

```

-- =====
-- 1. CONFIGURACIÓN INICIAL Y CREACIÓN DE LA BASE DE DATOS
-- =====

-- Si la base de datos ya existe, se elimina para asegurar una instalación
limpia.
IF DB_ID(N'RecetasFavoritasDB') IS NOT NULL
    DROP DATABASE RecetasFavoritasDB;
GO

CREATE DATABASE RecetasFavoritasDB;
GO

USE RecetasFavoritasDB;
GO

-- =====
-- 2. CREACIÓN DE TABLAS BASE
-- =====

-- TABLA: Usuarios
CREATE TABLE Usuarios (
    UsuarioID INT PRIMARY KEY IDENTITY(1,1),
    NombreUsuario NVARCHAR(100) NOT NULL UNIQUE,
    Email NVARCHAR(150) UNIQUE,
    TipoUsuario NVARCHAR(50), -- 'Profesional', 'Aficionado'
    FechaRegistro DATETIME DEFAULT GETDATE()
);

-- TABLA: Categorías
CREATE TABLE Categorías (
    CategoríaID INT PRIMARY KEY IDENTITY(1,1),
    NombreCategoría NVARCHAR(100) NOT NULL UNIQUE
);

-- TABLA: Recetas
CREATE TABLE Recetas (
    RecetaID INT PRIMARY KEY IDENTITY(1,1),
    Título NVARCHAR(200) NOT NULL,
    Descripción NVARCHAR(MAX),
    TiempoPreparaciónMin INT,
    TiempoCocciónMin INT,
    Porciones INT,
    NivelDificultad NVARCHAR(50), -- 'Fácil', 'Medio', 'Difícil'
    NotaChef NVARCHAR(MAX),
    CategoríaID INT FOREIGN KEY REFERENCES Categorías(CategoríaID),
    UsuarioID INT NULL FOREIGN KEY REFERENCES Usuarios(UsuarioID) -- Propietario
    de la receta
);

-- TABLA: Ingredientes
CREATE TABLE Ingredientes (
    IngredienteID INT PRIMARY KEY IDENTITY(1,1),
    NombreIngrediente NVARCHAR(150) NOT NULL UNIQUE,

```

```

        UnidadMedidaPredeterminada NVARCHAR(50)
    );

-- TABLA: RecetaIngredientes (Relación Muchos a Muchos)
CREATE TABLE RecetaIngredientes (
    RecetaID INT FOREIGN KEY REFERENCES Recetas(RecetaID),
    IngredienteID INT FOREIGN KEY REFERENCES Ingredientes(IngredienteID),
    Cantidad DECIMAL(10, 2) NOT NULL,
    UnidadMedida NVARCHAR(50),
    PRIMARY KEY (RecetaID, IngredienteID)
);

-- TABLA: PasosPreparacion
CREATE TABLE PasosPreparacion (
    PasoID INT PRIMARY KEY IDENTITY(1,1),
    RecetaID INT FOREIGN KEY REFERENCES Recetas(RecetaID),
    NumeroPaso INT NOT NULL,
    Instruccion NVARCHAR(MAX) NOT NULL,
    UNIQUE (RecetaID, NumeroPaso)
);
GO

-- =====
-- 3. INSERCIÓN DE DATOS DE PRUEBA Y CATÁLOGOS (SEED DATA)
-- =====

-- Insertar Categorías
INSERT INTO Categorías (NombreCategoria) VALUES
('Postres'),
('Platos Fuertes'),
('Entrantes'),
('Bebidas'),
('Salsas');

-- Insertar Usuarios
INSERT INTO Usuarios (NombreUsuario, Email, TipoUsuario) VALUES
('ChefGourmet', 'chef.gourmet@mail.com', 'Profesional'),
('AficionadoCocina', 'cocina.amateur@mail.com', 'Aficionado');

-- Insertar Ingredientes
INSERT INTO Ingredientes (NombreIngrediente, UnidadMedidaPredeterminada) VALUES
('Harina de trigo', 'gramos'),
('Azúcar', 'gramos'),
('Huevo', 'unidad'),
('Mantequilla sin sal', 'gramos'),
('Leche', 'ml'),
('Manzana', 'unidad'),
('Filete de Res', 'gramos'),
('Vino Tinto', 'ml');

-- Insertar Receta de Ejemplo 1 (Asignada a Aficionado)
DECLARE @CID_Postre INT = (SELECT CategoriaID FROM Categorías WHERE
NombreCategoria = 'Postres');
DECLARE @UID_Aficionado INT = (SELECT UsuarioID FROM Usuarios WHERE

