

PROYECTO FINAL

Integrantes:

Dantar Alejandro Ortiz Vega

José Ramon Preciado Torres



Introducción: Explicaremos a detalle lo que hicimos en nuestro proyecto de SQL Server a mongodb haciéndose que se visualizara en Visual Studio Code siguiendo los datos proporcionados por la maestra haciendo

1.-----Página 1 a Página 5

2.-----Pagina 5 a Página 6

3.-----Página 6 a Página 8

4.-----Página 8 a Página 14

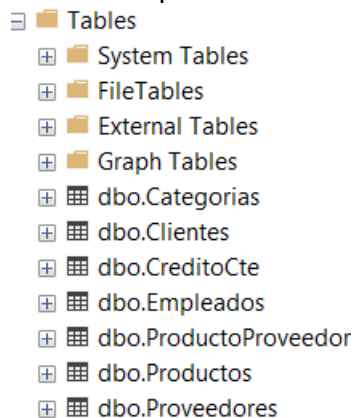
Conclusiones-----Página 15

1.- Exportación de SQL Server a Excel

- Tenemos creada nuestra base de datos en SQL Server



- Con sus respectivas tablas:



- Sus respectivos campos en código

```

CREATE TABLE Empleados(
    Id_Emp VARCHAR(8) PRIMARY KEY not null,
    Nombre_Emp VARCHAR(30) not null,
    Apellido_Emp VARCHAR(30) not null,
    Direccion_Emp TEXT,
    Tel_Emp BIGINT not null
);

CREATE TABLE Clientes(
    Id_Cte VARCHAR(8) PRIMARY KEY not null,
    Nombre_Cte VARCHAR(30) not null,
    Apellido_Cte VARCHAR(30) not null,
    Direccion_Cte TEXT,
    Tel_Cte BIGINT not null
);

CREATE TABLE Categorias(
    Id_Categoria INT PRIMARY KEY not null,
    Descripcion VARCHAR(30) not null
);

CREATE TABLE Productos(
    Id_Producto VARCHAR(8) PRIMARY KEY not null,
    Nombre_Prod VARCHAR(30) not null,
    CategoriaId INT not null,
    Color VARCHAR(20) not null,
    Talla VARCHAR(8) not null,
    Precio MONEY not null

    CONSTRAINT fk_Categorias FOREIGN KEY (CategoriaId) REFERENCES Categorias(Id_Categoria)
);

CREATE TABLE Proveedores(
    Id_Proveedor VARCHAR(8) PRIMARY KEY not null,
    Nombre VARCHAR(30) not null,
    Direccion TEXT,
    Correo VARCHAR(30) not null,
    Tel_Prov BIGINT not null
);

CREATE TABLE CreditoCte(
    CteId VARCHAR(8) not null,
    ProdId VARCHAR(8) not null,
    Cant_Debe MONEY not null,
    Fecha_Compra DATE,
    Unidades INT not null,
    EmpId VARCHAR(8) not null

    CONSTRAINT fk_Clientes FOREIGN KEY (CteId) REFERENCES Clientes(Id_Cte),
    CONSTRAINT fk_Productos FOREIGN KEY (ProdId) REFERENCES Productos(Id_Producto),
    CONSTRAINT fk_Empleados FOREIGN KEY (EmpId) REFERENCES Empleados(Id_Emp)
);

CREATE TABLE ProductoProveedor(
    ProdId VARCHAR(8) PRIMARY KEY not null,
    ProvId VARCHAR(8) not null,
    Cant INT not null

    CONSTRAINT fk_Producto FOREIGN KEY(ProdId) REFERENCES Productos(Id_Producto),
    CONSTRAINT fk_Proveedores FOREIGN KEY(ProvId) REFERENCES Proveedores(Id_Proveedor)
);

