**Análisis y Desarrollo de Software**

**FONRIO**

**Heidy Stephany Lozano León**

**Nicol Caterine Rodríguez Benítez**

**Kevin Santiago Saenz Reyes**

**Kevin Alexis Morales Avendaño**

**Ficha: 2996176**

**Servicio Nacional de Aprendizaje (SENA)  
Bogotá D.C.  
6 de junio 2025**

**Fabian Parra**

**FONRIO**

-- DDL

-- Creación de la base de datos

-- Creación de las tablas y sus atributos

-- Creación de la base de datos

CREATE DATABASE FONRIO;

USE FONRIO;

CREATE TABLE Estados (

ID\_Estado VARCHAR(20) PRIMARY KEY,

Nombre\_Estado VARCHAR(15) NOT NULL

);

CREATE TABLE Roles (

ID\_Rol VARCHAR(20) PRIMARY KEY,

Nombre VARCHAR(30) NOT NULL

);

CREATE TABLE Categorias (

ID\_Categoria VARCHAR(20) PRIMARY KEY,

Nombre\_Categoria VARCHAR(30) NOT NULL

);

CREATE TABLE Gamas (

ID\_Gama VARCHAR(20) PRIMARY KEY,

Nombre\_Gama VARCHAR(15) NOT NULL

);

CREATE TABLE Proveedores (

ID\_Proveedor VARCHAR(20) PRIMARY KEY,

Nombre\_Proveedor VARCHAR(45) NOT NULL,

Correo\_Electronico VARCHAR(30) UNIQUE NOT NULL,

Telefono VARCHAR(15) NOT NULL,

ID\_Estado VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Estado) REFERENCES Estados(ID\_Estado)

);

CREATE TABLE Clientes (

Documento\_Cliente VARCHAR(20) PRIMARY KEY,

Nombre\_Cliente VARCHAR(20) NOT NULL,

Apellido\_Cliente VARCHAR(20) NOT NULL,

Telefono VARCHAR(15) NOT NULL,

Fecha\_Nacimiento DATE NOT NULL,

Genero CHAR(1) CHECK (Genero IN ('F','M')) NOT NULL,

ID\_Estado VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Estado) REFERENCES Estados(ID\_Estado)

);

CREATE TABLE Empleados (

Documento\_Empleado VARCHAR(20) PRIMARY KEY,

Nombre\_Usuario NVARCHAR(30) NOT NULL,

Apellido\_Usuario NVARCHAR(30) NOT NULL,

Contrasena VARCHAR(20) NOT NULL,

Correo\_Electronico VARCHAR(100) UNIQUE NOT NULL,

Telefono VARCHAR(10) NOT NULL,

Genero CHAR(1) CHECK (Genero IN ('F','M')) NOT NULL,

ID\_Estado VARCHAR(20) NOT NULL,

ID\_Rol VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Estado) REFERENCES Estados(ID\_Estado),

FOREIGN KEY (ID\_Rol) REFERENCES Roles(ID\_Rol)

);

CREATE TABLE Contrasenas (

ID\_Contrasena INT IDENTITY(1,1) PRIMARY KEY,

Documento\_Empleado VARCHAR(20) UNIQUE NOT NULL,

Contrasena\_Hash VARCHAR(255) NOT NULL,

Fecha\_Creacion DATETIME DEFAULT GETDATE(),

FOREIGN KEY (Documento\_Empleado) REFERENCES Empleados(Documento\_Empleado)

);

CREATE TABLE Productos (

ID\_Producto VARCHAR(20) PRIMARY KEY,

Nombre\_Producto VARCHAR(30) NOT NULL,

Descripcion TEXT NOT NULL,

Precio\_Venta MONEY NOT NULL,

Stock\_Minimo INT NOT NULL,

ID\_Categoria VARCHAR(20) NOT NULL,

ID\_Estado VARCHAR(20) NOT NULL,

ID\_Gama VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Categoria) REFERENCES Categorias(ID\_Categoria),

FOREIGN KEY (ID\_Estado) REFERENCES Estados(ID\_Estado),

FOREIGN KEY (ID\_Gama) REFERENCES Gamas(ID\_Gama)

);

CREATE TABLE Compras (

ID\_Entrada VARCHAR(20) PRIMARY KEY,

Precio\_Compra MONEY NOT NULL,

ID\_Producto VARCHAR(20) NOT NULL,

Documento\_Empleado VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Producto) REFERENCES Productos(ID\_Producto),

