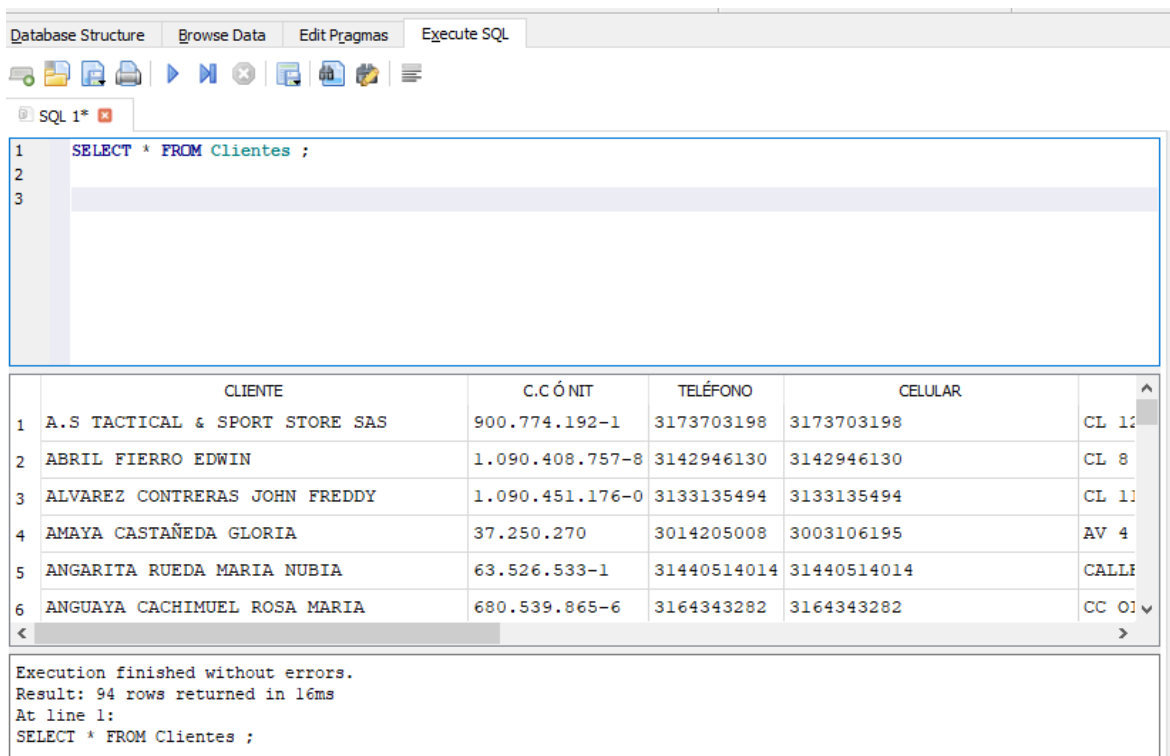


Ejercicios de la tabla- Evidencia

1. Seleccionar todas las columnas de una tabla

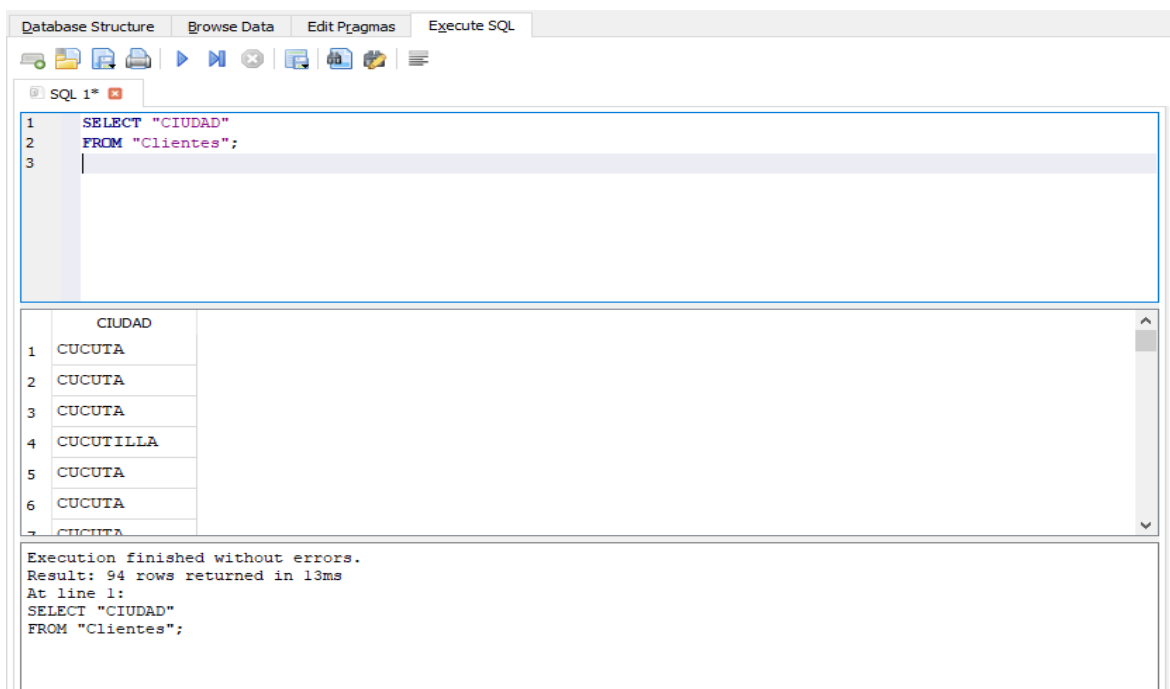


The screenshot shows the SQL Developer interface with the 'Execute SQL' tab active. The query editor contains the statement `SELECT * FROM Clientes ;`. The results pane displays a table with 6 rows and 6 columns. The columns are CLIENTE, C.CÓNIT, TELÉFONO, CELULAR, and two unlabeled columns. The data is as follows:

	CLIENTE	C.CÓNIT	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	CALLI
6	ANGUAYA CACHIMUEL ROSA MARIA	680.539.865-6	3164343282	3164343282	CC OI

