

## LENGUAJE SQL CONSULTAS DQL

Estructura básica de una consulta:

SELECT

FROM

Estructura condicionada:

SELECT

FROM

WHERE

### 1.SELECT

Este comando permite obtener datos, mezclar datos de tablas diferentes, realizar cálculos sobre los datos y agrupar datos.

Sintaxis del comando SELECT:

-\*: selecciona todas las columnas

-DISTINCT: muestra una lista de elementos sin duplicados

### 2.FROM

Este comando indica de qué tabla seleccionamos los datos que obtenemos en el SELECT.

Lo normal en una consulta es que los datos se encuentren en varias tablas. Para utilizar datos de diferentes tablas necesitamos enlazarlas a través del FROM. Para ello utilizamos la partícula JOIN y para ser más precisos a la hora de conseguir los datos relacionamos las tablas a través de una condición indicada con la partícula ON. Ejemplo:

```
SELECT *  
FROM piezas  
JOIN existencias ON piezas.tipo=existencias.tipo;
```

También tenemos otras partículas como:

RIGHT JOIN: que saca todos los valores al lado derecho del JOIN independientemente de que los de la izquierda sean nulos

LEFT JOIN: que saca todos los valores al lado izquierdo del JOIN independientemente de que los de la derecha sean nulos

### 3.WHERE

Este comando es el que nos permite definir la condición que se debe cumplir para que se devuelvan las filas.

[ WHERE <search\_condition> ]

En el WHERE podemos utilizar varios comandos según lo que queremos sacar por pantalla:

-Operadores Aritméticos:

> : Mayor que

< : Menor que

>= : Mayor igual que

<= : Menor igual que

= : Igual

<> : Distinto

-Valores Lógicos:

AND : devuelve verdadero si las expresiones a su izquierda y derecha son ambas verdaderas.

OR : Devuelve verdadero si cualquiera de las dos expresiones a izquierda y derecha del OR, son verdaderas; pudiendo sólo una ser verdadera.

-BETWEEN : nos permite obtener datos que se encuentran entre un rango de valores. Ejemplo:

```
SELECT tipo,modelo,precio
FROM piezas
WHERE precio BETWEEN 3 AND 8;
```

-IN : permite obtener registros cuyos valores estén en una lista de valores. Ejemplo:

```
SELECT tipo,modelo,precio
FROM piezas
WHERE precio IN (3,5, 8);
```

-LIKE : permite obtener registros cuyo valor en un campo cumpla una condición textual. Puede tener estos símbolos: % una serie cualquiera de caracteres; \_ un carácter cualquiera. Ejemplo:

```
/* Selecciona nombres que empiecen por S */
SELECT nombre
FROM personas
WHERE nombre LIKE 'S%';
```

-IS NULL : devuelve verdadero si el valor que examina es nulo. Ejemplo:

```
SELECT nombre,apellidos  
FROM personas  
WHERE telefono IS NULL;
```

-Funciones:

REPLACE : Buscar el texto a buscar en un determinado texto y lo cambia por el indicado como texto de reemplazo.

LENGTH : calcula el número de caracteres de una cadena y puede comparar números de caracteres.

-Agrupaciones:

GROUP BY : se indica las columnas por las que se agrupa. la función de este apartado es crear un único registro por cada valor distinto en las columnas del grupo.

COUNT(\*) : Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna, por tanto solo nos saca una fila y una columna.

SUM : suma los valores de la expresión.

AVG: calcula la media aritmética sobre la expresión indicada.

-Condiciones con HAVING:

Se utiliza para restringir el resultado de una expresión agrupada. Ejemplo:

```
SELECT tipo,modelo, cantidad, SUM(Cantidad)  
FROM existencias  
GROUP BY tipo,modelo  
HAVING SUM(Cantidad)>500;
```

-Subconsultas:

Se trata de una técnica que permite utilizar el resultado de una tabla SELECT en otra consulta SELECT.

```
SELECT nombre_empleado, paga  
FROM empleados  
WHERE paga <(SELECT paga FROM empleados  
                WHERE nombre_empleado='Martina');
```

Subconsultas de varias filas usando instrucciones.

ALL: Compara con todos los registros de la consulta

IN: No usa comparador, ya que sirve para comprobar si un valor se encuentra en el resultado de la subconsulta.

