**Guía de aprendizaje BD\_01**

**Robinson Puentes Cardenas**

**CC. 1111336624**

**Kevin Santiago torres higuera**

**CC. 1031650532**

Ficha: 2996234

Servicio Nacional De Aprendizaje SENA

Análisis y Desarrollo de Software

Instructor: Mauricio Estupiñán

17 de mayo de 2025

**DESARROLLO DE LA ACTIVIDAD.**

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

UPPER (nombre): convierte el nombre del producto a mayúsculas.

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

ORDER BY precio ASC: ordena los productos del más barato al más caro.

LIMIT 1: selecciona solo el primer resultado (el más barato).

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

SELECTnombre, precio: Selecciona las columnas de la tabla

FROM producto: Indica de donde será tomada la información

ORDER BY precio DESC: Ordena los productos por su precio de mayor a menor.

DESC: significa descendente.

LIMIT 1: Limita la salida a solo un resultado.

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

**SELECT nombre:** Tabla que queremos mostrar.

**FROM producto:** Tabla que queremos consultar

**WHERE codigo\_fabricante = 2:** Esta cláusula filtra los resultados para incluir solo los productos donde el valor de la columna codigo\_fabricante sea igual a 2.

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

**SELECT** nombre: Esta parte indica qué columnas quieres mostrar en el resultado.

**FROM producto:** Aquí le dices a SQL de qué tabla debe sacar los datos.

**WHERE precio <= 120:** Esta es una condición de filtro.

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

**SELECT nombre:** Muestra únicamente el nombre de los productos.

**FROM producto:** Indica que los datos provienen de la tabla producto.

**WHERE precio >= 400:** Filtra los productos para mostrar solo aquellos con un precio mayor o igual a 400 euros.

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

**SELECT nombre**: consulta qué columna quieres que se muestre

**FROM producto**: indica de qué tabla quieres obtener la información.

**WHERE precio < 400**: filtra los resultados.

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

**operador BETWEEN.**

**SELECT nombre, precio**: indica que queremos mostrar

**FROM producto**: Especifica de donde vamos a obtener la información

**WHERE precio > 80 AND precio < 300**: definimos el filtro para los productos.

**AND**: Conecta dos o más condiciones

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

**operador BETWEEN.**

**SELECT nombre, precio**: Indica que queremos ver el nombre y el precio de los productos.

**FROM producto**: La tabla a consultar

**WHERE precio BETWEEN 60 AND 200**: Esta es la cláusula clave que utiliza el operador BETWEEN.

**precio BETWEEN 60 AND 200**: Se usa para incluir solo los productos que cuesten entre 60€, y 200€ .

**26. Lista todos los productos que tengan un precio mayor que 200€ y que el identificador**

**de fabricante sea igual a 6.**

**SELECT nombre, precio, codigo\_fabricante**: Indica que queremos ver.

**FROM producto**: Indica que estamos consultando la tabla producto.

**WHERE precio > 200 AND codigo\_fabricante = 6**: Filtra los resultados

**27. Lista todos los productos donde el identificador de fabricante sea 1, 3 o 5. Sin utilizar**

**el operador IN.**

**SELECT nombre, codigo\_fabricante**: Indicamos que queremos ver el nombre del producto y su código de fabricante.

**FROM producto**: Especificamos que la información la obtendremos de la tabla "producto".

**WHERE:** Filtra los resultados.

**28. Lista todos los productos donde el identificador de fabricante sea 1, 3 o 5. Utilizando**

**el operador IN.**

**SELECT nombre, codigo\_fabricante**: Indicamos que queremos ver el nombre del producto y su código de fabricante.

**FROM producto**: Indicamos de que tabla obtendremos la información.

**WHERE codigo\_fabricante IN (1, 3, 5)**: Esta es la cláusula que filtra los resultados

**IN (1, 3, 5)**: El operador IN permite verificar si el valor de una columna coincide con alguno de los valores dentro de la lista entre paréntesis. En este caso, estamos buscando filas donde el valor de codigo\_fabricante sea igual a 1, o igual a 3, o igual a 5.

**29. Lista el nombre y el precio de los productos en céntimos (Habrá 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**: Consulta las columnas que queremos mostrar:

* **nombre**: Selecciona la columna con el nombre del producto.
* **precio \* 100**: Multiplica el valor de la columna precio por 100 para convertirlo a céntimos.
* **AS céntimos**: Crea un alias (un nombre temporal) para la columna resultante de la multiplicación. En lugar de mostrarse como precio \* 100, la columna se mostrará con el encabezado "céntimos".

