

# El SQL Server Management Studio


## 1. Introducción

SQL Server Management Studio es la herramienta gráfica de SQL Server que permite definir y gestionar todas las bases de datos almacenadas en el servidor SQL Server.

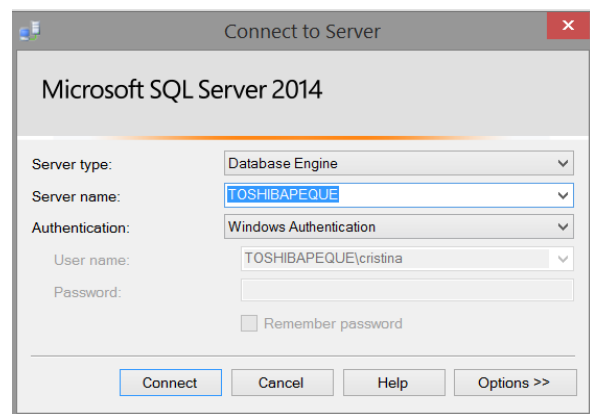
En este tema veremos cómo utilizar SQL Server Management Studio para manejar las bases de datos del servidor.

Nota. Para ilustrar las explicaciones se han incluido imágenes de la versión SQL Server 2005 en español y de la versión SQL Server 2014 en inglés.

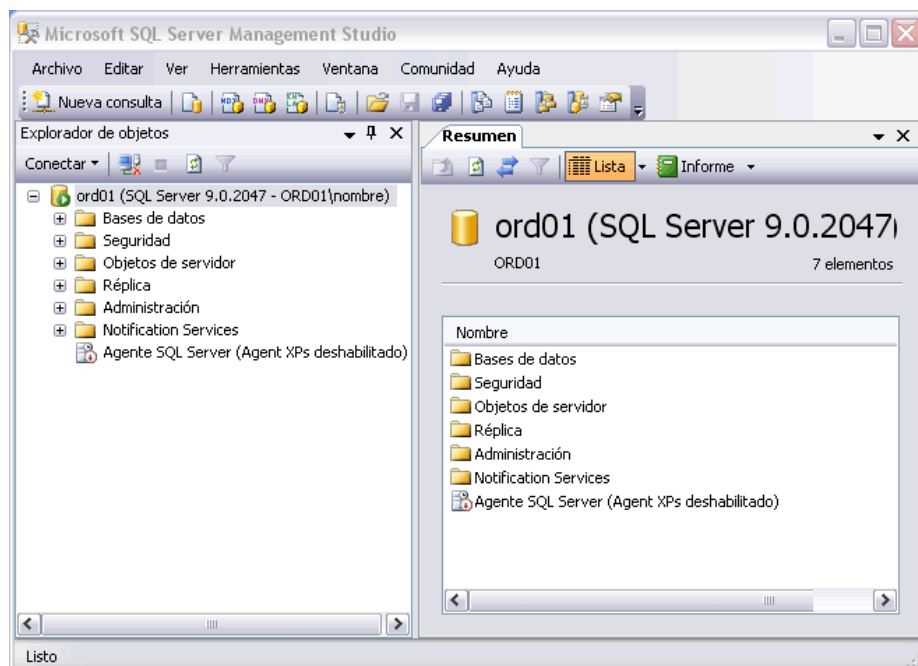
## 2. Entrada al SQL Server Management Studio.

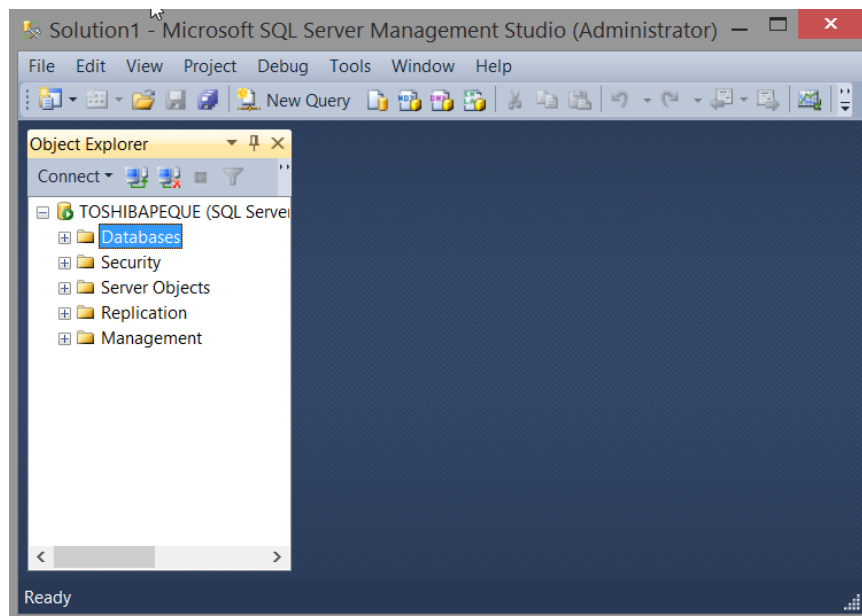
Aunque trabajemos en modo local, la entrada a la herramienta es la misma. Para empezar entramos a través del acceso directo . Ojo que para no tener problemas tienes que arrancar el programa en modo administrador (si no has cambiado las propiedades del archivo para que sea el modo por defecto, haz clic con el botón secundario del ratón y selecciona la opción *Abrir como administrador*).

Lo primero que deberemos hacer es establecer la conexión con el servidor:



Seleccionamos el nombre del servidor y pulsamos el botón *Conectar*. Se abrirá la ventana inicial del SQL Server Management Studio (en adelante SSMS):





En la parte izquierda tenemos abierto el panel Explorador de Objetos *OBJECT EXPLORER* en el que aparece debajo del nombre del servidor con el que estamos conectados una serie de carpetas *FOLDERS* y objetos que forman parte del servidor.

En el panel de la derecha se muestra la zona de trabajo, que varía según lo que tengamos seleccionado en el Explorador de objetos o según la versión de SSMS que utilices, en la primera imagen vemos el contenido de la carpeta que representa el servidor ord01.

En la parte superior tenemos el menú de opciones y la barra de herramientas Estándar.



Con las siguientes opciones (pueden cambiar de una versión a otra):

Nueva consulta/New Query	Consulta de motor de Base de datos
Consulta MDX de Analysis Services	Consulta DMX de Analysis Services
Consulta MXLA de Analysis Services	Consulta de SQL Server Mobile
Abrir archivo	Guardar
Guardar todo	Servidores registrados
Resumen	Explorador de Objetos
Explorador de Plantillas	Ventana de Propiedades

### 3. Las bases de datos

#### 3.1. Estructura interna de una base de datos

Antes de empezar tenemos que tener claro cómo se organiza la información en una base de datos SQL Server .

Las bases de datos de SQL Server utilizan tres tipos de archivos:

- Archivos de datos principales

En una base de datos SQLServer los datos se pueden repartir en varios archivos para mejorar el rendimiento de la base de datos.

El archivo de datos principal *PRIMARY FILE* es el punto de partida de la base de datos y apunta a los otros archivos de datos de la base de datos. Cada base de datos tiene obligatoriamente un archivo de datos principal. La extensión recomendada para los nombres de archivos de datos principales es *.mdf*.

- Archivos de datos secundarios

Los archivos de datos secundarios son todos los archivos de datos menos el archivo de datos principal. Puede que algunas bases de datos no tengan archivos de datos secundarios, mientras que otras pueden tener varios archivos de datos secundarios. La extensión de nombre de archivo recomendada para los archivos de datos secundarios es .ndf.

Además los archivos de datos se pueden agrupar en grupos de archivos *FILE GROUPS*. Para cada base de datos pueden especificarse hasta 32.767 archivos y 32.767 grupos de archivos.

- Archivos de registro

Los archivos de registro (archivos de log) *LOG FILE* almacenan toda la información de registro que se utiliza para recuperar la base de datos, el también denominado registro de transacciones. Como mínimo, tiene que haber un archivo de registro por cada base de datos, aunque puede haber varios. La extensión recomendada para los nombres de archivos de registro es .ldf.

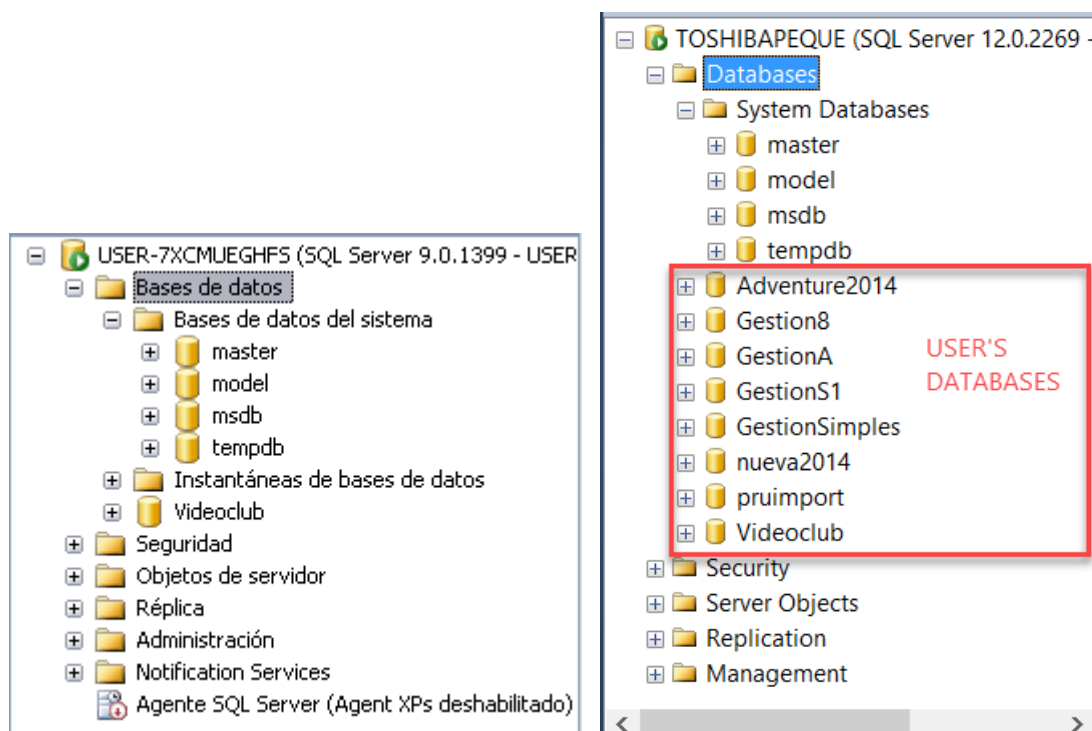
SQL Server no exige (sólo recomienda) las extensiones de nombre de archivo .mdf, .ndf y .ldf, pero estas extensiones ayudan a identificar las distintas clases de archivos y su uso.

Cada base de datos tiene al menos 2 archivos (un archivo de datos principal y un archivo de registro) y opcionalmente grupos de archivos.

Los archivos de datos y de registro de SQL Server se pueden colocar en sistemas de archivos FAT o NTFS. Se recomienda utilizar NTFS por los aspectos de seguridad que ofrece. No se pueden colocar grupos de archivos de datos de lectura y escritura, y archivos de registro, en un sistema de archivos NTFS comprimido. Sólo las bases de datos de sólo lectura y los grupos de archivos secundarios de sólo lectura se pueden colocar en un sistema de archivos NTFS comprimido.

### 3.2. Crear una base de datos en SSMS

En el Explorador de objetos, si desplegamos la carpeta Bases de datos nos aparecen Bases de datos del sistema, después las bases de datos **Instantáneas...** (una base de datos instantánea *SNAPSHOT* es una vista estática de sólo lectura *READ-ONLY STATIC VIEW* de una base de datos denominada base de datos de origen *SOURCE DATABASE*) y finalmente las bases de datos de usuario (en la imagen Videoclub es una base de datos de usuario).

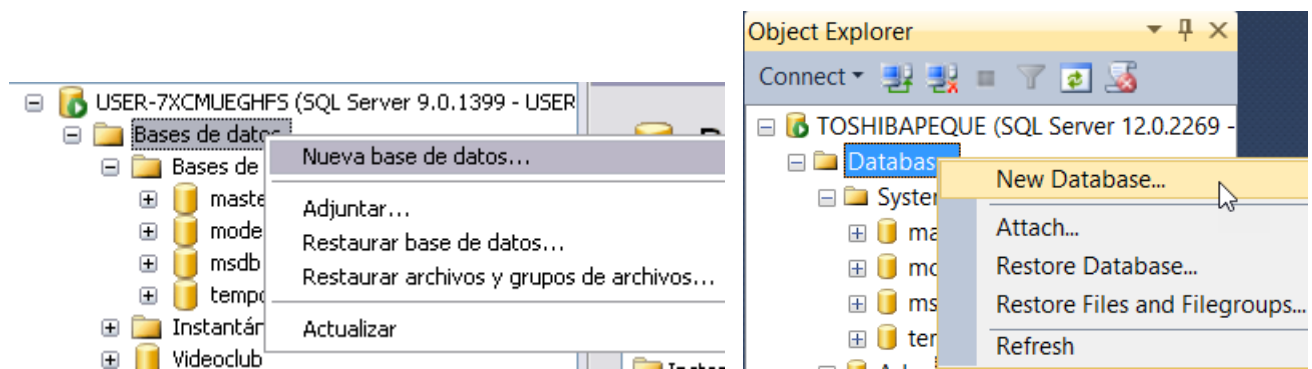


Después de la instalación, en la carpeta Bases de datos del sistema se habrá creado una especial denominada **master** se utiliza como base de datos de usuario por defecto.

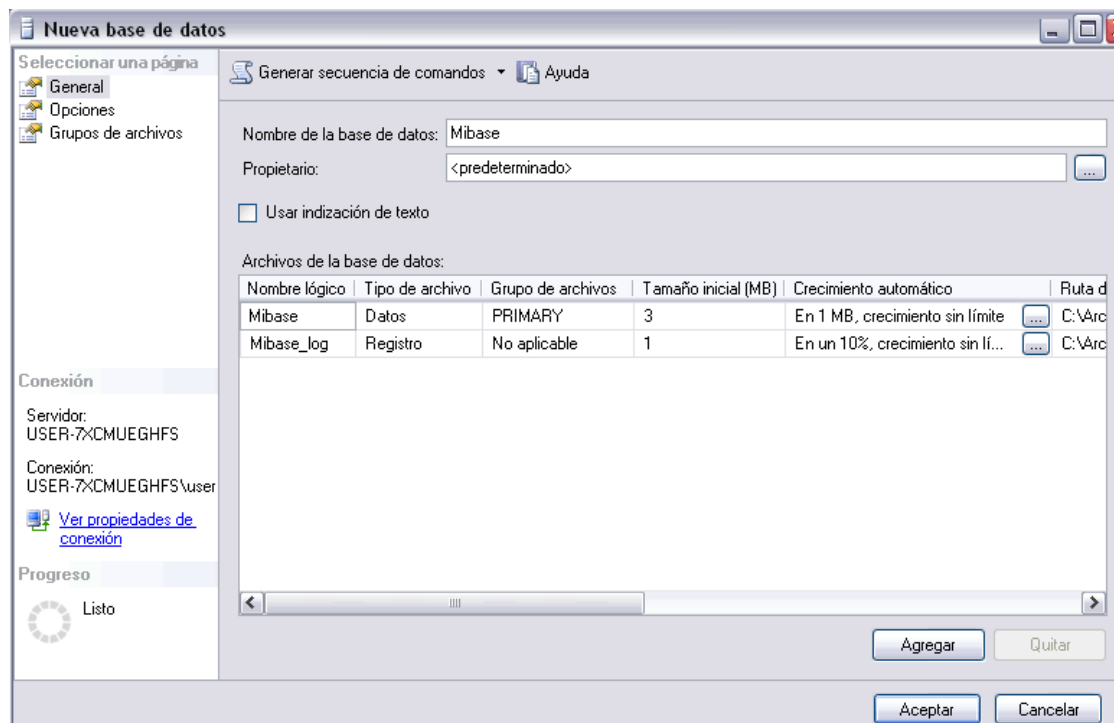
Las demás bases de datos del sistema forman también parte del diccionario de datos y las utiliza el sistema para llevar a cabo su gestión.

