

Департамент образования города Москвы

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

Институт цифрового образования
Департамент информатики, управления и технологий

Самостоятельная работа 1
по дисциплине «Проектный практикум по разработке ETL-решений»

Тема: «Интеграция данных из разных источников (баз данных)»

Направление подготовки 38.03.05 – бизнес-информатика
Профиль подготовки «Аналитика данных и эффективное управление»
(очная форма обучения)

Выполнила:
Студентка группы АДЭУ-211
St_88

Москва
2025

ВВЕДЕНИЕ

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

Задачи:

1. Создать исходные таблицы в PostgreSQL с различными наборами данных.
2. Настроить целевые таблицы в MySQL для приема данных.
3. Разработать процессы трансформации данных в Pentaho.
4. Реализовать механизмы обработки ошибок и валидации данных.
5. Создать представления для связанных данных.

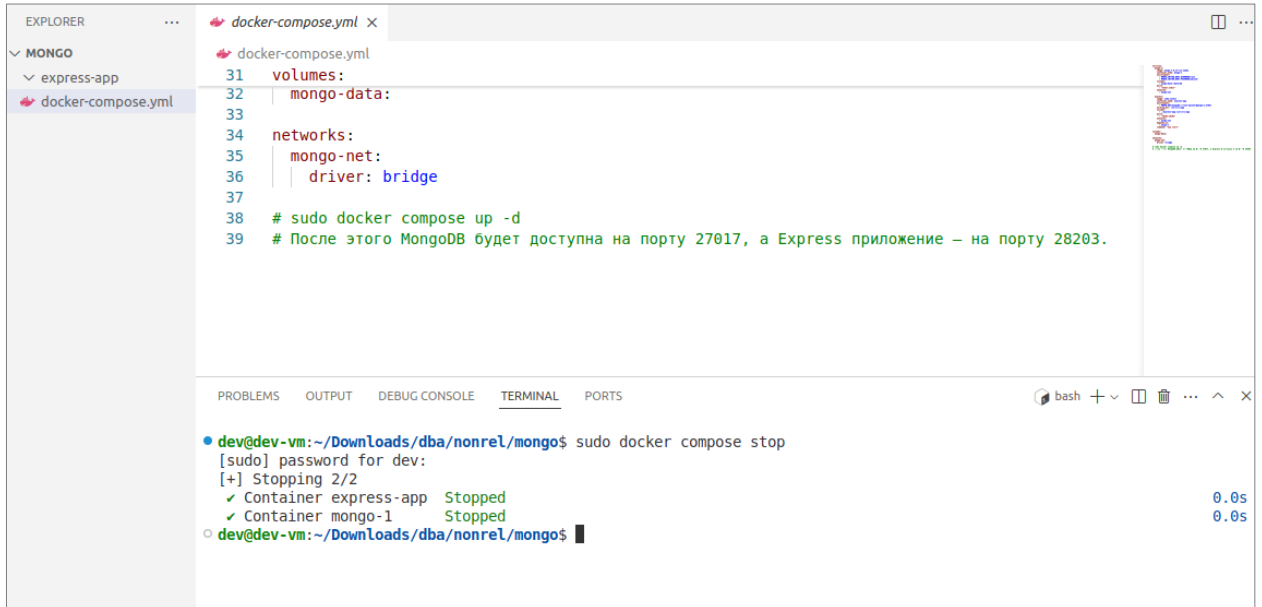
Вариант 1.

1. Создать таблицу products (id, name, category, price, stock_quantity, supplier_id)
2. Создать таблицу target_products с полями: id, name, category, price, stock_quantity, supplier_id, last_updated
3. Фильтрация товаров с количеством меньше 10
4. Расчет средней цены по категориям
5. Добавление метки времени обновления

ХОД РАБОТЫ

1. Подготавливаем среду для работы

Останавливаем контейнер MongoDB:



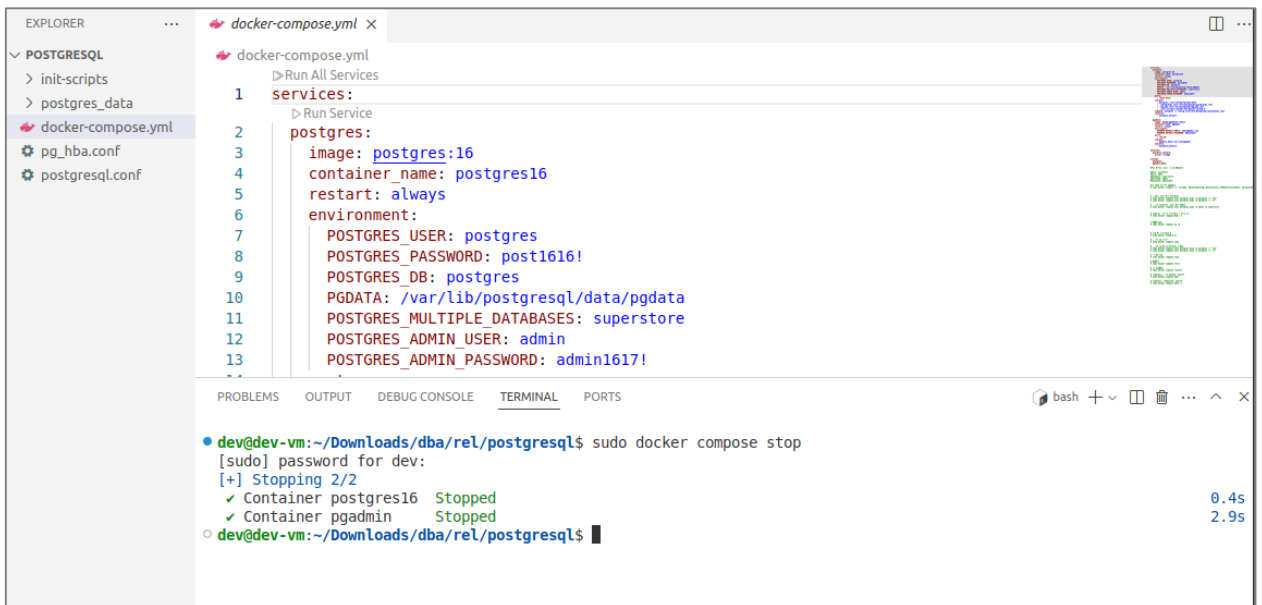
The screenshot shows the VS Code interface with the Explorer on the left showing a project structure with 'MONGO' and 'express-app'. The main editor shows a 'docker-compose.yml' file with the following content:

```
31 volumes:
32   mongo-data:
33
34 networks:
35   mongo-net:
36     driver: bridge
37
38 # sudo docker compose up -d
39 # После этого MongoDB будет доступна на порту 27017, а Express приложение – на порту 28203.
```

The TERMINAL panel at the bottom shows the execution of the stop command:

```
dev@dev-vm:~/Downloads/dba/nonrel/mongo$ sudo docker compose stop
[sudo] password for dev:
[+] Stopping 2/2
✓ Container express-app Stopped 0.0s
✓ Container mongo-1 Stopped 0.0s
dev@dev-vm:~/Downloads/dba/nonrel/mongo$
```

Далее останавливаем контейнер PostgreSQL:



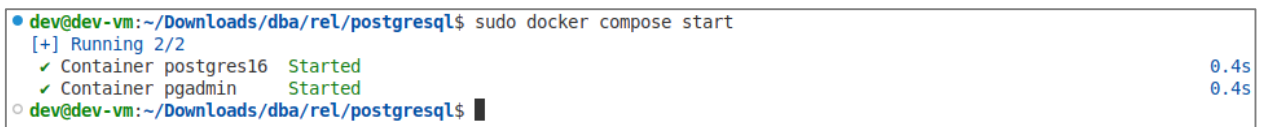
The screenshot shows the VS Code interface with the Explorer on the left showing a project structure with 'POSTGRES' and 'init-scripts'. The main editor shows a 'docker-compose.yml' file with the following content:

```
1 services:
2   postgres:
3     image: postgres:16
4     container_name: postgres16
5     restart: always
6     environment:
7       POSTGRES_USER: postgres
8       POSTGRES_PASSWORD: post1616!
9       POSTGRES_DB: postgres
10      PGDATA: /var/lib/postgresql/data/pgdata
11      POSTGRES_MULTIPLE_DATABASES: superstore
12      POSTGRES_ADMIN_USER: admin
13      POSTGRES_ADMIN_PASSWORD: admin1617!
```