**FROM producto**: Indica que estamos obteniendo los datos de la tabla producto.

**30. Lista los nombres de los fabricantes cuyo nombre empiece por la letra S.**

**SELECT nombre**: Especificamos la columna que queremos mostrar

**FROM fabricante**: consultamos la tabla llamada fabricante.

**WHERE nombre LIKE 'S%'**: Esta es la cláusula WHERE que filtra los resultados utilizando el operador LIKE.

**LIKE**: El operador LIKE se utiliza para realizar búsquedas de patrones en cadenas de texto.

**'S%'**: Este es el patrón de búsqueda.

**S**: Especifica que la cadena debe comenzar con la letra "S".

**%**: Este es un comodín que representa cualquier secuencia de cero o más caracteres. Por lo tanto, 'S%' coincidirá con cualquier nombre que comience con "S", seguido de cualquier otra letra o ninguna.

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

**'%e'**: A diferencia del ejercicio anterior en este caso el comodín va ubicado en la parte de atrás de lo que queremos mostrar en la búsqueda.

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

**'%w%':** a diferencia de los ejercicios anteriores en este caso el comodín va ubicado a los dos costados de lo que queremos mostrar en la búsqueda.

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

**SELECT nombre**: La columna que queremos mostrar.

**FROM fabricante**: La tabla que queremos consultar.

**WHERE** filtra los resultados

**LENGTH**: devuelve el número de caracteres de la cadena de texto en la columna.

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

**Portátil en el nombre.**

**'%Portátil%' :** a diferencia de los ejercicios anteriores en este caso el comodín va ubicado a los dos costados de la palabra que queremos mostrar en la búsqueda.

**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**: columna que queremos mostrar.

**FROM producto**: Tabla que queremos consultar.

**WHERE:** Clausula que filtra los resultados utilizando condiciones

**LIKE '%Monitor%'**: Esta condición selecciona los productos que contienen el nombre a buscar.

**AND**: Este operador lógico exige que ambas condiciones sean verdaderas para que una fila sea seleccionada.

**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**: Lo que queremos mostrar.

**FROM producto**: La tabla que queremos consultar.

**WHERE precio >= 180**: Esta cláusula filtra los resultados para incluir solo los productos cuyo precio sea mayor o igual a 180€.

**ORDER BY precio DESC, nombre ASC**: Esta cláusula especifica el orden en que se mostrarán los resultados:

**ORDER BY precio DESC**: Primero, los resultados se ordenarán por la columna "precio" en orden descendente.

**nombre ASC**: En segundo lugar, dentro de los grupos de productos que tienen el mismo precio, los resultados se ordenarán por la columna "nombre" en orden ascendente.

1.1.4 Consultas multitabla (Composición interna)

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

8. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett

Packardy Seagate. Sin utilizar el operador IN.

**Mysql 1**

**SELECT p.nombre AS nombre\_producto, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto de la tabla producto (alias p) y le asignamos el alias nombre\_producto, y seleccionamos el nombre del fabricante de la tabla fabricante (alias f) y le asignamos el alias nombre\_fabricante.

**FROM producto p, fabricante f**: Especificamos las dos tablas que vamos a utilizar en la consulta: producto con el alias p y fabricante con el alias f. La coma indica un JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Esta es la condición de JOIN. Vincula las filas de las dos tablas donde la columna codigo\_fabricante de la tabla producto coincide con la columna codigo (clave primaria) de la tabla fabricante.

**AND (f.nombre = 'Asus' OR f.nombre = 'Hewlett-Packard' OR f.nombre = 'Seagate')**: Esta condición adicional filtra los resultados para incluir solo aquellos productos cuyo fabricante tiene el nombre 'Asus' o 'Hewlett-Packard' o 'Seagate'.

**Sintaxis SQL2 (JOIN explícito con ON):**

SQL

**Mysql 2**

**SELECT p.nombre AS nombre\_producto, f.nombre** AS **nombre\_fabricante: Al igual que en la sintaxis SQL1, seleccionamos el nombre del producto y el nombre del fabricante, utilizando alias para mayor claridad.**

**FROM producto p JOIN fabricante f ON p.codigo\_fabricante = f.codigo: Aquí utilizamos la cláusula JOIN explícita (en este caso, un INNER JOIN que es el comportamiento predeterminado). La condición para unir las tablas se especifica con la cláusula ON. Vinculamos las filas donde p.codigo\_fabricante es igual a f.codigo.**

**WHERE f.nombre = 'Asus' OR f.nombre = 'Hewlett-Packard' OR f.nombre = 'Seagate': Similar a la sintaxis SQL1, esta cláusula WHERE filtra los resultados para mostrar solo los productos de los fabricantes especificados.**

9. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett Packardy Seagate. Utilizando el operador IN**.**

**Mysql 1**

**SELECT p.nombre AS nombre\_producto, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto (alias nombre\_producto) y el nombre del fabricante (alias nombre\_fabricante).

