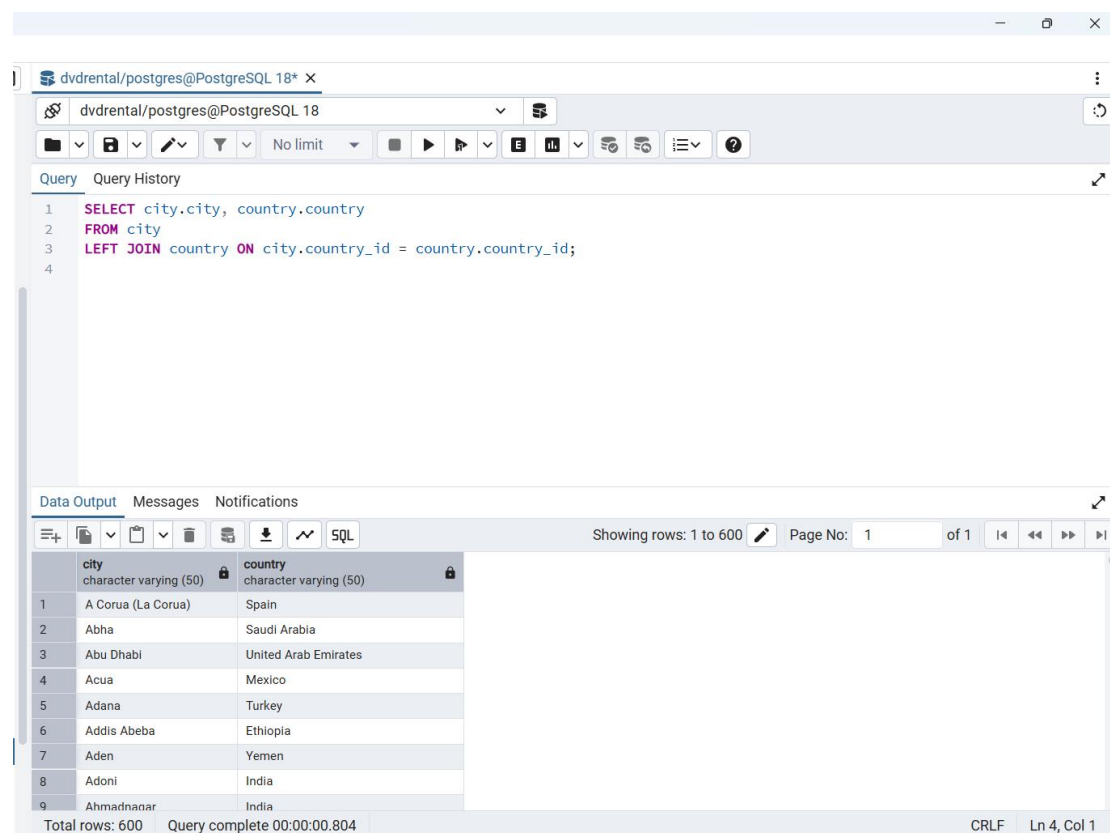


## Assignment 10: DVD Rental JOIN Queries

This assignment focuses on practicing **LEFT JOIN**, **RIGHT JOIN**, and **FULL JOIN** operations using the DVD Rental sample database. The goal is to understand how relational tables connect through foreign keys and how different JOIN types affect the result set.

### Query 1

**Description:** Retrieve the city and country names together by joining the city and country tables. This query demonstrates how location data is linked through the `country_id` field and how a **LEFT JOIN** keeps all rows from the left table.



The screenshot shows a PostgreSQL query editor interface. The query being executed is:

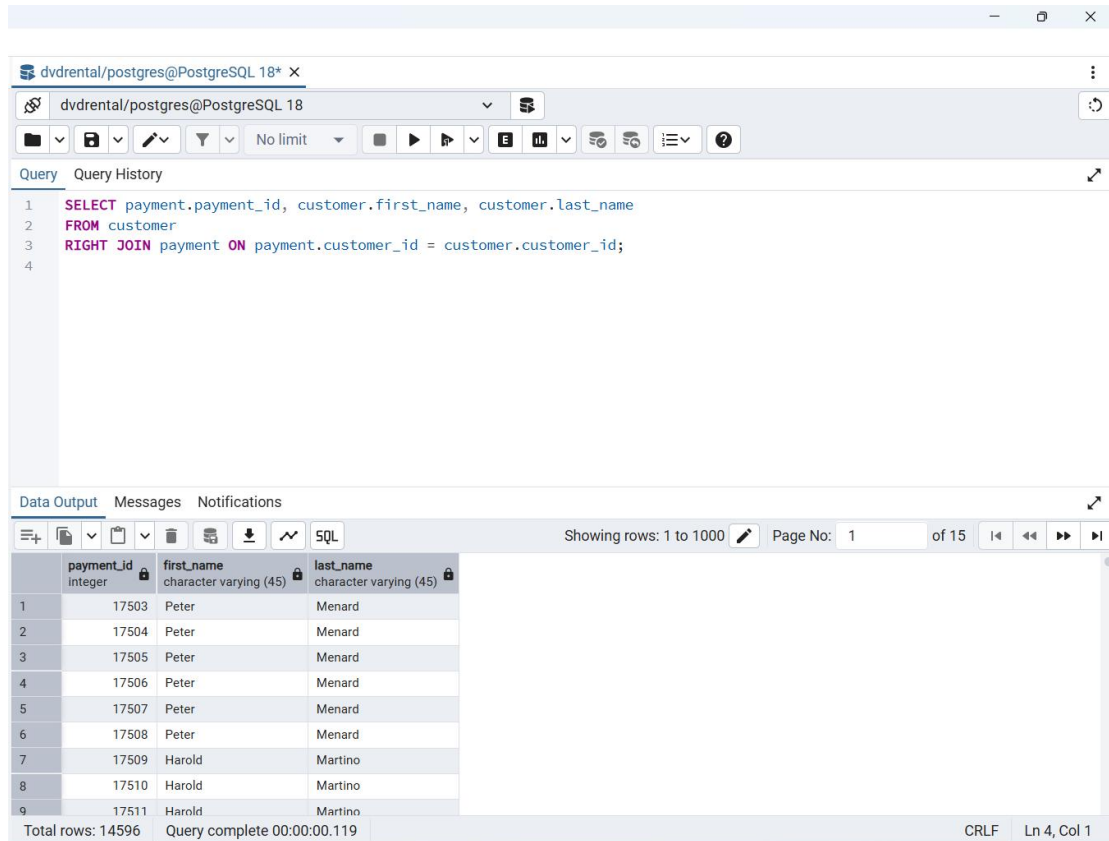
```
1 SELECT city.city, country.country
2 FROM city
3 LEFT JOIN country ON city.country_id = country.country_id;
4
```

The results are displayed in a table with two columns: `city` and `country`. The first 9 rows are shown, with a total of 600 rows. The status bar indicates the query is complete and shows the current line and column.

	city	country
1	A Corua (La Corua)	Spain
2	Abha	Saudi Arabia
3	Abu Dhabi	United Arab Emirates
4	Acua	Mexico
5	Adana	Turkey
6	Addis Abeba	Ethiopia
7	Aden	Yemen
8	Adoni	India
9	Ahmadnagar	India

## Query 2

**Description:** The payment\_id along with the customer's first\_name and last\_name. This query shows how RIGHT JOIN ensures that all payment records appear, even if some payments are not associated with a customer.



The screenshot shows a PostgreSQL query editor window titled "dvdrental/postgres@PostgreSQL 18\* X". The query is as follows:

```
1 SELECT payment.payment_id, customer.first_name, customer.last_name
2 FROM customer
3 RIGHT JOIN payment ON payment.customer_id = customer.customer_id;
4
```