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

2. Selección de una columna de una tabla

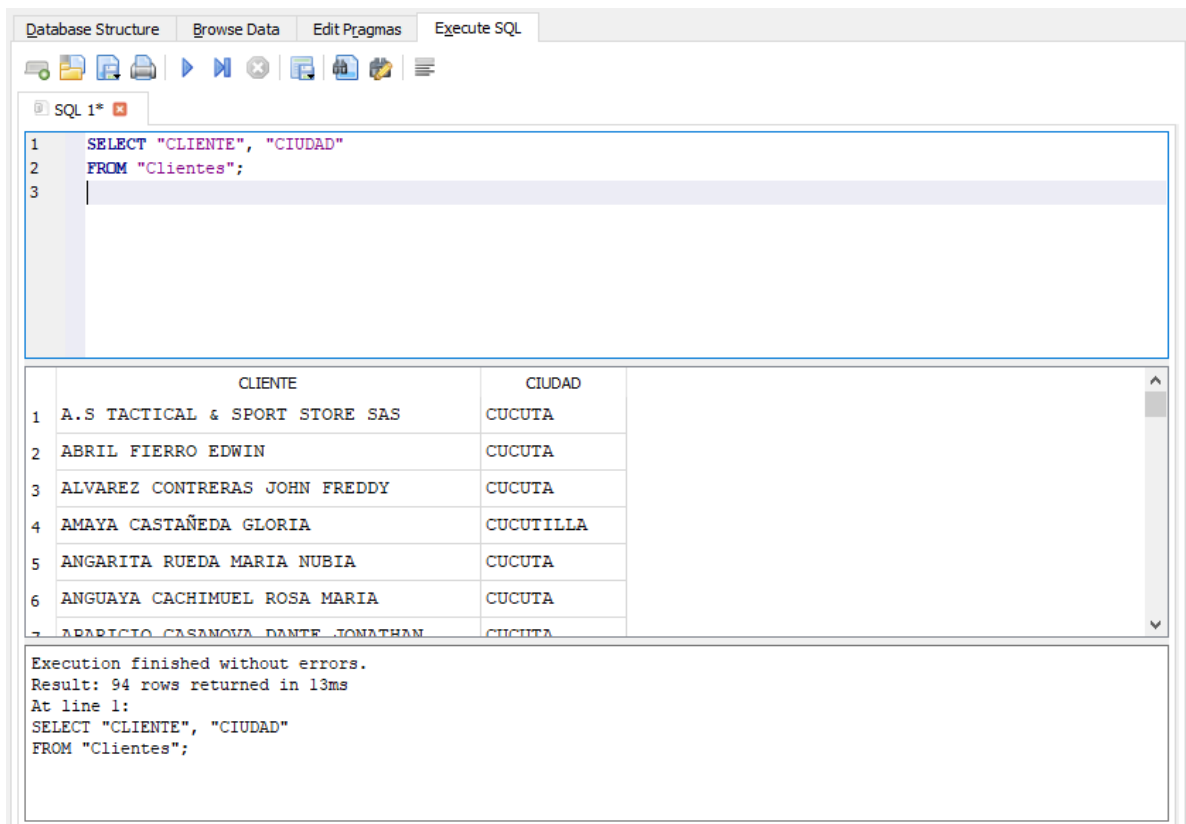


The screenshot shows the SQL Developer interface with the 'Execute SQL' tab active. The query editor contains the statement `SELECT "CIUDAD" FROM "Clientes";`. The results pane displays a table with 6 rows and 1 column. The column is labeled CIUDAD. The data is as follows:

	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



The screenshot shows the SQL Developer interface with the 'Execute SQL' tab selected. The SQL editor contains the following query:

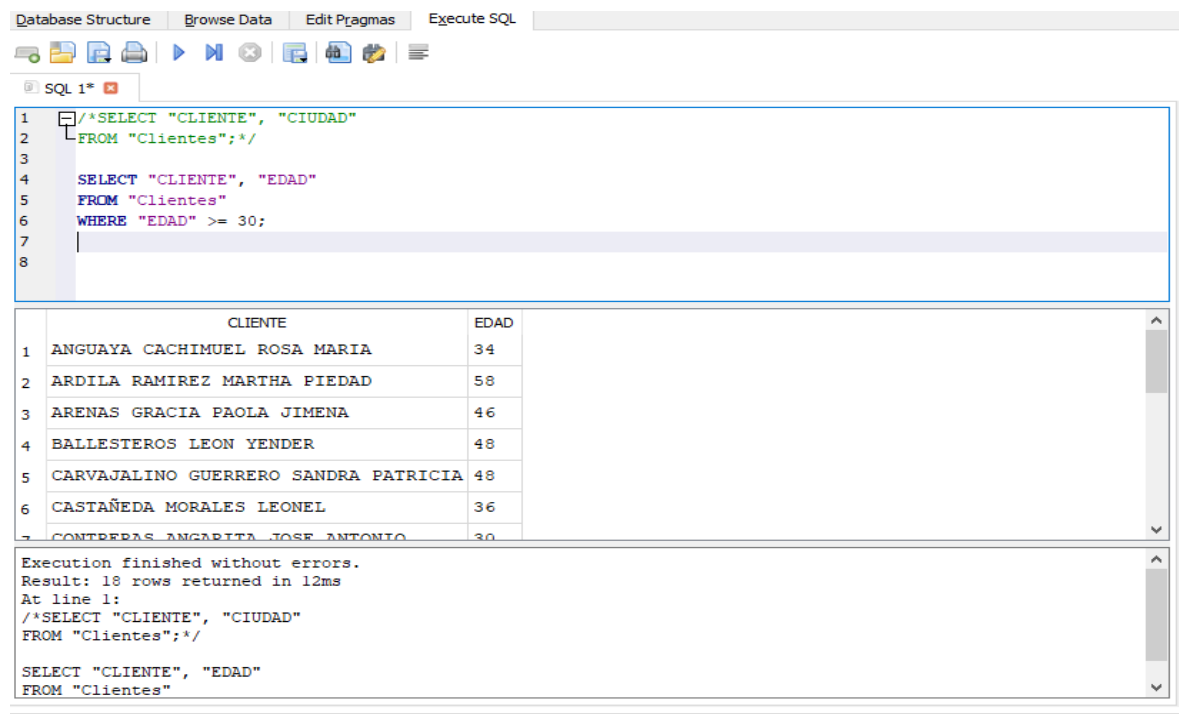
```
1 SELECT "CLIENTE", "CIUDAD"
2 FROM "Clientes";
3
```

The results are displayed in a table with two columns: CLIENTE and CIUDAD. The table contains 9 rows of data.