```

- Y sus agregaciones hechas en SQL

```

INSERT INTO Empleados VALUES ('EM-01', 'Agustin', 'Hernandez', 'Av. Maximiliano Cepeda #8', 1237892850),
('EM-02', 'Rosalia', 'Sanchez', 'Jr. Irene Otero #6648', 2574125800),
('EM-03', 'Maria', 'Clemente', 'Cl. Ximena Ballesteros #7070 Piso 7', 7539514258),
('EM-04', 'Fermin', 'Lucena', 'Cl. Dylan Sáenz #3559 Piso 8', 7485147529),
('EM-05', 'John', 'Cuellar', 'Av. José Casanova #8', 7485413099),
('EM-06', 'Rocio', 'Aguilera', 'Urb. Camila Rivero #69', 9865236951),
('EM-07', 'Salud', 'Arana', 'Cl. Juan Manuel Piña #0511', 9875145844),
('EM-08', 'Jesus', 'Cabezas', 'Av. Oliva Alaniz #9207', 3215684554);

INSERT INTO Clientes VALUES('C-01', 'Pere', 'Rodenas', 'Jr. Nadia Guardado #6', 3121458521),
('C-02', 'Clotilde', 'Alfonso', 'Av. Valentina Sanches #637', 3129863830),
('C-03', 'Joshua', 'Porrás', 'Urb. Camila Godínez #37 Piso 1', 31210926731),
('C-04', 'Matias', 'Navarro', 'Jr. Juan Sedillo #820', 3124584245),
('C-05', 'Valentin', 'Rosales', 'Jr. Magdalena Ríos #422 Piso 1', 3128456947),
('C-06', 'Gabriel', 'Chacon', 'Jr. Isidora Méndez #1', 3127485474),
('C-07', 'Silvana', 'Carmona', 'Jr. Ricardo Ornelas #9 Dpto. 539', 3127485485),
('C-08', 'Francisco', 'Leal', 'Av. Fabiana Crespo #28', 3129653262);

INSERT INTO Categorías VALUES(1, 'Pantalones'),
(2, 'Abrigos'),
(3, 'Zapatos'),
(4, 'Camisas'),
(5, 'Shorts');

INSERT INTO Productos VALUES('PR-01', 'Jeans Refill Jogger', 1, 'Azul', 'Ch', 150.00),
('PR-02', 'Zapatillas Frida Kollection', 3, 'Verde', 'Gd', 200.00),
('PR-03', 'Abrigo Sahara', 2, 'cafe', 'Ch', 400.00),
('PR-04', 'Short Lady Sun', 5, 'Blanco', 'M', 900.00),
('PR-05', 'Blusa 18 Forever', 4, 'Beige', 'Ch', 870.00),
('PR-06', 'Camisa Casual Refill', 4, 'Rayas', 'M', 640.00),
('PR-07', 'Pantalon Dockers', 1, 'Negro', 'Gd', 510.00),
('PR-08', 'Short Puma', 5, 'Gris', 'M', 470.00),
('PR-09', 'Tenis Nike Air Max', 3, 'Blanco', 'Ch', 790.00),