FOREIGN KEY (Documento\_Empleado) REFERENCES Empleados(Documento\_Empleado)

);

CREATE TABLE Detalle\_Compras (

Fecha\_Entrada DATE NOT NULL,

Cantidad INT NOT NULL,

ID\_Proveedor VARCHAR(20) NOT NULL,

ID\_Entrada VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Proveedor) REFERENCES Proveedores(ID\_Proveedor),

FOREIGN KEY (ID\_Entrada) REFERENCES Compras(ID\_Entrada)

);

CREATE TABLE Ventas (

ID\_Venta VARCHAR(20) PRIMARY KEY,

Documento\_Cliente VARCHAR(20) NOT NULL,

Documento\_Empleado VARCHAR(20) NOT NULL,

FOREIGN KEY (Documento\_Cliente) REFERENCES Clientes(Documento\_Cliente),

FOREIGN KEY (Documento\_Empleado) REFERENCES Empleados(Documento\_Empleado)

);

CREATE TABLE Detalle\_Ventas (

Cantidad INT NOT NULL,

Fecha\_Salida DATE NOT NULL,

ID\_Producto VARCHAR(20) NOT NULL,

ID\_Venta VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Producto) REFERENCES Productos(ID\_Producto),

FOREIGN KEY (ID\_Venta) REFERENCES Ventas(ID\_Venta)

);

CREATE TABLE Detalle\_Devoluciones (

ID\_Devolucion VARCHAR(20) PRIMARY KEY,

Cantidad\_Devuelta INT NOT NULL,

ID\_Venta VARCHAR(20) NOT NULL,

FOREIGN KEY (ID\_Venta) REFERENCES Ventas(ID\_Venta)

);

CREATE TABLE Devoluciones (

ID\_Devolucion VARCHAR(20) PRIMARY KEY,

Fecha\_Devolucion DATE NOT NULL,

FOREIGN KEY (ID\_Devolucion) REFERENCES Detalle\_Devoluciones(ID\_Devolucion)

);

--DML Inserción de datos en las tablas

INSERT INTO Estados (ID\_Estado, Nombre\_Estado) VALUES

('EST001', 'Activo'),

('EST002', 'Inactivo'),

('EST003', 'En proceso');

INSERT INTO Roles (ID\_Rol, Nombre) VALUES

('ROL001', 'Administrador'),

('ROL002', 'Empleado');

INSERT INTO Categorias (ID\_Categoria, Nombre\_Categoria) VALUES

('CAT001', 'Celulares'),

('CAT002', 'Cargadores'),

('CAT003', 'Audifonos'),

('CAT004', 'Forros'),

('CAT005', 'Strap\_Phone');

INSERT INTO Gamas (ID\_Gama, Nombre\_Gama) VALUES

('GAM001', 'Baja'),

('GAM002', 'Media'),

('GAM003', 'Alta'),

('GAM004', 'Media'),

('GAM005', 'Alta');

-- Elimina las gamas con ID GAM004 y GAM005 de la tabla Gamas

DELETE FROM Gamas

WHERE ID\_Gama IN ('GAM004', 'GAM005');

INSERT INTO Proveedores(ID\_Proveedor,Nombre\_Proveedor,Correo\_electronico,Telefono,ID\_Estado) VALUES

('PROV001', 'Claro\_Colombia', 'Claro1@proveedor.com', '3012345678', 'EST001'),

('PROV002', 'Alkosto', 'Alkosto@proveedor.com', '3023456789', 'EST001'),

('PROV003', 'Éxito', 'Éxito@proveedor.com', '3034567890', 'EST002'),

('PROV004', 'Mercado\_Libre', 'Mercado\_Libre@proveedor.com', '3045678901', 'EST001'),

('PROV005', 'Accesorios\_Colombia', 'Accesorios@proveedor.com', '3056789012', 'EST003');

INSERT INTO Clientes(Documento\_Cliente,Nombre\_Cliente,Apellido\_Cliente,Telefono,Fecha\_Nacimiento,Genero,ID\_Estado ) VALUES

