

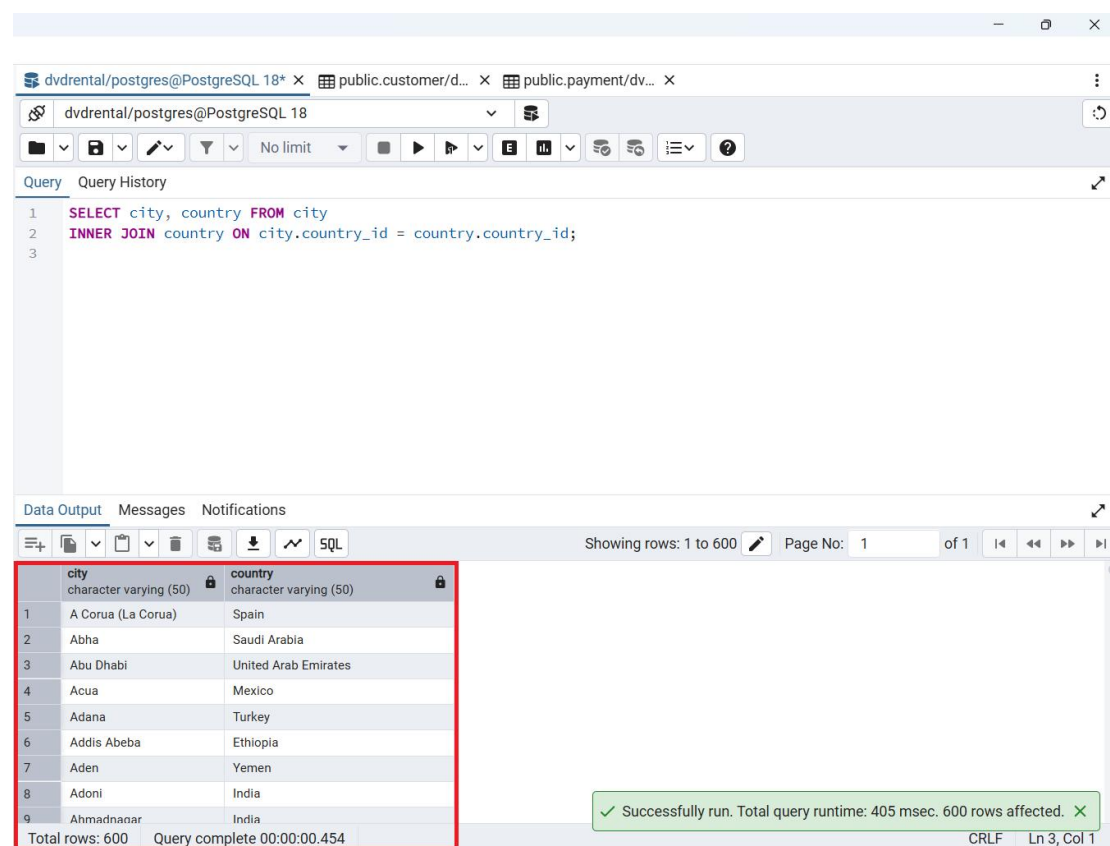
## Assignment 9: DVD Rental Queries

This assignment focuses on practicing **INNER JOIN operations** using the DVD Rental database. The goal is to understand how related tables can be connected through key fields to retrieve combined information.

These exercises strengthen your understanding of **relational data**, **foreign keys**, and how JOINS allow you to merge data from multiple tables in a meaningful way.

### Query 1

**Description:** Retrieve the city and country names together by joining the city and country tables. This query shows how location data is connected through the country\_id field.



