

SQL SERVER

SQL SERVER es un motor de bases de datos que permite modelar, diseñar y almacenar bases de datos relacionales para su funcionamiento.

Para crear y administrar una base de datos en SQL SERVER es necesario comprender las siguientes definiciones.

Base de Datos: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Servicio Motor de Bases de Datos: Microsoft SQL SERVER presenta varios servicios dentro de los cuales se encuentra el “*Motor de bases de datos*”, hay otros como: “*Reporting Services*”, “*Analysis Services*”, “*Integration Services*”. Para la creación y utilización de bases de datos se requiere ejecutar el servicio de “**Motor de Bases de Datos**”.

Servidor: Es un equipo de computador donde se encuentra instalado el Servicio de SQL SERVER, este servidor proporciona acceso a un conjunto de usuarios que pueden acceder a las bases de datos para su utilización.

Instancia: Es un grupo o espacio dentro de un servidor que permite la creación de bases de datos. Ejemplo:

Bajo la instancia Server100/**Institucion** se encuentran almacenadas las siguientes bases de datos: Institucion, Parqueadero, Transportadora.

Bajo la instancia Server100/**Direccion** se encuentran almacenadas las siguientes bases de datos: Pizzería, Restaurante, Banco.

Observar que bajo el **Server100** se encuentran instaladas dos instancias: “**Institucion**” y “**Direccion**” y dentro de estas instancias se encuentran N bases de datos almacenadas.

Autenticación: Es el tipo de acceso que permite Microsoft SQL SERVER para utilizar las bases de datos, hay dos tipos de autenticación: *Windows Authentication* y *SQL SERVER Authentication*.

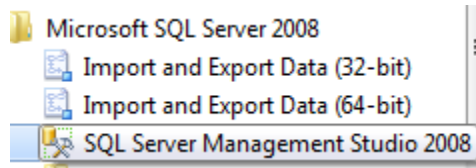
Windows Authentication: Este tipo de autenticación es utilizada dentro de las empresas cuando existe un directorio activo y una red configurada. Cuando el usuario accede al servicio de SQL SERVER no ingresa usuario ni contraseña, ya que SQL SERVER obtiene la información del equipo de la persona. El nombre de usuario y contraseña fue ingresado cuando la persona ingreso al dominio de la empresa que ocurre cada vez que prende el computador.

Introducción SQL SERVER –John Cañaveral

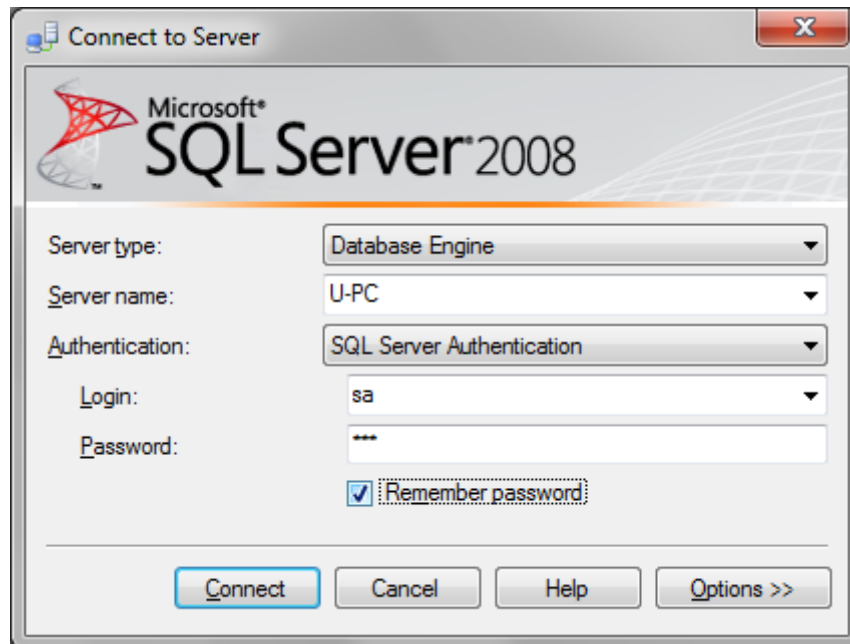
SQL SERVER Authentication: Es la más utilizada en la actualidad. Se requiere un nombre de usuario y contraseña para ingresar al servicio de SQL SERVER.

Paso a Paso: Acceder al Servicio de Motor de Bases de Datos SQL SERVER

1. Abrir Microsoft SQL SERVER 2008, clic en la opción “SQL SERVER Management Studio 2008”, como se presenta en la siguiente figura:



2. Se presenta la siguiente pantalla:



Ingresar las opciones que solicita:

Server Type: Motor Base de Datos

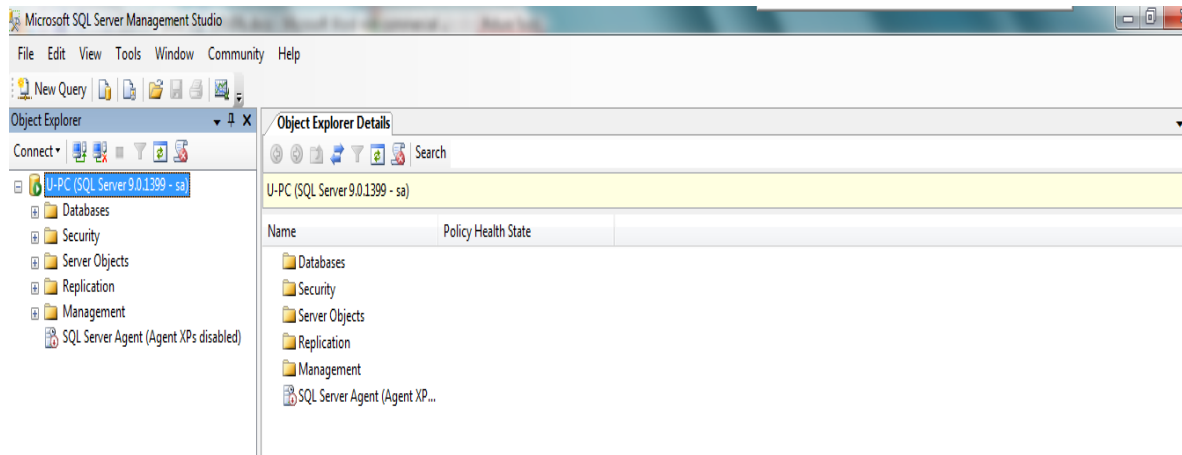
Server name: NombreServidor\\instancia

Authentication: SQL Server Authentication, Ingresar login y password

Clic en Conectar.

