

Módulo 1 de Desarrollo de Base de datos

- Instalación completa de SQL SERVER y SQL SERVER MANAGEMENT STUDIO (video de clase)
- Tipos de inicio de Sesión en SQL SERVER (Modo Windows-Modo SQL) (video de clase)
- Conceptos básicos del lenguaje SQL y entorno gráfico SQL SERVER, Object Explorer (El explorador de objetos) (video de clase)
- Modo gráfico de SQL SERVER
- Bases de datos del sistema
- Lenguaje de Definición de Datos (DDL) (Create database, Create table, Claves primarias, claves Foráneas)
 - Crear una base datos en modo gráfico
 - Barra de herramientas del editor de consultas SQL SERVER
 - Cambiar el contexto de la base de datos (USE)
 - Comando GO
 - Crear una base de datos usando grupos lógicos (.mdf y .ldf) con sentencias en modo comando
 - Crear y guardar un script. SQL desde el editor de consultas
 - Eliminar una base datos en modo gráfico
 - Eliminar una base de datos con sentencia en modo comando
 - Tipos de datos en SQL SERVER
 - Crear una tabla en modo gráfico (atributos con tipos de datos, clave primaria identity, nulos).
 - Duplicar una tabla a partir de una tabla existente usando el editor de consultas
 - Duplicar o copiar una tabla usando el modo gráfico
 - Crear una tabla con sentencia en modo comando
(atributos con tipos de datos, clave primaria identity, nulos)
 - Eliminar una tabla en modo gráfico
 - Eliminar una tabla con sentencia en modo comando
 - Alter table** (adicionar una columna, eliminar una columna, cambiar nombre de la tabla, cambiar el tipo de dato en una columna)
 - Cambiar la posición de una columna en SQL SERVER
 - Integridad referencial en la definición de claves foráneas y generación del modelo relacional (modo gráfico y Transact-SQL).

Bases de datos del sistema

Base de datos del sistema	Descripción
Base de datos maestra (máster)	Registra toda la información del sistema para una instancia de SQL Server.
Base de datos msdb	La utiliza el Agente SQL Server para programar alertas y trabajos.
Base de datos model	Se utiliza como plantilla para todas las bases de datos creadas en la instancia de SQL Server. Las modificaciones hechas a la base de datos model, como el tamaño de la base de datos, la intercalación, el modelo de recuperación y otras opciones de base de datos, se aplicarán a las bases de datos que se creen con posterioridad.
Base de datos tempdb	Área de trabajo que contiene objetos temporales o conjuntos de resultados intermedios.

DDL(Lenguaje de definición de datos

)

El DDL (Data Definition Language, o Data Description Language según autores), es la parte del SQL dedicada a la definición de la base de datos, consta de sentencias para definir la estructura de la base de datos, permite definir gran parte del nivel interno de la base de datos. Por este motivo estas sentencias serán utilizadas normalmente por el administrador de la base de datos.

La definición de la estructura de la base de datos incluye tanto la creación inicial de los diferentes objetos que formarán la base de datos, como el mantenimiento de esa estructura. Las sentencias del DDL utilizan unos verbos que se repiten para los distintos objetos. Por ejemplo, para crear un objeto nuevo el verbo será CREATE y a continuación el tipo de objeto a crear. CREATE DATABASE es la sentencia para crear una base de datos, CREATE TABLE nos permite crear una nueva tabla, CREATE INDEX crear un nuevo índice... Para eliminar un objeto utilizaremos el verbo DROP (DROP TABLE, DROP INDEX...) y para modificar algo de la definición de un objeto ya creado utilizamos el verbo ALTER (ALTER TABLE, ALTER INDEX...).

Los objetos que veremos en este tema son:

Bases de datos

Tablas

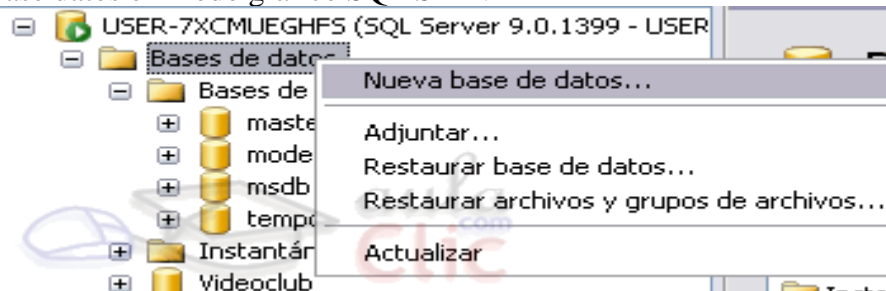
Vistas

Índices

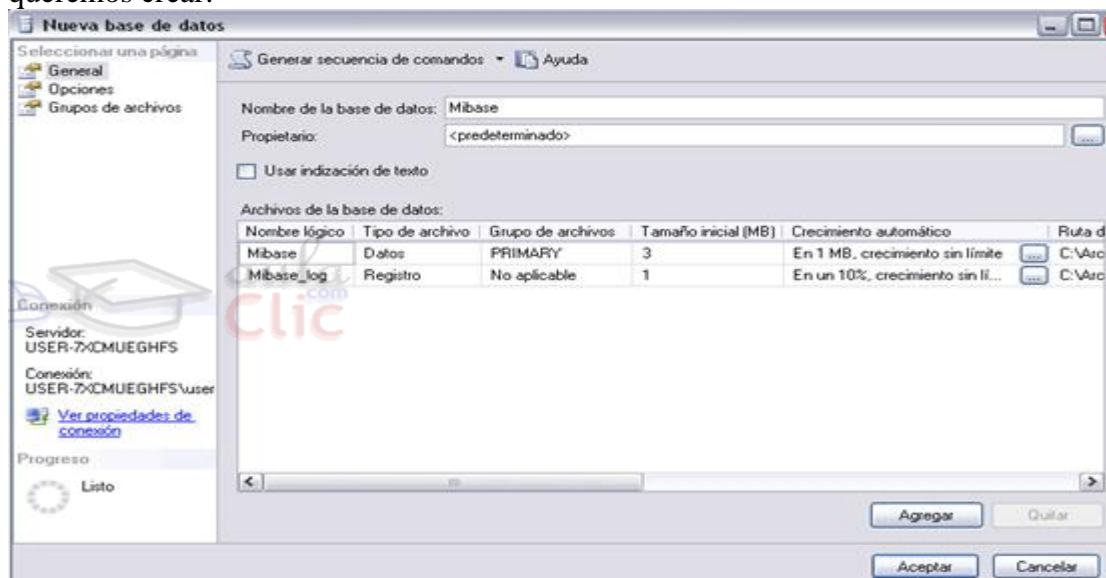
Crear una base de datos en modo gráfico SQL SERVER

Para crear una nueva base de datos de usuario nos posicionamos sobre la carpeta Bases de datos y con el botón derecho del ratón desplegamos el menú contextual del que elegimos la opción Nueva base de datos.

Crear una base de datos en modo gráfico SQL SERVER



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



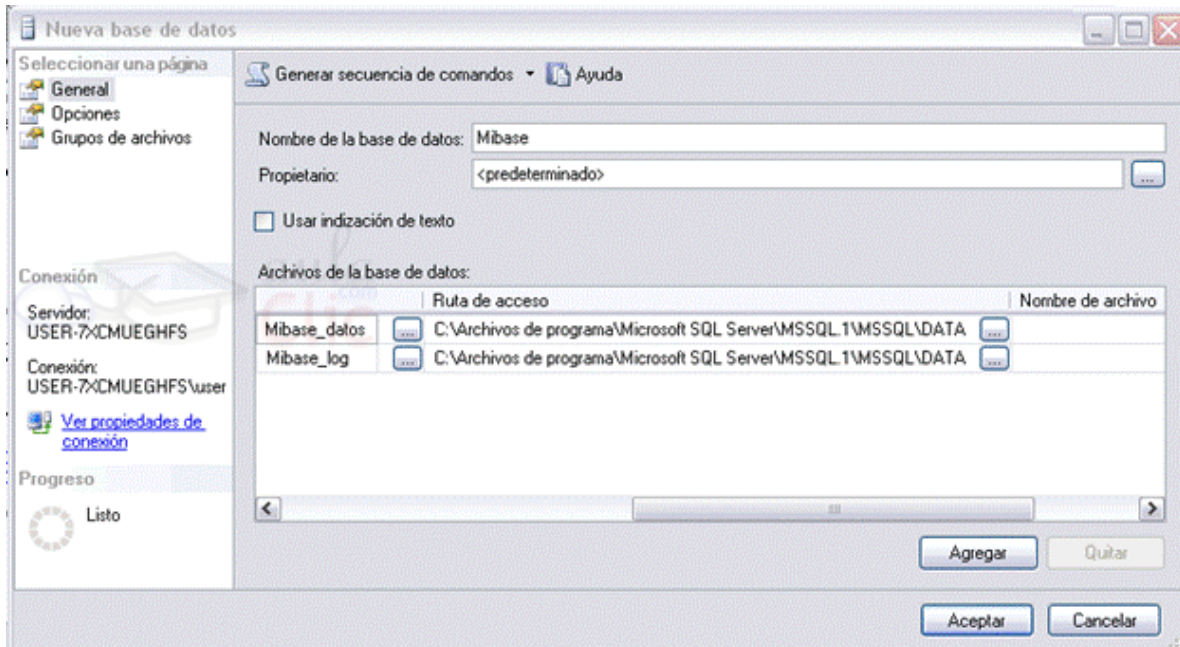
Lo mínimo a introducir será el campo Nombre de la base de datos, éste es el nombre de la base de datos lógica, la base de datos a la que nos referiremos dentro del SSMS, a nivel conceptual (en la imagen Mibase).

Esta base de datos está asociada a dos archivos físicos, en la parte inferior aparecen esos archivos. Para facilitarnos la tarea, al teclear el nombre de la base lógica, 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.

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


Si no indicamos ninguna ubicación podemos ver que los guarda en la carpeta del SQL Server/MSSQL.n/MSSQL/DATA.

n representa un número que puede variar de una instalación a otra.



