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

Макарова Виктория Сергеевна

**САМОСТОЯТЕЛЬНАЯ РАБОТА 1**

. Интеграция данных из разных источников (баз данных)

Направление подготовки

38.03.05 Бизнес-информатика

Профиль подготовки

Аналитика данных и эффективное управление

Курс обучения: 4

Форма обучения: очная

Преподаватель: кандидат технических наук,

доцент Босенко Тимур Муртазович

Москва

2025

**Тема:** разработка ETL-процесса для интеграции данных между PostgreSQL и MySQL с использованием Pentaho Data Integration.

**Задачи:**

− Создать исходные таблицы в PostgreSQL с различными наборами данных.

− Настроить целевые таблицы в MySQL для приема данных.

− Разработать процессы трансформации данных в Pentaho.

− Реализовать механизмы обработки ошибок и валидации данных.

− Создать представления для связанных данных.

**Необходимое ПО:**

- Конфигурация devops\_dba\_25.ova.

**Ход работы**

**Шаг 1**. Проверяем доступность СУБД Postgre SQL (локальная СУБД). Выполнение задания производим в pgAdmin4, который использует для работы HTTP (Рис. 1). Перед запускаем отключаем mongo, затем перезапускаем postgre (Рис. 2 - Рис.3).

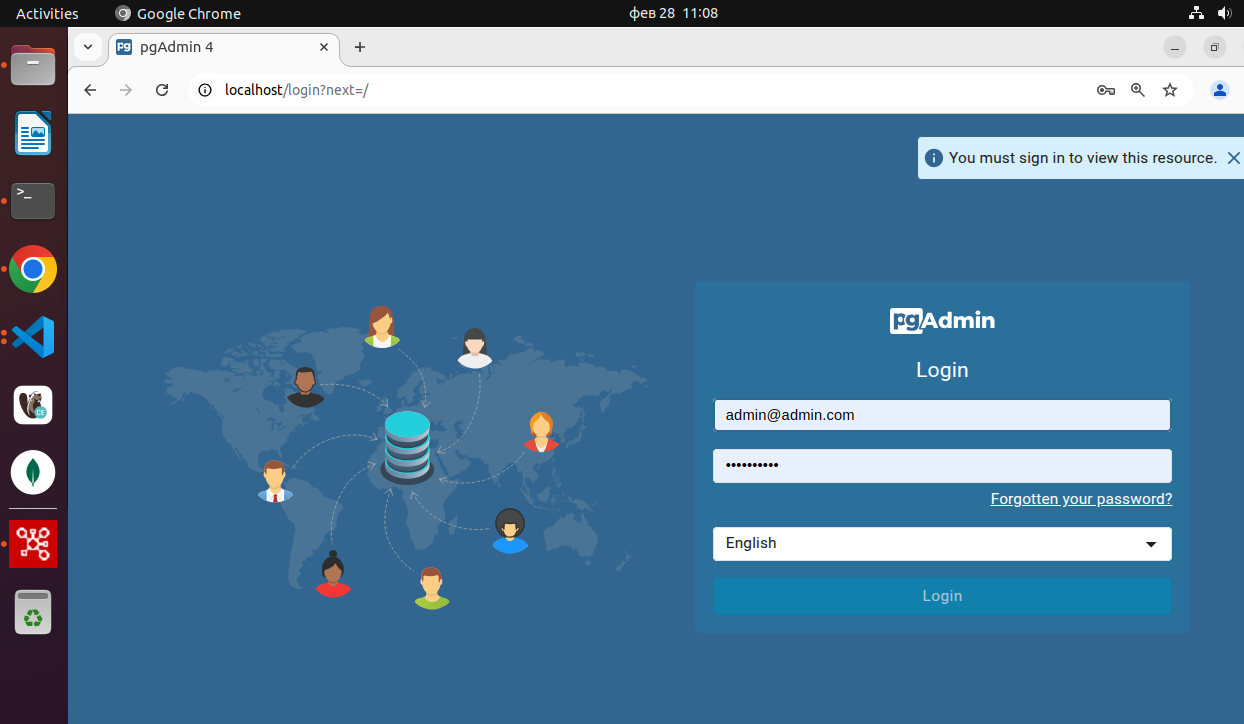


Рис. 1

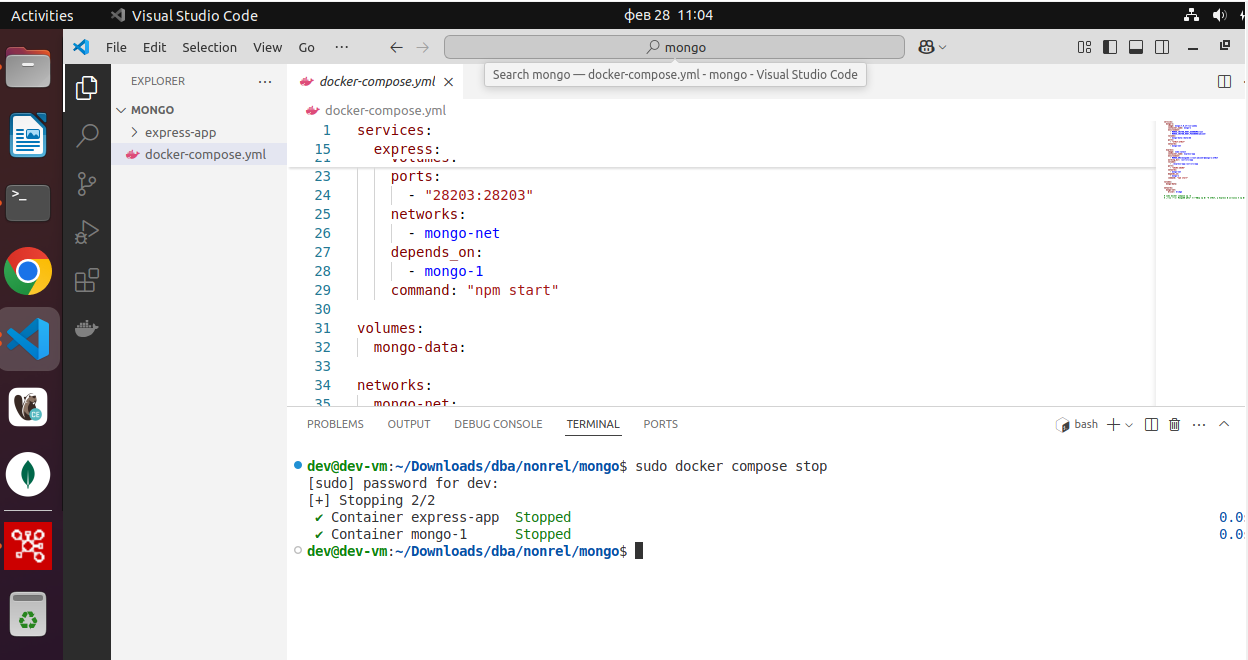


Рис. 2

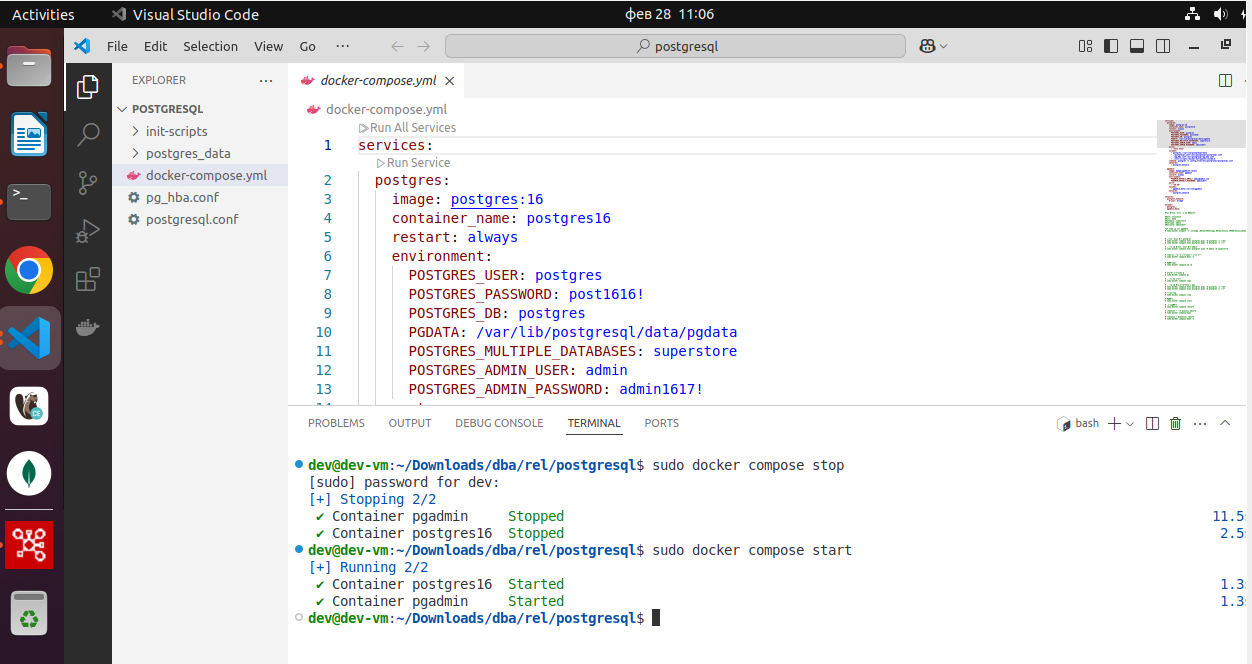


Рис. 3

**Шаг 2.** Создаем базу данных со своим уникальным логином id – st\_94 (Рис. 4).

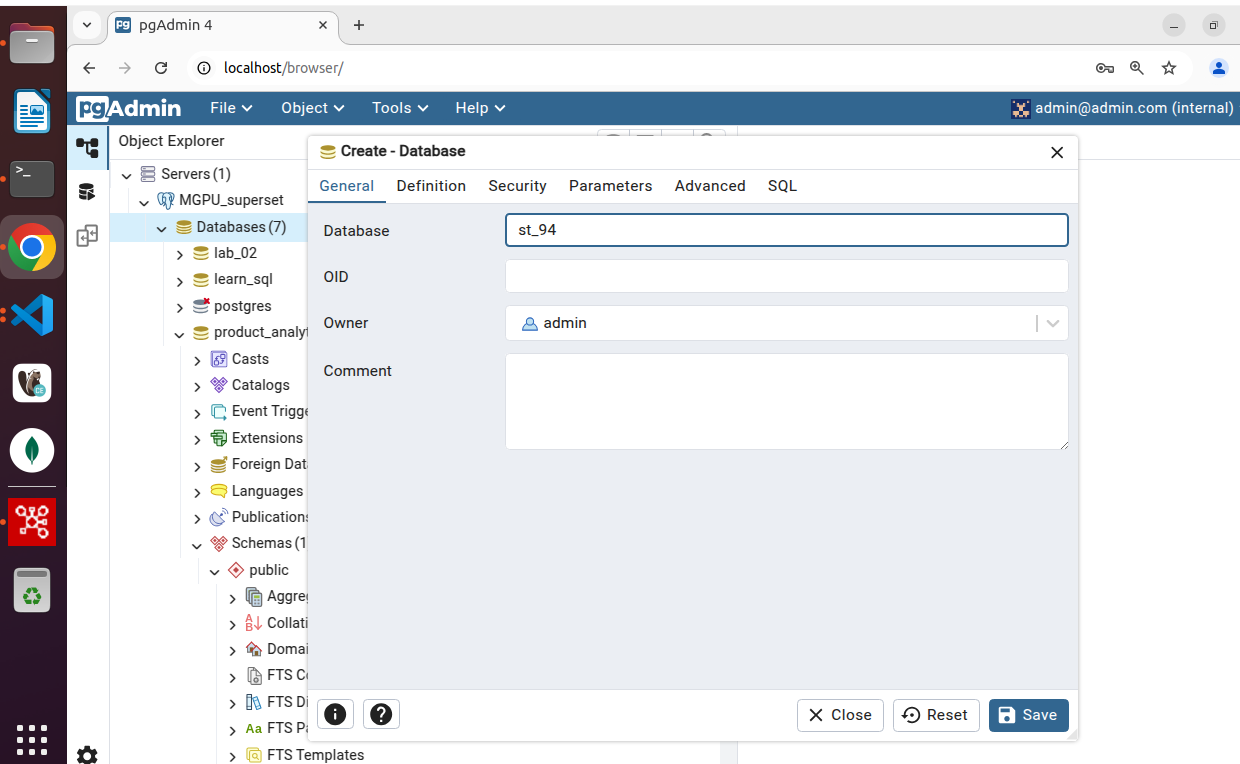


Рис. 4

**Задание 1**

**Шаг 3.** Создаем таблицу и данные, согласно варианту st\_94, таблица

sales (id, product\_id, sale\_date, quantity, price, discount) (Рис. 5).

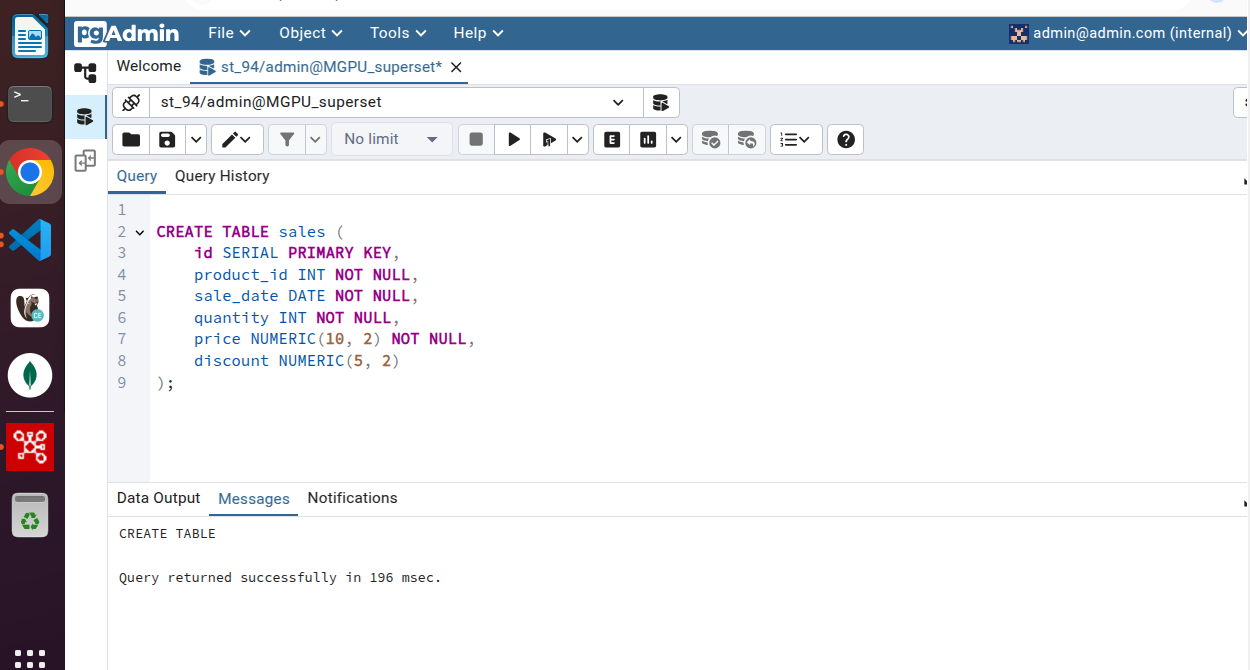


Рис. 5

CREATE TABLE sales (

id SERIAL PRIMARY KEY,

product\_id INT NOT NULL,

sale\_date DATE NOT NULL,

quantity INT NOT NULL,

price NUMERIC(10, 2) NOT NULL,

discount NUMERIC(5, 2)

);

Заполняем таблицу данными (Рис. 6)

