Департамент образования города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»

Институт цифрового образования
Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

«Проектный практикум по разработке ETL-решений »

Самостоятельная работа №1

Тема: «Разработка ETL-процесса для интеграции данных между PostgreSQL и MySQL с использованием Pentaho Data Integration.»

Выполнила:

Студентка группы АДЭУ-211

Кравцова Алёна Евгеньевна

Руководитель:

Босенко Т.М

Москва

2024

Цель работы: интегрировать данные PostgreSQL и MySQL посредством Pentaho.

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;
– Учетная запись в Mysql.

Индивидуальное задание: Вариант 6.

Создать таблицу suppliers (id, company_name, contact_person, country, rating)	Создать таблицу verified_suppliers с полем verification_status	,Фильтр поставщиков с рейтингом выше 4	,Группировка по странам	,Проверка статуса верификации

Шаг 1. Проверка подключения к PostgresSQL

Остановим и заново запустим контейнер с PostgresSQL (Рис. 1).

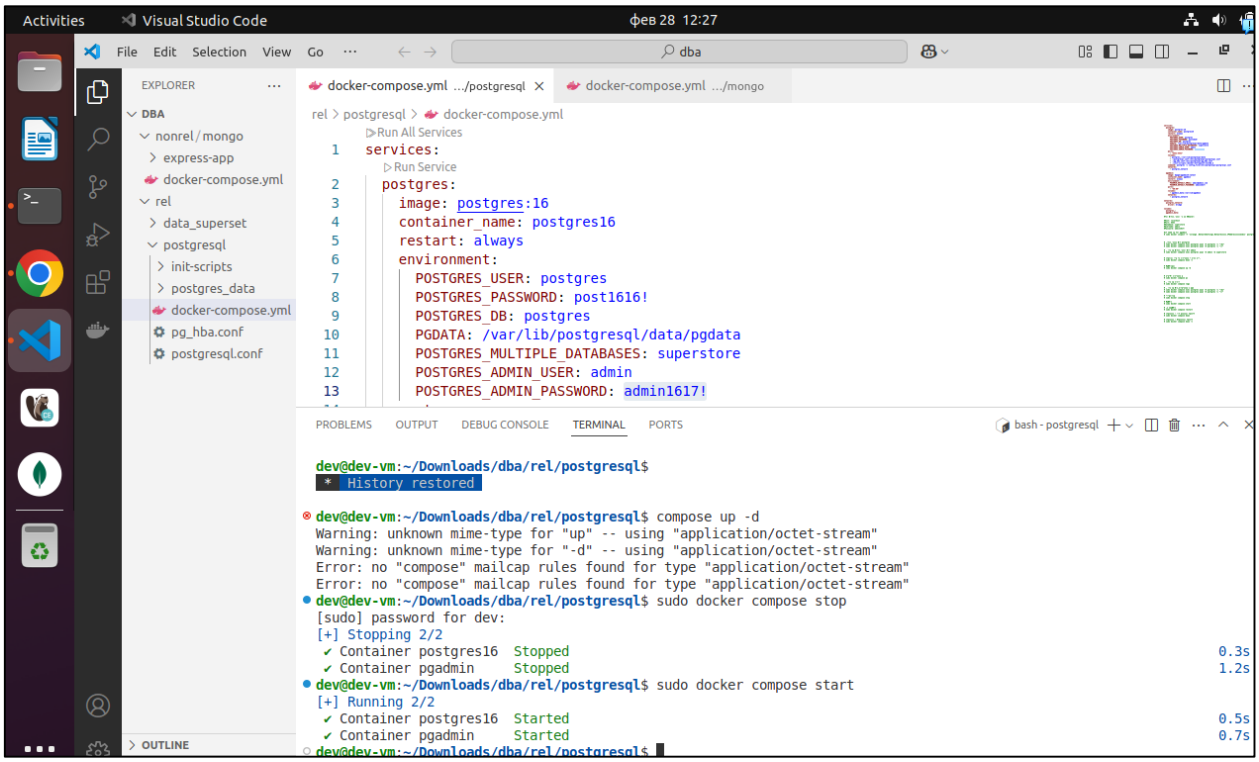

Рис. 1 – Остановка и повторный запуск PostgreSQL

Далее зайдем в pgAdmin и убедимся, что доступ к серверу есть (Рис. 2).

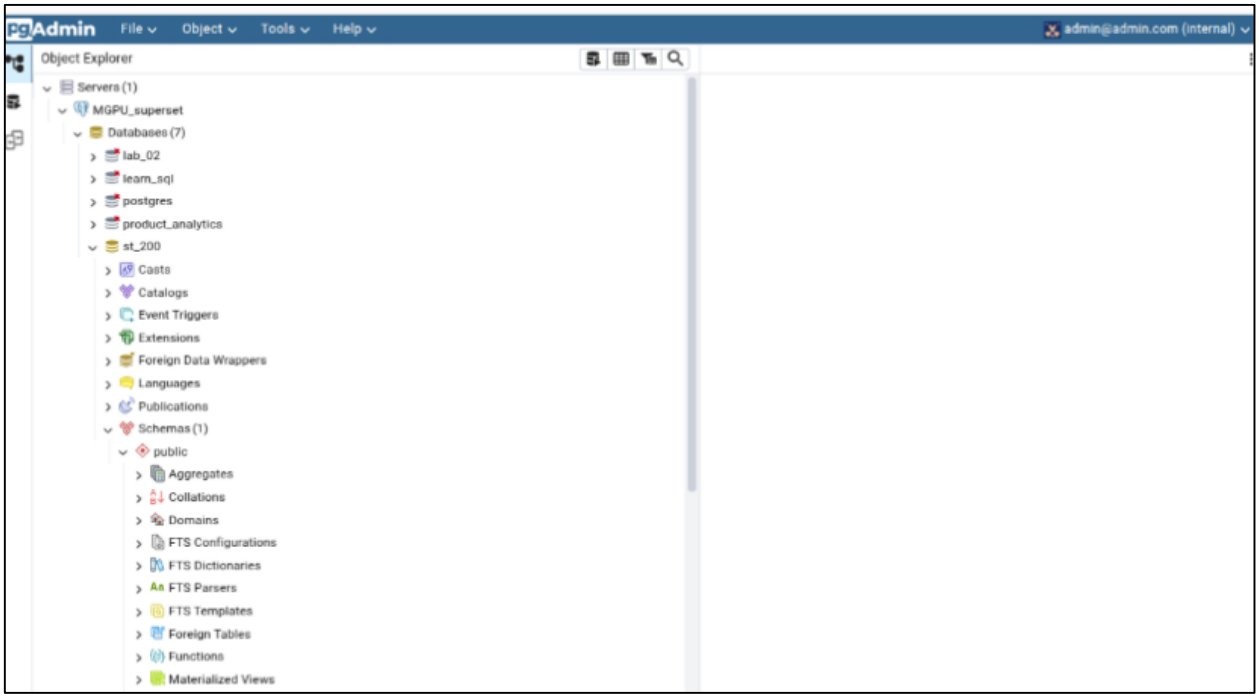

Рис. 2 – Успешное подключение к серверу

Создадим БД для дальнейшей работы (Рис. 3).

