

Guía N° 3

Tema: Modelo Relacional

I. Objetivos

1. Crear una base de datos con Lenguaje SQL
2. Crear tablas y definir tipos de datos con lenguaje SQL
3. Implementar la integridad referencial en una base de datos

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

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:

- Añadir información a una tabla
- Actualizar la información que ya existe en una tabla
- Eliminar información que ya existe en una tabla
- Ver la información contenida en una table

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	0x151C2F0002000000D000E0014002100FFFFFFFF
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F0002000000D000E0014002100FFFFFFFF
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F0002000000D000E0014002100FFFFFFFF
4	4	Dairy Products	Cheeses	0x151C2F0002000000D000E0014002100FFFFFFFF
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F0002000000D000E0014002100FFFFFFFF
6	6	Meat/Poultry	Prepared meats	0x151C2F0002000000D000E0014002100FFFFFFFF
7	7	Produce	Dried fruit and bean curd	0x151C2F0002000000D000E0014002100FFFFFFFF
8	8	Seafood	Seaweed and fish	0x151C2F0002000000D000E0014002100FFFFFFFF

En la tabla anterior se puede identificar como:

Campo: CategoryID

Registro: 1, Bevarages, Soft drinks, coffees, teas etc.

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:

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

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

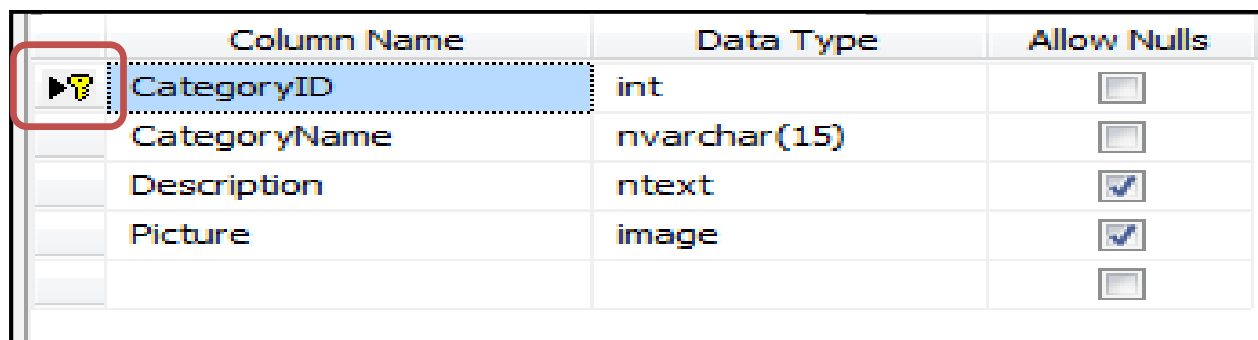
La propiedad permitir valores nulos muestra si se permiten valores nulos en ese campo, si un campo no permite valores nulos, debe ingresarse un valor no nulo para ese campo en cada registro antes de poder guardar el registro.


Propiedades de las relaciones

Las relaciones tienen las siguientes características:

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos o campos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están normalizadas.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.

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.

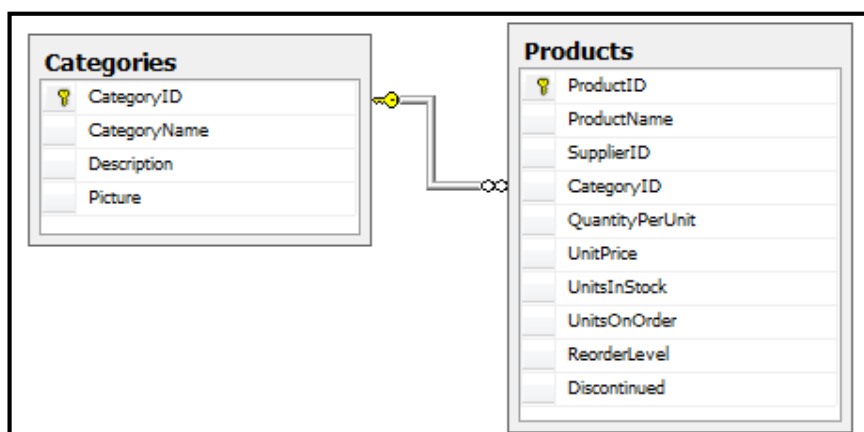
Cuando se agrega una restricción PRIMARY KEY a una o varias columnas de la tabla, el Motor de base de datos examina los datos y metadatos de columnas existentes para garantizar que se cumplen las siguientes reglas en las claves principales:

1. 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.
2. 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".

