ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
(ГАОУ ВО МГПУ)
Институт цифрового образования
Департамент информатики, управления и технологий

Практическая работа № 2.1

по дисциплине «Платформы Data Engineering»

Выполнила:
студентка группы БД-251м
Направление подготовки/Специальность
38.04.05 - Бизнес-информатика
Савкина Мария Алексеевна
St_84
Вариант 25

Проверил:
Кандидат технических наук, доцент
Босенко Тимур Муртазович

Москва 2025

Вариант 25. Задание:

Бизнес-кейс и вопрос для анализа: Время от заказа до отгрузки. Найти 5 подкатегорий с самым долгим средним временем подготовки заказа к отгрузке.

Mart-модель, которую необходимо создать в dbt: mart_order_processing_time

Архитектура DWH (граф зависимостей):

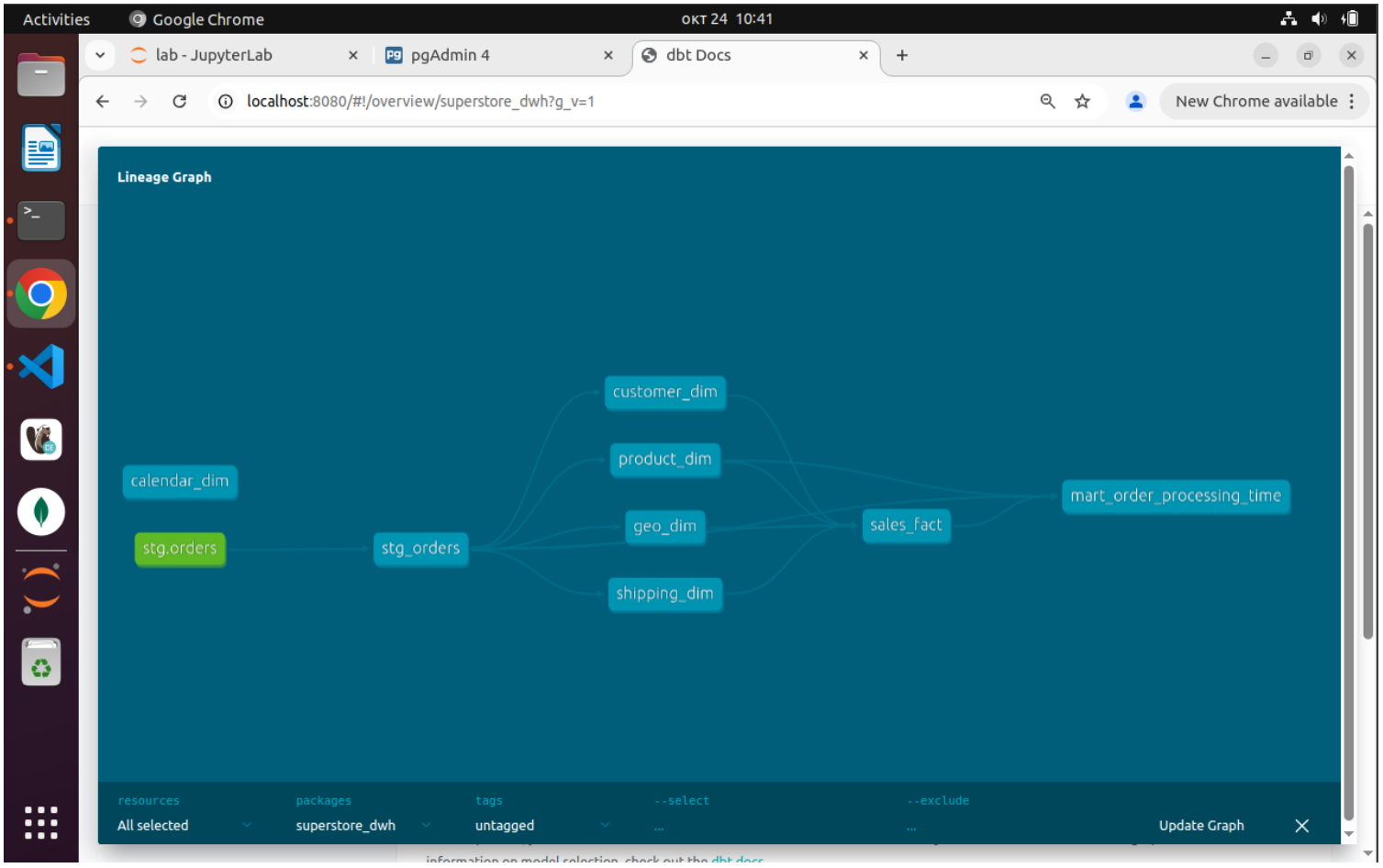

Рис 1. Граф зависимостей (lineage) проекта

Ключевые фрагменты кода:

1) Код модели stg_orders.sql

```
-- models/staging/stg_orders.sql
-- Эта модель читает данные из исходной таблицы stg.orders,
-- приводит их к нужным типам и исправляет ошибку с почтовым кодом.
-- Все последующие модели будут ссылаться на эту, а не на исходную таблицу.

SELECT
-- Приводим все к нижнему регистру для консистентности в dbt
"order_id",
("order_date")::date as order_date,
("ship_date")::date as ship_date,
"ship_mode",
"customer_id",
"customer_name",
"segment",
"country",
"city",
"state",
-- Исправляем проблему с Burlington прямо здесь, один раз и навсегда
CASE
WHEN "city" = 'Burlington' AND "postal_code" IS NULL THEN '05401'
ELSE "postal_code"
END as postal_code,
"region",
"product_id",
"category",
"subcategory" as sub_category, -- переименовываем для соответствия
"product_name",
"sales",
"quantity",
"discount",
"profit"
FROM {{ source('stg', 'orders') }}
```

2) Код модели sales_fact.sql.

```