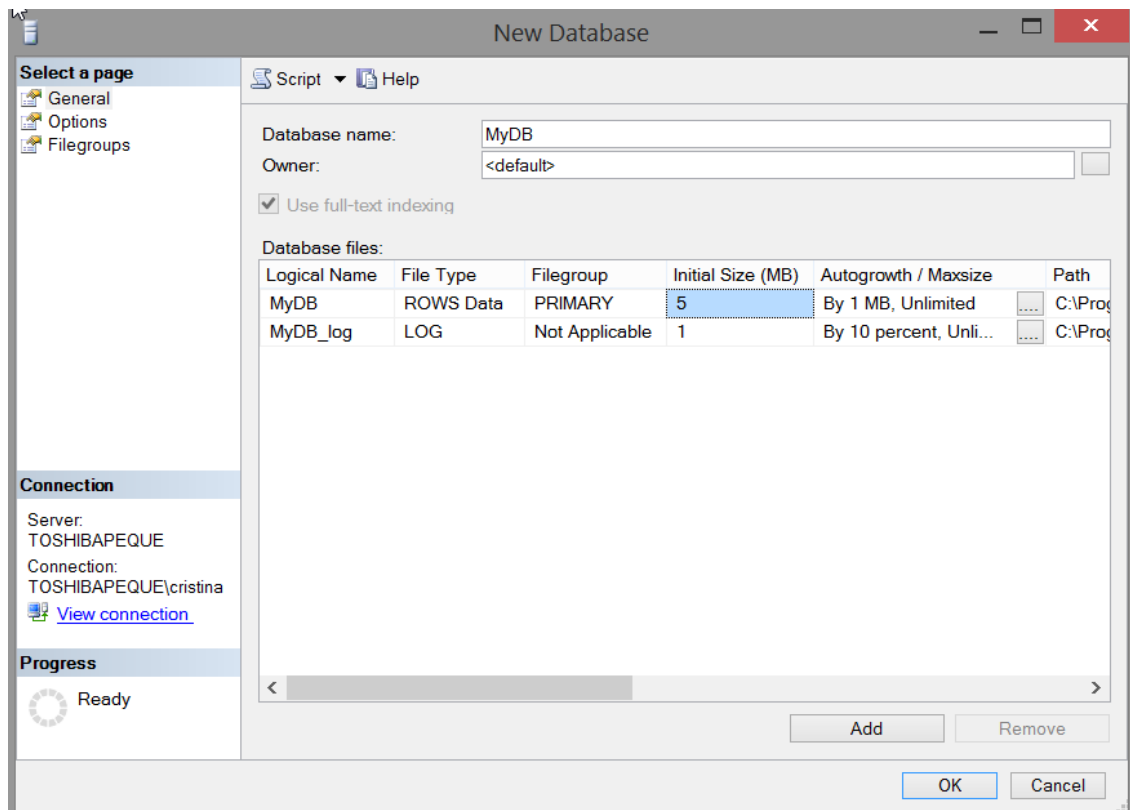
Base de datos	Descripción
<b>master</b>	Controla las bases de datos de usuario y el funcionamiento de SQL Server de forma global al realizar un seguimiento de información como las cuentas de usuario, las variables de entorno configurables y los mensajes de error del sistema.
<b>model</b>	Proporciona una plantilla o un prototipo para las nuevas bases de datos de usuario.
<b>tempdb</b>	Proporciona un área de almacenamiento para tablas temporales y otras necesidades de almacenamiento temporal.
<b>msdb</b>	Ofrece un área de almacenamiento para información de programación e historial de trabajos.

Las bases de datos de los usuarios se deben crear preferentemente fuera de la carpeta Bases de datos del sistema. Para crear una nueva base de datos de usuario nos posicionamos sobre la carpeta Bases de datos y con el botón secundario del ratón desplegamos el menú contextual del que elegimos la opción **Nueva base de datos...**



Se abre a continuación el cuadro de diálogo donde definiremos la base de datos que queremos crear:





Lo mínimo a introducir será el campo **Nombre de la base de datos**, éste es el nombre de la base de datos lógica *LOGICAL NAME*, la base de datos a la que nos referiremos dentro del SSMS, a nivel conceptual (en la imagen Mibase), es el nombre que aparece luego en el Explorador de objetos.

Esta base de datos está asociada a como mínimo dos archivos físicos, en la parte inferior aparecen esos archivos. Para facilitarnos la tarea, al teclear el nombre lógico, se rellenan automáticamente los nombres de los archivos físicos, el de datos con el mismo nombre y el del archivo de registro con el mismo nombre seguido de `_log`. Estos nombres son los nombres que se asumen por defecto pero los podemos cambiar, posicionando el cursor en el nombre y cambiándolo. Nosotros los cambiaremos aquí para distinguirlos mejor.

Para cada archivo físico podemos definir una serie de parámetros como el **tipo de archivo** (si es de datos o de Registro *LOG*) y su ocupación inicial (**Tamaño inicial**).

Si no indicamos ninguna ubicación podemos ver que los guarda en la carpeta del SQL Server/.../MSSQL/DATA. La ruta exacta varía de una versión a otra pero es muy similar.

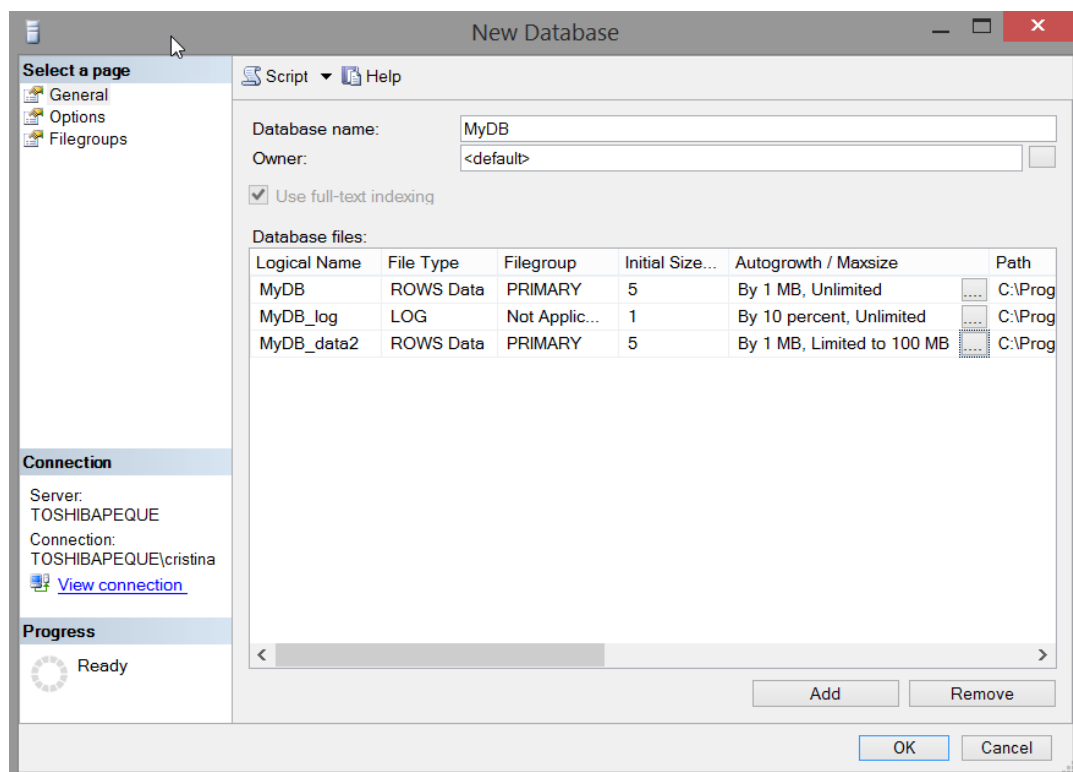
La ruta completa de los archivos físicos la tenemos en la columna **Ruta de acceso/Path**.

Se pueden definir otros parámetros que de momento no comentaremos pero que volveremos a ver y explicaremos cuando estudiemos la instrucción SQL para crear una bd.

Estos son los archivos mínimos en los que se almacenará la base de datos, pero como ya vimos anteriormente se puede almacenar en más archivos, en este caso los tenemos que definir todos en esta ventana, a continuación de los dos obligatorios.

Para añadir más archivos físicos disponemos del botón **Agregar**.

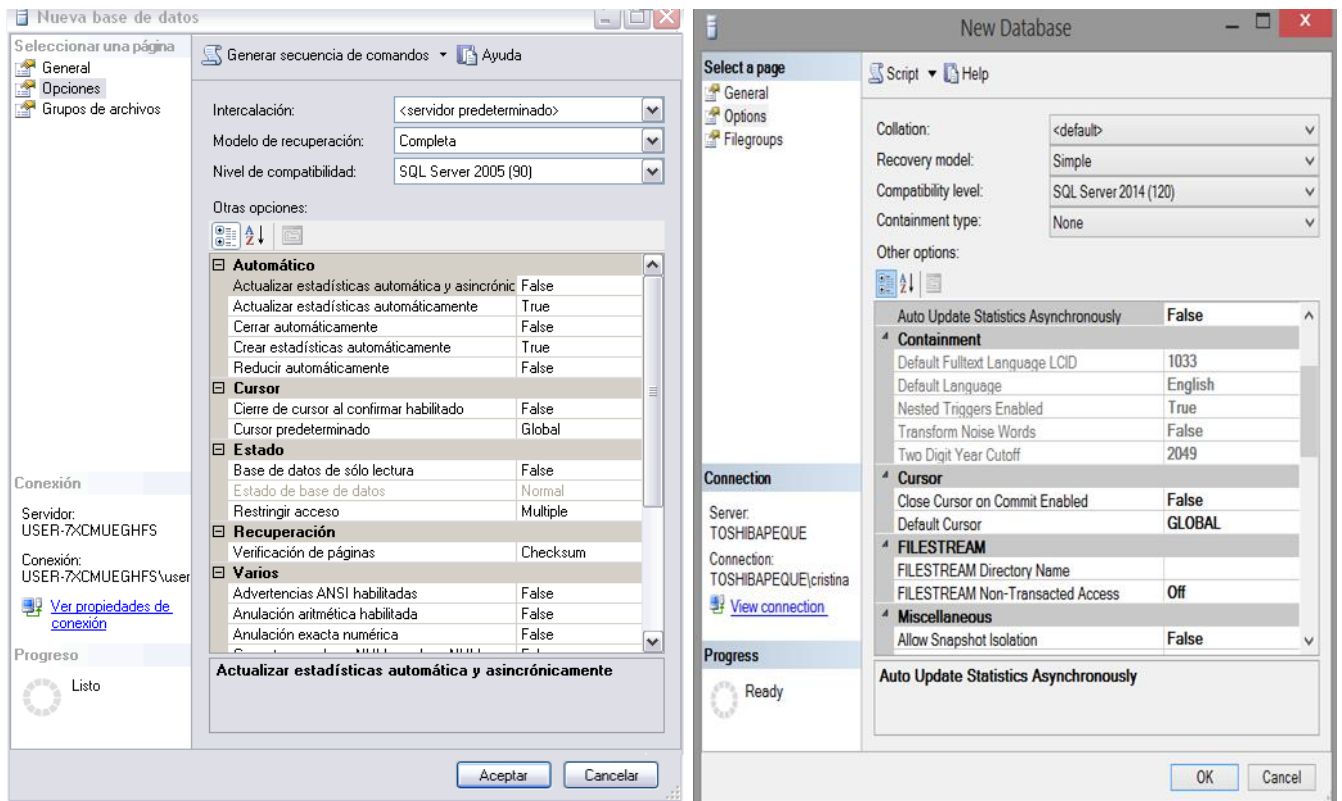
Al pulsar el botón **Agregar** se crea una nueva fila en la tabla de archivos físicos donde deberemos escribir el nombre del archivo, su tipo (desplegando la lista podemos elegir entre de Datos/ROWS data o de registro/LOG) y demás parámetros.



Al agregar un nuevo archivo se activa el botón **Quitar/Remove** siempre que estemos posicionados encima de un archivo secundario para poder así eliminarlo si lo queremos.

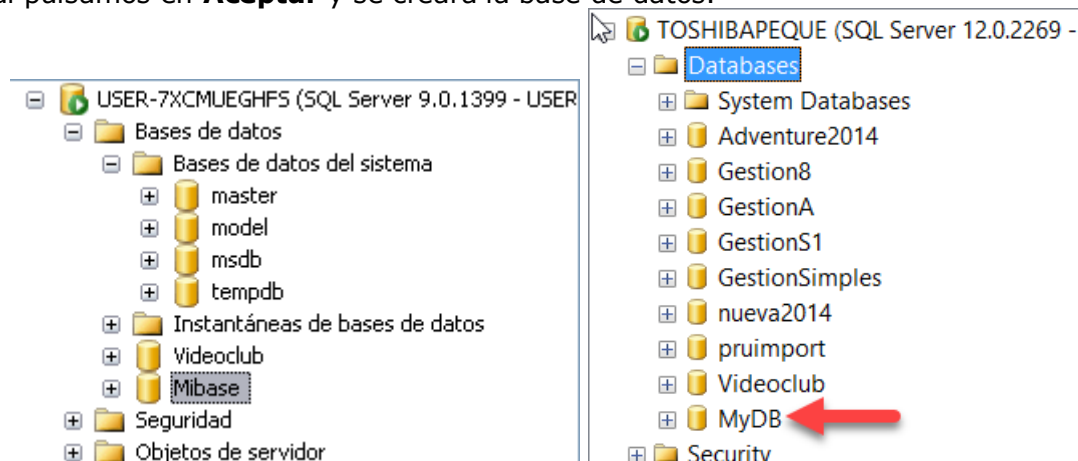
No podemos eliminar ni el de datos primario, ni el de registro inicial.

Si nos fijamos en la zona de la izquierda, vemos que nos encontramos en la pestaña **General**, podemos cambiar otros parámetros de la base de datos pulsando en **Grupos de archivos** o en **Opciones** :



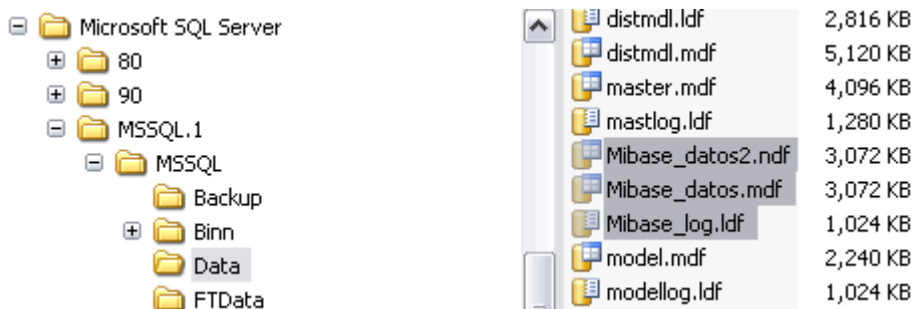
De momento no nos extenderemos sobre estas opciones.

Al final pulsamos en **Aceptar** y se creará la base de datos.

