

MANUAL RAPIDO SQL

LENGUAJE SQL

Existen tres tipos de Lenguajes SQL, cada uno con distintos tipos de instrucciones o comandos:

- los DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
- Los DCL que permiten llevar a cabo el control y la gestión del sistema (usuarios, permisos, ...)

1.0 Comandos DDL

CREATE Utilizado para crear nuevas tablas, campos e índices

DROP Empleado para eliminar tablas e índices

ALTER Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

1.1 Comandos DML

SELECT Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

INSERT Utilizado para cargar lotes de datos en la base de datos en una única operación.

UPDATE Utilizado para modificar los valores de los campos y registros especificados

DELETE Utilizado para eliminar registros de una tabla de una base de datos

1.1.2 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

FROM Utilizada para especificar la tabla de la cual se van a seleccionar los registros

WHERE Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar

GROUP BY Utilizada para separar los registros seleccionados en grupos específicos

HAVING Utilizada para expresar la condición que debe satisfacer cada grupo

ORDER BY Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

1.1.3 Operadores Lógicos

AND Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.

OR Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.

NOT Negación lógica. Devuelve el valor contrario de la expresión.

1.1.4 Operadores de Comparación

< Menor que

> Mayor que

<> Distinto de

<= Menor ó Igual que

>= Mayor ó Igual que

= Igual que

BETWEEN Utilizado para especificar un intervalo de valores.

LIKE Utilizado en la comparación de un modelo

In Utilizado para especificar registros de una base de datos

1.1.5 Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

- AVG Utilizada para calcular el promedio de los valores de un campo determinado
- COUNT Utilizada para devolver el número de registros de la selección
- SUM Utilizada para devolver la suma de todos los valores de un campo determinado
- MAX Utilizada para devolver el valor más alto de un campo especificado
- MIN Utilizada para devolver el valor más bajo de un campo especificado

1.1.6 Funciones aritméticas

- ABS(n): Devuelve el valor absoluto de "n".
- ROUND(m, n): Redondea el número "m" con el número de decimales indicado en "n", si no se indica "n" asume cero decimales.
- SQRT(n): Devuelve la raíz cuadrada del parámetro que se le pase.
- POWER(m, n): Devuelve la potencia de "m" elevada el exponente "n".

1.1.7 Funciones de cadenas

- LENGTH(c) o len(c): Devuelve la longitud (número de caracteres) de una cadena.
- LOWER(c): Devuelve la cadena "c" con todas las letras convertidas a minúsculas.
- UPPER(c): Devuelve la cadena "c" con todas las letras convertidas a mayúsculas.
- LTRIM(c): Elimina los espacios por la izquierda de la cadena "c".
- RTRIM(c): Elimina los espacios por la derecha de la cadena "c".
- REPLACE(c, b, s): Sustituye en la cadena "c" el valor buscado "b" por el valor indicado en "s".
- REPLICATE(c, n): Devuelve el valor de la cadena "c" el número de veces "n" indicado.
- LEFT(c, n): Devuelve "n" caracteres por la izquierda de la cadena "c".
- RIGHT(c, n): Devuelve "n" caracteres por la derecha de la cadena "c".
- SUBSTRING(c, m, n): Devuelve una sub-cadena obtenida de la cadena "c", a partir de la posición "m" y tomando "n" caracteres.
- CONCAT(c,m): Devuelve una cadena donde ha concatenado la cadena c y la cadena m.
- ASCII (cad): Devuelve el código ASCII de la cadena
- INSTR(cad1, cad2[,comienzo[,m]]): Función que busca un conjunto de caracteres dentro de una cadena. Nos devuelve la posición de cad2 en cad1 empezando a buscar en comienzo

1.1.8 Funciones de manejo de fechas

- YEAR(d): Devuelve el año correspondiente de la fecha "d".
- MONTH(d): Devuelve el mes de la fecha "d".
- DAY(d): Devuelve el día del mes de la fecha "d".
- DATEADD(f, n, d): Devuelve una fecha "n" periodos (días, meses años, según lo indicado) superior a la fecha "d". Si se le pasa un número "n"

negativo, devuelve una fecha "n" periodos inferior. De gran utilidad en consultas.

1.1.9 Funciones de conversión

Estas funciones **suelen ser específicas de cada gestor de datos**, ya que cada SGBDR utiliza nombres diferentes para los distintos tipos de datos (aunque existen similitudes se dan muchas diferencias).

Las funciones de conversión nos permiten cambiar valores de un tipo de datos a otro. Por ejemplo si tenemos una cadena y sabemos que contiene una fecha, podemos convertirla al tipo de datos fecha. Así, por ejemplo:

- En SQL Server tenemos funciones como CAST, CONVERT o PARSE, que en función de lo que especifiquemos en sus parámetros convertirán los datos en el tipo que le indiquemos. Ver funciones de conversión de SQL Server.
- En Oracle tenemos funciones como TO CHAR, TO DATE, TO NUMBER. Ver funciones de conversión de Oracle.
- En MySQL solucionamos la mayor parte de las conversiones con la función CAST. Ver conversiones en MySQL.

Algunos ejemplos simples de uso de funciones

Una vez vistas estas funciones básicas, vamos a poner ejemplos sencillos de la sintaxis de alguna de ellas:

```
SELECT ABS(100), ABS(-100), ROUND(10.12345, 2), SQRT(9)
SELECT UPPER(LTRIM(RTRIM(' PEPE '))), REPLICATE('AB', 20),
SUBSTRING('ABCDEFGHI', 3, 5)
SELECT MONTH(CAST('20060715' AS DATETIME)), DATEADD(dd, 10, GETDATE())
```

La función GETDATE() es específica de SQL Server, y nos devuelve la fecha y hora actuales del sistema.