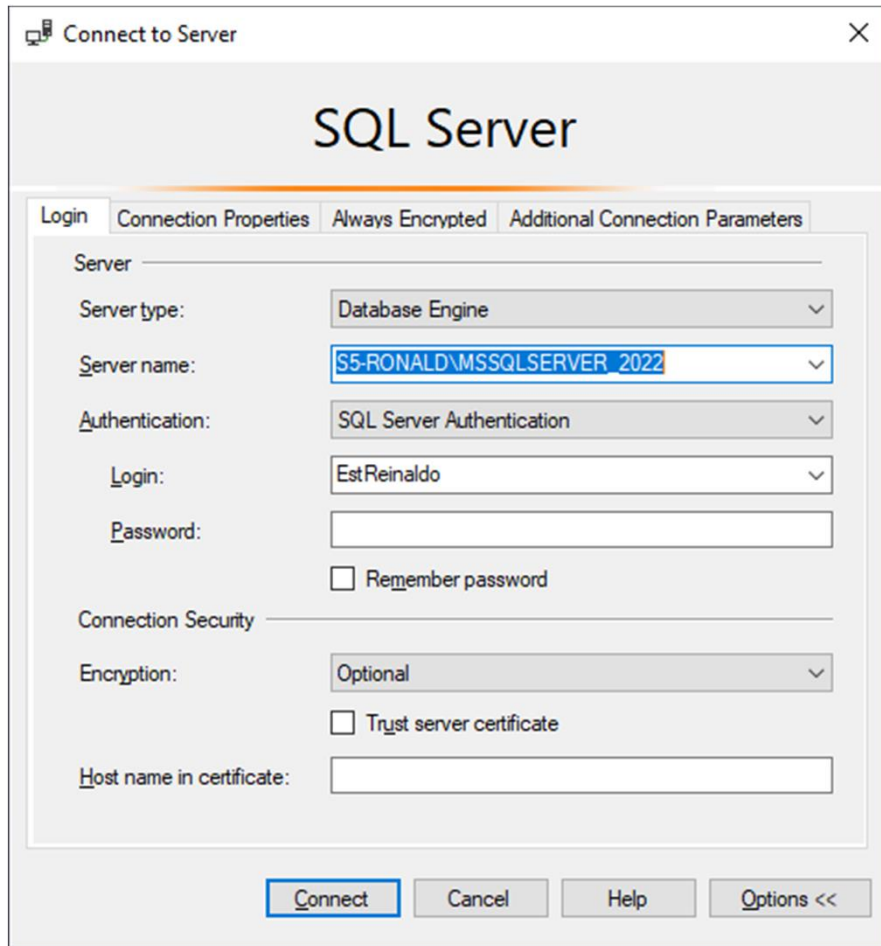


Base de Datos I

II. MODELO RELACIONAL Y CONSTRUCCIÓN DE BD

4.2. Management Studio – Creación de Base de Datos



The screenshot shows the 'Connect to Server' dialog box with the 'Login' tab selected. The dialog is titled 'SQL Server'. The 'Server' section contains the following fields:

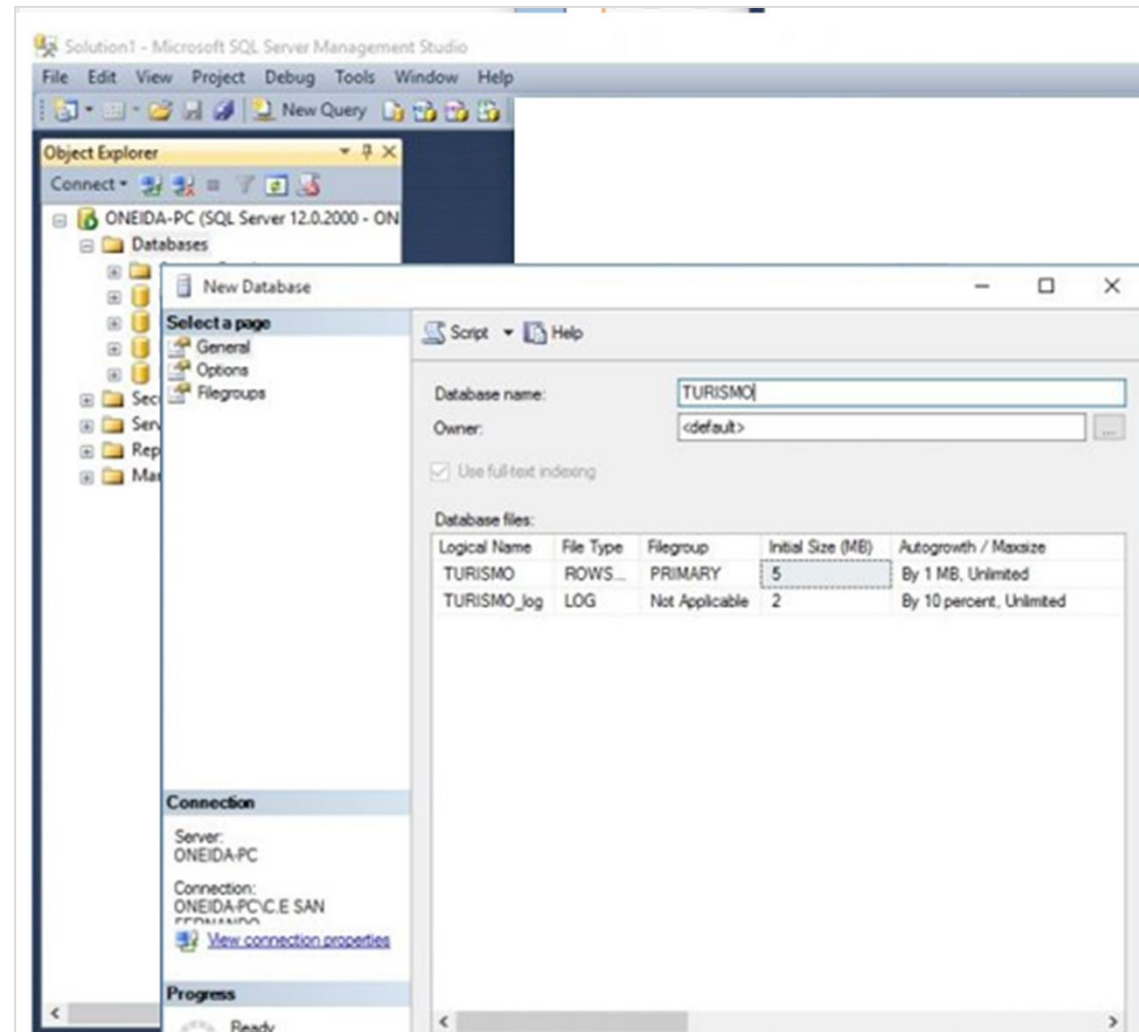
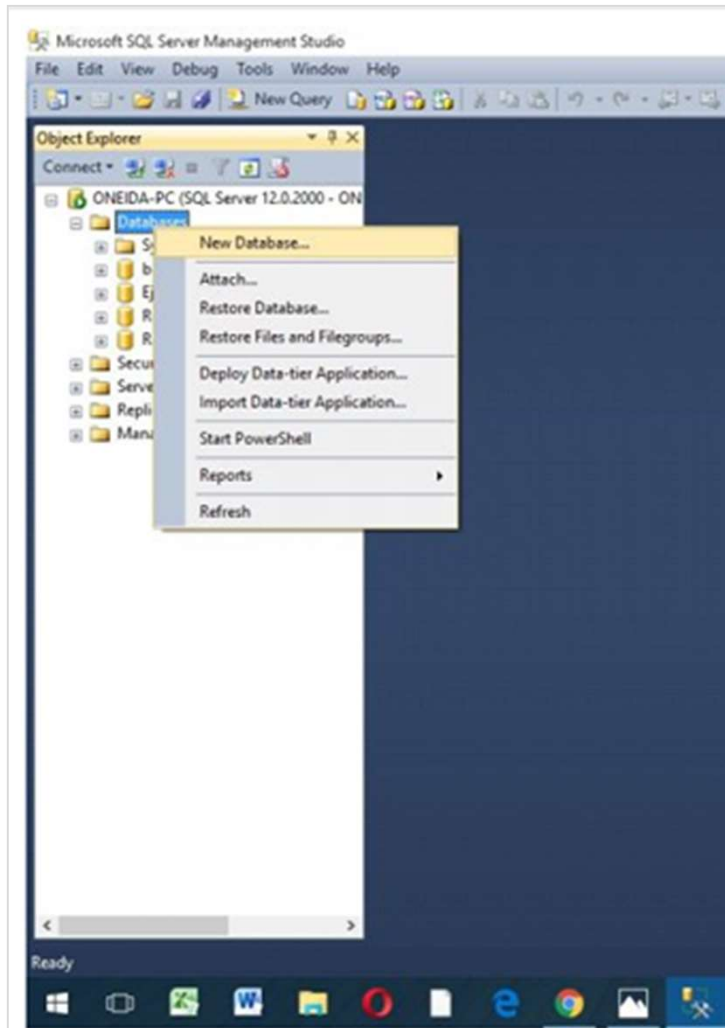
- Server type: Database Engine (dropdown)
- Server name: S5-RONALD\MSSQLSERVER_2022 (dropdown)
- Authentication: SQL Server Authentication (dropdown)
- Login: EstReinaldo (dropdown)
- Password: (text field)
- ☐ Remember password

