

## Sentencia SELECT

## **SELECT [ DISTINCT ] listaColumnas**

**FROM** listaTablas

[ WHERE condición para filas]

[ GROUP BY listaColumnas por las que se quiere agrupar

[ **HAVING** condición para los grupos] ]

[ ORDER BY listaColumnas [ ASC | DESC ] ]

```
Inner join
select ...
from A join B on (A.x=B.y)

Outer join
select * from A left [outer] join B on (condición)
Obtiene todas las filas relacionadas de A y B, y todas las no relacionadas de A.
select * from A right [outer] join B on (condición)
Obtiene todas las filas relacionadas de A y B, y todas las no relacionadas de B.
```

expresión [NOT] IN (listaValores)

expresión [NOT] BETWEEN expresión AND expresión

Columna IS [NOT] NULL

```
expresión [NOT] LIKE 'cadena'
```

La cadena de caracteres cadena admite los comodines % (tanto por ciento) y \_ (subrayado)

Para unir SELECT
UNION

UNION UNION ALL

```
COUNT(*)

número de filas

COUNT([DISTINCT] expr) número de valores distintos en expr

SUM([DISTINCT] expr) suma de todos los valores en expr

AVG([DISTINCT] expr) promedio de todos los valores en expr

MIN( expr) el más pequeño de todos los valores en expr

MAX( expr) el mayor de todos los valores en expr
```

## Subconsultas

Podemos utilizar los operadores de comparación para preguntar si el valor de una determinada expresión es mayor, menor, igual, etc. que el resultado de la subconsulta, siempre y cuando ésta devuelva una única fila y una única columna, es decir, un valor escalar.

expr operador\_comparación (orden select)

En otros casos, las subconsultas, dependiendo del valor esperado y del tipo de comparación deseado se pueden utilizar de 3 formas:

- expr operador\_comparación {ALL|[ANY|SOME]} (orden select)
- expr [NOT] IN (orden select)
- [NOT] EXISTS (orden select)

	pack ( cod varchar(7)) CP (cod) CAj (cod)> articulo	ptienea (	pack varchat(/), articulo varchar(7)) CP (nack articulo)	CAj (pack) -> pack CAj (pack) -> pack	stock (	articulo varchar(7),	disponible number(11),	Próximamente', 24 horas', 3/4 días', 1/2	semanas'))	CAi (articulo) —> articulo		cesta (	articulo varchar(7),	usuario varchar(50),	recha date )  (P (articulo usuario)	CAj (articulo) —> articulo		pedido (	numPedido number(11),	usuario varchar(50),	fecha date )	CA (numeratio) —> nenario VNN			linped (	numPedido number(11),	articulo varchar(7).	importe number(9.2),	cantidad number(11))	CA (Innea, numPedido)	CAj (arnculo) —> arnculo vniv CAj (numPedido) —> pedido	VNN (precio)
BD TiendaOnLine	articulo ( cod varchar(7), nombre varchar(45), pvp number(7,2), marca varchar(15),	imagen blob, urlimagen varchar(100), especificaciones text)	CP (cod)  CAi (marca) —> marca	camara (	cod varchar(7), resolucion varchar(15),	sensor varchar(45),	tipo varchar(45),	objetivo varchar(15),	pantalla varchar(20),	Zoom varchar(40)) CP (cod)	CAj (cod)> articulo		tv (	cod varchar(7),	panel varchar(45),	resolucion varchar(15),	hdreadyfullhd varchar(6),	tdt number(1) ) CP (cod)	CAj (cod)> articulo		memoria (	tipo varchar(7),	CP (cod)	CAj (cod)> articulo		objetivo (	tipo varchar(15).	montura varchar(15),	focal varchar(10),	apertura varchar(10),	especiales varchar(35) )  CP (cod)	CAj (cod)> articulo
BD Tief	provincia ( codp varchar(2), nombre varchar(25) ) CP (codp)	codm varchar(4), pueblo varchar(50),	25	CAj (provincia)> provincia VNN (pueblo)	usuario (	ciliali varchar(30), nombre varchar(35).	apellidos varchar(55),	dni varchar(12), telefono varchar(15).	calle varchar(45),	calle2 varchar(45),	codpos varchar(5),	pueblo varchar(4),	provincia varchan(z), nacido date)	CP (email)	CAlt (dni)	CAj (pueblo, provincia) —>	Jocalidad VIVIN	VNN (nombre)	direnvio (	email varchar(50)	calle varchar(45),	calle2 varchar(45),	codpos varchar(5),	pueblo varchar(4),	CP (email)	CAj (pueblo, provincia) —>		CAj (email) —> usuario	marca (	marca varchar(15),	empresa varchar(60),	logo blob )  CP (marca)