

1.1.3 Consultas sobre una tabla

1. Lista el nombre de todos los productos que hay en la tabla `producto`.

```
SELECT nombre  
FROM producto
```

2. Lista los nombres y los precios de todos los productos de la tabla `producto`.

```
SELECT nombre, precio  
FROM producto
```

3. Lista todas las columnas de la tabla `producto`.

```
SELECT *  
FROM producto
```

4. Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD).

```
SELECT nombre, precio AS Euros, precio * 1.09 AS Dolar  
FROM producto
```

5. Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD). Utiliza los siguientes alias para las columnas: nombre de producto, euros, dólares.

```
SELECT nombre AS Nombre_de_Producto, precio AS Euros,  
FORMAT ((Precio * 1.09), "C") AS dólares  
FROM producto
```

6. Lista los nombres y los precios de todos los productos de la tabla `producto`, convirtiendo los nombres a mayúscula.

```
SELECT UPPER (nombre) AS Nombre, precio  
FROM producto
```

7. Lista los nombres y los precios de todos los productos de la tabla `producto`, convirtiendo los nombres a minúscula.

```
SELECT LOWER (nombre) AS Nombre, precio  
FROM producto
```

8. Lista el nombre de todos los fabricantes en una columna, y en otra columna obtenga en mayúsculas los dos primeros caracteres del nombre del fabricante.

```
SELECT nombre, UPPER (SUBSTRING(nombre, 1, 2)) AS Iniciales
FROM fabricante
```

9. Lista los nombres y los precios de todos los productos de la tabla `producto`, redondeando el valor del precio.

```
SELECT nombre, ROUND(precio) FROM producto
```

10. Lista los nombres y los precios de todos los productos de la tabla `producto`, truncando el valor del precio para mostrarlo sin ninguna cifra decimal.

```
SELECT nombre, FORMAT(precio, "##")
FROM producto
```

11. Lista el código de los fabricantes que tienen productos en la tabla `producto`.

```
SELECT codigo_fabricante
FROM producto
GROUP BY codigo_fabricante
```

12. Lista el código de los fabricantes que tienen productos en la tabla `producto`, eliminando los códigos que aparecen repetidos.

Manera 1

```
SELECT DISTINCT codigo_fabricante
FROM producto
```

Manera 2

```
SELECT codigo_fabricante
FROM producto
GROUP BY codigo_fabricante
```

13. Lista los nombres de los fabricantes ordenados de forma ascendente.

```
SELECT nombre
FROM fabricante
ORDER BY nombre ASC
```

14. Lista los nombres de los fabricantes ordenados de forma descendente.

```
SELECT nombre
FROM fabricante
ORDER BY nombre DESC
```

15. Lista los nombres de los productos ordenados en primer lugar por el nombre de forma ascendente y en segundo lugar por el precio de forma descendente.

```
SELECT nombre, precio
FROM producto
ORDER BY nombre ASC, precio DESC
```

16. Devuelve una lista con las 5 primeras filas de la tabla fabricante.

```
SELECT *
FROM fabricante
LIMIT 5
```

17. Devuelve una lista con 2 filas a partir de la cuarta fila de la tabla fabricante. La cuarta fila también se debe incluir en la respuesta.

```
SELECT *
FROM fabricante
LIMIT 3,2
```

18. Lista el nombre y el precio del producto más barato. (Utilice solamente las cláusulas ORDER BY y LIMIT)

```
SELECT nombre, precio
FROM producto
ORDER BY precio ASC
LIMIT 0,1
```

19. Lista el nombre y el precio del producto más caro. (Utilice solamente las cláusulas ORDER BY y LIMIT)

```
SELECT nombre, precio
FROM producto
ORDER BY precio DESC
LIMIT 0,1
```

20. Lista el nombre de todos los productos del fabricante cuyo código de fabricante es igual a 2.

```
SELECT nombre
FROM producto
WHERE codigo_fabricante = 2
```

21. Lista el nombre de los productos que tienen un precio menor o igual a 120€.

```
SELECT nombre
FROM producto
WHERE precio <= 120
```

22. Lista el nombre de los productos que tienen un precio mayor o igual a 400€.

```
SELECT nombre
FROM producto
WHERE precio >= 400
```

23. Lista el nombre de los productos que **no tienen** un precio mayor o igual a 400€.

```
SELECT nombre
FROM producto
WHERE precio < 400
```

