

```
-- Creacion de la base de datos
```

```
CREATE DATABASE FINAL_BDA_4TN_DiDomenico;
```

```
USE FINAL_BDA_4TN_DiDomenico;
```

```
-- Creacion de las tablas
```

```
CREATE TABLE tipos_comprobantes (  
    id_tipo_comprobante INT PRIMARY KEY,  
    descrip_comprobante NVARCHAR(100)  
);
```

```
CREATE TABLE tipos_documento (  
    id_tipo_documento INT PRIMARY KEY,  
    descrip_documento NVARCHAR(100)  
);
```

```
CREATE TABLE clientes (  
    id_cliente INT PRIMARY KEY,  
    apellido_nombre NVARCHAR(100),  
    id_tipo_documento INT,  
    nro_documento INT,  
    FOREIGN KEY (id_tipo_documento) REFERENCES tipos_documento(id_tipo_documento)  
);
```

```
CREATE TABLE comprobantes (  
    nro_comprobante INT PRIMARY KEY,  
    fecha DATE,  
    id_tipo_comprobante INT,  
    id_cliente INT,  
    importe DECIMAL(10,2),  
    FOREIGN KEY (id_tipo_comprobante) REFERENCES tipos_comprobantes  
        (id_tipo_comprobante),  
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)  
);
```

```
-- Carga de los datos
```

```
INSERT INTO tipos_documento (id_tipo_documento, descrip_documento)  
VALUES
```

```
(0, 'Ningun Documento'),  
(1, 'D.N.I.'),  
(2, 'C.U.I.L.'),  
(3, 'C.U.I.T.'),  
(4, 'Cedula de Identificacion'),  
(5, 'Pasaporte'),  
(6, 'Libreta Civica'),  
(7, 'Libreta de Enrolamiento');
```

```
INSERT INTO tipos_comprobantes (id_tipo_comprobante, descrip_comprobante)  
VALUES
```

```
(1, 'Factura A'),  
(2, 'Nota Debito A'),  
(3, 'Nota Credito A'),  
(4, 'Recibos A'),  
(5, 'Nota Venta Contado A'),  
(6, 'Factura B'),  
(7, 'Nota Debito B'),  
(8, 'Nota Credito B'),  
(9, 'Recibo B'),  
(10, 'Nota Venta Contado B');
```

```

(11, 'Facturas C'),
(12, 'Nota Debito C'),
(13, 'Nota Credito C'),
(15, 'Recibo C'),
(16, 'Nota Venta Contado C'),
(81, 'Tique Factura A'),
(82, 'Tique Factura B'),
(83, 'Tique'),
(111, 'Tique Factura C'),
(112, 'Tique Nota Credito A'),
(113, 'Tique Nota Credito B'),
(114, 'Tique Nota Credito C');

INSERT INTO clientes (id_cliente, apellido_nombre, id_tipo_documento,
nro_documento)
VALUES
(1, 'Garcia, Luis', 1, 98765432),
(2, 'Mendez, Sofia', 2, 87654321),
(3, 'Hernandez, Pablo', 4, 76543210),
(4, 'Ruiz, Elena', 3, 65432109),
(5, 'Vazquez, Miguel', 1, 54321098),
(6, 'Romero, Laura', 1, 43210987),
(7, 'Silva, Carlos', 1, 32109876),
(8, 'Molina, Ana', 7, 21098765),
(9, 'Santos, Jorge', 6, 10987654),
(10, 'Nieves, Carmen', 5, 19876543);

INSERT INTO comprobantes (nro_comprobante, fecha, id_tipo_comprobante, id_cliente,
importe)
VALUES
(1, '2023-01-01', 1, 1, 100.00),
(2, '2023-02-02', 2, 2, 200.00),
(3, '2023-03-03', 3, 3, 300.00),
(4, '2023-04-04', 4, 4, 400.00),
(5, '2023-05-05', 5, 5, 500.00),
(6, '2023-06-06', 6, 6, 600.00),
(7, '2023-07-07', 7, 7, 700.00),
(8, '2023-08-08', 8, 8, 800.00),
(9, '2023-09-09', 9, 9, 900.00),
(10, '2023-10-10', 10, 10, 1000.00),
(11, '2023-11-11', 11, 1, 1100.00),
(12, '2023-12-12', 12, 2, 1200.00),
(13, '2024-01-13', 13, 3, 1300.00),
(14, '2024-02-14', 15, 4, 1400.00),
(15, '2024-03-15', 16, 5, 1500.00),
(16, '2024-04-16', 81, 6, 1600.00),
(17, '2024-05-17', 82, 7, 1700.00),
(18, '2024-06-18', 83, 8, 1800.00),
(19, '2024-07-01', 111, 9, 1900.00),
(20, '2024-07-10', 112, 10, 2000.00);

-- UAI_ARCHIVO
-- Generacion de TXT practico - 4to Ing. Sistemas