```

```

NombreUsuario = 'AficionadoCocina');
DECLARE @R1ID INT;

INSERT INTO Recetas (Titulo, Descripcion, TiempoPreparacionMin,
TiempoCoccionMin, Porciones, NivelDificultad, NotaChef, CategoriaID, UsuarioID)
VALUES
('Tarta de Manzana Clásica', 'Una receta tradicional de tarta de manzana.', 30,
45, 8, 'Medio', 'Usar manzanas firmes como Granny Smith para mejor textura.',
@CID_Postre, @UID_Aficionado);
SET @R1ID = SCOPE_IDENTITY();

-- Ingredientes para Receta 1
INSERT INTO RecetaIngredientes (RecetaID, IngredienteID, Cantidad, UnidadMedida)
VALUES
(@R1ID, (SELECT IngredienteID FROM Ingredientes WHERE NombreIngrediente =
'Harina de trigo'), 300, 'gramos'),
(@R1ID, (SELECT IngredienteID FROM Ingredientes WHERE NombreIngrediente =
'Azúcar'), 150, 'gramos'),
(@R1ID, (SELECT IngredienteID FROM Ingredientes WHERE NombreIngrediente =
'Huevo'), 2, 'unidad'),
(@R1ID, (SELECT IngredienteID FROM Ingredientes WHERE NombreIngrediente =
'Manzana'), 5, 'unidad');

-- Pasos para Receta 1
INSERT INTO PasosPreparacion (RecetaID, NumeroPaso, Instruccion) VALUES
(@R1ID, 1, 'Mezclar la harina y el azúcar.'),
(@R1ID, 2, 'Añadir los huevos, amasar y refrigerar la masa.'),
(@R1ID, 3, 'Rellenar la masa con las manzanas cortadas y hornear por 45
minutos.');
```

GO

```

-- =====
-- 4. CREACIÓN DE VISTAS (Views)
-- =====

-- 4.1. VistaRecetasDetalle
CREATE VIEW VistaRecetasDetalle AS
SELECT
    R.RecetaID,
    R.Titulo,
    R.TiempoPreparacionMin,
    R.TiempoCoccionMin,
    R.Porciones,
    R.NivelDificultad,
    C.NombreCategoria AS Categoria,
    U.NombreUsuario AS Creador
FROM
    Recetas R
JOIN
    Categorias C ON R.CategoriaID = C.CategoriaID
LEFT JOIN
    Usuarios U ON R.UsuarioID = U.UsuarioID;
GO
```

```

-- 4.2. VistaIngredientesPorReceta
CREATE VIEW VistaIngredientesPorReceta AS
SELECT
    R.Titulo AS Receta,
    R.RecetaID,
    RI.Cantidad,
    RI.UnidadMedida,
    I.NombreIngrediente AS Ingrediente
FROM
    Recetas R
JOIN
    RecetaIngredientes RI ON R.RecetaID = RI.RecetaID
JOIN
    Ingredientes I ON RI.IngredienteID = I.IngredienteID;
GO

```

```

-- 4.3. VistaPasosPreparacionOrdenados
CREATE VIEW VistaPasosPreparacionOrdenados AS
SELECT
    R.Titulo AS Receta,
    R.RecetaID,
    PP.NumeroPaso,
    PP.Instruccion
FROM
    Recetas R
JOIN
    PasosPreparacion PP ON R.RecetaID = PP.RecetaID;
GO

```

```

-- =====
-- 5. CREACIÓN DE FUNCIONES (Functions)
-- =====

```

```

-- 5.1. FN_CalcularTiempoTotal (Escalar)
CREATE FUNCTION FN_CalcularTiempoTotal (
    @PreparacionMin INT,
    @CoccionMin INT
)
RETURNS INT
AS
BEGIN
    RETURN @PreparacionMin + @CoccionMin;
END
GO

```

```

-- 5.2. FN_FormatearNivelDificultad (Escalar)
CREATE FUNCTION FN_FormatearNivelDificultad (
    @Nivel NVARCHAR(50)
)
RETURNS NVARCHAR(50)
AS
BEGIN
    DECLARE @Resultado NVARCHAR(50);
    SET @Resultado =

```

```

        CASE @Nivel
            WHEN 'Fácil' THEN '*' (Principiante)'
            WHEN 'Medio' THEN '** (Intermedio)'
            WHEN 'Difícil' THEN '*** (Avanzado)'
            ELSE 'N/A'
        END;
    RETURN @Resultado;
END
GO

```

```

-- 5.3. FN_BuscarRecetasPorIngrediente (TVF)