The 'Connection Security' section contains the following fields:

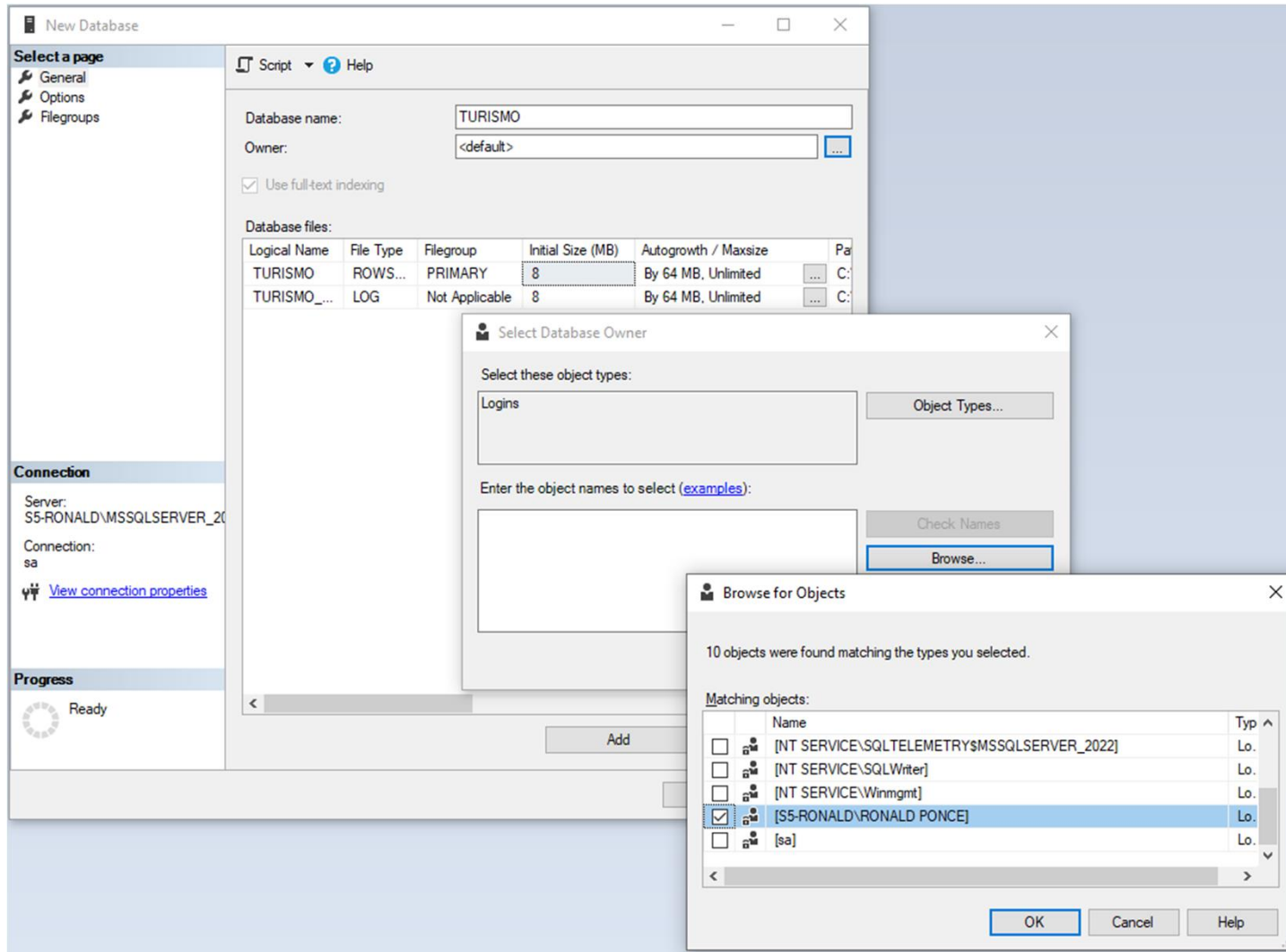
- Encryption: Optional (dropdown)
- ☐ Trust server certificate
- Host name in certificate: (text field)

At the bottom, there are four buttons: Connect, Cancel, Help, and Options <<.

4.2. Management Studio – Creación de Base de Datos



4.2. Management Studio – Creación de Base de Datos



4.2. Management Studio – Creación de Base de Datos

New Database

Select a page
General
Options
Filegroups

Script Help

Database name:

Owner:

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
TURISMO	ROWS...	PRIMARY	200	By 64 MB, Unlimited	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSER...
TURISMO_...	LOG	Not Applicable	40	By 64 MB, Unlimited	C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSER...