('CLI006', 'Juan', 'Pérez', '3144574273', '1990-05-10', 'M', 'EST001'),

('CLI007', 'Ana', 'Gómez', '3144574272', '1985-03-15', 'F', 'EST001'),

('CLI008', 'Luis', 'Ramírez', '3124574271', '2000-07-20', 'M', 'EST002'),

('CLI009', 'Carla', 'López', '3144774271', '1998-12-01', 'F', 'EST001'),

('CLI010', 'Alex', 'Moreno', '3144574275', '1995-09-30', 'M', 'EST003');

INSERT INTO Empleados(Documento\_Empleado,Nombre\_Usuario,Apellido\_Usuario,Contrasena,Correo\_electronico,Telefono,Genero,ID\_Estado,ID\_Rol) VALUES

('EMP006', 'Pedro', 'Cruz', '1234', 'pedrc@correo.com', '3023456784', 'M', 'EST001', 'ROL001'),

('EMP007', 'Lucía', 'Mendoza', 'abcd', 'luciaa@correo.com', '3023456781', 'F', 'EST001', 'ROL002'),

('EMP008', 'Miguel', 'Ortiz', 'pass', 'miguell@correo.com', '3023406789', 'M', 'EST002', 'ROL002'),

('EMP009', 'Paula', 'Reyes', 'clave', 'paulaa@correo.com', '3023454789', 'F', 'EST001', 'ROL001'),

('EMP010', 'Chris', 'Vega', 'test', 'chriss@correo.com', '3013456789', 'F', 'EST003', 'ROL002');

INSERT INTO Productos(ID\_Producto,Nombre\_Producto,Descripcion,Precio\_Venta,Stock\_Minimo,ID\_Categoria,ID\_Estado,ID\_Gama) VALUES

('PROD006', 'Redmi Note 12', 'Celular Xiaomi gama media con buena batería', 850000, 5, 'CAT001', 'EST001','GAM001'),

('PROD007', 'Motorola G60', 'Smartphone Motorola con cámara de 108MP', 950000, 4, 'CAT001', 'EST001','GAM001'),

('PROD008', 'Samsung A54', 'Celular Samsung gama media-alta', 1250000, 6, 'CAT001', 'EST001','GAM001'),

('PROD009', 'iPhone 13', 'iPhone 13 de 128GB', 3300000, 2, 'CAT001', 'EST001','GAM001'),

('PROD010', 'Cargador Tipo C', 'Cargador rápido universal tipo C', 50000, 10, 'CAT002', 'EST001','GAM001');

INSERT INTO Compras(ID\_Entrada,Precio\_Compra,ID\_Producto,Documento\_Empleado ) VALUES

('COM001', 300000, 'PROD006', 'EMP006'),

('COM002', 60000, 'PROD007', 'EMP007'),

('COM003', 500000, 'PROD008', 'EMP008'),

('COM004', 1500000, 'PROD009', 'EMP009'),

('COM005', 200000, 'PROD010', 'EMP010');

INSERT INTO Detalle\_Compras(Cantidad,ID\_Entrada,Fecha\_Entrada,ID\_Proveedor ) VALUES

(10, 'COM001','2025-02-01','PROV001'),

(15, 'COM002','2025-03-04','PROV002'),

(20, 'COM003','2025-12-05','PROV003'),

(5, 'COM004','2025-10-11','PROV004'),

(8, 'COM005','2025-9-12','PROV005');

INSERT INTO Ventas(ID\_Venta,Documento\_Cliente,Documento\_Empleado)VALUES

('VEN001', 'CLI006', 'EMP006'),

('VEN002', 'CLI007', 'EMP007'),

('VEN003', 'CLI008', 'EMP008'),

('VEN004', 'CLI009', 'EMP009'),

('VEN005', 'CLI010', 'EMP010');

INSERT INTO Detalle\_Ventas(Cantidad,ID\_Producto,ID\_Venta, Fecha\_Salida ) VALUES

(2, 'PROD006', 'VEN001','2025-02-01'),

(1, 'PROD007', 'VEN002','2025-01-06'),

(3, 'PROD008', 'VEN003','2025-04-04'),

(1, 'PROD009', 'VEN004','2025-06-09'),

(2, 'PROD010', 'VEN005','2025-09-20');

INSERT INTO Detalle\_Devoluciones(ID\_Devolucion,Cantidad\_Devuelta,ID\_Venta) VALUES

