

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

ДИСЦИПЛИНА:

Проектный практикум по разработке ETL-решений

Самостоятельная работа

Тема:

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

Выполнил(а): Морозова Валерия АДЭУ-211

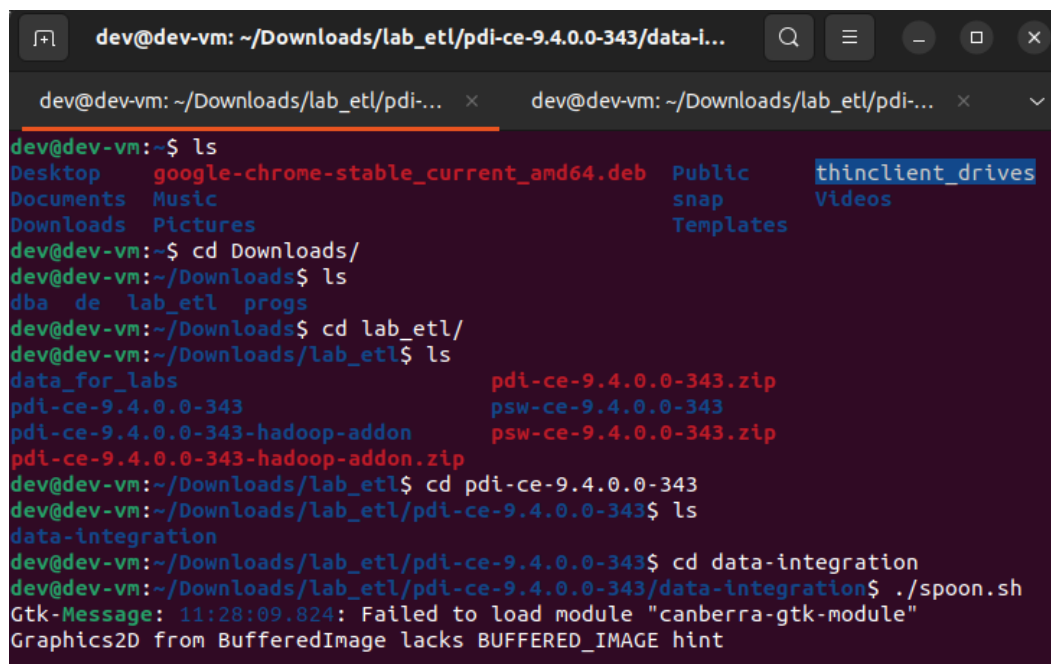
Преподаватель:

Москва

2025

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.



```
dev@dev-vm: ~/Downloads/lab_etl/pdi-ce-9.4.0.0-343/data-i...
dev@dev-vm: ~/Downloads/lab_etl/pdi-... x dev@dev-vm: ~/Downloads/lab_etl/pdi-... x
dev@dev-vm:~$ ls
Desktop  google-chrome-stable_current_amd64.deb  Public  thinclient_drives
Documents Music                               snap    Videos
Downloads Pictures                               Templates
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$ 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
Gtk-Message: 11:28:09.824: Failed to load module "canberra-gtk-module"
Graphics2D from BufferedImage lacks BUFFERED_IMAGE hint
```

Рисунок 1. Запуск Pentaho Data Integration

```
1 services:
2   mongo-1:
3     image: mongo:7.0.17-rc1-jammy
4     container_name: mongo-1
5     environment:
6       - MONGO_INITDB_ROOT_USERNAME=root
7       - MONGO_INITDB_ROOT_PASSWORD=abc123!
8     volumes:
9       - mongo-data:/data/db
10    ports:
11      - "27017:27017"
12    networks:
13      - mongo-net
14
15 express:
```

```
dev@dev-vm:~$ cd dba
bash: cd: dba: No such file or directory
dev@dev-vm:~$ cd dba/
bash: cd: dba/: No such file or directory
dev@dev-vm:~$ cd Downloads/
dev@dev-vm:~/Downloads$ cd dba/
dev@dev-vm:~/Downloads/dba$ cd nonrel/
dev@dev-vm:~/Downloads/dba/nonrel$ cd mongo/
dev@dev-vm:~/Downloads/dba/nonrel/mongo$ sudo docker compose stop
[sudo] password for dev:
[+] Stopping 2/2
  ✓ Container express-app   Stopped
  ✓ Container mongo-1      Stopped
dev@dev-vm:~/Downloads/dba/nonrel/mongo$
```