**FROM producto p, fabricante f**: Especificamos las tablas producto (alias p) y fabricante (alias f) para el JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Condición de JOIN que relaciona productos con sus fabricantes a través del código.

**AND f.nombre IN ('Asus', 'Hewlett-Packard', 'Seagate')**: Filtra los resultados para incluir solo los productos cuyo fabricante tenga un nombre que se encuentre dentro de la lista ('Asus', 'Hewlett-Packard', 'Seagate'). El operador IN simplifica la comparación con múltiples valores.

**Mysql 2**

**SELECT p.nombre AS nombre\_producto, f.nombre AS nombre\_fabricante**: Similar a la sintaxis SQL1, seleccionamos el nombre del producto y del fabricante con alias.

**FROM producto p JOIN fabricante f ON p.codigo\_fabricante = f.codigo**: Realizamos un INNER JOIN explícito entre las tablas producto y fabricante utilizando la condición p.codigo\_fabricante = f.codigo.

**WHERE f.nombre IN ('Asus', 'Hewlett-Packard', 'Seagate')**: Filtramos los resultados utilizando el operador IN para seleccionar solo los productos de los fabricantes especificados.

10. Devuelve un listado con el nombre y el precio de todos los productos de los

fabricantes cuyo nombre termine por la vocal e.

**Mysql 1**

**SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto (alias nombre\_producto), el precio del producto (p.precio), y el nombre del fabricante (alias nombre\_fabricante).

**FROM producto p, fabricante f**: Especificamos las tablas producto (alias p) y fabricante (alias f) para el JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Condición de JOIN que relaciona productos con sus fabricantes a través del código.

**AND f.nombre LIKE '%e'**: Filtra los resultados para incluir solo aquellos productos cuyo fabricante tiene un nombre que termina con la letra 'e'. El operador LIKE con el patrón '%e' busca cadenas que finalizan con 'e'.

**Mysql 2**

**SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante**: Similar a la sintaxis SQL1, seleccionamos el nombre y precio del producto, y el nombre del fabricante con alias.

**FROM producto p JOIN fabricante f ON p.codigo\_fabricante = f.codigo**: Realizamos un INNER JOIN explícito entre las tablas producto y fabricante utilizando la condición p.codigo\_fabricante = f.codigo.

**WHERE f.nombre LIKE '%e'**: Filtramos los resultados utilizando el operador LIKE con el patrón '%e' para seleccionar solo los productos de los fabricantes cuyo nombre termina con '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.

**Mysql 1**

**SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto (alias nombre\_producto), el precio del producto (p.precio), y el nombre del fabricante (alias nombre\_fabricante).

**FROM producto p, fabricante f**: Especificamos las tablas producto (alias p) y fabricante (alias f) para el JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Condición de JOIN que relaciona productos con sus fabricantes a través del código.

**AND f.nombre LIKE '%w%'**: Filtra los resultados para incluir solo aquellos productos cuyo fabricante tiene un nombre que contiene la letra 'w' en cualquier posición. El operador LIKE con el patrón '%w%' busca cadenas que contienen 'w'.

**Sintaxis SQL2 (JOIN explícito con ON):**

**Mysql 2**

**SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto (alias nombre\_producto), el precio del producto (p.precio), y el nombre del fabricante (alias nombre\_fabricante).

**FROM producto p, fabricante f**: Especificamos las tablas producto (alias p) y fabricante (alias f) para el JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Condición de JOIN que relaciona productos con sus fabricantes a través del código.

**AND f.nombre LIKE '%w%'**: Filtra los resultados para incluir solo aquellos productos cuyo fabricante tiene un nombre que contiene la letra 'w' en cualquier posición. El operador LIKE con el patrón '%w%' busca cadenas que contienen 'w'.

**Sintaxis SQL2 (JOIN explícito con ON):**

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)

**Mysql 1**

**SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del producto (alias nombre\_producto), el precio del producto (p.precio), y el nombre del fabricante (alias nombre\_fabricante).

