

SQL Programming

Módulo 1 – Introducción a SQL Server

Introducción a SQL Server

¿Qué es SQL Server?

SQL Server es un sistema de administración de bases de datos relacionales basado en la arquitectura cliente-servidor. La finalidad de SQL SERVER es registrar, administrar y analizar datos asegurando la disponibilidad y seguridad de la información en una red corporativa.

El Motor de base de datos es el componente principal que se instala como un servicio en sistemas operativos Windows y Linux para almacenar, procesar y persistir los datos.

El Motor de base de datos proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones consumidoras de datos más exigentes de una organización.



Transact-SQL

Transact-SQL es el lenguaje que emplea para enviar peticiones entre el cliente y el servidor.

Es un lenguaje exclusivo de SQL Server, pero basado en el lenguaje SQL estándar, utilizado por casi todos los tipos de bases de datos relacionales que existen.



Historia

SQL se ha establecido como el lenguaje estándar de base de datos relacionales.

Hay numerosas versiones de SQL. La versión original se desarrolló en el laboratorio de investigación de San José, California (*San Jose Research Center*) de IBM, este lenguaje originalmente denominado *Sequel*, se implementó como parte del proyecto System R, a principios de 1970 [McJones 97]. Desde entonces ha evolucionado a lo que ahora se conoce como SQL (*Structured Query Language*, o lenguaje estructurado de consultas).



Historia

En 1986, ANSI (*American National Standards Institute*, Instituto Nacional Americano de Normalización) e ISO (*International Standards Organization*, Organización Internacional de Normalización) publicaron una norma de SQL denominada SQL-86.

En 1987 IBM publicó su propia norma de SQL denominada SAA-SQL (*System Application Architecture Database*, Interfaz de base de datos para arquitecturas de aplicación de sistemas).

En 1989 se publicó una norma extendida para implementaciones de SQL pueden ser compatibles sólo con SQL-89, no siéndolo con SQL-92. SQL (SQL-89) y actualmente los SGBD son compatibles al menos con esta norma. La norma actual de SQL de ANSI/ISO es la SQL-92. Se debe tener en cuenta que algunas

Bases de sistema de SQL Server

Las **bases de datos de sistema** son bases de datos creadas por SQL Server para autoadministrarse. Son bases de datos que no pueden borrarse y donde el sistema guarda información interna y de otras bases de datos (metadatos).

BASE MASTER

La base de datos maestra registra toda la información de sistema de un sistema SQL Server. Dentro de esta información se incluyen los metadatos de todas las instancias, como las cuentas de inicio de sesión, los extremos, los servidores vinculados y la configuración del sistema.

Asimismo, Master es la base de datos que registra la existencia de las demás bases de datos, la ubicación de los archivos de las bases de datos y la información de inicialización de SQL Server. SQL Server no puede iniciarse si la base de datos master no está disponible.

Bases de sistema de SQL Server

BASE MSDB

La base MSDB principalmente tiene la función de dar soporte al Agente de SQL Server, de tal modo que almacena la definición y planificación de JOB's, Planes de Mantenimiento, etc. (incluyendo su historial de ejecución), como así también almacena la definición de otros objetos como Operadores, Alertas, etc.

BASE MODEL

Esta base de datos sirve de modelo o plantilla. Siempre que se crea una nueva base de datos (CREATE DATABASE), se realiza una copia de la base de datos MODEL, heredando de ésta su configuración y contenido, salvo que se especifique lo contrario.

Bases de sistema de SQL Server

BASE TEMPDB

Almacena tanto los objetos temporales (tablas temporales, procedimientos almacenados temporales, etc.), como los resultados intermedios que pueda necesitar crear el motor de base de datos, por ejemplo, durante la ejecución de consultas que utilizan las cláusulas GROUP BY, ORDER BY, DISTINCT, etc.

Además, TEMPDB se crea de nuevo siempre que se inicia la instancia SQL Server, tomando su tamaño por defecto. Conforme requiera más espacio, TEMPDB crecerá hasta el tamaño que necesite, y dado que el crecimiento de un

archivo implica esperas debido a la entrada / salida, es muy importante en entornos críticos dimensionar correctamente TEMPDB para que se cree con un tamaño apropiado, y evitar dichas esperas de entrada/salida en tiempo de ejecución, justo cuando estamos ofreciendo servicio a los usuarios. Evidentemente, estamos desplazando dichas esperas al momento de inicio de la instancia, pero también conseguimos una ventaja adicional al crear de una vez TEMPDB: limitamos la fragmentación (también muy importante en entornos críticos).

Autenticación en SQL Server

SQL Server admite dos modos de autenticación, el modo de autenticación de Windows y el modo mixto.