Рисунок 2. Отключение mongo

```
80 # Проверка портов
81 # sudo docker compose logs
82
83 # Проверка пользователей и баз
84 # sudo docker compose exec postgres psql -U postgres -c "\du"
85 # sudo docker compose exec postgres psql -U postgres -c "\l"
86
87 # Остановка
88 # sudo docker compose stop
89
90 # Запуск
91 # sudo docker compose start
92
93 # Перезапуск
94 # sudo docker compose restart
```

```
dev@dev-vm:~/Downloads/dba/rel/postgresql$
* History restored

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

Рисунок 3. Отключение и запуск postgresQL

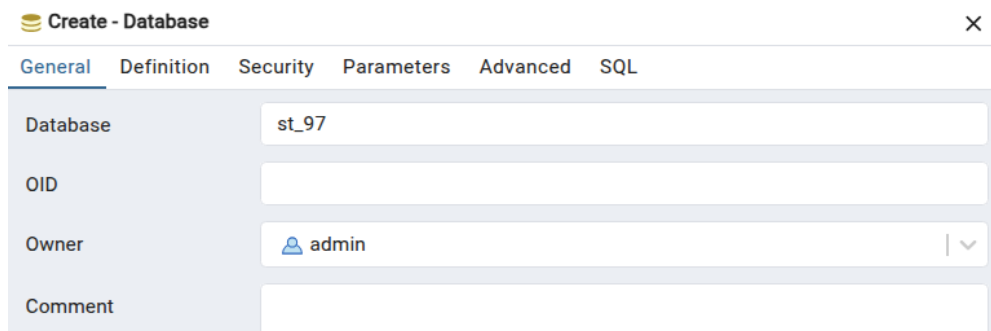


Рисунок 4. Создание собственной базы

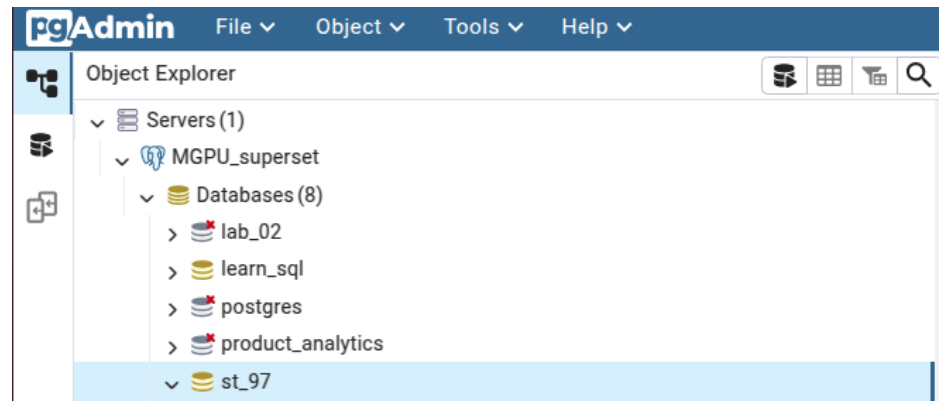


Рисунок 5. База st_97 успешно создана