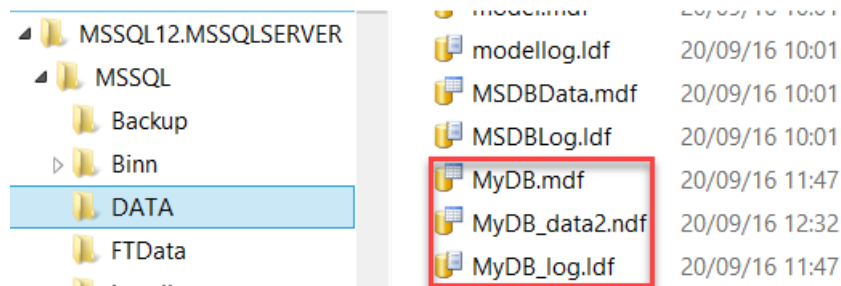


Aparecerá en la carpeta **Bases de datos**, si no se ve pulsa en el icono Actualizar

Desde el Explorador de **Windows** podemos ver que en la carpeta indicada se han creado los archivos físicos con los nombres que le hemos indicado.





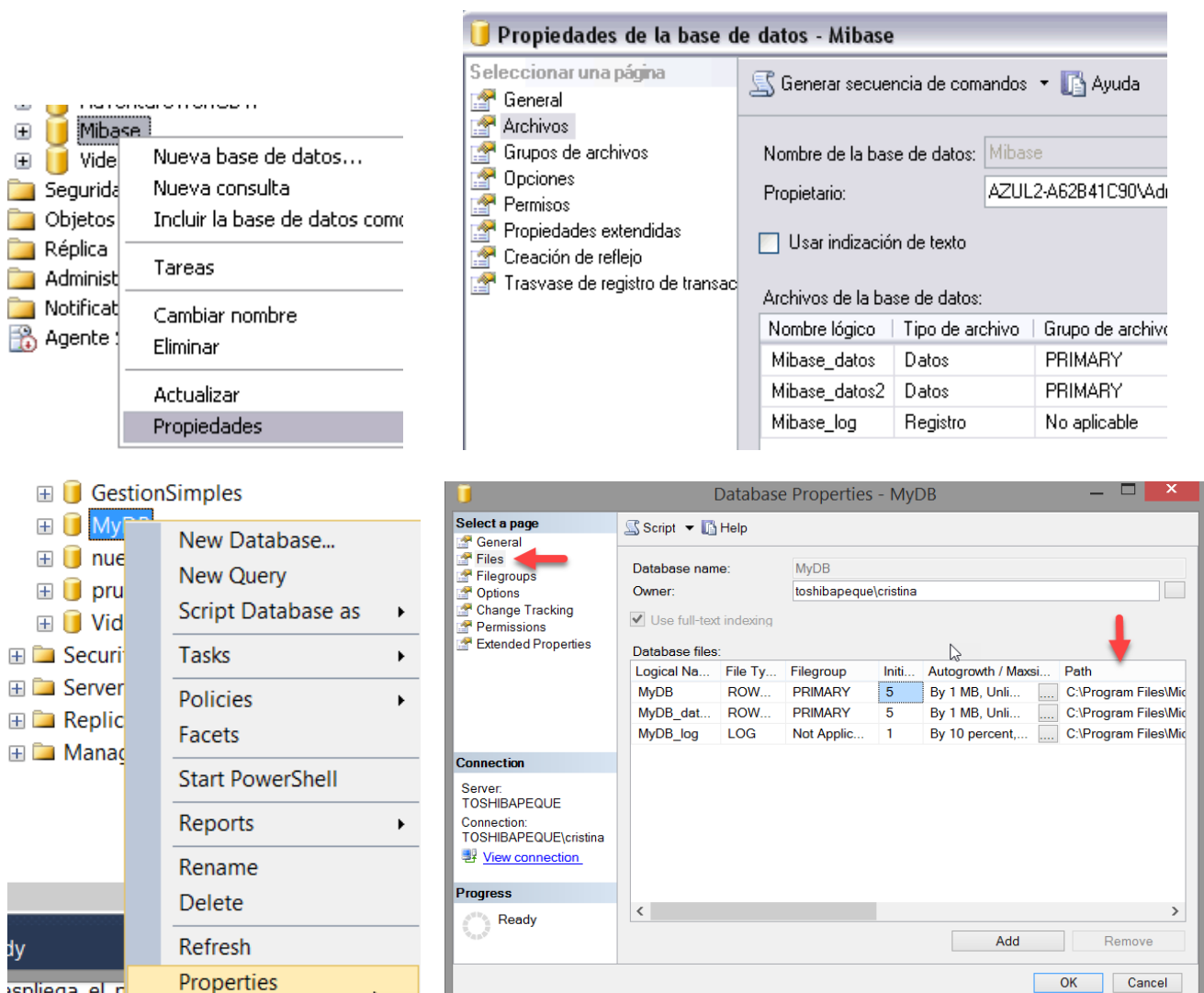


## 4. Utilizar una base de datos en varios servidores

En este punto veremos una manera sencilla de poder llevar una base de datos de un ordenador a otro, esta no es la forma más correcta de hacerlo (existen otras formas como importar, exportar bases de datos y realizar copias de seguridad) pero es la más sencilla y para empezar puede servir.

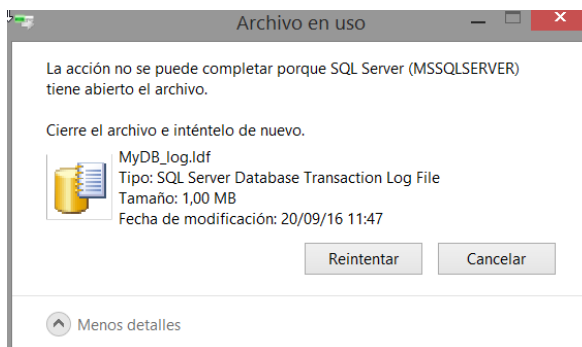
Vamos a suponer que tenemos una base de datos en el servidor de clase y queremos llevarnos la base de datos a nuestra casa para seguir trabajando con ella. Sabemos que la base de datos está guardada en los archivos físicos que definimos cuando definimos la base de datos, por lo tanto podríamos copiar estos archivos de un lado a otro. Veamos cómo hacerlo.

Estos archivos se localizan en una determinada ruta, la que indicamos al crear la base de datos. Esta información se encuentra en las propiedades de la base de datos, para acceder a la ventana de propiedades de la base de datos (parecida a la que se abrió cuando creamos la base de datos nueva) utilizamos como siempre el menú contextual de la base de datos en cuestión que nos llevará a la ventana Propiedades de la base de datos:



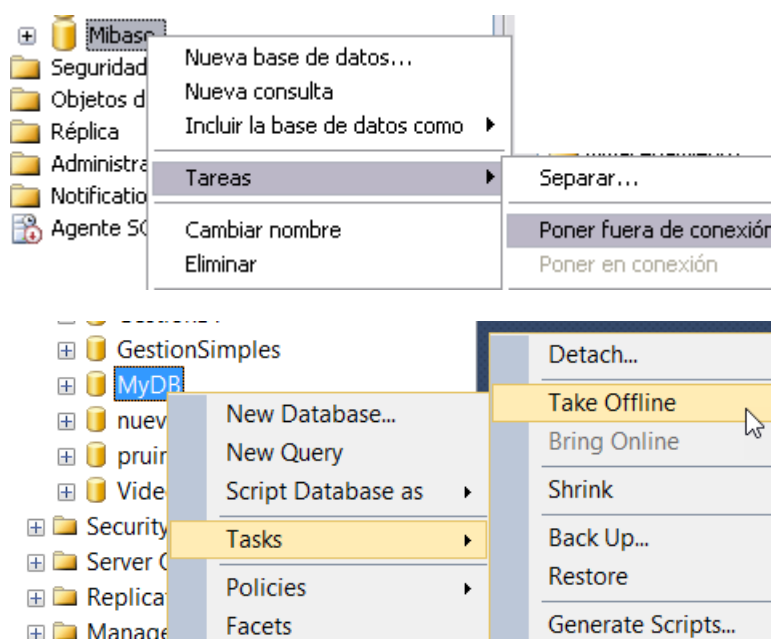


Si intentamos desde fuera del SSMS, por ejemplo desde el Explorador de Windows, copiar estos archivos a otro soporte, puede que no sea posible directamente, Windows nos advierte de que los archivos están en uso y por lo tanto no se pueden copiar.



En tal caso debemos desconectar la base de datos del servidor del siguiente modo:

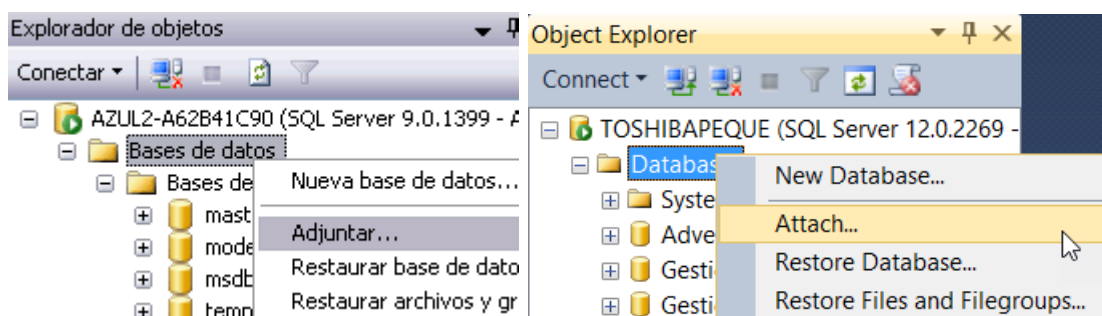
En SSMS despliega el menú contextual de la base de datos a copiar y selecciona la opción **Poner fuera de conexión**:



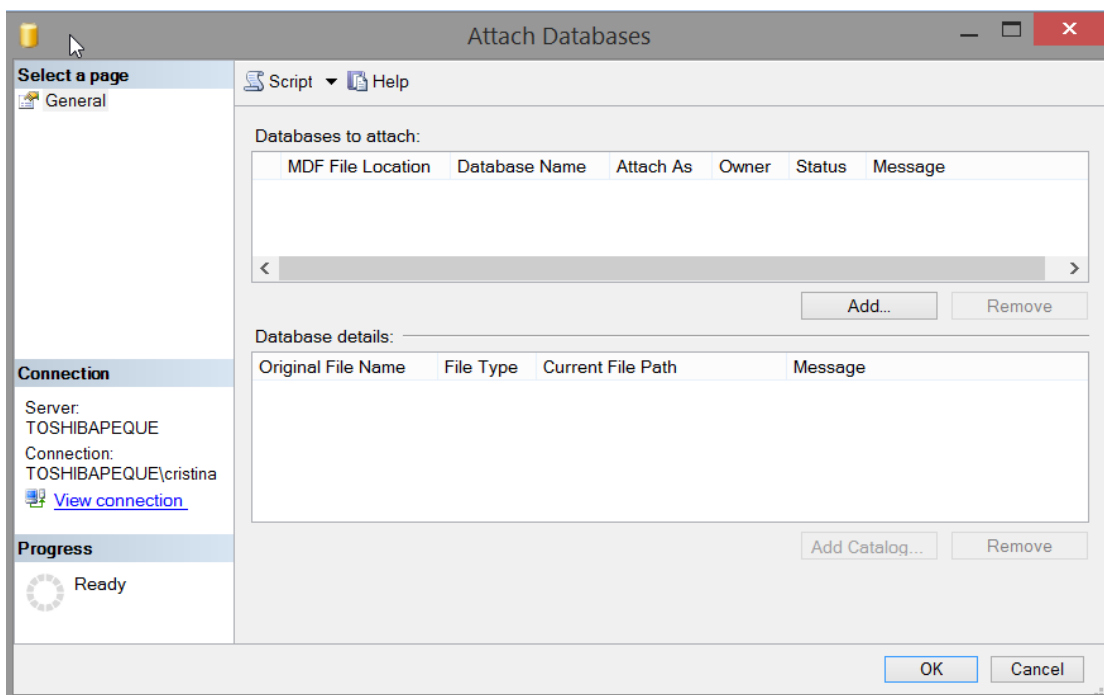
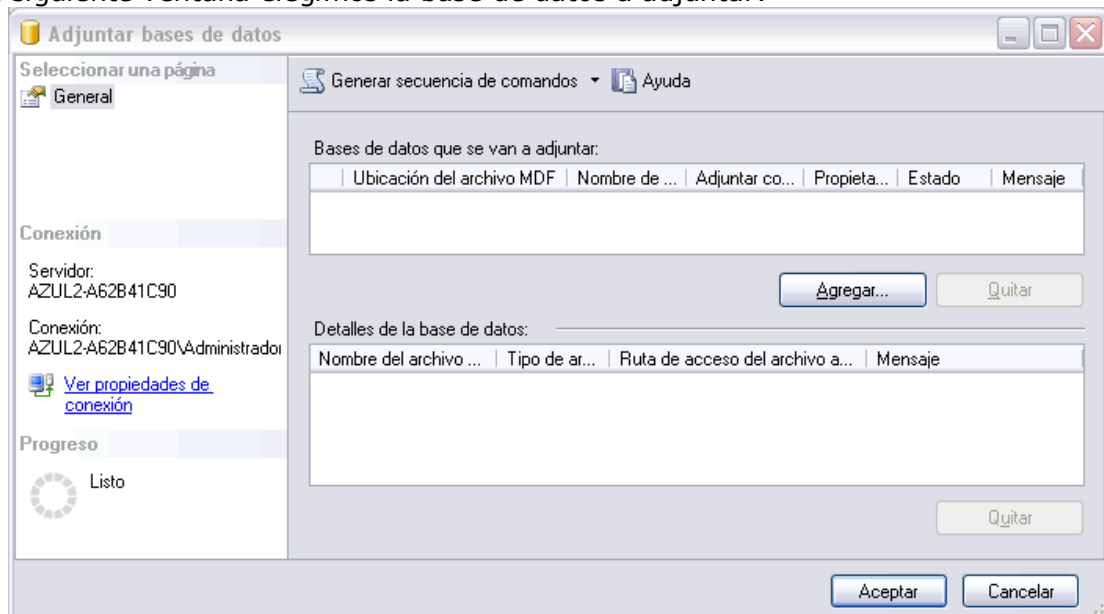
Aparecerá un símbolo a la izquierda de la base de datos indicándonos que la base de datos está desconectada, a partir de este momento Windows nos dejará copiar los archivos.

Para poder utilizar la base de datos en nuestro servidor de casa, lo que tenemos que hacer es pegar los archivos en la ubicación que queramos, y luego indicar al SQL Server que vamos a utilizar esta base de datos, de la siguiente manera:

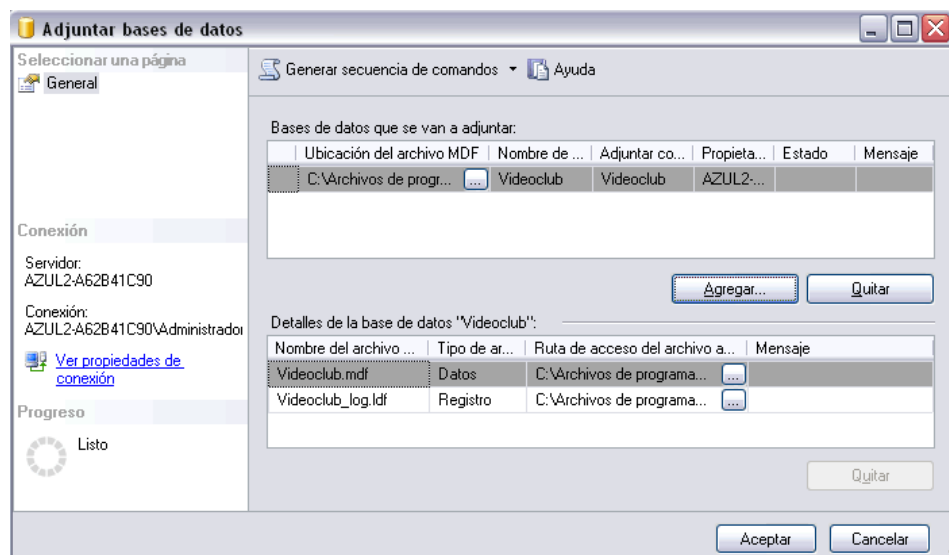
En el Explorador de objetos, sobre la carpeta Bases de datos despliega el menú contextual y elije **Adjuntar...**



En la siguiente ventana elegimos la base de datos a adjuntar:



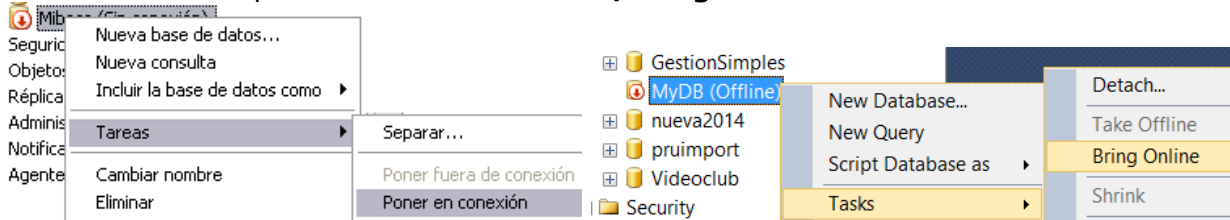
Pulsando en **Agregar** indicamos el archivo de datos primario en su ubicación y automáticamente se adjuntará la base de datos lógica asociada a este archivo.