-- Создает таблицу фактов, объединяя все измерения
SELECT
-- Суррогатные ключи из измерений
cd.cust_id,
pd.prod_id,
sd.ship_id,
gd.geo_id,
-- Ключи для календаря
to_char(o.order_date, 'yyyymmdd')::int AS order_date_id,
to_char(o.ship_date, 'yyyymmdd')::int AS ship_date_id,
-- Бизнес-ключ и метрики
o.order_id,
o.sales,
```

```
o.profit,
o.quantity,
o.discount
FROM {{ ref('stg_orders') }} AS o
LEFT JOIN {{ ref('customer_dim') }} AS cd ON o.customer_id = cd.customer_id
LEFT JOIN {{ ref('product_dim') }} AS pd ON o.product_id = pd.product_id
LEFT JOIN {{ ref('shipping_dim') }} AS sd ON o.ship_mode = sd.ship_mode
LEFT JOIN {{ ref('geo_dim') }} AS gd ON o.postal_code = gd.postal_code AND o.city = gd.city AND
o.state = gd.state
```

3) Код модели mart_order_processing_time.sql (Вариант 25)

```
SELECT
   p.sub_category,
   AVG(DATE_PART('day', s.ship_date - s.order_date)) AS avg_processing_days
FROM {{ ref('sales_fact') }} AS f
LEFT JOIN {{ ref('product_dim') }} AS p
   ON f.prod_id = p.prod_id
LEFT JOIN {{ ref('stg_orders') }} AS s
   ON f.order_id = s.order_id
GROUP BY p.sub_category
ORDER BY avg_processing_days DESC
LIMIT 5
```

4) Файл schema.yml с тестами всех моделей

```
# Путь к файлу: models/marts/schema.yml

version: 2

models:
- name: shipping_dim
columns:
- name: ship_id
tests:
- unique
- not_null

- name: customer_dim
columns:
- name: cust_id
tests:
- unique
- not_null

- name: geo_dim
columns:
- name: geo_id
tests:
- unique
- not_null

- name: product_dim
```

```
columns:
- name: prod_id
tests:
- unique
- not_null

- name: sales_fact
columns:
- name: cust_id
tests:
- relationships:
arguments:
to: ref('customer_dim')
field: cust_id

- name: mart_order_processing_time
columns:
- name: sub_category
tests:
- not_null
- unique
- name: avg_processing_days
tests:
- not_null
```

