

1. Show the cities of agents booking an order for a customer whose id is 'c002'. Use joins; no subqueries.

```
SELECT a.city
FROM orders o,
      agents a,
      customers c
WHERE o.aid = a.aid
      AND c.cid = o.cid
      AND c.cid = 'c002';
```

The screenshot shows a PostgreSQL query editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" tab, which contains the following SQL code:

```
INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
VALUES(1026, 'may', 'c002', 'a05', 'p03', 800, 740.00);

-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to C

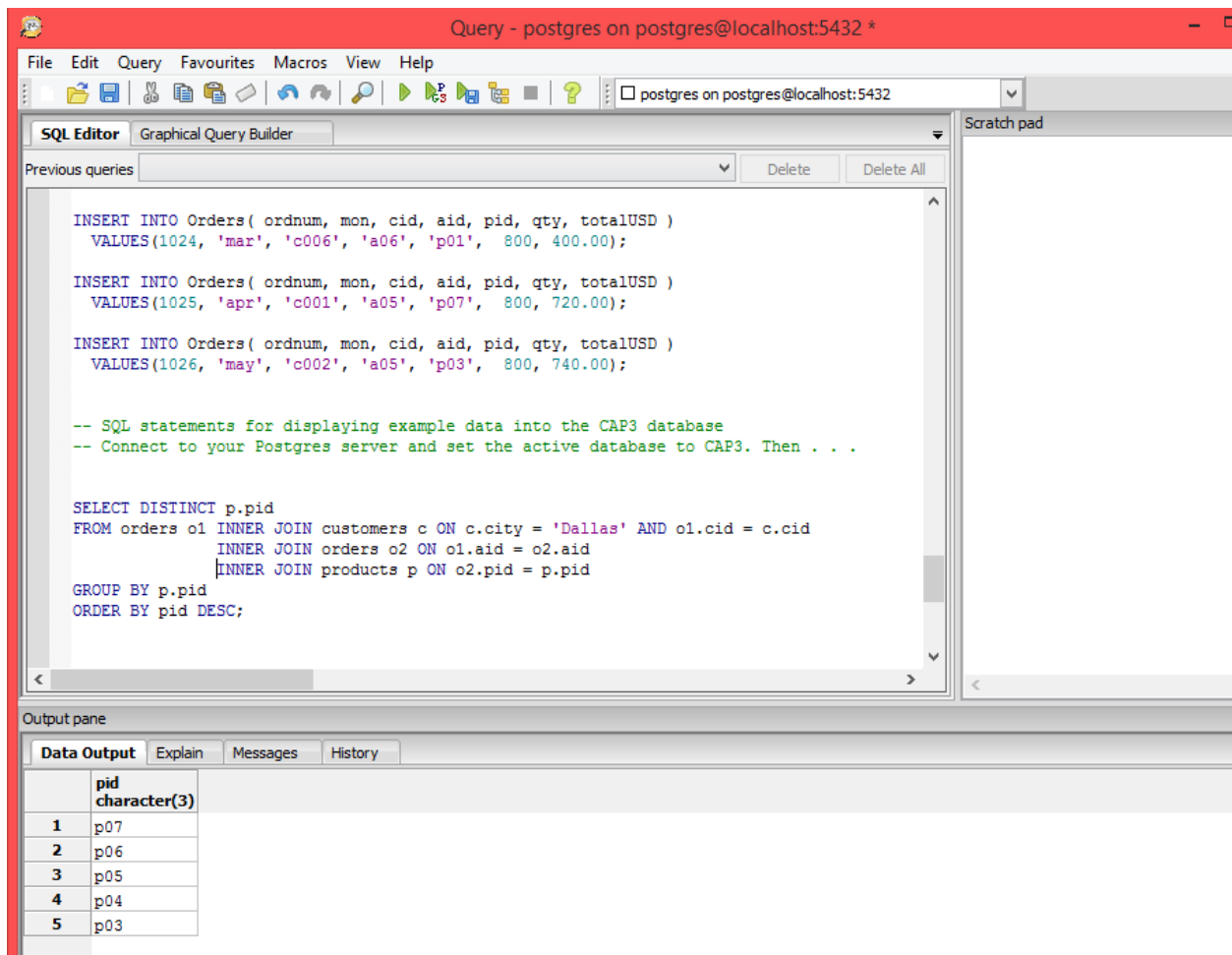
SELECT a.city
FROM orders o,
      agents a,
      customers c
WHERE o.aid = a.aid
      AND c.cid = o.cid
      AND c.cid = 'c002';
```

