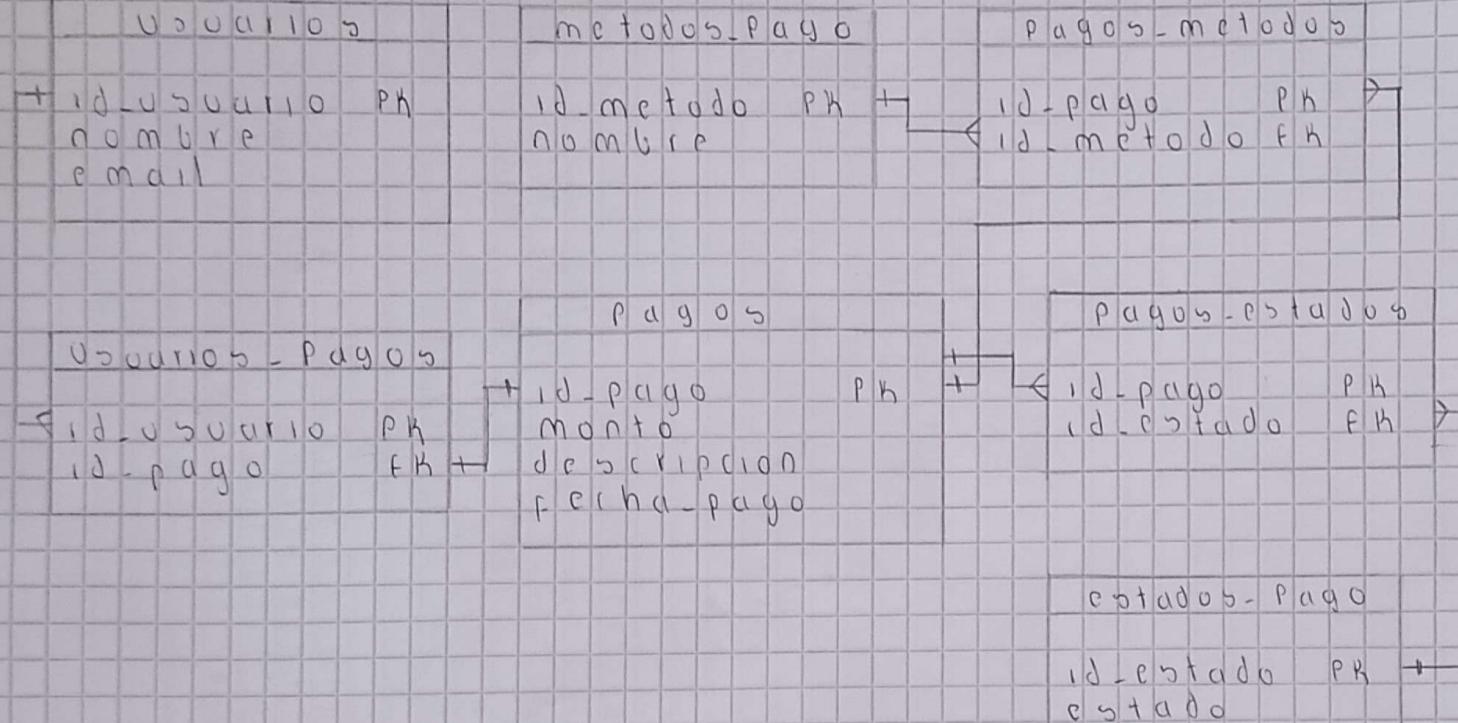


MEM - 1



CREATE DATABASE IF NOT EXISTS PagarelaPagos;  
USE PagarelaPagos;

DELIMITER //

CREATE PROCEDURE CrearTablaUsuarios()  
BEGIN  
DROP TABLE IF EXISTS Usuarios;  
CREATE TABLE Usuarios (  
id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100) NOT NULL,  
email VARCHAR(100) NOT NULL UNIQUE  
);  
  
INSERT INTO Usuarios (nombre, email) VALUES  
('Ana Torres', 'ana@email.com'),  
('Juan Perez', 'juan@email.com'),  
('Maria Lopez', 'maria@email.com'),  
('Carlos Garcia', 'carlos@email.com'),  
('Laura Martinez', 'laura@email.com'),  
('Pedro Gomez', 'pedro@email.com'),  
('Sofia Hernandez', 'sofia@email.com'),  
('Miguel Torres', 'Miguel@email.com'),  
('Elena Ruiz', 'Elena@email.com'),  
('Kevin Culma', 'Kevin@email.com');  
END;  
//

CREATE PROCEDURE CrearTablaMetodoPago()

BEGIN  
DROP TABLE IF EXISTS Metodo\_Pago;  
CREATE TABLE Metodo\_Pago (  
id\_metodo INT AUTO\_INCREMENT PRIMARY KEY,  
nombre VARCHAR(50) NOT NULL UNIQUE  
);  
  