24. Lista todos los productos que tengan un precio entre 80€ y 300€. Sin utilizar el operador BETWEEN.

```
SELECT *
FROM producto
WHERE precio <= 80 AND precio >= 300
```

25. Lista todos los productos que tengan un precio entre 60€ y 200€. Utilizando el operador BETWEEN.

```
SELECT *
FROM producto
WHERE precio BETWEEN 60 AND 200
```

26. Lista todos los productos que tengan un precio mayor que 200€ y que el código de fabricante sea igual a 6.

```
SELECT *
FROM producto WHERE precio > 200
AND codigo_fabricante = 6
```

27. Lista todos los productos donde el código de fabricante sea 1, 3 o 5. Sin utilizar el operador IN.

```
SELECT *
FROM producto
WHERE codigo_fabricante = 1
OR codigo_fabricante = 3
OR codigo_fabricante = 5
```

28. Lista todos los productos donde el código de fabricante sea 1, 3 o 5. Utilizando el operador IN.

```
SELECT *
FROM producto
WHERE codigo_fabricante IN (1,3,5)
```

29. Lista el nombre y el precio de los productos en céntimos (Habrás que multiplicar por 100 el valor del precio). Cree un alias para la columna que contiene el precio que se llame `céntimos`.

```
SELECT nombre, precio*100 AS céntimos
FROM producto
```

30. Lista los nombres de los fabricantes cuyo nombre empiece por la letra `s`.

```
SELECT nombre
FROM fabricante
WHERE nombre LIKE 's%'
```

31. Lista los nombres de los fabricantes cuyo nombre termine por la vocal `e`.

```
SELECT nombre
FROM fabricante
WHERE nombre LIKE '%e'
```

32. Lista los nombres de los fabricantes cuyo nombre contenga el carácter `w`.

```
SELECT nombre
FROM fabricante
WHERE nombre LIKE '%w%'
```

33. Lista los nombres de los fabricantes cuyo nombre sea de 4 caracteres.

```
SELECT nombre
FROM fabricante
WHERE LENGTH(nombre)=4
```

34. Devuelve una lista con el nombre de todos los productos que contienen la cadena `Portátil` en el nombre.

```
SELECT nombre
FROM producto
WHERE nombre LIKE '%Portátil%'
```

35. Devuelve una lista con el nombre de todos los productos que contienen la cadena `Monitor` en el nombre y tienen un precio inferior a 215 €.

```
SELECT nombre
FROM producto
WHERE nombre LIKE '%Monitor%' AND precio < 215
```

36. Lista el nombre y el precio de todos los productos que tengan un precio mayor o igual a 180€. Ordene el resultado en primer lugar por el precio (en orden descendente) y en segundo lugar por el nombre (en orden ascendente).

```
SELECT nombre, precio
FROM producto
WHERE precio >=180
ORDER BY precio DESC, nombre ASC
```

1.1.4 Consultas multitable (Composición interna)

Resuelva todas las consultas utilizando la sintaxis de SQL1 y SQL2.

1. Devuelve una lista con el nombre del producto, precio y nombre de fabricante de todos los productos de la base de datos.

```
SELECT producto.nombre, precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
```

2. Devuelve una lista con el nombre del producto, precio y nombre de fabricante de todos los productos de la base de datos. Ordene el resultado por el nombre del fabricante, por orden alfabético.

```
SELECT producto.nombre, precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
ORDER BY fabricante.nombre ASC
```

3. Devuelve una lista con el código del producto, nombre del producto, código del fabricante y nombre del fabricante, de todos los productos de la base de datos.

```
SELECT producto.codigo, producto.nombre, fabricante.codigo,
fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
```

4. Devuelve el nombre del producto, su precio y el nombre de su fabricante, del producto más barato.

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
ORDER BY precio LIMIT 0,1
```

5. Devuelve el nombre del producto, su precio y el nombre de su fabricante, del producto más caro.

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
ORDER BY precio DESC LIMIT 0,1
```

6. Devuelve una lista de todos los productos del fabricante Lenovo.

```
SELECT producto.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Lenovo"
```

7. Devuelve una lista de todos los productos del fabricante Crucial que tengan un precio mayor que 200€.

```
SELECT producto.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Crucial" AND producto.precio > 200
```

8. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett-Packard y Seagate. Sin utilizar el operador IN.

```
SELECT producto.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus" OR fabricante.nombre="Hewlett-Packard" OR fabricante.nombre="Seagate"
```

9. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett-Packard y Seagate. Utilizando el operador IN.

```
SELECT producto.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre IN ("Asus", "Hewlett-Packard", "Seagate")
```

10. Devuelve un listado con el nombre y el precio de todos los productos de los fabricantes cuyo nombre termine por la vocal e.