	CLIENTE	CIUDAD
1	A.S TACTICAL & SPORT STORE SAS	CUCUTA
2	ABRIL FIERRO EDWIN	CUCUTA
3	ALVAREZ CONTRERAS JOHN FREDDY	CUCUTA
4	AMAYA CASTAÑEDA GLORIA	CUCUTILLA
5	ANGARITA RUEDA MARIA NUBIA	CUCUTA
6	ANGUAYA CACHIMUEL ROSA MARIA	CUCUTA
7	ARABICIO CASANOVA DANTE JONATHAN	CUCUTA
8		
9		

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

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



The screenshot shows the SQL Developer interface with the 'Execute SQL' tab selected. The SQL editor contains the following query:

```
1 /*SELECT "CLIENTE", "CIUDAD"
2 FROM "Clientes";*/
3
4 SELECT "CLIENTE", "EDAD"
5 FROM "Clientes"
6 WHERE "EDAD" >= 30;
7
8
```

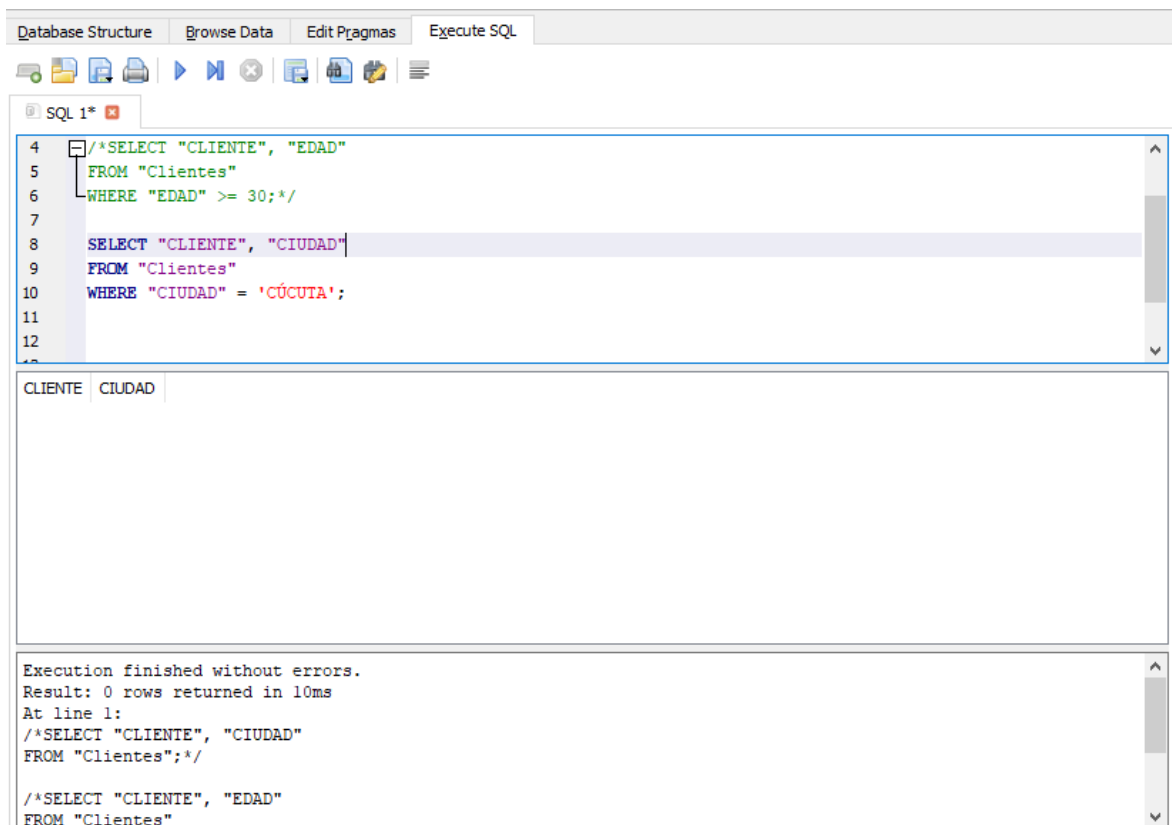
The results are displayed in a table with two columns: CLIENTE and EDAD. The table contains 7 rows of data.

	CLIENTE	EDAD
1	ANGUAYA CACHIMUEL ROSA MARIA	34
2	ARDILA RAMIREZ MARTHA PIEDAD	58
3	ARENAS GRACIA PAOLA JIMENA	46
4	BALLESTEROS LEON YENDER	48
5	CARVAJALINO GUERRERO SANDRA PATRICIA	48
6	CASTAÑEDA MORALES LEONEL	36
7	CONTRERAS ANGARITA JOSE ANTONIO	30

Execution finished without errors.
Result: 18 rows returned in 12ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

SELECT "CLIENTE", "EDAD"
FROM "Clientes"

