

Actividad 19

- Lista los clientes que han generado pagos con un monto total mayor a \$20

```
SELECT c.customer_id,concat(c.first_name," ",c.last_name) as nombre,  
sum(p.amount) as total  
FROM payment p  
LEFT JOIN customer c on p.customer_id=c.customer_id  
GROUP BY customer_id  
HAVING total>20  
ORDER BY total
```

The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT c.customer_id,concat(c.first_name," ",c.last_name) as nombre, sum(p.amount) as total FROM payment p LEFT JOIN customer c on p.customer_id=c.customer_id GROUP BY customer_id HAVING total>20 ORDER BY total`. The results table has three columns: customer_id, nombre, and total. It lists 17 customers with their total payment amounts, sorted in descending order. The first customer is JEFFERY PINGSON with a total of 20.92.

customer_id	nombre	total
416	JEFFERY PINGSON	20.92
161	GERALDINE PINKINS	20.93
138	CLAUDE HESZOG	20.94
83	LOUISE JENKINS	20.94
120	SYLVA KOTZ	20.95
46	CATHERINE CAMPBELL	20.95
589	TRACY HERRMANN	20.95
59	CHERYL MURPHY	20.95
272	MAY CALDWELL	20.95
167	SALLY PIERCE	20.95
551	CLAYTON BARBEE	20.95
561	DAN STELL	20.95
178	MARION SWIDER	20.96
386	TODD TAN	20.96
176	KYNG ANKRETH	20.96

- Liste los empleados que han generado pagos con un monto total mayor a \$100

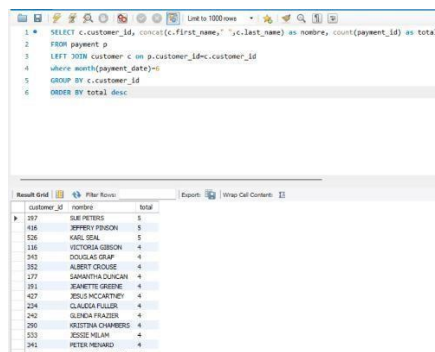
```
SELECT s.staff_id,concat(s.first_name," ",s.last_name) as nombre,  
sum(p.amount) as total  
FROM payment p  
LEFT JOIN staff s on p.staff_id=s.staff_id  
GROUP BY staff_id  
HAVING total>100
```

The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT s.staff_id,concat(s.first_name," ",s.last_name) as nombre, sum(p.amount) as total FROM payment p LEFT JOIN staff s on p.staff_id=s.staff_id GROUP BY staff_id HAVING total>100`. The results table has three columns: staff_id, nombre, and total. It lists two employees: Mike Hillyer with a total of 4281.66 and Jon Stephens with a total of 3935.35.

staff_id	nombre	total
1	Mike Hillyer	4281.66
2	Jon Stephens	3935.35

- Calcule el numero de pagos realizados por cada cliente en el mes de junio

```
SELECT c.customer_id, concat(c.first_name,"",c.last_name) as nombre,
count(payment_id) as total
FROM payment p
LEFT JOIN customer c on p.customer_id=c.customer_id
where month(payment_date)=6
GROUP BY c.customer_id
ORDER BY total desc
```



The screenshot shows a SQL query editor with the following query:

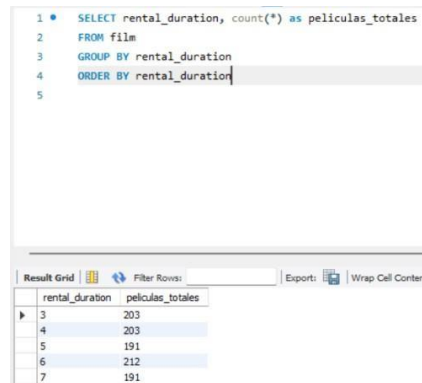
```
1 SELECT c.customer_id, concat(c.first_name,"",c.last_name) as nombre, count(payment_id) as total
2 FROM payment p
3 LEFT JOIN customer c on p.customer_id=c.customer_id
4 where month(payment_date)=6
5 GROUP BY c.customer_id
6 ORDER BY total desc
```