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre LIKE '%e'
```

11. Devuelve un listado con el nombre y el precio de todos los productos cuyo nombre de fabricante contenga el carácter `w` en su nombre.

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre LIKE '%w%'
```

12. Devuelve un listado con el nombre de producto, precio y nombre de fabricante, de todos los productos que tengan un precio mayor o igual a 180€. Ordene el resultado en primer lugar por el precio (en orden descendente) y en segundo lugar por el nombre (en orden ascendente)

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE producto.precio >= 180
ORDER BY precio DESC , producto.nombre ASC
```

13. Devuelve un listado con el código y el nombre de fabricante, solamente de aquellos fabricantes que tienen productos asociados en la base de datos.

```
SELECT fabricante.codigo, fabricante.nombre
FROM producto LEFT JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
GROUP BY fabricante.codigo
```

1.1.5 Consultas multitable (Composición externa)

Resuelva todas las consultas utilizando las cláusulas `LEFT JOIN` y `RIGHT JOIN`.

1. Devuelve un listado de **todos los fabricantes** que existen en la base de datos, junto con los productos que tiene cada uno de ellos. El listado deberá mostrar también aquellos fabricantes que no tienen productos asociados.

```
SELECT fabricante.codigo, fabricante.nombre, producto.nombre
FROM fabricante LEFT JOIN producto
ON producto.codigo_fabricante = fabricante.codigo
```

2. Devuelve un listado donde sólo aparezcan aquellos fabricantes que no tienen ningún producto asociado.

```
SELECT fabricante.codigo, fabricante.nombre, producto.nombre
FROM fabricante LEFT JOIN producto
ON producto.codigo_fabricante = fabricante.codigo
WHERE producto.nombre IS NULL
ORDER BY codigo_fabricante
```


3. ¿Pueden existir productos que no estén relacionados con un fabricante?
Justifique su respuesta.

No, porque la relación entre tablas exige que cada campo de código_fabricante en la tabla PRODUCTO esta vinculado a la tabla FABRICANTE por su campo código, ya que se definió una Foreign Key (∞ a 1) en su creación, se debería eliminar esa relación para poder existir productos sin fabricante.

1.1.6 Consultas resumen

1. Calcula el número total de productos que hay en la tabla productos.

```
SELECT COUNT(*)  
FROM Producto
```

2. Calcula el número total de fabricantes que hay en la tabla fabricante.

```
SELECT COUNT(*)  
FROM fabricante
```

3. Calcula el número de valores distintos de código de fabricante aparecen en la tabla productos.

```
SELECT COUNT(DISTINCT(codigo_fabricante))  
FROM producto
```

4. Calcula la media del precio de todos los productos.

```
SELECT AVG (precio)  
FROM producto
```

5. Calcula el precio más barato de todos los productos.

```
SELECT MIN(precio)  
FROM producto
```

6. Calcula el precio más caro de todos los productos.

```
SELECT MAX(precio)  
FROM producto
```

7. Lista el nombre y el precio del producto más barato.

```
SELECT nombre, precio  
FROM producto ORDER BY precio ASC LIMIT 0,1
```

8. Lista el nombre y el precio del producto más caro.

```
SELECT nombre, precio
FROM producto
ORDER BY precio DESC LIMIT 0,1
```

9. Calcula la suma de los precios de todos los productos.

```
SELECT SUM(precio)
FROM producto
```

10. Calcula el número de productos que tiene el fabricante Asus.

```
SELECT COUNT(*) AS Cantidad
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus"
```

11. Calcula la media del precio de todos los productos del fabricante Asus.

```
SELECT AVG(precio) AS MEDIA
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus"
```

12. Calcula el precio más barato de todos los productos del fabricante Asus.

```
SELECT (precio)
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus"
ORDER BY precio LIMIT 0,1
```

13. Calcula el precio más caro de todos los productos del fabricante Asus.

```
SELECT (precio)
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus"
ORDER BY precio DESC LIMIT 0,1
```

14. Calcula la suma de todos los productos del fabricante Asus.

```
SELECT SUM(precio)
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Asus"
```

15. Muestra el precio máximo, precio mínimo, precio medio y el número total de productos que tiene el fabricante Crucial.