('DEV001', 1, 'VEN001'),

('DEV002', 1, 'VEN002'),

('DEV003', 2, 'VEN003'),

('DEV004', 1, 'VEN004'),

('DEV005', 1, 'VEN005');

INSERT INTO Devoluciones(ID\_Devolucion,Fecha\_Devolucion) VALUES

('DEV001', '2025-03-01'),

('DEV002', '2025-03-03'),

('DEV003', '2025-03-05'),

('DEV004', '2025-03-07'),

('DEV005', '2025-03-09');

--DQL Consultas

-- Consulta total de compras por cliente

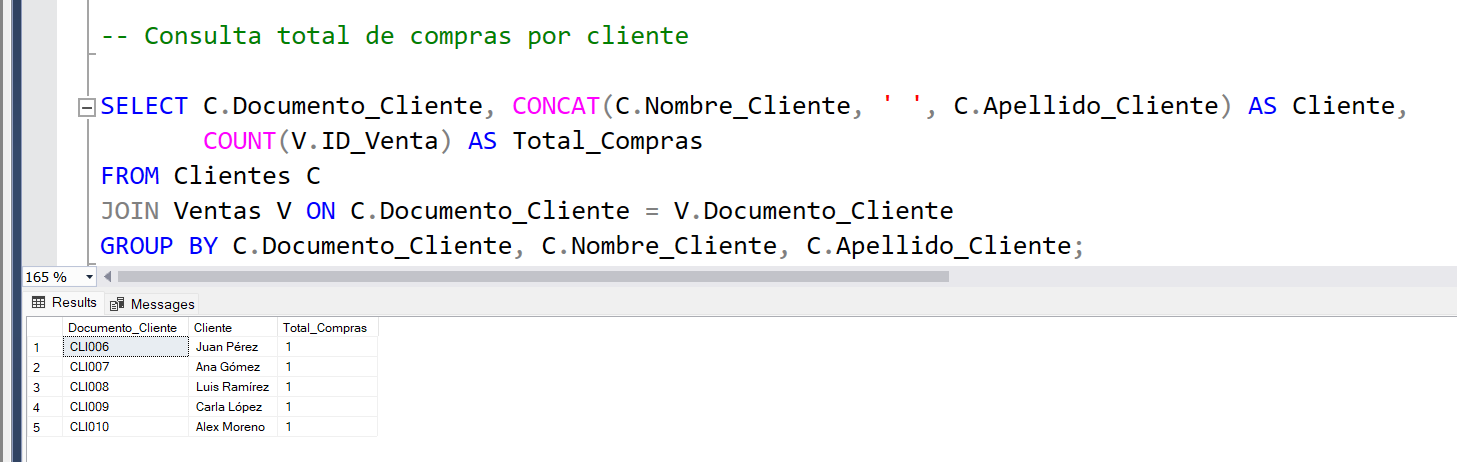
SELECT C.Documento\_Cliente, CONCAT(C.Nombre\_Cliente, ' ', C.Apellido\_Cliente) AS Cliente,

COUNT(V.ID\_Venta) AS Total\_Compras

FROM Clientes C

JOIN Ventas V ON C.Documento\_Cliente = V.Documento\_Cliente

GROUP BY C.Documento\_Cliente, C.Nombre\_Cliente, C.Apellido\_Cliente;