The TERMINAL panel at the bottom shows the execution of the stop command:

```
dev@dev-vm:~/Downloads/dba/rel/postgresql$ sudo docker compose stop
[sudo] password for dev:
[+] Stopping 2/2
✓ Container postgres16 Stopped 0.4s
✓ Container pgadmin Stopped 2.9s
dev@dev-vm:~/Downloads/dba/rel/postgresql$
```

Повторно запускаем docker-контейнер PostgreSQL:



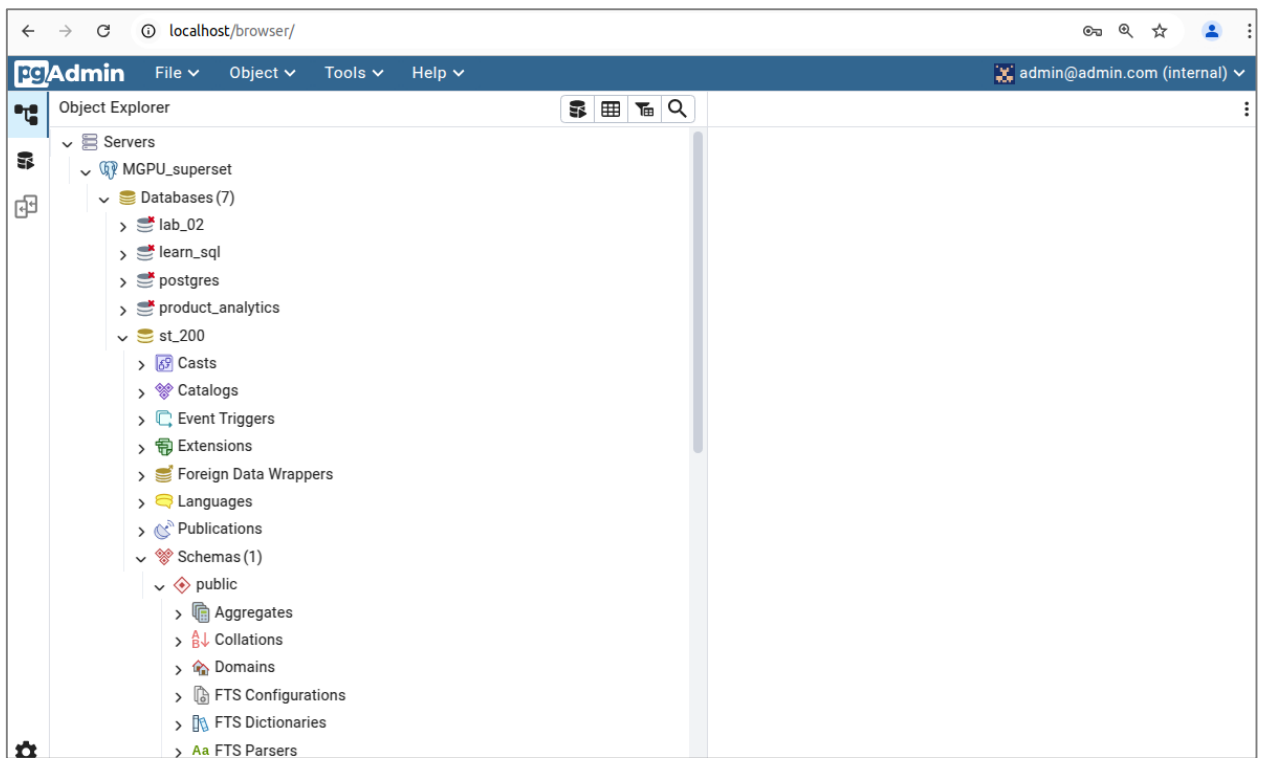
The screenshot shows the VS Code interface with the Explorer on the left showing a project structure with 'POSTGRES' and 'init-scripts'. The main editor shows a 'docker-compose.yml' file with the following content:

```
1 services:
2   postgres:
3     image: postgres:16
4     container_name: postgres16
5     restart: always
6     environment:
7       POSTGRES_USER: postgres
8       POSTGRES_PASSWORD: post1616!
9       POSTGRES_DB: postgres
10      PGDATA: /var/lib/postgresql/data/pgdata
11      POSTGRES_MULTIPLE_DATABASES: superstore
12      POSTGRES_ADMIN_USER: admin
13      POSTGRES_ADMIN_PASSWORD: admin1617!
```

The TERMINAL panel at the bottom shows the execution of the start command:

```
dev@dev-vm:~/Downloads/dba/rel/postgresql$ sudo docker compose start
[+] Running 2/2
✓ Container postgres16 Started 0.4s
✓ Container pgadmin Started 0.4s
dev@dev-vm:~/Downloads/dba/rel/postgresql$
```

Проверяем доступность СУБД Postgre SQL на локальном сервере:



2. Работа с PostgreSQL

Непосредственно в PostgreSQL создаем новую базу данных по имени идентификатора студента st_88:

Create - Database

×

General

Definition

Security

Parameters

Advanced

SQL

Database

st_88

OID

Owner

admin

⌵

Comment

i

?

×

Close

↺

Reset

💾

Save

Далее с помощью SQL-запроса создаем таблицу товаров `products_my`, состоящую из следующих полей:

`id` – идентификатор записи;

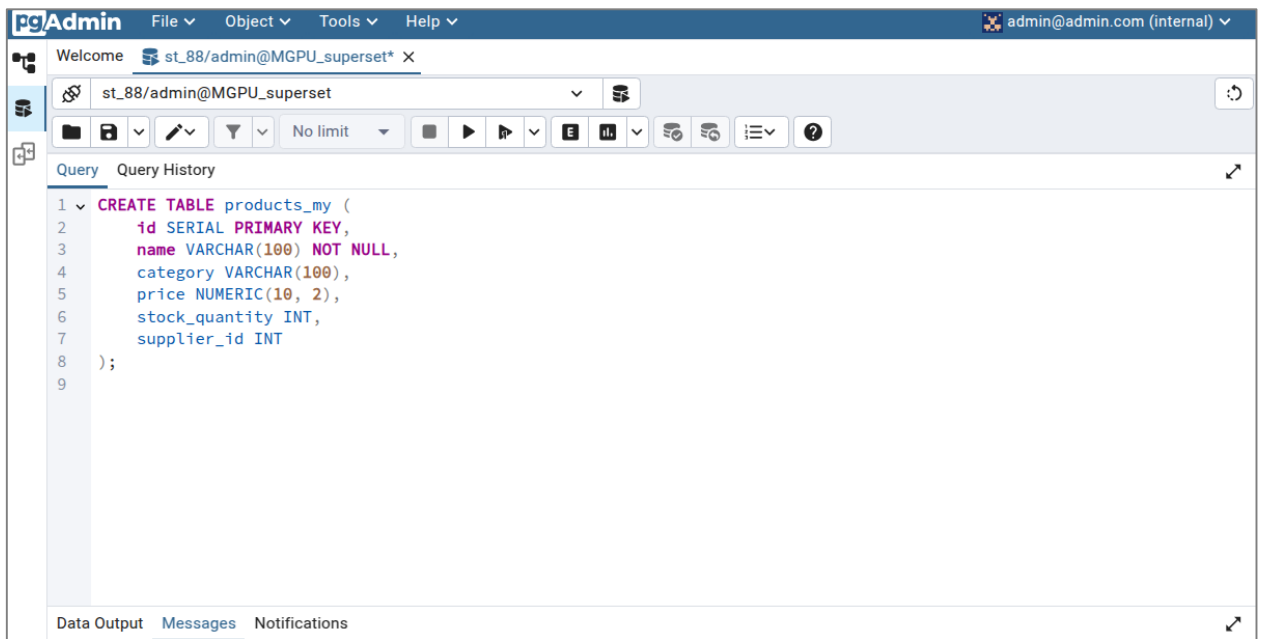
`name` – наименование товара;

`category` – категория товара;

