



TECNOLÒGICO NACIONAL DE MÈXICO

CAMPUS JEREZ

MATERIA: TALLER DE BASE DE DATOS

DOCENTE: ISC SALVADOR ACEVEDO SANDOVAL

5° SEMESTRE

TEMA 2: LENGUAJE DE MANIPULACIÓN DE DATOS

ACTIVIDAD 2: EJERCICIOS SQL [CONSULTAS CON
FUNCIONES DE AGREGACIÓN]

ALUMNA: LIZA AREMY SANTANA CONTRERAS

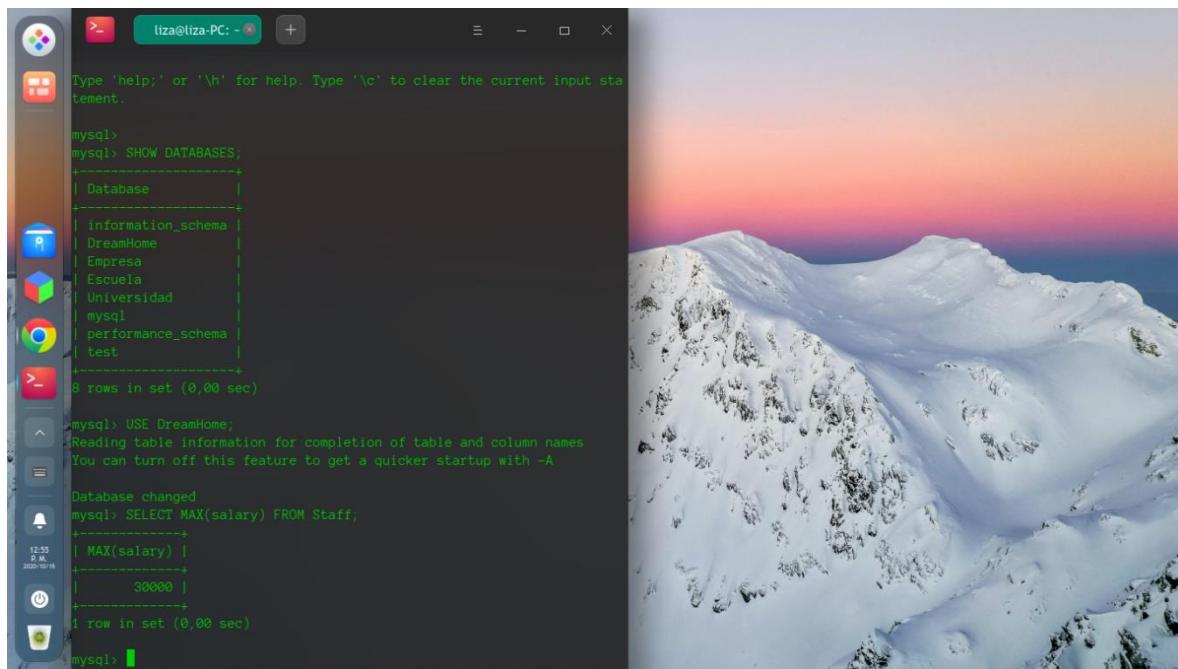
NO. CONTROL: 16070005

16 DE OCTUBRE DE 2020

JEREZ DE GARCIA SALINAS

DREAMHOME - MySQL

1. Mostrar el salario del empleado que gana mas



The terminal window displays the following MySQL session:

```
liza@liza-PC: ~
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| DreamHome |
| Empresa |
| Escuela |
| Universidad |
| mysql |
| performance_schema |
| test |
+-----+
8 rows in set (0,00 sec)

mysql> USE DreamHome;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT MAX(salary) FROM Staff;
+-----+
| MAX(salary) |
+-----+
|      30000 |
+-----+
1 row in set (0,00 sec)

mysql>
```

2. Mostrar el salario del empleado que gana menos



```
liza@liza-PC: ~ + - x
+-----+
| information_schema |
| DreamHome |
| Empresa |
| Escuela |
| Universidad |
| mysql |
| performance_schema |
| test |
+-----+
8 rows in set (0,00 sec)

mysql> USE DreamHome;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT MAX(salary) FROM Staff;
+-----+
| MAX(salary) |
+-----+
| 30000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(salary) FROM Staff;
+-----+
| MIN(salary) |
+-----+
| 9000 |
+-----+
1 row in set (0,00 sec)

mysql>
12:56 21 M 2020-10-16
```

3. Muestre cuales es el promedio del salario que perciben los trabajadores



```
liza@liza-PC: ~ + - x
+-----+
| test |
+-----+
8 rows in set (0,00 sec)

mysql> USE DreamHome;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

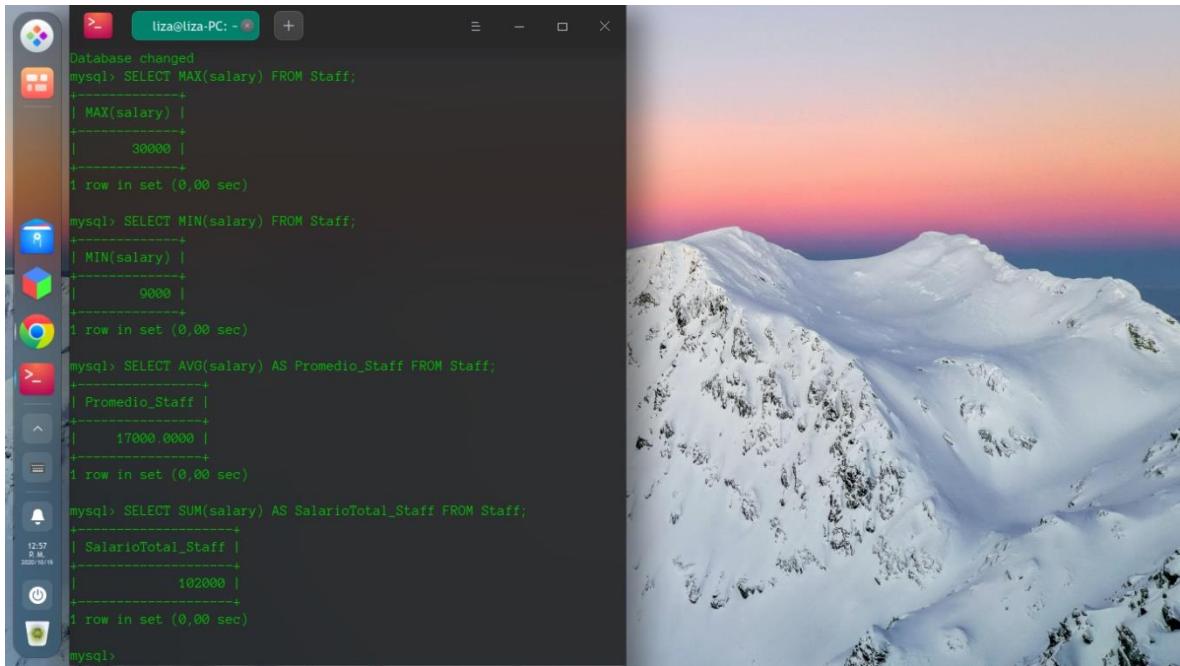
Database changed
mysql> SELECT MAX(salary) FROM Staff;
+-----+
| MAX(salary) |
+-----+
| 30000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(salary) FROM Staff;
+-----+
| MIN(salary) |
+-----+
| 9000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(salary) AS Promedio_Staff FROM Staff;
+-----+
| Promedio_Staff |
+-----+
| 17000.0000 |
+-----+
1 row in set (0,00 sec)

mysql>
```