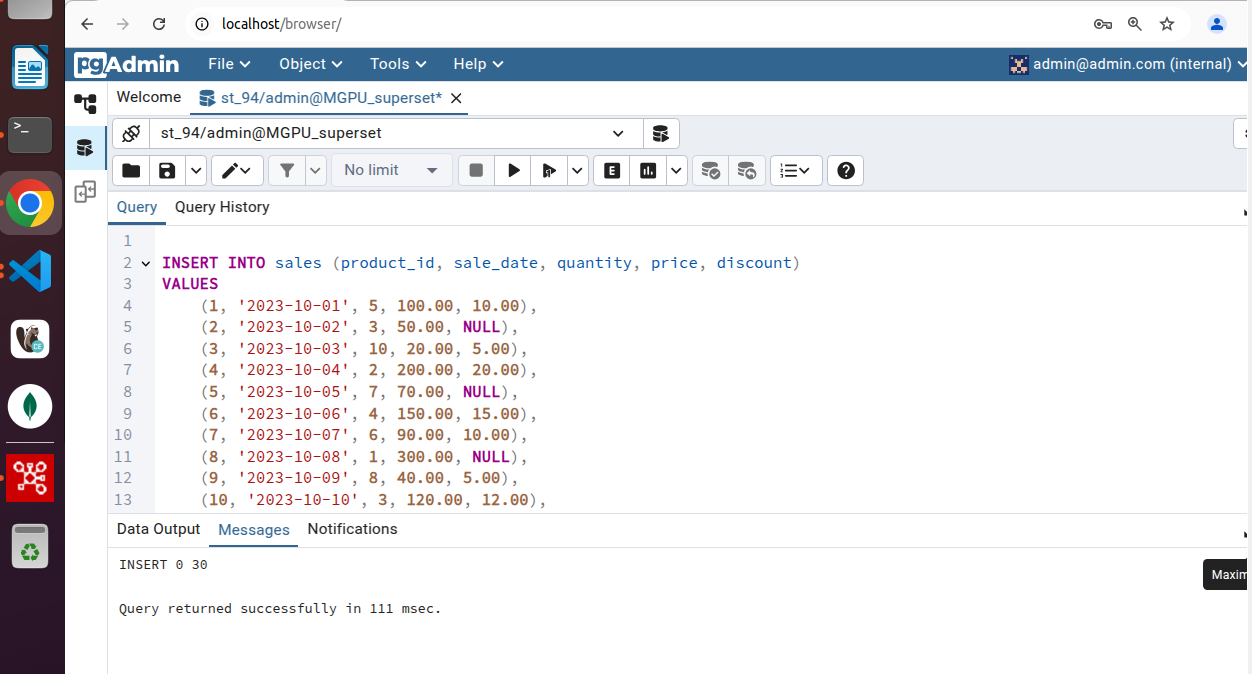


Рис. 6

INSERT INTO sales (product\_id, sale\_date, quantity, price, discount)

VALUES

(1, '2023-10-01', 5, 100.00, 10.00),

(2, '2023-10-02', 3, 50.00, NULL),

(3, '2023-10-03', 10, 20.00, 5.00),

(4, '2023-10-04', 2, 200.00, 20.00),

(5, '2023-10-05', 7, 70.00, NULL),

(6, '2023-10-06', 4, 150.00, 15.00),

(7, '2023-10-07', 6, 90.00, 10.00),

(8, '2023-10-08', 1, 300.00, NULL),

(9, '2023-10-09', 8, 40.00, 5.00),

(10, '2023-10-10', 3, 120.00, 12.00),

(11, '2023-10-11', 5, 80.00, NULL),

(12, '2023-10-12', 2, 250.00, 25.00),

(13, '2023-10-13', 4, 60.00, 6.00),

(14, '2023-10-14', 6, 110.00, NULL),

(15, '2023-10-15', 9, 30.00, 3.00),

(16, '2023-10-16', 3, 180.00, 18.00),

(17, '2023-10-17', 7, 50.00, NULL),

(18, '2023-10-18', 2, 220.00, 22.00),

(19, '2023-10-19', 5, 70.00, 7.00),

(20, '2023-10-20', 4, 130.00, NULL),

(21, '2023-10-21', 6, 90.00, 9.00),

(22, '2023-10-22', 1, 350.00, 35.00),

(23, '2023-10-23', 8, 45.00, NULL),

(24, '2023-10-24', 3, 160.00, 16.00),

(25, '2023-10-25', 5, 75.00, 7.50),

(26, '2023-10-26', 2, 280.00, NULL),

(27, '2023-10-27', 4, 65.00, 6.50),

(28, '2023-10-28', 6, 95.00, 9.50),

(29, '2023-10-29', 7, 55.00, NULL),

(30, '2023-10-30', 3, 140.00, 14.00);

Данные успешно добавлены (Рис. 7).

