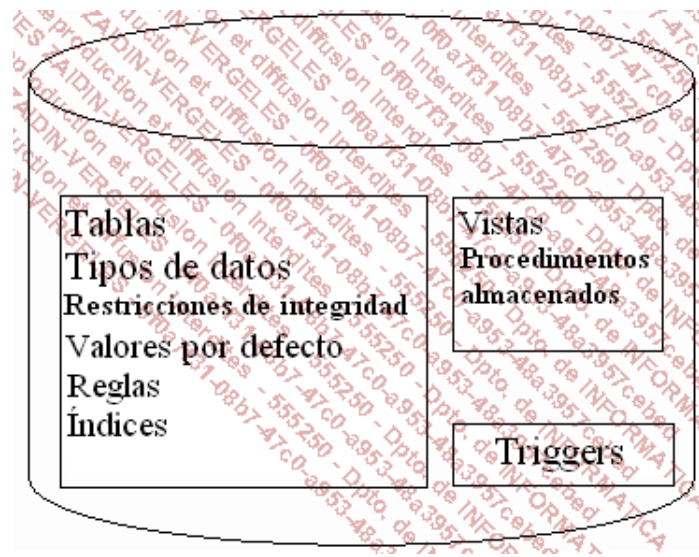


Base de datos SQL Server

1. Objetos de base de datos

Las bases de datos contienen cierto número de objetos lógicos. Es posible agrupar estos objetos en tres grandes categorías:

- Gestión y almacenamiento de los datos: tablas, tipos de datos, restricciones de integridad, valores por defecto, reglas e índices.
- Acceso a los datos: vistas y procedimientos almacenados.
- Gestión de integridad compleja: triggers (procedimientos almacenados que se ejecutan automáticamente en el momento de la ejecución de una orden SQL que modifique el contenido de una tabla: INSERT, UPDATE y DELETE). El trigger está siempre asociado a una tabla y a una instrucción SQL. Permite establecer reglas de integridad complejas entre varias tablas o mantener datos no normalizados.