Finalmente pulsamos en **Aceptar** y aparece la base de datos en nuestro servidor.

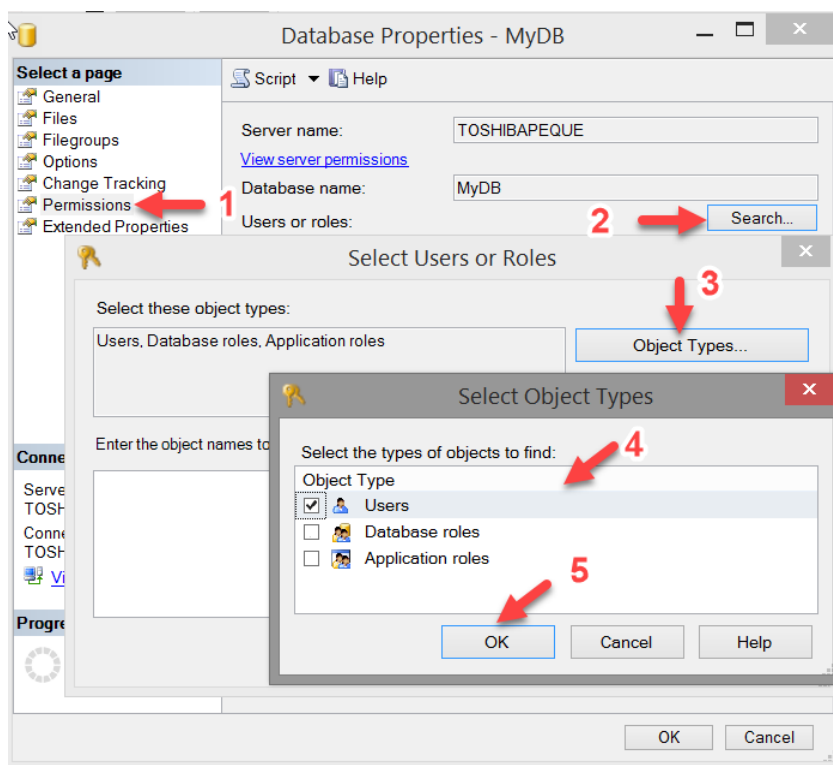
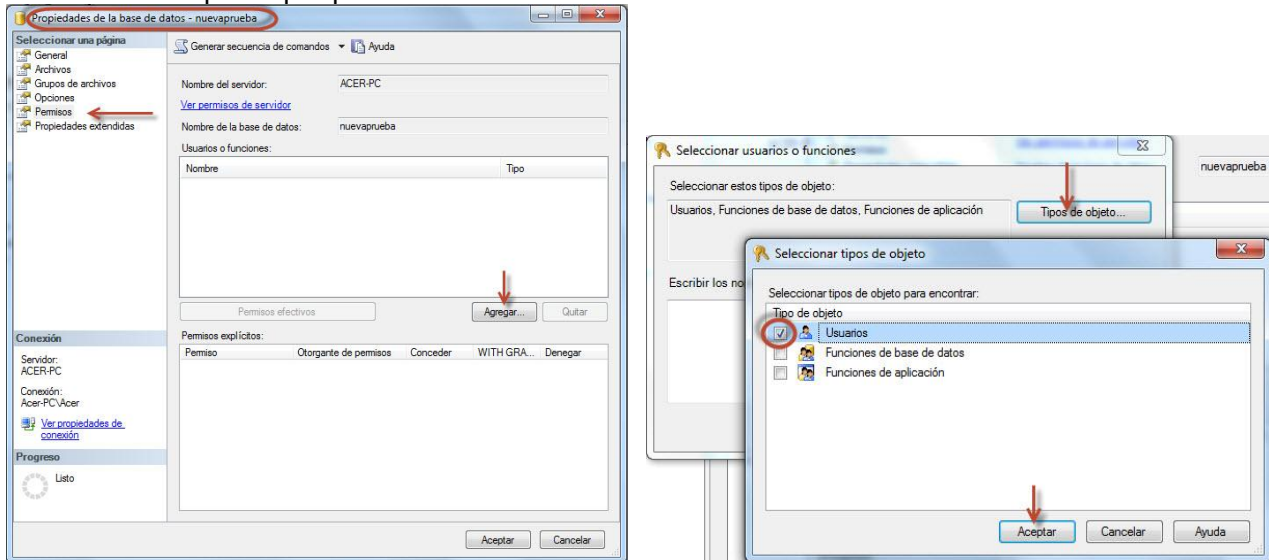
La opción **Adjuntar** sólo se utiliza la primera vez, cuando todavía no tenemos la base de datos registrada en el servidor de casa.

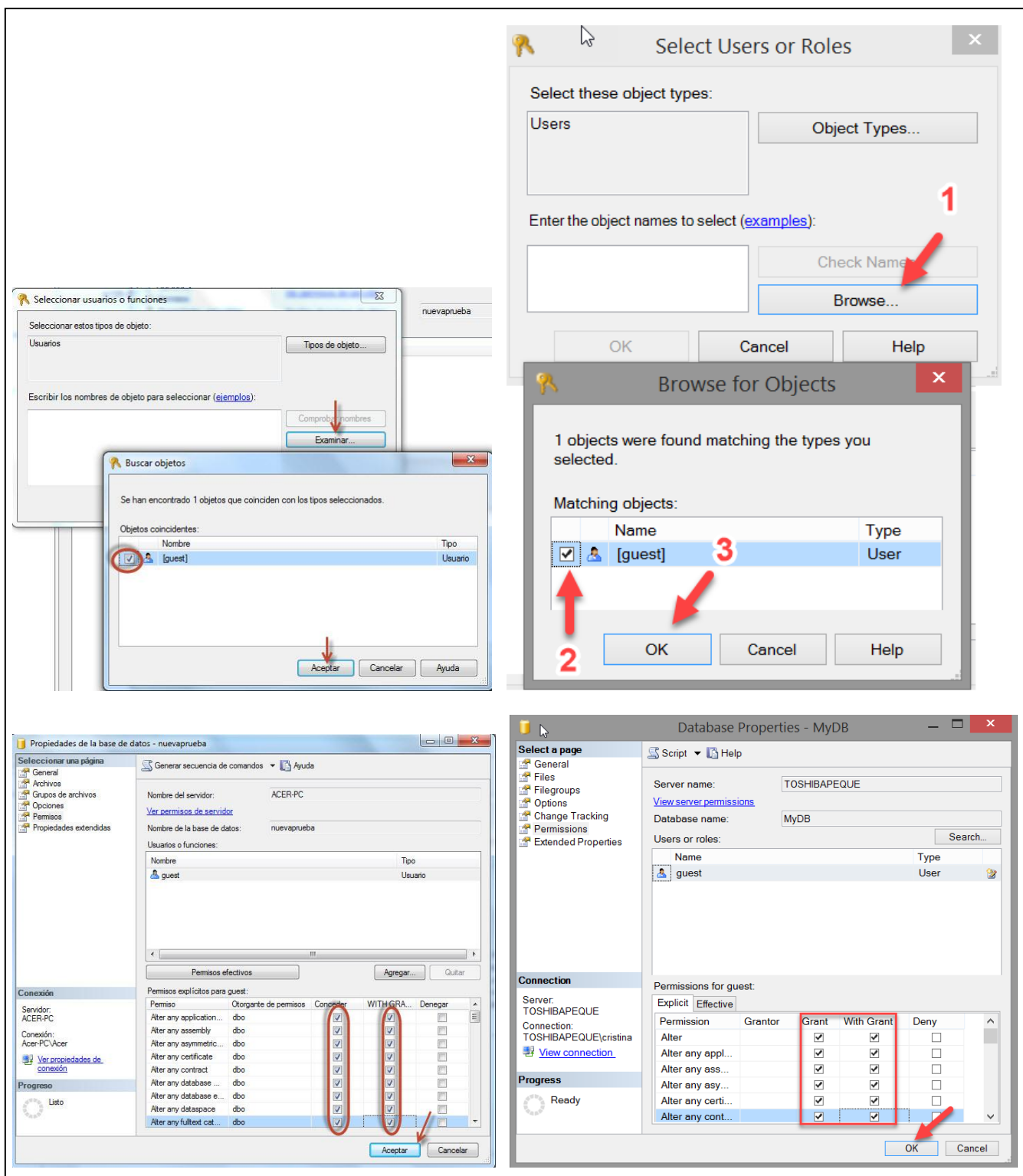
Nota. Si en nuestro servidor de casa ya teníamos la base de datos definida y conectada al servidor, no nos dejará pegar los archivos sustituyendo los existentes en nuestra máquina, primero tenemos que poner la base de datos fuera de conexión para poder pegarlos, y después de pegarlos volvemos a conectar la base de datos con el servidor, de la misma manera pero eligiendo esta vez la opción **Poner en conexión/Bring Online**



Nota. Un problema que puede ocurrir es que no tengamos en el otro servidor los permisos suficientes para poder trabajar con la base de datos adjuntada, en este caso puedes seguir estos pasos:

En el servidor donde se creó la base de datos (propietario), cambiar las propiedades de la base de datos para que permita a otros usuarios modificarla:





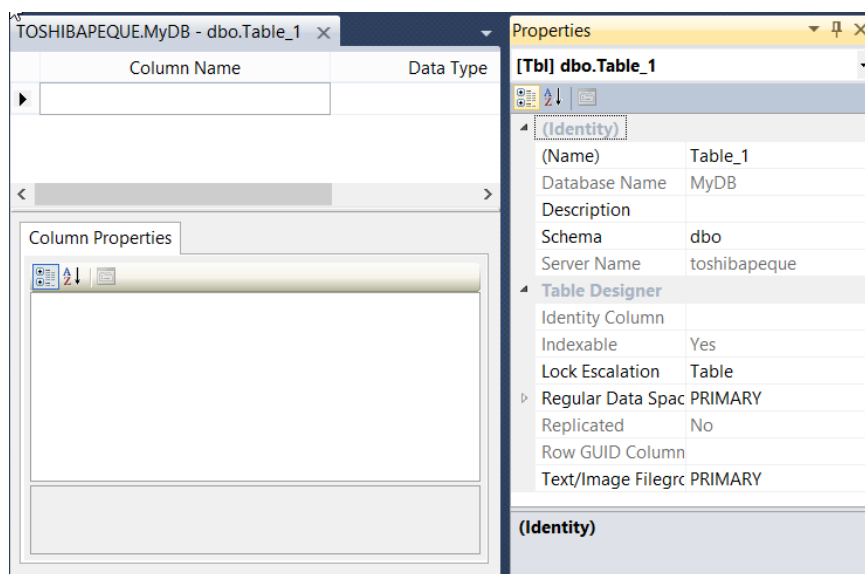
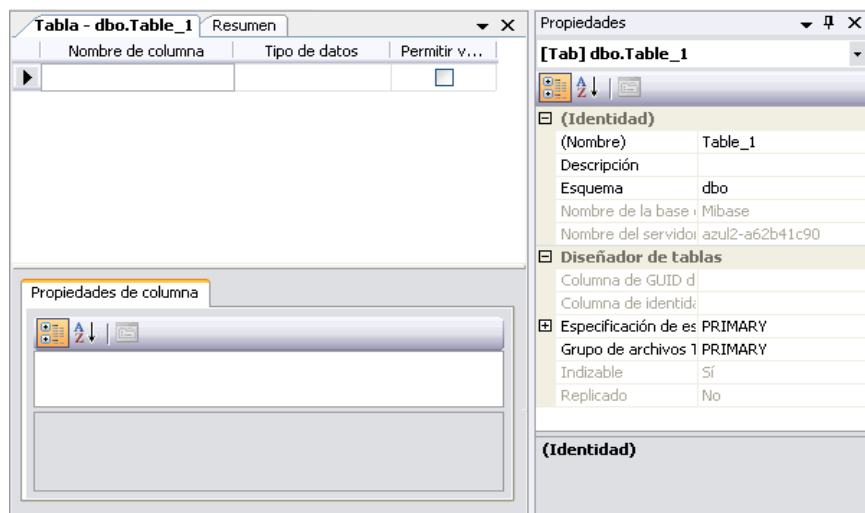
## 5. Las tablas

### 5.1. Crear una nueva tabla

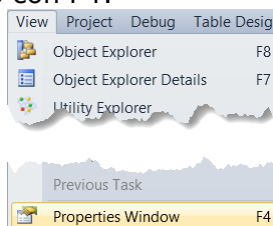
Para crear una nueva tabla primero nos tenemos que posicionar en la base de datos donde queremos que se almacene la tabla, desplegar el menú contextual y seleccionar la opción **Nueva tabla**



En la ventana que se abre debemos definir las columnas de la tabla:



Si el panel Propiedades no aparece lo puedes visualizar con la opción *Propiedades* del menú *Ver* o con F4:

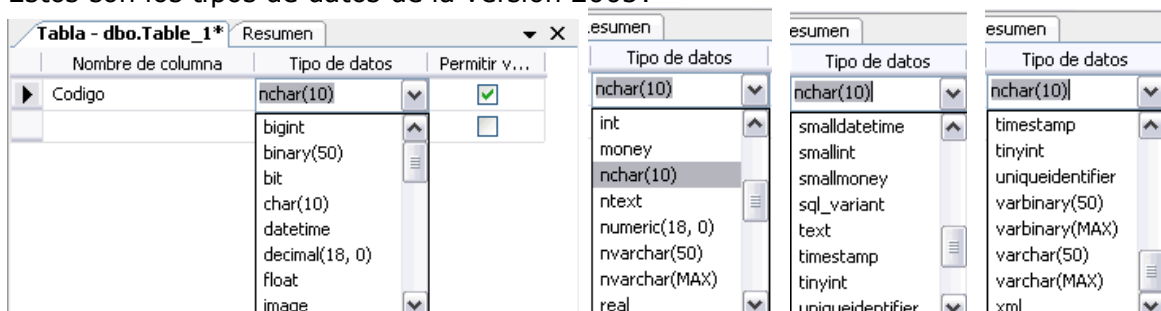


A cada columna se le asigna un nombre, un tipo de datos, y opcionalmente una serie de propiedades, en este tema veremos las básicas y las demás las veremos con más detalle cuando veamos la instrucción SQL CREATE TABLE.

