

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 PostgreSQL

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

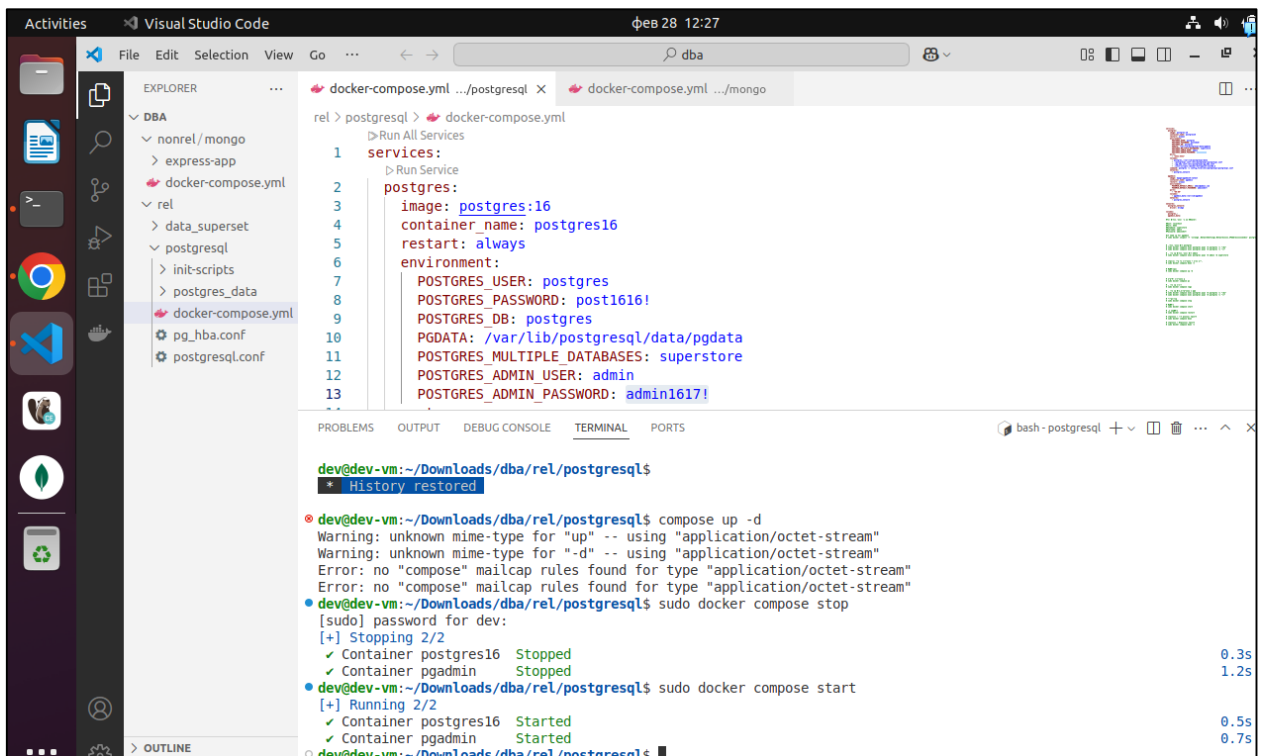


Рис. 1 – Остановка и повторный запуск PostgreSQL

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

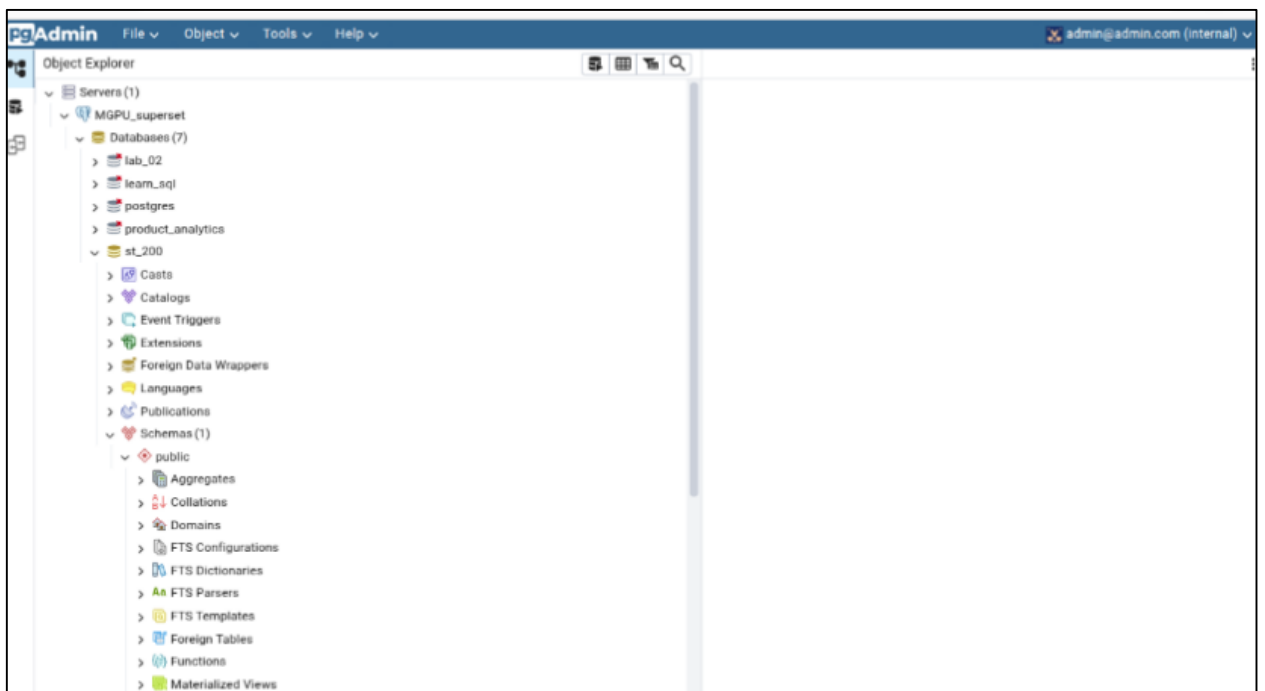


Рис. 2 – Успешное подключение к серверу

Создадим БД для дальнейшей работы (Рис. 3).