Introducción SQL SERVER –John Cañaveral

3. En la parte izquierda se muestra en forma de árbol el conjunto de herramientas al que se pueden acceder: Bases de Datos, Security, Server Objects, Replication, Management, Notification Services, SQL Server Agent.



3. Expandir la opción Bases de Datos, se presenta la siguiente imagen:



Observar que se muestran cada una de las base de datos en forma de carpeta Windows que se encuentran almacenadas bajo el servidor e instancia al cual se accedió. Al momento de acceder a la base de datos, se muestra un subconjunto de carpetas que permiten administrar la base de datos. A continuación se describen cada uno de ellos:

Database Diagrams: Permite la visualización de las tablas y sus relaciones de la base de datos. Normalmente por esta opción se relacionan las tablas para tener un mayor control sobre las mismas.

Tables: Permite la creación de las tablas de la base de datos, así como la inserción de los registros de las tablas, claves primarias, relaciones etc.

Views: Permite crear vistas sobre un conjunto de tablas relacionadas. Las vistas son una forma de crear conjunto de datos con necesidades específicas; las vista solo son creadas con fines de consulta. Es una alternativa para ver ciertos registros de un grupo de tablas relacionadas.

Introducción SQL SERVER –John Cañaveral

Synonyms: Permite hacer referencia a nuestros objetos de base de datos de una manera más entendible, clara, o simplemente para abreviar los nombres de tal manera que faciliten su uso en las consultas.

Programability: Permite la creación de procedimientos almacenados, funciones y otras funcionalidades para realizar cálculos sobre las tablas de la base de datos.

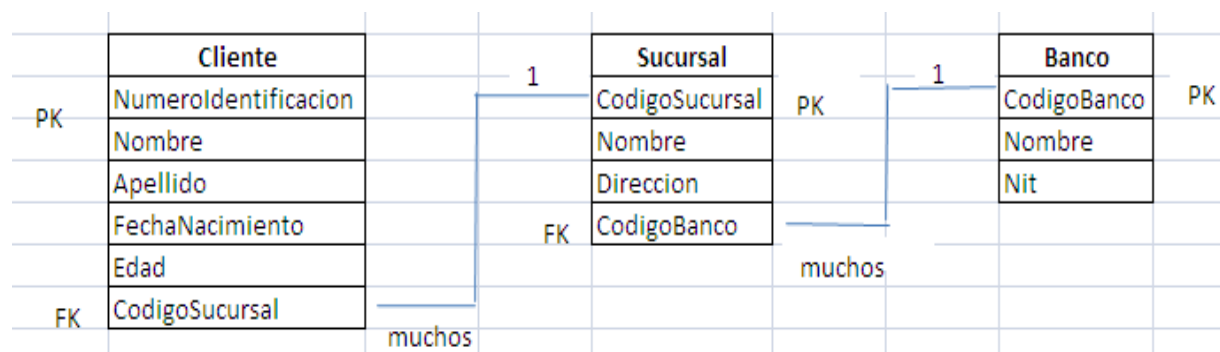
Service Broker: Permite la creación de aplicaciones distribuidas y confiables cuando estas acceden a diferentes bases de datos.

Storage: Permite la creación de configuraciones para la búsqueda sobre columnas de texto completo que usualmente no se pueden realizar por los filtros normales dentro de una consulta.

Security: Permite administrar los usuarios y roles para el acceso de cada uno de los objetos de la base de datos.

Paso a Paso: Creación de una Base de Datos SQL SERVER 2008

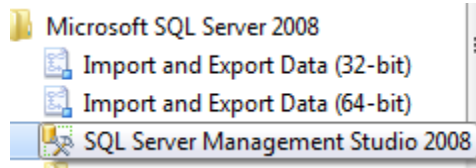
Problema: Diseñar la base de datos para un Banco que permita administrar las sucursales a las que pertenece y los clientes de las sucursales. El cliente solo puede pertenecer a una única sucursal. El modelo de datos es el siguiente:



Una vez modelada la base de datos en papel, se procede con la creación de la base de datos.

1. Abrir Microsoft SQL SERVER 2008, clic en la opción “SQL SERVER Management Studio 2008”, como se presenta en la siguiente figura:

Introducción SQL SERVER –John Cañaveral



Ingresar al servicio de base de datos:

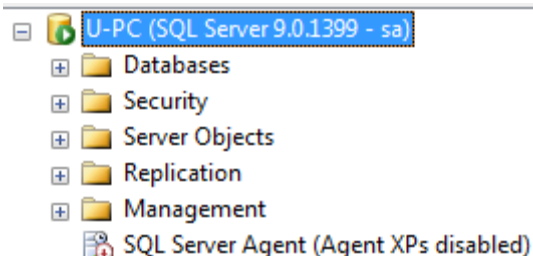
Tipo Servicio: Database Engine

Server Name: Nombre servidor\Instancia

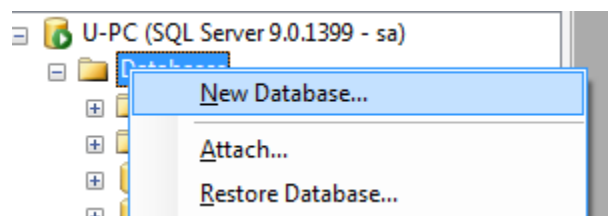
Authentication: SQL Server Authentication, ingresar Login y Password.



