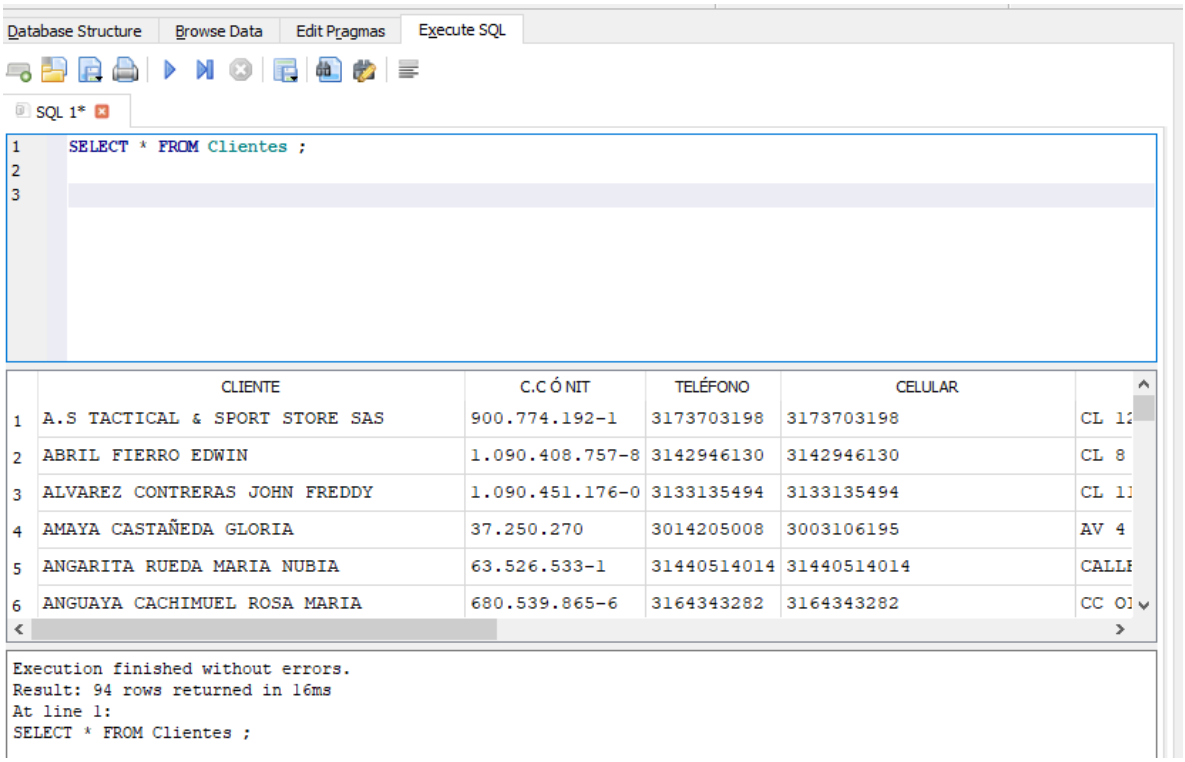
Daniel David Sanchez Florez

Ejercicios de la tabla- Evidencia

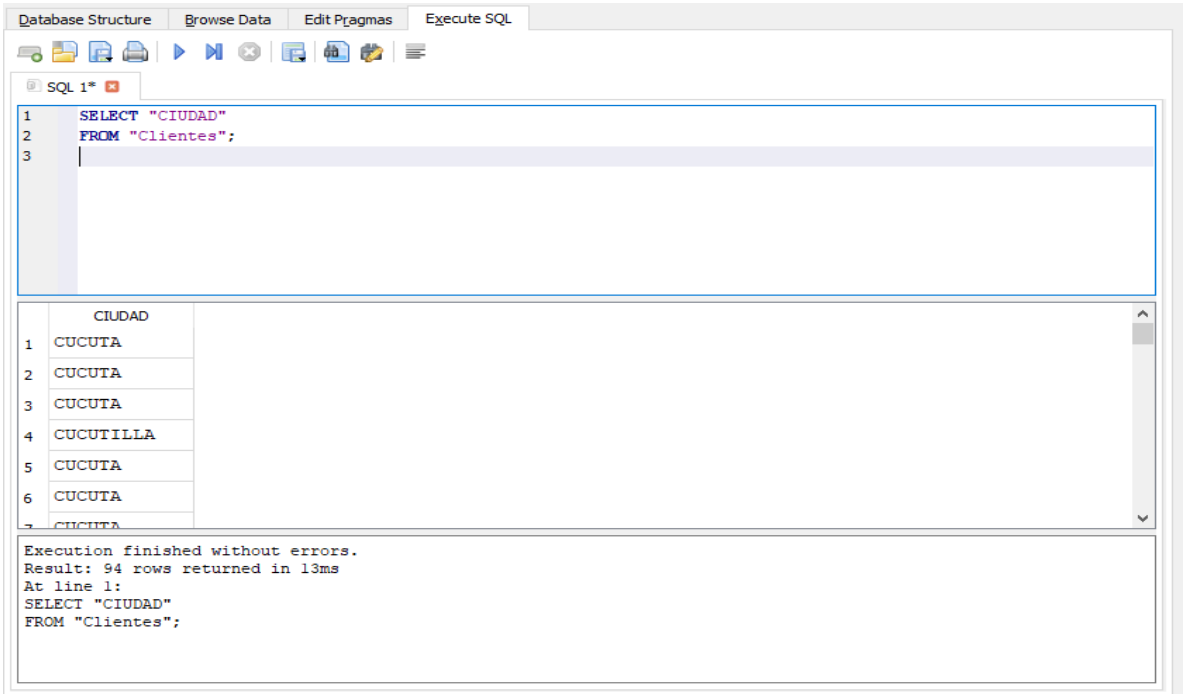
1. Seleccionar todas las columnas de una tabla



	CLIENTE	C.C Ó NIT	TELÉFONO	CELULAR	
1	A.S TACTICAL & SPORT STORE SAS	900.774.192-1	3173703198	3173703198	CL 12
2	ABRIL FIERRO EDWIN	1.090.408.757-8	3142946130	3142946130	CL 8
3	ALVAREZ CONTRERAS JOHN FREDDY	1.090.451.176-0	3133135494	3133135494	CL 11
4	AMAYA CASTAÑEDA GLORIA	37.250.270	3014205008	3003106195	AV 4
5	ANGARITA RUEDA MARIA NUBIA	63.526.533-1	31440514014	31440514014	CALLE
6	ANGUAYA CACHIMUEL ROSA MARIA	680.539.865-6	3164343282	3164343282	CC 01

```
Execution finished without errors.
Result: 94 rows returned in 16ms
At line 1:
SELECT * FROM Clientes ;
```

