

## T02 Consultas

### Select-from

Para realizar consultas sobre una base de datos vamos a utilizar la orden select de SQL. En este momento veremos la expresión mínima de la orden, formada por dos cláusulas, *select* y *from*, que obligatoriamente tendremos que especificar en cada consulta que realicemos.

Vamos a recuperar toda la información que se encuentra en la tabla profesores:

```
select * from profesores
```

dni	nombre	categoria	ingreso
21111222	EVA GOMEZ	TEU	1993-10-01
21222333	MANUEL PALOMAR	TEU	1989-06-16
21333444	RAFAEL ROMERO	ASO6	1992-06-16

Al especificar en la lista de columnas un asterisco le indicamos al SGBD que deseamos la información de todas las columnas definidas para la tabla profesores.

```
select categoria from profesores
```

categoria
TEU
TEU
ASO6

Ahora solo hemos obtenido la columna categoria desechando la salida de las otras columnas. Podemos pedir, eso sí, cuantas columnas queramos y estén definidas en la tabla.

```
select nombre, categoria from profesores
```

nombre	categoria
EVA GOMEZ	TEU
MANUEL PALOMAR	TEU
RAFAEL ROMERO	ASO6

Una opción de la que podemos hacer uso es la eliminación de duplicados en la salida usando el modificador `distinct`.

```
select distinct categoria from profesores
```

#### Contenidos

[1 Select-from](#)

[2 Where](#)

[3 Order by](#)

categoria
TEU
ASO6

## Where

Con la orden *select-from* obtenemos la información de las columnas requeridas de toda la tabla. Si únicamente queremos información de aquellas filas que cumplen una determinada condición utilizaremos la cláusula *where*.

```
select nombre
from profesores
where categoria = 'TEU'
```

nombre
EVA GOMEZ
MANUEL PALOMAR

En la construcción de tales condiciones podemos utilizar las conectivas lógicas AND, OR, y NOT, así como los paréntesis para alterar la evaluación de izquierda a derecha. También, los operadores de comparación >, <, >=, <=, <>, !=. Ante la duda, hay que consultar la tabla de precedencia de operadores, que en el caso de MySQL [se puede encontrar aquí](#): : no es lo mismo "A and B or C" que "A and (B or C)".

```
select nombre
from profesores
where categoria = 'TEU' or categoria = 'ASO6'
```

nombre
EVA GOMEZ
MANUEL PALOMAR
RAFAEL ROMERO

## Order by

Podemos ordenar la salida producida por nuestra orden select por valores ascendentes o descendentes de una columna en particular.

```
select creditos, descripcion
from asignaturas
order by creditos
```

creditos	descripcion
4.5	HISTORIA DE LA INFORMATICA
6.0	DISEÑO Y GESTION DE BASES DE DATOS
6.0	FUNDAMENTOS DE LAS BASES DE DATOS
6.0	PROGRAMACION CONCURRENTE
9.0	FUNDAMENTOS DE LA PROGRAMACION

El resultado anterior estaba ordenado ascendentemente. Se puede especificar `DESC` para hacer la ordenación de forma descendente (igual que se puede hacer `ASC` para ascendente, aunque no es necesario, como ya se ha visto)

```
select creditos, descripcion
from asignaturas
where creditos > 4.5
order by creditos DESC
```

creditos	descripcion
9.0	FUNDAMENTOS DE LA PROGRAMACION
6.0	DISEÑO Y GESTION DE BASES DE DATOS
6.0	FUNDAMENTOS DE LAS BASES DE DATOS
6.0	PROGRAMACION CONCURRENTE

Pueden aplicarse criterios más complejos de ordenación. El siguiente es un ejemplo de ordenación por dos atributos: a igualdad de los valores del primer atributo, el orden lo determinan los valores del segundo. Nótese que es posible combinar en el criterio de ordenación distintos tipos de datos.

```
select creditos, descripcion
from asignaturas
order by creditos, descripcion
```

creditos	descripcion
4.5	HISTORIA DE LA INFORMATICA
6.0	DISEÑO Y GESTION DE BASES DE DATOS
6.0	FUNDAMENTOS DE LAS BASES DE DATOS
6.0	PROGRAMACION CONCURRENTE
9.0	FUNDAMENTOS DE LA PROGRAMACION