4. Crear una consulta que muestre la cantidad que gasta la empresa en salarios



The terminal window displays the following MySQL queries and their results:

```
liza@liza-PC: ~
Database changed
mysql> SELECT MAX(salary) FROM Staff;
+-----+
| MAX(salary) |
+-----+
|      30000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(salary) FROM Staff;
+-----+
| MIN(salary) |
+-----+
|      9000 |
+-----+
1 row in set (0,00 sec)

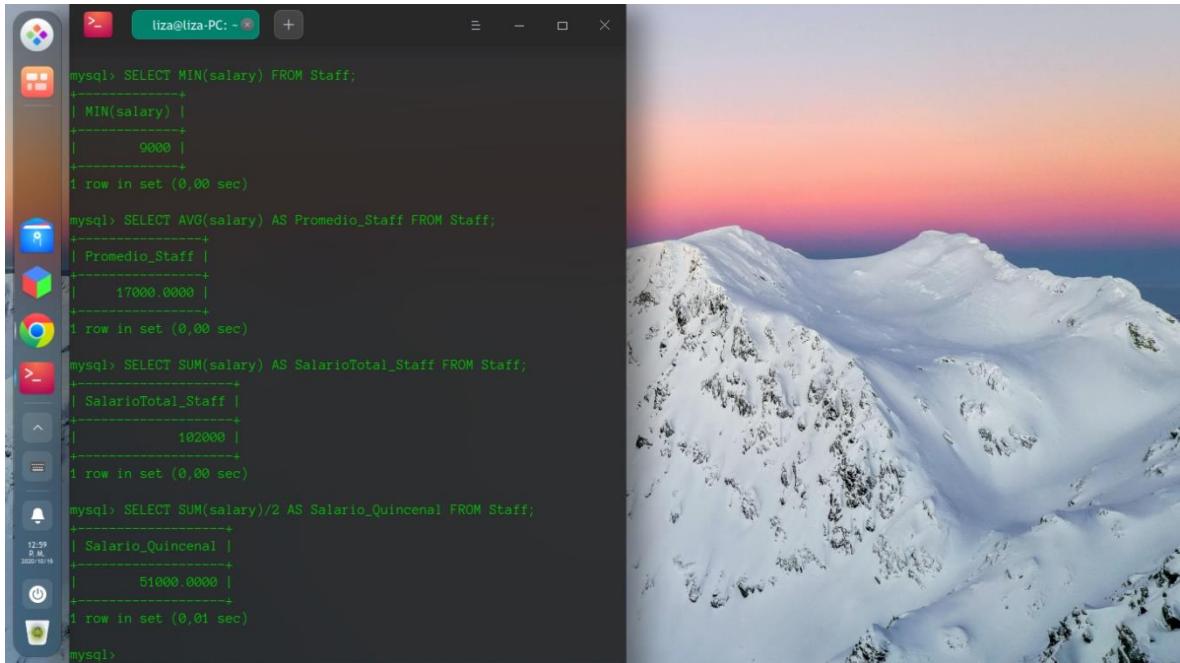
mysql> SELECT AVG(salary) AS Promedio_Staff FROM Staff;
+-----+
| Promedio_Staff |
+-----+
| 17000.0000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
+-----+
| SalarioTotal_Staff |
+-----+
|     102000 |
+-----+
1 row in set (0,00 sec)

12:57
27.10.16
2020-10-16

mysql>
```

5. Crear una consulta que muestre la cantidad que gasta la empresa en salarios quincenales (suponiendo que el dato almacenado es mensual)



The terminal window displays the following MySQL queries and their results, including a new query to calculate bi-weekly salaries:

```
liza@liza-PC: ~
Database changed
mysql> SELECT MIN(salary) FROM Staff;
+-----+
| MIN(salary) |
+-----+
|      9000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(salary) AS Promedio_Staff FROM Staff;
+-----+
| Promedio_Staff |
+-----+
| 17000.0000 |
+-----+
1 row in set (0,00 sec)

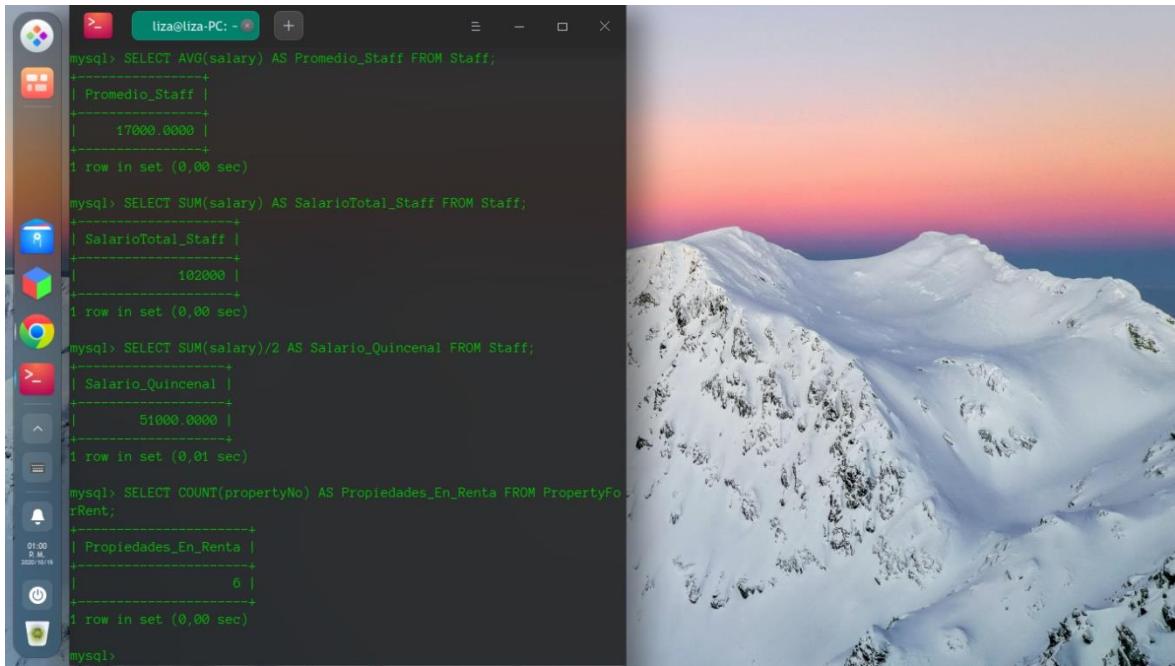
mysql> SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
+-----+
| SalarioTotal_Staff |
+-----+
|     102000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
+-----+
| Salario_Quincenal |
+-----+
| 51000.0000 |
+-----+
1 row in set (0,01 sec)

12:59
27.10.16
2020-10-16

mysql>
```

6. Mostrar cuantas propiedades en renta existen



```
liza@liza-PC: ~
mysql> SELECT AVG(salary) AS Promedio_Staff FROM Staff;
+-----+
| Promedio_Staff |
+-----+
| 17000.0000 |
+-----+
1 row in set (0,00 sec)

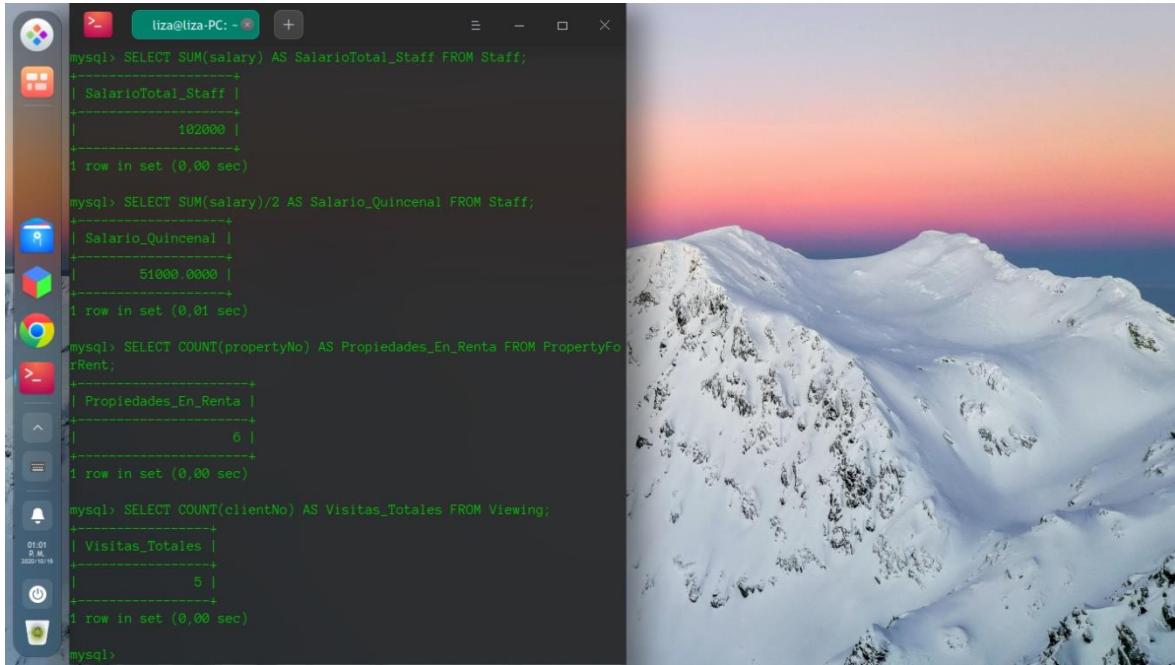
mysql> SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
+-----+
| SalarioTotal_Staff |
+-----+
| 102000 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
+-----+
| Salario_Quincenal |
+-----+
| 51000.0000 |
+-----+
1 row in set (0,01 sec)

mysql> SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;
+-----+
| Propiedades_En_Renta |
+-----+
| 6 |
+-----+
1 row in set (0,00 sec)

mysql>
```