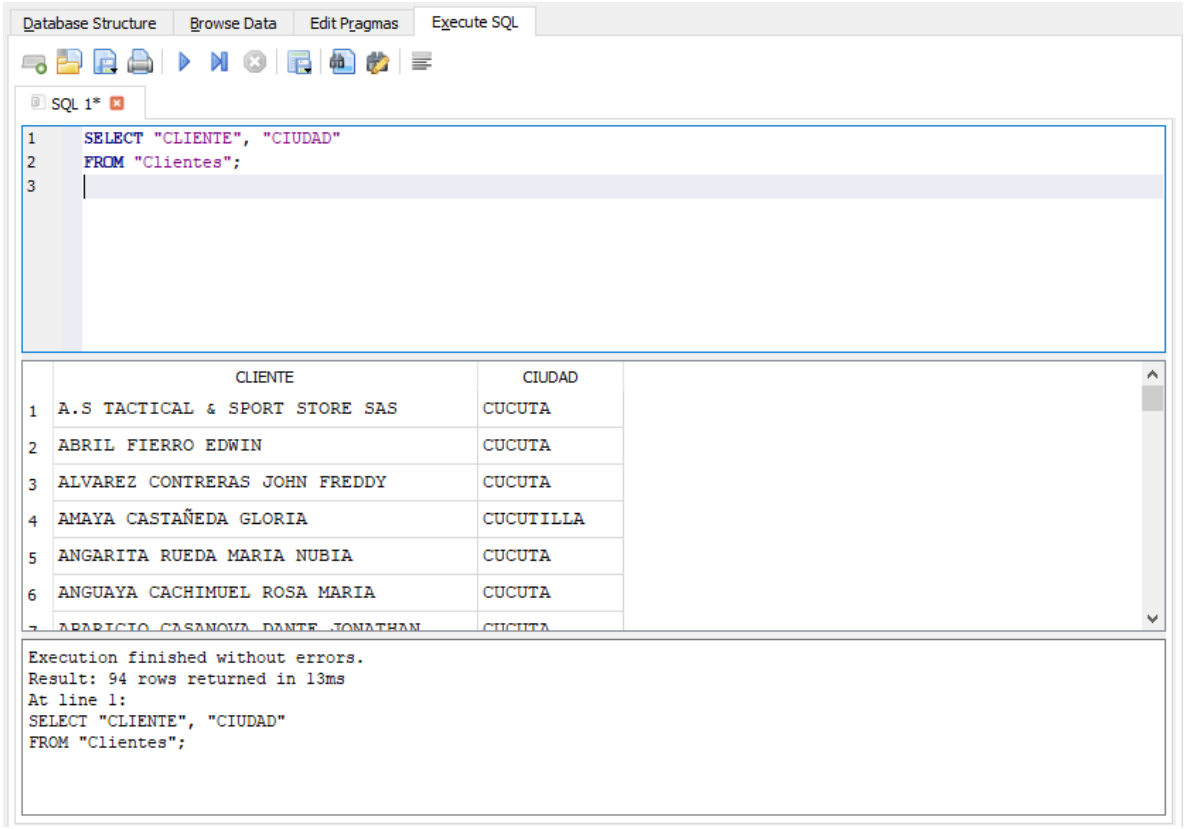
2. Selección de una columna de una tabla



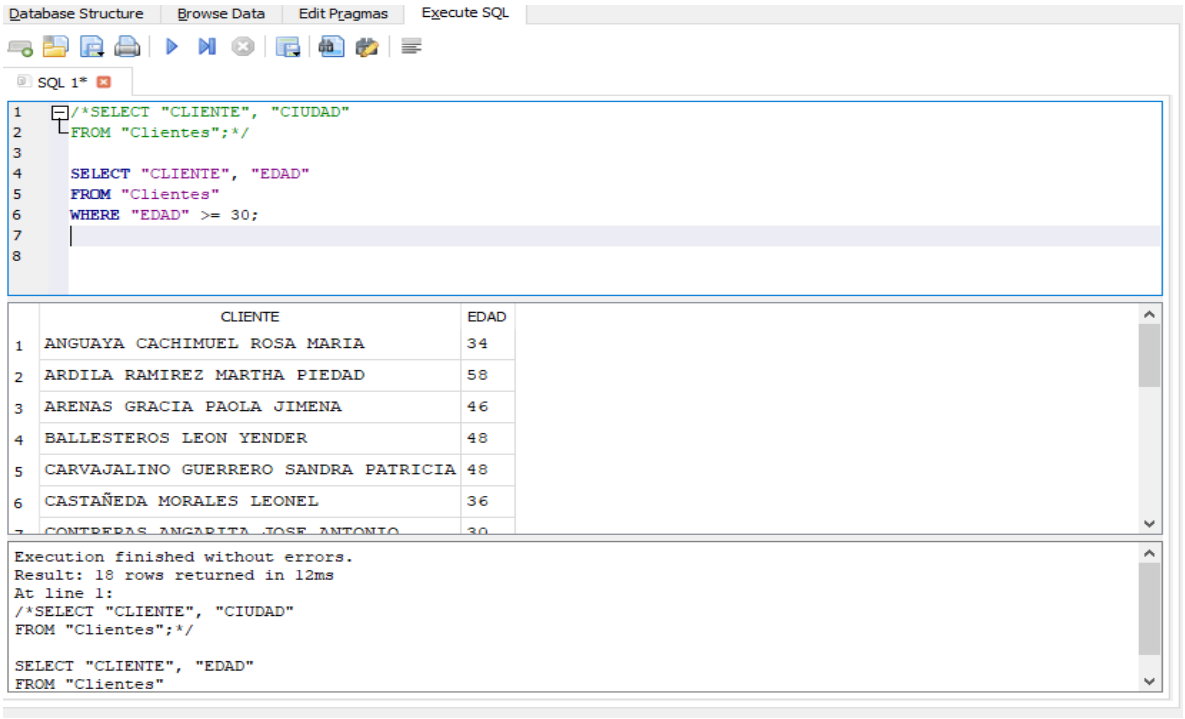
	CIUDAD
1	CUCUTA
2	CUCUTA
3	CUCUTA
4	CUCUTILLA
5	CUCUTA
6	CUCUTA

```
Execution finished without errors.
Result: 94 rows returned in 13ms
At line 1:
SELECT "CIUDAD"
FROM "Clientes";
```

