SQL

SQL (lenguaje estándar utilizado para bases de datos relacionales). Lenguaje de consulta estructurado. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Unifica características del algebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de una base de datos, de una forma sencilla y en un lenguaje similar al inglés utilizado para la comunicación entre personas.

Es un lenguaje normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos, o sobre las estructuras de los mismos.

Usos de SQL

- **Definición de datos**: permite que el usuario defina la estructura y la organización de los datos almacenados, así como las relaciones existentes entre ellos.
- **Recuperación de datos**: permite a un usuario o un programa recuperar y utilizar los datos almacenados en la base de datos.
- **Manipulación de datos:** permite a un usuario o un programa actualizar la base de datos añadiendo datos nuevos, borrando datos viejos y modificando los almacenados previamente.
- **Control de acceso:** Se utiliza para restringir la capacidad de un usuario para recuperar, añadir y modificar los datos, protegiendo los datos almacenados contra acceso no autorizado.
- **Compartición de información:** Para coordinar la compartición de datos entre usuarios concurrentes, asegurándose que no haya interferencia entre ellos.
- **Integridad de datos:** Define restricciones de integridad en la base de datos, protegiéndola de alteraciones indebidas a actualizaciones inconsistentes o fallos del sistema.

Beneficios

- **Independencia de los proveedores:** Esfuerzo mínimo de conversión entre un DBMS y otro.
- **Portabilidad entre sistemas informáticos**: las aplicaciones SQL que comienzan en sistemas monousuarios, pueden ser transferidos a sistemas mayores según van creciendo.
- **Fundamento relacional**: SQL es un lenguaje de base de datos relacionales, y ha llegado a ser popular junto con el modelo de base de datos.
- **Estructura de alto nivel parecida al inglés**: las sentencias son frases sencillas en inglés.
- Consulta interactiva ad hoc: proporciona a los usuarios acceso temporario a los datos almacenados.
- **Múltiples vistas de datos:** se pueden ser distintas vistas de la estructura y contenidos de una base de datos a diferentes usuarios.
- **Definición dinámica de datos:** la estructura de una base de datos puede ser modificada y ampliada dinámicamente, incluso mientras los usuarios están accediendo a los contenidos de la base de datos.
- **Arquitectura cliente/servidor:** los programas utilizan una red local para comunicarse con los servidores de base de datos que almacenan los datos compartidos.

SQL Server

SQL Server es un Sistema de gestión de base de datos relacionales diseñado para trabajar con grandes cantidades de información.

Arquitectura cliente/servidor

Utiliza la ***arquitectura cliente/servidor*** para separar la carga de trabajo en tareas que se ejecutan en equipos servidores y tareas que se ejecutan en equipos clientes:

- El cliente es el responsable de la logística empresarial y de presentar los datos del usuario.
- SQL Server administra las bases de datos y asigna los recursos disponibles del servidor entre distintas solicitudes.

Permite diseñar y distribuir aplicaciones para mejorar una gran variedad de entornos. Las interfaces de programación de los clientes proporcionan los medios para que las aplicaciones se ejecuten en equipos clientes independientes y se comuniquen con el servidor a través de la red.

Permite diseñar y distribuir aplicaciones para mejorar una gran variedad de entornos. Las interfaces de programación de los clientes proporcionan los medios para que las aplicaciones se ejecuten en equipos clientes independientes y se comuniquen con el servidor a través de la red.

Entorno Cliente/Servidor:

- Mantener las relaciones entre los datos de una base de datos.
- Asegurar que los datos estén correctamente almacenados y que no se infrinjan las reglas que definen las relaciones entre los datos.
- Recuperar todos los datos hasta un punto de coherencia garantizada, en caso de que produzca un error del sistema.

Objetos de SQL Server

Objeto de base de datos	Descripción
Tabla	Define un conjunto de filas que tienen columnas asociadas. Tipo de datos: Define los valores permitidos para los datos de una columna o variable.
SQL Server proporciona tipos de datos del sistema.	Los usuarios crean tipos de datos definidos por el usuario.
Restricción	Define reglas relativas a los valores permitidos en las columnas y es el mecanismo estándar para exigir la integridad de los datos.
Valor predeterminado	Define un valor que se almacena en una columna si no se especifica explícitamente ningún otro valor.
Regla	Contiene información que define los valores válidos que se pueden almacenar en una columna o en un tipo de datos.
Índice	Es una estructura de almacenamiento que proporciona acceso rápido para la recuperación de los datos y puede exigir la integridad de los datos.
Vista	Proporciona una forma de ver los datos de una o varias tablas o vistas de una base de datos.
Función	Definida por el usuario puede devolver un valor escalar o una tabla. Las funciones se utilizan para encapsular la lógica que se realiza con frecuencia.
Procedimiento almacenado	Es una colección con nombre de instrucciones Transact-SQL compiladas previamente que se ejecutan juntas.

