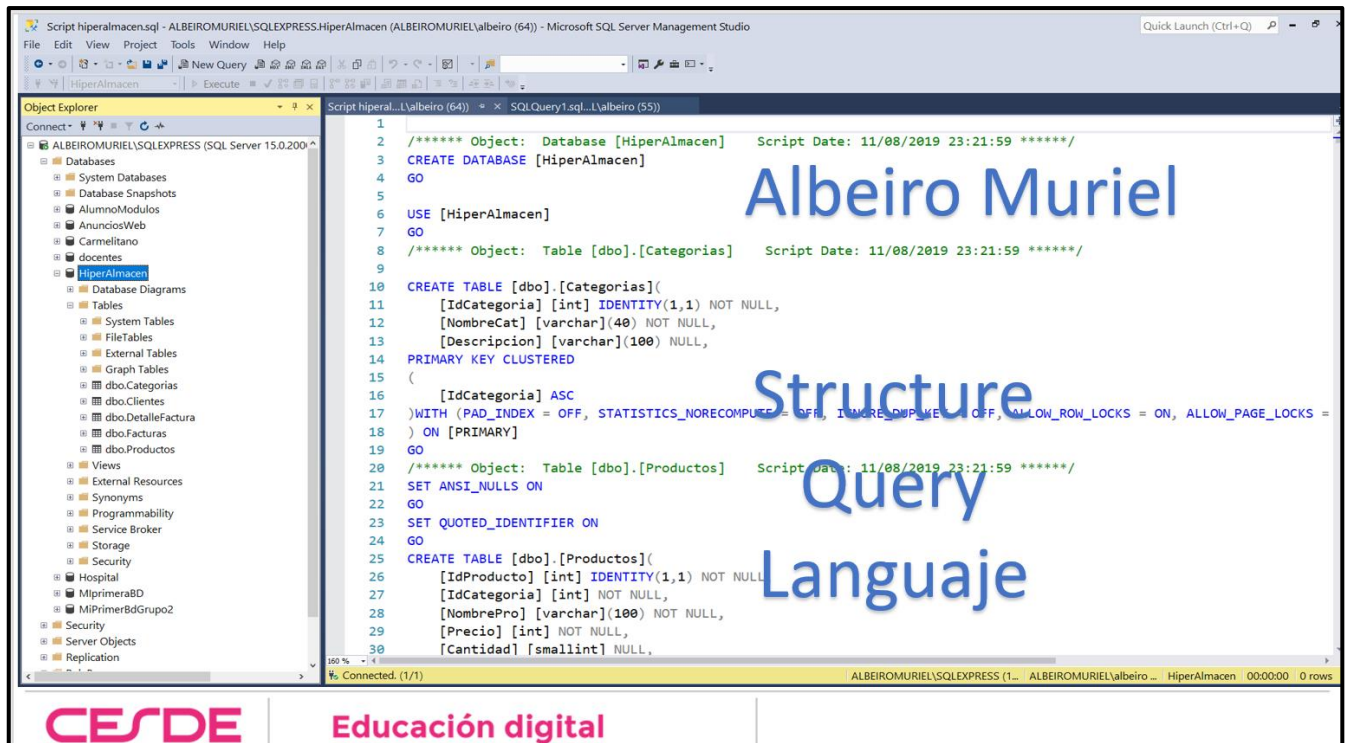


SQL



STRUCTURED QUERY LANGUAGE

STRUCTURED QUERY LANGUAGE

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

EXISTEN TRES TIPOS DE COMANDOS SQL:

Los DLL(Data Definition Language) que permiten crear y definir nuevas bases de datos, campos e índices. Los DML(Data Manipulation Language) que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos. Los DCL(Data Control Language) que se encargan de definir las permisos sobre los datos

Lenguaje de Definición de Datos (DDL)

El lenguaje de definición de datos (en inglés Data Definition Language, o DDL), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

Comando	Descripción
---------	-------------

CREATE	Crear tablas, campos e índices
--------	--------------------------------

DROP	Eliminar tablas e índices
------	---------------------------

ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
-------	---

TRUNCATE Este comando borra todo el contenido de una tabla. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE.

Lenguaje de manipulación de datos (DML)

Lenguaje de Manipulación de Datos, por sus siglas en Inglés, Data Manipulation Language (DML).

Es el que permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos. Existen cuatro operaciones básicas: SELECT, INSERT, UPDATE y DELETE.

