



GUSTAVO CORONEL
DESARROLLA SOFTWARE

ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER



Microsoft®
SQL Server®

Eric Gustavo Coronel Castillo
LIMA – PERU
2022



GUSTAVO CORONEL
DESARROLLA SOFTWARE

ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Administración de Base de Datos SQL Server

Derechos Reservados © 2022 GUSTAVO CORONEL

Primera Edición

LIMA - PERÚ



Presentación

Microsoft SQL Server es quizás la base de datos más utilizada, no solo por las instituciones educativas, sino también en las empresas. Entre sus principales características que la hacen la preferida tenemos su facilidad para su instalación, herramienta administrativa visual y muy intuitiva, bases de datos muy fáciles de transportar, integración con todos los productos de Microsoft, como por ejemplo Office, Reporting Service, SSIS y SSAS, pero, sobre todo, está en constante evolución, y también puedes utilizarla para ciencia de datos, para lo cual se integra con R, Python y Java.

Tiene una edición Express que la puedes utilizar de manera gratuita y una edición Developer que puedes utilizarla de manera gratuita para desarrollo.

Durante el desarrollo del curso aprenderás:

1. Diseñar, implementar y administrar la seguridad de servidores y bases de datos SQL Server.
2. Diseñar, implementar y administrar estrategias de backup.
3. Realizar tareas de importación y exportación de datos utilizando diferentes herramientas que te proporciona SQL Server.
4. Gestionar alertas y notificaciones.
5. Automatizar tareas administrativas.
6. Realizar auditorías de bases de datos.

Al estudiar este curso te estás preparando en los temas fundamentales para que puedas desempeñarte como DBA de servidores de base de datos SQL Server.

Eric Gustavo Coronel Castillo
INSTRUCTOR



Índice

CAPÍTULO 1 INTRODUCCIÓN A SQL SERVER	7
QUÉ ES SQL SERVER	7
VENTAJAS	12
SERVICIOS	13
<i>Contexto</i>	<i>13</i>
<i>Tecnologías de Microsoft SQL Server.....</i>	<i>14</i>
<i>SQL Server Database Engine</i>	<i>15</i>
<i>SQL Server Agent</i>	<i>15</i>
<i>SQL Server Analysis Services.....</i>	<i>15</i>
<i>SQL Server Browser</i>	<i>15</i>
<i>SQL Server Integration Services.....</i>	<i>15</i>
<i>SQL Server VSS Witer</i>	<i>16</i>
<i>SQL Server Reporting Services</i>	<i>16</i>
<i>SQL Server Master Data Services.....</i>	<i>16</i>
<i>SQL Server Data Quality Services.....</i>	<i>16</i>
HERRAMIENTAS DE TERCEROS.....	17
<i>Administración de la base de datos de SQL Server.....</i>	<i>17</i>
<i>Monitoreo de rendimiento de SQL Server.....</i>	<i>17</i>
<i>Información de SQL server</i>	<i>17</i>
CAPÍTULO 2 PREPARANDO EL ENTORNO DE TRABAJO	18
EDICIONES DE SQL SERVER.....	18
REQUISITOS GENERALES	20
HERRAMIENTAS.....	21
<i>Herramientas básicas</i>	<i>21</i>
<i>Base de datos de ejemplo</i>	<i>22</i>
<i>SQL Server Management Studio.....</i>	<i>23</i>
<i>Administrador de configuración de SQL Server.....</i>	<i>24</i>
CONECTARSE CON SSMS.....	25
AUTORIZACIÓN DE CONEXIONES ADICIONALES	29
<i>Crear un inicio de sesión con autenticación de Windows.....</i>	<i>29</i>
<i>Crear un inicio de sesión con autenticación SQL Server.....</i>	<i>32</i>
CONEXIÓN REMOTA	35
<i>Contexto</i>	<i>35</i>
<i>Habilitar protocolos</i>	<i>35</i>
<i>Configurar un puerto fijo.....</i>	<i>37</i>
<i>Abrir puertos del firewall</i>	<i>38</i>
<i>Conexión desde otro equipo.....</i>	<i>41</i>
CAPÍTULO 3 TRABAJANDO CON BASES DE DATOS	44
ARQUITECTURA DE BASES DE DATOS SQL SERVER.....	44



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

<i>Estructura Lógica</i>	44
<i>Estructura Física</i>	45
CREACIÓN DE UNA BASE DE DATOS.....	47
<i>Contexto</i>	47
<i>Concepto de base de datos</i>	47
<i>Crear base de datos usando T-SQL</i>	48
<i>Creación una base de datos usando SSMS</i>	50
GESTIÓN DE ARCHIVOS.....	52
<i>Grupos de archivos</i>	52
<i>Grupo de archivos predeterminado</i>	52
<i>Ejemplo de archivos y grupos de archivos</i>	53
PRACTICA	55
<i>Ejercicio 1</i>	55
<i>Ejercicio 2</i>	56
CAPÍTULO 4 GESTIÓN DE SEGURIDAD	57
CONTEXTO	57
<i>Securables</i>	57
<i>Principals</i>	57
<i>Permissions</i>	57
MODELO DE SEGURIDAD EN SQL SERVER	58
JERARQUIA DE SEGURIDAD	59
TIPOS DE AUTENTICACIÓN.....	60
<i>Autenticación Windows</i>	60
<i>Autenticación SQL Server</i>	60
<i>Inicios de sesión preestablecidos</i>	60
CONFIGURAR AUTENTICACIÓN DEL SERVIDOR.....	61
<i>Windows Authentication mode</i>	61
<i>SQL Server and Windows Authentication mode</i>	61
INICIOS DE SESIÓN	62
<i>Crear Inicios de Sesión</i>	62
<i>Utilizando la Instrucción CREATE LOGIN</i>	63
ROLES DEL SERVIDOR.....	64
ROLES DE BASE DE DATOS.....	67
<i>Roles Fijos de Base de Datos</i>	67
<i>Gestionando Roles de Base de Datos</i>	68
ROLES DE APLICACIÓN	70
USUARIOS DE BASE DE DATOS	71
<i>Contexto</i>	71
<i>Creación de un Usuario desde SSMS</i>	72
<i>Utilizando la Instrucción CREATE USER</i>	73
PERMISOS	74
<i>Tipos de Permisos</i>	74
<i>Asignación de Permisos o Privilegios</i>	76



CAPÍTULO 5 IMPORTACIÓN Y EXPORTACIÓN DE DATOS	77
INTRODUCCIÓN	77
<i>Contexto</i>	<i>77</i>
<i>In-Process vs. Out-of-Process Operation.....</i>	<i>79</i>
<i>Archivos de formato.....</i>	<i>80</i>
<i>Consideraciones de rendimiento</i>	<i>81</i>
USO DE SSIS PARA IMPORTAR Y EXPORTAR DATOS.....	82
<i>Contexto</i>	<i>82</i>
<i>Exportar datos</i>	<i>83</i>
<i>Importar datos.....</i>	<i>84</i>
<i>Ejercicios propuestos.....</i>	<i>85</i>
UTILITARIO BCP PARA IMPORTAR Y EXPORTAR DATOS	86
<i>Utilitario BCP</i>	<i>86</i>
<i>Copiar filas de una tabla en un archivo de datos utilizando conexión de confianza.....</i>	<i>87</i>
<i>Copiar filas de una tabla en un archivo de datos utilizando autenticación de modo mixta.....</i>	<i>88</i>
<i>Copiar datos de un archivo en una tabla</i>	<i>89</i>
<i>Copiar una columna específica en un archivo de datos.....</i>	<i>91</i>
<i>Copiar una fila específica en un archivo de datos</i>	<i>92</i>
<i>Copiar datos de una consulta en un archivo de datos.....</i>	<i>93</i>
<i>Crear un archivo de formato no XML.....</i>	<i>95</i>
<i>Crear un archivo de formato XML</i>	<i>96</i>
<i>Usar un archivo de formato para importar de forma masiva con bcp.....</i>	<i>97</i>
<i>Exportar a Formato CSV.....</i>	<i>99</i>
IMPORTACIÓN UTILIZANDO BULK INSERT	100
<i>Contexto</i>	<i>100</i>
<i>Importar datos desde un archivo de datos utilizando un archivo de formato.....</i>	<i>100</i>
<i>Importar datos desde un archivo de texto</i>	<i>102</i>
<i>Importar datos desde un archivo de datos en su formato nativo</i>	<i>103</i>
IMPORTACIÓN UTILIZANDO OPENROWSET	104
<i>Contexto</i>	<i>104</i>
<i>Importar datos desde un archivo de datos utilizando un archivo de formato.....</i>	<i>104</i>
CONSULTAS DISTRIBUIDAS	105
<i>Configuración</i>	<i>105</i>
<i>Ejemplos</i>	<i>105</i>
CAPÍTULO 6 CURSOS VIRTUALES	106
CUPONES.....	106
FUNDAMENTOS DE PROGRAMACIÓN CON JAVA.....	106
JAVA ORIENTADO A OBJETOS.....	107
PROGRAMACIÓN CON JAVA JDBC	108
PROGRAMACIÓN CON ORACLE PL/SQL	109



Capítulo 1 INTRODUCCIÓN A SQL SERVER

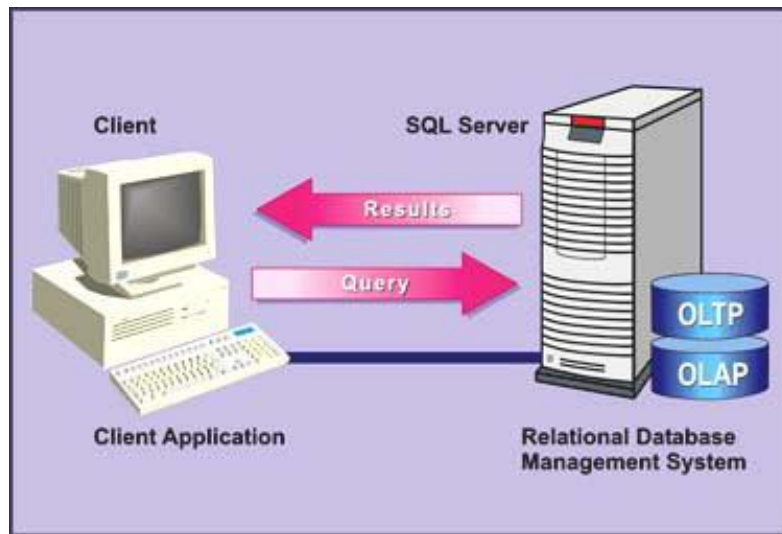
QUÉ ES SQL SERVER



Las bases de datos son exactamente lo que dice su nombre, un almacén de información que se administra mediante un motor o gestor de base de datos y un lenguaje de base de datos. SQL Server es el gestor de base de datos de Microsoft, que se utiliza para almacenar información trabajando en conjunto con sitios web y otros tipos de aplicaciones. Es un gestor de bases de datos relacionales; cuenta con una interfaz que conecta la base de datos con las aplicaciones en las cuales se genera y se utiliza dicha información.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER



SQL son las siglas de **Structured Query Language** (lenguaje de consulta estructurado) y es el lenguaje que utilizan los gestores de bases de datos relacionales y el que deben utilizar las aplicaciones para que puedan interactuar con las bases de datos.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsain
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

Tabla *cliente*

<i>número-cuenta</i>	<i>saldo</i>
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

<i>id-cliente</i>	<i>número-cuenta</i>
19.283.746	C-101
19.283.746	C-201
01.928.374	C-215
67.789.901	C-102
18.273.609	C-305
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

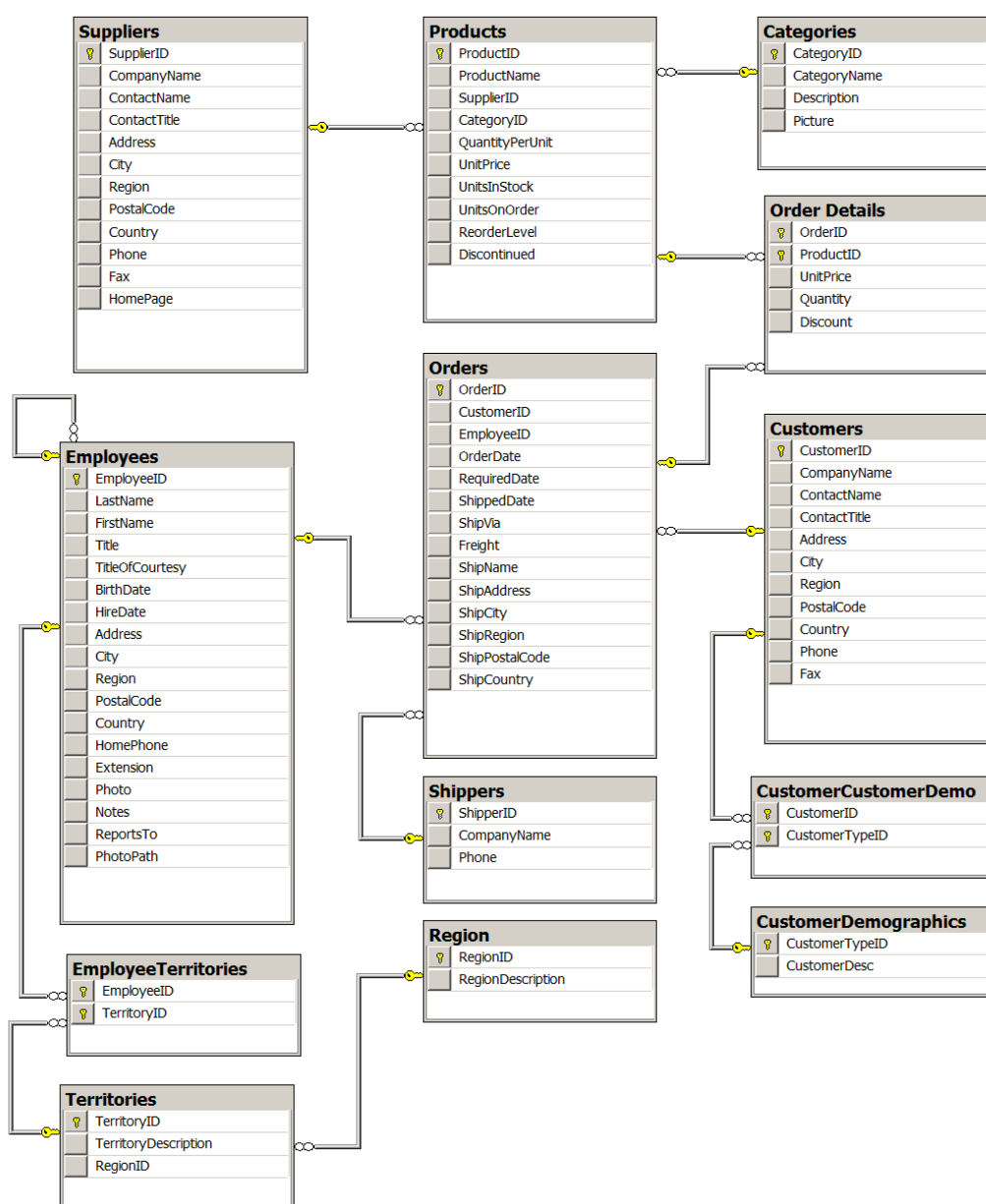
Tabla *cuenta*

Tabla de relación *cliente-cuenta*

El componente principal de una base de datos relacional es la tabla, la cual está formada por filas y columnas, y existe relación entre las tablas de una base de datos, lo que permite tener información consistente, los datos de una tabla tienen reglas a las que se les llama restricciones, como por ejemplo clave primaria, clave foránea, clave única, entre otras.



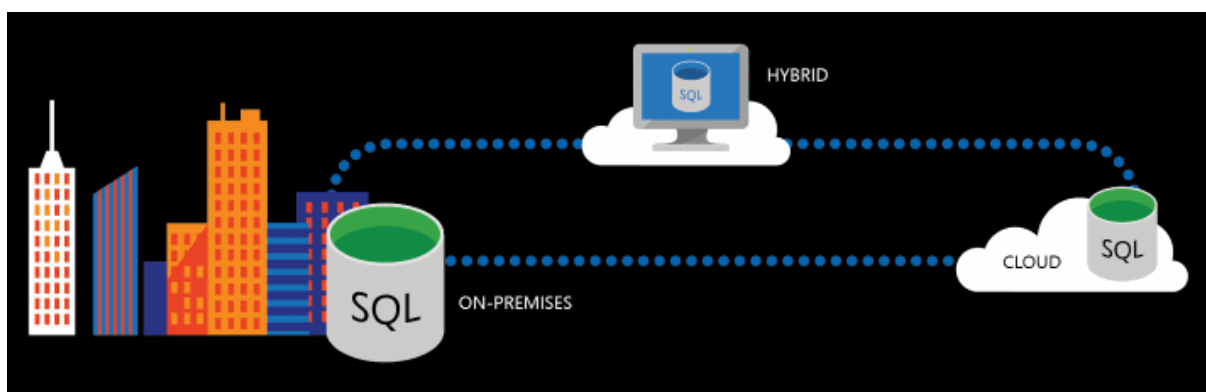
ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER



Los conjuntos de tablas de una base de datos se pueden graficar en un diagrama de base de datos, tal como se ilustra en la imagen.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER



Microsoft está en constante optimización de los servicios de SQL Server, pero también, añadiendo nuevos servicios para responder a los diferentes tipos de necesidades de las empresas y los negocios.

Entre sus ventajas que ofrece SQL Server encontramos las siguientes:

Líder del sector

Cree aplicaciones de misión crítica para procesamiento de transacciones en línea (OLTP) con escalabilidad, rendimiento y alta disponibilidad.

Seguridad avanzada

Proteja los datos en reposo y en movimiento. SQL Server es la base de datos menos vulnerable según la base de datos de vulnerabilidades del NIST.

BI móvil integral

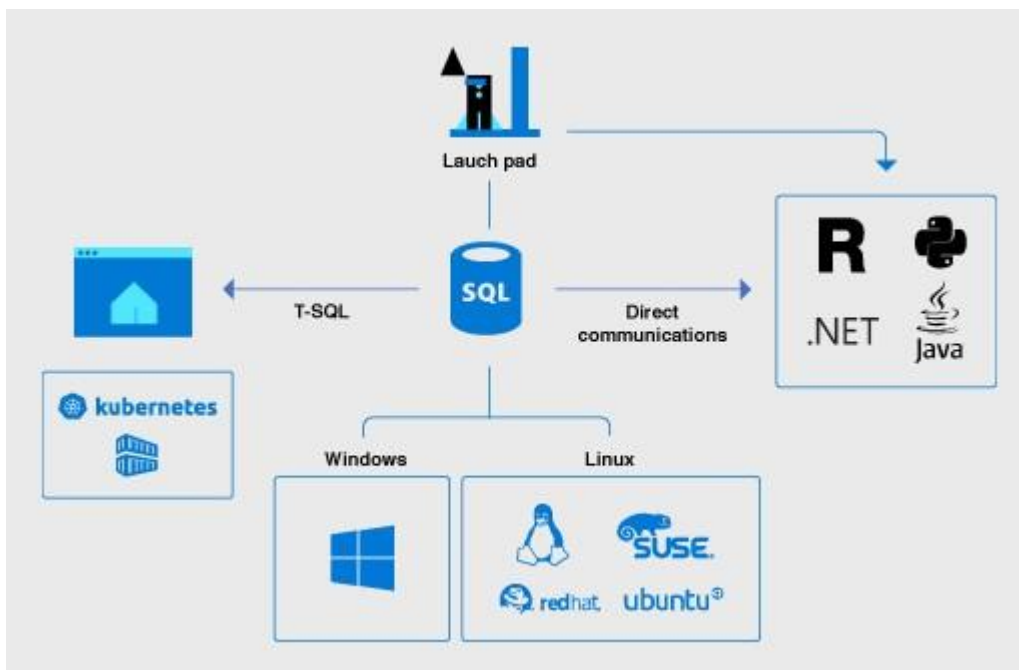
Transforme datos en conocimiento práctico. Proporcione información en cualquier dispositivo, con o sin conexión, a una quinta parte del coste de otras soluciones de autoservicio.

Diseñado para la nube híbrida

Obtenga una plataforma y herramientas coherentes para mover más fácilmente las cargas de trabajo entre su centro de datos, la nube privada o Microsoft Azure.



VENTAJAS



A continuación, se describen las ventajas y desventajas:

1. Soporte de transacciones.
2. Permite escalabilidad, estabilidad y seguridad.
3. Soporta procedimientos almacenados.
4. Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
5. Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
6. Permite administrar información de otros servidores de la red desde un solo punto.
7. Es un producto muy maduro con mucha base implantada y muy buena comunidad.



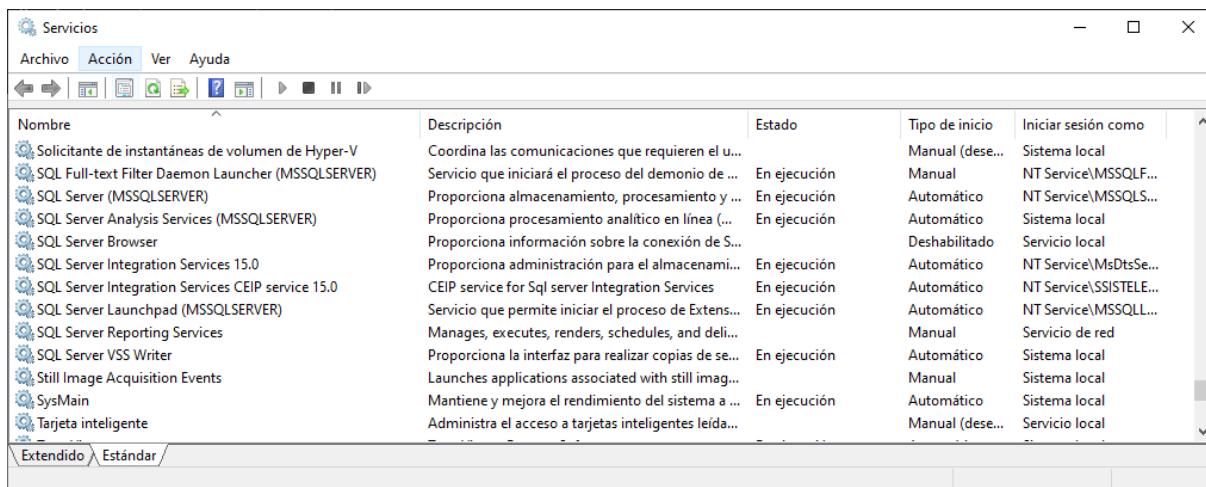
SERVICIOS

Contexto



Tener un repositorio de datos no es suficiente, hoy en día los motores de base de datos están evolucionando y adaptando a las nuevas necesidades que el mercado exige, y SQL Server se puede considerar como una plataforma de muchos servicios centrados en los datos.

Tecnologías de Microsoft SQL Server



Nombre	Descripción	Estado	Tipo de inicio	Iniciar sesión como
Solicitante de instantáneas de volumen de Hyper-V	Coordina las comunicaciones que requieren el u...		Manual (dese...	Sistema local
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Servicio que iniciará el proceso del demonio de ...	En ejecución	Manual	NT Service\MSSQLF...
SQL Server (MSSQLSERVER)	Proporciona almacenamiento, procesamiento y ...	En ejecución	Automático	NT Service\MSSQLS...
SQL Server Analysis Services (MSSQLSERVER)	Proporciona procesamiento analítico en línea (...)	En ejecución	Automático	Sistema local
SQL Server Browser	Proporciona información sobre la conexión de S...		Deshabilitado	Servicio local
SQL Server Integration Services 15.0	Proporciona administración para el almacenami...	En ejecución	Automático	NT Service\MsDtsSe...
SQL Server Integration Services CEIP service 15.0	CEIP service for Sql server Integration Services	En ejecución	Automático	NT Service\SSISTELE...
SQL Server Launchpad (MSSQLSERVER)	Servicio que permite iniciar el proceso de Extens...	En ejecución	Automático	NT Service\MSSQLL...
SQL Server Reporting Services	Manages, executes, renders, schedules, and deli...		Manual	Servicio de red
SQL Server VSS Writer	Proporciona la interfaz para realizar copias de se...	En ejecución	Automático	Sistema local
Still Image Acquisition Events	Launches applications associated with still imag...		Manual	Sistema local
SysMain	Mantiene y mejora el rendimiento del sistema a ...	En ejecución	Automático	Sistema local
Tarjeta inteligente	Administra el acceso a tarjetas inteligentes leída...		Manual (dese...	Servicio local

Microsoft SQL Server es un conjunto completo de tecnologías y herramientas de administración y análisis de datos para el entorno empresarial.

Cuando se instala SQL Server se habilitan una serie de servicios o programas que se ejecutan en segundo plano y el administrador de base de datos debe ser capaz de identificarlos para gestionarlos adecuadamente.

- SQL Server Database Engine
- SQL Server Agent
- SQL Server Analysis Services
- SQL Server Browser
- SQL Server Integration Services
- SQL Server VSS Writer
- SQL Server Reporting Services
- Master Data Services (Servicios de Datos Maestros)
- Data Quality Services (Servicios de Calidad de Datos)
- Replication (Replicación ó Duplicación)

SQL Server Database Engine

SQL Server Database Engine (Motor de Base de Datos) es el servicio básico del producto y se encarga de las operaciones para el almacenamiento, procesamiento y manejo de la seguridad de los datos. Controla el acceso de las aplicaciones a la base de datos y ejecuta las transacciones.

El motor de base de datos le permite crear bases de datos relacionales para procesamiento de transacciones en línea (OLTP) o procesamiento analítico de datos en línea (OLAP), incluyendo la creación de los objetos de la base de datos: tablas, índices, vistas, procedimientos almacenados, y usuarios.

SQL Server Agent

Este servicio se encarga de controlar las tareas programadas y dispara las replicas que se encuentren habilitadas.

SQL Server Analysis Services

SQL Server Analysis Services (SSAS) te permite construir y desplegar bases de datos para análisis como soporte a las soluciones de inteligencia de negocios.

SSAS te permite diseñar, construir y administrar bases de datos multidimensionales con data consolidada de las bases de datos transaccionales u operacionales y de otros orígenes de datos, utilizando la técnica de datos agregados.