```
SELECT fabricante.nombre, MAX(precio) AS Maximo, MIN(precio) AS
Minimo, AVG(precio) AS Media, COUNT(precio) AS Cantidad
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE fabricante.nombre="Crucial"
```

16. Muestra el número total de productos que tiene cada uno de los fabricantes. El listado también debe incluir los fabricantes que no tienen ningún producto. El resultado mostrará dos columnas, una con el nombre del fabricante y otra con el número de productos que tiene. Ordene el resultado descendientemente por el número de productos.

```
SELECT fabricante.nombre, COUNT(producto.nombre) AS Cantidad
FROM producto RIGHT JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
GROUP BY fabricante.nombre
ORDER BY Cantidad DESC
```

17. Muestra el precio máximo, precio mínimo y precio medio de los productos de cada uno de los fabricantes. El resultado mostrará el nombre del fabricante junto con los datos que se solicitan.

```
SELECT fabricante.nombre, MAX(producto.precio) AS MaxPrecio,
MIN(producto.precio) AS MinPrecio, AVG(producto.precio) AS Media
FROM producto RIGHT JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
GROUP BY fabricante.nombre
```

18. Muestra el precio máximo, precio mínimo, precio medio y el número total de productos de los fabricantes que tienen un precio medio superior a 200€. No es necesario mostrar el nombre del fabricante, con el código del fabricante es suficiente.

```
SELECT codigo_fabricante, MAX(precio) AS MaxPrecio, MIN(precio)
AS MinPrecio, AVG(precio) AS Media, COUNT(nombre) AS Cantidad
FROM producto
GROUP BY codigo_fabricante
HAVING Media > 200
```

19. Muestra el nombre de cada fabricante, junto con el precio máximo, precio mínimo, precio medio y el número total de productos de los fabricantes que tienen un precio medio superior a 200€. Es necesario mostrar el nombre del fabricante.

```
SELECT fabricante.codigo, fabricante.nombre,  
MAX(producto.precio) AS MaxPrecio, MIN(producto.precio) AS  
MinPrecio, AVG(producto.precio) AS Media, COUNT(producto.nombre)  
AS Cantidad  
FROM producto RIGHT JOIN fabricante  
ON producto.codigo_fabricante = fabricante.codigo  
GROUP BY fabricante.nombre  
HAVING Media > 200
```

20. Calcula el número de productos que tienen un precio mayor o igual a 180€.

```
SELECT COUNT(codigo) AS Cantidad  
FROM producto  
WHERE precio >= 180
```

21. Calcula el número de productos que tiene cada fabricante con un precio mayor o igual a 180€.

```
SELECT fabricante.nombre, COUNT(producto.codigo) AS Cantidad  
FROM producto INNER JOIN fabricante  
ON producto.codigo_fabricante= fabricante.codigo  
WHERE precio >= 180  
GROUP BY fabricante.nombre
```

22. Lista el precio medio los productos de cada fabricante, mostrando solamente el código del fabricante.

```
SELECT codigo_fabricante, AVG(producto.precio) AS Media  
FROM producto  
GROUP BY codigo_fabricante
```

23. Lista el precio medio los productos de cada fabricante, mostrando solamente el nombre del fabricante.

```
SELECT fabricante.nombre, AVG(producto.precio) AS Media  
FROM producto INNER JOIN fabricante  
ON producto.codigo_fabricante= fabricante.codigo  
GROUP BY fabricante.nombre
```

24. Lista los nombres de los fabricantes cuyos productos tienen un precio medio mayor o igual a 150€.

```
SELECT fabricante.nombre, AVG(producto.precio) AS Media  
FROM producto INNER JOIN fabricante  
ON producto.codigo_fabricante= fabricante.codigo  
GROUP BY fabricante.nombre  
HAVING Media > 150
```

25. Devuelve un listado con los nombres de los fabricantes que tienen 2 o más productos.