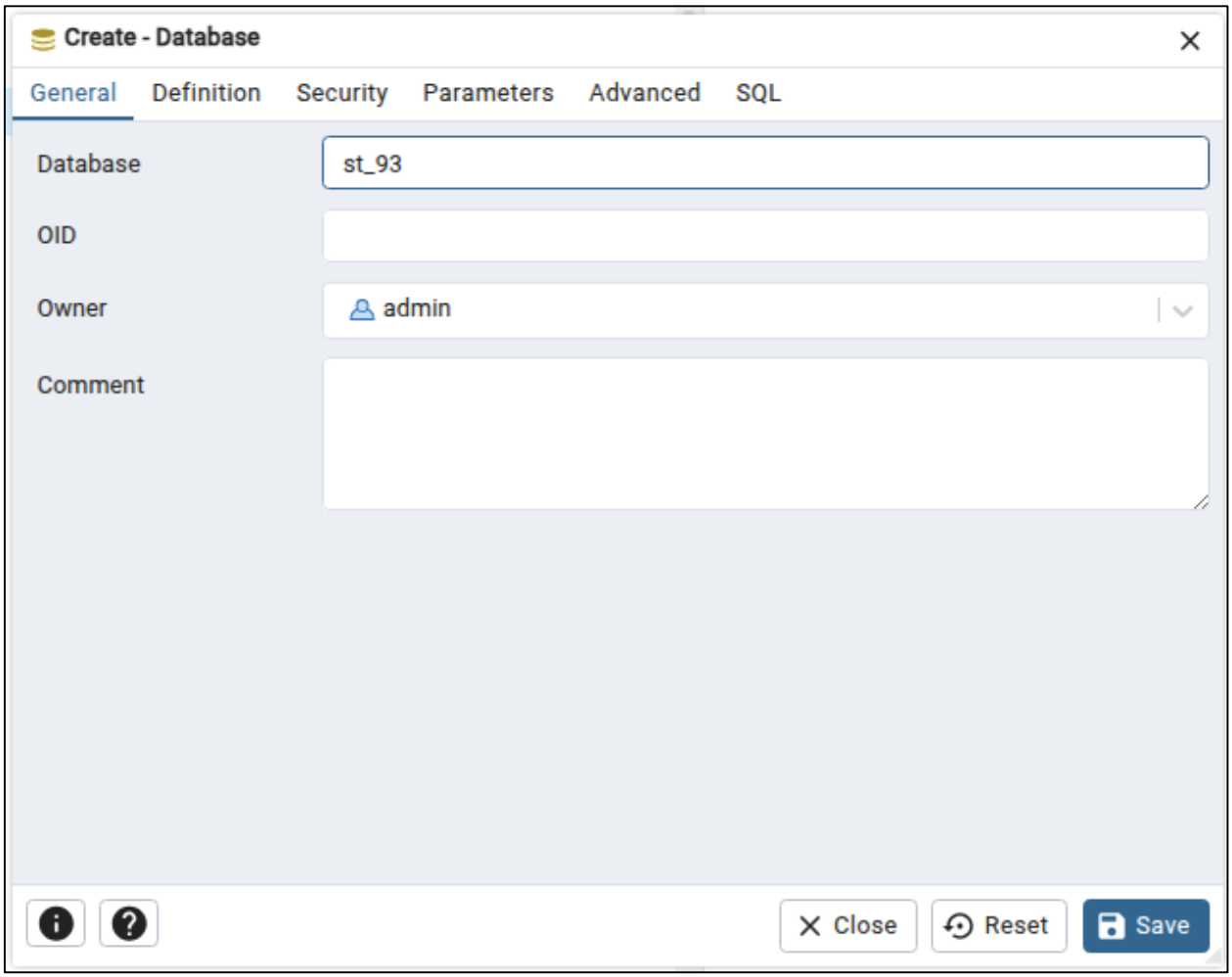

Рис. 3 – Создание БД

Далее необходимо создать таблицу таблицу suppliers (id, company_name, contact_person, country, rating) (Рис. 4).

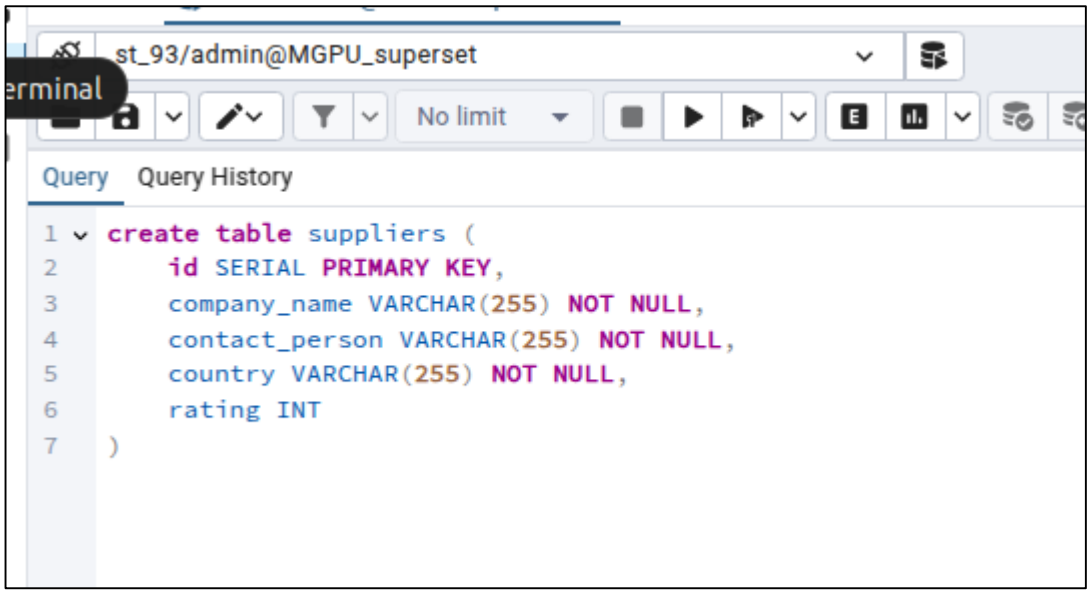

Рис. 4 – Скрипт для создания таблицы в pgAdmin

Таблица создана успешно (Рис. 5).

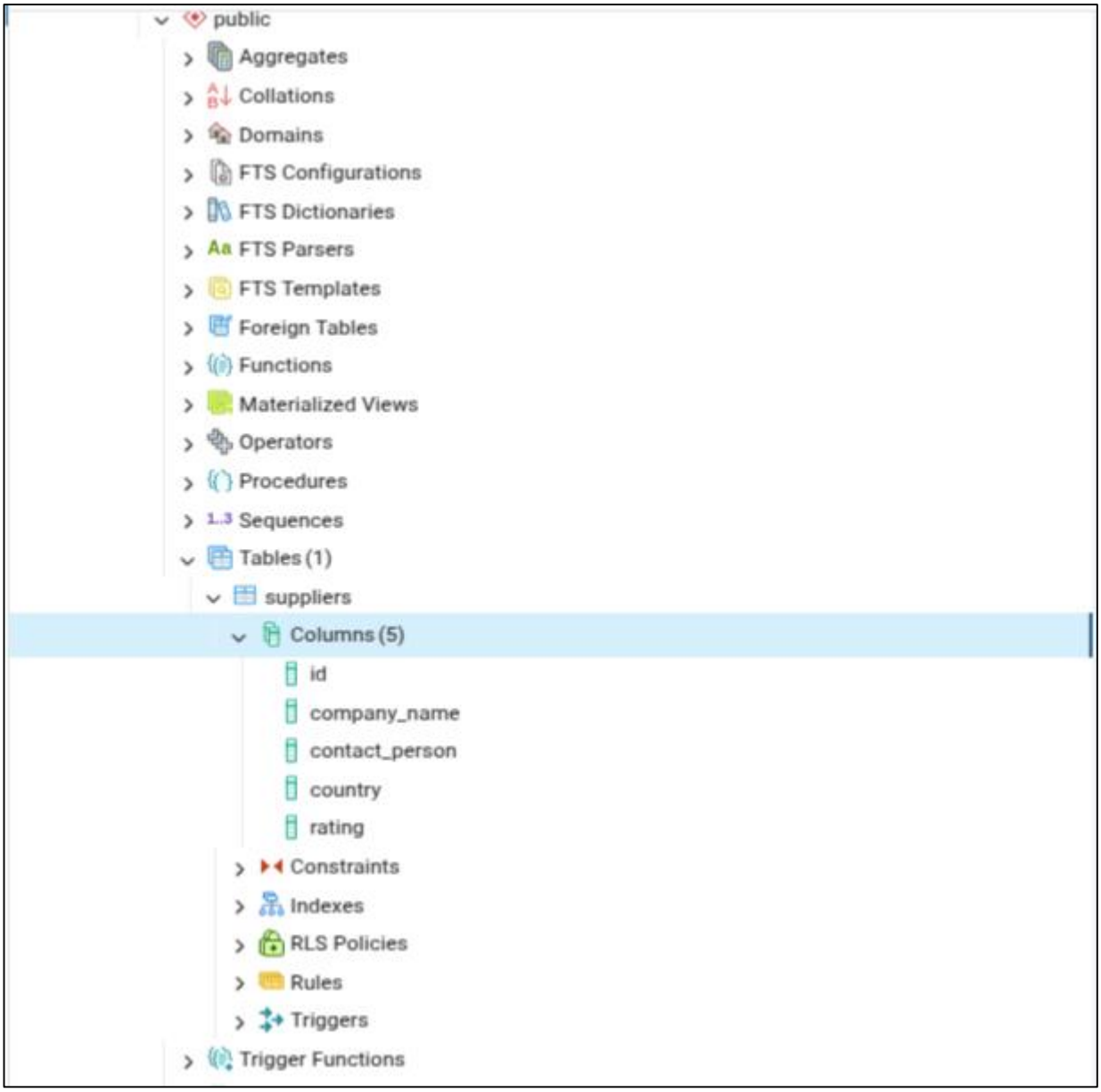

Рис. 5 – Созданная таблица в pgAdmin

Далее сделаем insert для загрузки сгенерированных данных (Рис. 6).