SQL Server Browser

Servicio que facilita la conexión con los clientes, por ejemplo, permitiendo la conexión sin especificar el puerto.

SQL Server Integration Services

SQL Server Integration Services (SSIS) es una plataforma que brinda la posibilidad de crear soluciones de integración de datos a nivel empresarial. Permite migrar datos desde diversos orígenes, transformarlos para estandarizarlos, y almacenarlos en un repositorio común. Una de sus aplicaciones más comunes es la actualización de datawarehouses.

SSIS se basa en la construcción de paquetes que contienen las tareas a ejecutar. Los paquetes permiten programar tareas casi sin escribir líneas de código.

SQL Server VSS Witer

Servicio que permite tener acceso a los archivos de datos, incluso cuando esté corriendo la instancia.

SQL Server Reporting Services

SQL Server Reporting Services (SSRS) es una plataforma que a través de un conjunto de herramientas y servicios permite acceder a una variedad de orígenes de datos para generar y publicar reportes en diferentes formatos.

Incluye APIs que permiten a los desarrolladores crear reportes a la medida de las necesidades de los usuarios.

SQL Server Master Data Services

SQL Server Master Data Services (MDS) es la solución SQL Server para la administración de la data maestra. Mediante Master Data Management (MDM) puedes definir listas de datos no transaccionales para mantenimiento de la data maestra.

Master Data Services incluye jerarquías, seguridad granular, transacciones, control de versiones de datos y reglas de negocios.

SQL Server Data Quality Services

SQL Server Data Quality Services (SSDQS) es una solución basada en el conocimiento de la calidad de datos. Te permite la construcción de una base de conocimiento de calidad de los datos (Data Quality Knowledge Base o DQKB) y utilizarla para realizar varias tareas de calidad de datos críticas como corrección, enriquecimiento, normalización y consolidación.



HERRAMIENTAS DE TERCEROS

Administración de la base de datos de SQL Server

Mejorar considerablemente la productividad de los DBA, asegurarse un rendimiento máximo de la base de datos y mitigar los riesgos de los cambios. Las herramientas de administración de base de datos **Toad for SQL Server** y **ApexSQL** proporcionan capacidades líderes de la industria y flujos de trabajo intuitivos para administrar fácilmente más bases de datos.

Monitoreo de rendimiento de SQL Server

Optimice fácilmente e implemente ajustes con **Spotlight on SQL Server Enterprise**. Garantice el máximo rendimiento gracias al monitoreo, el diagnóstico y la optimización inigualables. Obtenga acceso a los detalles que necesita para un óptimo ajuste de SQL Server. Obtenga información general intuitiva sobre el estado y el rendimiento, las acciones y alertas automatizadas, y el soporte para los dispositivos móviles.

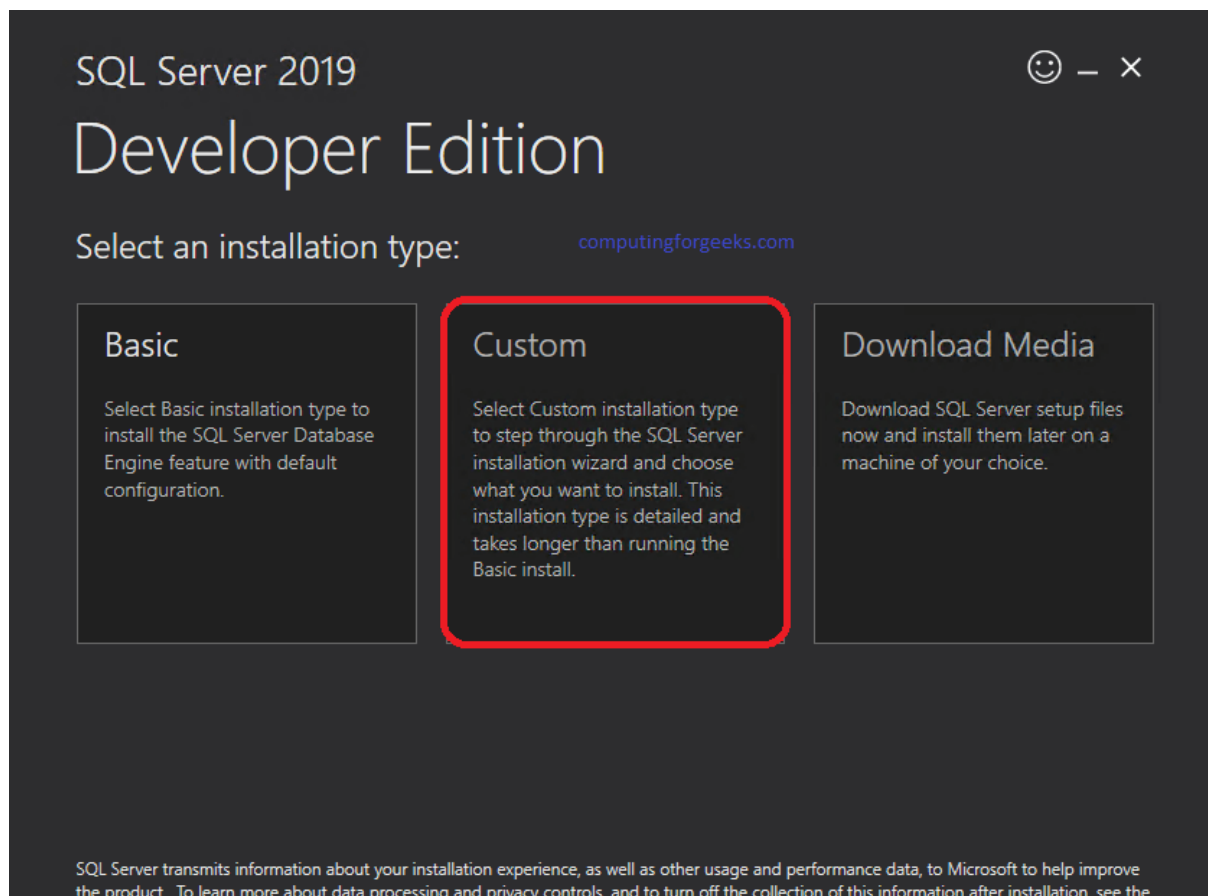
Información de SQL server

Utilice **Foglight** para monitorear fácilmente entornos de bases de datos complejos y diversos. Obtenga información valiosa de todas las áreas de los recursos de la base de datos, análisis de la carga de trabajo y control de cambios. Reciba un panorama completo de sus entornos multiplataforma, incluidos código abierto y NoSQL, para que pueda colaborar con todos los equipos a fin de encontrar y solucionar los problemas de rendimiento, antes de que pongan en riesgo su base de datos.



Capítulo 2 PREPARANDO EL ENTORNO DE TRABAJO

EDICIONES DE SQL SERVER



Estas son las ediciones disponibles de SQL Server:

- Express** : Edición gratuita. Incluye funcionalidad del core-RDBMS. Limitado a 10G de tamaño de disco. Ideal para desarrollo y pruebas.
- Edición Estándar** : Incluye funcionalidad central y capacidades de Business Intelligence.
- Edición Enterprise** : Edición SQL Server con todas las funciones. Incluye capacidades avanzadas de seguridad y almacenamiento de datos.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Edición Developer : Incluye todas las características de la Edición Enterprise y sin limitaciones, y se puede descargar y utilizar de forma gratuita solo con fines de desarrollo.

Después de descargar/adquirir SQL Server, la instalación se ejecuta con **SQLSetup.exe**, que está disponible como una GUI o un programa de línea de comandos.

La instalación a través de cualquiera de estos requerirá que especifique una clave de producto y ejecute alguna configuración inicial que incluya funciones habilitadas, servicios separados y la configuración de los parámetros iniciales para cada uno de ellos. Los servicios y características adicionales se pueden habilitar en cualquier momento ejecutando el programa SQLSetup.exe en la línea de comandos o en la versión de la GUI.

El lenguaje de desarrollo de SQL Server es **Transact-SQL** (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).



REQUISITOS GENERALES

Para que tengas un funcionamiento aceptable del SQL Server se realizan las siguientes recomendaciones:

Característica	Mínimo	Recomendable
Procesador	Core i3, AMD Athlon	Intel Core i5, AMD Ryzen o superiores
Memoria RAM	4 GB	16 GB
Espacio en disco duro	512 MB	2 GB
Sistema operativo	Para escritorio Windows 10 Pro	

Opciones de instalación recomendadas:

Opción	Valor
Tipo de instancia	Instancia por defecto
Modo de autenticación	Mixta
Analysis Services	Multidimensional

Para más información, consultar el siguiente enlace:

<https://docs.microsoft.com/es-es/sql/database-engine/install-windows/install-sql-server>

HERRAMIENTAS

Motor de base de datos de SQL Server incluye varias herramientas. En este documento se describen las primeras herramientas que necesitará para empezar a trabajar con este motor de base de datos.

Se puede obtener acceso a todas las herramientas desde el menú Inicio. No siempre se instalan las mismas herramientas, como SQL Server Management Studio, de forma predeterminada. Debe seleccionarlas como parte de los componentes cliente durante la instalación.

Herramientas básicas

- **SQL Server Management Studio**

Es la herramienta principal para administrar el motor de base de datos y escribir código Transact-SQL. Se encuentra en el shell de Visual Studio. No se incluye en SQL Server Express, pero está disponible como descarga independiente en el Centro de descarga de Microsoft.

- **SQL Server Configuration Manager**

Se instala con SQL Server y las herramientas cliente. Permite habilitar protocolos de servidor, configurar opciones de protocolo, como los puertos TCP, configurar servicios de servidor para que se inicien automáticamente y configurar equipos cliente para que se conecten de la forma preferida. Esta herramienta configura los elementos de conectividad más avanzados, pero no habilita las características.



Base de datos de ejemplo

Las bases de datos de ejemplo no están incluidas en el proceso de instalación de SQL Server.

La mayoría de los ejemplos descritos en los Libros en pantalla usan la base de datos ejemplo **AdventureWorks**, si vas a usar esta base de datos debes descargarla e instalarla para que puedas tenerla en tu servidor.

A continuación, tienes el enlace desde donde puedes descargar los scripts de las bases de datos ejemplo:

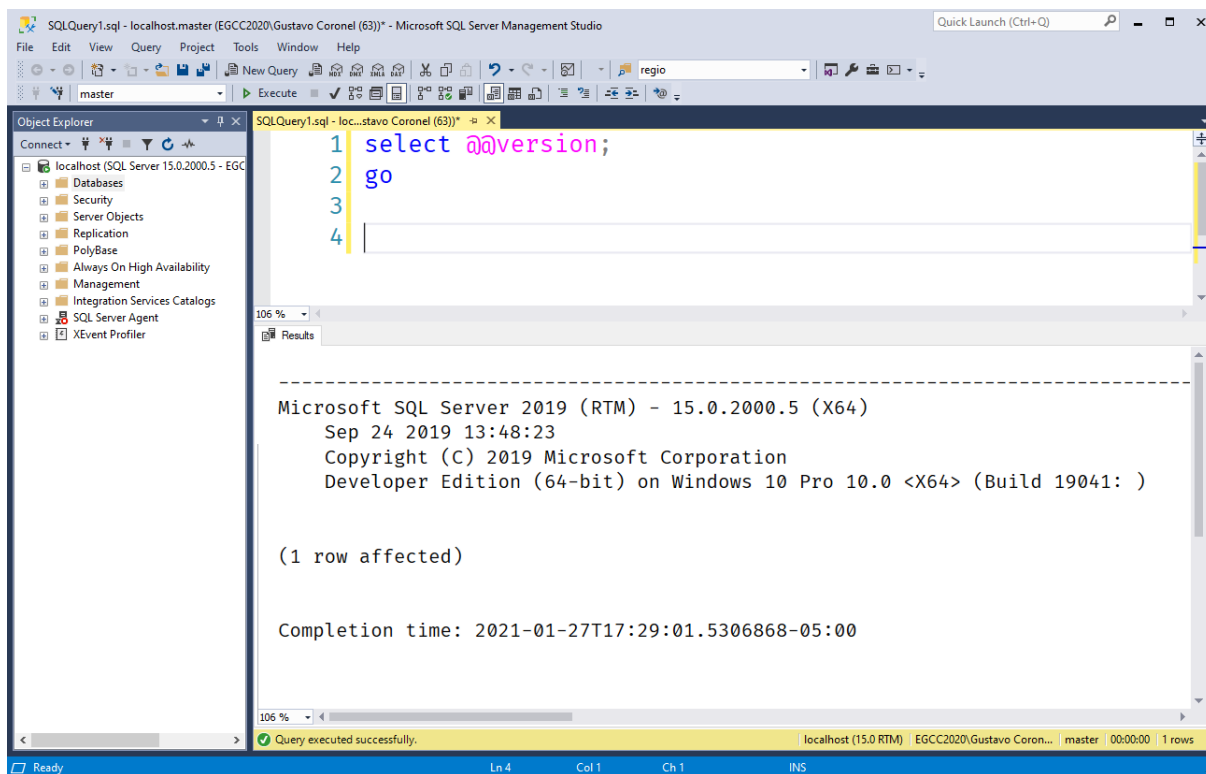
<https://github.com/microsoft/sql-server-samples/tree/master/samples/databases>

En el siguiente repositorio también vas a encontrar scripts para crear bases de datos ejemplo:

<https://github.com/gcoronelc/databases>

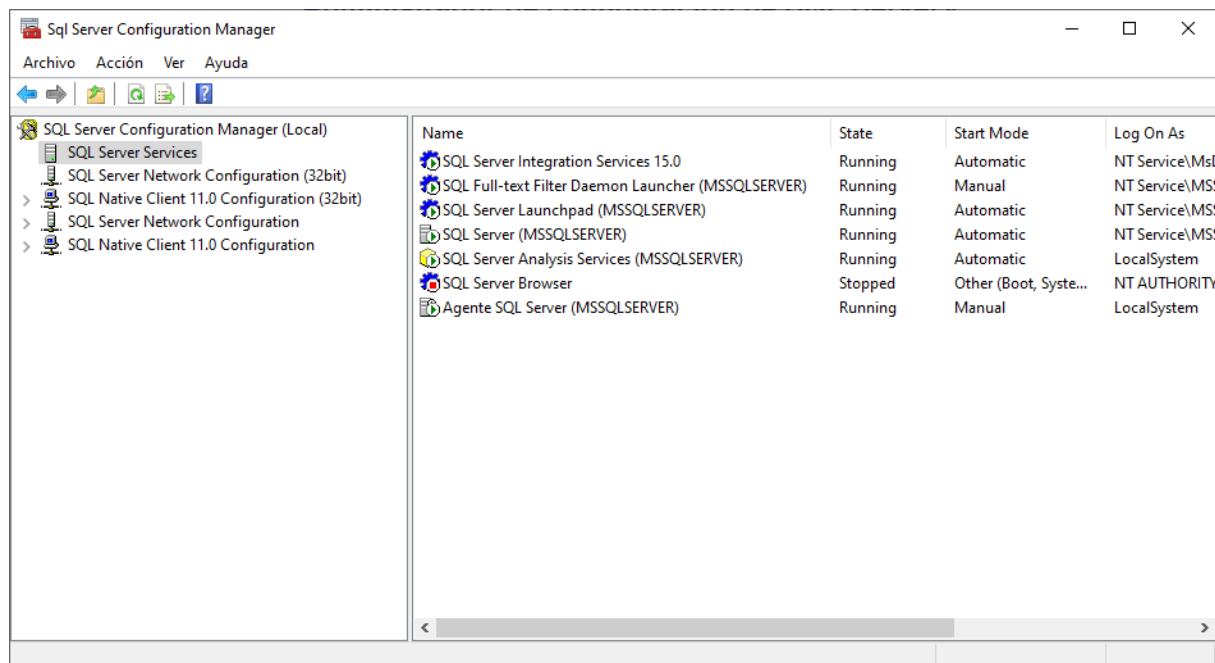
SQL Server Management Studio

En el menú **Inicio**, debes buscar y ejecutar **SQL Server Management Studio**.



Administrador de configuración de SQL Server

En el menú **Inicio**, debes buscar y ejecutar **Administrador de configuración de SQL Server**.



Esta herramienta te permite:

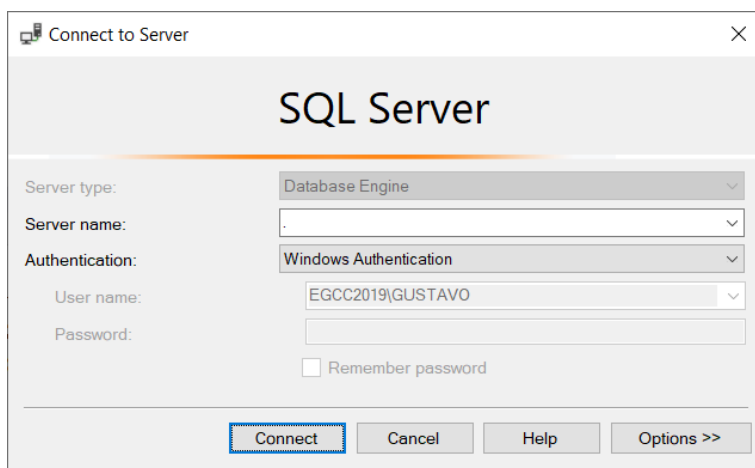
- Gestionar los servicios del servidor.
- Gestionar los protocolos de acceso al servidor.

CONECTARSE CON SSMS

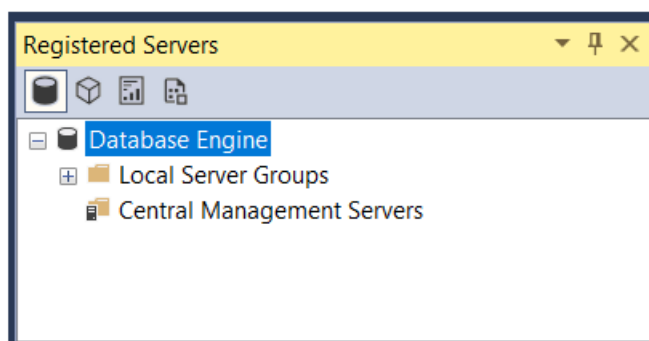
Resulta sencillo conectarse al Motor de base de datos desde herramientas que se ejecutan en el mismo equipo si conoce el nombre de la instancia y si se conecta como miembro del grupo Administradores del equipo. Los procedimientos siguientes deben realizarse en el mismo equipo en el que se ha instalado SQL Server.

Para determinar el nombre de la instancia de motor de base de datos:

1. Inicia una sesión en **Windows** como miembro del grupo **Administradores** y ejecuta **SQL Server Management Studio**.
2. En el cuadro de diálogo **Conectar con el servidor**, hacer clic en **Cancelar**.



3. Si no aparece la vista **Servidores registrados**, en el menú **Ver**, debes hacer click en **Servidores registrados**.

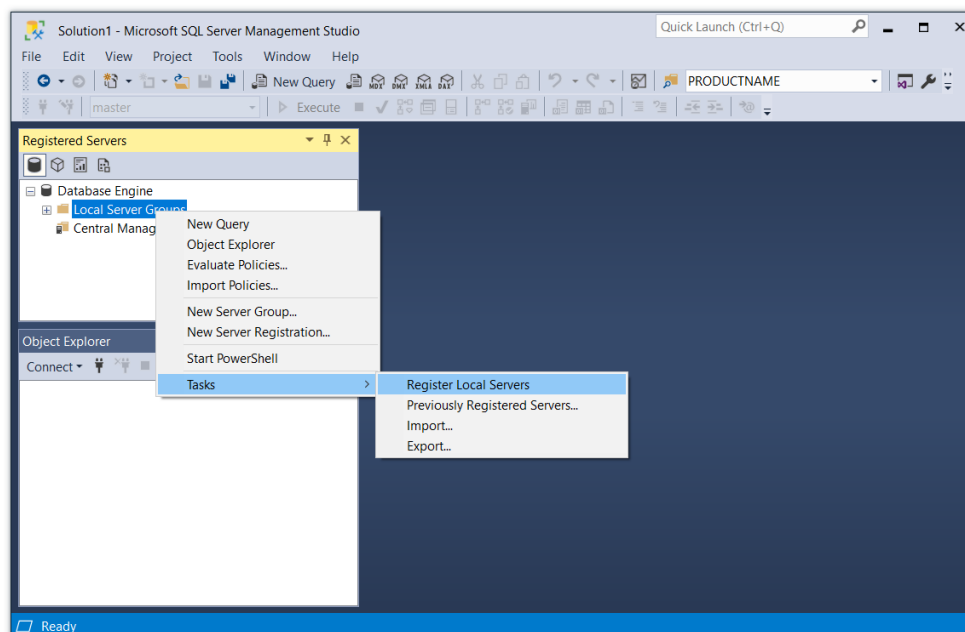


4. Con la opción **Motor de base de datos** seleccionado en la barra de herramientas de la vista **Servidores registrados**, expandir **Motor de base de datos**, hacer clic con el botón secundario en **Grupos de servidores locales**,

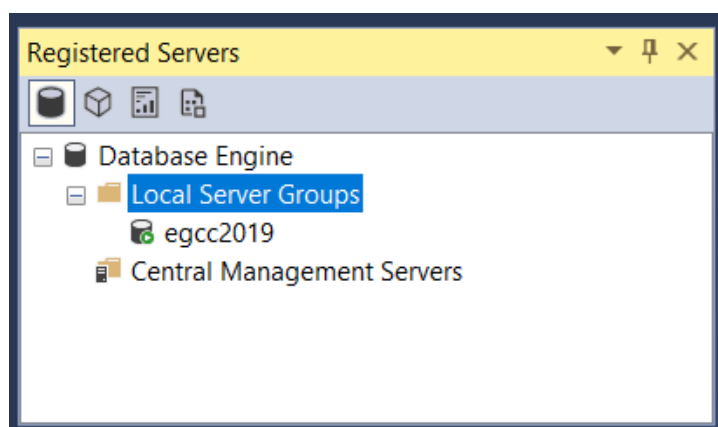


ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

seleccione **Tareas** y, a continuación, hacer clic en **Registrar servidores locales**.



5. Se muestran todas las instancias de Motor de base de datos instaladas en el equipo. La instancia predeterminada no tiene nombre y aparece como el nombre del equipo. Una instancia con nombre aparece como el nombre del equipo seguido de una barra inversa (\) y del nombre de la instancia. En SQL Server Express, la instancia se denomina **<nombre_equipo>\sqlexpress** a no ser que se haya cambiado el nombre durante la instalación.

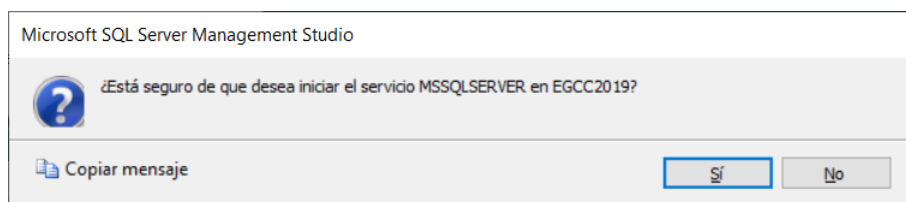
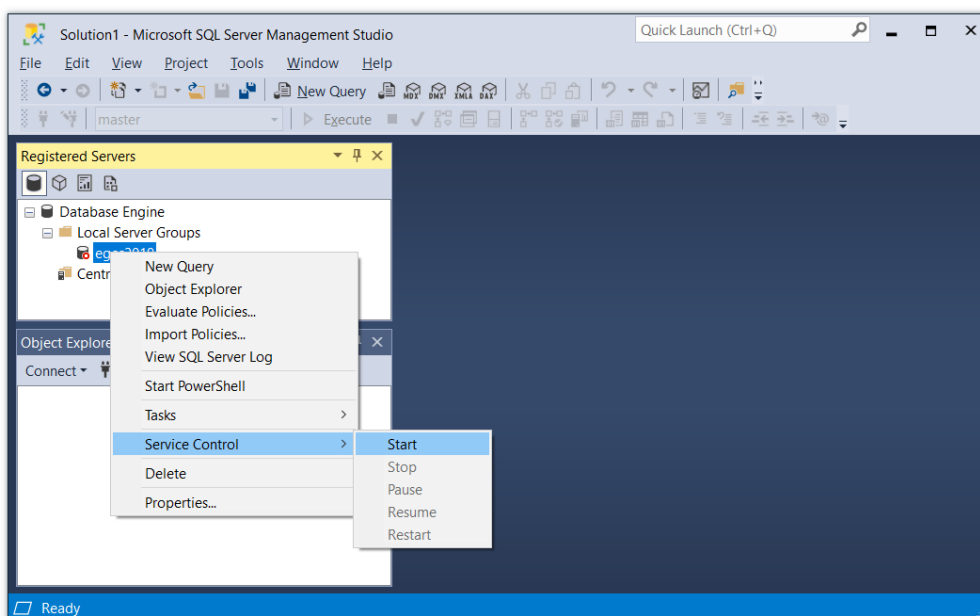




ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Para comprobar que el motor de base de datos está en ejecución:

6. En **Servidores registrados**, si el nombre de la instancia de SQL Server tiene un punto verde con una flecha blanca junto al nombre, el motor de base de datos está en ejecución y no es necesario realizar ninguna otra acción.
7. Si el nombre de la instancia de SQL Server tiene un punto rojo con un cuadrado blanco junto al nombre, el motor de base de datos se encuentra detenido. Haga clic con el botón secundario en el nombre de motor de base de datos, haga clic en **Control de servicios** y, a continuación, haga clic en **Iniciar**. Después de un cuadro de diálogo de confirmación, el motor de base de datos debería iniciarse y el color del punto debería cambiar a verde con una flecha blanca.