Результаты:

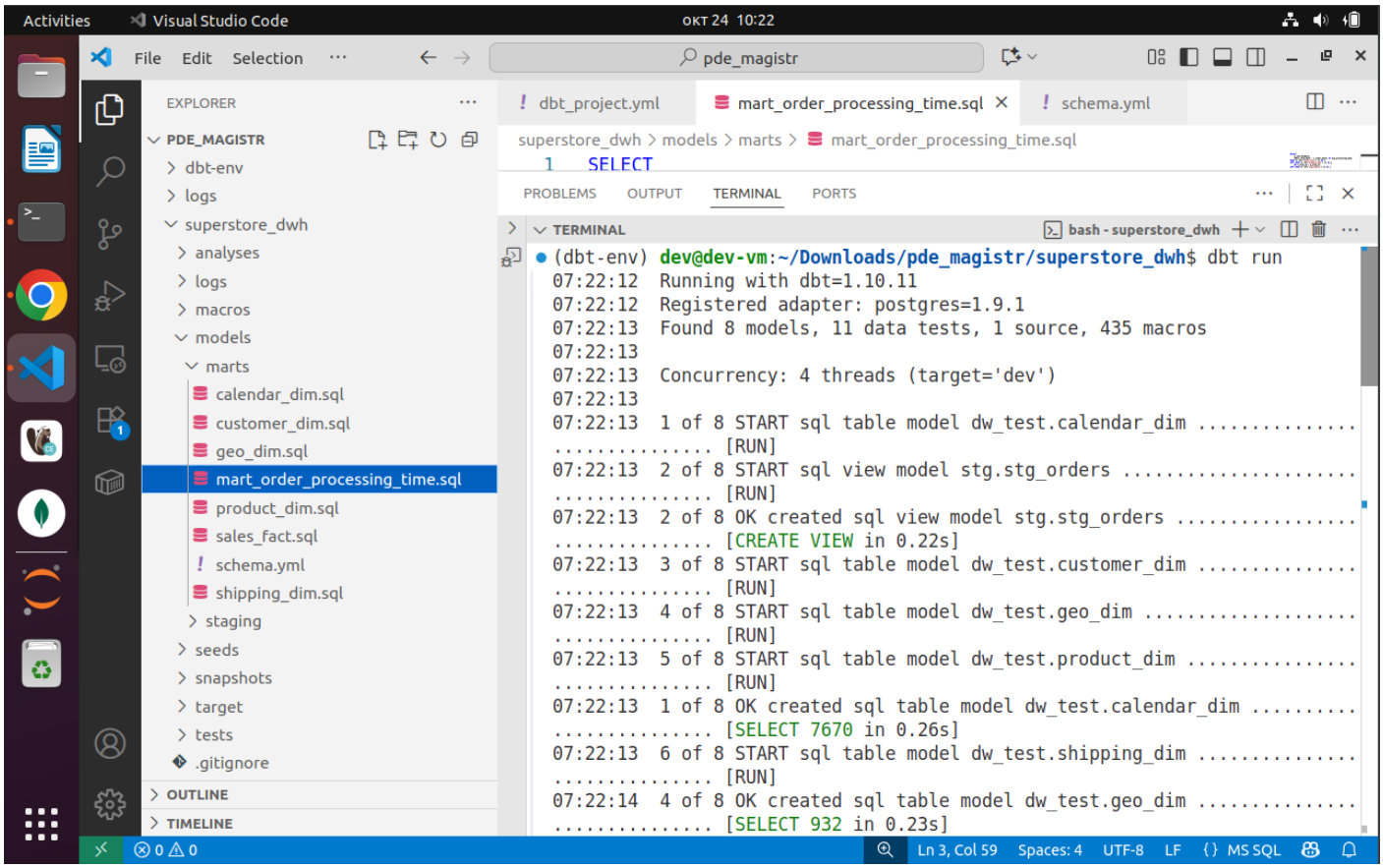

Рис. 2. Скриншот успешного выполнения dbt run (часть 1)

