

Consultas de Funciones agrupadas

BASE DE DATOS LIBRERÍA

1. Muestra cuántos libros tienen el campo precio asignado:

```
mysql> select count(precio) from libros;
+-----+
| count(precio) |
+-----+
|             10 |
+-----+
1 row in set (0,00 sec)
```

2. Muestra la cantidad total de libros que no tienen precio asignado:

```
mysql> select count(*) from libros where precio is null;
+-----+
| count(*) |
+-----+
|         2 |
+-----+
1 row in set (0,00 sec)
```

3. Muestra el libro más caro (solo aparece el precio sin mas datos):

```
mysql> select max(precio) from libros;
+-----+
| max(precio) |
+-----+
|          16 |
+-----+
1 row in set (0,00 sec)
```

4. Muestra el libro más barato (solo aparece el precio sin mas datos):

```
mysql> select min(precio) from libros;
+-----+
| min(precio) |
+-----+
|           8 |
+-----+
1 row in set (0,00 sec)
```

5.Muestra el precio medio de los libros cuyo autor es "J.K. Rowling"

```
mysql> select avg(precio) from libros where autor="JK Rowling";
+-----+
| avg(precio) |
+-----+
|      15.3333 |
+-----+
1 row in set (0,00 sec)
```

Group by: Escribe entre select y from, la funcion de agrupamiento que te pide la consulta (min, max, count...) y el campo por el que agrupas, es este caso autor, por ejemplo, SELECT autor, min(precio) FROM libros GROUP BY autor;

1.Muestra la cantidad de libros de cada autor:

```
mysql> select count(*),autor from libros group by autor;
+-----+-----+
| count(*) | autor          |
+-----+-----+
|         4 | JK Rowling     |
|         2 | Katherin Neville |
|         1 | William Moulton |
|         1 | Lewis Carroll  |
|         2 | Ken Follett    |
|         2 | Paolo Coelho   |
+-----+-----+
6 rows in set (0,00 sec)
```

2.Muestra el precio total de los libros de cada autor:

```
mysql> select sum(precio),autor from libros group by autor;
+-----+-----+
| sum(precio) | autor          |
+-----+-----+
|          46 | JK Rowling     |
|          20 | Katherin Neville |
|          10 | William Moulton |
|          11 | Lewis Carroll  |
|          20 | Ken Follett    |
|          11 | Paolo Coelho   |
+-----+-----+
6 rows in set (0,00 sec)
```

3.Muestra el precio máximo de los libros de cada autor:

```
mysql> select max(precio),autor from libros group by autor;
```

```
+-----+-----+
| max(precio) | autor          |
+-----+-----+
|          16 | JK Rowling     |
|          10 | Katherin Neville |
|          10 | William Moulton |
|          11 | Lewis Carroll  |
|          12 | Ken Follett    |
|          11 | Paolo Coelho   |
+-----+-----+
6 rows in set (0,00 sec)
```

4.Muestra el precio medio de los libros de cada autor:

```
mysql> select avg(precio),autor from libros group by autor;
```

```
+-----+-----+
| avg(precio) | autor          |
+-----+-----+
|      15.3333 | JK Rowling     |
|      10.0000 | Katherin Neville |
|      10.0000 | William Moulton |
|      11.0000 | Lewis Carroll  |
|      10.0000 | Ken Follett    |
|      11.0000 | Paolo Coelho   |
+-----+-----+
6 rows in set (0,01 sec)
```

BASE DE DATOS GALERIA

1.Muestra el precio máximo de las pinturas que haya comprado el cliente
"12345678Q":

```
mysql> select max(precio) from pintura where dni_cliente="12345678Q";
```

```
+-----+
| max(precio) |
+-----+
|      300.00 |
+-----+
1 row in set (0,00 sec)
```

2.Muestra el precio medio de las pinturas de cada artista:

```
mysql> select dni_artista,avg(precio) from pintura group by dni_artista;
+-----+-----+
| dni_artista | avg(precio) |
+-----+-----+
| 11111111A   | 216.000000  |
| 22222222B   | 187.500000  |
| 33333333C   | 233.000000  |
| 44444444D   | 458.333333  |
+-----+-----+
4 rows in set (0,00 sec)
```

3.Muestra el precio final total que ha pagado cada cliente por la compra de pinturas:

```
mysql> select dni_cliente, sum(preciofinal) from pintura group by dni_cliente;
+-----+-----+
| dni_cliente | sum(preciofinal) |
+-----+-----+
| 12345678Q   | 745 |
| 12345678W   | 250 |
| 12345678Y   | 250 |
| 88888888X   | 1051 |
+-----+-----+
4 rows in set (0,00 sec)
```

4.Muestra el dinero total que ha ganado cada artista por la venta de sus pinturas:

```
mysql> select dni_artista,sum(preciofinal) from pintura group by dni_artista;
+-----+-----+
| dni_artista | sum(preciofinal) |
+-----+-----+
| 11111111A   | 745 |
| 22222222B   | 250 |
| 33333333C   | 200 |
| 44444444D   | 1101 |
+-----+-----+
4 rows in set (0,00 sec)
```

BASE DE DATOS HORARIOS

1.Muestra cuántas asignaturas imparte cada profesor:

```
mysql> select count(nombre), nomprofesor from asignatura group by nomprofesor;
```

```
+-----+-----+
| count(nombre) | nomprofesor |
+-----+-----+
|             1 | Profesor1   |
|             1 | Profesor2   |
|             4 | Profesor3   |
|             2 | Profesor4   |
|             1 | Profesor5   |
+-----+-----+
5 rows in set (0,00 sec)
```

2.Muestra la cantidad de aulas que hay en el centro:

```
mysql> select count(numaula) from aula;
```

```
+-----+
| count(numaula) |
+-----+
|             9 |
+-----+
1 row in set (0,00 sec)
```

3.Muestra las aulas que tengan la capacidad máxima y mínima:

```
mysql> select max(capacidad), min(capacidad) from aula;
```

```
+-----+-----+
| max(capacidad) | min(capacidad) |
+-----+-----+
|             95 |             40 |
+-----+-----+
1 row in set (0,00 sec)
```

4.Muestra la cantidad de profesores que hay en cada departamento:

```
mysql> select departamento, count(nombre) from profesor group by departamento;
```

```
+-----+-----+
| departamento | count(nombre) |
+-----+-----+
| Informática  |             3 |
| Matemáticas  |             1 |
| Idiomas      |             1 |
+-----+-----+
3 rows in set (0,00 sec)
```