	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE ESTUDIOS TECNOLOGICOS</b> <b>ESCUELA DE COMPUTACION</b>
<b>CICLO 01-2016</b>	<b>GUIA DE LABORATORIO Nº 3</b>
	<b>Nombre de la practica:</b> Diseño y creación de una base de datos relacional en SQL Server Management Studio <b>Lugar de ejecución:</b> Laboratorio de Informática <b>Tiempo estimado:</b> 3 horas <b>Materia:</b> Base de datos <b>Docentes:</b> Blanca Iris Cañas y Evelyn Hernández

## I. Objetivos

1. Crear una base de datos
2. Crear tablas implementando en los campos diferentes tipos de datos
3. Identificar las llaves primarias y foráneas en una tabla

## II. Introducción Teórica

### Modelo relacional

Este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Su idea fundamental es el uso de "**relaciones**". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario eventual de la base de datos.

La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

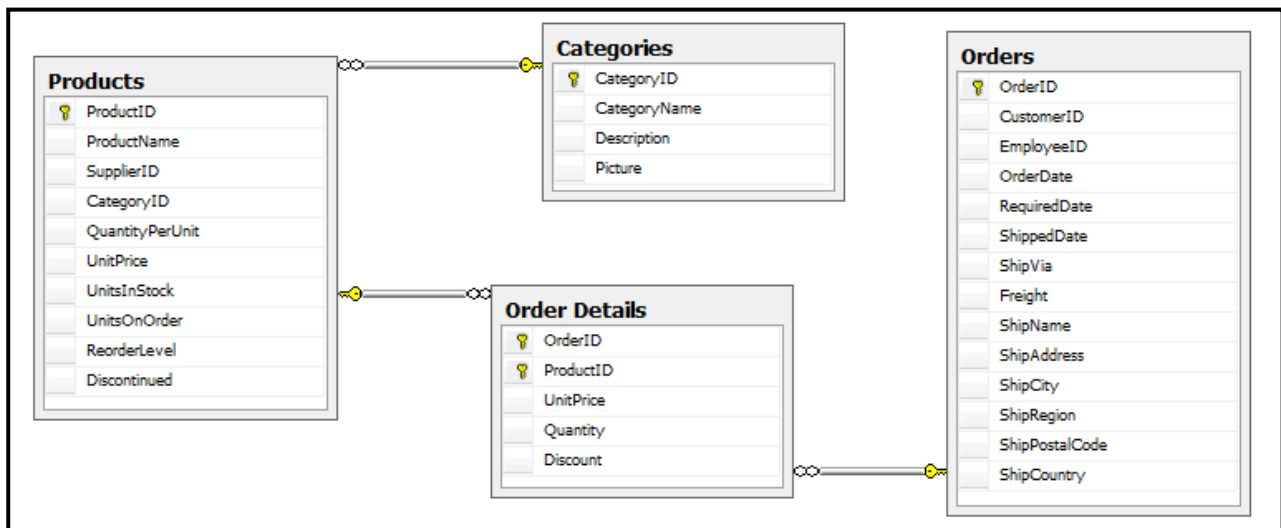
### Bases de datos relacionales

Una base de datos relacional almacena los datos en varios lugares denominados tablas y además controla la manera en que esas tablas se relacionan entre sí. En ocasiones encontrará las siglas RDBMS, del inglés Relational Database Management System, para hacer referencia a una base de datos relacional.

Por ejemplo: una base de datos que sirve para llevar un control de las ventas de una empresa, podría contener información sobre productos, categorías, pedidos y detalle de pedidos, los datos sobre cada uno de estos temas se almacenarían en tablas individuales, que tendrían los siguientes nombres:

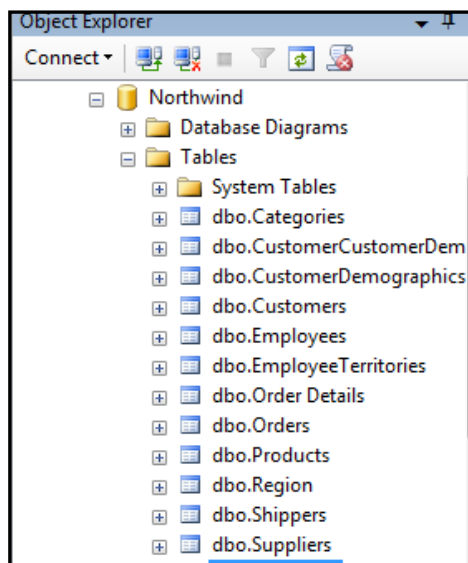
- Products
- Categories
- Orders
- Order Details

Además el RDBMS supervisaría los hechos que relacionan estas tablas entre sí, por ejemplo cada categoría tendría diferentes productos y cada producto estaría en diferentes órdenes de pedido



## Tablas

Las tablas son los objetos que almacenan los datos, una de las normas básicas para la base de datos es que cada tabla debería almacenar información sobre una entidad en concreto esto se conoce como regla de normalización.



Gran parte del trabajo que se realiza con una base de datos gira en torno a las tablas, cada base de datos admite estas cuatro operaciones básicas:

1. Añadir información a una tabla
2. Actualizar la información que ya existe en una tabla
3. Eliminar información que ya existe en una tabla
4. Ver la información contenida en una tabla

Por lo general estas operaciones se realizan ejecutando instrucciones SQL.

## Registros, campos y valores

Cada tabla se compone de registros y campos, registro es toda la información sobre una de las entidades dentro de una tabla, campo es un dato en concreto almacenado en una tabla.

	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	0x151C2F00020000000D000E0014002100FFFFFFFF
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F00020000000D000E0014002100FFFFFFFF
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F00020000000D000E0014002100FFFFFFFF
4	4	Dairy Products	Cheeses	0x151C2F00020000000D000E0014002100FFFFFFFF
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F00020000000D000E0014002100FFFFFFFF
6	6	Meat/Poultry	Prepared meats	0x151C2F00020000000D000E0014002100FFFFFFFF
7	7	Produce	Dried fruit and bean curd	0x151C2F00020000000D000E0014002100FFFFFFFF
8	8	Seafood	Seaweed and fish	0x151C2F00020000000D000E0014002100FFFFFFFF

En la tabla anterior se puede identificar como:

<b>Campo</b>	CategoryID
<b>Registro</b>	1, Bevarages, Soft drinks, coffees, teas etc.

En función de lo que se desea hacer, a veces conviene manipular los registros y otras veces los campos, por ejemplo, si quisiera saber lo que hay almacenado en la base datos sobre una categoría en concreto, recuperaría el registro de esa categoría de la tabla correspondiente, en cambio, si deseara saber los nombres de todas las categorías, tendría que revisar el contenido del campo CategoryName en todos los registros de la misma tabla.

Normalmente en una base de datos a los registros se le conoce como fila y a los campos como columnas.

### Propiedades de campo

Una base de datos completa como SQL Server le permite crear las diferencias entre cada campo especificando propiedades de campo:

	Column Name	Data Type	Allow Nulls
► ?	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(15)	<input type="checkbox"/>
	Description	ntext	<input checked="" type="checkbox"/>
	Picture	image	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Esta vista muestra la información del esquema de la tabla Categories en lugar de los datos, el esquema de una base de datos es una manera de hacer referencia a toda la información de diseño que determina lo que se puede almacenar en esa base de datos.