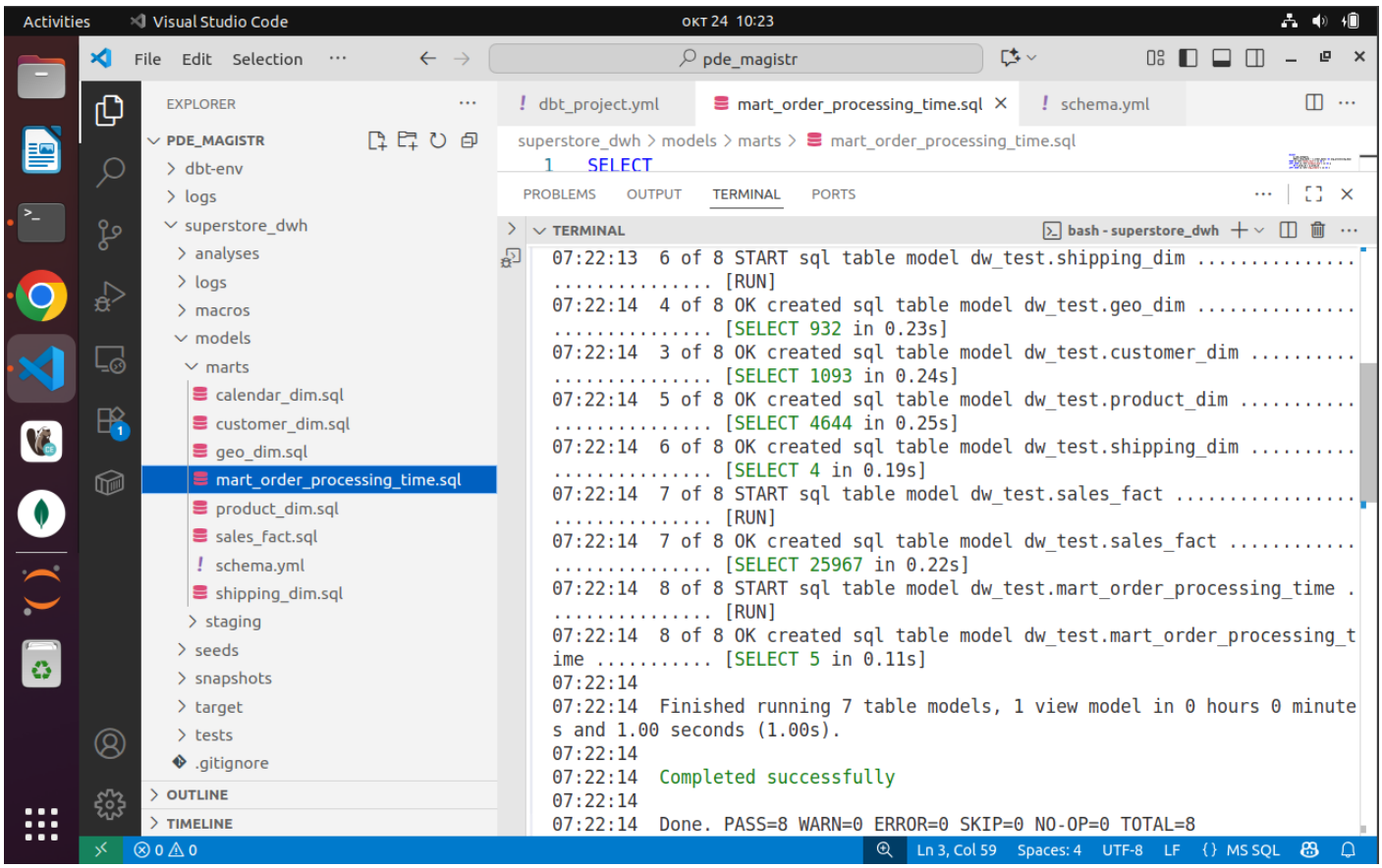

Рис. 3. Скриншот успешного выполнения dbt run (часть 2)

