SCRIPT COMPLETO - HOSPITAL

```
CREATE DATABASE HospitalDB;
GO
USE HospitalDB;
GO

CREATE PROCEDURE sp_crear_tabla_pacientes
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE nome='Paciente')
  BEGIN
    CREATE TABLE pacientes(
PacienteID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
fechaNacimiento DATE NOT NULL,
Genero CHAR(1) CHECK (Genero IN (M, F)),
Telefono VARCHAR (20) UNIQUE
)
  END
END;
GO
EXEC SP_crear_tabla_pacientes;
GO

CREATE PROCEDURE sp_crear_tabla_doctores
AS
BEGIN
  CREATE TABLE Doctores (
DoctorID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Especialidad VARCHAR (100) NOT NULL,
Email VARCHAR (100) UNIQUE
)
  END
END;
GO
EXEC sp_crear_tabla_doctores;
GO

CREATE PROCEDURE sp_crear_tabla_habitaciones
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Habitaciones')
  BEGIN
CREATE TABLE Habitaciones (
  HabitacionID PRIMARY KEY IDENTITY,
Numero INT UNIQUE NOT NULL,
Tipo VARCHAR (50) CHECK (Tipo IN ('General', 'UCI', 'Privado')),
Estado VARCHAR (20) CHECK (Estado IN ('Disponible', 'ocupada'))
)
END
  END;
GO
EXEC sp_crear_tabla_habitaciones;
```

```
GO

CREATE TABLE citas (
citaID INT PRIMARY KEY IDENTITY,
PacienteID INT FOREIGN KEY REFERENCES paciente (pacienteID),
DoctorID INT FOREIGN KEY REFERENCES Doctores (DoctorID),
fechacita DATETIME NOT NULL,
Estado_VARCHAR (20) CHECK (Estado IN ('Pendiente' 'Realizada')
)
  END
END;
GO
EXEC sp_crear_tabla_citas;
GO

CREATE PROCEDURE sp_Crear_tabla_hospitalizaciones
AS
BEGIN
  If NOT EXIST (SELECT * FROM sysobjects WHERE name=hospitalizaciones)
BEGIN
  CREATE TABLE Hospitalizaciones (
HospID INT PRIMARY KEY IDENTITY,
PacienteID INT FOREIGN KEY REFERENCES pacientes (pacienteID),
HabitacionID INT FOREIGN KEY REFERENCES Habitaciones (habitaciones)
fechaIngreso DATE NOT NULL,
fechaAlta DATE Null.
)
  END
END;
  GO

CREATE PROCEDURE sp-Crear_tabla_medicamentos
AS
BEGIN
  If NOT EXISTS (SELECT * FROM sysobjects WHERE name=Medicamentos
BEGIN
  CREATE TABLE Medicamentos (
MedicamentoID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (100) NOT NULL,
stock INT CHECK (stock >=0),
Precio DECIMAL (10,2) CHECK (precio >=0)
)
  END
END;
  GO
EXEC sp-crear-tabla-medicamentos;
GO

CREATE PROCEDURE sp-Insertar-paciente
@ Nombre VARCHAR (50),
@ Apellido VARCHAR (50),
@ fechaNacimiento DATE,
@ Genero CHAR (1),
@ Telefono VARCHAR (20)
AS
  BEGIN
If EXISTS (SELECT 1 FROM Pacientes WHERE Telefono = @ Telefono)
  BEGIN
```

```
PRINT 'El Pacrente ya está registro con este teléfono.';
  RETURN;
END
  INSERT INTO Pacrente (Nombre, Apellido, fecha Nacimiento)
  VALUES (@Nombre, @Apellido, @fecha Nacimiento, @Genero, @Teléfono)
  END;
GO

CREATE PROCEDURE sp-insertar-doctor
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Especialidad VARCHAR (100),
@Email VARCHAR (100)
AS
BEGIN
  If EXISTS (SELECT 1 FROM Doctores WHERE Email = @Email)
BEGIN
  PRINT 'Doctor ya registrado con ese email';
RETURN;
END
END
GO

CREATE PROCEDURE sp-insertar_habitación
@Numero INT,
@Tipo VARCHAR (50),
@Estado VARCHAR (20) = 'Disponible'
AS
  BEGIN
If EXISTS (SELECT 1 FROM Habitaciones WHERE Numero = @Numero)
  BEGIN
PRINT 'Ya existe esa habitación';
RETURN;
END
  INSERT INTO Habitaciones (Numero, tipo, Estado)
VALUES (@Numero, @Tipo, @Estado)
  END;
GO

CREATE PROCEDURE sp-insertar_cita
@PacrenteID INT,
@Doctor INT,
@fechacita DATETIME
AS
BEGIN

 If NOT EXISTS (SELECT 1 FROM Pacientes WAERE PacrenteID)
BEGIN
  PRINT 'Paciente no existe';
  RETURN;
END
If NOT EXISTS (SELECT 1 FROM Doctores WHERE Doctor)
  BEGIN
PRINT 'Doctor no existe';
RETURN;
  END
INSER INTO citas (PacienteID, Doctor, fechaCita)
VALUES (@PacienteID, @Doctor ID, @fechaCita)
```

```
END
 INSERT INTO citas (PacienteID, DoctorID, fechacita)
 VALUES (@PacienteID, @DoctorID, @fechacita)
END;
GO

CREATE PROCEDURE sp_insertar_tratamiento
@Nombre VARCHAR (100),
@Costo DECIMAL (10,2)
AS
  BEGIN
 INSERT INTO Tratamientos (Nombre, costo)
END;
GO

CREATE PROCEDURE sp_insertar_medicamento
@Nombre VARCHAR (100),
@Stock INT,
@Precio DECIMAL (10,2)
AS
  BEGIN
INSERT INTO Medicamentos (Nombres, Stock, precio)
VALUES (@Nombre, @Stock, @precio)
END;
GO

EXEC sp_Insertar_Paciente 'Juan', 'Perez', 1985-05-10, 'M', '3001111111';
EXEC sp_Insertar_Paciente 'Ana', Gómez, 1990-07-20, 'F', 3002222222;
EXEC sp_Insertar_paciente 'Luis', Martinez, 1978-03-15, 'M' 3003333333;
EXEC sp_Insertar_paciente 'Maria', lopez, 1982-12-01, 'F' 3004444444;
EXEC sp_Insertar_paciente 'Pedro', Ramirez, 1995-09-09 'M' 3005555555;
EXEC sp_Insertar_Paciente 'Lucia', fernandez, 1987-11-30, 'F' 3006666666;
EXEC sp_Insertar_Paciente 'Carlos', Torres 2000-06-22, 'M' 3007777777;
EXEC sp_Insertar_paciente 'laura', Jiménez 1992-04-18, 'F' 3008888888;
EXEC sp_Insertar_Paciente 'Diego', Castro 1989-08-05 'M' 3009999999;
EXEC sp_Insertar_paciente 'Rosa', Hernandez 1993-10-14 'F' 3010000000;

EXEC sp_Insertar_doctor 'Andres', Santos', Cardiologia, andres@hospital.com';
EXEC sp_Insertar_doctor 'Beatriz', Mora', Pediatria, beatriz@hospital.com';
EXEC sp_Insertar_doctor 'Camilo', Pardo', Cirugia, Camilo@hospital.com';
EXEC sp_Insertar_doctor 'Daniela', Vargas', Neurologia, daniela@hospital.com';
EXEC sp_Insertar_doctor 'Esteban', Suarez', Medicina Interna, esteban@hospital.com';
EXEC sp_Insertar_doctor 'Felipe', Lopez', Traumatologia, felipe@hospital.com';
EXEC sp_Insertar_doctor 'Gabriela', Moreno', oncologia, gabriela@hospital.com';
EXEC sp_Insertar_doctor 'Hector', Silva', Dermatologia, hector@hospital.com';
EXEC sp_Insertar_doctor 'Isabel', Cruz', Psiquiatria, Isabel@hospital.com';
EXEC sp_Insertar_doctor 'Jorge', Castillo', Ginecologia, Jorge@hospital.com';

EXEC sp_Insertar_habitacion 101, 'General' Disponible;
EXEC sp_Insertar_habitacion 102, 'UCI' Disponible;
EXEC sp_Insertar_habitacion 103, 'Privada' Disponible;
EXEC sp_Insertar_habitacion 104, 'General' Disponible;
EXEC sp_Insertar_habitacion 105, 'privada' Disponible;
EXEC sp_Insertar_habitacion 106, 'General' Disponible;
EXEC sp_Insertar_habitacion 107, 'UCI' Disponible;
EXEC sp_Insertar_habitacion 108, 'General' Disponible;