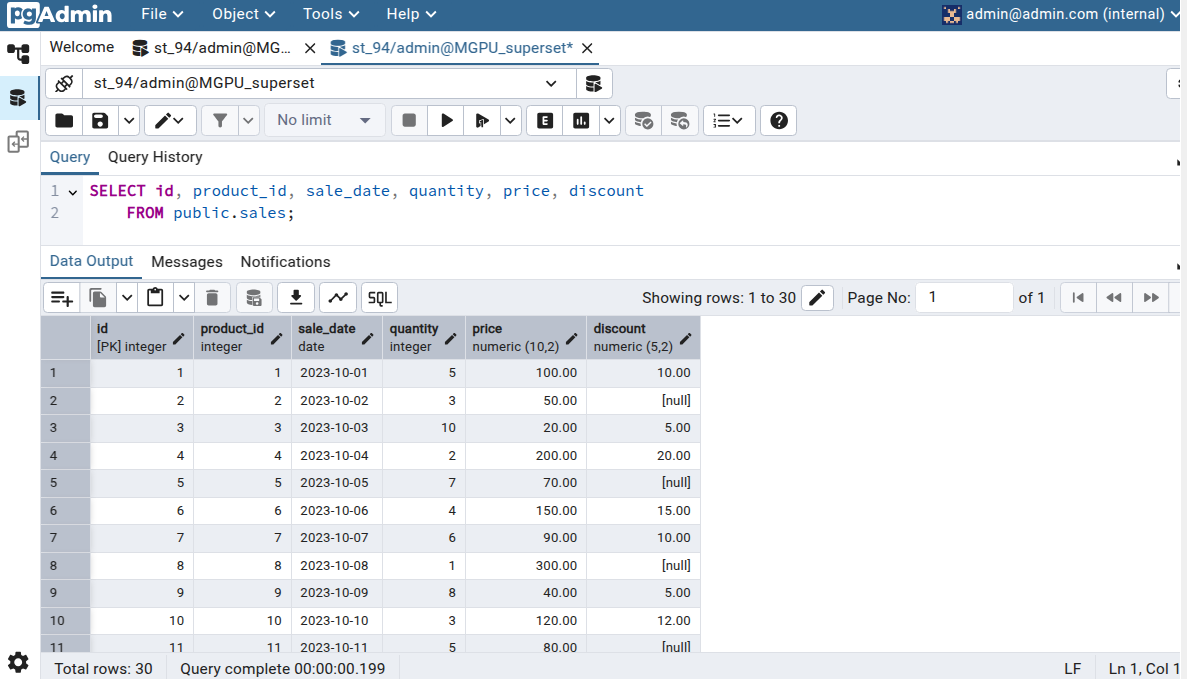


Рис. 7

**Шаг 4.** Запуск Pentaho (Рис. 8)

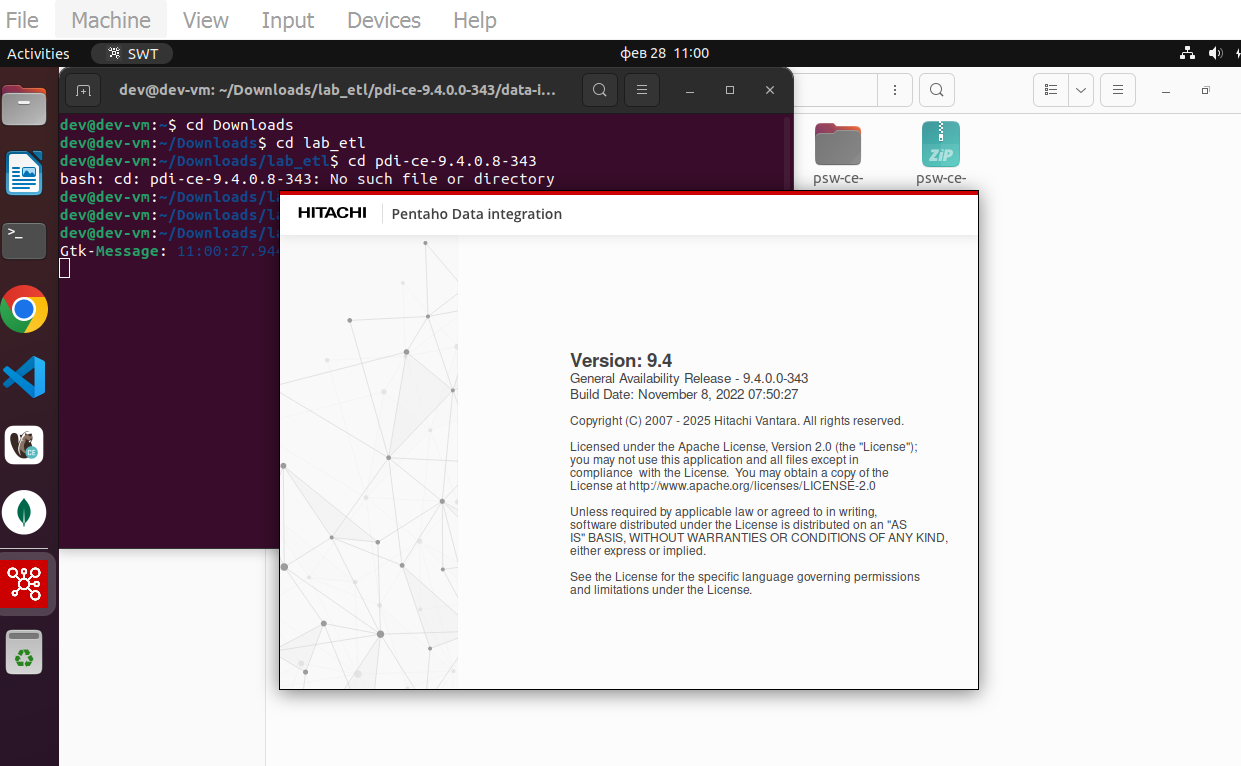


Рис. 8

**Шаг 5.** Сборка трансформации Postgree\_to\_MySQL (Рис. 9).

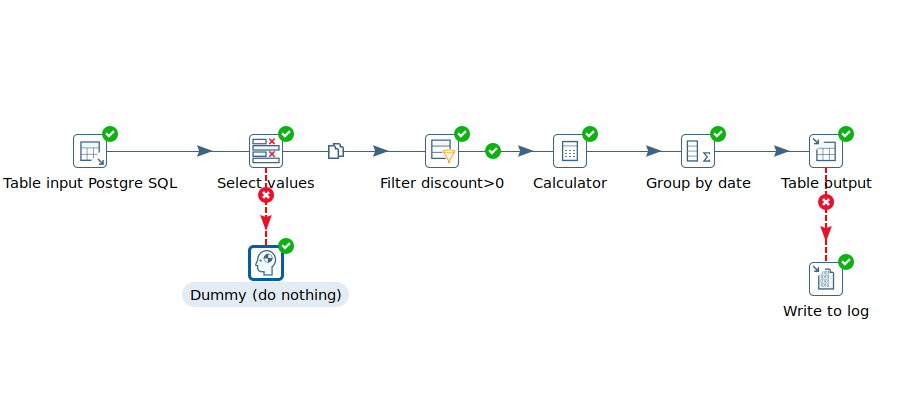


Рис. 9

Настройка объекта **Table input Postgre SQL** (Рис. 10-Рис.11):

Проверка соединения.

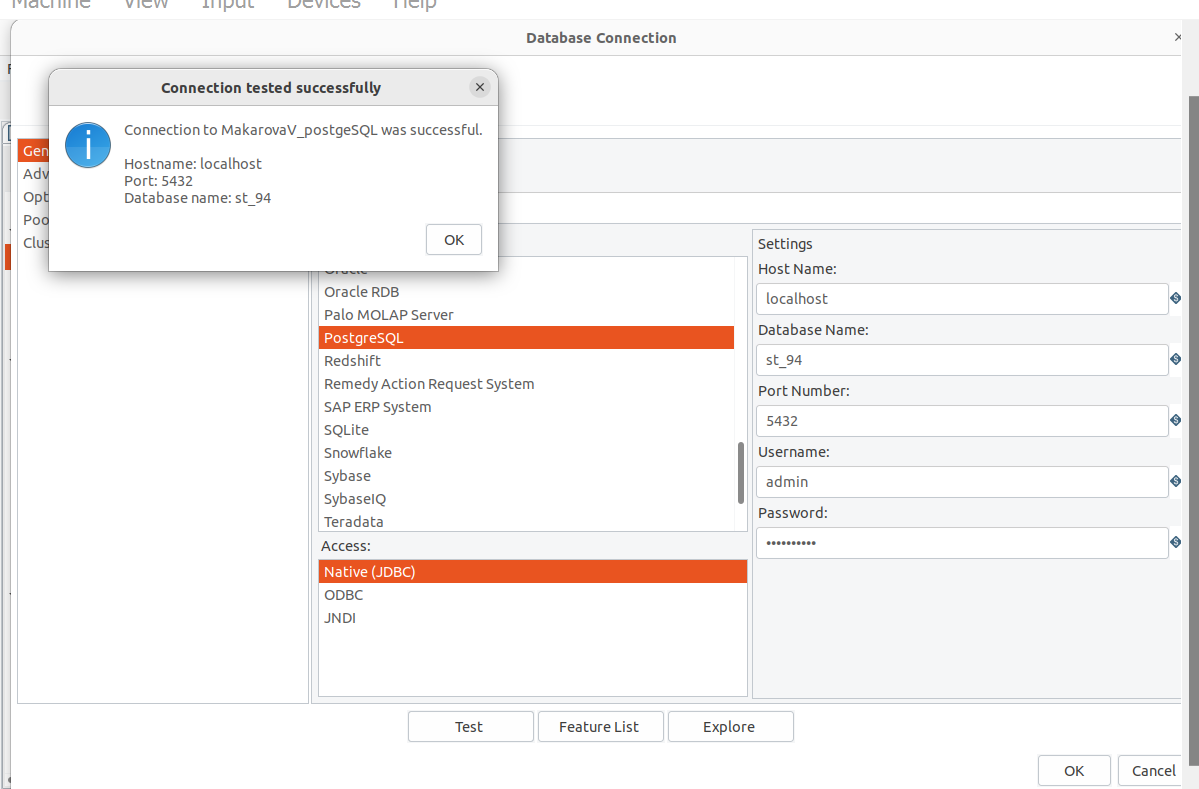


Рис. 10

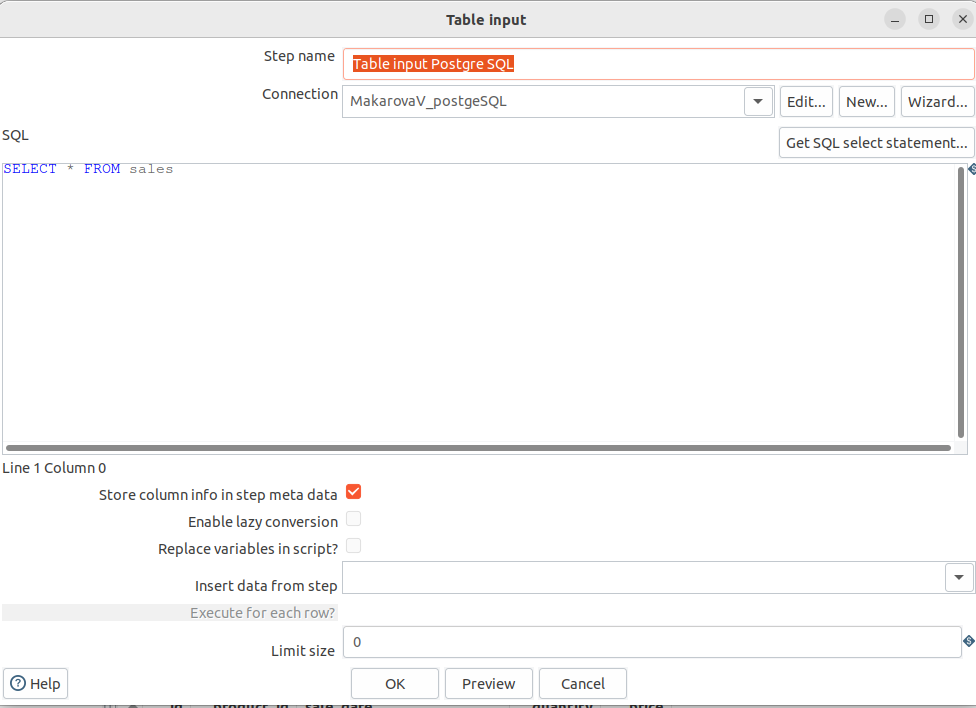


Рис. 11

Настройка объекта **Select Values** (Рис. 12):