```

```

CREATE FUNCTION FN_BuscarRecetasPorIngrediente (
    @NombreIngrediente NVARCHAR(150)
)
RETURNS TABLE
AS
RETURN
(
    SELECT
        R.RecetaID,
        R.Titulo,
        RI.Cantidad,
        RI.UnidadMedida
    FROM
        Recetas R
    JOIN
        RecetaIngredientes RI ON R.RecetaID = RI.RecetaID
    JOIN
        Ingredientes I ON RI.IngredienteID = I.IngredienteID
    WHERE
        I.NombreIngrediente = @NombreIngrediente
);
GO

```

```

-- =====
-- 6. CREACIÓN DE PROCEDIMIENTOS ALMACENADOS (Stored Procedures)
-- =====

```

```

-- 6.1. SP_CrearReceta

```

```

CREATE PROCEDURE SP_CrearReceta
    @Titulo NVARCHAR(200),
    @Descripcion NVARCHAR(MAX),
    @TiempoPreparacionMin INT,
    @TiempoCoccionMin INT,
    @Porciones INT,
    @NivelDificultad NVARCHAR(50),
    @NotaChef NVARCHAR(MAX) = NULL,
    @CategoriaID INT,
    @UsuarioID INT,
    @NuevoRecetaID INT OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    -- Validaciones... (omitiendo por brevedad, asumidas en el código anterior)

```

```

        INSERT INTO Recetas (Titulo, Descripcion, TiempoPreparacionMin,
TiempoCoccionMin, Porciones, NivelDificultad, NotaChef, CategoriaID, UsuarioID)
        VALUES (@Titulo, @Descripcion, @TiempoPreparacionMin, @TiempoCoccionMin,
@Porciones, @NivelDificultad, @NotaChef, @CategoriaID, @UsuarioID);
        SET @NuevoRecetaID = SCOPE_IDENTITY();
END
GO

```

-- 6.2. SP\_EliminarReceta

```

CREATE PROCEDURE SP_EliminarReceta
    @RecetaID INT

```

AS

BEGIN

```

    SET NOCOUNT ON;

```

```

    -- Debe eliminar primero las dependencias por clave externa

```

```

    DELETE FROM RecetaIngredientes WHERE RecetaID = @RecetaID;

```

```

    DELETE FROM PasosPreparacion WHERE RecetaID = @RecetaID;

```

```

    DELETE FROM Recetas WHERE RecetaID = @RecetaID;

```

END

GO

-- 6.3. SP\_AgregarIngredienteAReceta

```

CREATE PROCEDURE SP_AgregarIngredienteAReceta

```

```

    @RecetaID INT,

```

```

    @IngredienteID INT,

```

```

    @Cantidad DECIMAL(10, 2),

```

```

    @UnidadMedida NVARCHAR(50)

```

AS

BEGIN

```

    SET NOCOUNT ON;

```

```

    IF EXISTS (SELECT 1 FROM RecetaIngredientes WHERE RecetaID = @RecetaID AND
IngredienteID = @IngredienteID)

```

```

        BEGIN

```

```

            UPDATE RecetaIngredientes

```

```

            SET Cantidad = @Cantidad, UnidadMedida = @UnidadMedida

```

```

            WHERE RecetaID = @RecetaID AND IngredienteID = @IngredienteID;

```

```

        END

```

```

        ELSE

```

```

        BEGIN

```

```

            INSERT INTO RecetaIngredientes (RecetaID, IngredienteID, Cantidad,
UnidadMedida)

```

```

            VALUES (@RecetaID, @IngredienteID, @Cantidad, @UnidadMedida);

```

```

        END

```

END

GO

-- 6.4. SP\_AgregarPasoPreparacion

```

CREATE PROCEDURE SP_AgregarPasoPreparacion

```

```

    @RecetaID INT,

```

```

    @Instruccion NVARCHAR(MAX),

```

```

    @NumeroPaso INT = NULL

```

AS

BEGIN

```

    SET NOCOUNT ON;

```

```
IF @NumeroPaso IS NULL
BEGIN
    SELECT @NumeroPaso = ISNULL(MAX(NumeroPaso), 0) + 1
    FROM PasosPreparacion
    WHERE RecetaID = @RecetaID;
END
INSERT INTO PasosPreparacion (RecetaID, NumeroPaso, Instruccion)
VALUES (@RecetaID, @NumeroPaso, @Instruccion);
END
GO

-- =====
-- 7. FIN DEL SCRIPT DE DESPLIEGUE
-- =====
```