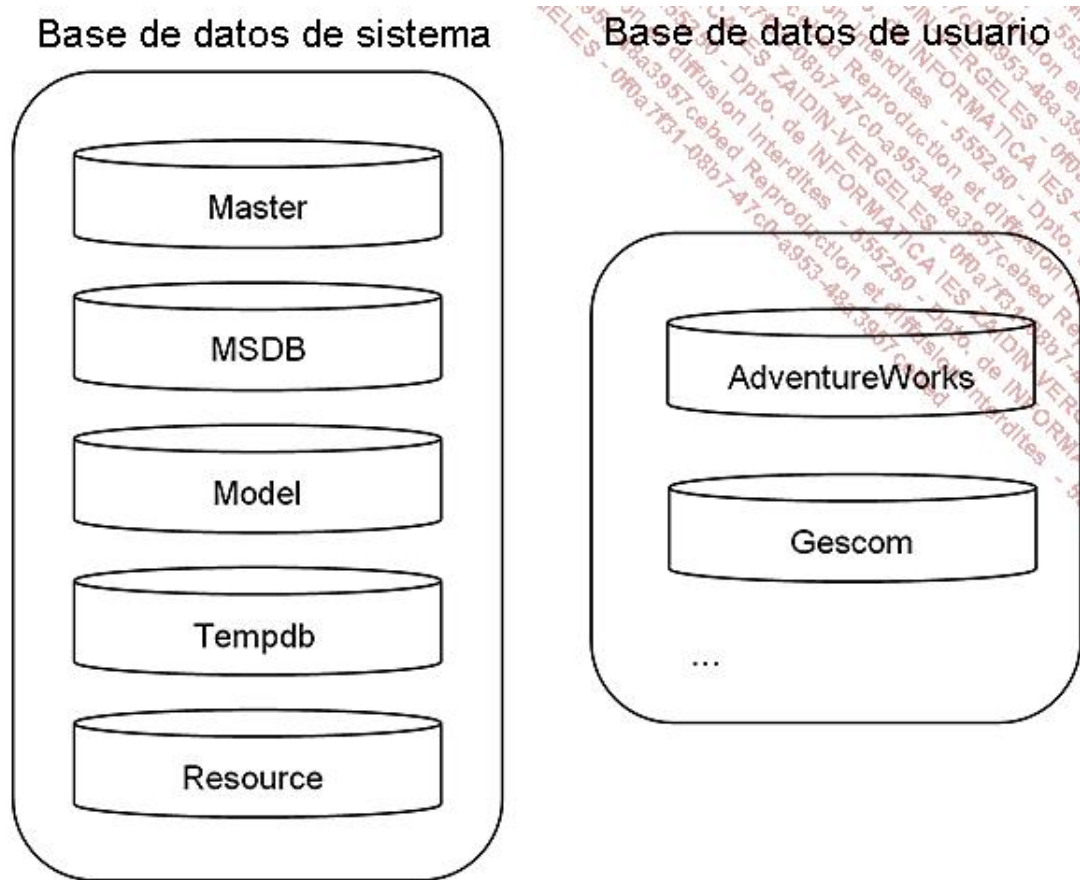
Objeto de base de datos

Nombre completo de los **objetos**

La regla aplicada para denominar a los objetos permite una identificación perfecta. El nombre completo se compone de la siguiente manera: *servidor.nombreBase.propietario.objeto*. Por defecto, sólo es necesario el nombre de los objetos. Este concepto se explica en profundidad en el capítulo Gestión de la base de datos.

2. Bases de datos de sistema y tablas de sistema

Para gestionar el conjunto de datos almacenados, SQL Server se utiliza a sí mismo. Por lo tanto, existen bases de datos de sistema y en cada base de datos de usuario algunas tablas de sistema. La inserción y la actualización de datos en estas tablas no se efectúan nunca de manera directa, sino a través de comandos Transact SQL o de procedimientos almacenados.



Organigrama de bases de datos



Los nombres de base de datos y de tablas de sistema son fijos y conocidos por SQL Server. Por lo tanto, no es necesario renombrar una tabla o una base de datos de sistema.

Master

Es la base de datos principal de SQL Server, donde está almacenado el conjunto de datos estratégicos para el buen funcionamiento del servidor (cuentas de conexión, opciones de configuración, existencia de bases de datos de usuario y las referencias a los ficheros que componen estas bases...).

Model

Esta base contiene el conjunto de elementos incluidos en toda base de datos de usuario nueva. Por defecto, sólo están las tablas de sistema, aunque es posible añadir otros elementos.

Tempdb

La base **Tempdb** es un espacio temporal de almacenamiento compartido. Permite gestionar las tablas temporales locales o globales, las tablas de trabajo intermedias para hacer ordenaciones, por ejemplo, pero también los conjuntos de resultado de los cursores. La base **Tempdb** se crea nuevamente, con su tamaño inicial, en el momento del inicio de la instancia. Por lo tanto, no se puede conservar de manera permanente ninguna información en la base de datos **Tempdb**. Los objetos temporales se eliminan cuando su propietario se desconecta.

Msdb

Contiene la información utilizada por el servicio SQL Server Agent para lanzar una alerta, avisar a un operador o ejecutar una tarea planificada. **Msdb** contiene también el histórico de ejecución de las tareas.

Resource

Esta base de datos de sólo lectura contiene la definición de todos los nuevos elementos definidos a partir de SQL Server 2005. Los objetos de sistema se definen aquí de manera lógica y aparecen en el esquema de usuario **sys**. Con esta base de datos, la migración de SQL Server 2000 a SQL Server 200x es más sencilla, ya que la simple adición de la base de datos origen permite obtener el conjunto de objetos definidos en SQL Server 2005 sin necesidad de

tocar la base de datos master.

Base de datos de usuario

Las bases de datos de usuario van a contener los datos proporcionados por los usuarios. Las bases presentes en el esquema anterior (AdventureWorks y Gescom) son las bases de ejemplo utilizadas en la documentación oficial de SQL Server y en este manual.