Se muestran las tres propiedades más importantes de cada campo de la tabla:

- El nombre de la columna (Column Name)
- El tipo de datos (Data Type)

- Si se permiten los valores nulos (Allow Nulls)

Por medio del **nombre de la columna** de un campo (o columna) se puede hacer referencia a ese campo en la tabla, por lo general se asignaran nombres significativos a los campos.

El **tipo de dato** de un campo determina los datos que se pueden almacenar en él, por ejemplo:

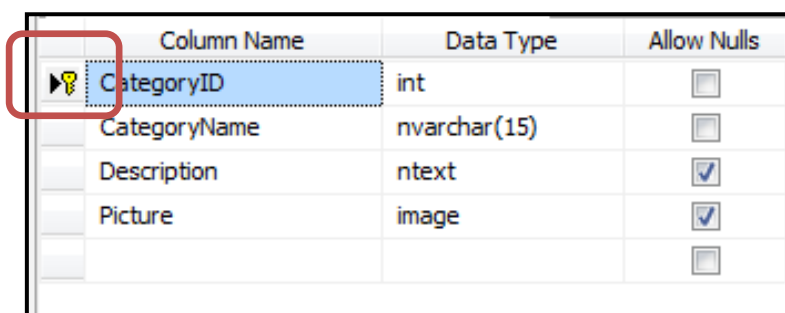
<b>varchar(30)</b>	<b>Texto con un tamaño máximo de 30 caracteres</b>
<b>datetime</b>	Información de fecha y hora
<b>int</b>	Números enteros

La propiedad de longitud de un campo especifica la cantidad máxima de datos que se puede almacenar en él.

La propiedad **permitir valores nulos** muestra si se permiten valores nulos en ese campo, si un campo no permite valores nulos, debe ingresarse un valor no nulo para ese campo en cada registro antes de poder guardar el registro.

Empleando las propiedades de campo para distinguir un campo de otro, contribuirá a mantener el orden y la organización de la base de datos, sirve para establecer reglas que la base de datos impone automáticamente para que los datos que se almacenen sigan cierta lógica.

### Claves y restricciones



	Column Name	Data Type	Allow Nulls
▶	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(15)	<input type="checkbox"/>
	Description	ntext	<input checked="" type="checkbox"/>
	Picture	image	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

En la imagen anterior se observará que al lado izquierdo del campo CategoryID posee un símbolo de llave, esta indica que esa columna es la clave principal de la tabla.

Una **clave principal** es una información de identificación exclusiva que le permite buscar un registro determinado dentro de una tabla, en la misma tabla no puede haber dos registros con el mismo valor en el campo de la clave principal, la clave principal podría estar compuesta por un solo campo o por varios

### PRIMARY KEY

Al crear una tabla puede crear una sola restricción PRIMARY KEY como parte de la definición de tabla. Si la tabla ya existe, puede agregar una restricción PRIMARY KEY, siempre que no exista ya otra restricción PRIMARY KEY. Una tabla puede contener una sola restricción PRIMARY KEY.

Cuando se agrega una restricción PRIMARY KEY a una o varias columnas de la tabla, el Motor de base de datos examina los datos y metadatos de columnas existentes para garantizar que se cumplen las siguientes reglas en las claves principales:

- **Las columnas no admiten valores NULL.**  
Las columnas de restricciones PRIMARY KEY que se especifiquen al crear la tabla se convierten implícitamente a NOT NULL. Una columna dispersa no se puede utilizar como parte de una clave principal porque las columnas dispersas deben permitir valores NULL.

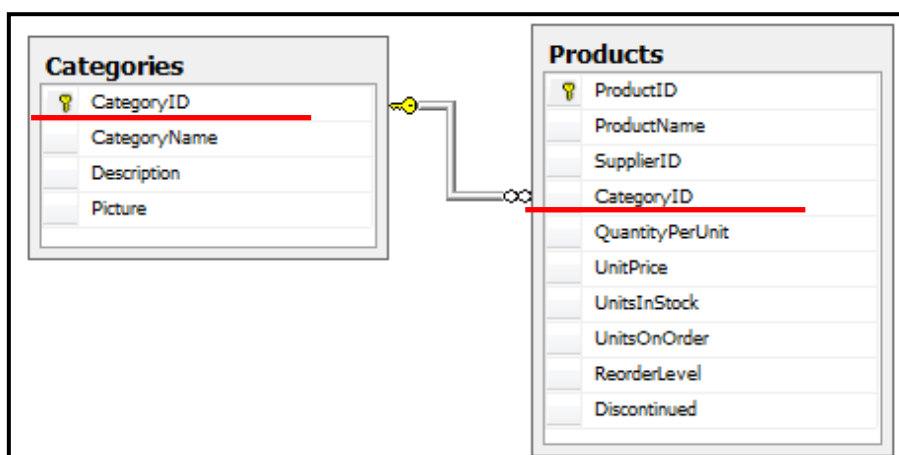
- **No puede haber valores duplicados.**

Si se agrega una restricción PRIMARY KEY a una columna que tiene valores duplicados o permite valores NULL, el Motor de base de datos devuelve un error y no agrega la restricción.

### Claves externas o foráneas

Una clave externa es un campo (o campos) que señala la clave principal de otra tabla. El propósito de la clave externa es **asegurar la integridad referencial de los datos**. En otras palabras, sólo se permiten los valores que se esperan que aparezcan en la base de datos.

La integridad referencial asegura que se mantengan las referencias entre las claves principales y las externas. Por ejemplo:



Se controla que si se agrega un código de categoría (CategoryID) en la tabla "Products", tal código exista previamente en la tabla "Categories".

**Nota: Los campos CategoryID de las tablas Categories y Productos deben tener las mismas propiedades de campo**

### FOREIGN KEY

Cuando se crea una tabla, se puede crear una restricción FOREIGN KEY como parte de la definición de esa tabla. Si ya existe una tabla, puede agregar una restricción FOREIGN KEY, siempre que dicha restricción esté vinculada a una restricción PRIMARY KEY, una tabla puede contener varias restricciones FOREIGN KEY.

Elimine una restricción FOREIGN KEY para quitar el requisito de integridad referencial entre las columnas de clave externa y las columnas de la clave principal.