--Consulta total de ventas por empleado

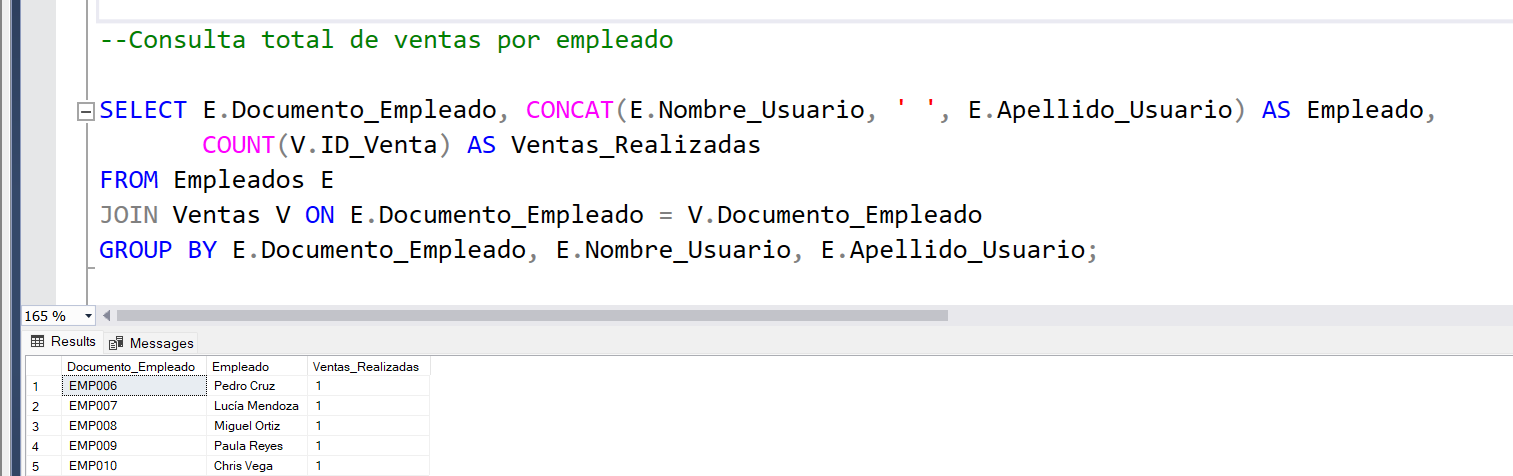
SELECT E.Documento\_Empleado, CONCAT(E.Nombre\_Usuario, ' ', E.Apellido\_Usuario) AS Empleado,

COUNT(V.ID\_Venta) AS Ventas\_Realizadas

FROM Empleados E

JOIN Ventas V ON E.Documento\_Empleado = V.Documento\_Empleado

GROUP BY E.Documento\_Empleado, E.Nombre\_Usuario, E.Apellido\_Usuario;



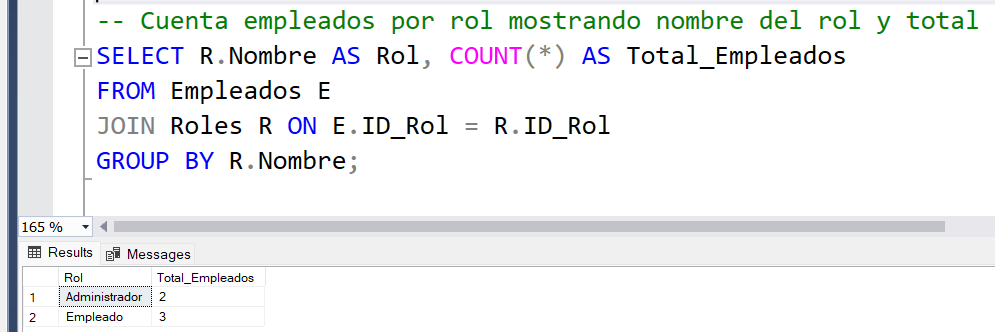
-- Cuenta empleados por rol mostrando nombre del rol y total

SELECT R.Nombre AS Rol, COUNT(\*) AS Total\_Empleados

FROM Empleados E

JOIN Roles R ON E.ID\_Rol = R.ID\_Rol

GROUP BY R.Nombre;



-- Cuenta total de personas por género (empleados y clientes juntos)