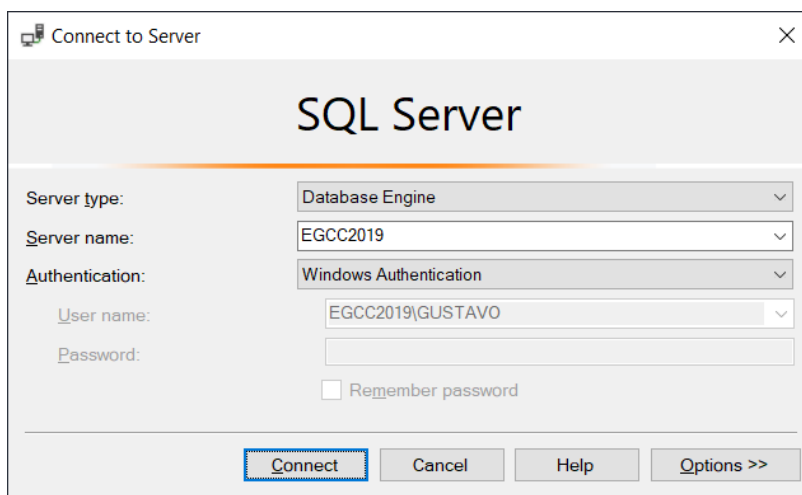




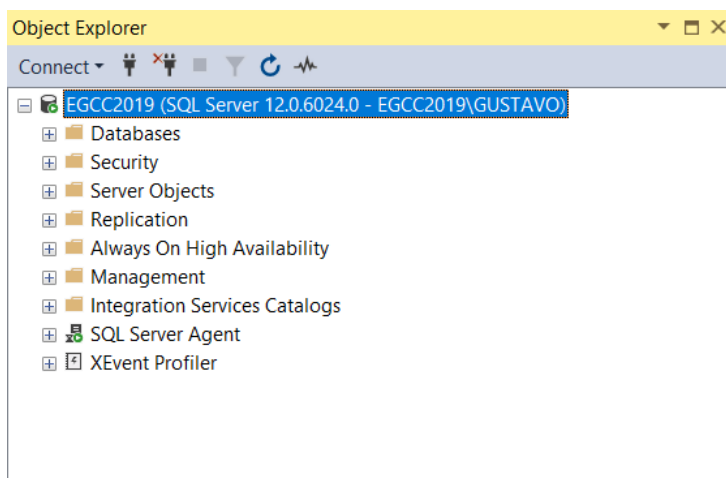
ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Para conectarse al motor de base de datos:

1. En **Management Studio**, en el menú **Archivo**, haga clic en **Conectar Explorador de objetos**. Se abre el cuadro de diálogo **Conectar con el servidor**. En el control **Tipo de servidor** se muestra el último tipo de componente utilizado.



2. Seleccione **Motor de base de datos**.
3. En el control **Nombre del servidor**, escriba el nombre de la instancia del motor de base de datos. Para la instancia predeterminada de SQL Server, el nombre de servidor es el nombre del equipo. Para una instancia con nombre de SQL Server, el nombre del servidor tiene el siguiente formato: **<nombre_equipo>\<nombre_instancia>**, a continuación, tiene un ejemplo: **EGCC2019\SQLEXPRESS**.
4. Hacer click en **Conectar**.



AUTORIZACIÓN DE CONEXIONES ADICIONALES

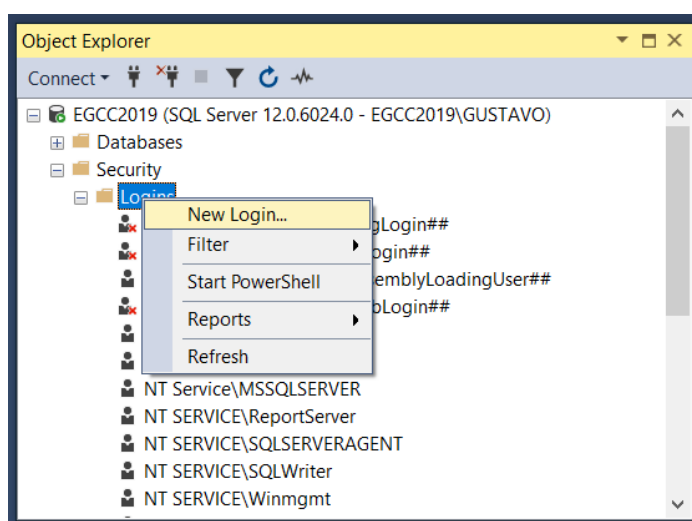
Ahora que te has conectado a SQL Server como administrador, una de las primeras tareas que debes realizar es autorizar a otros usuarios a conectarse. Para ello debes crear un inicio de sesión y concédale autorización para obtener acceso a una base de datos como usuario.

Los inicios de sesión pueden ser inicios de sesión con autenticación de Windows, que utilizan las credenciales de Windows, o bien inicios de sesión con autenticación de SQL Server, que almacenan la información de autenticación en SQL Server y son independientes de las credenciales de Windows.

Se recomienda utilizar la autenticación de Windows siempre que sea posible.

Crear un inicio de sesión con autenticación de Windows

1. En la tarea anterior, se conectó al motor de base de datos utilizando Management Studio. En el **Explorador de objetos**, expanda la instancia del servidor, expanda **Seguridad**, haga clic con el botón secundario en **Inicios de sesión** y, a continuación, haga clic en **Nuevo inicio de sesión**.



2. Aparecerá el cuadro de diálogo **Inicio de sesión - Nuevo**.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

3. En la página **General**, en el control **Nombre de inicio de sesión**, escriba un inicio de sesión de Windows con el formato **<domain>\<login>**.

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: EGCC2019\operador Search...

☒ Windows authentication
☐ SQL Server authentication

4. En el control **Base de datos predeterminada**, seleccione **AdventureWorks**, si está disponible. Si no lo está, seleccione **master**.

Progress

Ready

Default database: master

Default language: <default>

Remove

OK Cancel

5. En la página **Roles del servidor**, si el nuevo inicio de sesión va a ser administrador, haga clic en **sysadmin**; de lo contrario, déjelo en blanco.

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

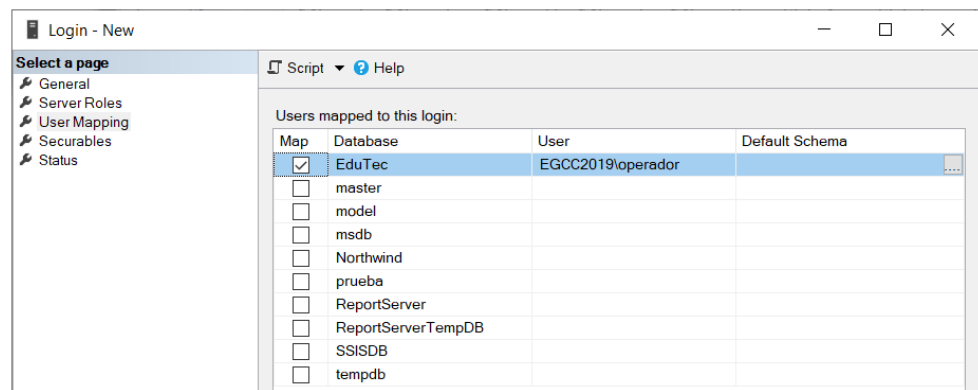
Script Help

Server role is used to grant server-wide security privileges to a user.

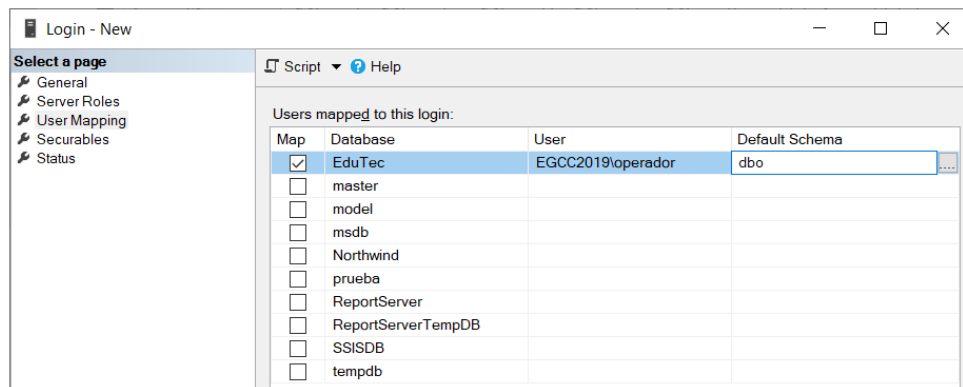
Server roles:

- ☐ bulkadmin
- ☐ dbcreator
- ☐ diskadmin
- ☐ processadmin
- ☒ public
- ☐ securityadmin
- ☐ serveradmin
- ☐ setupadmin
- ☒ sysadmin

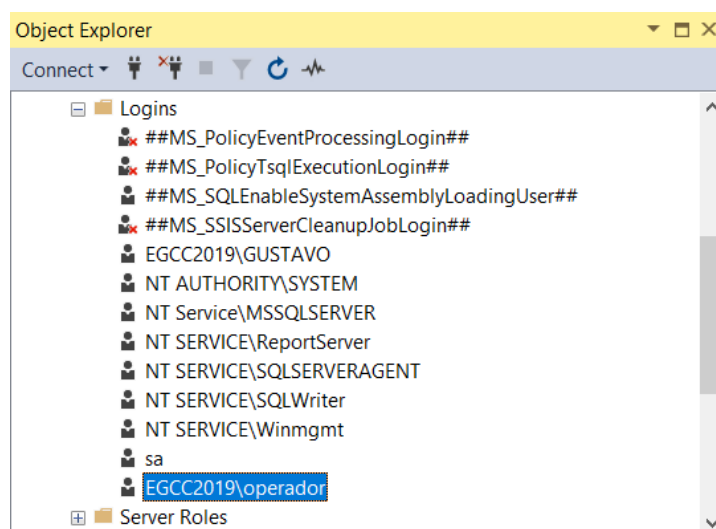
6. En la página **Asignación de usuarios**, seleccione **Asignar** para la base de datos **EduTec**, si está disponible. Si no lo está, seleccione master. Observe que en la columna **Usuario** se ha rellenado con el inicio de sesión. Al cerrar el cuadro de diálogo, se creará el usuario en la base de datos.



7. En el cuadro **Esquema predeterminado**, escriba **dbo** para asignar al usuario el esquema del propietario de la base de datos.

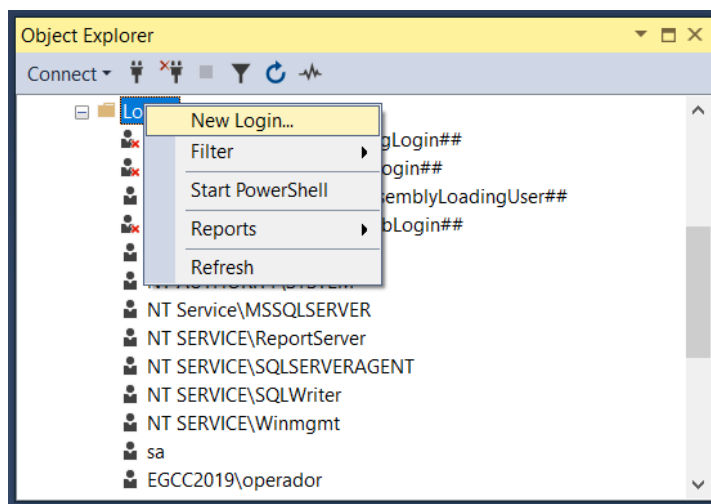


8. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.

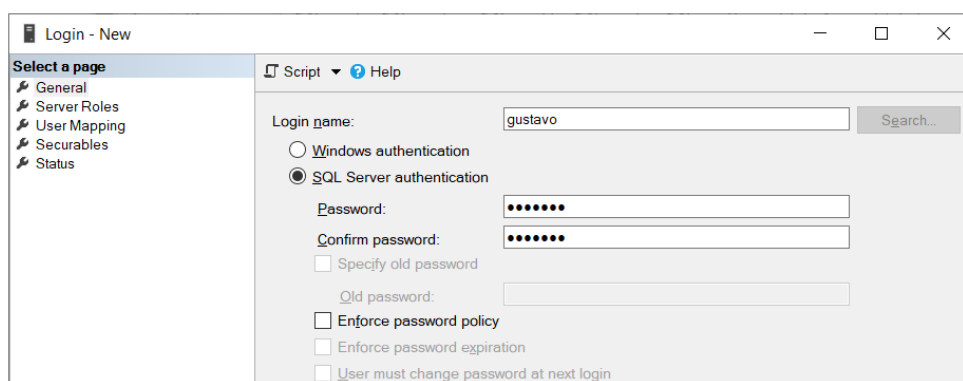


Crear un inicio de sesión con autenticación SQL Server

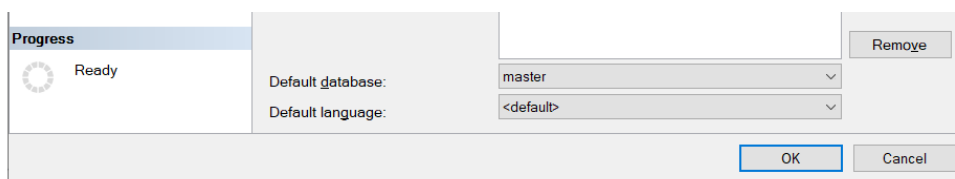
1. En el **Explorador de objetos**, expanda la instancia de servidor, expanda **Seguridad**, haga clic con el botón secundario en **Inicios de sesión** y, a continuación, haga clic en **Nuevo inicio de sesión**.



2. Aparecerá el cuadro de diálogo **Inicio de sesión - Nuevo**.
3. En la página **General**, en el cuadro **Nombre de inicio de sesión**, escriba un inicio de sesión, por ejemplo, **gustavo**. Seleccione **Autenticación de SQL Server** y escriba una contraseña para el nuevo inicio de sesión, también debe deshabilitar la opción **Exigir expiración de contraseña**.



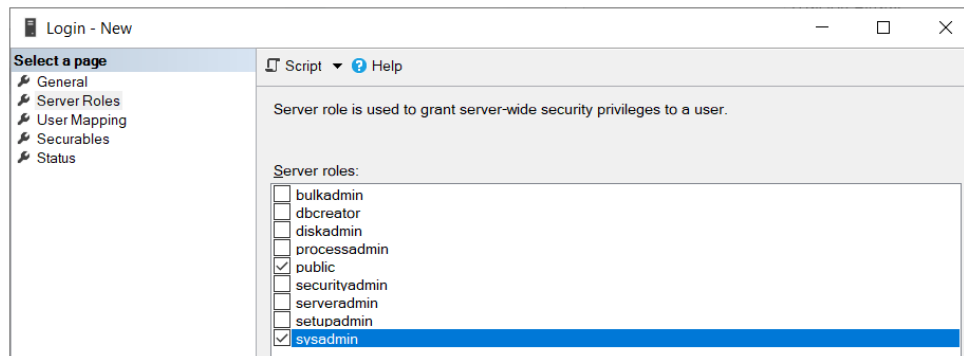
4. En el cuadro **Base de datos predeterminada**, seleccione **EduTec**, si está disponible. Si no lo está, seleccione **master**.



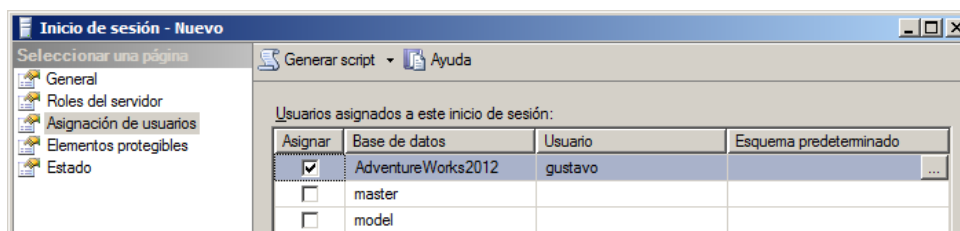


ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

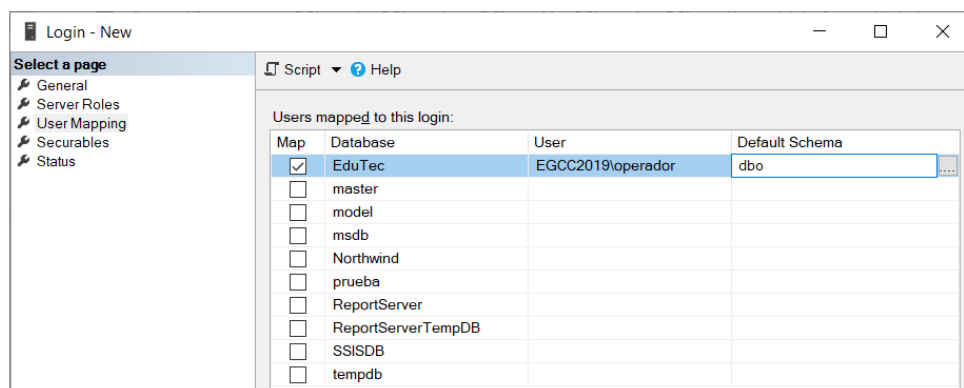
5. En la página **Roles del servidor**, si el nuevo inicio de sesión va a ser administrador, haga clic en **sysadmin**; de lo contrario, déjelo en blanco.



6. En la página **Asignación de usuarios**, seleccione **Asignar** para la base de datos EduTec, si está disponible. si no lo está, seleccione master. Observe que el cuadro **Usuario** se ha rellenado con el inicio de sesión. Al cerrar el cuadro de diálogo, se creará el usuario en la base de datos.



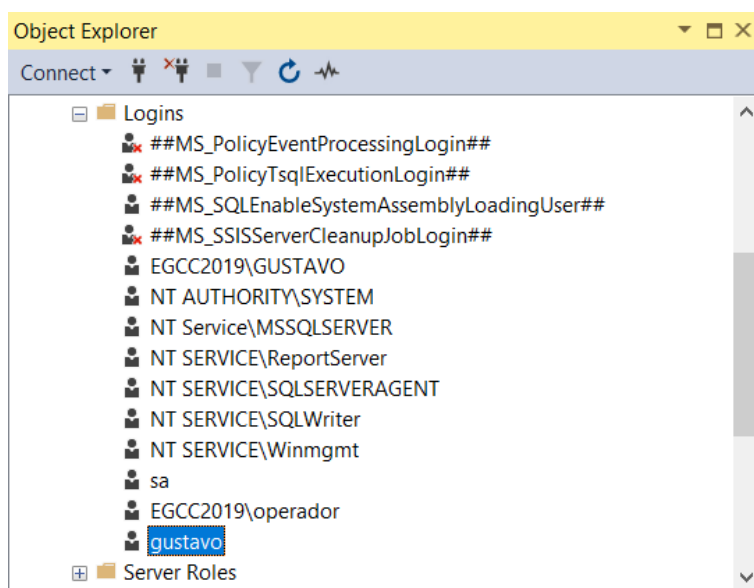
7. En el cuadro **Esquema predeterminado**, escriba **dbo** para asignar al usuario el esquema del propietario de la base de datos.



8. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.

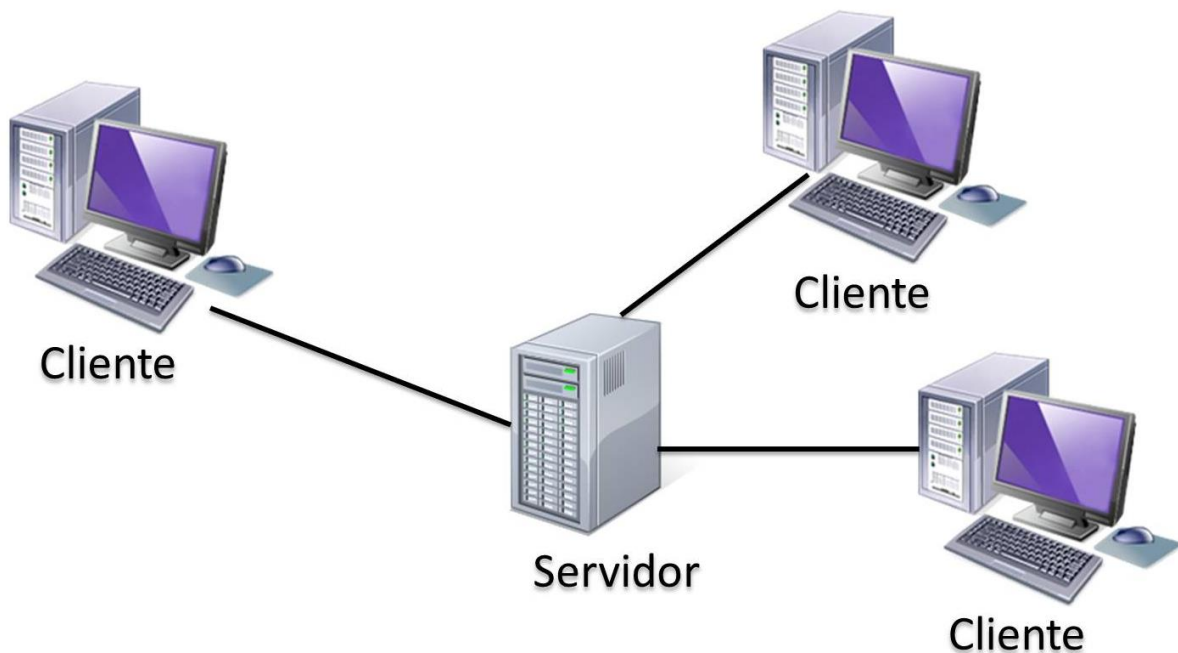


ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER





CONEXIÓN REMOTA



Contexto

Cuando se instala SQL Server, por defecto, para mejorar la seguridad, no se puede obtener acceso al motor de base de datos desde otros equipos de la red. En esta sección aprenderás a habilitar los protocolos, configurar los puertos y configurar el Firewall de Windows para conectarse desde otros equipos.

Habilitar protocolos

Para mejorar la seguridad, SQL Server se instalan con conectividad de red limitada. Las conexiones al motor de base de datos se pueden realizar desde herramientas que se ejecuten en el mismo equipo, no desde otros equipos. Si tiene previsto realizar las tareas de desarrollo en el mismo equipo que motor de base de datos, no necesita habilitar otros protocolos. Management Studio se conectará a motor de base de datos mediante el protocolo de memoria compartida. Este protocolo ya está habilitado.

Si tiene previsto conectarse al motor de base de datos desde otro equipo, debe habilitar un protocolo, como TCP/IP.

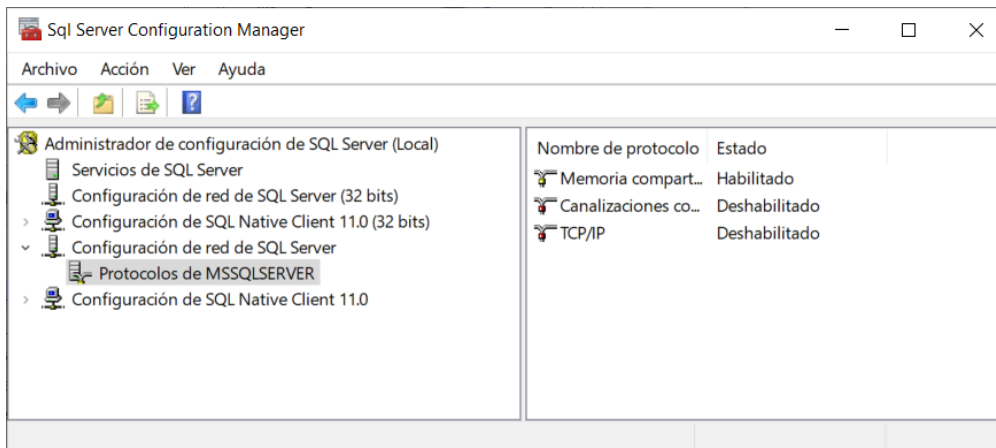
Cómo habilitar conexiones TCP/IP desde otro equipo:

1. En el menú **Inicio**, elija **Todos los programas**, **Microsoft SQL Server**, por último, **Administrador de configuración de SQL Server**.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

2. En **Administrador de configuración de SQL Server**, expanda **Configuración de red de SQL Server** y, a continuación, haga clic en **Protocolos de <NombreInstancia>**.

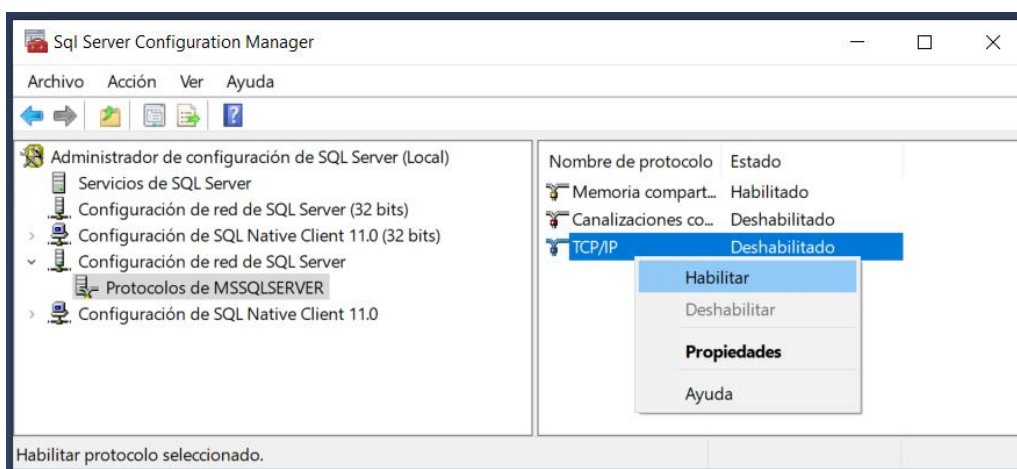


La instancia predeterminada (una instancia sin nombre) aparece como **MSSQLSERVER**. Si ha instalado una instancia con nombre, el nombre proporcionado aparece en la lista. Las versiones EXPRESS se instala como **SQLEXPRESS**, a menos que se haya cambiado el nombre durante la instalación.

3. En la lista de protocolos, haga clic con el botón secundario en el protocolo que desee habilitar (**TCP/IP**) y, a continuación, haga clic en **Habilitar**.

Nota

Debe reiniciar el servicio **SQL Server** después de realizar los cambios en los protocolos de red; sin embargo, esto se completa en la siguiente tarea.