- La autenticación de Windows es el modo predeterminado, y a menudo se denomina seguridad integrada debido a que este modelo de seguridad de SQL Server está estrechamente integrado con Windows. Para iniciar sesión en SQL Server, se confía en las cuentas de usuario y grupo específicas de Windows. Los usuarios de Windows que ya hayan sido autenticados no tienen que presentar credenciales adicionales.
- El modo mixto admite la autenticación tanto de Windows como de SQL Server. Los pares de nombre de usuario y contraseña se mantienen en SQL Server.



Importante: Se recomienda utilizar la autenticación de Windows siempre que sea posible. Este modo de autenticación usa una serie de mensajes cifrados para autenticar usuarios en SQL Server. Cuando se utilizan inicios de sesión de SQL Server, los nombres de inicio de sesión de SQL Server y las contraseñas cifradas se pasan a través de la red, lo que hace que sea menos segura.

Lenguajes de Definición y Manipulación

Lenguaje de Definición de Datos

Las instrucciones del lenguaje de definición de datos (DDL) definen estructuras de datos. Use estas instrucciones para crear, modificar o quitar estructuras y objetos de bases de datos.

ALTER	CREATE	DROP
DISABLE TRIGGER	ENABLE TRIGGER	TRUNCATE TABLE

Lenguaje de Manipulación de Datos

El lenguaje de manipulación de datos (DML) afecta a la información almacenada en la base de datos. Use estas instrucciones para insertar, actualizar y modificar los registros de las tablas de la base de datos.

BULK INSERT	DELETE	INSERT
UPDATE	MERGE	SELECT



Lenguajes de Definición y Manipulación

Lenguaje de Control de Datos

Los comandos DCL se utilizan para el control de acceso y la gestión de permisos para los usuarios en nuestra base de datos.

GRANT	DENY	REVOKE
-------	------	--------

Lenguaje de control de Transacciones

Con los comandos TCL podemos gestionar y controlar las transacciones T-SQL para que podamos estar seguros de que nuestra transacción se realiza correctamente y que no se viola la integridad de nuestra base de datos.

BEGIN TRAN	COMMIT TRAN	ROLLBACK
------------	-------------	----------



Select

Devuelve todos los estados posibles para la columna seleccionada del modelo. Los valores devueltos varían dependiendo de si la columna especificada contiene valores discretos, valores numéricos de datos discretos o valores numéricos continuos.

El siguiente ejemplo hace uso del * y retorna todas las filas y todas las columnas de una tabla.

Sintaxis

```
SELECT * FROM Production.Product;
```

El siguiente ejemplo hace uso de la cláusula top y retorna 10 productos.

Sintaxis

```
SELECT top 10 ProductID  
FROM Production.Product;
```

Select

El siguiente ejemplo hace uso de la cláusula OFFSET y retorna todo el contenido de la tabla a excepción de los 10 primeros productos.

Sintaxis

```
SELECT ProductID  
FROM Production.Product  
ORDER BY ProductID  
OFFSET 10 ROWS;
```

El siguiente ejemplo hace uso de la cláusula OFFSET para posicionarse en el décimo registro y retorna 5 productos.

Sintaxis

```
SELECT ProductID  
FROM Production.Product  
ORDER BY ProductID  
OFFSET 10 ROWS FETCH NEXT 5 ROWS ONLY;
```

Comentarios

Los comentarios se pueden insertar en una línea independiente. El servidor no evalúa el comentario.

Sintaxis

-- Aquí un Comentario



Comentario de bloque

Indica texto proporcionado por el usuario. El servidor no evalúa el texto situado entre `/*` y `*/`.

Sintaxis

`/*`

Aquí un

Comentario

`*/`



Alias y Literales

Hay dos tipos de alias que se utilizan con mayor frecuencia: **alias de columna** y **alias de tabla**. Los alias de columna existen para ayudar en la organización del resultado. El uso de un alias de columna hace el resultado mucho más legible.

El segundo tipo de alias es el alias de tabla. Esto se alcanza al colocar un alias directamente luego del nombre de tabla en la cláusula FROM. Es conveniente cuando desea obtener información de dos tablas separadas (el término técnico es 'realizar uniones', este punto le veremos más adelante).

En el siguiente ejemplo se renombra la columna *Name* por *Producto*.