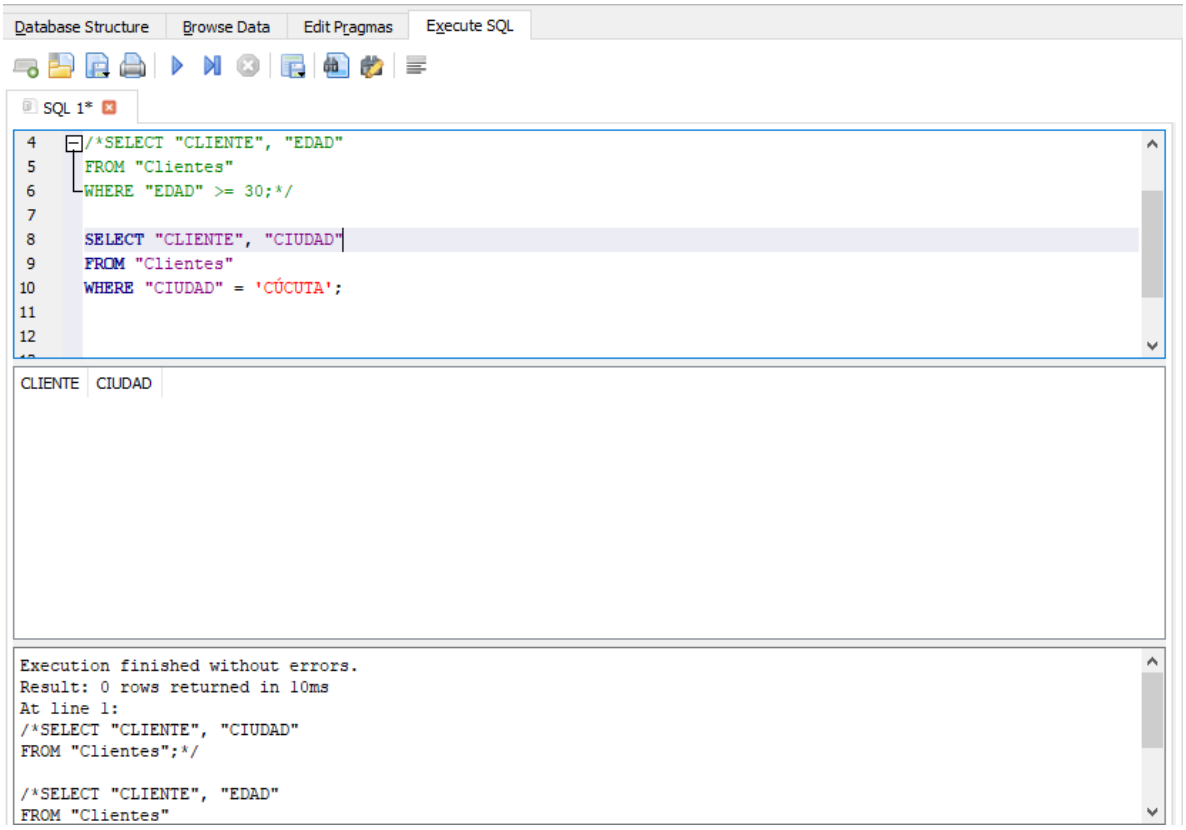
3. Selección de dos columnas de una tabla



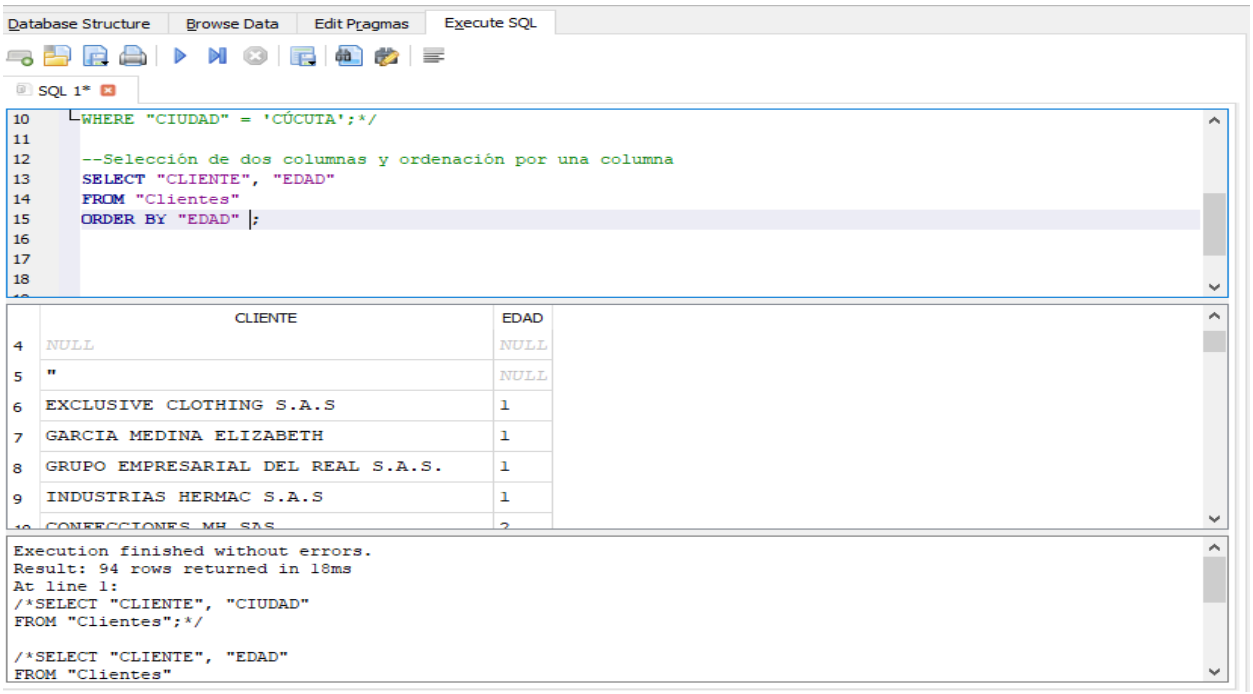
4. Selección de dos (o más) columnas de una tabla y filtrado mediante comparación numérica en WHERE



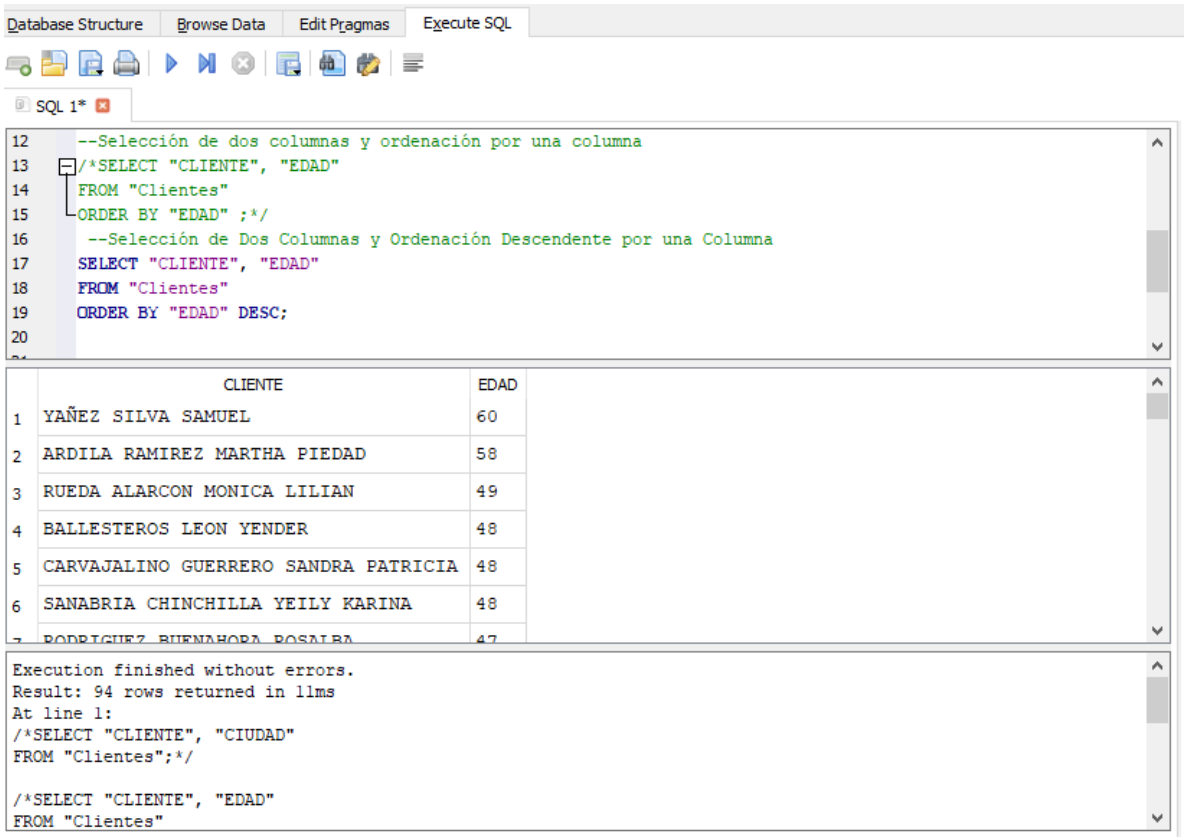
5. **Selección de Dos Columnas y Filtrado Utilizando una Condición de Igualdad en WHERE**



