



Bases de datos MS SQL Server 2005

En este capítulo veremos qué es una base de datos, los elementos que la componen, la diferencia entre una base de datos de sistema y una de usuario, los catálogos de bases de datos y de sistema, y cómo consultarlos.

Esta página se ha dejado en blanco intencionalmente.

Capítulo 2

Bases de datos MS SQL Server 2005

Contenido

- ❑ *¿Qué es una base de datos?*
 - ✓ *Objetos de una base de datos SQL Server 2005*
- ❑ *Las bases de datos de sistema*
 - ✓ *Las tablas de sistema*
 - *El Catálogo de la Base de Datos*
 - *El Catálogo del Sistema*
 - ✓ *Consultas a los Catálogos*
 - *Con vista del catálogo*
 - *Con procedimiento almacenado del sistema*
 - *Con vista de esquema de información*
- ❑ *El Code Editor (Editor de Código)*
 - ✓ *Carga de SQL Query Editor*
 - ✓ **Ejercicio 4:** *Consulta al Catálogo de la base de datos AdventureWorks*

Esta página se ha dejado en blanco intencionalmente.

Bases de datos MS SQL Server 2005

SQL (Structured Query Language) es un lenguaje de manejo de datos creado por IBM en los años setenta como una herramienta para facilitar el acceso de los usuarios a los datos almacenados en las computadoras centrales. El lenguaje fue adoptado por otros fabricantes de productos para bases de datos, por lo que fue necesario definir un estándar. Este estándar se conoce como **SQL ANSI**.

En la actualidad existe en el mercado muchos productos basados en el estándar SQL ANSI: PL/SQL de Oracle, SQL Server de Microsoft, System 11 de Sybase, DB2 de IBM, etc. Todos los fabricantes diseñan su producto SQL basándose en el estándar ANSI, pero cada uno le añade "extensiones" propias (mejoras o añadidos al lenguaje) para satisfacer los requerimientos particulares de sus clientes.

El producto de Microsoft, SQL Server, ahora en su versión 2005 emplea un lenguaje basado en el estándar que Microsoft llama **Transact-SQL**.

¿Qué es una base de datos?

SQL Server utiliza un tipo de base de datos denominado base de datos relacional.

Las bases de datos relacionales son aquellas en las que los datos están organizados en tablas. Las tablas están organizadas agrupando datos acerca del mismo tema y contienen columnas y filas de información. Las tablas se relacionan entre sí por medio del motor de base de datos cuando se solicita.

En general, se puede pensar en una base de datos como un conjunto de datos relacionados. En los primeros productos de base de datos, una base de datos por lo regular era solo un archivo, tal como empleados.dbf, el cual contenía una sola tabla de datos. Dentro del archivo empleados.dbf había columnas que relacionaban los datos del empleado, como remuneración, fecha de contratación, nombre, número de seguro social, etc. Había una fila para cada persona de la empresa, con los valores correspondientes en las columnas apropiadas. Los índices, empleados para acelerar el acceso a los datos, estaban en un archivo separado, como lo estaba cualquier elemento relacionado con la seguridad.

En SQL Server, una base de datos no necesariamente está atada a un archivo, es mas bien un concepto lógico con base en un conjunto de objetos relacionados. Por ejemplo, en SQL Server, una base de datos contiene no solo los datos, sino además la estructura de la base de datos, todos los índices, la seguridad de la base de datos y tal vez otros objetos como vistas o procedimientos almacenados, relacionados con esa base de datos en particular.

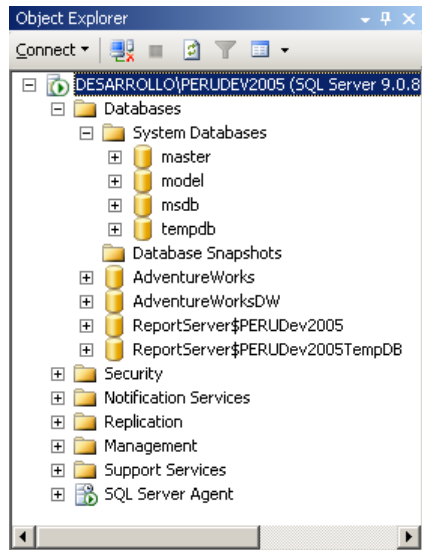
Objetos de una base de datos SQL Server 2005

Una base de datos relacional está compuesta de diferentes tipos de objetos. Los siguientes son algunos de los objetos más comunes:

- **Tablas (tables)**, son los objetos que contienen los datos de la base de datos. Una tabla es una colección de columnas donde los datos están organizados en un formato fila-columna similar al de una hoja de cálculo. Cada fila representa un registro único, y cada columna es un campo dentro del registro.
- **Columnas (columns)**, son las partes de la tabla que almacenan los datos. A una columna debe asignársele un tipo de dato y un nombre único.
- **Tipos de datos (datatypes)**, hay varios tipos de datos de donde elegir, como carácter, numérico o fecha. A una columna en una tabla se le asigna un solo tipo de dato.
- **Claves principales (primary keys)**, las claves son esenciales para las bases de datos relacionales. Las claves principales garantizan que cada fila es única en una tabla, proporcionando una forma de identificar de manera única cada elemento que se almacena.
- **Claves foráneas (foreign keys)**, las claves foráneas son columnas que hacen referencia a claves principales o restricciones únicas de otras tablas. SQL Server usa las claves principales y externas para relacionar datos almacenados en tablas separadas al efectuar consultas.
- **Restricciones (constraints)**, son mecanismos de integridad de datos implementadas por el sistema, con base en el servidor.
- **Disparadores o desencadenantes (triggers)**, son procedimientos almacenados que se activan cuando se agrega, se modifica o se elimina datos de una base de datos. Las reglas de negocio y la integridad de datos se suele definir mediante triggers. Por ejemplo, un trigger puede asegurar que todo libro en una biblioteca tenga asignado un editor.
- **Indices (indexes)**, pueden ayudarle a organizar los datos a efecto de que las consultas se ejecuten con mayor rapidez.
- **Statistics (estadísticas)**, es la distribución estadística de los elementos de una columna ó conjunto de columnas. Esta distribución es utilizada para establecer la eficiencia de un índice en la recuperación de datos.
- **Vistas (views)**, básicamente las vistas son consultas almacenadas en la base de datos que pueden hacer referencia a una o varias tablas. Puede crearlas y guardarlas a fin de utilizarlas con facilidad en el futuro. Por lo regular las vistas excluyen ciertas columnas de una tabla, o bien, vinculan varias tablas. También pueden emplearse como mecanismos de seguridad para que ciertos usuarios puedan consultarlas y otros no.

- **Sinónimos (synonyms)**, ahora SQL Server 2005 soporta sinónimos. Un sinónimo permite que un objeto pueda ser referenciado utilizando un nombre alternativo. A veces, para referirse a un objeto es necesario calificarlo completamente, lo que genera una referencia larga y compleja para el objeto. En este caso, podemos crear un sinónimo que permita establecer la referencia al objeto utilizando un nombre más corto y sencillo.
- **Procedimientos almacenados (stored procedures)**, éstos son similares a las macros que se utilizan en productos como Excel o Word. Podemos escribir código Transact-SQL y almacenarlo bajo un nombre. Al ejecutar el procedimiento almacenado, usted de hecho ejecuta el código Transact-SQL contenido en él. Un uso sería tomar el código Transact-SQL que ejecuta un informe semanal, guardarlo como un procedimiento almacenado, y a partir de ahí ejecutar solamente el procedimiento almacenado para generar informes. También pueden emplearse como mecanismos de seguridad.
- **Funciones definidas por el usuario (user-defined functions)**, adicionalmente a las funciones proporcionadas por Transact-SQL, podemos crear nuestras propias funciones para ampliar la capacidad del lenguaje.
- **Assemblies (ensamblados)**, en SQL Server 2005 es un objeto que referencia a un módulo de aplicación con código manejado (archivo .dll) que se ha creado bajo el CLR (Common Language Runtime) del .NET Framework.
- **Tipos de datos definidos por el usuario (user-defined data types)**, son tipos de datos con nombre propio creados en base a los tipos de datos estándar de SQL Server.
- **Tipos definidos por el usuario (user-defined types)**, son tipos creados en SQL Server 2005 que establecen referencias a tipos del CLR (Common Language Runtime) del .NET Framework.
- **Reglas (rules)**, se asignan a las columnas para garantizar que los datos que se registran en ellas cumplan con determinadas condiciones establecidas por la regla. Por ejemplo, puede utilizar una regla para asegurar que el campo o columna de números telefónicos solo contenga dígitos.
- **Valores predeterminados (defaults)**, se puede asignar un valor predeterminado a una columna de modo que si no se registra dato durante una operación INSERT en la columna específica, se utiliza dicho valor.
- **Usuarios (users)**, permiten establecer los permisos de quienes acceden a una base de datos. Una cuenta de usuario que accede a una base de datos obtiene sus permisos a través de un usuario de dicha base de datos.