Create - Database

General Definition Security Parameters Advanced SQL

Database: st_93

OID:

Owner: admin

Comment:

Close Reset Save

Рис. 3 – Создание БД

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

```
1 create table suppliers (  
2     id SERIAL PRIMARY KEY,  
3     company_name VARCHAR(255) NOT NULL,  
4     contact_person VARCHAR(255) NOT NULL,  
5     country VARCHAR(255) NOT NULL,  
6     rating INT  
7 )
```

Рис. 4 – Скрипт для создания таблицы в pgAdmin

Таблица создана успешно (Рис. 5).

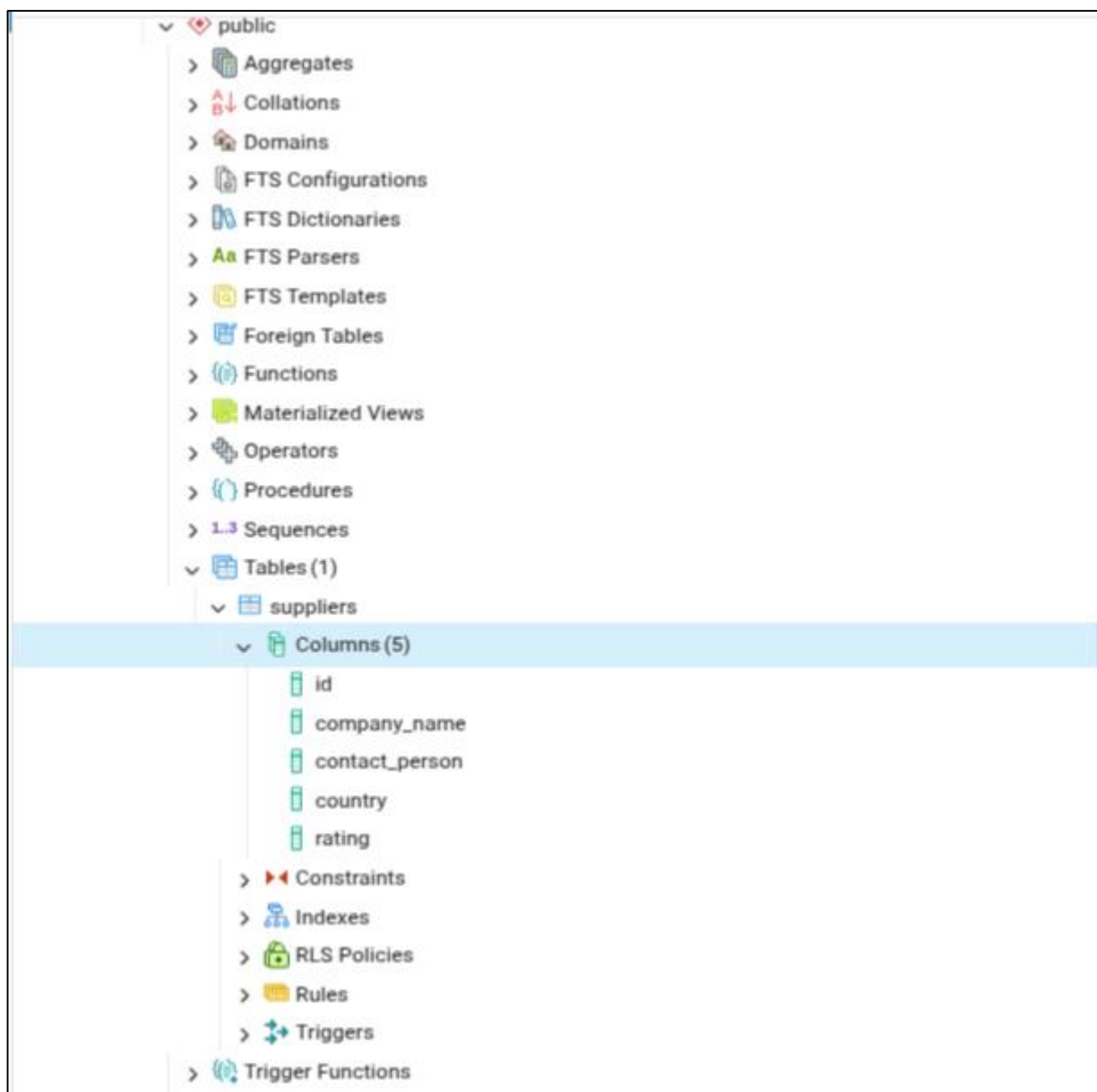


Рис. 5 – Созданная таблица в pgAdmin

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

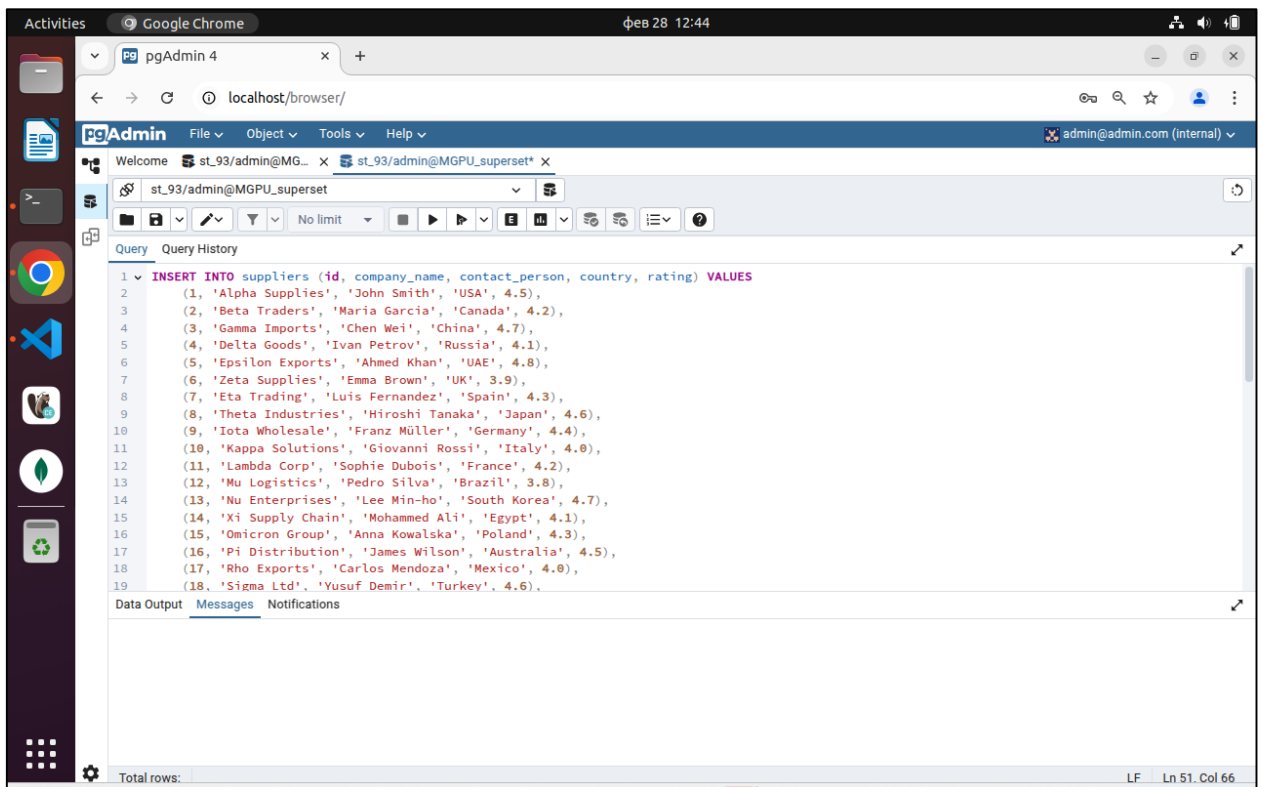


Рис. 6 – скрипт для загрузки данных

Данные успешно загружены (Рис. 7).