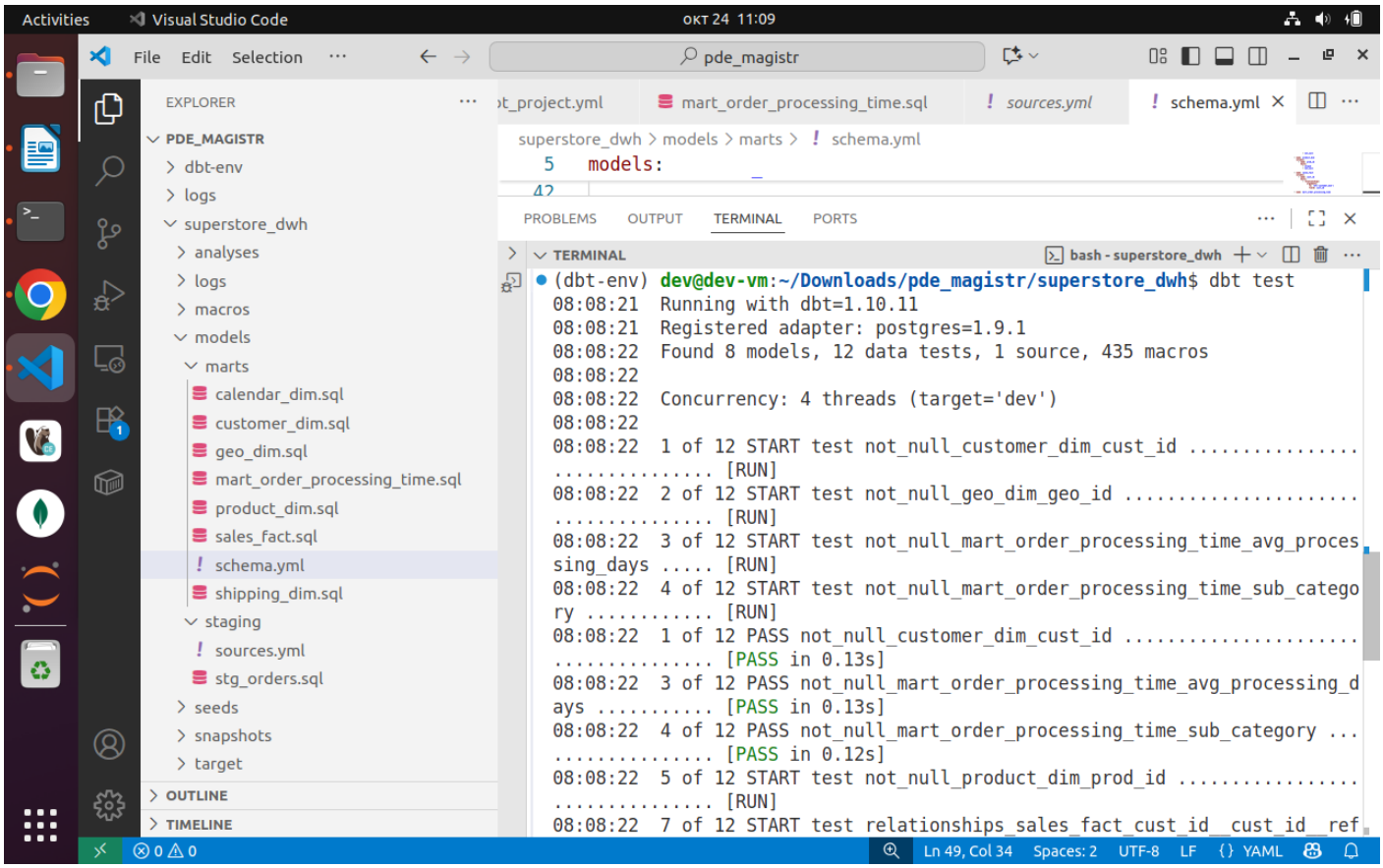

Рис. 4. Скриншот успешного выполнения dbt test (часть 1)