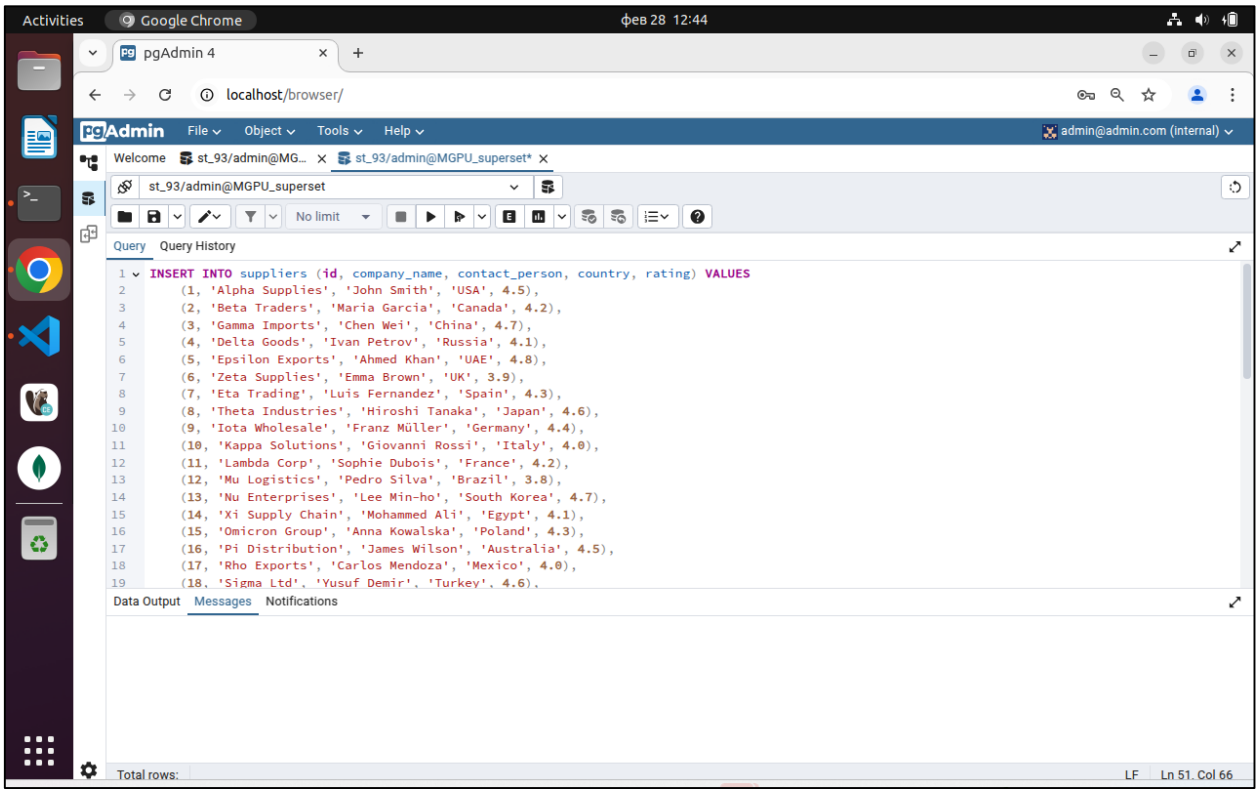

Рис. 6 – скрипт для загрузки данных

Данные успешно загружены (Рис. 7).

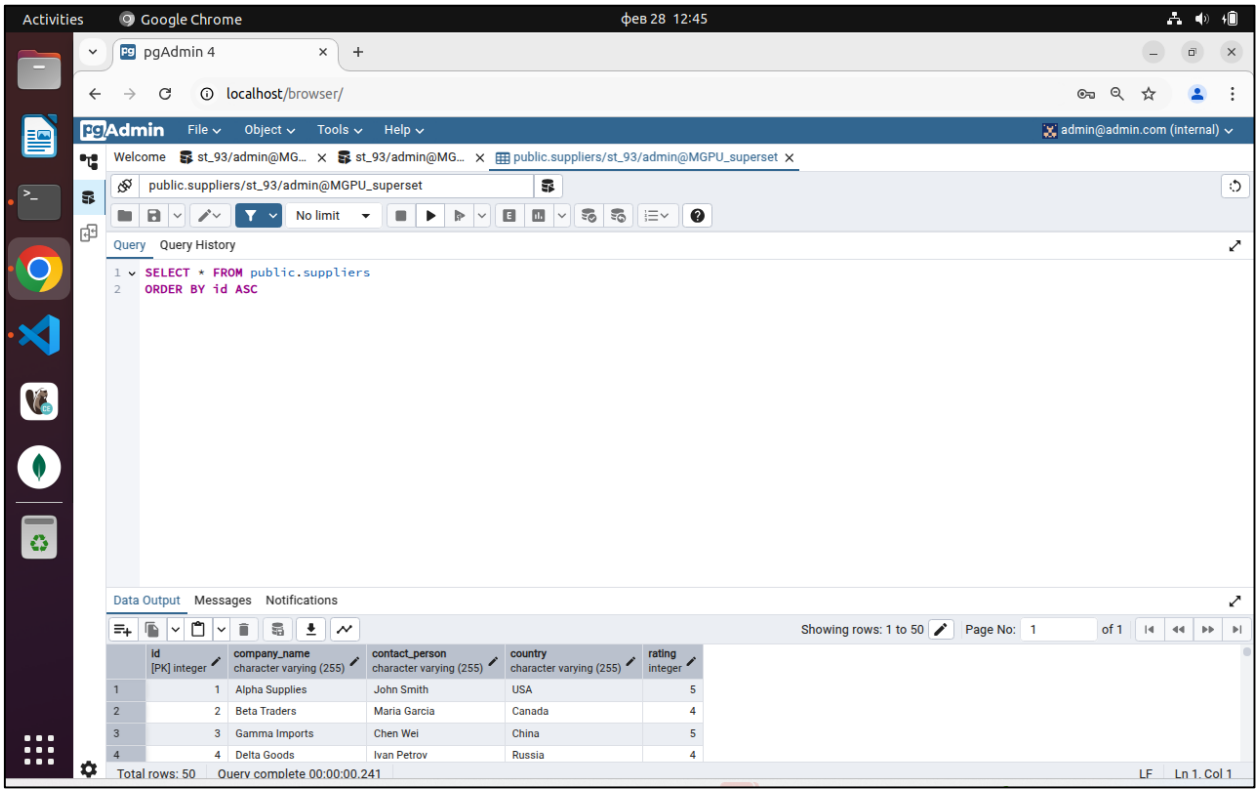

Рис. 7 – Результат загрузки данных

Далее перейдем к phpMyAdmin, доступ к целевой БД есть (Рис. 8).

