

El Lenguaje SQL

Autor

Martín Vladimir Alonso Sierra Galvis

Gestión de Datos, Maestría en Ciencia de Datos
Pontificia Universidad Javeriana Cali

Versión 2.0
Santiago de Cali, febrero de 2023

Tabla de contenido

SQL	2
¿Qué es SQL?	2
¿Cómo comenzar a usar SQL?	3
Sentencias SQL	5
La sentencia SELECT	5
Otras sentencias	6
INSERT	6
UPDATE	7
DELETE	8
Referencias	9

SQL

¿Qué es SQL?

El Lenguaje de Consulta Estructurado, SQL por sus siglas en inglés, es un lenguaje estándar, enfocado al manejo de bases de datos relacionales. Con este lenguaje podemos gestionar la base de datos, realizando acciones como crear usuarios, crear conexiones o crear tablas. Además podemos interactuar con los datos creando nuevos registros, actualizando o eliminando registros existentes y consultando determinados registros que necesitemos. Muchas de las aplicaciones que usamos a diario están soportadas por grandes y complejas bases de datos relacionales y utilizan SQL para procesar los datos y mostrarnos información de la mejor manera posible.

El lenguaje SQL cumple con las siguientes características:

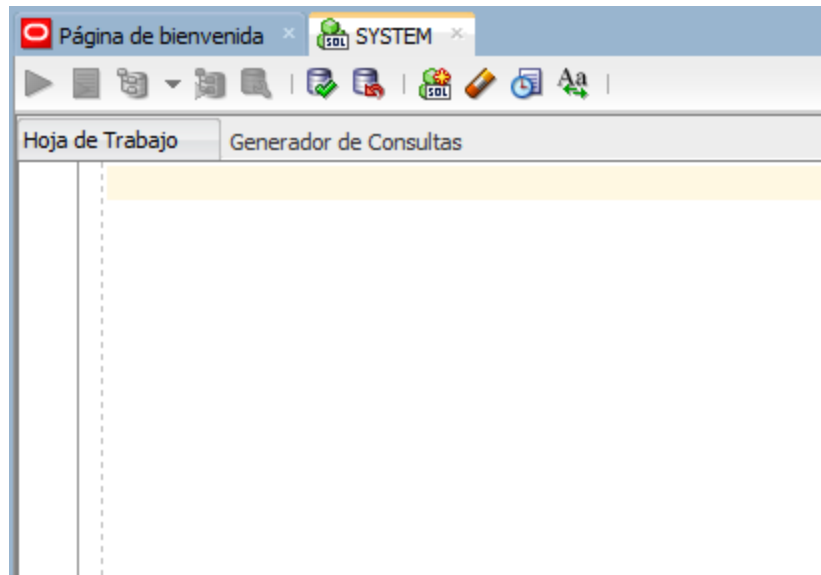
1. Es **unificado**, ya que lo utilizan todo tipo de usuarios, desde Administradores de Bases de Datos, pasando por Ingenieros de Software, hasta Científicos de Datos.
2. Es **no procedimental**, es decir, los usuarios especifican **qué** quieren, a diferencia de otros lenguajes donde se debe especificar **cómo** conseguirlo. Tal es el caso de los lenguajes de programación.
3. Es **relacionalmente completo** pues permite que los usuarios realicen todo tipo de consultas de datos.

Es importante que tengamos en cuenta que para poder utilizar SQL es necesario tener un software intermediario, el Sistema Gestor de Bases de Datos Relacionales. Dependiendo del sistema gestor que usemos, las sentencias SQL pueden tener algunas ligeras variaciones. En esta unidad nos enfocaremos en Oracle Database 21c Express Edition, por lo que las sentencias que encontraremos en este documento son propias de dicha versión del gestor. En este punto es importante haber revisado la lectura **Instalación de Herramientas**, donde encontrarás los pasos para instalar tanto Oracle Database como SQL Developer, su cliente gráfico.

¿Cómo comenzar a usar SQL?

Existen dos herramientas principales que nos permiten interactuar con Oracle Database mediante la ejecución de SQL. La primera es el Símbolo de Sistema, también llamado cmd o línea de comandos. Es la opción más utilizada por profesionales que se encuentran a cargo de la base de datos como, por ejemplo, los Administradores de Bases de Datos. Esto porque, muchas veces, las bases de datos se encuentran en servidores que no cuentan con una interfaz gráfica sino que toda la interacción se realiza por medio de una terminal. El problema con la línea de comandos es que no es una herramienta muy amigable para aquellas personas que no se encuentran familiarizadas con la base de datos. Es aquí donde entra en escena la segunda herramienta: SQL Developer, específicamente, su módulo Hoja de Trabajo de SQL, donde podremos escribir cualquier tipo de sentencia SQL. Como se mencionó en anteriores lecturas, esta es la herramienta recomendada para comenzar a trabajar con la base de datos, y es la que usaremos en esta unidad para mostrar los ejemplos. A modo de conocimiento general, existe una tercera forma de ejecutar SQL, los lenguajes de programación, mismos que son la herramienta predilecta de Ingenieros de Sistemas e Ingenieros de Software por igual. Aunque en esta unidad no tendremos acercamiento a ningún lenguaje de programación, a lo largo del curso usaremos uno en particular para realizar ciertas tareas de Gestión de Datos en Ciencia de Datos: el lenguaje de programación Python.