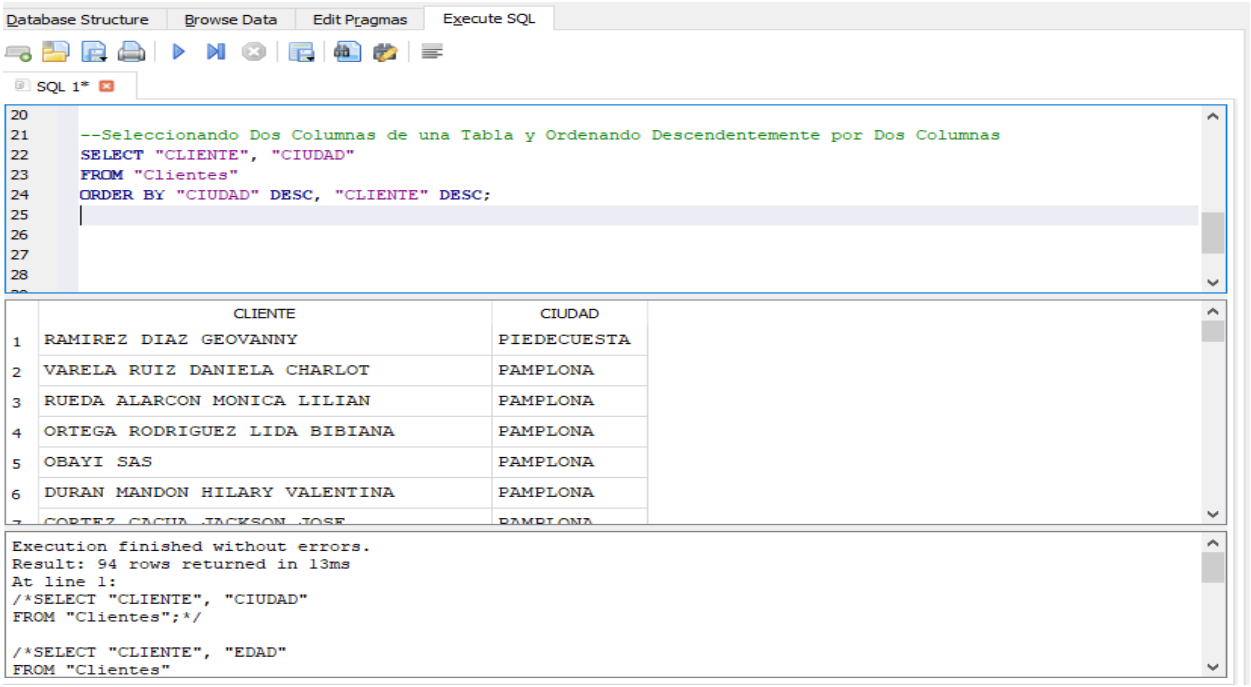
6. **Selección de dos columnas y ordenación por una columna**



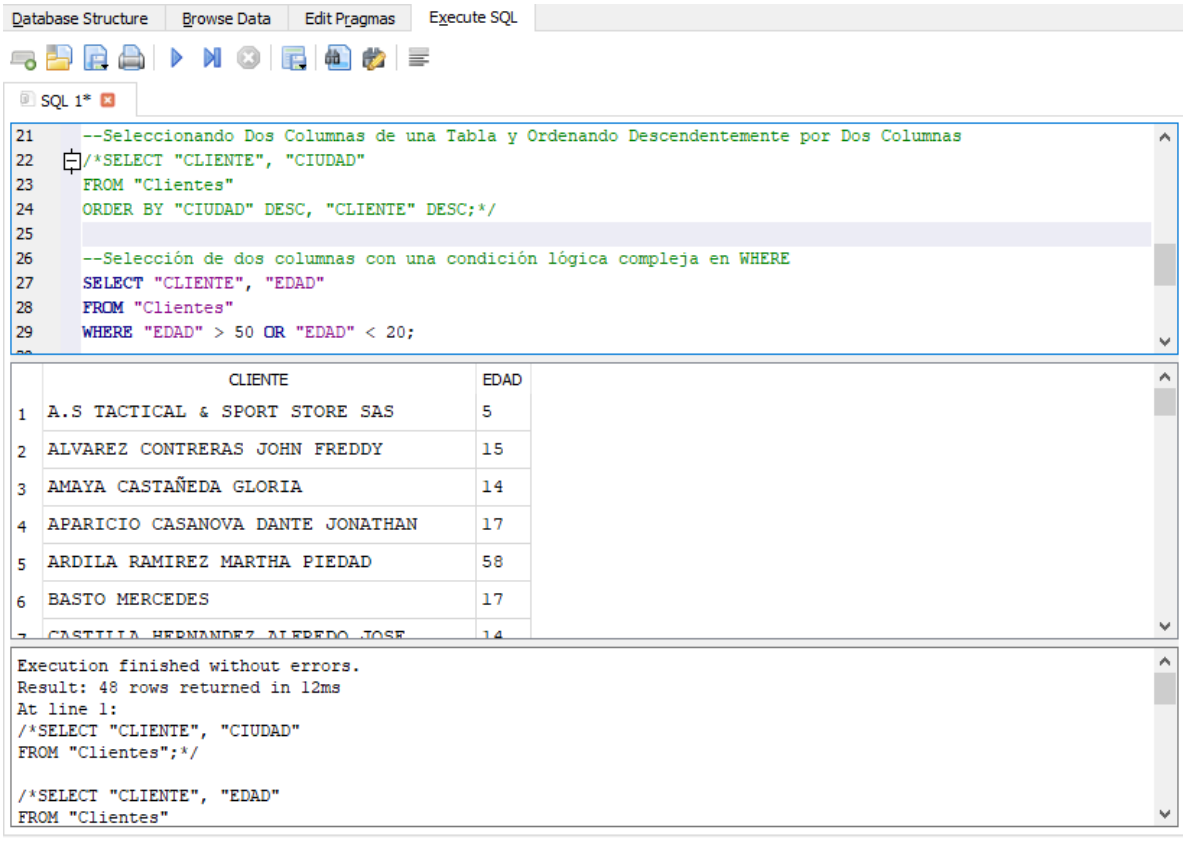
7. Selección de Dos Columnas y Ordenación Descendente por una Columna