Las bases de datos de sistema



Cuando se instala SQL Server, se crean cinco bases de datos de sistema, dos bases de datos de usuario, y dos bases de datos del Report Server.

Las bases de datos de sistema contienen las tablas de sistema, las que a su vez contienen metadatos; es decir, los datos que permiten operar y administrar el sistema.

Las bases de datos de usuario son como cualquier base de datos que los usuarios crean. Las creadas durante la instalación del servidor sirven como bases de datos de ejemplo.

Las bases de datos del Report Server almacenan datos como definición de reportes, reportes almacenados en caché, y la metadata de los reportes, que permiten operar y administrar los reportes.

La siguiente lista describe cada una de las bases de datos creadas durante la instalación de mi SQL Server 2005.

- **master (sistema):** Es la principal base de datos del sistema. Controla los usuarios y las operaciones sobre el servidor manteniendo datos como cuentas de usuario, variables de entorno, mensajes de error del sistema, etc.
- **model (sistema):** Proporciona una plantilla o modelo para cualquier base de datos nueva. Cuando se crea una base de datos, todo el contenido de la model se copia en la nueva base de datos.
- **msdb (sistema):** Almacena toda la data que utiliza el SQL Server Agent para programar alertas y trabajos, y para registrar operadores.
- **tempdb (sistema):** Para almacenamiento de tablas temporales.
- **Resource (sistema):** Es una base de datos de solo lectura que contiene copias de todos los objetos de sistema que vienen con SQL Server 2005. Estos objetos se encuentran físicamente en esta base de datos, pero aparecen lógicamente en el esquema **sys** de cada base de datos.
- **AdventureWorks (usuario):** Base de datos de Adventure Works Cycles, una multinacional que fabrica y vende bicicletas a nivel mundial.

- **AdventureWorksDW (usuario):** Base de datos de análisis de Adventure Works Cycles.
- **ReportServer\$PERUDev2005 (Report Server):** Base de datos que almacena la metadata de los reportes y que permite la operación y administración de reportes.
- **ReportServer\$PERUDev2005TempDB (Report Server):** Base de datos que almacena información de la sesión y del almacenamiento en caché de una sesión de Reporting Services.

Las tablas de sistema

Las tablas de sistema almacenan metadatos acerca del sistema y de los objetos de la base de datos.

El Catálogo de la Base de Datos

Cada una de las bases de datos, ya sea de sistema o de usuario, contiene una colección de tablas de sistema que almacenan datos que describen a una base de datos específica. Esta colección de tablas se conoce como el **Catálogo de la Base de Datos**, y pertenecen al esquema **sys** de la base de datos.

A partir de esta versión de SQL Server, las tablas de sistema no pueden ser consultadas directamente. Para consultar el Catálogo de una base de datos podemos utilizar las **Vistas del Catálogo (Catalog Views)** ó las **Vistas de Esquema de Información (Information Schema Views)**.

El siguiente cuadro muestra algunas de las vistas del catálogo que forman el Catálogo de la Base de Datos.

Vista del Catálogo	Base de datos	Descripción
sys.sysobjects	Todas	Tiene una fila por cada objeto almacenado en la base de datos.
sys.sysusers	Todas	Tiene una fila por cada usuario que tiene acceso a la base de datos.
sys.sysindexes	Todas	Tiene una fila por cada índice creado para las tablas de la bases de datos.

El Catálogo del Sistema

Es una colección de tablas de sistema que almacenan datos acerca del sistema y todas las bases de datos. Esta colección de tablas solo está presente en la base de datos de sistema **master**.

El siguiente cuadro muestra algunas de las vistas del catálogo que forman el **Catálogo del Sistema**.

Tabla de sistema	Base de datos	Descripción
sys.databases	master	Tiene una fila por cada base de datos en el servidor.
sys.syslogins	master	Tiene una fila por cada login name que puede conectarse al servidor.
sys.sysmessages	master	Tiene una fila por cada mensaje de error del sistema o advertencia que el servidor puede retornar.

Consultas a los Catálogos

Cuando se desea obtener información de un catálogo podemos hacerlo mediante una consulta a una vista del catálogo, utilizando un procedimiento almacenado del sistema, o mediante vistas de esquemas de información.