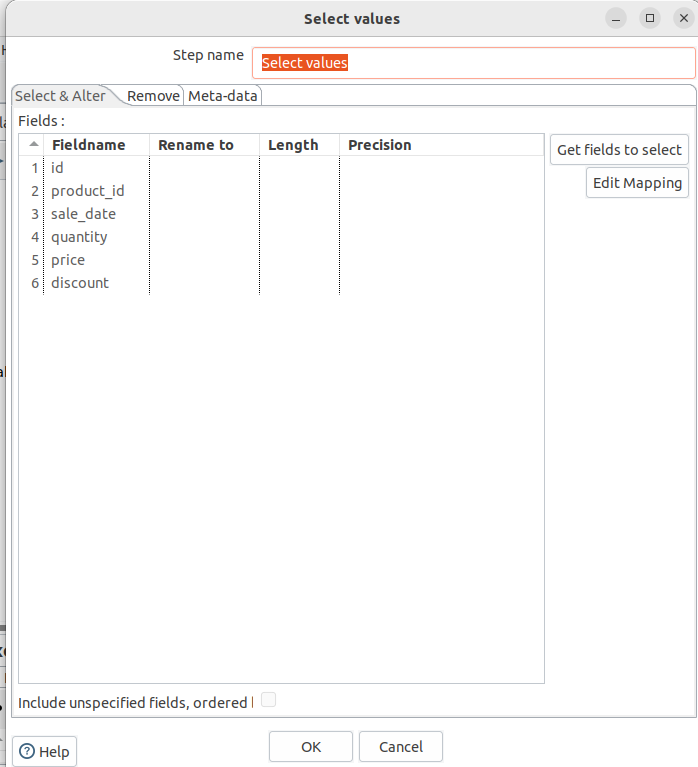


Рис. 12

**Задание 3**

В соотвествии с заданием варианта st\_94 добавляем фильтр: **Фильтрация продаж со скидкой**

Настройка объекта **Filter Row** (Рис. 13):

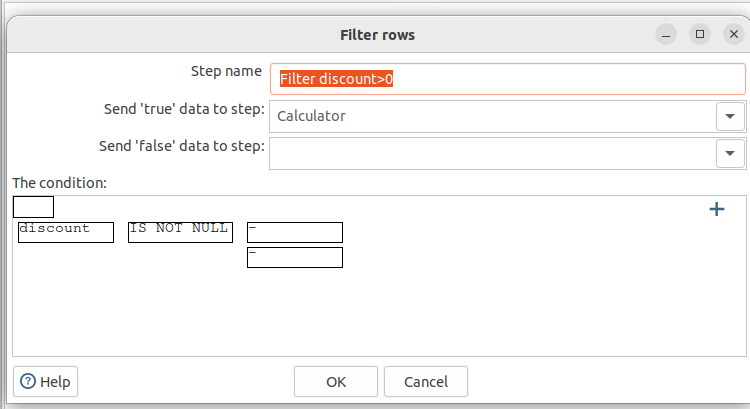


Рис 13

**Задание 4**

Для расчета общей выручки по датам необходимо рассчитать выручку путем умножения количества проданного товара на стоимость.

Настройка объекта **Calculator** (Рис. 14):

Добавим новое поле revenue.

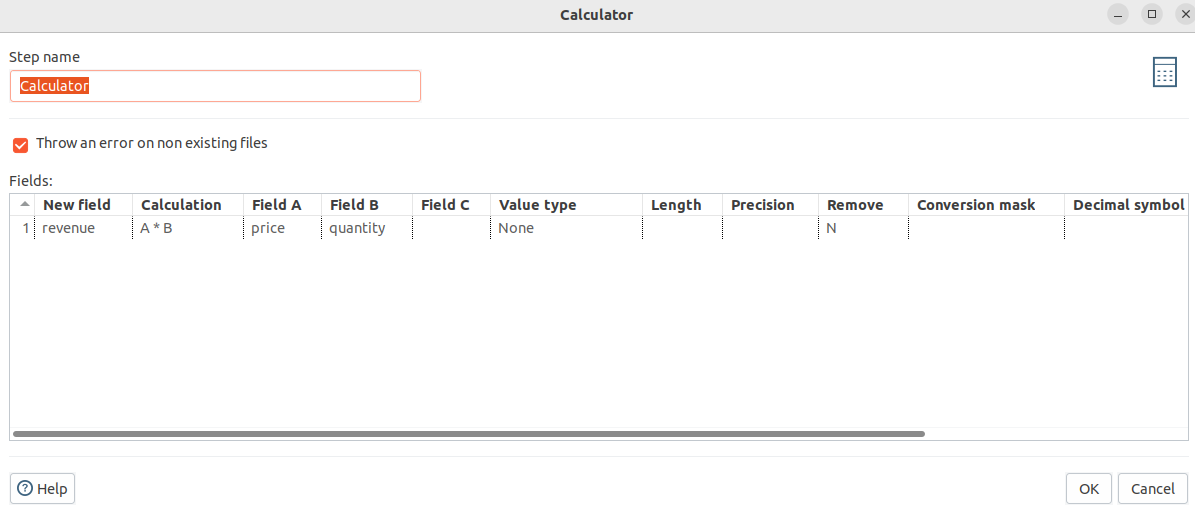
****

Рис. 14

Настройка объекта **Group by Date** (Рис. 15):

Группируем поле revenue по дате и применяем агрегирование - сумма.

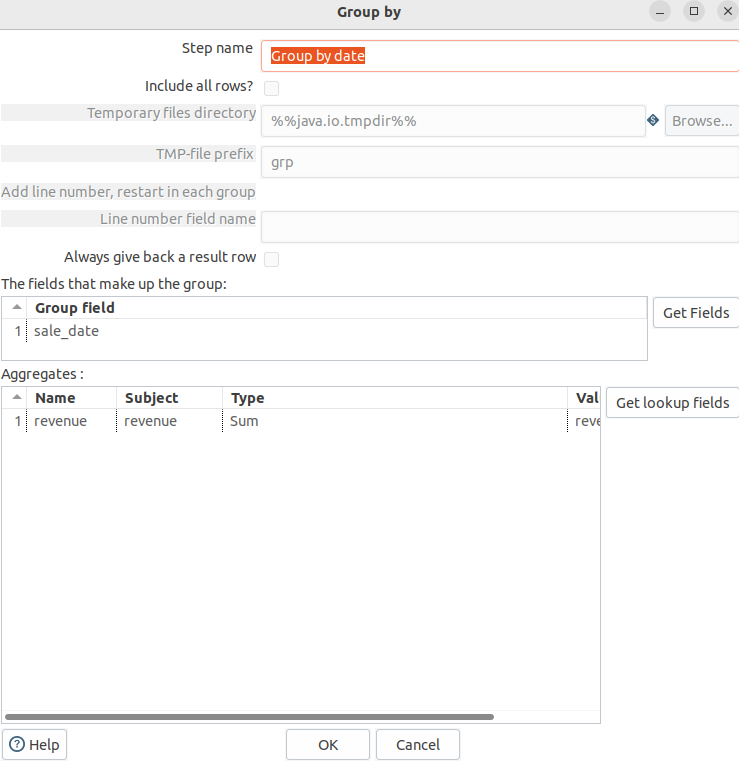


Рис. 15

Настройка объекта **Table output** (Рис. 16):