Connection
Server: S5-RONALD\MSSQLSERVER_20
Connection: sa
[View connection properties](#)

Progress
Ready

Add Remove

OK Cancel

4.2. Management Studio – Creación de Base de Datos

New Database

Select a page: General Options Filegroups

Script Help

Collation: Latin1_General_100_BIN2_UTF8

Recovery model: Full

Compatibility level: SQL Server 2022 (160)

Containment type: None

Other options:

Full Text Search Full Text Search Index Full Text Search Indexes

State

Database Read-Only	False
Database State	NORMAL
Encryption Enabled	False
Restrict Access	MULTI_USER

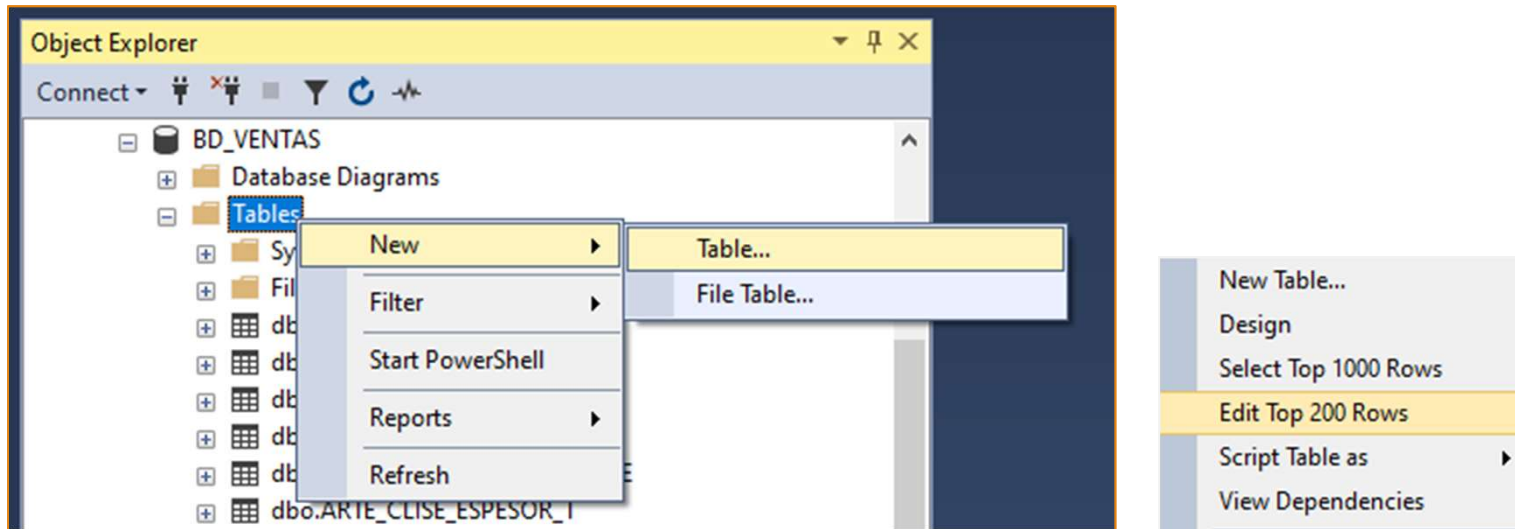
Restrict Access

MULTI_USER
SINGLE_USER
RESTRICTED_USER

OK Cancel

4.2. Management Studio – Creación de Tablas de Datos

1. En SSMS, en el **Explorador de objetos**, conéctese a la instancia de Motor de base de datos que contiene la base de datos que se va a modificar.
2. En el **Explorador de objetos**, expanda el nodo **Bases de datos** y, a continuación, expanda la base de datos que contendrá la nueva tabla.
3. En Explorador de objetos, haga clic con el botón derecho en el nodo **Tablas** de la base de datos. Seleccione **Nueva tabla**.



4. Haga click sobre la Tabla creada y seleccione “**Design**” para escribir los nombres de columna, elija los tipos de datos y elija si desea permitir valores NULL o NOT NULL para cada columna
5. Si desea ver los datos de la Tabla creada seleccione “**Edit top 200...**” como se muestra en la ilustración siguiente:

4.2. Management Studio – Creación de Tablas de Datos

Para especificar más propiedades para una columna, como valores de identidad o columna calculada, seleccione la columna y, en la pestaña propiedades de columna, elija las propiedades adecuadas. (SQL Server Management Studio).

Para especificar una columna como clave principal, haga clic con el botón derecho en la columna y seleccione **Establecer clave principal**. ([Create Primary Keys](#)).