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



Aparecerá dentro de la carpeta Bases de datos. Si no se ve pulsa en el icono Actualizar o la tecla F5. 

Barra de herramientas del Editor de consultas SQL SERVER

Cuando el Editor de consultas está abierto, la barra de herramientas del Editor SQL aparece con los botones siguientes.



También puede agregar la barra de herramientas del Editor SQL seleccionando el menú Ver y, a continuación, las opciones Barras de herramientas y Editor SQL. Si agrega la barra de herramientas del Editor SQL cuando no está abierta ninguna ventana del Editor de consultas, no habrá ningún botón disponible.

Íconos de opciones de la barra de herramientas del editor de izquierda a derecha:

Conexión mediante la barra de herramientas del editor

Se abre el cuadro de diálogo Conectar al servidor. Utilice este cuadro de diálogo para establecer una conexión a un servidor.

También puede conectarse a la base de datos mediante el menú contextual.

Cambio de la conexión mediante la barra de herramientas del editor

Se abre el cuadro de diálogo Conectar al servidor. Use este cuadro de diálogo para establecer una conexión a un servidor diferente.

También puede cambiar las conexiones mediante el menú contextual.

Bases de datos disponibles mediante la barra de herramientas del editor

Cambia la conexión a una base de datos distinta del mismo servidor.

Ejecución mediante la barra de herramientas del editor

Ejecuta el código seleccionado o, si no se ha seleccionado ningún código, ejecuta todo el código del editor de consultas.

También puede ejecutar una consulta con F5 o en el menú contextual.

Cancelación de la ejecución de consultas mediante la barra de herramientas del editor

Envía una solicitud de cancelación al servidor. Algunas consultas no pueden cancelarse inmediatamente, sino que deben esperar a una condición de cancelación adecuada. Cuando se cancelan las transacciones, podrían producirse retrasos mientras se revierten.

También puede cancelar una consulta en ejecución si selecciona Alt + Interrumpir.

Análisis mediante la barra de herramientas del editor

Comprueba la sintaxis del código seleccionado. Si no se ha seleccionado ningún código, comprueba la sintaxis de todo el código en la ventana del Editor de consultas.

También puede comprobar el código en el Editor de consultas con Ctrl + F5.

Visualización del plan de ejecución estimado mediante la barra de herramientas del editor

Solicita un plan de ejecución de consulta desde el procesador de consultas sin ejecutar la consulta y muestra el plan en la ventana Plan de ejecución. Este plan utiliza estadísticas de índice para calcular el número de filas que se prevé que van a devolverse en cada parte de la ejecución de la consulta. El plan de consultas real que se utiliza puede ser diferente del plan de ejecución calculado. Esto puede ocurrir si el número de filas devueltas es distinto del calculado y el procesador de consultas cambia el plan para conseguir mayor eficacia.

También puede mostrar un plan de ejecución estimado con Ctrl + L o en el menú contextual.

Opciones de consulta mediante la barra de herramientas del editor

Abre el cuadro de diálogo Opciones de consulta. Use este cuadro de diálogo para configurar las opciones predeterminadas de la ejecución de consultas y los resultados de las mismas.

También puede seleccionar Opciones de consulta en el menú contextual.

IntelliSense habilitado mediante la barra de herramientas del editor

Especifica si la funcionalidad de IntelliSense está disponible en el editor de consultas del motor de base de datos. Esta opción está establecida de forma predeterminada.

También puede seleccionar IntelliSense habilitado si presiona Ctrl + B y luego Ctrl I, o bien en el menú contextual.

Inclusión del plan de ejecución real mediante la barra de herramientas del editor

Ejecuta la consulta, devuelve los resultados de la consulta y usa el plan de ejecución para la consulta. Las consultas se muestran como un plan de consultas gráfico en la ventana Plan de ejecución.

También puede seleccionar Incluir plan de ejecución real con Ctrl + M o en el menú contextual.

Inclusión de estadísticas de consultas activas mediante la barra de herramientas del editor

Proporciona información en tiempo real sobre el proceso de ejecución de consultas a medida que los controles fluyen de un operador de plan de consulta a otro.

También puede seleccionar Include Live Query Statistics (Incluir estadísticas de consultas activas) en el menú contextual.

Inclusión de estadísticas de cliente mediante la barra de herramientas del editor

Incluye una ventana Estadísticas de clientes que contiene estadísticas sobre la consulta y los paquetes de red, así como el tiempo transcurrido de la consulta.

También puede seleccionar Include Live Query Statistics (Incluir estadísticas de consultas activas) mediante Mayús + Alt + S o en el menú contextual.

Resultados a texto mediante la barra de herramientas del editor

Devuelve los resultados de la consulta como texto en la ventana Resultados.

También puede devolver los resultados a texto con Ctrl + T o en el menú contextual.

Resultados a cuadrícula mediante la barra de herramientas del editor

Devuelve los resultados de la consulta como una o varias cuadrículas en la ventana Resultados. Esta opción está habilitada de forma predeterminada.

También puede devolver los resultados a texto con Ctrl + D o en el menú contextual.

Resultados a archivo mediante la barra de herramientas del editor

Cuando se ejecuta la consulta, se abre el cuadro de diálogo Guardar resultados. En Guardar en, seleccione la carpeta en la que desea guardar el archivo. En Nombre de archivo, escriba el nombre del archivo y, luego, seleccione Guardar para guardar los resultados de la consulta como un archivo de Informe con la extensión .rpt. Para acceder a las opciones avanzadas, seleccione la flecha abajo en el botón Guardar y, luego, seleccione Guardar con codificación.

También puede devolver los resultados a texto con Ctrl + Mayús + F o en el menú contextual.

Conversión en comentario de las líneas seleccionadas mediante la barra de herramientas del editor

Convierte la línea actual en comentario agregando al principio de la línea un operador de comentario (--).

También puede convertir en comentario una línea con Ctrl + K y Ctrl + C.

Eliminación de las marcas de comentario de las líneas seleccionadas mediante la barra de herramientas del editor

Convierte la línea actual en una instrucción de código activo quitando los operadores de comentario (--) del principio de la línea.

También puede quitar la marca de comentario de una línea si presiona Ctrl + K y, luego, Ctrl + U.

Reducción de la sangría mediante la barra de herramientas del editor

Mueve el texto de la línea a la izquierda quitando los espacios en blanco al principio de la línea.

Aumento de la sangría de línea mediante la barra de herramientas del editor

Mueve el texto de la línea a la derecha agregando espacios en blanco al principio de la línea.

Especificación de los valores de los parámetros de plantilla mediante la barra de herramientas del editor

Abre un cuadro de diálogo que puede utilizar para especificar los valores de los parámetros en los procedimientos almacenados y funciones.

Cambiar el contexto de la base de datos (USE)

```
USE {database_name}  
[;]
```

database_name

Es el nombre de la base de datos o instantánea de base de datos a la que se cambia el contexto de usuario. Los nombres de base de datos y de instantánea de base de datos deben cumplir las mismas reglas que los identificadores.

Observaciones

Cuando un inicio de sesión de SQL Server se conecta a SQL Server, el inicio de sesión se conecta automáticamente a su base de datos predeterminada y adquiere el contexto de seguridad de un usuario de la base de datos. Si no se ha creado ningún usuario de la base de datos para el inicio de sesión de SQL Server, el inicio de sesión se conecta como **guest**. Si el usuario de la base de datos no tiene permiso **CONNECT** en la base de datos, la instrucción **USE** no funcionará correctamente. Si no se ha asignado ninguna base de datos predeterminada al inicio de sesión, su base de datos predeterminada se establecerá en **master**.

USE se ejecuta en tiempo de compilación y de ejecución, y surte efecto inmediatamente. Por lo tanto, las instrucciones que aparecen en un lote después de la instrucción **USE** se ejecutan en la base de datos especificada.

Permisos

Requiere permiso **CONNECT** en la base de datos de destino.

Ejemplos

En el siguiente ejemplo se cambia el contexto de la base de datos al de la base de datos **mibase**

```
USE mi base;
```


utilidades de SQL Server: comando GO

SQL Server proporciona comandos que no son instrucciones de Transact-SQL, pero que las utilidades sqlcmd y osql, y el Editor de código de SQL Server Management Studio sí reconocen. Estos comandos se pueden usar para facilitar la legibilidad y la ejecución de lotes y scripts.

GO indica a las utilidades de SQL Server el final de un lote de instrucciones Transact-SQL.

GO no es una instrucción Transact-SQL, sino un comando reconocido por las utilidades sqlcmd y osql, así como por el Editor de código de SQL Server Management Studio.

Las utilidades de SQL Server interpretan GO como una señal de que deben enviar el lote actual de instrucciones Transact-SQL a una instancia de SQL Server. El lote actual de instrucciones está formado por todas las instrucciones especificadas desde el último comando GO o desde el comienzo de la sesión o script ad hoc si se trata del primer comando GO.

Una instrucción Transact-SQL no puede ocupar la misma línea que un comando GO. Sin embargo, la línea sí puede contener comentarios.

Las aplicaciones de SQL Server pueden enviar varias instrucciones Transact-SQL a una instancia de SQL Server para su ejecución como un lote. En ese momento, las instrucciones del lote se compilan en un único plan de ejecución. Los programadores que ejecutan instrucciones en las utilidades de SQL Server o que generan Scripts de instrucciones Transact-SQL para ejecutarlas a través de las utilidades de SQL Server utilizan el comando GO para indicar el final de un lote.