The screenshot shows a PostgreSQL query editor interface. The query editor displays the following SQL query:

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

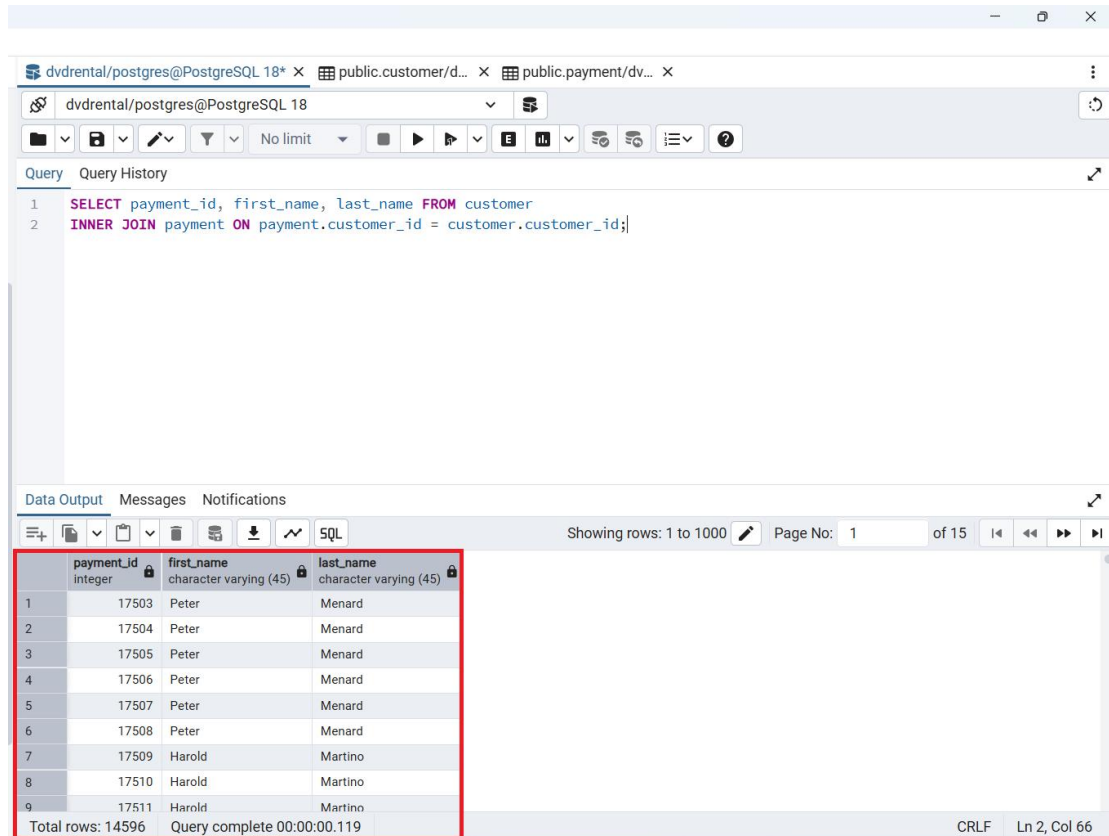
The query results are displayed in a table with two columns: city and country. The table shows 600 rows of data. The first few rows are highlighted in a red box:

city	country
A Corua (La Corua)	Spain
Abha	Saudi Arabia
Abu Dhabi	United Arab Emirates
Acua	Mexico
Adana	Turkey
Addis Abeba	Ethiopia
Aden	Yemen
Adoni	India
Ahmadnagar	India

The status bar at the bottom indicates: Total rows: 600, Query complete 00:00:00.454. A green message box at the bottom right states: Successfully run. Total query runtime: 405 msec. 600 rows affected.

## Query 2

**Description:** Display each payment's ID along with the customer's first and last name. This query demonstrates how payment records are linked to the customers who made them.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

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

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 9 rows highlighted by a red box. The total number of rows is 14596.

	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

Total rows: 14596 Query complete 00:00:00.119

### Query 3

**Description:** List each rental's ID together with the customer's first and last name.

The screenshot shows a PostgreSQL query editor interface. The top bar displays the connection 'dvdrental/postgres@PostgreSQL 18\*' and open tabs for 'public.customer/d...' and 'public.rental/dvdr...'. The query editor contains the following SQL code:

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

Below the query editor, the 'Data Output' tab is active, 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 contains 10 rows of data, which are highlighted with a red border. The status bar at the bottom indicates 'Total rows: 16044' and 'Query complete 00:00:00.138'.

	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

Total rows: 16044 Query complete 00:00:00.138 CRLF Ln 3, Col 1