EXEC sp_Insertar_habitacion 109, 'Privada' Disponible;
EXEC sp_Insertar_habitacion 110, 'General' Disponible;
```

```
EXEC sp_Insertar_medicamento 'Paracetamol', 500, 2000;
EXEC sp_Insertar_medicamento 'Ibuprofeno' 300, 2500;
EXEC sp_Insertar_medicamento 'Amoxicilina' 200, 5000;
EXEC sp_Insertar_medicamento 'Omeprazol' 400, 3500;
EXEC sp_Insertar_medicamento 'Metformina' 250, 6000;
EXEC sp_Insertar_medicamento 'Losartan' 180, 7000;
EXEC sp_Insertar_medicamento 'Atorvastatina' 220, 8000;
EXEC sp_Insertar_medicamento 'Insulina' 150, 20000;
EXEC sp_Insertar_medicamento 'Clonazepam' 100 12000;
EXEC sp_Insertar_medicamento 'Vitamina C' 600, 1500;

EXEC sp_Insertar_cita 1,1, 2025-09-15 09:00;
EXEC sp_Insertar_cita 2,2, 2025-09-16 10:30;
EXEC sp_Insertar_cita 3,3, 2025-09-17 14:00;
EXEC sp_Insertar_cita 4,4, 2025-09-18 08:00;
EXEC sp_Insertar_cita 5,5, 2025-09-18 15:00;
EXEC sp_Insertar_cita 6,6, 2025-09-19 09:30;
EXEC sp_Insertar_cita 7,7, 2025-09-19 11:00;
EXEC sp_Insertar_cita 8,8, 2025-09-20 13:00;
EXEC sp_Insertar_cita 9,9, 2025-09-21 10:00;
EXEC sp_Insertar_cita 10,10 2025-01-22 16:00;

EXEC sp_Insertar_hospitalizacion 1,1, 2025-09-01;
EXEC sp_Insertar_hospitalizacion 2,2, 2025-09-02;
EXEC sp_Insertar_hospitalizacion 3,3, 2025-09-03;
EXEC sp_Insertar_hospitalizacion 4,4, 2025-09-04;
EXEC sp_Insertar_hospitalizacion 5,5, 2025-09-05;
EXEC sp_Insertar_hospitalizacion 6,6, 2025-09-06;
EXEC sp_Insertar_hospitalizacion 7,7, 2025-09-07;
EXEC sp_Insertar_hospitalizacion 8,8, 2025-09-08;
EXEC sp_Insertar_hospitalizacion 9,9, 2025-09-09;
EXEC sp_Insertar_hospitalizacion 10,10 2025-09-10;

CREATE VIEW vw_pacientes_hospitalizados AS
SELECT p.Nombre, p.Apellido, h.Numero_h, Tipo, h.Estado
FROM paciente p
JOIN Hospitalizaciones hs ON p.PacienteID= hs.PacienteID
JOIN Habitaciones h ON hs.HabitacioneID= h.HacibitacionID;
GO

CREATE VIEW vw_citas_pendientes_doctor AS
SELECT d.Nombre, d.Apellido, COUNT (c.CitaID) AS TotalPendientes
FROM Doctores d
JOIN citas c ON d.DoctorID= DoctorID
WHERE c.Estado ='pendiente'
GROUP BY d.Nombre, d.Apellido;
GO

CREATE VIEW vw_medicamentos_bajo_stock AS
SELECT Nombre, stock, precio
FROM Medicamentos
WHERE stock < 200;
GO

CREATE VIEW vw_pacientes_por_especialidad AS
SELECT d.Especialidad, COUNT (c.CitaID) AS pacientesAtendidos
FROM Doctores d
JOIN citas c ON d.DoctorID=c.DoctorID
```

GROUP BY d.Especialidad;
GO

SCRIPT COMPLETO - BD RESTAURANTE

```
CREATE DATABASE Restaurante;
GO
USE Restaurante;
GO

CREATE PROCEDURE sp_Crear_tabla_Clientes
AS
  BEGIN
IF NOT EXISTS (SELECT* FROM SYSOBJECTS WHERE name='Clientes')
  BEGIN
CREATE TABLE clientes (
ClienteID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Telefono VARCHAR (20),
Email VARCHAR (100) UNIQUE,
FechaRegistro DATE DEFAULT GETDATE ()
  END
ELSE
  END;
GO
EXEC sp_Crear_tabla_clientes;
GO

CREATE PROCEDURE sp_Crear_tabla_mesas
AS
  BEGIN
CREATE TABLE Mesas (
MesaID INT PRIMARY KEY IDENTITY,
NumeroMesa INT UNIQUE NOT NULL,
Capacidad INT CHECK (Capacidad>0),
Estado VARCHAR (20) DEFAULT 'Libre'
)
END
  ELSE
END;
  GO
EXEC sp_crear_tabla_mesas;
GO

CREATE PROCEDURE sp_Crear_tabla_empleados
AS
BEGIN
 IF NOT EXISTS (SELECT* FROM SYSOBJECTS WHERE name='Empleados')
 BEGIN
CREATE TABLE Empleados(
EmpleadoID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT, NULL,
Cargo VARCHAR (50) NOT NULL,
Telefono VARCHAR (20),
Email VARCHAR (100) UNIQUE
)
END;
 GO
```

```
EXEC sp_crear_tabla_empleados;
GO

CREATE PROCEDURE sp_crear_tabla_menu
AS
  BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Menu')
BEGIN
  CREATE TABLE Menu(
PlatoID INT PRIMARY KEY IDENTITY,
NombrePlato VARCHAR (100) NOT NULL,
Categoria VARCHAR (60),
Precio DECIMAL (10,2) CHECK (Precio>0)
)
 END;
GO
 EXEC sp-crear-tabla_menu;
GO

CREATE PROCEDURE sp-crear_tabla_reservas
AS
BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Reservas')
BEGIN
  CREATE TABLE Reservas (
ReservaID)
ClienteID)
MesaID)
fechaReserva DATETIME NOT NULL,
Estado VARCHAR (20) DEFAULT
)
 END;
EXEC sp_crear_tabla_reservas;
GO

CREATE PROCEDURE sp_crear_tabla_pedidos
 AS
BEGIN
 IF NOT EXISTS (SELECT*FROM sysobjects
WHERE name='Pedidos' AND)
BEGIN
  CREATE TABLE Pedidos (
PedidoID INT PRIMARY KEY IDENTITY,
ClienteID INT FOREIGN KEY REFERENCES
Clientes (ClienteID)
EmpleadoID INT FOREIGN KEY
REFERENCES Empleados
fechaPedido DATATIME DEFAULT
)
END;
GO
EXEC sp-crear_tabla_Pedidos;
GO

CREATE PROCEDURE sp_crear_tabla_facturas
AS
 BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects
```

```
WHERE name='facturas AND')
 BEGIN
CREATE TABLE facturas(
 facturaID INT PRIMARY KEY IDENTITY,
 PedidoID INT FOREIGN KEY REFERENCES
Pedidos
 fechafactura DATETIME DEFAULT
GETDATE
 Total DECIMAL (10,2) CHECK (Total>=0)
)
END
GO
 EXEC sp-crear_tabla_facturas;
GO

CREATE PROCEDURE sp-insertar_cliente
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Teléfono VARCHAR (20),
@Email VARCHAR (100)
AS
 BEGIN
 If EXISTS (SELECT 1 FROM Clientes WHERE
BEGIN
 RETURN;
END
 If (@Nombre = "OR @Apellido="")
BEGIN
END

INSERT INTO Clientes (Nombre, Apellido, Teléfono, Email)
 VALUES (@Nombre, @Apellido, @Teléfono, @Email);
END;
 GO

EXEC sp-Insertar_Cliente 'Carlos, 'Gomez', '3101234567', Carlos@gmail.com';
EXEC sp-Insertar-Cliente 'Maria, 'Lopez', '3112345678', 'maria@gmail.com';
EXEC sp-Insertar-Cliente 'Juan, 'Martinez', 3123456789', Juan@gmail.com;
EXEC sp-Insertar-Cliente 'Ana', Rodriguez', '3134567890', ana@gmail.com';
EXEC sp-Insertar-Cliente 'Pedro', Sánchez', 3145678901', 'Pedro@gmail.com';
EXEC sp-Insertar_cliente 'Laura', Torres', 3156789012', laura@gmail.com',
EXEC sp-Insertar-Cliente 'Andrés, Ramirez', 3167890123, andres@gmail.com';
EXEC sp-Insertar-Cliente 'Sofia, 'Morales', 3178901234', Sofia@gmail.com;
EXEC sp-Insertar-Cliente 'Diego, 'Castro', 3189012345', diego@gmail.com;
EXEC sp-Insertar-Cliente 'Camila', Hernández', 3190123456', Camila@gmail.com,

CREATE PROCEDURE sp_Insertar_mesa
@NumeroMesa INT,
@Capacidad INT,
@Estado VARCHAR (20)
 AS
BEGIN
END;
 GO

EXEC sp-Insertar_mesa 1, 2 'libre';
EXEC sp_Insertar_mesa 2, 4 'libre';