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

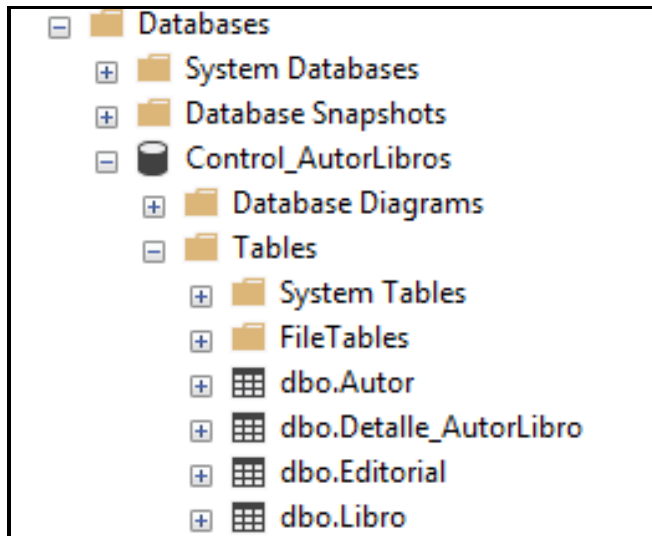
- Máquina con SQL Server 2012, 2014 o 2016
- Guía Número 3 de Modelamiento y diseño de base de datos

IV. Procedimiento

Ejercicio 1. Creando la base de datos y tablas

1. Para este ejercicio debe ejecutar el **script de la guía**, el cual lo tiene que descargar de los recursos de la Guía 3 de la página de la Universidad.

- Después de ejecutar el script, actualice (Refresh) la carpeta **Databases** tendrá la siguiente base de datos con las tablas correspondientes:



Ejercicio 2. Creando la relación entre las tablas

- Identificar cuales tablas tienen llaves primarias

Tablas	Campos	Llave primaria
Autor	CodigoAutor PrimerNombre Primer Apellido FechaNacimiento Nacionalidad	CodigoAutor
Libro	CodigoLibro Titulo ISBN AñoEdicion CodigoEditorial	CodigoLibro
Editorial	CodigoEditorial Nombre País	CodigoEditorial
Detalle_AutorLibro	CodigoAutor CodigoLibro Fecha	

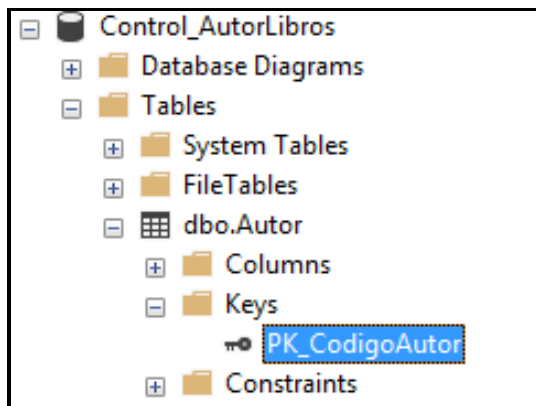
- Creando la llave primaria de la tabla Autor, utilizando siempre el script que descargo, después del último comando GO, digitar la siguiente consulta:

```
--Creando llaves primarias
--Tabla Autor

ALTER TABLE Autor
ADD CONSTRAINT PK_CodigoAutor
PRIMARY KEY (CodigoAutor)

GO
```

3. Seleccione desde la instrucción ALTER hasta el comando GO, y ejecute (EXECUTE) la consulta
4. Actualice (Refresh) la carpeta Tables y compruebe en la carpeta Keys que se creó la llave primaria



5. Crear las otras llaves primarias

```
--Tabla Libro
ALTER TABLE Libro
ADD CONSTRAINT PK_CodigoLibro
PRIMARY KEY (CodigoLibro)

GO

--Tabla Editorial
ALTER TABLE Editorial
ADD CONSTRAINT PK_CodigoEditorial
PRIMARY KEY (CodigoEditorial)

GO
```

6. Verifique en cada tabla que se han creado las llaves primarias (Primary Key), así como lo realizo en el paso 4 de este ejercicio
7. Identificar las llaves foráneas

Tablas	Campos	Llave foránea
Autor	CodigoAutor PrimerNombre Primer Apellido FechaNacimiento Nacionalidad	
Libro	CodigoLibro Titulo ISBN AñoEdicion CodigoEditorial	CodigoEditorial
Editorial	CodigoEditorial Nombre País	
Detalle_AutorLibro	CodigoAutor CodigoLibro Fecha	CodigoAutor CodigoLibro

Notas:

- Una editorial puede editar uno o más libros, por esa razón la llave foránea se coloca en la tabla Libro
- 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.