The results are displayed in a table grid with the following data:

customer_id	nombre	total
107	SLR PETERS	5
416	JEFFERY PERSON	5
526	KARL SEAL	5
116	VICTORIA GIBSON	4
343	DOUGLAS GRAP	4
352	ALBERT GROSSE	4
177	SARANTHA DUNCAN	4
191	JEANETTE GREENE	4
427	JOSE MC CARTNEY	4
224	CLAUSIA FULLER	4
242	OLIVERA FRADDER	4
260	KRISTINA CHAMBERS	4
533	JOSIE NELAN	4
341	PETER HOWARD	4

- Calcule la cantidad de películas por cada duración de renta (rental_duration)

```
SELECT rental_duration, count(*) as peliculas_totales
FROM film
GROUP BY rental_duration
ORDER BY rental_duration
```



The screenshot shows a SQL query editor with the following query:

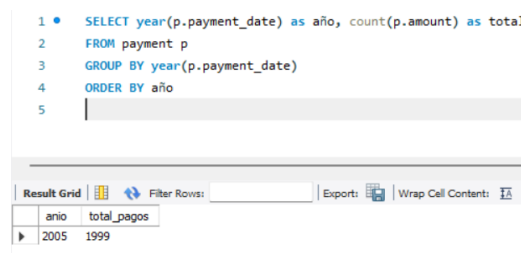
```
1 SELECT rental_duration, count(*) as peliculas_totales
2 FROM film
3 GROUP BY rental_duration
4 ORDER BY rental_duration
5
```

The results are displayed in a table grid with the following data:

rental_duration	peliculas_totales
3	203
4	203
5	191
6	212
7	191

- Calcule la cantidad total de pagos recibidos cada año

```
SELECT year(p.payment_date) as año,
count(p.amount) as total
FROM payment p
GROUP BY year(p.payment_date)
ORDER BY año
```

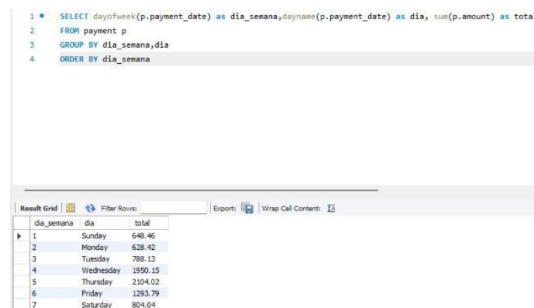


```
1 • SELECT year(p.payment_date) as año, count(p.amount) as total
2 FROM payment p
3 GROUP BY year(p.payment_date)
4 ORDER BY año
5
```

año	total_pagos
▶ 2005	1999

- Calcule el monto total de pagos por cada día de la semana

```
SELECT dayofweek(p.payment_date) as
dia_semana,dayname(p.payment_date) as dia, sum(p.amount) as total
FROM payment p
GROUP BY dia_semana,dia
ORDER BY dia_semana
```

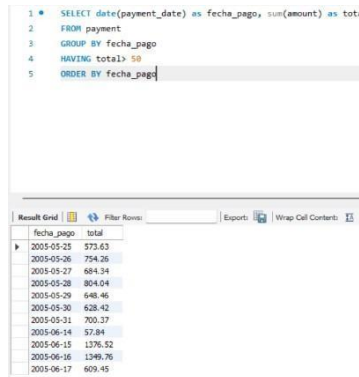


```
1 • SELECT dayofweek(p.payment_date) as dia_semana,dayname(p.payment_date) as dia, sum(p.amount) as total
2 FROM payment p
3 GROUP BY dia_semana,dia
4 ORDER BY dia_semana
```

dia_semana	dia	total
▶ 1	Sunday	648.46
2	Monday	628.42
3	Tuesday	788.13
4	Wednesday	950.15
5	Thursday	2104.02
6	Friday	1293.79
7	Saturday	804.04

- Liste los días donde el total de pagos fue mayor a \$50

```
SELECT date(payment_date) as fecha_pago, sum(amount) as total
FROM payment
GROUP BY fecha_pago
HAVING total > 50
ORDER BY fecha_pago
```



The screenshot shows a SQL query editor with the following query:

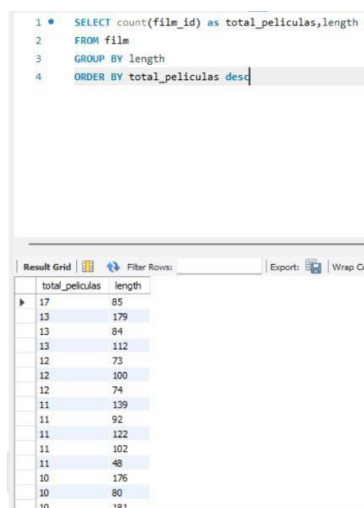
```
1 • SELECT date(payment_date) as fecha_pago, sum(amount) as total
2 FROM payment
3 GROUP BY fecha_pago
4 HAVING total > 50
5 ORDER BY fecha_pago
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'fecha_pago' and 'total'. The results are as follows:

fecha_pago	total
2005-05-25	573.63
2005-05-26	754.26
2005-05-27	684.34
2005-05-28	804.04
2005-05-29	646.46
2005-05-30	628.42
2005-05-31	700.37
2005-06-01	57.84
2005-06-02	1176.52
2005-06-03	1349.76
2005-06-04	609.45

- Calcule cuantas películas hay en cada duración(length) específica

```
SELECT count(film_id) as total_peliculas, length
FROM film
GROUP BY length
ORDER BY total_peliculas desc
```



The screenshot shows a SQL query editor with the following query:

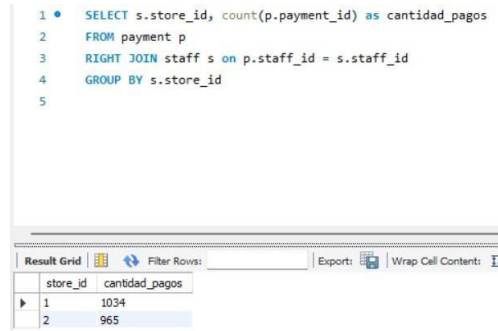
```
1 • SELECT count(film_id) as total_peliculas, length
2 FROM film
3 GROUP BY length
4 ORDER BY total_peliculas desc
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'total_peliculas' and 'length'. The results are as follows:

total_peliculas	length
17	85
13	179
13	84
13	112
12	73
12	100
12	74
11	139
11	92
11	122
11	102
11	48
10	176
10	80
10	181

- Calcule la cantidad de pagos realizados en cada tienda

```
SELECT s.store_id, count(p.payment_id) as cantidad_pagos
FROM payment p
RIGHT JOIN staff s on p.staff_id = s.staff_id
GROUP BY s.store_id
```

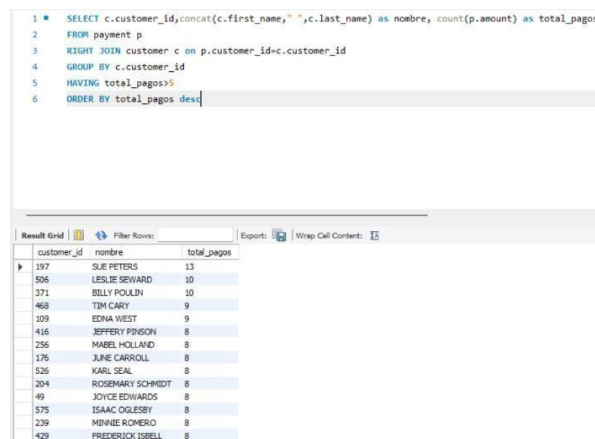


```
1 • SELECT s.store_id, count(p.payment_id) as cantidad_pagos
2 FROM payment p
3 RIGHT JOIN staff s on p.staff_id = s.staff_id
4 GROUP BY s.store_id
5
```

store_id	cantidad_pagos
1	1034
2	965

- Calcule el monto total de pagos por cada cliente, pero solo para aquellos que han realizado mas de 5 pagos

```
SELECT c.customer_id, concat(c.first_name, " ", c.last_name) as nombre,
count(p.amount) as total_pagos
FROM payment p
RIGHT JOIN customer c on p.customer_id=c.customer_id
GROUP BY c.customer_id
HAVING total_pagos>5
ORDER BY total_pagos desc
```