EXEC sp-Insertar_mesa 3, 4 'libre';
```

```
EXEC sp_Insertar_mesa 4, 6 'libre';
EXEC sp_Insertar_mesa 5, 2 'libre';
EXEC sp_Insertar_mesa 6, 8 'libre';
EXEC sp_Insertar_mesa 7, 2 'libre';
EXEC sp_Insertar_mesa 8, 4 'libre';
EXEC sp_Insertar_mesa 9, 6 'libre';
EXEC sp_Insertar_mesa 10, 2 'libre';

CREATE PROCEDURE sp_Insertar_empleado
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Cargo VARCHAR (50),
@Telefono VARCHAR (20),
@Email VARCHAR (100),
AS
BEGIN
 IF EXISTS (SELECT 1 FROM Empleados WHERE
END;
 GO

 INSER INTO Empleados (Nombre, Apellido, Cargo, Telefono, Email)
VALUES (@Nombre, @Apellido, @Cargo, @Telefono @Email);
END;
 GO

EXEC sp_Insertar_empleado 'luis, 'fernández ', Mesero '3201111111', luis@gmail.com);
EXEC sp_Insertar_empleado 'Carolina', Vega 'Cajera '3202222222 'Carolina@gmail.com
EXEC sp_Insertar_empleado 'Daniel,' Perez, 'Cocinero '3203333333 'Daniel@gmail.com'
EXEC sp_Insertar_empleado ' Juliana, Ortiz' Mesera '3204444444' Juliana@gmail.com;
EXEC sp_Insertar_empleado 'Jorge, Mendoza', Administrador '3205555555' Jorge@gmail.com,
EXEC sp_Insertar_empleado 'Natalia, reyes', Cocinera '3206666666' natalia@gmail.com;
EXEC sp_Insertar_empleado ' Andres, Suarez, Mesero '3207777777' andres@gmail.com;
EXEC sp_Insertar_empleado 'Paula, Moreno, Hostess '3208888888' Paola@gmail.com;
EXEC sp_Insertar_empleado 'felipe, Castillo, Mesero '3209999999' felipe@gmail.com;
EXEC sp_Insertar_empleado 'Valentina, Rios, chef '3210000000' valentina@gmail.com

CREATE PROCEDURE sp_Insertar_plato
@NombrePlato VARCHAR (100),
@Categoria VARCHAR (50),
@Precio DECIMAL (10,2)
 AS
BEGIN
 IF EXIST (SELECT 1 FROM Menu WHERE
NombrePlato = @NombrePlato)
 BEGIN
RETURN;
 END;
GO

 INSERT INTO Menu (NombrePlato, Categoria, Precio)
VALUES (@Nombreplato, @Categoria, @Precio)
 END;
GO

EXEC sp_Insertar_plato 'Bandeja Paisa,' Plato fuerte,' 25000;
EXEC sp_Insertar_plato 'Ajiaco; Plato fuerte; 22000;
EXEC sp_Insertar_plato 'Empanadas' 'Entrada', 8000;
EXEC sp_Insertar_plato 'Arepa con Queso' Entrada' 6000
```

```
EXEC SP_Insertar_Plato 'Cazuela de Mariscos', 'Plato fuerte', 35000;
EXEC SP_Insertar_Plato 'Jugo de Mango', 'Bebida', 6000;
EXEC SP_Insertar_Plato 'Postre de tres leches', 'postre', 10000;
EXEC SP_Insertar_Plato 'flan de Caramelo', 'postre', 9000;
EXEC SP_Insertar_Plato 'Sancocho', 'Plato fuerte', 28000;

CREATE PROCEDURE SP_Insertar_reserva
@ClienteID INT,
@MesaID INT,
@fechaReserva DATATIME,
@Estado VARCHAR (20)
BEGIN
  END

INSERT INTO Reservas (clienteID, MesaID, fechaReserva, Estado)
  VALUES (@clienteID, @MesaID, @fechaReserva, @Estado);
END;
  GO

EXEC SP_Insertar_reserva 1,1 '2025-09-12 19:00; Pendiente;
EXEC SP_Insertar_reserva 2,2 '2025-09-12 20:00 Pendiente;
EXEC SP_Insertar_reserva 3,3 '2025-09-12 21:00 Pendiente;
EXEC SP_Insertar_reserva 4,4 '2025-09-13 19:30 Pendiente;
EXEC SP_Insertar_reserva 5,5 '2025-09-13 20:30 Pendiente;
EXEC SP_Insertar_reserva 6,6 '2025-09-13 21:00 Pendiente;
EXEC SP_Insertar_reserva 7,7 '2025-09-14 19:00 pendiente;
EXEC SP_Insertar_reserva 8,8 '2025-09-14 20:00 Pendiente;
EXEC SP_Insertar_reserva 9,9 '2025-09-14 21:00 pendiente;
EXEC SP_Insertar_reserva 10,10 '2025-09-15 19:00 Pendiente;

CREATE PROCEDURE SP_Insertar_pedido
@ClienteID INT,
@EmpleadoID INT,
@Estado VARCHAR (20)
  AS
BEGIN
RETURN;
  END

INSERT INTO Pedidos (clienteID, EmpleadoID, Estado)
  VALUES (@clienteID, @EmpleadoID, @Estado);
END;
  GO

EXEC SP_Insertar_Pedido 1,1, 'En proceso';
EXEC SP_Insertar_Pedido 2,2, 'En proceso';
EXEC SP_Insertar_Pedido 3,3, 'En proceso';
EXEC SP_Insertar_pedido 4,4, 'En proceso';
EXEC SP_Insertar_Pedido 5,5 'En proceso';
EXEC SP_Insertar_pedido 6,6, 'En proceso';
EXEC SP_Insertar_Pedido 7,7, 'En proceso';
EXEC SP_Insertar_Pedido 8,8, 'En proceso';
EXEC SP_Insertar_Pedido 9,9, 'En proceso';
EXEC SP_Insertar_Pedido 10,10 'En Proceso';

CREATE PROCEDURE SP_Insertar_factura
@PedidoID
@Total DECIMAL (10,2)
```

```
AS
 BEGIN
 IF NOT EXISTS (SELECT 1 FROM  pedidos
 WHERE Pedido ID = @Pedido ID)
RETURN;
 END

INSERT INTO facturas (Pedido ID, Total)
VALUES (@Pedido ID, @Total);
 END;
GO

EXEC sp_Insertar_factura 1, 35000;
EXEC sp_Insertar_factura 2, 50000;
EXEC sp_Insertar_factura 3, 25000;
EXEC sp_Insertar_factura 4, 60000;
EXEC sp_Insertar_factura 5, 42000;
EXEC sp_Insertar_factura 6, 38000;
EXEC sp_Insertar_factura 7, 30000;
EXEC sp_Insertar_factura 8, 70000;
EXEC sp_Insertar_factura 9, 45000;
EXEC sp_Insertar_factura 10, 55000;

CREATE VIEW vw_Cliente_reservas_activas AS
SELECT c.Nombre, c.Apellido, r.fechaReserva, r.Estado
FROM Clientes c
JOIN Reservas r ON c.ClienteID = r.ClienteID
WHERE r.Estado = 'Pendiente';

CREATE VIEW vw_mesas_ocupadas AS
SELECT MesaID, NumeroMesa, Estado
FROM Mesas
WHERE Estado='Ocupada';

CREATE VIEW vw_Pedidos_empleados AS
SELECT e.Nombre, e.Apellido, COUNT (p.PedidoID) AS TotalPedidos
FROM Empleados e
LEFT JOIN Pedidos p ON e.EmpleadoID = p.EmpleadoID
GROUP BY e.Nombre, e.Apellido;

CREATE VIEW vw_Platos_Caros AS
SELECT NombrePlato, Categoria, Precio
FROM Menu
WHERE Precio > 50000;

CREATE VIEW vw_facturacion_Clientes AS
SELECT c.Nombre, c.Apellido, SUM (f.Total) AS TotalGastado
FROM Clientes c
JOIN Pedidos p ON c.ClienteID = p.ClienteID
JOIN facturas f ON p.PedidoID = f.PedidoID
GROUP BY c.Nombre, c.Apellido;
```

SCRIPT COMPLETO - BD SUPERMERCADO

```
CREATE DATABASE supermercado;
GO
USE supermercado;
GO

CREATE PROCEDURE sp_Crear_tabla_Clientes
AS
 BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Clientes')
 BEGIN
CREATE TABLE Clientes(
Cliente ID PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Teléfono VARCHAR (20),
Email VARCHAR (100) UNIQUE
fechaRegistro DATE DEFAULT GETDATE()
 END
END;
 GO
EXEC sp_Crear_tabla_Clientes;
GO

CREATE PROCEDURE sp_crear_tabla_productos
AS
 BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Productos')
 BEGIN
 CREATE TABLE productos(
 ProductoID PRIMARY KEY IDENTITY,
 Nombre Producto VARCHAR (100) NOT NULL,
 Categoria VARCHAR (50),
 Precio DECIMAL (10,2) CHECK (precio>0),
 Stock INT CHECK (Stock >=0)
 )
 END
END;
 GO
EXEC sp_Crear_tabla_Productos;