7. Mostrar cuantas visitas a las propiedades se han hecho



```
liza@liza-PC: ~
mysql> SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
+-----+
| SalarioTotal_Staff |
+-----+
| 102000 |
+-----+
1 row in set (0,00 sec)

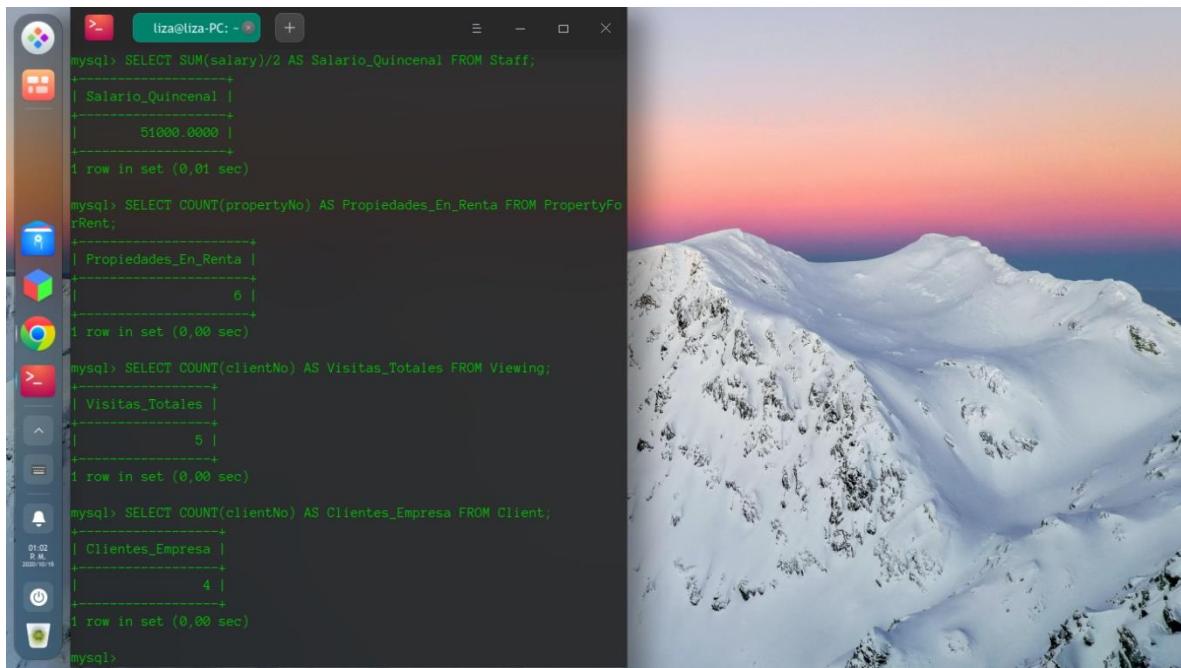
mysql> SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
+-----+
| Salario_Quincenal |
+-----+
| 51000.0000 |
+-----+
1 row in set (0,01 sec)

mysql> SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;
+-----+
| Propiedades_En_Renta |
+-----+
| 6 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT COUNT(clientNo) AS Visitas_Totales FROM Viewing;
+-----+
| Visitas_Totales |
+-----+
| 5 |
+-----+
1 row in set (0,00 sec)

mysql>
```

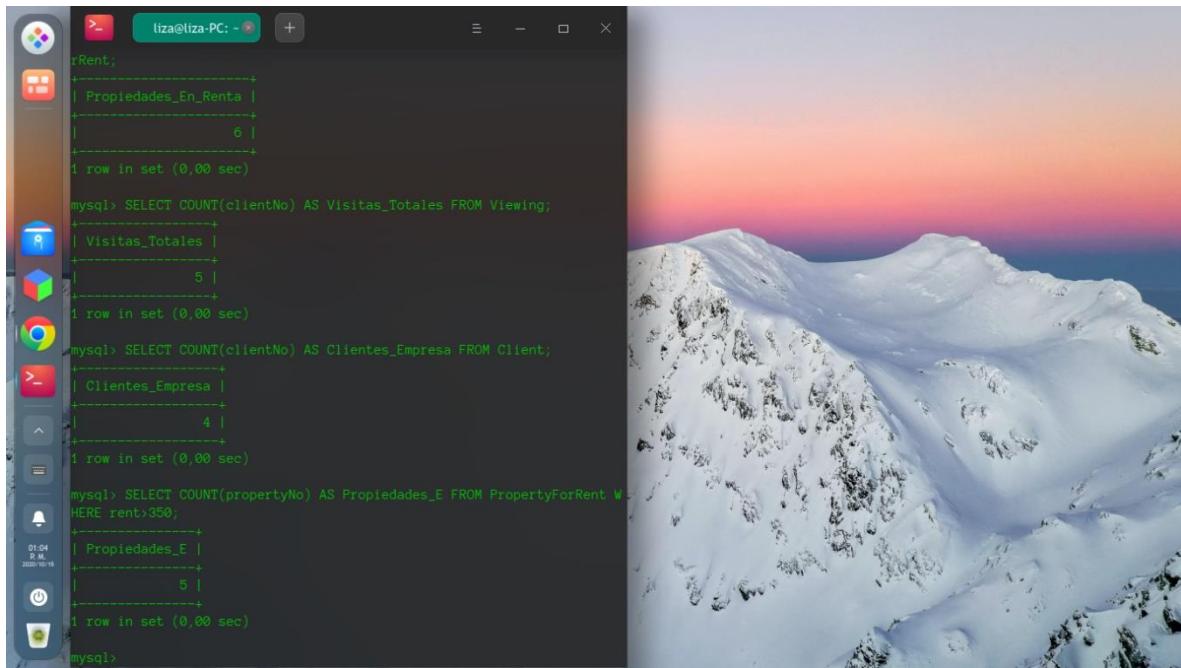
8. Mostrar la cantidad de clientes que atiende la empresa



```
liza@liza-PC: ~ +  
mysql> SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;  
+-----+  
| Salario_Quincenal |  
+-----+  
| 51000.0000 |  
+-----+  
1 row in set (0,01 sec)  
  
mysql> SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;  
+-----+  
| Propiedades_En_Renta |  
+-----+  
| 6 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS Visitas_Totales FROM Viewing;  
+-----+  
| Visitas_Totales |  

```

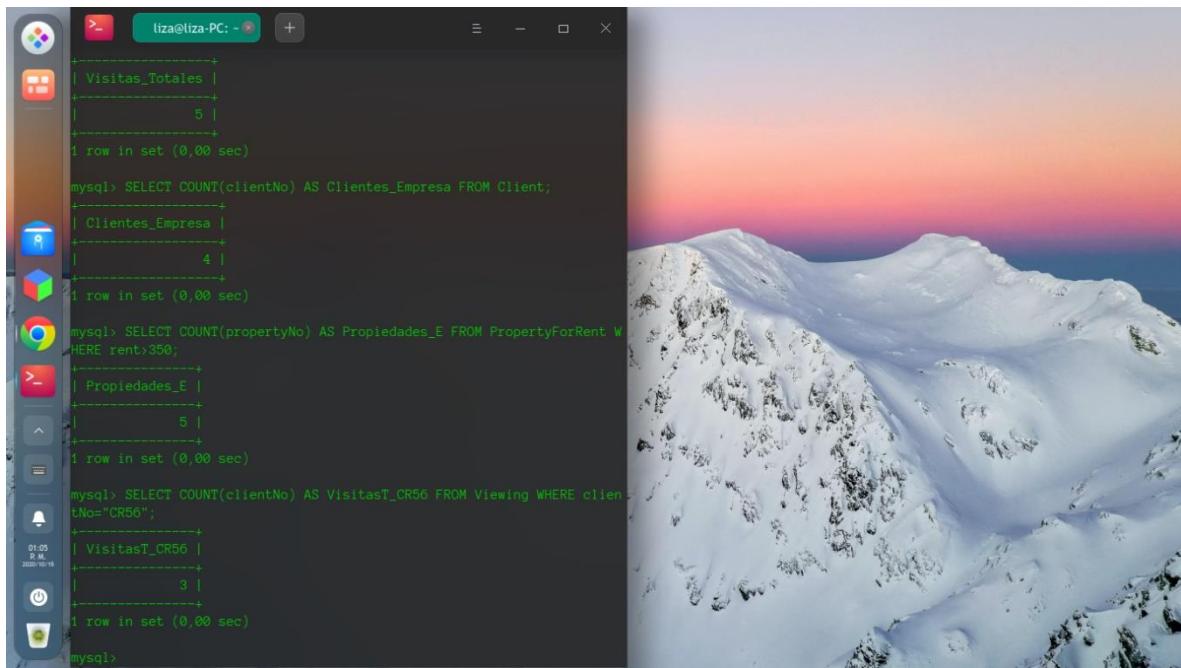
9. Mostrar cuantas propiedades en renta que cuesten mas de 350 euros existen



```
liza@liza-PC: ~ +  
mysql> SELECT COUNT(propertyNo) AS Propiedades_E FROM PropertyForRent WHERE rent>350;  
+-----+  
| Propiedades_E |  
+-----+  
| 5 |  
+-----+  
1 row in set (0,00 sec)  
  

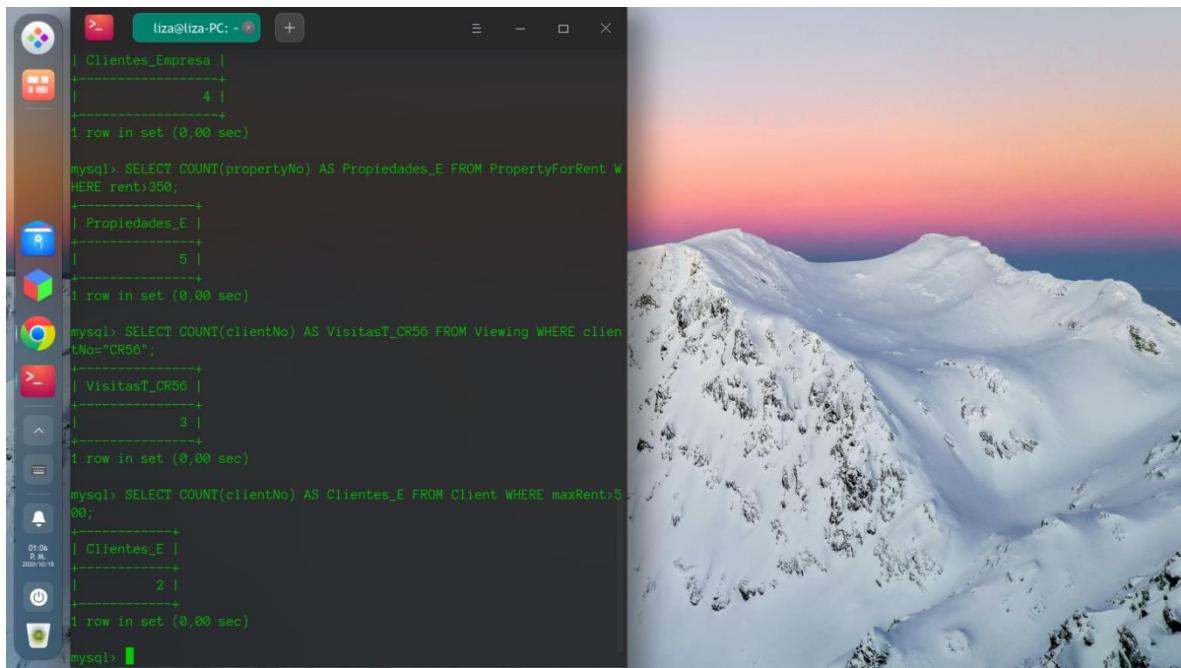
```