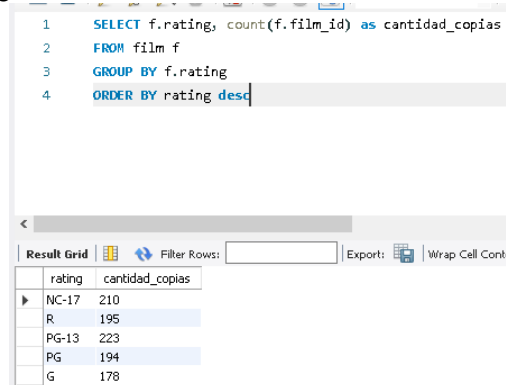


```
1 • SELECT c.customer_id, concat(c.first_name, " ", c.last_name) as nombre, count(p.amount) as total_pagos
2 FROM payment p
3 RIGHT JOIN customer c on p.customer_id=c.customer_id
4 GROUP BY c.customer_id
5 HAVING total_pagos>5
6 ORDER BY total_pagos desc
```

customer_id	nombre	total_pagos
197	SUE PETERS	13
906	LESLIE SHWARD	10
371	BILLY POULIN	10
468	TIM CARY	9
109	EDNA WEST	9
416	JEFFERY PINSON	8
256	HABEL HOLLAND	8
176	JUNE CARROLL	8
526	KARL SEAL	8
204	ROSEMARY SCHMIDT	8
49	JOYCE EDWARDS	8
575	ISAAC OULESEY	8
229	MIRNIE ACKERD	8
429	FREDERICK ISRELL	8

- Calcule el número de películas (puede haber más de 1 película por cada título) según su clasificación (rating)

```
SELECT f.rating, count(f.film_id) as cantidad_copias
FROM film f
GROUP BY f.rating
ORDER BY rating desc
```



The screenshot shows a SQL query editor with the following query:

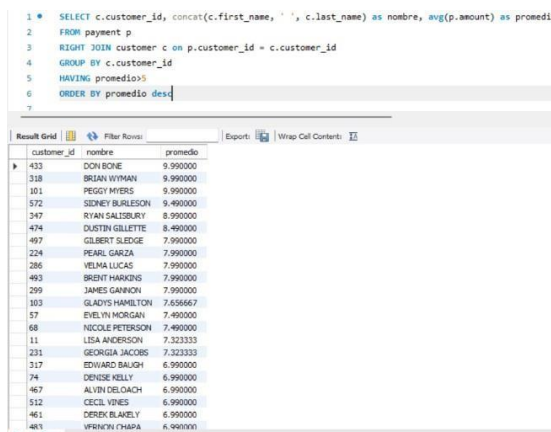
```
1 SELECT f.rating, count(f.film_id) as cantidad_copias
2 FROM film f
3 GROUP BY f.rating
4 ORDER BY rating desc
```

Below the query, a 'Result Grid' is displayed with the following data:

rating	cantidad_copias
NC-17	210
R	195
PG-13	223
PG	194
G	178

- Liste los clientes que han realizado pagos con un monto promedio mayor a \$5

```
SELECT c.customer_id, concat(c.first_name, ' ', c.last_name) as nombre,
avg(p.amount) as promedio
FROM payment p
RIGHT JOIN customer c on p.customer_id = c.customer_id
GROUP BY c.customer_id
HAVING promedio > 5
ORDER BY promedio desc
```



The screenshot shows a SQL query editor with the following query:

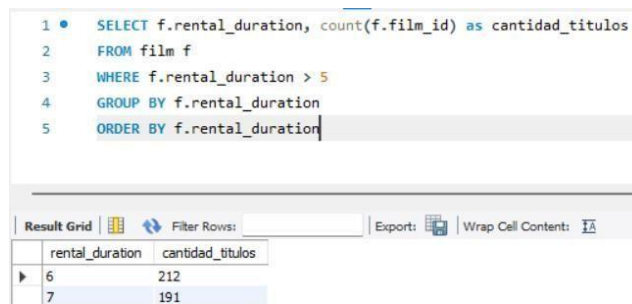
```
1 SELECT c.customer_id, concat(c.first_name, ' ', c.last_name) as nombre, avg(p.amount) as promedio
2 FROM payment p
3 RIGHT JOIN customer c on p.customer_id = c.customer_id
4 GROUP BY c.customer_id
5 HAVING promedio > 5
6 ORDER BY promedio desc
```