Desde el paso 8 hasta el paso 13 de este ejercicio se crean las siguientes relaciones:

Claves primarias	Claves foráneas
<p>Tabla: Autor Campo: CodigoAutor</p> <p>Tabla: Libro Campo: CodigoLibro</p>	<p>Tabla: Detalle_AutorLibro Campo: CodigoAutor Hace referencia a la clave primaria CodigoAutor de la tabla Autor</p> <p>Tabla: Detalle_AutorLibro Campo: CodigoLibro Hace referencia a la clave primaria CodigoLibro de la tabla Libro</p>

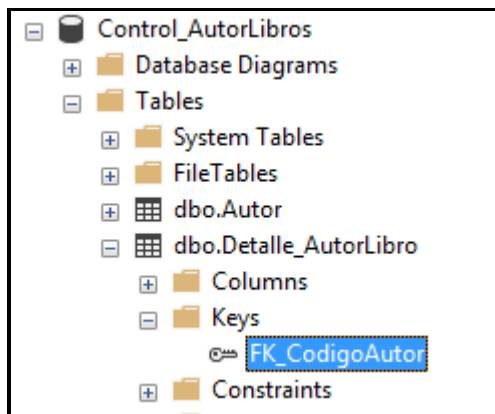
8. Creando la llave foránea, con este paso ya se está creando la relación entre las tablas, digite la siguiente consulta, después del último comando GO

```
--Creacion de llaves foraneas
--Relacion entre las tablas Autor y Detalle_AutorLibro

ALTER TABLE Detalle_AutorLibro
ADD CONSTRAINT FK_CodigoAutor
FOREIGN KEY (CodigoAutor)
REFERENCES Autor(CodigoAutor)

GO
```

9. Seleccione desde la instrucción ALTER hasta el comando GO y ejecute la consulta
10. Actualice (Refresh) la carpeta Tables y verifique que se creó en la tabla Detalle_AutorLibro la llave foránea



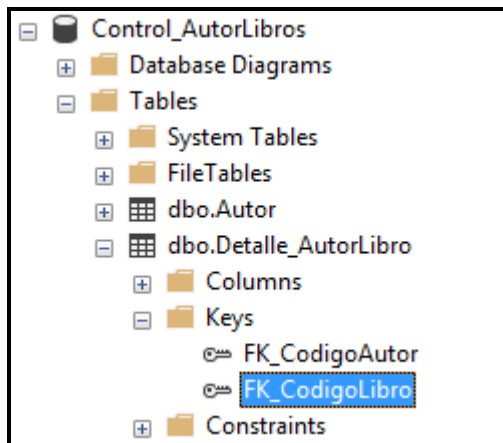
11. Creando la llave foránea entre la tabla Libro y Detalle_AutorLibro, en el mismo script de la guía, y después del último GO, digite la siguiente consulta:

```
--Relacion entre las tablas Libro y Detalle_AutorLibro

ALTER TABLE Detalle_AutorLibro
ADD CONSTRAINT FK_CodigoLibro
FOREIGN KEY (CodigoLibro)
REFERENCES Libro(CodigoLibro)

GO
```

12. Seleccione desde la instrucción ALTER hasta el comando GO y ejecute la consulta
13. Actualice (Refresh) la carpeta Tables y verifique que se creó en la tabla Detalle_AutorLibro la llave foránea



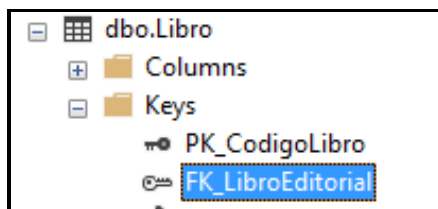
14. Por ultimo crear la llave foránea en la tabla Libro, para relacionar las tablas Editorial y Libro, digitar y ejecutar la siguiente consulta:

```
--Relacion entre las tablas Editorial y Libro

ALTER TABLE Libro
ADD CONSTRAINT FK_LibroEditorial
FOREIGN KEY (CodigoEditorial)
REFERENCES Editorial(CodigoEditorial)

GO
```