('PR-10', 'Abrigo Wall Street', 2, 'Negro', 'Gd', 680.00),
('PR-11', 'Zapatos Brantano', 3, 'Negro', 'Ch', 470.00);

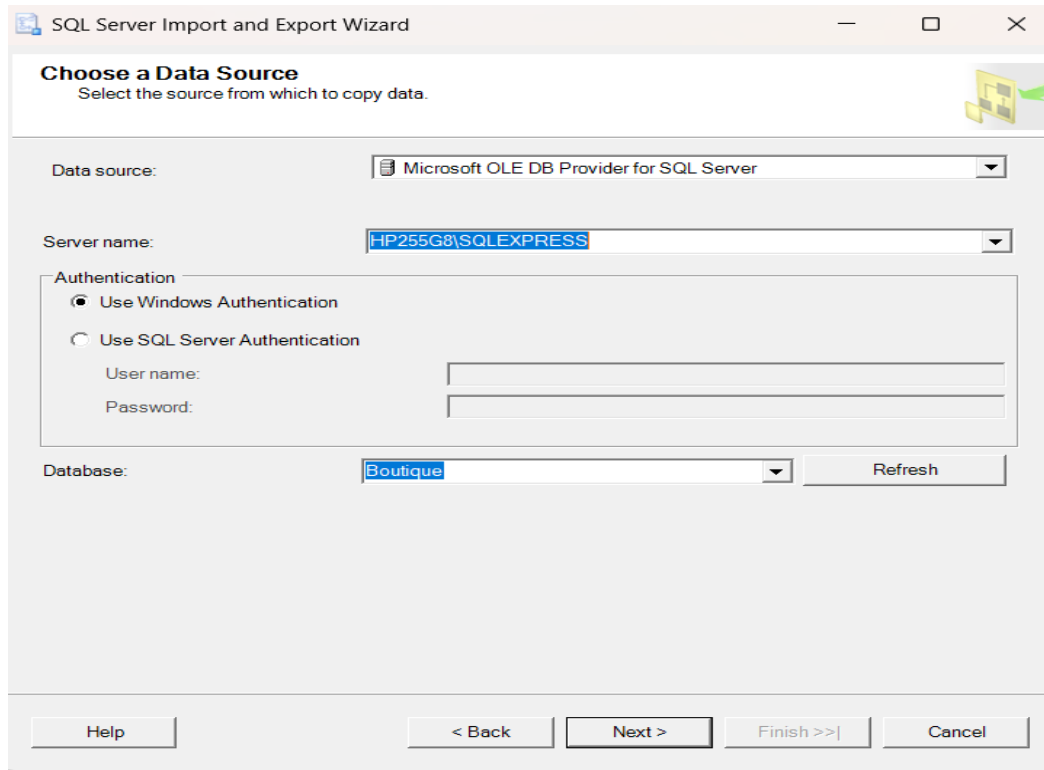
INSERT INTO Proveedores VALUES('PV-01', 'Sugarlips', 'Urb. Alex Del Valle #76868', 'sugarlips@gmail.com', 3121234567),
('PV-02', 'LASHshowroom', 'Av. Luana Medrano #575 Piso 42', 'lashowroom@gmail.com', 3127894561),
('PV-03', 'HanaFashion', 'Av. Christopher Arevalo #785 Hab. 822', 'hanafashion@gmail.com', 3124568521),
('PV-04', 'ZaaSmart', 'Cl. Javier Gaytán #38713 Dpto. 951', 'zaasmart@gmail.com', 3124865247),
('PV-05', 'Tekstila', 'Cl. Maite Marín #17', 'tekstila@gmail.com', 3127895623),
('PV-06', 'Modastock', 'Av. Mariangel Almanza #7106 Piso 72', 'modastock@gmail.com', 3127419623),
('PV-07', 'Ganga Lotes', 'Cl. Alex Córdoba #71038 Piso 6', 'gangalotes@gmail.com', 3128457485),
('PV-08', 'BrandsGateway', 'Urb. Anthony Vega # 7496', 'brandsgateway@gmail.com', 3129586952);

INSERT INTO CreditoCte VALUES('C-01', 'PR-08', 1410, '2023/05/11', 3, 'EM-01'),
('C-01', 'PR-10', 680, '2023/05/21', 1, 'EM-02'),
('C-03', 'PR-11', 940, '2023/02/05', 2, 'EM-03'),
('C-07', 'PR-01', 450, '2023/05/14', 3, 'EM-03'),
('C-03', 'PR-05', 870, '2023/01/17', 1, 'EM-04'),
('C-02', 'PR-05', 1740, '2023/05/14', 2, 'EM-05'),
('C-04', 'PR-06', 1280, '2023/05/07', 2, 'EM-01'),
('C-06', 'PR-07', 1530, '2023/11/08', 3, 'EM-06'),
('C-07', 'PR-04', 900, '2023/12/22', 1, 'EM-07'),
('C-02', 'PR-02', 600, '2023/03/21', 3, 'EM-01');

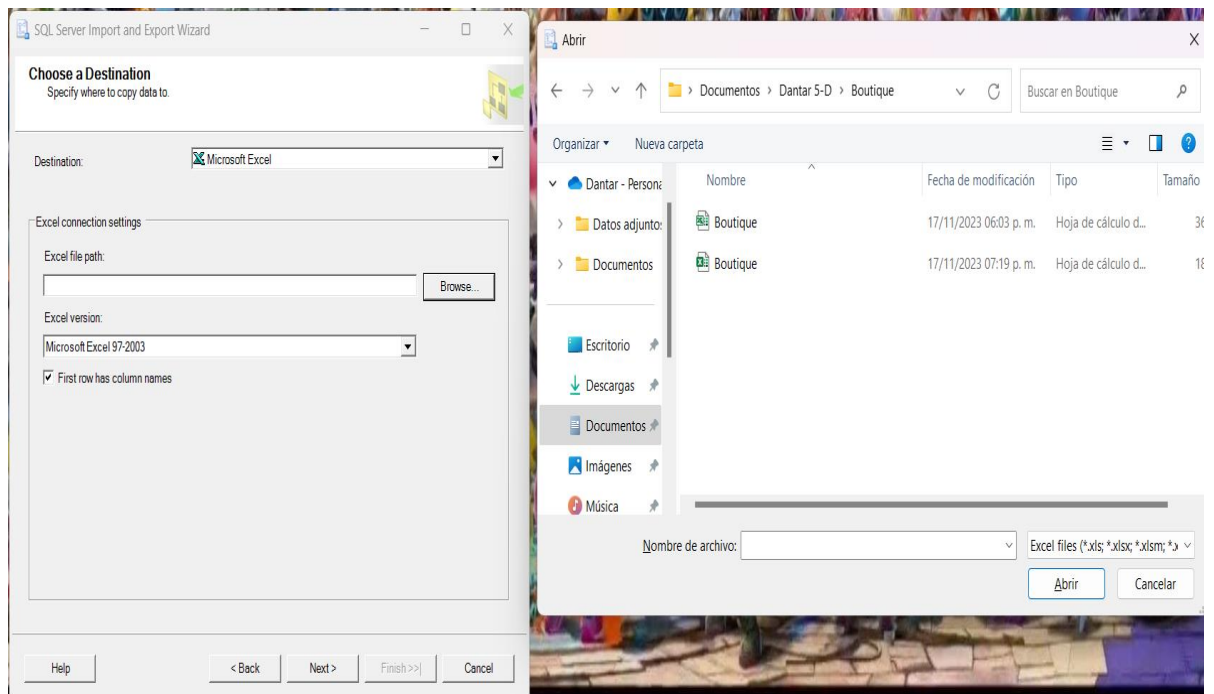
INSERT INTO CreditoCte VALUES('C-07', 'PR-01', 450, '2023/05/14', 3, 'EM-03');
INSERT INTO ProductoProveedor VALUES('PR-01', 'PV-01', 70),
('PR-02', 'PV-01', 50),
('PR-03', 'PV-02', 68),
('PR-04', 'PV-04', 100),
('PR-05', 'PV-03', 40),
('PR-06', 'PV-04', 75),
('PR-07', 'PV-05', 60),
('PR-08', 'PV-06', 70),
('PR-09', 'PV-07', 90),
('PR-10', 'PV-08', 85),
('PR-11', 'PV-08', 100);