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



Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

```

4  /*SELECT "CLIENTE", "EDAD"
5  FROM "Clientes"
6  WHERE "EDAD" >= 30;*/
7
8  SELECT "CLIENTE", "CIUDAD"
9  FROM "Clientes"
10 WHERE "CIUDAD" = 'CÚCUTA';
11
12

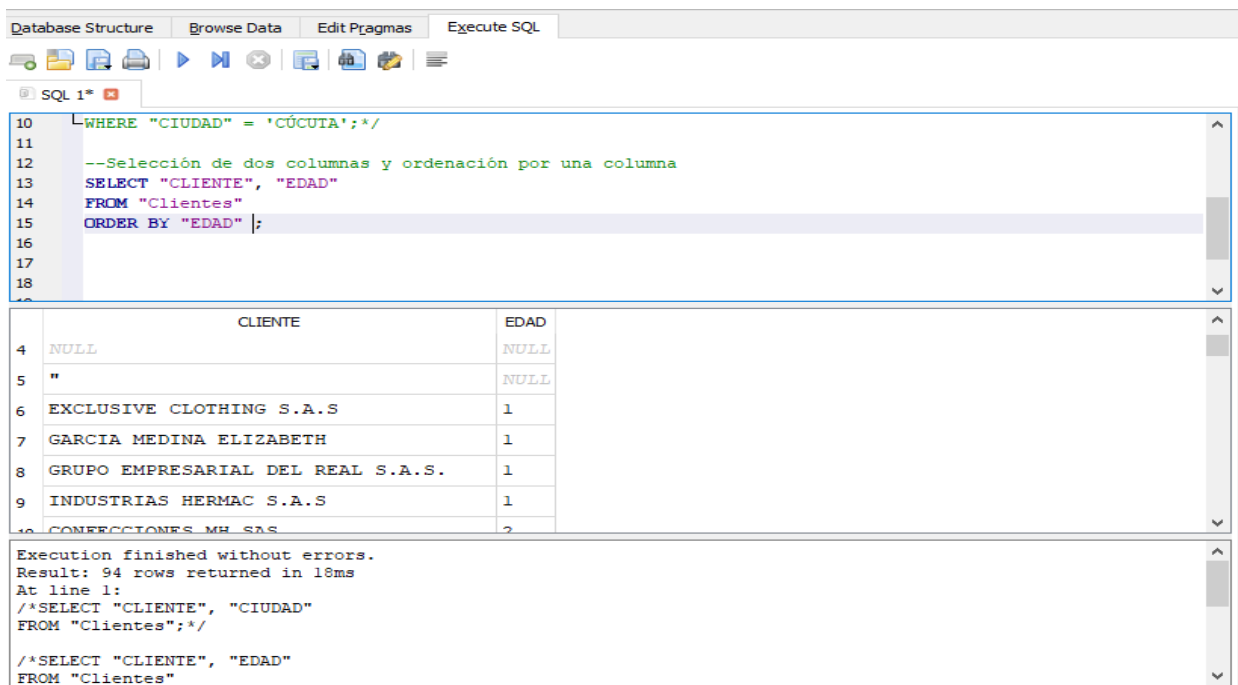
```

CLIENTE	CIUDAD
---------	--------

Execution finished without errors.
Result: 0 rows returned in 10ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

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

6. Selección de dos columnas y ordenación por una columna



Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

```

10 WHERE "CIUDAD" = 'CÚCUTA';*/
11
12 --Selección de dos columnas y ordenación por una columna
13 SELECT "CLIENTE", "EDAD"
14 FROM "Clientes"
15 ORDER BY "EDAD" |;
16
17
18

```

CLIENTE	EDAD
NULL	NULL
"	NULL
EXCLUSIVE CLOTHING S.A.S	1
GARCIA MEDINA ELIZABETH	1
GRUPO EMPRESARIAL DEL REAL S.A.S.	1
INDUSTRIAS HERMAC S.A.S	1
CONFECCIONES MH SAS	2

Execution finished without errors.
Result: 94 rows returned in 10ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

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

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

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```

12 --Selección de dos columnas y ordenación por una columna
13 /*SELECT "CLIENTE", "EDAD"
14 FROM "Clientes"
15 ORDER BY "EDAD" ;*/
16 --Selección de Dos Columnas y Ordenación Descendente por una Columna
17 SELECT "CLIENTE", "EDAD"
18 FROM "Clientes"
19 ORDER BY "EDAD" DESC;
20

```

	CLIENTE	EDAD
1	YÁÑEZ SILVA SAMUEL	60
2	ARDILA RAMIREZ MARTHA PIEDAD	58
3	RUEDA ALARCON MONICA LILIAN	49
4	BALLESTEROS LEON YENDER	48
5	CARVAJALINO GUERRERO SANDRA PATRICIA	48
6	SANABRIA CHINCHILLA YEILY KARINA	48
7	RODRIGUEZ BUENAHORA ROSAIRA	47

Execution finished without errors.
Result: 94 rows returned in 11ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

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

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

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```

20
21 --Seleccionando Dos Columnas de una Tabla y Ordenando Descendentemente por Dos Columnas
22 SELECT "CLIENTE", "CIUDAD"
23 FROM "Clientes"
24 ORDER BY "CIUDAD" DESC, "CLIENTE" DESC;
25
26
27
28

```

	CLIENTE	CIUDAD
1	RAMIREZ DIAZ GEOVANNY	PIEDRECUESTA
2	VARELA RUIZ DANIELA CHARLOT	PAMPLONA
3	RUEDA ALARCON MONICA LILIAN	PAMPLONA
4	ORTEGA RODRIGUEZ LIDA BIBIANA	PAMPLONA
5	OBAYI SAS	PAMPLONA
6	DURAN MANDON HILARY VALENTINA	PAMPLONA
7	CORTEZ CACHA JACKSON JOSE	PAMPLONA

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

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

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

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

```

21 --Seleccionando Dos Columnas de una Tabla y Ordenando Descendentemente por Dos Columnas
22 /*SELECT "CLIENTE", "CIUDAD"
23 FROM "Clientes"
24 ORDER BY "CIUDAD" DESC, "CLIENTE" DESC;*/
25
26 --Selección de dos columnas con una condición lógica compleja en WHERE
27 SELECT "CLIENTE", "EDAD"
28 FROM "Clientes"
29 WHERE "EDAD" > 50 OR "EDAD" < 20;

```

	CLIENTE	EDAD
1	A.S TACTICAL & SPORT STORE SAS	5
2	ALVAREZ CONTRERAS JOHN FREDDY	15
3	AMAYA CASTAÑEDA GLORIA	14
4	APARICIO CASANOVA DANTE JONATHAN	17
5	ARDILA RAMIREZ MARTHA PIEDAD	58
6	BASTO MERCEDES	17
7	CASTILLA HERNANDEZ ALFREDO JOSE	14

Execution finished without errors.
Result: 48 rows returned in 12ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