Проверка подключения:

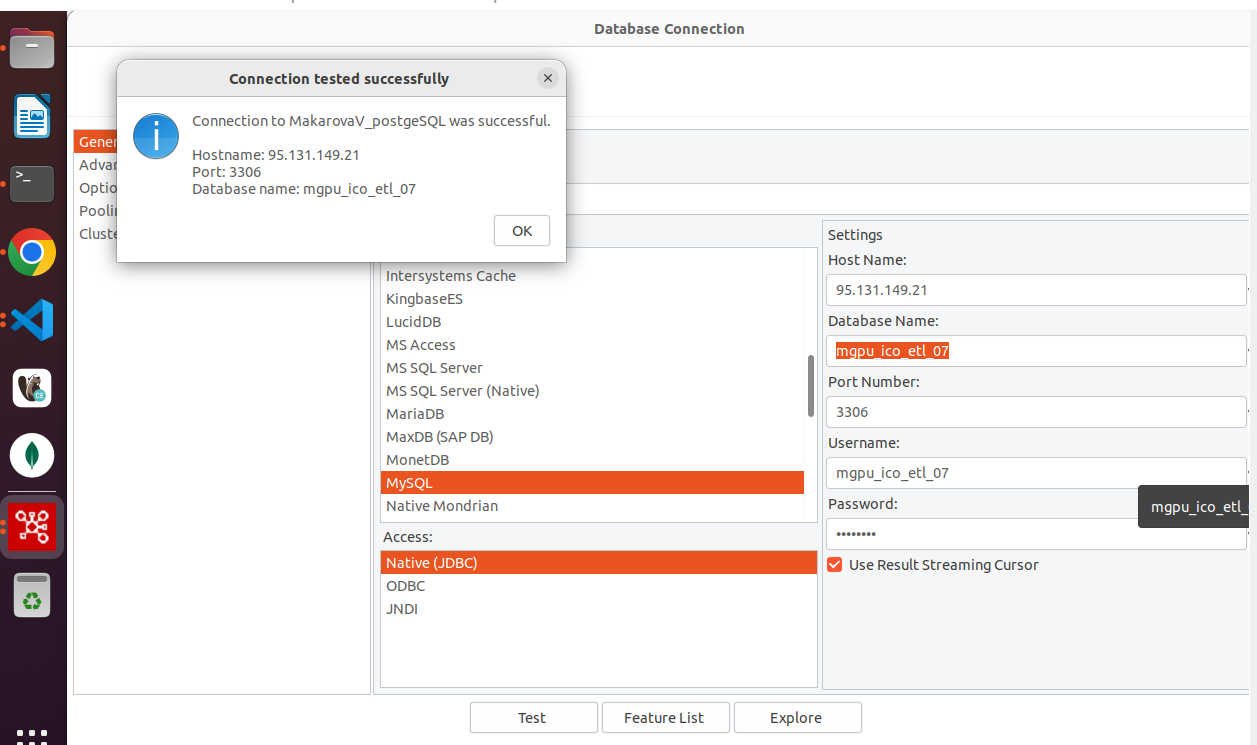


Рис. 16

**Задание 2**

Создадим целевую таблицу **sales\_analytics** в СУБД Mysql (Рис. 17).



Рис. 17

Выберем таблицу **sales\_analytics** (Рис. 18).

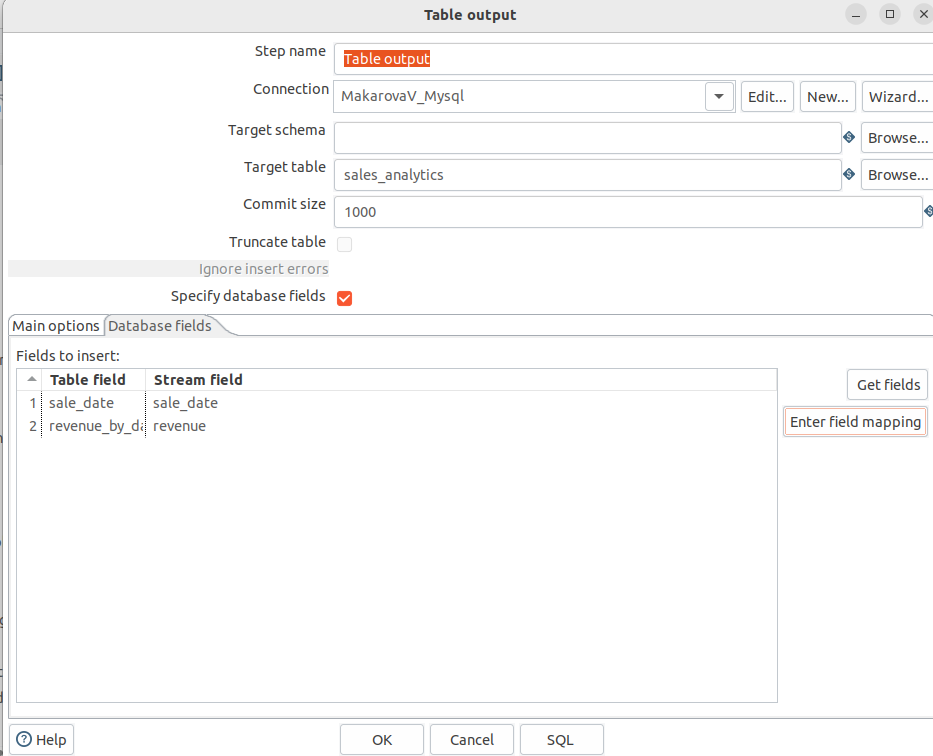


Рис. 18

Проверим маппинг полей (Рис. 19).

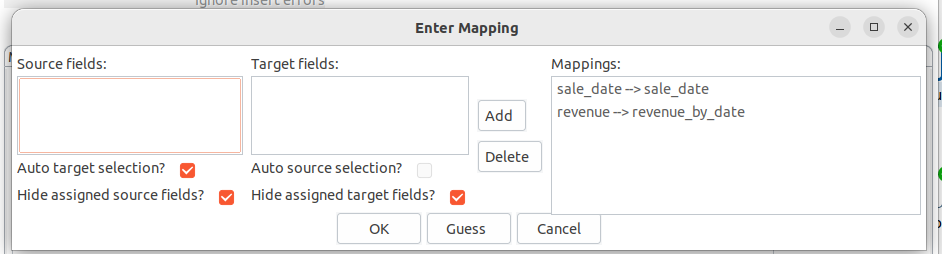


Рис. 19

**Шаг 6**. Запуск **Job CSV\_to\_MySQL.**

В объекте Transformation укажем созданную трансформацию Postgree\_to\_Mysql (Рис. 20).

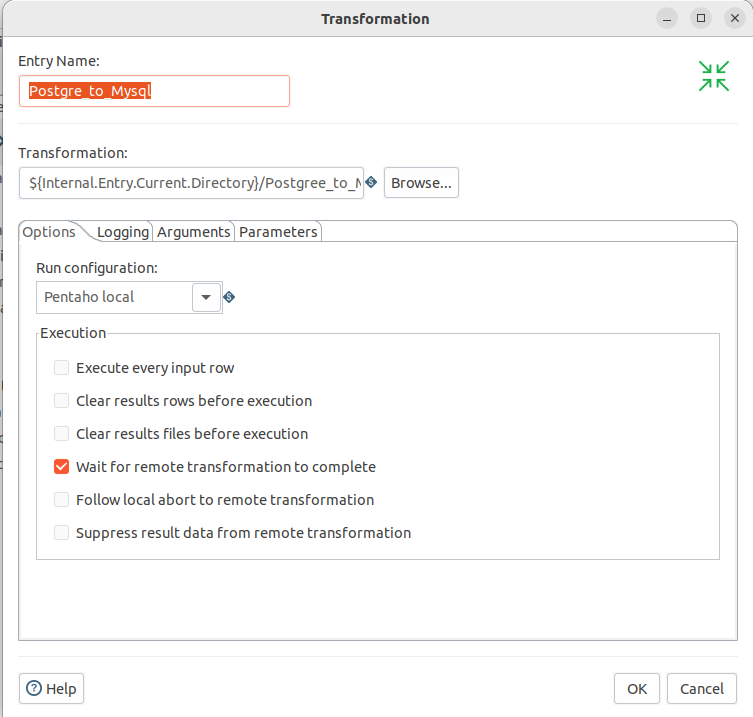


Рис. 20

Job был выполнен без ошибок (Рис. 21).