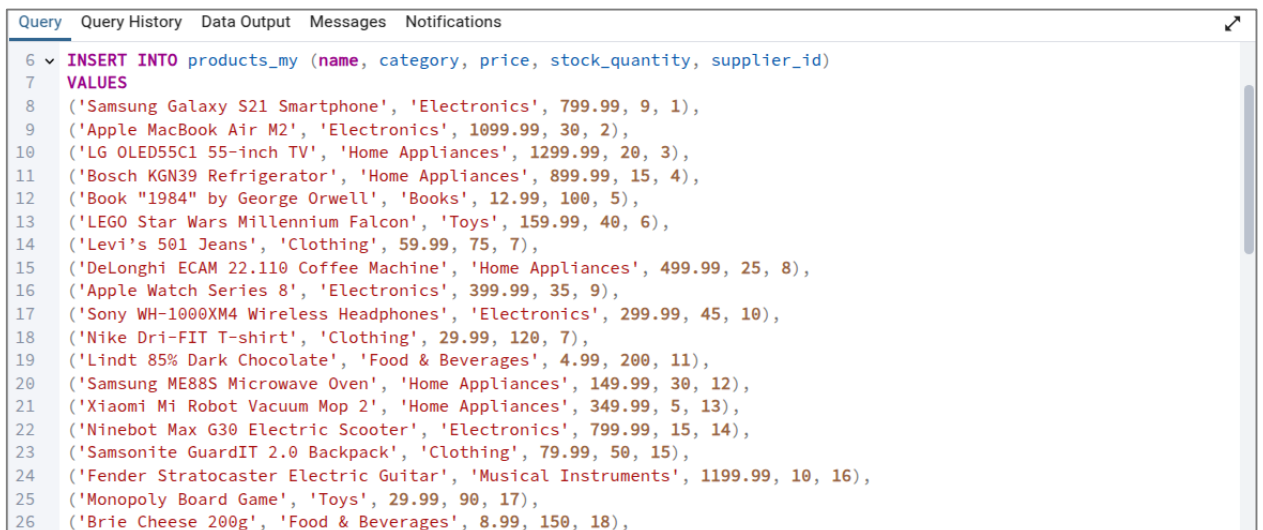
`price` – цена товара;

`stock_quantity` – количество товара;

`supplier_id` – идентификатор поставщика товара.



Далее заранее подготавливаем код для генерации данных о товарах в таблицу и запускаем скрипт:



Проверяем, что данные успешно записаны в таблицу:

The screenshot shows the PgAdmin web interface. The 'Data Output' tab is active, displaying a table with 14 rows. The table has the following columns: id (integer, PK), name (character varying (100)), category (character varying (100)), price (numeric (10,2)), stock_quantity (integer), and supplier_id (integer). The data includes various products like Samsung Galaxy S21, Apple MacBook Air M2, LG OLED55C1 TV, Bosch KGN39 Refrigerator, etc.

id	name	category	price	stock_quantity	supplier_id
71	Samsung Galaxy S21 Smartphone	Electronics	799.99	9	1
72	Apple MacBook Air M2	Electronics	1099.99	30	2
73	LG OLED55C1 55-inch TV	Home Appliances	1299.99	20	3
74	Bosch KGN39 Refrigerator	Home Appliances	899.99	15	4
75	Book "1984" by George Orwell	Books	12.99	100	5
76	LEGO Star Wars Millennium Falcon	Toys	159.99	40	6
77	Levi's 501 Jeans	Clothing	59.99	75	7
78	DeLonghi ECAM 22.110 Coffee Machine	Home Appliances	499.99	25	8
79	Apple Watch Series 8	Electronics	399.99	35	9
80	Sony WH-1000XM4 Wireless Headphones	Electronics	299.99	45	10
81	Nike Dri-FIT T-shirt	Clothing	29.99	120	7
82	Lindt 85% Dark Chocolate	Food & Beverages	4.99	200	11
83	Samsung ME88S Microwave Oven	Home Appliances	149.99	30	12
84	Xiaomi MI Robot Vacuum Mop 2	Home Appliances	349.99	5	13

3. Работа с MySQL

Далее переходим в MySQL и создаем целевую таблицу, в которую будут записываться данные по итогам выполнения работы. Структура таблицы аналогична исходной, добавляется лишь поле `last_updated` для отображения метки даты последнего обновления данных:

The screenshot shows the phpMyAdmin interface. The 'SQL' tab is active, displaying a SQL query to create a new table named 'target_products'. The query defines the table structure with columns: id (INT AUTO_INCREMENT PRIMARY KEY), name (VARCHAR(100) NOT NULL), category (VARCHAR(100)), price (DECIMAL(10, 2)), stock_quantity (INT), supplier_id (INT), and last_updated (DATE).

```

1 CREATE TABLE target_products (
2   id INT AUTO_INCREMENT PRIMARY KEY,
3   name VARCHAR(100) NOT NULL,
4   category VARCHAR(100),
5   price DECIMAL(10, 2),
6   stock_quantity INT,
7   supplier_id INT,
8   last_updated DATE
9 );
10

```

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: target_products

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Слежение Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 name	varchar(100)	utf8mb4_unicode_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 category	varchar(100)	utf8mb4_unicode_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 price	decimal(10,2)			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	5 stock_quantity	int			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	6 supplier_id	int			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	7 last_updated	date			Да	NULL			Изменить Удалить Ещё

↑ ☐ Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс Пространственный

Удалить из центральных столбцов

Печать Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после last_updated Вперёд

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
Изменить Переименовать Удалить	PRIMARY	BTREE	Да	Нет	id	0	A	Нет	

Также заранее подготовим вторую таблицу в MySQL для вывода результата выполнения 4 задания по варианту 1 – расчет средней цены в разрезе категорий товаров. Данная таблица состоит из следующих полей:

id – идентификатор записи;

category – категория товаров;

average_price – средняя цена товаров в данной категории;

median_price – медианная цена товаров в данной категории.

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01

Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Процедуры События Триггеры Ещё

Выполнить SQL-запрос(ы) к базе данных mgpu_ico_etl_01:

```

1 CREATE TABLE price_per_category (
2   id SERIAL PRIMARY KEY,
3   category VARCHAR(100) NOT NULL,
4   average_price NUMERIC(10, 2),
5   median_price NUMERIC(10, 2)
6 );

```


Сервер: localhost:3306 » База данных: mgrp_ico_etl_01 » Таблица: price_per_category

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Слежение Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 category	varchar(100)	utf8mb4_unicode_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 average_price	decimal(10,2)			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 median_price	decimal(10,2)			Да	NULL			Изменить Удалить Ещё

☐ Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс Пространственный
 Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов

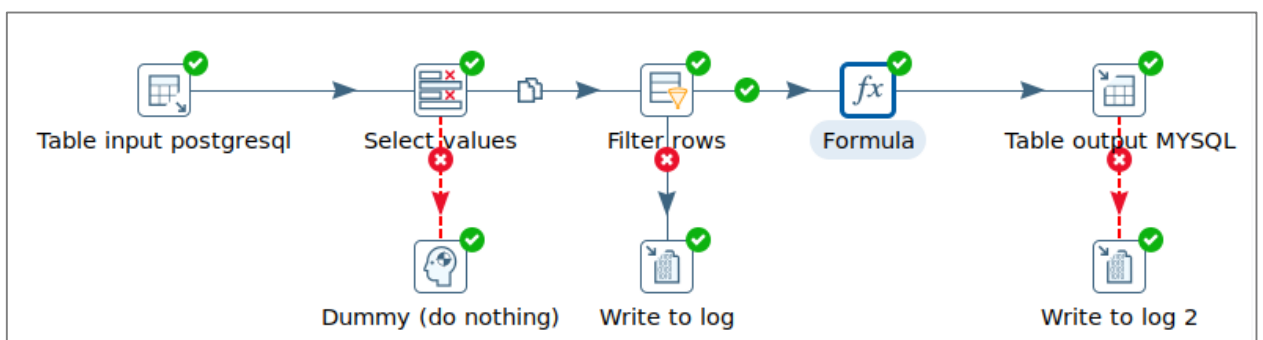
4. Работа с Pentaho

Запускаем Pentaho с помощью терминальной команды `./spoon.sh`:

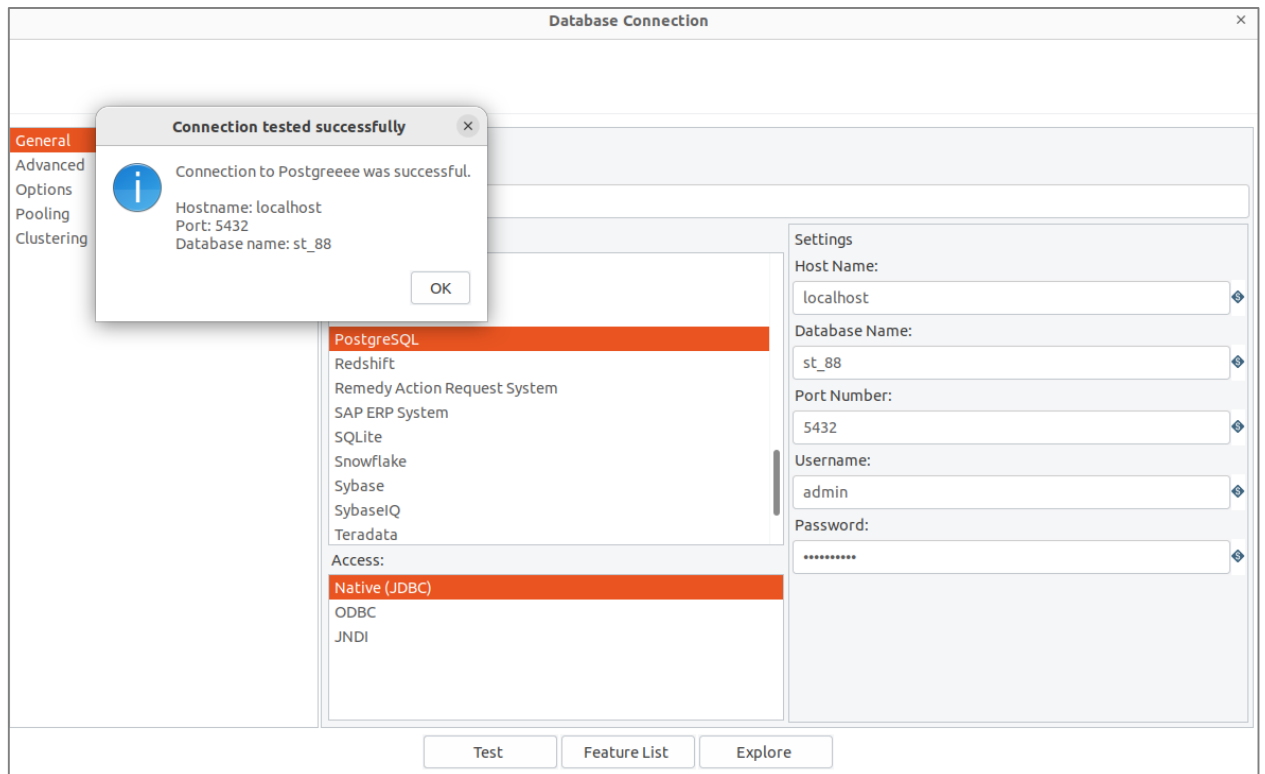
```
dev@dev-vm: ~/Downloads/lab_etl/pdi-ce-9.4.0.0-343/data-i...
dev@dev-vm:~$ cd Downloads
dev@dev-vm:~/Downloads$ ls
dba de lab_etl progs
dev@dev-vm:~/Downloads$ cd lab_etl/
dev@dev-vm:~/Downloads/lab_etl$ ls
data_for_labs pdi-ce-9.4.0.0-343 psw-ce-9.4.0.0-343
pdi-ce-9.4.0.0-343-hadoop-addon pdi-ce-9.4.0.0-343-hadoop-addon.zip psw-ce-9.4.0.0-343.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$ 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
```

4.1. Создание трансформации для выполнения индивидуальных заданий №1, 2, 3, 5

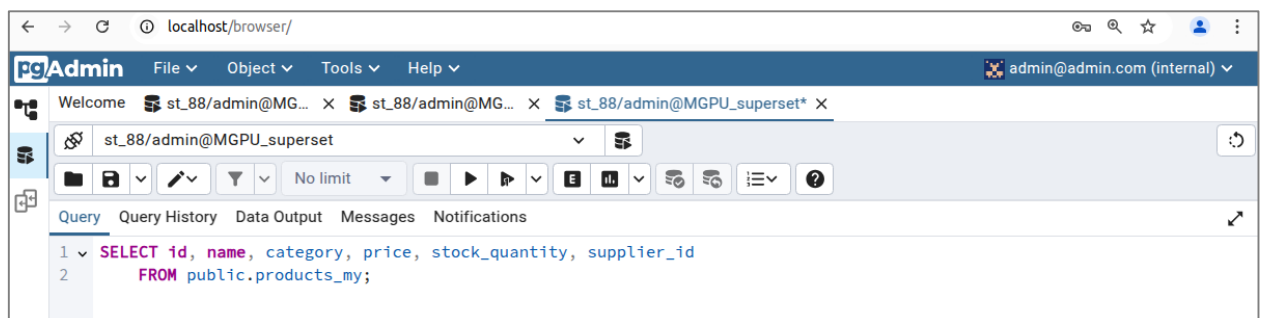
В первую очередь начинаем создавать трансформацию для перекачки данных из базы PostgreSQL в MySQL. Готовая трансформация будет иметь следующий вид и далее будет описана подробно:



Добавляем компонент импорта данных из таблицы. Настраиваем подключение к PostgreSQL и проверяем успешность:



Далее генерируем скрипт для получения данных из таблицы *products_my* в самом pgAdmin:



Возвращаемся в Pentaho и вставляем скрипт в компонент *Table input*:

Table input

Step name

Table input postgresql

Connection

Postgreeee

Edit...

New...

Wizard...

SQL

Get SQL select statement...

```
SELECT id, name, category, price, stock_quantity, supplier_id
FROM public.products_my;
```

Line 2 Column 25

Store column info in step meta data

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size

0

Help

OK

Preview

Cancel

Просматриваем импортируемые данные о товарах:

Rows of step: Table input postgresql (100 rows)						
	id	name	category	price	stock_quantity	supplier_id
1	71	Samsung Galaxy S21 Smartphone	Electronics	799.99	9	1
2	72	Apple MacBook Air M2	Electronics	1099.99	30	2
3	73	LG OLED55C1 55-inch TV	Home Appliances	1299.99	20	3
4	74	Bosch KGN39 Refrigerator	Home Appliances	899.99	15	4
5	75	Book "1984" by George Orwell	Books	12.99	100	5
6	76	LEGO Star Wars Millennium Falcon	Toys	159.99	40	6
7	77	Levi's 501 Jeans	Clothing	59.99	75	7
8	78	DeLonghi ECAM 22.110 Coffee Machine	Home Appliances	499.99	25	8
9	79	Apple Watch Series 8	Electronics	399.99	35	9
10	80	Sony WH-1000XM4 Wireless Headphones	Electronics	299.99	45	10
11	81	Nike Dri-FIT T-shirt	Clothing	29.99	120	7
12	82	Lindt 85% Dark Chocolate	Food & Beverages	4.99	200	11
13	83	Samsung ME88S Microwave Oven	Home Appliances	149.99	30	12
14	84	Xiaomi Mi Robot Vacuum Mop 2	Home Appliances	349.99	5	13
15	85	Ninebot Max G30 Electric Scooter	Electronics	799.99	15	14
16	86	Samsonite GuardIT 2.0 Backpack	Clothing	79.99	50	15
17	87	Fender Stratocaster Electric Guitar	Musical Instruments	1199.99	10	16
18	88	Monopoly Board Game	Toys	29.99	90	17
19	89	Brie Cheese 200g	Food & Beverages	8.99	150	18
20	90	Moleskine Classic Notebook	Stationery	24.99	200	19
21	91	Book "Harry Potter and the Sorcerer's Stone"	Books	14.99	85	5
22	92	Mobil 1 5W-30 Motor Oil	Automotive	39.99	70	20
23	93	JBL Charge 5 Portable Speaker	Electronics	179.99	40	10
24	94	Adidas Ultraboost Running Shoes	Clothing	129.99	60	7
25	95	Lavazza Coffee Beans 1kg	Food & Beverages	18.99	250	11
26	96	Xiaomi Redmi Note 12 Smartphone	Electronics	349.99	80	1
27	97	Redmond RMC-M90 Multicooker	Home Appliances	99.99	50	12

Далее добавляем компонент *Select values* для вывода полей таблицы. Какие-либо переименования в данном случае не требуются:

Select values

Step name

Select values

Select & Alter

Remove

Meta-data

Fields :

	Fieldname	Rename to	Length	Precision
1	id			
2	name			
3	category			
4	price			
5	stock_quantity			
6	supplier_id			

Get fields to select

Edit Mapping

Далее по заданию №3 необходимо отфильтровать товары с количеством меньше 10. Т.к. в основном в сгенерированных данных количество больше 10, то мы будем убирать данные, где количество меньше 10. Для фильтрации будем использовать компонент *Filter rows*:

Filter rows

Step name

Filter rows

Send 'true' data to step:

Send 'false' data to step:

The condition:

+

stock_quantity

>=

10

(Integer)

Товары, количество которых меньше 10, будут записываться в логи:

Write to log

Step name:

Log level:

Print header: ☒

Limit rows?: ☐

Nr of rows to print:

Write to log:

Fields

	Field
1	id
2	name
3	stock_quantity

По заданию №5 необходимо добавить метку времени обновления. Для этого используем компонент *Formula*: создаем новое поле *last_updated*, в которое будет записываться текущая дата:

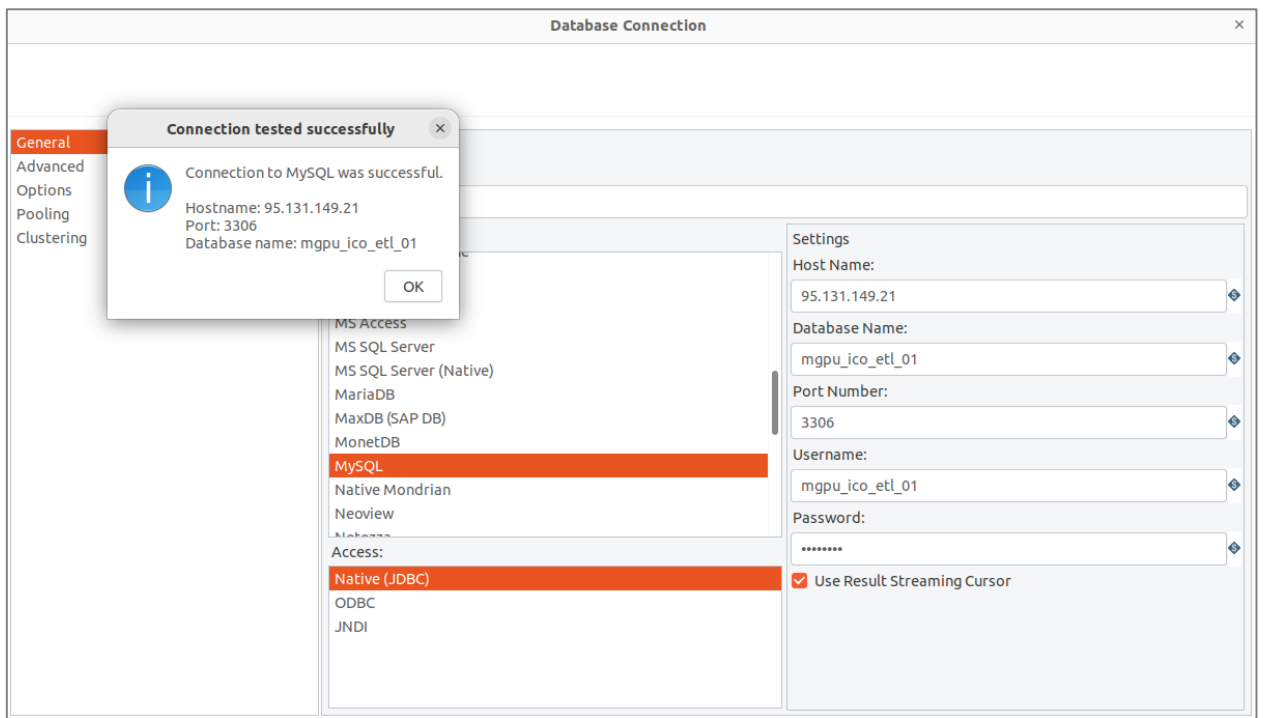
Formula

Step name:

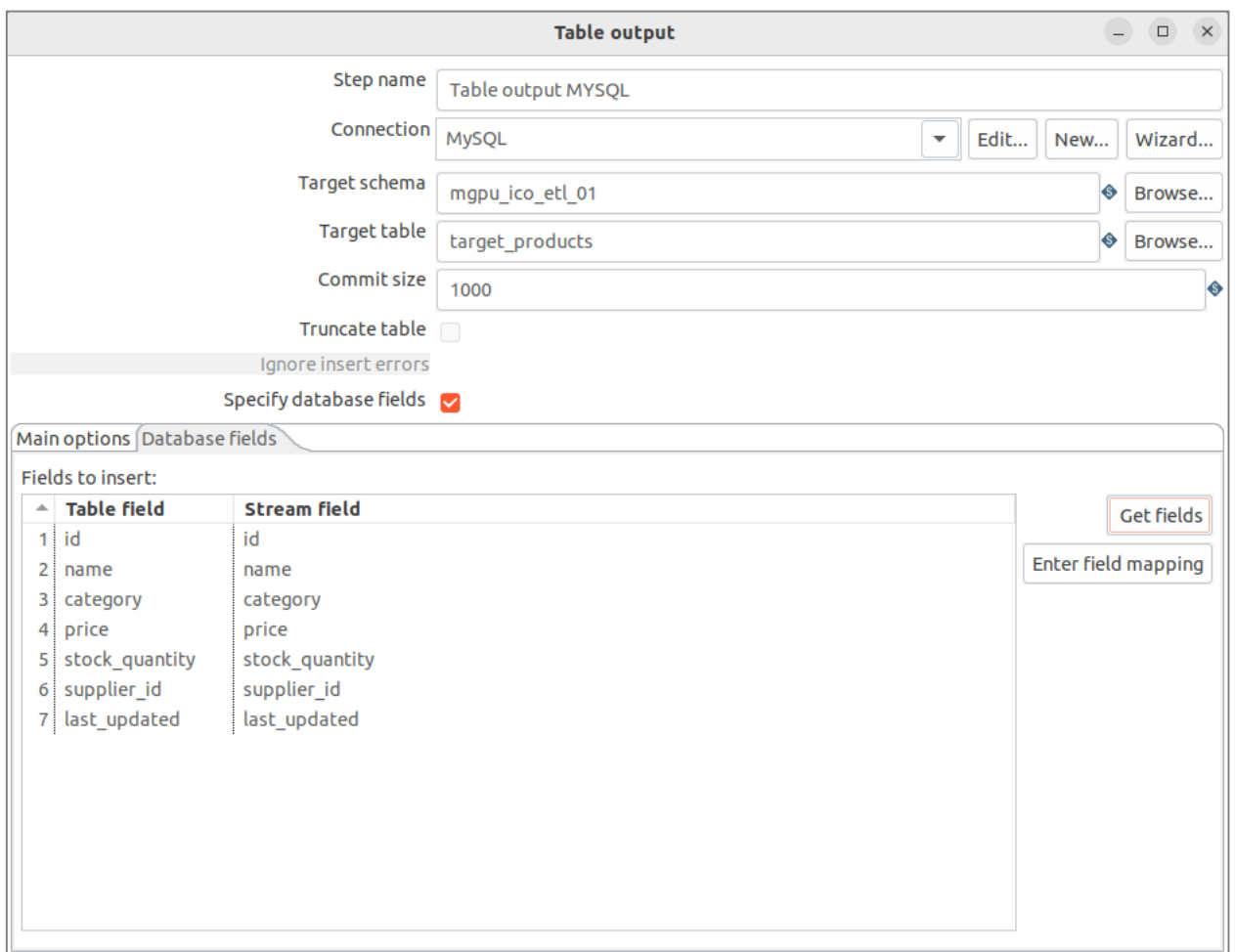
Fields:

	New field	Formula	Value type	Length	Precision	Replace value
1	last_updated	TODAY()	Date			

На данном этапе работа по трансформации данных закончена, поэтому далее добавляем компонент *Table output* для экспорта результата в таблицу MySQL. Создаем новое подключение к базе данных и проверяем успешность:

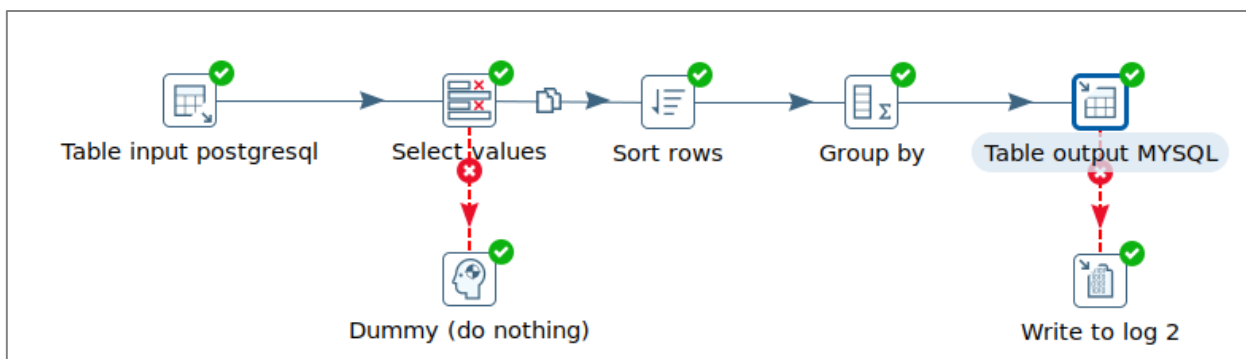


Выбираем нужную, созданную ранее, целевую таблицу *target_products* и делаем маппинг полей:



4.2. Создание трансформации для выполнения индивидуальных заданий №4

Перейдем к выполнению индивидуального задания №4 – расчет средней цены по категориям. Для этого создадим отдельную трансформацию:



Она состоит аналогично из компонентов подключения и импорта данных из PostgreSQL (используется созданная ранее таблица *products_my* со сгенерированными данными о товарах).

