

КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-
ПЕТЕРБУРГА

Санкт-Петербургское государственное
бюджетное профессиональное образовательное учреждение
«КОЛЛЕДЖ ЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

«МДК.07.01 Управление и автоматизация баз данных»

ОТЧЁТ

по лабораторной работе №4

«Организация и обработка данных»

Работу выполнил студент

325гр.:

Шлычков И. Д.

Преподаватель: Фомин А.В.

Санкт-Петербург 2025

Заходим в нашу БД и переходим в Views чтобы сделать новые образы.
Рисунок 1

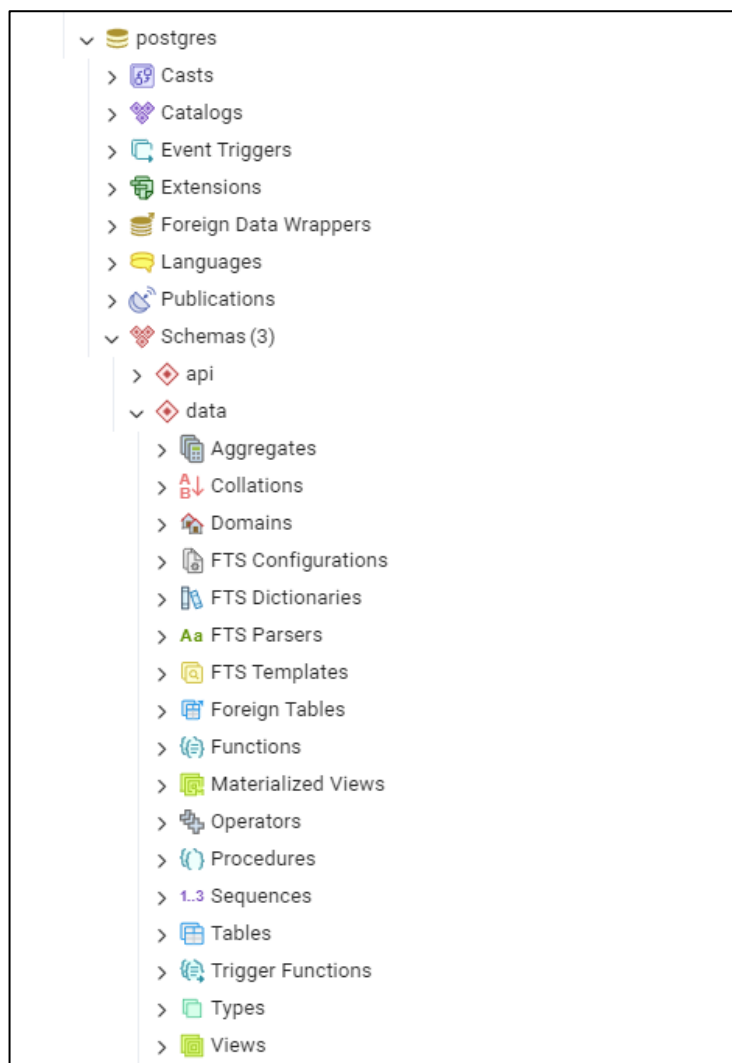
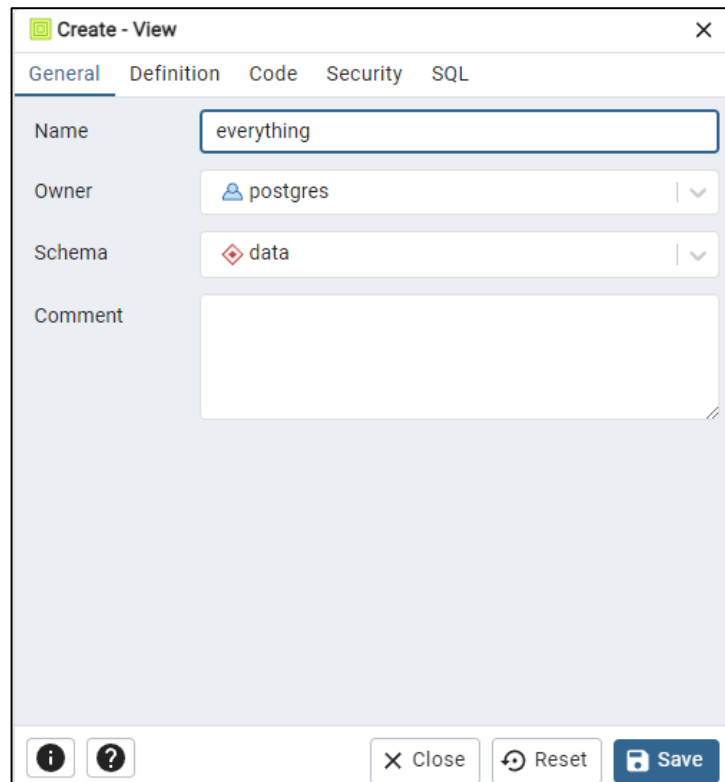


