
SQL Server

Ailyn Rebollar Pérez

SQL (Structured Query Language)

- ▶ Lenguaje que permite ***definir estructuras*** para ***almacenar y modificar datos*** así como ***tener restricciones*** específicas para la seguridad de los mismos. También permite ***hacer consultas*** a una base de datos.

Historia

- ❑ En 1974 IBM desarrolló la primera versión de SQL que se llamaba Sequel.
- ❑ En 1986 ISO y ANSI publicaron un estándar para SQL llamado SQL-86.
- ❑ Estándar SQL-1999: Se agregaron **expresiones regulares**, consultas recursivas, **triggers** y características orientadas a objetos.

Historia

- ❑ Estándar SQL-2003: Se introducen características de XML, se agrega la sentencia sequence y columnas autonuméricas.
- ❑ Estándar SQL-2008: Se introduce la cláusula **ORDER BY**, disparadores del tipo INSTEAD OF y sententecia TRUNCATE.
- ❑ Estándar SQL-2011: Se incluyen datos temporales y mejoras de la cláusula FETCH.

Historia

- ❑ Estándar SQL-2016: Permite búsquedas de patrones, compatibilidad con ficheros JSON.
- ❑ SQL se considera el primer lenguaje establecido como el estándar para bases de datos relacionales.

Partes de SQL

► **DDL (*Data Definition Language*):** Nos permite definir esquemas, relaciones/tablas, modificar y eliminarlas por medio de comandos:

- **CREATE** : Sirve para crear.
- **DROP**: Sirve para eliminar.
- **ALTER**: Sirve para modificar.

Partes de SQL

► ***DML (Data Manipulation Language)***: Nos permite hacer consultas a la base de datos, insertar tuplas, borrarlas, eliminarlas y modificarlas por medio de comandos:

- ***INSERT***: Sirve para insertar/agregar tuplas a una tabla.
- ***DELETE***: Sirve para eliminar/borrar tuplas.
- ***UPDATE***: Sirve para actualizar tuplas.
- ***SELECT***: Sirve para realizar consultas a la BD.

Partes de SQL



Integrity: Nos permite tener restricciones que deben cumplir los datos almacenados en una base de datos además de poder actualizar los mismos sin dañarlos.

- **PRIMARY KEY:** Indica que un atributo es llave primaria.
- **FOREIGN KEY:** Indica que un atributo es llave foránea.
- **NOT NULL:** Indica que un atributo no puede ser nulo.
- **UNIQUE:** Indica que un atributo es único.

Partes de SQL

- ▶ **Definición de Vistas:** Nos permite definir vistas por medio de comandos del DDL, sin embargo, aquí las vistas son tablas donde mostramos sólo cierta información o ciertos atributos.
- ▶ **Control de Transacciones:** Por medio de comandos podemos tener medidas para el inicio y final de cada transacción y mantener la base en un estado consistente, un ejemplo es usando **COMMIT** y **ROLLBACK**.

Partes de SQL

- ▶ **SQL incrustado/incorporado y SQL dinámico:** Permite definir cómo las sentencias SQL pueden incorporarse o interactuar dentro de lenguajes de programación de propósito general como C, Java, C++.
- ▶ **Autorización:** Permite dar accesos específicos a las tablas, vistas, esquemas, etc a través de comandos DDL:
 - **GRANT:** Concede/otorga permisos.
 - **REVOKE:** Quita permisos.
 - **DENY:** Niega permisos.

SQL DDL

El DDL permite definir la especificación de tablas y de información respecto a éstas como:

- ❑ El esquema de una tabla.
- ❑ Los tipos de valores asociados a cada atributo.
- ❑ Restricciones de integridad.
- ❑ La seguridad y autorización de información para cada tabla.
- ❑ Definición de índices en las tablas.

Esquema (SQL Server)



“Es un contenedor con nombre para los objetos de una base de datos” [1] y éste nos permite organizar estos elementos que pueden ser tablas, vistas, procedimientos almacenados, triggers, etc. para tener un mejor control y administración de permisos o autorizaciones a ciertos usuarios.

➤ **Sintaxis:**

```
CREATE SCHEMA nombre_esquema [autorización usuario];
```

Esquema (SQL Server)

➤ Características:

- ❑ Al crear una tabla sin esquema, por default se crea en un esquema público (en el caso de SQL Server dbo)
- ❑ Se puede **crear** (CREATE), **modificar** (ALTER) y **eliminar** (DROP) un esquema.
- ❑ Se puede asignar permisos al momento de crear un esquema con **GRANT**, **DENY** y **REVOKE**. (Esto es opcional)
- ❑ **ALTER** se usa para cambiar/transferir una tabla de un esquema a otro.

Creación de un Esquema



```
CREATE SCHEMA VENTAS;  
CREATE SCHEMA BANCOS;  
CREATE SCHEMA "Alumnos";  
CREATE SCHEMA "Academicos";
```



```
CREATE SCHEMA Alumnos AUTHORIZATION ailyn;  
CREATE SCHEMA Ventas AUTHORIZATION gerardo;  
CREATE SCHEMA Bancos AUTHORIZATION user30;
```

Modificación de un Esquema



ALTER SCHEMA ALUMNOS TRANSFER dbo.Alumnos;

ALTER SCHEMA PLANTAS TRANSFER ANIMALES.Tipos;

Eliminación de un Esquema




DROP SCHEMA ALUMNOS;


DROP SCHEMA PLANTAS;

DROP SCHEMA Vehiculos;

Tipos de datos más comunes (SQL Server)



int: Se usa para guardar números enteros y su rango es de -2, 147, 483, 648 a 2, 147, 483, 647.




bigint: Se usa para guardar números enteros y su rango es de -9, 223, 372, 036, 854, 775, 808 a 9, 223, 372, 036, 854, 775, 807.



int: Se usa para guardar números enteros y su rango es de -32, 768 a 32, 767.

