Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики управления и технологий

Мареев Георгий Александрович БД-241м

**Лабораторная работа 2. Моделирование данных и SQL для Data Engineering**

**Вариант 13**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

кандидат технических наук

Москва

2025

**Введение**

*Цель*: освоить принципы проектирования баз данных, создания структуры таблиц и загрузки данных в PostgreSQL.

**Основная часть**

Просмотр всех запущенные контейнеры:

docker ps

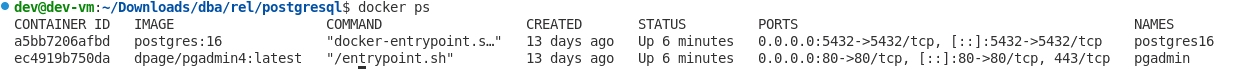


Рисунок 1 Все запущенные контейнеры

Перезапуск контейнеров:

dev@dev-vm:~/Downloads/dba/rel/postgresql$ sudo docker compose stop

[sudo] password for dev:

[+] Stopping 2/2

✔ Container pgadmin Stopped 1.5s

✔ Container postgres16 Stopped 0.2s

dev@dev-vm:~/Downloads/dba/rel/postgresql$ sudo docker compose start

[+] Running 2/2

✔ Container postgres16 Started 0.3s

✔ Container pgadmin Started 0.3s

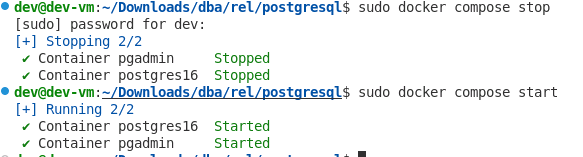
****

Рисунок 2 Перезапуск контейнеров

Проверка ip-адресса контейнера postgres

sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgres16

172.18.0.3



Рисунок 3 IP-aдресс контейнера postgres

Запуск pgadmin в браузере на виртуальной машине:

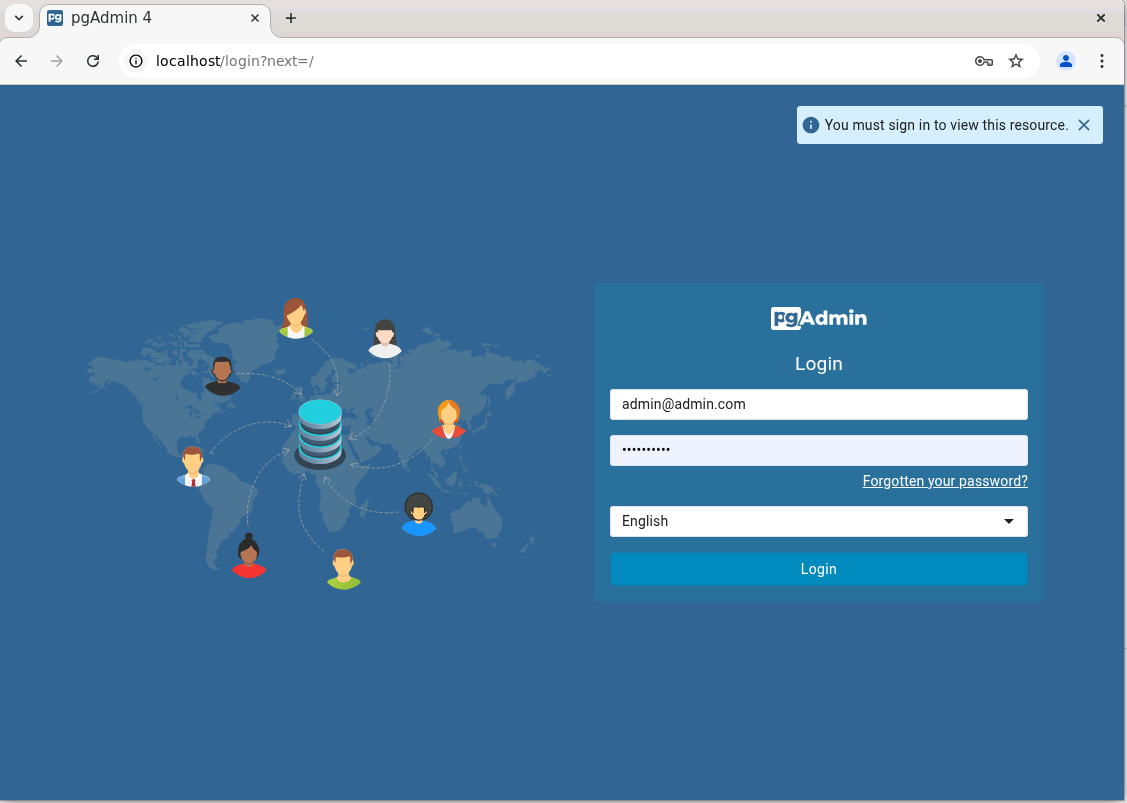


Рисунок 4 pgadmin браузере на виртуальной машине

Состояние MGPU\_superset

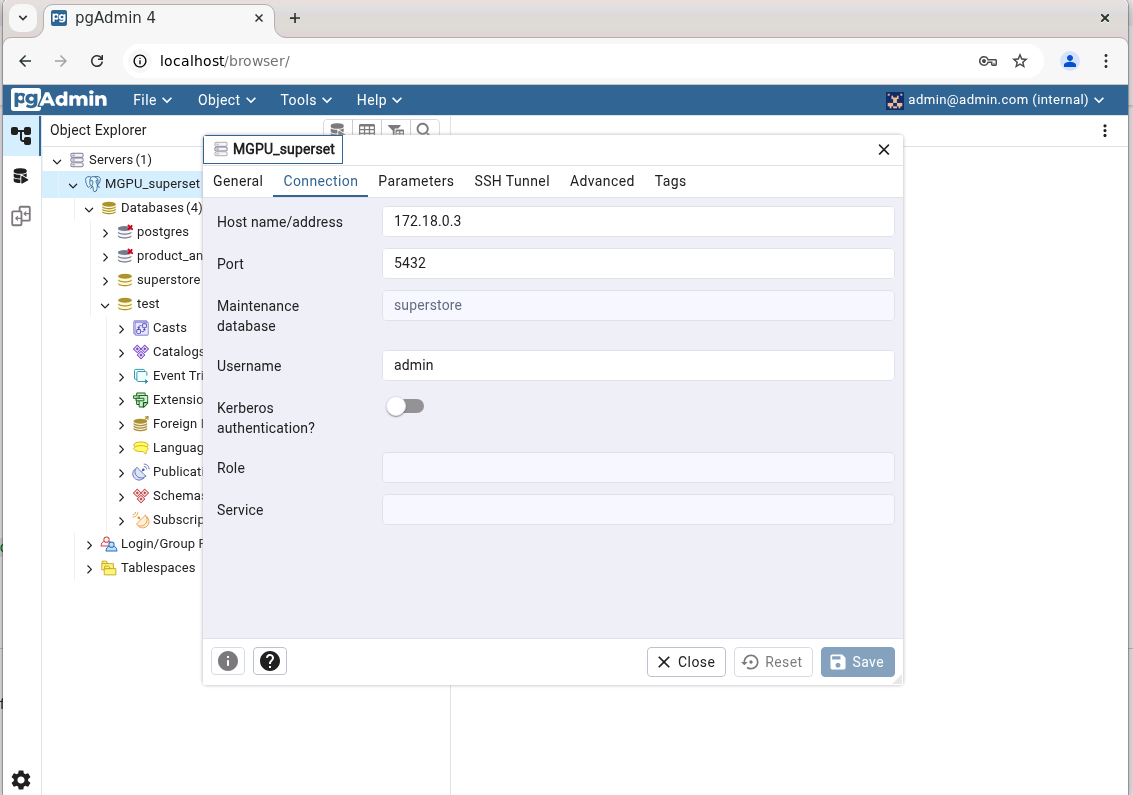


Рисунок 5 Состояние MGPU\_superset

Создание бд lab\_02

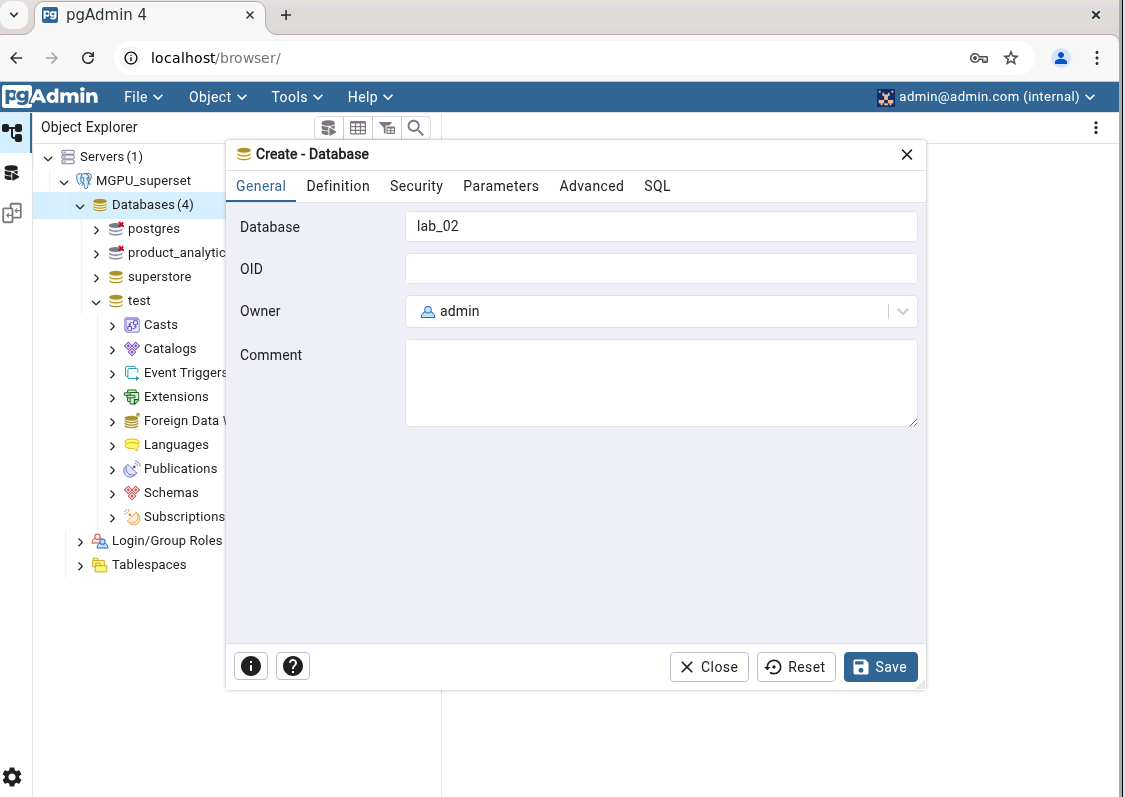
****

Рисунок 6 Создание бд lab\_02

Запуск pgadmin в браузере на локальной машине:



Рисунок 7 pgadmin в браузере на локальной машине

Создание Schema (stg):

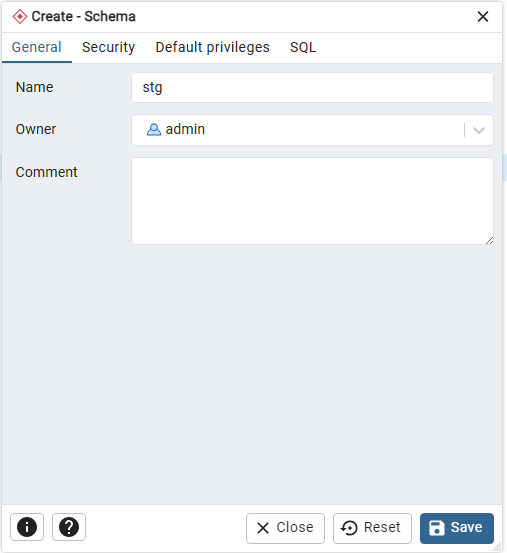


Рисунок 8 Создание stg (staging layer)

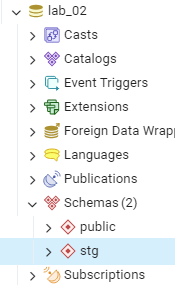


Рисунок 9 stg на общей схеме

Создание Schema (dw):

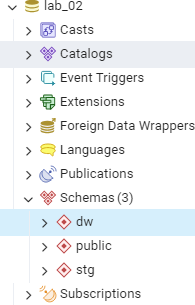


Рисунок 10 Создание dw (data warehouse layer)

Заполнение stg данными из stg.orders.sql

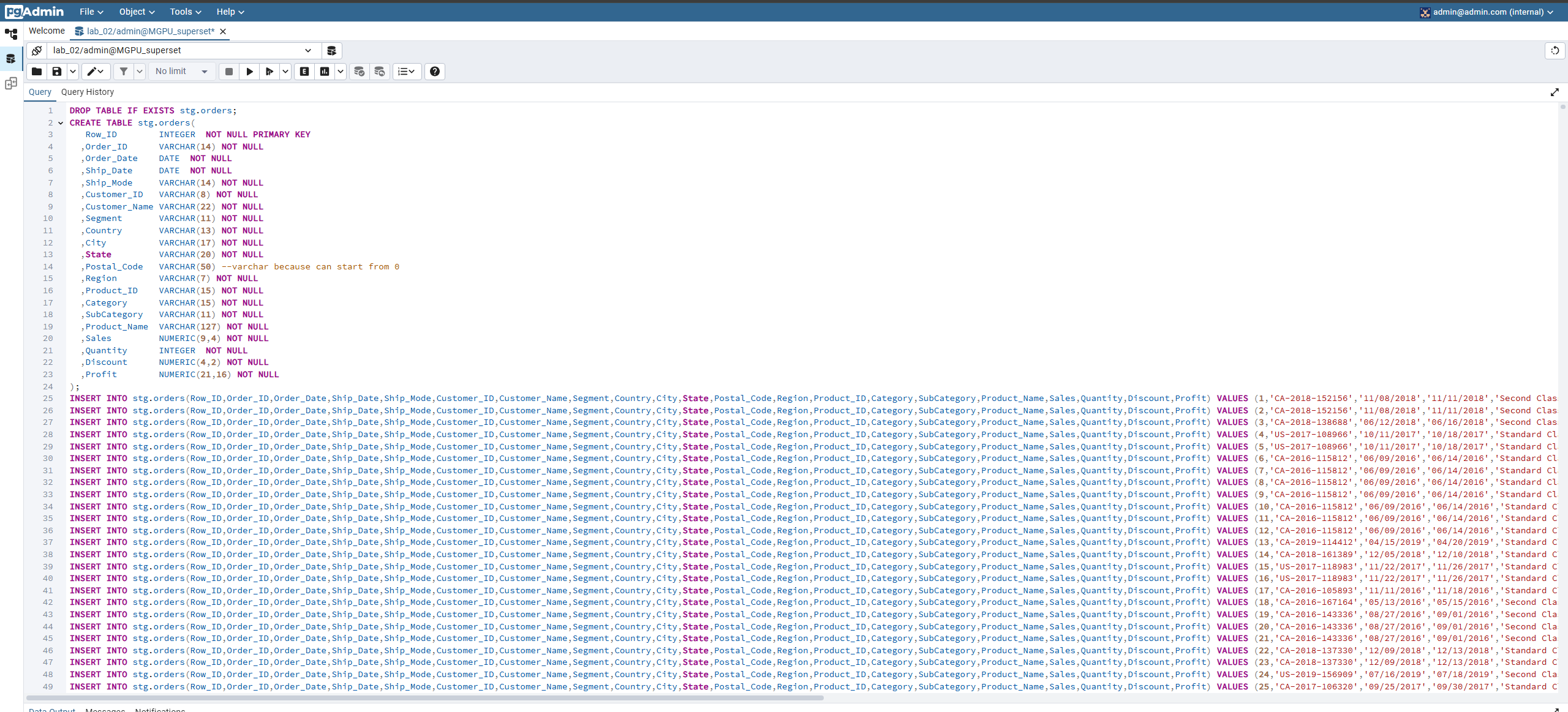


Рисунок 11 Заполнение stg данными

stg-Tables-orders-Columns:

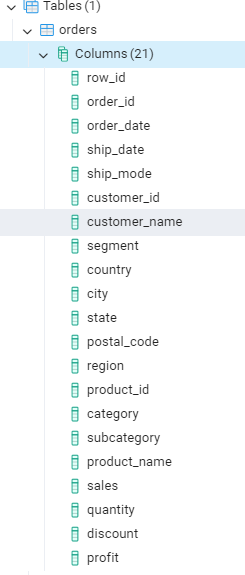


Рисунок 12 Состояние orders

Проверка количества данных:

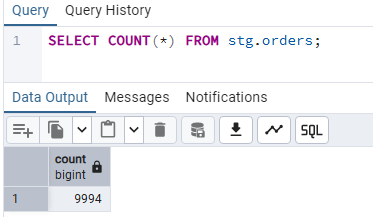


Рисунок 13 Проверка количества данных

**public (raw data layer)**

orders:

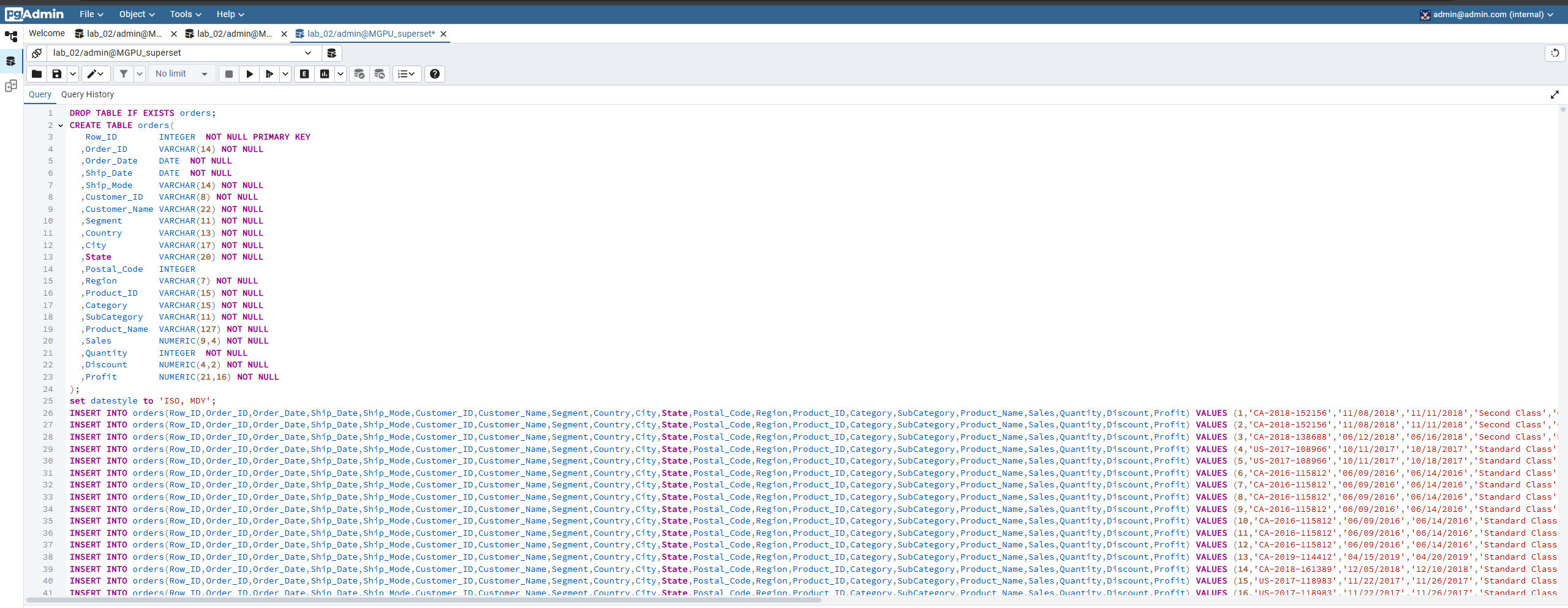


Рисунок 14 Заполнение orders

Проверка данных:

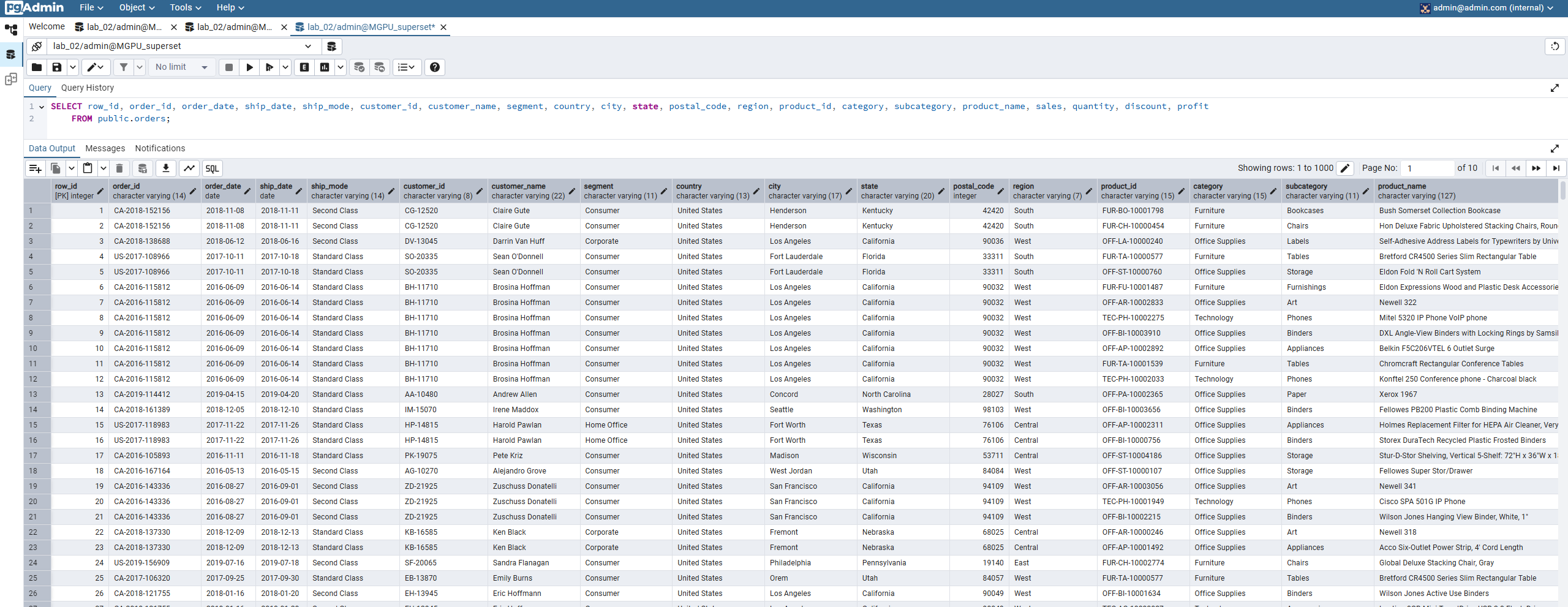
****

Рисунок 15 Проверка данных в orders

Данные дублированы

people:

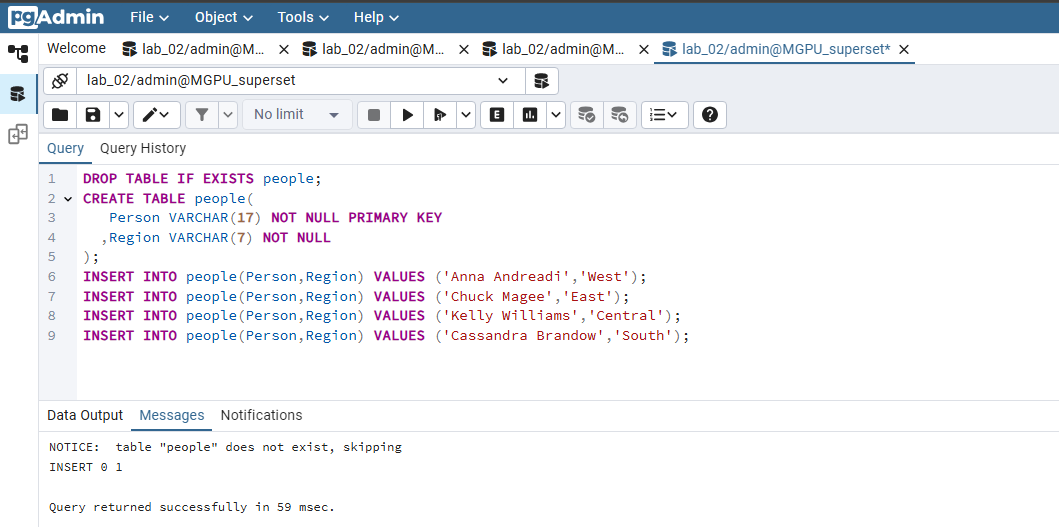


Рисунок 16 Заполнение people

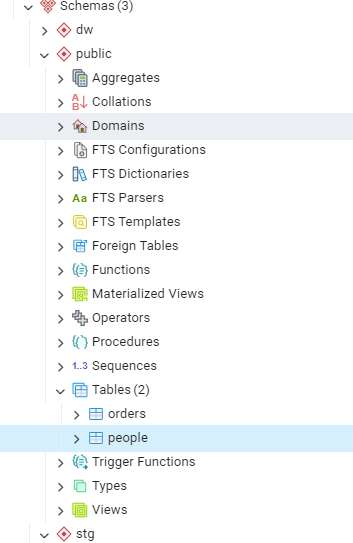


Рисунок 17 Состояние

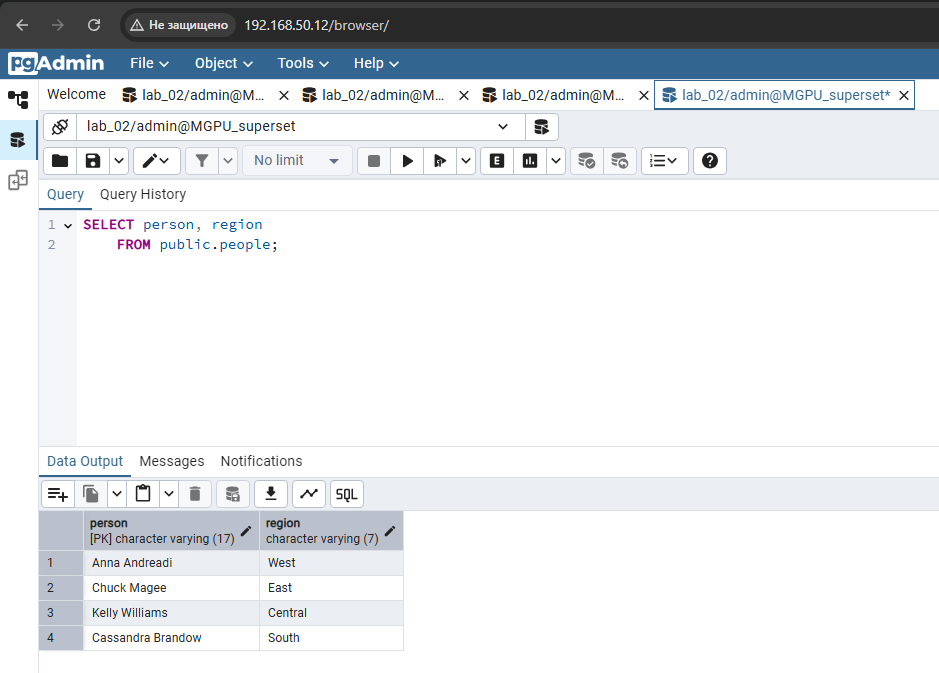


Рисунок 18 Проверка данных people

returns:

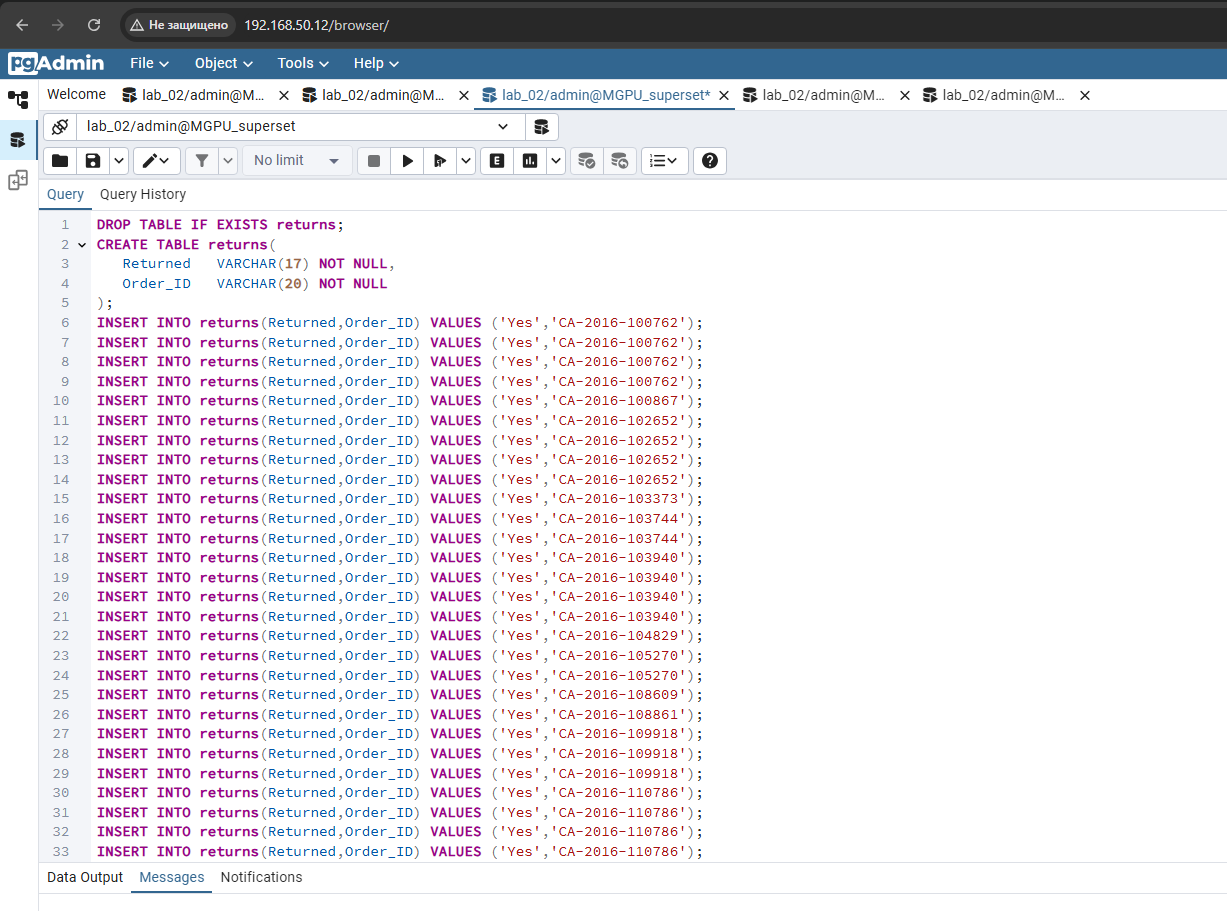


Рисунок 19 Заполнение данных returns

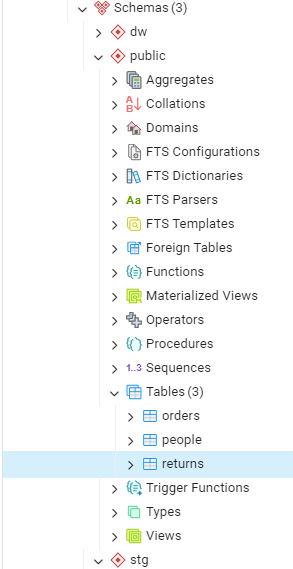


Рисунок 20 Состояние

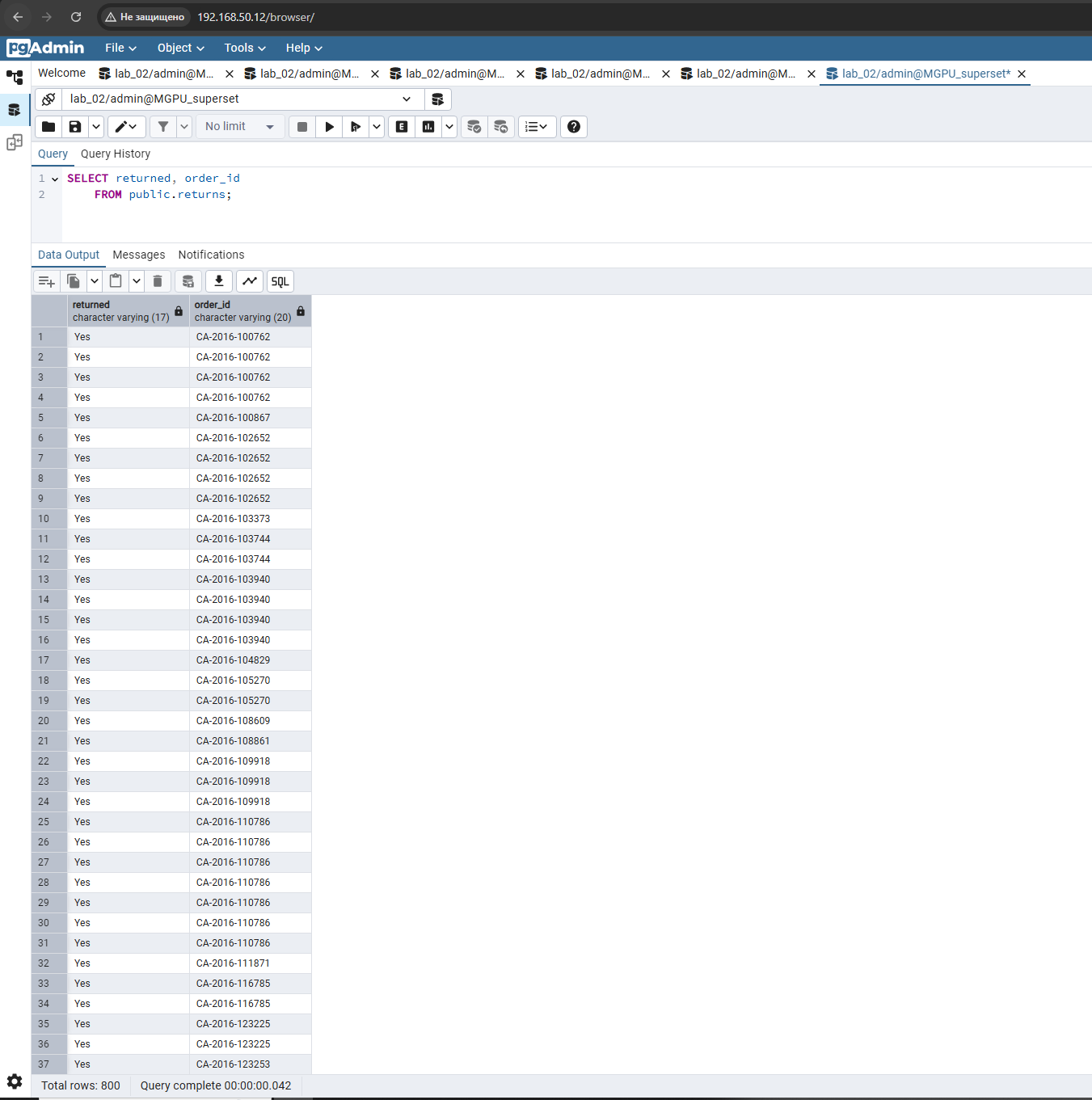


Рисунок 21 Проверка данных returns

**Dw (Data Warehouse layer)**

скрипты из файла from\_stg\_to dw.sql

Создание справочников:

* shipping\_dim

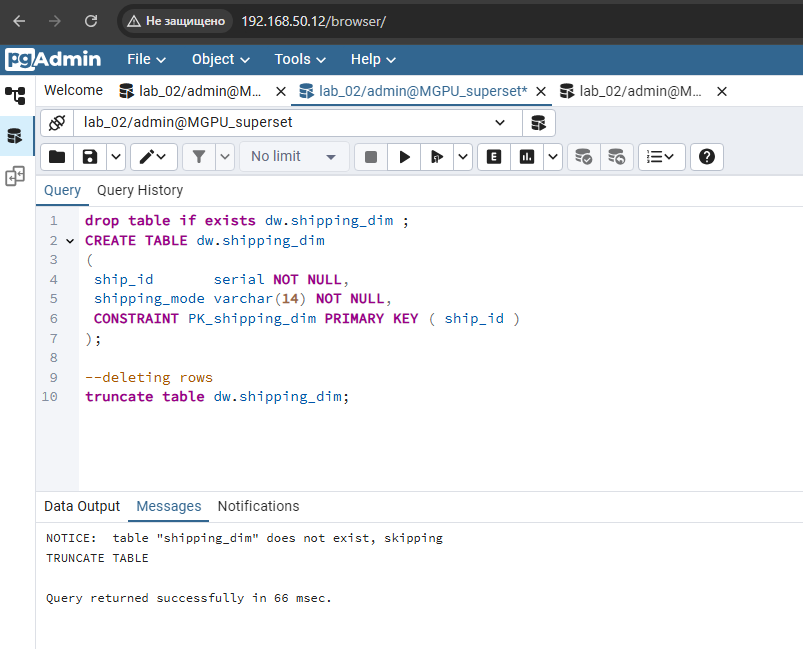


Рисунок 22 Создание справочника shipping\_dim

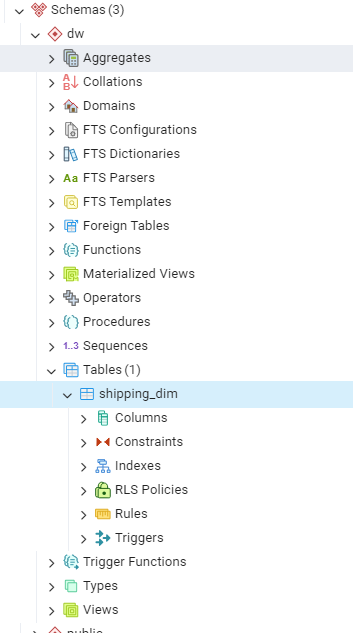


Рисунок 23 Справочник shipping\_dim на схеме

Проверка и вывод

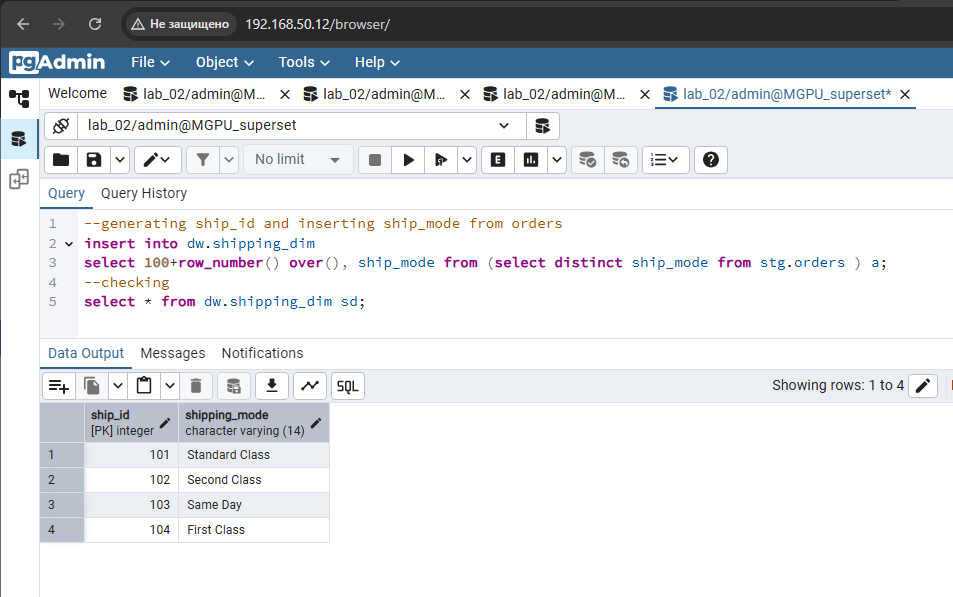


Рисунок 24 Проверка справочника shipping\_dim

customer\_dim

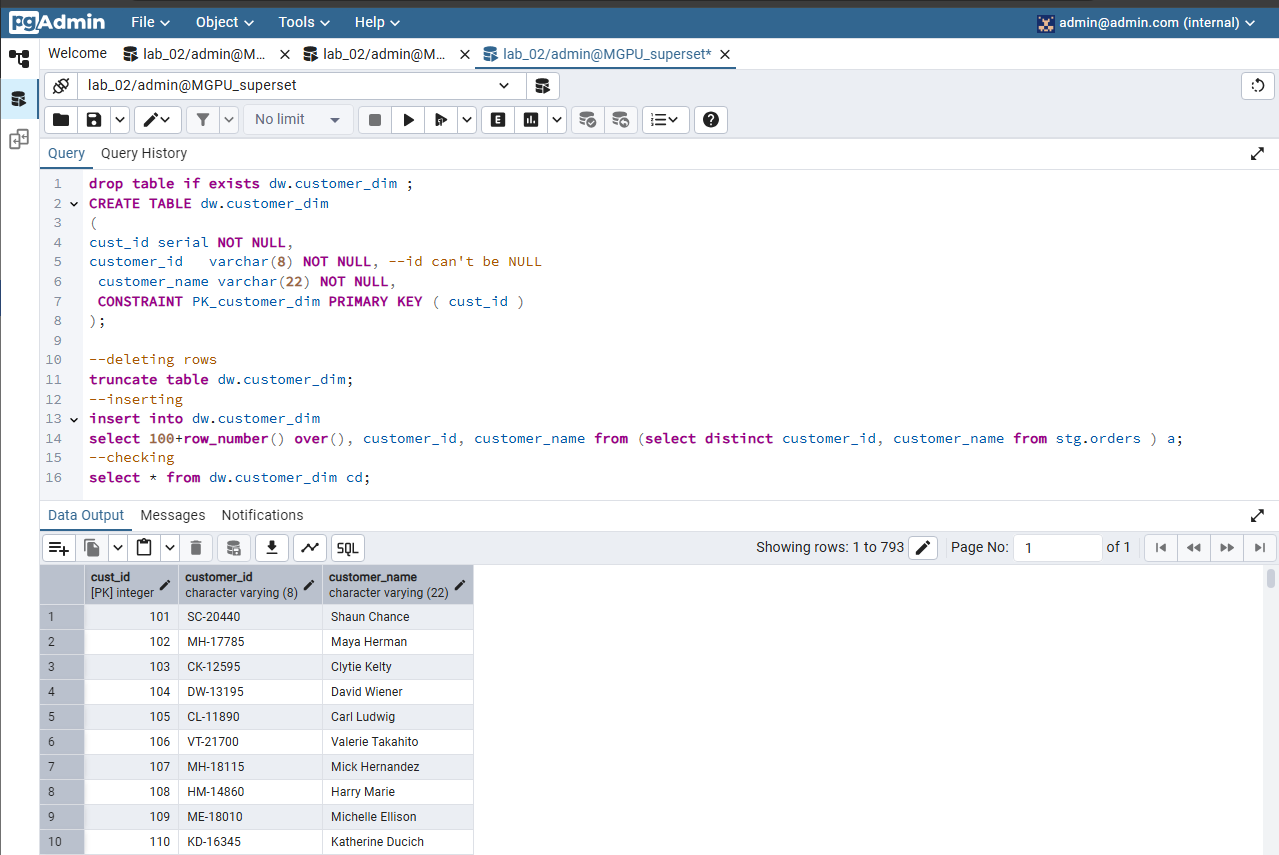


Рисунок 25 Создание и проверка справочника customer\_dim

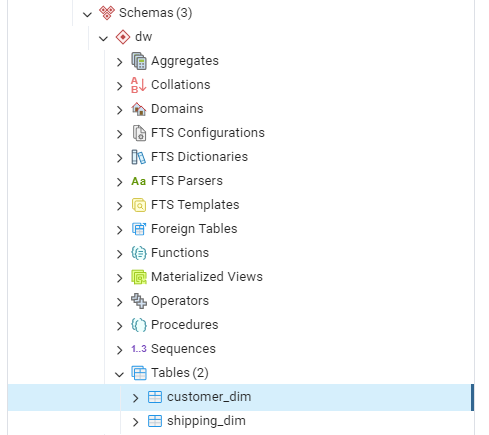


Рисунок 26 Справочник customer\_dim на схеме

geo\_dim

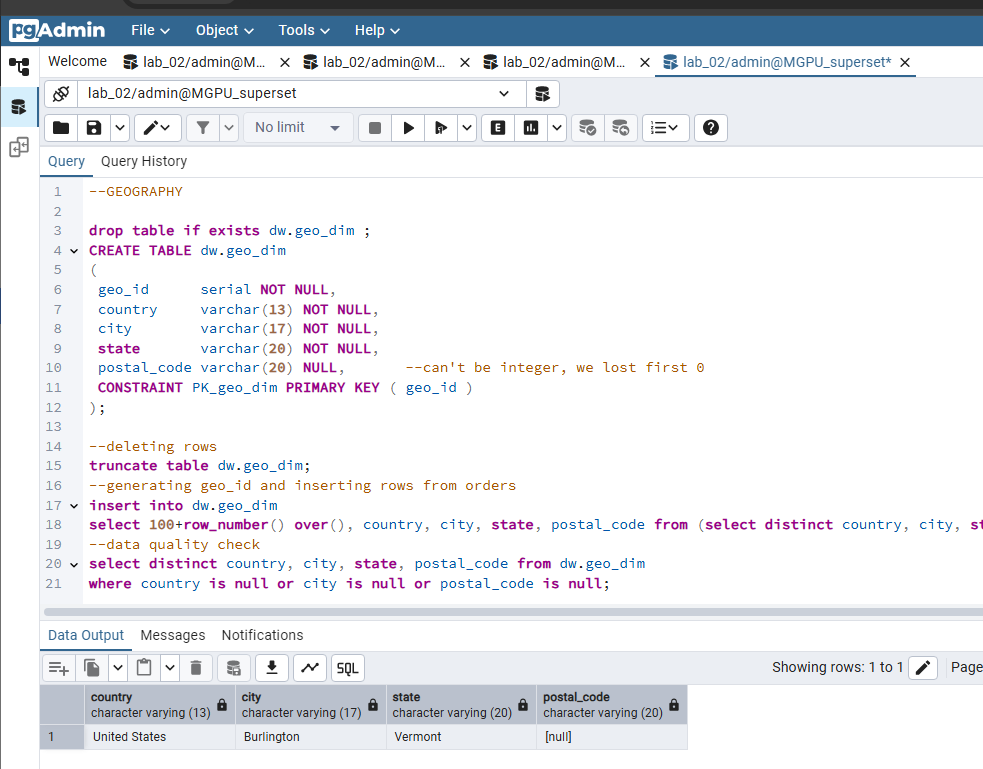


Рисунок 27 Создание и проверка справочника geo\_dim

United States/Vermont/Burlington – нет почтового индекса

Обновление и проверка данных(почтового индекса) для Burlington:

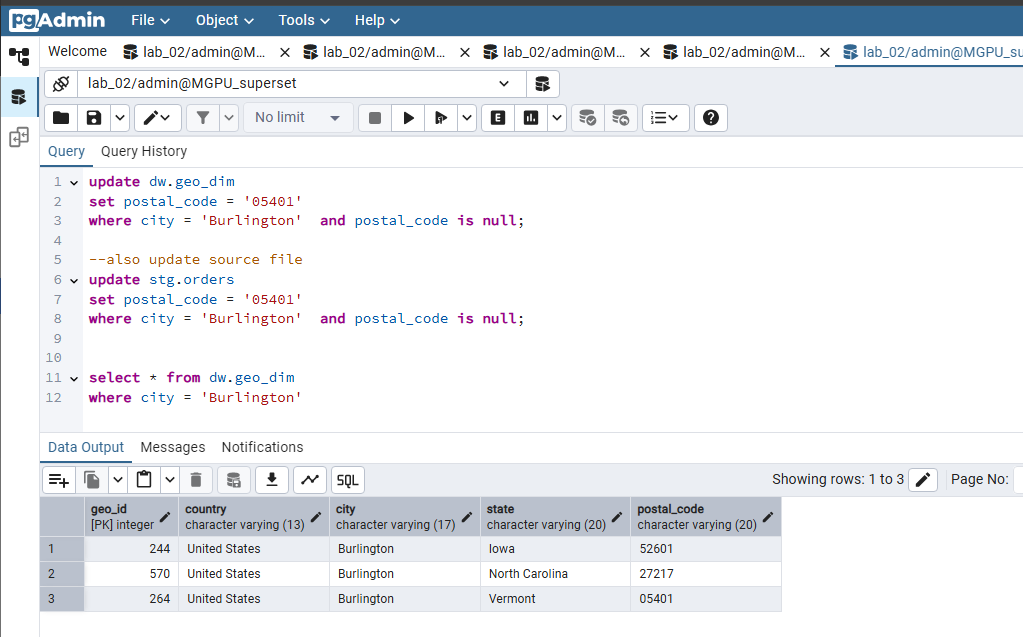


Рисунок 28 Добавление почтового индекса для Burlington

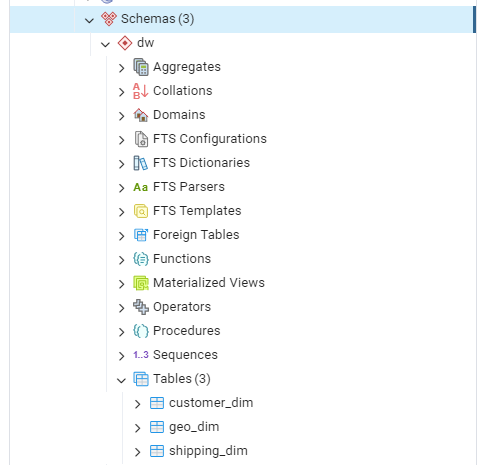


Рисунок 29 geo\_dim на схеме

product\_dim

создание и проверка:

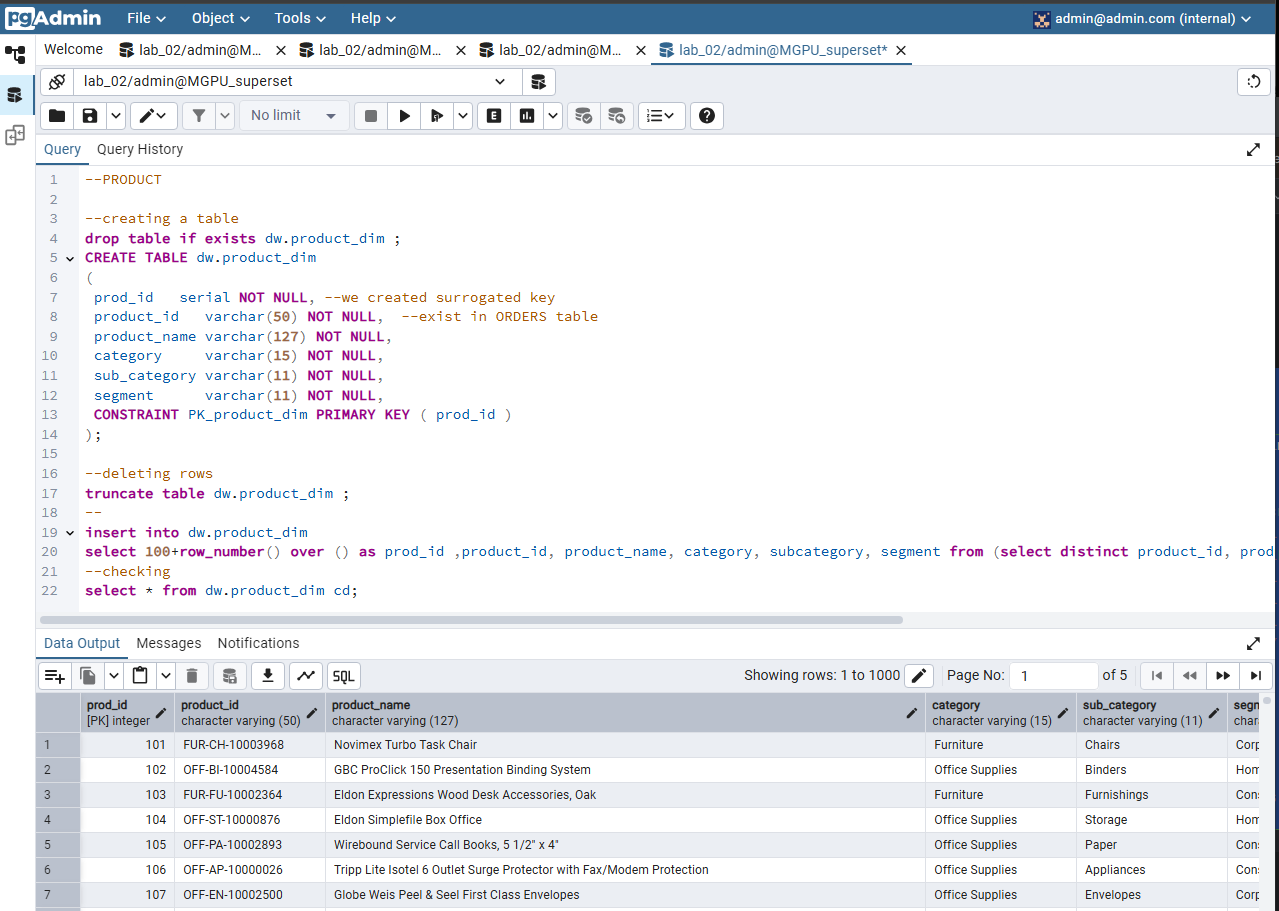


Рисунок 30 Создание и проверка справочника product\_dim

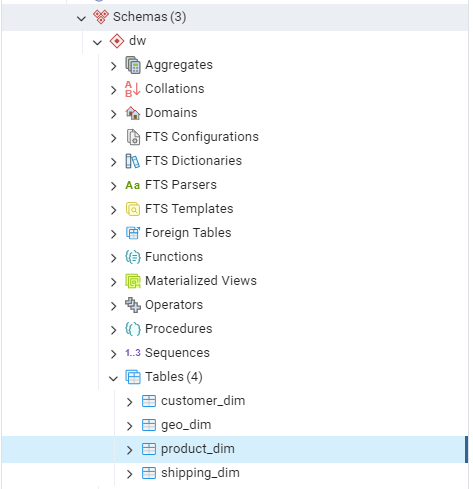


Рисунок 31 product\_dim на схеме

calendar\_dim

создание и проверка:

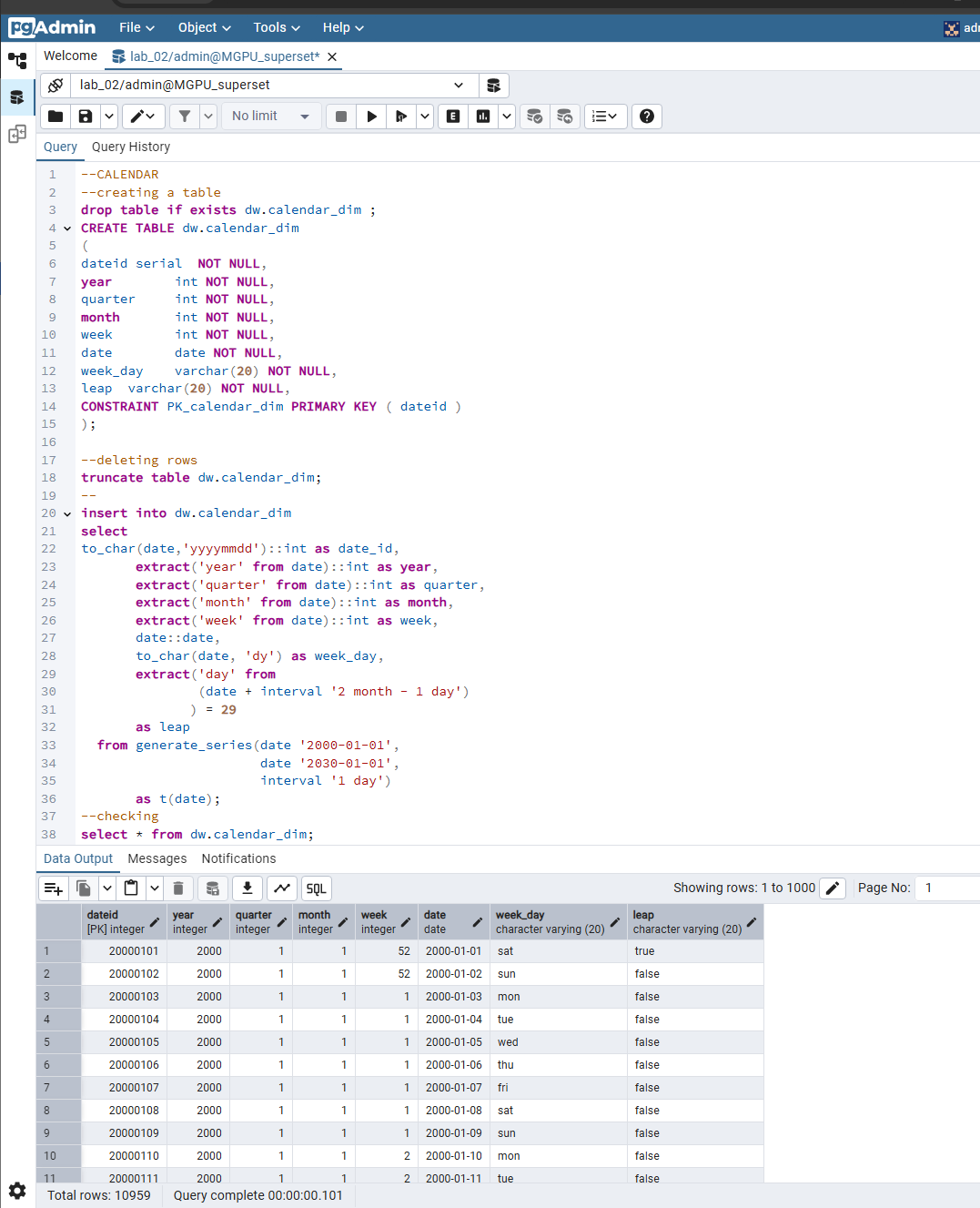


Рисунок 32 Создание и проверка справочника calendar\_dim

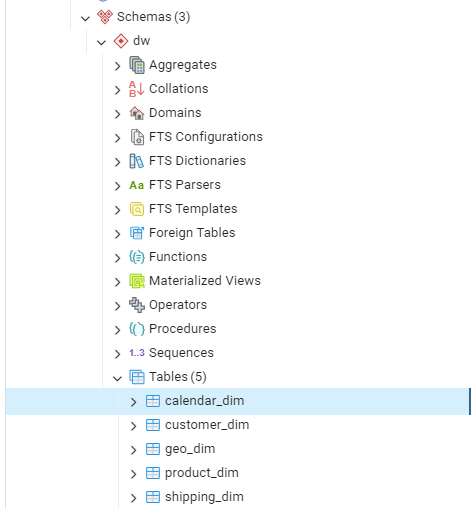


Рисунок 33 calendar\_dim на схеме

Метрики sales\_fact

Создание и проверка

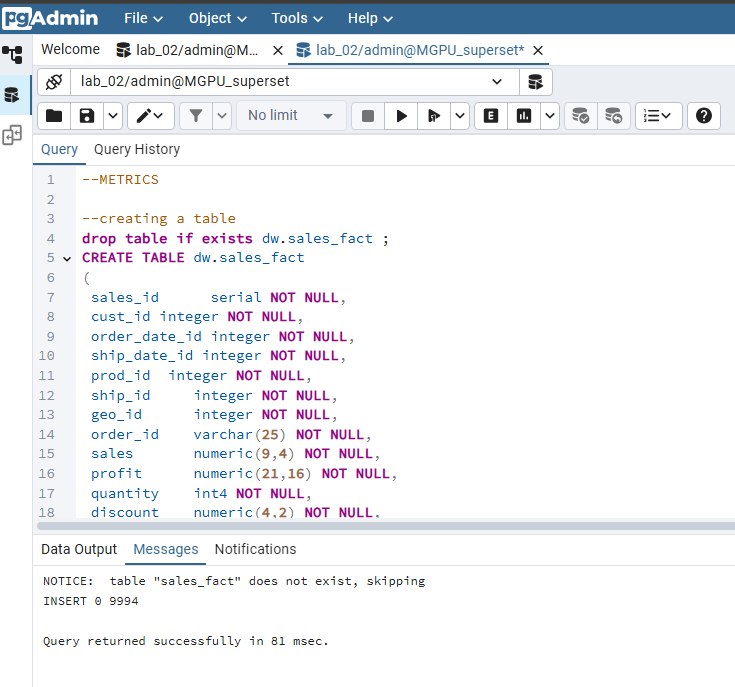


Рисунок 34 Создание метрик (sales\_fact)