10. Mostrar cuantas visitas a la propiedad CR56 se han hecho



```
liza@liza-PC: ~ +  
+-----+  
| Visitas_Totales |  
+-----+  
| 5 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS Clientes_Empresa FROM Client;  
+-----+  
| Clientes_Empresa |  
+-----+  
| 4 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(propertyNo) AS Propiedades_E FROM PropertyForRent WHERE rent>350;  
+-----+  
| Propiedades_E |  
+-----+  
| 5 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo="CR56";  
+-----+  
| VisitasT_CR56 |  
+-----+  
| 3 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql>
```

11. Mostar la cantidad de clientes que puedan pagar una renta mayor a 500 euros atiende la empresa



```
liza@liza-PC: ~ +  
+-----+  
| Clientes_Empresa |  
+-----+  
| 4 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(propertyNo) AS Propiedades_E FROM PropertyForRent WHERE rent>350;  
+-----+  
| Propiedades_E |  
+-----+  
| 5 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo="CR56";  
+-----+  
| VisitasT_CR56 |  
+-----+  
| 3 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;  
+-----+  
| Clientes_E |  
+-----+  
| 2 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql>
```

12. Calcular el promedio de la renta que pueden pagar los clientes



```
liza@liza-PC: ~ +  
+-----+  
| Propiedades_E |  
+-----+  
| 5 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo='CR56';  
+-----+  
| VisitasT_CR56 |  
+-----+  
| 3 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;  
+-----+  
| Clientes_E |  
+-----+  
| 2 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT AVG(maxRent) AS Promedio_Renta FROM Client;  
+-----+  
| Promedio_Renta |  
+-----+  
| 531.2500 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql>
```

13. Mostrar el total de rentas recaudadas al mes



```
liza@liza-PC: ~ +  
+-----+  
| VisitasT_CR56 |  
+-----+  
| 3 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;  
+-----+  
| Clientes_E |  
+-----+  
| 2 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT AVG(maxRent) AS Promedio_Renta FROM Client;  
+-----+  
| Promedio_Renta |  
+-----+  
| 531.2500 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;  
+-----+  
| Rentas_Recaudadas |  
+-----+  
| 2825 |  
+-----+  
1 row in set (0,00 sec)  
  
mysql>
```

14. Mostrar cual es la renta mas cara pagada y cual es la mas barata



```
liza@liza-PC: ~ + ... - x
mysql> SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;
+-----+
| Clientes_E |
+-----+
| 2 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(maxRent) AS Promedio_Renta FROM Client;
+-----+
| Promedio_Renta |
+-----+
| 531.2500 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
+-----+
| Rentas_Recaudadas |
+-----+
| 2825 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
+-----+
| Max_Renta |
+-----+
| 650 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
+-----+
| Min_Renta |
+-----+
| 350 |
+-----+
1 row in set (0,00 sec)

mysql>
```

15. Calcular el promedio de la renta que recibe la empresa



```
liza@liza-PC: ~ + ... - x
mysql> SELECT AVG(maxRent) AS Promedio_Renta FROM Client;
+-----+
| Promedio_Renta |
+-----+
| 531.2500 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
+-----+
| Rentas_Recaudadas |
+-----+
| 2825 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
+-----+
| Max_Renta |
+-----+
| 650 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
+-----+
| Min_Renta |
+-----+
| 350 |
+-----+
1 row in set (0,00 sec)

mysql>
```

16. Mostrar el total de rentas que pueden pagar los clientes al mes



```
liza@liza-PC: ~ + - x
mysql> SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
+-----+
| Rentas_Recaudadas |
+-----+
|      2825      |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
+-----+
| Max_Renta |
+-----+
|      650      |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
+-----+
| Min_Renta |
+-----+
|      350      |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
+-----+
| PromedioR_Mensuales |
+-----+
|    470.8333     |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(maxRent) AS MaximoR_Mensuales FROM Client;
+-----+
| MaximoR_Mensuales |
+-----+
|      2125      |
+-----+
1 row in set (0,00 sec)

mysql>
```

17. Mostrar el total de rentas recaudadas por rentar CASAS



```
liza@liza-PC: ~ + - x
mysql> SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
+-----+
| Max_Renta |
+-----+
|      650      |
+-----+
1 row in set (0,00 sec)

mysql> SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
+-----+
| Min_Renta |
+-----+
|      350      |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
+-----+
| PromedioR_Mensuales |
+-----+
|    470.8333     |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(maxRent) AS MaximoR_Mensuales FROM Client;
+-----+
| MaximoR_Mensuales |
+-----+
|      2125      |
+-----+
1 row in set (0,00 sec)

mysql>
```

The image shows a terminal window on the left and a scenic view of a snow-covered mountain range under a sunset sky on the right.

```

liza@liza-PC: ~
mysql> SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
+-----+
| Min_Renta |
+-----+
|      350 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
+-----+
| PromedioR_Mensuales |
+-----+
|     470.8333 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(maxRent) AS MaximoR_Mensuales FROM Client;
+-----+
| MaximoR_Mensuales |
+-----+
|      2125 |
+-----+
1 row in set (0,00 sec)

mysql> SELECT SUM(rent) AS RentasM_Casas FROM PropertyForRent WHERE type='House';
+-----+
| RentasM_Casas |
+-----+
|      1250 |
+-----+
1 row in set (0,00 sec)

mysql>

```

18. EXPLICAR COMO FUNCIONA max y min UTILIZADO EN CAMPOS VARCHAR

MAX, MIN

Devuelven el mínimo o el máximo de un conjunto de valores contenidos en un campo específico de una consulta. Su sintaxis es:

- Min(expr)
- Max(expr)

Parece que la forma en que MIN () y MAX () trabajan en las cadenas CHAR o VARCHAR es por orden alfabético, donde A es el valor más pequeño y Z es el más grande, y así sucesivamente.

Compara las columnas ENUM y SET por sus valores de cadena, en lugar de sus posiciones relativas dentro del conjunto.