Comando	Descripción
---------	-------------

SELECT	Muestra los registros de las tablas
--------	-------------------------------------

INSERT	Insertar o agregar registros a las tablas.
--------	--

UPDATE	Para modificar los registros de las tablas.
--------	---

DELETE	Elimina todos los registros de la tabla
--------	---

Lenguaje de Control de Datos (DCL)

Permite crear roles, permisos e integridad referencial, así como el control al acceso a la base de datos.

Comando	Descripción
---------	-------------

GRANT	Usado para otorgar privilegios de acceso de usuario a la base de datos.
-------	---

REVOKE	Utilizado para retirar privilegios de acceso otorgados con el comando GRANT
--------	---

TIPOS DE DATOS EN SQL SERVER

Ya explicamos que al crear una tabla debemos elegir la estructura adecuada, esto es, definir los campos y sus tipos más precisos, según el caso.

Para almacenar valores NUMERICOS SQL Server dispone de varios tipos.

Para almacenar valores ENTEROS, por ejemplo, en campos que hacen referencia a cantidades, usamos:

integer o int: Su rango es de -2000000000 a 2000000000 aprox. El tipo "integer" tiene subtipos:

smallint: Puede contener hasta 5 dígitos. Su rango va desde -32000 hasta 32000 aprox.

tinyint: Puede almacenar valores entre 0 y 255.

bigint: De -9000000000000000000 hasta 9000000000000000000 aprox.

Para almacenar valores numéricos EXACTOS con decimales, especificando la cantidad de cifras a la izquierda y derecha del separador decimal, utilizamos:

decimal o numeric (t,d):

Pueden tener hasta 38 dígitos, guarda un valor exacto.

El primer argumento indica el total de dígitos y el segundo, la cantidad de decimales.

Por ejemplo.

Si queremos almacenar valores entre -99.99 y 99.99 debemos definir el campo como tipo "decimal(4,2)".

Si no se indica el valor del segundo argumento, por defecto es "0".

Por ejemplo, si definimos "decimal(4)" se pueden guardar valores entre -9999 y 9999.

El rango depende de los argumentos, también los bytes que ocupa.

Se utiliza el punto como separador de decimales.

Si ingresamos un valor con más decimales que los permitidos, redondea al más cercano; por ejemplo, si definimos "decimal(4,2)" e ingresamos el valor "12.686", guardará "12.69", redondeando hacia arriba; si ingresamos el valor "12.682", guardará "12.67", redondeando hacia abajo.

numéricos APROXIMADOS

Para almacenar valores numéricos APROXIMADOS con decimales utilizamos:

float y real: De $1.79E+308$ hasta $1.79E+38$. Guarda valores aproximados.

real: Desde $3.40E+308$ hasta $3.40E+38$. Guarda valores aproximados.

Para almacenar valores MONETARIOS empleamos:

money: Puede tener hasta 19 dígitos y sólo 4 de ellos puede ir luego del separador decimal; entre -9000000000000000.5808 aprox y 9000000000000000.5807 .

smallmoney: Entre -200000.3648 y 200000.3647 aprox.

Para todos los tipos numéricos:

- si intentamos ingresar un valor fuera de rango, no lo permite.
- si ingresamos una cadena, SQL Server intenta convertirla a valor numérico, si dicha cadena consta solamente de dígitos, la conversión se realiza, luego verifica si está dentro del rango, si es así, la ingresa, sino, muestra un mensaje de error y no ejecuta

la sentencia. Si la cadena contiene caracteres que SQL Server no puede convertir a valor numérico, muestra un mensaje de error y la sentencia no se ejecuta.

Por ejemplo, definimos un campo de tipo decimal(5,2), si ingresamos la cadena '12.22', la convierte al valor numérico 12.22 y la ingresa; si intentamos ingresar la cadena '1234.56', la convierte al valor numérico 1234.56, pero como el máximo valor permitido es 999.99, muestra un mensaje indicando que está fuera de rango. Si intentamos ingresar el valor '12 y .25', SQL Server no puede realizar la conversión y muestra un mensaje de error.

Es importante elegir el tipo de dato adecuado según el caso, el más preciso.

Por ejemplo, si un campo numérico almacenará valores positivos menores a 255, el tipo "int" no es el más adecuado, conviene el tipo "tinyint", de esta manera usamos el menor espacio de almacenamiento posible.

Si vamos a guardar valores monetarios menores a 200000 conviene emplear "smallmoney" en lugar de "money".