***No use un punto y coma como terminador de instrucción después de GO.**

Ejemplo:

En ejemplo se crean dos lotes. El primer lote solo contiene una instrucción USE mibase para establecer el contexto de base de datos. Y la segunda instrucción crea una nueva tabla dentro de la base de datos mibase.

```
USE mibase;
```

```
GO
```

```
Create table mitabla(  
Id int identity(1,1) primary key,  
Nombre varchar(50) not null,  
Cedula varchar(50) not null  
)
```

```
GO
```

Crear una base de datos usando grupos lógicos (.mdf y .ldf) con sentencias en modo comando

Estructura interna

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

Archivos de datos principales (.mdf): Archivo requerido

Archivos de registro (.ldf): Archivo requerido

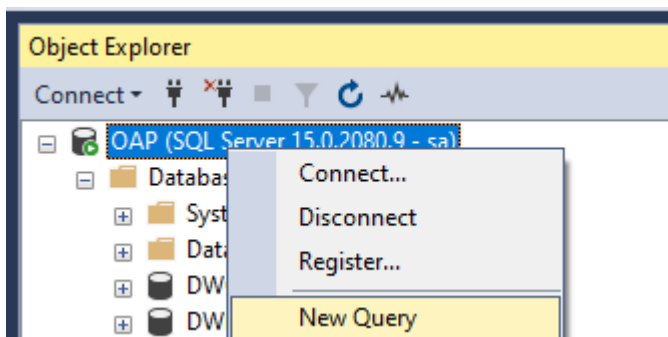
Comando mínimo para la creación de una base de datos:

CREATE DATABASE nbBasedeDatos

Comando completo con grupos lógicos:

Ejemplo explicado en clase:

1. Crear una nueva consulta



2. Sentencia completa para la creación de una base de datos con grupos lógicos

-- mi base propia

create database mibase

on primary

(Name=mibasededatos, filename='D:\data\mibasedatos.mdf' --Master Database Files

,size=50mb

,Filegrowth=25%

)

log on

(Name=mibasededatosLog, filename='D:\data\mibasedatos.ldf' --Log Database Files

,size=25mb

,Filegrowth=25%

)

Después de ejecutar la sentencia la base de datos se creará en la ubicación indicada en la instrucción Filename, y Aparecerá dentro de la carpeta Bases de datos. Si no se visualiza automáticamente debe hacer click en el ícono Actualizar o la tecla F5.



Crear y guardar un script. SQL desde el editor de consultas

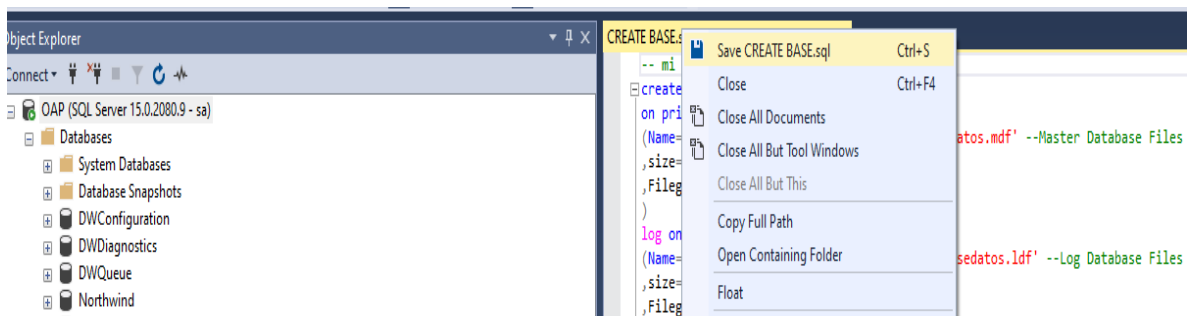
1. Crea una nueva consulta



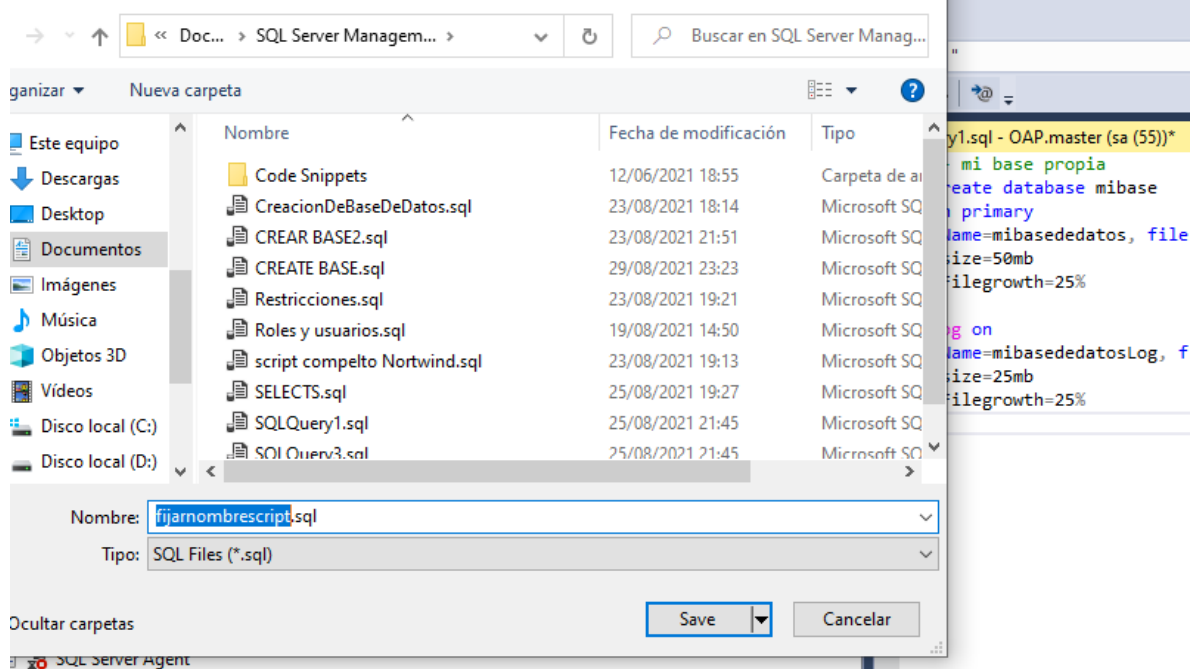
2. Escribir la consulta en el editor de consultas SQL SERVER

```
USE mibase;
GO
Create table mitabla(
Id int identity(1,1) primary key,
Nombre varchar(50) not null,
Cedula varchar(50) not null
)
GO
```

3. Hacer click derecho sobre la pestañan de la consulta, elegir la opción **Save** (guardar)

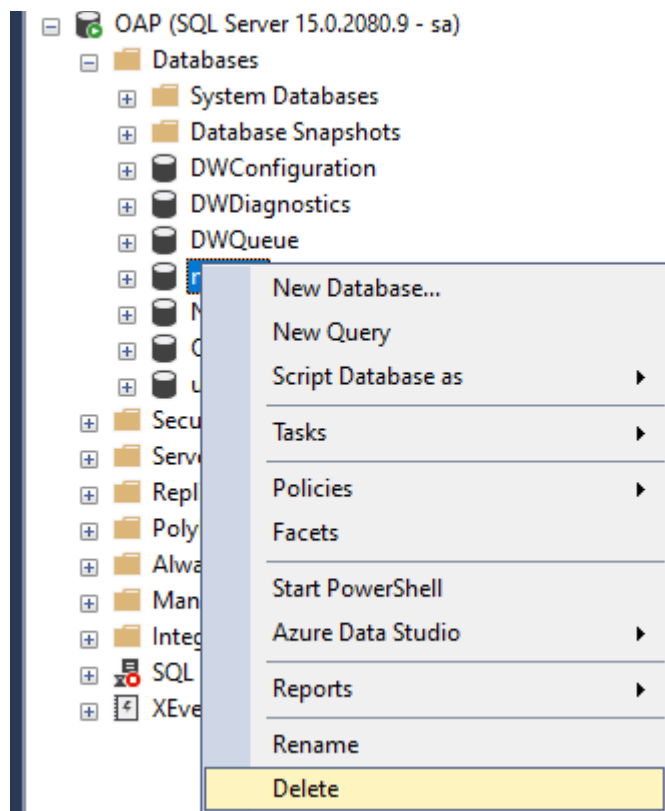


4. Modificar el nombre de la consulta para identificarla posteriormente y elegir una ubicación para almacenarla en el equipo.



Eliminar una base de datos en modo gráfico

1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos de SQL SERVER, a continuación, expándala.
2. Expanda Bases de datos, haga clic con el botón derecho en la base de datos que quiera eliminar y, luego, haga clic en Eliminar (Delete).
3. Confirme que ha seleccionado la base de datos correcta y haga clic en Aceptar.



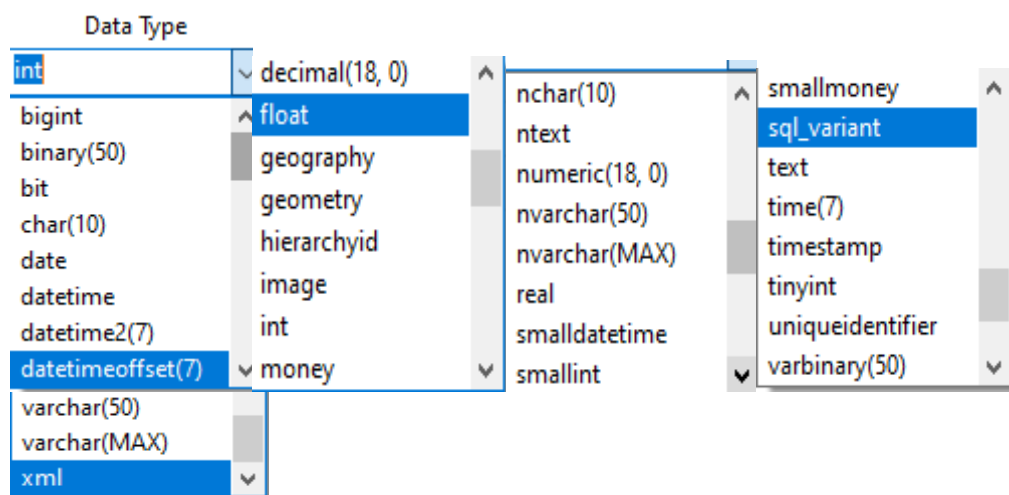
Eliminar una base de datos con sentencia en modo comando