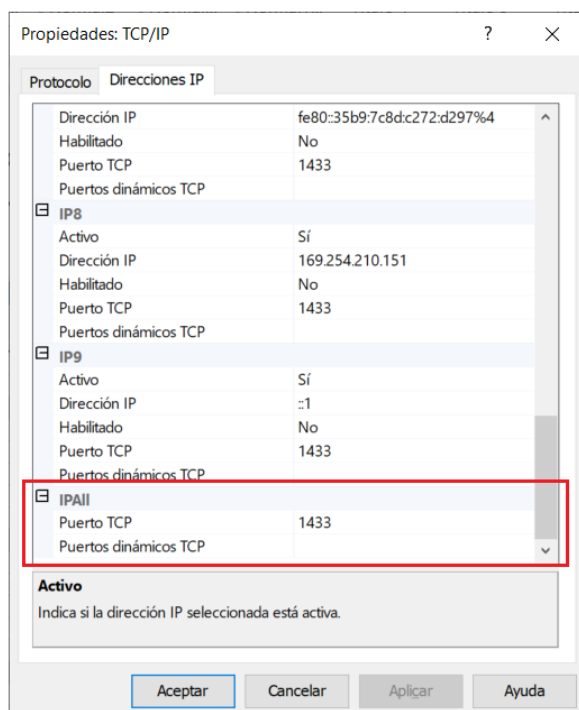


Configurar un puerto fijo

Para mejorar la seguridad, está activo el Firewall de Windows. Si desea conectarse a esta instancia desde otro equipo, debe abrir un puerto de comunicaciones en el firewall. La instancia predeterminada del motor de base de datos escucha en el puerto **1433**; por tanto, no tiene que configurar un puerto fijo. No obstante, las instancias con nombre incluidas las de SQL Server Express escuchan en puertos dinámicos. Para poder abrir un puerto en el firewall, debe configurar primero que el motor de base de datos escuche en un puerto específico conocido como puerto fijo o estático; de lo contrario, es posible que el Motor de base de datos escuche en un puerto distinto cada vez que se inicie.

Para configurar SQL Server para escuchar en un puerto específico:

1. En el **Administrador de configuración de SQL Server**, expanda **Configuración de red de SQL Server** y, a continuación, haga clic en la instancia de servidor que desee configurar, por ejemplo, **Protocolos de MSSQLSERVER**.
2. En el panel derecho, haga doble clic en **TCP/IP**.
3. En el cuadro de diálogo **Propiedades de TCP/IP**, haga clic en la pestaña **Direcciones IP**.

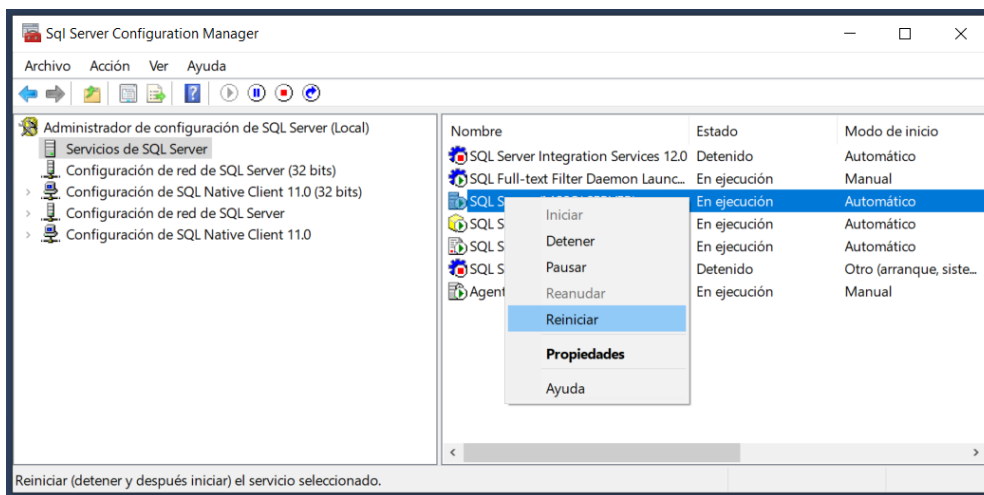


4. En el cuadro **Puerto TCP** de la sección **IPAll**, escriba un número de puerto disponible. Para este tutorial, se utilizará 49172.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

5. Haga clic en **Aceptar** para cerrar el cuadro de diálogo y en **Aceptar** cuando aparezca una advertencia que indique que debe reiniciarse el servicio.
6. En el panel izquierdo, haga clic en **Servicios de SQL Server**.
7. En el panel derecho, haga clic con el botón secundario en la instancia de SQL Server y, a continuación, haga clic en **Reiniciar**. Cuando se reinicia el motor de base de datos, escucha en el puerto 49172.



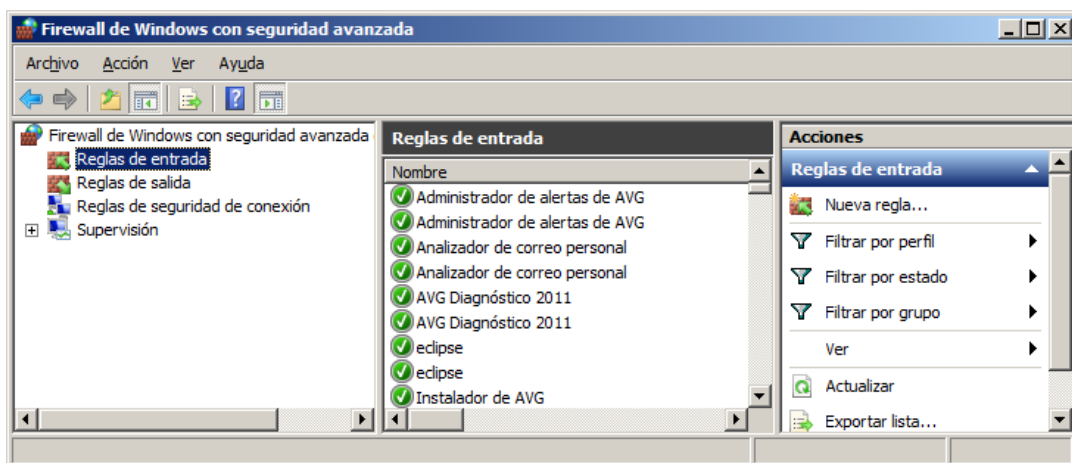
Abrir puertos del firewall

Los sistemas FIREWALL ayudan a evitar el acceso no autorizado a los recursos de los equipos.

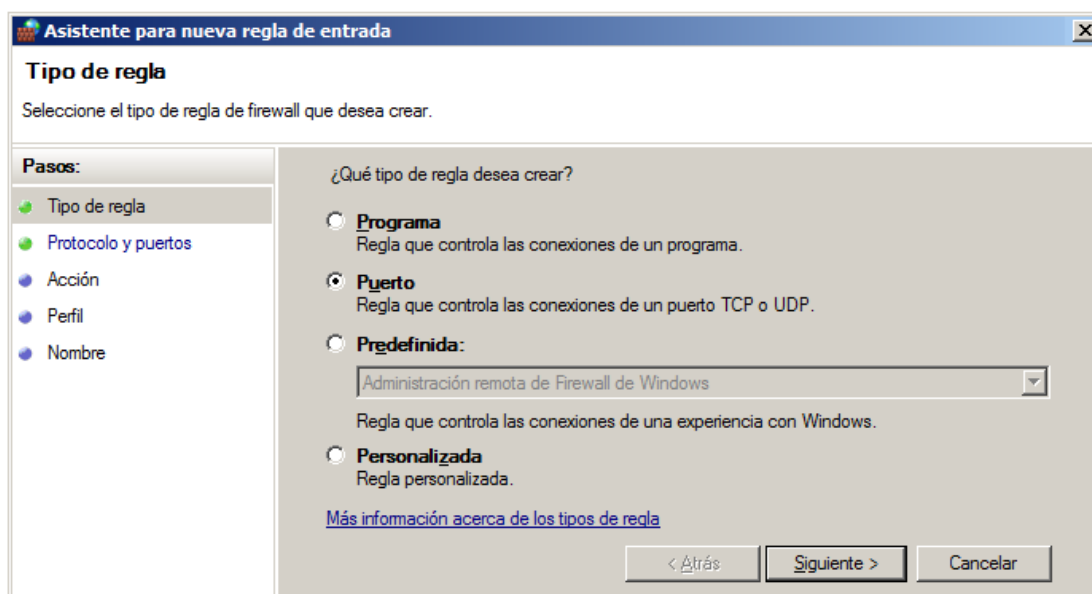
Para conectarse a SQL Server desde otro equipo cuando el firewall está activo, debe abrir un puerto en el firewall.

Después de configurar el motor de base de datos para utilizar un puerto fijo, siga estas instrucciones para abrir ese puerto en el Firewall de Windows. (No es necesario configurar un puerto fijo para la instancia predeterminada, porque ya está fijada en el puerto TCP 1433).

1. En el menú **Inicio**, haga clic en **Ejecutar**, escriba **WF.msc** y, a continuación, haga clic en **Aceptar**.



- En la opción **Firewall de Windows con seguridad avanzada** del panel izquierdo, haga clic en **Reglas de entrada** y, a continuación, haga clic en **Nueva regla** en el panel **Acciones**.
- En el cuadro de diálogo **Tipo de regla**, seleccione **Puerto** y, a continuación, haga clic en **Siguiente**.





ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

4. En el cuadro de diálogo **Protocolo y puertos**, seleccione **TCP**. Seleccione **Puertos locales específicos** y, a continuación, escriba el número de puerto de la instancia del motor de base de datos. Use **1433** para la instancia predeterminada. Escriba **49172** si está configurando una instancia con nombre y ha configurado el puerto fijo de la tarea anterior. Haga clic en **Siguiente**.

The screenshot shows the 'Asistente para nueva regla de entrada' (New Incoming Rule Wizard) dialog box, specifically the 'Protocolo y puertos' (Protocol and ports) step. The left pane shows the 'Pasos' (Steps) list with 'Protocolo y puertos' selected. The main area contains the following options:

- ¿Se aplica esta regla a TCP o UDP?
 - ☒ **TCP**
 - ☐ **UDP**
- ¿Se aplica esta regla a todos los puertos locales o a unos puertos locales específicos?
 - ☐ **Todos los puertos locales**
 - ☒ **Puertos locales específicos:**
Ejemplo: 80, 443, 5000-5010

At the bottom, there are three buttons: '< Atrás', 'Siguiente >' (highlighted), and 'Cancelar'. A link 'Más información acerca de protocolos y puertos' is also present.

5. En el cuadro de diálogo **Acción**, seleccione **Permitir la conexión** y, a continuación, haga clic en **Siguiente**.

The screenshot shows the 'Asistente para nueva regla de entrada' (New Incoming Rule Wizard) dialog box, specifically the 'Acción' (Action) step. The left pane shows the 'Pasos' (Steps) list with 'Acción' selected. The main area contains the following options:

- ¿Qué medida debe tomarse si una conexión coincide con las condiciones especificadas?
 - ☒ **Permitir la conexión**
Esto incluye las conexiones protegidas mediante IPsec y las que no lo están.
 - ☐ **Permitir la conexión si es segura**
Esto incluye solamente las conexiones autenticadas mediante IPsec. Éstas se protegerán mediante la configuración de reglas y propiedades de IPsec del nodo Regla de seguridad de conexión.
 - ☐ **Bloquear la conexión**

At the bottom, there are three buttons: '< Atrás', 'Siguiente >' (highlighted), and 'Cancelar'. A link 'Más información acerca de las acciones' is also present.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

6. En el cuadro de diálogo **Perfil**, seleccione los perfiles que describan el entorno de conexión del equipo cuando desee conectarse al motor de base de datos y, a continuación, haga clic en **Siguiente**.

The screenshot shows the 'Asistente para nueva regla de entrada' (New Incoming Rule Wizard) dialog box, specifically the 'Perfil' (Profile) step. The title bar reads 'Asistente para nueva regla de entrada'. The main heading is 'Perfil' with the instruction 'Especifique los perfiles en los que se va a aplicar esta regla.' (Specify the profiles to which this rule will be applied). On the left, a 'Pasos:' (Steps) pane lists: 'Tipo de regla' (Rule type), 'Protocolo y puertos' (Protocol and ports), 'Acción' (Action), 'Perfil' (Profile), and 'Nombre' (Name). The 'Perfil' step is currently selected. The main area asks '¿Cuándo se aplica esta regla?' (When does this rule apply?). It contains three options: 'Dominio' (Domain) with a checked checkbox and description 'Se aplica cuando un equipo está conectado a su dominio corporativo.'; 'Privado' (Private) with a checked checkbox and description 'Se aplica cuando un equipo está conectado a una ubicación de redes privadas.'; and 'Público' (Public) with an unchecked checkbox and description 'Se aplica cuando un equipo está conectado a una ubicación de redes públicas.' Below these is a link 'Más información acerca de los perfiles' (More information about profiles). At the bottom are three buttons: '< Atrás' (Back), 'Siguiente >' (Next), and 'Cancelar' (Cancel).

7. En el cuadro de diálogo **Nombre**, escriba un nombre y una descripción para esta regla. Después, haga clic en **Finalizar**.

The screenshot shows the 'Asistente para nueva regla de entrada' (New Incoming Rule Wizard) dialog box, specifically the 'Nombre' (Name) step. The title bar reads 'Asistente para nueva regla de entrada'. The main heading is 'Nombre' with the instruction 'Especifique el nombre y la descripción de esta regla.' (Specify the name and description of this rule). On the left, the 'Pasos:' (Steps) pane is the same as in the previous step, but 'Nombre' is now selected. The main area has two input fields: 'Nombre:' (Name) with the text 'SQL Server Laboratorio' entered, and 'Descripción (opcional):' (Description (optional)) with the text 'Configuración de puerto para prueba de laboratorio.' entered. At the bottom are three buttons: '< Atrás' (Back), 'Finalizar' (Finish), and 'Cancelar' (Cancel).

Conexión desde otro equipo

Ahora que ha configurado el motor de base de datos para escuchar en un puerto fijo y ha abierto este puerto en el firewall, puede conectarse a SQL Server desde otro equipo.

Cuando el servicio **SQL Server Browser** está ejecutándose en el equipo servidor y el firewall haya abierto el puerto UDP 1434, la conexión se podrá realizar utilizando el nombre del equipo y el nombre de la instancia. Para mejorar la seguridad, el ejemplo no utiliza el servicio SQL Server Browser.

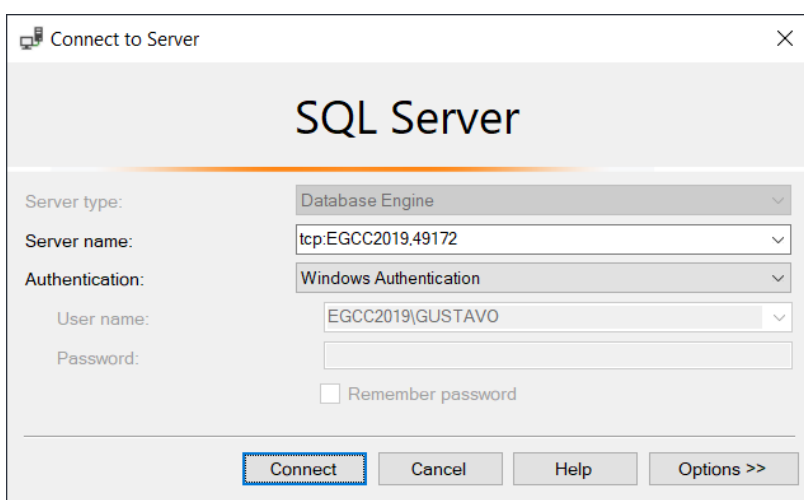
Para conectarse al motor de base de datos desde otro equipo:

1. En un segundo equipo que incluya las herramientas de cliente de SQL Server, inicie una sesión con una cuenta autorizada para conectarse a SQL Server y abra **Management Studio**.
2. En el cuadro de diálogo **Conectar con el servidor**, confirme **Motor de base de datos** en el cuadro **Tipo de servidor**.
3. En el cuadro **Nombre del servidor**, escriba **tcp:** para especificar el protocolo, seguido del nombre del equipo, una coma y el número de puerto. Para conectarse a la instancia predeterminada, el puerto 1433 está implícito y se puede omitir, por lo que deberá escribir **tcp:<nombre_computador>**. En nuestro ejemplo, escriba **tcp:<computer_name>,49172**.

Nota

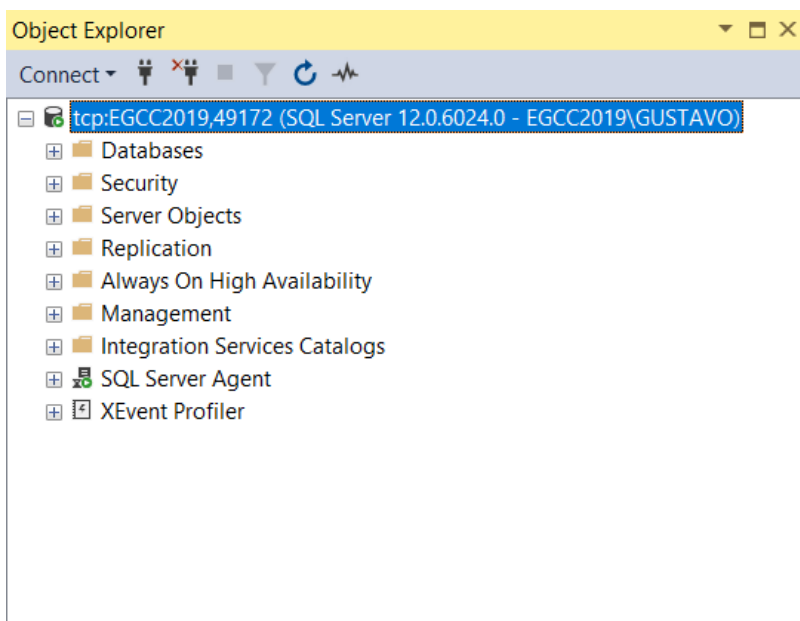
Si omite **tcp:** en el cuadro **Nombre del servidor**, el cliente probará todos los protocolos habilitados en el orden especificado en la configuración del cliente.

4. En el cuadro **Autenticación**, confirme **Autenticación de Windows** y, a continuación, haga clic en **Conectar**.

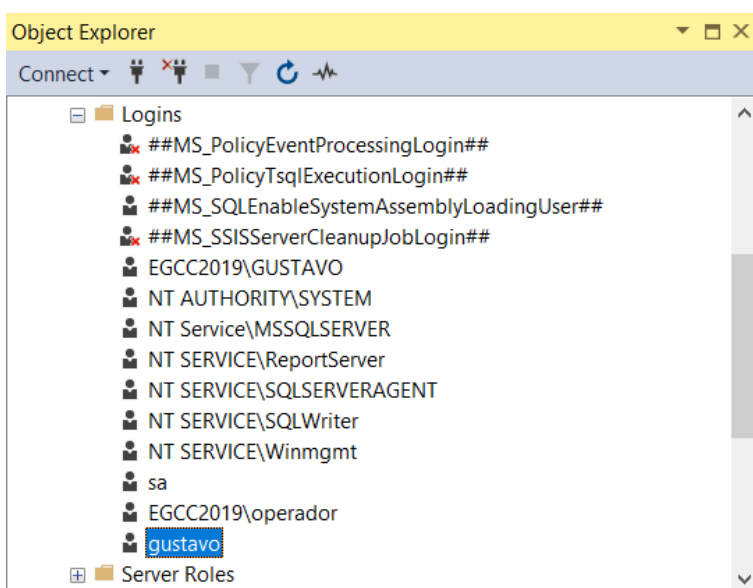




5. El resultado lo puedes ver en la siguiente imagen.



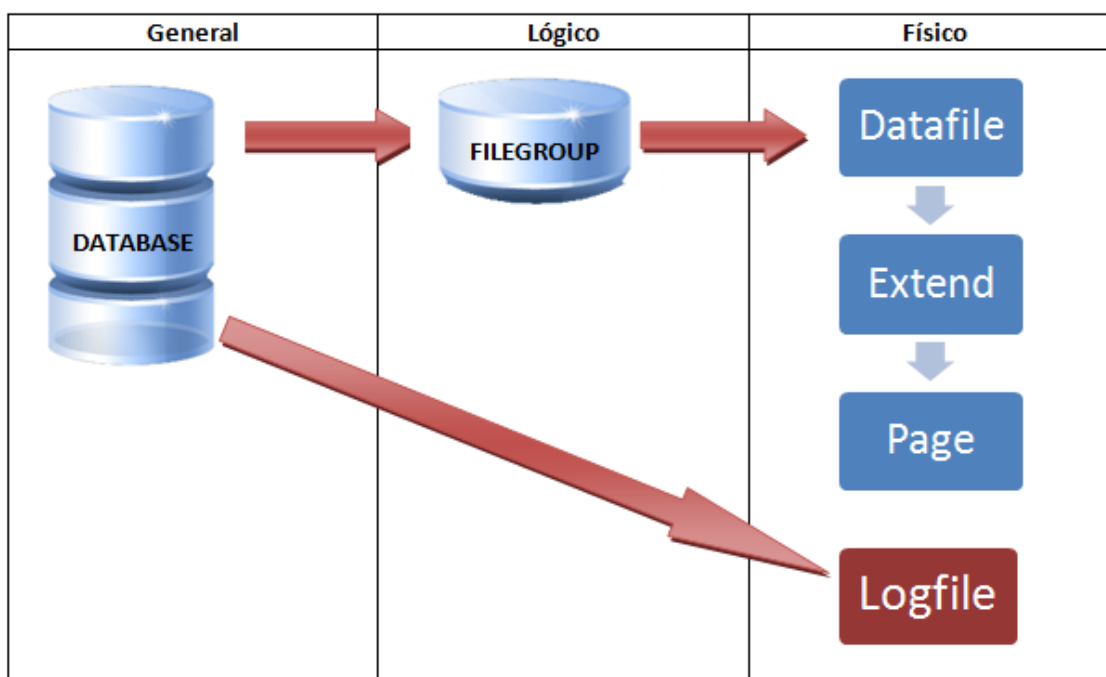
6. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.





Capítulo 3 TRABAJANDO CON BASES DE DATOS

ARQUITECTURA DE BASES DE DATOS SQL SERVER



La arquitectura interna de las bases de datos en SQL Server está compuesta por 2 tipos de estructura, la estructura lógica y la estructura física. Es muy importante conocer cómo es que estas estructuras están compuestas y cuál es la relación que tienen los objetos de base de datos con cada una de estas estructuras.

Estructura Lógica

Desde el punto de vista lógico, la base de datos debe tener al menos 1 "FileGroup" el cual contiene a toda la metadata de la misma base de datos, es decir tablas y vistas de sistema, a este "FileGroup" inicial se le conoce como "Primario" y está presente en todas las bases de datos.

Todos los objetos de usuario que contengan data, ya sean tablas o índices, deben estar ligados a un "FileGroup", esto se puede definir al momento de ejecutar la sentencia DDL de creación del objeto, si no se indica a que "FileGroup" estará ligado ese objeto, este pertenecerá al "FileGroup" por defecto definido en la base de datos. La base de datos solo puede tener definido 1 solo "FileGroup" por defecto.

Las bases de datos pueden tener hasta 32767 “FileGroups” definidos, según los límites establecidos. Uno de los propósitos de los “FileGroups” es poder distribuir la data a través de varios discos duros físicos, de esta manera se puede obtener mayor rendimiento en las operaciones de I/O debido a que más de un disco trabajara al mismo tiempo. Otro de los propósitos es poder esconder la ubicación física real de la información a los programadores, ya que para ellos la tabla “X” pertenece al “FileGroup” “A”, pero no saben en que data files físicamente se encuentra la información de la tabla “X”.

Nota

Para saber las capacidades máximas de SQL Sever, puedes consultar el siguiente enlace:

<https://docs.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server>

Los “FileGroups” pueden contener 1 o más “Datafiles”, y cada uno de estos datafiles se puede encontrar en discos diferentes, lo cual también agilizará las consultas y los ingresos de información a las tablas que se encuentren asignadas a este “FileGroup”, debido a que SQL Server distribuirá la información uniformemente a través de todos los “DataFiles” del “FileGroup”.

Estructura Física

Desde el punto de vista físico, como ya hemos visto, tenemos los “DataFiles” que son en realidad los archivos de datos, es decir donde se guarda toda la información de la base de datos. Un “DataFile” solo puede pertenecer a 1 “FileGroup”.

Internamente los “DataFiles” están divididos en “Extends” y estos a su vez en “Pages”. Las “Pages” son la unidad mínima de almacenamiento dentro de la base de datos. Un “Page” tiene 8 Kb de tamaño en espacio de disco. Un “Extend” tiene 8 “Pages” contiguas que lo conforman, es decir, un “Extend” tiene como tamaño 64 Kb de espacio en disco.

En un “Page” solo puede haber información de 1 sola tabla, es decir el espacio de un “Page” no es compartido entre tablas o índices. En el caso de los “Extends”, estos pueden ser de dos tipos:

- **Mixed:** Los cuales son compartidos hasta por 8 objetos, uno por cada “Page”.
- **Uniform:** Los cuales solo pertenecen a un solo objeto, es decir que todos los “Pages” pertenecen a un solo objeto.