```

- Exportamos la base de SQL a Excel



- Se crea un documento de Excel con el nombre de la base de datos



Uno es el archivo en formato Excel 97-2003 para la exportación de SQL a Excel, luego ese archivo se convierte al formato tradicional de Excel para su lectura en mongo








Se exportaron correctamente cada tabla de la base de datos con sus respectivos campos

- Nos vamos a guardar como y le ponemos de nombre, el nombre de la tabla que queramos exportar

Es importante convertirlo al formato mostrado en pantalla y posicionarnos en el campo que deseamos convertir (CSV UTF-8 delimitado por comas)



- Hacemos lo mismo con cada tabla

 Categorías	18/11/2023 07:55 p. m.	Archivo de valores...	1 KB
 Clientes	18/11/2023 07:57 p. m.	Archivo de valores...	1 KB
 CreditoCte	18/11/2023 10:52 p. m.	Archivo de valores...	1 KB
 Empleados	18/11/2023 10:55 p. m.	Archivo de valores...	1 KB
 ProductoProveedor	18/11/2023 10:58 p. m.	Archivo de valores...	1 KB
 Productos	18/11/2023 11:07 p. m.	Archivo de valores...	1 KB
 Proveedores	18/11/2023 11:09 p. m.	Archivo de valores...	1 KB

3.-MONGODB-EXCEL

- Creamos la base de datos en mongodb y vamos creando colección por colección

Create Database



Database Name

Boutique

Collection Name

Categorías

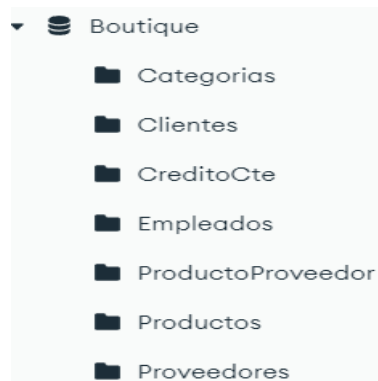
☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