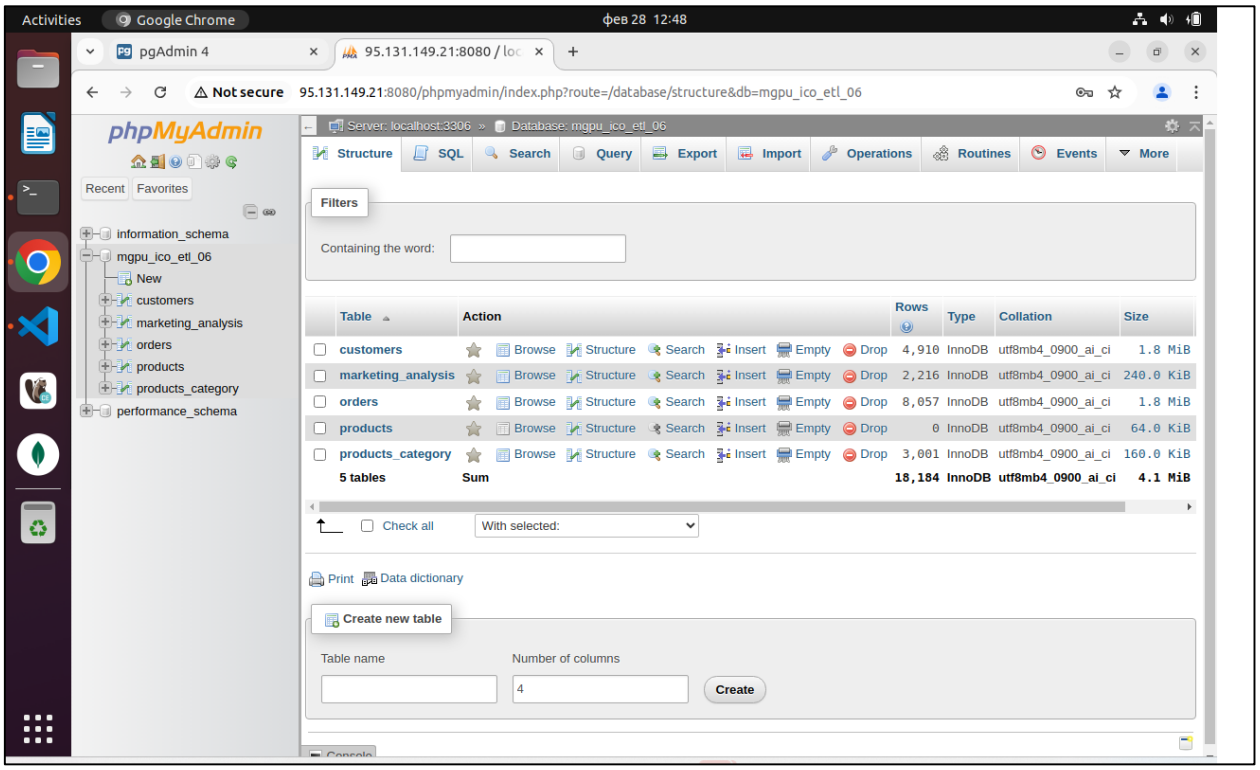

Рис. 8 – Подключение к mgpu_ico_etl_06

Создадим таблицу там в соответствии с заданием и добавим поле verification_status (Рис. 9).

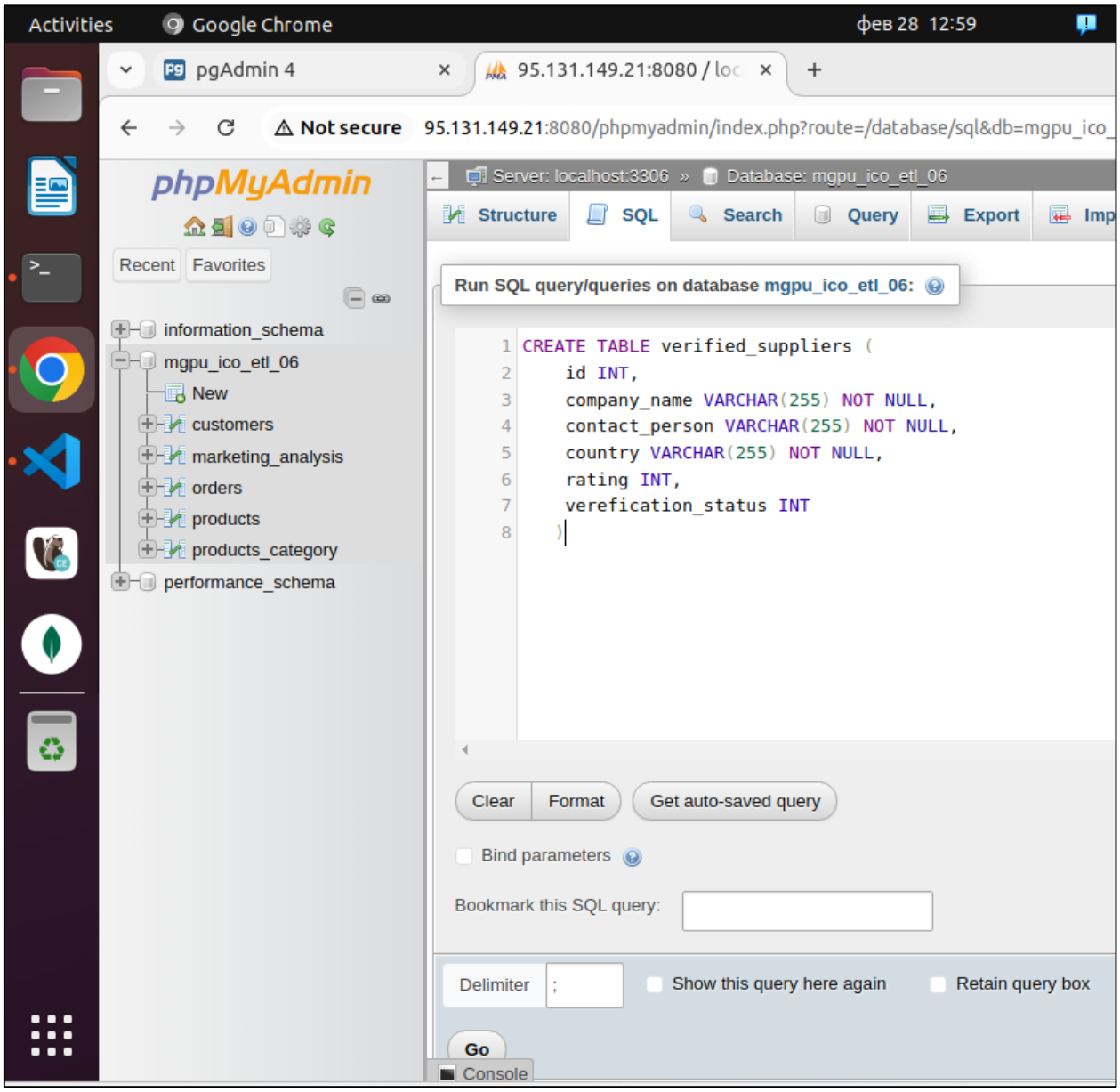

Рис. 9 – Скрипт для создания таблицы в phpMyAdmin

Таблица успешно создана (Рис. 10).

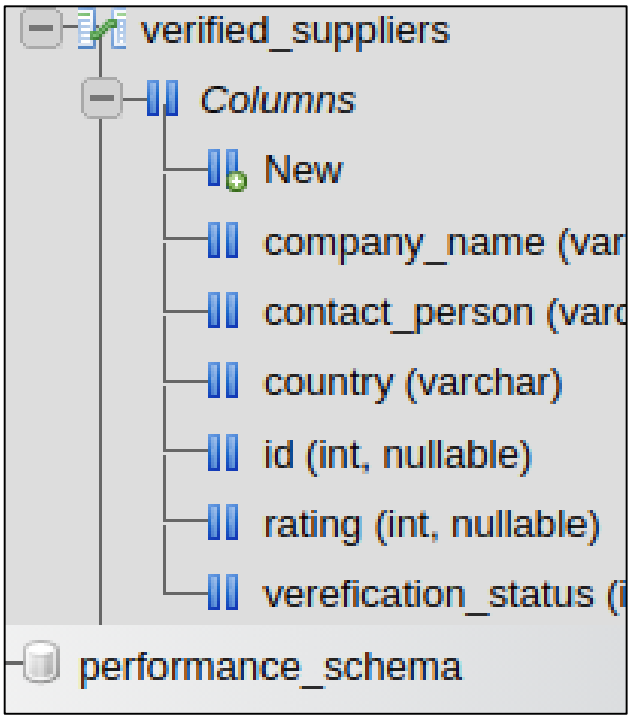

Рис. 10 – Созданная таблица в phpMyAdmin

Далее перейдем в pentaho и создадим новую трансформацию (Рис. 11)

