

SQLite - Select

ALUMNO: Bisutti German

DNI: 45928501

Data Manipulation Language, SELECT.

Para esta práctica utilizaremos la base de datos Ordenes.sqlite incluida en el trabajo.

Nos centraremos en la recuperación de datos de la base de datos, veremos el comando de SQL: SELECT. Este comando es uno de los más importantes y complejos en SQL, veremos algunos ejemplos y ustedes tendrán contestar las consignas, para ello pueden probar sus comandos en SQLITE.

Para listar todos los nombres de producto y sus precios usamos:

```
SELECT producto_nombre, producto_precio FROM productos;
```

Como ven los datos no viene ordenados, aparecen en el orden en que están guardados en la base de datos.

Para ver la misma información ordenada por precio usamos:

```
SELECT producto_nombre, producto_precio FROM productos ORDER BY  
producto_precio;
```

- 1) Escribir el comando para listar el nombre de los clientes y su dirección, ordenado por el nombre de cliente.

```
SELECT cliente_nombre, cliente_direccion FROM CLientes ORDER BY cliente_nombre;
```

- 2) Escribir el comando para listar los nombres de proveedores y su país, ordenado por el país.

```
SELECT proveedor_nombre, proveedor_pais FROM Proveedores ORDER BY proveedor_pais;
```

También pueden ordenar en forma descendente incluyendo DESC después del campo por el que ordenan.

Para listar todos los datos que cumplan con un criterio usamos SELECT combinado con WHERE y el criterio de búsqueda. Por ejemplo para buscar todos los productos que tengan un precio mayor a 10 usamos:

```
SELECT producto_nombre, producto_precio FROM productos where  
producto_precio > 10;
```

- 3) Escribir el comando para listar, de la tabla Orden_Items, todos los Producto_ID y su precio donde la cantidad pedida sea mayor a 100.

```
SELECT producto_ID, item_precio FROM Orden_Items WHERE cantidad > 100;
```

- 4) Escribir el comando para listar todos los productos con un precio inferior a 4.

```
SELECT * FROM Productos WHERE producto_precio < 4;
```

- 5) También puede utilizar los siguientes operadores. Identifique la función de cada uno:

= Verifica igualdad entre datos

!= Verifica Desigualdad entre datos

< Indica número menor que

> Indica numero mayor que

<= Indica numero menor o igual que

>= Indica número mayor o igual que

<> Verifica si dos valores son diferentes

Para realizar búsquedas más complejas pueden usar los operadores lógicos AND (y) OR (o) NOT (no). Por ejemplo, para listar el nombre y precio de todos los productos del proveedor BRS01 y de precio mayor a 7 usamos:

```
SELECT producto_nombre, producto_precio FROM productos WHERE  
proveedor_id='BRS01' AND producto_precio > 7;
```

- 6) Listar los nombre de producto y precio del proveedor DLL01 y de precio menor a 4.

```
SELECT producto_nombre, producto_precio FROM productos WHERE proveedor_id = 'DLL01' and  
producto_precio < 4;
```

- 7) Listar los nombre de producto y precio de los proveedores BRS01 y FNG01.

```
SELECT producto_nombre, producto_precio FROM productos WHERE proveedor_id = 'BRS01' OR proveedor_id  
= 'FNG01';
```

- 8) Listar los nombres de productos con un precio entre 4 y 10.

```
SELECT producto_nombre, producto_precio FROM productos WHERE producto_precio > 4 AND  
producto_precio < 10;
```

- 9) Investigar con la bibliografía recomendada o en internet las sentencias IN y BETWEEN.