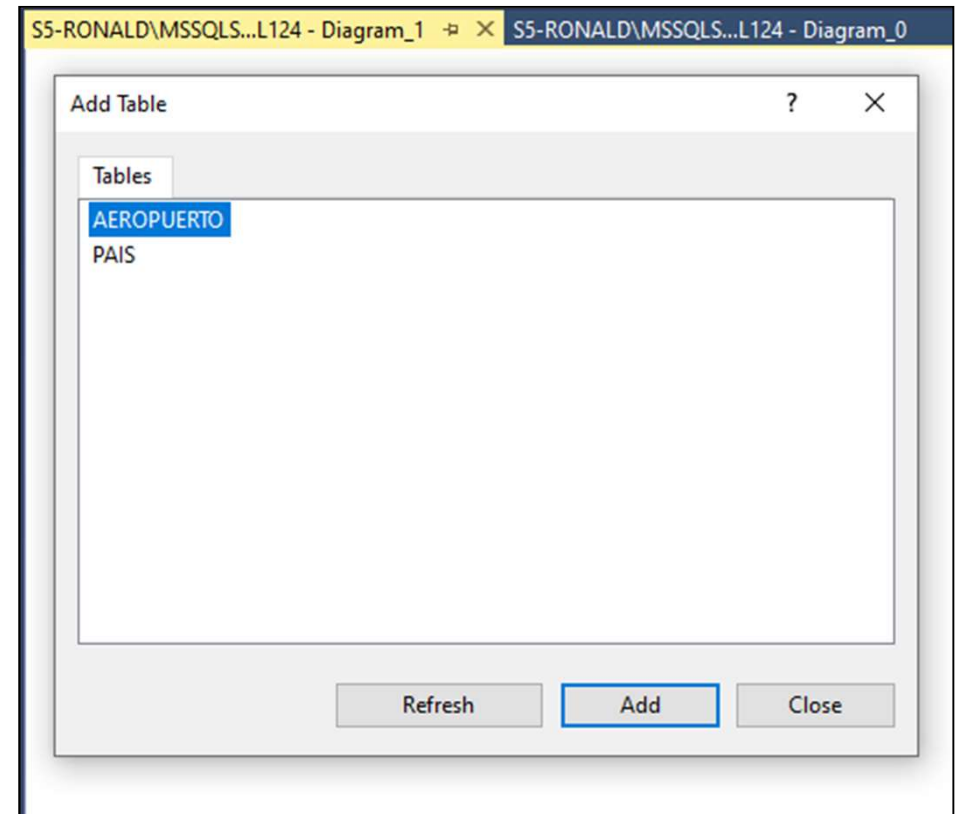
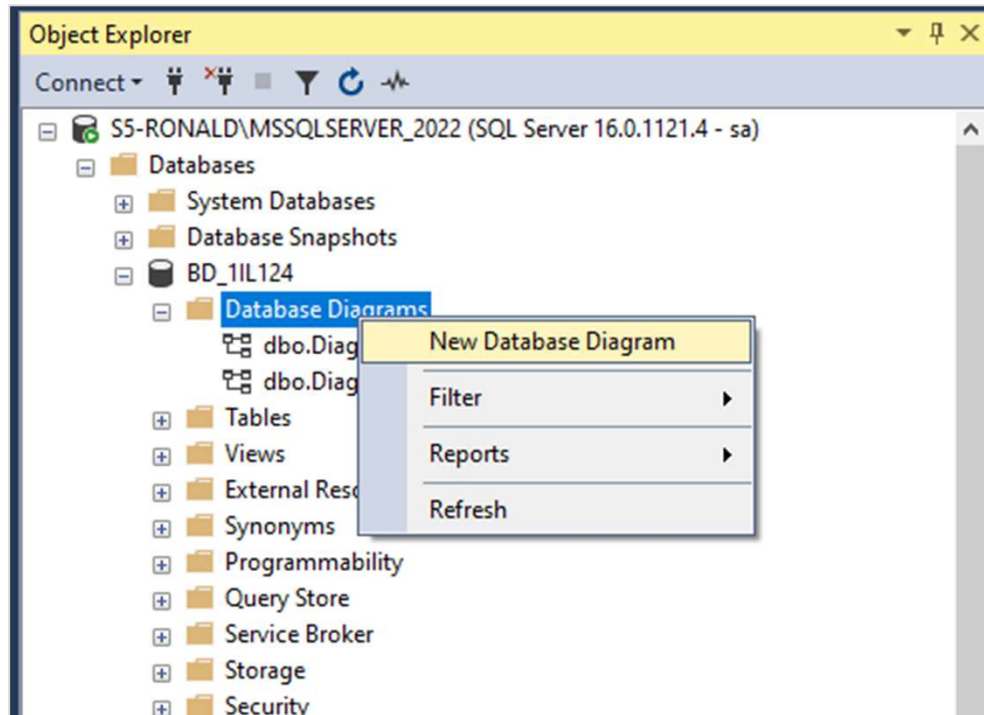
Para crear relaciones de clave externa, restricciones **CHECK** o índices, haga clic con el botón secundario en el panel Diseñador de tablas y seleccione un objeto de la lista como se muestra en la ilustración siguiente

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
FirstName	nvarchar(30)	<input type="checkbox"/>
LastName	nvarchar(50)	<input type="checkbox"/>
▶ ModifiedDate	smalldatetime	<input checked="" type="checkbox"/>

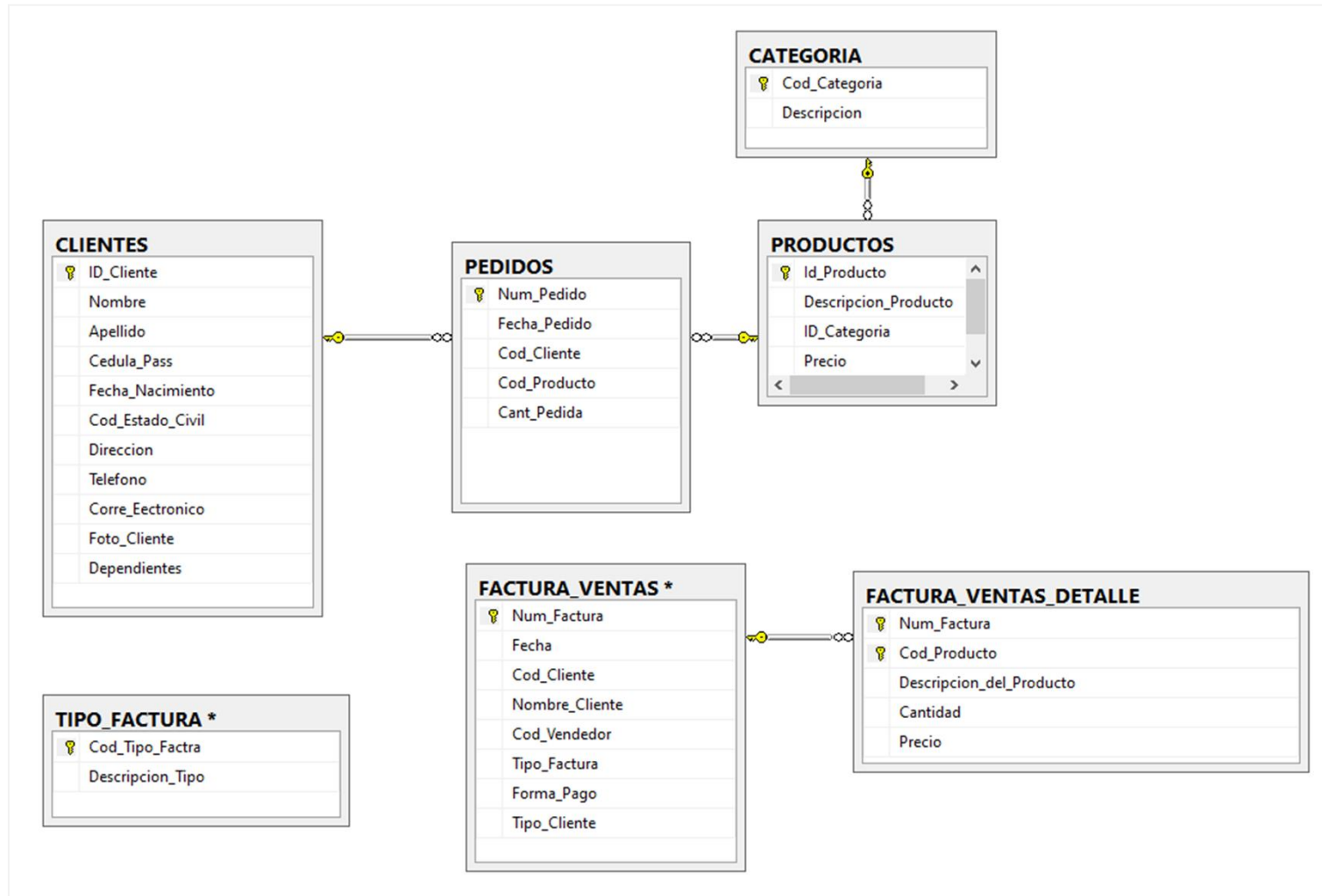
Column Name	Data Type	Allow Nulls
🔑 PersonID	int	<input type="checkbox"/>
FirstName	nvarchar(30)	<input type="checkbox"/>
LastName	nvarchar(50)	<input type="checkbox"/>
ModifiedDate	smalldatetime	<input type="checkbox"/>
▶		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
🔑 NO_FACTURA	varchar(15)	<input type="checkbox"/>
🔑 COD_FABRICA	char(6)	<input type="checkbox"/>
🔑 GRADO	varchar(20)	<input type="checkbox"/>
ESPESOR	numeric(5, 1)	<input type="checkbox"/>
PESO_NETO_KG	real	<input type="checkbox"/>
PRECIO_FOB_KG	real	<input type="checkbox"/>