2. Se presenta la siguiente imagen, expandir la carpeta Databases.

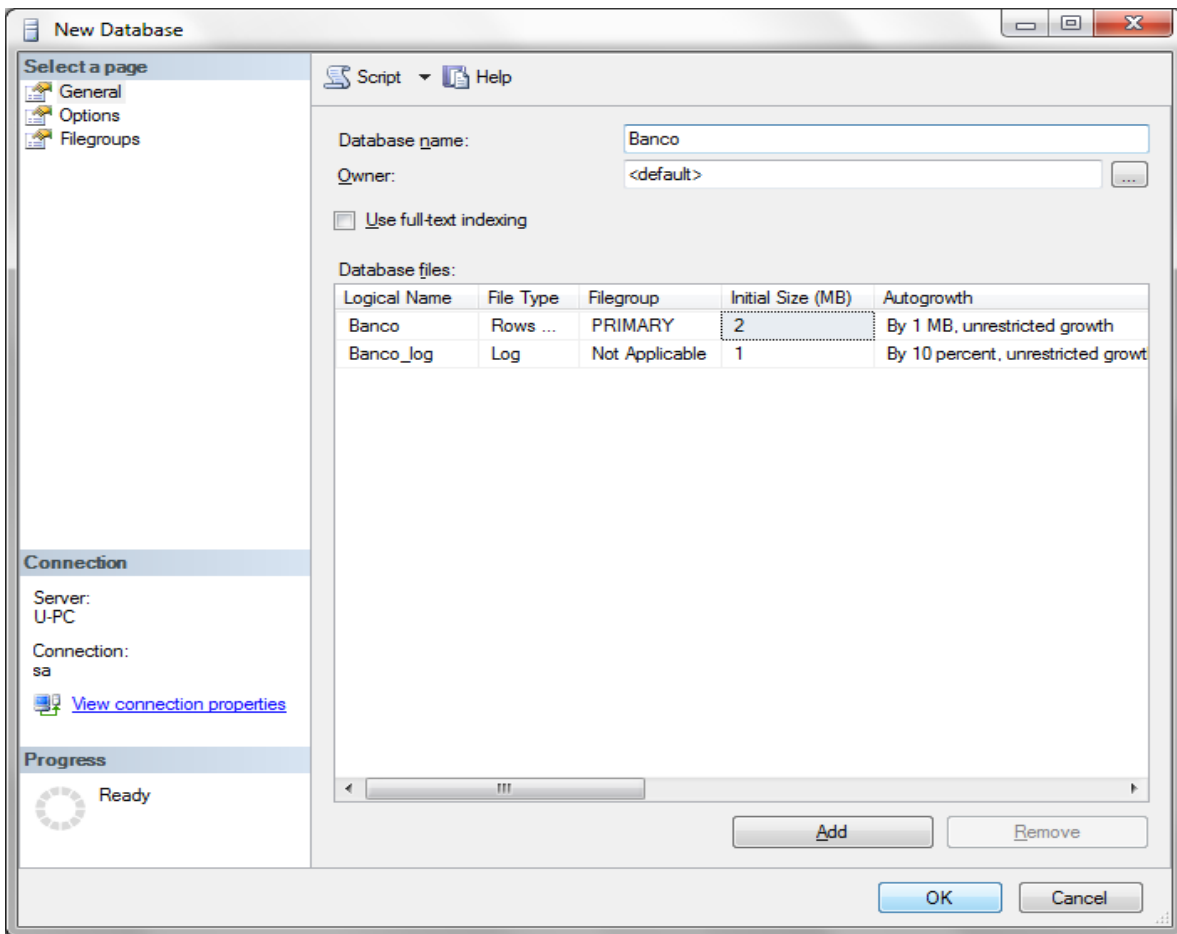


3. Clic derecho sobre la carpeta Databases y seleccionar la opción “Nueva Base de Datos”

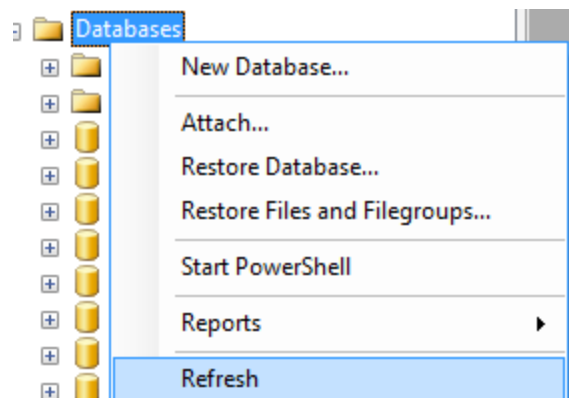


4. En el nombre de la base de datos ingresar “Banco”, clic en Aceptar.

Introducción SQL SERVER –John Cañaveral

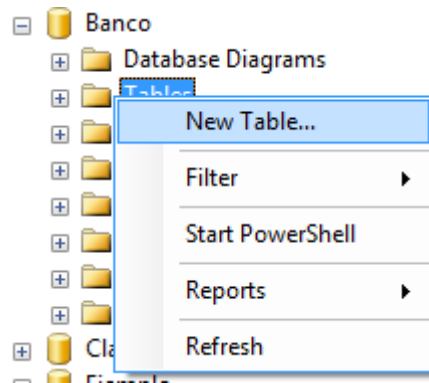


5. La base de datos es creada. Actualizar la carpeta Databases para ver la base de datos recién creada, clic derecho sobre Databases y clic en la opción Actualizar.



5. Ahora se procede con la creación de las tablas. Expandir la carpeta de la base de datos “Banco”, clic derecho sobre Tables, “New table”.

Introducción SQL SERVER –John Cañaveral



6. Para crear la tabla, se inicia con la creación de las columnas y finalmente se asigna el nombre a la tabla. Iniciar con la creación de la tabla Banco que tiene las siguientes columnas:

CodigoBanco, tipo dato: int, no permitir nulos

Nombre, tipo dato: varchar(50), no permitir nulos

Nit, tipo dato: varchar(10), no permitir nulos

U-PC.Banco - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls
▶	CodigoBanco	int	<input type="checkbox"/>
	Nombre	varchar(50)	<input type="checkbox"/>
	Nit	varchar(10)	<input type="checkbox"/>
			<input type="checkbox"/>

Cada columna o atributo de cada tabla en SQL SERVER tiene 3 conceptos que se deben de configurar:

- *Nombre de la columna:* el nombre de la columna o atributo
- *Tipo de dato:* Define el tipo de dato de la columna, por lo general los más utilizados son int (numero entero), datetime (fecha), varchar(longitud) para almacenar texto con una determinada longitud.
- *Permite nulos:* Define si el campo permite Nulos, esto indica si el campo es requerido o no. Ocurre cuando el registro es insertado en la tabla y el valor para la celda no es proporcionado, en este caso se almacena el valor NULL

Nota: La palabra reservada NULL indica que no es NADA y representa la no existencia de información en una celda.