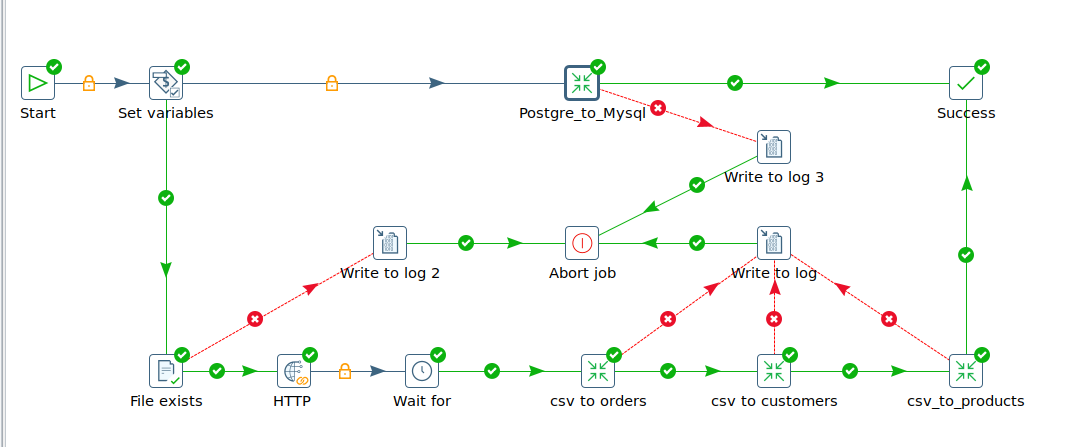


Рис. 21

Логгирование (Рис. 22)

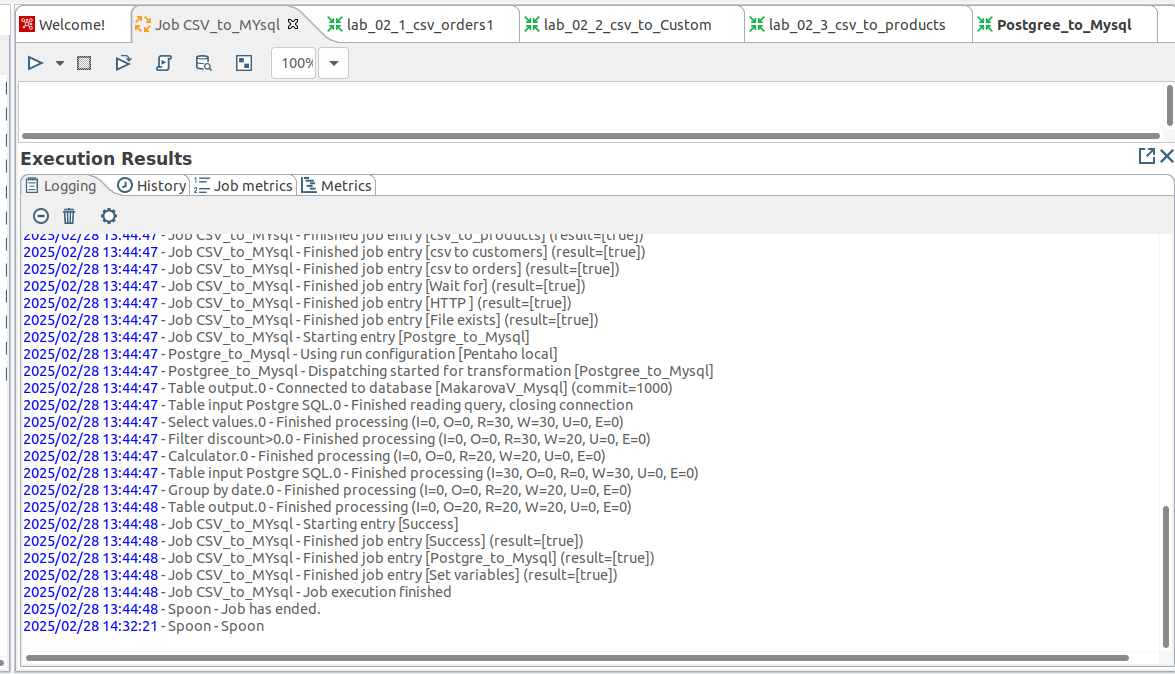


Рис. 22

Проверим загрузку данных в СУБД Mysql.

Данные таблицы **sales\_analytics** с рассчитанной выручкой по дням (Рис. 23).

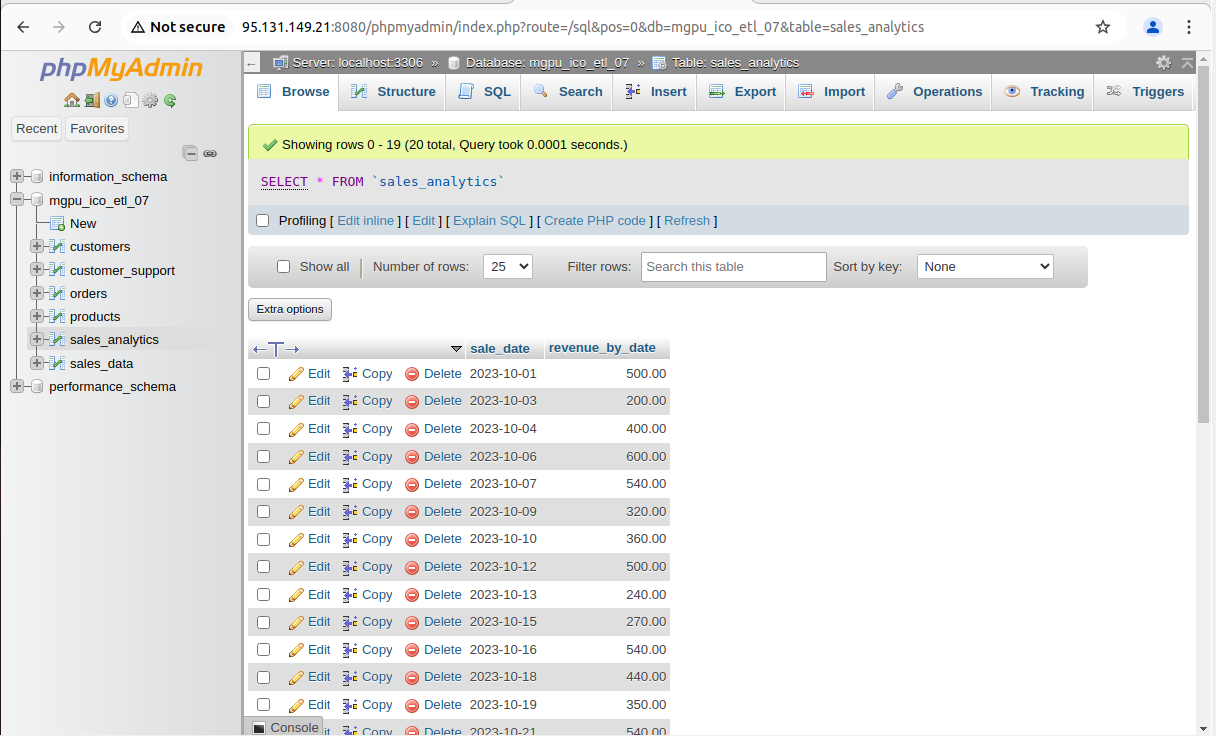


Рис. 23

Данные таблиц **orders, customers, products** (Рис. 24 – Рис. 26).