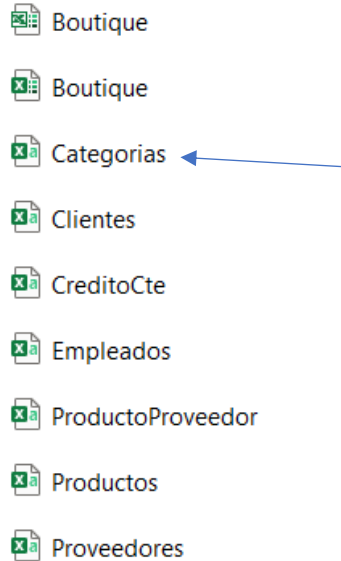


Así quedaría la base hecha en mongodb

- El guardar en el formato delimitado por comas era para importar los datos a las colecciones



- Le damos en ADD DATA, seguido de Import file y nos despliega el navegador de documentos



- Así queda, convierte los datos del Excel al tipo de dato que maneja mongodb, hacemos lo mismo con cada colección hecha en mongodb

Specify Fields and Types [Learn more about data types](#)

	<input checked="" type="checkbox"/> Id_Categoria	<input checked="" type="checkbox"/> Descripcion
	Int32	String
1	1	Pantalones
2	2	Abrigos
3	3	Zapatos
4	4	Camisas
5	5	Shorts

```
_id: ObjectId('65596b3e6acc86ba3263095d')  
Id_Categoria: 1  
Descripcion: "Pantalones"
```

```
_id: ObjectId('65596b3e6acc86ba3263095e')  
Id_Categoria: 2  
Descripcion: "Abrigos"
```

```
_id: ObjectId('65596b3e6acc86ba3263095f')  
Id_Categoria: 3  
Descripcion: "Zapatos"
```

```
_id: ObjectId('65596b3e6acc86ba32630960')  
Id_Categoria: 4  
Descripcion: "Camisas"
```

```
_id: ObjectId('65596b3e6acc86ba32630961')  
Id_Categoria: 5  
Descripcion: "Shorts"
```

4.- Consultas Visual Studio Code

- Se hicieron las consultas hechas en SQL Server

#Obtener el nombre y apellido de los clientes que fueron atendidos por Agustín

-SQL Server

```
SELECT Clientes.Nombre Cte, Clientes.Apellido Cte, Productos.Nombre Prod  
FROM Clientes, CreditoCte, Empleados, Productos WHERE CreditoCte.EmpId = Empleados.Id_Emp  
AND Clientes.Id_Cte = CreditoCte.CteId AND CreditoCte.ProdId = Productos.Id_Producto AND Empleados.Nombre_Emp='Agustin';
```

-mongodb


```

Consulta1.mongodb > ...
Disconnected. Click here to connect.
db.CreditoCte.aggregate([
  {
    $match:
    {
      EmpId: db.Empleados.findOne({ Nombre_Emp: "Agustin" }).Id_Emp
    }
  },
  {
    $lookup:
    {
      from: "Clientes", localField: "CteId", foreignField: "Id_Cte", as: "cliente_info"
    }
  },
  {
    $project:
    {
      Nombre_Cte: "$cliente_info.Nombre_Cte", Apellido_Cte: "$cliente_info.Apellido_Cte"
    }
  }
])

```

#Obtener los productos comprados por Clotilde

-SQL Server

```

SELECT Clientes.Nombre_Cte, Clientes.Apellido_Cte, Productos.Nombre_Prod, Productos.Precio, CreditoCte.Unidades
FROM Clientes, CreditoCte, Productos WHERE Productos.Id_Producto = CreditoCte.ProdId AND CreditoCte.CteId = Clientes.Id_Cte AND Clientes.Nombre_Cte = 'Clotilde';

```

-Mongodb

```

use('Boutique')

db.CreditoCte.aggregate([
  {
    $match:
    {
      CteId: db.Clientes.findOne({ Nombre_Cte: "Clotilde" }).Id_Cte
    }
  },
  {
    $lookup:
    {
      from: "Productos", localField: "ProdId", foreignField: "Id_Producto", as: "producto_info"
    }
  },
  {
    $project:
    {
      Nombre_Prod: "$producto_info.Nombre_Prod"
    }
  }
])

```

#Obtener el nombre y apellido de los clientes que hicieron una compra entre \$700-1000