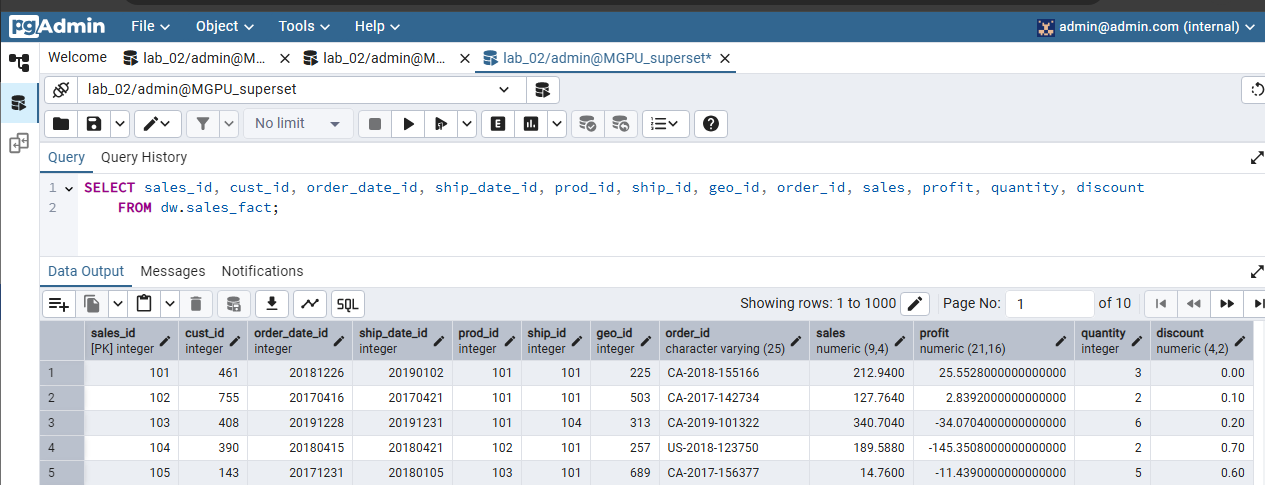


Рисунок 35 Проверка метрик (sales\_fact)

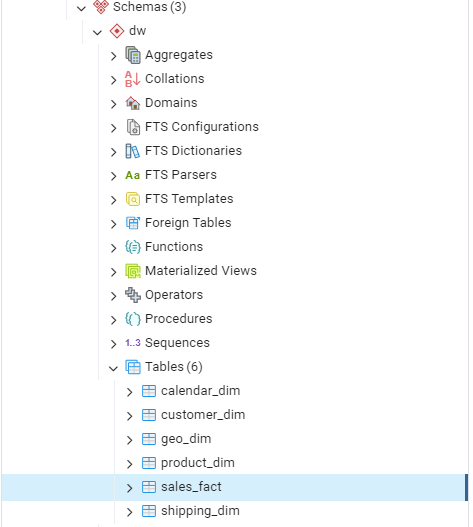


Рисунок 36 sales\_fact на схеме

Проверка качества данных

Проверка количества данных:

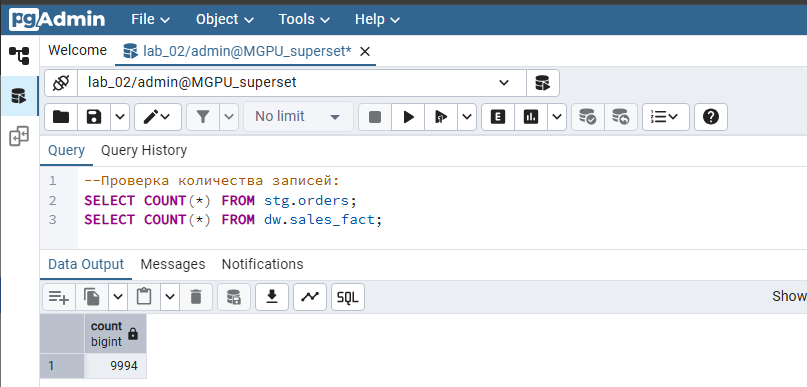


Рисунок 37 Проверка количества данных

Проверка целостности данных:

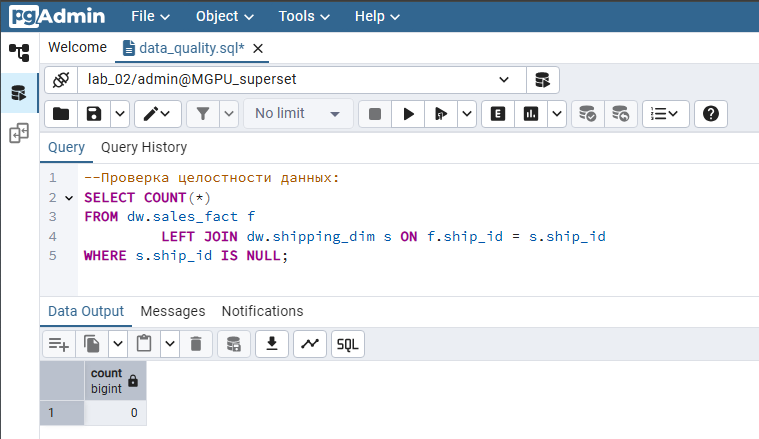


Рисунок 38 Проверка целостности данных

Проверка корректности агрегатов:

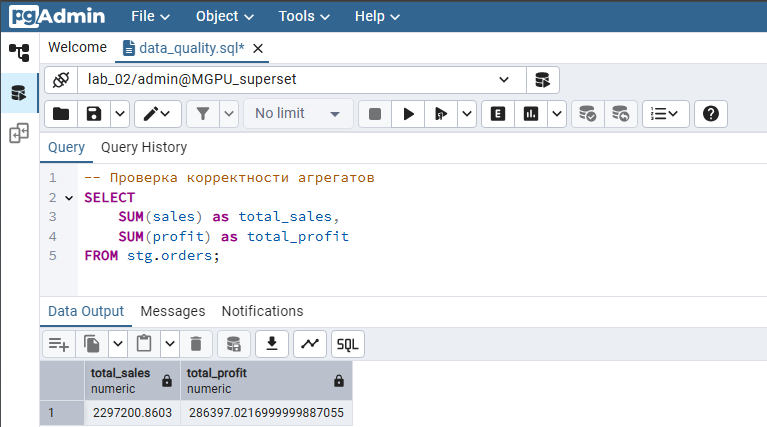


Рисунок 39 Проверка корректности агрегатов



Рисунок 40 Рисунок 39 Проверка корректности агрегатов

**Индивидуальное задание. Вариант 13**

**Задание:**

1. Создать представление по клиентам.

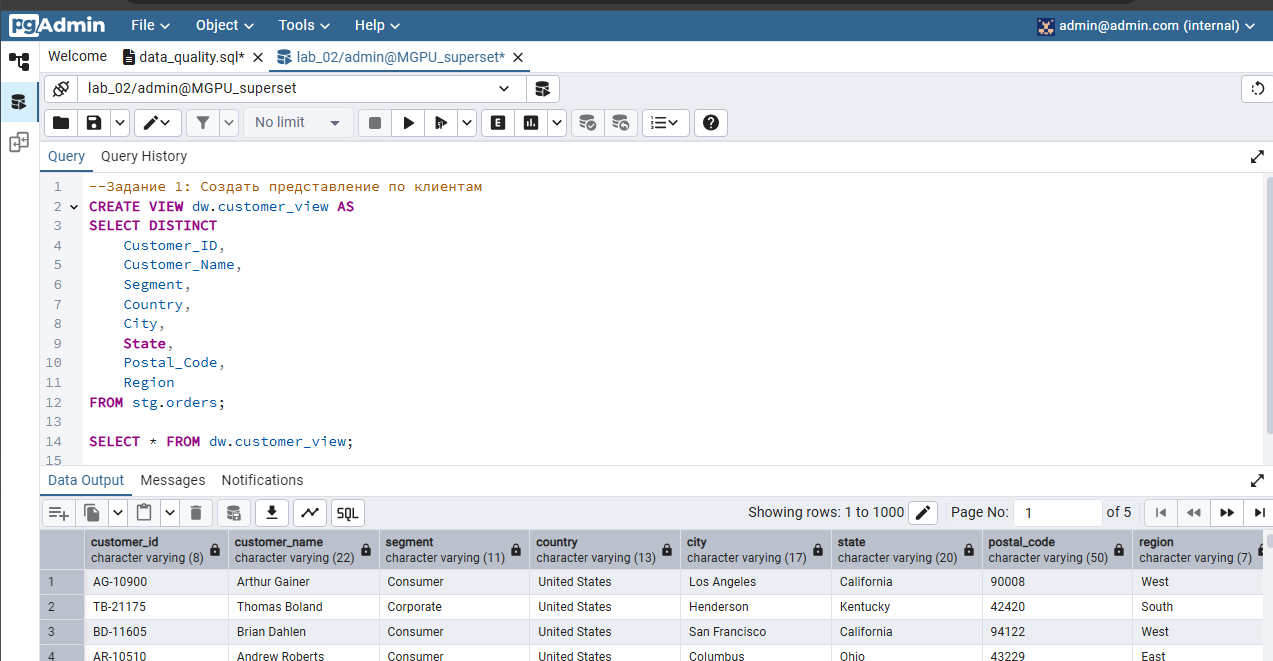


Рисунок 41 Создание представления по клиентам

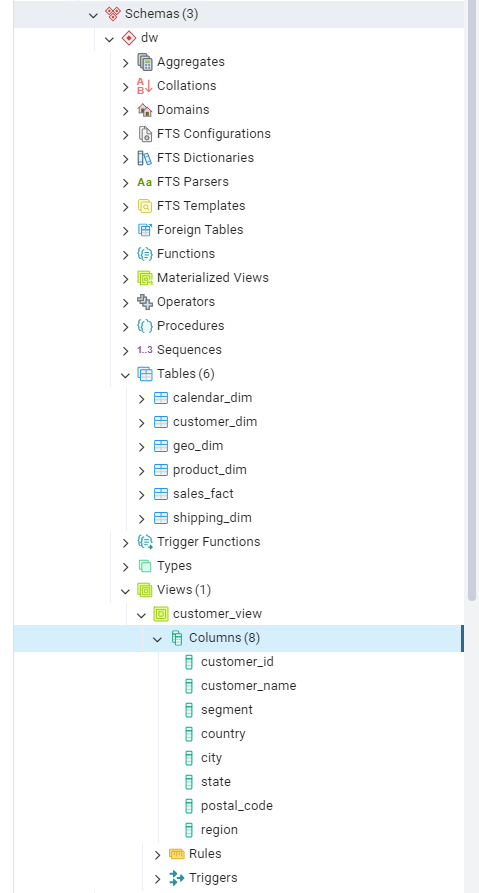


Рисунок 42 View1 на схеме

1. Определить продажи по способам доставки.