3. Las tablas de sistema

Las tablas de sistema siempre están presentes en SQL Server. Sin embargo, se recomienda no trabajar directamente con estas tablas. Para buscar información, es necesario pasar por el esquema de información y, concretamente, por las vistas definidas en el esquema del usuario **sys** cuando es posible.

En la tabla siguiente, se hace referencia a algunas tablas de sistema.


Catálogo de sistema (presente únicamente en la base Master)	
Tabla de sistema	Función
syslogins	Una línea para cada usuario o grupo Windows puede conectarse al servidor SQL.
sysmessages	Una línea para cada mensaje de error definido y para cada idioma.
sysdatabases	Una línea por base de datos de usuario.
sysconfigures	Una línea para cada opción de configuración del servidor.
sysusers	Una línea para cada usuario definido en la base.
syscolumns	Una línea para cada columna de las tablas, vistas y para cada parámetro de los procedimientos almacenados.
sysobjects	Una línea para cada objeto de la base de datos.

Las tablas de sistema son utilizadas directamente por el motor de SQL Server. Las aplicaciones que utilizan SQL Server no deben, en ningún caso, acceder directamente a estas tablas, ni siquiera en modo sólo lectura. Es más: como la estructura de estas tablas evoluciona con cada versión de SQL Server, si alguna aplicación accede directamente a las tablas de sistema, podemos encontrarnos con la imposibilidad de migrar a una nueva versión de SQL Server a no ser que la aplicación se escriba de nuevo.

 SQL Server no tiene en cuenta los triggers que puedan estar definidos sobre tablas de sistema ya que pueden entorpecer el buen funcionamiento de algunas operaciones.

4. Extracción de metadatos

Para consultar los datos contenidos en las tablas de sistema, se desaconseja hacer directamente una consulta de tipo SELECT. Es preferible hacer a través de procedimientos almacenados, funciones de sistema y vistas del esquema de información.

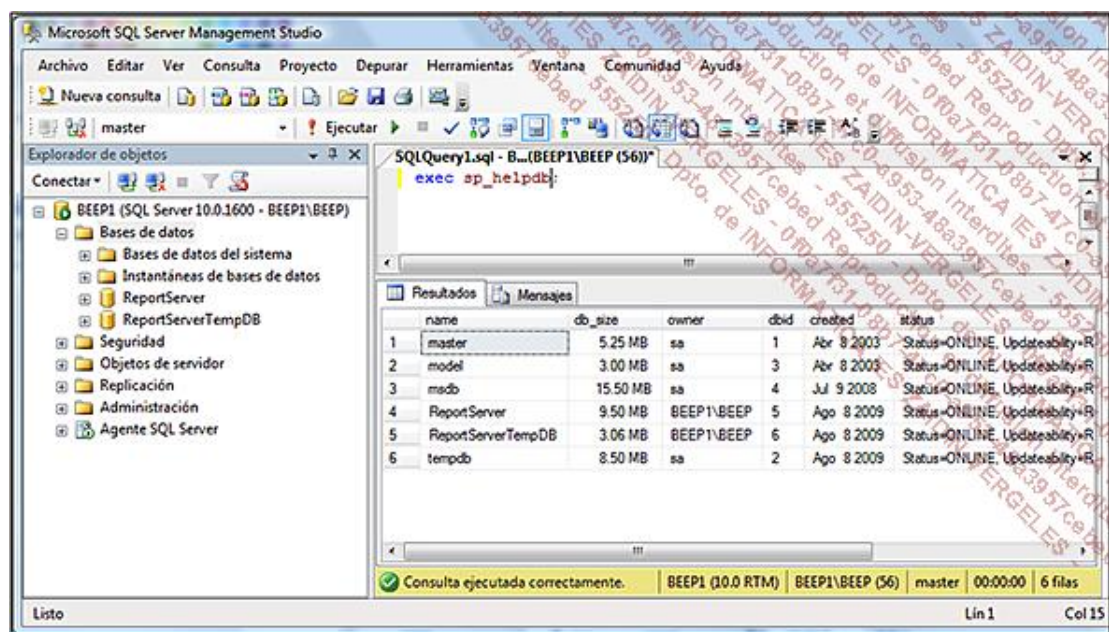
 Como administrador, es posible modificar el contenido de las tablas de sistema. Esta operación debe evitarse ya que puede tener consecuencias irreversibles y fatídicas. La única manera de solucionar un problema de este tipo sería restaurar una copia de seguridad.