Далее добавляется компонент сортировки данных по категориям (подготовка данных к дальнейшей группировке):

The screenshot shows the 'Sort rows' configuration window. The 'Step name' is 'Sort rows'. The 'Sort directory' is set to '%java.io.tmpdir%'. The 'TMP-file prefix' is 'out'. The 'Sort size (rows in memory)' is '1000000'. The 'Free memory threshold (in %)' is empty. The 'Compress TMP Files?' checkbox is checked. The 'Only pass unique rows? (verifies keys only)' checkbox is unchecked. The 'Fields' section shows a table with one row: 'category' is selected for sorting, with 'Ascending' order, 'Case sensitive compare?' checked, 'Sort based on current locale?' checked, 'Collator Strength' set to 'default', and 'Presorted?' set to 'No'.

	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	category	Y	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	default	No

Группируем поля для расчета средней цены товаров в разрезе каждой категории. Добавляются новые агрегированные поля для расчета средней и медианной цен товаров из каждой категории:

Group by

Step name

Group by

Include all rows?

☐

Temporary files directory

%%java.io.tmpdir%%

Browse...

TMP-file prefix

grp

Add line number, restart in each group

Line number field name

Always give back a result row

☐

The fields that make up the group:

Group field

1 category

Get Fields

Aggregates :

	Name	Subject	Type
1	average_price	price	Average (Mean)
2	median_price	price	Median

Get lookup fields

Используем компонент экспорта данных в MySQL. Настраиваем подключение к базе данных и выбираем нужную таблицу:

Table output

Step name

Table output MYSQL

Connection

mysql

Edit...

New...

Wizard...

Target schema

mgpu_ico_etl_01

Browse...

Target table

price_per_category

Browse...

Commit size

1000

Truncate table

☐

Ignore insert errors

☐

Specify database fields

☒

Main options

Database fields

Fields to insert:

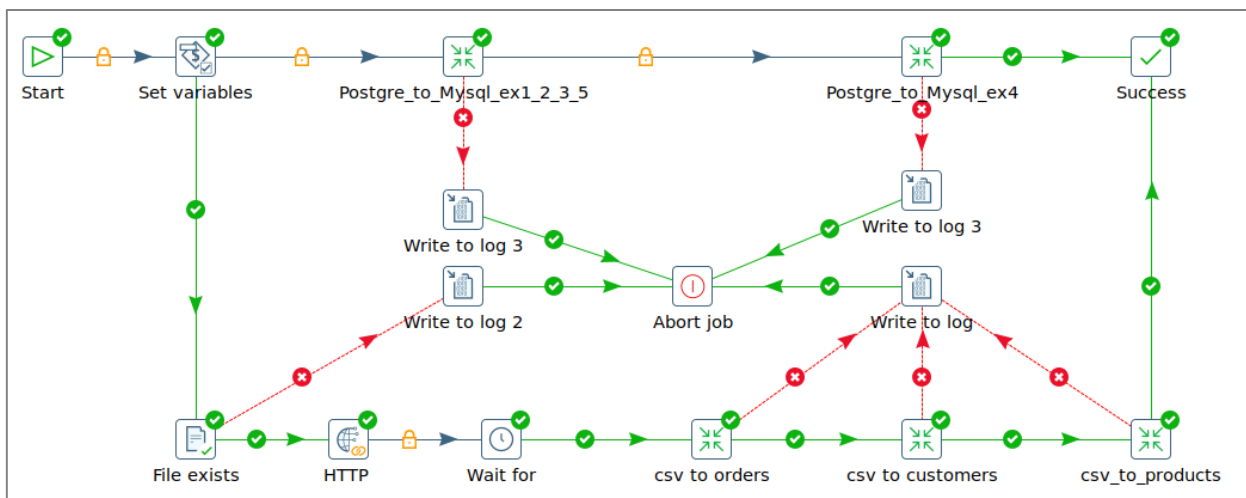
	Table field	Stream field
1	category	category
2	average_price	average_price
3	median_price	median_price

Get fields

Enter field mapping

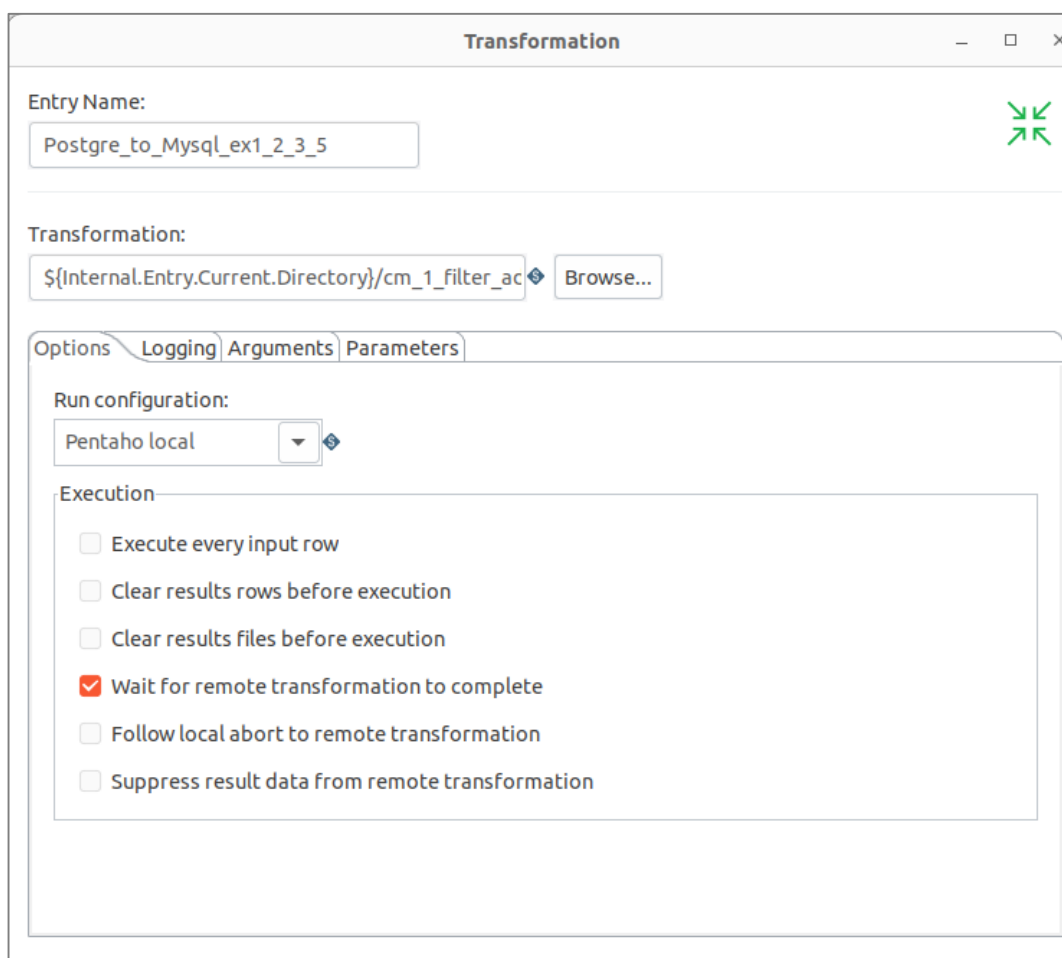
4.3. Объединение трансформаций в единую работу

Итоговая работа выглядит следующим образом:



В рамках данной работы были объединены только что созданные две трансформации для перекачки данных из PostgreSQL в MySQL, а также три трансформации для импорта данных из CSV-файла и экспорта в базу данных MySQL (они были рассмотрены подробно ранее в лабораторной работе №2).