1. Conéctese con el Motor de base de datos.
2. En la barra Estándar, haga clic en Nueva consulta
3. Escriba la sentencia: DROP DATABASE [database_name"]

Ejemplo:

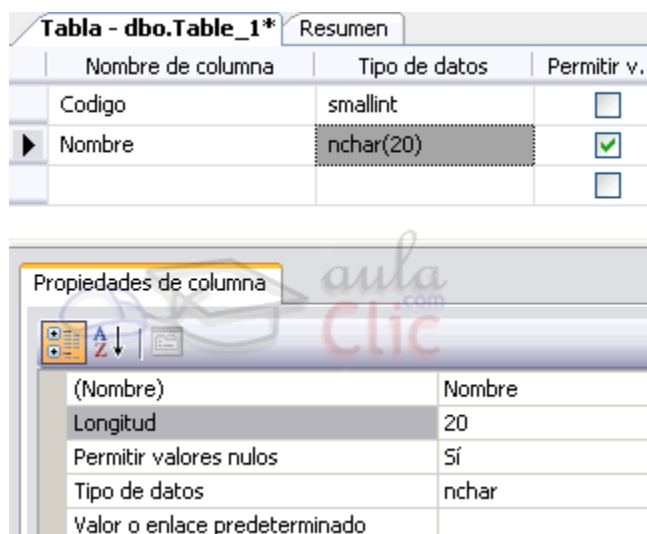
```
USE master ;  
GO  
DROP DATABASE mibase ;  
GO
```

Tipos de datos en SQL SERVER



Podemos elegir entre todos los tipos que se visualizan en la imagen.

Algunos tipos no necesitan más, como por ejemplo el tipo entero (int), y otros se pueden completar con una longitud, como los tipos alfanuméricos:



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 numérico o decimal no tienen la propiedad longitud, pero sí las propiedades escala y precisión, los valores que permiten definir el tamaño del campo.

Definición de tipos de datos en SQL Server 2019

En Microsoft SQL Server, cada columna, variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de datos es un atributo que especifica el tipo de datos que el objeto puede contener: datos de enteros, caracteres, moneda, fecha y hora, cadenas binarias, etc.

SQL Server proporciona un conjunto de tipos de datos del sistema que define todos los tipos de datos que pueden utilizarse con SQL Server. También puede definir sus propios tipos de datos en Transact-SQL o Microsoft .NET Framework. Los tipos de datos de alias están basados en los tipos de datos proporcionados por el sistema.

Tipos de datos en SQL Server 2019

A continuación, mostramos, agrupados por categorías (numéricos exactos, numéricos aproximados, fecha y hora, cadenas de caracteres, cadenas de caracteres Unicode, cadenas binarias, otros tipos de datos).

Números exactos

Tipos de datos numéricos exactos que utilizan datos enteros. Para ahorrar espacio en la base de datos, se debe usar el tipo de datos más pequeño que puede contener todos los valores posibles de manera confiable. Por ejemplo, para un campo que guarde la edad de una persona, sería suficiente con *tinyint*, dado que una persona no vivirá más de 255 años.

Tipo de datos	Intervalo	Almacenamiento
bigint	De -2^{63} (-9.223.372.036.854.775.808) a $2^{63}-1$ (9.223.372.036.854.775.807)	8 bytes
int	De -2^{31} (-2.147.483.648) a $2^{31}-1$ (2.147.483.647)	4 bytes
smallint	De -2^{15} (-32.768) a $2^{15}-1$ (32.767)	2 bytes
tinyint	De 0 a 255	1 byte
numeric[(p[, s])]	<p>Números de precisión y escala fijas. Cuando se utiliza la precisión máxima, los valores válidos se sitúan entre $-10^{38} + 1$ y $10^{38} - 1$. Los sinónimos ISO para decimal son dec y dec(p, s). numeric es funcionalmente idéntico a decimal.</p> <p>p (precisión): el número total máximo de dígitos decimales que se almacenarán. Este número incluye los lados derecho e izquierdo del separador decimal. La precisión debe ser un valor comprendido entre 1 y la precisión máxima de 38. La precisión predeterminada es 18.</p> <p>s (escala): el número de dígitos decimales que se almacenarán a la derecha del separador decimal. Este</p>	<p>Según la precisión:</p> <p>1 – 9: 5 bytes 10-19: 9 bytes 20-28: 13 bytes 29-38: 17 bytes</p>

Tipo de datos	Intervalo	Almacenamiento
	número se extrae de p para determinar el número máximo de dígitos a la izquierda del separador decimal. La escala debe ser un valor comprendido entre 0 y p , y solo se puede especificar si se especifica la precisión. La escala predeterminada es 0, por lo que $0 \leq s \leq p$. Los tamaños de almacenamiento máximo varían según la precisión.	
decimal [(p , s)]	idéntico a numeric [(p , s)]	
money	De -922.337.203.685.477,5808 a 922.337.203.685.477,5807	8 bytes
smallmoney	De – 214.748,3648 a 214.748,3647	4 bytes
bit	Tipo de datos entero que puede aceptar los valores 1, 0 o NULL	1 bit

El tipo de datos **int** es el principal tipo de datos de valores enteros de SQL Server. El tipo de datos **bigint** está pensado para usarse cuando los valores enteros pueden exceder el intervalo admitido por el tipo de datos **int**. **bigint** se encuentra entre **smallmoney** y **int** en el gráfico de prioridad de tipo de datos.

Los tipos de datos **money** y **smallmoney** tienen una precisión de una diezmilésima de las unidades monetarias que representan. En Informática, los tipos de datos **money** y **smallmoney** tienen una precisión de una centésima de las unidades monetarias que representan. Use un punto para separar las unidades parciales de moneda, como céntimos, de las unidades completas de moneda. Por ejemplo, 2.15 puede especificar 2 dólares y 15 centavos.

Tipos de datos numéricos que tienen precisión y escala fijas. Los tipos **decimal** y **numeric** son tipos de datos numéricos de precisión y escala fijas y son sinónimos, se pueden usar indistintamente.

El Motor de base de datos de SQL Server optimiza el almacenamiento de las columnas de tipo **bit**. Si hay menos de ocho columnas de tipo **bit** en una tabla, las columnas se almacenan como 1 byte. Si hay entre 9 y 16 columnas de tipo **bit**, se almacenan como 2 bytes, y así sucesivamente. Los valores de cadena TRUE y FALSE se pueden convertir en valores de tipo **bit**: TRUE se convierte en 1 y FALSE en 0. La conversión a bit promueve cualquier valor distinto de cero a uno.

Números aproximados

Tipos de datos numéricos y aproximados que se utilizan con datos numéricos de coma flotante. Los datos de coma flotante son aproximados; por tanto, no todos los valores del rango del tipo de datos se pueden representar con exactitud. El sinónimo ISO para **real** es **float(24)** .

Tipo de datos	Intervalo	Almacenamiento
float	De $-1,79\text{E}+308$ a $-2,23\text{E}-308$, 0 y de $2,23\text{E}-308$ a $1,79\text{E}+308$	Depende del valor de n
real	De $-3,40\text{E} + 38$ a $-1,18\text{E} - 38$, 0 y de $1,18\text{E} - 38$ a $3,40\text{E} + 38$	4 bytes

Sintaxis para float

float [(n)], donde n es el número de bits que se usan para almacenar la mantisa del número de **float** en notación científica y, por tanto, dicta su precisión y el tamaño de almacenamiento. Si se especifica n , debe ser un valor entre **1** y **53**. El valor predeterminado de n es **53**.

Valor n	Precisión	Almacenamiento
1-24	7 dígitos	4 bytes
25-53	15 dígitos	8 bytes

SQL Server trata n como uno de dos valores posibles. Si $1 \leq n \leq 24$, n se trata como 24. Si $25 \leq n \leq 53$, n se trata como 53.

El tipo de datos SQL Server **float[(n)]** cumple con el estándar ISO para todos los valores de *n* desde **1** hasta **53**. El sinónimo de **double precision** es **float(53)** .

Fecha y hora

Tipo de datos	Intervalo	Almacenamiento
date	De 0001-01-01 a 9999-12-31 (de 1582-10-15 a 9999-12-31 para Informática). Del 1 de enero del año 1 E. C. al 31 de diciembre de 9999 E. C. (del 15 de octubre de 1582 E. C. al 31 de diciembre de 9999 E. C. para Informática)	3 bytes, fijo
datetime	Define una fecha que se combina con una hora del día con fracciones de segundos basada en un reloj de 24 horas. Del 01.01.53 hasta el 31.12.99. De 00:00:00 a 23:59:59,997	8 bytes
smalldatetime	Define una fecha que se combina con una hora del día. La hora está en un formato de día de 24 horas , con segundos siempre a cero (: 00) y sin fracciones de segundo. De 1900-01-01 a 2079-06-06. Del 1 de enero de 1900 hasta el 6 de junio	4 bytes, fijo

Tipo de datos	Intervalo	Almacenamiento
	<p>de 2079.</p> <p>De 00:00:00 a 23:59:59.</p> <p>2007-05-09 23:59:59 se redondeará a 2007-05-10 00:00:00</p>	
datetime2	<p>Define una fecha que se combina con una hora del día basada en un reloj de 24 horas. datetime2 se puede considerar como una extensión del tipo datetime existente que tiene un rango de fechas mayor, un valor predeterminado mayor de precisión fraccionaria y una precisión opcional especificada por el usuario.</p> <p>De 0001-01-01 a 31.12.99.</p> <p>Del 1 de enero del año 1 después de Cristo al 31 de diciembre de 9999.</p> <p>De 00:00:00 a 23:59:59.9999999</p>	<p>De 0 a 7 dígitos, con una precisión de 100 ns. La precisión predeterminada es 7 dígitos</p>
time	<p>Define una hora de un día. La hora no distingue la zona horaria y está basada en un reloj de 24 horas.</p> <p>De 00:00:00.0000000 a 23:59:59.9999999</p>	<p>5 bytes (fijo) es el valor predeterminado con el valor predeterminado de 100 ns de precisión de fracciones de</p>

Tipo de datos	Intervalo	Almacenamiento
	(de 00:00:00.000 a 23:59:59.999 en Informática)	segundo. En Informática, el valor predeterminado es 4 bytes y fijo, con el valor predeterminado de 1 ms de precisión de fracciones de segundo
	Define una fecha que se combina con una hora del día con reconocimiento de zona horaria y basado en un reloj de 24 horas. De 0001-01-01 a 31.12.99	
	Del 1 de enero del año 1 E. C. al 31 de diciembre de 9999 E. C. De 00:00:00 a 23:59:59.9999999.	6 bytes para una precisión inferior a 3. 7 bytes para las precisiones 3 y 4. Todas las demás precisiones
datetimeoffset	De -14:00 a +14:00	requieren 8 bytes

Cadenas de caracteres

Los tipos de datos de caracteres son de tamaño fijo, **char**, o de tamaño variable, **varchar**. A partir de SQL Server 2019 (15.x), cuando se usa una intercalación con UTF-8 habilitado, estos tipos de datos almacenan el intervalo completo de datos de caracteres Unicode y usan la codificación de caracteres UTF-8.