Procedimientos almacenados de sistema

Los procedimientos almacenados de sistema se mantienen, la mayor parte de ellos, por razones de compatibilidad ascendente. Su utilización se desaconseja.

Para consultar las tablas de sistema, hay numerosos procedimientos almacenados. Todos ellos comienzan por **sp_**. Entre todos los procedimientos almacenados que existen, mencionamos los siguientes:

Procedimiento almacenado	Descripción
sp_help [nombre_objeto]	Información del objeto indicado.
sp_helpdb [nombre_base_datos]	Información de la base de datos indicada.
sp_helpindex [nombre_tabla]	Información de los índices de la tabla indicada.
sp_helplogins [nombre_conexión]	Información de la conexión indicada.
sp_who	Lista de los usuarios actualmente conectados.



El catálogo

SQL Server tiene vistas de sistema que permiten obtener información de sistema. Todas estas vistas están presentes en el esquema **sys**.

Con el objetivo de navegar lo mejor posible en estas vistas, éstas se agrupan en temas:




- Objetos, tipos e índices
- Servidores asociados
- CLR
- Reflejo
- Service Broker
- Seguridad
- Transacciones
- Configuración del servidor
- Información del servidor

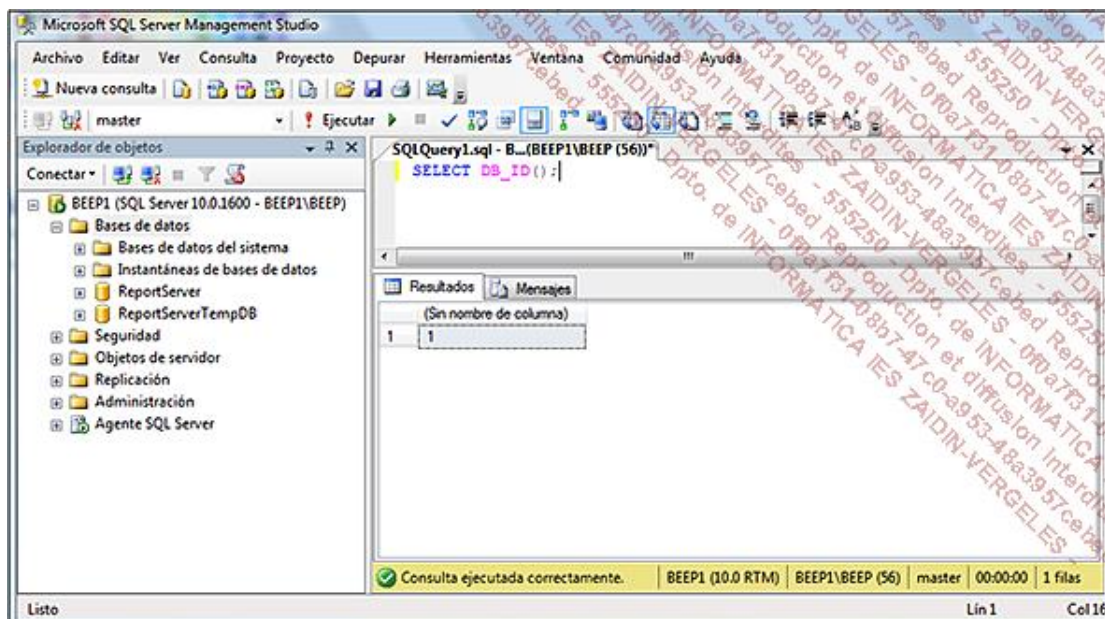
- Entorno de ejecución
- Almacenamiento
- Puntos de ruptura
- Particionamiento
- Trazas y eventos

Funciones de sistema

Las funciones de sistema se utilizan con comandos Transact SQL. De esta manera, es posible recuperar valores relativos a la base de datos en la que se trabaja, del servidor o de los usuarios.