DREAMHOME – DB2

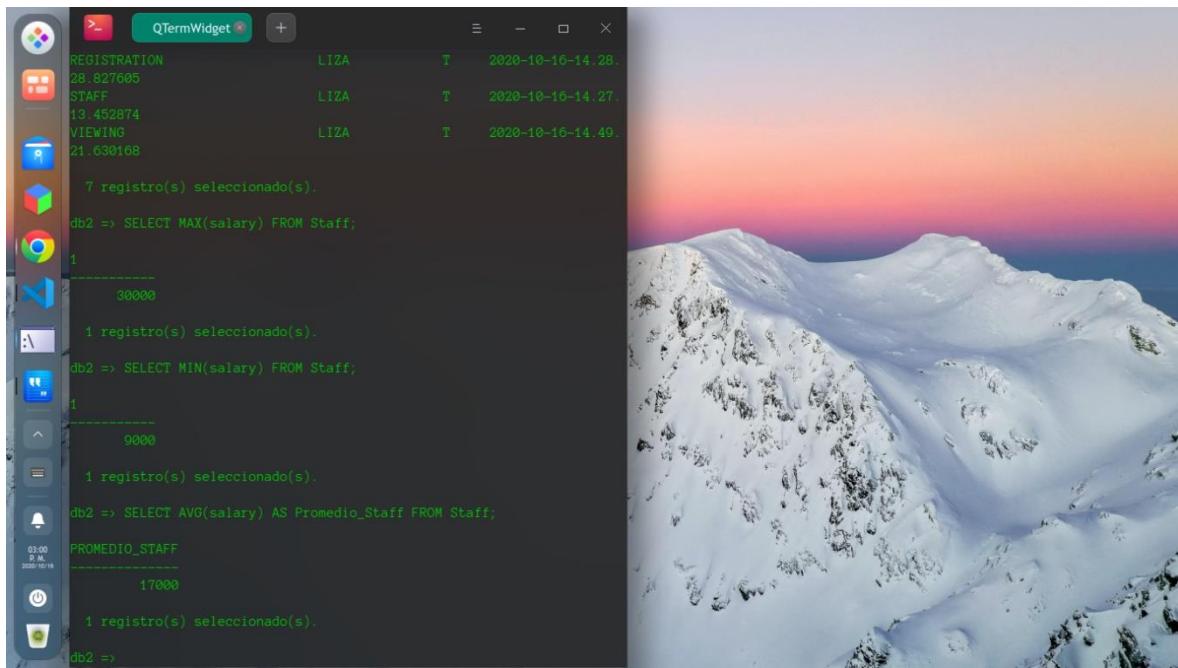
1. Mostrar el salario del empleado que gana mas

A screenshot of a dual-monitor setup. The left monitor displays a terminal window titled "QTermWidget" showing a table of database records. The table has columns "Nombre", "Tipo", and "Hora creación". The data includes entries like "BRANCH", "CLIENT", "PRIVATEOWNER", "PROPERTYFORRENT", "REGISTRATION", "STAFF", "VIEWING", and "21.630168". Below the table, the terminal shows the command "db2 => SELECT MAX(salary) FROM Staff;" followed by the result "1 30000". The right monitor shows a 3D rendering of a snow-covered mountain peak against a sunset sky.

2. Mostrar el salario del empleado que gana menos

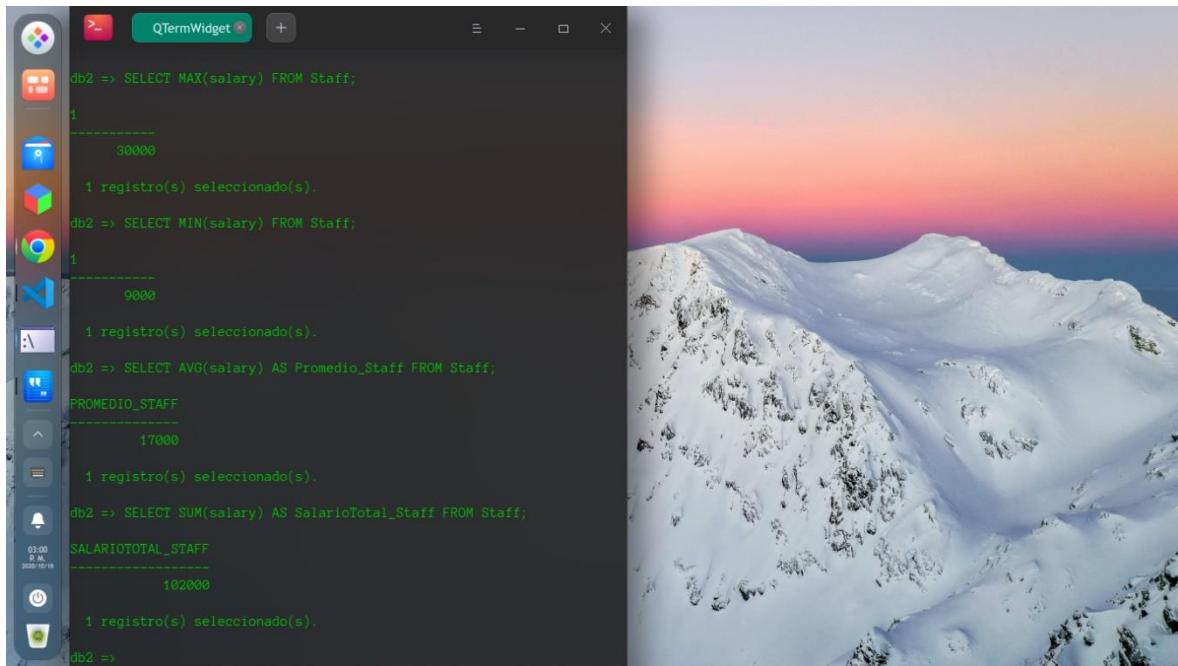
A screenshot of a Linux desktop environment. On the left, there's a vertical dock with various icons for file management, system settings, and productivity tools. The main window is a terminal titled "QTermWidget" which shows several database queries and their results. One query finds the maximum salary (30000), another finds the minimum salary (9000). The background of the desktop is a scenic image of a snow-covered mountain range under a sunset sky.

3. Muestre cual es el promedio del salario que perciben los trabajadores



```
REGISTRATION          LIZA      T    2020-10-16-14.28.  
28.827605  
STAFF                LIZA      T    2020-10-16-14.27.  
13.452874  
VIEWING              LIZA      T    2020-10-16-14.49.  
21.630168  
  
7 registro(s) seleccionado(s).  
db2 => SELECT MAX(salary) FROM Staff;  
1  
-----  
30000  
  
1 registro(s) seleccionado(s).  
db2 => SELECT MIN(salary) FROM Staff;  
1  
-----  
9000  
  
1 registro(s) seleccionado(s).  
db2 => SELECT AVG(salary) AS Promedio_Staff FROM Staff;  
  
PROMEDIO_STAFF  
-----  
17000  
  
1 registro(s) seleccionado(s).  
db2 =>
```

4. Crear una consulta que muestre la cantidad que gasta la empresa en salarios



```
db2 => SELECT MAX(salary) FROM Staff;  
1  
-----  
30000  
  
1 registro(s) seleccionado(s).  
db2 => SELECT MIN(salary) FROM Staff;  
1  
-----  
9000  
  
1 registro(s) seleccionado(s).  
db2 => SELECT AVG(salary) AS Promedio_Staff FROM Staff;  
  
PROMEDIO_STAFF  
-----  
17000  
  
1 registro(s) seleccionado(s).  
db2 => SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;  
  
SALARIOTOTAL_STAFF  
-----  
102000  
  
1 registro(s) seleccionado(s).  
db2 =>
```

5. Crear una consulta que muestre la cantidad que gasta la empresa en salarios quincenales (supniendo que el dato almacenado es mensual)



```
db2 => SELECT MIN(salary) FROM Staff;
+
-----+
      9000
      1 registro(s) seleccionado(s).

db2 => SELECT AVG(salary) AS Promedio_Staff FROM Staff;
PROMEDIO_STAFF
-----
      17000
      1 registro(s) seleccionado(s).

db2 => SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
SALARIOTOTAL_STAFF
-----
      102000
      1 registro(s) seleccionado(s).

db2 => SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
SALARIO_QUINCENAL
-----
      51000
      1 registro(s) seleccionado(s).

db2 =>
```