-- Intentemos ingresar un valor fuera del rango definido, una cantidad

-- que supera el rango del tipo "tinyint", el valor 260 (genera error):

-- Intentamos ingresar un precio que supera el rango del tipo "smallmoney",

-- el valor 250000 (genera error):

-- Intentamos ingresar una cadena que SQL Server no pueda convertir a valor

-- numérico en el campo "precio" (genera error):

-- Ingresamos una cadena en el campo "cantidad" (lo convierte a valor numérico):

Tomado de

<http://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=24&punto=&inicio=>



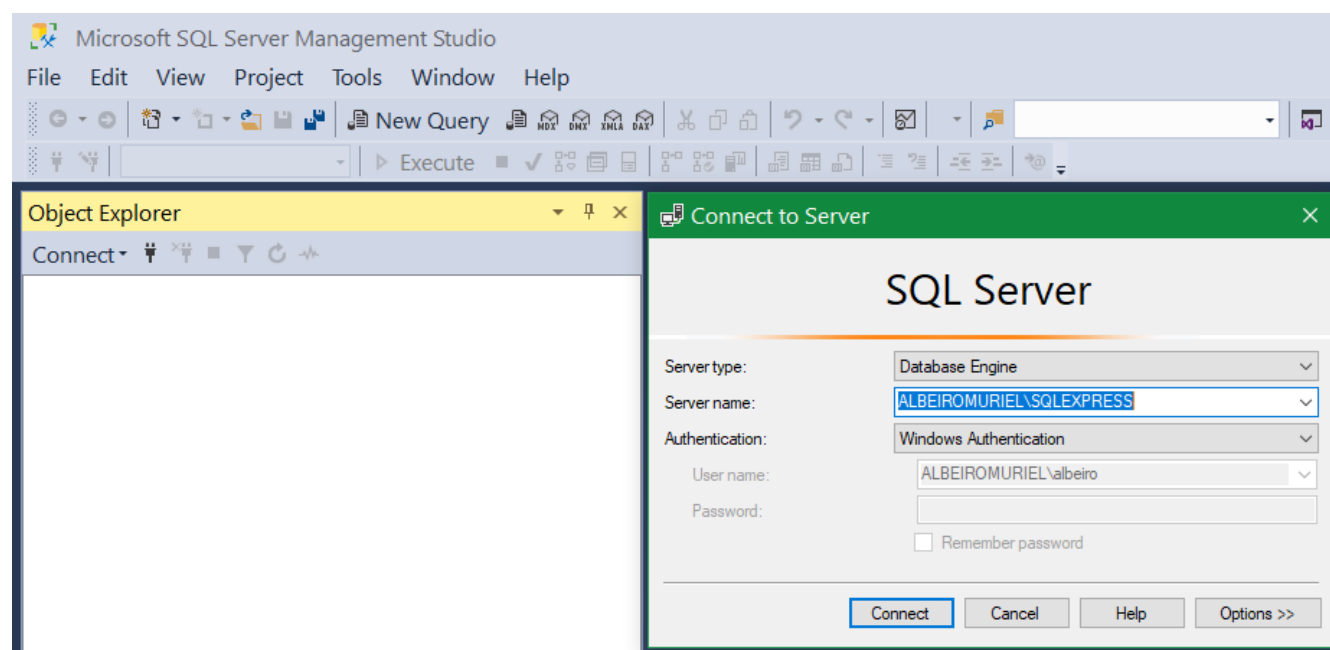
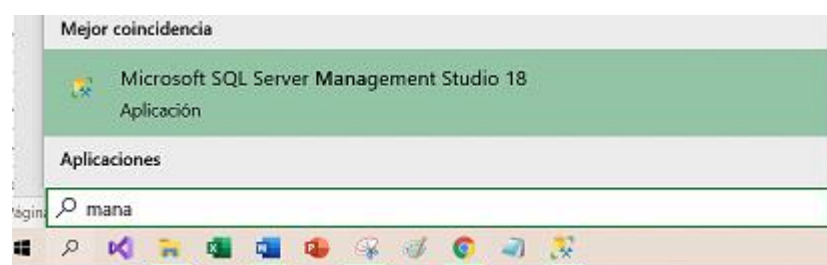