Настраиваем путь корректный путь к трансформациям:

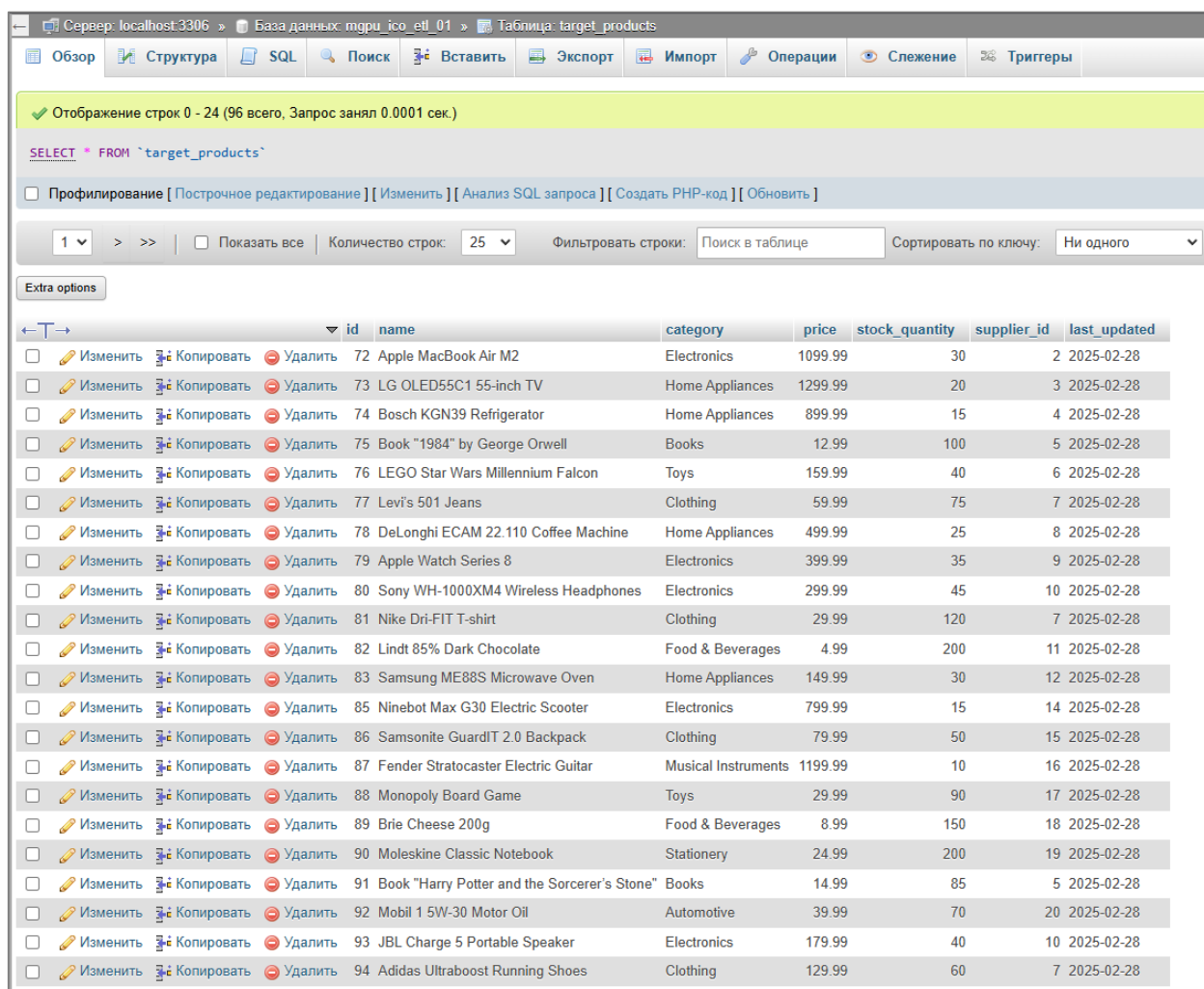


Запускаем работу – выполнена без ошибок.

5. Проверка результатов

Для проверки результатов выполнения работы необходимо перейти в phpMyAdmin.

Результат выполнения перекачки данных из PostgreSQL в MySQL, фильтрации по количеству товаров и добавлению метки о дате обнрвления успешно записан в таблицу *target_products*:



Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: target_products

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Слежение Триггеры

Отображение строк 0 - 24 (96 всего, Запрос занял 0.0001 сек.)

SELECT * FROM `target_products`

Профилерование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> | ☐ Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ни одного

Extra options

			id	name	category	price	stock_quantity	supplier_id	last_updated
<input type="checkbox"/>	Изменить	Копировать	Удалить	72	Apple MacBook Air M2	Electronics	1099.99	30	2 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	73	LG OLED55C1 55-inch TV	Home Appliances	1299.99	20	3 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	74	Bosch KGN39 Refrigerator	Home Appliances	899.99	15	4 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	75	Book "1984" by George Orwell	Books	12.99	100	5 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	76	LEGO Star Wars Millennium Falcon	Toys	159.99	40	6 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	77	Levi's 501 Jeans	Clothing	59.99	75	7 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	78	DeLonghi ECAM 22.110 Coffee Machine	Home Appliances	499.99	25	8 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	79	Apple Watch Series 8	Electronics	399.99	35	9 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	80	Sony WH-1000XM4 Wireless Headphones	Electronics	299.99	45	10 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	81	Nike Dri-FIT T-shirt	Clothing	29.99	120	7 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	82	Lindt 85% Dark Chocolate	Food & Beverages	4.99	200	11 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	83	Samsung ME88S Microwave Oven	Home Appliances	149.99	30	12 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	85	Ninebot Max G30 Electric Scooter	Electronics	799.99	15	14 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	86	Samsonite GuardIT 2.0 Backpack	Clothing	79.99	50	15 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	87	Fender Stratocaster Electric Guitar	Musical Instruments	1199.99	10	16 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	88	Monopoly Board Game	Toys	29.99	90	17 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	89	Brie Cheese 200g	Food & Beverages	8.99	150	18 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	90	Moleskine Classic Notebook	Stationery	24.99	200	19 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	91	Book "Harry Potter and the Sorcerer's Stone"	Books	14.99	85	5 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	92	Mobil 1 5W-30 Motor Oil	Automotive	39.99	70	20 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	93	JBL Charge 5 Portable Speaker	Electronics	179.99	40	10 2025-02-28
<input type="checkbox"/>	Изменить	Копировать	Удалить	94	Adidas Ultraboost Running Shoes	Clothing	129.99	60	7 2025-02-28

Результат расчета средних цен товаров в разрезе категорий успешно записан в таблицу *price_per_category*:

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: price_per_category

Обзор Структура SQL Поиск Вставить Экспорт Импорт

Отображение строк 0 - 21 (22 всего, Запрос занял 0.0003 сек.) [average_price: 1199.99... - 11.49...]

SELECT * FROM `price_per_category` ORDER BY `price_per_category`.`average_price` DESC

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировка

Extra options

				id	category	average_price	median_price
<input type="checkbox"/>				15	Musical Instruments	1199.99	1199.99
<input type="checkbox"/>				10	Health & Wellness	989.99	989.99
<input type="checkbox"/>				7	Electronics	488.74	299.99
<input type="checkbox"/>				22	Travel Accessories	437.50	437.50
<input type="checkbox"/>				12	Home Appliances	436.24	399.99
<input type="checkbox"/>				3	Baby Products	399.99	399.99
<input type="checkbox"/>				16	Networking	399.99	399.99
<input type="checkbox"/>				6	Cosmetics	199.99	199.99
<input type="checkbox"/>				19	Sports & Fitness	157.49	124.99
<input type="checkbox"/>				1	Accessories	149.99	149.99
<input type="checkbox"/>				5	Clothing	90.90	79.99
<input type="checkbox"/>				13	Home Textiles	89.99	89.99
<input type="checkbox"/>				21	Toys	86.66	69.99
<input type="checkbox"/>				17	Outdoor & Travel	81.10	49.99
<input type="checkbox"/>				9	Furniture	79.99	79.99
<input type="checkbox"/>				18	Photography	74.99	74.99
<input type="checkbox"/>				14	Lighting	49.99	49.99
<input type="checkbox"/>				11	Home & Kitchen	49.99	39.99
<input type="checkbox"/>				2	Automotive	39.99	39.99
<input type="checkbox"/>				20	Stationery	24.99	24.99
<input type="checkbox"/>				4	Books	20.74	19.99
<input type="checkbox"/>				8	Food & Beverages	11.49	10.99

Имеем, что всего было представлено 22 категории товаров, из них наивысшая средняя цена товара у категории Музыкальные инструменты, а наименьшая – у Еды.