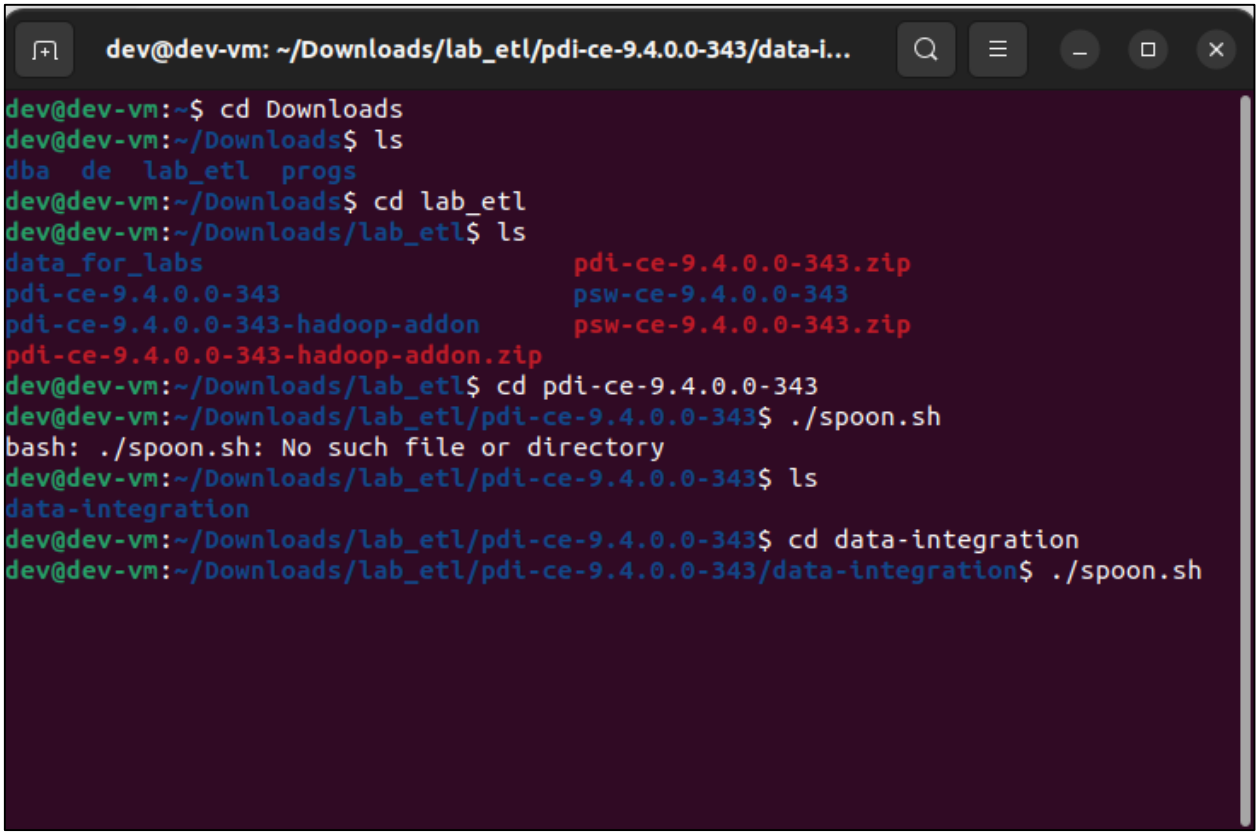

Рис. 11 –Запуск pentaho

Для выполнения задания 3,4,5 в трансформации будет реализованы следующие компоненты:

– Загрузка данных из postgresSql;

– Select rows;

– Проверка верификации пользователя (если рейтинг – 0, то пользователь неверефицирован);

– Фильтр по рейтингу выше 4;

– Загрузка данных в MySQL.

Подключимся к PostgreSQL для получения данных из созданной ранее таблицы (Рис. 12, Рис. 13).