Вариант10

Задание 1. Создать таблицу payments (id, customer_id, amount, date, method)

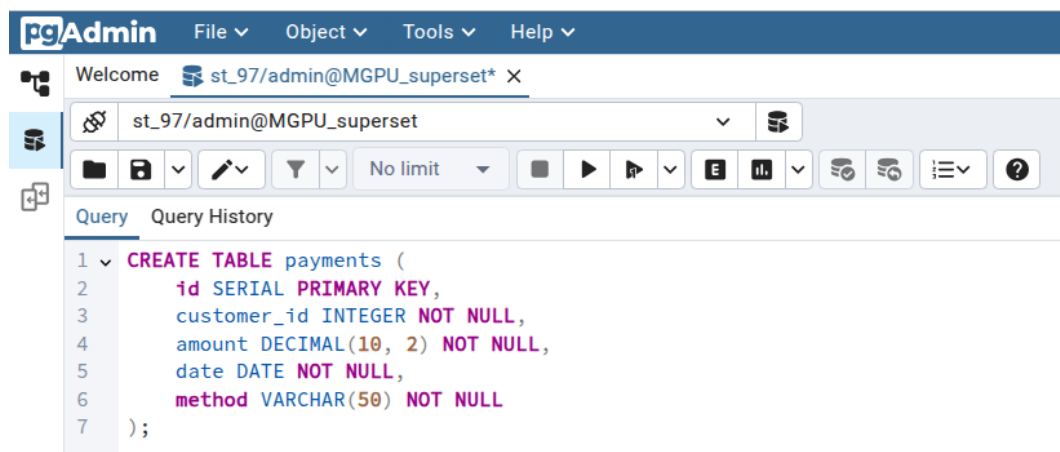


Рисунок 6. Запрос на создание таблицы в postgresSQL

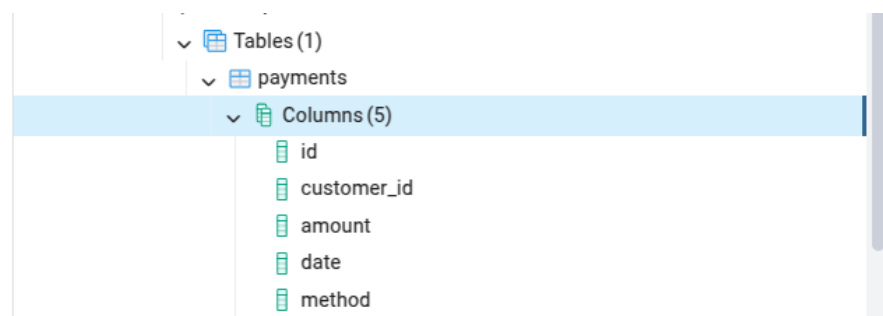


Рисунок 7. Проверка таблицы payments в базе

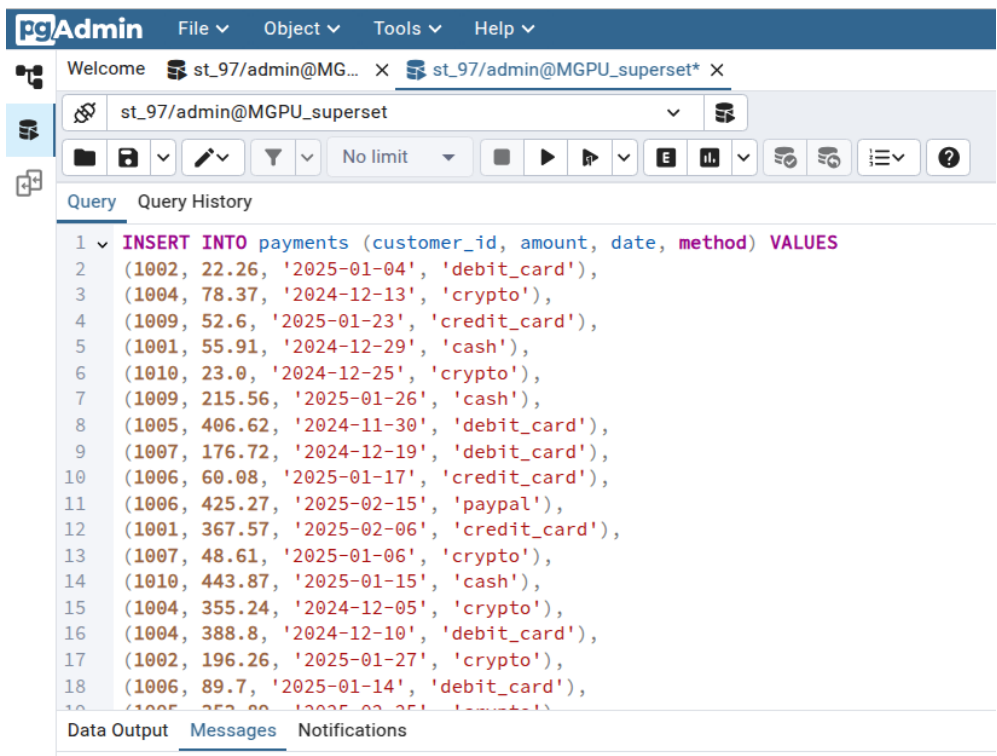


Рисунок 8. Заполнение таблицы сгенерированными данными

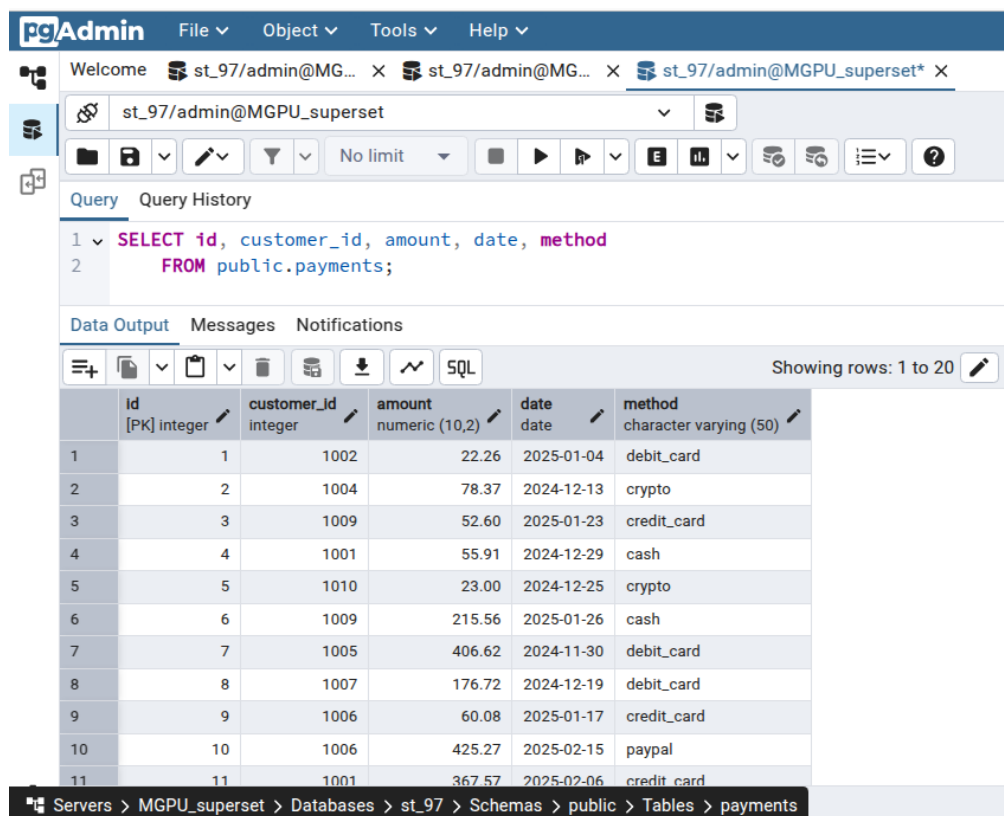


Рисунок 9. Генерация запроса для проверки содержания таблицы

Задание 2. Создать таблицу payment_analytics с полями для анализа транзакций

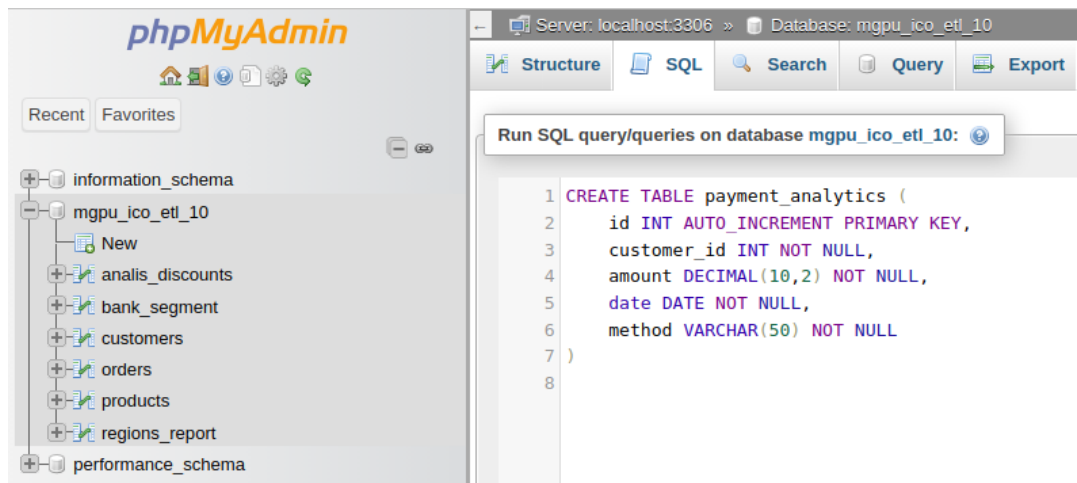


Рисунок 10. Создание таблицы через запрос в MySQL

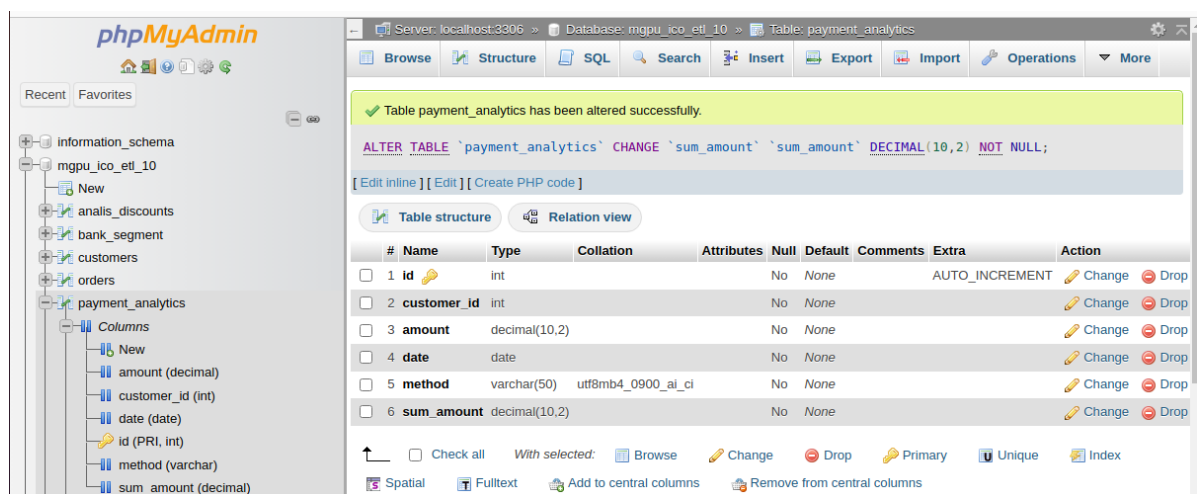


