

## PRUEBA TÉCNICA SQL

1.- Seleccionar todos los clientes ordenados por nombre del cliente

```
SELECT *  
FROM CLIENTES  
ORDER BY NOM_CLIENTE;
```

2.- Seleccionar los clientes cuya localidad sea Madrid

```
SELECT *  
FROM CLIENTES  
WHERE LOC_CLIENTE ='Madrid';
```

3.- Seleccionar todos los clientes cuyo código postal pertenezca a la provincia de Sevilla (los que comienzan por 41)

```
SELECT *  
FROM CLIENTES  
WHERE LEFT(CAST(CP_CLIENTE AS varchar), 2) = '41';
```

4.- Seleccionar todos los clientes cuyo código sea inferior a 10 o cuyo nombre de cliente comience por la letra A

```
SELECT *  
FROM CLIENTES  
WHERE COD_CLIENTE<10 AND LEFT(NOM_CLIENTE,1)= 'A';
```

5.- Contar cuántos clientes existen por cada localidad

```
SELECT LOC_CLIENTE, COUNT(*) AS CUANTOS  
FROM CLIENTES  
GROUP BY LOC_CLIENTE;
```

6.- Borrar los clientes cuyo nombre comience por la letra Z y termine por la letra Z

```
DELETE FROM CLIENTES  
WHERE LEFT(NOM_CLIENTE,1)= 'Z' AND RIGHT(NOM_CLIENTE,1)= 'Z';
```

7.- Modificar el valor del código postal de los clientes cuya localidad sea Madrid con el valor 28001

```
UPDATE CLIENTES SET CP_CLIENTE = 28001  
WHERE LOC_CLIENTE='Madrid';
```

8.- Seleccionar los clientes cuyo importe de sus VENTAS sea distinto de cero.

```
SELECT *  
FROM CLIENTES, VENTAS  
WHERE VENTAS.COD_CLIENTE=CLIENTES.COD_CLIENTE  
AND VENTAS.IMPORTE <>0;
```

9.- Seleccionar todos los clientes junto con el importe de las ventas de cada uno.

```
SELECT CLIENTES.*, VENTAS.IMPORTE
```

```
FROM CLIENTES LEFT JOIN VENTAS  
ON VENTAS.COD_CLIENTE=CLIENTES.COD_CLIENTE;
```

10.- Modificar el importe de las ventas de los clientes a cero para aquellos clientes cuyo nombre contenga al menos una letra A.

```
UPDATE VENTAS SET IMPORTE = 0  
WHERE COD_CLIENTE IN  
(SELECT COD_CLIENTE  
FROM CLIENTES  
WHERE NOM_CLIENTE LIKE '%A%');
```

11. Subirlo a Github para compartirlo.