 GO

CREATE PROCEDURE sp_Crear_tabla_empleados
AS
 BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Empleados')
BEGIN
 CREATE TABLE Empleados(
EmpleadoID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Cargo VARCHAR (50)
Teléfono VARCHAR (20)
Email VARCHAR (100) UNIQUE
)
END
```

```
END;
 GO
EXEC sp_crear_tabla_empleados;
GO

CREATE PROCEDURE sp_crear_tabla_proveedores
 AS
BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=proveedores)
AS
 BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=ventas
AS
 BEGIN
VentasID INT PRIMARY KEY IDENTITY,
ClienteID INT FOREIGN KEY REFERENCES clientes (ClientesID),
EmpleadoID INT FOREIGN KEY REFERENCES empleados (EmpleadoID),
fecha venta DATETIME DEFAULT GETDATE (),
Total DECIMAL (10,2) CHECK (total >=0)
)
 END
END;
 GO
EXEC sp_crear_tabla_ventas;
 GO

CREATE PROCEDURE sp_crear_tabla_detalleventa
AS
 BEGIN
 CREATE TABLE DetalleVenta (
DetalleID INT PRIMARY KEY IDENTITY,
VentaID INT FOREIGN KEY REFERENCES venta (ventaID),
ProductoID INT FOREIGN KEY REFERENCES producto (productoID),
Cantidad INT CHECK (cantidad >0),
Subtotal DECIMAL (10,2) CHECK (subtotal >=0)
)
 END
END;
 GO
EXEC sp_crear_tabla_detalleventa;
 GO

CREATE PROCEDURE sp_crear_tabla_inventario
 AS
BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Inventario')
BEGIN
CREATE TABLE Inventario(
InventarioID
ProductoID
StockActual
UltimaActualizacion DATETIME DEFAULT GETDATE()
)
 END
END;
GO
 EXEC sp_crear_tabla_inventario;
 GO
```

```
CREATE PROCEDURE sp-Insertar-Cliente
@Nombre VARCHAR (60),
@Apellido VARCHAR (50),
@Telefono VARCHAR (20),
@Email VARCHAR (100)
 AS
BEGIN
 IF EXISTS (SELECT 1 FROM Clientes WHERE Email= @Email)
 BEGIN
END)
 INSERT INTO Clientes (Nombre, Apellido, Telefono, Email)
 VALUES (@Nombre, @Apellido, @Telefono, @Email);
END;
 GO

EXEC sp-Insertar-Cliente 'Camila', Ramirez, 3101111111', camila@sup-com;
EXEC sp-Insertar-Cliente 'Juan', Perez 3102222222', Juan@sup.com;
EXEC sp-Insertar-Cliente 'Sofia', López, 3103333333', Sofia@sup.com;
EXEC sp-Insertar-Cliente 'Carlos', Martinez, 3104444444', Carlos@sup.com;
EXEC sp-Insertar-Cliente 'Valentina', Morales, 3105555555', valentina@sup.com;
EXEC sp-Insertar-Cliente 'Diego', Hernández, 3106666666', diego@sup.com;
EXEC sp-Insertar-cliente 'laura', Castro, 3107777777', laura@sup.com;
EXEC sp-Insertar-cliente 'Andrés', Garcia, 3108888888 'andres@sup.com;
EXEC sp-Insertar-cliente 'Natalia', ortiz, 3109999999 'natalia@sup.com;
EXEC sp-Insertar-Cliente 'felipe', vega, 3110000000 'felipe@sup.com;

CREATE PROCEDURE sp-Insertar-producto
@NombreProducto VARCHAR (100),
@Categoria VARCHAR (50),
@Precio DECIMAL (10,2),
@Stock INT
AS
 BEGIN
 If EXISTS (SELECT 1 FROM Productos WHERE NombreProducto
 BEGIN;
END);

 INSERT INTO productos (NombreProducto, Categoria, Precio, Stock)
VALUES (@NombreProducto, @categoria, @precio, @stock);
 END;
GO

EXEC sp-Insertar-producto 'Arroz 1Kg', 'Granos', 3500, 100;
EXEC sp-Insertar-producto 'frijoles 1Kg' 'Granos', 4000, 80;
EXEC sp-Insertar-producto 'Leche 1L', 'lácteos' 3200, 150;
EXEC sp-Insertar-producto 'Queso 500g', 'lacteos' 8000, 70;
EXEC sp-Insertar-producto 'Huevos 30u', 'Proteinas' 12000, 60;
EXEC sp-Insertar-producto 'Carne res 1Kg', 'proteinas' 25000, 50;
EXEC sp-Insertar-producto 'Pan tajado', 'Panaderia' 4500, 90;
EXEC sp-Insertar-producto 'Aceite 1L', 'Aceites' 12000, 120;
EXEC sp-Insertar-producto 'Manzanas 1Kg', 'frutas' 6000, 40;
EXEC sp-Insertar-producto 'Bananos 1Kg', 'frutas' 5000, 65;

CREATE PROCEDURE sp-Insertar-Proveedor
@NombreProveedor VARCHAR (100),
@Telefono VARCHAR (20),
@Email VARCHAR (100),
@Direccion VARCHAR (150)
```

```
AS
 BEGIN
 IF EXISTS (SELECT 1 FROM proveedores WHERE Email)
 BEGIN
RETURN;
 END
INSERT INTO proveedores (NombreProveedor, Telefono, Email, Direccion)
 END;
GO

EXEC SP_Insertar_Proveedor 'Distribuidora Central' 3209999999, Central@sup.com,
EXEC SP_Insertar_Proveedor 'Lacteos Andinos' 3201111111, lacteos@sup.com,
EXEC SP_Insertar_Proveedor 'Granos Norte' 3202222222, granos@sup.com,
EXEC SP_Insertar_Proveedor 'Carnes Premium' 3203333333, Carnes@sup.com,
EXEC SP_Insertar_Proveedor 'Frutas del Valle' 3204444444, frutas@sup.com,
EXEC SP_Insertar_Proveedor 'Panadería la 14' 3205555555, panaderia@sup.com
EXEC SP_Insertar_Proveedor 'Aceite el dorado' 3206666666, aceite@sup.com,
EXEC SP_Insertar_Proveedor 'Verduras frescas' 3207777777, verduras@sup.com,
EXEC SP_Insertar_Proveedor 'Huevos del campo' 3208888888, Huevos@sup.com,
EXEC SP_Insertar_Proveedor 'Importa Caribe' 3210000000, Caribe@sup.com;

CREATE PROCEDURE SP_Insertar_Venta
@ ClienteID INT,
@ EmpleadoID INT,
@ Total DECIMAL (10,2)
 AS
BEGIN
 IF NOT EXIST (SELECT 1 FROM clientes WHERE)
 BEGIN
END;

INSERT INTO ventas (ClienteID, EmpleadoID, Total) VALUES);
 END;
GO

EXEC SP_Insertar_Venta 1,1, 35000;
EXEC SP_Insertar_Venta 2,2, 50000;
EXEC SP_Insertar_Venta 3,3, 22000;
EXEC SP_Insertar_Venta 4,4, 60000;
EXEC SP_Insertar_Venta 5,5, 45000;
EXEC SP_Insertar_Venta 6,6, 38000;
EXEC SP_Insertar_Venta 7,7, 70000;
EXEC SP_Insertar_Venta 8,8, 25000;
EXEC SP_Insertar_Venta 9,9, 32000;
EXEC SP_Insertar_Venta 10,10 40000;

CREATE PROCEDURE SP_Insertar_Inventario
@ ProductoID INT,
@ StockActual INT
 AS
BEGIN
 IF NOT EXISTS (SELECT 1 FROM Productos WHERE)
BEGIN
 RETURN;
END
 -
 INSERT INTO Inventario (ProductoID, stockActual) VALUES
 END;
```

60

```
EXEC SP_Insertar_Inventario 1, 100;
EXEC SP_Insertar_Inventario 2, 80;
EXEC SP_Insertar_Inventario 3, 150;
EXEC SP_Insertar_Inventario 4, 70;
EXEC SP_Insertar_Inventario 5, 60;
EXEC SP_Insertar_Inventario 6, 50;
EXEC SP_Insertar_Inventario 7, 90;
EXEC SP_Insertar_Inventario 8, 120;
EXEC SP_Insertar_Inventario 9, 40;
EXEC SP_Insertar_Inventario 10, 55;

CREATE VIEW vw_Ventas_Clientes AS
SELECT C.Nombre, C.Apellido, SUM (V.Total) AS TotalComprado
FROM Clientes C
JOIN Ventas V ON C.ClienteID = V.ClienteID
GROUP BY C.Nombre, C.Apellido;

CREATE VIEW vw_Productos_bajo_Stock AS
SELECT NombreProducto, Stock
FROM Productos
WHERE Stock < 20;

CREATE VIEW vw_Ventas_empleados AS