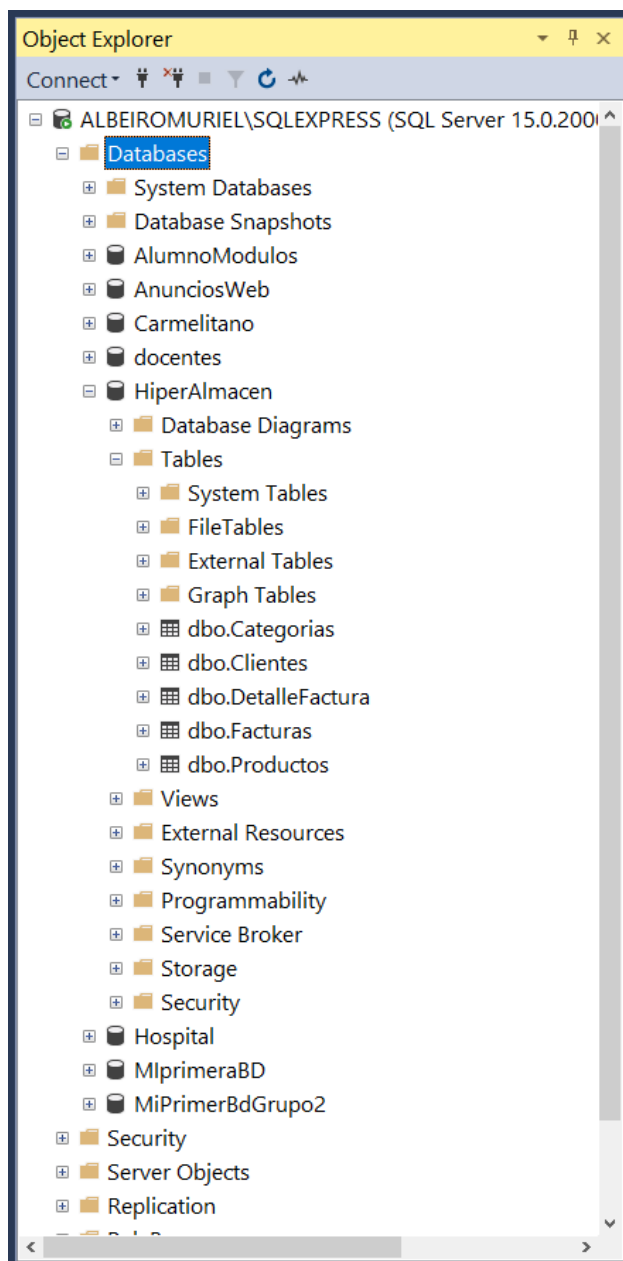
Iniciar SQL Server

Hay varias formas de ingresar a SQL Server, una buscar el icono del Management Studio en el escritorio o en archivos de programas



Otra forma. En el buscador de Windows escribir Management Studio.

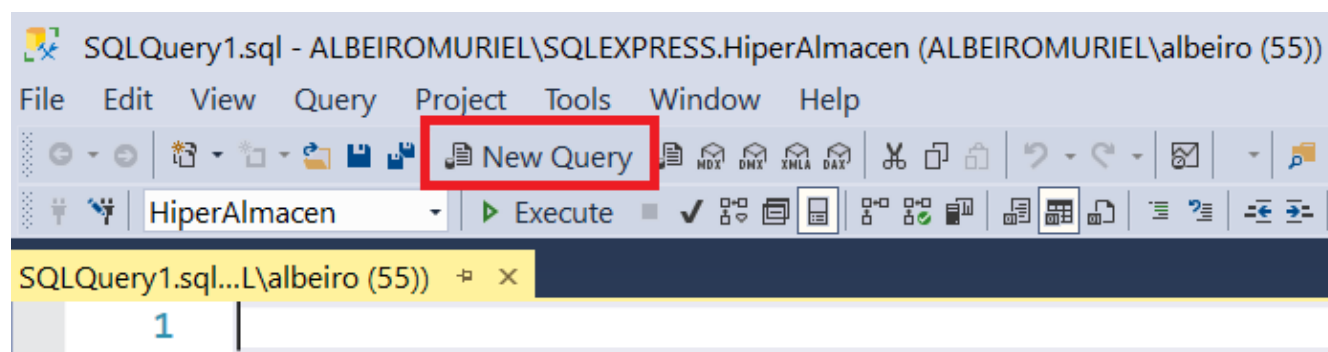




Para crear bases de datos o tablas, la forma más rápida es utilizar clic derecho sobre la opción que desea. Este curso se enfocará en los comandos Transact Sql para realizar todas las acciones de sobre la base de datos.

En la figura se puede observar el entorno de base de datos y tablas en el Explorador de Objetos del SQL Server.

Ventana de consultas



En la ventana de consultas se realizarán todos los comandos para crear, modificar o eliminar base de datos o cualquiera de sus componentes.

Una base de datos está conformada por:

- ✓ Tablas
- ✓ Consultas
- ✓ Procedimientos almacenados
- ✓ Otros elementos

CREAR UNA BASE DATOS

Las bases de datos, tablas, índices, columnas, alias, vistas, rutinas almacenadas, desencadenadores, eventos, variables, particiones, espacios de tabla, puntos de guardado, etiquetas, usuarios, roles, se conocen colectivamente como identificadores y tienen ciertas reglas para nombrar.