INSERT INTO Metodo\_Pago (nombre) VALUES  
('Tarjeta Credito'), ('Tarjeta Debito'), ('Transferencia  
Bancaria'), ('PSE'), ('PayPal'), ('Neguri'), ('Daviplata'),  
('Cripto'), ('Contraentrega'), ('Bono de Mlapago');  
END;  
//

CREATE PROCEDURE CrearTablaEstadosPago()

BEGIN  
DROP TABLE IF EXISTS Estados\_Pago;  
CREATE TABLE Estados\_Pago (

```

    id-estado INT AUTO_INCREMENT PRIMARY KEY,
    estado VARCHAR(50) NOT NULL UNIQUE
);

INSERT INTO Estados_Pago (estado) VALUES
('Pendiente'), ('Pagado'), ('Fallido'), ('En proceso'),
('Meembolsado'), ('Cancelado'), ('Expirado'), ('Revisando'),
('Autenticado'), ('Rechazado');

END;
//



CREATE PROCEDURE CrearTablaPagos()
BEGIN
DROP TABLE IF EXISTS Pagos;
CREATE TABLE Pagos(
    id-pago INT AUTO_INCREMENT PRIMARY KEY,
    monto DECIMAL(10,2) NOT NULL,
    descripcion VARCHAR(200),
    fecha_pago DATETIME NOT NULL
);

INSERT INTO Pagos (monto, descripcion, fecha_pago) VALUES
(120000, 'Compra en linea', '2025-09-16 14:00'),
(80000, 'Suscripcion mensual', '2025-09-17 10:00'),
(55000, 'Factura de servicios', '2025-09-18 08:30'),
(99000, 'Renovacion membresia', '2025-09-19 09:00'),
(320000, 'Compra electrodomestico', '2025-09-19 16:00'),
(25000, 'Pago curso online', '2025-09-21 11:00'),
(45000, 'Recargar celular', '2025-09-21 15:00'),
(75000, 'Compra ropa', '2025-09-22 13:00'),
(38000, 'Suscripcion app', '2025-09-23 18:00'),
(67000, 'Compra libros', '2025-09-23 20:00');
END;
//


CREATE PROCEDURE CrearTablaUsuarios_Pagos()
BEGIN
DROP TABLE IF EXISTS Usuarios_Pagos;
CREATE TABLE Usuarios_Pagos(
    id-usuario INT NOT NULL,
    id-pago INT NOT NULL,
    PRIMARY KEY (id-usuario, id-pago),
    FOREIGN KEY (id-usuario) REFERENCES Usuarios (id-usuario),
    FOREIGN KEY (id-pago) REFERENCES Pagos (id-pago)
);