```
SELECT fabricante.nombre
FROM producto LEFT JOIN fabricante
ON producto.codigo_fabricante= fabricante.codigo
GROUP BY fabricante.nombre
HAVING COUNT(producto.codigo) >= 2
```

26. Devuelve un listado con los nombres de los fabricantes y el número de productos que tiene cada uno con un precio superior o igual a 220 €. No es necesario mostrar el nombre de los fabricantes que no tienen productos que cumplan la condición.

Ejemplo del resultado esperado:

nombre	total
Lenovo	2
Asus	1
Crucial	1

```
SELECT fabricante.nombre , COUNT(producto.codigo) AS Total
FROM producto LEFT JOIN fabricante
ON producto.codigo_fabricante= fabricante.codigo
WHERE producto.precio >= 220
GROUP BY fabricante.codigo
ORDER BY Total DESC
```

27. Devuelve un listado con los nombres de los fabricantes y el número de productos que tiene cada uno con un precio superior o igual a 220 €. El listado debe mostrar el nombre de todos los fabricantes, es decir, si hay algún fabricante que no tiene productos con un precio superior o igual a 220€ deberá aparecer en el listado con un valor igual a 0 en el número de productos.

Ejemplo del resultado esperado:

nombre	total
Lenovo	2
Crucial	1
Asus	1
Huawei	0
Samsung	0
Gigabyte	0
Hewlett-Packard	0
Xiaomi	0
Seagate	0

```
SELECT fabricante.nombre, COUNT(producto.codigo) AS Total FROM
fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante WHERE
producto.precio >= 220
GROUP BY fabricante.codigo
UNION
SELECT fabricante.nombre, "0"
FROM fabricante
WHERE fabricante.codigo NOT IN ( SELECT fabricante.codigo
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE producto.precio >= 220
GROUP BY fabricante.codigo)
ORDER BY Total DESC
```

28. Devuelve un listado con los nombres de los fabricantes donde la suma del precio de todos sus productos es superior a 1000 €.

```
SELECT fabricante.nombre
FROM fabricante LEFT JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
GROUP BY fabricante.nombre
HAVING SUM(producto.precio) > 1000
```

29. Devuelve un listado con el nombre del producto más caro que tiene cada fabricante. El resultado debe tener tres columnas: nombre del producto, precio y nombre del fabricante. El resultado tiene que estar ordenado alfabéticamente de menor a mayor por el nombre del fabricante.

```
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto, fabricante
WHERE producto.codigo_fabricante = fabricante.codigo
AND producto.precio = ( SELECT MAX(producto.precio)
                        FROM producto
                        WHERE producto.codigo_fabricante = fabricante.codigo )
ORDER BY fabricante.nombre
```

1.1.7 Subconsultas (En la cláusula **WHERE**)

1.1.7.1 Con operadores básicos de comparación

1. Devuelve todos los productos del fabricante **Lenovo**. (Sin utilizar **INNER JOIN**).

```
SELECT *
FROM producto
WHERE codigo_fabricante = ( SELECT codigo
                           FROM fabricante
                           WHERE nombre = "Lenovo")
```

2. Devuelve todos los datos de los productos que tienen el mismo precio que el producto más caro del fabricante `Lenovo`. (Sin utilizar `INNER JOIN`).

```
SELECT *
FROM producto
WHERE precio = ( SELECT MAX(precio)
                  FROM producto
                  WHERE codigo_fabricante = ( SELECT codigo
                                                FROM fabricante
                                                WHERE nombre = "Lenovo"))
```

3. Lista el nombre del producto más caro del fabricante `Lenovo`.

```
SELECT producto.nombre, producto.precio
FROM fabricante, producto
WHERE fabricante.codigo = producto.codigo_fabricante
AND fabricante.nombre="Lenovo"
AND producto.precio= (SELECT MAX(precio)
                      FROM fabricante INNER JOIN producto
                      ON fabricante.codigo=producto.codigo_fabricante
                      WHERE fabricante.nombre="Lenovo")
```

4. Lista el nombre del producto más barato del fabricante `Hewlett-Packard`.

```
SELECT producto.nombre, producto.precio
FROM fabricante, producto
WHERE fabricante.codigo = producto.codigo_fabricante
AND fabricante.nombre="Hewlett-Packard"
AND producto.precio = ( SELECT MIN(precio)
                        FROM fabricante INNER JOIN producto
                        ON fabricante.codigo = producto.codigo_fabricante
                        WHERE fabricante.nombre = 'Hewlett-Packard')
```

5. Devuelve todos los productos de la base de datos que tienen un precio mayor o igual al producto más caro del fabricante `Lenovo`.

```
SELECT *
FROM producto
WHERE precio >= (SELECT MAX(precio)
                 FROM fabricante INNER JOIN producto
                 ON fabricante.codigo = producto.codigo_fabricante
                 WHERE fabricante.nombre = 'Lenovo')
```

6. Lista todos los productos del fabricante `Asus` que tienen un precio superior al precio medio de todos sus productos.