Sentencias de manipulación de datos

Este tipo de sentencias son utilizadas para crear y manejar objetos y propiedades de una base de datos, tales como tablas, vistas, procedimientos, etc. Las sentencias son:

Para crear un objeto:

CREATE nombre_objeto

Para modificar un objeto:

ALTER nombre_objeto

Para eliminar un objeto:

DROP nombre objeto

Creación de una tabla

Sintaxis de una sentencia básica de creación de tabla:

```
CREATE TABLE nombre_tabla
(nom_col1 tipo_dato [IDENTITY(inicio,incremento)][NULL|NOT NULL],
nom_col2 tipo_dato,
nom_col_n tipo_dato
[CONSTRAINT nombre_constraint PRIMARY KEY(nom_col1)],
[CONSTRAINT nombre_constraint FOREIGN KEY (nom_col2)
REFERENCES tabla2 (nombre_columna_tabla2)]
)
```

En donde:

- CREATE TABLE nombre_tabla: se debe especificar el nombre de la tabla a crear.
- Nom_col1 tipo_dato: se describen los nombres de columnas (nombre de campo) con su tipo de dato, cuando sean más de dos campos se separa con comas.
- IDENTITY(inicio, incremento): es una propiedad que se utiliza para definir a una columna como columna identidad. Columna cuyo contenido se genera automáticamente iniciando con el valor que se le indica en inicio que es el primer valor a insertarse en la tabla e incremento es un valor que indica en cuánto se incrementa el siguiente valor a insertarse.

  Provee un valor único e incremental, y se suele utilizar en conjunto con la PRIMARY KEY. Solo puede ser asignada a tipos de datos numéricos, y puede existir solo una columna con esta propiedad por tabla.

- CONSTRAINT: esta propiedad es opcional y se utiliza para hacer cumplir la integridad de los datos. Algunas de las Constraint que se pueden crear son:
- PRIMARY KEY: para establecer la clave principal o primaria de la tabla
- FOREIGN KEY: para establecer las claves secundarias o foráneas de la tabla con respecto a otras.
- Lo que está entre [ ] es opcional incorporarlo o no en la sentencia.
- Lo que se separa por la barra vertical "|" es que debe elegirse uno o el otro.
- Las palabras en mayúsculas son palabras reservadas del lenguaje es decir que deben escribirse tal como se presenta en la sintaxis y no deben utilizarse para nombres de otros elementos como objetos, campos, etc.

Tipos de datos

Tipo genérico	Tipo de dato almacenado	Nombre en SQL Server	Cantidad de bytes que ocupa en el almacenamiento
Binario	Almacenan cadenas de bits expresados en números hexadecimales.	**binary**	Longitud fija (hasta 8 Kb)
		varbinary	Puede variar en el nro. de dígitos hexadecimales (hasta 8 Kb.)
		image	Longitud variable y puede exceder los 8 Kb.
Alfanumérico o Caracter	Combinación de etras, símbolos y números.	**char**	Longitud fija (hasta 8 Kb)
		varchar	Puede variar el nro. de caracteres (hasta 8Kb.)
		text	Pueden ser caracteres ASCII y exceder los 8 Kb.
Fecha y Hora	Fechas y horas (día, mes y año y hora, minuto y segundos según el formato del sistema	**datetime**	Fechas entre el 01/01/1753 hasta 31/12/9999 (8 by)
		smalldatetime	Fechas entre el 01/01/1900 hasta 06/06/2079 (4 by)
Numéricos Decimal (punto fijo)	Números con decimales de coma fija.	**decimal**	Pueden tener hasta 38 dígitos
		numeric	Equivalente al decimal.
Numéricos Decimal (punto flotante)	Números con decimales aproximados	**float**	Entre 1,79E + 308 hasta 1,79 + 308
		real	Entre 3,40E + 38 hasta 3,40 + 38.

