7. Consultas resumen.

7.1. Introducción.....1
7.2. Funciones de agregación.....1
7.2.1. Contar valores distintos COUNT(DISTINCT columna).....2
7.3. Agrupamiento de filas (GROUP BY).....2
7.4. Condición de agrupamiento (HAVING).....2
7.5. Algunos errores:.....3
7.5.1. Error al contar el número de filas distintas.....3
7.5.2. Error al intentar utilizar una función de agregación en la cláusula WHERE.....3
7.5.3. Error al usar COUNT(*) y COUNT(columna) al hacer un LEFT JOIN.....4

7.1. Introducción.

Recordando la sintaxis para realizar una consulta con la sentencia SELECT de MySQL:

```
SELECT [DISTINCT] expresión_select [, expresión_select …]
    [FROM tabla]
    [WHERE condición_where]
    [GROUP BY {nombre_columna | expresión | posición} [ASC | DESC], … [WITH
        ROLLUP]]
    [HAVING condición_where]
    [ORDER BY {nombre_columna | expresión | posición} [ASC | DESC], …]
    [LIMIT {[offset,] row COUNT | row COUNT OFFSET offset}]
```

Es muy importante conocer **en qué orden se ejecuta cada una de las cláusulas** que forman la sentencia SELECT. El orden de ejecución es el siguiente:

1. Cláusula FROM.
2. Cláusula WHERE (Es opcional, puede ser que no aparezca).
3. Cláusula GROUP BY (Es opcional, puede ser que no aparezca).
4. Cláusula HAVING (Es opcional, puede ser que no aparezca).
5. Cláusula SELECT.
6. Cláusula ORDER BY (Es opcional, puede ser que no aparezca).
7. Cláusula LIMIT (Es opcional, puede ser que no aparezca).

En esta unidad vamos a trabajar con dos nuevas cláusulas GROUP BY y HAVING.

7.2. Funciones de agregación.

Estas funciones realizan una operación específica sobre todas las filas de un grupo, definido por GRUPO BY o en su defecto afectando a todas las filas a mostrar. Las funciones de agregación más comunes son:

MAX(expr)	Valor máximo del grupo
MIN(expr)	Valor mínimo del grupo
SUM(expr)	Suma de todos los valores del grupo
AVG(expr)	Valor medio del grupo
COUNT(*)	Número de filas que tiene el resultado de la consulta incluido NULL
COUNT(columna)	Número de valores no nulos que hay en esa columna

Importante: Las funciones de agregación sólo se pueden usar en las cláusulas SELECT y HAVING.

Las funciones COUNT(*) y COUNT(columna) devolverán resultados diferentes cuando haya valores nulos en la columna que estamos usando en la función.

Ejemplos:
Supongamos que tenemos los siguientes valores en la tabla alumno:

id	nombre	apellido1	apellido2	fecha_nacimiento	es_repetidor	teléfono
1	María	S-anchez	Pérez	1990/12/01	no	NULL
2	Juan	Sáez	Vega	1998/05/02	no	618253876
3	Pepe	Ramírez	Gea	1988/01/03	no	NULL
4	Lucía	López	Ruiz	1993/06/13	sí	678516294

La consulta:

```
mysql>SELECT COUNT(teléfono) FROM alumno;
```

Devolverá el valor 2, solo hay dos teléfonos no nulos. Mientras que la consulta:

```
mysql>SELECT COUNT(*) FROM alumno;
```

Devolverá el valor 4, porque hay cuatro filas que mostrar.

7.2.1. Contar valores distintos COUNT(DISTINCT columna)

Supongamos que tenemos los siguientes valores en la tabla producto:

id	nombre	precio	código_fabricante
1	Disco duro SATA3 1TB	86	5
2	Memoria RAM DDR4 8GB	120	4
3	Disco SSD 1 TB	150	5
4	GeForce GTX 1050Ti	185	5

Y nos piden calcular el número de valores distintos de código de fabricante que aparecen en la tabla producto.

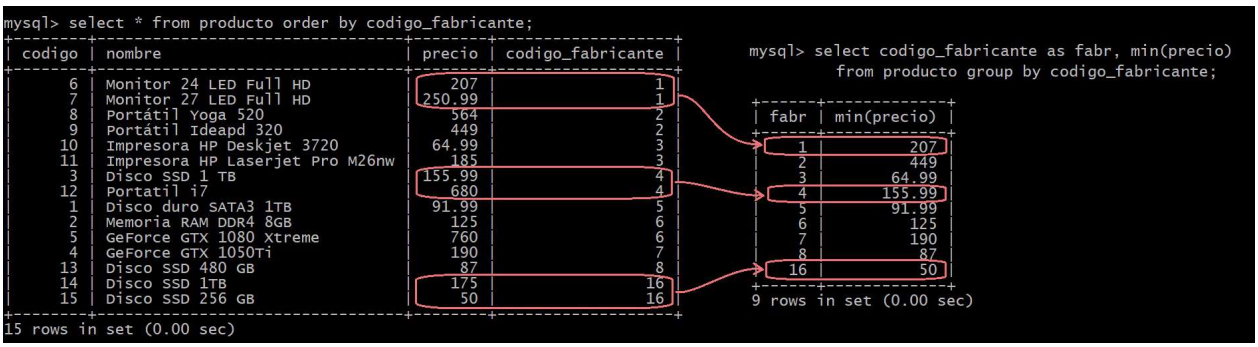
```
mysql>SELECT COUNT(DISTINCT código_fabricante) FROM producto;
```

