

LMD

Las sentencias que se utilizan en este lenguaje son:

- *SELECT: se utiliza para extraer información de una BD.*
- *INSERT: inserta uno o varios registros en una tabla.*
- *DELETE: borra registros de una tabla*
- *UPDATE: modifica registros de una tabla.*

De estas sentencias la más utilizada es select y es en la que más nos centraremos.

Sentencia SELECT

Sintaxis general

*SELECT [DISTINCT] listaColumnas
FROM listaTablas
[WHERE filtro];*

En el SELECT indicaremos el nombre de las columnas que queramos que aparezcan en el resultado de nuestra consulta. En el FROM indicaremos la tabla donde se encuentran las columnas.

Y si queremos que se cumpla una condición utilizaremos WHERE.

DISTINCT es para evitar la salida de tuplas(filas) duplicadas. Se considera tuplas duplicadas cuando el valor de todos los campos son iguales.

<i>Proveedor</i>	<i>Artículo</i>	<i>Precio</i>	
<i>María</i>	<i>Mesa</i>	<i>125</i>	<i>No son tuplas duplicadas porque los artículos tienen precios distintos</i>
<i>María</i>	<i>Mesa</i>	<i>250</i>	

Por ejemplo si queremos obtener las provincias de las que tenemos proveedores realizaremos la siguiente consulta:

Sin distinct	Con distinct
Select provincia from proveedores	Select distinct provincia from proveedores
Resultado: Alicante Alicante Alicante Murcia Castellón Castellón	Resultado: Alicante Murcia Castellón

Solo hay que poner el distinct cuando sea necesario, como en el caso del ejemplo anterior.

Ejemplos utilización select:

- Obtenemos el dni y el nombre de los profesores de nuestra BD

*Select dni, nombre
From profesores;*

- Si queremos obtener todos los datos de una tabla utilizamos el asterisco (*). Obtenemos todos los datos de la tabla profesores.

```
Select * from profesores;
```

- Si necesitamos que se cumpla alguna condición utilizamos el WHERE. Obtenemos el dni y el nombre de los profesores que son de Alicante

```
Select dni, nombre  
From profesores  
Where ciudad='Alicante';
```

En las condiciones se pueden utilizar los operadores lógico AND, OR y NOT y operadores de comparación <, >, >=, <=, =, <>(distinto). Los números van sin comillas, lo demás (texto, fecha) con comillas simples. Si tenemos campos donde su valor en algunos registros es nulo, podemos obtener los registros con ese campo nulo indicando en el WHERE is null. También podemos obtener los registros que tienen valor en el campo con is not null.

- Obtenemos el nombre de los artículos cuyo precio sea mayor que 50

```
Select nombre  
From artículo  
Where precio > 50;
```

- Obtenemos el nombre de los artículos cuyo precio sea mayor que 50 y menor que 100

*Select nombre
From artículo
Where precio > 50 and precio < 100;*

- *Obtenemos las piezas suministradas por el proveedor 4 y el 5*

*Select numpieza
from preciosum
where numvend=4 or numvend=5;*

- *Obtener el nombre de los clientes que no tienen teléfono.*

*Select nombre
From cliente
Where telefono is null*

- *Obtener el nombre de los clientes que tienen teléfono*

*Select nombre
From cliente
Where telefono is not null*

La salida de puede ordenar de forma ascendente o descendentes. Por defecto la ordenación es ascendente.

Ejemplo: Ordena el resultado por la columna precio de forma ascendente

Select cod_artículo, nombre, precio

*From artículo
Where precio > 50
Order by precio*

*Si quiero ordenar de forma descendente tengo que poner
order by precio desc*

*Para ordenar por más de una campo, estos van separados
por coma.*

Ejemplo:

*Select nombre, apellido, teléfono
from cliente
order by nombre, apellido*

*Ordenara por nombre y para los nombres que sean iguales
los ordena por apellidos.*

*Carla León
Ernesto Pérez
María López
María Pérez
Pedro Giménez*

*En temas posteriores veremos más sentencias que se
añadirán al select, hay que tener en cuenta que la sentencia
order by siempre va al final de todas las demás sentencias.*