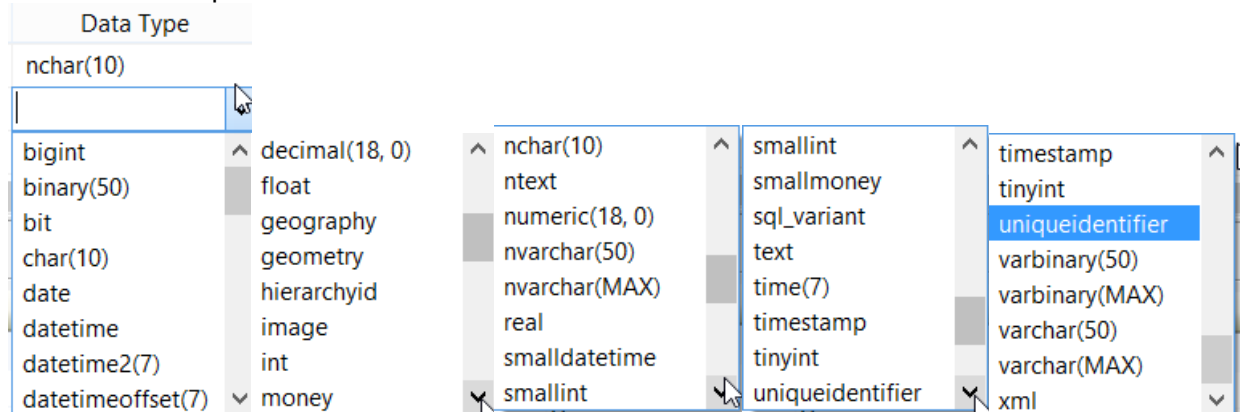
De momento no tenemos definida ninguna columna, al teclear un nombre se crea una primera entrada en esta tabla con la definición de la primera columna. En la columna Tipo de datos elegimos qué tipo de valores se podrán almacenar en la columna.

## 5.2. Tipos de datos

Estos son los tipos de datos de la versión 2005:



Estos son los tipos de la versión 2014:



Podemos elegir entre todos los tipos que aparecen arriba, aquí citaremos los más útiles. Los tipos enteros *INTEGER* sirven para almacenar números enteros (sin decimales) positivos y negativos más o menos grandes/pequeños, estos son:

- **bigint** (*BIG INTEGER* entero largo de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807)
- **int** (entero para valores desde -2.147.483.648 a 2.147.483.647)
- **smallint** (*SMALL INTEGER* entero corto para valores desde -32.768 a 32.767)
- **tinyint** (*TINY INTEGER* almacena un byte/ocho bits para valores desde 0 a 255)

El tipo **bit** almacena un bit (0 o 1)

**Float** y **real** para números almacenados en coma flotante.

**Decimal** y **numeric** para números almacenados en coma fija.

**Money** para números que se refieran a cantidades monetarias.

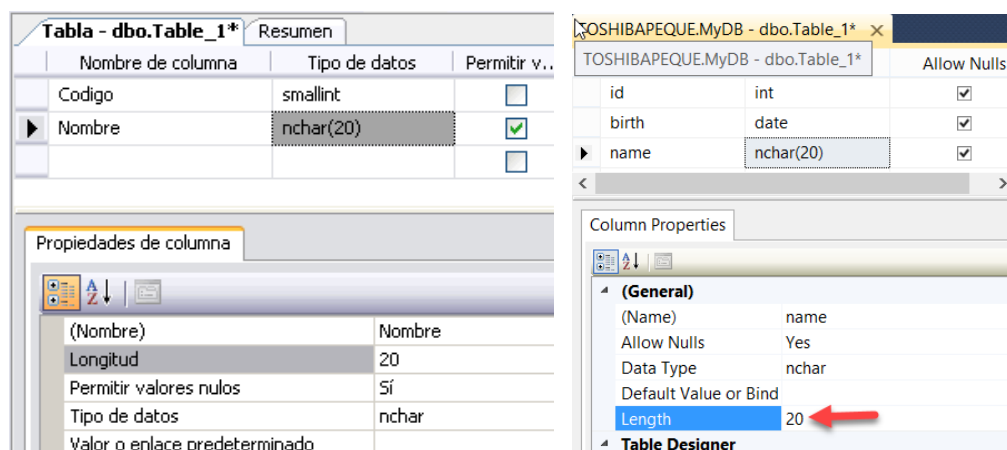
**Date**, **Datetime** son para fechas sin hora y con hora.

**Char** (*CHARACTER*), **nchar** para valores alfanuméricos (carácter) de longitud fija.

**Varchar**, **nvarchar** para valores alfanuméricos de longitud **variable**.

**Geography** para coordenadas geográficas.

Algunos tipos no necesitan más, como por ejemplo el tipo entero (**int**), y otros se pueden completar con una **longitud**, aparecen en la lista con números entre paréntesis). Por ejemplo:



En este ejemplo hemos definido una columna (**Codigo**) de tipo Entero corto *SMALLINT*, y una columna (**Nombre**) que almacenará hasta 20 caracteres alfanuméricos (**nchar(20)**), en este caso la longitud la indicamos en la pestaña **Propiedades de columna** en la propiedad **Longitud**.

Las propiedades de la columna pueden variar dependiendo del tipo de datos de la columna seleccionada, por ejemplo los campos enteros no tienen la propiedad **longitud**, ya que el propio tipo define la longitud del campo, en cambio los campos de tipo **numeric** o **decimal** no tienen la propiedad **longitud** pero sí las propiedades **escala** y **precisión**, los valores que permiten definir el tamaño del campo, la **precisión** es el número total de dígitos numéricos y **escala** el nº de decimales.



### 5.3. Valores nulos

También podemos indicar si la columna permitirá valores nulos o no, cambiando la propiedad **Permitir valores nulos/Allow Nulls** que aparece en las propiedades de la columna, o bien simplemente marcando o desmarcando la casilla de la columna **Permitir valores nulos/Allow Nulls** que se encuentra al lado de la columna **Tipo de datos/ Data Type**. Si la casilla está marcada, el usuario podrá no rellenar el campo cuando inserte una fila de datos en la tabla.

### 5.4. Columna con contador

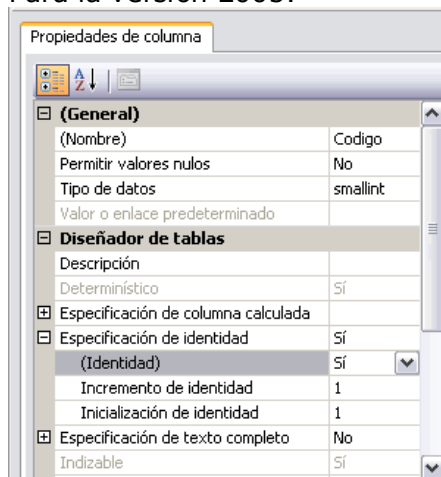
En la mayoría de los sistemas gestores de bases de datos tenemos un tipo de datos de tipo contador, autonumérico, autoincremental, etc. Este tipo hace que el propio sistema sea el encargado de rellenar el campo con un valor que va incrementando conforme se crean más filas de datos en la tabla.

Las columnas de este tipo se utilizan normalmente para numerar las filas de la tabla, como no habrán dos filas con el mismo valor (el sistema se encarga de incrementar el valor cada vez que se crea una nueva fila), estos campos se suelen utilizar como claves primarias.

En SQL Server no existe el tipo de datos Contador pero se consigue el mismo funcionamiento asignando a la columna un tipo de datos numérico y definiendo la columna como columna de identidad *IDENTITY COLUMN*.

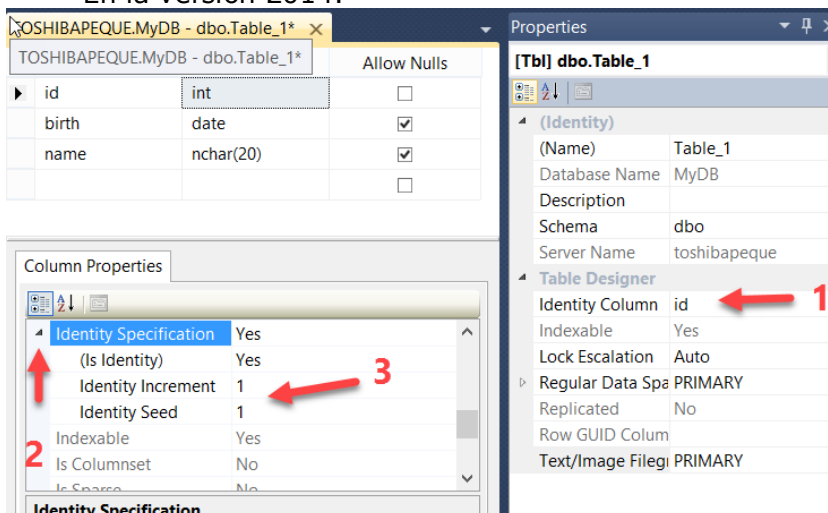
Aunque este tipo de columnas se utiliza frecuentemente como clave primaria, SQL Server no le asigna automáticamente esta función, la tenemos que definir nosotros mismos, pero sí fuerza a que sea una columna sin valores nulos y no se puede definir más de una columna de identidad por tabla.

La forma de definir una columna de identidad varía un poco de la versión 2005 a la 2014. Para la versión 2005:



En las propiedades de la columna marcamos **Sí** en la propiedad (**Identidad**) y a continuación podemos indicar en qué valor queremos que empiece el contador (**Iniciación de identidad**) y en cuánto incrementará cada vez que se cree un nuevo registro (**Incremento de identidad**).

En la versión 2014:

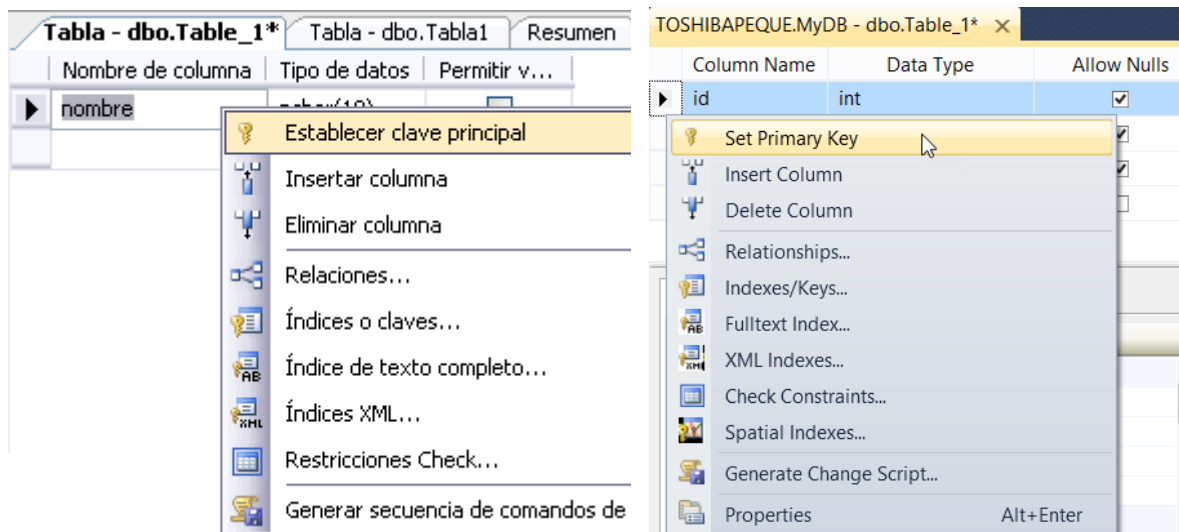


Primero tenemos que indicar en las propiedades de la tabla qué columna es de identidad (1). Para cambiar los valores de incremento (**Identity Increment**) e inicialización (**Identity Seed**), abrimos la propiedad Identity Specification (2) y modificamos los valores que queramos (3).

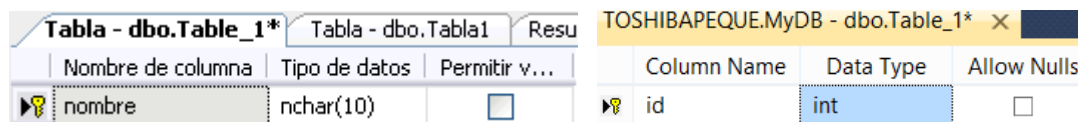


## 5.5. Clave primaria

Para definir una columna como clave primaria, posicionamos el puntero del ratón sobre la columna, desplegamos el menú contextual y seleccionamos la opción **Establecer Clave principal**:

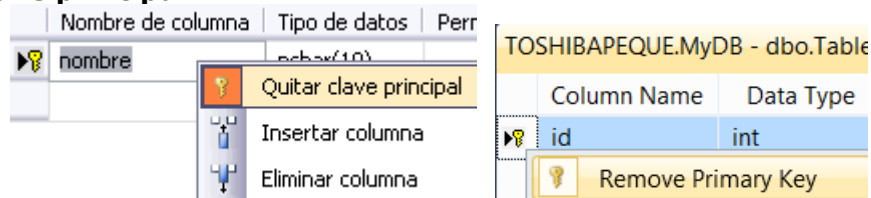


Aparecerá una llave a la izquierda del nombre, símbolo de las claves principales y si el campo permitía nulos automáticamente se deshabilita la casilla:



Para definir una clave primaria compuesta por varias columnas, seleccionamos las columnas manteniendo pulsada la tecla Ctrl y luego seleccionamos la opción.

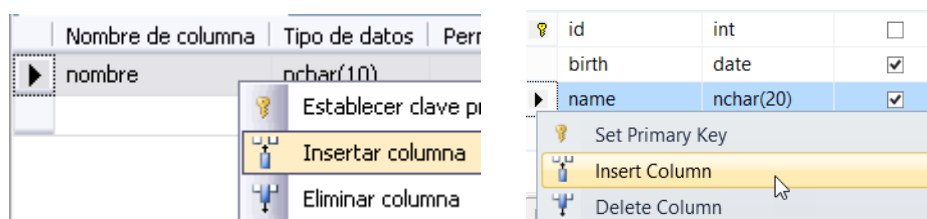
Para quitar una clave principal, hacemos lo mismo pero en esta ocasión seleccionamos la opción **Quitar clave principal**.



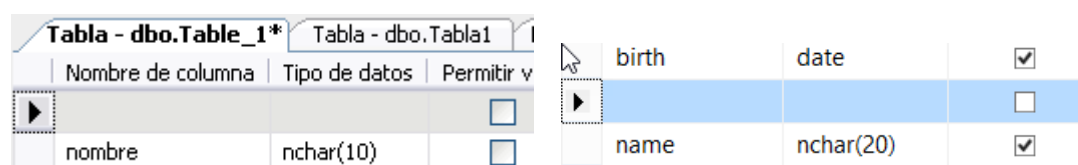
También podemos utilizar el icono de la barra de herramientas.

## 5.6. Añadir o eliminar columnas

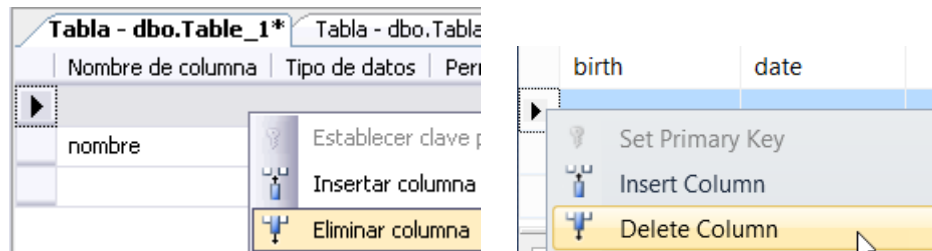
Una vez definidas algunas columnas, si queremos añadir una nueva columna entre dos, nos posicionamos en la segunda y seleccionamos la opción **Insertar columna** del menú contextual.