6. Mostrar cuantas propiedades en renta existen



```
db2 => SELECT AVG(salary) AS Promedio_Staff FROM Staff;
PROMEDIO_STAFF
-----
      17000
      1 registro(s) seleccionado(s).

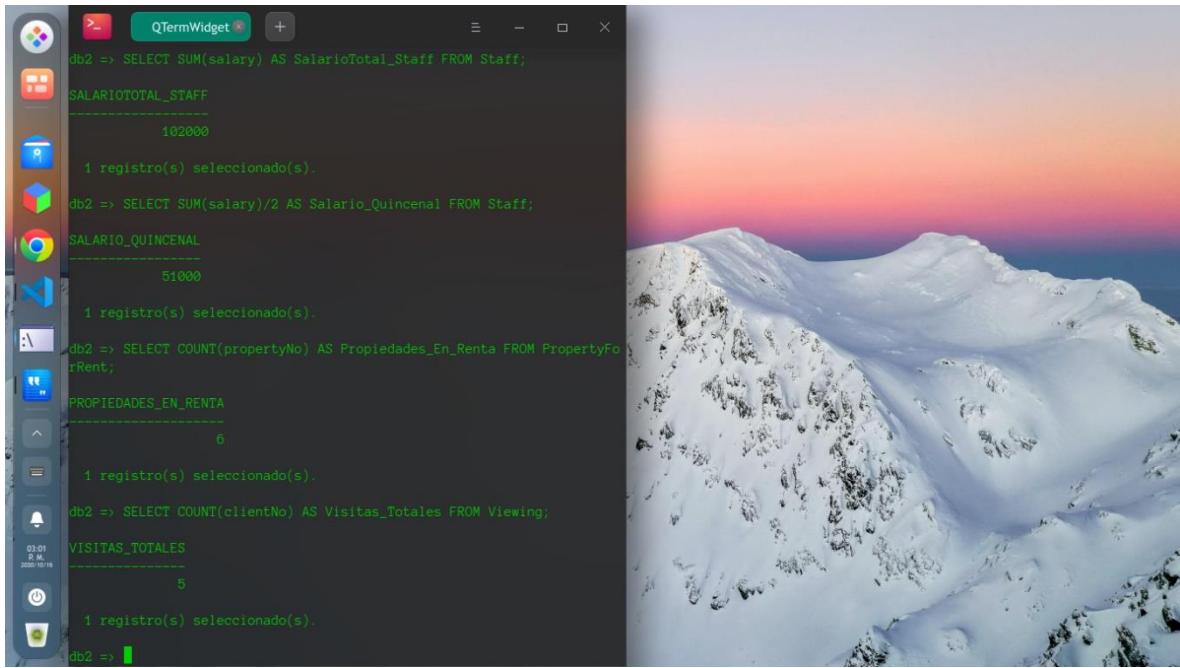
db2 => SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
SALARIOTOTAL_STAFF
-----
      102000
      1 registro(s) seleccionado(s).

db2 => SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
SALARIO_QUINCENAL
-----
      51000
      1 registro(s) seleccionado(s).

db2 => SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;
PROPIEDADES_EN_RENTA
-----
      6
      1 registro(s) seleccionado(s).

db2 =>
```

7. Mostrar cuantas visitas a las propiedades se han hecho



```
db2 => SELECT SUM(salary) AS SalarioTotal_Staff FROM Staff;
SALARIOTOTAL_STAFF
-----
102000

1 registro(s) seleccionado(s).

db2 => SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
SALARIO_QUINCENAL
-----
51000

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;
PROPIEDADES_EN_RENTA
-----
6

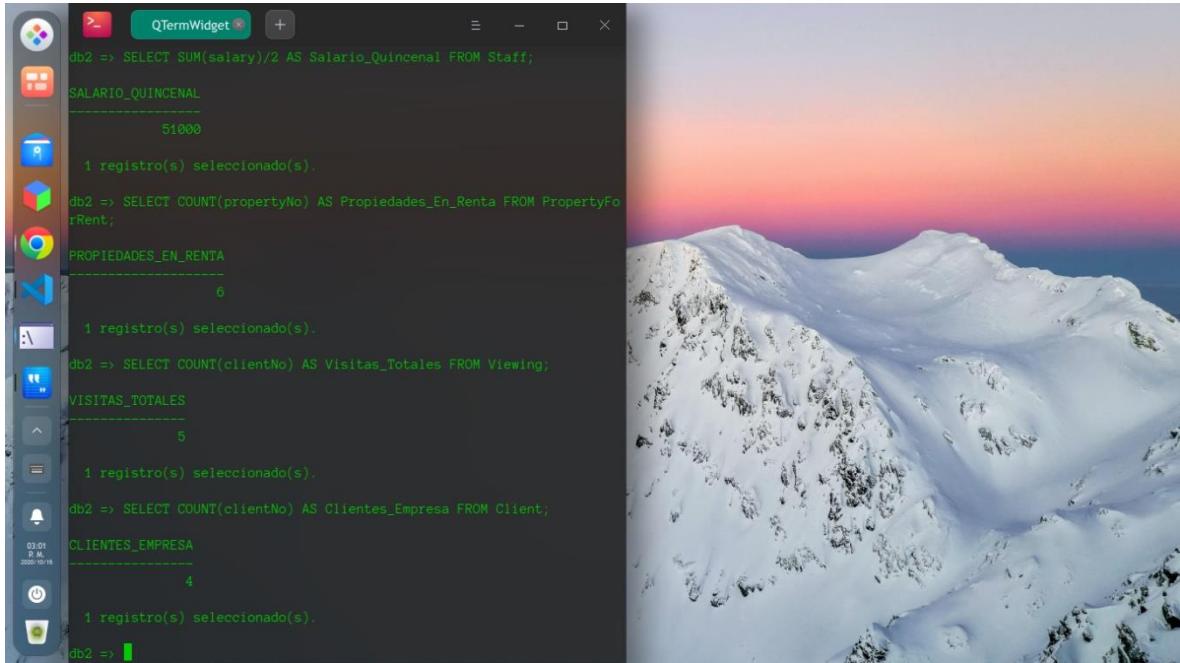
1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS Visitas_Totales FROM Viewing;
VISITAS_TOTALES
-----
5

1 registro(s) seleccionado(s).

db2 =>
```

8. Mostrar la cantidad de clientes que atiende la empresa



```
db2 => SELECT SUM(salary)/2 AS Salario_Quincenal FROM Staff;
SALARIO_QUINCENAL
-----
51000

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(propertyNo) AS Propiedades_En_Renta FROM PropertyForRent;
PROPIEDADES_EN_RENTA
-----
6

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS Visitas_Totales FROM Viewing;
VISITAS_TOTALES
-----
5

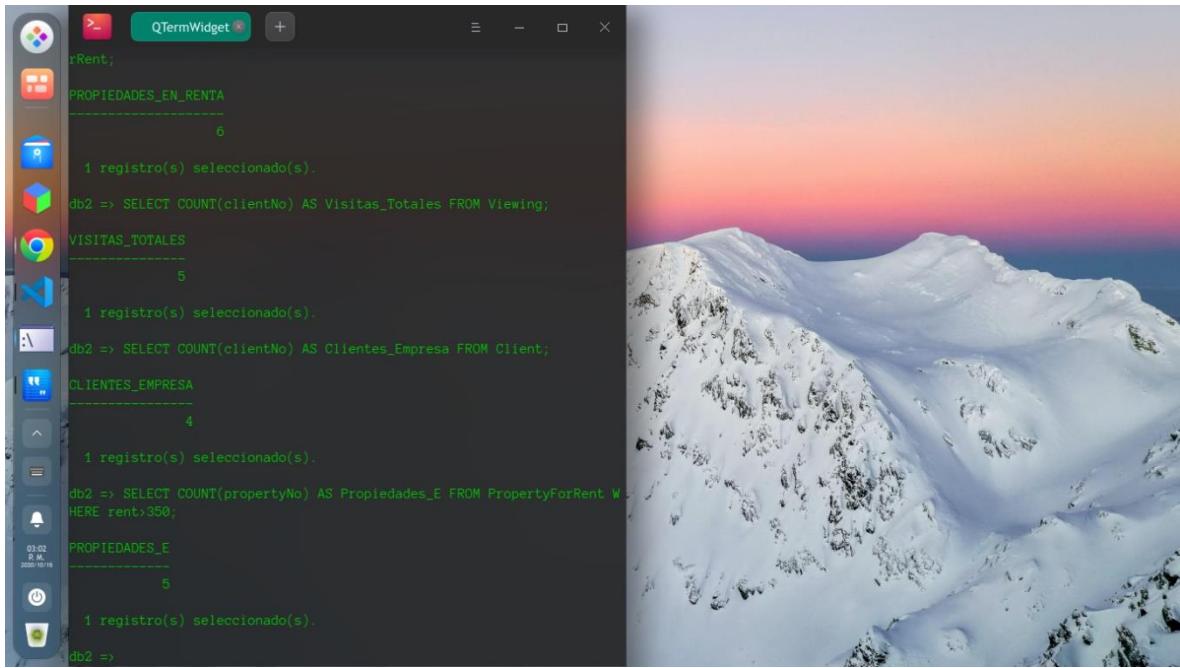
1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS Clientes_Empresa FROM Client;
CLIENTES_EMPRESA
-----
4

1 registro(s) seleccionado(s).

db2 =>
```