SELECT Genero, COUNT(\*) AS Cantidad

FROM (

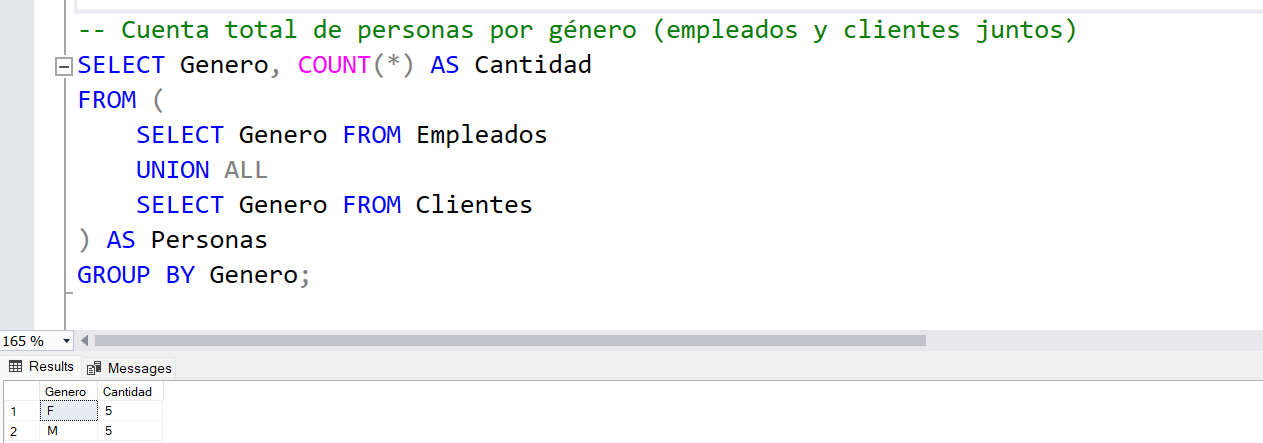
SELECT Genero FROM Empleados

UNION ALL

SELECT Genero FROM Clientes

) AS Personas

GROUP BY Genero;

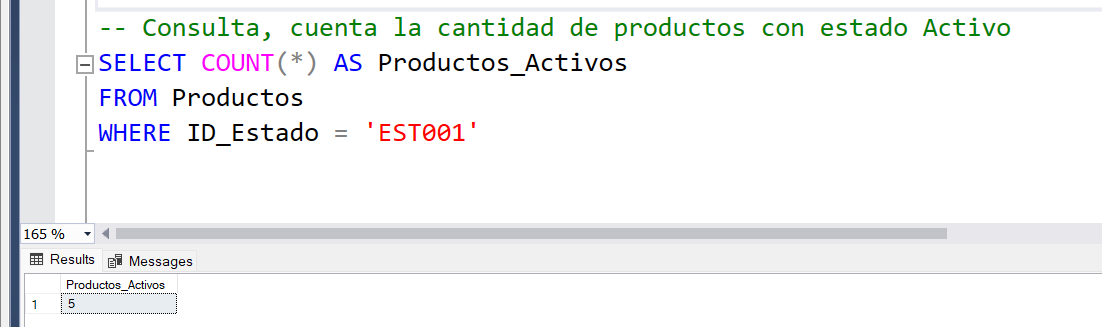


-- Consulta, cuenta la cantidad de productos con estado Activo

SELECT COUNT(\*) AS Productos\_Activos

FROM Productos

WHERE ID\_Estado = 'EST001'



-- Lista correos electrónicos indicando si son de proveedores o empleados

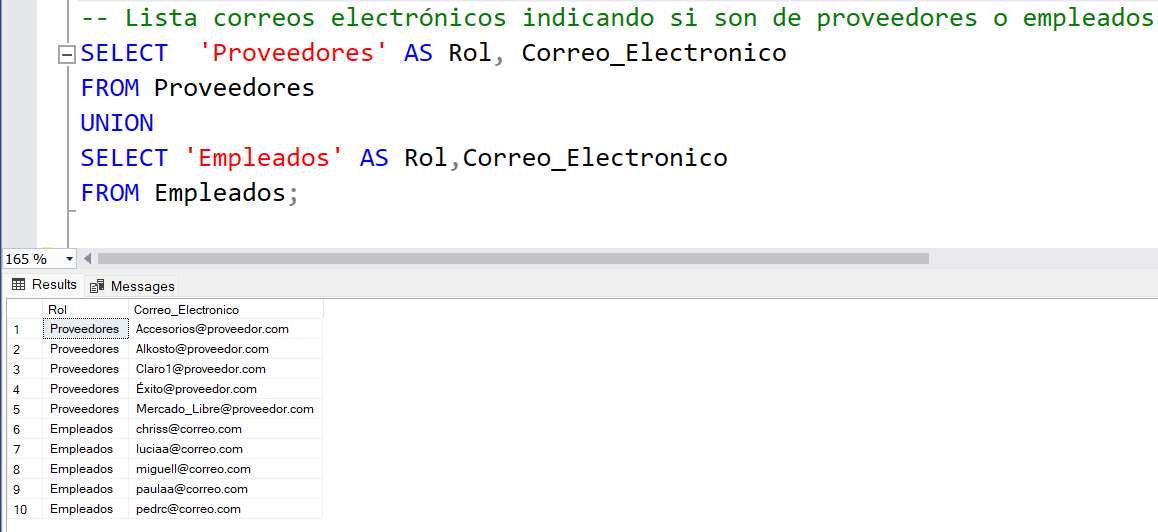
SELECT 'Proveedores' AS Rol, Correo\_Electronico

FROM Proveedores

UNION

SELECT 'Empleados' AS Rol,Correo\_Electronico

FROM Empleados;



-- Muestra todas las gamas ordenadas alfabéticamente por nombre

SELECT \* FROM Gamas ORDER BY Nombre\_Gama;

-- Muestra una lista de los productos vendidos con la fecha de salida,

--el nombre completo del cliente y la cantidad, ordenados desde la venta más reciente.