La nueva columna se colocará delante:

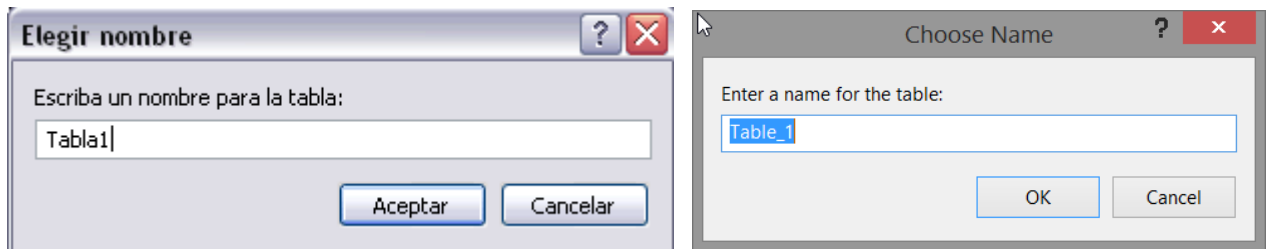


Del mismo modo si queremos eliminar la definición de una columna, nos posicionamos en la columna a eliminar y seleccionamos la opción **Eliminar columna**:

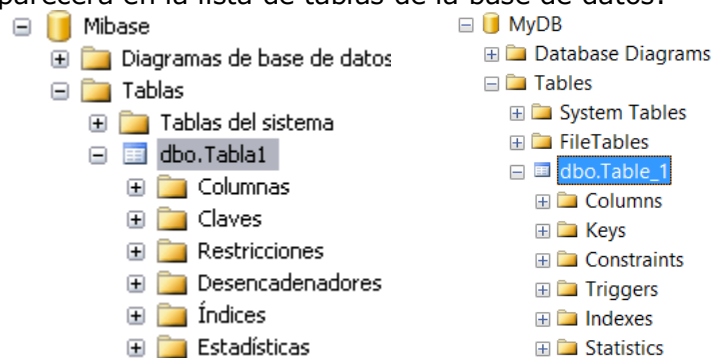


O simplemente hacemos clic en la zona a la izquierda del nombre y pulsamos la tecla Supr.

Finalmente guardamos la tabla, nos pedirá el nombre de la tabla:

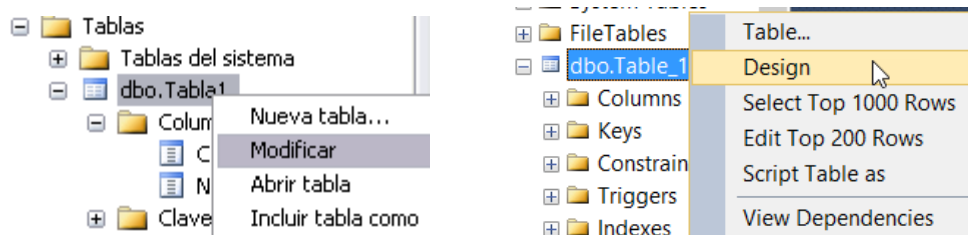


La nueva tabla aparecerá en la lista de tablas de la base de datos:



## 5.7. Modificar la definición de una tabla

Para entrar a la ventana de definición de la tabla utilizamos la opción **Modificar** de su menú contextual:



Se abrirá la ventana que ya conocemos para definir las columnas de la tabla.

## 5.8. Insertar datos en la tabla

Ahora que tenemos la tabla creada podemos rellenarla con datos. Para eso debemos abrir la tabla:



Se abrirá una ventana parecida a esta:

Tabla - dbo.Cientes		
	Codigo	Nombre
	1	Luis
	2	Juan
	NULL	Ana
*	NULL	NULL

TOSHIBAPEQUE.MyDB - dbo.Table_1			
	id	birth	name
	1	1960-02-01	John
	2	1960-12-03	Rose
*	NULL	NULL	NULL

La primera columna sirve para indicarnos el estado de una fila, por ejemplo el \* nos indica que es una nueva fila, esta fila realmente no está en la tabla, nos sirve de contenedor para los nuevos datos que queramos insertar.

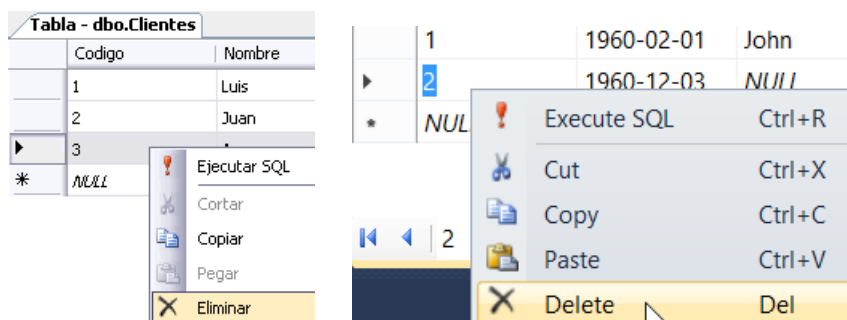
Para insertar una nueva fila de datos sólo tenemos que rellenar los campos que aparecen en esa fila (la del \*), al cambiar de fila los datos se guardarán automáticamente en la tabla a no ser que alguno infrinja alguna regla de integridad, en ese caso SQL Server nos devuelve un mensaje de error para que corrijamos el dato erróneo, si no lo podemos corregir entonces sólo podemos deshacer los cambios.

## 5.9. Modificar datos

Para modificar un valor que ya está en una fila de la tabla sólo tenemos que posicionarnos en el campo y rectificar el valor. En cuanto modificamos un valor, la fila aparece con un lápiz escribiendo (ver imagen), este lápiz nos indica que la fila se ha modificado y tiene nuevos datos por guardar. Al salir de la fila ésta se guardará automáticamente a no ser que el nuevo valor infrinja alguna regla de integridad. Si queremos salir de la fila sin guardar los cambios, tenemos que cancelar la actualización pulsando la tecla ESC.

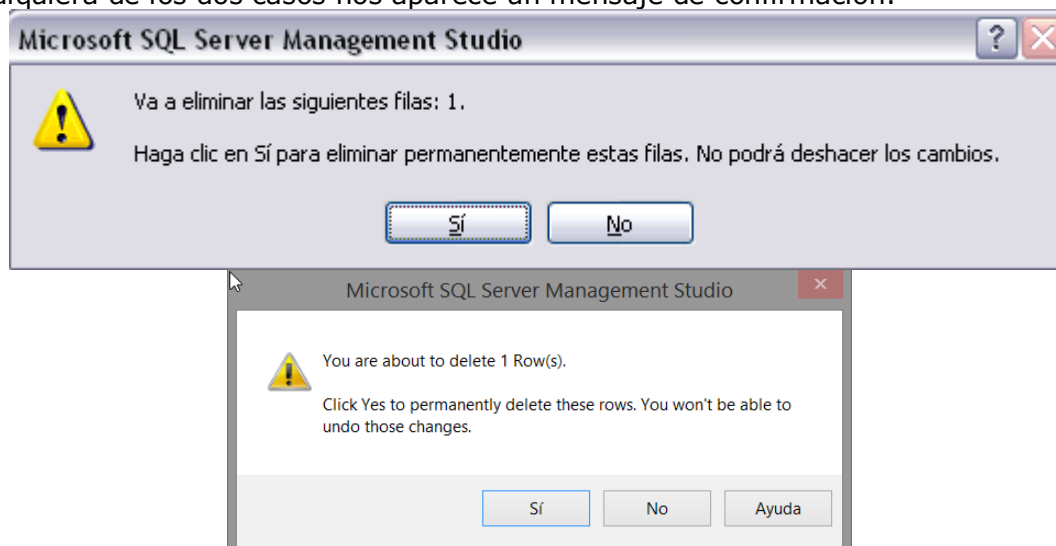
## 5.10. Eliminar filas

Para eliminar una fila completa desplegamos su menú contextual y seleccionamos la opción **Eliminar**:



O simplemente nos posicionamos en la columna de la izquierda encima de la fila a eliminar y pulsamos la tecla Supr.

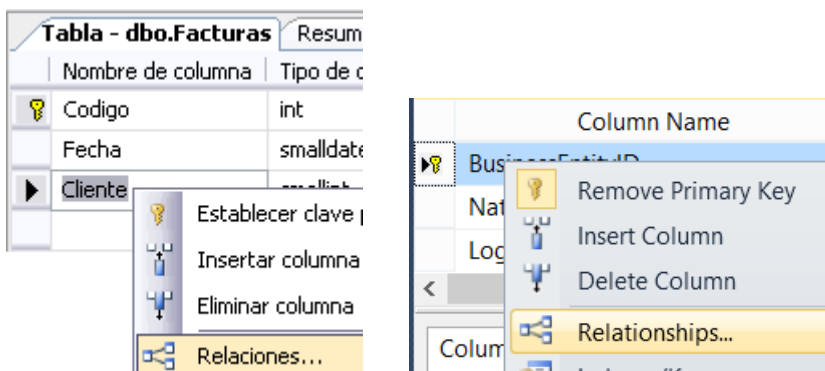
En cualquiera de los dos casos nos aparece un mensaje de confirmación.



## 6. Relacionar tablas

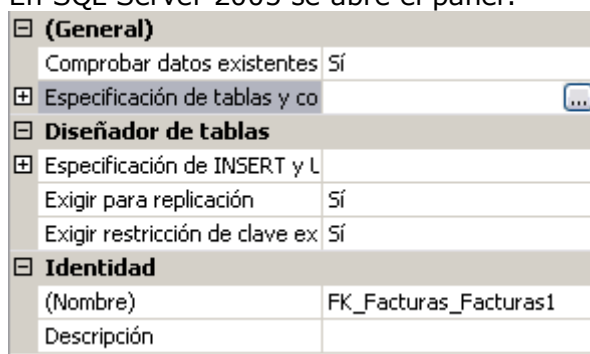
Como ya hemos visto, en una base de datos relacional, las relaciones entre las tablas se implementan mediante la definición de claves ajenas, son campos que contienen valores que señalan a un registro en otra tabla, en esta relación así creada, la tabla referenciada se considera principal y la que contiene la clave ajena es la subordinada. También se habla de tabla padre *PARENT* (para la principal) y tabla hijo *CHILD* (para la subordinada).

Desde el entorno gráfico del SSMS podemos definir claves ajenas entrando en el diseño de la tabla y desplegando el menú contextual del campo que va a ser clave ajena:



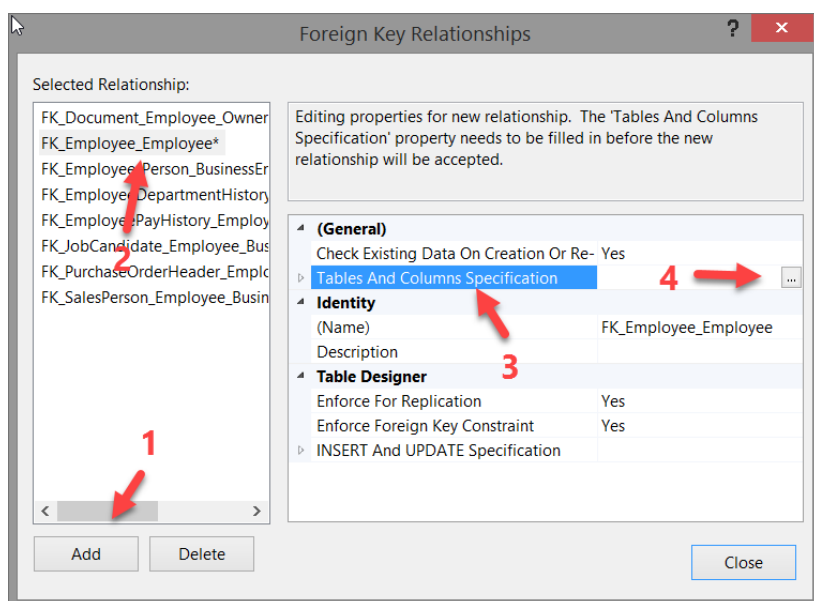
Seleccionamos la opción Relaciones.../Relationships...

En SQL Server 2005 se abre el panel:



Al pulsar el botón que se encuentra en la fila **Especificación de tablas y columnas** se abre el diálogo donde definiremos la relación como veremos más abajo.

En SQL Server 2014 se abre el siguiente panel:



Aparecen a la izquierda todas las relaciones definidas y al pulsar en Add (1) aparece una nueva relación en la lista (2) con un \* para indicar que es la que estamos editando.

En la parte derecha se ven las propiedades de la relación seleccionada.

Para completar la definición seleccionamos la fila *Tables And Columns Specification* (3), aparece el botón (4) que pulsamos y se abrirá el diálogo siguiente:

Nombre de la relación:

FK\_Facturas\_Facturas

Tabla de clave principal: Tabla de clavna:

Cientes Facturas

Codigo	Cliente

Aceptar Cancelar

Relationship name:

FK\_Employee\_Department

Primary key table: Foreign key table:

Department (HumanResources) Employee (HumanResources)

DepartmentID BusinessEntityID

OK Cancel

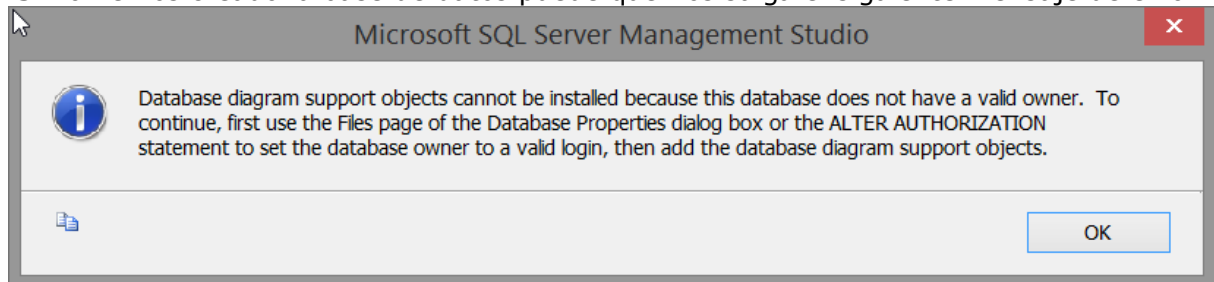
En la parte derecha de la ventana tenemos la tabla en la que estamos (la que va a contener la clave ajena), no se puede modificar, y debajo el campo que va a actuar como clave ajena, sólo nos queda elegir en el desplegable de la izquierda la tabla a la que hace referencia la clave ajena y debajo seleccionar el campo de dicha tabla al que referencia la clave ajena.

De esta forma hemos definido una relación entre las tablas Facturas/Employee y Cientes/Department. Para ver las relaciones existentes entre las diferentes tablas tenemos los diagramas.