Рисунок 1 – база данных

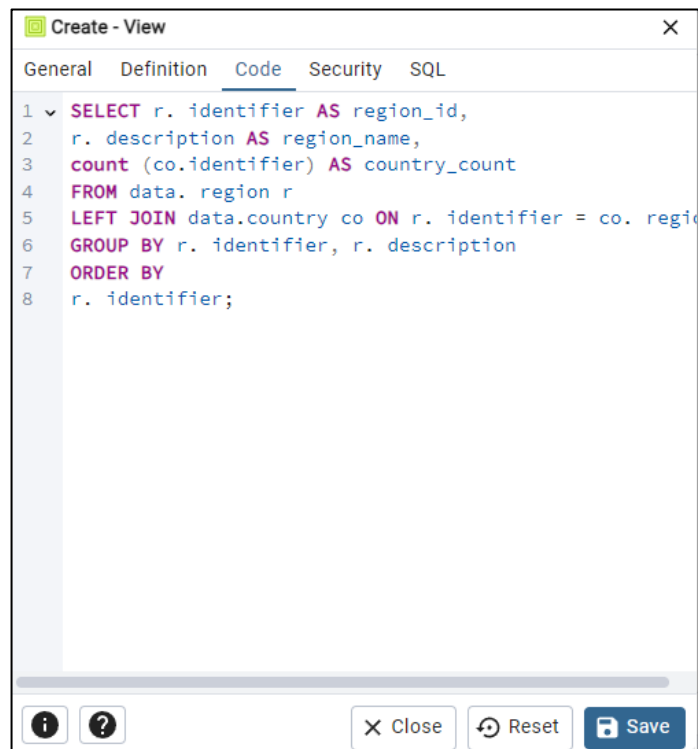
Было создано представление everything, которое объединяет данные из нескольких таблиц.

Это представление позволяет получить полную информацию из связанных таблиц в одном запросе. Рисунок 2-3



The screenshot shows the 'Create - View' dialog box with the 'General' tab selected. The 'Name' field contains 'everything'. The 'Owner' dropdown is set to 'postgres'. The 'Schema' dropdown is set to 'data'. The 'Comment' field is empty. At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

Рисунок 2 – создание представления



The screenshot shows the 'Create - View' dialog box with the 'Code' tab selected. The SQL code is as follows:

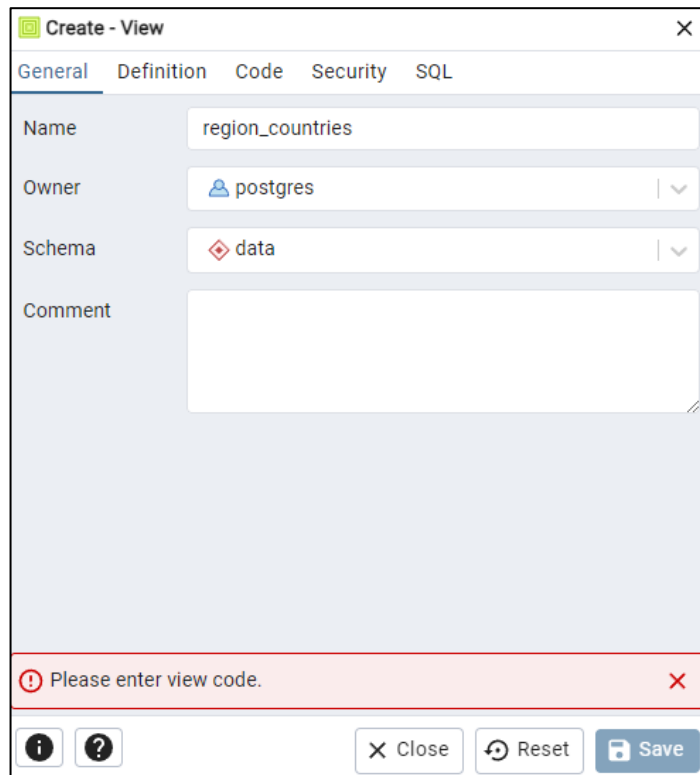
```
1 SELECT r. identifier AS region_id,  
2 r. description AS region_name,  
3 count (co.identifier) AS country_count  
4 FROM data. region r  
5 LEFT JOIN data.country co ON r. identifier = co. region_id  
6 GROUP BY r. identifier, r. description  
7 ORDER BY  
8 r. identifier;
```

At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

Рисунок 3 – код для представления

Создание представления region_countries

Представление region_countries было создано для отображения списка стран по регионам. Данное представление упрощает анализ распределения стран по регионам. Рисунок 4-5

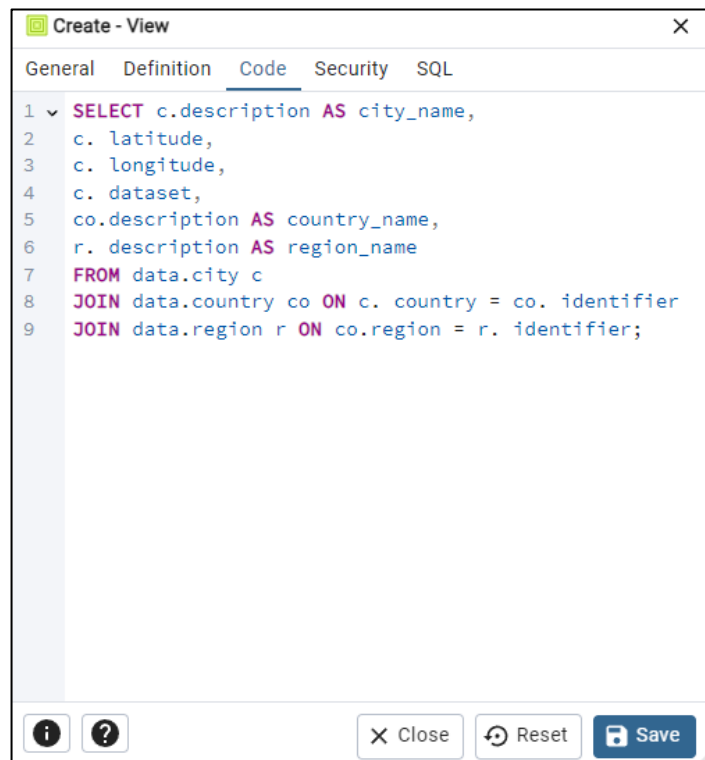


The screenshot shows the 'Create - View' dialog box with the following fields:

- Name: region_countries
- Owner: postgres
- Schema: data
- Comment: (empty text area)

A red error message at the bottom states: "Please enter view code." The dialog has tabs for General, Definition, Code, Security, and SQL. At the bottom are buttons for Close, Reset, and Save.