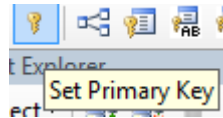
Nota: No es recomendable crear columnas con nombres con espacios, eje: Nombre Banco, es mejor unirlos con _, asi: Nombre_Banco

7. Definir la clave primaria para el campo CodigoBanco. Seleccionar la columna CodigoBanco.

Introducción SQL SERVER –John Cañaveral

U-PC.Banco - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls
▶	CodigoBanco	int	<input type="checkbox"/>
	Nombre	varchar(50)	<input type="checkbox"/>
	Nit	varchar(10)	<input type="checkbox"/>
			<input type="checkbox"/>

Clic en el icono superior que representa una llave.

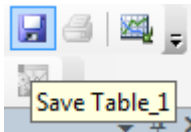


La clave primaria de la tabla es asignada.

U-PC.Banco - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls
▶	CodigoBanco	int	<input type="checkbox"/>
	Nombre	varchar(50)	<input type="checkbox"/>
	Nit	varchar(10)	<input type="checkbox"/>
			<input type="checkbox"/>

Nota: Cuando la tabla posee más de una clave primaria basta con realizar el mismo procedimiento manteniendo presionado la tecla Control.

8. Una vez creada la estructura de la tabla, proceder a guardar la tabla. Clic en el



icono de ingresar el nombre de la tabla Banco.

Choose Name

Enter a name for the table:

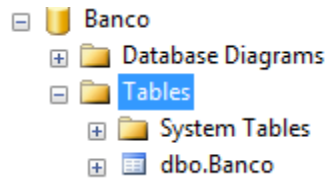
Banco

OK

Cancel

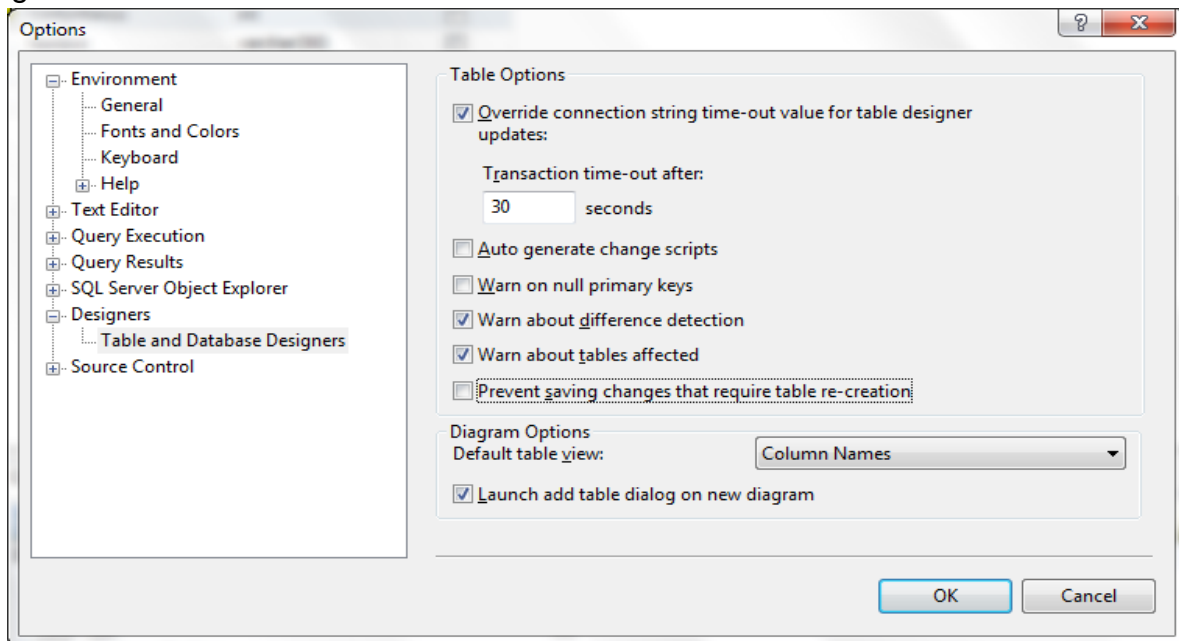
La tabla es almacenada. Para verla, refrescar la carpeta Tables: clic derecho sobre la carpeta y clic en Actualizar,

Introducción SQL SERVER –John Cañaveral



Nota: En el caso de que sea necesario modificar la estructura de la tabla, es necesario habilitar esta opción a SQL SERVER 2008: clic en Tools—Opciones—Designers, quitar la opción: ☐ Prevent saving changes that require table re-creation

Figura 1



Clic en Aceptar.

9. Crear las dos tablas Cliente y Sucursal con la siguiente configuración:

Cliente

Numeroidentificacion, varchar(10), No permite nulos
Nombre, varchar(50), No permite nulos
Apellido, varchar(50), No permite nulos
FechaNacimiento, datetime, No permite nulos
Edad, int, permite nulos
CodigoSucursal, int, No permite nulos

Sucursal


CodigoSucursal, int, No permite nulos
Nombre, varchar(50), No permite nulos

Introducción SQL SERVER –John Cañaveral


Direccion, varchar(50), No permite nulos

CodigoBanco, int, No permite nulos


Como se presenta en la siguiente imagen:

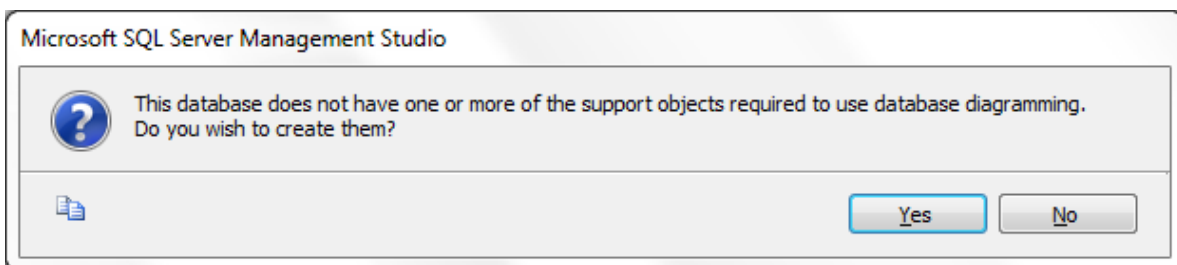
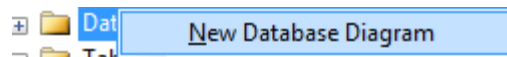
U-PC.Banco - dbo.Cliente			
	Column Name	Data Type	Allow Nulls
	NumeroIdentificacion	varchar(10)	<input type="checkbox"/>
	Nombre	varchar(50)	<input type="checkbox"/>
	Apellido	varchar(50)	<input type="checkbox"/>
	FechaNacimiento	datetime	<input type="checkbox"/>
	Edad	int	<input checked="" type="checkbox"/>
	CodigoSucursal	int	<input type="checkbox"/>

Cliente

U-PC.Banco - dbo.Sucursal			
	Column Name	Data Type	Allow Nulls
	CodigoSucursal	int	<input type="checkbox"/>
	Nombre	varchar(50)	<input type="checkbox"/>
	Direccion	varchar(50)	<input type="checkbox"/>
	CodigoBanco	int	<input type="checkbox"/>

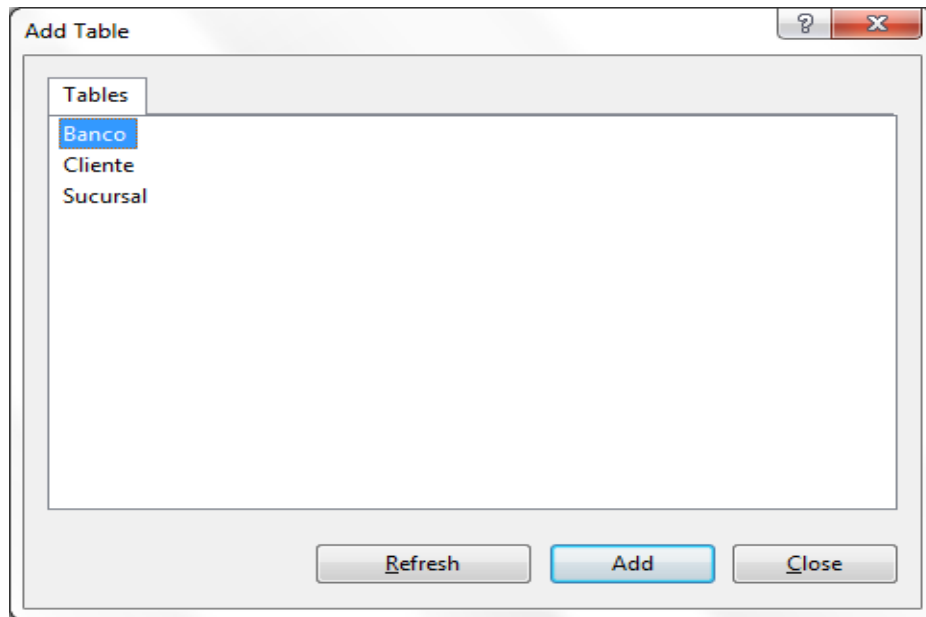
Sucursal

10. Una vez creadas las tablas, se procede con la creación de las relaciones entre las tablas de acuerdo al diseño que se realizó inicialmente. En esta base de datos existen las relaciones: Banco – Sucursales (1-muchos), Sucursal – Cliente (1-muchos). Para crear las relaciones, se crea el diagrama de base de datos; en la carpeta  Database Diagrams , clic derecho – New Database Diagram—clic en Sí.

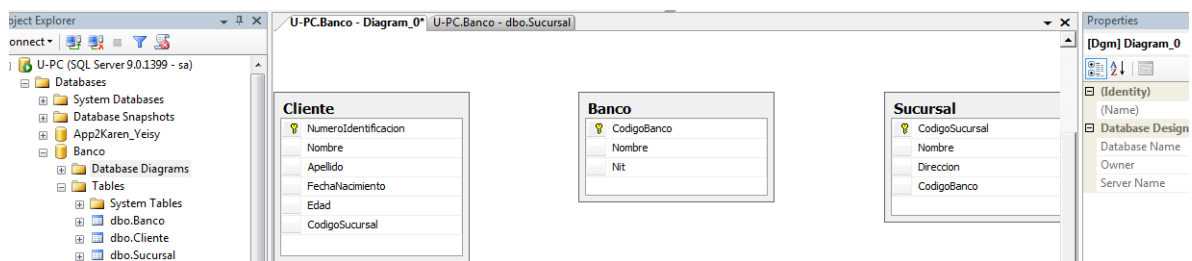


11. Adicionar las tablas que se van a relacionar.

Introducción SQL SERVER –John Cañaverál



12. Se presentan las tablas en el diagrama



13. Para relacionar dos tablas con tipo de relación 1-muchos, siempre se debe de iniciar en sentido padre (1)-hijo (muchos); iniciamos con la relación Banco (padre)-Sucursal (hijo). Seleccionar el campo Clave de la tabla Banco, el cual es **CodigoBanco** arrastrarlo con clic izquierdo presionado hacia el campo de la tabla Sucursal CodigoBanco, una vez el cursor se encuentre por encima del campo CodigoBanco soltar el clic. Se presenta la siguiente imagen:

Introducción SQL SERVER –John Cañaveral

Tables and Columns

Relationship name:
FK_Sucursal_Banco

Primary key table:
Banco

Foreign key table:
Sucursal

Primary key table	Foreign key table
CodigoBanco	CodigoBanco

OK Cancel

Se presenta en la parte superior el nombre de la relación, ingresar Relacion_Banco_Sucursal y clic en Ok.