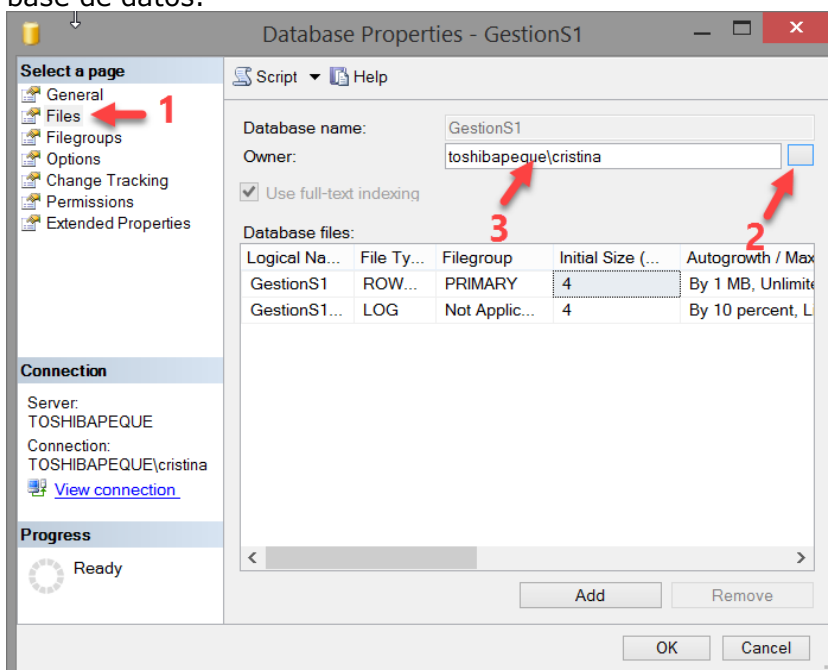
Primero debemos definir el diagrama, para ello seleccionamos la opción correspondiente:



Nota. Si no hemos creado la base de datos puede que nos salga el siguiente mensaje de error:



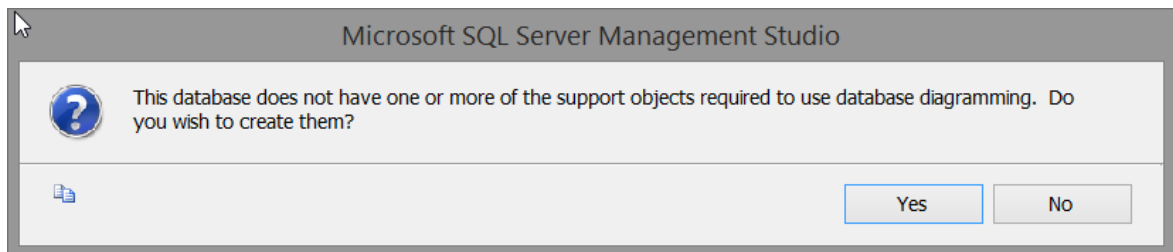
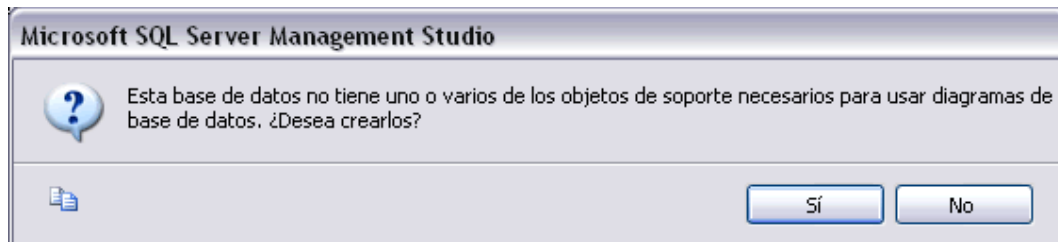
Podemos solucionarlo haciendo lo que nos sugiere, entrar en la ventana Propiedades de la base de datos:



En la opción *Files* (1) pulsar el botón para completar el campo *Owner* (2), se abrirá un diálogo para indicar un propietario válido (nuestro usuario), al aceptar aparecerá el usuario (3).

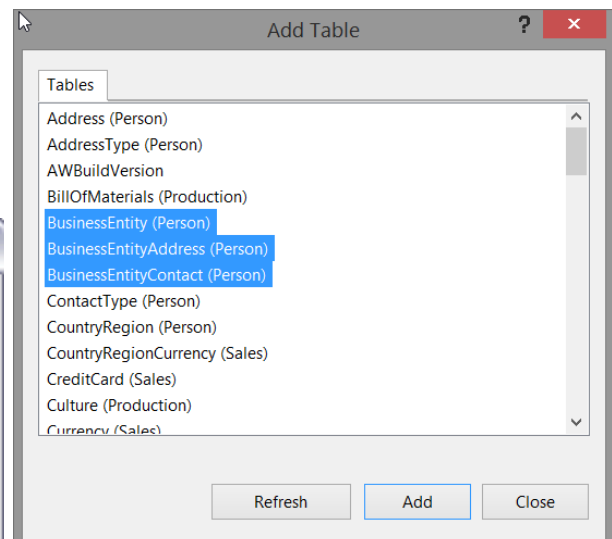
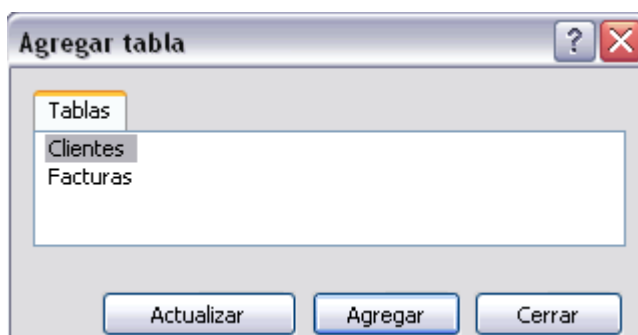
Guardamos pulsando en OK.

Si no tenemos todavía ningún diagrama creado en la base de datos, nos aparece un mensaje:

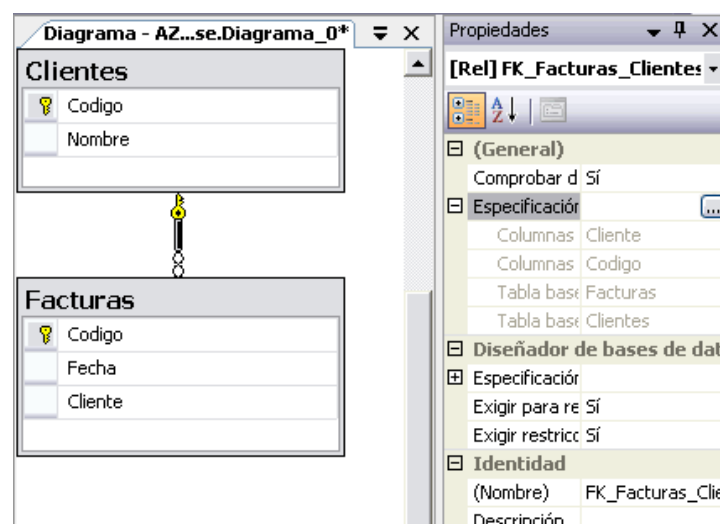


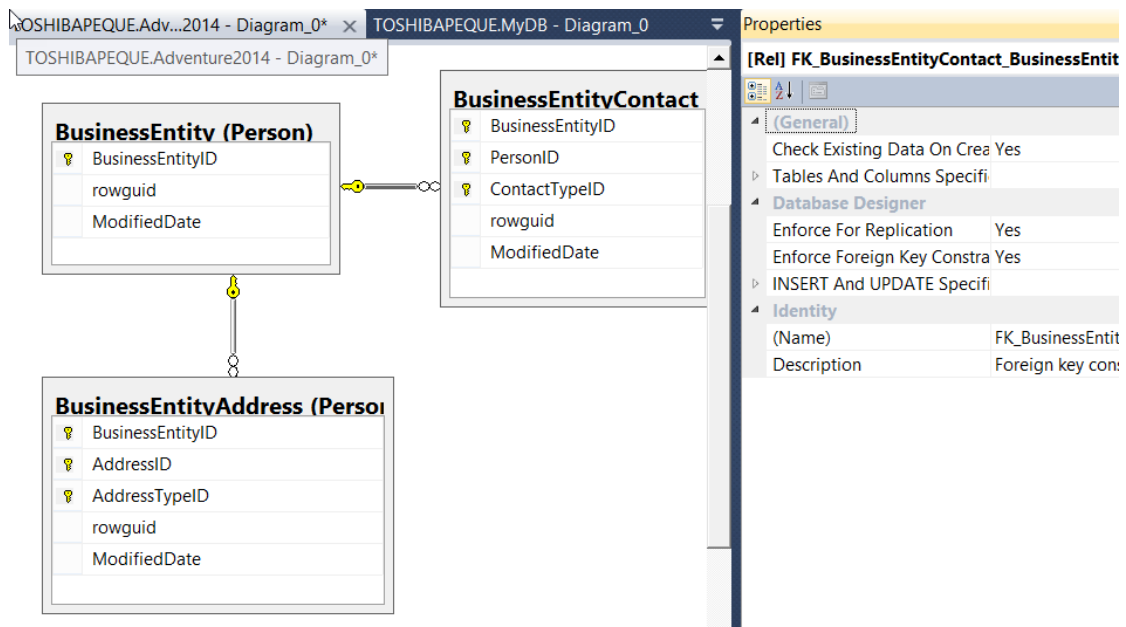
Elegimos Sí y se crea digamos el soporte donde se pintará el diagrama.

A continuación si elegimos crear un nuevo diagrama nos preguntará las tablas a incluir en el diagrama:



Seleccionamos cada una (con CTRL podemos seleccionar varias saltadas, con May seleccionamos un bloque) y pulsamos **Agregar/Add**, cuando hayamos agregado al diagrama todas las que queremos pulsamos en **Cerrar/Close** y aparecerán en el diagrama las tablas con las relaciones que tengan definidas en ese momento y la ventana Propiedades:





Para ver las propiedades de una relación hacemos clic sobre ella. La llave indica la tabla principal (padre) y el símbolo infinito señala la tabla que contiene la clave ajena (hijo).

En el examinador de objetos en la carpeta **Diagramas de base de datos** aparecen todos los diagramas definidos hasta el momento:



Hemos aprendido hasta ahora lo básico para poder crear una base de datos y rellenarla con tablas relacionadas entre sí y con datos, ahora veamos cómo recuperar esos datos.

## 7. Las consultas

### 7.1. Abrir una nueva consulta

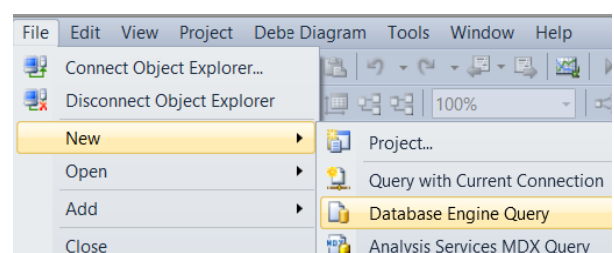
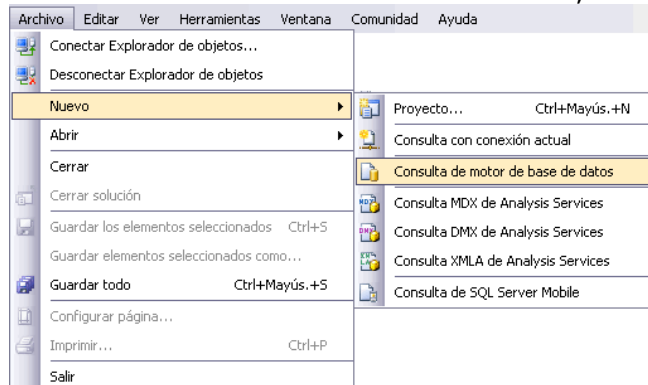
Vamos a ver ahora cómo crear consultas/queries SQL y ejecutarlas desde el entorno del SSMS.

Para ello debemos abrir la zona de trabajo de tipo Query, abriendo una nueva consulta, seleccionando previamente la base de datos sobre la que vamos a trabajar y pulsando el botón



de la barra de herramientas.

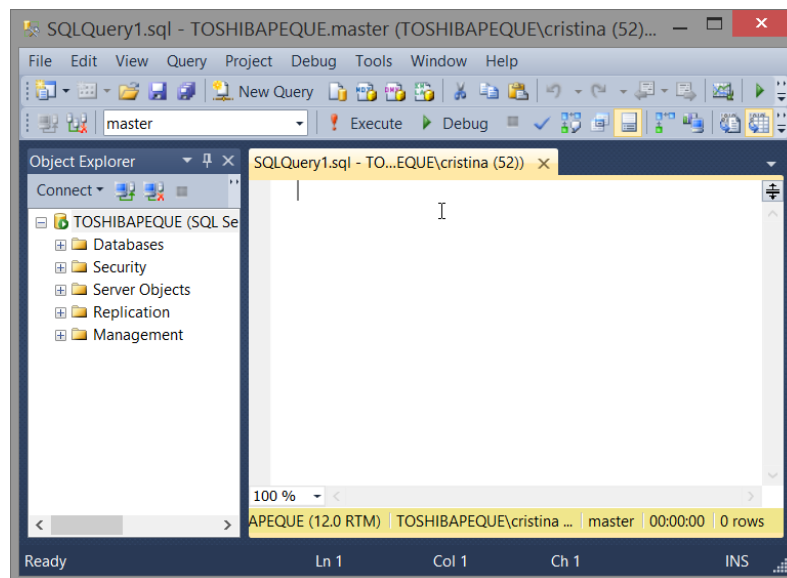
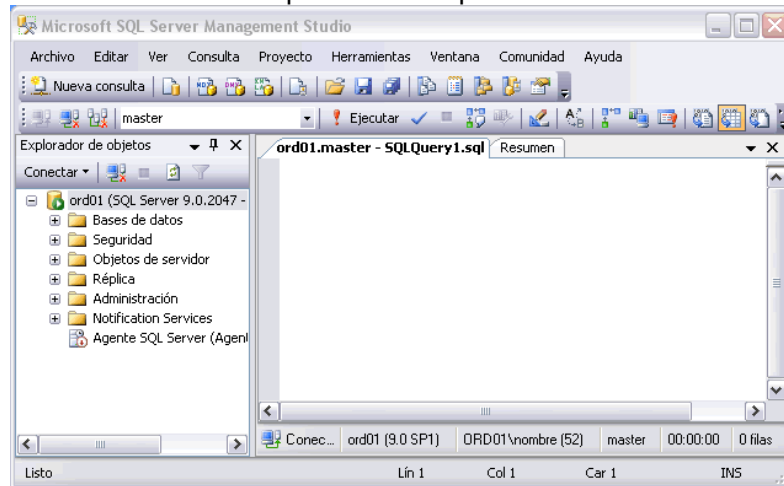
Si queremos realizar la consulta sobre una base de datos alojada en otro servidor con el cual todavía no hemos establecido conexión, seleccionamos de la barra de menús la opción:



En este último caso nos aparecerá el cuadro de diálogo para establecer la conexión (el mismo que vimos al principio del tema).



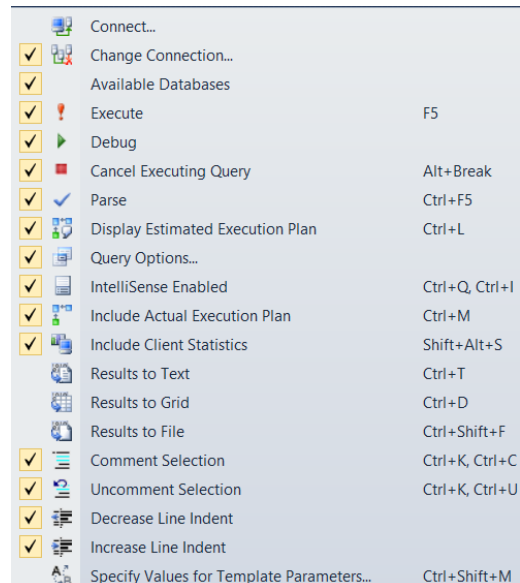
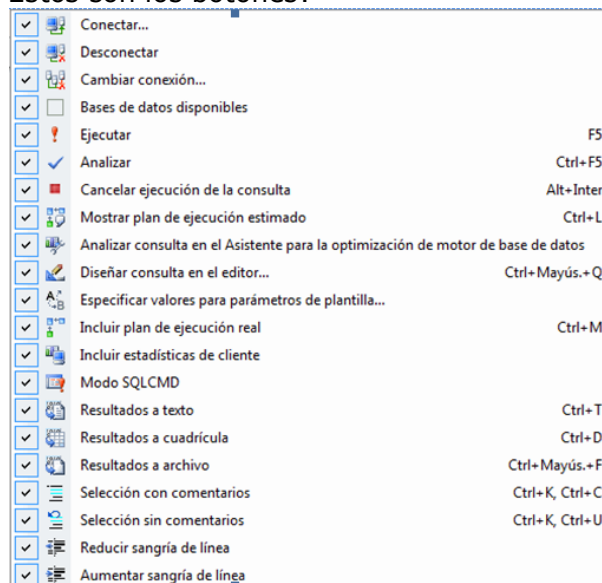
A continuación se abrirá un nuevo panel donde podremos teclear las sentencias SQL:



Además aparece una nueva barra de herramientas (la barra Editor SQL) que nos permitirá ejecutar los comandos más útiles del modo query.