Esta consulta devolverá: 2

7.3. Agrupamiento de filas (GROUP BY)

La cláusula GROUP BY nos permite crear **grupos de filas** que tienen los mismos valores en las columnas por las que se desea agrupar.

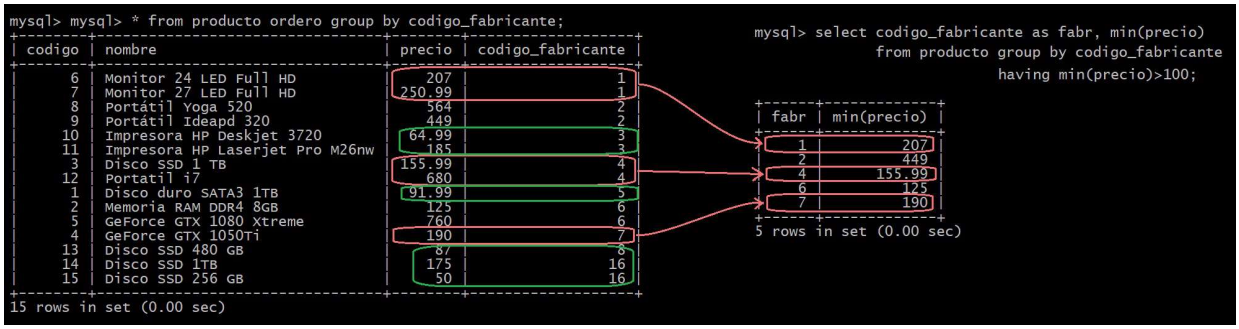
En la siguiente captura se han agrupado los productos de una tienda por código_fabricante y se muestra el producto más barato de cada fabricante: min(precio).



7.4. Condición de agrupamiento (HAVING)

La cláusula HAVING nos permite crear **filtros** sobre los grupos de filas que tienen los mismos valores en las columnas por las que se ha agrupado.

En la siguiente captura se han filtrado los fabricantes que tienen un precio mínimo inferior a 100. Para ello se ha de agrupar primero por código_fabricante y mostrar solo los que tengan un precio mínimo superior a 100.



7.5. Algunos errores:

7.5.1. Error al contar el número de filas distintas

Supongamos que queremos saber el número de proveedores que tenemos. Es decir, el número de códigos de fabricantes distintos que tenemos en la tabla de productos.

```
mysql>SELECT DISTINCT  COUNT(codigo_fabricante) FROM producto;
```

Nos devolvería en total de productos que tenemos, pues contaría el número de filas que tienen codigo_fabricante. Como todos los productos tienen un fabricante, luego el número total de productos.

```
mysql>SELECT COUNT( DISTINCT codigo_fabricante) FROM producto;
```

En este caso solo mostraría los códigos de fabricantes que son distintos. Luego si nos ofrece el total de fabricantes que son proveedores de la tienda.

```
mysql> select * from producto order by codigo_fabricante;
+--------+---------------------------------+--------+--------------------+
| codigo | nombre                          | precio | codigo_fabricante  |
+--------+---------------------------------+--------+--------------------+
|      6 | Monitor 24 LED Full HD          |    207 |                  1 |
|      7 | Monitor 27 LED Full HD          | 250.99 |                  1 |
|      8 | Portátil Yoga 520               |    564 |                  2 |
|      9 | Portátil Ideapd 320             |    449 |                  2 |
|     10 | Impresora HP Deskjet 3720       |  64.99 |                  3 |
|     11 | Impresora HP Laserjet Pro M26nw |    185 |                  3 |
|      3 | Disco SSD 1 TB                  | 155.99 |                  4 |
|     12 | Portatil i7                     |    680 |                  4 |
|      1 | Disco duro SATA3 1TB            |  91.99 |                  5 |
|      2 | Memoria RAM DDR4 8GB            |    125 |                  6 |
|      5 | GeForce GTX 1080 Xtreme         |    760 |                  6 |
|      4 | GeForce GTX 1050Ti              |    190 |                  7 |
|     13 | Disco SSD 480 GB                |     87 |                  8 |
|     14 | Disco SSD 1TB                   |    175 |                 16 |
|     15 | Disco SSD 256 GB                |     50 |                 16 |
+--------+---------------------------------+--------+--------------------+
15 rows in set (0.00 sec)

mysql> SELECT DISTINCT COUNT(codigo_fabricante) FROM producto;
+---------------------------+
| COUNT(codigo_fabricante) |
+---------------------------+
|                        15 |
+---------------------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(DISTINCT codigo_fabricante) FROM producto;
+-----------------------------------+
| COUNT(DISTINCT codigo_fabricante) |
+-----------------------------------+
|                                 9 |
+-----------------------------------+
1 row in set (0.00 sec)
```