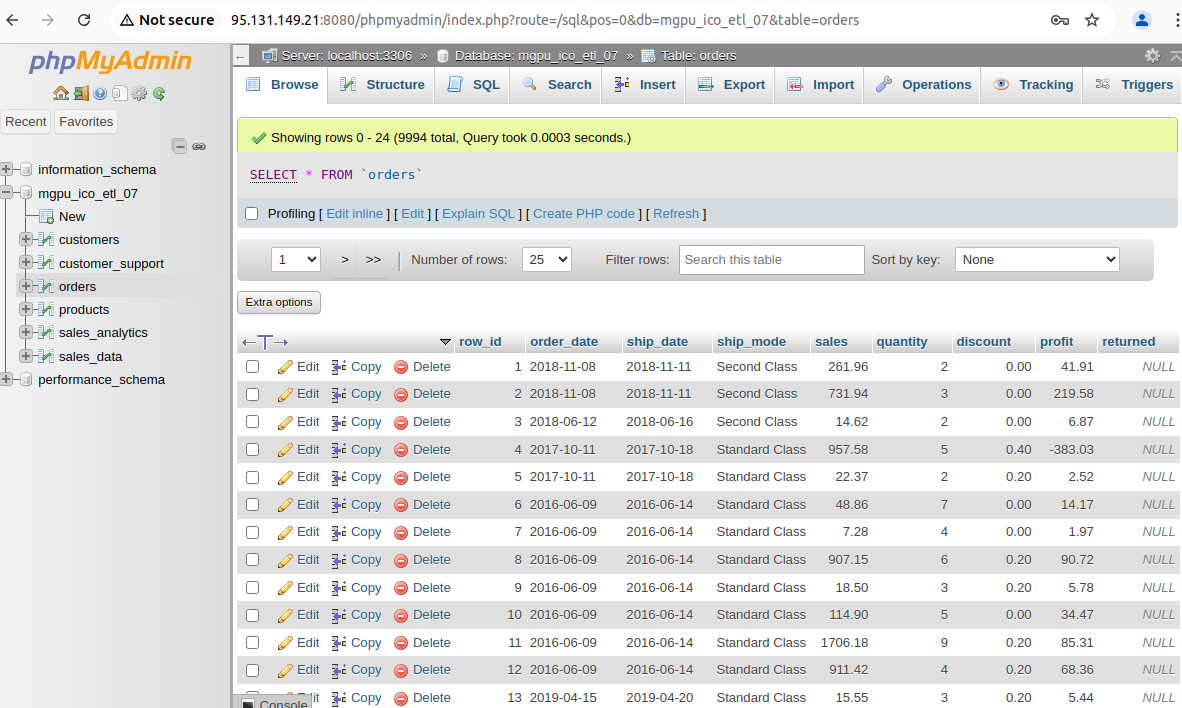


Рис. 24

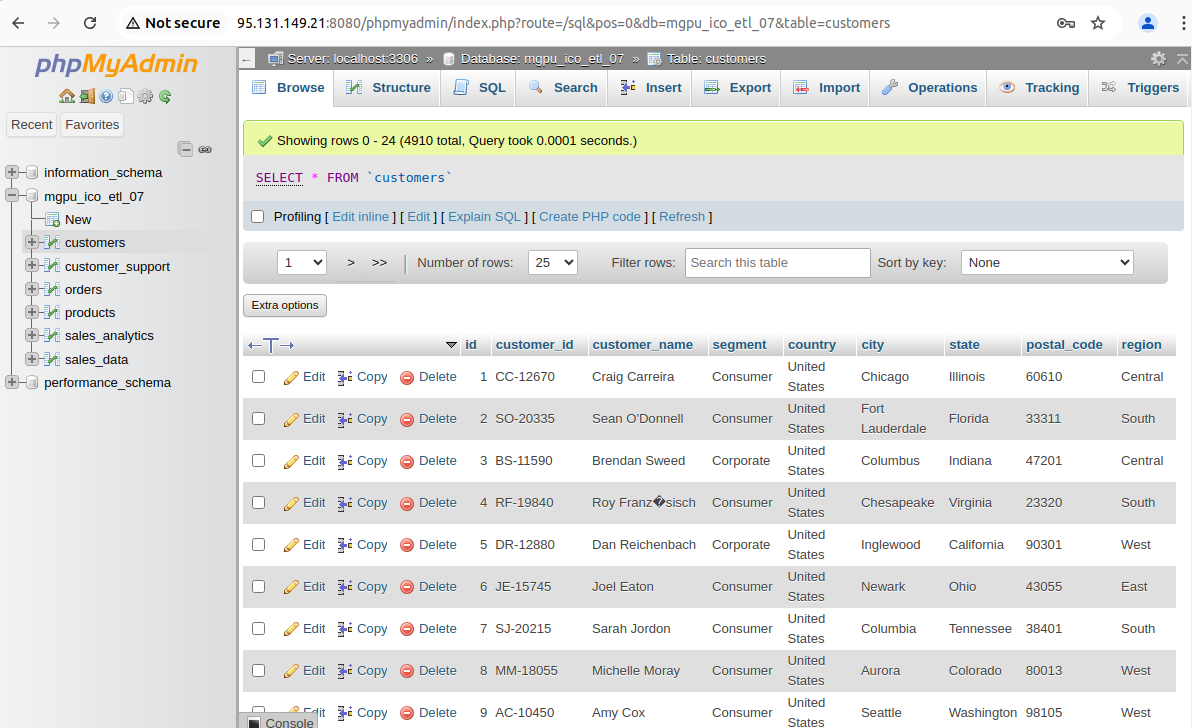


Рис. 25

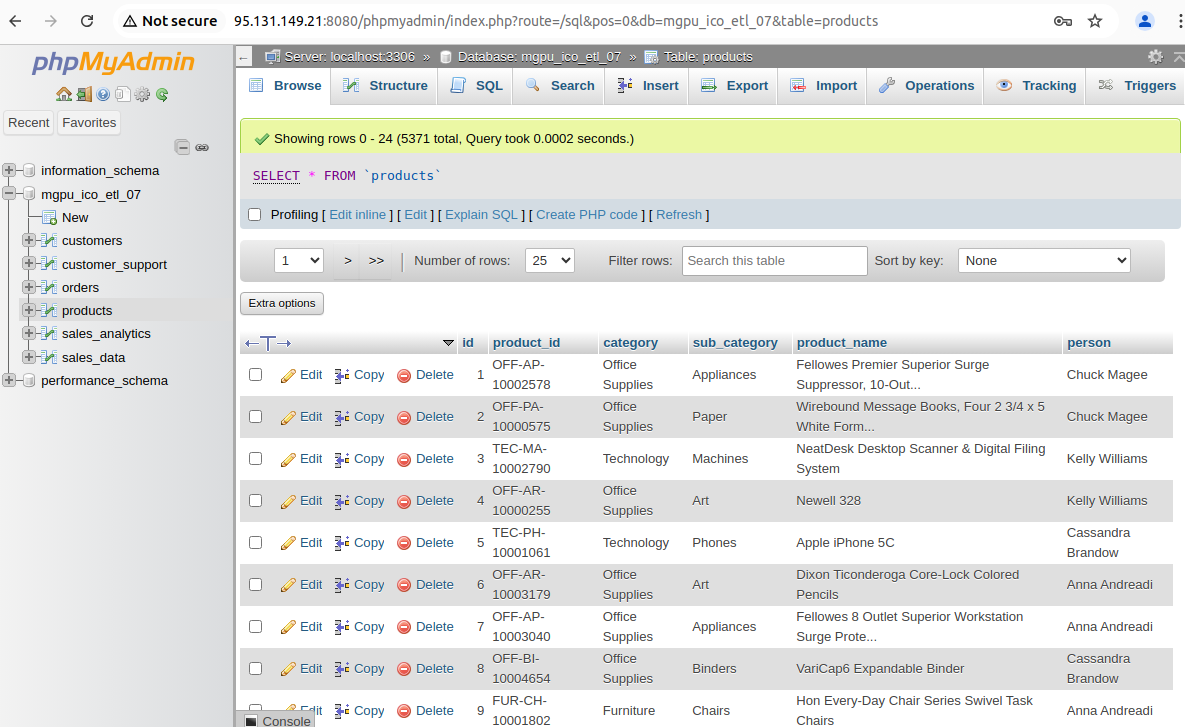


Рис. 26

**Вывод:** Поставленные задачи были выполнены, а именно:

− Создать исходные таблицы в PostgreSQL с различными наборами данных.

− Настроить целевые таблицы в MySQL для приема данных.

− Разработать процессы трансформации данных в Pentaho.

− Реализовать механизмы обработки ошибок и валидации данных.