Foreign Key Relationship

Selected Relationship:
Relacion_Banco_Sucursal*

Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

☒ (General)

☐ Check Existing Data On Creation Yes

☒ Tables And Columns Specification

☒ Database Designer

Enforce For Replication Yes

Enforce Foreign Key Constraint Yes

☒ INSERT And UPDATE Specification

☒ Identity

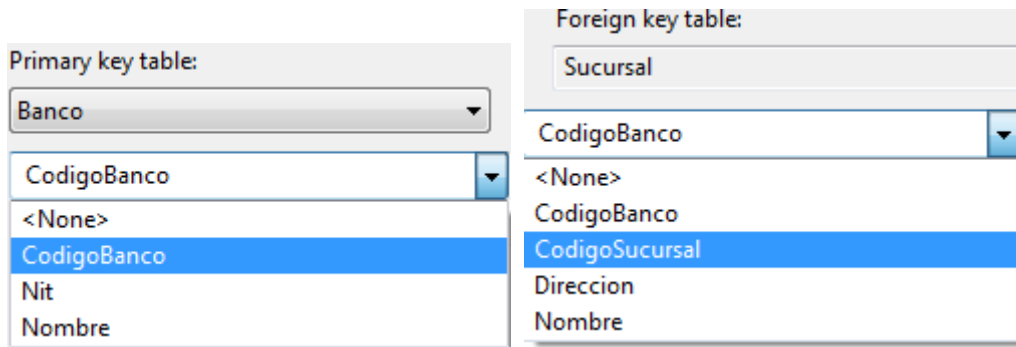
(Name)	Description
Relacion_Banco_Sucursal	

OK Cancel

Una vez definidos los campos, se presenta una ventana donde se permite configurar otras opciones de la relación. Clic en OK.

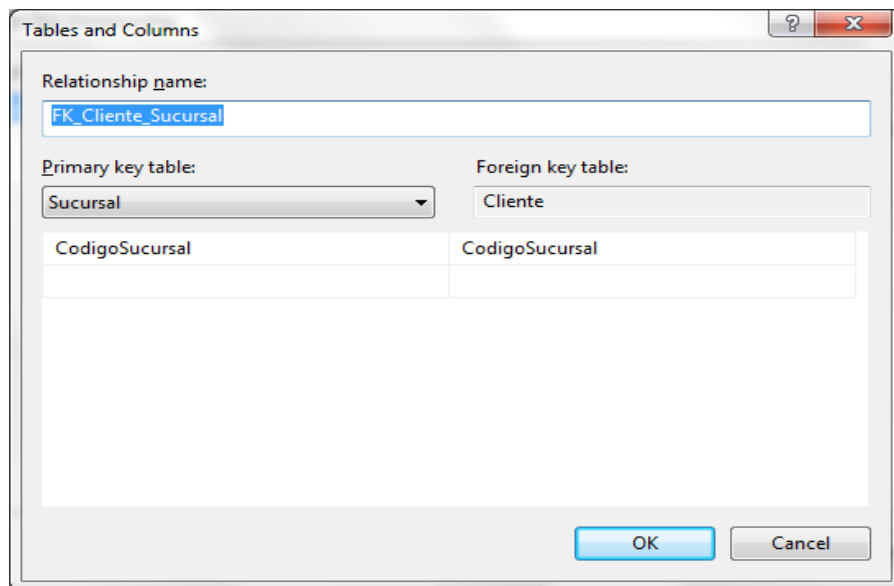
Nota: Observar que en el caso de haber soltado el clic sobre un campo de la tabla sucursal que no tiene relación con CodigoBanco, es posible modificar el campo en ambos sentidos (Banco-Sucursal), como se muestra a continuación.

Introducción SQL SERVER –John Cañaveral



Nota: Antes de relacionar dos campos de dos tablas, se debe de revisar que estos cumplan exactamente con el mismo tipo de datos y longitud. Ejemplo: CodigoBanco int de la tabla Banco y CodigoBanco int de la tabla Sucursal.

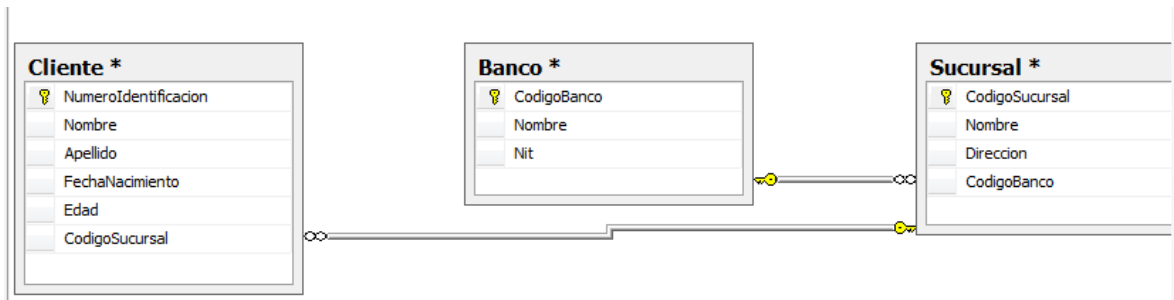
14. Relacionar las tablas Sucursal y Cliente. Seleccionar el campo clave de la tabla Sucursal quien es la tabla padre en esta relación y soltarla sobre el campo CodigoSucursal de la tabla Cliente.