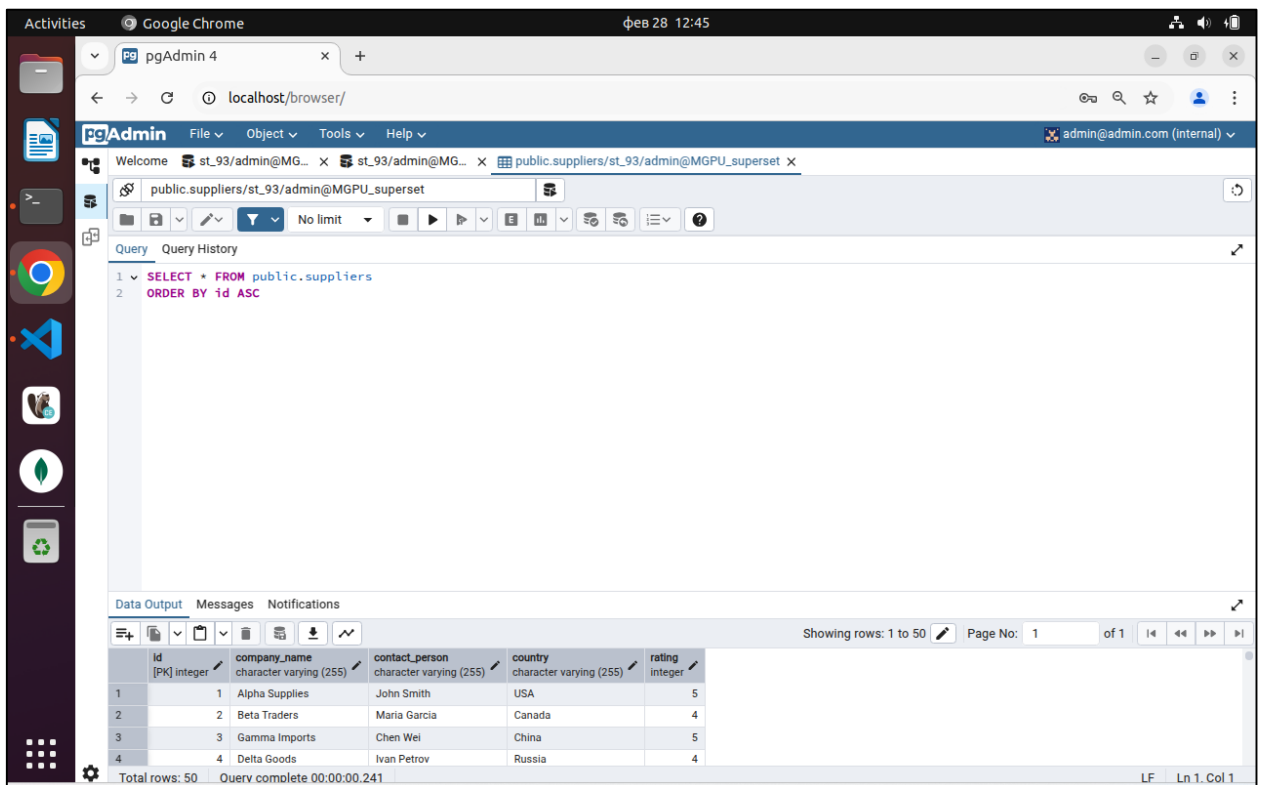


Рис. 7 – Результат загрузки данных

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

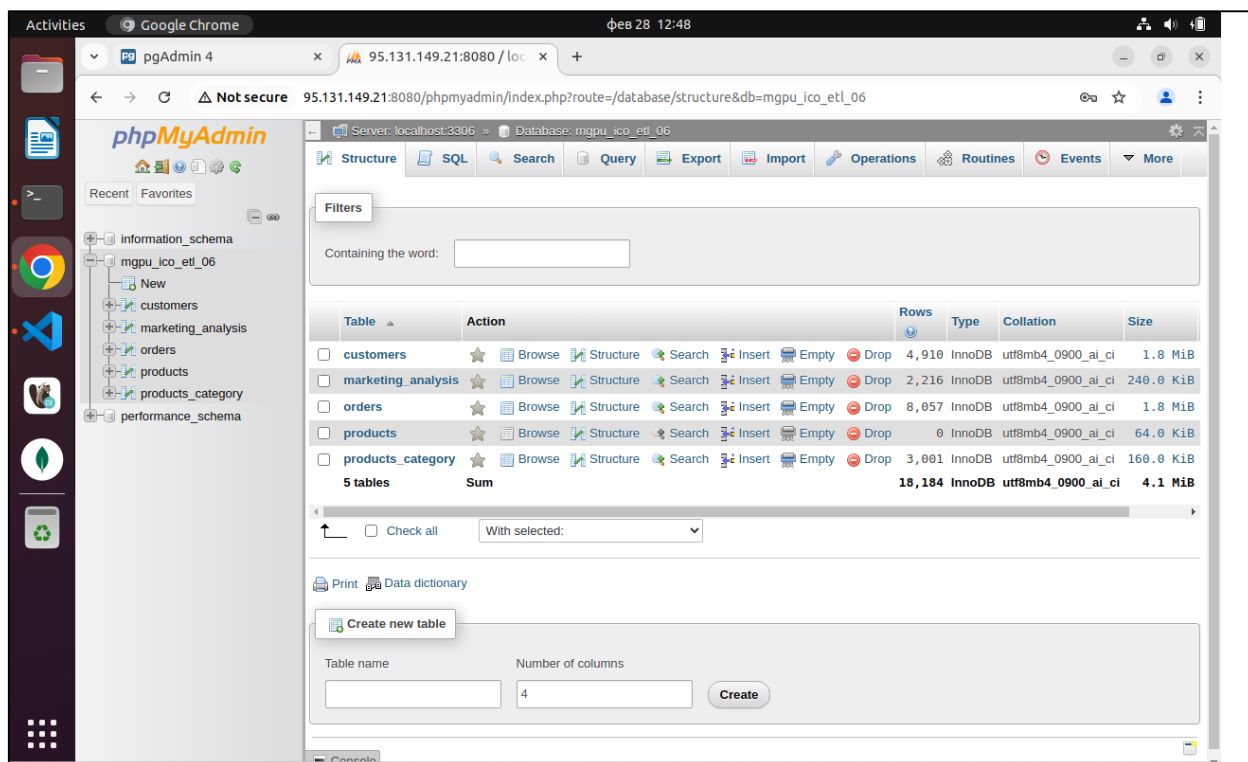


Рис. 8 – Подключение к mgpu_ico_etl_06