-SQL Server

```
SELECT Clientes.Nombre Cte, Clientes.Apellido Cte, Productos.Nombre Prod, CreditoCte.Unidades, (Productos.Precio * CreditoCte.Unidades) AS SubTotal
FROM Clientes, CreditoCte, Productos WHERE Clientes.Id Cte = CreditoCte.CteId
AND CreditoCte.ProdId = Productos.Id Producto AND (Productos.Precio * CreditoCte.Unidades) between 700 and 1000;
```

-mongodb

```
use('Boutique')

db.CreditoCte.aggregate([
  {
    $lookup: {
      from: "Productos",
      localField: "ProdId",
      foreignField: "Id_Producto",
      as: "producto_info"
    }
  },
  {
    $unwind: "$producto_info"
  },
  {
    $addFields: {
      "producto_info.Precio": {
        $replaceAll: {
          input: "$producto_info.Precio",
          find: ",",
          replacement: "."
        }
      }
    }
  },
  {
    $addFields: {
      "producto_info.Precio": {
        $replaceAll: {
          input: "$producto_info.Precio",
          find: " €",
          replacement: ""
        }
      }
    }
  }
])
```

```

    $addFields: {
      "TotalCompra": {
        $multiply: [ {
          $toDouble: "$producto_info.Precio"
        },
        "$Unidades"
      ]
    }
  },
  {
    $match: {
      $expr: {
        $and: [
          { $gte: [ "$TotalCompra", 700 ] },
          { $lte: [ "$TotalCompra", 1000 ] }
        ]
      }
    }
  },
  {
    $lookup: {
      from: "Clientes",
      localField: "CteId",
      foreignField: "Id_Cte",
      as: "cliente_info"
    }
  },
  {
    $unwind: "$cliente_info"
  },
  {
    $project: {
      Nombre_Cte: "$cliente_info.Nombre_Cte",
      Apellido_Cte: "$cliente_info.Apellido_Cte"
    }
  }
}

```

#Obtener el nombre y apellido de clientes que compraron en la categoría de zapatos

-SQL Server

```

SELECT Clientes.Nombre Cte, Clientes.Apellido Cte, Productos.Nombre Prod, Productos.Precio, CreditoCte.Unidades
FROM Clientes, CreditoCte, Productos, Categorías WHERE Clientes.Id Cte = CreditoCte.CteId AND CreditoCte.ProdId = Productos.Id Producto
AND Productos.CategoriaId = Categorías.Id Categoría AND Categorías.Descripción='Zapatos';

```

-mongodb

```

use('Boutique')
db.CreditoCte.aggregate([
  {
    $lookup:
    {
      from: "Productos", localField: "ProdId", foreignField: "Id_Producto", as: "producto_info"
    }
  },
  {
    $match:
    {
      "producto_info.CategoriaId": db.Categorias.findOne({ Descripcion: "Zapatos" }).Id_Categoria
    }
  },
  {
    $lookup:
    {
      from: "Clientes", localField: "CteId", foreignField: "Id_Cte", as: "cliente_info"
    }
  },
  {
    $project:
    {
      Nombre_Cte: "$cliente_info.Nombre_Cte", Apellido_Cte: "$cliente_info.Apellido_Cte"
    }
  }
])

```

#Obtener los productos comprados por Clotilde en la fecha 2023-05-14

-SQL Server

```

SELECT c.Nombre Cte, c.Apellido Cte, p.Nombre Prod, p.Precio, cr.Fecha Compra, cr.Unidades, (p.Precio*cr.Unidades) AS subtotal,
((p.Precio*cr.Unidades)*0.16) AS iva, ((p.Precio*cr.Unidades)+(p.Precio*cr.Unidades)*0.16)) AS total
FROM Clientes c INNER JOIN CreditoCte cr ON c.Id Cte = cr.CteId INNER JOIN Productos p ON p.Id Producto = cr.ProdId
WHERE cr.Fecha Compra = '2023-05-14' AND c.Nombre Cte = 'Clotilde';

```

```

SELECT c.Nombre Cte, c.Apellido Cte, cr.Fecha Compra, SUM(cr.Unidades) AS Unidades, SUM(p.Precio*cr.Unidades) AS subtotal,
SUM((p.Precio*cr.Unidades)*0.16) AS iva, SUM((p.Precio*cr.Unidades)+(p.Precio*cr.Unidades)*0.16)) AS total
FROM Clientes c INNER JOIN CreditoCte cr ON c.Id Cte = cr.CteId INNER JOIN Productos p ON p.Id Producto = cr.ProdId
WHERE cr.Fecha Compra = '2023-05-14' AND c.Nombre Cte = 'Clotilde' GROUP BY c.Nombre Cte, c.Apellido Cte, cr.Fecha Compra;