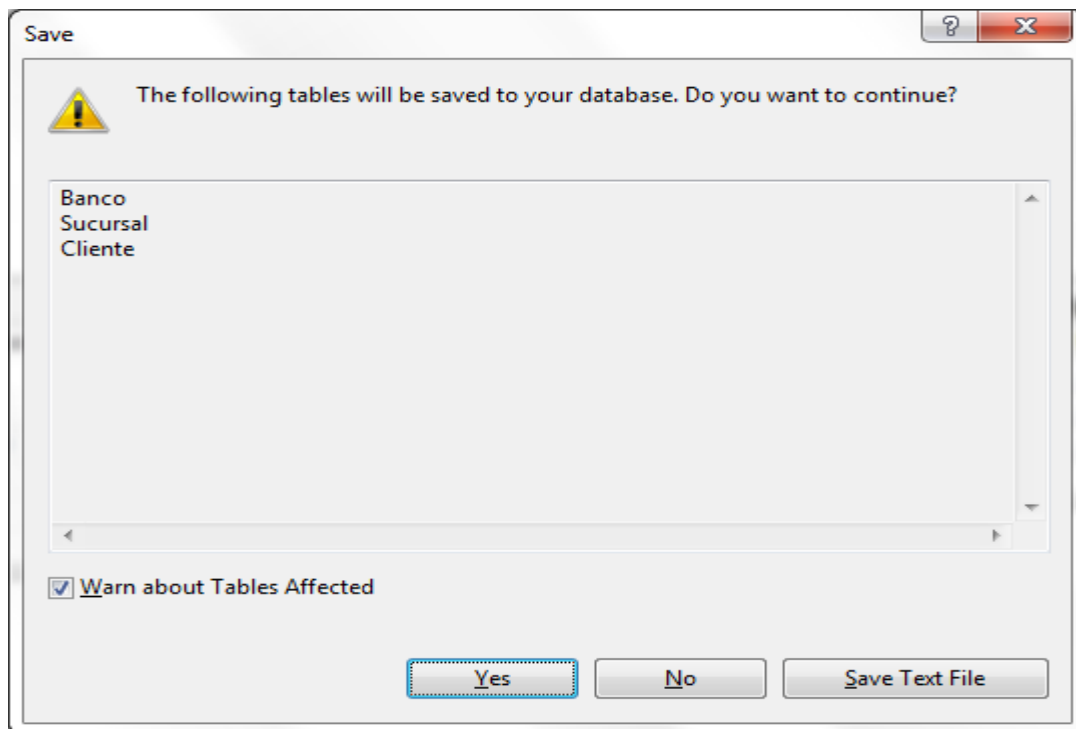
Clic en Ok, nuevamente clic en OK.

15. Las relaciones son creadas, como se presenta en la siguiente imagen:

Introducción SQL SERVER –John Cañaveral



El símbolo * asterisco indica que las relaciones de las tablas no han sido guardadas, clic en , se presenta una ventana solicitando el nombre del diagrama, ingresar un nombre para el diagrama, se presenta la siguiente imagen:



SQL SERVER pregunta que si desea almacenar los cambios sobre las tablas, clic en Sí.

Ha finalizado la creación de la base de datos, sus tablas y relaciones, ahora se debe proceder con la inserción de los registros.

16. Para insertar los registros, se debe de tener en cuenta iniciar con las tablas maestras o padres, en este caso el orden de las tablas para insertar los registros es:

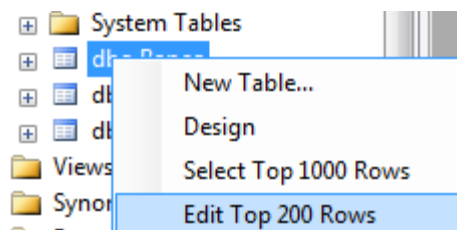
a. *Banco*, porque la tabla Sucursal tiene el campo CodigoBanco con la cual existe la referencia por la relación que se creó en ambas tablas.

Introducción SQL SERVER –John Cañaveral

- b. *Sucursal*, porque la tabla Cliente tiene el campo CodigoSucursal con la cual existe la referencia por la relación que se creó en ambas tablas.
- c. *Cliente*

Nota: Cuando se menciona **referencia física** significa que antes de ingresar el valor para el campo CodigoBanco o CodigoSucursal, este campo debe de existir en la tabla con la cual está relacionada; en el caso de no existir SQL SERVER genera un mensaje de error informando el problema en la relación de ambas tablas.




17. Clic derecho sobre la tabla Banco – Editar las primeras 200 filas



18. Se presenta la siguiente imagen:

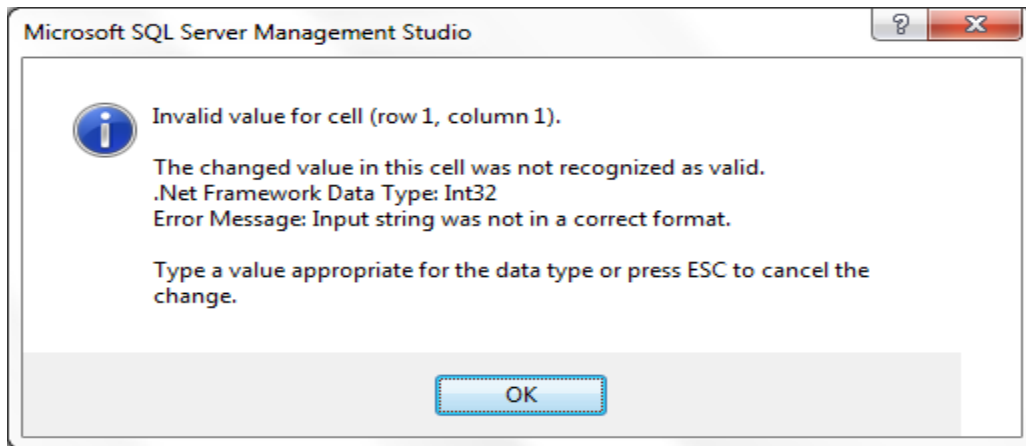
U-PC.Banco - dbo.Banco			
	CodigoBanco	Nombre	Nit
*	NULL	NULL	NULL

19. Ingresar los valores para cada campo, CodigoBanco, Nombre y Nit. Presionar Enter después de finalizar con el primer registro.

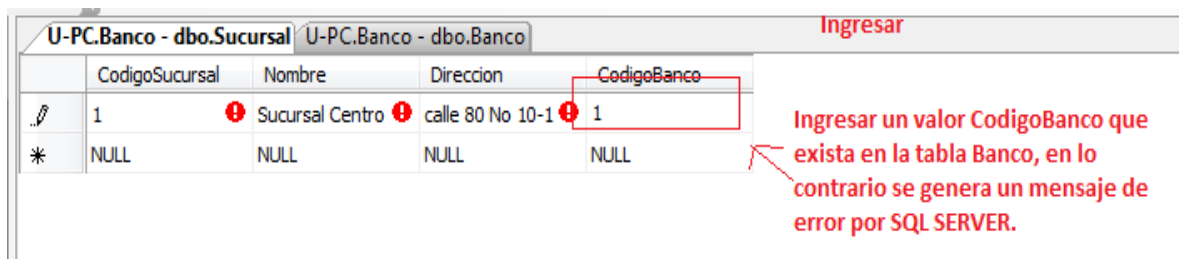
U-PC.Banco - dbo.Banco			
	CodigoBanco	Nombre	Nit
	1	 Bancolombia	 1023
*	NULL	NULL	NULL

Nota: En el proceso de inserción de datos sobre las tablas, se debe de tener cuidado con el tipo de dato definido en la columna, por ejemplo si se ingresara el texto “hola” en el campo CodigoBanco se genera un mensaje de error a causa de que la columna CodigoBanco solo recibe valores enteros porque así está definida en la estructura de la tabla.