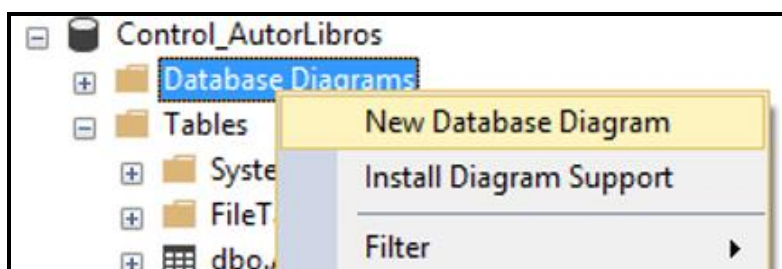
15. Actualice (Refresh) la carpeta Tables y verifique que se creó en la tabla Libro la llave foránea



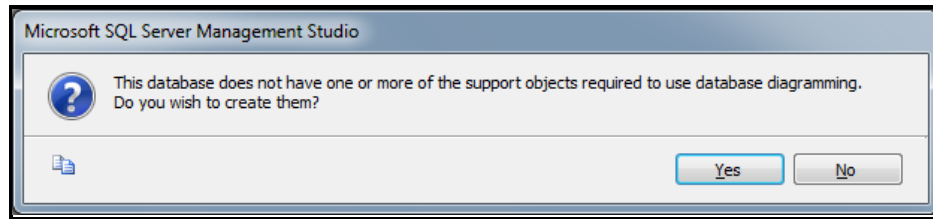
16. Guardar los cambios del script

Ejercicio 3. Creando el diagrama de base de datos

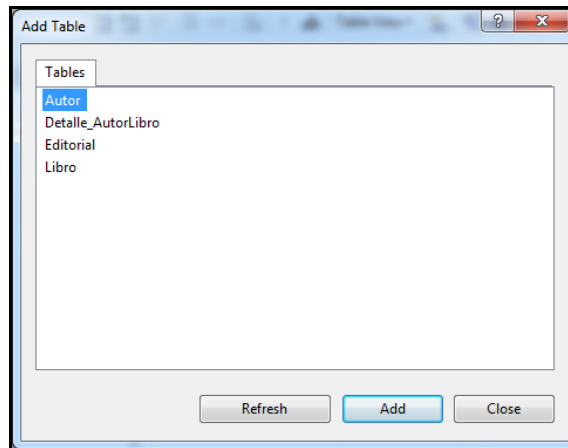
1. Haga clic derecho sobre la opción Diagrama de Base de datos (Database Diagrams)
2. Seleccione la opción **Nuevo Diagrama de Base de datos (New Database Diagram)**



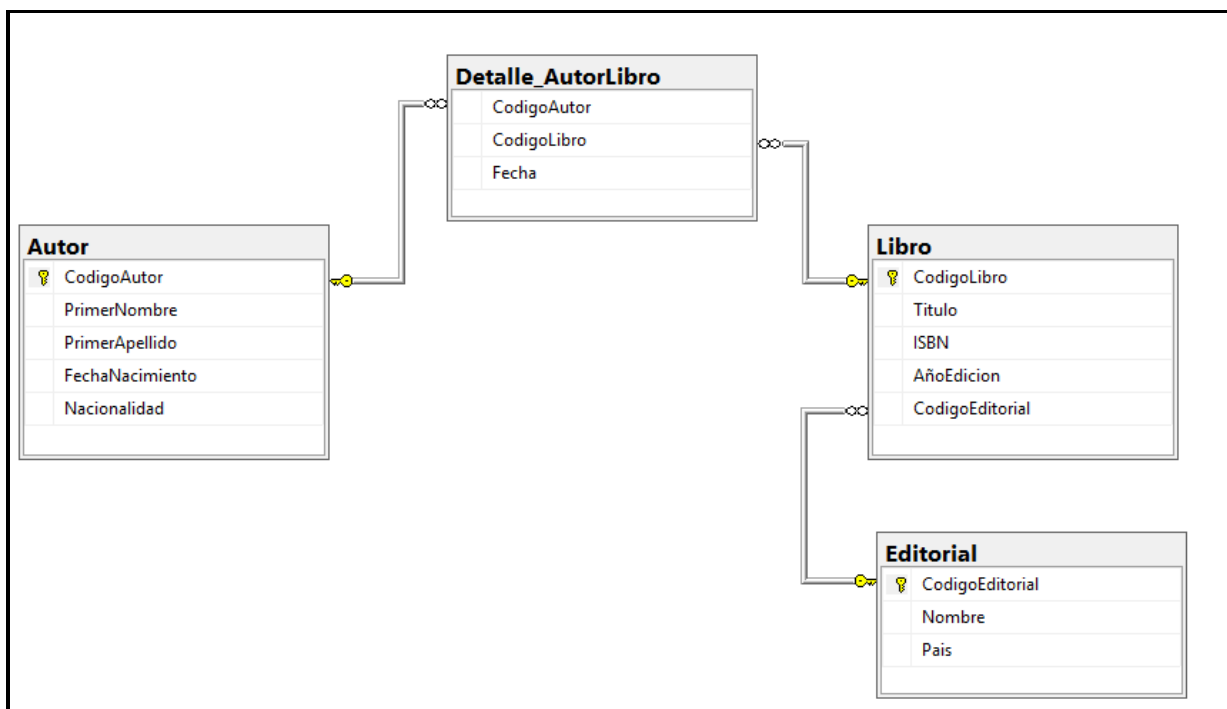
3. En la ventana emergente hacer clic en Si (Yes)