Volviendo a SQL Developer, cuando creamos una conexión a la base de datos, por ejemplo, al usuario SYSTEM, al momento de ingresar a dicha conexión se despliega automáticamente una pestaña con un espacio para escribir, como se muestra a continuación. Esta es la Hoja de Trabajo de SQL.



La anterior imagen nos muestra que pestaña siempre va a tener el mismo nombre de la conexión. Si por alguna razón esta no aparece al ingresar a la conexión, o la cerramos por equivocación, la podemos abrir dando click en el botón Hoja de Trabajo de SQL, que se encuentra en la barra inmediatamente superior. El botón lo identificamos por el siguiente ícono.



Sentencias SQL

La sentencia SELECT

Una **sentencia SQL** no es otra cosa que un comando del lenguaje SQL que usaremos para realizar determinada acción sobre la base de datos o sobre los datos almacenados en ella. Las sentencias SQL se dividen en dos categorías principales: DDL o Data Definition Language; y DML o Data Manipulation Language. En nuestro caso, para realizar Gestión de Datos en el ámbito de la Ciencia de Datos, nos van a interesar las sentencias pertenecientes a la segunda categoría, en especial, la sentencia **SELECT**.

La sentencia SELECT es la sentencia más utilizada de todas y, en la Gestión de Datos para la Ciencia de Datos, es la más importante. Con ella le indicamos al Sistema Gestor de la Base de Datos Relacional cuáles datos queremos recuperar o extraer de la base de datos. Su sintaxis básica es la siguiente.

SELECT * **FROM** [Tabla];

En la anterior expresión, SELECT es la cláusula con la que indicamos la intención de seleccionar datos. Por su parte, la cláusula FROM hace referencia a la tabla de la que vamos a extraer los datos. El nombre de dicha tabla iría donde se encuentra la expresión [Tabla]. El * significa que queremos obtener los datos de todas las columnas de la tabla.

Pongamos como ejemplo la base de datos de libros. Si de la tabla Libros queremos seleccionar todos los registros y, de esos registros, los datos de todas las columnas de la tabla, entonces la sentencia SELECT sería la siguiente.

SELECT * **FROM** Libros;

Ahora bien, si queremos obtener los datos de determinadas columnas, entonces debemos especificar cuáles son dichas columnas. Por ejemplo, si de la tabla Libros necesitamos únicamente los datos de las columnas nombre y id_editorial, la sentencia quedaría así.

SELECT nombre, id_editorial **FROM** Libros;

Finalmente, se puede presentar el caso en que deseemos seleccionar los datos de algunos registros específicos. Para ello podemos utilizar la cláusula WHERE, encargada de realizar verificación de valores en columnas. La sintaxis sería la siguiente.

SELECT nombre **FROM** Libros **WHERE** id_editorial=2;

Como resultado, esta sentencia nos devuelve los datos de la columna nombre de todos aquellos registros cuyo valor en la columna id_editorial sea igual a 2. Estas estructuras de la sentencia SELECT que hemos visto hasta el momento se utilizan de la misma forma para recuperar información de todas las tablas de la base de datos. Lo único que cambiaría serían los nombres de las tablas, las columnas y los valores a verificar.

Otras sentencias

Las sentencias que veremos en esta sección son más utilizadas por Administradores de Datos o Ingenieros de Software debido a que están más enfocadas a roles con propiedad sobre los datos. Aun así, no está demás que aprendamos un poco sobre ellas por si en algún momento las llegamos a necesitar.

INSERT

Con la sentencia INSERT podemos ingresar nuevos registros en la base de datos. Es muy común utilizarla en la etapa de creación de la base de datos para poblar las tablas con datos. Si en algún momento hemos llenado un formulario en una aplicación, ya sea web o móvil, muy seguramente INSERT está ejecutándose en el servidor para almacenar nuestros datos. Como Científicos de Datos, es muy raro que utilicemos esta sentencia directamente, pues, como veremos en futuras unidades, existen herramientas que la ejecutan automáticamente. La sintaxis de INSERT es.