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

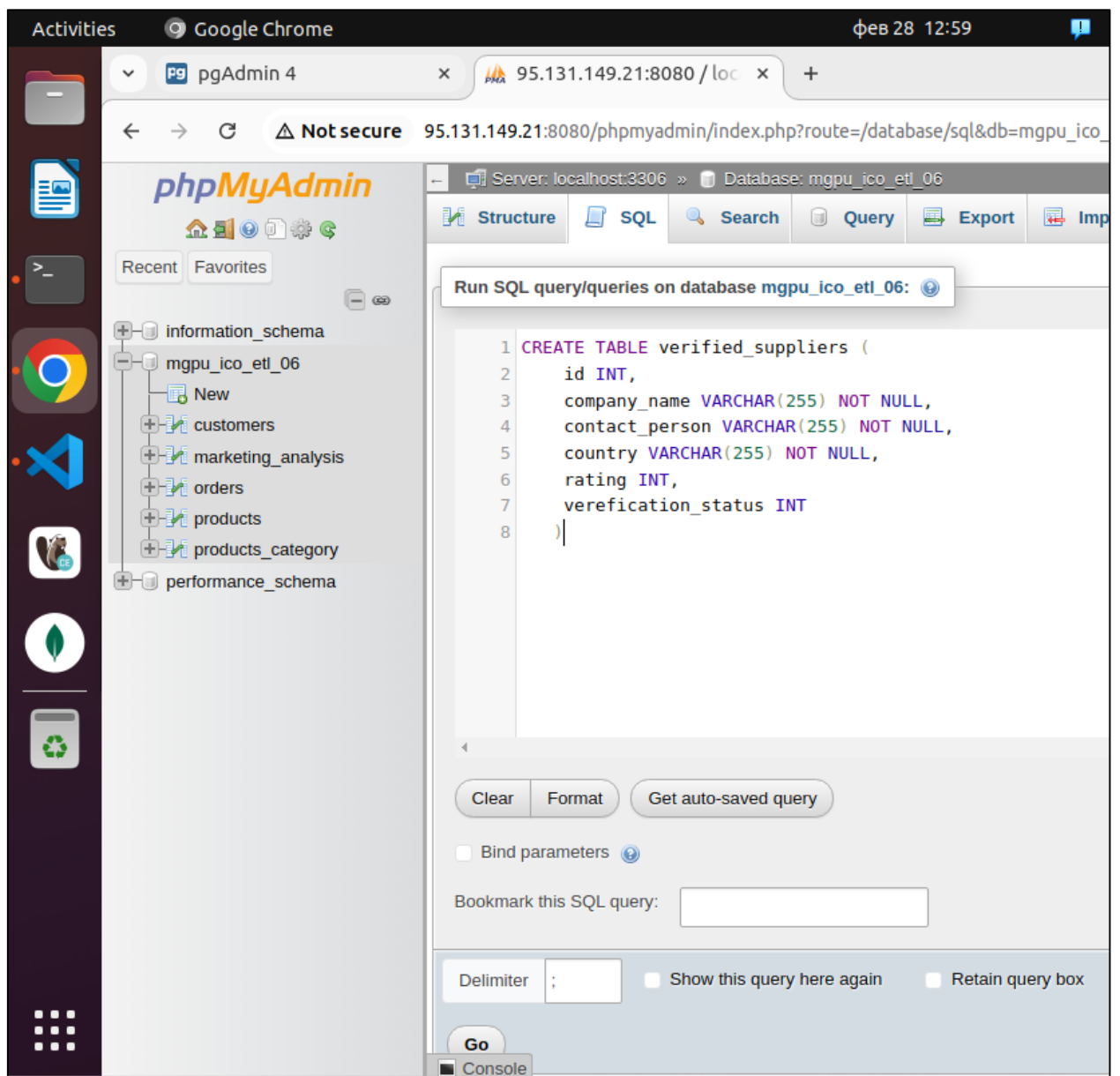


Рис. 9 – Скрипт для создания таблицы в phpMyAdmin

Таблица успешно создана (Рис. 10).