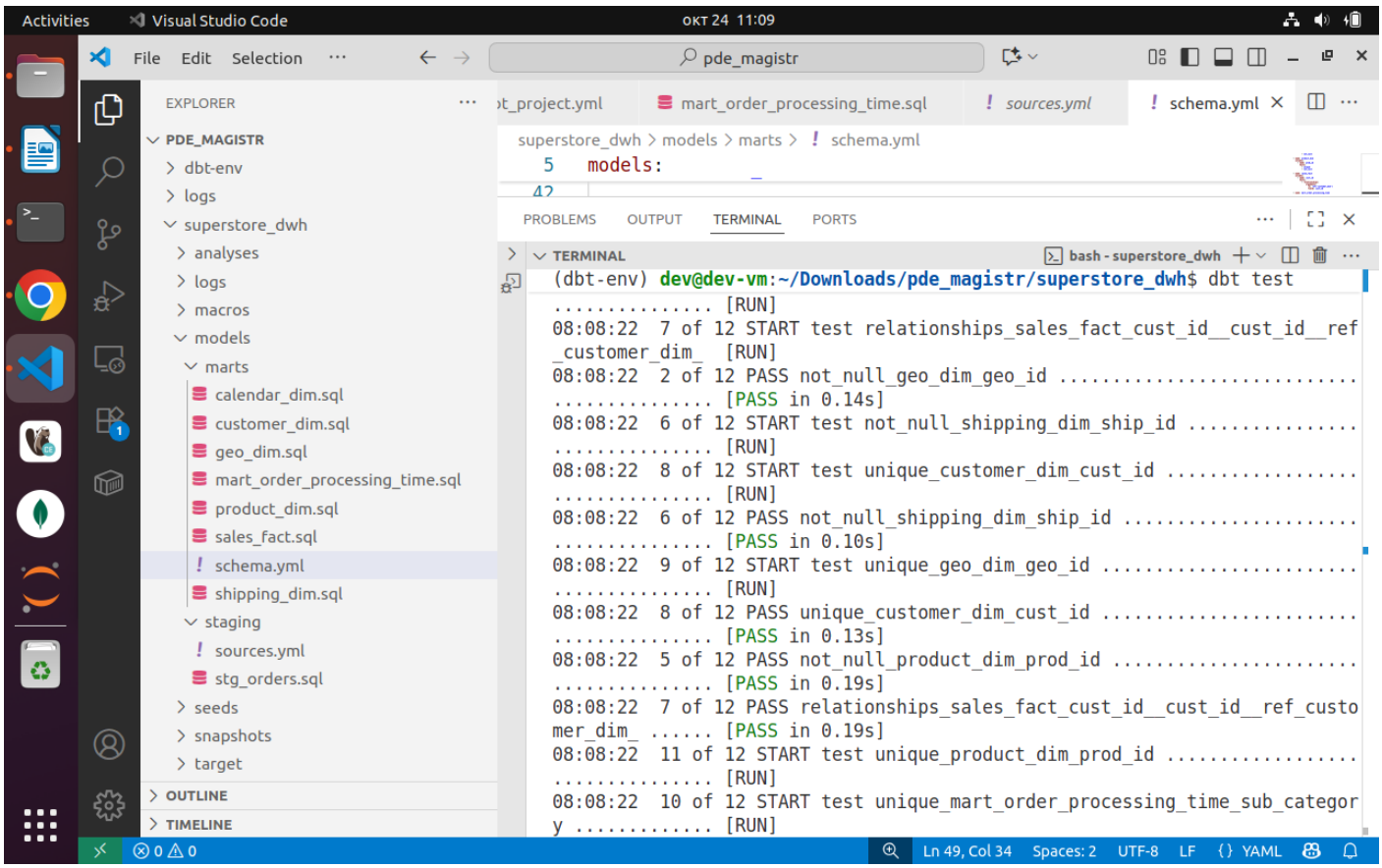

Рис. 5. Скриншот успешного выполнения dbt test (часть 2)