### III. Requerimientos

- Máquina con SQL Server 2012
- Guía Número 3 de base de datos

## IV. Procedimiento

### Parte 1: Iniciando sesión desde SQL Server Managment Studio

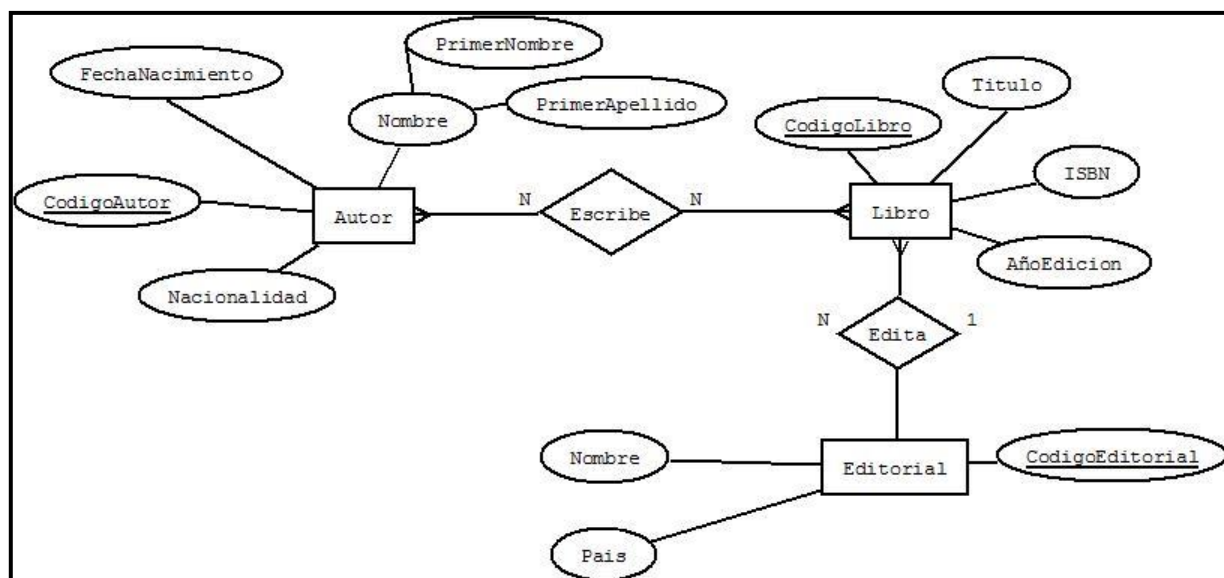
1. Hacer clic en el botón **Inicio**
2. Hacer clic en la opción **Todos los programas** y hacer clic en **Microsoft SQL Server 2012**

Para conectarse con el servidor de base de datos elija los siguientes parámetros de autenticación:

- **Tipo de servidor:** Database Engine
- **Nombre del servidor:** S02-PCNumMaquina  
Nota: NumMaquina es el número de la maquina local
- **Autenticación:** SQL Server Authentication
- **Login:** sa
- **Password:** 123456

### Parte 2. Diseño de una base de datos relacional

Para esta parte de la guía se tomará el siguiente modelo E-R:



Interpretación del modelo E-R:

- Un Autor puede escribir muchos libros y un libro puede ser escrito por muchos autores
- Una Editorial puede editar muchos libros pero un libro solo puede ser editado por una editorial específica

### Ejercicio 1. Crear la base de datos

1. Crear la base de datos **Gestion\_Libros\_SuCarnet**
2. Debe tener las siguientes propiedades:
  - a. Tamaño inicial (Initial size): 4MB
  - b. Auto incremento (Autogrowth): 2MB

- c. Propietario (Owner): sa

## Ejercicio 2. Crear las tablas de la base de datos

El diagrama E-R nos muestra tres entidades:

Entidades	Atributos
<b>Autor</b>	CodigoAutor Nombre (PrimerNombre y Primer Apellido) FechaNacimiento Nacionalidad
<b>Libro</b>	CodigoLibro Titulo ISBN AñoEdicion
<b>Editorial</b>	CodigoEditorial Nombre País

La relación que existe entre la entidad Editorial y Libro es de uno a muchos

La relación que existe entre Autor y Libro es de muchos a muchos, este tipo de relaciones no son recomendables y debemos tratar de evitarlas utilizando **TABLAS INTERMEDIAS** en las que se utilizarían relaciones de uno a muchos.

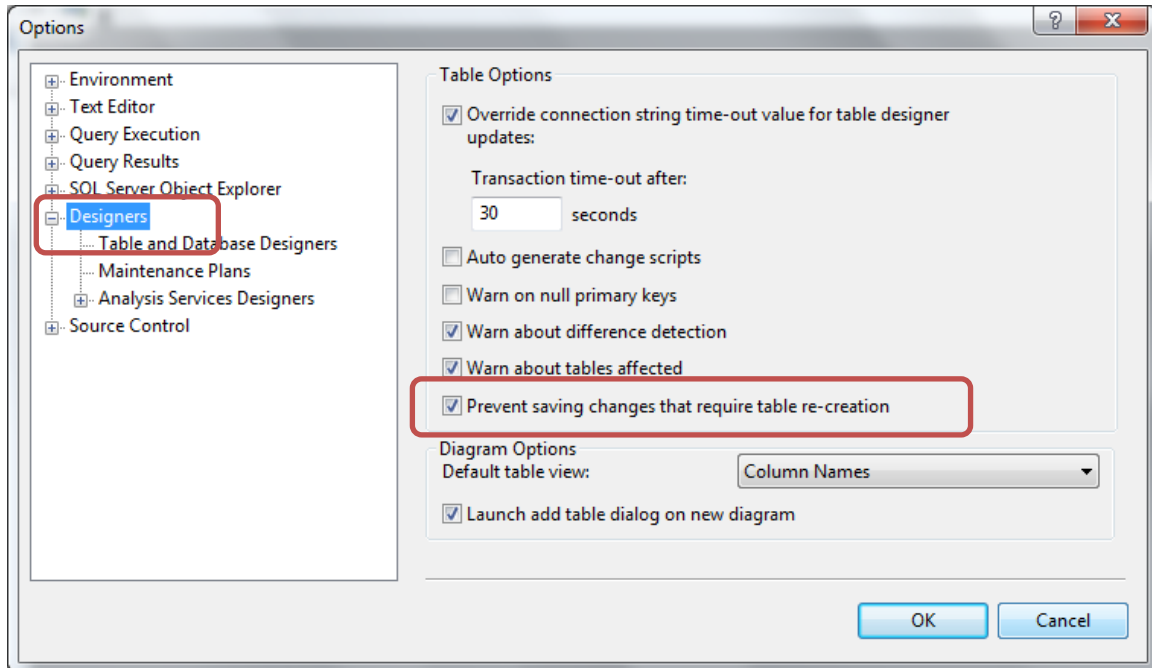
Por lo tanto las tablas a crear en este ejercicio son:

Tablas	Campos
<b>Autor</b>	<u>CodigoAutor</u> (Llave primaria) Nombre (PrimerNombre y Primer Apellido) FechaNacimiento Nacionalidad
<b>Libro</b>	<u>CodigoLibro</u> (Llave primaria) Titulo ISBN AñoEdicion CodigoEditorial (Llave foránea)
<b>Editorial</b>	<u>CodigoEditorial</u> (Llave primaria) Nombre País
<b>Detalle_AutorLibro</b>	CodigoAutor (Llave foránea) CodigoLibro (Llave foránea) Fecha

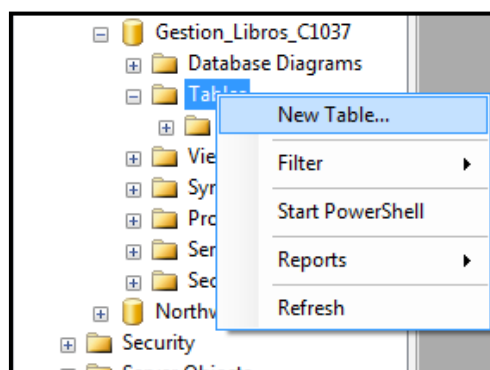
### Nota:

- A la tabla Libro se agregó el campo CodigoEditorial por la relación que existe entre esta tabla y la tabla Editorial, por lo tanto la llave foránea de la tabla Libro es la llave primaria de la tabla Editorial
- Se creó la tabla Detalle\_AutorLibro ya que esta tabla intermedia rompe la relación de muchos a muchos que existe entre Autor y Libro, se le agregaron los campos CodigoAutor y CodigoLibro para crear la relación de uno a muchos, de igual forma estos campos son las llaves foráneas de esta tabla ya que son las llaves primarias de las tablas Autor y Libro respectivamente.