4.2. Management Studio – Establecer Relaciones entre Tablas



4.2. Management Studio – Establecer Relaciones entre Tablas



4.2. Management Studio – Integridad Referencial

ELIMINAR EN CASCADA: Cuando creamos una clave foránea utilizando esta opción, elimina las filas de referencia en la tabla secundaria cuando la fila referenciada se elimina en la tabla primaria que tiene una clave primaria. Es importante indicar que esta opción no cuenta con un deshacer por lo cual su aplicación debe ser cuidadosa

ACTUALIZAR CASCADA: Cuando creamos una clave externa utilizando ACTUALIZAR CASCADA, las filas de referencia se van a actualizar en la tabla secundaria cuando la fila referenciada se actualiza en la tabla principal que tiene una clave primaria. Su utilización es adecuada y conveniente toda vez que nos ayuda a realizar actualizaciones masivas de las claves o i llaves sin tener que ir tabla a atabla

Actividad practica: Actualización y eliminación en Cascada entre las tablas CATEGORIA y PRODUCTOS

1. Creación de la regla ELIMINAR y ACTUALIZAR CASCADA en una clave externa usando el SQL Server Managment Studio.
2. Crear la regla ELIMINAR y ACTUALIZAR CASCADA en una clave foránea utilizando el script T-SQL
3. Se lanzará en una tabla con una clave externa en cascada ELIMINAR o ACTUALIZAR

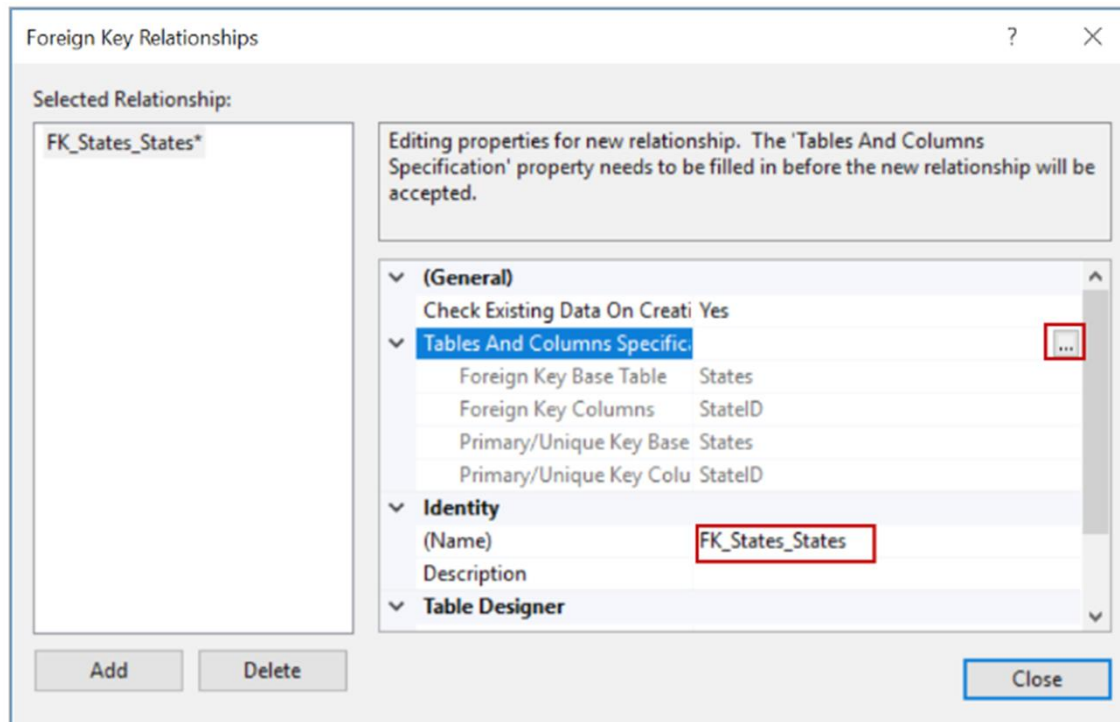
```
ALTER TABLE [dbo].[PRODUCTOS] WITH CHECK ADD CONSTRAINT [FK_ID_Categoria] FOREIGN KEY([ID_Categoria])
REFERENCES [dbo].[CATEGORIA] ([Cod_Categoria])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
```

4.2. Management Studio – Integridad Referencial

Crear una clave foránea con las reglas **ELIMINAR** y **ACTUALIZAR CASCADA**

Inicie sesión en SQL Server con SQL Server Management Studio, diríjase a la carpeta **Claves (Keys)** en la tabla secundaria. Haga clic derecho en la carpeta **Claves (Keys)** y seleccione **Nueva clave externa**.

Edite la **especificación de tablas y columnas** haciendo clic en... como se muestra en la siguiente imagen.



4.2. Management Studio – Integridad Referencial

Seleccione la tabla primaria y la columna de clave principal en la tabla primaria. Seleccione la columna de clave externa en la tabla secundaria. Haga clic en OK y consulte la imagen que se muestra a continuación.

The screenshot shows the 'Tables and Columns' dialog box in SQL Server Enterprise Manager. The dialog has a title bar with a question mark and a close button. The 'Relationship name:' field contains 'FK_States_Countries'. Below this, there are two sections: 'Primary key table:' and 'Foreign key table:'. The 'Primary key table:' dropdown is set to 'Countries', and the 'Foreign key table:' dropdown is set to 'States'. Below these, there are two columns: 'CountryID' under the primary key table and 'CountryID' under the foreign key table. The 'CountryID' column under the foreign key table is highlighted in blue. At the bottom right, there are 'OK' and 'Cancel' buttons. The 'OK' button is highlighted with a red border.