Rescribir el comando de la respuesta 7) utilizando la sentencia IN.

```
SELECT producto_nombre, producto_precio FROM productos WHERE proveedor_id IN ('BRS01', 'FNG01');
```

Rescribir el comando de la respuesta 8) utilizando la sentencia BETWEEN.

```
SELECT producto_nombre, producto_precio FROM productos WHERE producto_precio BETWEEN 4 AND 10;
```

Si queremos listar todos los productos del proveedor DLL01 podemos usar:

```
SELECT producto_nombre from productos WHERE proveedor_ID='DLL01';
```

Pero si usamos la siguiente sentencia no nos devuelve la información solicitada:

```
SELECT producto_nombre from productos WHERE proveedor_ID='dll01';
```

La diferencia está en el uso de mayúsculas o minúsculas en Proveedor_ID. Para evitar esta fificultad podemos utilizar el operador LIKE.

```
SELECT producto_nombre from productos WHERE proveedor_ID LIKE 'dll01';
```

Además utilizando este operador podemos usar el carácter % como comodín, que representa cualquier cadena de caracteres, como en la sentencia siguiente.

```
SELECT producto_nombre from productos WHERE producto_nombre LIKE 'auto%';
```

10) Listar el nombre y precio de todos los productos de peluche.

```
SELECT producto_nombre, producto_precio FROM Productos WHERE producto_nombre LIKE '%peluche';
```

11) Listar el nombre y precio de todos los osos de la tabla de productos.

```
SELECT producto_nombre, producto_precio FROM Productos WHERE producto_nombre LIKE '%oso%';
```

También se pueden realizar operaciones entre los campos o un campo y una expresión, además se puede nombrar cualquier columna u operación con un alias usando la sentencia AS. Por ejemplo de la tabla Orden_items podemos calcular el precio total de cada ítem como el producto de la cantidad por el precio unitario de la siguiente manera:

```
SELECT producto_id, cantidad, item_precio, cantidad*item_precio AS  
precio_total FROM orden_items;
```

12) Ahora nos solicitan hacer una lista de precios de los productos con el precio por menor y por mayor. El precio por menor es el cargado en la base de datos, y el precio por mayor tiene un descuento del 10%. Listar todos los productos con su nombre, precio por menor y precio por mayor, nombrar el campo precio por menor: precio_menor y el precio por mayor: precio_mayor.

```
SELECT producto_nombre, producto_precio AS precio_por_menor, producto_precio - (producto_precio*0.1)  
AS precio_por_mayor FROM Productos;
```

13) Ahora nos solicitan realizar la misma lista, pero el precio de la base de datos es el precio por mayor, y el precio por menor es un 25% mayor.

```
SELECT producto_nombre, producto_precio AS precio_por_mayor, producto_precio + (producto_precio*0.25)  
AS precio_por_menor FROM Productos;
```

SQL nos permite realizar un resumen de los datos sin tener que mostrarnos todos los datos, por ejemplo, queremos conocer el precio del producto más caro, podemos usar:

```
SELECT max(producto_precio) AS precio_mayor FROM Productos;
```

O podemos contar la cantidad de productos con:

```
SELECT COUNT(*) AS Cant_productos FROM Productos;
```

14) También existen los siguientes comandos de resumen. Identifique la función de cada uno:

MAX (): Analiza los datos y muestra el campo con el máximo valor

MIN (): Analiza los datos y muestra el campo con el mínimo valor

COUNT (): Analiza los datos y cuenta el número de filas o registros

SUM (): Analiza los datos y suma los valores del campo de forma numérica

AVG (): Analiza los datos y calcula el promedio de los valores de los campos

15) Contar la cantidad de ítems pedidos de la orden de compra nro. 20005. Para esto van a tener que buscar en la tabla Orden_Items y sumar todos los campos cantidad de la orden de compra 20005 (van a tener que combinar el comando SELECT con WHERE)

```
SELECT SUM (cantidad) FROM Orden_items WHERE orden_numero = '20005';
```

16) Calcular el precio total de la orden de compra Nro. 20005. Recordar que para calcular el precio total van a tener que hacer la suma de todos los productos de cantidad por Item_Precio .

```
SELECT SUM (cantidad * item_precio) FROM Orden_Items WHERE orden_numero = '20005';
```

17) Calcular la cantidad total a pedir (entre todas las órdenes de compra) del producto BNBG01.

```
SELECT SUM (cantidad) FROM Orden_items WHERE producto_ID = 'BNBG01';
```



(CC)Luis Pérez – 2020

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.es>