SELECT E.Nombre, E.Apellido, COUNT (V.VentaID) AS NumVentas
FROM Empleados E
JOIN Ventas V ON E.EmpleadoID = V.EmpleadoID
GROUP BY E.Nombre, E.Apellido;

CREATE VIEW vw_Inventario_actual AS
SELECT P.NombreProducto, I.StockActual, I.UltimaActualizacion
FROM Inventario I
JOIN Productos P ON I.ProductoID = P.ProductoID;

CREATE VIEW vw_Productos_mas_vendidos AS
SELECT P.NombreProducto, TotalVendido
FROM DetalleVenta D
JOIN Productos P ON D.ProductosID = ProductoID
GROUP BY P.NombreProducto
ORDER BY TotalVendido DESC;
```

SCRIPT COMPLETO - BANCO

```
CREATE DATABASE BancoDB;
GO
USE BancoDB;
GO

CREATE PROCEDURE sp_Crear_tabla_Clientes
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Cliente')
  BEGIN
    CREATE TABLE Clientes (
ClienteID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Email VARCHAR (100) UNIQUE NOT NULL,
Telefono VARCHAR (20),
fechaRegistro DATE DEFAULT GETDATE ()
)
  END
END;
GO
EXEC sp_Crear_tabla_clientes;
GO

CREATE PROCEDURE sp_Crear_tabla_Cuentas
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Cuentas')
  BEGIN
  CREATE TABLE Cuentas(
CuentaID INT PRIMARY KEY IDENTITY,
ClienteID INT FOREIGN KEY REFERENCES Clientes (ClienteID),
NumeroCuenta VARCHAR (20) UNIQUE NOT NULL,
Saldo DECIMAL (12,2) CHECK (saldo >=0),
TipoCuenta VARCHAR (20) CHECK (TipoCuenta IN ('Ahorros',
fechaApertura DATE DEFAULT GETDATE ()
)
END
END;
GO
EXEC sp_Crear_tabla_Cuentas;
GO

CREATE PROCEDURE sp_Crear_tabla_transacciones
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name)
  BEGIN
CREATE TABLE Transacciones(
  TransaccionID PRIMARY KEY IDENTITY,
  CuentaID INT FOREIGN KEY REFERENCES Cuentas (cuentaID),
  Monto DECIMAL (12,2) NOT NULL,
  Tipo VARCHAR (20) CHECK (tipo IN ('Deposito','Retiro')),
  fecha DATETIME DEFAULT GETDATE ()
  )
```

```
END
END;
GO
EXEC sp_crear_tabla_transacciones;
GO

CREATE PROCEDURE sp_crear_tabla_prestamos
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='prestamo')
  BEGIN
    CREATE TABLE prestamos(
PrestamoID PRIMARY KEY IDENTITY,
ClienteID INT FOREIGN KEY REFERENCES clientes (clienteID),
Monto DECIMAL (12,2) NOT NULL CHECK
Interes DECIMAL (5,2) NOT NULL,
Estado VARCHAR (20) CHECK (Estado IN ('Activo','pagado')),
fechaInicio DATE DEFAULT GETDATE()
)
  END
END;
GO
EXEC sp_crear_tabla_prestamos;
GO

CREATE PROCEDURE sp_crear_tabla_sucursales
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=Surcusales)
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE 'Surcursales'
BEGIN
  CREATE TABLE    Sucursales(
SucursalID  INT    PRIMARY KEY  IDENTITY,
Nombre VARCHAR    (50)  NOT NULL,
Ciudad  VARCHAR    (50)  NOT NULL,
Direccion VARCHAR  (100)
)
  END
END;
GO
EXEC  sp_crear_tabla_sucursales;
GO

CREATE  PROCEDURE  sp_crear_tabla_empleados
AS
BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Empleados')
  BEGIN
  CREATE TABLE Empleados(
EmpleadoID  INT  PRIMARY KEY  IDENTITY,
Nombre  VARCHAR  (50) NOT NULL,
Apellido  VARCHAR  (50) NOT NULL,
Cargo  VARCHAR  (50),
SucursalID INT  FOREIGN  KEY  REFERENCES  sucursales (SucursalID),
fechaIngreso
)
  END
```

```
IF NOT EXISTS (SELECT 1 FROM Cuentas WHERE CuentaID=@Cuenta)
BEGIN
    PRINT 'Cuenta no existe';
    RETURN;
END
INSERT INTO Transacciones (CuentaID, Monto, Tipo)
VALUES (@CuentaID, @Monto, @Tipo)
END;
GO

CREATE PROCEDURE sp-insertar-prestamo
@ClienteID INT,
@Monto DECIMAL (12,2),
@Interes DECIMAL (5,2),
@Estado VARCHAR (20)
AS
BEGIN
  IF NOT EXISTS (SELECT 1 FROM Clientes WHERE ClienteID)
BEGIN
  PRINT 'Cliente no existe';
  RETURN;
END
INSERT INTO Prestamos (ClienteID, Monto, Interes, Estado)
VALUES (@ClienteID, @Monto, @Interes, @Estado)
END;
GO

CREATE PROCEDURE sp_insertar_surcursal
@Nombre VARCHAR (50),
@Ciudad VARCHAR (50),
@Direccion VARCHAR (100)
AS
  BEGIN
INSERT INTO Surcusales (Nombre, Ciudad, Direccion)
VALUES (@Nombre, @Ciudad, @Direccion)
END;
GO

CREATE PROCEDURE sp_insertar_empleado
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Cargo VARCHAR (50),
@SucursalID INT
AS
BEGIN
  IF NOT EXISTS (SELECT 1 FROM Sucursales WHERE SucursalID)
BEGIN
PRINT 'Sucursal no existe';
  RETURN;
END
INSERT INTO Empleados (Nombre, Apellido, Cargo, SucursalID)
VALUES (@Nombre, @Apellido, @Cargo, @SucursalI)
END;
GO

INSERT INTO Reportes (Descripcion) VALUES (@Descripcion)
END;
GO
```

```
END;
GO
EXEC sp_crear_tabla_empleados;
GO
CREATE PROCEDURE sp_crear_tabla_reportes
AS
BEGIN
 IF NOT EXISTS (SELECT* FROM sysobjects WHERE 'Reportes')
BEGIN
 CREATE TABLE Reportes (
ReporteID INT PRIMARY KEY IDENTITY,
Descripcion VARCHAR (200),
FechaGeneracion DATE DEFAULT GETDATE()
)
 END
END;
GO
EXEC sp_crear_tabla_reportes;
GO
CREATE PROCEDURE sp_insertar_cliente
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Email VARCHAR (100),
@Telefono VARCHAR (20)
AS
 BEGIN
 IF EXISTS (SELECT 1 FROM Clientes WHERE Email = @Email)
BEGIN
IF EXISTS (SELECT 1 FROM Clientes (Nombre, Apellido, Email, telefono)
VALUES (@Nombre, @Apellido, @Email, @Telefono)
END;
GO

CREATE PROCEDURE sp_insertar_Cuenta
@ClienteID INT,
@NumeroCuenta VARCHAR (20),
@Saldo DECIMAL (12,2),
@TipoCuenta VARCHAR (20)
AS
BEGIN
 IF NOT EXISTS (SELECT 1 FROM Clientes WHERE clienteID)
 BEGIN
IF NOT EXISTS (SELECT 1 FROM Clientes WHERE clienteID = @ClienteID)
 BEGIN
 PRINT 'Cliente no existe';
 RETURN;
END
 INSERT INTO Cuentas (ClienteID, NumeroCuenta, Saldo, TipoCuenta)
 VALUES (@ClienteID, @NumeroCuenta, @Saldo, @TipodeCuenta)
END;
GO

CREATE PROCEDURE sp_insertar_transación
@CuentaID INT,
@Monto DECIMAL (12,2),
@Tipo VARCHAR (20)
AS
BEGIN
```

```
EXEC SP_Insertar_Cliente 'Juan', 'Perez', 'Juanp@gmail.com', '3001111111';
EXEC SP_Insertar_Cliente 'Ana', 'Martinez', 'Ana.m@gmail.com', '3002222222';
EXEC SP_Insertar_Cliente 'Luis', 'Gómez', 'luisg@gmail.com', '3003333333';
EXEC SP_Insertar_Cliente 'Maria', 'López', 'maria1@gmail.com', '3004444444';
EXEC SP_Insertar_Cliente 'Pedro', 'Ramirez', 'pedrar@gmail.com', '3005555555';
EXEC SP_Insertar_Cliente 'Lucia', 'Fernández', 'lucia8@gmail.com', '3006666666';
EXEC SP_Insertar_Cliente 'Carlos', 'Torres', 'carlost@gmail.com', '3007777777';