Below the SQL Editor is the "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, showing the results of the query in a table:

	city text
1	Tokyo
2	Duluth

2. Show the ids of products ordered through any agent who makes at least one order for a customer in Dallas, sorted by pid from highest to lowest. Use joins; no subqueries.

```
SELECT DISTINCT p.pid
FROM orders o1 INNER JOIN customers c ON c.city = 'Dallas' AND o1.cid = c.cid
      INNER JOIN orders o2 ON o1.aid = o2.aid
      INNER JOIN products p ON o2.pid = p.pid
GROUP BY p.pid
ORDER BY pid DESC;
```



The screenshot shows a PostgreSQL SQL Editor window titled "Query - postgres on postgres@localhost:5432 *". The editor contains the following SQL code:

```
INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
VALUES(1024, 'mar', 'c006', 'a06', 'p01', 800, 400.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
VALUES(1025, 'apr', 'c001', 'a05', 'p07', 800, 720.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
VALUES(1026, 'may', 'c002', 'a05', 'p03', 800, 740.00);

-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

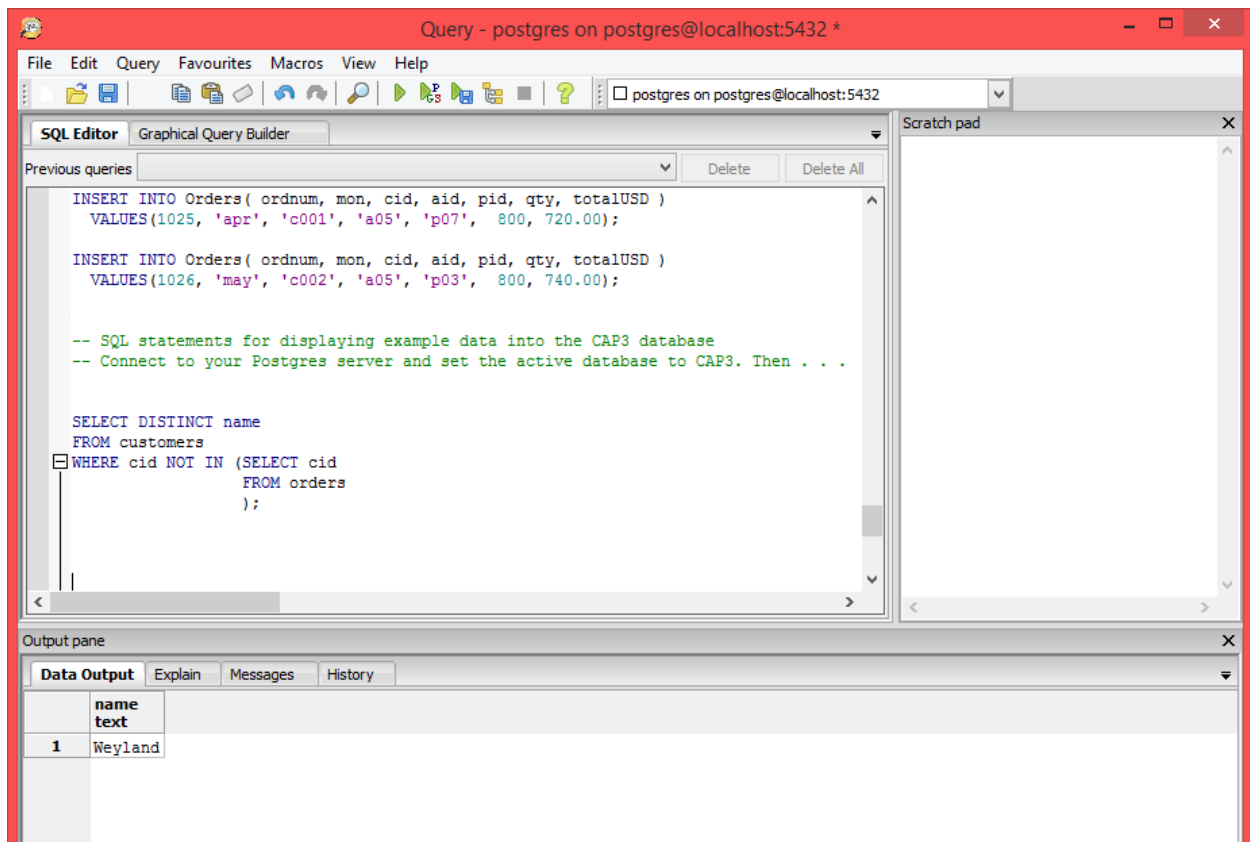
SELECT DISTINCT p.pid
FROM orders o1 INNER JOIN customers c ON c.city = 'Dallas' AND o1.cid = c.cid
      INNER JOIN orders o2 ON o1.aid = o2.aid
      INNER JOIN products p ON o2.pid = p.pid
GROUP BY p.pid
ORDER BY pid DESC;
```

