

# Actividad 2.2. Consultas simples: SELECT, FROM, WHERE

## OCW VJ1220 Bases de datos

### Objetivo

Usar la sentencia SELECT para realizar consultas simples sobre una sola tabla con y sin restricciones.

### Qué hacer

Lee en el libro los apartados del 4.4 y 4.5 del **capítulo 4**. Los conceptos sobre la estructura básica de la sentencia SELECT que aparecen en la lectura, y que vas a trabajar en esta actividad, son:

- **en qué orden** se evalúan las cláusulas en una sentencia simple: SELECT, FROM, WHERE
- qué se especifica en la cláusula **FROM**
- qué se especifica en la cláusula **SELECT**
- qué **expresiones** se pueden incluir en la cláusula SELECT
- qué utilidad tiene la cláusula **WHERE**
- qué **expresiones** se pueden incluir en la cláusula WHERE

Puedes repasar los conceptos que has estudiado en el vídeo 3 titulado **Consultas en SQL: SELECT, FROM, WHERE**. Después realiza los ejercicios que encontrarás a continuación. Los ejercicios trabajan sobre la base de datos del apartado 4.2 del libro. Es importante que te familiarices con las tablas de esta base de datos y que entiendas el significado de los datos que almacenan.

### Ejercicios

Los ejercicios que hay a continuación trabajan sobre la base de datos del apartado 4.2 del libro.

1. Los nulos en **facturas.iva** y **facturas.dto** de la base de datos que se describe en el apartado 4.2 del libro debemos interpretarlos como el valor 0 (ojo: es un mal uso, aunque habitual, de los nulos). ¿Qué debemos escribir en el SELECT de la siguiente sentencia para ver todos los datos de la factura pero sin ver los nulos en **iva** y **dto**, sino ver ceros en su lugar?

```
SELECT ...  
FROM facturas;
```

2. Vamos a suponer que la tabla de **facturas** tiene 1000 filas y que la tabla de **clientes** tiene 130 filas. Fíjate que la columna **facturas.codcli** es clave ajena a la tabla **clientes**: mediante ella se hace referencia al cliente al que pertenece la factura. ¿Cuántas filas crees que devolverá la siguiente sentencia?

```
SELECT codcli
FROM facturas;
```

¿Cuántas filas crees que devolverá la siguiente sentencia?

```
SELECT DISTINCT codcli
FROM facturas;
```

¿Cuántas filas crees que devolverá la siguiente sentencia?

```
SELECT DISTINCT codfac
FROM facturas;
```

3. Escribe una sentencia **SELECT** que muestre los tipos de IVA que se han aplicado en las facturas. Por ejemplo, si hay facturas con 16% de IVA, en el resultado debe salir solamente una vez el valor 16 ya que es un tipo de IVA aplicado.
4. ¿A qué consulta de datos responde la siguiente sentencia?

```
SELECT codfac, linea, codart, precio, cant, COALESCE(dto,0) AS dto
FROM lineas_fac
ORDER BY codfac, linea;
```

¿Qué debemos añadir a la sentencia anterior para obtener también el importe de cada línea?

5. Mediante la palabra clave **AS** podemos renombrar columnas en el resultado de las consultas. ¿Qué nombre crees que es apropiado darle a la tercera columna de esta sentencia? Fíjate bien en la función que estamos usando.

```
SELECT codfac, fecha, EXTRACT(year FROM fecha) AS .....
FROM facturas;
```

6. Escribe una sentencia **SELECT** que obtenga como resultado los años en que hay facturas.



7. ¿Qué nombre crees que debemos darle a la tercera columna de la siguiente sentencia? Recuerda que en el libro aparecen los operadores y las funciones que podemos usar con las fechas.

```
SELECT codfac, CURRENT_DATE - fecha AS .....
FROM facturas
ORDER BY 2 DESC;
```

8. Fíjate en la estructura de la tabla de **provincias**:

Columna	Tipo	Modificadores
codpro	character varying(2)	not null
nombre	character varying(30)	not null

Sabiendo que el operador || concatena cadenas, escribe una sentencia SELECT que muestre un listado como el siguiente. Fíjate que aparece el nombre de cada columna seguido del código de la provincia entre paréntesis. Ten en cuenta que el nombre de las provincias debe aparecer con la inicial en mayúscula, independientemente de como se haya almacenado en la base de datos (por ejemplo, algunas provincias aparecen escritas en mayúsculas en la base de datos).

```
provincias
-----
Alava (01)
Albacete (02)
Alicante (03)
Almeria (04)
Asturias (33)
Avila (05)
Badajoz (06)
Balears (07)
...
```

9. ¿A qué consulta responde la siguiente sentencia?

```
SELECT *
FROM facturas
WHERE dto = 0 OR dto IS NULL;
```

¿Cómo podemos hacer la misma restricción utilizando la función COALESCE?

10. Escribe una sentencia SELECT que muestre los meses del año pasado en los que compraron algo los clientes con código 45, 54, 87 y 102. Sabemos que han comprado algo porque tienen alguna factura con fecha del año pasado.

11. Escribe una sentencia SELECT que muestre el nombre, la dirección y el código postal de aquellos clientes en los que el código postal empieza por 50. En el listado debe aparecer una columna llamada tipo en donde se muestre la palabra preferente junto a los clientes cuyo código postal tenga el tercer dígito mayor o igual que 5. Los clientes deben mostrarse ordenados por el nombre, con nombre y dirección con mayúscula sólo en la inicial de cada palabra.
12. Escribe una sentencia SELECT que muestre un listado de los artículos cuyo precio es inferior a 5 céntimos de euro. Muestra el precio original al doble de su valor. Ordena el resultado por el código del artículo.
13. Escribe una sentencia SELECT que muestre el código y la fecha de las facturas con un descuento del 25% y sin IVA (recuerda que en la base de datos de prácticas se considera que una factura no tiene iva si éste es cero o nulo). Ordena el resultado por la fecha de la factura.
14. Escribe una sentencia SELECT que muestre los nombres de las provincias cuya segunda letra sea una o. Estas provincias deben aparecer ordenadas alfabéticamente, con la inicial en mayúscula y el resto en minúsculas.
15. Escribe una sentencia SELECT que muestre un listado con el código y la fecha de las facturas del año pasado que pertenecen a clientes cuyo código está entre el 50 y el 80. Ordenar el resultado por la fecha descendentemente.