

1. Lenguaje SQL

El estándar actual de 2016, compatible con ficheros JSON para la consola de la línea de comandos interactiva SQL CLI, **no distingue entre mayúsculas y minúsculas (case insensitive)** a la hora de escribir sus sentencias. Establece una interfaz de programación de aplicaciones (API, application programming interface) que han de utilizar nuestros programas para lanzar sentencias SQL de una forma normalizada a través de esta serie de funciones bien definidas en el estándar. De manera que los fabricantes de SGBDR pueden crear controladores que soporten el API de SQL y/o del controlador del lenguaje de programación; por ejemplo, JDBC (*Java Database Connectivity*) u ODBC (*Open Database Connectivity*), entre otros que vayamos a utilizar para embeber nuestras sentencias SQL.

	JDBC (Java Database Connectivity)	ODBC (Open Database Connectivity)
Denominación	Programa de conectividad de BBDD desde Java.	Programa de conectividad abierta de BBDD.
¿Cuándo se creó?	En 1996, por la empresa Sun Microsystems que fue adquirida por Oracle en 2009 y que se encarga de ir actualizándolo.	En 1992, por la empresa Microsoft, que se encarga de ir actualizándolo.
¿Qué es?	Define una API estándar solo para el lenguaje Java que permite definir cómo un usuario puede acceder a sus datos desde la base de datos.	Define una API estándar abierta a cualquier SGBDR que lo soporte.

Tabla 6.1. Comparativa entre JDBC y ODBC.

Microsoft ha creado otra tecnología más potente llamada OLE DB (Object Linking and Embedding for Databases), que permite trabajar con todo tipo de BBDD, no solo relacionales sino también con las de grafos, orientadas a objetos, nativas XML o NoSQL (que se muestran en el módulo de lenguaje de marcas). OLE DB está incluida dentro de los componentes de Microsoft para acceso a datos (MDAC, Microsoft Data Access Components).

A. Bloques de instrucciones de SQL

Las instrucciones, las sentencias o los comandos SQL se pueden agrupar dependiendo de su utilidad o funcionalidad en seis grandes bloques:

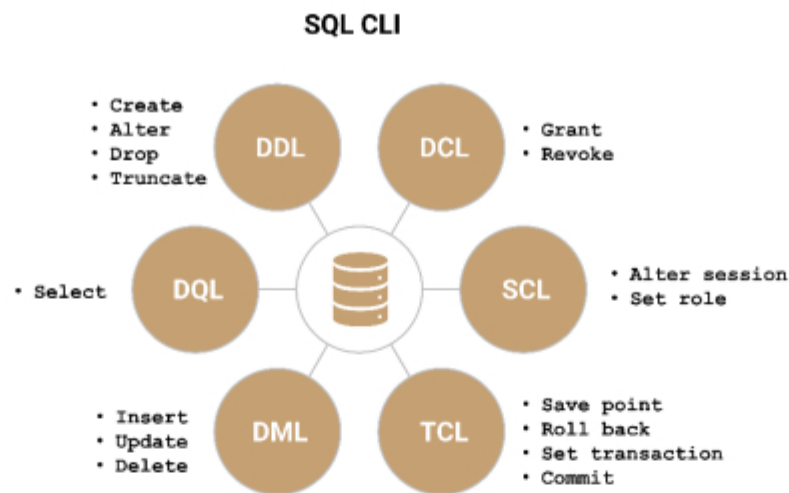


Fig. 6.1. Sublenguajes de SQL.

DDL (Data Definition Language)	El lenguaje de definición de datos agrupa las instrucciones CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT, SHOW y DESCRIBE.
DQL (Data Query Language)	El lenguaje de consultas de datos solo utiliza la instrucción SELECT. Se pueden ir anidando una dentro de otra, como veremos.
DML (Data Manipulation Language)	El lenguaje de manipulación de datos agrupa las instrucciones INSERT, UPDATE, DELETE, MERGE, CALL, EXPLAIN PLAN y LOCK TABLE.
DCL (Data Control Language)	El lenguaje de control de datos agrupa las instrucciones GRANT y REVOKE.
SCL (Session Control Language)	El lenguaje de control de sesiones gestiona las sesiones de trabajo con los diferentes roles de los usuarios. Agrupa las instrucciones ALTER SESSION y SET ROLE.
TCL (Transaction Control Language)	El lenguaje de control de transacciones agrupa las instrucciones SET TRANSACTION, COMMIT, ROLLBACK y SAVEPOINT.

Tabla 6.2. Detalle de sublenguajes de SQL.