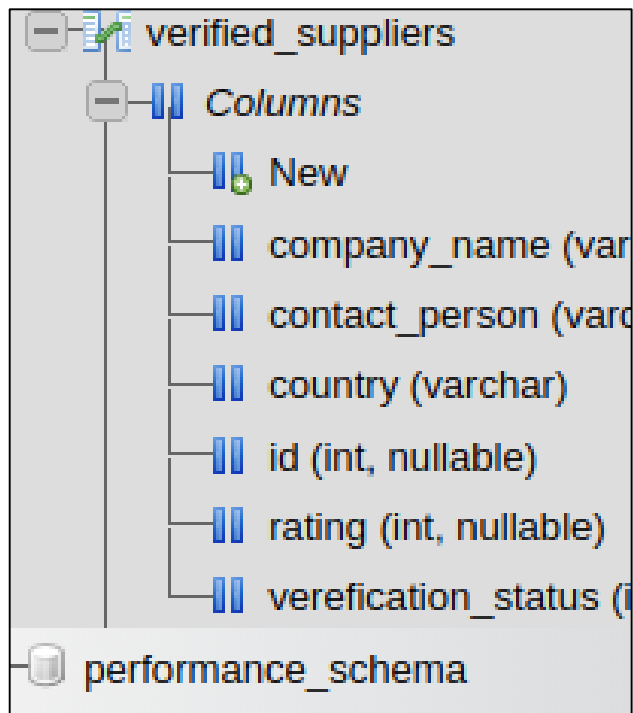


Рис. 10 – Созданная таблица в phpMyAdmin

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

```
dev@dev-vm: ~/Downloads/lab_etl/pdi-ce-9.4.0.0-343/data-i...
dev@dev-vm:~$ cd Downloads
dev@dev-vm:~/Downloads$ ls
dba  de  lab_etl  progs
dev@dev-vm:~/Downloads$ cd lab_etl
dev@dev-vm:~/Downloads/lab_etl$ ls
data_for_labs          pdi-ce-9.4.0.0-343.zip
pdi-ce-9.4.0.0-343     psw-ce-9.4.0.0-343
pdi-ce-9.4.0.0-343-hadoop-addon  psw-ce-9.4.0.0-343.zip
pdi-ce-9.4.0.0-343-hadoop-addon.zip
dev@dev-vm:~/Downloads/lab_etl$ cd pdi-ce-9.4.0.0-343
dev@dev-vm:~/Downloads/lab_etl/pdi-ce-9.4.0.0-343$ ./spoon.sh
bash: ./spoon.sh: No such file or directory
dev@dev-vm:~/Downloads/lab_etl/pdi-ce-9.4.0.0-343$ ls
data-integration
dev@dev-vm:~/Downloads/lab_etl/pdi-ce-9.4.0.0-343$ cd data-integration
dev@dev-vm:~/Downloads/lab_etl/pdi-ce-9.4.0.0-343/data-integration$ ./spoon.sh
```

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 неverified);
- Фильтр по рейтингу выше 4;
- Загрузка данных в MySQL.