```

```
INSEMT INTO Usuarios_Pagos (id_usuario, id_pago) VALUES  
 (1,1), (2,2), (3,3), (4,4), (5,5)  
 (6,6), (7,7), (8,8), (9,9), (10,10);  
END;  
//
```

```
CREATE PROCEDURE CrearTablaPagosMetodos()  
BEGIN  
DROP TABLE IF EXISTS Pagos_Metodos;  
CREATE TABLE Pagos_Metodos(  
    id_pago INT NOT NULL,  
    id_metodo INT NOT NULL,  
    PRIMARY KEY (id_pago, id_metodo),  
    FOREIGN KEY (id_pago) REFERENCES Pagos (id_pago),  
    FOREIGN KEY (id_metodo) REFERENCES Metodos_Pago  
        (id_metodo)  
);  
INSEMT INTO Pagos_Metodos (id_pago, id_metodo)  
VALUES  
 (9,1), (7,2), (3,3), (4,4), (5,5), (6,6), (1,7), (8,8),  
 (9,9), (10,10);  
END;  
//
```

```
CREATE TABLE PROCEDURE CrearTablaPagosEstados()  
BEGIN
```

```
DROP TABLE IF EXISTS Pagos_Estados;  
CREATE TABLE Pagos_Estados(  
    id_pago INT NOT NULL,  
    id_estado INT NOT NULL,  
    PRIMARY KEY (id_pago, id_estado),  
    FOREIGN KEY (id_pago) REFERENCES Pagos (id_pago),  
    FOREIGN KEY (id_estado) REFERENCES Estados_Pago  
        (id_estado)  
);
```

```
INSEMT INTO Pagos_Estados (id_pago, id_estado) VALUES  
 (1,2), (2,2), (3,1), (4,2), (5,2),  
 (6,2), (7,1), (8,2), (9,3), (10,4);
```

```
END;
```

```
//
```

```
DELIMITER;
```

```
CALL CrearTablaUsuarios();
CALL CrearTablaMetodosPago();
CALL CrearTablaEstadosPago();
CALL CrearTablaPagos();
CALL CrearTablaUsuariosPagos();
CALL CrearTablaPagosMetodos();
CALL CrearTablaPagosEstados();
```

```
CREATE OR REPLACE VIEW Vista_PagosPorUsuario AS
SELECT u.nombre AS Usuario, p.monto, p.descripcion, p.fecha_pago
FROM Pagos p
JOIN Usuarios_Pagos up ON up.id_pago = p.id_pago
JOIN Usuarios u ON up.id_usuario = u.id_usuario;
```

```
CREATE OR REPLACE VIEW Vista_PagosPorMetodo AS
SELECT m.nombre AS Metodo, COUNT(pm.id_pago) AS total
FROM Metodos_Pago m
LEFT JOIN Pagos_Metodos pm ON m.id_metodo = pm.id_metodo
GROUP BY m.id_metodo;
```

```
CREATE OR REPLACE VIEW Vista_PagosPorEstado AS
SELECT e.estado, COUNT(pe.id_pago) AS total
FROM Estados_Pago e
LEFT JOIN Pagos_Estados pe ON e.id_estado = pe.id_estado
GROUP BY e.id_estado;
```

```
CREATE OR REPLACE VIEW Vista_UsuariosConPagos AS
SELECT u.nombre AS Usuario, COUNT(up.id_pago) AS totalPagos
FROM Usuarios u
LEFT JOIN Usuarios_Pagos up ON u.id_usuario = up.id_usuario
GROUP BY u.id_usuario;
```

```
CREATE OR REPLACE VIEW Vista_PagosDetallados AS
SELECT p.id_pago, u.nombre AS Usuario, p.monto, m.nombre
AS Metodo, e.estado, p.fecha_pago
FROM Pagos p
JOIN Usuarios_Pagos up ON up.id_pago = p.id_pago
JOIN Usuarios u ON up.id_usuario = u.id_usuario
JOIN Pagos_Metodos pm ON pm.id_pago = p.id_pago
JOIN Metodos_Pago m ON pm.id_metodo = m.id_metodo
JOIN Pagos_Estados pe ON pe.id_pago = p.id_pago
JOIN Estados_Pago e ON pe.id_estado = e.id_estado,
```

```
SELECT * FROM Vista_PagosPorUsuario;
SELECT * FROM Vista_PagosPorMetodo;
SELECT * FROM Vista_PagosPorEstado;
SELECT * FROM Vista_UsuariosConPagos;
SELECT * FROM Vista_PagosDetallados;
```

## DICCIONARIO DE DATOS MEM - 7

### Tabla: Usuario

Campo	Tipo	Largo	Descripción	Restricciones
id_usuario	INT	-	Identificador único del usuario	PK, AUTOINCREMENT
nombre	VARCHAR	100	Nombre completo del usuario	NOT NULL
email	VARCHAR	100	Correo electrónico del usuario	NOT NULL, UNIQUE

### Tabla: Metodos\_Pago

Campo	Tipo	Largo	Descripción	Restricciones
id_metodo	INT	-	Identificador único del método de pago	PK, AUTOINCREMENT
nombre	VARCHAR	50	Nombre del método de pago	NOT NULL, UNIQUE

### Tabla: Estados\_Pago

Campo	Tipo	Largo	Descripción	Restricciones
id_estado	INT	-	Identificador único del estado del pago	PK, AUTOINCREMENT
estado	VARCHAR	50	Estado en que se encuentra el pago	NOT NULL, UNIQUE

### Tabla: Pagos

Campo	Tipo	Longitud	Descripción	Restricciones
id-pago	INT	-	Identificador en PK, AUTO-INCREMENTO del pago	
monto	DECIMAL	10,2	Valor del pago	NOT NULL
descripcion	NVARCHAR	200	Descripción del pago	NULLABLE
fecha_pago	DATETIME	-	Fecha y hora del pago	NOT NULL

### Tabla: Usuarios\_Pagos

Campo	Tipo	Longitud	Descripción	Restricciones
id-usuario	INT	-	Identificador único del usuario que realizó el pago	PK(id-pago), FK-> Usuarios.id-usuario
id-pago	INT	-	Identificador del pago	PK(id-usuario), FK-> Pagos.id-pago

### Tabla: Pagos-Métodos

Campo	Tipo	Longitud	Descripción	Restricciones
id-pago	INT	-	Identificador del pago	PK(id-metodo), FK-> Pagos.id-pago
id-metodo	INT	-	Identificador del método de pago	PK(id-pago), FK-> Metodos_Pago.id-metodo

## Tabla: Pagos-Estados

Campo	Tipo	Longitud	Descripción	Restricciones
id-pago	INT	-	Identificador único PK (id-estado), fk co del pago	Pagos, id-pago
id-estado	INT	-	Identificador del FK (id-pago), fk estado del pago	Estados-Pago, id-estado