**FROM producto p, fabricante f**: Especificamos las tablas producto (alias p) y fabricante (alias f) para el JOIN implícito.

**WHERE p.codigo\_fabricante = f.codigo**: Condición de JOIN que relaciona productos con sus fabricantes a través del código.

**AND p.precio >= 180**: Filtra los resultados para incluir solo los productos cuyo precio sea mayor o igual a 180€.

**ORDER BY p.precio DESC, p.nombre ASC**: Ordena el resultado:

* Primero por la columna p.precio en orden descendente (DESC).
* Segundo por la columna p.nombre en orden ascendente (ASC) dentro de los grupos de productos con el mismo precio.

**Mysql 2**

SELECT p.nombre AS nombre\_producto, p.precio, f.nombre AS nombre\_fabricante

FROM producto p JOIN fabricante f ON p.codigo\_fabricante = f.codigo`\*\*: Realizamos un `INNER JOIN` explícito entre las tablas `producto` y `fabricante` utilizando la condición `p.codigo\_fabricante = f.codigo`.

WHERE p.precio >= 180`\*\*: Filtramos los resultados para incluir solo los productos cuyo precio sea mayor o igual a 180€.

ORDER BY p.precio DESC, p.nombre ASC`\*\*: Ordena el resultado de la misma manera que en la sintaxis SQL1: primero por precio descendente y luego por nombre ascendente.

Ambas consultas devolverán el mismo resultado, mostrando el nombre del producto, su precio y el nombre del fabricante para aquellos productos que cuestan 180€ o más, ordenados según lo especificado.

13. Devuelve un listado con el identificador y el nombre de fabricante, solamente de

aquellos fabricantes que tienen productos asociados en la base de datos.

**Mysql 1**

**SELECT DISTINCT f.codigo AS codigo\_fabricante, f.nombre AS nombre\_fabricante**: Seleccionamos el código del fabricante (alias codigo\_fabricante) y el nombre del fabricante (alias nombre\_fabricante) de la tabla fabricante (alias f). Utilizamos DISTINCT para asegurarnos de que cada fabricante aparezca solo una vez en el resultado, incluso si tiene múltiples productos asociados.

**FROM fabricante f, producto p**: Especificamos las tablas fabricante (alias f) y producto (alias p) para el JOIN implícito.

**WHERE f.codigo = p.codigo\_fabricante**: Esta es la condición de JOIN. Vincula las filas de las dos tablas donde la columna codigo (clave primaria de fabricante) coincide con la columna codigo\_fabricante (clave foránea de producto). Esto asegura que solo se consideren los fabricantes que tienen al menos un producto en la tabla producto.

**Mysql 2**

**SELECT DISTINCT f.codigo AS codigo\_fabricante, f.nombre AS nombre\_fabricante**: Al igual que en la sintaxis SQL1, seleccionamos el código y el nombre del fabricante, utilizando DISTINCT para evitar duplicados.

**FROM fabricante f INNER JOIN producto p ON f.codigo = p.codigo\_fabricante**: Realizamos un INNER JOIN explícito entre las tablas fabricante (alias f) y producto (alias p). La condición para la unión se especifica con ON f.codigo = p.codigo\_fabricante. Un INNER JOIN devuelve solo las filas donde hay una coincidencia en ambas tablas según la condición especificada.

**1.1.5 Consultas multitabla (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 f.nombre AS nombre\_fabricante, p.nombre AS nombre\_producto**: Seleccionamos el nombre del fabricante de la tabla fabricante (alias f) y el nombre del producto de la tabla producto (alias p). Utilizamos AS para darles nombres más descriptivos a las columnas en el resultado.

**FROM fabricante f**: Comenzamos seleccionando todos los registros de la tabla fabricante. Esta será la tabla de la izquierda en nuestro LEFT JOIN.

**LEFT JOIN producto p ON codigo = p.codigo\_fabricante**: Realizamos un LEFT JOIN con la tabla producto. La condición de unión (ON) especifica que queremos relacionar las filas donde el codigo del fabricante en la tabla fabricante coincida con el codigo\_fabricante en la tabla producto.

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

**SELECT f.nombre AS nombre\_fabricante**: Seleccionamos el nombre del fabricante de la tabla fabricante.

**FROM fabricante f LEFT JOIN producto p ON f.codigo = p.codigo\_fabricante**: Realizamos el mismo LEFT JOIN que en la consulta anterior para relacionar fabricantes y productos.

