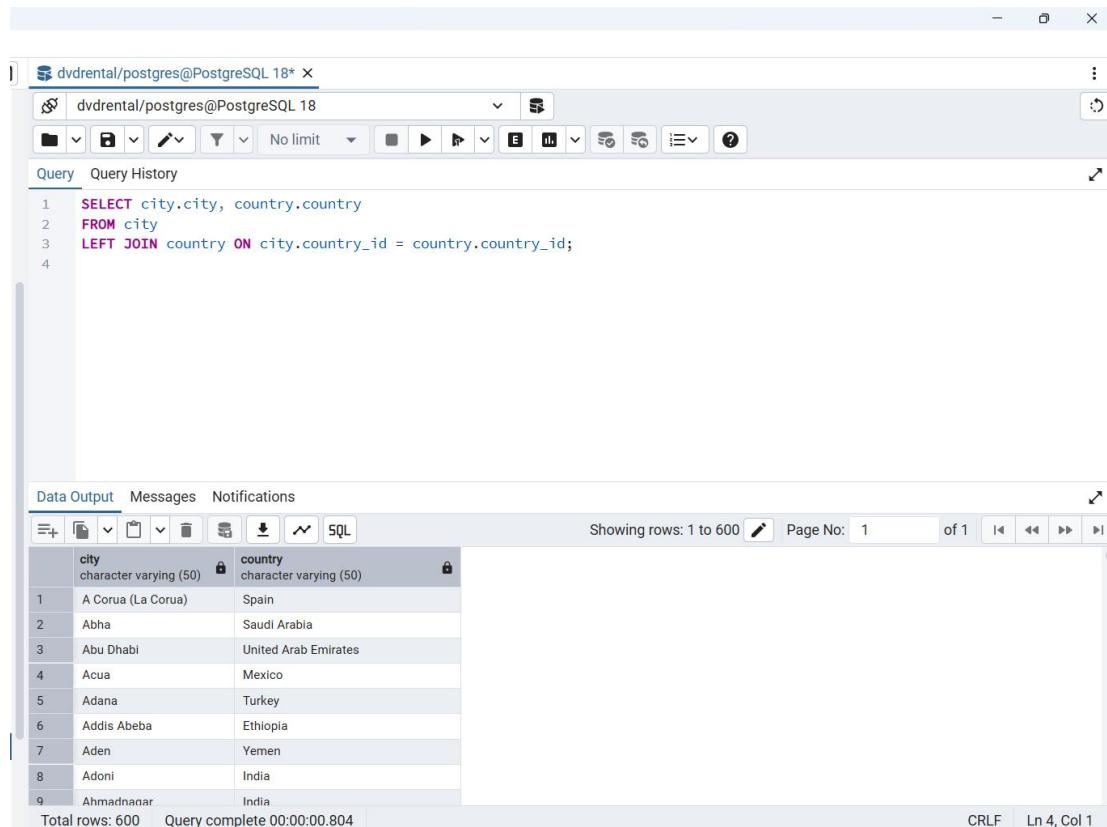


Assignment 10: DVDRental JOIN Queries

This assignment focuses on practicing **LEFT JOIN**, **RIGHT JOIN**, and **FULL JOIN** operations using the DVDRental 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 client interface with the following details:

- Connection:** dvdrental/postgres@PostgreSQL 18*
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Editor:** Shows the SQL query:

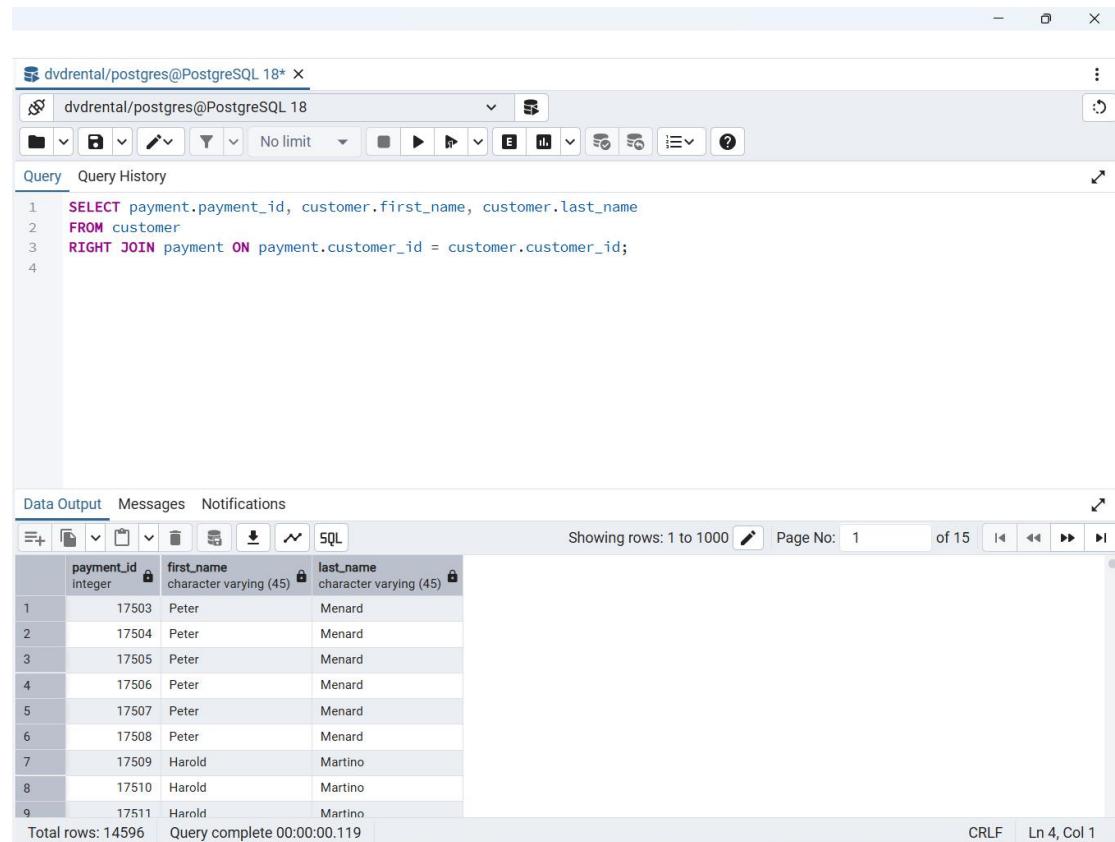
```
1 SELECT city.city, country.country
2 FROM city
3 LEFT JOIN country ON city.country_id = country.country_id;
4
```
- Data Output:** A table showing the joined data. The columns are `city` and `country`. The data includes:

	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

- Message Bar:** Shows "Showing rows: 1 to 600" and "Page No: 1 of 1".
- Bottom Status:** "Total rows: 600 Query complete 00:00:00.804" and "CRLF Ln 4, Col 1".

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 client interface with the following details:

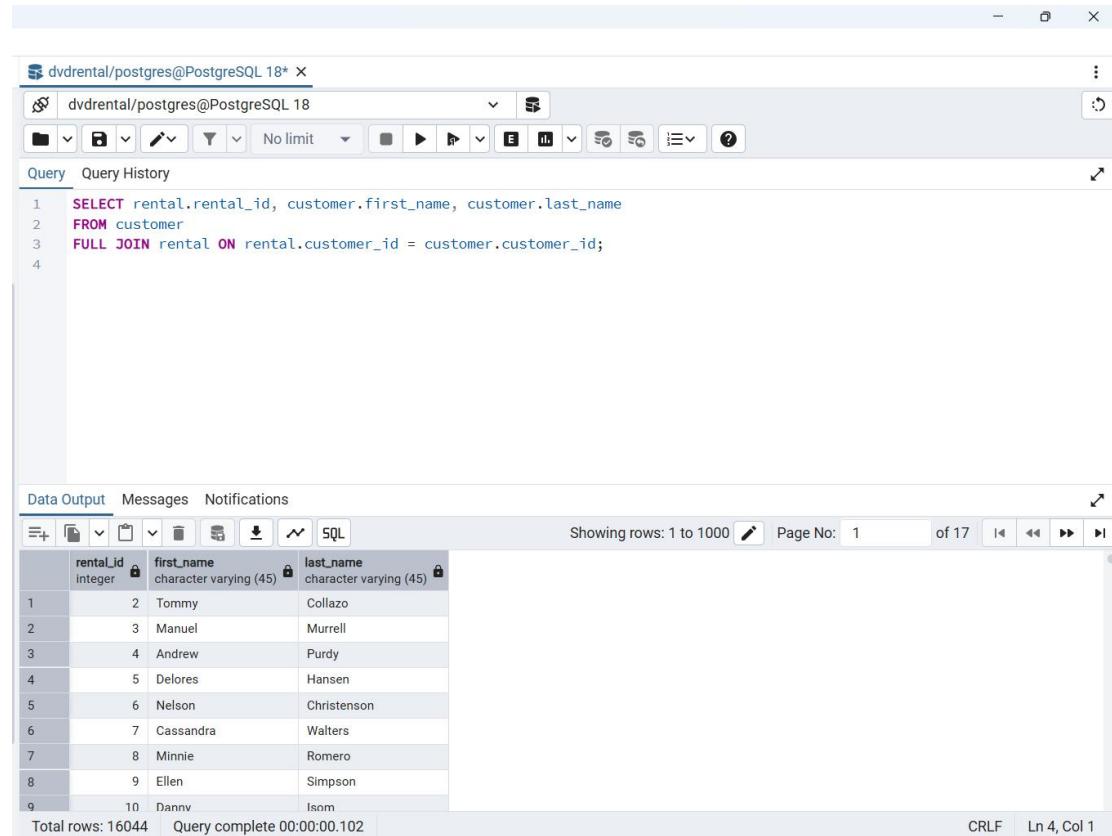
- Connection:** dvdrental/postgres@PostgreSQL 18*
- Toolbar:** Includes icons for file operations, search, and database management.
- Query Tab:** Contains the SQL code:

```
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
```
- Data Output Tab:** Displays the results of the query in a table format. The table has three columns: payment_id, first_name, and last_name. The data shows multiple rows for the same customer_id (17503, 17504, 17505, 17506, 17507, 17508) and two rows for customer_id 17511 (Harold).
- Table Headers:** payment_id (integer), first_name (character varying(45)), last_name (character varying(45)).
- Table Data:**

	payment_id	first_name	last_name
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
- Message Bar:** Showing rows: 1 to 1000 / Page No: 1 of 15.
- Bottom Status:** 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 pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

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

Data Output:

	rental_id	first_name	last_name
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	Danny	Isom

Total rows: 16044 Query complete 00:00:00.102 CRLF Ln 4, Col 1