Mencionamos algunas de ellas:

Funciones de sistema	Parámetro	Descripción
DB_ID 	Nombre	Devuelve el identificador de la base de datos.
USER_NAME 	ID	Devuelve el nombre de usuario a partir de su identificador.
COL_LENGTH	Columna	Longitud de la columna.
STATS_DATE	Índice 	Fecha de última actualización de las estadísticas.
DATALength	Tipo de datos	Longitud de la expresión.



Esquema de información

Es una serie de vistas que ofrecen una visualización de los parámetros de manera independiente de las tablas de sistema. El no hacer referencia directamente a las tablas de sistema impide posibles modificaciones que puedan aparecer en su estructura en próximas versiones. Además, estas vistas cumplen con la definición del estándar ANSI SQL para los esquemas de información. Cada vista presenta metadatos para el conjunto de los objetos presentes en la base de datos.

Las vistas del esquema de información:



CHECK_CONSTRAINTS

Visualiza el conjunto de restricciones del tipo CHECK definidas en la base de datos.

COLUMN_DOMAIN_USAGE

Visualiza el conjunto de columnas definidas en un tipo de datos de usuario.

COLUMN_PRIVILEGE

Visualiza el conjunto de privilegios, a nivel de columna, para el usuario actual o para otro usuario de la base de datos.

COLUMNS

Visualiza la definición del conjunto de columnas accesibles, en la base de datos actual, para el usuario actual.

CONSTRAINT_COLUMN_USAGE

Visualiza el conjunto de columnas para las que existe una restricción.

CONSTRAINT_TABLE_USAGE

Visualiza el conjunto de tablas para las que hay, al menos, una restricción definida.

DOMAIN_CONSTRAINTS

Visualiza el conjunto de tipos de datos de usuario que son accesibles para el usuario actual y que están asociados a una regla.

DOMAINS

Visualiza el conjunto de tipos de datos de usuario que son accesibles para el usuario actual.

KEY_COLUMN_USAGE

Visualiza el conjunto de columnas para las que hay definida una restricción de clave.

PARAMETERS

Visualiza las propiedades de los parámetros de las funciones definidas por el usuario y de los procedimientos almacenados accesibles para el usuario actual. Para las funciones, también se visualiza la información relativa al valor de retorno.

REFERENTIAL_CONSTRAINTS

Visualiza el conjunto de restricciones de clave foránea definidas en la base de datos actual.

ROUTINES

Visualiza el conjunto de procedimientos almacenados y de funciones accesibles para el usuario actual.

ROUTINE_COLUMNS

Visualiza las propiedades de cada columna reenviadas por las funciones accesibles para el usuario actual.

SCHEMATA

Visualiza las bases de datos para las que el usuario actual tiene permisos.

TABLE_CONSTRAINTS

Visualiza las restricciones a nivel de tabla de la base de datos actual.

TABLE_PRIVILEGES

Visualiza el conjunto de privilegios del usuario actual en la base de datos actual y el conjunto de privilegios que el usuario actual otorga a otros usuarios de la base de datos actual.

TABLAS

Visualiza el conjunto de tablas de la base de datos actual para las que el usuario actual tiene derechos.