```

-mongodb

```

use('Boutique')

db.CreditoCte.aggregate([
  {
    $match:
    {
      CteId: db.Clientes.findOne({ Nombre_Cte: "Clotilde" }).Id_Cte,
      Fecha_Compra: new Date("2023-05-14")
    }
  },
  {
    $lookup:
    {
      from: "Productos", localField: "ProdId", foreignField: "Id_Producto", as: "producto_info"
    }
  },
  {
    $unwind: "$producto_info"
  },
  {
    $addFields:
    {
      "producto_info.Precio": { $replaceAll: { input: "$producto_info.Precio", find: ",", replacement: "." } }
    }
  },
  {
    $addFields:
    {
      "producto_info.Precio": { $replaceAll: { input: "$producto_info.Precio", find: " €", replacement: "" } }
    }
  },
  {
    $lookup:
    {
      from: "Clientes", localField: "CteId", foreignField: "Id_Cte", as: "cliente_info"
    }
  }
])

```

```

    from: "Clientes", localField: "CteId", foreignField: "Id_Cte", as: "cliente_info"
  }
},
{
  $unwind: "$cliente_info"
},
{
  $project: {
    Nombre_Cte: "$cliente_info.Nombre_Cte",
    Apellido_Cte: "$cliente_info.Apellido_Cte",
    Fecha_Compra: "$Fecha_Compra",
    Unidades: "$Unidades",
    Subtotal: { $multiply: [ { $toDouble: "$producto_info.Precio" }, "$Unidades" ] },
    IVA: { $multiply: [ { $multiply: [ { $toDouble: "$producto_info.Precio" }, "$Unidades" ] }, 0.16 ] },
    Total: { $multiply: [ { $multiply: [ { $toDouble: "$producto_info.Precio" }, "$Unidades" ] }, 1.16 ] }
  }
}
])

```

- Por ultimo los archivos que se usan para modificar, eliminar, agregar y encontrar en nuestra base de datos Boutique

Agregar

```
use('Boutique')

db.Empleados.insertOne({
  Id_Emp: "EM-09",
  Nombre_Emp: "Teresa",
  Apellido_Emp: "Lucio",
  Direccion_Emp: "asd",
  ⚡ Tel_Emp: 3124568891
})
```

Un ejemplo de cómo insertar a un empleado

Modificar

```
use('Boutique')

db.users.updateOne(
  {"_id": ObjectId('') },
  ⚡ { $set: {Nombre_Prod: "655998306acc86ba3263098d"}},
  {upsert: true}
)
```

Se agarro cualquier id de un producto su id seguirá siendo el mismo lo que cambiará será el nombre del producto y sus referencias

Eliminar

```
use('Boutique')

db.users.deleteOne(
  ⚡ {"_id": ObjectId('655998306acc86ba3263098e')}
)
```

Se necesita únicamente del id ya que detectara en que colección esta y lo eliminara

Encontrar

```
use('Boutique')

db.Clientes.find({})
```

Esta es una búsqueda simple desplegándose todos los clientes de la boutique, pero se puede usar como referencia para búsquedas diferentes en la base

CONCLUSIONES

Bien en conclusión en base a lo aprendido este semestre sobre las bases de datos no relacionales pudimos entender y manejar un modelo de base de datos con su respectivo gestor en este caso fue mongodb que después de haber usado SQL Server pues teníamos un entendimiento de las bases de datos haciéndonos menos difícil esto aunque por mi parte(Dantar Alejandro Ortiz Vega) tal vez se me complicaba un poco las consultas complejas como en SQL pero con un poco de practica se arreglaba y con las actividades hechas en esta segunda parcial que mejor para poder entender todo esto, mongodb es un gestor no relacional accesible para todos al igual que es entendible para los usuarios que entiendan las bases de datos convencionales.

Bibliografías:

<https://dochub.mongodb.org/> Para la documentación de mongodb

Y el material de EDUC proporcionado por la maestra Evangelista Salazar para las referencias de nuestro proyecto y como hacerlo en base a los criterios de evaluación.