Caracteres validados para nombrar una base de datos

Los siguientes caracteres son válidos y permiten que los identificadores no estén entre comillas:

- ✓ ASCII: [0-9, az, AZ \$ _] (números 0-9, letras latinas básicas, tanto mayúsculas como minúsculas, signo de dólar, subrayado)
- ✓ Extendido: U + 0080 .. U + FFFF

Reglas adicionales

Hay una serie de otras reglas para los identificadores:

- ✓ Los identificadores se almacenan como Unicode (UTF-8)
- ✓ Los identificadores pueden o no distinguir entre mayúsculas y minúsculas. Ver Identifier mayúsculas y minúsculas .

- ✓ Los nombres de bases de datos, tablas y columnas no pueden terminar con caracteres de espacio
 - Los nombres de los identificadores pueden comenzar con un número, pero solo no pueden contener números a menos que estén entre comillas.
 - Un identificador que comienza con un número, seguido de una 'e', se puede analizar como un número de punto flotante y debe ser citado.
 - No se permite que los identificadores contengan el carácter NUL ASCII (U + 0000) y los caracteres complementarios (U + 10000 y superior).
 - Los nombres como 5e6, 9e no están prohibidos, pero se recomienda encarecidamente no usarlos, ya que podrían conducir a la ambigüedad en ciertos contextos, siendo tratados como un número o expresión.
 - Las variables de usuario no se pueden usar como parte de un identificador, o como un identificador en una declaración SQL.
- ✓ Longitud máxima
 - Las bases de datos, tablas, columnas, índices, restricciones, rutinas almacenadas, desencadenadores, eventos, vistas, espacios de tablas, servidores y grupos de archivos de registro tienen una longitud máxima de 64 caracteres.
 - Las etiquetas de instrucciones compuestas tienen una longitud máxima de 16 caracteres
 - Los alias tienen una longitud máxima de 256 caracteres, excepto los alias de columna en las sentencias CREATE VIEW , que se adhieren a los mismos y se comparan con la longitud máxima de columna de 64 caracteres (no con la longitud máxima de alias de 256 caracteres).
 - Los usuarios tienen una longitud máxima de 80 caracteres.
 - Los roles tienen una longitud máxima de 128 caracteres.

- o Los caracteres de múltiples bytes no cuentan para el límite de caracteres.

CREATE DATABASE

CREATE DATABASE crea una base de datos con el nombre dado. Para usar esta declaración, necesita el privilegio CREATE para la base de datos. CREATE SCHEMA es sinónimo de CREATE DATABASE.

Ejemplo

CREATE DATABASE bdfactura

Para verificar que la base de datos se crea correctamente, siga estas instrucciones.

- ✓ `exec sp_helpdb` – Muestra las bases de datos del server.

Comando USE base de datos. use factura

USE bdfactura

- ✓ `exec sp_helpdb bdfactura` – Muestra la estructura de base de datos actual.

ELIMINAR BASE DE DATOS DROP

Al eliminar una instantánea de base de datos, ésta se elimina de SQL Server y se eliminan también los archivos dispersos que utiliza. Cuando se quita una instantánea de base de datos, se terminan también todas sus conexiones de usuario.

DROP DATABASE bdfactura

Para quitar una instantánea de base de datos

- ✓ Identifique la instantánea de base de datos que desee quitar. Puede ver las instantáneas de una base de datos en SQL Server Management Studio.
- ✓ Ejecute una instrucción DROP DATABASE especificando el nombre de la instantánea de base de datos que se quitará.

INSTRUCCIÓN CREATE

TABLAS

Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro.

CREATE TABLE

use ejemploClase

```
CREATE TABLE empleados (  
  identificacion real,  
  nombres varchar(60),  
  Salario int,  
  edad tinyint,  
  FechaIngreso date,  
)
```

```
exec sp_columns empleados
```

```
insert into empleados values (1010546879, 'albeiro muriel', 3500000, 50, '12/1/2000')
```

```
select * from empleados
```

Modificar el nombre de una columna

- ✓ alter table empleados alter column identificacion bigint -- modificar el tipo de dato
- ✓ alter table empleados alter column nombres varchar(70) -- modificar el tipo de dato
- ✓ exec sp_rename 'empleados.nombres', 'nombEmp' -- Cambiar el nombre a un columna

Adicionar una columnar

alter table empleados

add idempleado int identity primary key not null -- agregar una clave primaria

alter table empleados drop column edad, salario