INSERT INTO [Tabla]([columna], [columna], [columna], ...) **VALUES** ([valor], [valor], [valor], ...);

La cláusula INSERT INTO le indica al gestor que vamos a ingresar un nuevo registro a la tabla definida en [Tabla]. En los paréntesis que siguen a esta definición debemos especificar las columnas en las que queremos ingresar datos. Por último, la cláusula VALUES ayuda a establecer cuáles son los datos que se van a ingresar. ¡Ojo! Los datos que coloquemos dentro de los paréntesis deben corresponder con el orden en como se especificaron las columnas. Veamos como quedaría la sentencia si deseamos ingresar un nuevo registro en la tabla Libros.

```
INSERT INTO Libros(id_libro, nombre, paginas, fecha_publicacion,  
id_editorial) VALUES(10, 'El Problema de los Tres Cuerpos', 400,  
01/01/2003, 3);
```

Cabe recordar que es importante que respetemos los tipos de datos de cada columna al momento de ingresar nuevos registros a la tabla.

UPDATE

Esta sentencia permite la actualización de datos en una tabla y se usa normalmente para corregir errores que se hayan cometido al momento de ingresar datos. Como Científicos de Datos, tendremos un contacto casi nulo con UPDATE puesto que, si en algún momento necesitamos hacer alguna corrección sobre los datos, esta la haremos con datos que ya se encuentran fuera de la base de datos, proceso que se conoce como limpieza y preparación de datos, tema que veremos en próximas unidades. La sintaxis básica es.

```
UPDATE [Tabla] SET [columna]=[valor], [columna]=[valor], ...;
```

La cláusula UPDATE establece que se va a actualizar la tabla especificada en [Tabla]. Seguidamente, la cláusula SET sirve para indicar las columnas con sus respectivos valores nuevos. Si queremos actualizar todos los valores de la columna nombre, para todos los registros, en la tabla Libros, la sentencia sería.

```
UPDATE Libros SET nombre='El Bosque Oscuro';
```

Ahora bien, si queremos actualizar solamente el valor de la columna nombre de un registro determinado, debemos aplicar la cláusula WHERE así.

UPDATE Libros **SET** nombre='El Bosque Oscuro' **WHERE** id_libro=10;

La anterior sentencia actualiza el valor de la columna nombre a 'El Bosque Oscuro' en el registro cuyo id_libro tenga el valor 10.

DELETE

La sentencia DELETE es la sentencia de más cuidado en la categoría DML. Ningún profesional quisiera pasar por el problema de borrar datos sensibles de una base de datos. Lo bueno es que, al igual que ocurre con UPDATE, al ser DELETE una sentencia que modifica datos existentes en una base de datos, la probabilidad de que la usemos es prácticamente cero. Si necesitamos eliminar algún dato, lo haremos en el proceso de limpieza y preparación de datos. Pero si por algún motivo necesitamos eliminar registros de una tabla, la sintaxis es.

DELETE FROM [Tabla];

Las cláusulas DELETE y FROM indican al gestor que se van a borrar **todos** los registros de la tabla especificada en [Tabla]. Normalmente esta acción es indeseada; siempre se va a preferir eliminar registros específicos. Para ello utilizamos la cláusula WHERE.

DELETE FROM [Tabla] **WHERE** [columna]=[valor];

Por ejemplo, si quisiéramos eliminar el registro que ingresamos a la base de datos en la sección INSERT, deberíamos utilizar la sentencia DELETE de la siguiente manera.

DELETE FROM Libros **WHERE** id_libro=10;

Para finalizar, y como dato curioso, en la actualidad las empresas y organizaciones no borran los datos de sus bases de datos debido a que estos se han convertido en un bien muy preciado. Precisamente, gracias a los datos almacenados por años, los Analistas de Datos y Científicos de Datos pueden realizar labores de extracción para aplicar algoritmos de analítica o Machine Learning que permitan visualizar tendencias o mejorar procesos, acciones que pueden terminar traduciéndose en ganancias.

Referencias

1. Bradford, L. *What is SQL?* Marzo de 2020. [En línea]: disponible en <https://learntocodewith.me/posts/sql-guide/#what-is-sql>. [Accedido en 2020].
2. Curry, C. *Oracle SQL All-in-One Quick Start Tutorial Series*. Enero de 2020. [En línea]: disponible en <https://www.youtube.com/watch?v=1o0c-zD3iFU&t=1625s>. [Accedido en 2020].
3. Vegas, J. *Introducción al SQL*. Abril de 1998. Departamento de Informática. Universidad de Valladolid. [En línea]: disponible en <https://www.infor.uva.es/~jvegas/cursos/bd/sqlplus/sqlplus.html>. [Accedido en 2023].