SELECT

DV.Fecha\_Salida,

CONCAT(C.Nombre\_Cliente, ' ', C.Apellido\_Cliente) AS Cliente,

P.Nombre\_Producto,

DV.Cantidad

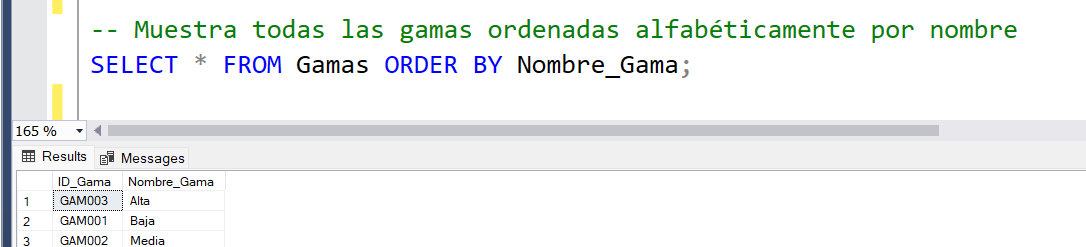
FROM Detalle\_Ventas DV

JOIN Ventas V ON DV.ID\_Venta = V.ID\_Venta

JOIN Clientes C ON V.Documento\_Cliente = C.Documento\_Cliente

JOIN Productos P ON DV.ID\_Producto = P.ID\_Producto

ORDER BY DV.Fecha\_Salida DESC;



-- Muestra el nombre de los proveedores activos y la cantidad de

--compras únicas que han realizado.

SELECT

PR.Nombre\_Proveedor,

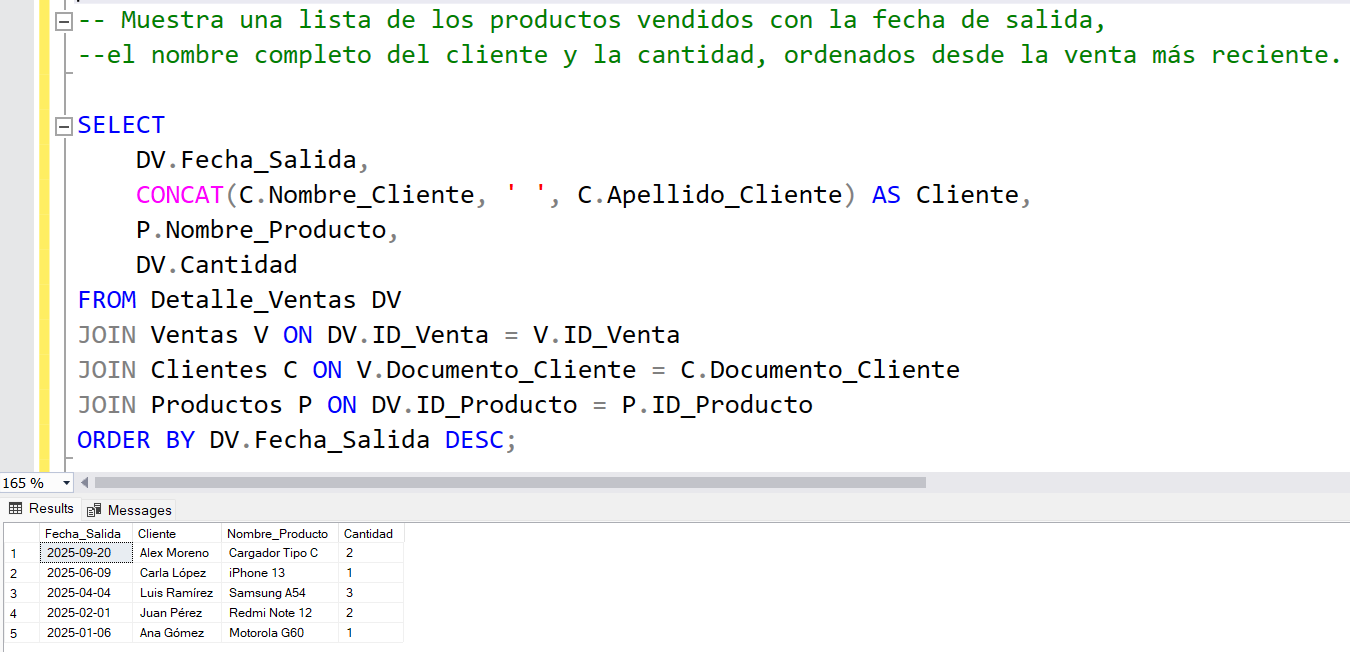
COUNT(DISTINCT DC.ID\_Entrada) AS Numero\_Compras