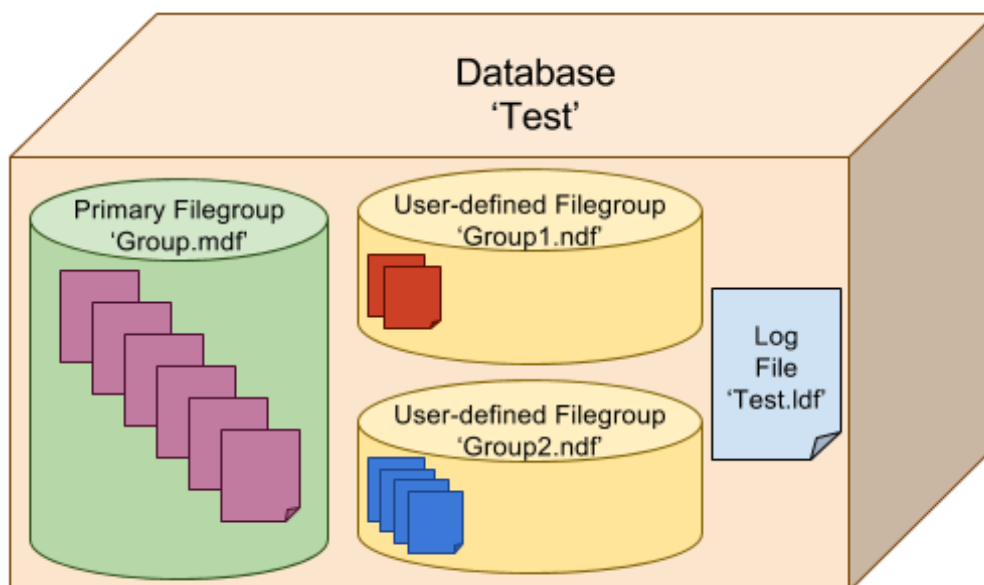
Normalmente cuando se crea una nueva tabla esta es asignada a un “Extend” de tipo “Mixed”, hasta alcanzar la utilización de hasta 8 “Pages”, a partir de ese momento se asignan “Extends” de tipo “Uniform” para optimizar el uso del espacio en la tabla.

Los “DataFiles” normalmente tienen 2 extensiones de archivo, las cuales son estándar mas no obligatorias, la extensión **“mdf”** que se utiliza para el primer “Datafile” perteneciente al “FileGroup” primario, y la extensión **“ndf”** que se utiliza para los demás datafiles que se agregan posteriormente a los demás “FileGroups” de la base de datos.

En el caso del “LogFile”, este no pertenece a un “FileGroup” en específico, en cambio archivo está ligado directamente a la base de datos. Las bases de datos de SQL Server solo pueden tener un solo “LogFile” activo al mismo tiempo, si bien se pueden crear múltiples “LogFiles” en la base de datos, solo uno podrá ser escrito, ya que solo uno puede estar activo, cuando este archivo se llene, la base de datos pasará a escribir al siguiente archivo de transacciones, y así sucesivamente. Por esta razón no es muy conveniente ni útil tener más de un “LogFile”.



CREACIÓN DE UNA BASE DE DATOS



Contexto

Ahora conocerás los fundamentos de la creación de una base de datos en SQL SERVER, así mismo realizaras una práctica sobre cómo crear una base de datos con la herramienta gráfica Management Studio y Transact-SQL.

Lógicamente es necesario que tengamos instalado el servidor de base de datos o dicho de otra manera: Microsoft SQL Server en cualquier versión, usualmente al instalar SQL SERVER ya viene incluida la herramienta Management Studio que te permite conectarte y manejar tu servidor SQL Server desde una interfaz gráfica en lugar de tener que usar la línea de comandos.

Concepto de base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, en otros términos, una base de datos es un “ALMACEN” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrarla y utilizarla fácilmente.

Crear base de datos usando T-SQL

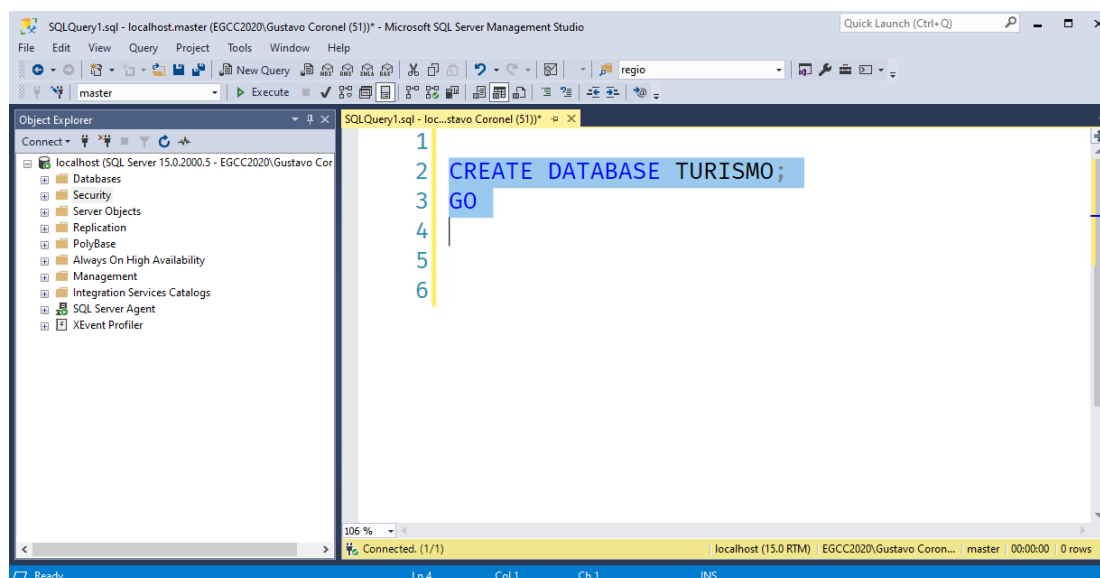
T-SQL es el lenguaje de manipulación de los datos en SQL Server, cada SGBD tiene un lenguaje propio, así, pues si utilizamos ORACLE el lenguaje de manipulación será el PL-SQL. Sin embargo, hay un lenguaje estándar llamado SQL y ese, aunque con limitantes en el sentido que no provee muchas funciones que los propios de las SGBD tienen, funciona en cualquier SGBD.

T-SQL o PL-SQL es el nombre dado a la modificación del lenguaje SQL por Microsoft y Oracle respectivamente, la esencia del lenguaje se mantiene, es por ello que existen muchas similitudes entre ambos. Ahora nos centraremos en el T-SQL.

Aprenderemos la forma de hacerlo mediante las Querys o consultas:

- Lo Primero que hacemos es dar click al botón **New Query** (o Nueva Consulta) y abrimos una nueva consulta.
- Debemos crear una base de datos y luego ponerla en uso entonces escribimos lo siguiente y lo ejecutamos:

```
CREATE DATABASE TURISMO;
```





ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Para ejecutar una sentencia, la seleccionamos con el mouse (toda la sentencia) y presionamos el botón **Ejecutar**.

```
-- Crea la base de datos

CREATE DATABASE TURISMO;
GO

-- Activar la BD TURISMO para poder hacer modificaciones

USE TURISMO;
GO
```

A continuación, tienes el script de creación de la base de datos TURISMO, pero indicando de manera explícita otros parámetros, como la ubicación de los archivos que la conforman:

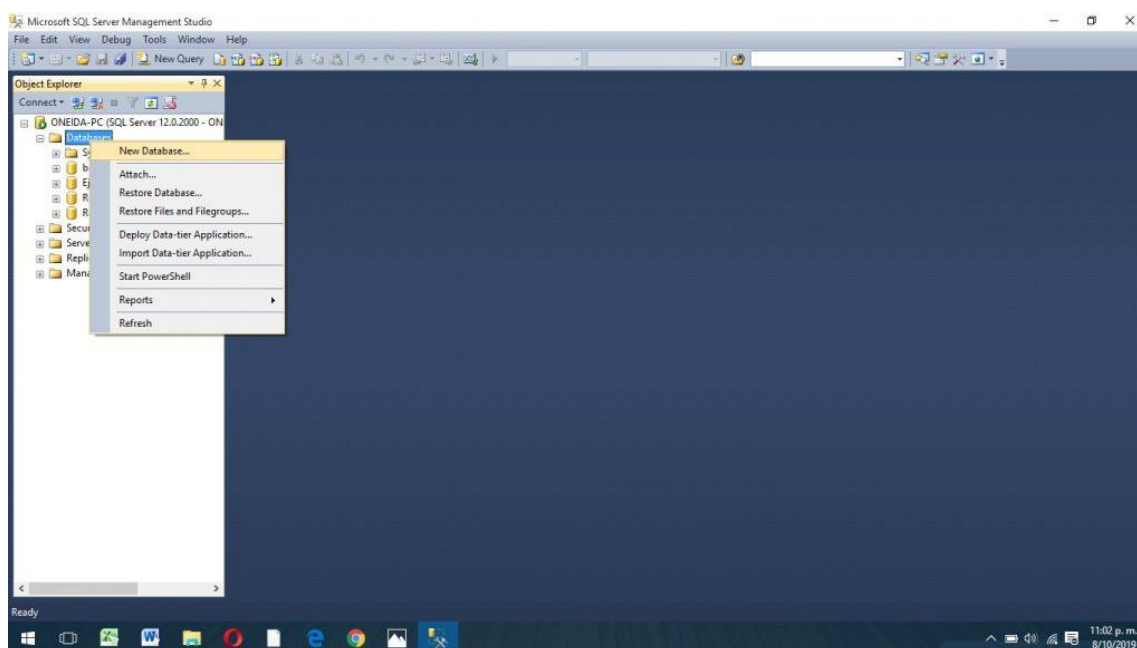
```
USE master;
GO

CREATE DATABASE TURISMO
ON
( NAME = TURIMO,
  FILENAME = 'E:\SQLServerDB\TURISMO.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = TURISMOLOG,
  FILENAME = 'E:\SQLServerDB\TURISMOLOG.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB );
GO
```

Creación una base de datos usando SSMS

Paso a seguir:

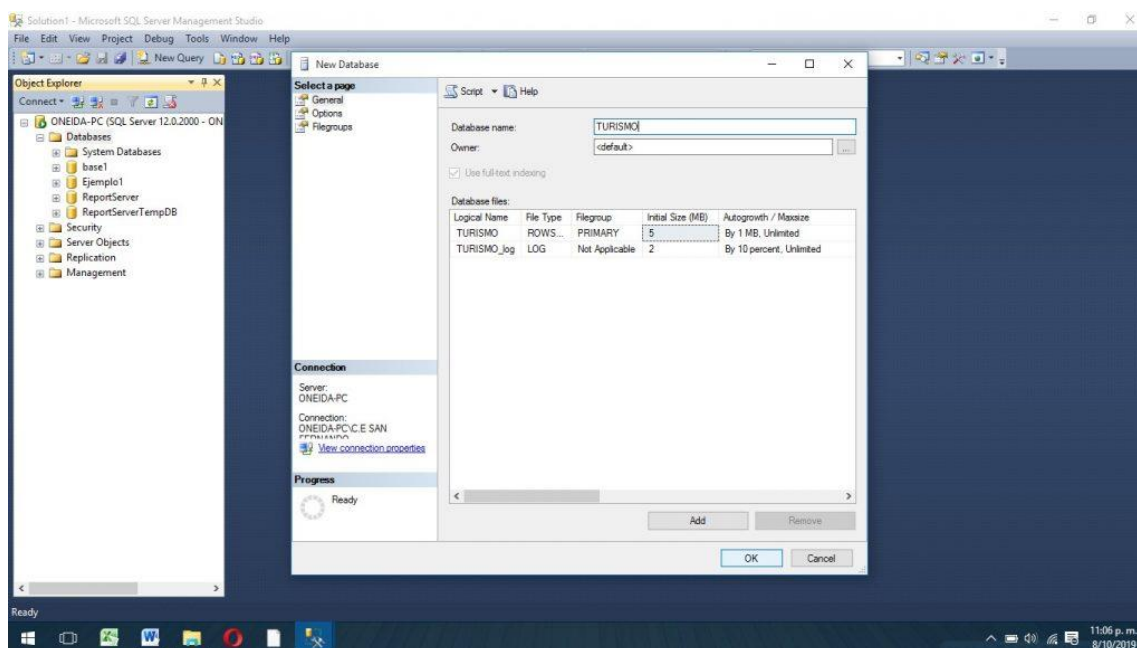
1. En el **Explorador de objetos**, conéctese a una instancia del motor de base de datos de SQL Server y expándala.
2. Hacer click con el botón derecho del Mouse en **Bases de datos** y luego hacer click en **Nueva base de datos** o **New database**.





ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

3. En el diálogo **Nueva base de datos**, especifique un nombre para la base de datos.



4. Si desea crear la base de datos aceptando todos los valores predeterminados, hacer click en el botón **Aceptar**; de lo contrario, continúe con siguientes los pasos opcionales.

Básicamente esos son los pasos para crear una base de datos en SQL Server de forma gráfica.

Es importante mencionar que ya teniendo creada la base de datos vacía, lo que corresponde hacer es crear las tablas en las cuales se almacena y guarda la información de forma organizada.

GESTIÓN DE ARCHIVOS

Grupos de archivos

El grupo de archivos contiene el archivo de datos principal y cualquier otro archivo secundario que no se encuentre en otro grupo de archivos.

Se pueden crear grupos de archivos definidos por el usuario para agrupar archivos con fines administrativos y de asignación y ubicación de datos.

Por ejemplo, **Data1.ndf**, **Data2.ndf** y **Data3.ndf** pueden crearse en tres unidades de disco, respectivamente, y asignarse al grupo de archivos **fgroup1**. Se puede crear una tabla específicamente para el grupo de archivos **fgroup1**. Las consultas de datos de la tabla se distribuirán por los tres discos, con lo que mejorará el rendimiento. Puede obtenerse la misma mejora del rendimiento con un solo archivo creado en un conjunto de bandas RAID (matriz redundante de discos independientes). No obstante, los archivos y grupos de archivos permiten agregar fácilmente nuevos archivos a discos nuevos.

Todos los archivos de datos se almacenan en los grupos de archivos que se indican en la tabla siguiente.

GRUPO DE ARCHIVOS	DESCRIPCIÓN
Principal	Grupo de archivos que contiene el archivo principal. Todas las tablas del sistema forman parte del grupo de archivos principal.
Tabla optimizada para memoria	Un grupo de archivos optimizados para memoria está basado en un grupo de archivos de FILESTREAM.
Definidos por el usuario	Cualquier grupo de archivos creado por el usuario al crear la base de datos o al modificarla.

Grupo de archivos predeterminado

Cuando se crean objetos en la base de datos sin especificar a qué grupo de archivos pertenecen, se asignan al grupo de archivos predeterminado. Siempre existe un grupo de archivos designado como predeterminado. Los archivos del grupo de archivos predeterminado deben ser lo suficientemente grandes como para dar cabida a todos los objetos nuevos no asignados a otros grupos de archivos.

El grupo de archivos PRINCIPAL es el predeterminado, a menos que se cambie mediante la instrucción ALTER DATABASE. Los objetos y las tablas del sistema no

se asignan al nuevo grupo de archivos predeterminado, sino que siguen asignados al grupo de archivos PRIMARY.

Ejemplo de archivos y grupos de archivos

En el siguiente ejemplo se crea una base de datos en una instancia de SQL Server. La base de datos tiene un archivo de datos principal, un grupo de archivos definido por el usuario y el archivo de registro.

El archivo de datos principal está en el grupo de archivos principal y el grupo de archivos definido por el usuario tiene dos archivos de datos secundarios. Una instrucción ALTER DATABASE hace que el grupo de archivos definido por el usuario sea el grupo predeterminado. A continuación, se crea una tabla que especifica el grupo de archivos definido por el usuario.

En este ejemplo se usa una ruta de acceso **E:\SQLServerDB** para los archivos.

Creación de la base de datos:

```
USE master;
GO

CREATE DATABASE TURISMO
ON PRIMARY
( NAME='TURISMO',
  FILENAME= 'E:\SQLServerDB\TURISMO.mdf',
  SIZE=4MB,
  MAXSIZE=10MB,
  FILEGROWTH=1MB),
FILEGROUP TURISMO_FG1
( NAME = 'TURISMO_FG1_Data1',
  FILENAME = 'E:\SQLServerDB\TURISMO_FG1_Data1.ndf',
  SIZE = 1MB,
  MAXSIZE=10MB,
  FILEGROWTH=1MB),
( NAME = 'TURISMO_FG1_Data2',
  FILENAME = 'E:\SQLServerDB\TURISMO_FG1_Data2.ndf',
  SIZE = 1MB,
  MAXSIZE=10MB,
  FILEGROWTH=1MB)
LOG ON
( NAME='TURISMOLOG',
  FILENAME = 'E:\SQLServerDB\TURISMOLOG.ldf',
```



```
SIZE=1MB,  
MAXSIZE=10MB,  
FILEGROWTH=1MB);  
GO
```

Cambiar el file group por defecto:

```
ALTER DATABASE TURISMO  
MODIFY FILEGROUP TURISMO_FG1 DEFAULT;  
GO
```

Creando una tabla en el file group de usuario:

```
USE TURISMO;  
CREATE TABLE MyTable  
(  
    id int PRIMARY KEY,  
    dato varchar(100)  
)  
ON TURISMO_FG1;  
GO
```



PRACTICA

Ejercicio 1

Crear la base de datos **LibroTSQL** y, como su nombre indica, se creará con la ayuda de un script Transact SQL. Crearemos una segunda base de datos, llamada LibroSSMS, desde la interfaz gráfica del Management Studio.

Los archivos de datos se definirán en un directorio específico. Conviene crear el directorio **c:\datos**. Evidentemente, guardar los datos en el disco **c:** es producto del ejemplo pedagógico.

Parámetros de la base de datos LibroTSQL

Archivo de datos:

- Tamaño: 10 MB
- Nombre físico: c:\datos\LibroTSQL.mdf
- Nombre lógico: LibroTSQL

Archivo de log:

- Tamaño: 8 MB
- Nombre físico: c:\datos\LibroTSQL_log.ldf
- Nombre lógico: LibroTSQL_log



Parámetros de la base de datos LibroSSMS

Archivo de datos:

- Tamaño: 15 MB
- Nombre físico: c:\datos\LibroSSMS.mdf
- Nombre lógico: LibroSSMS

Archivo de log:

- Tamaño: 8 MB
- Nombre físico: c:\datos\LibroSSMS_log.ldf
- Nombre lógico: LibroSSMS_log

Ejercicio 2

En la base de datos LibroTSQL, añadir el grupo de archivos **Data**. Este grupo de archivos se compone de dos archivos: **data1.ndf** y **data2.ndf**.

El archivo **data1.ndf** posee un tamaño fijo de **50 MB** mientras que el archivo **data2.ndf** tiene un tamaño inicial de **10 MB** que puede después crecer hasta un tamaño de **50 MB** en pasos de **10 MB**.

Capítulo 4

GESTIÓN DE SEGURIDAD

CONTEXTO



Para gestionar la seguridad en SQL Server debes entender los 3 tipos de elementos o entidades que se han definido.

Securables

Son los objetos a los que debes asegurar el acceso. Por ejemplo, una tabla es asegurable. Algunos elementos protegibles pueden estar contenidos dentro de otros, creando jerarquías anidadas llamadas "ámbitos" que pueden protegerse ellos mismos. Los ámbitos protegibles son servidor, base de datos y esquema .

Principals

Son entidades de seguridad que acceden y ejecutan acciones sobre los objetos. Al igual que otros componentes del modelo de autorización de SQL Server, los principals se pueden organizar en una jerarquía. Un inicio de sesión es un ejemplo de un principal.

Permissions

Cada asegurable de SQL Server tiene permisos asociados que se pueden otorgar a un principal.



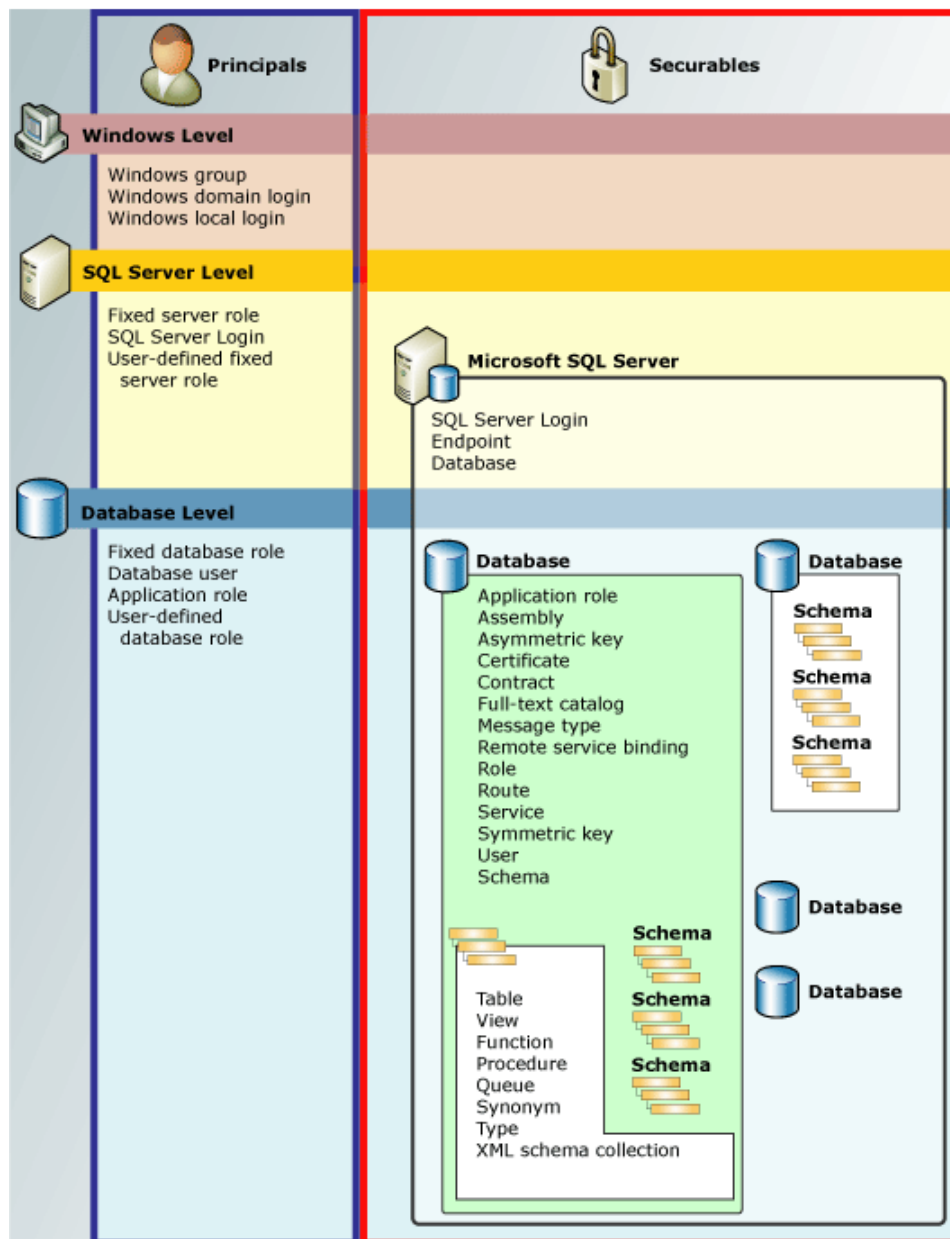
MODELO DE SEGURIDAD EN SQL SERVER



Para poder acceder a un asegurable y poder ejecutar alguna acción, como por ejemplo, realizar un consulta (SELECT) sobre una tabla se tienen cuatro niveles de seguridad como se ilustra en la imagen adjunta.



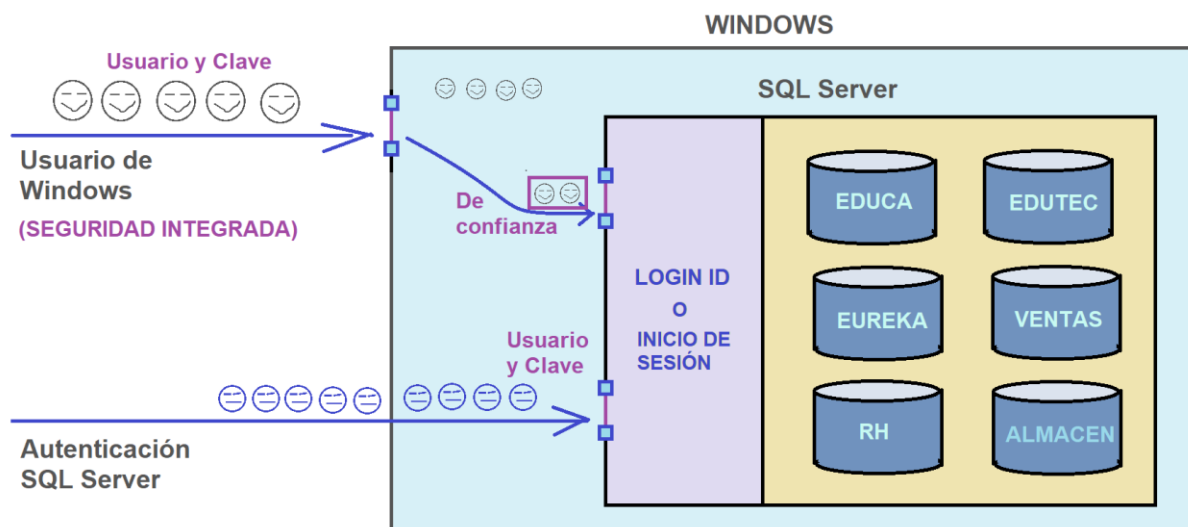
JERARQUIA DE SEGURIDAD



- **Principals:** Entidades de seguridad: Usuarios windows, usuarios sql server, Usuarios de BD.
- **Securables:** Recursos que pueden ser protegidos



TIPOS DE AUTENTICACIÓN



Autenticación Windows

En este caso es el Sistema Operativo Windows el que se encarga de validar el usuario y clave. Para poder iniciar sesión en SQL Server, es necesario haber iniciado sesión en Windows y estar autorizado el acceso a SQL Server.

Autenticación SQL Server

En este caso es el servidor SQL Server el encargado de validar el usuario y clave. Para poder iniciar sesión en SQL Server, es necesario tener un Login ID en el servidor. No es necesario iniciar sesión en Windows.