1. Antes de hacer este ejercicio realice los siguientes pasos:
  - a. Hacer clic en la opción **Tools**
  - b. Hacer clic en **Options...**
  - c. En la ventana Options, hacer clic en **Designers** y quitar el cheque a la opción **Prevent saving changes that require table re-creation**



2. Quitando este cheque podrá modificar el diseño de las tablas sin ningún problema
3. Haga clic en el botón OK
4. En el Explorador de objetos, haga clic derecho en el nodo Tablas (Tables) de la base de datos que ya creo en la práctica y, a continuación, haga clic en Nueva tabla (New Table...)



5. Escriba los nombres de columna, elija los tipos de datos y decida en cada columna si va a permitir valores NULL.  
Crear la siguiente tabla:



Column Name	Data Type	Allow Nulls
CodigoAutor	char(5)	<input type="checkbox"/>
PrimerNombre	varchar(25)	<input checked="" type="checkbox"/>
SegundoNombre	varchar(30)	<input checked="" type="checkbox"/>
FechaNacimiento	date	<input checked="" type="checkbox"/>
Nacionalidad	varchar(35)	<input checked="" type="checkbox"/>

- En el menú Archivo (File), seleccione Guardar (Save Table\_1)
- En el cuadro de diálogo Elegir nombre, en nuestro ejercicio la tabla es la de Autor y haga clic en Aceptar (OK).
- Para terminar el ejercicio cree las demás tablas:

**Tabla: Libro**

Column Name	Data Type	Allow Nulls
CodigoLibro	char(10)	<input type="checkbox"/>
Titulo	varchar(MAX)	<input checked="" type="checkbox"/>
ISBN	varchar(20)	<input type="checkbox"/>
AñoEdicion	char(4)	<input checked="" type="checkbox"/>
CodigoEditorial	char(5)	<input type="checkbox"/>

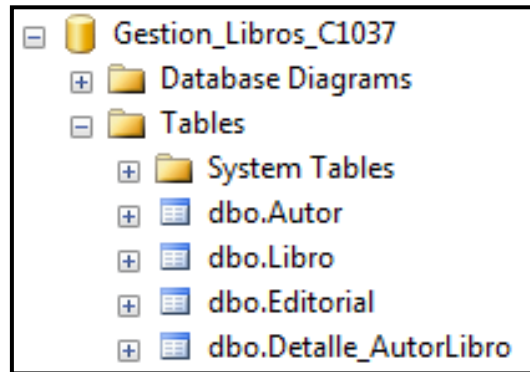
**Tabla: Editorial**

Column Name	Data Type	Allow Nulls
CodigoEditorial	char(5)	<input type="checkbox"/>
Nombre	varchar(45)	<input checked="" type="checkbox"/>
Pais	varchar(50)	<input checked="" type="checkbox"/>

**Tabla: Detalle\_AutorLibro**

Column Name	Data Type	Allow Nulls
CodigoAutor	char(5)	<input checked="" type="checkbox"/>
CodigoLibro	char(10)	<input checked="" type="checkbox"/>
Fecha	date	<input checked="" type="checkbox"/>

- Al final tendrá cuatro tablas creadas en la base de datos **Gestion\_Libros\_SuCarnet**:

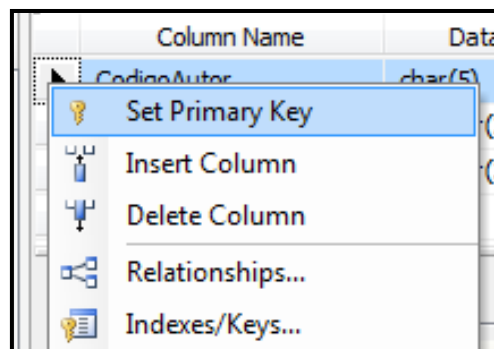


### Ejercicio 3. Establecer claves primarias (PRIMARY KEY) y claves externas o foráneas (FOREIGN KEY)


Ahora se creará las relaciones entre las tablas tomando siempre el modelo E-R

#### Estableciendo las claves principales o primarias:

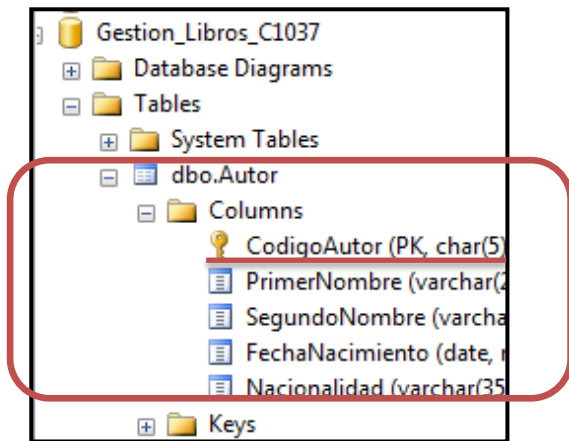
1. Cierre todas las tablas que tiene abierta
2. Hacer clic derecho sobre la tabla Autor y seleccione la opción Diseño (Design) y se abrirá de nuevo el diseño de la tabla
3. Hacer clic derecho sobre el campo CodigoAutor y seleccione Establecer clave principal (Set Primary Key)



4. Al final observará un símbolo como de llave a la par del campo: CodigoAutor

	Column Name	Data Type	Allow Nulls
	CodigoAutor	char(5)	<input type="checkbox"/>
	PrimerNombre	varchar(25)	<input checked="" type="checkbox"/>
	SegundoNombre	varchar(30)	<input checked="" type="checkbox"/>
	FechaNacimiento	date	<input checked="" type="checkbox"/>

5. Guarde los cambios en la tabla Autor y cerrar la tabla
6. Expanda la carpeta dbo.Autor y luego la carpeta Columns, observará los campos de la tabla y la asignación de la clave principal en el campo CodigoAutor



7. Asigne la clave principal a las siguientes tablas:

Tabla	Campo
<b>Libro</b>	CodigoLibro
<b>Editorial</b>	CodigoEditorial

### Estableciendo las claves externas o foráneas

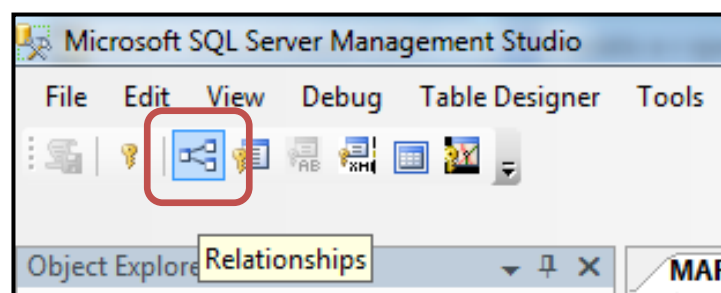
Siempre tomando en cuenta el modelo E-R, se observa que existe una relación de Muchos a Muchos entre las entidades Autor y Libro, pero como ya se definió la tabla intermedia Detalle\_AutorLibro ahora la relación queda de la siguiente manera:



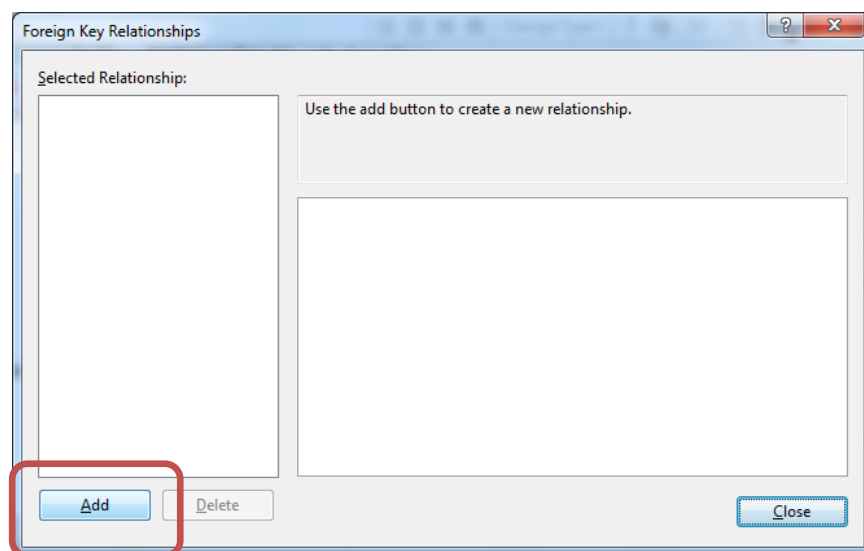
Y se lee de la siguiente manera:

Un autor puede escribir muchos libros y un libro puede ser escrito por muchos autores, por lo tanto en la tabla Detalle\_AutorLibro se almacena por separado el código del libro las veces que se quiera como también el código del autor las veces que sea necesario.

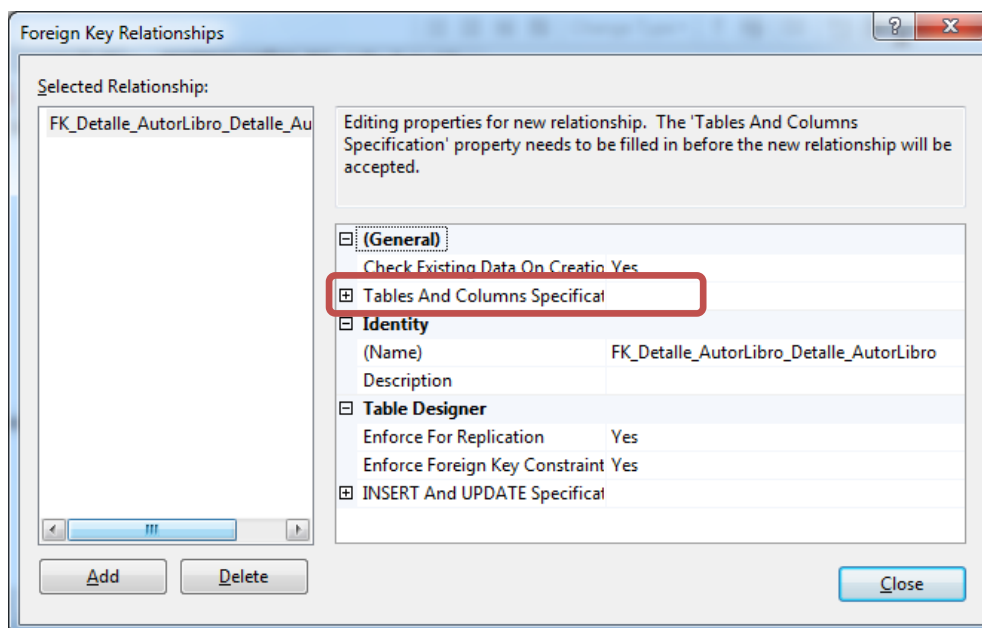
1. Haga clic derecho en la tabla que va a estar en el lado de la clave externa de la relación en este caso la tabla **Detalle\_AutorLibro** y, a continuación, haga clic en Diseño (Design)
2. La tabla se abre en el Diseñador de tablas (Table Designer).
3. En el menú Diseñador de tablas, haga clic en Relaciones (Relationships).



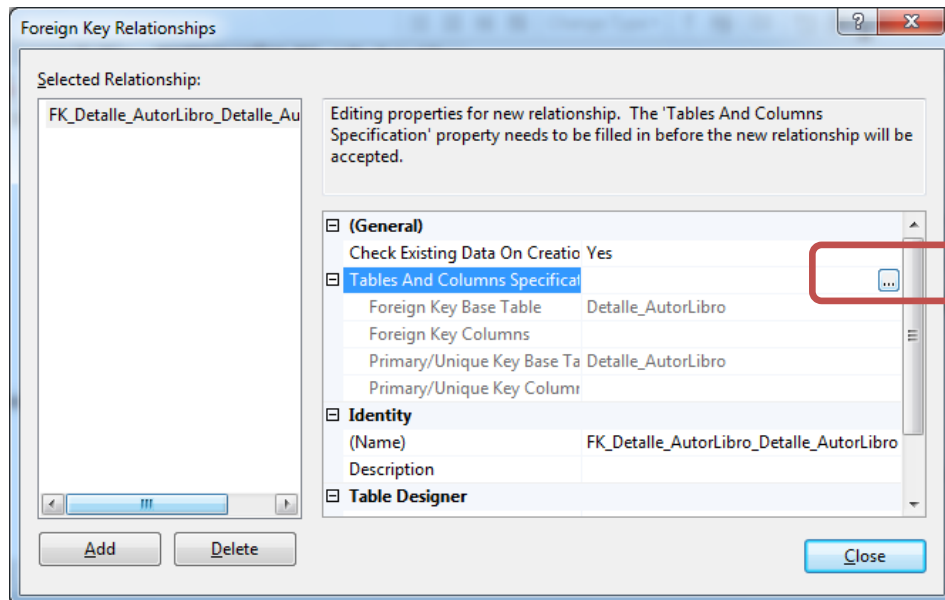
4. En el cuadro de diálogo Relaciones de clave externa (Foreign Key Relationship), haga clic en Agregar (Add)



5. La relación aparece en la lista Relación seleccionada con un nombre proporcionado por el sistema con el formato FK\_<tablename>\_<tablename>, donde tablename es el nombre de la tabla de clave externa.