Рисунок 4 – создание представления



The screenshot shows the 'Create - View' dialog box with the 'Code' tab selected. The SQL code is as follows:

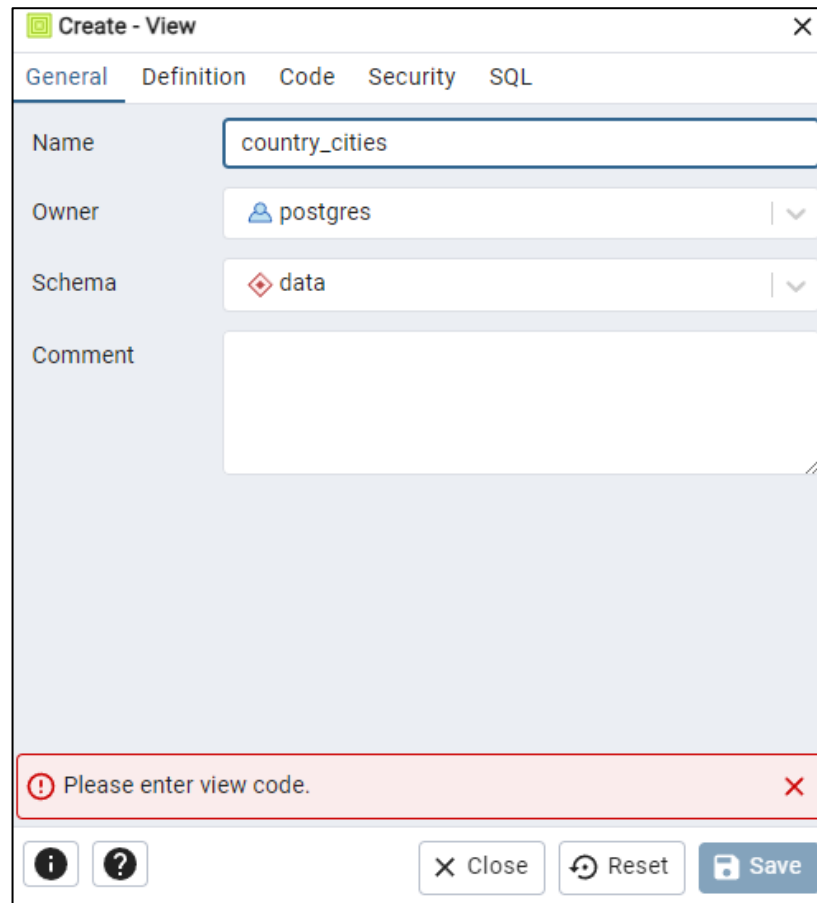
```
1 SELECT c.description AS city_name,  
2 c. latitude,  
3 c. longitude,  
4 c. dataset,  
5 co.description AS country_name,  
6 r. description AS region_name  
7 FROM data.city c  
8 JOIN data.country co ON c. country = co. identifier  
9 JOIN data.region r ON co.region = r. identifier;
```

The dialog has tabs for General, Definition, Code, Security, and SQL. At the bottom are buttons for Close, Reset, and Save.

Рисунок 5 – код для представления

Создание представления country_cities

Представление country_cities предоставляет список городов для каждой страны. Это представление удобно для быстрого доступа к информации о городах и их принадлежности к странам. Рисунок 6-7



The screenshot shows the 'Create - View' dialog box with the following details:

- Name:** country_cities
- Owner:** postgres
- Schema:** data
- Comment:** (empty text area)
- Error Message:** Please enter view code.
- Buttons:** Close, Reset, Save

Рисунок 6 – создание представления



The screenshot shows the 'Code' tab of the 'Create - View' dialog box with the following SQL code:

```
1 SELECT co.identifier AS country_id, co. description AS country_name,  
2 count (c. identifier) AS city_count  
3 FROM data.country co  
4 LEFT JOIN data.city c ON co. identifier = c. country  
5 GROUP BY co.identifier, co.description  
6 ORDER BY co.identifier;
```

Рисунок 7 – код для представления

Проверка созданных представлений

После создания всех представлений была выполнена проверка их работоспособности.

Результаты запросов к представлениям подтвердили корректность их работы.