Numéricos Enteros	Datos numéricos enteros positivos o negativos. Tinyint no tiene negativos	**bigint**	Entre 2^63 hasta 2^63 - 1 (8 by)
		int	Entre -2,147,483,648 hasta 2,147,483,648 (4by)
		smallint	Entre -32,768 hasta 32,767 (2by)
		tinyint	Entre 0 y 255 (1 by)
Numérico Moneda	Numérico decimal para cantidades monetarias positivas o negativas	**money**	Entre 922,337,203,685,477,5808 positivos y negativos (8 by)
		smallmoney	Entre 214,748,3648 (4 by)
	Especiales	**bit**	Datos que consisten de 1 o 0. Similar al True o False
Tipo genérico	**Tipo de dato almacenado**	**Nombre en SQL Server**	**Cantidad de bytes que ocupa en el almacenamiento**
		cursor	Es empleado en variables o procedimientos almacenados.
		timestamp	Indica la actividad que ocurre sobre una fila.
		uniqueidentifier	Es un identificador único de la fila dentro de la base de datos
		sql_variant	Almacena varios tipos de datos excepto el tect, ntext, timestamp, image.
		table	Almacena un resultado de una consulta.
Unicode	Se puede almacenar datos Unicode. Utilizan 2 byte por cada carácter	**nchar**	Longitud fija hasta 4000 caracteres Unicode.
		nvarchar	Longitud variable hasta 4000 caracteres unicode
		Ntext	Exceden los 4000 caracteres unicodes.

Sentencias de actualización de datos (DML)

Con las **sentencias de manipulación de datos** se pueden recuperar datos de tablas de una base de datos, ingresar nuevos registros, modificar o eliminar los existentes.

Las sentencias son:

- **Select:** Para recuperar los datos de una o más tablas.
- **Insert:** Para insertar o cargar datos nuevos.
  Insert into *nombre_tabla* **(***lista_columnas***)**
  Values (*lista_valores***)**
- **Update:** Para modificar datos existentes.
  Update *nombre_tabla*
  Set *columna1 =nuevo_valor1[,*
  Columna2=nuevo_valor2]
  Where *condición*

  Si la actualización de una fila no cumple con una restricción o regla, infringe la configuración de valores NULL o si el nuevo valor es de un tipo de datos incompatible, se cancela la instrucción, se devuelve un error y no se actualiza ningún registro

- **Delete:** Eliminar datos existentes.
  Delete *nombre_tabla*
  Where *condición.*
  Si no se especifica la cláusula WHERE, se borran todas las filas de una tabla.

Recuperación de datos

Permite ver o interactuar con los datos en las tablas de una base de datos. Se debe utilizar un conjunto de comandos e instrucciones definidos por el software del DBMS (Sistema gestor de base de datos).
Es una de las tareas que lleva a cabo con mayor frecuencia. Cuando existen varias tablas y las relaciones entre ellas son complejas, las consultas pueden llegar a ser francamente complicadas, tardando en ejecutarse una cantidad de tiempo nada despreciable.

Sentencia select

Select se utiliza para obtener un conjunto de registros que pueden proceder de una o más tablas. Es la sentencia más potente y compleja de SQL. Recupera datos de una base de datos y devuelve en forma de tablas que no quedan guardadas en la base.

La sintaxis de la sentencia completa es:

SELECT [DISTINCT|ALL] *lista_columnas*

[INTO nueva_tabla] --Propio de SQL Server

FROM l*ista_tablas*

[**WHERE** *condición*]

[**GROUP BY** *lista_columnas*]

[**HAVING** *condición*]

[**ORDER BY** *lista_columnas* [ASC|DESC]]

- *lista_columnas*: se especifica el nombre de la/s columna/s de las cuales se

quiere mostrar la información separadas por comas. Esta lista recupera y muestra las columnas en el orden especificado.

- DISTINCT|ALL. Distinct elimina filas duplicadas en el resultado de la consulta. Por defecto está definida la cláusula ALL es decir muestra todo.
- Select top: Si se quiere limitar la cantidad de filas devueltas por la sentencia select, se puede utilizar combinada con Top seguido por el número de filas.
- INTO nueva_tabla esta cláusula es propia de SQL Server. Define la creación de una nueva tabla a partir de la respuesta a la consulta especificada.
- FROM lista_tablas: se especifica el nombre de la/s tabla/s de las cuales se quiere obtener la información. Si hay más de una tabla se separan con comas y habrá que prestar atención a la forma en que las tablas deben relacionarse o combinarse en la sentencia select (Join).
- WHERE condición: Con frecuencia suele ser necesario aplicar algún tipo de condición que restringa el número de registros devuelto por una consulta. Estas son condiciones de búsqueda o bien, en alguna bibliografía se lo suele encontrar como test del where que se verá más adelante.
- GROUP BY: Establece la lista de columnas por las cuales se agrupará la Información.
- HAVING: Condición que permite filtrar los grupos generados por GROUP BY.
- ORDER BY: lista de columnas que debe utilizarse como criterio para ordenar los registros. La palabra reservada ASC indica que el orden ascendente es decir de mayor a menor para columnas numéricas; en order alfabético para columnas alfanuméricas según el código ASCII, y de la fecha más antigua a las más reciente si la columna es de tipo fecha. Si se indica DESC la columna se ordena en forma inversa a la anterior

Condiciones de búsqueda en el Where

Se utiliza si solo se quiere seleccionar una parte de las filas de una tabla, que cumplan con la condición especificada en esta cláusula.