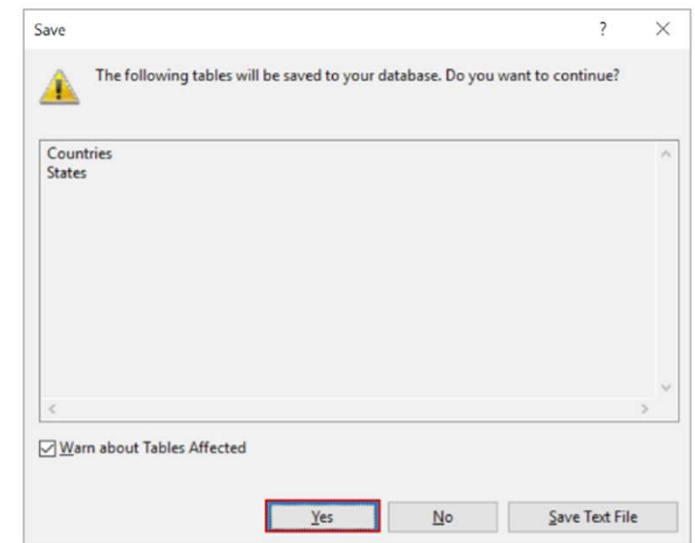
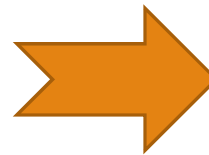
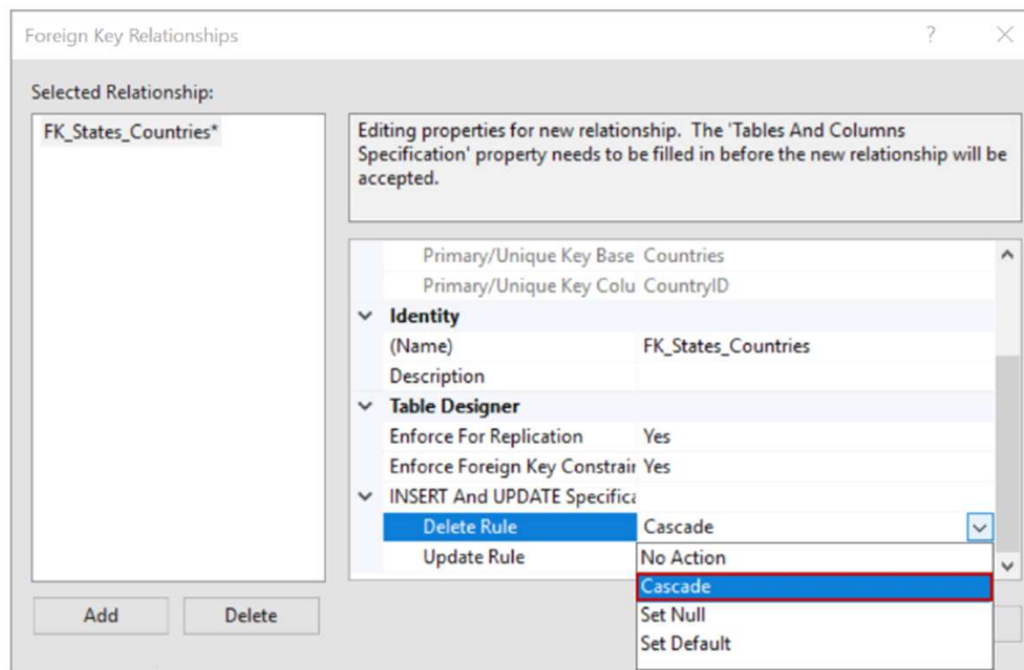
Primary key table:	Foreign key table:
Countries	States
CountryID	CountryID

OK Cancel

4.2. Management Studio – Integridad Referencial

Una vez que se haga clic en **Sí**, se va a crear una clave externa con la regla de eliminación. Podemos crear una clave foránea con la regla ACTUALIZAR CASCADA, seleccionando CASCADA como una acción para la regla de actualización en las especificaciones INSERTAR y ACTUALIZAR.

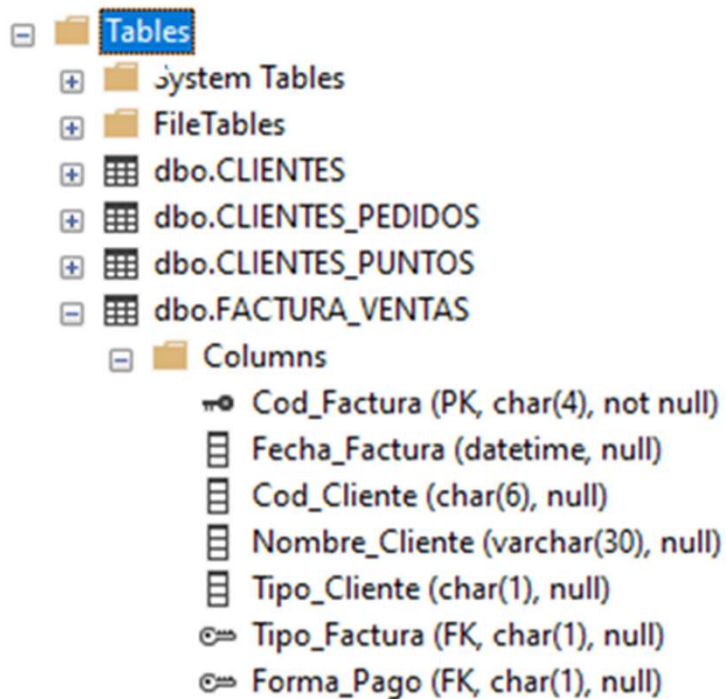
Finalmente haga clic en **Cerrar** y guarde la tabla en el diseñador. Haga clic en **Sí** en la ventana del mensaje de advertencia.