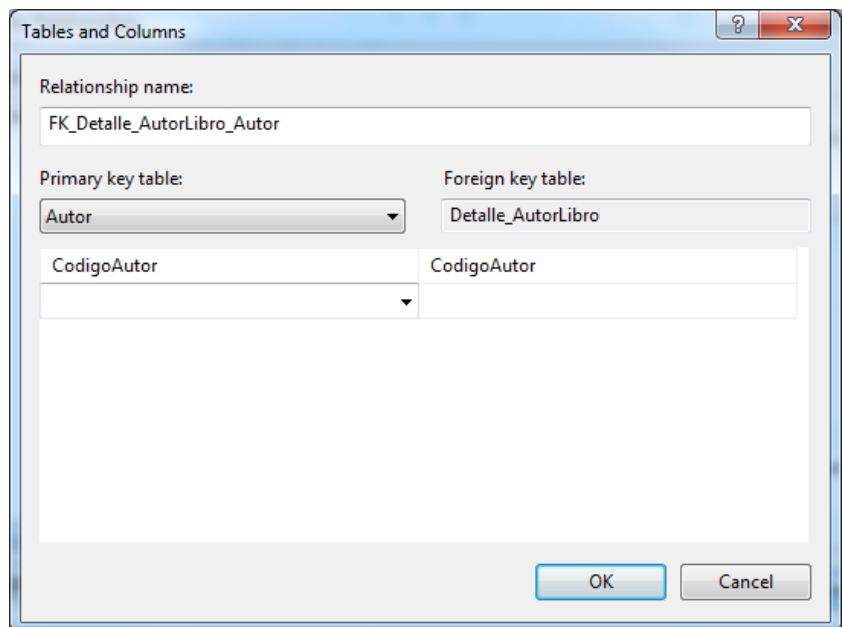
6. Haga clic en el símbolo + de la opción **Especificaciones de tablas y columnas (Tables and Columns Specification)** y a continuación, haga clic en los puntos suspensivos (...) que aparecen a la derecha de la propiedad.



7. En el cuadro de diálogo Tablas y columnas (Tables and Columns), en la lista desplegable **Clave principal (Primary key table)**, elija la tabla que estará en el lado de la clave principal de la relación en este caso la Tabla **Autor**.

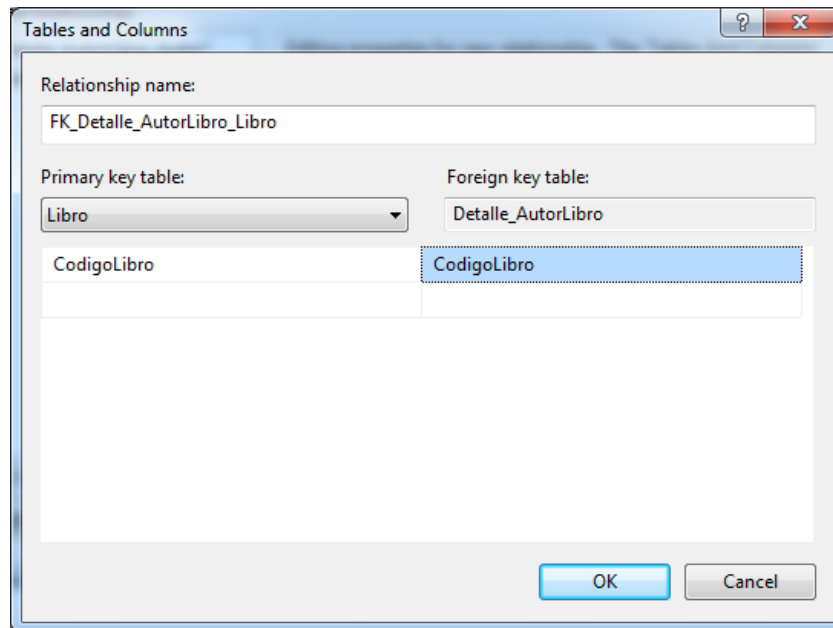
8. En la cuadrícula situada debajo, elija la columna que contribuyen a la clave principal de la tabla en este caso **CodigoAutor**

9. En la celda de la cuadrícula adyacente situada a la izquierda de cada columna, elija la columna de **Clave externa (Foreign key table)** en el ejercicio debe seleccionar el campo **CodigoAutor** correspondiente de la tabla de clave externa **Detalle\_AutorLibro**

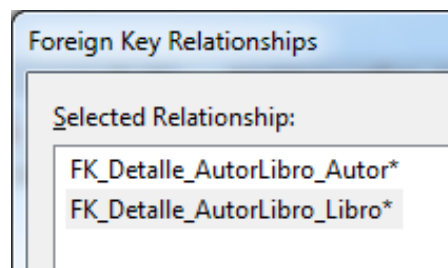


10. Haga clic en Aceptar (Ok) para crear la relación.

11. Ahora hacer clic en Agregar (Add), para crear la relación entre las tablas **Libro y Detalle\_AutorLibro**



12. Al final quedará de la siguiente manera:

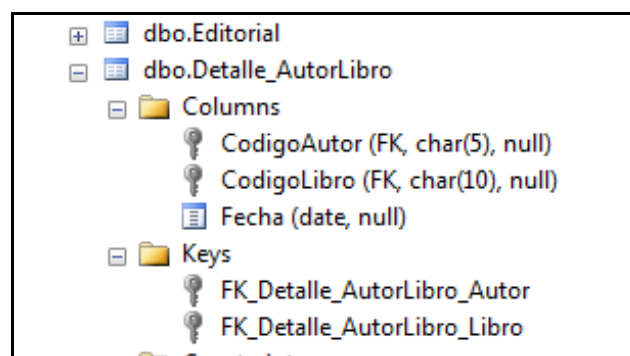


13. Hacer clic en la opción Cerrar (Close)

14. Cierre el diseñador de tablas

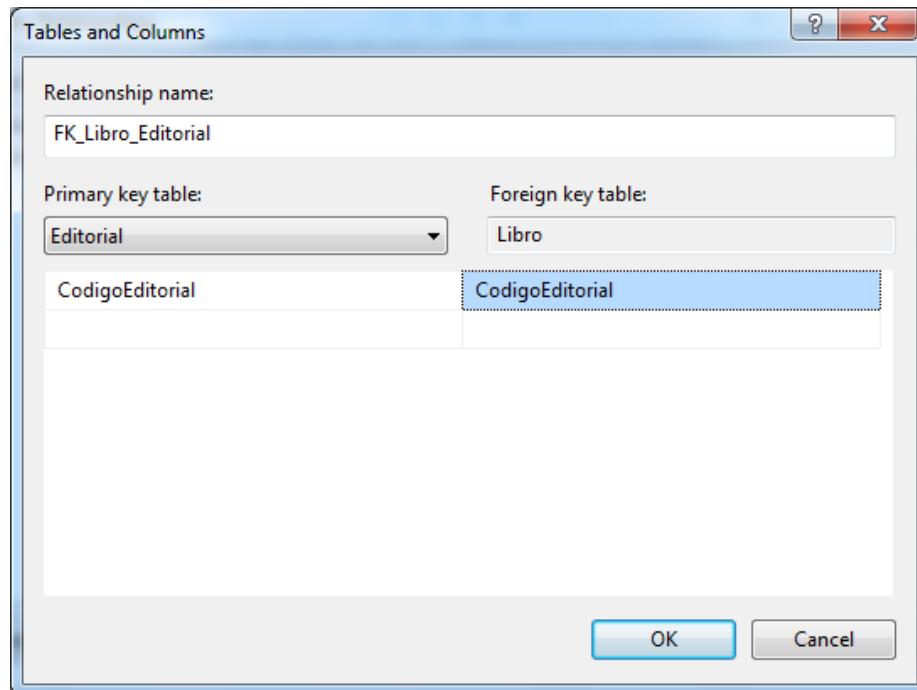
15. Le pedirá que guarde los cambios, haga clic en Si (Yes), y en la siguiente ventana haga clic en Si (Yes) para guardar los cambios en las tablas afectadas en la relación.