Inicios de sesión preestablecidos

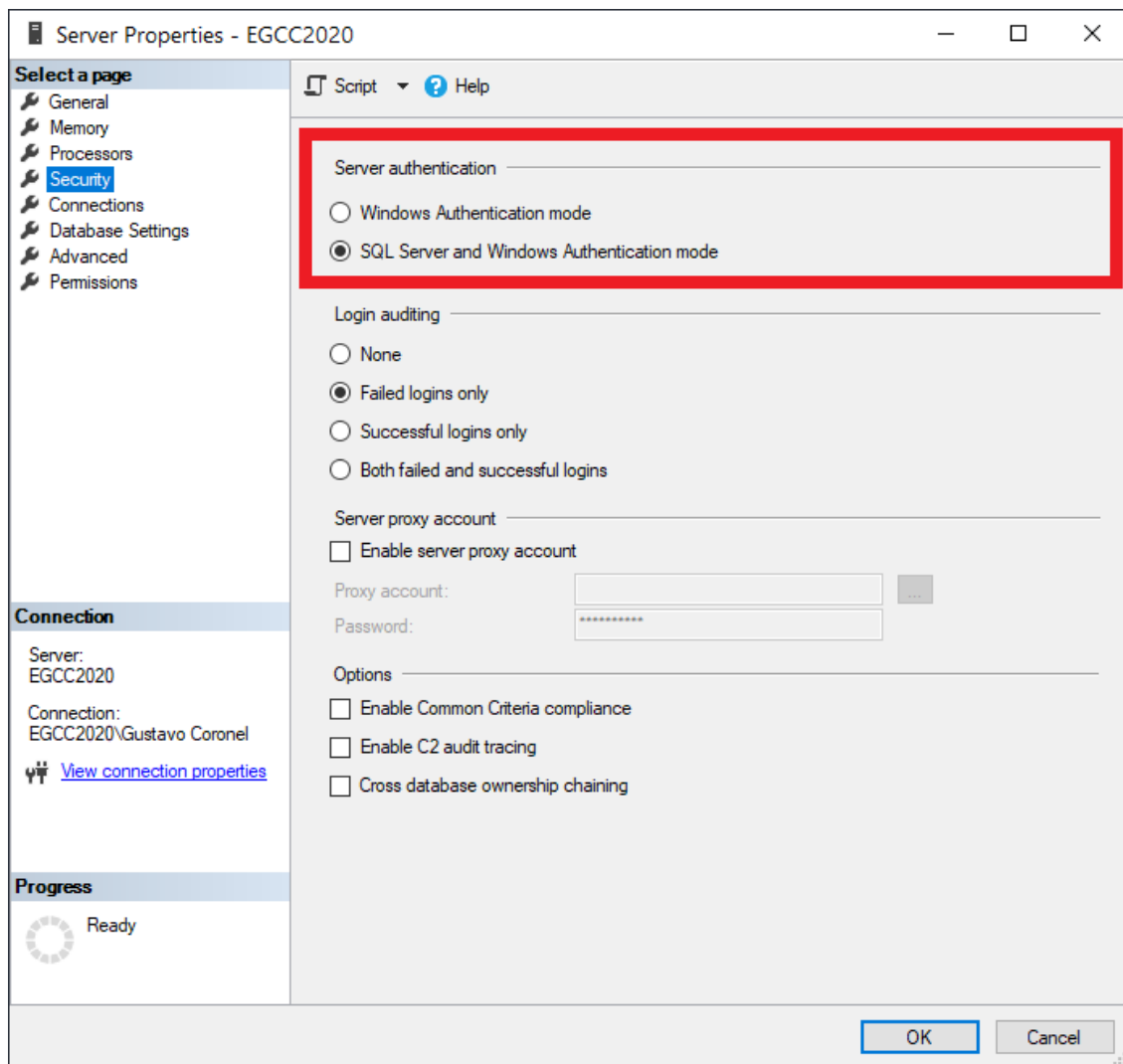
Al instalarse SQL Server se crean 2 Login ID (Inicios de sesión):

- La cuenta de servicio que se utiliza para iniciar el servicio SQL Server. Puede cambiarse sus privilegios.
- El inicio de sesión **sa**. Este no puede eliminarse ni modificarse. No estará disponible si solo está configurada la autenticación de Windows.

Pueden realizar cualquier tarea en SQL Server, esto es posible por que pertenecen al rol de servidor sysadmin.



CONFIGURAR AUTENTICACIÓN DEL SERVIDOR



Windows Authentication mode

El servidor SQL Server solo permite inicios de sesión de usuarios de confianza, esto quiere decir, de usuarios de Windows previamente autorizados.

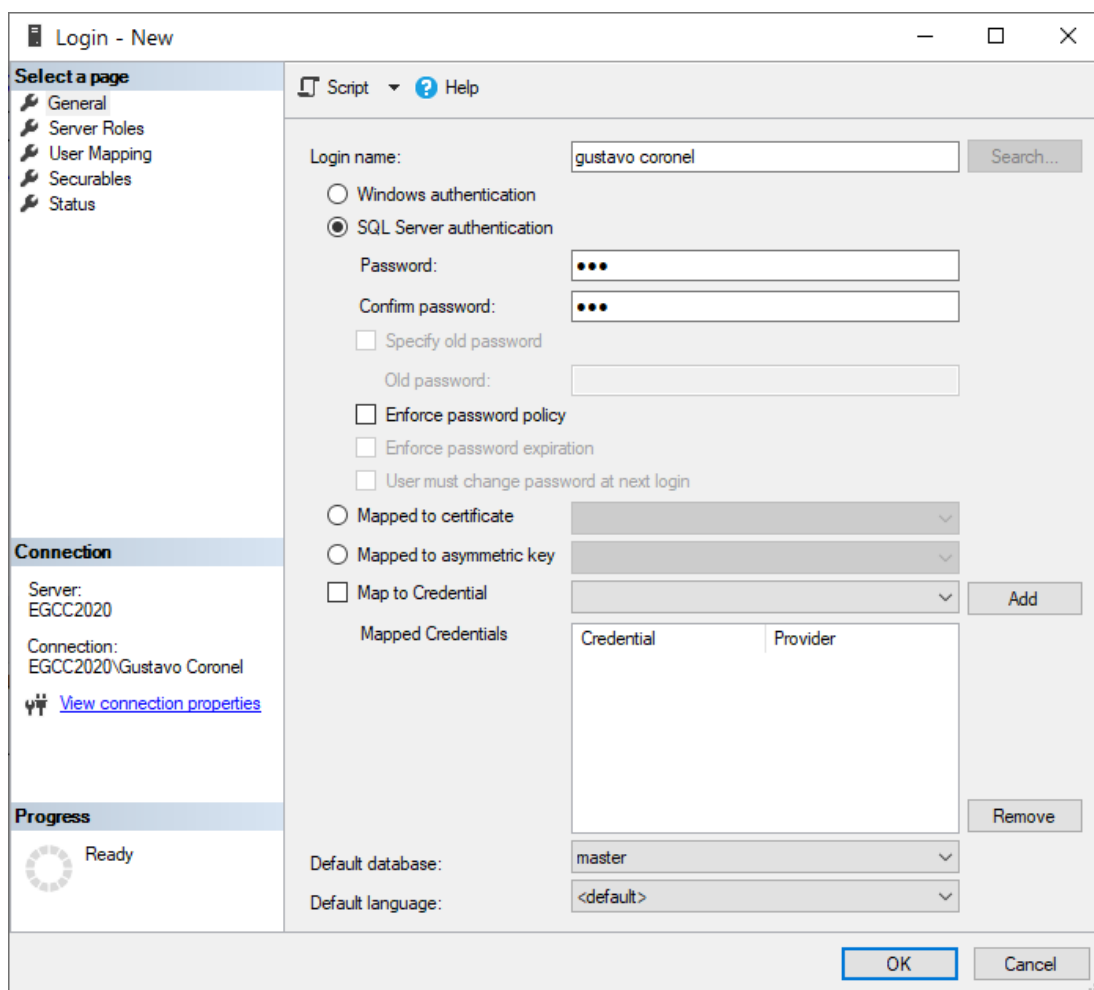
SQL Server and Windows Authentication mode

El servidor SQL Server permite inicios de sesión de usuarios de confianza y del mismo servidor.

INICIOS DE SESIÓN

Crear Inicios de Sesión

Puedes crear nuevos inicios de sesión de manera grafica utilizando SSMS, a continuación, tienes la imagen de la UI que te permite hacer esta tarea:



Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Connection

Server: EGCC2020

Connection: EGCC2020\Gustavo Coronel

[View connection properties](#)

Progress

Ready

Script Help

Login name: gustavo coronel Search...

☐ Windows authentication

☒ SQL Server authentication

Password: ...

Confirm password: ...

☐ Specify old password

Old password:

☐ Enforce password policy

☐ Enforce password expiration

☐ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Mapped Credentials

Credential	Provider
------------	----------

Add

Remove

Default database: master

Default language: <default>

OK Cancel



Utilizando la Instrucción CREATE LOGIN

También puedes utilizar el comando CREATE LOGIN para crear nuevos inicios de sesión, a continuación, tienes un ejemplo donde se está creando login [gustavo coronel] y se le esta asignando el rol de servidor [sysadmin]:

```
-- Creando el login [gustavo coronel]

USE [master]
GO

CREATE LOGIN [gustavo coronel]
WITH
    PASSWORD='sql',
    DEFAULT_DATABASE=[master],
    CHECK_EXPIRATION=OFF,
    CHECK_POLICY=OFF
GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [gustavo coronel]
GO
```

Puedes usar las instrucciones DROP LOGIN / ALTER LOGIN para eliminar y/o modificar información de un login.



ROLES DEL SERVIDOR

ROL	DESCRIPCIÓN
SYSADMIN	Pueden realizar cualquier actividad en el servidor.
SERVERADMIN	Pueden cambiar las opciones de configuración de todo el servidor y apagar el servidor.
SECURITYADMIN	Gestionan los inicios de sesión y sus propiedades. Pueden conceder, negar y rechazar los permisos a nivel de servidor. También pueden conceder permisos de GRANT, DENY y REVOKE a nivel de base de datos si tienen acceso a una base de datos. Además, pueden restablecer contraseñas para los inicios de sesión de SQL Server.
PROCESSADMIN	Pueden finalizar los procesos que se ejecutan en una instancia de SQL Server.
SETUPADMIN	Puede agregar y quitar servidores vinculados mediante instrucciones SQL.
BULKADMIN	Pueden ejecutar la instrucción BULK INSERT.
DISKADMIN	Se utiliza para administrar archivos de disco.
DBCREATOR	Pueden crear, alterar, eliminar y restaurar cualquier base de datos.
PUBLIC	Cada inicio de sesión de SQL Server pertenece al rol de servidor public. Cuando un servidor principal no ha concedido o denegado permisos específicos en un objeto, el usuario hereda los permisos concedidos al rol public en ese objeto. Sólo se deben asignar permisos públicos a cualquier objeto cuando se desee que el objeto esté disponible para todos los usuarios. No se puede cambiar la membresía de public.

Con Microsoft SQL Server es posible administrar roles de servidor definidos por el usuario.

Uno de los grandes beneficios de SQL Server, es la posibilidad de aplicar permisos de una forma granular sin necesidad de asignar un usuario a un rol de servidor que posiblemente le otorgue más permisos de los que realmente necesita.

Por motivos de compatibilidad con versiones anteriores y por comodidad, se siguen manteniendo los roles a nivel de servidor, los cuales agrupan a otras entidades de seguridad.

Se recomienda, sin embargo, asignar permisos más concretos siempre que sea posible.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Consultando los roles de servidor:

```
EXEC sp_helpsrvrole;  
GO
```

ServerRole	Description
sysadmin	System Administrators
securityadmin	Security Administrators
serveradmin	Server Administrators
setupadmin	Setup Administrators
processadmin	Process Administrators
diskadmin	Disk Administrators
dbcreator	Database Creators
bulkadmin	Bulk Insert Administrators

Consultando los roles de servidor y roles definidos por el usuario:

```
SELECT name FROM sys.server_principals  
WHERE type = 'R';  
GO
```

```
name  
-----  
public  
sysadmin  
securityadmin  
serveradmin  
setupadmin  
processadmin  
diskadmin  
dbcreator  
bulkadmin
```



Consultando los miembros de un rol:

```
EXEC sp_helpsrvrolemember 'sysadmin';  
GO
```

ServerRole	MemberName
sysadmin	sa
sysadmin	DESKTOP-835DM8C\Gustavo Coronel
sysadmin	NT SERVICE\SQLwriter
sysadmin	NT SERVICE\winmgmt
sysadmin	NT Service\MSSQLSERVER
sysadmin	NT SERVICE\SQLSERVERAGENT

La tabla siguiente explica los comandos que puedes utilizar para trabajar con roles de nivel de servidor:

COMANDO/FUNCION	DESCRIPCIÓN
CREATE SERVER ROLE	Comando que permite crear un rol de servidor definido por el usuario.
ALTER SERVER ROLE	Comando que permite modifica la pertenencia a un rol de servidor o modifica el nombre de un rol de servidor definido por el usuario.
DROP SERVER ROLE	Comando que permite eliminar un rol de servidor definido por el usuario.
IS_SRVROLEMEMBER	Función que permite determina la pertenencia a un rol de servidor.



ROLES DE BASE DE DATOS

Roles Fijos de Base de Datos

ROL	DESCRIPCIÓN
DB_OWNER	Los miembros de este rol pueden realizar todas las actividades de configuración y mantenimiento en la base de datos, incluyendo eliminar la base de datos.
DB_SECURITYADMIN	Los miembros de este rol pueden modificar la pertenencia a roles y administrar permisos.
DB_ACCESSADMIN	Los miembros de este rol pueden agregar o quitar el acceso a la base de datos para los inicios de sesión.
DB_BACKUPOPERATOR	Los miembros de este rol pueden crear copias de seguridad de la base de datos.
DB_DDLADMIN	Los miembros de este rol pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en la base de datos.
DB_DATAWRITER	Los miembros de este rol pueden agregar, eliminar o cambiar datos en todas las tablas de usuario.
DB_DATAREADER	Los miembros de este rol pueden leer todos los datos de las tablas de usuario.
DB_DENYDATAWRITER	Los miembros de este rol no pueden agregar, modificar, ni eliminar datos de tablas de usuario de la base de datos.
DB_DENYDATAREADER	Los miembros de este rol no pueden leer datos de las tablas de usuario de una base de datos.
PUBLIC	Todos los usuarios de la base de datos pertenecen al rol public. Cuando a un usuario no se le ha concedido ni denegado permisos específicos para un objeto, el usuario hereda los permisos concedidos al rol public para ese objeto.

Los roles de bases de datos son una muy buena alternativa para gestionar los permisos sobre objetos, también puedes crear roles de usuario para mejorar la gestión de permisos sobre grupos de usuarios.

Estos roles existen en todas las bases de datos. No se pueden cambiar los permisos asignados a los roles fijos de bases de datos.

A los usuarios de base de datos le puedes asignar uno o mas roles, por defecto, pertenecen al rol PUBLIC.



Gestionando Roles de Base de Datos

CARACTERÍSTICA	TIPO	DESCRIPCIÓN
SP_HELPDBFIXEDROLE	Metadatos	Devuelve una lista de los roles fijos de base de datos.
SP_DBFIXEDROLEPERMISSION	Metadatos	Muestra los permisos de un rol fijo de base de datos.
SP_HELPROLE	Metadatos	Proporciona información sobre los roles de la base de datos.
SP_HELPROLEMEMBER	Metadatos	Muestra los miembros de un rol de la base de datos.
SYS.DATABASE_ROLE_MEMBERS	Metadatos	Devuelve una fila para cada miembro de cada rol de base de datos.
IS_MEMBER	Metadatos	Indica si el usuario actual es miembro del grupo de Windows especificado o del rol base de datos.
CREATE ROLE	Comando	Crea un nuevo rol de base de datos.
ALTER ROLE	Comando	Cambia el nombre o la pertenencia de un rol de base de datos.
DROP ROLE	Comando	Elimina un rol de base de datos.
SP_ADDROLE	Comando	Crea un nuevo rol de base de datos.
SP_DROPROLE	Comando	Elimina un rol de base de datos.
SP_ADDROLEMEMBER	Comando	Agrega un rol de usuario de base de datos, un inicio de sesión de Windows o un grupo de Windows a un rol de base de datos.
SP_DROPROLEMEMBER	Comando	Elimina una cuenta de seguridad de un rol en la base de datos.
GRANT	Permisos	Proporciona permisos a un rol.
DENY	Permisos	Deniega permisos a un rol.
REVOKE	Permisos	Elimina los permisos otorgados o denegados anteriormente.

La tabla anterior muestra una lista de procedimientos e instrucciones que puedes utilizar para gestionar roles de base de datos.



Consultando roles fijos de la base de datos:

```
EXEC sp_helpdbfixedrole;  
GO
```

DbFixedRole	Description
db_owner	DB Owners
db_accessadmin	DB Access Administrators
db_securityadmin	DB Security Administrators
db_ddladmin	DB DDL Administrators
db_backupoperator	DB Backup Operator
db_datareader	DB Data Reader
db_datawriter	DB Data Writer
db_denydatareader	DB Deny Data Reader
db_denydatawriter	DB Deny Data Writer

Consultando los permisos de un rol fijo de base de datos:

```
EXEC sp_dbfixedrolepermission 'db_datawriter';  
GO
```

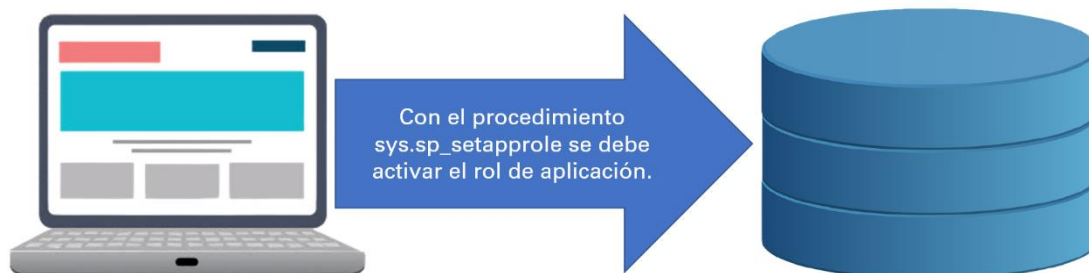
DbFixedRole	Permission
db_datawriter	DELETE permission on any object
db_datawriter	INSERT permission on any object
db_datawriter	UPDATE permission on any object

Creación de un rol de base de datos:

```
USE [EUREKABANK]  
GO  
  
CREATE ROLE [Operador] AUTHORIZATION [dbo]  
GO
```



ROLES DE APLICACIÓN



Un rol de aplicación es como cualquier otro rol en el sentido de que se pueden otorgar permisos sobre elementos asegurables a un rol de aplicación, sin embargo, la diferencia es que debe ser "activado" por una aplicación utilizando el procedimiento almacenado `sp_setapprole`.

Este tipo de rol puede resultar muy beneficioso desde el punto de vista de seguridad, ya que es la aplicación quien activará el rol y recién a partir de ese momento tendrá acceso a los recursos, los inicios de sesión no tendrán permisos respectivos, por ejemplo, si utiliza SSMS para acceder a la base de datos.

Los roles de aplicación no eliminan la necesidad de proporcionar credenciales de conexión iniciales. Primero se debe autenticar con éxito en el server/database utilizando un inicio de sesión antes de que se pueda activar un rol de aplicación.

Los roles de aplicación difieren de los roles de bases de datos en:

- No tiene miembros.
- Se habilitan usando `sp_setapprole` con la contraseña del rol de aplicación después de una autenticación exitosa.

A continuación, tienes un ejemplo de la creación de un rol de aplicación:

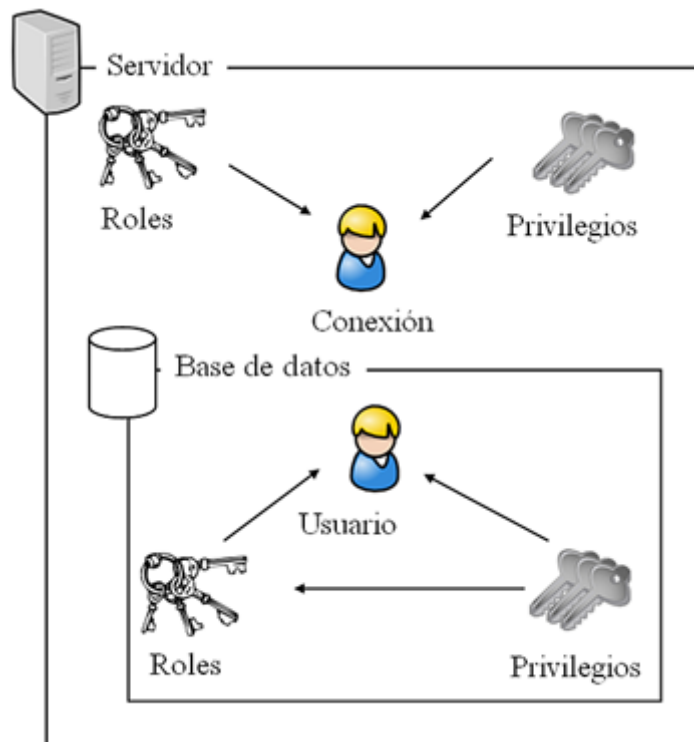
```
USE EUREKABANK;  
GO
```

```
CREATE APPLICATION ROLE EurekaAppRole  
WITH PASSWORD = '123456';  
GO
```



USUARIOS DE BASE DE DATOS

Contexto



El inicio de sesión no es suficiente para gestionar los objetos de una base de datos, es necesario que se encuentre vinculado con un usuario de base de datos, y este debe tener los permisos correspondientes o tener el rol que los tenga.

Todos los usuarios de base de datos, incluido guest (el invitado), pertenecen al grupo public. Los derechos que se detallan a continuación se pueden asignar directamente a public.

Es posible gestionar la asignación de privilegios a nivel del servidor, de la base de datos, del esquema o directamente a nivel del objeto.



Creación de un Usuario desde SSMS

La siguiente UI te permite crear un usuario de base de datos y vincularlo con un inicio de sesión:

Database User - New

Select a page

- General
- Owned Schemas
- Membership
- Securables
- Extended Properties

Connection

Server: EGCC2020

Connection: EGCC2020\Gustavo Coronel

[View connection properties](#)

Progress

Ready

User type: SQL user with login

User name: Gustavo

Login name: gustavo coronel

Default schema: dbo

OK Cancel



Utilizando la Instrucción CREATE USER

El siguiente script crea el usuario [Gustavo], lo vincula con el inicio de sesión [gustavo coronel] y establece el esquema [dbo] como el esquema por defecto:

```
USE [EUREKABANK]
GO

CREATE USER [Gustavo]
FOR LOGIN [gustavo coronel]
WITH DEFAULT_SCHEMA=[dbo]
GO
```



PERMISOS

Tipos de Permisos

PERMISO	DESCRIPCIÓN
CONTROL	Concede al <i>principal</i> todos los permisos definidos en el <i>securable</i> , incluyendo todos los <i>securables</i> que abarca dicho ámbito. Se incluyen permisos para conceder estos permisos a otros <i>principales</i> . De este modo, por ejemplo, CONTROL SERVER sería equivalente a pertenecer al rol de servidor <i>sysadmin</i> .
ALTER	Confiere los permisos necesarios para cambiar las propiedades de un <i>securable</i> y todos los del ámbito, salvo la propiedad de los mismos. Así, por ejemplo, el permiso ALTER sobre una base de datos incluye los permisos necesarios para crear, modificar y eliminar objetos del esquema.
ALTER ANY <securable>	Concede permisos para cambiar cualquier <i>securable</i> del tipo especificado. Por ejemplo, ALTER ANY LOGIN permite al <i>principal</i> modificar cualquier login del servidor.
TAKE OWNERSHIP	Permite al receptor del permiso tomar propiedad del elemento protegible para el que se concede este permiso.
IMPERSONATE <User>	Permite al <i>principal</i> receptor del permiso suplantar al usuario.
CREATE <securable>	Concede el permiso para crear el <i>securable</i> . Dependiendo de si el <i>securable</i> es a nivel de servidor, base de datos o esquema, será necesario contar con el permiso ALTER necesario.
VIEW DEFINITION	Concede el permiso para acceder a los metadatos.
REFERENCES	Concede permisos para crear un FOREIGN KEY en una tabla especificada, y la posibilidad de crear funciones y vistas con la cláusula WITH SCHEMABINDING que hace referencia a ese <i>securable</i> .



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

No todos los permisos pueden asignarse a todas las entidades protegibles. La siguiente tabla muestra una relación de los permisos y los objetos a los que se puede aplicar.