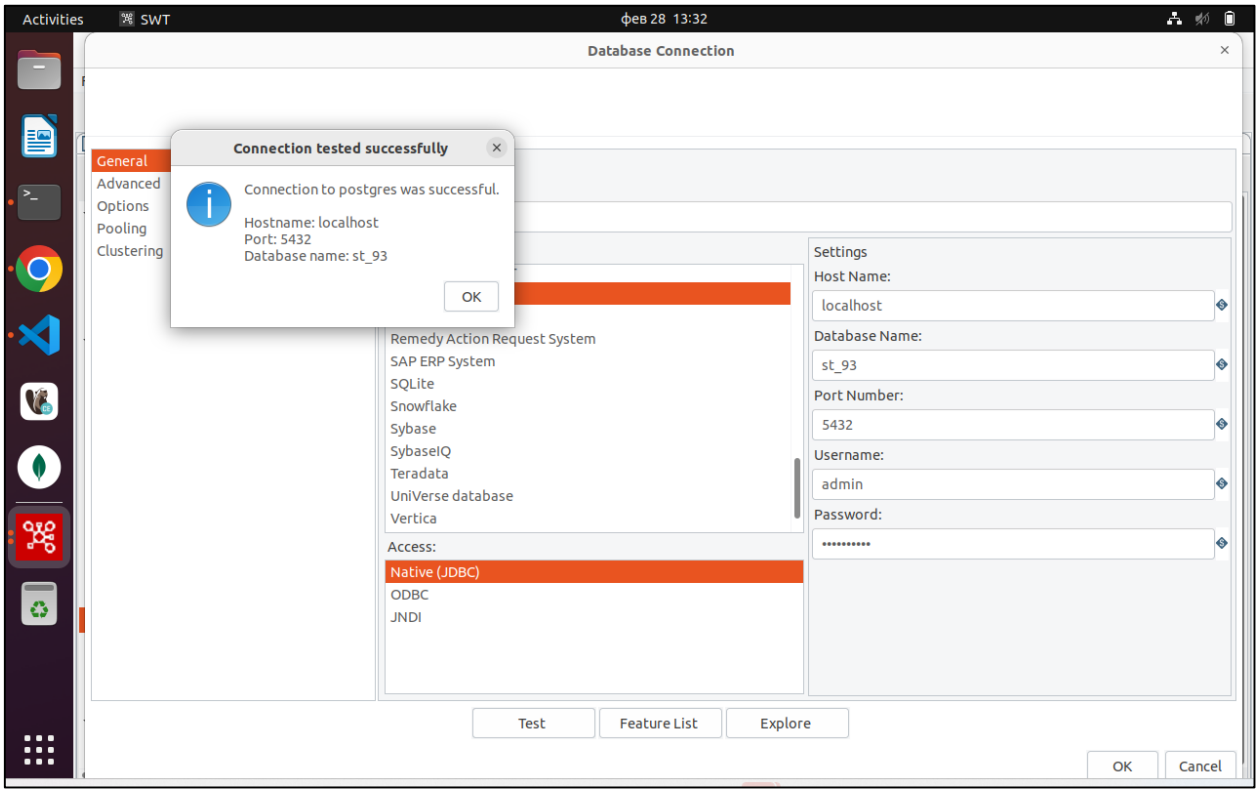

Рис. 12 – Подключение к PostgreSQL

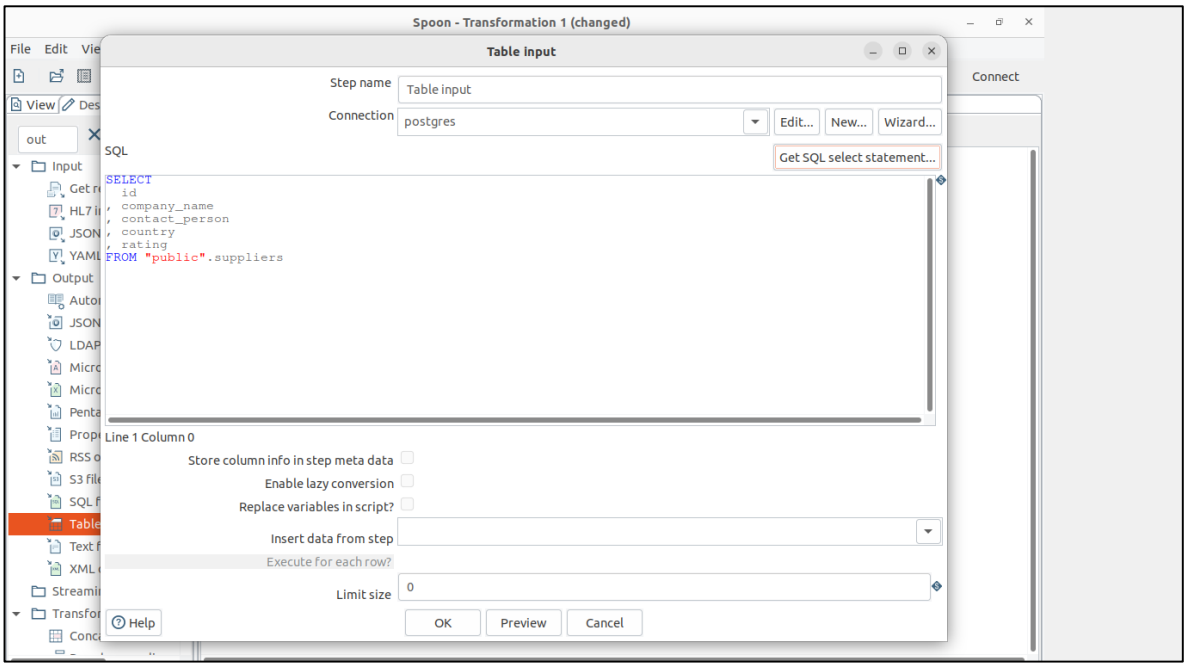

Рис. 13 – Импорт данных из PostgreSQL

Далее отфильтруем рейтинг выше 0 (Рис. 14).

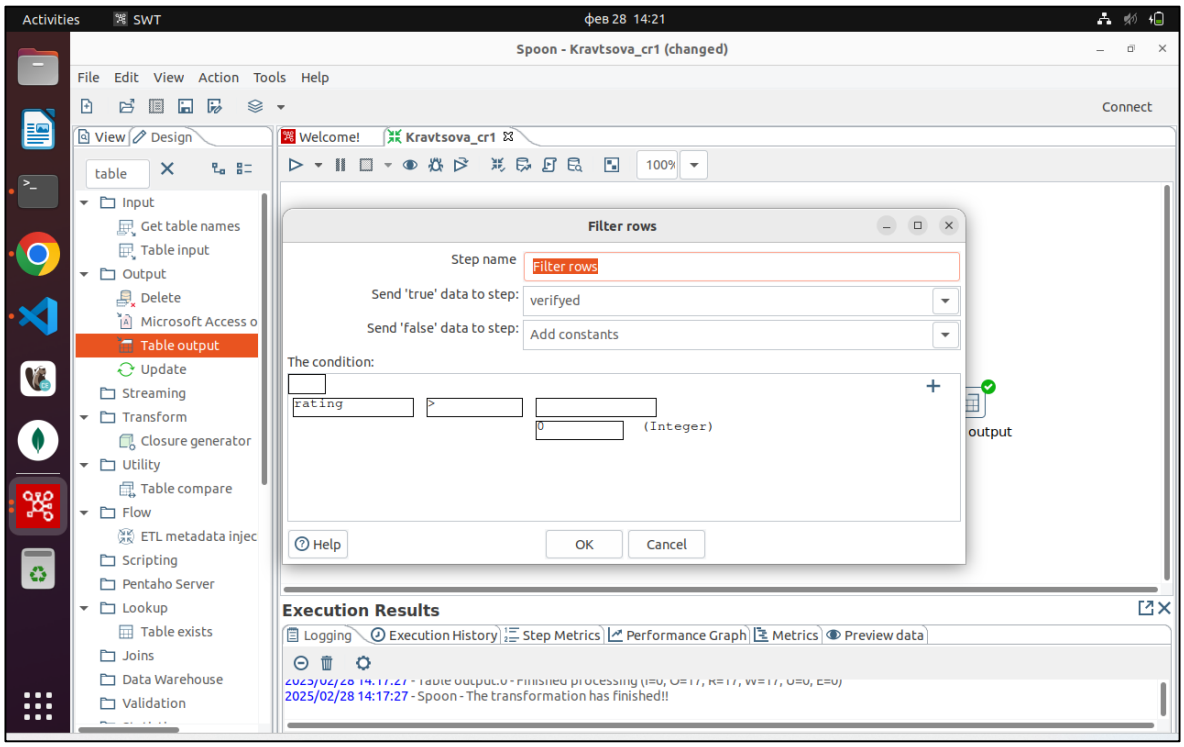

Рис. 14 – Фильтр по рейтингу

Далее необходимо определить статус верификации. Если рейтинг 0, то статус верификации 0 (неверефицирован) (Рис. 15).

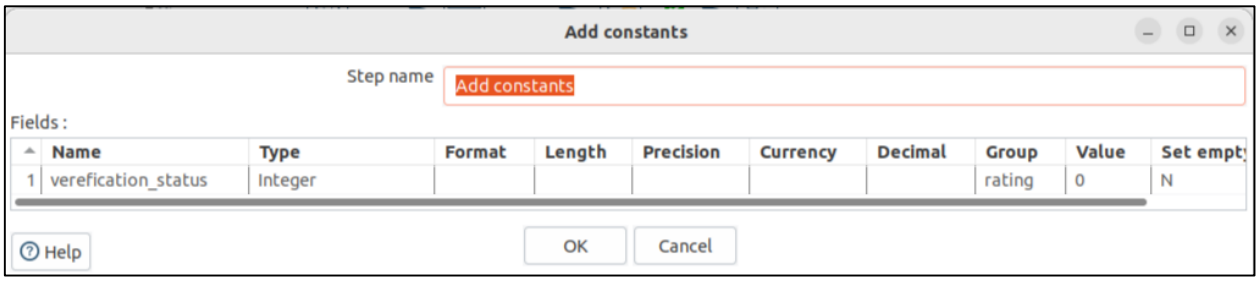

Рис. 15 – Значения не верефицированного пользователя

Если рейтинг больше 0, то статус верификации 1 (верифицирован) (Рис. 16).

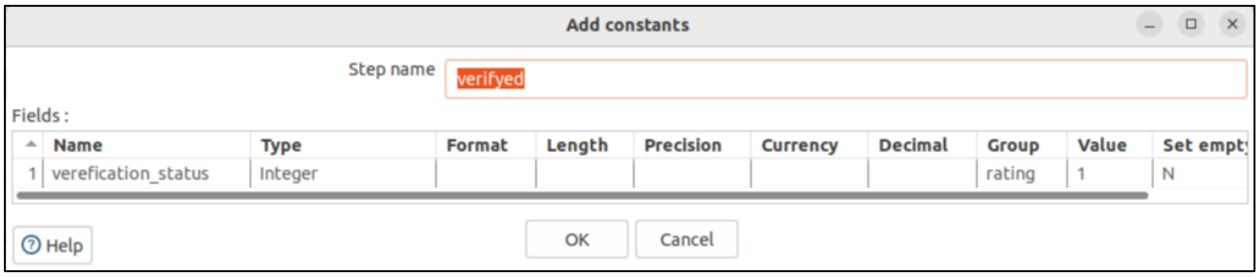

Рис. 16 – Значение верифицированного пользователя

Компонент *append streams* соединит значения, также посмотрим *preview data*, чтобы убедиться, что статус проставляется корректно (Рис. 17). Исходя из рисунка 17 видно, что пользователю с рейтингом 0 был проставлен статус не верифицированного пользователя.

