

## Resumen

---

El MySQL es un sistema gestor de bases de datos relacionales (SGBDR) corporativo de código abierto, muy utilizado para apoyar la gestión de los datos en aplicaciones web, a menudo junto con Apache y PHP.

Los tipos de datos *string* almacenan datos alfanuméricos en el conjunto de caracteres de la base de datos.

El MySQL proporciona los siguientes tipos de datos para gestionar datos alfanuméricos: CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM y SET.

El MySQL soporta todos los tipos de datos numéricos del SQL estándar: INTEGER (también abreviado por *INT*), SMALL, DECIMAL (también abreviado por *DEC* o *FIXED*) y NUMERIC. También soporta FLOAT, REAL, DOUBLE PRECISION (también llamado simplemente *DOUBLE*, o bien *REAL*), BIT y booleana.

El tipo de datos de que el MySQL dispone para almacenar datos que indiquen momentos temporales son: DATETIME, DATE, TIMESTAMP, TIME y YEAR.

La cláusula SELECT, del SQL, permite elegir columnas y valores (resultados de las expresiones) derivados de estas.

La cláusula FROM, del SQL, permite especificar las tablas en las que hay que ir a buscar las columnas o sobre las que se calcularán los valores resultantes de las expresiones.

La sentencia SELECT tiene más cláusulas aparte de las conocidas SELECT y FROM. Así, tiene una cláusula ORDER BY que permite ordenar el resultado de la consulta.

La cláusula WHERE, de SELECT, permite establecer los criterios de búsqueda sobre las filas generadas por la cláusula FROM.

La sintaxis básica de una cláusula SELECT es:

---

```
select <expresión / columna>, <expresión / columna>, ...  
from <tabla>, <tabla>, ...  
[Where <condición_de_búsqueda>]  
[Order by <expresión / columna>];
```

---

La opción DISTINCT acompañando la cláusula SELECT permite especificar que se quiere un único ejemplar para las filas repetidas.

La cláusula GROUP BY permite agrupar las filas resultantes de las cláusulas SELECT, FROM y WHERE según una o más de las columnas seleccionadas.

## BASES DE DATOS

Las cláusulas GROUP BY y HAVING añaden detrás la cláusula WHERE (si los hay) y antes de la cláusula ORDER BY (si los hay), por lo que ampliamos la sintaxis de la sentencia SELECT:

---

```
select [distinct] <expresión / columna>, <expresión / columna>, ...  
from <tabla>, <tabla>, ...  
[Where <condició_de_recerca>]  
[Group by <alias / columna>, <alias / columna>, ...]  
[Having <condició_sobre_grups>]  
[Order by <expresión / columna> [asc | desc], <expresión / columna> [asc | desc], ...];
```

---

Las funciones incorporadas proporcionadas por los SGBD se pueden utilizar dentro de expresiones y actúan con los valores de las columnas, variables o constantes en las cláusulas SELECT, WHERE y ORDER BY.

Las funciones y los operadores matemáticos principales proporcionados por el SGBD MySQL son ABS, Ceil, DIV, /, FLOOR, MOD, +, ROUND y TRUNCATE.

Las funciones y los operadores de cadenas de caracteres más importantes en el MySQL son CONCAT, FIND\_IN\_SET, INSTR, LCASE, LENGTH, LIKE, LOWER, LTrim, RTrim, substring, TRIM y UPPER.

Las funciones y los operadores de gestión de fechas más utilizadas en el MySQL son CURDATE, CURTIME, DATE\_FORMAT, DATE, DAYNAME, HOUR, MONTH, MONTHNAME, NOW, STR\_TO\_DATE, SYSDATE, SYSDATE y YEAR.

Los operadores de control de flujo que hay que conocer de MySQL están CASE, IF, IFNULL y NULLIFIES.

El MySQL permite la operación de unión:

---

```
sentència_select_sense_order_by  
union  
sentència_select_sense_order_by  
[Order by ...]
```

---

Los SGBD relacionales actuales (MySQL, Oracle, SQL Server, PostgreSQL, MS-Access ...) incorporan las combinaciones INNER, LEFT y RIGHT, con las que se pueden conseguir todos los tipos de combinaciones entre tablas.

La combinación LEFT JOIN permite combinar todas las filas de la tabla de la izquierda de JOIN con las filas con valores coincidentes de la tabla de la derecha, y proporciona valores nulos para las columnas de la tabla de la derecha cuando no hay filas con valores coincidentes.

La combinación RIGHT JOIN permite combinar todas las filas de la tabla de la derecha del JOIN con las filas con valores coincidentes de la tabla de la izquierda, y proporciona valores nulos para las columnas de la tabla de la izquierda cuando no hay filas con valores coincidentes.

## **BASES DE DATOS**

Una subconsulta es una sentencia `SELECT` que se incluye en la cláusula `where` de otra sentencia `SELECT`. La subconsulta se cierra entre paréntesis y no incluye el punto y coma finales.