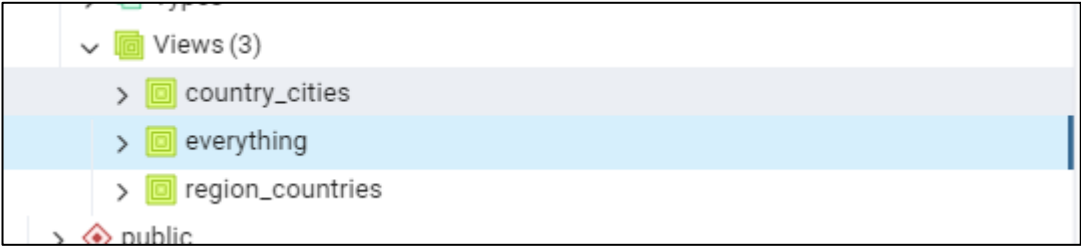


Рисунок 8 – убеждаемся, что всё сработало

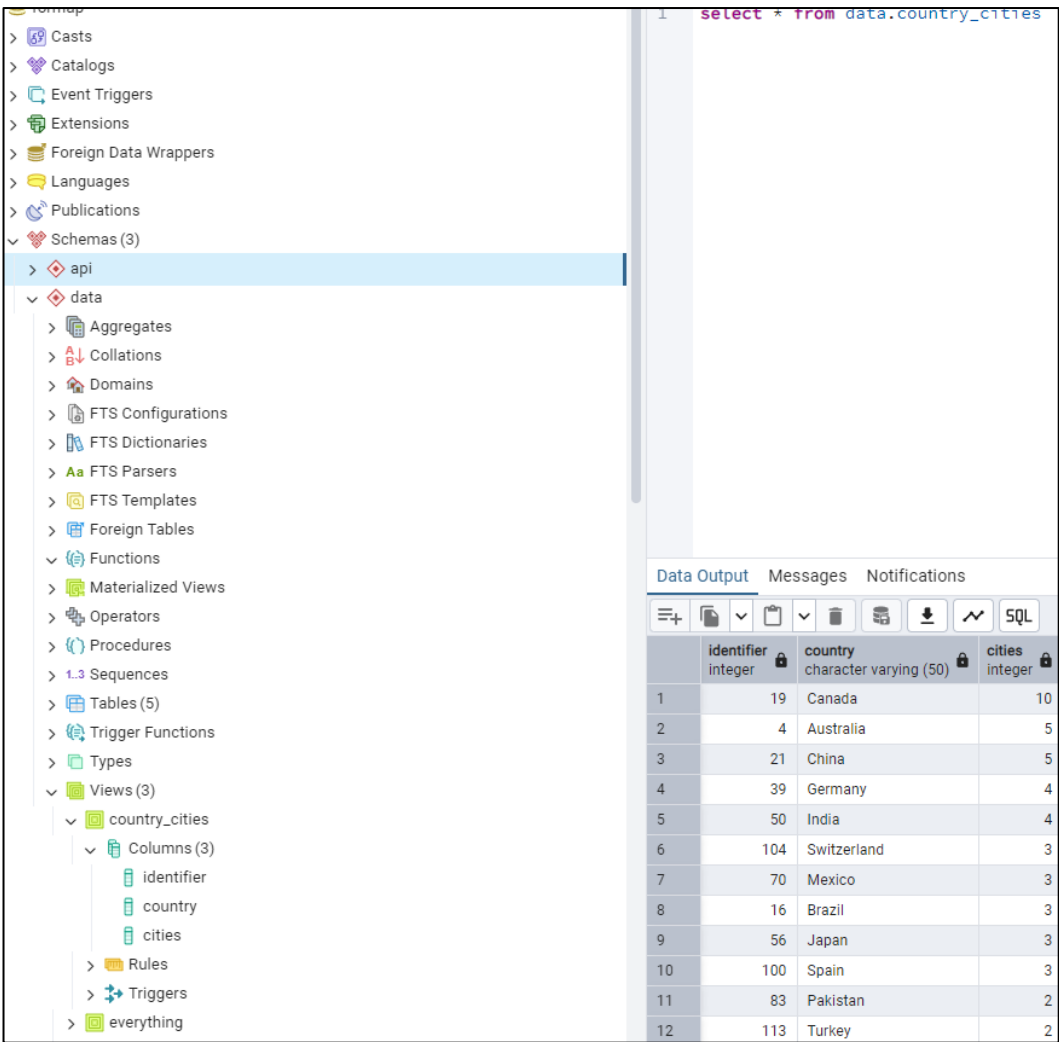


Рисунок 9 – выводим данные country_cities

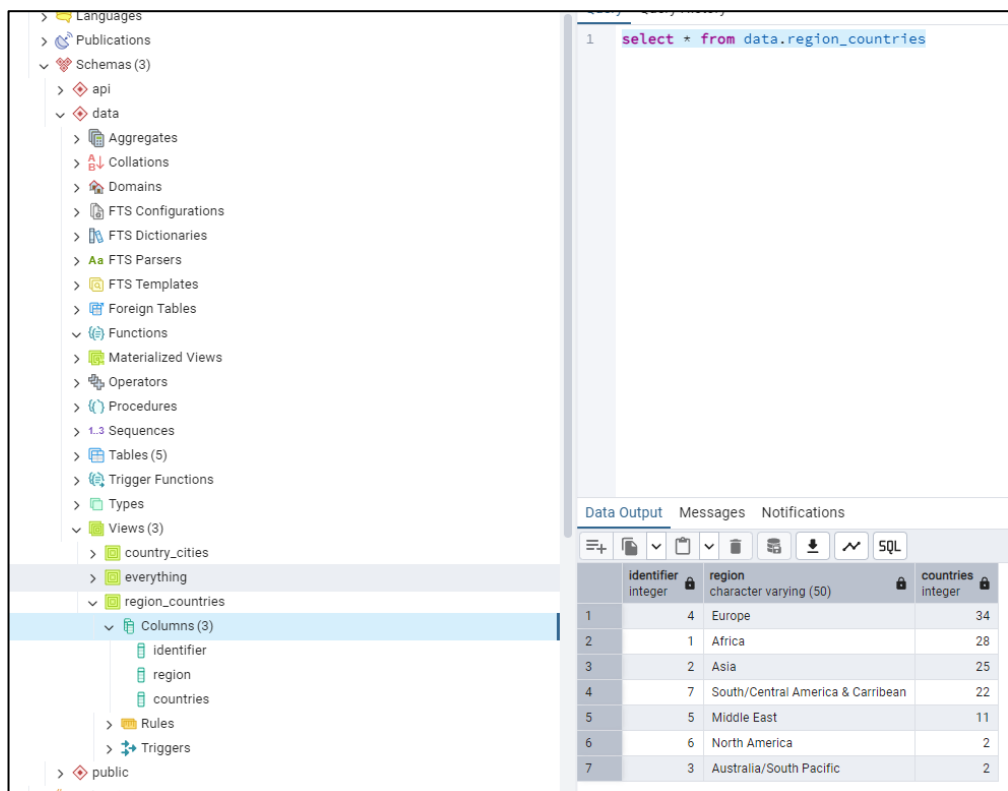


Рисунок 10 – выводим данные region_countries

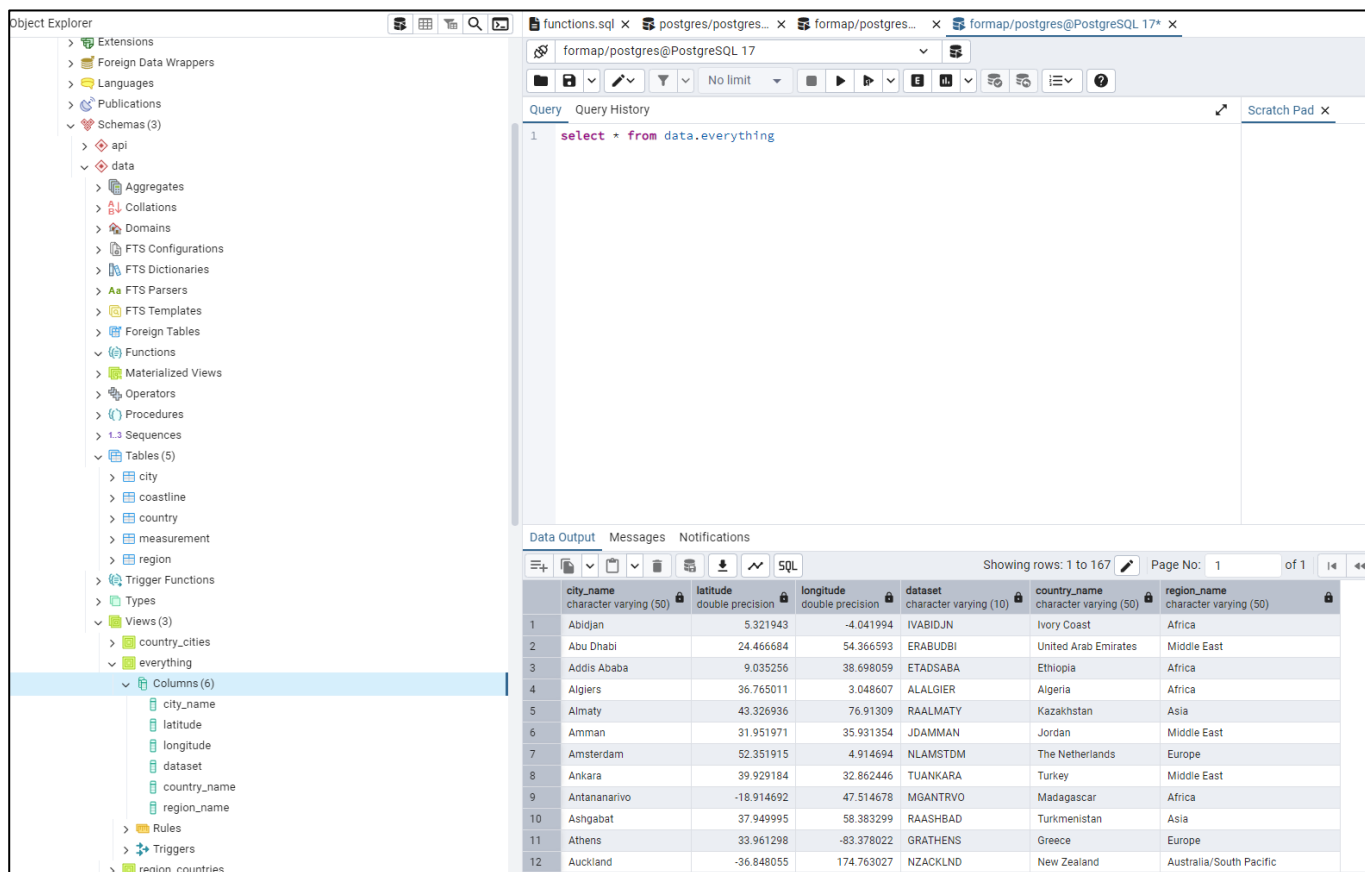


Рисунок 11 – выводим данные everything

Вывод

В ходе лабораторной работы были успешно созданы и протестированы представления `everything`, `region_countries` и `country_cities`. Эти представления позволяют упростить доступ к данным, сократить сложность запросов и улучшить структуру базы данных. Работа показала важность использования представлений для организации и обработки информации в базах данных.


```
SELECT r. identifier AS region_id,  
r. description AS region_name,  
count (co.identifier) AS country_count  
FROM data. region r  
LEFT JOIN data.country co ON r. identifier = co. region  
GROUP BY r. identifier, r. description  
ORDER BY  
r. identifier;
```

```
SELECT c.description AS city_name,  
c. latitude,  
c. longitude,  
c. dataset,  
co.description AS country_name,  
r. description AS region_name  
FROM data.city c  
JOIN data.country co ON c. country = co. identifier  
JOIN data.region r ON co.region = r. identifier;
```

```
SELECT co. identifier AS country_id, co. description AS  
country_name, count (c. identifier) AS city_count  
FROM data.country co  
LEFT JOIN data.city c ON co. identifier = c. country  
GROUP BY co.identifier, co.description  
ORDER BY co. identifier;
```