PERMISO	SECURABLES A LOS QUE SE APLICA
SELECT	Sinónimos, Tablas y columnas, Funciones con valores de tabla, Transact-SQL y Common Language Runtime (CLR), Vistas y columnas
VIEW CHANGE TRACKING	Tablas, Esquemas
UPDATE	Sinónimos, Tablas y columnas, Vistas y columnas, Objetos de secuencia.
REFERENCES	Funciones escalares y de agregado (Transact-SQL y CLR), Colas de Service Broker, Tablas y columnas, Funciones con valores de tabla (Transact-SQL y CLR), y columnas, Tipos, Vistas y columnas, Objetos de secuencia
INSERT	Sinónimos, Tablas y columnas, Vistas y columnas.
DELETE	Sinónimos, Tablas y columnas, Vistas y columnas.
EXECUTE	Procedimientos (Transact-SQL y CLR), Funciones escalares y de agregado (Transact-SQL y CLR), Sinónimos, Tipos CLR.
RECEIVE	Colas de Service Broker.
VIEW DEFINITION	Grupos de disponibilidad, Procedimientos (Transact-SQL y CLR), Colas de Service Broker, Funciones escalares y de agregado (Transact-SQL y CLR), Sinónimos, Inicios de sesión, usuarios y roles, Sinónimos, Tablas, Funciones con valores de tabla (Transact-SQL y CLR), Vistas, Objetos de secuencia.
ALTER	Procedimientos (Transact-SQL y CLR), Funciones escalares y de agregado (Transact-SQL y CLR), Colas de Service Broker, Tablas, Funciones con valores de tabla (Transact-SQL y CLR), Vistas.
TAKE OWNERSHIP	Grupos de disponibilidad, Roles, Procedimientos (Transact-SQL y CLR), Funciones escalares y de agregado (Transact-SQL y CLR), Roles del servidor, Sinónimos, Tablas, Funciones con valores de tabla (Transact-SQL y CLR), Vistas, Objetos de secuencia.
CONTROL	Grupos de disponibilidad, Procedimientos (Transact-SQL y CLR), Funciones escalares y de agregado (Transact-SQL y CLR), Inicios de sesión, usuarios y roles, Colas de Service Broker, Sinónimos, Tablas, Funciones con valores de tabla (Transact-SQL y CLR), Vistas, Objetos de secuencia



Asignación de Permisos o Privilegios

El siguiente script otorga el permiso **SELECT** sobre las tablas **Cuenta** y **Movimiento** al usuario **Gustavo**:

```
use [EUREKABANK]
GO

GRANT SELECT ON [dbo].[Movimiento] TO [Gustavo]
GO

GRANT SELECT ON [dbo].[Cuenta] TO [Gustavo]
GO
```

El siguiente script otorga el permiso **SELECT** sobre las tablas **Cliente** y **Empleado** al rol **Operador**:

```
use [EUREKABANK]
GO

GRANT SELECT ON [dbo].[Cliente] TO [Operador]
GO

GRANT SELECT ON [dbo].[Empleado] TO [Operador]
GO
```

El siguiente script asigna el rol **Operador** al usuario **Gustavo**:

```
USE [EUREKABANK]
GO

ALTER ROLE [Operador] ADD MEMBER [Gustavo]
GO
```

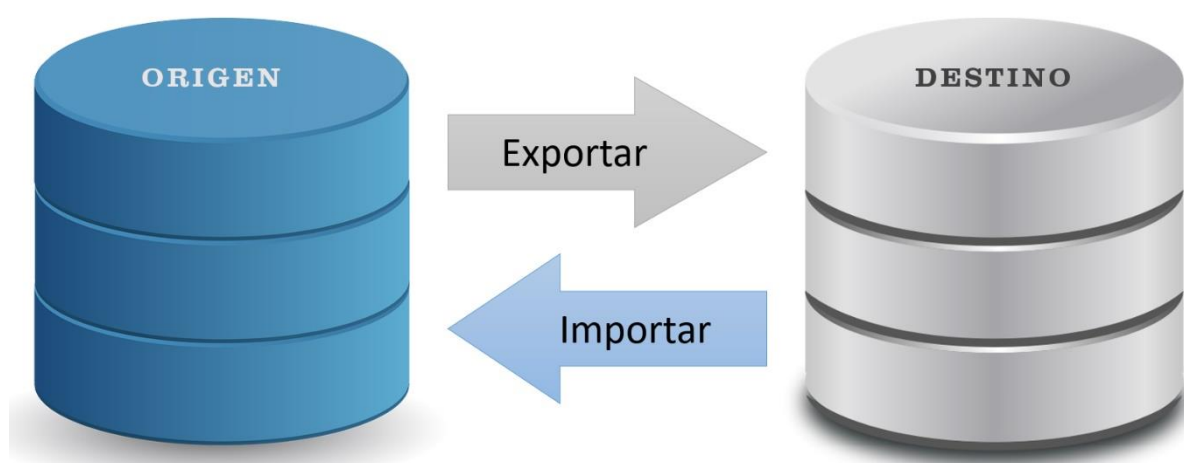


Capítulo 5

IMPORTACIÓN Y EXPORTACIÓN DE DATOS

INTRODUCCIÓN

Contexto



En muchos casos se necesita transferir datos de un servidor de SQL Server a otro servidor (que no necesariamente es SQL Server) o viceversa, esta operación puede resultar muy difícil si no se conoce las opciones que nos brinda SQL Server.



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Microsoft SQL Server admite la exportación masiva de datos desde una tabla o vista de SQL Server y la importación masiva de datos a una tabla. Los métodos básicos disponibles son:

Método	Descripción	Importa datos	Exporta datos
bcp, utilidad	Utilidad de línea de comandos (Bcp.exe) que importa y exporta datos masivamente y genera archivos de formato.	Sí	Sí
Instrucción BULK INSERT	Instrucción Transact-SQL que importa datos directamente de un archivo de datos en una tabla o vista de una base de datos.	Sí	No
INSERT ... SELECT * FROM OPENROWSET(BULK...)	Instrucción Transact-SQL que usa el proveedor de conjunto de filas bulk OPENROWSET para importar masivamente datos en una tabla de SQL Server especificando la función OPENROWSET(BULK...) para seleccionar datos en una instrucción INSERT.	Sí	No



In-Process vs. Out-of-Process Operation

La instrucción BULK INSERT y la función OPENROWSET(BULK) se ejecutan IN-PROCESS con SQL Server, comparten el mismo espacio de direcciones de memoria. Debido a que los archivos de datos son abiertos por un proceso de SQL Server, los datos no se copian entre el proceso del cliente y los procesos de SQL Server.

Por el contrario, la utilidad BCP se ejecuta fuera de proceso. Para mover datos entre los espacios de memoria de proceso, bcp debe utilizar el cálculo de referencias de datos entre procesos. El cálculo de referencias de datos entre procesos consiste en convertir los parámetros de una llamada de método en una secuencia de bytes. Ello puede incrementar significativamente la carga del procesador. Sin embargo, como bcp analiza los datos y los convierte en el formato de almacenamiento nativo en el proceso del cliente, pueden descargarse el análisis y la conversión de datos del proceso de SQL Server. En consecuencia, si tiene restricciones de CPU, puede obtener mejor rendimiento en la importación masiva en un equipo que tenga más de una CPU o en diferentes equipos, utilizando bcp en lugar de utilizar BULK INSERT o `INSERT ... SELECT * FROM OPENROWSET(BULK)`.



Archivos de formato

Las utilidades con las que contamos en SQL Server **bcp**, **BULK INSERT** e **INSERT ... SELECT * FROM OPENROWSET(BULK...)** admiten el uso de un archivo de formato especializado que almacena información de formato para cada campo de un archivo de datos. El archivo de formato puede contener también información acerca de la tabla de SQL Server correspondiente. El archivo de formato se puede utilizar para proporcionar toda la información de formato necesaria para la exportación e importación masivas de datos en una instancia de SQL Server.

Los archivos de formato proporcionan una forma flexible de interpretar los datos con el formato que tienen en el archivo de datos durante la importación, así como para dar formato a los datos del archivo de datos durante la exportación. Esta flexibilidad elimina la necesidad de escribir código para propósitos especiales con el fin de interpretar los datos o volver a darles formato según los requisitos específicos de SQL Server o la aplicación externa. Por ejemplo, si va a exportar masivamente datos que se van a cargar en una aplicación que requiere valores separados por comas, puede usar un archivo de formato para insertar comas como terminadores de campo en los datos exportados.

SQL Server admite dos tipos de archivos de formato: archivos de formato XML y archivos de formato no XML. Los archivos de formato no XML se admiten en versiones anteriores de SQL Server; los archivos de formato XML son admitidos desde SQL Server 2005.

La única herramienta que puede generar un archivo de formato es la utilidad **bcp**.



Consideraciones de rendimiento

Las consideraciones de rendimiento pueden ser importantes si va a importar grandes cantidades de datos. En determinados casos, el rendimiento puede mejorar si modifica la forma en que una operación de importación o exportación masiva controla uno o varios de los siguientes elementos:

- Comprobación de restricciones **CHECK**
- Forma de registrar las transacciones masivas. Relevante para bases de datos que normalmente utilizan el modelo de recuperación completa.
- Orden de los datos exportados
- Bloqueo de tablas
- Ejecución de desencadenadores



USO DE SSIS PARA IMPORTAR Y EXPORTAR DATOS



Contexto

SQL Server Integration Services (SSIS) forma parte del software de datos de Microsoft SQL Server, utilizado para multitud de tareas de migración de datos. Básicamente es una herramienta ETL que forma parte del Business Intelligence Suite de Microsoft y que se utiliza principalmente para conseguir la integración de datos.

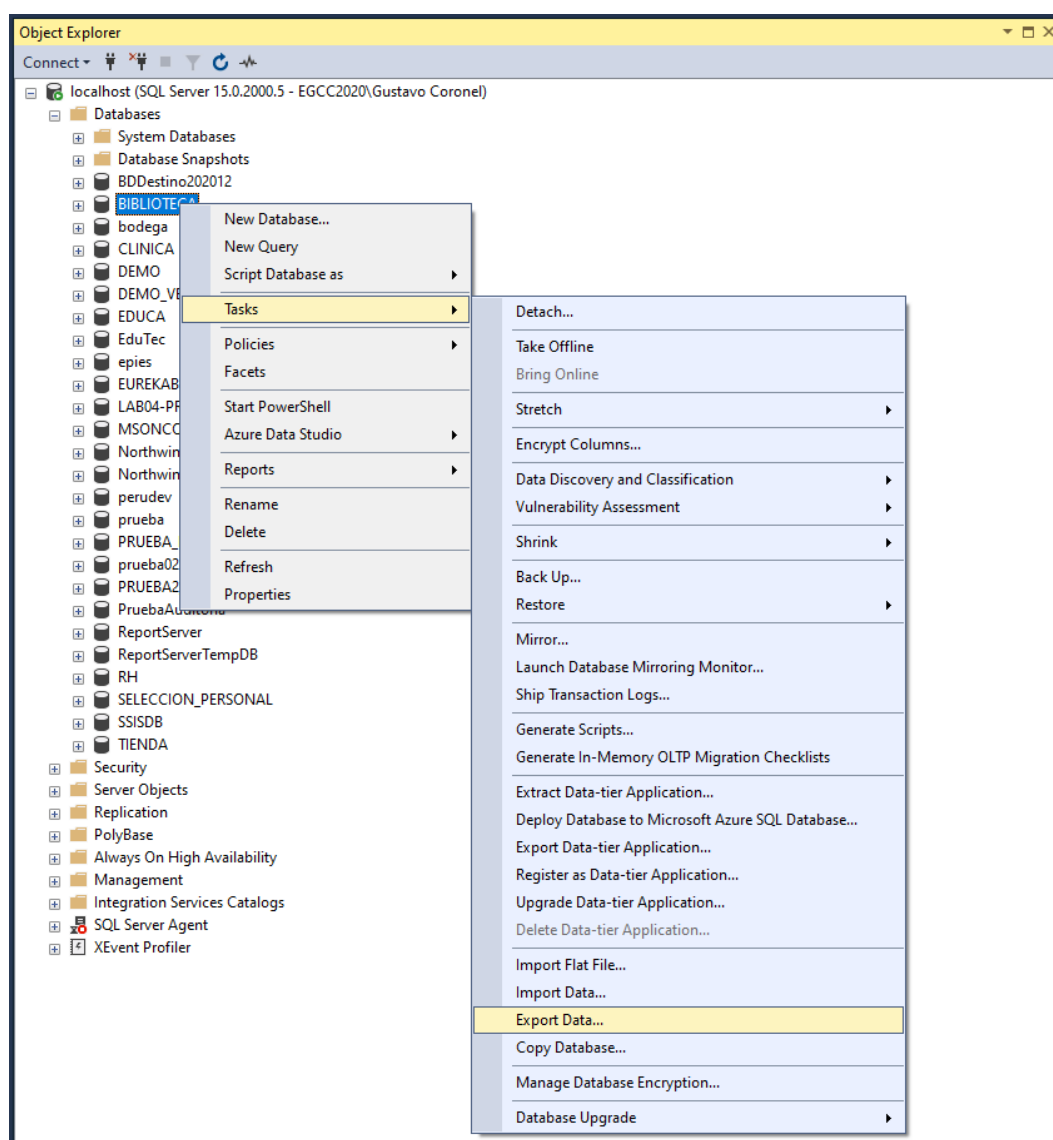
Esta solución está diseñada para resolver problemas relacionadas con la integración de datos y aplicaciones de workflows. Cuenta con una herramienta para soluciones ETL.

Desde el SSMS también asistentes que te permiten exportar e importar datos de una base de datos SQL Server.

Exportar datos

Puedes exportar datos de una base de datos SQL Server a diferentes destinos.

Para iniciar el asistente debes hacer click con el botón derecho del mouse sobre la base de datos de dónde vas a exportar datos, **Tasks** y finalmente **Export Data**, como se ilustra en la siguiente imagen:

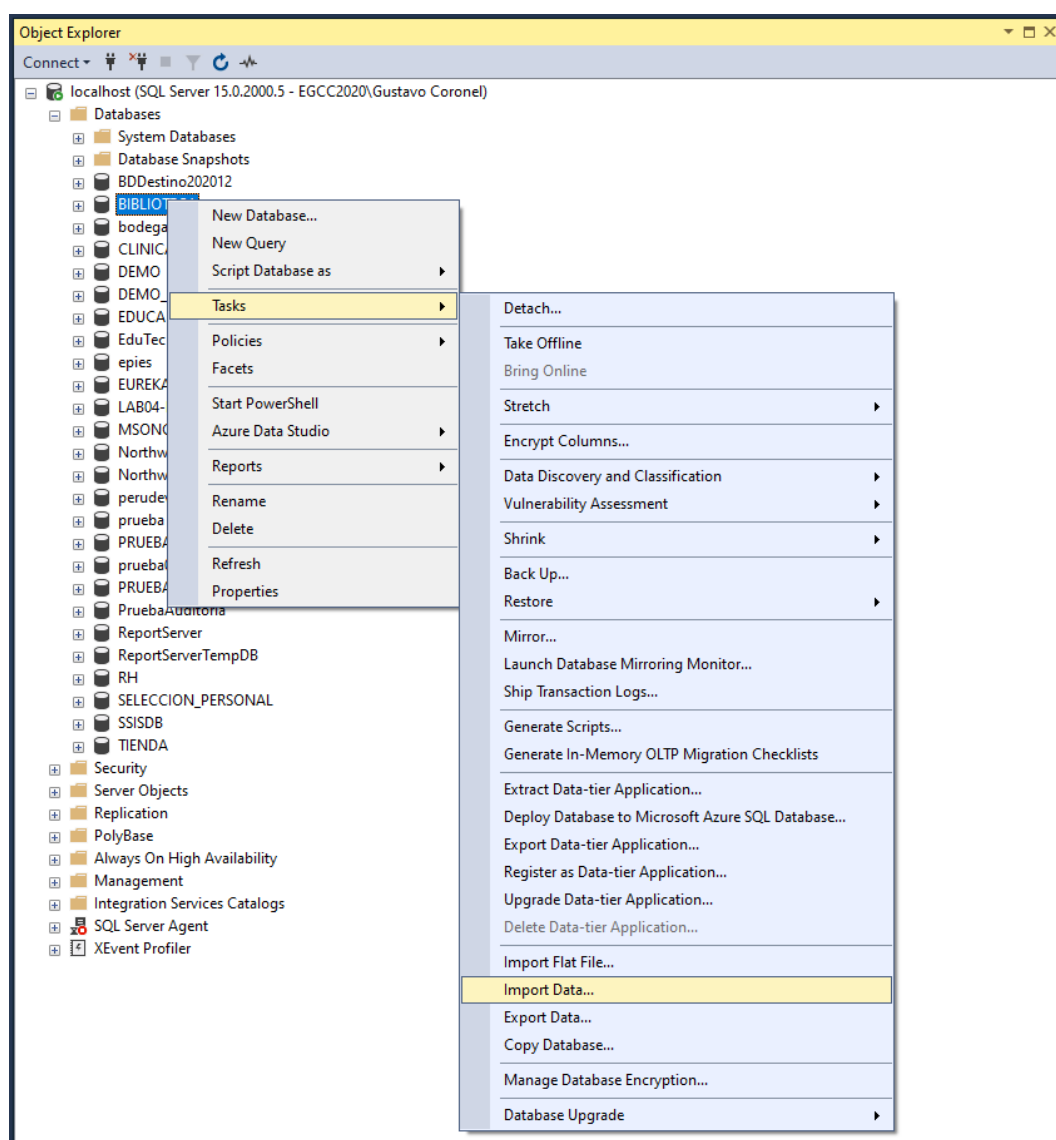


Luego debes seguir los pasos del asistente.

Importar datos

Puedes importar datos a una base de datos SQL Server desde diferentes orígenes.

Para iniciar el asistente debes hacer click con el botón derecho del mouse sobre la base de datos de dónde vas a exportar datos, **Tasks** y finalmente **Import Data**, como se ilustra en la siguiente imagen:



Luego debes seguir los pasos del asistente.



Ejercicios propuestos

Ejercicio 3

Exportar la tabla EDUTEC.DBO.CURSO a la base de datos DEMO tabla DBO.TBL_CURSO.

Ejercicio 4

Realizar los siguientes pasos:

1. Exportar la tabla EDUTEC.DBO.PROFESOR a un archivo de Excel de nombre PROFESORES.XLS.
2. Importar el archivo PROFESOR.XLS a la base de datos DEMO a una tabla de nombre DBO.TBL_PROFESOR.

Ejercicio 5

Realizar los siguientes pasos:

1. Exportar la tabla EDUTEC.DBO.ALUMNO a un archivo de texto ALUMNO.TXT con formato CSV.
2. Importar el archivo ALUMNO.TXT a la base de datos DEMO a una tabla de nombre DBO.TBL_ALUMNO.

Ejercicio 6

Exportar el resumen de ventas de la base de datos NORTHWIND a la base de datos DEMO a una tabla de nombre DBO.TBL_VENTAS.

UTILITARIO BCP PARA IMPORTAR Y EXPORTAR DATOS

Utilitario BCP

La utilidad **bcp** (**bcp.exe**) es una herramienta de línea de comandos que utiliza la API de importación masiva de Bulk Copy Program o Programa de copia masiva (BCP). La utilidad bcp realiza las tareas siguientes:

- Exportaciones masivas de datos de una tabla de SQL Server a un archivo de datos.
- Exportaciones masivas de una consulta.
- Importaciones masivas de datos de un archivo de datos a una tabla de SQL Server.
- Genera archivos de formato.

```
C:\>bcp -help
usage: bcp {dbtable | query} {in | out | queryout | format} datafile
      [-m maxerrors]           [-f formatfile]           [-e errfile]
      [-F firstrow]            [-L lastrow]              [-b batchsize]
      [-n native type]         [-c character type]       [-w wide character
type]
      [-N keep non-text native] [-v file format version] [-q quoted identifier]
      [-C code page specifier]  [-t field terminator]  [-r row terminator]
      [-i inputfile]            [-o outfile]           [-a packetsize]
      [-S server name]          [-U username]           [-P password]
      [-T trusted connection]   [-v version]         [-R regional enable]
      [-k keep null values]     [-E keep identity values]
      [-h "load hints"]         [-x generate xml format file]
      [-d database name]       [-K application intent] [-l login timeout]
```

Copiar filas de una tabla en un archivo de datos utilizando conexión de confianza

El siguiente ejemplo ilustra la opción **out** sobre la tabla **EduTec.dboCurso**. En este ejemplo se crea un archivo de datos con el nombre **Curso_A.dat** y se usa el formato de caracteres para copiar los datos de la tabla en ese archivo. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando **bcp**.

En la consola, escriba el siguiente comando:

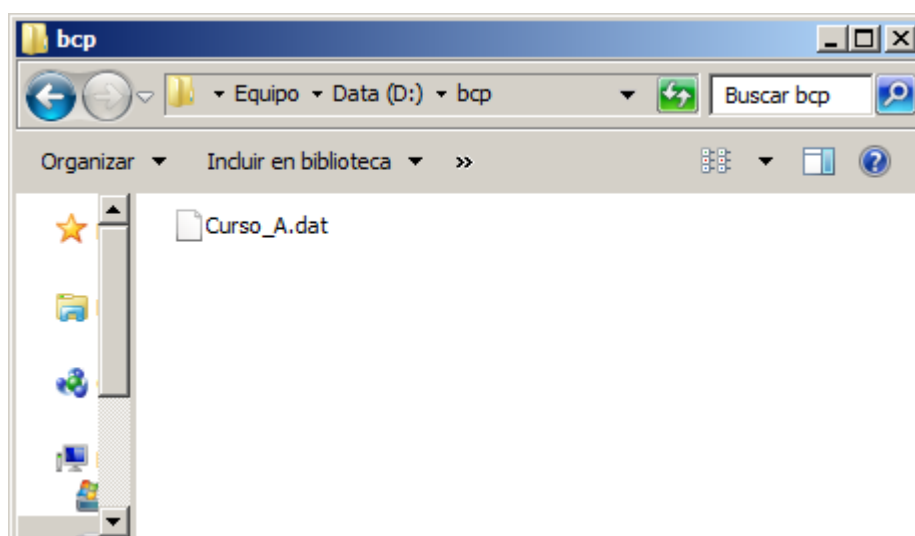
```
C:\>bcp edutec.dbo.Curso out d:\bcp\Curso_A.dat -T -c
```

Starting copy...

14 rows copied.

Network packet size (bytes): 4096

Clock Time (ms.) Total : 1 Average : (14000.00 rows per sec.)





Copiar filas de una tabla en un archivo de datos utilizando autenticación de modo mixta

El siguiente ejemplo ilustra la opción **out** sobre la tabla **EduTec.dboCurso**. En este ejemplo se crea un archivo de datos con el nombre **Curso_B.dat** y se usa el formato de caracteres para copiar los datos de la tabla en ese archivo.

En el ejemplo se supone que usa la autenticación de modo mixto; debe utilizar el modificador **-U** para especificar su identificador de inicio de sesión. Además, a menos que se esté conectando a la instancia predeterminada de SQL Server en el equipo local, use el modificador **-S** para especificar el nombre del servidor y, opcionalmente, un nombre de instancia, el modificador **-P** se utiliza para especificar la contraseña.

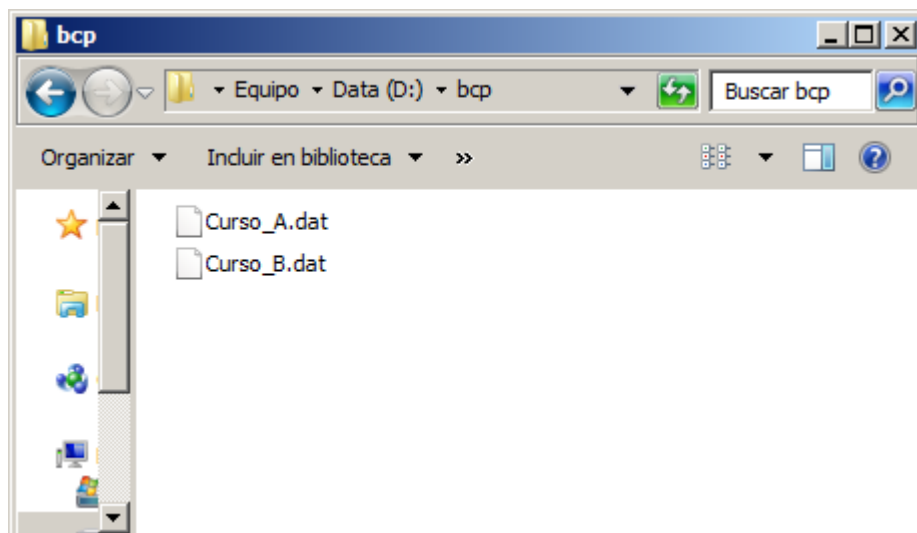
```
C:\>bcp edutech.dbo.Curso out d:\bcp\Curso_B.dat -c -Usa -Psql
```

```
Starting copy...
```

```
14 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 1      Average : (14000.00 rows per sec.)
```





Nota

Si no utiliza el modificar **–P** el sistema le solicitará la contraseña.

Copiar datos de un archivo en una tabla

El siguiente ejemplo ilustra la opción **in** y usa el archivo creado en el ejemplo anterior (Curso_B.dat). No obstante, en primer lugar, este ejemplo crea una copia vacía de la tabla de **EduTec.dbo.Curso**, **dbo.Curso2**, en la que se copian los datos. En el ejemplo se supone que se usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando **bcp**.

Para crear la tabla vacía, en el Editor de consultas, escriba el siguiente comando:

```
USE EduTec;  
GO
```

```
SELECT * INTO dbo.Curso2  
FROM dbo.Curso WHERE 1=2;  
GO
```

```
SELECT * FROM dbo.Curso2;  
GO
```

```
IdCurso IdTarifa NomCurso
```

```
-----
```

```
(0 filas afectadas)
```

Para realizar una copia masiva de los datos de caracteres en la nueva tabla, es decir, para importar los datos, en la consola escriba el siguiente comando:

```
C:\>bcp edutec.dbo.Curso2 in d:\bcp\Curso_B.dat -T -c
```

```
Starting copy...
```

```
14 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 343      Average : (40.82 rows per sec.)
```



ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

Para comprobar que el comando se ha ejecutado correctamente, muestre el contenido en el editor de consultas utilizando el siguiente comando:

```
USE EduTec;  
GO
```

```
SELECT * INTO dboCurso2  
FROM dbo.Curso WHERE 1=2;  
GO
```

```
IdCurso IdTarifa NomCurso
```

```
-----  
C001    A        SO Windows  
C002    B        Visual Basic .NET - Fundamentos  
C003    C        Visual Basic .NET - Cliente/Servidor  
C004    C        Visual Basic .NET - Web  
C005    B        Java - POO  
C006    C        Java - Cliente/Servidor  
C007    C        Java - Web  
C008    C        Java - Frameworks  
C009    B        PHP - Fundamentos  
C010    C        PHP - POO  
C011    C        PHP - Frameworks  
C012    B        C# - POO  
C013    C        C# - Web Services  
C014    C        Java - Web Services
```

```
(14 filas afectadas)
```



Copiar una columna específica en un archivo de datos

Para copiar una columna específica, puede usar la opción **queryout**. El siguiente ejemplo copia únicamente la columna **NomCurso** de la tabla **dbo.Curso** en un archivo de datos. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando **bcp**.