EXEC SP_Insertar_Cliente 'Laura', 'Hernández', 'laurah@gmail.com', '3008888888';
EXEC SP_Insertar_Cliente 'Diego', 'Castro', 'diegoc@gmail.com', '3009999999';
EXEC SP_Insertar_Cliente 'Paola', 'Jimenez', 'paolaj@gmail.com', '3010000000';
/*Sucursales*/
EXEC SP_Insertar_Sucursal 'Sucursal Centro', 'Bogota', 'Calle 10 #5-20';
EXEC SP_Insertar_Sucursal 'Sucursal Norte', 'Bogota', 'Cra 45 #120-55';
EXEC SP_Insertar_Sucursal 'Sucursal Sur', 'Medellin', 'Calle 80 #30-40';
EXEC SP_Insertar_Sucursal 'Sucursal Cali', 'Cali', 'Av. Colombia #45-33';
EXEC SP_Insertar_Sucursal 'Sucursal Cartagena', 'Cartagena', 'Cra 8 #20-10';
EXEC SP_Insertar_Sucursal 'Sucursal bucaramanga', 'bucaramanga', 'Calle 15 #8-22';
EXEC SP_Insertar_Sucursal 'Sucursal barranquilla', 'barranquilla', 'Av. 40 #18-90';
EXEC SP_Insertar_Sucursal 'Sucursal pereira', 'pereira', 'Cra 2 #6-33';
EXEC SP_Insertar_Sucursal 'Sucursal Manizales', 'Manizales', 'Calle 22 #15-44';
EXEC SP_Insertar_Sucursal 'Sucursal pasto', 'pasto', 'Cra 7 #25-10';

EXEC SP_Insertar_empleado 'Andres', 'Santos', 'Gerente', 1;
EXEC SP_Insertar_empleado 'Beatriz', 'Mora', 'Cajera', 1;
EXEC SP_Insertar_empleado 'Camilo', 'Pardo', 'Asesor', 2;
EXEC SP_Insertar_empleado 'Daniela', 'Vargas', 'Gerente', 2;
EXEC SP_Insertar_empleado 'Esteban', 'Suárez', 'Asesor', 3;
EXEC SP_Insertar_empleado 'Felipe', 'López', 'Cajero', 3;
EXEC SP_Insertar_empleado 'Gabriela', 'Moreno', 'Asesora', 4;
EXEC SP_Insertar_empleado 'Hector', 'Silva', 'Asesor', 5;
EXEC SP_Insertar_empleado 'Isabel', 'Cruz', 'Gerente', 6;
EXEC SP_Insertar_empleado 'Jorge', 'Castillo', 'Cajero', 7;

EXEC SP_Insertar_Cuenta 1, '1001', 500000, 'Ahorros';
EXEC SP_Insertar_Cuenta 2, '1002', 1500000, 'Corriente';
EXEC SP_Insertar_Cuenta 3, '1003', 250000, 'Ahorros';
EXEC SP_Insertar_Cuenta 4, '1004', 750000, 'Ahorros';
EXEC SP_Insertar_Cuenta 5, '1005', 1000000, 'Corriente';
EXEC SP_Insertar_Cuenta 6, '1006', 900000, 'Ahorros';
EXEC SP_Insertar_Cuenta 7, '1007', 600000, 'Corriente';
EXEC SP_Insertar_Cuenta 8, '1008', 1200000, 'Ahorros';
EXEC SP_Insertar_Cuenta 9, '1009', 350000, 'Ahorros';
EXEC SP_Insertar_Cuenta 10, '1010', 800000, 'Corriente';

EXEC SP_Insertar_transaccion 1, 200000, 'Deposito';
EXEC SP_Insertar_transaccion 2, 50000, 'Retiro';
EXEC SP_Insertar_transaccion 3, 100000, 'Deposito';
EXEC SP_Insertar_transaccion 4, 30000, 'Retiro';
EXEC SP_Insertar_transaccion 5, 25000, Deposito;
EXEC SP_Insertar_transaccion 6, 80000, Retiro;
EXEC SP_Insertar_transaccion 7, 150000, Deposito;
EXEC SP_Insertar_transaccion 8, 200000, Transferencia;
EXEC SP_Insertar_transaccion 9, 50000, Retiro;
EXEC SP_Insertar_transaccion 10, 100000, Deposito;

EXEC SP_Insertar_Prestamo 1, 2000000, 5.5, 'Activo';
EXEC SP_Insertar_Prestamo 2, 3500000, 6.2, 'Activo';
EXEC SP_Insertar_Prestamo 3, 1000000, 4.5, 'Pagado';
```

```
EXEC SP_Prestamo 4, 5000000, 7.0, 'Mora';
EXEC SP_Prestamo 5, 2200000, 5.0, 'Activo';
EXEC SP_Prestamo 6, 1500000, 4.8, 'Pagado';
EXEC SP_Prestamo 7, 3300000, 6.5, 'Mora';
EXEC SP_Prestamo 8, 4000000, 6.9, 'Activo';
EXEC SP_Prestamo 9, 1800000, 5.3, 'Pagado';
EXEC SP_Prestamo 10, 2700000, 5.7, 'Activo';

EXEC SP_Insertar_reporte 'Clientes con préstamos activos';
EXEC SP_Insertar_reporte 'Saldo total por cliente';
EXEC SP_Insertar_reporte 'Transacciones del último mes';
EXEC SP_Insertar_reporte 'Prestamo en mora';
EXEC SP_Insertar_reporte 'Empleados por sucursal';
EXEC SP_Insertar_reporte 'Reporte de ingresos por cuentas';
EXEC SP_Insertar_reporte 'Clientes frecuente';
EXEC SP_Insertar_reporte 'Historial de pagos';
EXEC SP_Insertar_reporte 'Reporte general de préstamos';
EXEC SP_Insertar_reporte 'Resumen de transacciones';

CREATE VIEW vw_Clientes_prestamos_activos AS
SELECT C.ClienteID, C.Nombre, C.Apellido, P.Monto, P.Estado
FROM Clientes C
JOIN Prestamos P ON C.ClienteID = P.ClienteID
WHERE P.Estado = 'Activo';
GO

CREATE VIEW vw_saldo_total_Cliente AS
SELECT C.ClientID, C.Nombre, SUM (Ct.Saldo) AS SaldoTotal
FROM Clientes C
JOIN Cuentas Ct ON C.ClienteID C.Nombre
GROUP BY C.ClienteID, C.Nombre;
GO

CREATE VIEW vw_transacciones_mes AS
SELECT t.TransaccionID, c.NumeroCuenta, t.Monto, t.tipo, t.fecha
FROM Transacciones t
JOIN Cuentas c ON t.CuentaID = c.CuentaID
WHERE MONTH (t.fecha) = MONTH (GETDATE()) AND YEAR = YEAR (GETDATE());
GO

CREATE VIEW vw_prestamos_mora AS
SELECT P.PrestamoID, C.Nombre, P.Monto, P.fechaInicio
FROM Prestamo P
JOIN Clientes C ON P.clientesID = C.ClientesID
WHERE P.Estado = 'Mora';
GO

CREATE VIEW vw_empleados_sucursal AS
SELECT S.Nombre AS Sucursal, e.Nombre, e.Apellido, e.Cargo
FROM Empleados e
JOIN Sucursales S ON e.SucursalID = S.SucursalID;
GO
```

SCRIPT COMPLETO - UNIVERSIDAD

```
CREATE DATABASE universidadDB;
GO
USE universidadDB;
GO

CREATE PROCEDURE sp-Crear_tabla_estudiantes
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=Estudiante
BEGIN
CREATE TABLE Estudiantes (
EstudianteID PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Email VARCHAR (100) UNIQUE NOT NULL,
fechaNacimiento DATE NOT NULL,
ProgramaID INT NULL
)
  END
END;
GO
EXEC sp-crear_tabla_estudiantes;
GO

CREATE PROCEDURE sp-crear_tabla_profesores
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=Profesores
BEGIN
CREATE TABLE Profesores (
ProfesorID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Apellido VARCHAR (50) NOT NULL,
Especialidad VARCHAR (50),
Email VARCHAR (100) UNIQUE NOT NULL
)
  END
END;
GO
EXEC sp-crear_tabla_profesores;
GO

CREATE PROCEDURE sp-Crear_tabla_Cursos
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Cursos')
BEGIN
  CREATE TABLE Cursos (
CursoID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (100) NOT Null,
Creditos INT CHECK (Creditos >0),
ProfesorID INT FOREIGN KEY REFERENCES Profesores (ProfesoresID)
)
  END
END,
```

```
GO
EXEC sp_Crear_tabla_cursos;
GO

CREATE PROCEDURE sp_Crear_tabla_programas
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='programas')
BEGIN
CREATE TABLE programas(
Programa ID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (100) NOT NULL,
facultad VARCHAR (100) NOT NULL
)
END
END;
GO