7.5.2. Error al intentar utilizar una función de agregación en la cláusula WHERE.

Las funciones de agregación o resumen sólo se pueden usar en las cláusulas SELECT Y HAVING, por lo tanto si intenta hacer uso de una función de agregación en la cláusula WHERE obtendrá un error.

Ejemplo: Devuelve un listado con los productos que tienen un precio superior al precio medio de todos los artículos que existen en la tabla productos.

```
mysql> select nombre, precio from producto where precio>=avg(precio);
ERROR 1111 (HY000): Invalid use of group function
```

Uso inapropiado de función resumen.
AVG() solo se puede usar con:
- SELECT
- HAVING

```
mysql> select nombre, precio from producto
            where precio>=(select avg(precio) from producto);
+-------------------------+--------+
| nombre                  | precio |
+-------------------------+--------+
| GeForce GTX 1080 Xtreme |    760 |
| Portátil Yoga 520       |    564 |
| Portátil Ideapd 320     |    449 |
| Portatil i7             |    680 |
+-------------------------+--------+
4 rows in set (0.00 sec)
```

La única forma de resolver este ejercicio es con una subconsulta. Cuando en una consulta se hace otra consulta. La segunda consulta está entre paréntesis.

7.5.3. Error al usar COUNT(*) y COUNT(columna) al hacer un LEFT JOIN

Realicemos una consulta que devuelva un listado con el número de productos que tiene cada fabricante. El listado debe incluir aquellos fabricantes que no tienen productos asociados indicando que tienen 0 productos.
En este caso es necesario realizar un RIGHT JOIN con las tablas producto y fabricante, de esa forma incluirá los fabricantes que no están relacionados con los articulos.
Por ejemplo, al ejecutar la utilizando count(*) nos aparece que los fabricantes Huawei, Acer, IBM y Xiaomi tienen un productos asociados cada uno, cosa que no es así, pues estos proveedores no tienen ningún producto asociado.
Pero count(*) ha contalo las filas en las que p.codigo_fabricante es NULL.

```
mysql> select f.nombre, count(*)
       from producto as p right join fabricante as f on p.codigo_fabricante=f.codigo
       group by f.codigo;
+-----------------+----------+
| nombre          | count(*) |
+-----------------+----------+
| Asus            |        2 |
| Lenovo          |        2 |
| Hewlett-Packard |        2 |
| Samsung         |        2 |
| Seagate         |        1 |
| Crucial         |        2 |
| Gigabyte        |        1 |
| Apple           |        1 |
| Xiaomi          |        1 |
| IBM             |        1 |
| Acer            |        1 |
| Dell            |        2 |
| Huawei          |        1 |
+-----------------+----------+
13 rows in set (0.00 sec)
```

Sin embargo, si utilizamos count(p.codigo_fabricante) en este caso no cuenta las filas NULL por lo que aparecen con 0 productos.

```
mysql> select f.nombre, count(p.codigo_fabricante)
       from producto as p right join fabricante as f on p.codigo_fabricante=f.codigo
       group by f.codigo;
+-----------------+----------------------------+
| nombre          | count(p.codigo_fabricante) |
+-----------------+----------------------------+
| Asus            |                          2 |
| Lenovo          |                          2 |
| Hewlett-Packard |                          2 |
| Samsung         |                          2 |
| Seagate         |                          1 |
| Crucial         |                          2 |
| Gigabyte        |                          1 |
| Apple           |                          1 |
| Xiaomi          |                          0 |
| IBM             |                          0 |
| Acer            |                          0 |
| Dell            |                          2 |
| Huawei          |                          0 |
+-----------------+----------------------------+
13 rows in set (0.00 sec)
```

Para contar el número de productos que tiene cada fabricante es necesario realizar un GROUP BY por el código del fabricante y contar el número de filas que tiene cada uno de los grupos que hemos creado. Pero debemos tener cuidado porque obtendremos resultados diferentes al contar el número de filas de cada grupo con COUNT (*) y COUNT(p.codigo_fabricante). La opción correcta es hacer uso de COUNT(p.codigo_fabricante) porque contará aquellas filas que tienen un valor distinto de NULL en la columna p.codigo_fabricante.