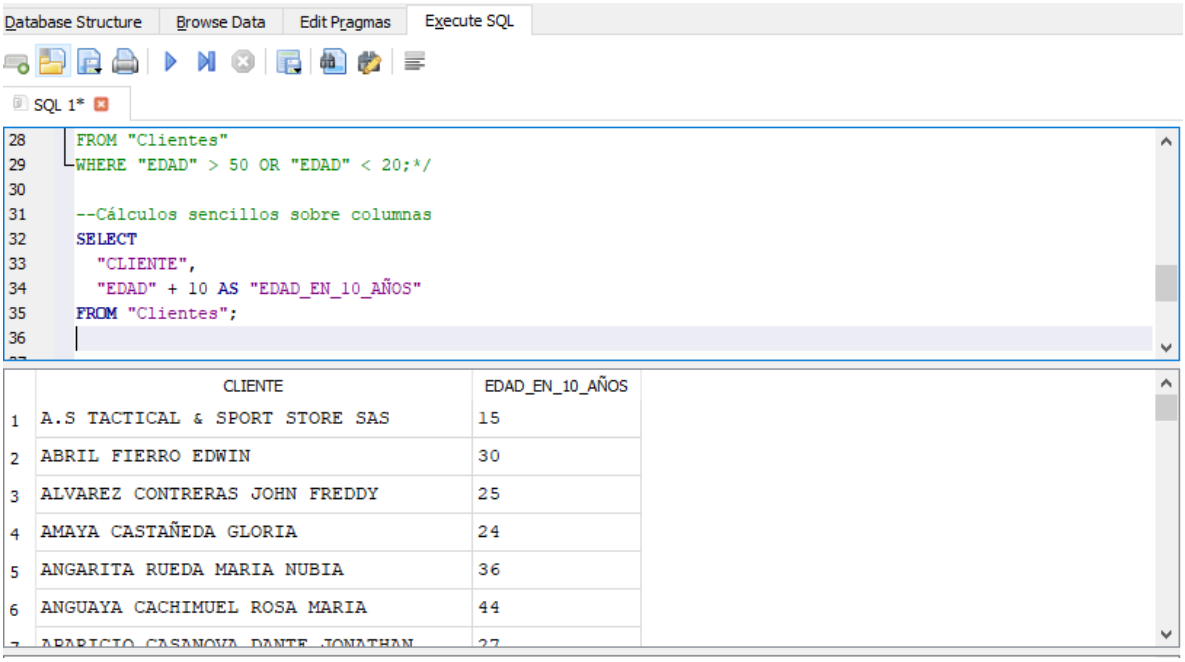
8. Seleccionando Dos Columnas de una Tabla y Ordenando Descendentemente por Dos Columnas



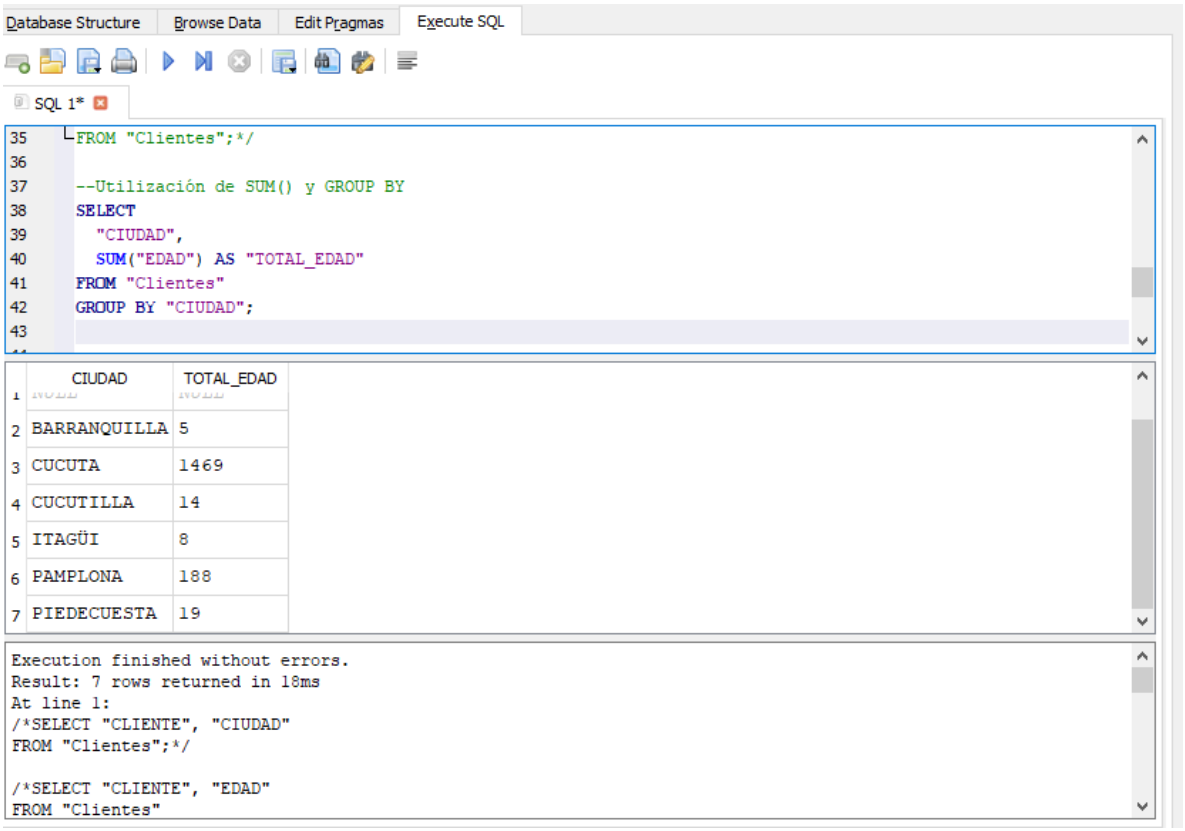
9. Selección de dos columnas con una condición lógica compleja en WHERE