En la consola ejecute el siguiente comando:

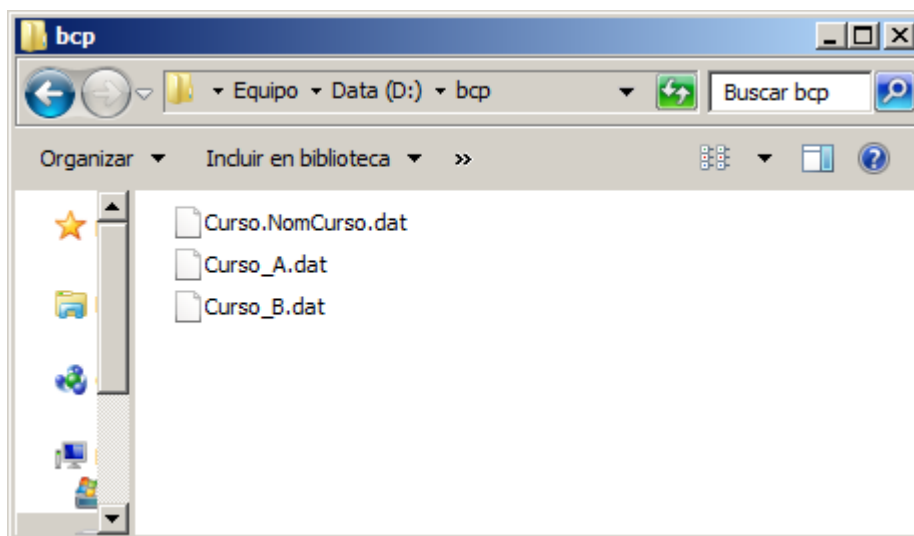
```
C:\>bcp "select NomCurso from EduTec.dbo.Curso" queryout  
d:\bcp\Curso.NomCurso.dat -T -c
```

Starting copy...

14 rows copied.

Network packet size (bytes): 4096

Clock Time (ms.) Total : 1 Average : (14000.00 rows per sec.)



Copiar una fila específica en un archivo de datos

Para copiar una fila específica, se puede usar la opción **queryout**. En el siguiente ejemplo se copia únicamente la fila correspondiente al profesor de código P002 de la tabla **EduTec.dbo.Profesor** en un archivo de datos de nombre P002.dat. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia de servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

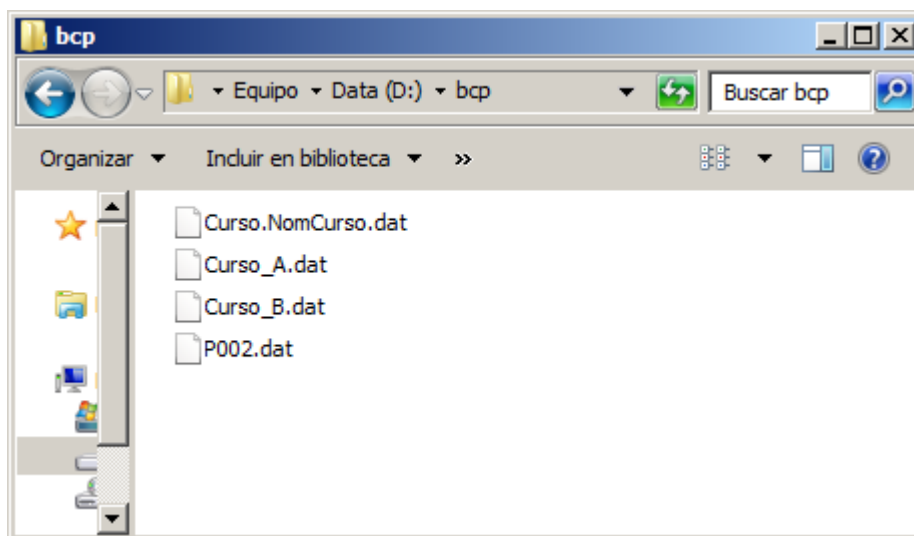
```
C:\>bcp "select * from EduTec.dbo.Profesor where IdProfesor='P002'"  
queryout d:\bcp\P002.dat -T -c
```

Starting copy...

1 rows copied.

Network packet size (bytes): 4096

Clock Time (ms.) Total : 31 Average : (32.26 rows per sec.)



Copiar datos de una consulta en un archivo de datos

Para copiar el conjunto de resultados de una instrucción Transact-SQL en un archivo de datos, use la opción **queryout**. El siguiente ejemplo copia los nombres de la tabla de **EduTec.dbo.Profesor**, ordenados por el apellido y después por el nombre, en el archivo de datos **Profesores.txt**. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

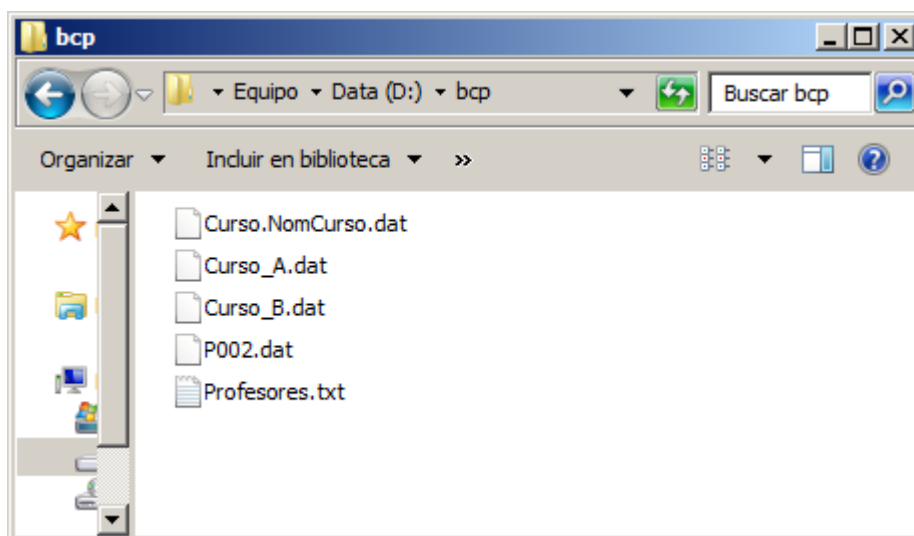
```
C:\>bcp "select ApeProfesor, NomProfesor from EduTec.dbo.Profesor order by 1, 2" queryout d:\bcp\Profesores.txt -T -c
```

Starting copy...

42 rows copied.

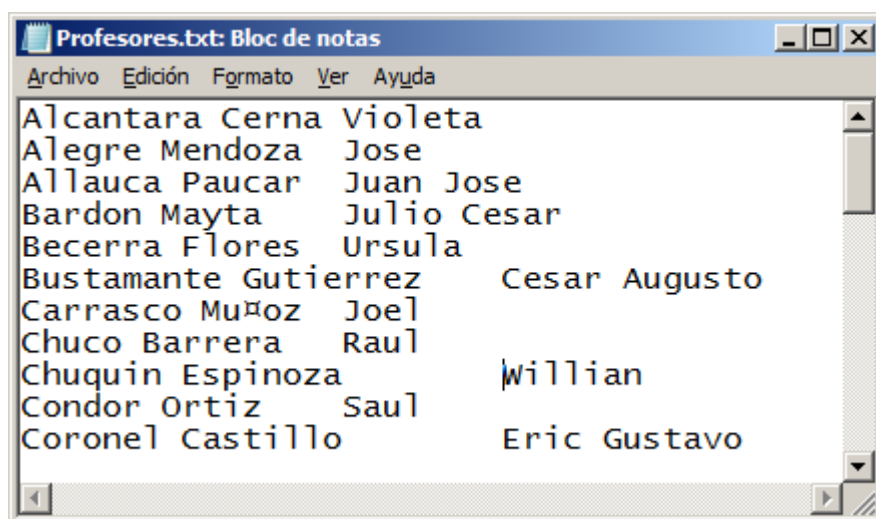
Network packet size (bytes): 4096

Clock Time (ms.) Total : 31 Average : (1354.84 rows per sec.)





ADMINISTRACIÓN DE BASE DE DATOS SQL SERVER

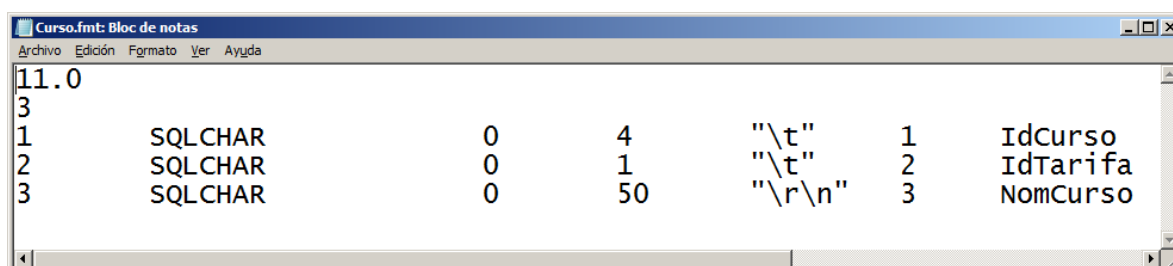
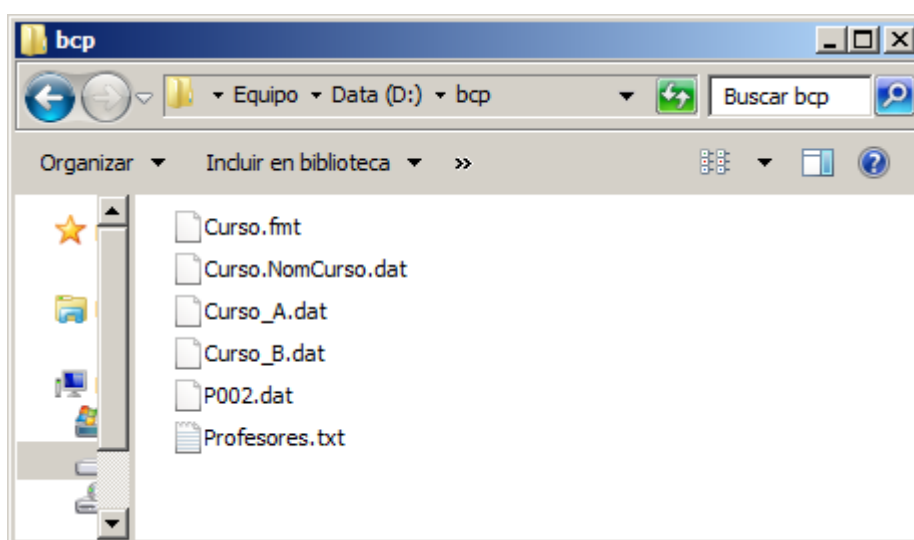


Crear un archivo de formato no XML

En el siguiente ejemplo se crea un archivo de formato no XML, **Curso.fmt**, para la tabla dbo.Curso de la base de datos EduTec. En el ejemplo se está usando autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

```
C:\>bcp EduTec.dbo.Curso format nul -T -c -f d:\bcp\Curso.fmt
```

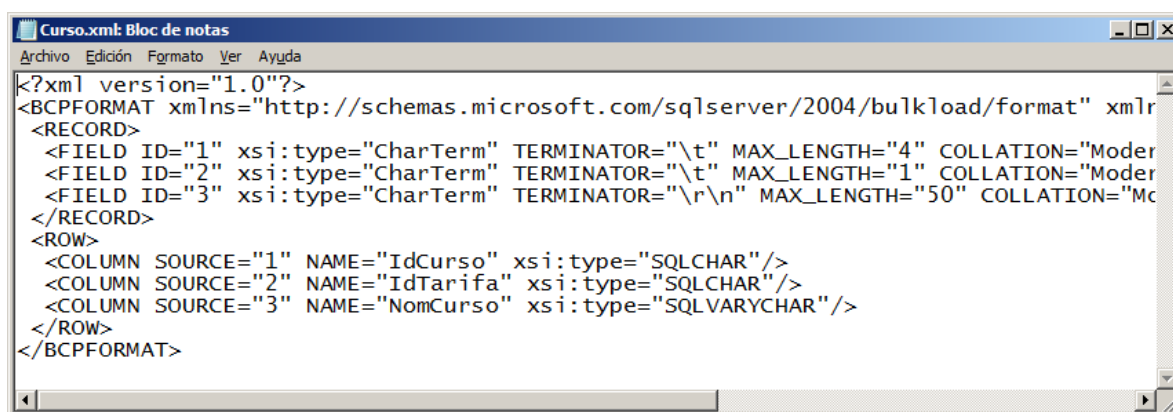
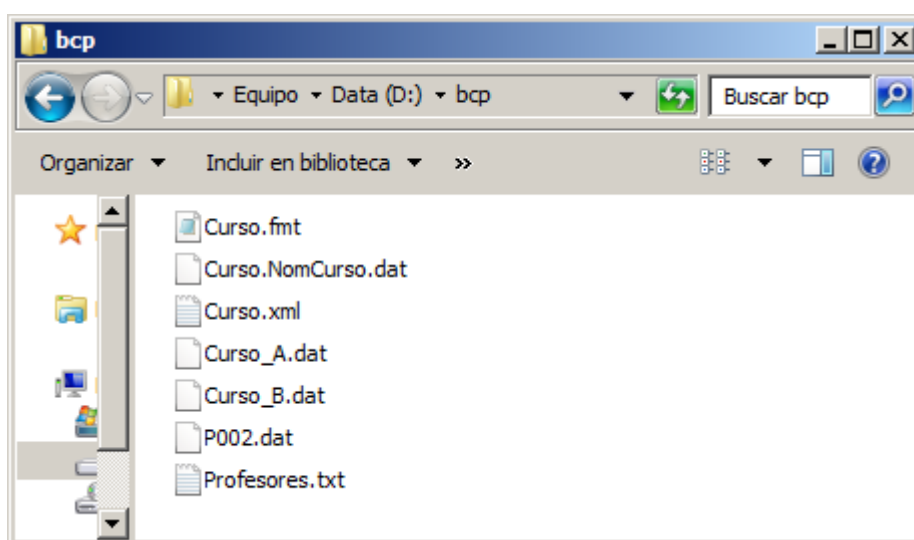


Crear un archivo de formato XML

El siguiente ejemplo crea un archivo de formato XML con el nombre Curso.xml para la tabla **dbo.Curso** de la base de datos EduTec. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

```
c:\>bcp EduTec.dbo.Curso format nul -T -c -x -f d:\bcp\Curso.xml
```





Usar un archivo de formato para importar de forma masiva con bcp

Para usar un archivo de formato creado anteriormente al importar datos a una instancia de SQL Server, se debe usar el modificador **-f** con la opción **in**. Por ejemplo, el siguiente comando copia de forma masiva el contenido del archivo de datos, Curso_A.dat, en una copia de la tabla dbo.Curso de nombre dbo.Curso3 mediante el archivo de formato creado anteriormente (Curso.xml). En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

Primero crearemos la tabla dbo.Curso3:

```
USE EduTec;
go

SELECT * INTO dbo.Curso3
FROM dbo.Curso WHERE 1=2;
GO

select * from dbo.Curso3;
go

IdCurso IdTarifa NomCurso
-----
(0 filas afectadas)
```

En el símbolo del sistema de Windows, escriba:

```
C:\>bcp EduTec.dbo.Curso3 in D:\bcp\Curso_B.dat -T -f D:\bcp\Curso.xml

Starting copy...

14 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total      : 234      Average : (59.83 rows per sec.)
```



Ahora, consultamos la tabla dbo.Curso3:

```
select * from dbo.Curso3;  
go
```

```
IdCurso IdTarifa NomCurso
```

```
-----  
C001    A        SO windows  
C002    B        Visual Basic .NET - Fundamentos  
C003    C        Visual Basic .NET - Cliente/Servidor  
C004    C        Visual Basic .NET - Web  
C005    B        Java - POO  
C006    C        Java - Cliente/Servidor  
C007    C        Java - Web  
C008    C        Java - Frameworks  
C009    B        PHP - Fundamentos  
C010    C        PHP - POO  
C011    C        PHP - Frameworks  
C012    B        C# - POO  
C013    C        C# - Web Services  
C014    C        Java - Web Services
```

(14 filas afectadas)



Exportar a Formato CSV

Exportar la tabla EDUTECH.DBO.CURSOS al archivo CURSOS.CSV.

Desde la consola de Windows:

```
bcp "SELECT * FROM dbo.Curso" queryout D:\BCP\Cursos.CSV -t, -c -S . -d  
EDUTECH -T
```

Configurar el servidor para utilizar Transact SQL:

```
sp_configure 'show advanced options', 1;  
RECONFIGURE;  
go
```

```
sp_configure;  
GO
```

```
sp_configure 'xp_cmdshell', 1;  
RECONFIGURE;  
GO
```

Utilizando Transact-SQL:

```
declare @comando varchar(400)  
set @comando = 'bcp "SELECT top 5 * FROM [EduTec].[dbo].[Profesor]"  
queryout D:\BCP\test.xls -Slocalhost -Usa -Psql -c -C RAW -t "," '  
exec master..xp_cmdshell @comando  
GO
```



IMPORTACIÓN UTILIZANDO BULK INSERT

Contexto

BULK INSERT carga datos de un archivo de datos a una tabla. Esta funcionalidad es parecida a la que ofrece la opción **in** del comando **bcp**, aunque el que lee el archivo de datos es el proceso de SQL Server.

Importar datos desde un archivo de datos utilizando un archivo de formato.

Utilizaremos el archivo de formato Curso.xml para importar el archivo de datos de Curso_A.dat en la tabla dbo.Curso4.

Primero crearemos la tabla dbo.Curso4:

```
USE EduTec;  
go
```

```
SELECT * INTO dbo.Curso4  
FROM dbo.Curso WHERE 1=2;  
GO
```

```
select * from dbo.Curso4;  
go
```

```
IdCurso IdTarifa NomCurso
```

(0 filas afectadas)



Ahora, procederemos a importar el archivo de datos:

```
bulk insert EduTec.dbo.Curso4
from 'D:\bcp\Curso_A.dat'
with (formatfile='D:\bcp\Curso.xml');

select * from dbo.Curso4;
go
```

IdCurso	IdTarifa	NomCurso
---------	----------	----------

C001	A	SO windows
C002	B	Visual Basic .NET - Fundamentos
C003	C	Visual Basic .NET - Cliente/Servidor
C004	C	Visual Basic .NET - Web
C005	B	Java - POO
C006	C	Java - Cliente/Servidor
C007	C	Java - Web
C008	C	Java - Frameworks
C009	B	PHP - Fundamentos
C010	C	PHP - POO
C011	C	PHP - Frameworks
C012	B	C# - POO
C013	C	C# - Web Services
C014	C	Java - Web Services

(14 filas afectadas)



Importar datos desde un archivo de texto

En este caso importaremos datos desde Profesores.txt en la tabla dbo.Profesor2.

Primero crearemos la tabla dbo.Profesor2:

```
select NomProfesor, ApeProfesor  
into dbo.Profesor2  
from dbo.Profesor  
where 1=2;  
(0 filas afectadas)
```

Ahora, procedemos a importar los datos:

```
bulk insert dbo.Profesor2  
from 'D:\bcp\Profesores.txt'  
with (fieldterminator='\t',rowterminator='\n');  
go
```

```
select * from dbo.Profesor2;  
go
```

NomProfesor	ApeProfesor
Alcantara Cerna	Violeta
Alegre Mendoza	Jose
Allauca Paucar	Juan Jose
Bardon Mayta	Julio Cesar
. . .	
. . .	



Importar datos desde un archivo de datos en su formato nativo

Importaremos los datos desde el archivo Curso_nat.dat en la tabla dbo.Curso5.

En primer lugar crearemos la tabla dbo.Curso5:

```
SELECT * INTO dbo.Curso5  
FROM dbo.Curso WHERE 1=2;  
GO
```

```
select * from dbo.Curso5;  
go
```

```
IdCurso IdTarifa NomCurso
```

```
-----
```

```
(0 filas afectadas)
```

Se procederá a crear el archivo con formato nativo utilizando el utilitario bcp:

```
C:\>bcp EduTec.dbo.Curso out D:\bcp\Curso-nat.Dat -n -T
```

```
Starting copy...
```

```
14 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 31      Average : (451.61 rows per sec.)
```

Ahora, procedemos a ejecutar la importación:

```
BULK INSERT dbo.Curso5  
FROM 'D:\bcp\Curso-nat.Dat'  
WITH (DATAFILETYPE='native');  
GO
```

```
(14 filas afectadas)
```



IMPORTACIÓN UTILIZANDO OPENROWSET

Contexto

Se utiliza para importaciones masivas de datos. Su sintaxis básica es:

```
INSERT ... SELECT * FROM OPENROWSET(BULK...)
```

Importar datos desde un archivo de datos utilizando un archivo de formato.

Utilizaremos el archivo de formato Curso.fmt para importar el archivo de datos de Curso_A.dat en la tabla dbo.Curso6.

Primero crearemos la tabla dbo.Curso6:

```
SELECT * INTO dbo.Curso6  
FROM dbo.Curso WHERE 1=2;  
GO
```

(0 filas afectadas)

Ahora, procederemos a ejecutar la importación:

```
INSERT INTO dbo.Curso6  
SELECT a.* FROM OPENROWSET( BULK 'D:\bcp\Curso_A.dat',  
    FORMATFILE = 'D:\bcp\Curso.fmt') AS a;
```

(14 filas afectadas)



CONSULTAS DISTRIBUIDAS

Configuración

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
go

sp_configure;
GO

sp_configure 'Ad Hoc Distributed Queries', 1;
RECONFIGURE;
GO
```

Ejemplos

Usando autenticación integrada:

```
SELECT a.*
FROM OPENROWSET('SQLNCLI', 'Server=localhost;Trusted_Connection=yes;',
    'SELECT * FROM EDUTEC.dbo.curso') AS a;
```

Usando autenticación SQL Server:

```
SELECT a.*
FROM OPENROWSET('SQLNCLI', 'Server=localhost;userid=sa;password=sql',
    'SELECT * FROM EDUTEC.dbo.curso') AS a;
```



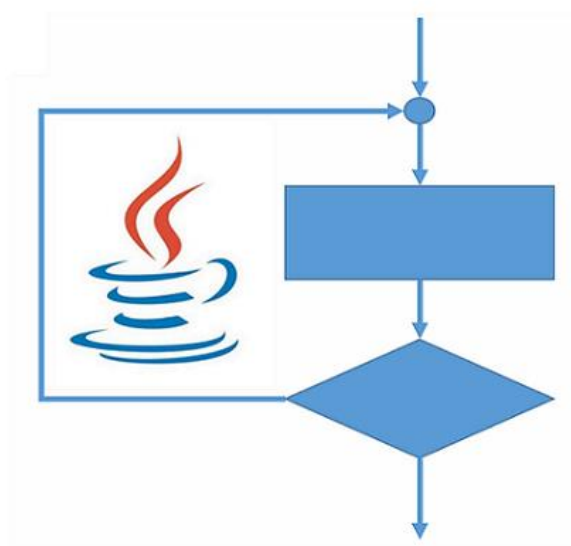
Capítulo 6 CURSOS VIRTUALES

CUPONES

En esta URL se publican cupones de descuento:

<http://gcoronelc.github.io>

FUNDAMENTOS DE PROGRAMACIÓN CON JAVA



Tener bases sólidas de programación muchas veces no es fácil, creo que es principalmente por que en algún momento de tu aprendizaje mezclas la entrada de datos con el proceso de los mismos, o mezclas el proceso con la salida o reporte, esto te lleva a utilizar malas prácticas de programación que luego te serán muy difíciles de superar.

En este curso aprenderás las mejores prácticas de programación para que te inicies con éxito en este competitivo mundo del desarrollo de software.

URL del Curso: <https://n9.cl/gcoronelc-java-fund>

Avance del curso: <https://n9.cl/gcoronelc-fp-avance>

Cupones de descuento: <http://gcoronelc.github.io>



JAVA ORIENTADO A OBJETOS



CURSO PROFESIONAL DE JAVA ORIENTADO A OBJETOS

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

I N S T R U C T O R

En este curso aprenderás a crear software aplicando la Orientación a Objetos, la programación en capas, el uso de patrones de software y Swing.

Cada tema está desarrollado con ejemplos que demuestran los conceptos teóricos y finalizan con un proyecto aplicativo.

URL del Curso: <https://bit.ly/2B3ixUW>

Avance del curso: <https://bit.ly/2RYGXIt>

Cupones de descuento: <http://gcoronelc.github.io>



PROGRAMACIÓN CON JAVA JDBC



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JAVA JDBC

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

I N S T R U C T O R

En este curso aprenderás a programar bases de datos Oracle con JDBC utilizando los objetos Statement, PreparedStatement, CallableStatement y a programar transacciones correctamente teniendo en cuenta su rendimiento y concurrencia.

Al final del curso se integra todo lo desarrollado en una aplicación de escritorio.

URL del Curso: <https://bit.ly/31apy0O>

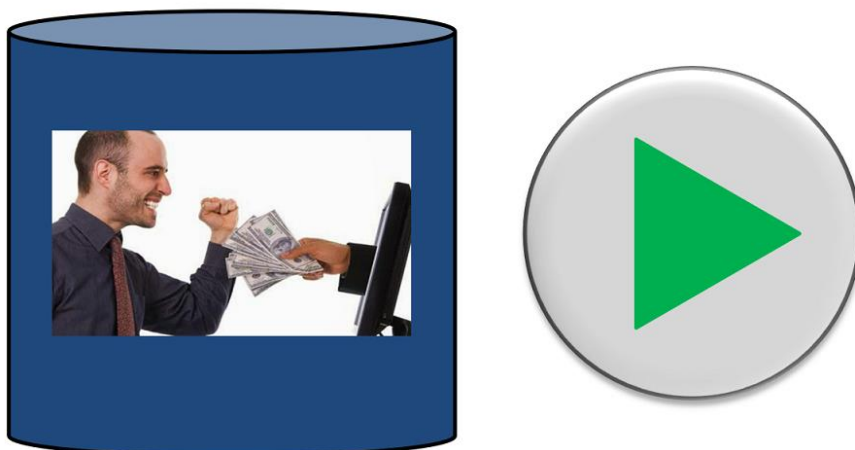
Avance del curso: <https://bit.ly/2vatZOT>

Cupones de descuento: <http://gcoronelc.github.io>



PROGRAMACIÓN CON ORACLE PL/SQL

ORACLE PL/SQL



En este curso aprenderás a programar las bases de datos ORACLE con PL/SQL, de esta manera estarás aprovechando las ventajas que brinda este motor de base de datos y mejorarás el rendimiento de tus consultas, transacciones y la concurrencia.

Los procedimientos almacenados que desarrolles con PL/SQL se pueden ejecutarlos de Java, C#, PHP y otros lenguajes de programación.

URL del Curso: <https://bit.ly/2YZjfxT>

Avance del curso: <https://bit.ly/3bcigYb>

Cupones de descuento: <http://gcoronelc.github.io>