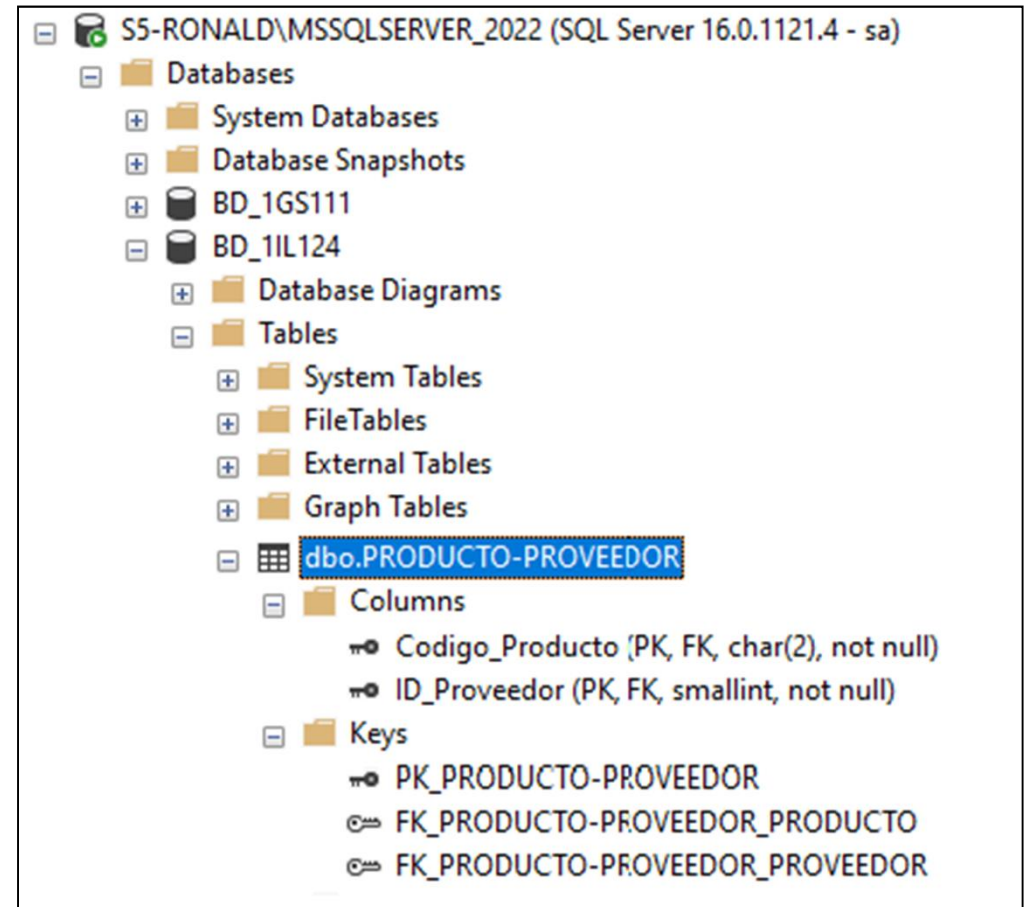
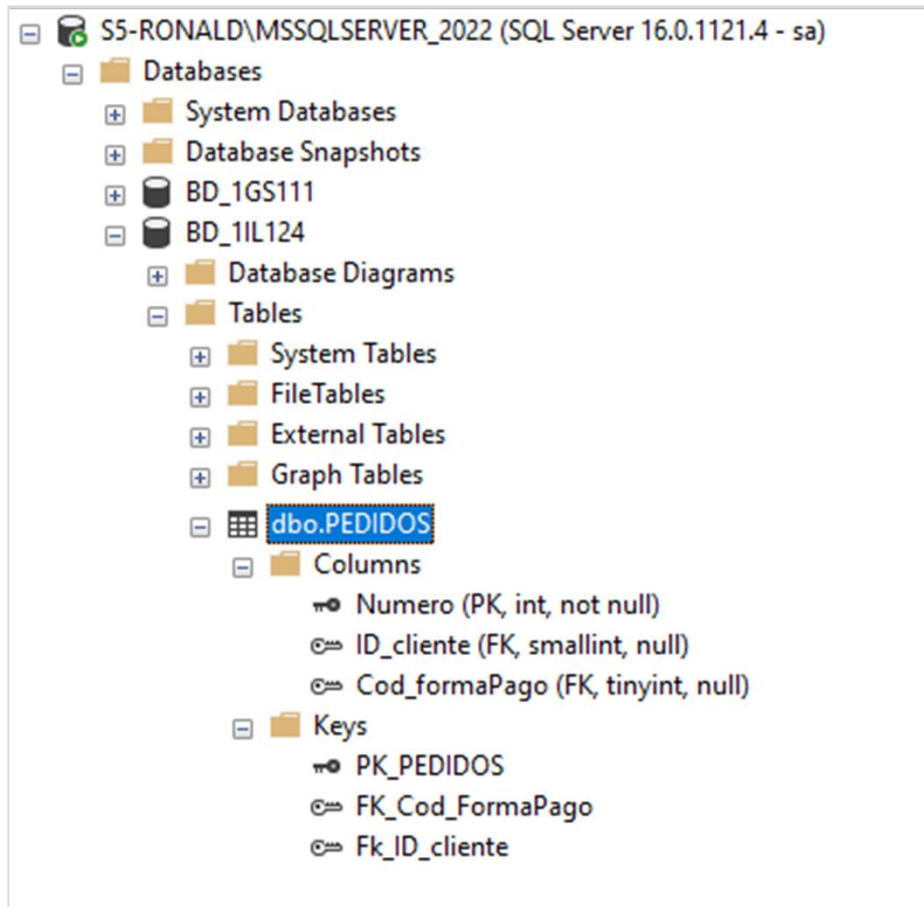
4.2. Management Studio – Integridad Referencial

Desde el Diagrama de Base de Datos no es muy clara que tipo de Relación se está modelando por lo cual para poder observar mejor esto, les recomendamos abrir la tabla de principal de FACTURA_VENTAS, en forma de observación de sus filas y columnas.

Expanda columnas y allí encontraran la identificación de las llaves, en modelo de las llaves primarias (llave más oscura que apunta a la izquierda) y las llaves foráneas de “**Constraint**” para exigir integridad referencial (llave más clara que apunta de forma invertida a la derecha)



4.2. Management Studio – Campo Consrraaint de Base de Datos



4.2. Management Studio – Integridad Referencial

Construya el siguiente script T-SQL, la cual crea una Tabla principal, una Tabla secundaria y una clave externa en la tabla secundaria con integridad y exigencia de la regla ELIMINAR y ACTUALIZAR en Cascada.

```
CREATE TABLE Countries
(
  CountryID INT PRIMARY KEY,
  CountryName VARCHAR(50),
  CountryCode VARCHAR(3)
)

CREATE TABLE States
(
  StateID INT PRIMARY KEY,
  StateName VARCHAR(50),
  StateCode VARCHAR(3),
  CountryID INT
)

ALTER TABLE [dbo].[States] WITH CHECK ADD CONSTRAINT [FK_States_Countries] FOREIGN KEY([CountryID])
REFERENCES [dbo].[Countries] ([CountryID])
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[States] CHECK CONSTRAINT [FK_States_Countries]
GO
```

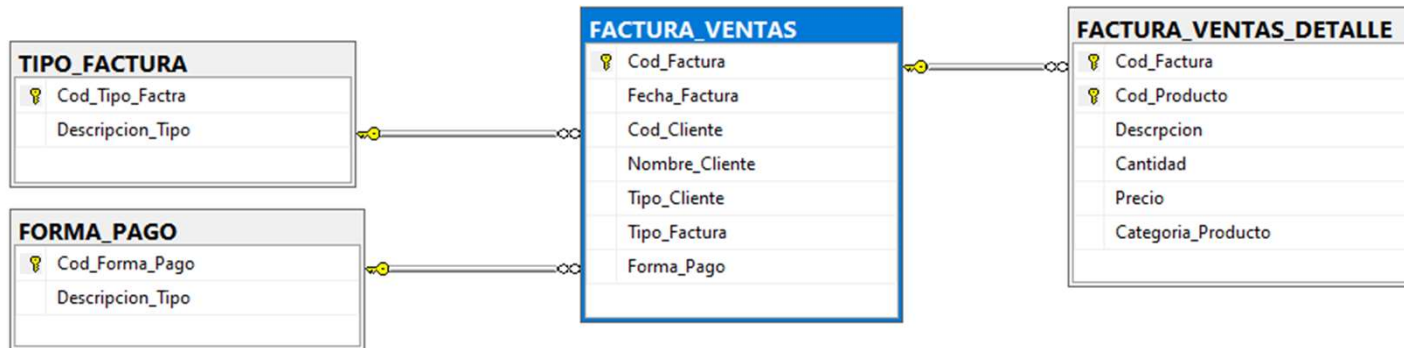
4.2. Management Studio – Integridad Referencial

```
USE [BD_IGS116]
GO
```

```
ALTER TABLE [dbo].[FACTURA_VENTAS]
    ADD CONSTRAINT FK_Tipo_Factura FOREIGN KEY (Tipo_Factura) REFERENCES TIPO_FACTURA (Cod_Tipo_Factra)
```

```
USE [BD_IGS116]
GO
```

```
ALTER TABLE [dbo].[FACTURA_VENTAS]
    ADD CONSTRAINT FK_Forma_Pago FOREIGN KEY (Forma_Pago) REFERENCES FORMA_PAGO (Cod_Forma_Pago)
```