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

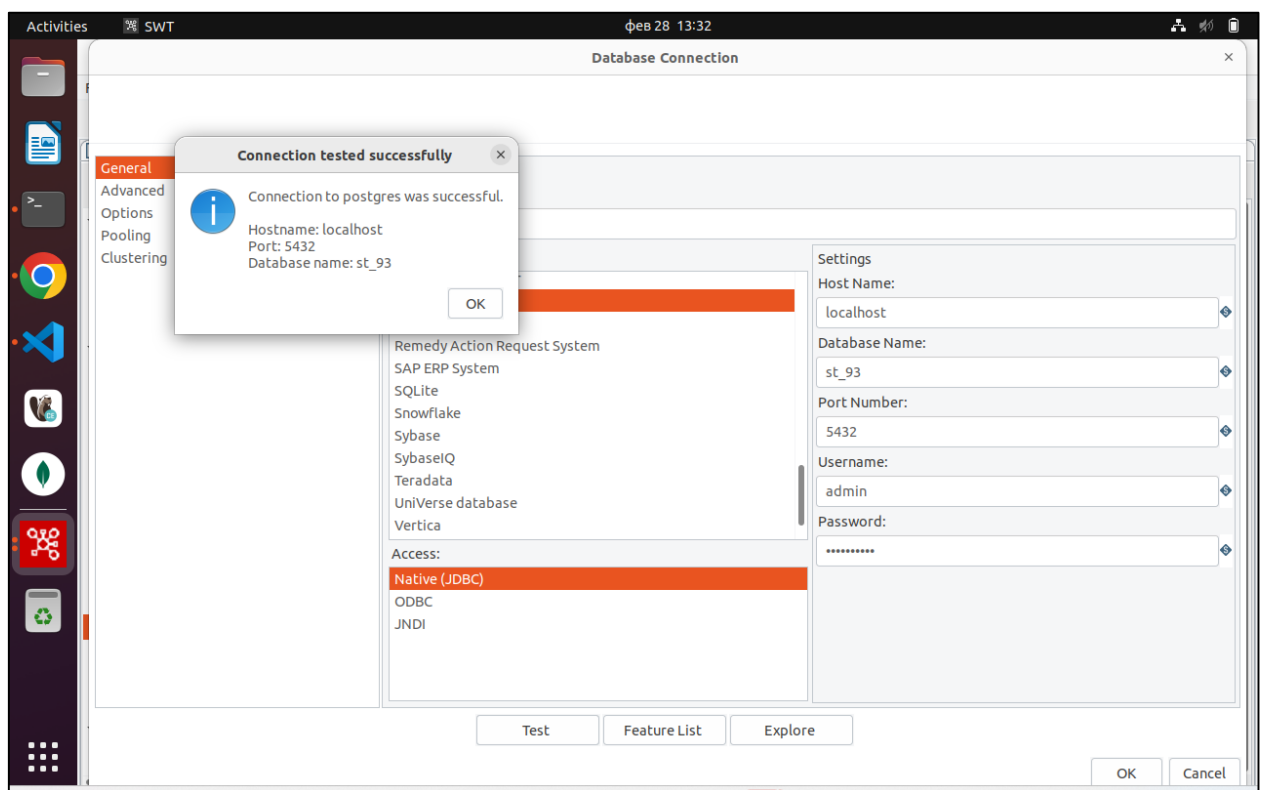


Рис. 12 – Подключение к PostgreSQL

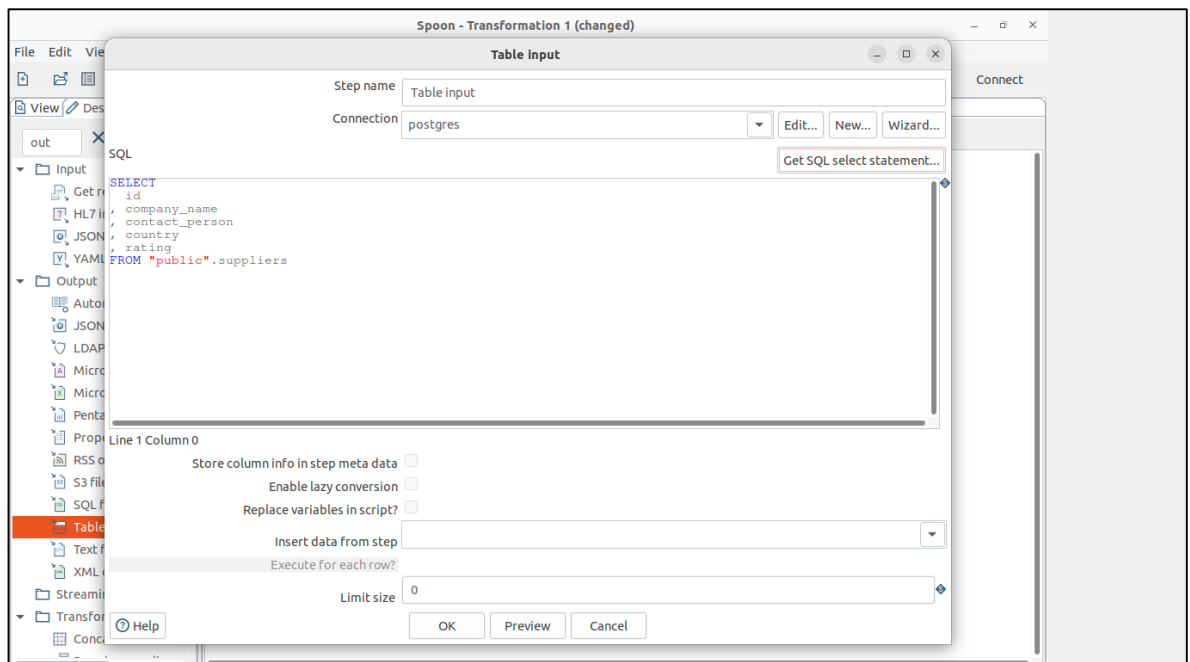


Рис. 13 – Импорт данных из PostgreSQL

Далее отфильтруем рейтинг выше 0 (Рис. 14).

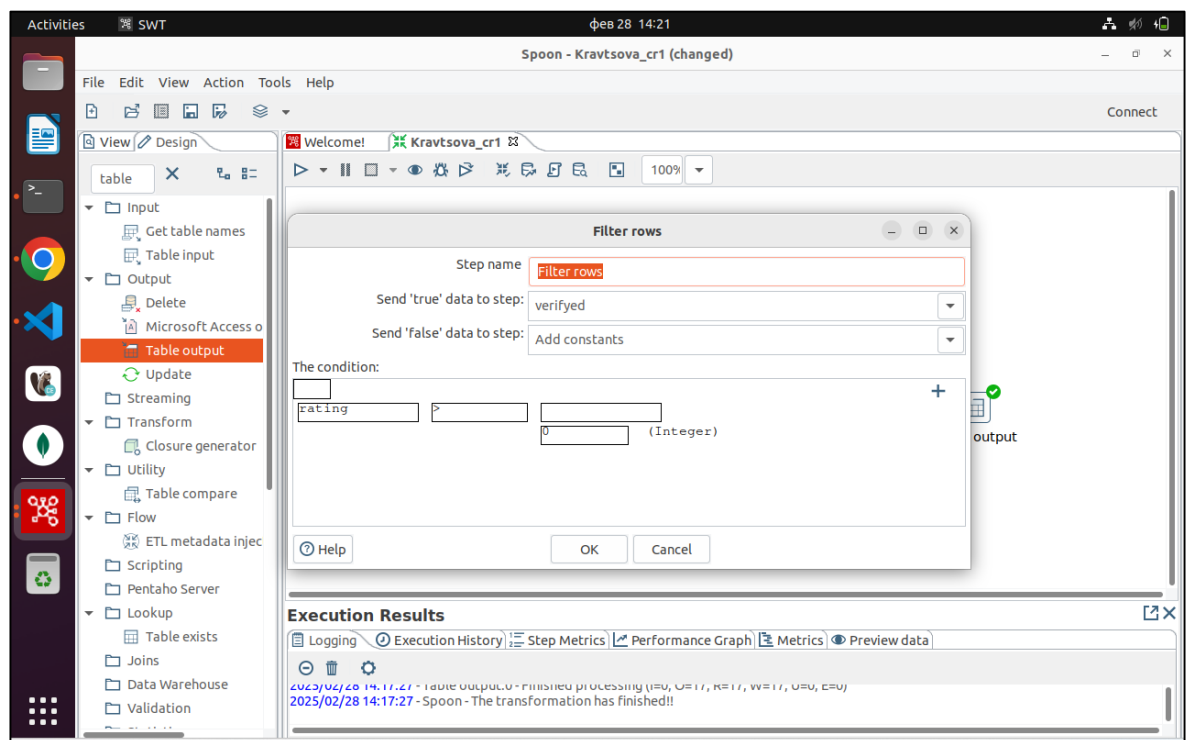


Рис. 14 – Фильтр по рейтингу

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

Step name: Add constants

Fields:

	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty
1	verification_status	Integer						rating	0	N

Buttons: Help, OK, Cancel

Рис. 15 – Значения не верифицированного пользователя

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

Step name: verified

Fields:

	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty
1	verification_status	Integer						rating	1	N

Buttons: Help, OK, Cancel

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.