9. Mostrar cuantas propiedades en renta que cuesten mas de 350 euros existen



```
rRent;
PROPIEDADES_EN_RENTA
-----
6

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS Visitas_Totales FROM Viewing;

VISITAS_TOTALES
-----
5

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS Clientes_Empresa FROM Client;

CLIENTES_EMPRESA
-----
4

1 registro(s) seleccionado(s).

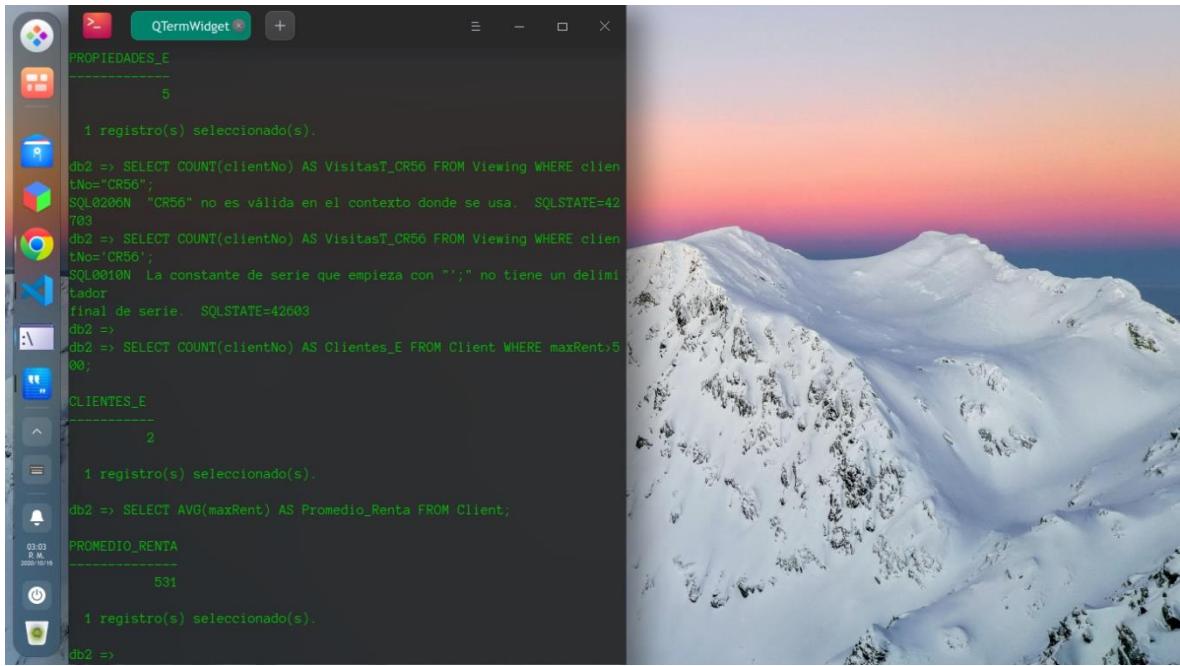
db2 => SELECT COUNT(propertyNo) AS Propiedades_E FROM PropertyForRent WHERE rent>350;

PROPIEDADES_E
-----
5

1 registro(s) seleccionado(s).

db2 =>
```

10. Mostar la cantidad de clientes que puedan pagar una renta mayor a 500 euros atiende la empresa



```
PROPIEDADES_E
-----
5

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo='CR56';
SQL0206N  "CR56" no es válida en el contexto donde se usa.  SQLSTATE=42703
db2 => SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo= 'CR56';
SQL0010N  La constante de serie que empieza con "";" no tiene un delimitador final de serie.  SQLSTATE=42603
db2 =>
db2 => SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;

CLIENTES_E
-----
2

1 registro(s) seleccionado(s).

db2 => SELECT AVG(maxRent) AS Promedio_Renta FROM Client;

PROMEDIO_RENTA
-----
531

1 registro(s) seleccionado(s).

db2 =>
```

11. Calcular el promedio de la renta que pueden pagar los clientes



```
QTermWidget + - X
PROPIEDADES_E
-----
5

1 registro(s) seleccionado(s).

db2 => SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo='CR56';
SQL0200N "CR56" no es válida en el contexto donde se usa. SQLSTATE=42703
db2 => SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo= 'CR56';
SQL0010N La constante de serie que empieza con "";" no tiene un delimitador final de serie. SQLSTATE=42603
db2 =>
db2 => SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;

CLIENTES_E
-----
2

1 registro(s) seleccionado(s).

db2 => SELECT AVG(maxRent) AS Promedio_Renta FROM Client;

PROMEDIO_RENTA
-----
531

1 registro(s) seleccionado(s).

db2 =>
```

12. Mostrar el total de rentas recaudadas al mes



```
QTermWidget + - X
SQL0200N "CR56" no es válida en el contexto donde se usa. SQLSTATE=42703
db2 => SELECT COUNT(clientNo) AS VisitasT_CR56 FROM Viewing WHERE clientNo='CR56';
SQL0010N La constante de serie que empieza con "";" no tiene un delimitador final de serie. SQLSTATE=42603
db2 =>
db2 => SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;

CLIENTES_E
-----
2

1 registro(s) seleccionado(s).

db2 => SELECT AVG(maxRent) AS Promedio_Renta FROM Client;

PROMEDIO_RENTA
-----
531

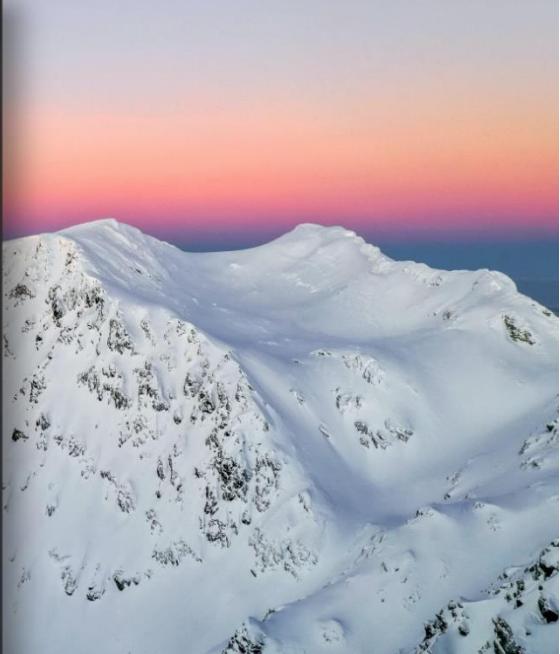
1 registro(s) seleccionado(s).

db2 => SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
RENTAS_RECAUDADAS
-----
2825

1 registro(s) seleccionado(s).

db2 =>
```

13. Mostrar cual es la renta mas cara pagada y cual es la mas barata



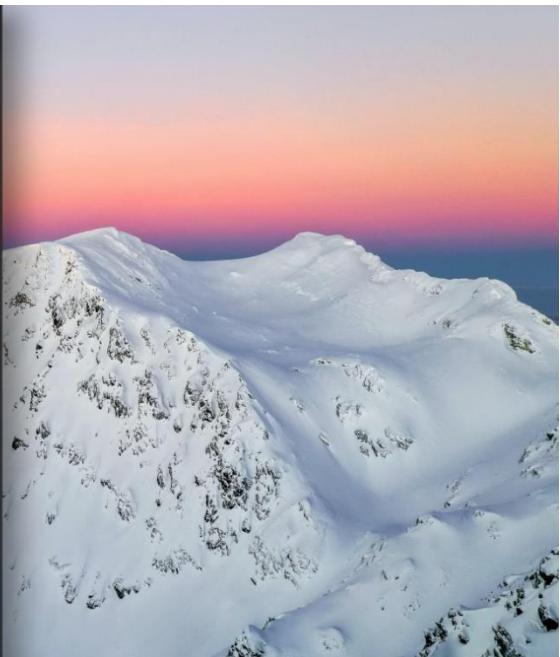
```
db2 => SELECT AVG(maxRent) AS Promedio_Renta FROM Client;
PROMEDIO_RENTA
-----
      531
1 registro(s) seleccionado(s).

db2 => SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
RENTAS_RECAUDADAS
-----
     2825
1 registro(s) seleccionado(s).

db2 => SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
MAX_RENTA
-----
      650
1 registro(s) seleccionado(s).

db2 => SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
MIN_RENTA
-----
      350
1 registro(s) seleccionado(s).

db2 =>
```



```
db2 => SELECT COUNT(clientNo) AS Clientes_E FROM Client WHERE maxRent>500;
CLIENTES_E
-----
      2
1 registro(s) seleccionado(s).

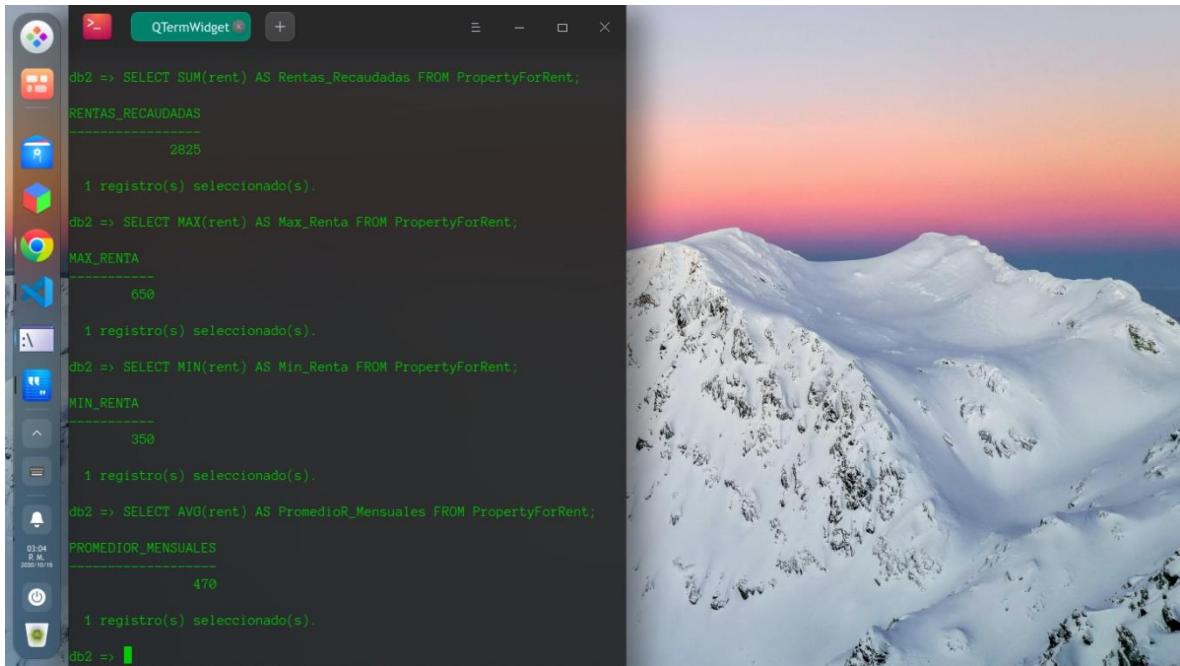
db2 => SELECT AVG(maxRent) AS Promedio_Renta FROM Client;
PROMEDIO_RENTA
-----
      531
1 registro(s) seleccionado(s).

db2 => SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
RENTAS_RECAUDADAS
-----
     2825
1 registro(s) seleccionado(s).

db2 => SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
MAX_RENTA
-----
      650
1 registro(s) seleccionado(s).

db2 =>
```