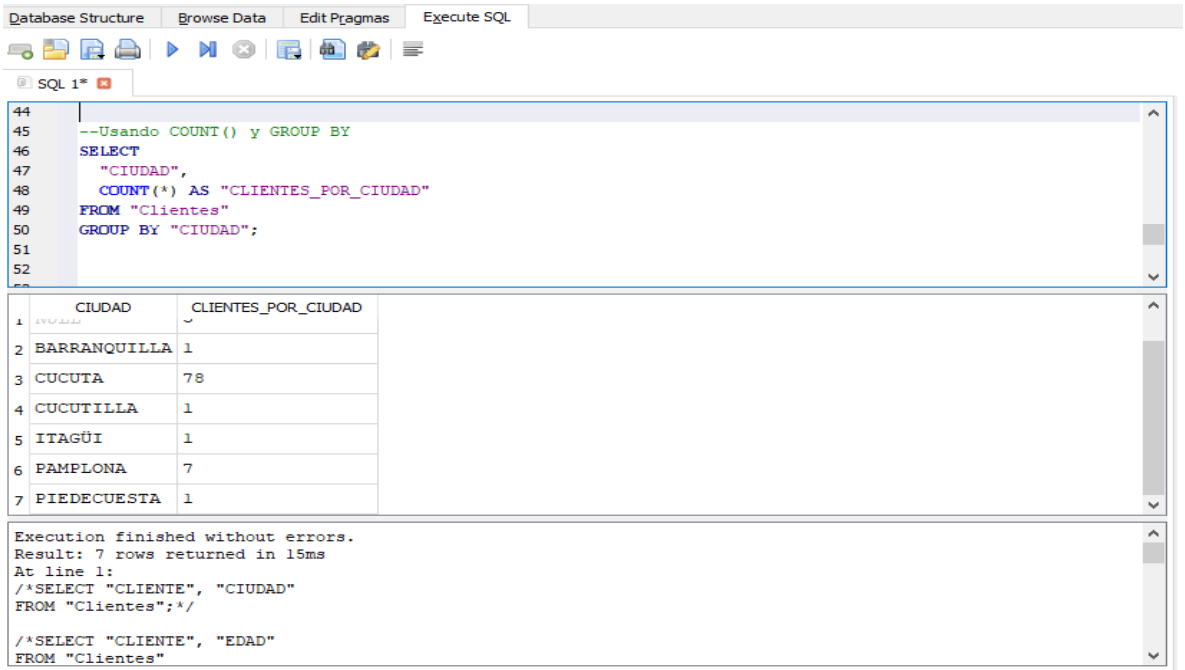
10. Cálculos sencillos sobre columnas



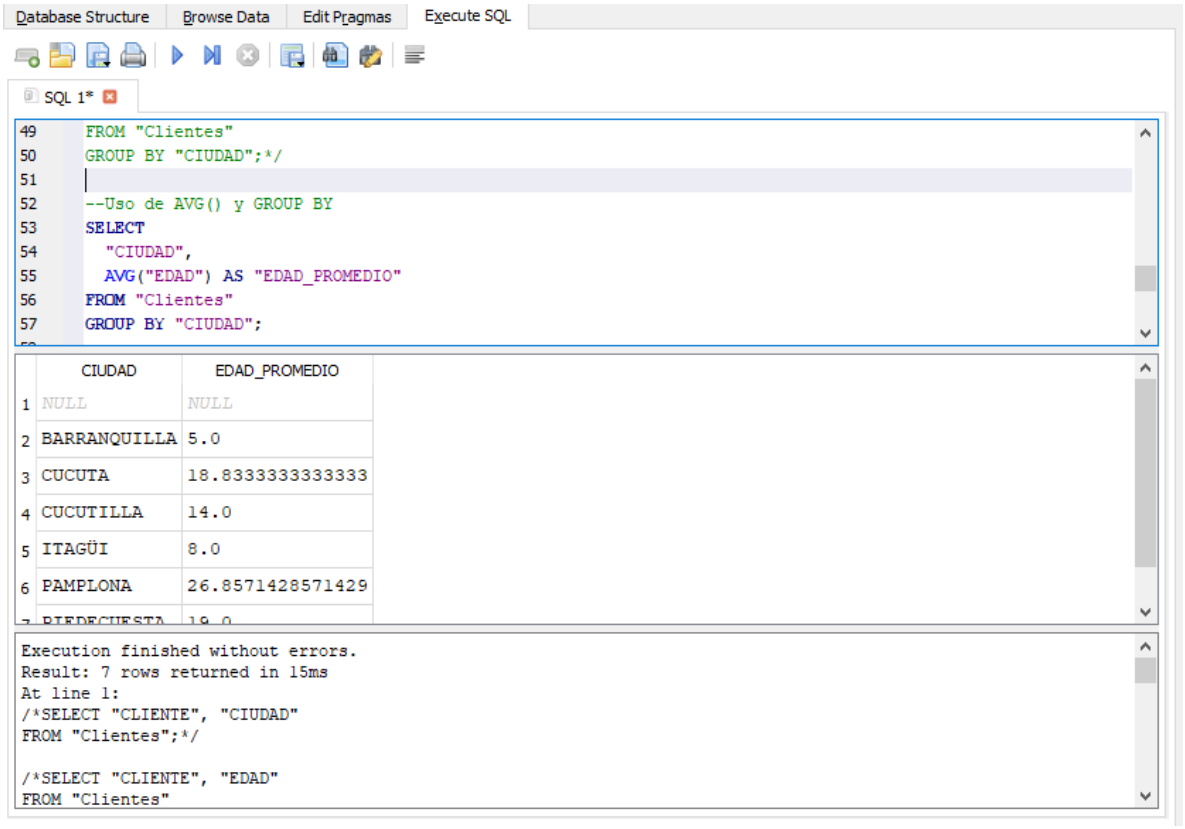
11. Utilización de SUM() y GROUP BY



12. Usando COUNT() y GROUP BY



13. Uso de AVG() y GROUP BY

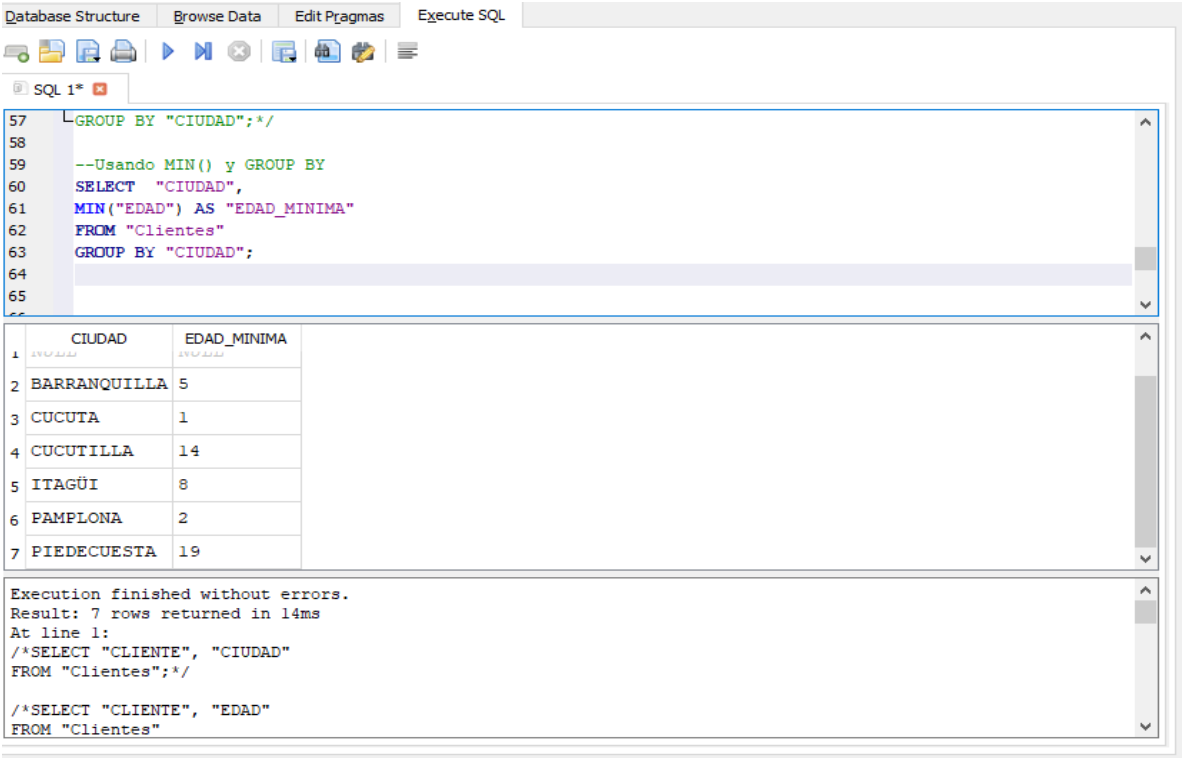


```
Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

49  FROM "Clientes"
50  GROUP BY "CIUDAD";*/
51
52  --Uso de AVG() y GROUP BY
53  SELECT
54    "CIUDAD",
55    AVG("EDAD") AS "EDAD_PROMEDIO"
56  FROM "Clientes"
57  GROUP BY "CIUDAD";
```