Si se especifica una intercalación cuyo formato no es UTF-8, estos tipos de datos solo almacenan un subconjunto de caracteres admitidos por la página de códigos correspondiente de esa intercalación.

Tipo de datos	Intervalo
char	<p><i>char [(n)]</i> Datos de cadena de tamaño fijo. n define el tamaño de la cadena en bytes y debe ser un valor entre 1 y 8000. Para juegos de caracteres de codificación de byte único, como el latino, el tamaño de almacenamiento es n bytes y el número de caracteres que se pueden almacenar también es n. Para los juegos de caracteres de codificación multibyte, el tamaño de almacenamiento sigue siendo n bytes, pero el número de caracteres que se pueden almacenar puede ser menor que n. El sinónimo ISO para <i>char</i> es <i>character</i></p>
varchar	<p><i>varchar [(n / max)]</i> Datos de cadena de tamaño variable. Utilice n para definir el tamaño de la cadena en bytes, que puede ser un valor comprendido entre 1 y 8000, o bien use max para indicar un tamaño de restricción de columna hasta un almacenamiento máximo de 2³¹-1 bytes (2 GB). Para juegos de caracteres de codificación de byte único, como el latino, el tamaño de almacenamiento es n bytes + 2 bytes y el número de caracteres que se pueden almacenar también es n. Para los juegos de caracteres de codificación multibyte, el tamaño de almacenamiento sigue siendo n bytes + 2 bytes, pero el número de caracteres que se pueden almacenar puede ser menor que n. Los sinónimos ISO para <i>varchar</i> son <i>charvarying</i> o <i>charactervarying</i></p>
text, ntext, image	<p>Tipos de datos de longitud fija y variable para almacenar valores de gran tamaño con datos de caracteres y binarios Unicode y no Unicode. Los datos Unicode utilizan el juego de caracteres UNICODE UCS-2.</p>

Tipo de datos	Intervalo
<p>Una idea equivocada habitual es pensar que en CHAR(n) y VARCHAR(n), la n define el número de caracteres. Pero en CHAR(n) y VARCHAR(n) la n define la longitud de la cadena en bytes (0-8.000). n nunca define números de caracteres que se pueden almacenar. Esto es similar a la definición de NCHAR(n) y NVARCHAR(n). La idea equivocada se produce porque cuando se usa la codificación de un solo byte, el tamaño de almacenamiento de CHAR y VARCHAR es de n bytes y el número de caracteres también es n.</p>	<p>Los tipos de datos ntext, text e image se quitarán en una versión futura de SQL Server. Evite su uso en nuevos trabajos de desarrollo y piense en modificar las aplicaciones que los usan actualmente.</p> <p>ntext: datos Unicode de longitud variable con una longitud máxima de cadena de $2^{30} - 1$ (1.073.741.823) bytes. El tamaño de almacenamiento, en bytes, es dos veces la longitud de cadena especificada. El sinónimo en ISO de ntext es national text.</p> <p>text: datos no Unicode de longitud variable en la página de códigos del servidor y con una longitud máxima de cadena de $2^{31}-1$ (2.147.483.647). Cuando la página de códigos del servidor utiliza caracteres de doble byte, el almacenamiento sigue siendo de 2.147.483.647 bytes. Dependiendo de la cadena de caracteres, el espacio de almacenamiento puede ser inferior a 2.147.483.647 bytes.</p> <p>image: datos binarios de longitud variable desde 0 hasta $2^{31}-1$ (2.147.483.647) bytes</p>

Sin embargo, para la codificación de varios bytes como UTF-8, los intervalos Unicode más altos (128-1.114.111) dan como resultado un carácter que usa dos o más bytes. Por ejemplo, en una columna definida como CHAR(10), Motor de base de datos puede almacenar 10 caracteres que usan la codificación de un solo byte (intervalo de Unicode 0-127), pero menos de 10 caracteres cuando se usa la codificación de varios bytes (intervalo de Unicode 128-1.114.111). Para obtener más información sobre el almacenamiento Unicode y los intervalos de caracteres, vea Diferencias de almacenamiento entre UTF-8 y UTF-16.

Cuando no se especifica *n* en una instrucción de definición de datos o de declaración de variable, la longitud predeterminada es 1. Cuando no se especifica *n* al usar las funciones CAST y CONVERT, la longitud predeterminada es 30.

Los objetos que utilizan **char** o **varchar** se asignan a la intercalación predeterminada de la base de datos, a menos que se asigne una intercalación específica por medio de la cláusula COLLATE. La intercalación controla la página de códigos utilizada para almacenar los datos de caracteres.

Cadenas de caracteres Unicode

Los tipos de datos de caracteres son de tamaño fijo, nchar, o de tamaño variable, nvarchar. A partir de SQL Server 2012 (11.x), cuando se usa una intercalación con carácter complementario (SC) habilitado, estos tipos de datos almacenan el intervalo completo de datos de caracteres Unicode y usan la codificación de caracteres UTF-16. Si se especifica una intercalación que no es de tipo SC, estos tipos de datos almacenan solo el subconjunto de datos de caracteres admitidos por la codificación de caracteres UCS-2.

Tipo de datos	Intervalo
nchar [(<i>n</i>)]	<p>Datos de cadena de tamaño fijo. <i>n</i> define el tamaño de la cadena en pares de bytes y debe ser un valor entre 1 y 4000. El tamaño de almacenamiento es dos veces <i>n</i> bytes. Para la codificación UCS-2, el tamaño de almacenamiento es el doble de <i>n</i> bytes y el número de caracteres que se pueden almacenar también en <i>n</i>. Para la codificación UTF-16, el tamaño de almacenamiento sigue siendo el doble de <i>n</i> bytes, pero el número de caracteres que se pueden almacenar puede ser menor que <i>n</i> porque los caracteres complementarios usan dos pares de bytes (también denominados par suplente). Los sinónimos ISO de nchar son national char y national character.</p>
nvarchar [(<i>n</i> max)]	<p>Datos de cadena de tamaño variable. <i>n</i> define el tamaño de la cadena en pares de bytes y puede ser un valor entre 1 y 4000. max indica que el tamaño máximo de almacenamiento es de 2³⁰-1 caracteres (2 GB). El tamaño de almacenamiento es el doble de <i>n</i> bytes + 2 bytes. Para la codificación UCS-2, el tamaño de almacenamiento es el doble de <i>n</i> bytes + 2 bytes y el número de caracteres que se pueden almacenar también en <i>n</i>. Para la codificación UTF-16, el tamaño de almacenamiento sigue siendo el doble de <i>n</i> bytes + 2 bytes, pero el número de caracteres que se pueden almacenar puede ser menor que <i>n</i> porque los caracteres complementarios usan dos pares de bytes (también denominados par suplente). Los sinónimos ISO de nvarchar son national char varying y national character varying.</p>

**Tipo de
datos**

Intervalo

Cadenas binarias

Tipos de datos binarios de longitud fija o variable.

Tipo de datos

Intervalo

binary [(n)]

Datos binarios de longitud fija con una longitud de n bytes, donde n es un valor que oscila entre 1 y 8000. El tamaño de almacenamiento es de n bytes.

varbinary [(n | max)]

Datos binarios de longitud variable. n puede ser un valor de 1 a 8000. **max** indica que el tamaño máximo de almacenamiento es de $2^{31}-1$ bytes. El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes. Los datos especificados pueden tener una longitud de 0 bytes. El sinónimo de ANSI SQL para **varbinary** es **binary varying**.

image

Datos binarios de longitud variable desde 0 hasta $2^{31}-1$ (2.147.483.647) bytes.

Cuando no se especifica n en una instrucción de definición de datos o de declaración de variables, la longitud predeterminada es 1. Cuando no se especifica n con la función CAST, la longitud predeterminada es 30.

Tipo de datos	Usar cuando...
binary	Los tamaños de las entradas de datos de columna sean coherentes
varbinary	Los tamaños de las entradas de datos de columna varíen considerablemente
varbinary(max)	Las entradas de datos de columna superen los 8000 bytes

Otros tipos de datos

Tipo de datos	Intervalo
hierarchyid	<p>El tipo de datos hierarchyid es un tipo de datos del sistema de longitud variable. Use hierarchyid para representar la posición en una jerarquía. Una columna de tipo hierarchyid no representa automáticamente un árbol. Dependerá de la aplicación generar y asignar los valores hierarchyid de tal forma que la relación deseada entre las filas se refleje en los valores. Un valor del tipo de datos hierarchyid representa una posición en una jerarquía de árbol.</p> <p>Muy compactos: el número medio de bits necesarios para representar un nodo en un árbol con n nodos depende del promedio de distribución ramificada secundarios (el promedio de elementos secundarios de un nodo). Para distribuciones ramificadas pequeñas (0-7), el tamaño es</p>

Tipo de datos

Intervalo

aproximadamente $6 \cdot \log A n$ bits, donde A es el promedio de distribución ramificada. Un nodo en una jerarquía organizativa de 100.000 personas con un promedio de nodos secundarios de 6 niveles supone aproximadamente 38 bits. Esto se redondea a 40 bits (o 5 bytes) para el almacenamiento.