16. Ahora despliegue las carpetas **Columns** y **Keys** de la tabla **Detalle\_AutorLibro** y observará la creación de las relaciones:



17. Crear la relación entre las **tablas Editorial y Libros**

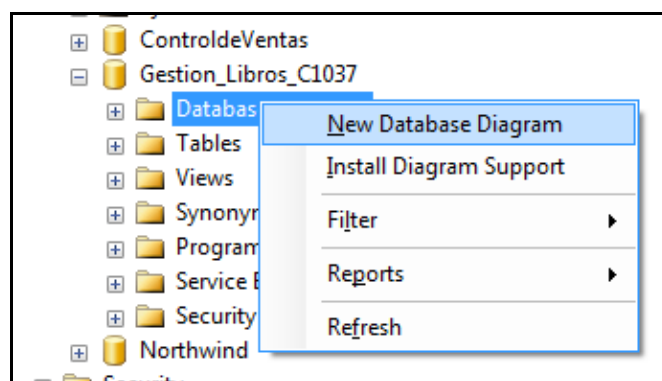
18. Ahora la tabla que llevará la clave foránea es la tabla Libros, ya que una editorial edita muchos libros pero un libro solo puede ser editado por una editorial (siguiendo nuestro modelo E-R)
19. Quedará de la siguiente manera la nueva relación



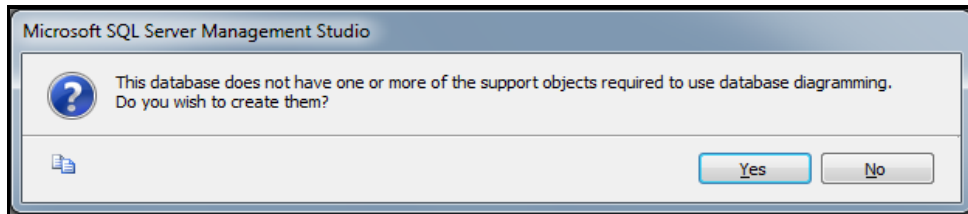
20. Guarde los cambios y observe la creación de la relación en la tabla Libros

#### Ejercicio 4. Creando el diagrama de base de datos

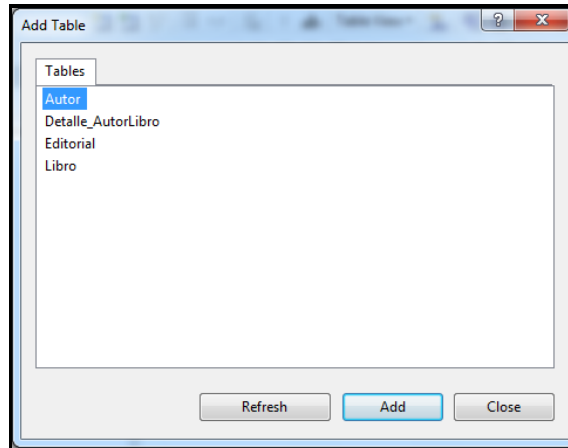
1. Cierre todas las tablas
2. Haga clic derecho sobre la opción Diagrama de Base de datos (Database Diagrams)
3. Seleccione la opción **Nuevo Diagrama de Base de datos (New Database Diagram)**



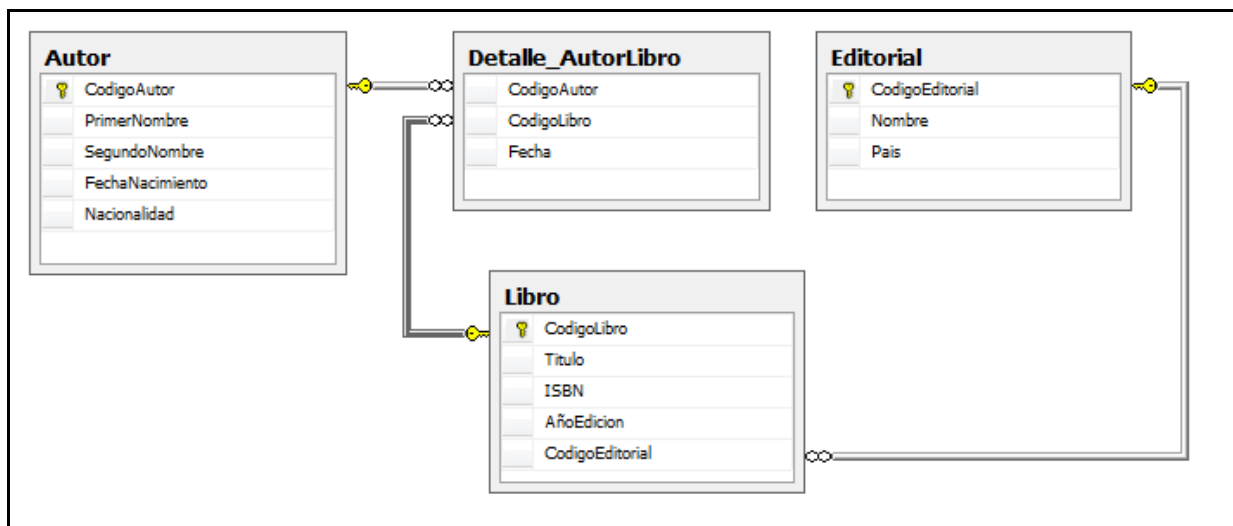
4. En la ventana emergente haga clic en Si (Yes)



5. Se habilita la venta Agregar tabla (Add Table)



6. Seleccione cada tabla y haga clic en Agregar (Add), realice este paso hasta que termine de agregar todas las tablas
7. Haga clic en Cerrar (Close)
8. Y observara que se ha creado el siguiente diseño de base de datos



9. Hasta aquí ya paso su modelo E-R a una base de datos Relacional

### Ejercicio 5. Agregando datos a las tablas

1. Haga clic derecho sobre la **tabla Autor**
2. Seleccione la opción **Edit top 200 rows**



- Se abrirá una hoja con filas y columnas en donde la columna son los campos y las filas los registros

	CodigoAutor	PrimerNombre	SegundoNombre	FechaNacimiento	Nacionalidad
*	NULL	NULL	NULL	NULL	NULL

- En donde podrá digitar por columna los siguientes registros:

CodigoAutor	PrimerNombre	SegundoNombre	FechaNacimiento	Nacionalidad
PL001	Pablo	Lopez	1960-08-19	Colombiana
CM002	Claudia	Martinez	1970-06-10	Salvadoreña
PM003	Patricio	Murry	1967-12-12	Española
NH004	Nuria	Hernandez	1980-09-03	Colombiana

- Cierre la edición de datos de la tabla Autor

- Agregar los siguientes registros:

**Tabla Editorial:**

CodigoEditorial	Nombre	Pais
ED001	Omega 2000	Colombia
ED002	Anaya Multimedia	España
ED003	McGrawHill	Inglaterra