Con vista del catálogo

Podemos utilizar una instrucción SELECT sencilla para leer una vista del catálogo:

```
SELECT * FROM sys.sysobjects WHERE xtype = 'U'
```

Muestra una lista de todas las tablas de usuario de la base de datos actual efectuando una consulta a la vista del catálogo **sys.sysobjects**.

Con procedimiento almacenado del sistema

Un procedimiento almacenado del sistema es un programa almacenado en la base de datos, y que puede ser invocado por una aplicación para ejecutar operaciones en la base de datos.

```
sp_help clientes
```

Consulta el Catálogo de la base de datos actual para mostrar toda la información acerca de la definición de la tabla **clientes**.

Con vista de esquema de información

Una vista es una consulta preconstruída que se almacena en la base de datos, y que ser ejecutada por un programa. SQL Server cuenta con un conjunto de vistas denominadas de esquema de información que permiten averiguar sobre el esquema de una base de datos. El identificador de estas vistas tiene la forma **information_schema.objetos**.

```
SELECT * FROM information_schema.tables
```

Muestra una lista de todas las tablas de la base de datos actual.

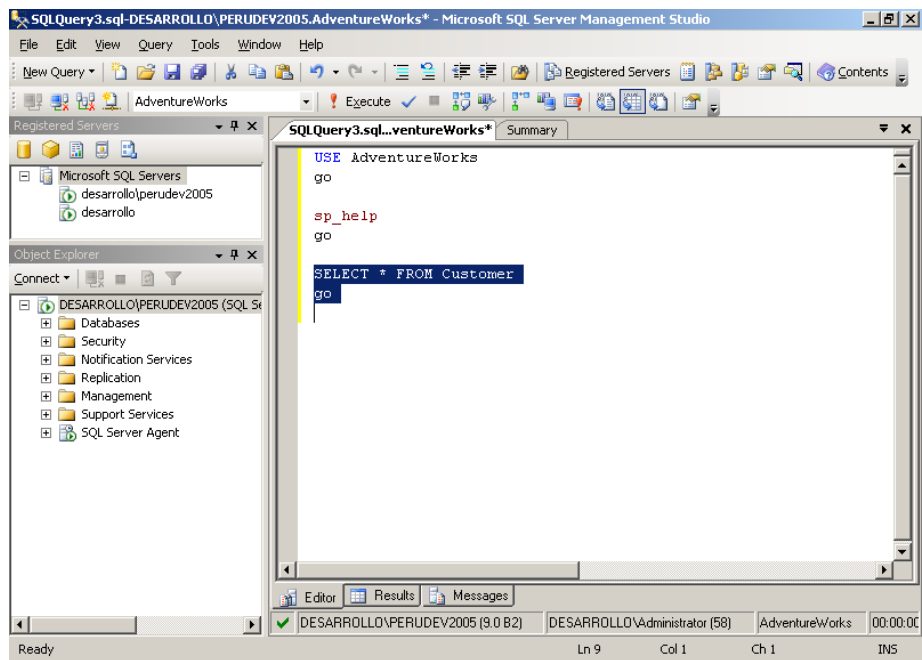
El Code Editor (Editor de Código)

Code Editor es un editor para crear y editar scripts (guiones). Reemplaza al **Analizador de Consultas (Query Analyzer)** de las versiones anteriores de SQL Server.

SQL Server Management Studio proporciona cuatro versiones del Code Editor: **SQL Query Editor**, **MDX Query Editor**, **XML Query Editor**, y **SQL Mobile Query Editor**.

Carga de SQL Query Editor

1. En **SQL Server Management Studio**, en el menú **File**, ejecute **New, New SQL Server Query**.
2. En el diálogo **Connect to SQL Server** seleccione el servidor SQL al que desea conectarse. Se añade una ficha **SQLQueryN.sql**.



SQL Query Editor trabaja como un editor de scripts, y como una ventana para ejecutar comandos interactivamente.

Para ejecutar el script contenido en el editor

Haga clic en el botón **Execute** de la barra de herramientas del editor. También puede ejecutarlo pulsando la tecla [F5].

Para ejecutar solo algunas instrucciones contenidas en el editor

Utilizando el mouse ó el teclado, seleccione las instrucciones que desea ejecutar. Luego, haga clic en el botón **Execute** de la barra de herramientas del editor. ó pulse la tecla [F5].