La comparación se realiza con prioridad a la profundidad. Dados dos valores hierarchyid a y b , $a < b$ significa que a viene antes que b en un corte transversal de prioridad a la profundidad del árbol. Los índices de los tipos de datos hierarchyid están en orden con prioridad a la profundidad y los nodos cercanos entre sí en un corte transversal de prioridad a la profundidad se almacenan casi uno junto a otro. Por ejemplo, los elementos secundarios de un registro se almacenan adyacentes a ese registro. Para más información, vea Datos jerárquicos (SQL Server).

Compatibilidad con inserciones y eliminaciones arbitrarias. Con el método GetDescendant siempre es posible generar un miembro del mismo nivel a la derecha de cualquier nodo determinado, a la izquierda de cualquier nodo determinado, o entre dos miembros cualesquiera del mismo nivel. Se mantiene la propiedad comparison cuando se inserta o elimina un número arbitrario de nodos de la jerarquía. La mayoría de las inserciones y eliminaciones conservan la propiedad compactness. Sin embargo, las inserciones entre dos nodos generarán valores hierarchyid con una representación ligeramente menos compacta.

La codificación usada en el tipo **hierarchyid** está limitada a 892 bytes. Por

Tipo de datos	Intervalo
	<p>consiguiente, el tipo hierarchid no podrá representar los nodos con demasiados niveles en su representación como para caber en los 892 bytes.</p>
uniqueidentifier	<p>Es un GUID de 16 bytes. Una columna o una variable local de tipo de datos uniqueidentifier se puede inicializar en un valor de las siguientes formas:</p> <p>Mediante el uso de las funciones NEWID o NEWSEQUENTIALID.</p> <p>Mediante la conversión a partir de una constante de cadena con el formato <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code>, donde cada <i>x</i> es un dígito hexadecimal en el intervalo 0-9 o a-f.</p> <p>Con los valores uniqueidentifier se pueden usar operadores de comparación. No obstante, no se implementa la ordenación mediante la comparación de los patrones de bits de los dos valores. Las únicas operaciones que se pueden realizar con un valor uniqueidentifier son comparaciones (=, <>, <, >, <=, >=) y comprobaciones para NULL (IS NULL e IS NOT NULL). No es posible utilizar otros operadores aritméticos.</p> <p>Con el tipo de datos uniqueidentifier se pueden usar todas las propiedades y restricciones de columna, excepto IDENTITY.</p>
sql_variant	<p>Tipo de datos que almacena valores de varios tipos de datos admitidos en SQL Server.</p> <p>sql_variant puede usarse en columnas, parámetros, variables y los valores devueltos de las funciones definidas por el usuario. sql_variant permite que</p>

Tipo de datos	Intervalo
xml	<p>estos objetos de base de datos admitan valores de otros tipos de datos. Una columna de tipo sql_variant puede contener filas de tipos de datos diferentes. Por ejemplo, una columna definida como sql_variant puede almacenar valores int, binario y char.</p> <p>sql_variant puede tener una longitud máxima de 8016 bytes. Esto incluye la información y el valor de tipo base. La longitud máxima del tipo base real es 8.000 bytes.</p>
Tipos de geometría espacial	<p>El tipo de datos espaciales planares, geometry, se implementa como un tipo de datos CLR (Common Language Runtime) en SQL Server. Este tipo representa datos en un sistema de coordenadas euclídeo (plano). SQL Server admite un conjunto de métodos para el tipo de datos espaciales geography. Este métodos incluyen a su vez métodos de geometry definidos por el estándar Open Geospatial Consortium (OGC) y un conjunto de extensiones de Microsoft para dicho estándar. La tolerancia de error para los métodos geometry puede ser de hasta $1,0e-7$ * extensiones. Las extensiones hacen referencia a la distancia máxima aproximada entre los puntos del objeto geometry.</p> <p>El tipo geometry está predefinido y está disponible en cada base de datos. Puede crear columnas de tabla de tipo geometry y operar con los</p>

Tipo de datos	Intervalo
	<p>datos geometry de la misma manera que con los demás tipos CLR. Se puede utilizar en columnas calculadas persistentes y no persistentes.</p>
Tipos de geografía espacial	<p>El tipo de datos de geografía espacial, geography, se implementa como un tipo de datos de .NET CLR (Common Language Runtime) en SQL Server. Este tipo representa los datos en un sistema de coordenadas de tierra redonda. El tipo de datos SQL Server geography almacena datos elipsoidales (globo), como coordenadas de latitud y longitud de GPS. SQL Server admite un conjunto de métodos para el tipo de datos espacial geography. Se incluyen los métodos de geography definidos por el estándar Open Geospatial Consortium (OGC) y un conjunto de extensiones de Microsoft a dicho estándar.</p> <p>La tolerancia de error para los métodos geography puede ser de hasta 1,0e-7 * extensiones. Las extensiones hacen referencia a la distancia máxima aproximada entre los puntos del objeto geography.</p> <p>El tipo geography está predefinido y está disponible en cada base de datos. Puede crear columnas de tabla de tipo geography y operar con los datos geography de la misma manera que con los demás tipos proporcionados por el sistema. Se puede utilizar en columnas calculadas persistentes y no persistentes.</p>

Crear una tabla en modo gráfico

1. En SSMS, en el Explorador de objetos, conéctese a la instancia de Motor de base de datos que contiene la base de datos que se va a modificar.
2. En el Explorador de objetos, expanda el nodo Bases de datos y, a continuación, expanda la base de datos que contendrá la nueva tabla.
3. En el Explorador de objetos, haga clic con el botón derecho en el nodo Tablas de la base de datos y, después, haga clic en Nueva tabla.
4. Escriba los nombres de columna, elija los tipos de datos y elija si desea permitir valores NULL para cada columna como se muestra en la ilustración siguiente:

OAP.universidad - dbo.Alumno* ↗ ✕		OAP.universidad - dbo.mitabla	
	Column Name	Data Type	Allow Nulls
🔑	CodAlumno	int	<input type="checkbox"/>
	Cedula	varchar(50)	<input type="checkbox"/>
	Nombre	varchar(100)	<input type="checkbox"/>
	Direccion	varchar(150)	<input type="checkbox"/>
▶	Telefono	tinyint	<input checked="" type="checkbox"/>
	Email	varchar(100)	<input type="checkbox"/>
	FechaNacimiento	date	<input type="checkbox"/>
	Edad	tinyint	<input type="checkbox"/>
			<input type="checkbox"/>

- Para especificar más propiedades para una columna, como la identidad o valores de columna calculada, haga clic en la columna y después, en la pestaña de propiedades de la columna, elija las propiedades adecuadas.

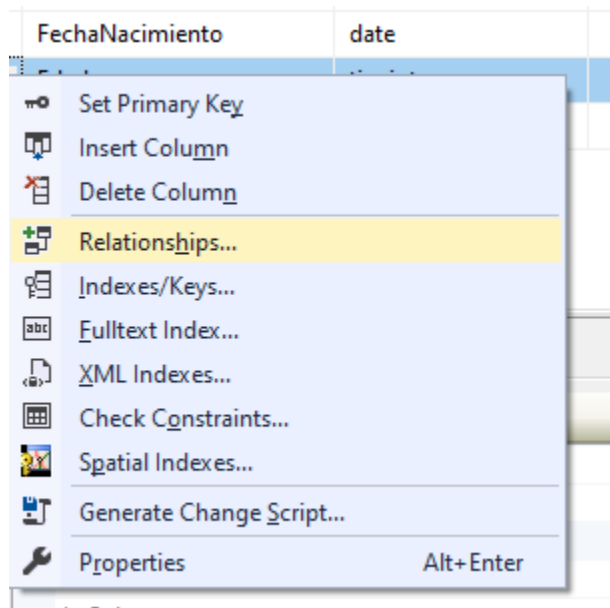
OAP.universidad - dbo.Alumno* X OAP.universidad - dbo.mi			
	Column Name	Data Type	Allow Null
🔔	CodAlumno	int	<input type="checkbox"/>
	Cedula	varchar(50)	<input type="checkbox"/>
	Nombre	varchar(100)	<input type="checkbox"/>
	Direccion	varchar(150)	<input type="checkbox"/>
	Telefono	tinyint	<input checked="" type="checkbox"/>
	Email	varchar(100)	<input type="checkbox"/>
	FechaNacimiento	date	<input type="checkbox"/>
	Edad	tinyint	<input type="checkbox"/>
			<input type="checkbox"/>

Column Properties	
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
> Full-text Specification	No
Has Non-SQL Server Subscriber	No
▼ Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

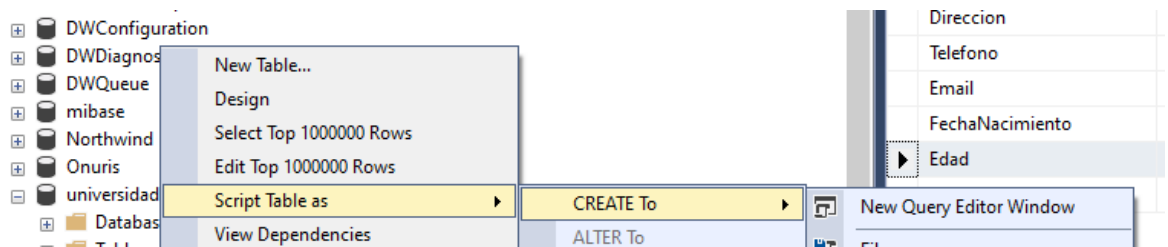
- Para especificar una columna como clave principal, haga clic con el botón derecho en la columna y seleccione Establecer clave principal. Para obtener más información, consulte Create Primary Keys.