Condiciones de comparación o test de comparación (=,<>,!0,>,>,>=,>=)
Calcula y compara los valores de dos expresiones por cada fila de datos como puede ser un campo con una constante o expresiones mas complejas.

```
select *
from facturas
where fecha < '10/07/2008'
```

Condición de búsqueda (o test) de correspondencia con un patrón de búsqueda
Comprueba si el valor de un dato de una columna se ajusta a un patrón especificado, se utiliza la palaba reservada **LIKE.**

- % : reemplaza a uno o más caracteres.
- ▪ _ (guion bajo): reemplaza a un solo carácter.
- ▪ []: reemplaza a cualquier carácter que se encuentre entre el rango [a-f] o bien en el conjunto [abcdef].
- ▪ ^ : reemplaza a cualquier carácter que NO se encuentre entre el rango [a-f] o bien en el conjunto [abcdef].

```
Select *
```

```
from articulos
Where descripcion like 'L%'
```

Condición de búsqueda (o test) de rango
Se utiliza el operador **BETWEEN**, este verifica si el valor de un a columna se encuentra entre dos valores devueltos por dos expresiones. Implica el uso de tres expresiones de SQL. La primera define el valor a comprobar, la segunda y la tercera definen los extremos del rango.

```
Select *
from articulos
Where pre_unitario between 50 and 100
```

Condición de búsqueda (o test) de pertenecía a un conjunto
Se utiliza el operador **IN** para verificar si el valor de una columna se encuentra entre la lista de valores.

```
Select *
from clientes
Where cod_cliente in (1,3,7,8,12)
```

Condición de búsqueda (o test) de valor nulo
Se utiliza el operador **NULL** para verificar si el valor de una columna es nulo o no.

```
Select *
from articulos
Where observaciones is null
```

Operador NOT
Se puede utilizar para seleccionar filas donde la condición de búsqueda sea falsa.

```
Select *
from articulos
Where pre_unitario not between 10 and 100
```

Condiciones de busqueda compuestas
Utilizando las reglas de la lógica, se pueden combinar estas condiciones de búsqueda simples para formas consultas complejas utilizando los operadores **OR** y **AND**.

- Operador OR: Se utiliza para combinar dos condiciones de búsqueda cuando una o la otra o ambas deben ser ciertas.
- Operador AND: Se utiliza para combinar dos condiciones cuando es necesario que ambas sean ciertas simultáneamente.

Combinación de tablas

Es el proceso en el cual se forman duplas de filas haciendo coincidir los contenidos de las columnas relacionadas se denomina componer (joining) las tablas. La tabla resultante se denomina composición.

Joing implícito

```
select ape_vendedor+' '+nom_vendedor Vendedor, barrio
from vendedores, barrios
where vendedores.cod_barrio=barrios.cod_barrio
```

Joing Explicito

- **INNER JOIN:** selecciona valores que tienen coincidencias en ambas tablas. Se utiliza de forma abreviada solo "**JOIN**" y se utiliza para obtener información de dos tablas y combinar dicha información. Solo se mostrarán los registros donde coincidan los valores de sus claves primarias y foráneas respectivas, es decir no se mostrarán los registros donde no haya conexión entre tabas.

  SELECT campos

  FROM tabla1

  JOIN tabla2

  ON condicion de combinacion;

- **LEFT JOIN:** Combinación interna que encuentra registros de la primera tabla que se coincidan en la segunda tabla y si un valor de la primera tabla no se coincide en la segunda no aparecerá. Se emplea la combinación externa para mostrar los registros de la tabla de la izquierda. Si no encuentra coincidencia con la tabla de la derecha, el registro devuelto será los de la tabla de la derecha seteados a NULL.

- **RIGHT JOIN:** Una combinación externa derecha opera del mismo modo, pero la tabla de la derecha es la localiza los registros en la tabla de la izquierda,

- **COSS JOIN:** Una combinación cruzada retoña todas las combinaciones de ambas tablas y no se incluye una condición de enlace. Este tipo de join no incluye una condición de enlace. Se genera un producto cartesiano en el que el numero de filas del resulta es el resultado de la multiplicación de la primera tabla por las filas de la segunda.

- **FULL JOIN:** retorna todos los registros de ambas tablas. Si un registro de la tabla de la izquierda no encuentra coincidencias en la tabla de la derecha devuelve los valores de la tabla de la derecha seteados a NULL. Y si se da el proceso inverso los campos de la tabla izquierda aparecen conteniendo "null".