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

10. Cálculos sencillos sobre columnas

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1*

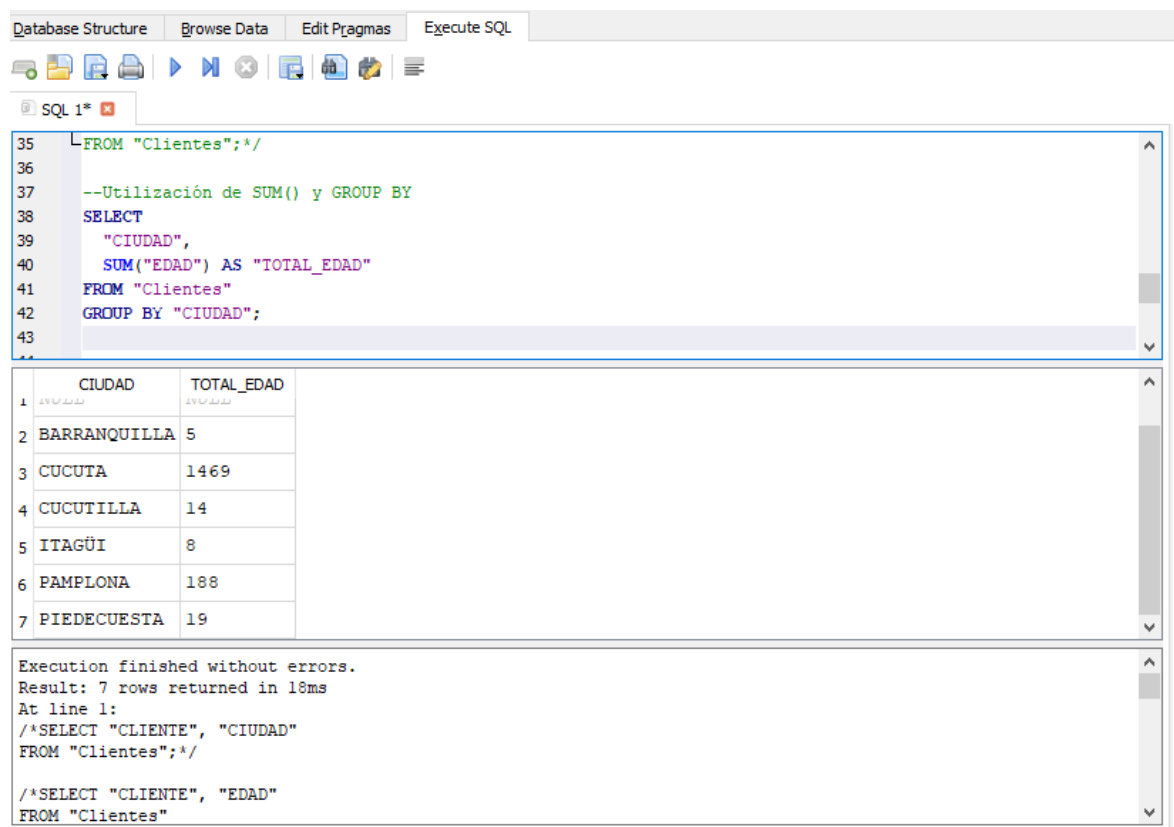
```

28 FROM "Clientes"
29 WHERE "EDAD" > 50 OR "EDAD" < 20;*/
30
31 --Cálculos sencillos sobre columnas
32 SELECT
33     "CLIENTE",
34     "EDAD" + 10 AS "EDAD_EN_10_AÑOS"
35 FROM "Clientes";
36

```

	CLIENTE	EDAD_EN_10_AÑOS
1	A.S TACTICAL & SPORT STORE SAS	15
2	ABRIL FIERRO EDWIN	30
3	ALVAREZ CONTRERAS JOHN FREDDY	25
4	AMAYA CASTAÑEDA GLORIA	24
5	ANGARITA RUEDA MARIA NUBIA	36
6	ANGUAYA CACHIMUEL ROSA MARIA	44
7	APARICIO CASANOVA DANTE JONATHAN	27

11. Utilización de SUM() y GROUP BY



The screenshot shows the SQL Developer interface with the 'Execute SQL' tab active. The SQL editor contains the following query:

```
35 FROM "Clientes";*/
36
37 --Utilización de SUM() y GROUP BY
38 SELECT
39     "CIUDAD",
40     SUM("EDAD") AS "TOTAL_EDAD"
41 FROM "Clientes"
42 GROUP BY "CIUDAD";
43
```

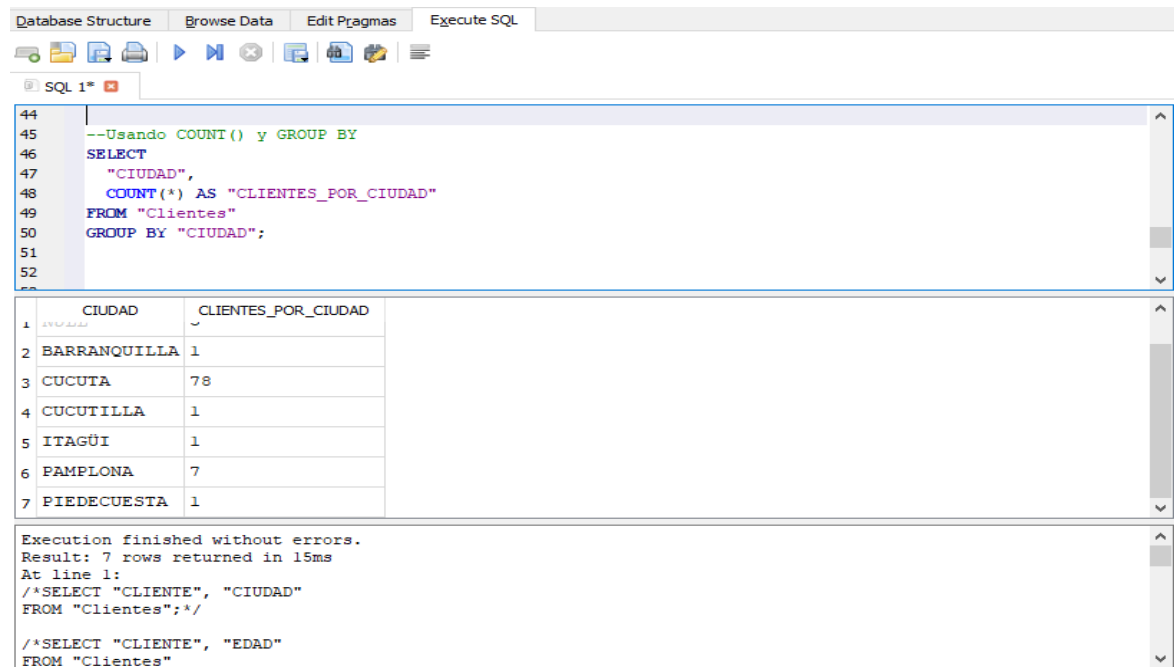
The result set is displayed below the query:

	CIUDAD	TOTAL_EDAD
1	BARRANQUILLA	5
3	CUCUTA	1469
4	CUCUTILLA	14
5	ITAGÜI	8
6	PAMPLONA	188
7	PIEDRECUESTA	19

Execution finished without errors.
Result: 7 rows returned in 18ms
At line 1:
/*SELECT "CLIENTE", "CIUDAD"
FROM "Clientes";*/

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

12. Usando COUNT() y GROUP BY



The screenshot shows the SQL Developer interface with the 'Execute SQL' tab active. The SQL editor contains the following query:

```
44
45 --Usando COUNT() y GROUP BY
46 SELECT
47     "CIUDAD",
48     COUNT(*) AS "CLIENTES_POR_CIUADAD"
49 FROM "Clientes"
50 GROUP BY "CIUDAD";
51
52
```

The result set is displayed below the query:

	CIUDAD	CLIENTES_POR_CIUADAD
2	BARRANQUILLA	1
3	CUCUTA	78
4	CUCUTILLA	1
5	ITAGÜI	1
6	PAMPLONA	7
7	PIEDRECUESTA	1

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

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

13. Uso de AVG() y GROUP BY

Database Structure Browse Data Edit Pragma 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	PIEDRECUESTA	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 Pragma 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
66

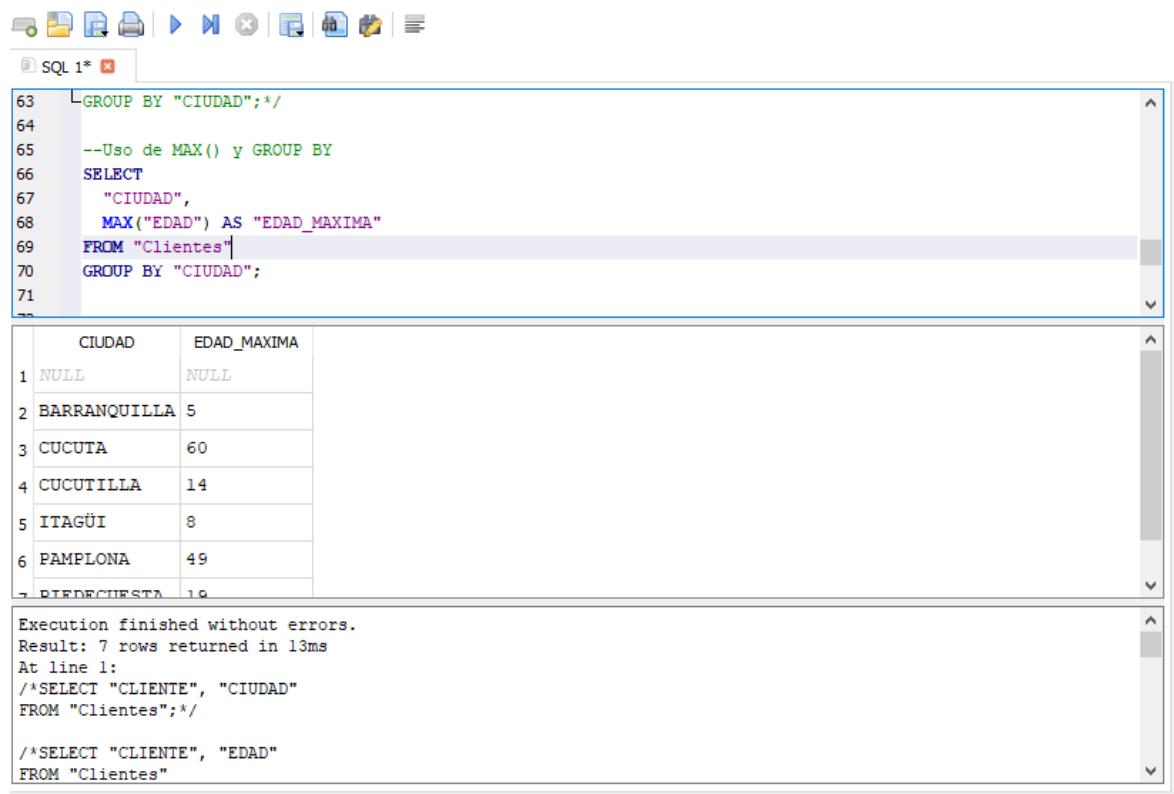
```

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



The screenshot shows the SQL Developer interface with a query window titled 'SQL 1*'. The query is as follows:

```
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
```

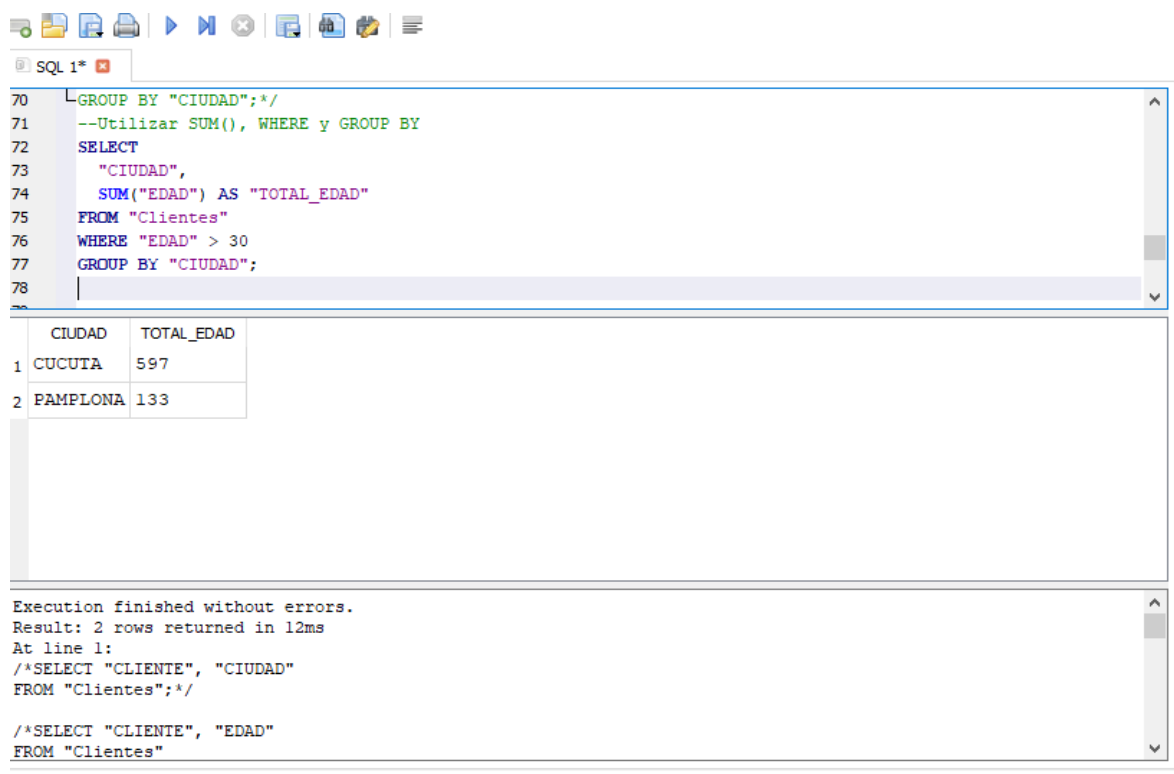
The result set displays the maximum age for each city:

	CIUDAD	EDAD_MAXIMA
1	NULL	NULL
2	BARRANQUILLA	5
3	CUCUTA	60
4	CUCUTILLA	14
5	ITAGÜI	8
6	PAMPLONA	49
7	PIEDRECUESTA	16

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



The screenshot shows the SQL Developer interface with a query window titled 'SQL 1*'. The query is as follows:

```
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
```

The result set displays the total age for cities where the age is greater than 30:

	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";*/

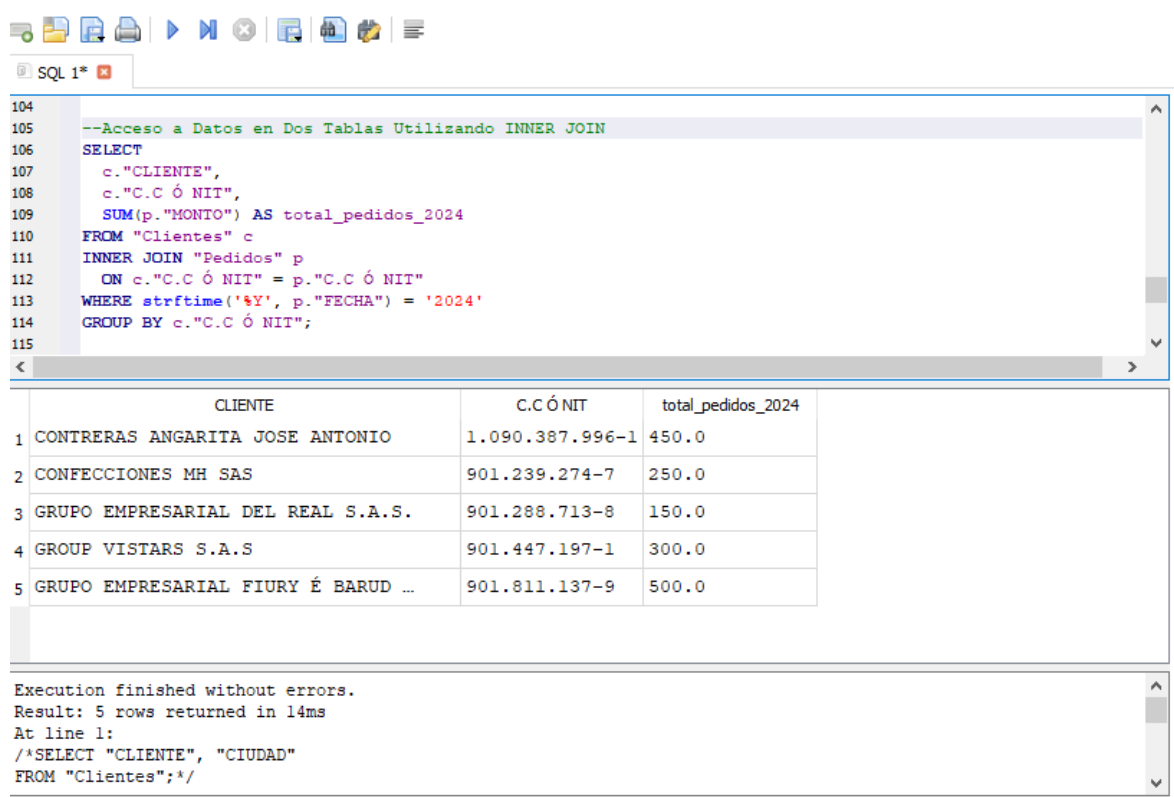
/*SELECT "CLIENTE", "EDAD"
FROM "Clientes"

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

```
SQL 1*
95 L */
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



The screenshot shows a SQL Developer window with a query titled "SQL 1*" containing the following SQL code:

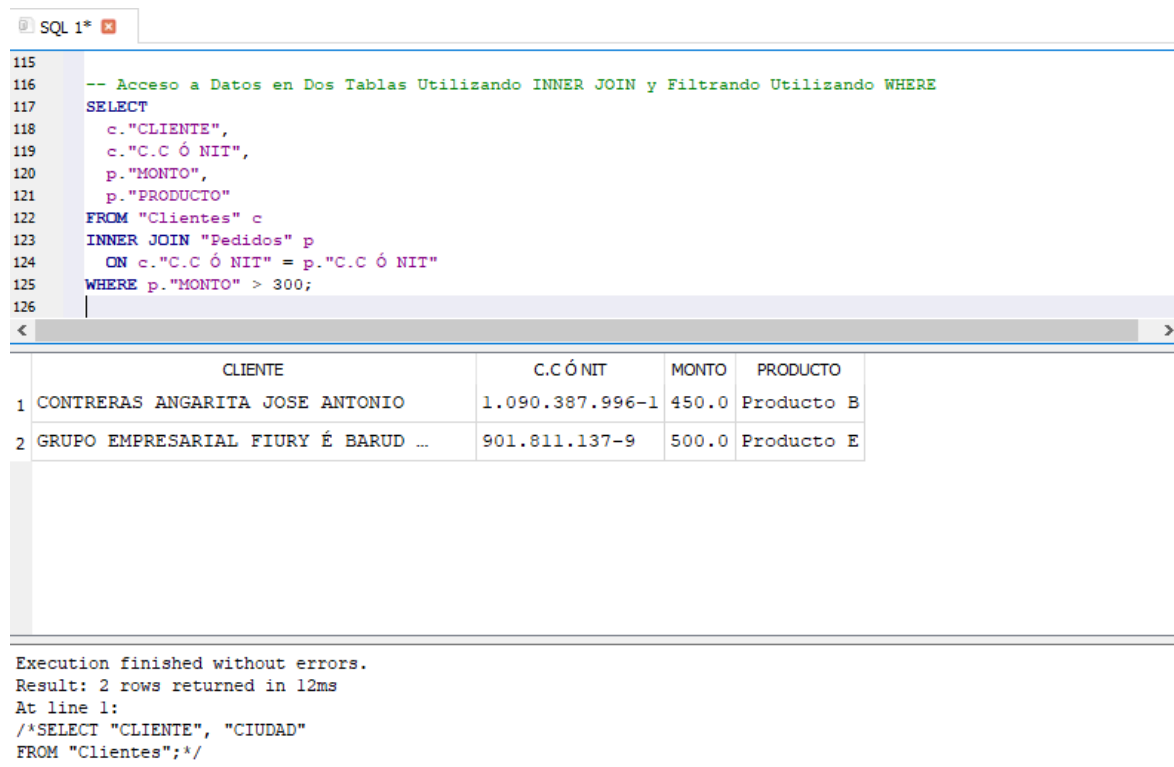
```
104
105 --Acceso a Datos en Dos Tablas Utilizando INNER JOIN
106 SELECT
107     c."CLIENTE",
108     c."C.C Ó NIT",
109     SUM(p."MONTO") AS total_pedidos_2024
110 FROM "Clientes" c
111 INNER JOIN "Pedidos" p
112     ON c."C.C Ó NIT" = p."C.C Ó NIT"
113 WHERE strftime('%Y', p."FECHA") = '2024'
114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

The results of the query are displayed in a table with 5 rows and 3 columns: CLIENTE, C.C Ó NIT, and total_pedidos_2024.

	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



The screenshot shows a SQL Developer window with a query titled "SQL 1*" containing the following SQL code:

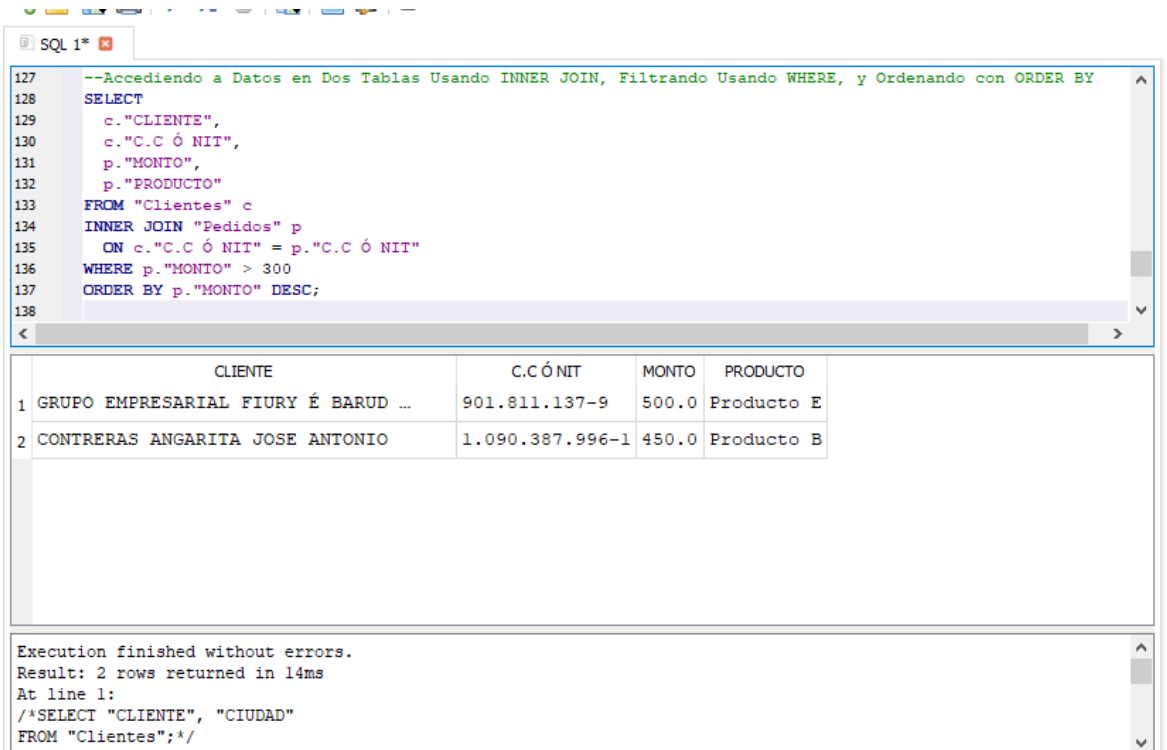
```
115
116 -- Acceso a Datos en Dos Tablas Utilizando INNER JOIN y Filtrando Utilizando WHERE
117 SELECT
118     c."CLIENTE",
119     c."C.C Ó NIT",
120     p."MONTO",
121     p."PRODUCTO"
122 FROM "Clientes" c
123 INNER JOIN "Pedidos" p
124     ON c."C.C Ó NIT" = p."C.C Ó NIT"
125 WHERE p."MONTO" >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

The results of the query are displayed in a table with 2 rows and 4 columns: CLIENTE, C.C Ó NIT, MONTO, and PRODUCTO.

	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



The screenshot shows a SQL IDE window titled "SQL 1*". The query editor contains the following SQL code:

```
--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;
```

Below the query editor, the results are displayed in a table with 4 columns: CLIENTE, C.C Ó NIT, MONTO, and PRODUCTO. The table contains 2 rows of data.

	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

At the bottom of the IDE, the execution status is shown: "Execution finished without errors. Result: 2 rows returned in 14ms. At line 1: /*SELECT "CLIENTE", "CIUDAD" FROM "Clientes";*/"