Estos son los botones:



En la imagen de la derecha tienes marcados los que te pueden ser útiles. No veremos todos los botones en este tema, a continuación veremos unos cuantos.

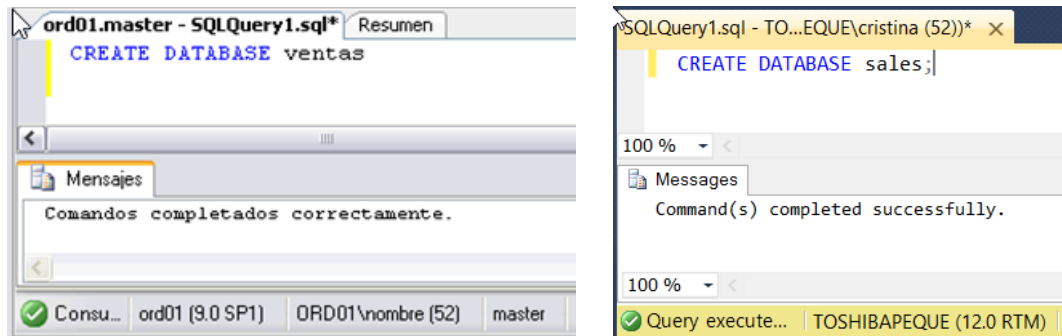
## 7.2. Escribir y ejecutar código TRANSACT-SQL

Sólo tenemos que teclear la sentencia a ejecutar, por ejemplo empezaremos por crear la base de datos.

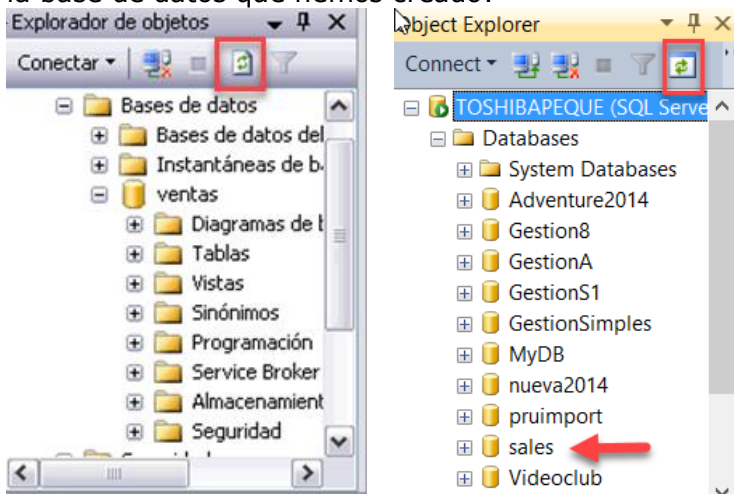
Utilizaremos la sentencia CREATE DATABASE mínima:

```
CREATE DATABASE ventas;
```

Al pulsar el botón **Ejecutar** se ejecuta la sentencia y aparece en la parte inferior el resultado de la ejecución, en la pestaña **Mensajes**:



Si ahora desplegamos la carpeta Bases de Datos del Explorador de Objetos, observaremos la base de datos que hemos creado:



Si la base de datos no aparece refresca la lista con el botón Actualizar/Refresh señalado en la imagen.

Si la ejecución de la sentencia produce un error, el sistema nos devolverá el mensaje de error escrito en rojo en la pestaña **Mensajes**.

Podemos incluir en una misma consulta varias sentencias SQL, cuando pulsamos Ejecutar se ejecutarán todas a detrás de la otra.

Si no queremos ejecutarlas todas podemos comentar las que no queremos que se ejecuten (como veremos en el siguiente punto) o bien seleccionar las que queremos ejecutar antes de pulsar el botón Ejecutar ya que si al dar la orden de ejecutar hay algunas líneas seleccionadas, el sistema entiende que sólo tiene que ejecutar dichas líneas.


## 7.3. Incluir comentarios

En el texto SQL de la consulta podemos querer incluir comentarios, son palabras que no son normalmente SQL sino texto nuestro que no deberá interpretar el sistema cuando ejecute la consulta.

Para indicar que lo que hay escrito es un comentario se escribe delante del comentario dos guiones --, el comentario se puede escribir en una o varias líneas o al final de una línea con instrucción SQL. Por ejemplo:

```
CREATE DATABASE ventas;  -- Esta instrucción sirve para crear la bd Ventas
--Esto es otro comentario
```

Los comentarios aparecen normalmente en color verde.

Otra forma más cómoda de definir comentarios es utilizando los botones  para comentar o "descomentar" las líneas seleccionadas en ese momento. ¡Ojo! Sólo sirven para comentar o "descomentar" líneas enteras.

También se pueden incluir bloques de comentarios (un comentario que se extiende en varias líneas), en este caso podemos escribir -- al principio de cada línea o encerrar el bloque formado por esas líneas entre los caracteres /\* y \*/. Ejemplo:

```
/* esto es un comentario de bloque
se extiende en varias líneas
CREATE DATABASE ventas2
y termina con los caracteres */
CREATE DATABASE ventas3
```

Fíjate que la primera instrucción CREATE DATABASE aparece en verde, está comentada, no se ejecutará mientras que la segunda aparece del color normal, no está comentada.

## 7.4. Guardar la consulta

Las consultas no son objetos de la base de datos, cuando se guarda una consulta, ésta se almacena fuera de la base de datos en un archivo de texto que normalmente tiene extensión .sql. Este archivo lo puedes abrir con el Bloc de notas y verás el mismo texto que desde la ventana del SSMS pero sin los colores.

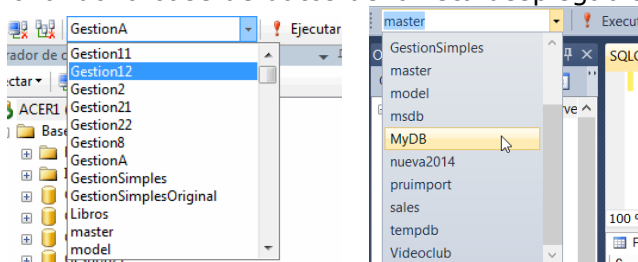
## 7.5. La base de datos predeterminada

Cuando ejecutamos consultas desde el editor, nos tenemos que fijar sobre qué base de datos se va a actuar.

Fijándonos en la pestaña de la consulta, en la parte inferior aparece el nombre del servidor y el nombre de la base de datos sobre la que se va a actuar (en las imágenes *master*).

Cuando creamos una nueva consulta, ésta actuará sobre la base de datos activa en ese momento. Por defecto la base de datos activa es la predeterminada (master). Si queremos que la base de datos activa sea por ejemplo la base de datos ventas, hacemos clic sobre su nombre en el Explorador de objetos, y ésta pasará a ser la base de datos activa. Si ahora creamos una nueva consulta, ésta actuará sobre la base de datos ventas.

También podemos cambiar o visualizar la base de datos sobre la que actuará la consulta seleccionando la base de datos de la lista desplegable que aparece en la barra Editor SQL




Si queremos crear una consulta que siempre actúe sobre una determinada base de datos y no nos queremos preocupar de qué base de datos tenemos activa en el momento de la ejecución, podemos añadir al principio de la consulta la instrucción USE nombreBaseDatos; esto hará que la base de datos activa pase a ser la indicada, y todas las instrucciones que aparezcan después, se ejecutarán sobre ella.

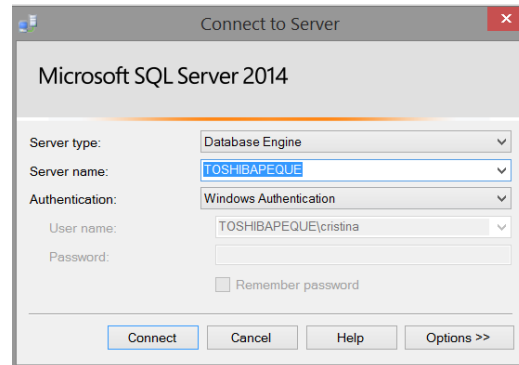
Por ejemplo:

```
USE ventas;
SELECT * FROM pedidos;
```

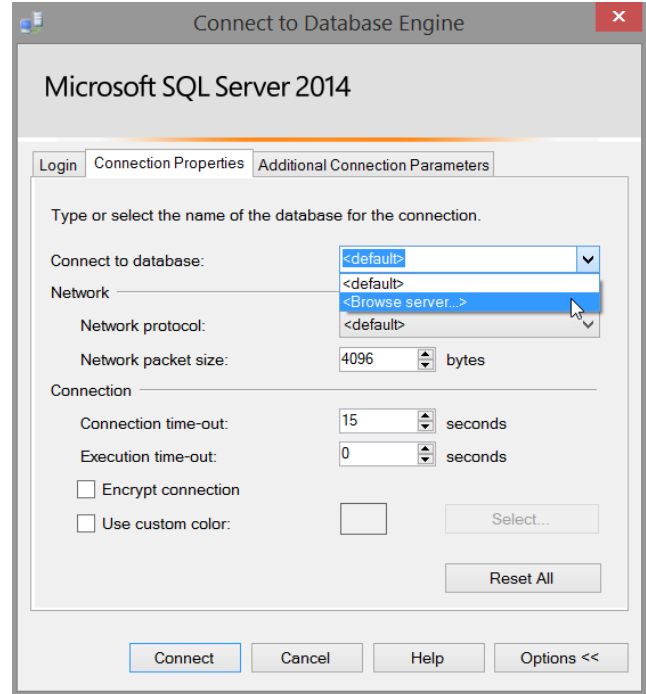
Obtiene todos los datos de la tabla *pedidos* que se encuentra en la base de datos *ventas*.

Si no utilizamos USE y almacenamos la consulta, al abrirla otra vez, cogerá como base de datos la predeterminada (no la activa) y se volverá a ejecutar sobre la base de datos master.

Normalmente utilizaremos como base de datos la nuestra y no la base de datos *master*, por lo que nos será útil cambiar el nombre de la base de datos predeterminada, esto lo podemos hacer cambiando la base de datos por defecto en el id de sesión. Para llegar a la ventana puedes hacerlo cuando vas a conectar con el servidor o bien pulsando el botón Cambiar conexión de la barra  :




Pulsamos en el botón **Opciones>>**,



En la pestaña **Propiedades de conexión**, en el cuadro **Conectar con base de datos:** Seleccionamos **<Examinar servidor>** para elegir la base de datos.

La elegimos y aceptamos. A partir de ese momento la base de datos elegida será la que SQL Server coja por defecto en todas las sesiones de ese usuario.

## 7.6. Intellisense

En la versión 2014, Management Studio incorpora Intellisense, la aplicación de Microsoft para autocompletar, utilizada en muchos de sus entornos como por ejemplo Microsoft Visual Studio. Ayuda a redactar instrucciones y minimiza los errores. Se puede activar o desactivar con el botón .

Para más información consulta la página:

[https://msdn.microsoft.com/es-es/library/ms174207\(v=sql.110\).aspx](https://msdn.microsoft.com/es-es/library/ms174207(v=sql.110).aspx)

## 7.7. Los colores del editor de código

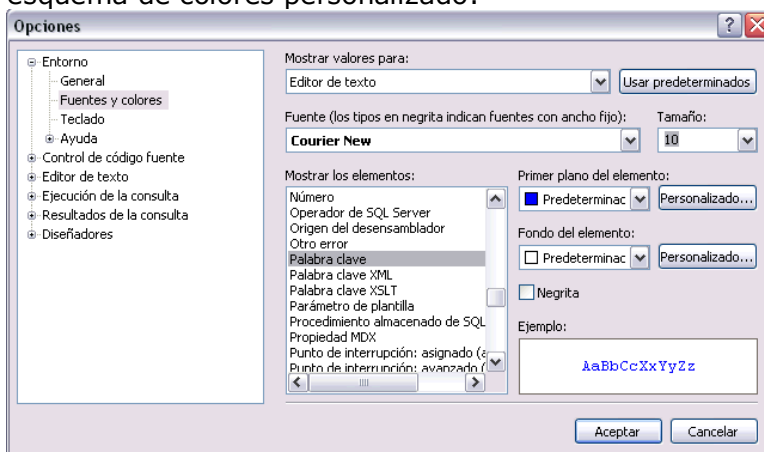
Para facilitarnos la redacción y corrección de las sentencias, el editor de SQL presenta las palabras de distintos colores según su categoría y podemos utilizar el panel Explorador de Objetos para arrastrar desde él los objetos sobre la zona de trabajo y así asegurarnos de que los nombres de los objetos (por ejemplo nombre de tabla, de columna, etc.) sean los correctos.

Como hemos dicho el texto que se escribe en este editor de código se colorea por categoría. Los colores son los mismos que se utilizan en todo el entorno SQL Server. En esta tabla aparecen los colores más comunes.

Color	Categoría
Rojo	Cadena de caracteres/String
Verde oscuro	Comentario/Comment
Comando SQLCMD	Negro sobre fondo plateado
Fucsia	Función del sistema
Verde	Tabla del sistema
Azul	Palabra clave/Keyword
Verde azulado	Números de línea o parámetro de plantilla
Rojo oscuro	Procedimiento almacenado de SQL Server
Gris oscuro	Operadores

## 7.8. Configurar un esquema de colores personalizado

En el menú **Herramientas, Opciones**, desplegando la opción **Entorno, Fuentes y colores**, se puede ver la lista completa de colores y sus categorías, así como configurar un esquema de colores personalizado:



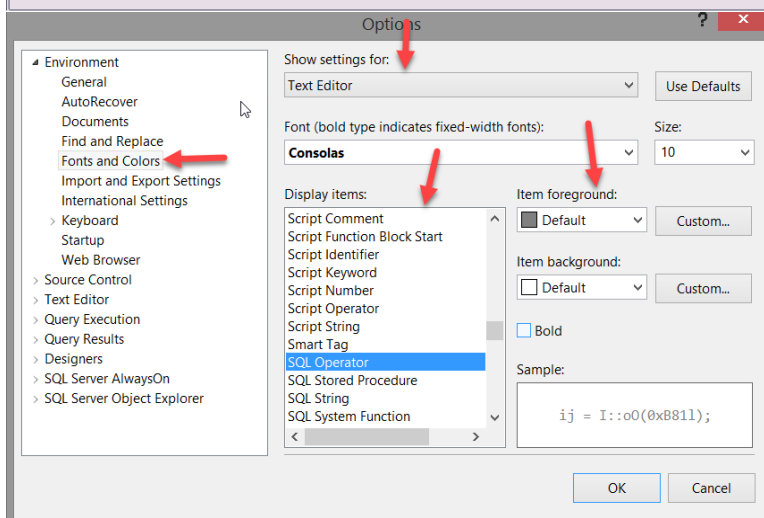
En la lista *Mostrar valores para:* /*Show settings for:*, seleccionamos el entorno que se verá afectado.

El botón *Usar predeterminados/Use Defaults* nos permite volver a la configuración predeterminada.

En la lista *Mostrar elementos* /*Display Items* seleccionamos el elemento a personalizar y cambiamos sus propiedades que aparecen en la parte derecha.

En *Ejemplo:/Sample:* vemos un ejemplo de cómo queda el elemento así configurado.

En 2014 la lista de elementos es mucho más larga, los que nos pueden interesar son Keyword (palabra clave), los que empiezan por Script (el texto que estamos escribiendo en el editor SQL), y los que empiezan por SQL.



Ahora sólo nos queda aprender a redactar sentencias SQL, cosa que haremos en los próximos temas.