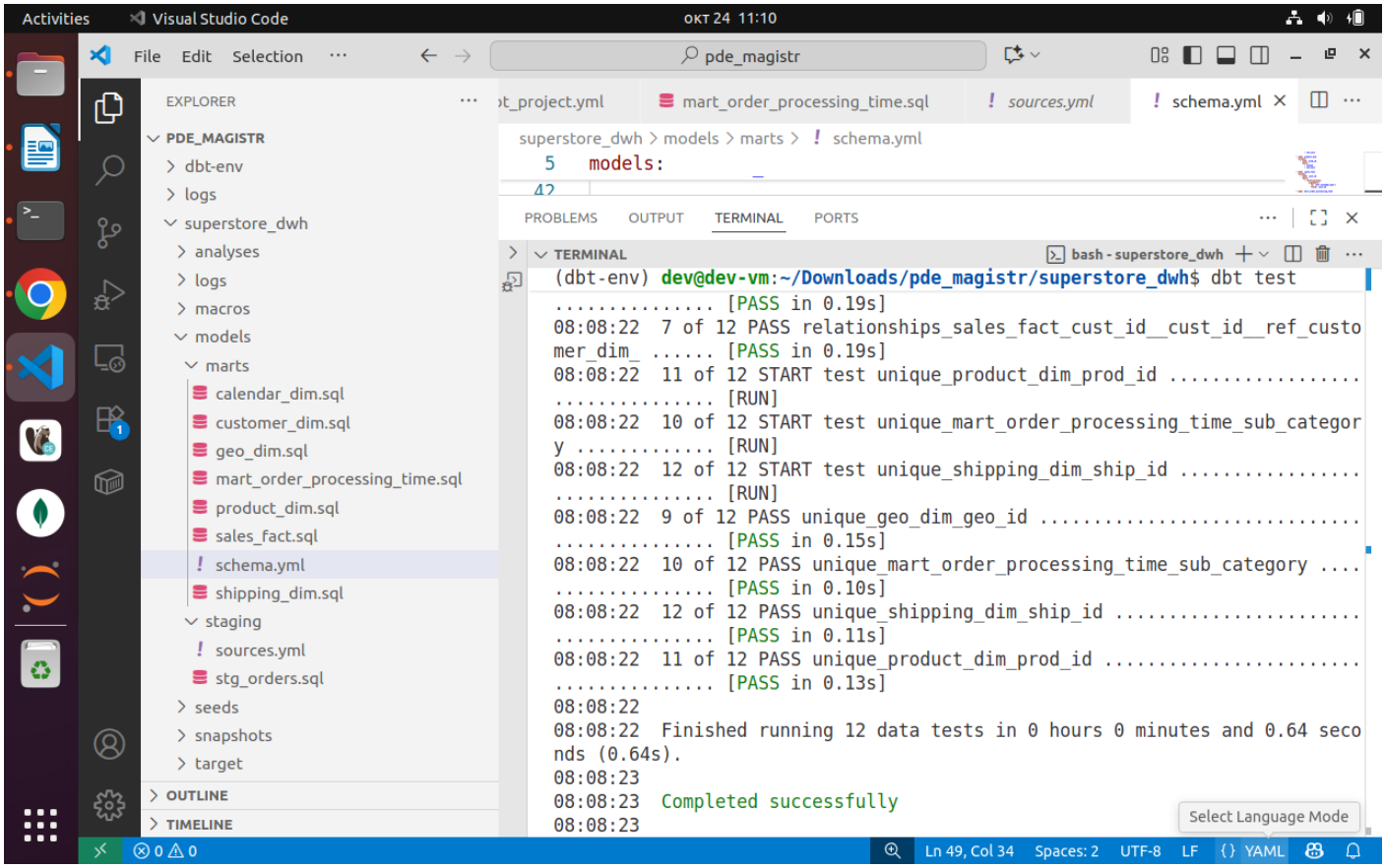

Рис. 6. Скриншот успешного выполнения dbt test (часть 3)