Sintaxis

```
SELECT Name as Producto, 'Hola Mundo' as Literal  
FROM    Production.Product;
```

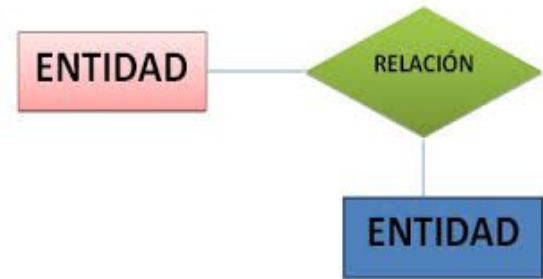
Diseño de la Base de Datos

Introducción

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en una computadora; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas se debe realizar un modelo de datos.

El modelo entidad-relación (E-R) es uno de los varios modelos conceptuales existentes para el diseño de bases de datos.

Suele cometerse el error de ir creando nuevas tablas a medida que se van necesitando, haciendo así el modelo de datos y la construcción física de las tablas simultáneamente.



Diseño de la Base de Datos - Introducción

El modelo de datos más extendido es el denominado **ENTIDAD/RELACIÓN (E/R)**. En el modelo E/R se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades.

Entidad

- Una entidad es cualquier "objeto" discreto sobre el que se tiene información.
- Cada ejemplar de una entidad se denomina *instancia*.
- Las entidades son modeladas en la base de datos como tablas.

Relación

- Una relación describe cierta interdependencia (de cualquier tipo) entre una o más entidades.
- Una relación no tiene sentido sin las entidades que relaciona.
- Las relaciones son definidas con claves primarias y claves foráneas, y mantienen la integridad referencial.

Diseño de la Base de Datos - Introducción




Cardinalidad de las Relaciones

Una relación describe cierta interdependencia (de cualquier tipo) entre una o más entidades.

Relaciones de **uno a uno**: una instancia de la entidad A se relaciona con una y solamente una de la entidad B.

Relaciones de **uno a muchos**: cada instancia de la entidad A se relaciona con varias instancias de la entidad B.

Relaciones de **muchos a muchos**: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

Tipo	Relación	Representación
1:1	Una a una: La cardinalidad máxima en ambas direcciones es 1.	1  1
1:N	Una a muchas: La cardinalidad máxima en una dirección es 1 y en la otra, muchos.	1  N
N:M	Muchas a muchas: La cardinalidad máxima en ambas direcciones es muchos.	N  M

Diseño de la Base de Datos

Atributos

Las entidades tienen atributos. Un *atributo* de una entidad representa alguna propiedad que nos interesa almacenar. En el modelo de Bases de Datos, los atributos son almacenados como columnas o campos de una tabla.

Consideraciones en el Planeamiento del Diseño Lógico de la Base de Datos

- Determinar el negocio y las necesidades del usuario. Considerando cuales son los problemas que hay que salvar y las tareas que los usuarios deberán completar.
- Crear Bases de Datos normalizadas. Prevenir la existencia innecesaria de información duplicada, inconsistencias en la base de datos, anomalías y problemas de pérdida de la información.

Diseño de la Base de Datos

Tablas

SQL emplea tablas como objetos de almacenamiento de datos, que los usuarios manipulan a través de sus aplicaciones. Las tablas son objetos compuestos por una estructura (conjunto de columnas) que almacenan información interrelacionada (filas) acerca de algún objeto en general.

Sus características:

- Las tablas tienen un solo nombre y es único en toda la base datos.
- Están compuestas por registros y campos (filas y columnas).
- Los registros y columnas pueden estar en diferentes órdenes.
- Una base de datos puede contener muchas tablas. Cada tabla almacena información.

Diseño de la Base de Datos

Restricciones de las Tablas

- Los nombres de las tablas deben ser únicos en la base de datos.
- Los nombres de las columnas (campos) deben ser únicos en la tabla.
- No puede haber dos registros con el mismo valor de la clave primaria.

Columnas No Descomponibles

- Son aquellas columnas que contienen cierta información que no puede ser almacenada en dos o más columnas.
- Son fáciles de actualizar.
- Son fáciles de consultar.
- Recomendadas para mantener la integridad de los datos.

Diseño de la Base de Datos

Restricciones en las columnas

- Not NULL

Clave Primaria (PRIMARY KEY)

- La Clave Primaria es una columna o un grupo de columnas que fuerzan la integridad de los datos en la tabla, asegurándose que cada fila en la tabla es la única (unicidad del registro).
- Solo puede haber una sola clave primaria por tabla.
- La clave primaria no permite valores nulos o duplicados.

Clave Foránea (FOREIGN KEY)

- La Clave Foránea referencia a la clave primaria de una tabla.
- Ésta puede referenciar a la clave primaria de la misma tabla o de otra.
- SQL Server valida la consistencia e integridad de los datos almacenados en una clave foránea.
- Fuerza la integridad referencia

¡Muchas gracias!

¡Sigamos trabajando!