**WHERE p.nombre IS NULL**: Esta cláusula WHERE filtra los resultados. Solo se incluirán en el listado aquellos fabricantes para los cuales la columna p.nombre (el nombre del producto) es NULL. Esto indica que no se encontró ningún producto asociado a ese fabricante en la tabla producto.

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

Si la base de datos tiene implementadas correctamente las restricciones de clave foránea, esta consulta no debería devolver ningún resultado. Esto confirmaría que todos los productos están efectivamente relacionados con un fabricante.

Sin embargo, si por alguna razón existieran inconsistencias en la base de datos (por ejemplo, si las restricciones de clave foránea no se aplicaron correctamente en algún momento), esta consulta mostraría los nombres de aquellos productos que no tienen un fabricante asociado.

**1.1.6 Consultas resumen**

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

COUNT(\*) se aplicó a la tabla producto para obtener el número total de productos.

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

COUNT(\*) se aplicó a la tabla fabricante para obtener el número total de fabricantes.

**3. Calcula el número de valores distintos de identificador de fabricante aparecen en la**

**tabla productos.**

COUNT(DISTINCT) Cuenta el número de valores únicos (distintos) que existen en la columna

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

AVG(columna) Calcula el valor promedio de los valores en la columna especificada

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

MIN(columna) Devuelve el valor mínimo encontrado en la columna especificada

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

MAX(columna) Devuelve el valor máximo encontrado en la columna especificada

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

MIN(columna) Devuelve el valor mínimo encontrado en la columna especificada

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

MAX(columna) Devuelve el valor máximo encontrado en la columna especificada

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

SUM(columna) Calcula la suma de todos los valores en la columna especificada

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

COUNT(\*) Cuenta el número de filas que cumplen con los criterios especificados después de la cláusula WHERE.

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

AVG(columna) Calcula la media (promedio) de los valores en la columna especificada

**Cláusula de unión:** JOIN (INNER JOIN implícito) Combina filas de las tablas producto y fabricante basándose en la relación entre ellas.

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

Función : MIN(columna) Devuelve el valor mínimo de la columna especificada (p.precio en este caso).

Cláusula de unión: JOIN (INNER JOIN implícito) Combina las tablas producto y fabricante basándose en la relación entre ellas.

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

**Función**  MAX(columna) Devuelve el valor máximo de la columna especificada

**Cláusula de unión:** JOIN (INNER JOIN implícito) Combina las tablas producto y fabricante basándose en la relación entre ellas.

**Cláusula de filtro:** WHERE Limita los resultados a los productos del fabricante 'Asus'

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

**Función agregada:** SUM(columna) Calcula la suma de todos los valores en la columna especificada

**Cláusula de unión:** JOIN (INNER JOIN implícito) Combina filas de las tablas producto y fabricante basándose en la relación entre ellas.

**Cláusula de filtro:** WHERE Restringe los resultados a aquellos productos cuyo fabricante es 'Asus'.

**15. Muestra el precio máximo, precio mínimo, precio medio y el número total de**

**productos que tiene el fabricante Crucial.**

**Funciones :** MAX(), MIN(), AVG(), COUNT(\*) **:** Realizan cálculos sobre un conjunto de valores.

**Cláusula de unión:** JOIN (INNER JOIN implícito) Combina las tablas producto y fabricante basándose en la relación entre ellas.

**Cláusula de filtro:** WHERE Limita los resultados a los productos del fabricante

**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 descendentemente por el**

**número de productos.**

**Función :** COUNT(columna)**:** Cuenta el número de valores no nulos en la columna especificada para cada grupo.

**Cláusula de unión:** LEFT JOIN **:** Devuelve todos los registros de la tabla izquierda y los registros coincidentes de la tabla derecha. Si no hay coincidencia, se devuelven valores NULL para las columnas de la tabla derecha.

**Cláusula de agrupación:** GROUP BY **:** Agrupa las filas que tienen los mismos valores en una o más columnas en una fila de resumen.

**Cláusula de ordenamiento:** ORDER BY **:** Ordena el conjunto de resultados de una consulta en función de una o más columnas.

**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.**

**Funciones:** MAX(), MIN(), AVG()**:** Realizan cálculos sobre un conjunto de valores para cada grupo.

**Cláusula de unión:** JOIN (INNER JOIN implícito)**:** Combina las tablas fabricante y producto basándose en la relación entre ellas.

**Cláusula de agrupación:** GROUP BY **:** Agrupa las filas con el mismo nombre de fabricante para aplicar las funciones de agregación a cada grupo.