The Output pane at the bottom shows the results of the query:

	pid character(3)
1	p07
2	p06
3	p05
4	p04
5	p03

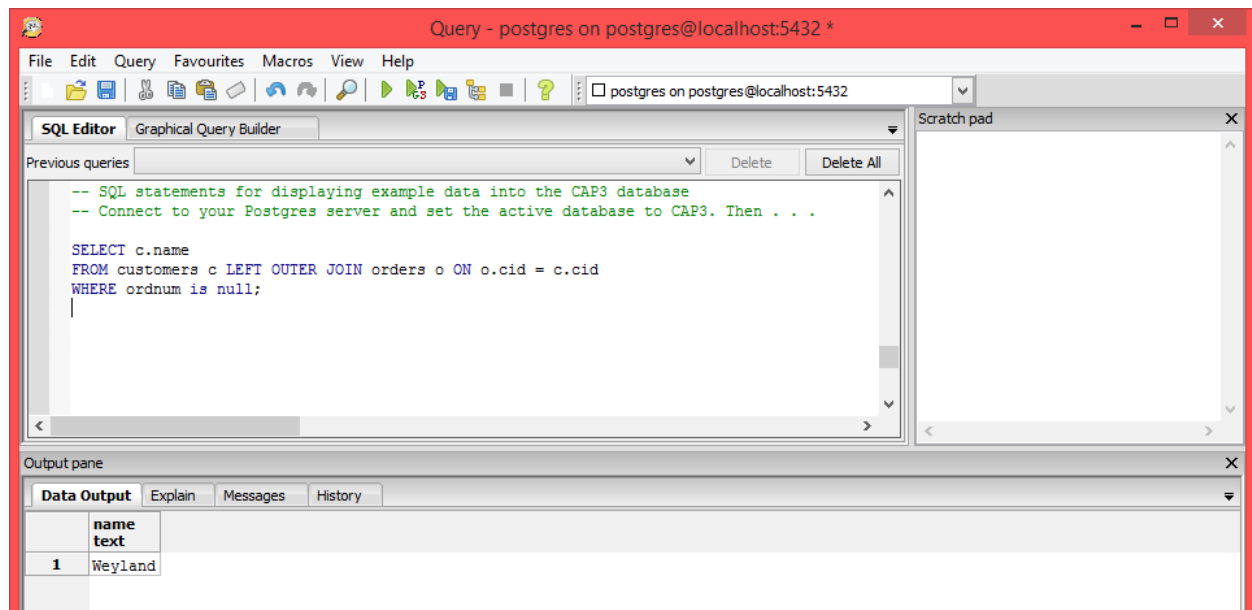
3. Show the names of customers who have never placed an order. Use a subquery.

```
SELECT DISTINCT name
FROM customers
WHERE cid NOT IN (SELECT cid
                  FROM orders
                  );
```