**Tabla Libros:**

CodigoLibro	Titulo	ISBN	AñoEdicion	CodigoEditorial
BDCOL00001	Fundamentos de base de datos	12333-8999988	2004	ED001
BDESP00002	La Biblia de SQL Server 2008	3444-99888-88	2008	ED002
PRCOL00002	Programación orientada a objetos	8999-9999444	2011	ED001
DWING00003	Diseño Web y Hojas de estilo	300096-99999	2010	ED003

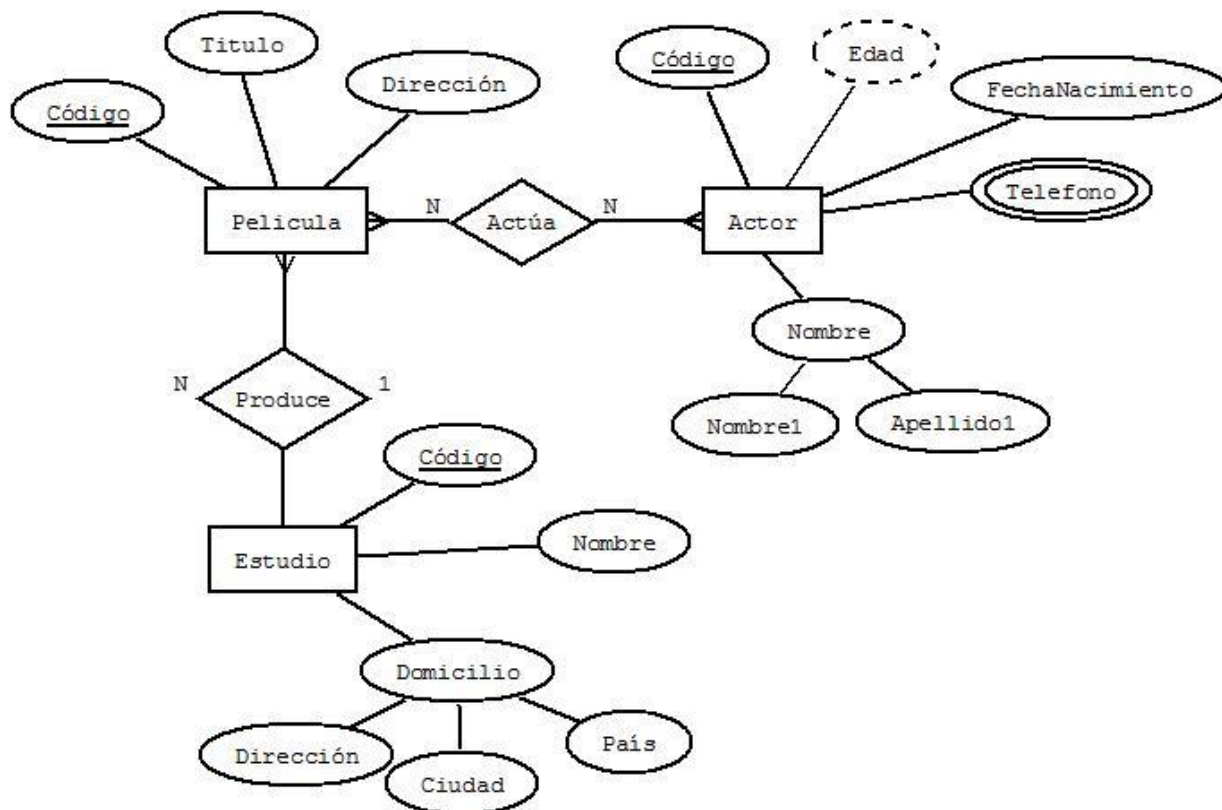
**Tabla Detalle\_AutorLibro**

CodigoAutor	CodigoLibro	Fecha
PL001	BDCOL00001	12/06/2002
NH004	BDCOL00001	12/06/2002
CM002	PRCOL00002	19/12/2009
PM003	BDESP00002	01/03/2000
PM003	DWING00003	23/03/2009

7. De esta forma ha agregado datos a las tablas en donde por medio de las relaciones que se han creado no puede agregar un dato en el campo que posee una llave foránea si este valor no existe en el campo que tiene la llave primaria
8. Realizar pruebas y analice los mensajes de error
9. Cierre todas las tablas

## V. Ejercicio complementario

Crear la siguiente base de datos en SQL Server:



Nombre de la base de datos: **Gestion\_Peliculas**

Crear:

1. La base de datos
2. Las tablas con sus correspondientes campos y propiedades
3. Las relaciones entre las tablas
4. Crear el diagrama de la base de datos
5. Agregar por los menos tres registros por cada tabla

El docente revisará su trabajo en la práctica.

## VI. Análisis de resultados

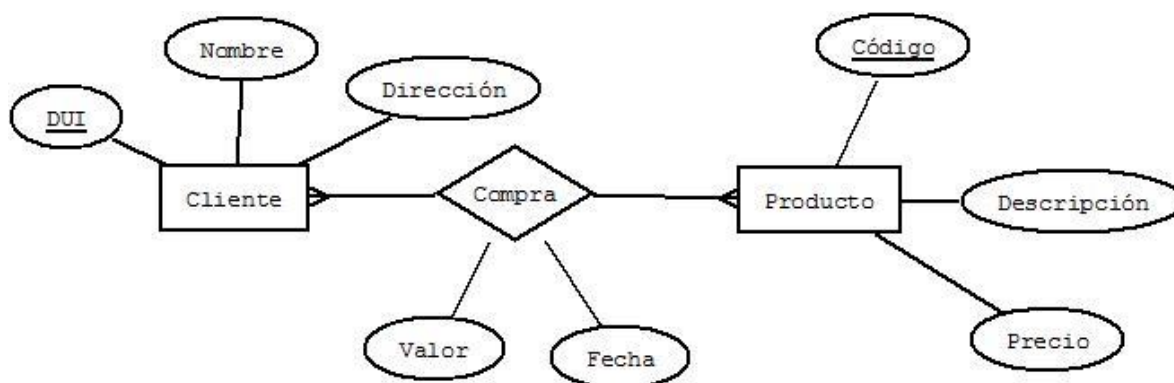
Tarea en parejas

Fecha de entrega: Según indicaciones de docente

Investigar:

1. Sobre las instrucciones SQL, las cuales se clasifican según su propósito en tres grupos:
  - a. El DDL (Data Description Language) Lenguaje de Descripción de Datos.
  - b. El DCL (Data Control Language) Lenguaje de Control de Datos.
  - c. El DML (Data Manipulation Language) Lenguaje de Manipulación de Datos.
2. Formato de los identificadores
3. Tipos de datos que se utilizan en SQL Server 2012

Tomando el siguiente Modelo E-R



Y por medio del lenguaje SQL investigue:

1. Sintaxis para crear la base de datos
2. Sintaxis para crear las tablas
3. Sintaxis para crear llave primaria y llave foráneas entre tablas, impleméntelo en las tablas del punto anterior

## VI. Fuente de consulta

1. La Biblia de SQL Server 2005

Madrid, España: Anaya, 2006

Autor: Mike Gundelerloy y Joseph L. Jorden

Biblioteca UDB – Clasificación: 005.361 G975 2006

2. Microsoft SQL Server 2008: Guía del Administrador

Madrid, España: ANAYA, 2009

Autor: William Stanek

Biblioteca UDB – Clasificación: 005.361 S784 2009