EXEC sp-crear-tabla_programas;
GO

CREATE PROCEDURE sp_crear_tabla_matriculas
AS
BEGIN
  CREATE TABLE Matriculas(
  Matricula ID INT PRIMARY KEY IDENTITY,
  EstudianteID FOREIGN KEY REFERENCES Curso (curso),
  CursoID INT FOREIGN KEY REFERENCES Estudiantes (Estudiante),
  fecha DATE DEFAULT GETDATE()
)
  END
END;
GO
EXEC sp_crear_tabla_matriculas;
GO

CREATE PROCEDURE sp_Crear-tabla_notas
AS
BEGIN
  IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Notas'
  BEGIN
CREATE TABLE Notas(
NOTAID INT PRIMARY KEY IDENTITY,
Matricula INT FOREIGN KEY REFERENCES Matriculas (MatriculasID),
Calificacion DECIMAL (4,2) CHECK (calificacion BETWEEN 0 AND 5)
)
END
  END;
GO
EXEC sp-Crear-tabla_notas;
GO

CREATE PROCEDURE sp-Crear_taba_bibliotecas
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name=Bibliotecas')
BEGIN
  CREATE TABLE Bibliotecas(
BibliotecaID INT PRIMARY KEY IDENTITY,
```

```
Nombre VARCHAR (100) NOT NULL,
ubicacion VARCHAR (100) NOT NULL
)
  END
END;
GO
EXEC sp_crear_tabla_bibliotecas;
GO

CREATE PROCEDURE sp_insertar_estudiante
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Email VARCHAR (100),
@fechaNacimiento DATE,
@ProgramaID INT=Null
AS
BEGIN
  IF EXISTS (SELECT 1 FROM estudiantes WHERE email=@Email)
BEGIN
  PRINT 'El estudiante ya está registrado con ese email.';
  RETURN;
END
INSERT INTO Estudiantes (Nombre, Apellido, Email, fechaNacimiento)
VALUES (@Nombre, @Apellido, @Email, @fechaNacimiento, @ProgramaID)
END;
GO

CREATE PROCEDURE sp_insertar_profesor
@Nombre VARCHAR (50),
@Apellido VARCHAR (50),
@Especialidad VARCHAR (50),
@Email VARCHAR (100)
AS
  BEGIN
IF EXISTS (SELECT 1 FROM Profesores WHERE Email= @Email)
  BEGIN
    PRINT 'Profesores (Nombre, Apellido, Especialidad, Email)
    RETURN;
END
INSERT INTO Profesores (Nombre, Apellido, Especialidad, Email)
VALUES (@Nombre, @Apellido, @Especialidad, @Email)
END;
GO

CREATE PROCEDURE sp_insertar_curso
@Nombre VARCHAR (100),
@Creditos INT,
@Profesor INT
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM Profesores WHERE profesorID
  BEGIN
PRINT 'Profesor no existe';
  RETURN;
END
INSERT INTO Cursos (Nombre, Creditos, profesorID)
VALUES (@Nombre, @Creditos, @profesorID)
END;
```

```
GO

CREATE PROCEDURE sp_insertar_matricula
@EstudianteID INT,
@CursoID INT
AS
  BEGIN
IF NOT EXISTS (SELECT 1 FROM Estudiantes WHERE EstudianteID
  BEGIN
  PRINT 'Estudiante no existe';
  RETURN;
END
IF NOT EXISTS (SELECT 1 FROM Curso WHERE CursoID=@CursoID)
BEGIN
  PRINT 'Curso no existe';
  RETURN;
END
INSERT INTO Matriculas (EstudianteID, CursoID)
VALUES (@EstudianteID, @CursoID)
  END;
GO

CREATE PROCEDURE sp_insertar_nota
@MatriculaID INT,
@Calificacion DECIMAL (4,2)
  AS
  BEGIN
PRINT 'Matricula no existe';
  RETURN;
END
INSERT INTO Notas (MatriculaID, Calificacion)
VALUES (@MatriculaID, @Calificacion)
END;
GO

EXEC sp_Insertar_programa 'Ingeneria de sistemas', facultad de Ingenieria;
EXEC sp_Insertar_programa 'Derecho', facultad de ciencias Jurridicas;
EXEC sp_Insertar_Programa 'Medicina', facultad de ciencia de la salud;
EXEC sp_Insertar_programa 'Administracion', facultad de economia;
EXEC sp_Insertar_programa 'psicologia', facultad de ciencia sociales;
EXEC sp_Insertar_programa 'Arquitectura', facultad de Arquitectura
EXEC sp_Insertar_Programa 'Contaduria Publica', facultad de economia;
EXEC sp_Insertar_programa 'Ingeneria civil', facultad de ingenieria;
EXEC sp_Insertar_programa 'Enfermeria', facultad de ciencia de salud;
EXEC sp_insertar_programa 'Comunicacion socia', facultad de humanidad;

EXEC sp_Insertar_estudiante 'Juan', Pérez', Juanp@uni.edu', 2000-05-10', 1;
EXEC sp_Insertar_estudiante 'Ana', Martinez, anam@uni.edu', 2001-03-22', 2;
EXEC sp_Insertar_estudiante 'Luis', Gomez', luisg@uni.edu', 1999-11-15', 4;
EXEC sp_Insertar_estudiante 'Maria', lopez', mariaL@uni.edu', 2002-01-30', 3;
EXEC sp_Insertar_estudiante 'Pedro' Ramirez', pedrorO@uni.edu', 2000-07-19', 4;
EXEC sp_Insertar_estudiante 'Lucia', fernandez', luciaf@uni.edu', 1998-09-25', 5;
EXEC sp_Insertar_estudiante 'Carlos', Torres', Carlosl@uni.edu', 2001-12-12', 6;
EXEC sp_Insertar_estudiante 'Laura', Hernandez', laurah@uni.edu', 2000-04-05', 7;
EXEC sp_Insertar_estudiante 'Diego' Castro', diegoc@uni.edu', 1999-06-17', 8;
EXEC sp_Insertar_estudiante 'paola' Jiménez', Paolar@uni.edu', 2002-02-28', 9;
```

```
EXEC  SP_Insertar_profesor 'Andrés', Santos', Programación, 'andres@uni.edu';
EXEC  SP_Insertar_profesor 'Beatriz', Mora', Derecho penal, 'beatriz@uni.edu';
EXEC  SP_Insertar_profesor 'Camilo', Pardo', Cirugía, 'Camilo@uni.edu';
EXEC  SP_Insertar_profesor 'Daniela', Vargas', Administración, 'daniela@uni.edu';
EXEC  SP_Insertar_Profesor 'Esteban', Suarez', psicología, 'esteban@uni.edu';
EXEC  SP_Insertar_Profesor 'felipe', Lopez', Diseño urbano, 'felipe@uni.edu';
EXEC  SP_Insertar_profesor 'Gabriela', Moreno', Contabilidad, 'gabriela@uni.edu';
EXEC  SP_Insertar_profesor 'Hector', Silva', Estructuras, 'hector@uni.edu';
EXEC  SP_Insertar_Profesor 'Isabel', Cruz', Enfermeria general, 'isabel@uni.edu';
EXEC  SP_Insertar_profesor 'Jorge', Castillo', Periodismo, 'jorge@uni.edu';

EXEC  SP_Insertar_Curso 'programacion I', 4, 1;
EXEC  SP_Insertar_Curso 'Derecho Constitucional', 3, 2;
EXEC  SP_Insertar_Curso 'Anatomia humana', 5, 3;
EXEC  SP_Insertar_Curso 'Teoria Administrativa', 4, 4;
EXEC  SP_Insertar_Curso 'psicologia del desarrollo', 3, 5;
EXEC  SP_Insertar_Curso 'Diseño Arquitectonico', 4, 6;
EXEC  SP_Insertar_Curso 'Contabilidad financiera', 4, 7;
EXEC  SP_Insertar_curso 'Mecánica del Suelos', 5, 8;
EXEC  SP_Insertar_Curso 'Cuidado del paciente', 3, 9;
EXEC  SP_Insertar_Curso 'Redaccion periodistica', 2, 10;

EXEC  SP_Insertar_matricula 1,1;
EXEC  SP_Insertar_matricula 2,2;
EXEC  SP_Insertar_matricula 3,3;
EXEC  SP_Insertar_matricula 4,4;
EXEC  SP_Insertar_matricula 5,5;
EXEC  SP_Insertar_matricula 6,6;
EXEC  SP_Insertar_matricula 7,7;
EXEC  SP_Insertar_matricula 8,8;
EXEC  SP_Insertar_matricula 9,9;
EXEC  SP_Insertar_matricula 10,10;

EXEC  SP_Insertar_nota 1, 4, 5;
EXEC  SP_Insertar_nota 2, 3, 8;