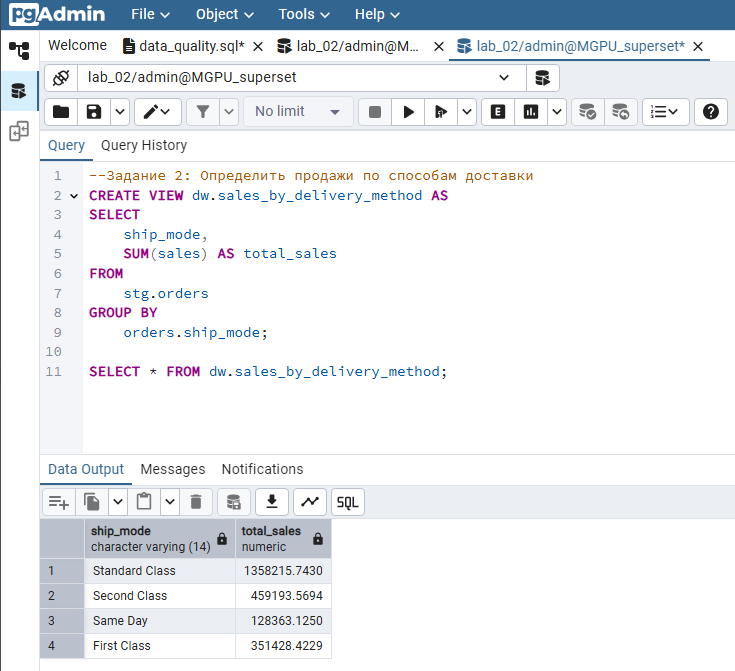


Рисунок 43 Определение продажи по способам доставки

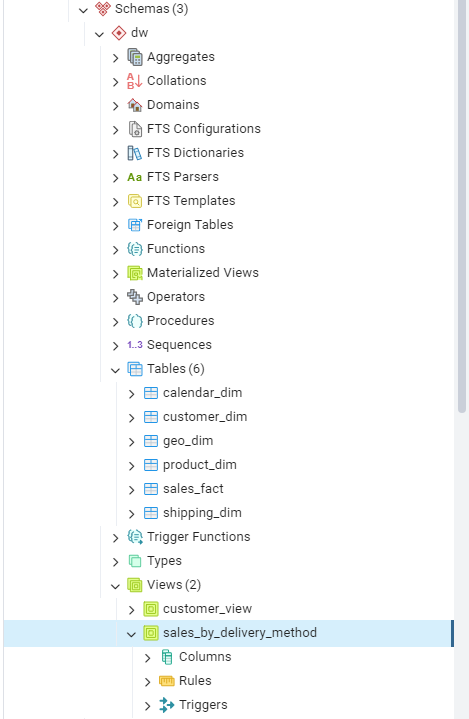


Рисунок 44 View2 на схеме

1. Рассчитать среднюю прибыль по городам.

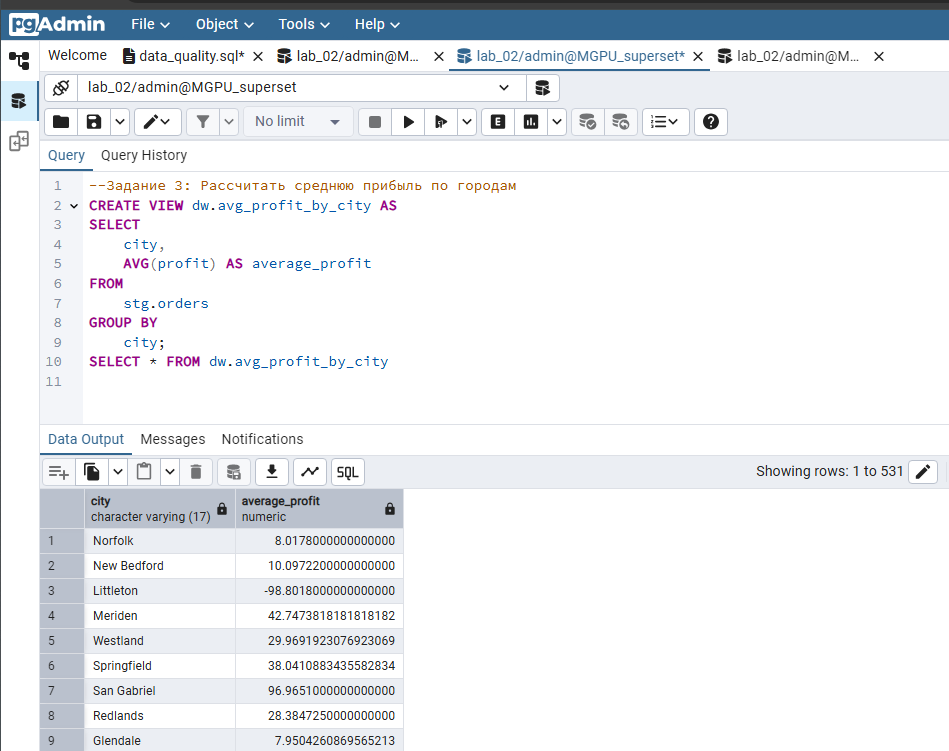
****

Рисунок 45 Расчёт средней прибыли по городам

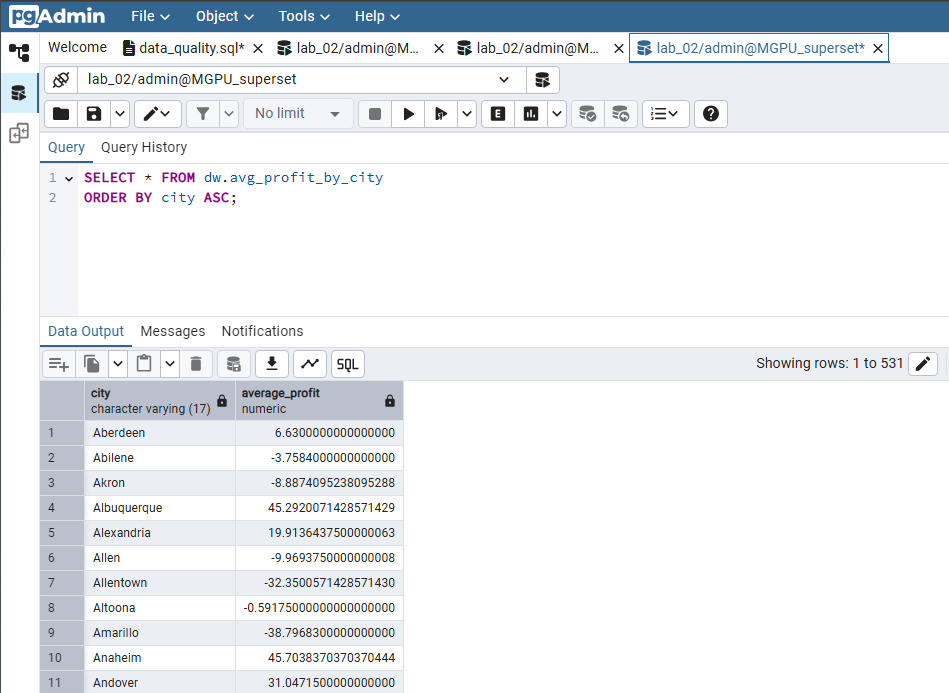
****

Рисунок 46 Вывод городов в алфавитном порядке

**Доп задания (\*попробовать выполнить):**

1. Оптимизировать индексы для ускорения запросов.
2. Создать представления (views) для часто используемых запросов.
3. Написать процедуру для автоматической загрузки данных.

**Контрольные вопросы:**

1. В чем разница между схемами stg и dw?

Staging – временная зона для сырых данных. Хранит неизменённые копии данных из разных источников. Если dw “сломается”, staging позволяет перезагрузить данные без обращения к исходным системам (буфет первичной обработки).

DW (data warehouse) – хранилище очищенных, преобразованных и структурированных данных для анализа (отчётов, дашбордов и тд). DW содержит справочники (измерения, dimensions), фактовые таблицы (facts).

Staging — это "песочница" для подготовки данных и страховка от потерь.

DW — слой для анализа, где справочники и факты работают вместе, обеспечивая скорость и удобство запросов.

1. Зачем нужны суррогатные ключи?

Суррогатные ключи помогают:

Изолировать DW от изменений в источниках, упрощают моделирование, обеспечивают уникальность, поддерживают историю изменений.

1. Какие преимущества дает деморализация данных в DW?

Скорость запросов, упрощение анализа, оптимизация для агрегаций, снижение нагрузки на СУБД.

**Заключение**

В результате выполнения данной работы были освоены ключевые принципы проектирования реляционных баз данных, включая концепцию многослойной архитектуры хранения данных, определение связей между сущностями и выбор оптимальных типов данных. Практическая реализация этапов загрузки данных позволила закрепить навыки работы с SQL-запросами. Полученные знания и опыт являются основой для дальнейшего изучения расширенных возможностей СУБД, таких как оптимизация запросов, транзакции и администрирование баз данных.