```
SELECT *
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE fabricante.nombre = 'Asus'
AND producto.precio > ( SELECT AVG(precio)
                        FROM fabricante INNER JOIN producto
                        ON fabricante.codigo = producto.codigo_fabricante
                        WHERE fabricante.nombre = 'Asus')
```

1.1.7.2 Subconsultas con `ALL` y `ANY`

8. Devuelve el producto más caro que existe en la tabla `producto` sin hacer uso de `MAX`, `ORDER BY` ni `LIMIT`.

```
SELECT *
FROM producto
WHERE precio >= ALL ( SELECT precio
                     FROM producto)
```

9. Devuelve el producto más barato que existe en la tabla `producto` sin hacer uso de `MIN`, `ORDER BY` ni `LIMIT`.

```
SELECT *
FROM producto
WHERE precio <= ALL ( SELECT precio
                     FROM producto)
```

10. Devuelve los nombres de los fabricantes que tienen productos asociados. (Utilizando `ALL` o `ANY`).

```
SELECT nombre
FROM fabricante
WHERE codigo = ANY ( SELECT codigo_fabricante
                    FROM producto)
```

11. Devuelve los nombres de los fabricantes que no tienen productos asociados. (Utilizando `ALL` o `ANY`).

```
SELECT nombre
FROM fabricante
WHERE codigo != ALL ( SELECT codigo_fabricante
                     FROM producto)
```


1.1.7.3 Subconsultas con IN y NOT IN

12. Devuelve los nombres de los fabricantes que tienen productos asociados.
(Utilizando IN o NOT IN).

```
SELECT nombre
FROM fabricante
WHERE codigo IN ( SELECT codigo_fabricante
                  FROM producto)
```

13. Devuelve los nombres de los fabricantes que no tienen productos asociados.
(Utilizando IN o NOT IN).

```
SELECT nombre
FROM fabricante
WHERE codigo NOT IN ( SELECT codigo_fabricante
                     FROM producto)
```

1.1.7.4 Subconsultas con EXISTS y NOT EXISTS

14. Devuelve los nombres de los fabricantes que tienen productos asociados.
(Utilizando EXISTS o NOT EXISTS).

```
SELECT nombre
FROM fabricante
WHERE EXISTS ( SELECT codigo_fabricante
               FROM producto
               WHERE producto.codigo_fabricante = fabricante.codigo)
```

15. Devuelve los nombres de los fabricantes que no tienen productos asociados.
(Utilizando EXISTS o NOT EXISTS).

```
SELECT nombre
FROM fabricante
WHERE NOT EXISTS ( SELECT codigo_fabricante
                  FROM producto
                  WHERE producto.codigo_fabricante = fabricante.codigo)
```

1.1.7.5 Subconsultas correlacionadas

16. Lista el nombre de cada fabricante con el nombre y el precio de su producto más caro.

```
SELECT fabricante.nombre, producto.precio
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE producto.precio = ( SELECT MAX(precio)
                          FROM producto
                          WHERE codigo_fabricante = fabricante.codigo )
```

17. Devuelve un listado de todos los productos que tienen un precio mayor o igual a la media de todos los productos de su mismo fabricante.

```
SELECT *
FROM producto AS tabla1
WHERE precio >= (SELECT AVG(precio)
                FROM producto AS tabla2
                WHERE tabla1.codigo_fabricante = tabla2.codigo_fabricante)
```

18. Lista el nombre del producto más caro del fabricante `Lenovo`.

```
SELECT producto.nombre
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE fabricante.nombre = "Lenovo"
AND producto.precio = ( SELECT MAX(precio)
                        FROM producto
                        WHERE codigo_fabricante = fabricante.codigo)
```

1.1.8 Subconsultas (En la cláusula `HAVING`)

7. Devuelve un listado con todos los nombres de los fabricantes que tienen el mismo número de productos que el fabricante `Lenovo`.

```
SELECT fabricante.nombre, COUNT(producto.codigo)
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
GROUP BY fabricante.codigo
HAVING COUNT(producto.codigo) >= ( SELECT COUNT(producto.codigo)
                                    FROM fabricante INNER JOIN producto
                                    ON fabricante.codigo = producto.codigo_fabricante
                                    WHERE fabricante.nombre = "Lenovo")
```