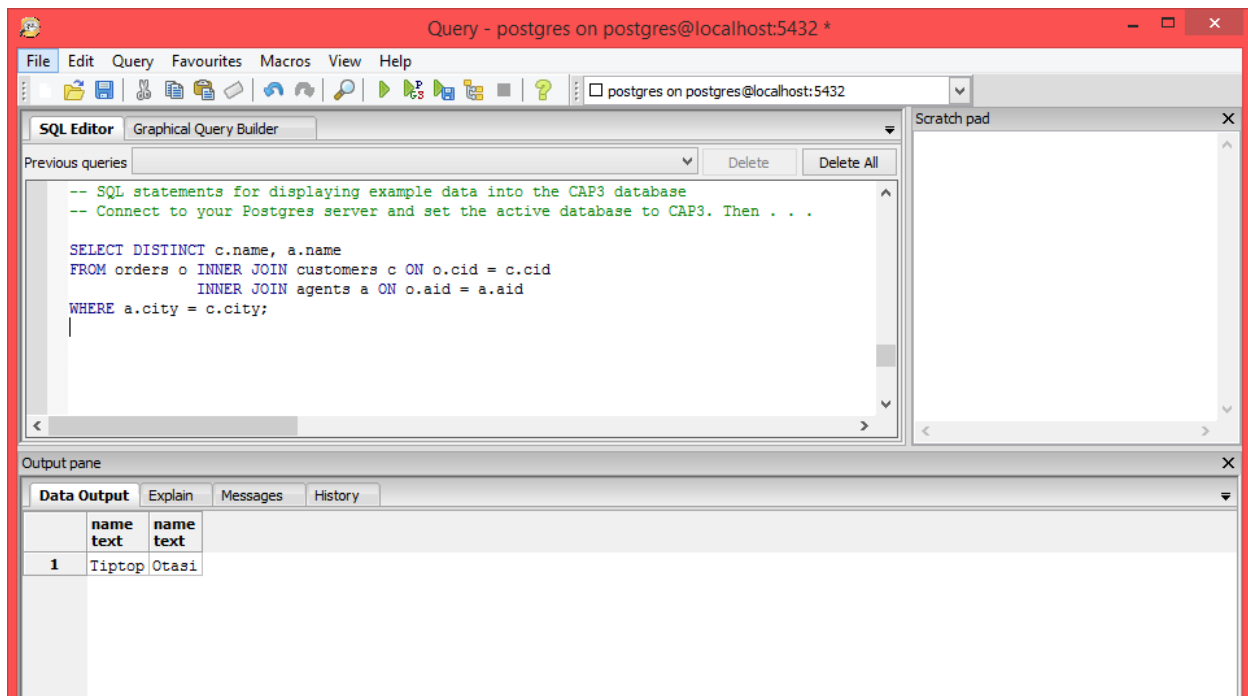
4. Show the names of customers who have never placed an order. Use an outer join.

```
SELECT c.name
FROM customers c LEFT OUTER JOIN orders o ON o.cid = c.cid
WHERE ordnum is null;
```