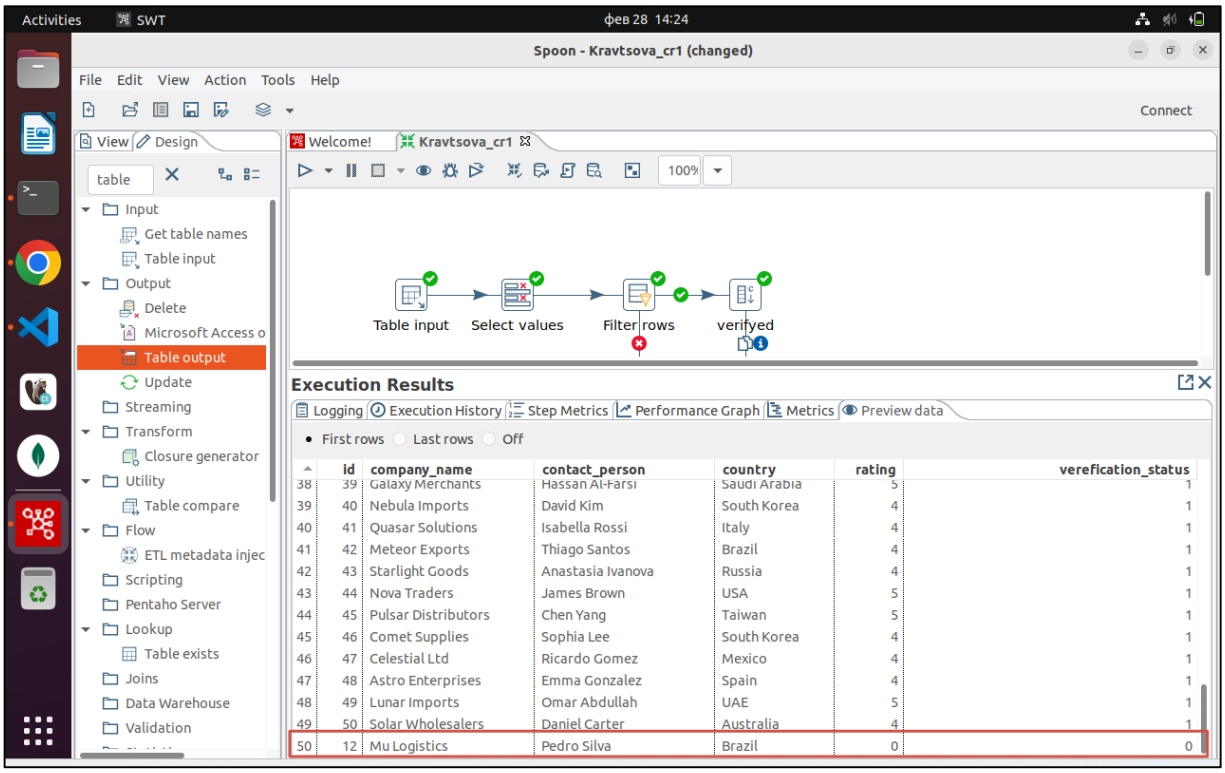

Рис. 17 – Проставленные статусы верификации

Далее по заданию отфильтруем записи, у которых рейтинг выше 4 – именно они и будут загружаться в хранилище (Рис. 18).

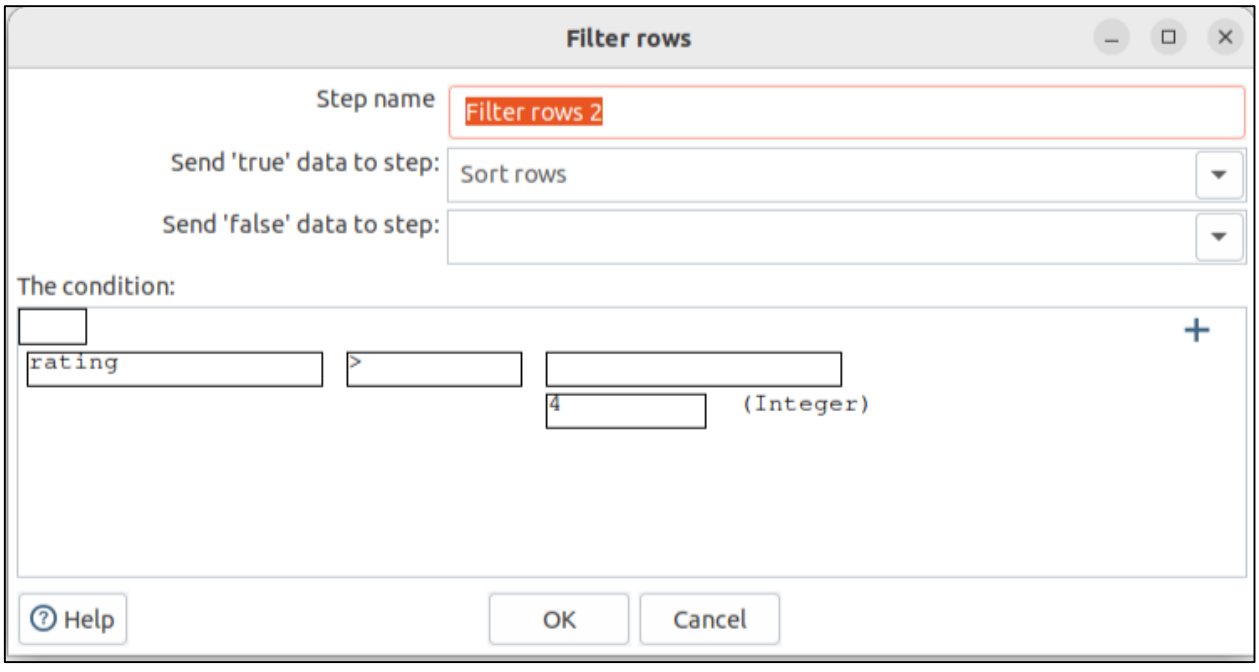

Рис. 18 – Фильтр по рейтингу

Далее подключимся к БД для загрузки данных (Рис. 19).

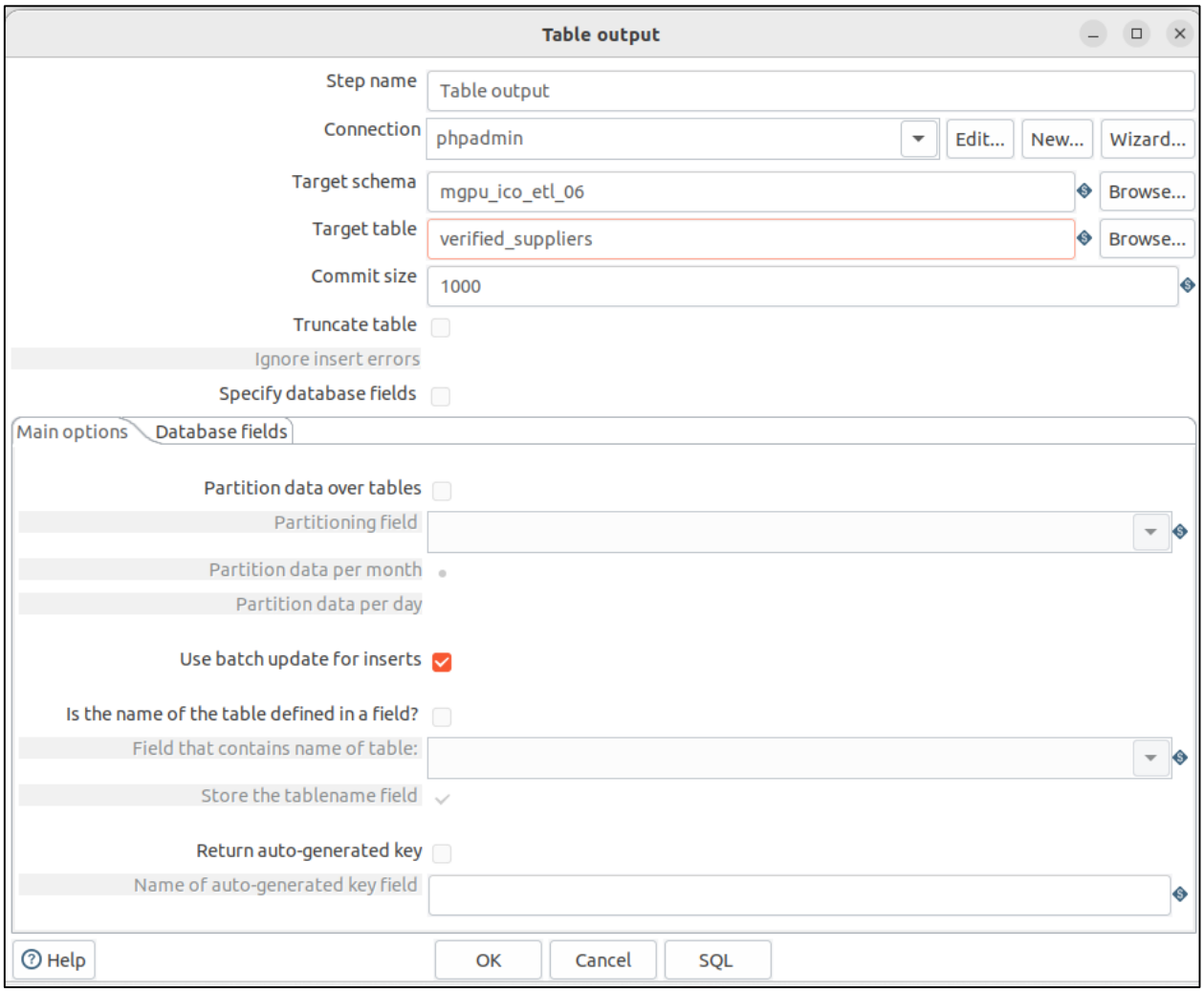

Рис. 19 – Подключение к базе MySQL

Запустим трансформацию (Рис. 20).