Como se puede observar en el diagrama superior existen tres tablas relacionadas a la tabla principal de **FACTURA_VENTAS**, sin embargo, las relaciones de **TIPO_FACTURA** y **FORMA_PAGO**, son diferentes a la relación que existe con **FACTURA_VENTAS_DETALLES**, ya que en esta última existe una llave foránea para permitir la relación MAESTRO . DETALLE entre estas dos tablas y las dos antes mencionadas lo que están exigiendo es Integridad Referencial para la Tabla Principal.

4.3. Transact SQL – Creación de Base de Datos

```
--Base de Datos I  
--Autoria:RIP
```

```
-- Scrip de Creación de una Base de Datos
```

```
USE master;  
GO
```

```
CREATE DATABASE BD_NUEVAJAVA ON  
(NAME = BD_NUEVAJAVA_dat,  
  FILENAME =  
'C:\SISTEMA5\BASEDATOS\DAT\BD_NUEVAJAVAdat.mdf',  
  SIZE = 20,  -- Tamaño de BD_dat inicial  
  MAXSIZE = 750,  
  FILEGROWTH = 5)          --Aumento de el tamaño
```

```
BD
```

```
LOG ON
```

```
(NAME = BD_NUEVAJAVA_log,  
  FILENAME =  
'C:\SISTEMA5\BASEDATOS\LOG\BD_NUEVAJAVAllog.ldf',  
  SIZE = 5 MB, -- Tamaño de BD_log inicial  
  MAXSIZE = 25 MB,  
  FILEGROWTH = 5 MB)  --Aumento de el tamaño BD
```

```
COLLATE Modern_Spanish_CI_AI;
```

```
GO
```

4.3. Transact SQL – Constraint de Foreign Key

```
USE [BD_1GS111]
GO
ALTER TABLE [dbo].[FACTURA_VENTAS]
    ADD CONSTRAINT FK_Cod_Cliente_ FOREIGN KEY (Cod_Cliente)
    REFERENCES CLIENTES (ID_Cliente)
GO
```

```
USE BD_REPUESTOS
```

```
ALTER TABLE [dbo].[ORDEN_TRABAJO] WITH NOCHECK ADD
CONSTRAINT [FK_PROVEEDOR] FOREIGN KEY ([COD_PROVEEDOR],
[COD_MARCA])
REFERENCES [dbo].[MARCA] ([COD_PROVEEDOR], [CODIGO])
ON UPDATE CASCADE
GO
```

4.3. Transact SQL – Constraint de Foreign Key

Esta Tabla debe ser creada primero:

```
CREATE TABLE Tipos_Cliente (  
    Tipo_ClienteID INT CONSTRAINT PK_TiposCliente PRIMARY KEY, -- Clave primaria personalizada  
    Descripcion VARCHAR(50) NOT NULL  
);
```

Esta tabla se crea después de Tipos_Clientes

```
CREATE TABLE Clientes (  
    ClienteID INT CONSTRAINT PK_Clientes PRIMARY KEY, -- Clave primaria de restricción personalizado  
    Nombre VARCHAR(50) NOT NULL,  
    Apellido VARCHAR(50) NOT NULL,  
    Email VARCHAR(100),  
    Telefono VARCHAR(15),  
    FechaRegistro DATETIME DEFAULT GETDATE(),  
    Tipo_ClienteID INT,  
    CONSTRAINT FK_Clientes_TipoCliente FOREIGN KEY (Tipo_ClienteID)  
        REFERENCES Tipos_Cliente(Tipo_ClienteID) -- Clave foránea personalizada  
);
```

4.3. Transact SQL – Usos Orden de Inserción de Datos

Insertar en la Tabla Tipos_Cliente: Primero, insertamos los tipos de clientes en la tabla Tipos_Cliente. Esto es necesario para asegurar que las claves foráneas (Tipo_ClienteID) en la tabla Clientes tengan referencias válidas.

```
INSERT INTO Tipos_Cliente (Tipo_ClienteID, Descripcion) VALUES (1, 'Común');
INSERT INTO Tipos_Cliente (Tipo_ClienteID, Descripcion) VALUES (2, 'Empresa');
INSERT INTO Tipos_Cliente (Tipo_ClienteID, Descripcion) VALUES (3, 'Frecuente');
INSERT INTO Tipos_Cliente (Tipo_ClienteID, Descripcion) VALUES (4, 'Vip');
```

Insertar en la Tabla Clientes: Después, insertamos los registros de clientes en la tabla Clientes, especificando el tipo de cliente correspondiente para cada registro, basándonos en los valores previamente insertados en Tipos_Cliente.

```
INSERT INTO
Clientes (ClienteID, Nombre, Apellido, Email, Telefono, Tipo_ClienteID) VALUES (1, 'Juan', 'Pérez', 'juan.perez@example.com', '5551234567', 1);
-- Cliente Común
INSERT INTO Clientes (ClienteID, Nombre, Apellido, Email, Telefono, Tipo_ClienteID) VALUES (2, 'María', 'Gómez', 'maria.gomez@panama.com', '5557654321', 2); --
Cliente Empresa
INSERT INTO Clientes (ClienteID, Nombre, Apellido, Email, Telefono, Tipo_ClienteID) VALUES (3, 'Carlos', 'López', 'carlos.lopez@chiriqui.com', '5559876543', 3); --
Cliente Frecuente
INSERT INTO Clientes (ClienteID, Nombre, Apellido, Email, Telefono, Tipo_ClienteID) VALUES (4, 'Ana', 'Martínez', 'ana.martinez@herrera.com', '5556547890', 4); --
Cliente Vip
INSERT INTO Clientes (ClienteID, Nombre, Apellido, Email, Telefono, Tipo_ClienteID) VALUES (5, 'Pedro', 'Fernández', 'pedro.fernandez@colon.com', '5553210987', 1); --
Cliente Común
```