VIEW_COLUMN_USAGE

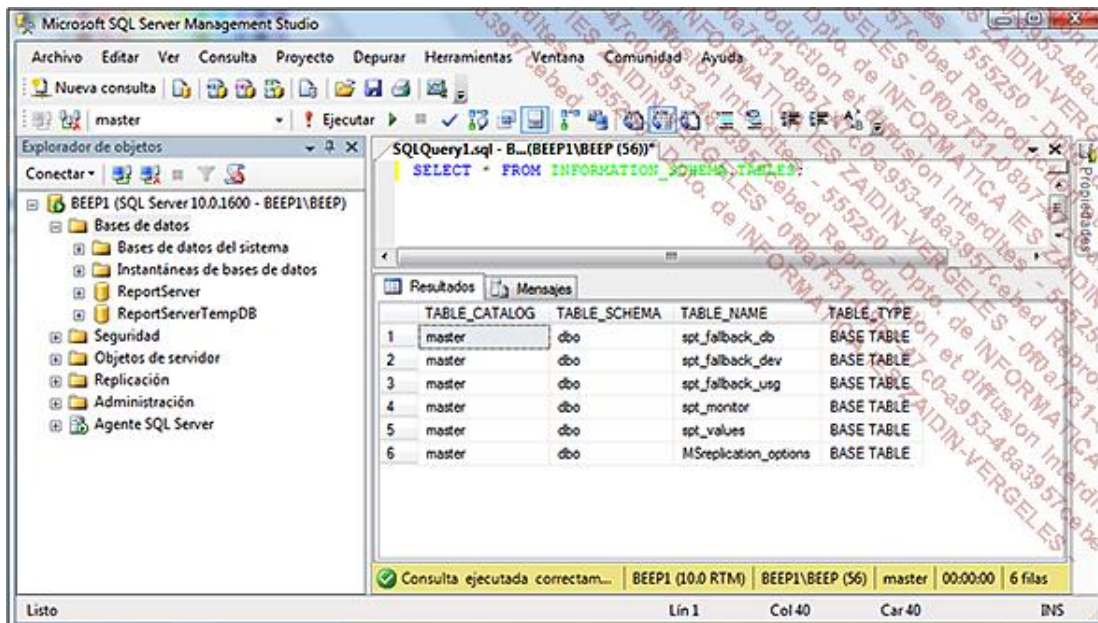
Visualiza el conjunto de columnas de tablas que se utilizan en la definición de una o varias vistas.

VIEW_TABLE_USAGE

Visualiza el conjunto de tablas de la base de datos actual que se utilizan a la hora de definir las vistas.

VIEWS

Visualiza el conjunto de vistas accesibles para el usuario actual en la base de datos actual.



5. Las tareas del administrador

El administrador de la base de datos tiene como objetivo principal mejorar el funcionamiento de la base de datos. Aunque SQL Server ofrece numerosas herramientas y algoritmos de autooptimización, todavía queda mucho trabajo para el administrador.

Las principales tareas del administrador son:

- Gestionar los servicios SQL Server;
- Gestionar las instancias SQL Server;
- Establecer los procesos de copia de seguridad y restauración;
- Configurar la disponibilidad de los datos de acuerdo con la política de la empresa;
- Gestionar las configuraciones de red;

- Importar y exportar datos.

Además de las competencias en sistemas que debe tener el administrador para ser capaz de gestionar lo mejor posible la instancia SQL Server, es importante que conozca las diferentes posibilidades ofrecidas por SQL Server para la automatización de las tareas con SQL Agent.

Para medir el resultado de su trabajo y comparar las diferentes opciones de configuración que puede tener que realizar, el administrador debe ser capaz de utilizar las herramientas y métodos asociados a SQL Server.

Por último, y sin duda el aspecto más importante, la administración de la base debe inscribirse en un proceso más global que implique al administrador desde el diseño de la base de datos para tomar las mejores decisiones en términos de arquitectura desde el momento de su diseño. El administrador podrá, de esta manera, intervenir en la creación de la base de datos y en las decisiones que se tomen, como por ejemplo, el número de archivos que utilizar, los índices, las vistas y los procedimientos almacenados que es necesario definir para optimizar el tráfico de red pero también simplificar la gestión de derechos de acceso. De la misma manera, será el administrador el que aconseje sobre la partición o no de una tabla.