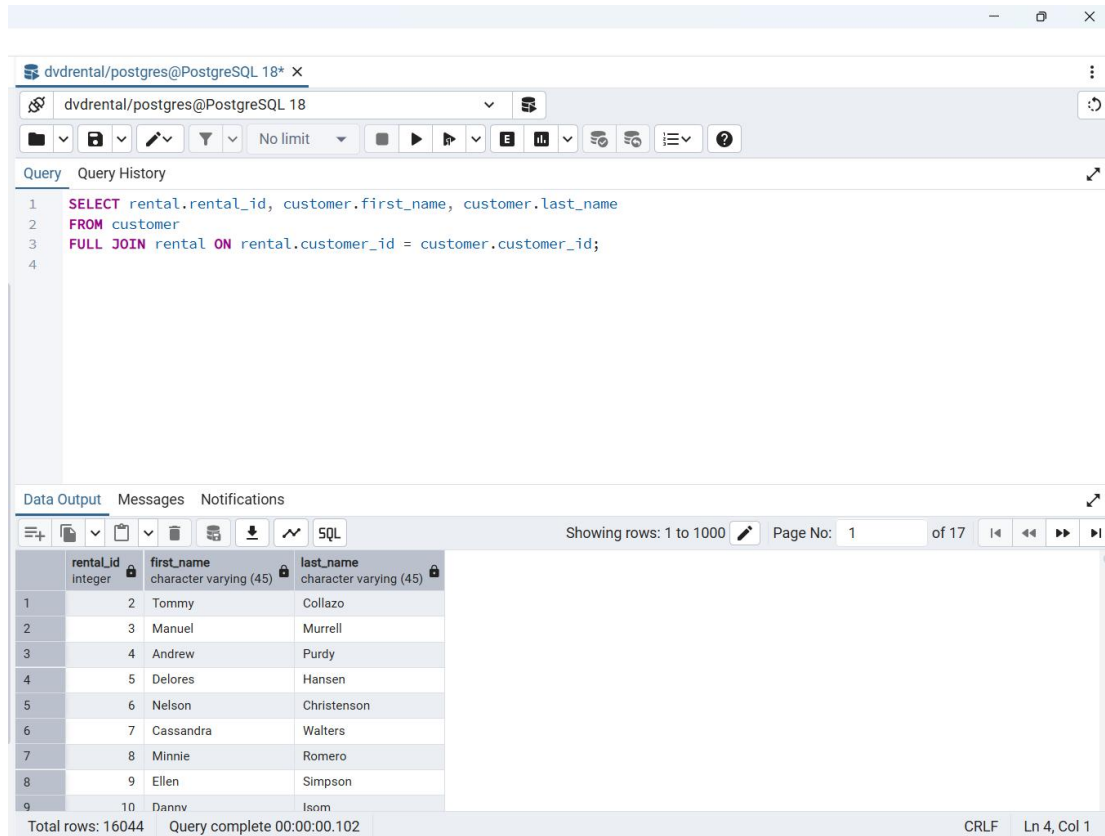
The "Data Output" tab is selected, showing the results of the query. The results are displayed in a table with the following columns: payment\_id (integer), first\_name (character varying (45)), and last\_name (character varying (45)). The table shows 9 rows of data, with the first 8 rows having a customer associated with them, and the 9th row (payment\_id 17511) having no customer associated with it, demonstrating the effect of the RIGHT JOIN.

	payment_id integer	first_name character varying (45)	last_name character varying (45)
1	17503	Peter	Menard
2	17504	Peter	Menard
3	17505	Peter	Menard
4	17506	Peter	Menard
5	17507	Peter	Menard
6	17508	Peter	Menard
7	17509	Harold	Martino
8	17510	Harold	Martino
9	17511	Harold	Martino

Total rows: 14596 Query complete 00:00:00.119 CRLF Ln 4, Col 1

### Query 3

**Description:** The rental\_id along with the customer's first\_name and last\_name. This query demonstrates how FULL JOIN returns all customers and all rentals, matching them where possible and showing unmatched rows from both sides.



The screenshot shows a PostgreSQL query editor window titled "dvdrental/postgres@PostgreSQL 18\* X". The query is as follows:

```
1 SELECT rental.rental_id, customer.first_name, customer.last_name
2 FROM customer
3 FULL JOIN rental ON rental.customer_id = customer.customer_id;
4
```

The "Data Output" tab is selected, showing the results of the query. The results are displayed in a table with the following columns: rental\_id (integer), first\_name (character varying (45)), and last\_name (character varying (45)). The table shows 10 rows of data, which are the first 10 rows of the 16044 total rows returned by the query.

	rental_id integer	first_name character varying (45)	last_name character varying (45)
1	2	Tommy	Collazo
2	3	Manuel	Murrell
3	4	Andrew	Purdy
4	5	Delores	Hansen
5	6	Nelson	Christenson
6	7	Cassandra	Walters
7	8	Minnie	Romero
8	9	Ellen	Simpson
9	10	Dannv	Isom

At the bottom of the window, it indicates "Total rows: 16044" and "Query complete 00:00:00.102". The status bar shows "CRLF" and "Ln 4, Col 1".