	CIUDAD	EDAD_PROMEDIO
1	NULL	NULL
2	BARRANQUILLA	5.0
3	CUCUTA	18.8333333333333
4	CUCUTILLA	14.0
5	ITAGÜI	8.0
6	PAMPLONA	26.8571428571429
7	PIEDECUESTA	19.0

```
Execution finished without errors.
Result: 7 rows returned in 15ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"
```

14. Usando MIN() y GROUP BY

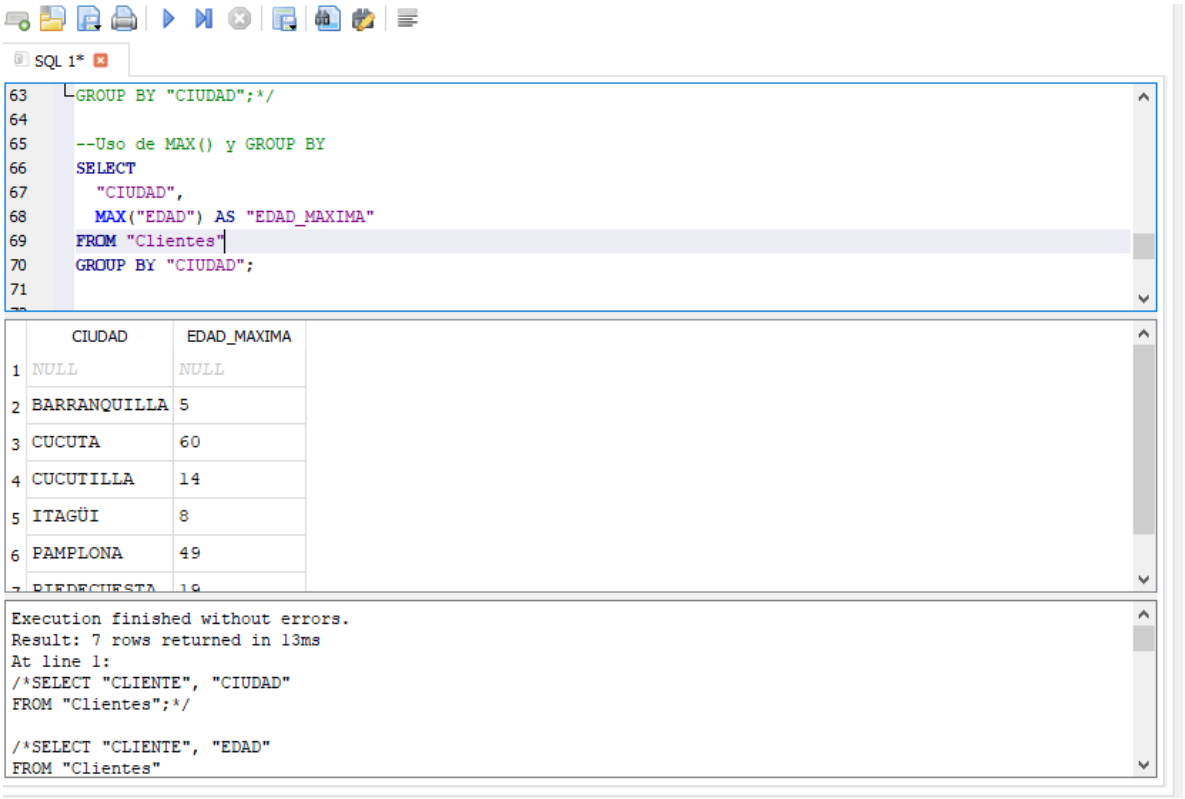


```
Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

57  GROUP BY "CIUDAD";*/
58
59  --Usando MIN() y GROUP BY
60  SELECT  "CIUDAD",
61  MIN("EDAD") AS "EDAD_MINIMA"
62  FROM "Clientes"
63  GROUP BY "CIUDAD";
64
65
```

	CIUDAD	EDAD_MINIMA
2	BARRANQUILLA	5
3	CUCUTA	1
4	CUCUTILLA	14
5	ITAGÜI	8
6	PAMPLONA	2
7	PIEDECUESTA	19

```
Execution finished without errors.
Result: 7 rows returned in 14ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"
```

15. Uso de MAX() y GROUP BY

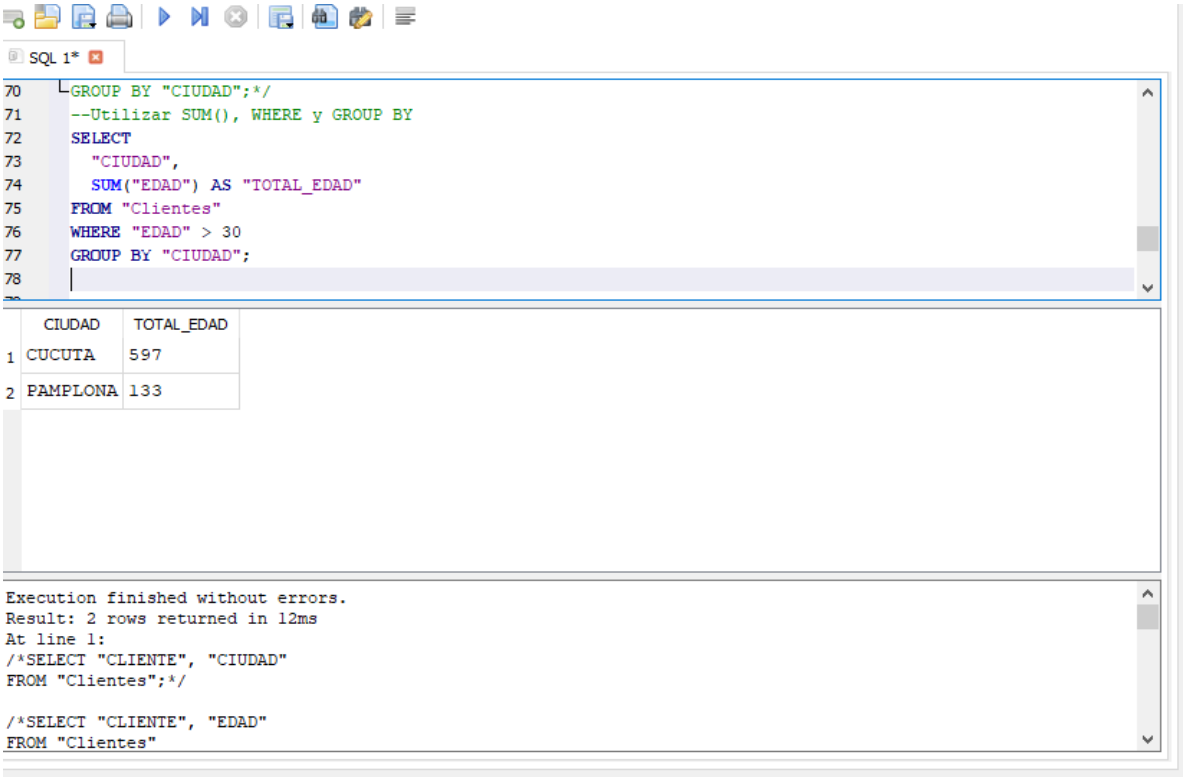


```
SQL 1*
63  GROUP BY "CIUDAD";*/
64
65  --Uso de MAX() y GROUP BY
66  SELECT
67    "CIUDAD",
68    MAX("EDAD") AS "EDAD_MAXIMA"
69  FROM "Clientes"
70  GROUP BY "CIUDAD";
71
```