OAP.universidad - dbo.Alumno* X OAP.universidad	
Column Name	Data Type
CodAlumno	int
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
Set Primary Key	
Insert Column	
Delete Column	
Relationships...	
Indexes/Keys...	
Fulltext Index...	
XML Indexes...	
Check Constraints...	
Spatial Indexes...	
Generate Change Script...	
Properties Alt+Enter	

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



Duplicar una tabla a partir de una tabla existente usando el editor de consultas



1. Expanda las tablas de la base de datos, haga click derecho sobre la tabla, elija la opción **Script Table as**.
2. Luego elija la opción **CREATE to**
3. Elija la opción **New Query Editor Windows**
4. Elija los campos que desea reutilizar para la nueva tabla, cambie el nombre de la nueva tabla, y finalmente ejecute la consulta.

```
SQLQuery5.sql - OAP...iversidad (sa (54))* X SQLQuery4
USE [universidad]
GO

/***** Object: Table [dbo].[mitabla]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[minuevatabla](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NOT NULL,
    [Cedula] [varchar](50) NOT NULL,
)
```

Duplicar una tabla desde modo gráfico

Para duplicar una tabla

1. Asegúrese de que está conectado a la base de datos en la que desea crear la tabla y de que la base de datos está seleccionada en el Explorador de objetos.
2. En el Explorador de objetos, haga clic con el botón derecho en Tablas y, luego, haga clic en Nueva tabla.
3. En el Explorador de objetos, haga clic con el botón derecho en la tabla que quiera copiar y, después, haga clic en Diseño.
4. Seleccione las columnas de la tabla existente y haga clic en Copiar en el menú Edición.
5. Vuelva a la ventana nueva y seleccione la primera fila.
6. En el menú Edición, haga clic en Pegar.
7. En el menú Archivo, haga clic en Guardar nombre de tabla.
8. En el cuadro de diálogo Elegir nombre, escriba un nombre para la tabla nueva y haga clic en Aceptar.

Crear una tabla con sentencia en modo comando (agregar columnas, tipos de datos ,clave primaria, identity)

1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos.
2. En la barra de Estándar, haga clic en Nueva consulta.
3. Escriba la consulta de creación de los campos y propiedad en la ventana de consulta y haga clic en Ejecutar.

```
use universidad
]create table Alumno (
  CodAlumno int identity (1,1) primary key,
  Cedula varchar(50) not null,
  Nombre varchar(100)not null,
  Direccion varchar(150) not null,
  Telefono tinyint not null,
  Email varchar(100) not null,
  FechaNacimiento date not null
.)
```

Eliminar una tabla en modo gráfico

Limitaciones y restricciones

No se puede eliminar una tabla a la que haga referencia una restricción FOREIGN KEY. Primero se debe quitar la restricción FOREIGN KEY o la tabla de referencia. Si la tabla de referencia y la tabla que tiene la clave principal se van a quitar en la misma instrucción DROP TABLE, la tabla de referencia debe aparecer primero.

Cuando se quita la tabla, las reglas o valores predeterminados de la tabla pierden sus enlaces y se quitan automáticamente las restricciones o desencadenadores asociados con la tabla. Si vuelve a crear una tabla, debe volver a enlazar las reglas y valores predeterminados apropiados, volver a crear los desencadenadores y agregar todas las restricciones necesarias.

Si quita una tabla que contenga una columna de tipo varbinary (max) con el atributo FILESTREAM, los datos almacenados en el sistema de archivos no se quitarán.

DROP TABLE y CREATE TABLE no se deberían ejecutar en la misma tabla en el mismo lote. De lo contrario, podría producirse un error inesperado.

Las vistas o procedimientos almacenados que hagan referencia a la tabla quitada se deben eliminar o modificar explícitamente para quitar la referencia a la tabla.

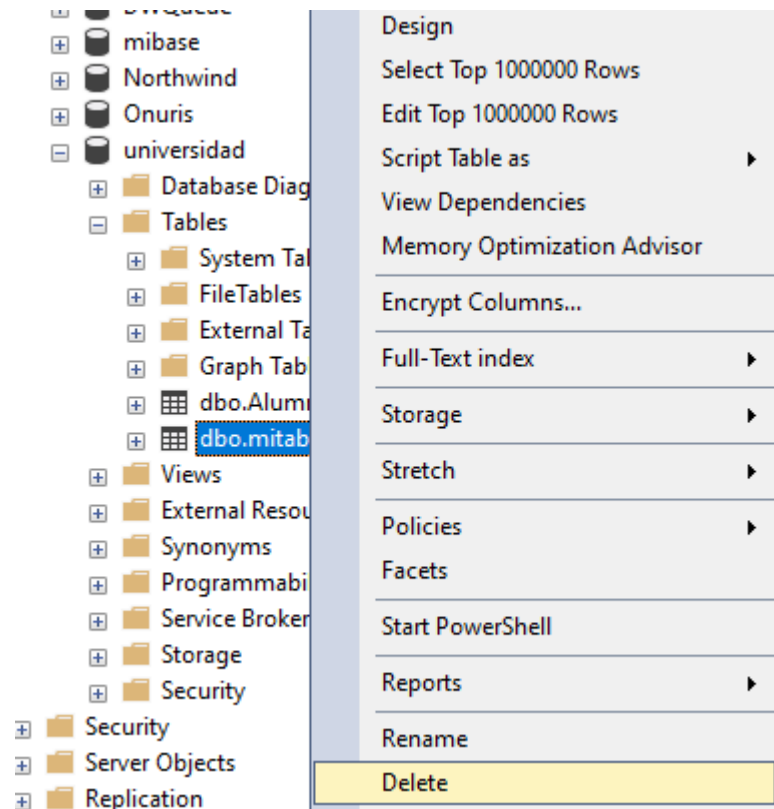
Seguridad

Permisos

Se requiere el permiso ALTER en el esquema al que pertenece la tabla, el permiso CONTROL en la tabla o la pertenencia al rol fijo de base de datos db_ddladmin.

Para eliminar una tabla de la base de datos en modo gráfico:

1. En el Explorador de objetos, seleccione la tabla que desea eliminar.
2. Haga clic con el botón derecho en la tabla y elija Eliminar en el menú contextual.
3. Un cuadro de mensaje le pedirá que confirme la eliminación. Haga clic en Sí.



Eliminar una tabla con sentencia en modo comando

Para eliminar una tabla en el Editor de consultas

1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos.
2. En la barra de Estándar, haga clic en Nueva consulta.
3. Digite la siguiente consulta en la ventana de consulta y haga clic en Ejecutar.

DROP TABLE [database_name]

Ejemplo: DROP TABLE mibase;

Alter table

Agregar columnas a una tabla con el comando Alter table:

Limitaciones y restricciones

Al usar la instrucción ALTER TABLE para agregar columnas a una tabla, se agregan automáticamente las columnas al final de la tabla. Si desea que las columnas aparezcan en un orden concreto en la tabla, use SQL Server Management Studio. Sin embargo, tenga en cuenta que esto no es un procedimiento recomendado del diseño de base de datos. El procedimiento recomendado es especificar el orden en que las columnas se devuelven en el nivel de aplicación y de consulta. No debe confiar en el uso de SELECT * para devolver todas las columnas en un orden esperado según el orden en que están definidos en la tabla. Especifique siempre las columnas por nombre en las consultas y aplicaciones en el orden en que desea que aparezcan.

Seguridad

Permisos

Requiere el permiso ALTER en la tabla.

Ejemplo: en este ejemplo se agregan dos nuevas columnas a la tabla Alumno

```
ALTER TABLE dbo.Alumno ADD column_b VARCHAR (20) NULL, column_c INT NULL;
```

Eliminar columnas a una tabla con el comando Alter table:

Limitaciones y restricciones

No puede eliminar una columna que tenga una restricción CHECK. Primero debe eliminar la restricción.

No puede eliminar una columna que tiene restricciones PRIMARY KEY o FOREIGN KEY u otras dependencias excepto si usa el Diseñador de tablas. Al utilizar el Explorador de objetos o Transact-SQL, primero debe quitar todas las dependencias de la columna.

Seguridad

Permisos

Requiere el permiso ALTER en la tabla.

En el ejemplo siguiente se muestra cómo eliminar una columna de la tabla Alumno:

```
ALTER TABLE dbo.Alumno DROP COLUMN column_b;
```

Si la columna contiene restricciones u otras dependencias, se devolverá un mensaje de error. Resuelva el error eliminando las restricciones a las que hace referencia.

Cambiar el nombre de columnas

Limitaciones y restricciones

Cambiar el nombre de una columna automáticamente no cambiará las referencias a esa columna. Es necesario modificar de forma manual los objetos que hacen referencia a la columna cuyo nombre se ha cambiado. Por ejemplo, si se cambia el nombre de una columna de una tabla y en un desencadenador existe una referencia a esa columna, es necesario modificar el desencadenador para reflejar el nuevo nombre de la columna

Seguridad

Permisos

Requiere el permiso ALTER en el objeto.

Para cambiar el nombre de una columna mediante el Explorador de objetos (modo gráfico)

1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos.
2. En el Explorador de objetos, haga clic con el botón derecho en la tabla en la que quiere cambiar nombres de columnas y elija Cambiar nombre.
3. Escriba un nuevo nombre de columna.

Para cambiar el nombre de una columna mediante el Diseñador de tablas

1. En el Explorador de objetos, haga clic con el botón derecho en la tabla en la que quiere cambiar nombres de columnas y elija Diseño.
2. En Nombre de columna, seleccione el nombre que desea cambiar y escriba uno nuevo.
3. En el menú Archivo, haga clic en *Guardar nombre de tabla.

Para cambiar el nombre de una columna desde el editor de consultas Transact-SQL

En el ejemplo siguiente se cambia el nombre de la columna **Cedula** de la tabla **Alumno** a **Documento** en la base de datos de **Universidad**.