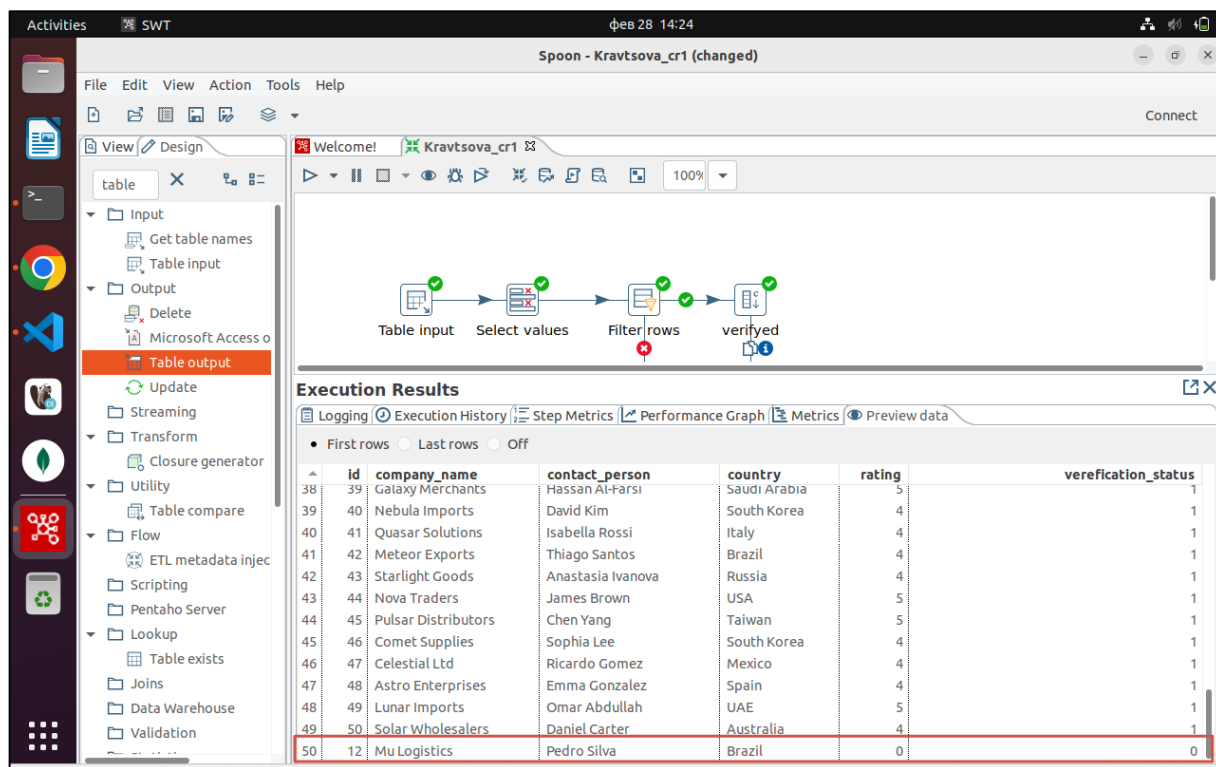


Рис. 17 – Проставленные статусы верификации

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

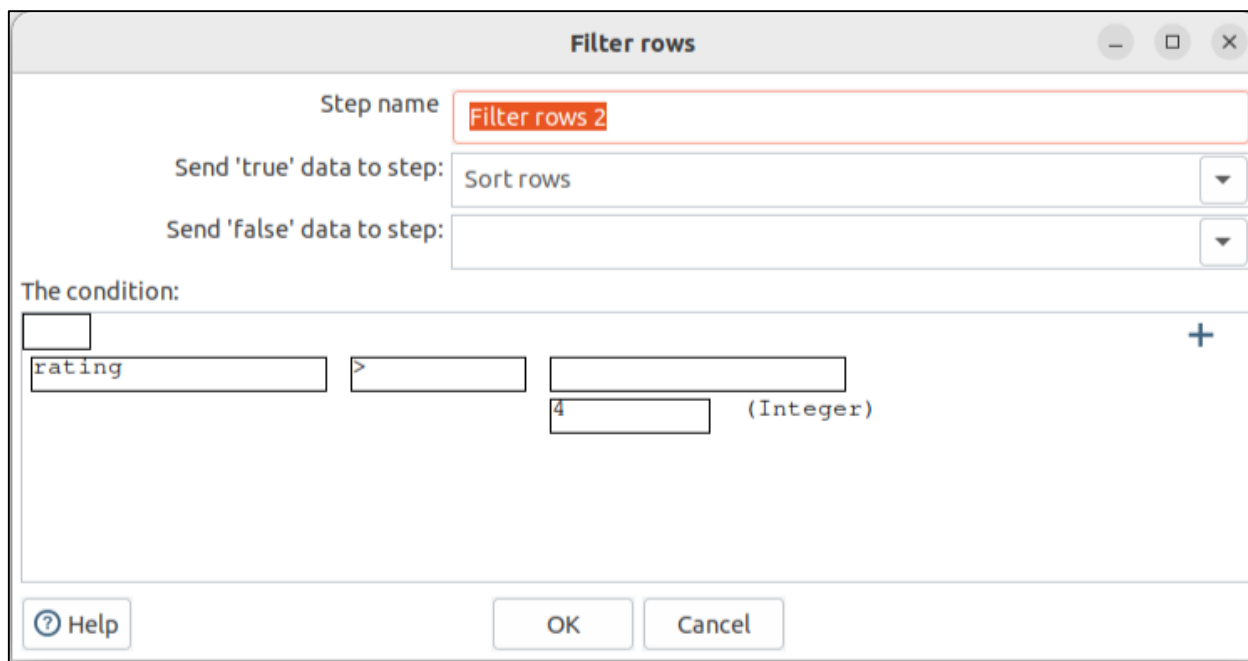


Рис. 18 – Фильтр по рейтингу

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

Table output

Step name: Table output

Connection: phpadmin [Edit...] [New...] [Wizard...]