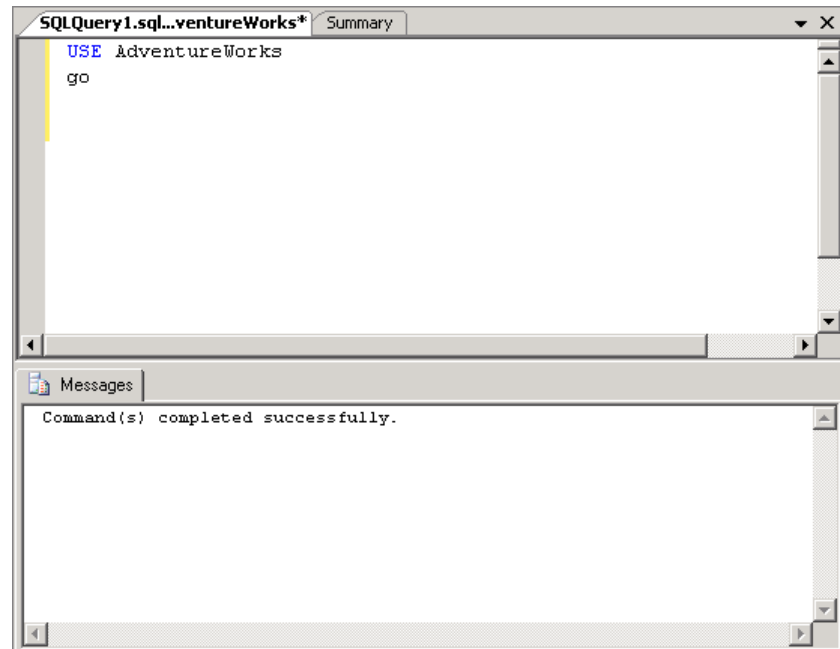
Para guardar el script

Haga clic en el botón **Save** de la barra de herramientas, ó ejecute **File, Save SQLQueryN.sql As.**

Ejercicio 4: Consulta al Catálogo de la base de datos AdventureWorks

1. En **SQL Query Editor**, escriba y luego ejecute el siguiente código:

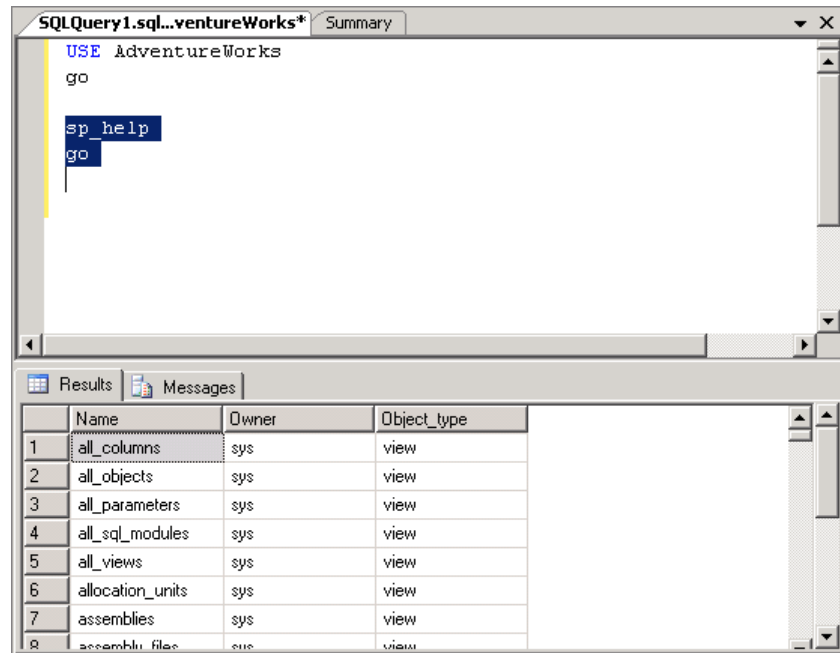
```
USE AdventureWorks  
go
```



En la barra de herramientas del editor, la lista **Available Databases** cambia de **master** a **AdventureWorks**.