CREATE PROCEDURE usp_GenerarTXT @clienteId INT, @fileName NVARCHAR(255)
AS
BEGIN

```

```

DECLARE @text VARCHAR(118);
DECLARE @cmd VARCHAR(1000);

SELECT
    @text = CONCAT(
        FORMAT(c.fecha, 'yyyyMMdd'),
        RIGHT('000' + CAST(tc.id_tipo_comprobante AS NVARCHAR), 3),
        RIGHT('000000000000000000' + CAST(c.nro_comprobante AS NVARCHAR), 20),
        RIGHT('000000000000000000' + CAST(c.nro_comprobante AS NVARCHAR), 20),
        RIGHT('00' + td.id_tipo_documento, 2),
        RIGHT('000000000000000000' + CAST(cl.nro_documento AS NVARCHAR), 20),
        LEFT(cl.apellido_nombre + SPACE(30), 30),
        RIGHT('0000000000000000' + REPLACE(FORMAT(c.importe, '0.00'), '.', ''), 15)
    )
FROM comprobantes c
INNER JOIN clientes cl
ON c.id_cliente = cl.id_cliente AND cl.id_cliente = @clienteId
INNER JOIN tipos_comprobantes tc
ON c.id_tipo_comprobante = tc.id_tipo_comprobante
INNER JOIN tipos_documento td
ON cl.id_tipo_documento = td.id_tipo_documento;

IF @text IS NOT NULL
BEGIN
    SET @cmd = 'echo ' + @text + ' > C:\Users\Nicol\Desktop\UAI-2024\BDA\FINAL_BDA\' + @fileName + '.txt';
    EXEC xp_cmdshell @cmd, no_output;
END
ELSE
BEGIN
    PRINT 'No se encontraron comprobantes para el cliente especificado.';
END
END

-- EXEC generarTXT @clienteId = 1, @fileName = 'nombre_del_archivo';

-- UAI_VISTA
-- Creacion de la Vista Funcional al modelo
CREATE VIEW vw_Comprobante AS
SELECT
    c.nro_comprobante,
    c.fecha,
    tc.descripcion_comprobante AS descripcion_comprobante,
    cl.apellido_nombre AS nombre_y_apellido,
    c.importe
FROM comprobantes c
INNER JOIN clientes cl
ON c.id_cliente = cl.id_cliente
INNER JOIN tipos_comprobantes tc
ON c.id_tipo_comprobante = tc.id_tipo_comprobante;
--SELECT * FROM vw_Comprobante;

-- UAI_SP

```

```
-- SP QUE MUESTRE EL TOTAL FACTURADO EN UN RANGO DE FECHAS
CREATE PROCEDURE usp_TotalFacturado
    @fechaInicio DATE,
    @fechaFin DATE
AS
BEGIN
    SELECT SUM(importe) AS TotalFacturado
    FROM comprobantes
    WHERE fecha BETWEEN @fechaInicio AND @fechaFin;
END
-- EXEC TotalFacturado @fechaInicio = '2023-01-01', @fechaFin = '2023-12-31';

-- UAI_FN
-- FUNCION QUE MUESTRE EL COMPROBANTE CON SUS DATOS ORIGINALES MAS LA BASE IMPONIBLE DEL MISMO (IVA 21%)
CREATE FUNCTION fn_ObtenerComprobanteConImpuesto(@comprobanteID INT)
RETURNS TABLE
AS
RETURN
(
    SELECT
        c.nro_comprobante, -- Número de comprobante
        c.fecha, -- Fecha del comprobante
        c.id_tipo_comprobante, -- Tipo de comprobante
        c.id_cliente, -- Cliente
        c.importe, -- Importe del comprobante
        c.importe * 1.21 AS BaseImponible -- Base imponible con IVA del 21%
    FROM comprobantes AS c
    WHERE c.nro_comprobante = @comprobanteID
);
-- SELECT * FROM fn_ObtenerComprobanteConImpuesto(1);

-- UAI_TR
-- Agregar el atributo acumulado de compras a la entidad clientes y realizar un triggers que actualice este de forma automatica en cada carga de comprobante
ALTER TABLE clientes
ADD acumulado_compras DECIMAL(18, 2) DEFAULT 0;

CREATE TRIGGER Tr_ActualizarAcumuladoCompras
ON comprobantes
AFTER INSERT
AS
BEGIN
    UPDATE clientes
    SET acumulado_compras = acumulado_compras + i.importe
    FROM clientes c
    INNER JOIN inserted i ON c.id_cliente = i.id_cliente;
END
```