En esta unidad nos centraremos en las instrucciones DDL y DQL. El resto se irán viendo en otras unidades.

B. Palabras reservadas de SQL

Como en cualquier lenguaje de programación, en nuestro caso DSL que es SQL, vamos a tener ciertos identificadores que ya están reservados por el propio lenguaje. Por tanto, estas palabras reservadas solo se podrán utilizar en el lugar correcto para construir las sentencias SQL, además de las que acabamos de mencionar en las categorías o bloques: DDL, DQL, DML, DCL y TCL. Las que más vamos a utilizar son:

ALL	AND	ANY	AVG	BETWEEN	COUNT	EXISTS
FROM	GROUP BY	HAVING	IN	IS NULL	LIKE	MAX
MIN	NOT	OR	ORDER BY	SUM	UNIQUE	WHERE

Tabla 6.3. Palabras reservadas más utilizadas en SQL.

No son las únicas, iremos descubriendo más cuando las vayamos necesitando. También hay que comentar que cada SGBDR puede que tenga algunas propias que no sigan el estándar SQL y, por tanto, solo funcionen con ese fabricante. Nosotros vamos a tratar de seguir el estándar internacional de SQL de 2016 para que esté soportado por cualquiera de los fabricantes de SGBDR que existen.

C. Tipos básicos de dominios de SQL

Para que SQL pueda integrarse con los lenguajes de programación, necesitamos que sus tipos básicos de datos soporten los mismos dominios que estos. Aunque puede que haya alguna variación de unos a otros, en SQL estándar están soportados los siguientes:

Para gestionar números enteros o con decimales:

<i>integer</i> o <i>int</i>	Permite trabajar con números enteros, será el HW o el SGBDR quien limitará su rango de valores. Existe tipos llamados <i>tinyint</i> , <i>smallint</i> y <i>mediumint</i> para rangos más pequeños, o tipos llamados <i>bigint</i> para rangos más grandes.
<i>numeric (p, d)</i>	Permite trabajar con números decimales. El usuario definirá: - con el parámetro <i>p</i> : el número de cifras enteras - con el parámetro <i>d</i> : el número de cifras decimales.
<i>real, double precision</i>	Permite trabajar con números en coma flotante con <i>real</i> de precisión simple o con <i>double</i> de doble precisión utilizando el estándar IEEE 754.
<i>float (n)</i>	Permite trabajar con números en coma flotante; en este caso es el usuario quien decide con el parámetro <i>n</i> el número de cifras para la precisión de este.

Tabla 6.4. Tipos básicos de datos en SQL para números enteros o con decimales.

Para gestionar fechas y/o horas:

datetime	Permite trabajar con valores que contengan partes de fecha y/o hora.
timestamp	Permite trabajar con valores que contengan partes de fecha y/o hora. La diferencia con datetime es que hace referencia al huso horario UTC, por lo que dependiendo del país en el que nos encontremos podemos hacer el cálculo adecuado para mostrar el valor según nuestro huso horario.
date	Permite trabajar con valores que contengan solo la fecha. Puede que alguna vez nos interese tener estos valores de las fechas y las horas por separado; por eso, disponemos de los siguientes tipos de campos.
time	Permite trabajar con valores que contengan solo la hora.
year	Permite trabajar con valores que contengan solo las 4 cifras del año dentro del rango de 1901 a 2155. Si queremos guardar el año actual lo podemos obtener de la función NOW().

Tabla 6.5. Tipos básicos de datos en SQL para fechas y/o horas.

Para gestionar valores alfanuméricos:

character o char(n)	Permite guardar una cadena de caracteres de longitud fija <i>n</i> .
character varying o varchar(n)	Permite guardar una cadena de caracteres de longitud variable, aunque el usuario con el valor <i>n</i> indicará su tamaño máximo.

Tabla 6.6. Tipos básicos de datos en SQL para valores alfanuméricos.

Para gestionar otros tipos de datos:

boolean o bit	Permite guardar valores ciertos (true) o falsos (false).
binary , varbinary , blob , text , enum y set	Para campos más grandes. Pueden servir para guardar imágenes u objetos multimedia, entre otros.
point	Nuevo tipo de datos especiales que sirve para almacenar valores geodésicos.
json	Nuevo tipo de datos especiales para almacenar vectores (arrays) de texto para almacenar el contenido de algún fichero JSON con el que queramos trabajar.

Tabla 6.7. Tipos básicos de datos en SQL para otros tipos de datos.

Para más información podemos consultar directamente la web del fabricante. Por ejemplo, MySQL en esta [página web](#).