Below the query, a 'Result Grid' is displayed with the following data:

customer_id	nombre	promedio
433	DON BONE	9.990000
318	BRIAN WYMAN	9.990000
101	PEGGY MYERS	9.990000
572	SENEE BURGESSON	9.490000
347	RYAN SALISBURY	8.990000
474	DUSTIN GILLETTE	8.490000
497	GILBERT SLEDGE	7.990000
224	PEARL GARZA	7.990000
286	VELMA LUCAS	7.990000
493	BRENT HARKINS	7.990000
299	JAMES GANNON	7.990000
103	GLADYS HAMILTON	7.656667
57	EVELYN MORGAN	7.490000
68	NICOLE PETERSON	7.490000
11	LISA ANDERSON	7.323333
231	GEORGIA JACOBS	7.323333
317	EDWARD BALUGH	6.990000
74	DENISE KELLY	6.990000
467	ALVIN DELOACH	6.990000
512	CECIL VINES	6.990000
461	DEREK BLAKELY	6.990000
481	VPRINCHY THAPA	6.990000

- Calcule la cantidad de títulos de películas en cada duración de renta (rental_duration) mayor a 5 días

```
SELECT f.rental_duration, count(f.film_id) as cantidad_titulos
FROM film f
WHERE f.rental_duration > 5
GROUP BY f.rental_duration
ORDER BY f.rental_duration
```



The screenshot shows a SQL query editor with the following query:

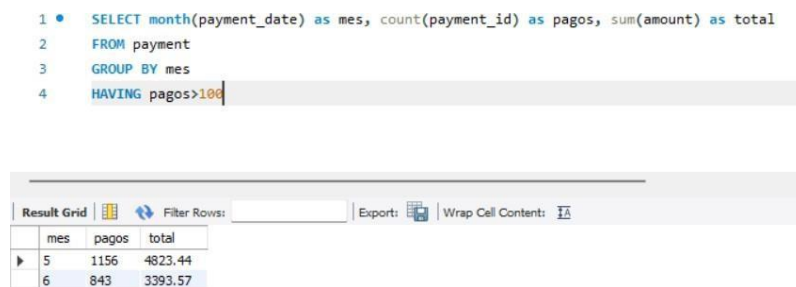
```
1 • SELECT f.rental_duration, count(f.film_id) as cantidad_titulos
2 FROM film f
3 WHERE f.rental_duration > 5
4 GROUP BY f.rental_duration
5 ORDER BY f.rental_duration
```

Below the query editor, there is a 'Result Grid' showing the results of the query:

rental_duration	cantidad_titulos
6	212
7	191

- Liste el total de pagos recibidos por cada mes, pero solo los meses con más de 100 pagos

```
SELECT month(payment_date) as mes, count(payment_id) as pagos, sum(amount) as total
FROM payment
GROUP BY mes
HAVING pagos > 100
```



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT month(payment_date) as mes, count(payment_id) as pagos, sum(amount) as total
2 FROM payment
3 GROUP BY mes
4 HAVING pagos > 100
```

Below the query editor, there is a 'Result Grid' showing the results of the query:

mes	pagos	total
5	1156	4823.44
6	843	3393.57

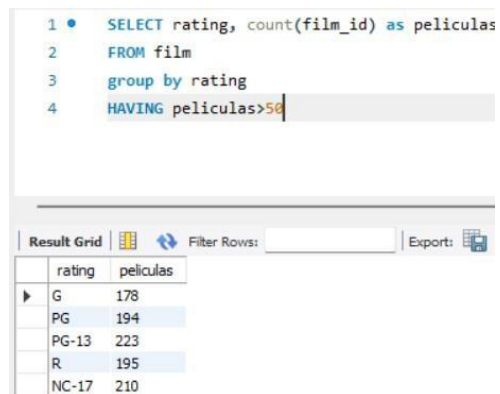
- Liste las clasificaciones de películas(rating) que tienen mas de 50 peliculas registradas

```
SELECT rating, count(film_id) as peliculas
```

```
FROM film
```

```
group by rating
```

```
HAVING peliculas>50
```



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT rating, count(film_id) as peliculas
2 FROM film
3 group by rating
4 HAVING peliculas>50
```

Below the query editor, there is a "Result Grid" section with a table showing the results of the query. The table has two columns: "rating" and "peliculas". The results are as follows:

rating	peliculas
G	178
PG	194
PG-13	223
R	195
NC-17	210