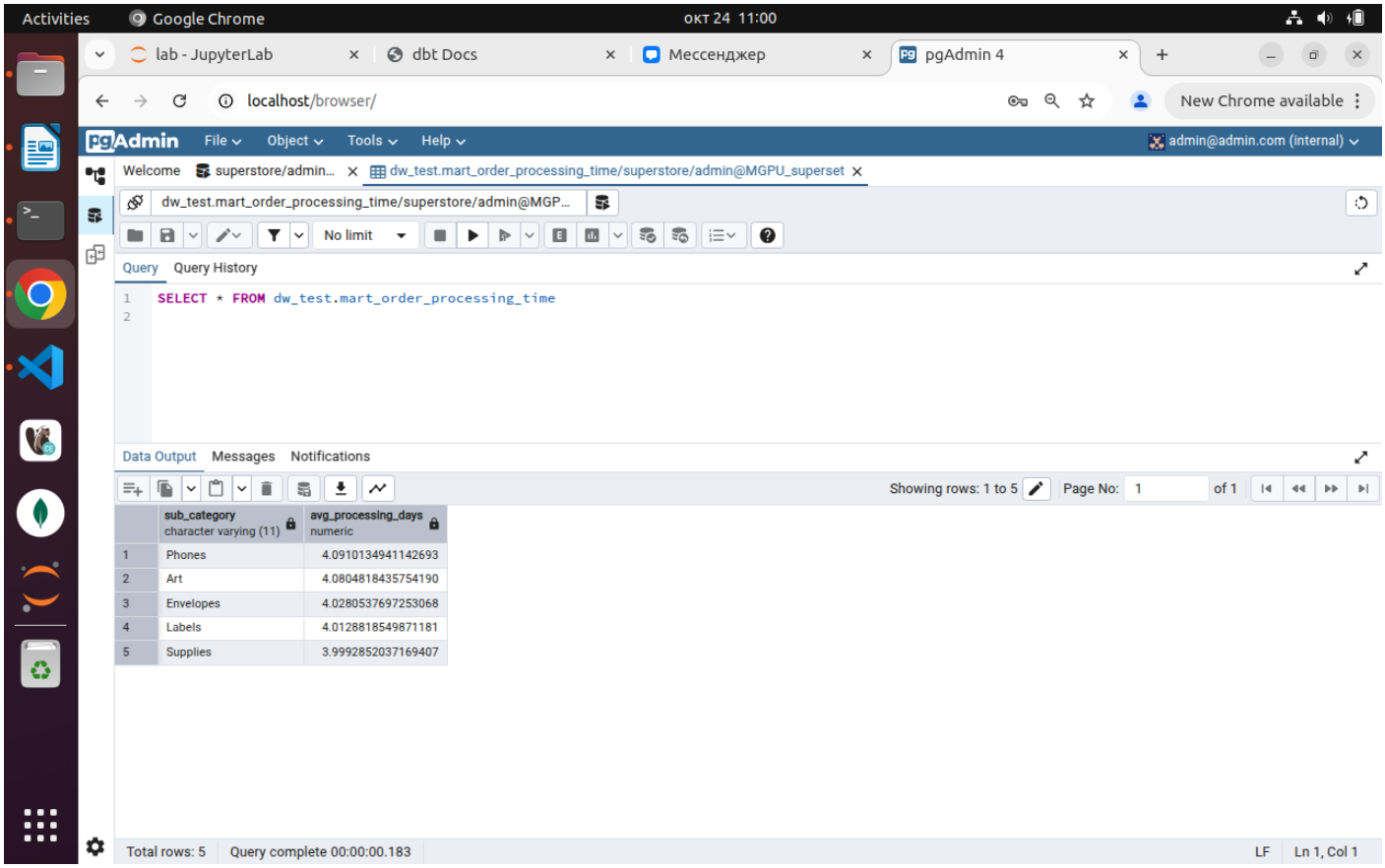

Рис. 7. Результат выполнения модели mart_order_processing_time

Выводы

В ходе выполнения практической работы по разработке и тестированию dbt-моделей для бизнес-логики мною были выявлены следующие преимущества dbt для реализации DWH по сравнению с написанием DDL/DML скриптов вручную:

Автоматическое построение графа зависимостей (lineage) в dbt docs наглядно демонстрирует работу проекта, что может быть полезно как на этапе разработки, так и при работе с проектом другими членами команды. Автоматическое документирование проекта.

Встроенное тестирование данных, которое позволяет автоматически проверять качество данных по заданным параметрам, прописанным в .yml файле с помощью команды dbt test.

Автоматизация рутинных процессов, поскольку dbt сам определяет правильную последовательность сборки моделей, и, как следствие, сокращение времени на реализацию типовых задач.