FROM Proveedores PR

JOIN Detalle\_Compras DC ON PR.ID\_Proveedor = DC.ID\_Proveedor

WHERE PR.ID\_Estado = 'EST001'

GROUP BY PR.Nombre\_Proveedor;



--TRIGGER

-- Resta la cantidad vendida del stock mínimo del producto después de una venta.

CREATE TRIGGER trg\_UpdateStockAfterVenta

ON Detalle\_Ventas

AFTER INSERT

AS

BEGIN

-- Actualizar el stock restando la cantidad vendida

UPDATE p

SET p.Stock\_Minimo = p.Stock\_Minimo - i.Cantidad

FROM Productos p

INNER JOIN inserted i ON p.ID\_Producto = i.ID\_Producto;

END;

-- Trigger que actualiza el Stock\_Minimo en Productos

-- al registrar una nueva compra en Detalle\_Compras

CREATE TRIGGER trg\_UpdateStockAfterCompra

ON Detalle\_Compras

AFTER INSERT

AS

BEGIN

UPDATE p

SET p.Stock\_Minimo = p.Stock\_Minimo + i.Cantidad

FROM Productos p

INNER JOIN Compras c ON p.ID\_Producto = c.ID\_Producto

INNER JOIN inserted i ON c.ID\_Entrada = i.ID\_Entrada;

END;

--Procedimientos Almacenados en SQL Server

--Consultar productos con bajo stock

CREATE PROCEDURE sp\_ProductosBajoStock

@Minimo INT

AS

BEGIN

SELECT ID\_Producto, Nombre\_Producto , Stock\_Minimo

FROM Productos

WHERE Stock\_Minimo <= @Minimo;

END;

--Consultar resumen de ventas por fecha

CREATE PROCEDURE sp\_ResumenVentasPorFecha

@FechaInicio DATE,

@FechaFin DATE

AS

BEGIN

SELECT

V.ID\_Venta,

DV.Fecha\_Salida AS Fecha,

C.Nombre\_Cliente + ' ' + C.Apellido\_Cliente AS Cliente,

DV.ID\_Producto,

P.Nombre\_Producto AS Producto,

DV.Cantidad,

P.Precio\_Venta,

(DV.Cantidad \* P.Precio\_Venta) AS Total

FROM Ventas V

INNER JOIN Clientes C ON V.Documento\_Cliente = C.Documento\_Cliente

INNER JOIN Detalle\_Ventas DV ON V.ID\_Venta = DV.ID\_Venta

INNER JOIN Productos P ON DV.ID\_Producto = P.ID\_Producto

WHERE DV.Fecha\_Salida BETWEEN @FechaInicio AND @FechaFin;

END;

-- Registrar ventas

CREATE PROCEDURE sp\_RegistrarVenta

@Documento\_Cliente VARCHAR(20),

@Fecha DATETIME,

@ID\_Producto VARCHAR(20),

@Cantidad INT

AS

BEGIN

DECLARE @ID\_Venta VARCHAR(20);

SET @ID\_Venta = LEFT(NEWID(), 20);

INSERT INTO Ventas (ID\_Venta, Documento\_Cliente, Documento\_Empleado)

VALUES (@ID\_Venta, @Documento\_Cliente, );

INSERT INTO Detalle\_Ventas (ID\_Venta, ID\_Producto, Cantidad, Fecha\_Salida)

VALUES (@ID\_Venta, @ID\_Producto, @Cantidad, @Fecha);

END;