2. Ahora, añada las instrucciones siguientes, y ejecútelas.

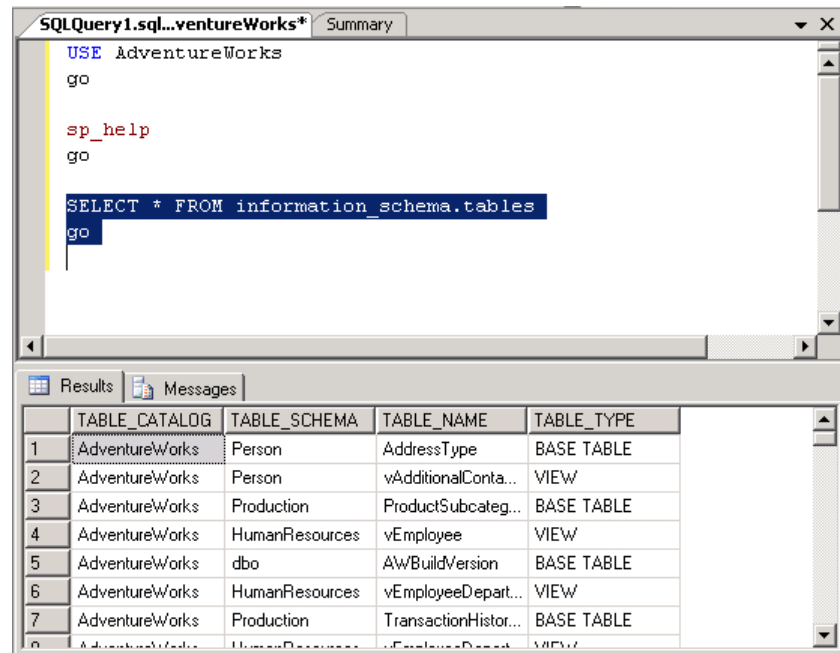
```
sp_help  
go
```



El procedimiento almacenado del sistema **sp_help**, ejecutado sin parámetros, muestra una lista de los objetos de base de datos contenidos en la base de datos actual (**AdventureWorks**).

3. Ejecute la siguiente instrucción para ver las tablas contenidas en **AdventureWorks**:

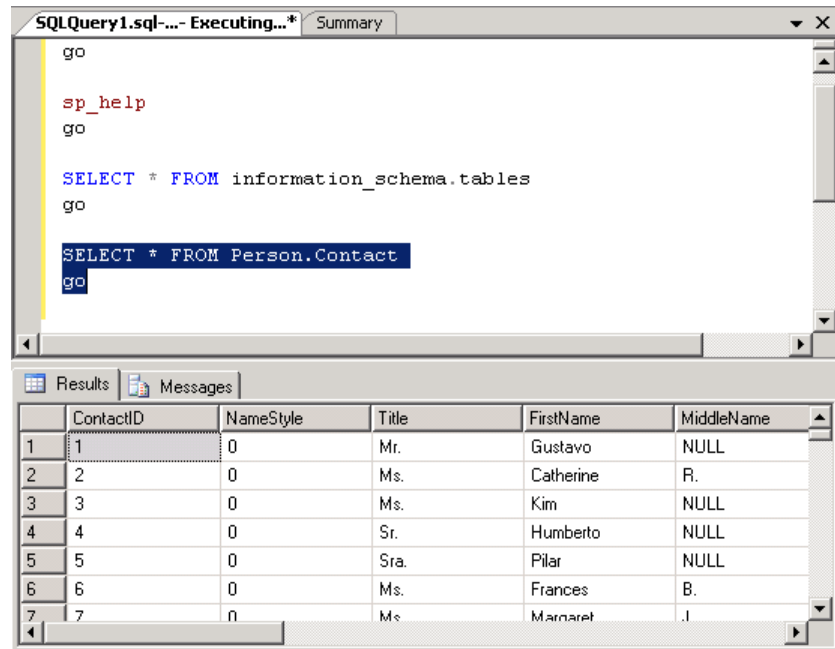
```
SELECT * FROM information_schema.tables  
go
```



4. Para ver el contenido de la tabla **Contact**, ejecute la siguiente instrucción:

```
SELECT * FROM Person.Contact
go
```

Según el Catálogo de la base de datos, la tabla **Contact** se encuentra en el esquema **Person**.



5. Para guardar el script, en el menú **File** ejecute **Save SQLQuery1.sql As**.
6. En el diálogo **Save File As**, seleccione la carpeta en la que almacenará el script.
7. En **File name**, digite el nombre del archivo. Digite por ejemplo, **Ejercicio004.sql**, y haga clic en **Save**.

Esta página se ha dejado en blanco intencionalmente.