Use Universidad;

GO

EXEC sp_rename 'Alumno.Cedula', 'Documento', 'COLUMN';

Para modificar el tipo de datos de una columna modo gráfico

1. En el Explorador de objetos, haga clic con el botón derecho en la tabla que contenga columnas cuya escala quiera cambiar y, después, haga clic en Diseño.
2. Seleccione la columna en la que desea modificar el tipo de datos.
3. En la pestaña Propiedades de columna, haga clic en la celda de la cuadrícula de la propiedad Tipo de datos y elija un tipo de datos en la lista desplegable.
4. En el menú Archivo, haga clic en *Guardar nombre de tabla.

Para modificar el tipo de datos de una columna desde editor de consultas

1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos.
2. En la barra de Estándar, haga clic en Nueva consulta.
3. Digite la siguiente sentencia en la ventana de consulta y haga clic en Ejecutar.

En este ejemplo se modifica el tipo de datos del campo Documento de VARCHAR a INT en la tabla Alumno.

```
Use Universidad;  
GO  
ALTER TABLE dbo.Alumno ALTER COLUMN Documento INT;
```

Para cambiar el orden de las columnas

1. En el Explorador de objetos, haga clic con el botón derecho en la tabla cuyas columnas quiera reordenar y seleccione Diseñar.
2. Seleccione la casilla que hay a la izquierda del nombre de la columna cuyo orden desea cambiar.
3. Arrastre la columna a otra ubicación en la tabla.

Con editor de consultas- Transact-SQL
Para cambiar el orden de las columnas

***Esta tarea no es compatible con las instrucciones Transact-SQL en SQL SERVER.**

Integridad referencial en la definición de claves foráneas y generación del modelo relacional (modo gráfico y Transact-SQL).

Una clave foránea o foreign key es un campo (o campos) que señala la clave primaria de otra tabla. El propósito de la clave foránea es asegurar la integridad referencial de los datos. La columna o columnas señaladas como llave foránea, solo podrán tener valores que ya existan en la clave primaria PRIMARY KEY de la otra tabla.

La clave foránea en resumen se refiere a una columna o varias columnas, que sirven para señalar cual es la clave primaria de otra tabla. Es importante que comprendas que hay dos etapas en las que podemos agregar llaves foráneas a las tablas: la primera es al momento de crear la tabla en Sql Server, ya que podemos definir de una vez las llaves foráneas y campos de la tabla de nuestra Base de Datos; la segunda opción es agregar las llaves foráneas en tablas ya creadas, esto consiste en modificar la estructura previamente diseñada.

Crear relaciones con claves foráneas

Permisos

La creación de una tabla nueva con una clave externa requiere el permiso CREATE TABLE en la base de datos y el permiso ALTER en el esquema en el que se crea la tabla.

La creación de una clave externa en una tabla existente requiere el permiso ALTER en la tabla.

Límites y restricciones

Una restricción de clave externa no tiene que estar vinculada solo a una restricción de clave principal en otra tabla. Las claves externas también se pueden definir para hacer referencia a las columnas de una restricción UNIQUE de otra tabla.

Si se especifica un valor distinto de NULL en la columna de una restricción FOREIGN KEY, el valor debe existir en la columna a la que se hace referencia. De lo contrario, se devuelve un mensaje de error de infracción de clave externa. Para asegurarse de que todos los valores de la restricción de clave externa compuesta se comprueben, especifique NOT NULL en todas las columnas que participan.

Las restricciones FOREIGN KEY solo pueden hacer referencia a las tablas de la misma base de datos en el mismo servidor. La integridad referencial entre bases de datos debe implementarse a través de desencadenadores. Para obtener más información, vea CREATE TRIGGER.

Las restricciones FOREIGN KEY pueden hacer referencia a otra columna de la misma tabla, lo que se conoce como autorreferencia.

Una restricción FOREIGN KEY especificada en el nivel de columna solo puede incluir una columna de referencia. Esta columna debe tener el mismo tipo de datos que la columna en la que se define la restricción.

Una restricción FOREIGN KEY especificada en el nivel de tabla debe tener el mismo número de columnas de referencia que la lista de columnas de la restricción. El tipo de datos de cada columna de referencia debe ser también el mismo que el de la columna correspondiente de la lista de columnas.

El Motor de base de datos no tiene un límite predefinido en el número de restricciones FOREIGN KEY que puede contener una tabla y que hacen referencia a otras tablas. El Motor de base de datos tampoco limita el número de restricciones FOREIGN KEY que pertenecen a otras tablas y que hacen referencia a una tabla concreta. No obstante, el número real de restricciones FOREIGN KEY utilizadas está limitado por la configuración del hardware y por el diseño de la base de datos y de la aplicación. Una tabla puede hacer referencia a otras 253 tablas y columnas como claves externas (referencias de salida) como máximo. SQL Server 2016 (13.x) y versiones posteriores aumentan el límite para la cantidad de otras tablas y columnas que pueden hacer referencia a las columnas de una sola tabla (referencias de entrada) de 253 a 10 000. (Requiere al menos el nivel de compatibilidad 130). El aumento tiene las siguientes restricciones:

Se admiten más de 253 referencias de clave externa para las operaciones DELETE y UPDATE DML. Las operaciones MERGE no se admiten.

Una tabla con una referencia de clave externa a sí misma seguirá limitada a 253 referencias de clave externa.

Actualmente, no hay disponibles más de 253 referencias de clave externa para índices de almacén de columnas, tablas optimizadas para memoria o Stretch Database.

Las restricciones FOREIGN KEY no se exigen en tablas temporales.

Si la clave externa se define en una columna de tipo definido por el usuario CLR, la implementación del tipo debe admitir el orden binario. Para obtener más información, vea Tipos definidos por el usuario de CLR.

Una columna de tipo varchar(max) solo puede participar en una restricción FOREIGN KEY si la clave principal a la que hace referencia se define también como tipo varchar(max) .

Crear una relación de clave externa en el Diseñador de tablas

Uso de SQL Server Management Studio

1. En el Explorador de objetos, haga clic con el botón derecho en la tabla que va a estar en el lado de la clave externa de la relación y, después, haga clic en Diseño.

La tabla se abre en el Diseñador de tablas.

2. En el menú Diseñador de tablas, haga clic en Relaciones.

3. En el cuadro de diálogo Relaciones de clave externa, haga clic en Agregar.

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.

4. Haga clic en la relación en la lista Relación seleccionada.
5. Haga clic en Especificaciones de tablas y columnas en la cuadrícula situada a la derecha y, después, haga clic en los puntos suspensivos (...) situados a la derecha de la propiedad.
6. En el cuadro de diálogo Tablas y columnas, en la lista desplegable Clave principal, elija la tabla que estará en el lado de la clave principal de la relación.
7. En la cuadrícula situada debajo, elija las columnas que contribuyen a la clave principal de la tabla. En la celda de la cuadrícula adyacente situada a la derecha de cada columna, elija la columna de clave externa correspondiente de la tabla de clave externa.

El Diseñador de tablas sugerirá un nombre para la relación. Para cambiar este nombre, edite el contenido del cuadro de texto Nombre de la relación.

8. Elija Aceptar para crear la relación.
9. Cierre la ventana del diseñador de tablas y guarde los cambios para que el cambio de relación de clave externa surta efecto.

Ejercicio:

Vamos a crear cuatro tablas en Sql Server para hacer las relaciones respectivas y agregar las claves primarias y foráneas en la Base de Datos. Las tablas a crear son: Alumnos, Asignatura, profesor e Inscripción. Las primeras tres tablas tendrán únicamente las claves primarias y la tabla inscripción tendrá las claves foráneas que estarán relacionadas con las llaves primarias (campos Id) de las tablas Alumnos, Asignatura y Profesor.

```
create database Universidad
on primary
(Name=Universidaddata, filename='D:\data\Universidad.mdf' --Master Database Files
,size=50mb
,Filegrowth=25%
)
log on
(Name=UniversidadLog, filename='D:\data\Universidad.ldf' --Log Database Files
,size=25mb
,Filegrowth=25%
)
```

GO

Use Universidad

Go

/* Creando tabla Alumno, asignatura y profesor */

```
Create table Alumno(
IdAlumno int primary key identity(1,1),
NombreAlumno varchar(50) not null,
ApellidoAlumno varchar(50) not null,
Direccion varchar(50),
Fecha_nacimiento date
);
```

```
Create table Asignatura(
IdAsignatura int primary key identity(1,1),
NombreAsignatura varchar(20) not null,
Creditos int not null
);
```

```
Create table Profesor(
IdProfesor int primary key identity(1,1),
NombreProfesor varchar(50) not null,
ApellidoProfesor varchar(50) not null,
Direccion varchar(50),
Fecha_nacimiento date,
Nivel_Academico int not null
);
```

/* Creando tabla Inscripción con campos para llaves foraneas */

```
Create table Inscripcion(  
  IdInscripcion int primary key identity(1,1),  
  IdAsignatura int not null,  
  IdAlumno int not null,  
  IdProfesor int not null,  
  Fecha date,  
  CONSTRAINT fk_Alumno FOREIGN KEY (IdAlumno) REFERENCES Alumno (IdAlumno),  
  CONSTRAINT fk_Asignatura FOREIGN KEY (IdAsignatura) REFERENCES Asignatura (IdAsignatura),  
  CONSTRAINT fk_Profesor FOREIGN KEY (IdProfesor) REFERENCES Profesor (IdProfesor)  
);
```

Agregar llave primaria en tabla ya creada

Supongamos que es necesario modificar tablas existentes y agregar llaves primarias o foráneas. Con el siguiente código modificamos la tabla profesor para agregar su clave primaria:

```
ALTER TABLE Profesor ADD CONSTRAINT pk_Profesor_Id PRIMARY KEY  
(IdProfesor)
```

Agregar clave foránea sql a Tabla ya creada

Supongamos que la tabla Profesor no ha sido relacionada con la tabla Inscripción, entonces en caso de no existir dicha clave foránea, la podemos agregar con el siguiente código Sql:

```
ALTER TABLE Inscripcion
```

```
ADD FOREIGN KEY (IdProfesor) REFERENCES Profesor(IdProfesor);
```