Target schema: mgpu_ico_etl_06 [Browse...]

Target table: verified_suppliers [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options | **Database fields**

Partition data over tables: ☐

Partitioning field: [Dropdown]

Partition data per month: [Dropdown]

Partition data per day: [Dropdown]

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table: [Dropdown]

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: [Text]

[?] Help [OK] [Cancel] [SQL]

Рис. 19 – Подключение к базе MySQL

Запустим трансформацию (Рис. 20).

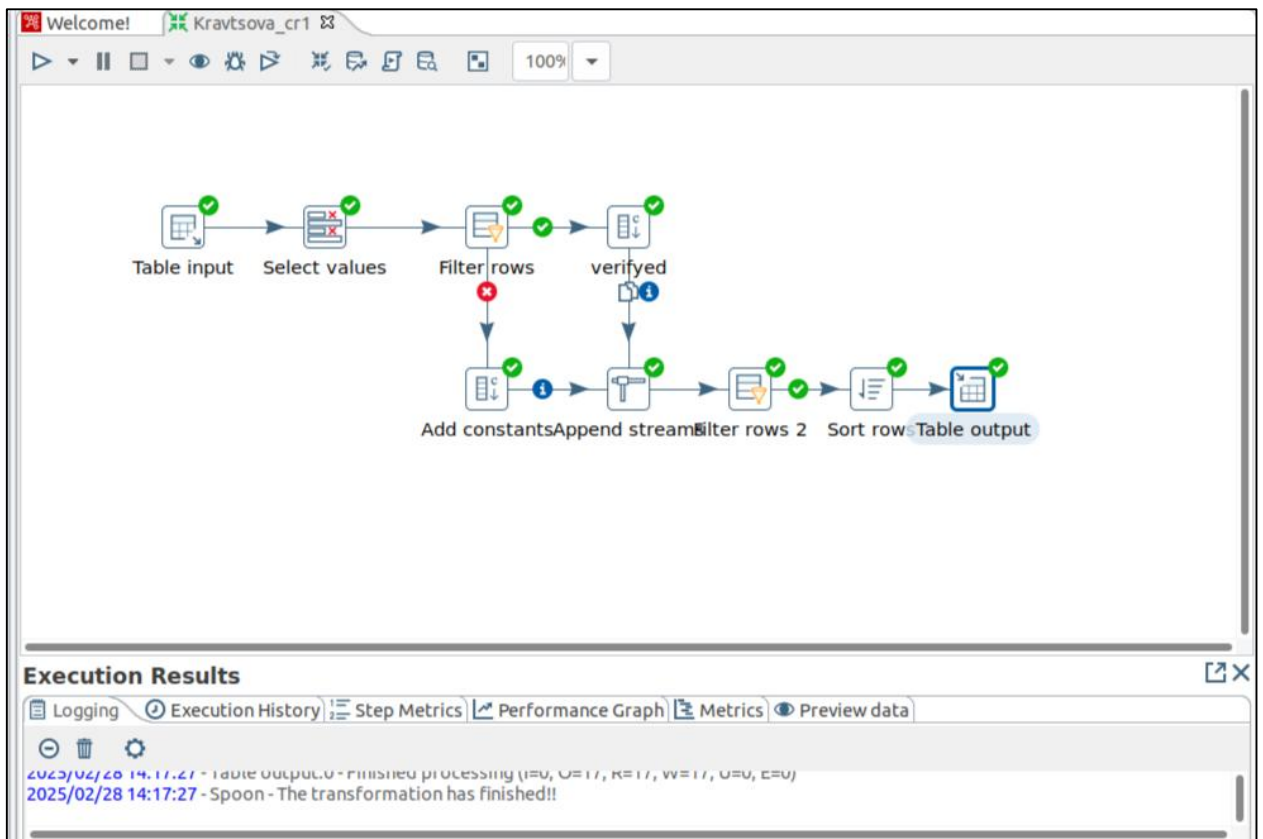


Рис. 20 – Успешное выполнение трансформации

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

The screenshot shows the phpMyAdmin interface for the 'verified_suppliers' table. The table has 25 rows and 6 columns: id, company_name, contact_person, country, rating, and verification_status. The data is displayed in a table format with a search bar and pagination controls. The table is located in the 'mgpu_ico_etl_06' database.

id	company_name	contact_person	country	rating	verification_status
49	Lunar Imports	Omar Abdullah	UAE	5	1
45	Pulsar Distributors	Chen Yang	Taiwan	5	1
44	Nova Traders	James Brown	USA	5	1
39	Galaxy Merchants	Hassan Al-Farsi	Saudi Arabia	5	1
37	Orion Trading	Sandra Mendes	Portugal	5	1
34	Skyline Ltd	Daniel Schmidt	Germany	5	1
32	Altitude Exports	Chang Liu	China	5	1
26	Zenith Distributors	Fatima Noor	Pakistan	5	1
25	Apex Merchants	Samuel Adams	USA	5	1
24	Omega Global	Mikhail Orlov	Kazakhstan	5	1
18	Sigma Ltd	Yusuf Demir	Turkey	5	1
16	Pi Distribution	James Wilson	Australia	5	1
13	Nu Enterprises	Lee Min-ho	South Korea	5	1
8	Theta Industries	Hiroshi Tanaka	Japan	5	1
5	Epsilon Exports	Ahmed Khan	UAE	5	1
3	Gamma Imports	Chen Wei	China	5	1
1	Alpha Supplies	John Smith	USA	5	1

Рис. 21 – Успешная загрузка данных

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

```
Run SQL query/queries on table mgpu_ico_etl_06.verified_suppliers:
1 SELECT country, COUNT(company_name) As company_count
2 from verified_suppliers
3 GROUP BY country;
```

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.