Результаты экспорта данных из CSV-файла (данные успешно записались в соответствующие таблицы MySQL):

Таблица товаров:

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: products

Обзор

Структура

SQL

Поиск

Вставить

Экспорт

Импорт

Операции

Слежение

Триггеры

✔ Отображение строк 0 - 24 (5371 всего, Запрос занял 0.0008 сек.)

SELECT * FROM `products`

Профилирование

Построчное редактирование

Изменить

Анализ SQL запроса

Создать PHP-код

Обновить

1

>

>>

Количество строк: 25

Фильтровать строки: Поиск в таблице

Сортировать по ключу: Ни одного

Extra options

			id	product_id	category	sub_category	product_name	person	
<input type="checkbox"/>	Изменить	Копировать	Удалить	1	OFF-AP-10002578	Office Supplies	Appliances	Fellowes Premier Superior Surge Suppressor, 10-Out...	Chuck Magee
<input type="checkbox"/>	Изменить	Копировать	Удалить	2	OFF-PA-10000575	Office Supplies	Paper	Wirebound Message Books, Four 2 3/4 x 5 White Form...	Chuck Magee
<input type="checkbox"/>	Изменить	Копировать	Удалить	3	TEC-MA-10002790	Technology	Machines	NeatDesk Desktop Scanner & Digital Filing System	Kelly Williams
<input type="checkbox"/>	Изменить	Копировать	Удалить	4	OFF-AR-10000255	Office Supplies	Art	Newell 328	Kelly Williams
<input type="checkbox"/>	Изменить	Копировать	Удалить	5	TEC-PH-10001061	Technology	Phones	Apple iPhone 5C	Cassandra Brandow
<input type="checkbox"/>	Изменить	Копировать	Удалить	6	OFF-AR-10003179	Office Supplies	Art	Dixon Ticonderoga Core-Lock Colored Pencils	Anna Andreadi
<input type="checkbox"/>	Изменить	Копировать	Удалить	7	OFF-AP-10003040	Office Supplies	Appliances	Fellowes 8 Outlet Superior Workstation Surge Prote...	Anna Andreadi
<input type="checkbox"/>	Изменить	Копировать	Удалить	8	OFF-BI-10004654	Office Supplies	Binders	VariCap6 Expandable Binder	Cassandra Brandow
<input type="checkbox"/>	Изменить	Копировать	Удалить	9	FUR-CH-10001802	Furniture	Chairs	Hon Every-Day Chair Series Swivel Task Chairs	Anna Andreadi

Таблица клиентов:

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: customers

Обзор

Структура

SQL

Поиск

Вставить

Экспорт

Импорт

Операции

Слежение

Триггеры

✔ Отображение строк 0 - 24 (4910 всего, Запрос занял 0.0003 сек.)

SELECT * FROM `customers`

☐ Профилирование

[Построчное редактирование]

[Изменить]

[Анализ SQL запроса]

[Создать PHP-код]

[Обновить]

1 > >> | Количество строк: 25 | Фильтровать строки: | Сортировать по ключу:

Extra options

↶ ↷

▼ id customer_id customer_name segment country city state postal_code region

☐

✎ Изменить

📄 Копировать

🗑 Удалить

1 CC-12670 Craig Carreira Consumer United States Chicago Illinois 60610 Central

☐

✎ Изменить

📄 Копировать

🗑 Удалить

2 SO-20335 Sean O'Donnell Consumer United States Fort Lauderdale Florida 33311 South

☐

✎ Изменить

📄 Копировать

🗑 Удалить

3 BS-11590 Brendan Sweed Corporate United States Columbus Indiana 47201 Central

☐

✎ Изменить

📄 Копировать

🗑 Удалить

4 RF-19840 Roy Franz-sisch Consumer United States Chesapeake Virginia 23320 South

☐

✎ Изменить

📄 Копировать

🗑 Удалить

5 DR-12880 Dan Reichenbach Corporate United States Inglewood California 90301 West

☐

✎ Изменить

📄 Копировать

🗑 Удалить

6 JE-15745 Joel Eaton Consumer United States Newark Ohio 43055 East

☐

✎ Изменить

📄 Копировать

🗑 Удалить

7 SJ-20215 Sarah Jordon Consumer United States Columbia Tennessee 38401 South

☐

✎ Изменить

📄 Копировать

🗑 Удалить

8 MM-18055 Michelle Moray Consumer United States Aurora Colorado 80013 West

Таблица заказов:

Сервер: localhost:3306 » База данных: mgpu_ico_etl_01 » Таблица: orders

Обзор

Структура

SQL

Поиск

Вставить

Экспорт

Импорт

Операции

Слежение

Триггеры

✔ Отображение строк 0 - 24 (9994 всего, Запрос занял 0.0046 сек.) [returned: 1... - 1...]

SELECT * FROM `orders` ORDER BY `orders`.`returned` DESC

☐ Профилирование

[Построчное редактирование]

[Изменить]

[Анализ SQL запроса]

[Создать PHP-код]

[Обновить]

1

> >>

Количество строк: 25

Фильтровать строки:

Сортировать по ключу:

Ни одного

Extra options

← →

▼

row_id

order_date

ship_date

ship_mode

sales

quantity

discount

profit

returned ▼ 1

☐

Изменить

Копировать

Удалить

139

2018-10-13

2018-10-19

Standard Class

65.88

6

0.00

18.45

1

☐

Изменить

Копировать

Удалить

56

2018-06-17

2018-06-18

First Class

208.56

6

0.00

52.14

1

☐

Изменить

Копировать

Удалить

137

2018-10-13

2018-10-19

Standard Class

4.02

2

0.00

1.97

1

☐

Изменить

Копировать

Удалить

138

2018-10-13

2018-10-19

Standard Class

76.18

3

0.20

26.66

1

☐

Изменить

Копировать

Удалить

203

2016-08-03

2016-08-05

First Class

2.60

1

0.20

0.29

1

☐

Изменить

Копировать

Удалить

57

2018-06-17

2018-06-18

First Class

32.40

5

0.00

15.55

1

☐

Изменить

Копировать

Удалить

21

2016-08-27

2016-09-01

Second Class

22.72

4

0.20

7.38

1

☐

Изменить

Копировать

Удалить

20

2016-08-27

2016-09-01

Second Class

213.48

3

0.20

16.01

1

☐

Изменить

Копировать

Удалить

19

2016-08-27

2016-09-01

Second Class

8.56

2

0.00

2.48

1

☐

Изменить

Копировать

Удалить

91

2018-09-17

2018-09-22

Standard Class

73.58

2

0.20

8.28

1

ВЫВОДЫ

В ходе выполнения данной самостоятельной работы были реализованы все поставленные задачи:

1. была создана исходная таблица в PostgreSQL со сгенерированным набором данных о товарах;
2. были созданы и настроены целевые таблицы в MySQL для приема данных;
3. были созданы процессы трансформации данных в Pentaho, реализованы механизмы фильтрации, группировки и добавления данных;
4. выполнены все индивидуальные задания варианта №1

Таким образом, была достигнута главная цель - разработать ETL-процесс для интеграции данных между PostgreSQL и MySQL с использованием Pentaho Data Integration.

ПРИЛОЖЕНИЯ

1. Скрипт для создания исходной таблицы *products_my* в PostgreSQL:

```
CREATE TABLE products_my (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    category VARCHAR(100),  
    price NUMERIC(10, 2),  
    stock_quantity INT,  
    supplier_id INT  
);
```

2. Скрипт для создания целевой таблицы *target_products* в MySQL:

```
CREATETABLE target_products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    category VARCHAR(100),  
    price DECIMAL(10, 2),  
    stock_quantity INT,  
    supplier_id INT,  
    last_updated DATE  
);
```

3. Скрипт для создания целевой таблицы *price_per_category* в MySQL:

```
CREATE TABLE price_per_category (  
    id SERIAL PRIMARY KEY,  
    category VARCHAR(100) NOT NULL,  
    average_price NUMERIC(10, 2),  
    median_price NUMERIC(10, 2)  
);
```