EXEC  SP_Insertar_nota 3, 4, 2;
EXEC  SP_Insertar_nota 4, 3, 5;
EXEC  SP_Insertar_nota 5, 4, 7;
EXEC  SP_Insertar_nota 6, 3, 9;
EXEC  SP_Insertar_nota 7, 4, 1;
EXEC  SP_Insertar_nota 8, 2, 8;
EXEC  SP_Insertar_nota 9, 4, 0;
EXEC  SP_Insertar_nota 10, 3, 6;

CREATE VIEW vw_estudiantes_programas AS
SELECT e.Nombre, e.Apellido, p.Nombre AS Programa, p.facultad
FROM Estudiantes e
JOIN Programas p ON e.ProgramaID = p.ProgramaID;
GO

CREATE VIEW vw_Promedio_notas_estudiante AS
SELECT e.nombre e.Apellido, AVG(n.Calificacion) AS promedio
FROM Estudiantes e
JOIN Matriculas m ON e.EstudianteID = m.estudianteID
JOIN Notas n ON m.matriculaID = n.MatriculaID
GROUP BY e.Nombre, e.Apellido;
GO
```

```
CREATE VIEW vw_Cursos-matriculados AS
SELECT C.Nombre AS curso, COUNT(m.MatriculaID) AS
FROM curso C
LEFT JOIN Matriculas m ON C.CursoID= m.CursoID
GROUP BY C.Nombre;
GO

CREATE VIEW vw_profesores_curso AS
SELECT P.Nombre, P.Apellido, C.Nombre AS curso
FROM Profesores p
JOIN Cursos C ON P.ProfesoresID= C.ProfesorID;
GO

CREATE VIEW vw_estudiantes_bajo_rendimiento AS
SELECT e.Nombre, e.Apellido, n.Calificacion
FROM Estudiantes e
JOIN Matriculas m ON e.EstudianteID=m.EstudianteID
JOIN Notas m ON m.MatriculaID = n.MatriculaID
WHERE n.Calificacion <3;
GO
```

SCRIPT COMPLETO-TIENDAONLINE

```
CREATE DATABASE TiendaOnline;
GO
USE TiendaOnline;
GO

CREATE PROCEDURE sp_crear_tabla_usuarios
AS
BEGIN
 IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='usuarios')
  BEGIN
CREATE TABLE usuarios(
usuarioID INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (50) NOT NULL,
Email VARCHAR (100) UNIQUE NOT NULL,
Contrasena VARCHAR (100) NOT NULL,
fechaRegistro DATE DEFAULT GETDATE ()
)
  END
END;
GO
  EXEC sp_crear_tabla_usuarios;

CREATE PROCEDURE sp_crear_tabla_categorias
AS
  BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Categoria')
BEGIN
  CREATE TABLE categorias(
  CategoriaID INT PRIMARY KEY IDENTITY,
  Nombre VARCHAR (50) NOT NULL
)
  END
END;
GO
EXEC sp_crear_tabla_categorias;

CREATE PROCEDURE sp_crear_tabla_Productos
AS
  BEGIN
  IF NOT EXISTS (SELECT* FROM sysobjects WHERE Name='productos')
BEGIN
  CREATE TABLE Productos (
PRODUCTOID `INT PRIMARY KEY IDENTITY,
Nombre VARCHAR (100) Not NULL,
Precio DECIMAL (10,2) CHECK (Precio >0),
Stock INT CHECK (Stock >=0),
CategoriaID INT FOREIGN KEY REFERENCES
)
END
END;
GO
EXEC sp_crear_tabla_Productos;

CREATE PROCEDURE sp_crear_tabla_carritos
 AS
```

```
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name = carrito
BEGIN
CREATE TABLE Carritos(
CarritoID INT PRIMARY KEY IDENTITY,
UsuarioID INT FOREIGN KEY REFERENCES usuarios
FechaCreacion DATE DEFAULT GETDATE()
)
END
END;
GO
EXEC sp_crear_tabla_carritos;

CREATE PROCEDURE sp-crear_tabla_pedidos
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name= pedidos
BEGIN
CREATE TABLE pedidos (
PedidoID INT PRIMARY KEY IDENTITY,
UsuarioID INT FOREIGN KEY REFERENCES usuarios (usuarioID),
fechaPedido DATE NOT NULL,
Estado VARCHAR (20) CHECK
)
END
END;
GO

CREATE PROCEDURE sp-crear-tabla-pagos
AS
BEGIN
IF NOT EXISTS (SELECT* FROM sysobjects WHERE name='Pagos'
BEGIN
CREATE TABLE pagos(
PagoID INT PRIMARY KEY IDENTITY,
PedidoID INT FOREIGN KEY REFERENCES pedidos (PedidoID),
Monto DECIMAL (10,2) CHECK (Monto >0),
fechaPago DATE NOT NULL,
Metodo VARCHAR (20) CHECK (Metodo IN)
)
END
END;
GO
EXEC sp-crear_tabla-pagos;

CREATE PROCEDURE sp-crear-tabla-envios
AS
BEGIN
CREATE TABLE Envios (
EnvioID INT PRIMARY KEY IDENTITY,
PedidoID INT FOREIGN KEY REFERENCES pedidos (PedidoID),
Direccion VARCHAR (200) NOT NULL,
fechaEnvio DATE,
Estado VARCHAR (20) CHECK
)
END
END;
GO
```

```
CREATE PROCEDURE Sp_Insertar_Usuario
@Nombre VARCHAR (50),
@Email VARCHAR (100),
@Contrasena VARCHAR (100)
AS
BEGIN
IF EXISTS (SELECT 1 FROM usuarios WHERE Email=@Email)
BEGIN
PRINT 'El correo ya está registrado';
RETURN;
END
INSERT INTO usuarios (Nombre, Email, Contrasena)
VALUES (@Nombre, @Email, @Contrasena);
END;
GO

CREATE PROCEDURE sp_Insertar_categoria
@Nombre VARCHAR (50)
AS
BEGIN
IF EXISTS (SELECT 1 FROM categorias WHERE Nombre = @Nombre)
BEGIN
PRINT 'la categoria ya existe';
RETURN;
END
INSERT INTO Categorias (Nombre)
VALUES (@Nombre);
END;
INSER INTO categorias (Nombre)
VALUES (@Nombre);
END;
GO

CREATE PROCEDURE sp_Insertar_Producto
@Nombre VARCHAR (100),
@Precio DECIMAL (10,2),
@Stock INT,
@Categoria INT
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM categorias WHERE CategoriaID)
BEGIN
PRINT 'La categoria no existe';
RETURN;
END
INSERT INTO Productos (Nombre, Precio, Stock, CategoriaID)
VALUES (@Nombre, @Precio, @Stock, @CategoriaID);
END;
GO

CREATE PROCEDURE sp_Insertar_Carrito
@UsuarioID INT
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM usuarios WHERE usuarioID=@usuarioID)
BEGIN
PRINT 'Usuario no existe';
RETURN;
```

```
END
 INSERT INTO carritos (usuarioID)
VALUES (@usuarioID);
  END;
GO

CREATE PROCEDURE sp_Insertar_Pedido
@usuarioID INT,
@fechaPedido DATE,
@Estado VARCHAR (20)
  AS
BEGIN

CREATE PROCEDURE sp_Insertar_Pago
@PedidoID INT,
@Monto DECIMAL (10,2),
@fechaPago DATE,
@Metodo VARCHAR (20)
  AS
BEGIN
  END
INSERT INTO pagos (PedidoID, Monto, fecha pago, Metodo)
VALUES (@PedidoID, @Monto, @fecha pago, @Metodo);
  END;
GO

CREATE VIEW vw_Productos_mas_vendidos AS
SELECT p.Nombre, COUNT (pe.PedidoID) AS TotalPedidos
FROM Productos p
JOIN Pedidos pe ON pe.usuarioID IS NOT NULL
  GROUP BY p.Nombre;
GO

CREATE VIEW vw_Usuarios_pedidos AS
SELECT pe.PedidoID, pe.fechaPedido, pe.Estado
FROM usuarios u
JOIN pedidos pe ON u.usuarioID = pe.usuarioID;
GO

CREATE VIEW vw_Pedidos_pagos AS
SELECT pe.PedidoID, pe.Estado, pa.Monto, pa.Metodo, pa.fecha pago
FROM pedidos pe
JOIN pagos pa ON pe.PedidoID = pa.PedidoID;
GO

CREATE VIEW vw_envios_Pendientes AS
SELECT e.EnvioID, e.Direccion, e.Estado, pe.fechaPedido
FROM Envios e
JOIN pedidos pe ON e.PedidoID = pe.pedidoID
WHERE e.Estado = 'Pendiente';
GO

CREATE VIEW vw_Ventas_por_usuario AS
SELECT u.Nombre, SUM (pa.Monto) AS TotalGastado
FROM usuarios u
JOIN pagos pa ON pe.PedidoID = pa.pedidoID
GROUP BY u.Nombre;
GO
```