14. Calcular el promedio de la renta que recibe la empresa



```
db2 => SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
RENTAS_RECAUDADAS
-----
2825
1 registro(s) seleccionado(s).

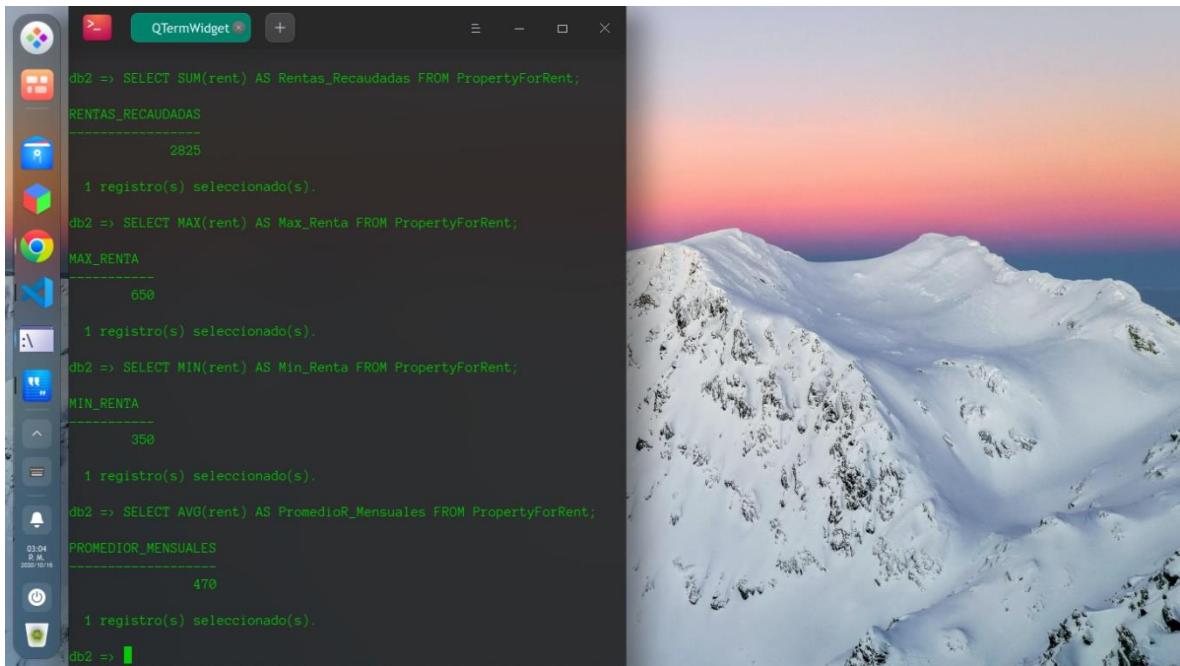
db2 => SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
MAX_RENTA
-----
650
1 registro(s) seleccionado(s).

db2 => SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
MIN_RENTA
-----
350
1 registro(s) seleccionado(s).

db2 => SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
PROMEDIOR_MENSUALES
-----
470
1 registro(s) seleccionado(s).

db2 => 
```

15. Mostrar el total de rentas que pueden pagar los clientes al mes



```
db2 => SELECT SUM(rent) AS Rentas_Recaudadas FROM PropertyForRent;
RENTAS_RECAUDADAS
-----
2825
1 registro(s) seleccionado(s).

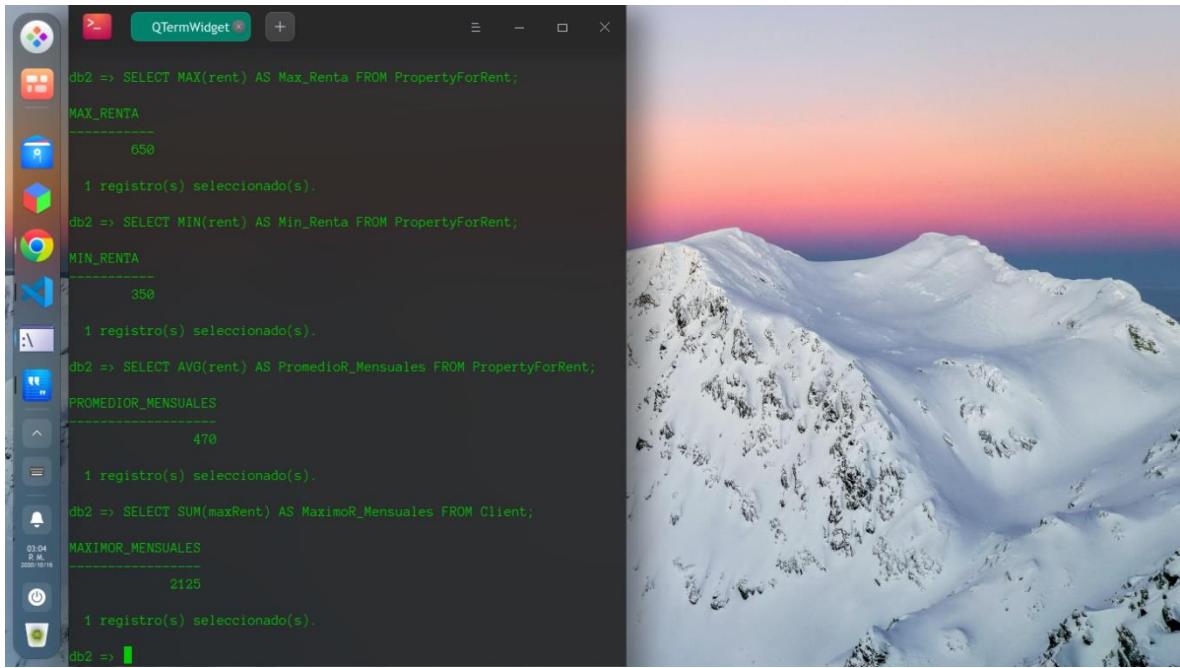
db2 => SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
MAX_RENTA
-----
650
1 registro(s) seleccionado(s).

db2 => SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
MIN_RENTA
-----
350
1 registro(s) seleccionado(s).

db2 => SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
PROMEDIOR_MENSUALES
-----
470
1 registro(s) seleccionado(s).

db2 => 
```

16. Mostrar el total de rentas recaudadas por rentar CASAS



```
db2 => SELECT MAX(rent) AS Max_Renta FROM PropertyForRent;
MAX_RENTA
-----
650
1 registro(s) seleccionado(s).

db2 => SELECT MIN(rent) AS Min_Renta FROM PropertyForRent;
MIN_RENTA
-----
350
1 registro(s) seleccionado(s).

db2 => SELECT AVG(rent) AS PromedioR_Mensuales FROM PropertyForRent;
PROMEDIOR_MENSUALES
-----
470
1 registro(s) seleccionado(s).

db2 => SELECT SUM(maxRent) AS MaximoR_Mensuales FROM Client;
MAXIMOR_MENSUALES
-----
2125
1 registro(s) seleccionado(s).

db2 => |
```