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



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



8. Guarde el diseño de la base de datos, coloque como nombre: Diagram_Guia3
9. Cierre el diagrama de la base de datos

Ejercicio 4. Eliminando la base de datos del servidor

1. Eliminar la base, en el mismo script de la guía, digite la siguiente consulta:

```
--Eliminando la base de datos

DROP DATABASE Control_AutorLibros
```

2. Seleccione la consulta y ejecútela
3. Actualice (Refresh) la carpeta Databases y verifique que ya no está la base de datos en el servidor de SQL Server

Ejercicio para trabajarlo en la clase

En una nueva área de edición de consultas crear la siguiente base de datos en SQL Server:

Nombre de la base de datos: **Hotel_SuCarnet**

Tablas:

Tablas	Campos
Hotel	CodigoHotel Nombre Direccion SitioWeb
Habitacion	CodigoHabitacion CodigoHotel Tipo Precio
Huesped	CodigoHuesped Nombre Email Dirección Telefono
Reservacion	CodigoReserva CodigoHotel CodigoHuesped CodigoHabitacion FechaInicio FechaFin

Haciendo uso de TRANSACT – SQL crear:

1. La base de datos
2. Las tablas con sus correspondientes campos y propiedades
3. Crear las siguientes restricciones en los campos:
 - a. Unique:
 - i. Tabla Hotel (Nombre, SitioWeb)
 - ii. Tabla Huesped (E-mail)
 - b. Check:
 - i. Tabla Habitacion (Precio mayor que 25, Tipo: Doble, individual)
 - ii. Tabla Reservacion (Fecha fin tiene que ser mayor a la fecha de inicio)
 - c. Default:
 - i. Tabla Reservacion (Fecha inicio por defecto puede ser la fecha actual del sistema)
4. Crear las relaciones entre las tablas identificando las llaves foráneas y primarias por cada tabla
5. Guarde el archivo de las consultas con el nombre **Consultas_Hotel.sql**, y enviarlo al docente

V. Ejercicio complementario

PROBLEMA: GESTIÓN DE ALUMNOS

La secretaría de una Universidad necesita crear una base de datos que controle información sobre las FACULTADES, los ALUMNOS, los PROFESORES, las MATERIAS y las CARRERAS ofertadas.

Con respecto a la base de datos deberá tomar en cuenta los siguientes aspectos:

- a) Una facultad puede poseer muchas carreras
- b) Un alumno puede estar matriculado en una sola carrera
- c) Cada materia puede pertenecer a una o varias carreras (por ejemplo. Matemáticas, programación etc.)
- d) Cada materia puede estar impartida por uno o más profesores y un profesor puede impartir una o más materias

Haciendo uso de TRANSACT – SQL crear:

1. La base de datos
2. Las tablas con sus correspondientes campos y propiedades
3. Las relaciones entre las tablas
4. Restricciones en todos los campos que crea conveniente
5. Crear el diagrama de la base de datos

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

1. Libros en pantalla de SQL Server 2008 R2
[http://msdn.microsoft.com/es-es/library/ms176061\(v=sql.105\).aspx](http://msdn.microsoft.com/es-es/library/ms176061(v=sql.105).aspx)
[http://msdn.microsoft.com/es-es/library/bb510741\(v=sql.105\).aspx](http://msdn.microsoft.com/es-es/library/bb510741(v=sql.105).aspx)