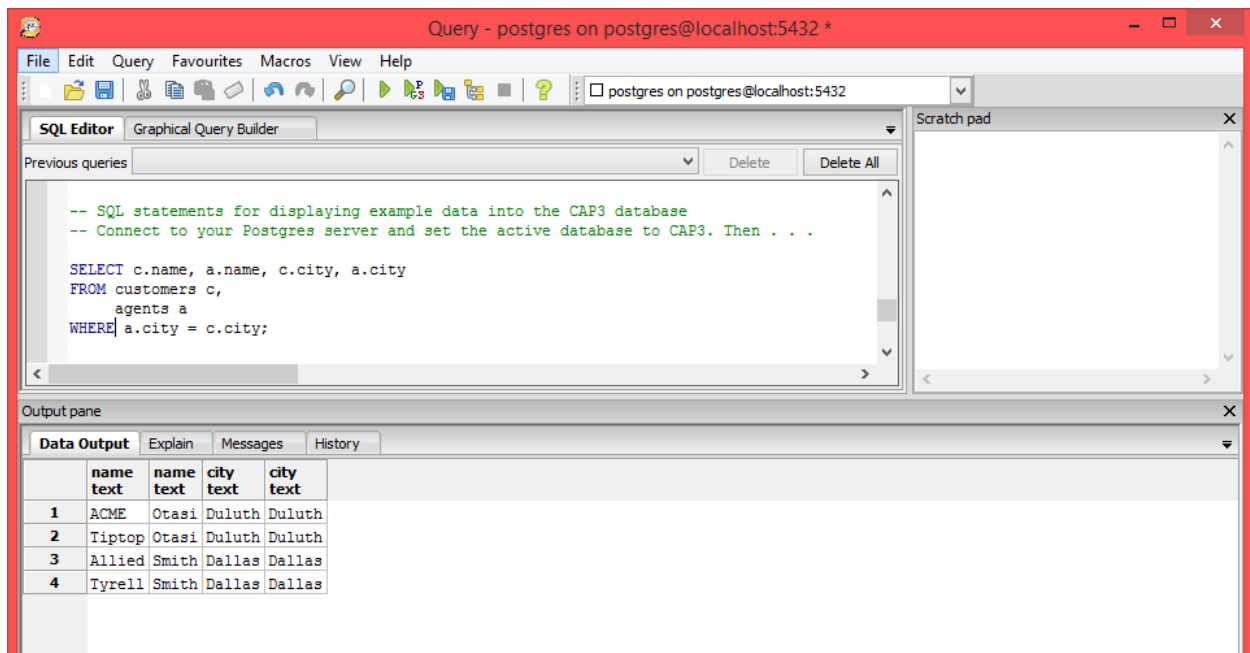
5. Show the names of customers who placed at least one order through an agent in their own city, along with those agent(s) names.

```
SELECT DISTINCT c.name, a.name
FROM orders o INNER JOIN customers c ON o.cid = c.cid
            INNER JOIN agents a ON o.aid = a.aid
WHERE a.city = c.city;
```



6. Show the names of customers and agents living in the same city, along with the name of the shared city, regardless of whether or not the customer has ever placed an order with that agent.

```
SELECT c.name, a.name, c.city, a.city
FROM customers c,
      agents a
WHERE a.city = c.city;
```



The screenshot shows a PostgreSQL query editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" with a "Graphical Query Builder" tab. The query text is:

```
-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

SELECT c.name, a.name, c.city, a.city
FROM customers c,
      agents a
WHERE a.city = c.city;
```

Below the editor is the "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, showing a table with 4 rows and 4 columns:

	name text	name text	city text	city text
1	ACME	Otasi	Duluth	Duluth
2	Tiptop	Otasi	Duluth	Duluth
3	Allied	Smith	Dallas	Dallas
4	Tyrell	Smith	Dallas	Dallas

7. Show the name and city of customers who live in the city that makes the fewest different kinds of products. (Hint: Use count and group by on the Products table.)

```
SELECT c.name, p1.city
FROM products p1,
     products p2,
     customers c
WHERE c.city = p1.city
     AND p1.city > p2.city
GROUP BY p1.city, c.name
ORDER BY count(p1.city);
```

Or

```
SELECT name, city
FROM customers
WHERE city IN (SELECT city
              FROM products
              GROUP BY city
              ORDER BY count(city)
              LIMIT 1
              );
```

The screenshot shows a PostgreSQL query editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" with a "Graphical Query Builder" tab. It contains a query that filters customers by the city with the fewest products. The "Output pane" at the bottom shows the "Data Output" tab with a table of results.

```
-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to C

SELECT name, city
FROM customers
WHERE city IN (SELECT city
              FROM products
              GROUP BY city
              ORDER BY count(city)
              LIMIT 1
              );
```

	name	city
	text	text
1	Tiptop	Duluth
2	ACME	Duluth