	CIUDAD	EDAD_MAXIMA
1	NULL	NULL
2	BARRANQUILLA	5
3	CUCUTA	60
4	CUCUTILLA	14
5	ITAGÜI	8
6	PAMPLONA	49
7	RIEDECUESTA	19

```
Execution finished without errors.
Result: 7 rows returned in 13ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"
```

16. Utilizar SUM(), WHERE y GROUP BY



```
SQL 1*
70  GROUP BY "CIUDAD";*/
71  --Utilizar SUM(), WHERE y GROUP BY
72  SELECT
73    "CIUDAD",
74    SUM("EDAD") AS "TOTAL_EDAD"
75  FROM "Clientes"
76  WHERE "EDAD" > 30
77  GROUP BY "CIUDAD";
78
```

	CIUDAD	TOTAL_EDAD
1	CUCUTA	597
2	PAMPLONA	133

```
Execution finished without errors.
Result: 2 rows returned in 12ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"
```

17. Uso de COUNT(), WHERE y GROUP BY

```
78
79  --Uso de COUNT(), WHERE y GROUP BY
80  SELECT
81    "CIUDAD",
82    COUNT(*) AS "NUMERO_DE_CLIENTES"
83  FROM "Clientes"
84  WHERE "EDAD" > 30
85  GROUP BY "CIUDAD";
86
```

	CIUDAD	NUMERO_DE_CLIENTES
1	CUCUTA	13
2	PAMPLONA	3

```
Execution finished without errors.
Result: 2 rows returned in 12ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"
```

NOTA PARA EJERCICIO 18 – 19 Y 20 CREAMOS OTRA TABLA (pedidos) Y LE ASIGNAMOS CLAVE PRIMARIA A UN DATO DE CLIENTES

SQL 1*

```
95   */
96   --Rellenamos la tabla
97   INSERT INTO "Pedidos" ("C.C Ó NIT", "FECHA", "MONTO", "PRODUCTO")
98   VALUES
99     ('901.239.274-7', '2024-02-15', 250.00, 'Producto A'),
100    ('1.090.387.996-1', '2024-03-10', 450.00, 'Producto B'),
101    ('901.447.197-1', '2024-06-17', 300.00, 'Producto C'),
102    ('901.288.713-8', '2024-08-22', 150.00, 'Producto D'),
103    ('901.811.137-9', '2024-11-03', 500.00, 'Producto E');
104
```

- **Se relleno con cedulas de la tabla clientes con la clave foránea “c.c o NIT” que esta vinculada a la clave primaria de la tabla clientes**

18. Acceso a Datos en Dos Tablas Utilizando INNER JOIN

SQL 1*

```
104
105  --Acceso a Datos en Dos Tablas Utilizando INNER JOIN
106  SELECT
107    c."CLIENTE",
108    c."C.C Ó NIT",
109    SUM(p."MONTO") AS total_pedidos_2024
110  FROM "Clientes" c
111  INNER JOIN "Pedidos" p
112    ON c."C.C Ó NIT" = p."C.C Ó NIT"
113  WHERE strftime('%Y', p."FECHA") = '2024'
114  GROUP BY c."C.C Ó NIT";
115
```

	CLIENTE	C.C Ó NIT	total_pedidos_2024
1	CONTRERAS ANGARITA JOSE ANTONIO	1.090.387.996-1	450.0
2	CONFECCIONES MH SAS	901.239.274-7	250.0
3	GRUPO EMPRESARIAL DEL REAL S.A.S.	901.288.713-8	150.0
4	GROUP VISTARS S.A.S	901.447.197-1	300.0
5	GRUPO EMPRESARIAL FIURY É BARUD …	901.811.137-9	500.0

```
Execution finished without errors.
Result: 5 rows returned in 14ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/
```

19. Acceso a Datos en Dos Tablas Utilizando INNER JOIN y Filtrando Utilizando WHERE

SQL 1*

```
115
116  -- Acceso a Datos en Dos Tablas Utilizando INNER JOIN y Filtrando Utilizando WHERE
117  SELECT
118    c."CLIENTE",
119    c."C.C Ó NIT",
120    p."MONTO",
121    p."PRODUCTO"
122  FROM "Clientes" c
123  INNER JOIN "Pedidos" p
124    ON c."C.C Ó NIT" = p."C.C Ó NIT"
125  WHERE p."MONTO" > 300;
126
```

	CLIENTE	C.C Ó NIT	MONTO	PRODUCTO
1	CONTRERAS ANGARITA JOSE ANTONIO	1.090.387.996-1	450.0	Producto B
2	GRUPO EMPRESARIAL FIURY É BARUD …	901.811.137-9	500.0	Producto E

```
Execution finished without errors.
Result: 2 rows returned in 12ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/
```

20. Accediendo a Datos en Dos Tablas Usando INNER JOIN, Filtrando Usando WHERE, y Ordenando con ORDER BY

SQL 1*

```
--Accediendo a Datos en Dos Tablas Usando INNER JOIN, Filtrando Usando WHERE, y Ordenando con ORDER BY
SELECT
  c."CLIENTE",
  c."C.C Ó NIT",
  p."MONTO",
  p."PRODUCTO"
FROM "Clientes" c
INNER JOIN "Pedidos" p
  ON c."C.C Ó NIT" = p."C.C Ó NIT"
WHERE p."MONTO" > 300
ORDER BY p."MONTO" DESC;
```

	CLIENTE	C.C Ó NIT	MONTO	PRODUCTO
1	GRUPO EMPRESARIAL FIURY É BARUD …	901.811.137-9	500.0	Producto E
2	CONTRERAS ANGARITA JOSE ANTONIO	1.090.387.996-1	450.0	Producto B

```
Execution finished without errors.
Result: 2 rows returned in 14ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/
```