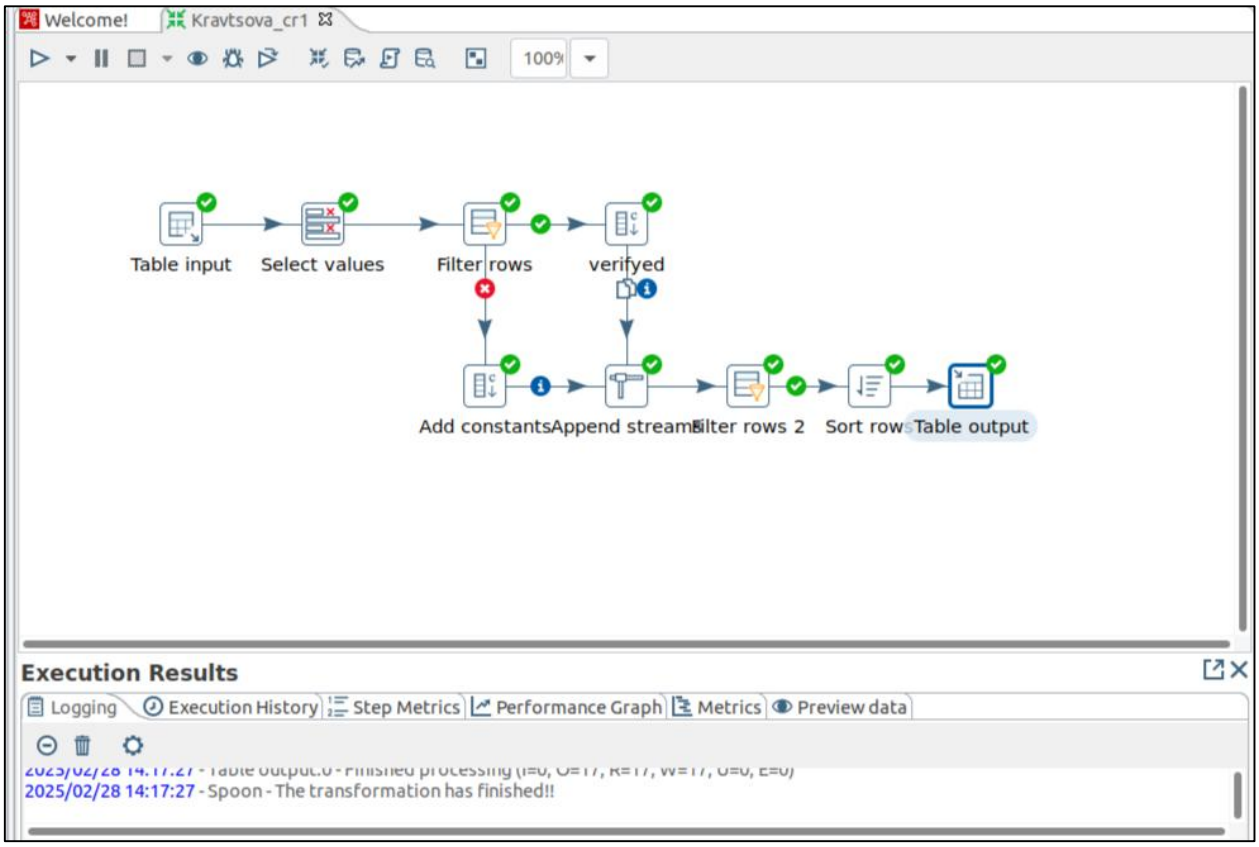

Рис. 20 – Успешное выполнение трансформации

Трансформация выполнена успешно, все данные корректно загружены в хранилище (Рис. 21).

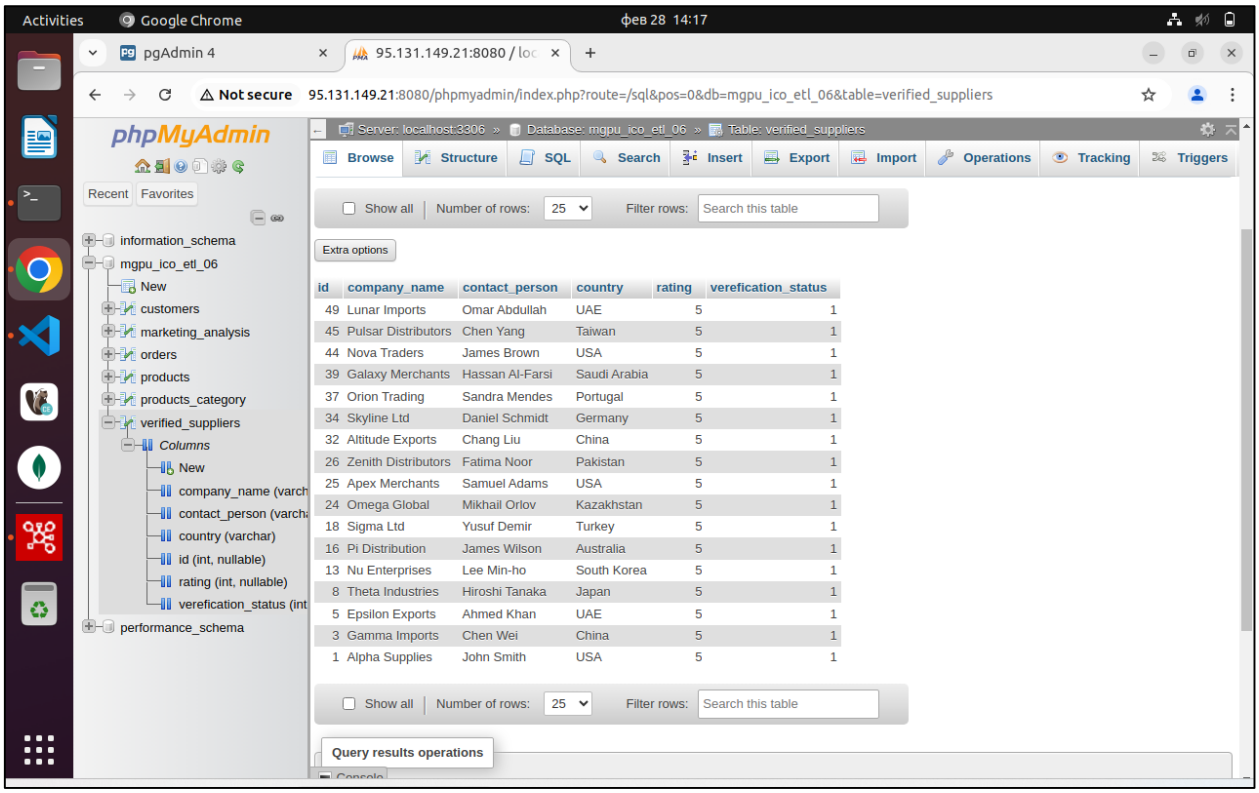

Рис. 21 – Успешная загрузка данных

Напишем запрос, который посчитает количество компаний в каждой стране (группировка по стране) (Рис. 22, Рис.23).

Рис. 22 – Запрос на количество компаний в каждой стране

country	company_count
UAE	2
Taiwan	1
USA	3
Saudi Arabia	1
Portugal	1
Germany	1
China	2
Pakistan	1
Kazakhstan	1
Turkey	1
Australia	1
South Korea	1
Japan	1

Рис. 23 –Результат запроса

Вывод: в итоге выполнения работы был получен опыт работы с разными базами данных, а том числе и с PostgreSQL. В рамках одной трансформации были задействованы разные БД, что дает четкое понимание о том, как правильно подключать и выгружать данные.