Tipos de datos más comunes (SQL Server)



decimal y numeric(s,p): Se usa para guardar números con punto decimal pero para usarlos hay que especificar la precisión y escala.

- ❑ Precisión (p) : Número de dígitos después del punto.
- ❑ Escala (s): Número de dígitos en total para representar el número.

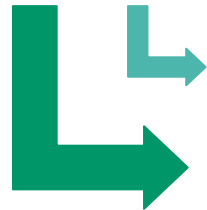
Tipos de datos más comunes (SQL Server)

► **Ejemplo:** Supongamos que queremos almacenar en la col1 el **123** en **formato decimal (5,2)** y en col2 el **12345.12** en **formato numeric(10,5)**.

Entonces estos se verían de la forma:

● col1 decimal (5,2)

123.00

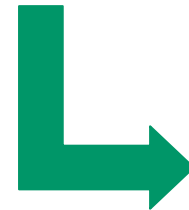


precisión(2)

escala(5)

● col2 numeric(10,5)

12345.12000



precisión(5)

escala(10)

Tipos de datos más comunes (SQL Server)

► **money:** Guarda números pero con formato de dinero que se escoja puede ser en pesos, dólares, etc.

Además su rango va de -922, 337, 203, 685, 447.5808 a 922, 337, 203, 685, 447.5807.

Soporta de 2 a 4 decimales

Tipos de datos más comunes (SQL Server)

▶ **char/character(n):** Se usa para guardar cadenas o string de tamaño exacto n si se da el parámetro o si no se especifica almacena de 1 a 8, 000 caracteres.

▶ **varchar(n):** Se usa para guardar cadenas o strings de máxima longitud n si se da el parámetro, sino se almacena máximo $2^{31} - 1$ caracteres.

▶ **text:** Guarda string al igual que varchar de a lo más $2^{31} - 1$ pero se recomienda usar éste tipo de dato para guardar comentarios, o campos que no sabemos el tamaño.

Tipos de datos más comunes (SQL Server)

► **date:** Almacena fechas y el formato por default es YYYY-MM-DD donde YYYY es para representar el año, MM el mes y DD el día, sin embargo, notemos que si se almacena así es en enteros.

También podemos almacenar las fechas como varchar, donde el nombre del mes salga completo o abreviado.

Asimismo, podemos cambiar el separador que es un - por una /.

Tipos de datos más comunes (SQL Server)

► **time:** Almacena la hora en forma de enteros con el formato hh-mm-ss donde hh representan las horas, mm los minutos y ss los segundos.

También podemos modificar el formato para que se aparezca el **AM** y **PM**.

Y podemos cambiar el separador que es - a :

Tipos de datos más comunes (SQL Server)

- ▶ **datetime**: Almacena la fecha y hora en forma de enteros con el formato YYYY-MM-DD espacio hh-mm-ss donde YYYY representa el año, MM el mes, DD el día, hh las horas, mm los minutos y ss los segundos.

Podemos cambiar de igual forma los separadores y los formatos, como ya se vio en ***date*** y ***time***.

Tipos de datos más comunes (SQL Server)

- ▶ **datetime**: Almacena la fecha y hora en forma de enteros con el formato YYYY-MM-DD espacio hh-mm-ss donde YYYY representa el año, MM el mes, DD el día, hh las horas, mm los minutos y ss los segundos.

Podemos cambiar de igual forma los separadores y los formatos, como ya se vio en ***date*** y ***time***.

Creación de una Base de Datos (SQL Server)



```
CREATE DATABASE nombre_bd;
```

```
CREATE DATABASE facultad_ciencias;
```

```
CREATE DATABASE multas;
```

```
CREATE DATABASE vehiculos;
```

Creación de tablas (SQL Server)

- ▶ `CREATE TABLE` nombre_tabla(
nombre_atributo₁ tipo restricción,
atributo₂ tipo restricción,
....
atributo_n tipo restricción);
- ▶ `CREATE TABLE` schema_nombre.nombre_tabla(
atributo₁ tipo restricción,
....
atributo_n tipo restricción);

Creación de tablas (SQL Server)

- ▶ `CREATE TABLE` alumnos(
num_cuenta int PRIMARY KEY,
nombre varchar(30) NOT NULL,
....
f_nac date);
- ▶ `CREATE TABLE` computacion.alumnos(
num_cuenta int PRIMARY KEY,
....
f_nac date);

Bibliografía

- [1] [Definición de Esquema SQL Server](#), consultado el día 10 de Septiembre de 2019.
- [2] [Creación de un esquema SQL Server](#), consultado el día 10 de Septiembre de 2019.
- [3] [Tipos de Datos en SQL Server](#), consultado el día 10 de Septiembre de 2019.
- [4] [Creación de una Tabla en SQL Server](#), consultado el día 10 de Septiembre de 2019.
- [5] Silberschatz Abraham, *Database System Concepts*, McGraw-Hill, 6ta edición, 2011.