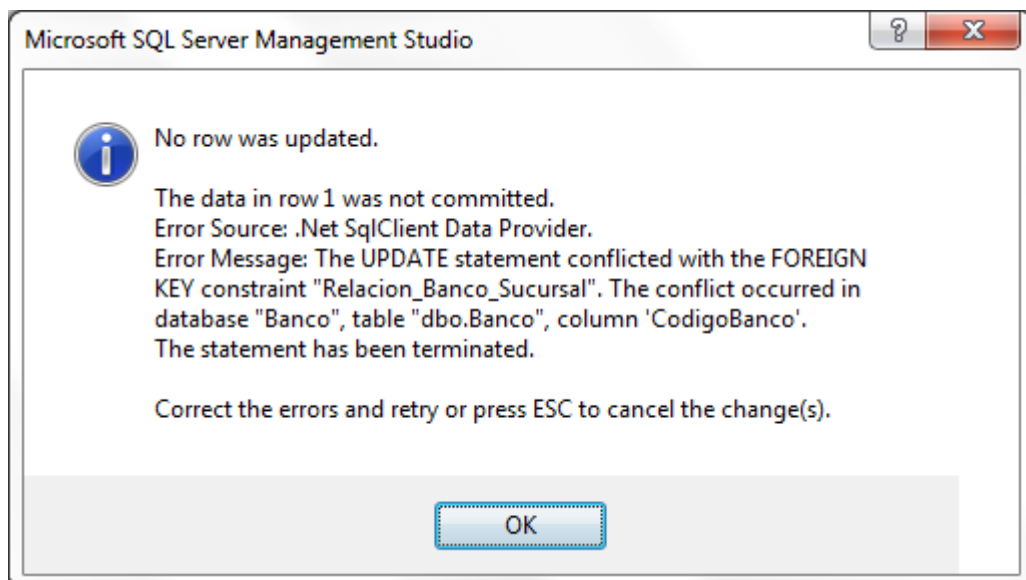
Introducción SQL SERVER –John Cañaveral



20. Realizar el mismo procedimiento con la tabla Sucursal.




Nota: En el caso de que el valor ingresado en la columna CodigoBanco de la tabla Sucursal no exista como clave primaria del campo CodigoBanco de la tabla Banco se genera el siguiente mensaje de error:



Introducción SQL SERVER –John Cañaveral

Este error informa que la clave foránea CódigoBanco de la tabla Sucursal no existe en la tabla Banco.

21. Realizar el mismo procedimiento con la tabla Cliente.

U-PC.Banco - dbo.Cliente						
	NumeroIdentifi...	Nombre	Apellido	FechaNacimiento	Edad	CodigoSucursal
	7012089	Juan	Perez	2010-01-01 00:..	10	1
*	NULL	NULL	NULL	NULL	NULL	NULL

22. Ingresar 2 registros para Banco, 3 registros para Sucursal y 5 registros para Clientes.

23. Finaliza la creación de la base de datos y la información esta almacenada.

Tipos de Datos SQL SERVER 2008

Tipo Dato	Descripción	Longitud	Ejemplo
Int	Número entero positivos o negativos.	Desde – 2.147.483.648 a 2.147.483.647 (requiere de 4 bytes por valor).	1, -10, 200, 5000
numeric(longitud entera, longitud decimales)	Número decimal con una longitud de enteros y una longitud de decimales	Los datos pueden tener un máximo de 30 dígitos, que pueden estar todos a la derecha de la coma decimal. El tipo de dato almacena una representación exacta del número.	Numeric(10,2): 12345678.12 Numeric(5,1):9874,2 Numeric(18,5): 1234567891123.12
decimal(longitud entera, longitud decimales)	Número decimal con una longitud de enteros y una longitud de decimales	Los datos pueden tener un máximo de 30 dígitos, que pueden estar todos a la derecha de la coma decimal. El tipo de dato almacena una representación	decimal(10,2): 12345678.12 decimal (5,1):9874,2 decimal (18,5): 1234567891123.12

Introducción SQL SERVER –John Cañaveral

		exacta del número.	
char	Los datos deben tener la misma longitud fija	La definida por el usuario	char(2) = 'la' char(1) = 'a' char(4)='hola'
varchar(longitud)	Los datos pueden variar en el número de caracteres	La definida por el usuario	varchar(5)="hola" varchar(5)="hola1" varchar(3)="hol" varchar(3)="a"
nvarchar(longitud)	Los datos pueden variar en el número de caracteres (hasta 4000 caracteres Unicode)	La definida por el usuario	nvarchar(5)="hola" nvarchar(5)="hola1" nvarchar(3)="hol" nVarchar(3)="a"
nchar(longitud)	Los datos deben tener la misma longitud fija (hasta 4000 caracteres Unicode)	La definida por el usuario	nchar(2) = 'la' nchar(1) = 'a' nchar(4)='hola'
Datetime	Los datos fecha están comprendidos entre el 1 de Enero de 1753 hasta el 31 de diciembre de 9999	Almacena: año, mes, día, hora, minuto, segundo, milisegundo	2011-09-20 10:58:06.207
SmallDatetime	Los datos fecha están comprendidos entre en el 1 de Enero de 1900 hasta el 31 de diciembre de 2079	Almacena: año, mes, día, hora, minuto, segundo,	2011-09-20 10:59:00
bit	Consisten un 1 o un 0. Se usan para representar valores lógicos VERDADERO o FALSO, SI o NO	True o False	True False
Smallint	Desde -32,768 a 32.767	Número entero mas pequeño que int	32767
Float	Datos numéricos aproximados que consisten de	Desde -1.79E + 308 a 1.79E + 308	1057844

Introducción SQL SERVER –John Cañaveral

	datos con una aproximación tanto como el sistema de numeración binaria pueda ofrecer		
Ntext	Cualquier campo Unicode	os datos pueden exceder los 4000 caracteres Unicode	"hola - * como estan"