Рисунок 11. Добавлен столбец для группировки по сумме продаж



Рисунок 12. Импорт данных из PostgreSQL

Examine preview data					
Rows of step: Table input PostgreSQL (20 rows)					
▲	id	customer_id	amount	date	method
1	1	1002	22.26	2025/01/04 00:00:00.000	debit_card
2	2	1004	78.37	2024/12/13 00:00:00.000	crypto
3	3	1009	52.6	2025/01/23 00:00:00.000	credit_card
4	4	1001	55.91	2024/12/29 00:00:00.000	cash
5	5	1010	23.0	2024/12/25 00:00:00.000	crypto
6	6	1009	215.56	2025/01/26 00:00:00.000	cash
7	7	1005	406.62	2024/11/30 00:00:00.000	debit_card
8	8	1007	176.72	2024/12/19 00:00:00.000	debit_card
9	9	1006	60.08	2025/01/17 00:00:00.000	credit_card
10	10	1006	425.27	2025/02/15 00:00:00.000	paypal
11	11	1001	367.57	2025/02/06 00:00:00.000	credit_card
12	12	1007	48.61	2025/01/06 00:00:00.000	crypto
13	13	1010	443.87	2025/01/15 00:00:00.000	cash
14	14	1004	355.24	2024/12/05 00:00:00.000	crypto

Рисунок 13. Предпросмотр данных

Select values

Step name

Select values

Select & Alter

Remove

Meta-data

Fields :

▲	Fieldname	Rename to	Length	Precision
1	id			
2	customer_id			
3	amount			
4	date			
5	method			

Get fields to select

Edit Mapping

Рисунок 14. Выбор нужных столбцов

Задание 3. Фильтр платежей по методу

Filter rows

Step name

Filter rows

Send 'true' data to step:

Memory group by

Send 'false' data to step:

Write to log 2

The condition:

method <> [crypto]

Рисунок 15. Фильтр платежей по методу

Загружаются все методы кроме криптовалютного.

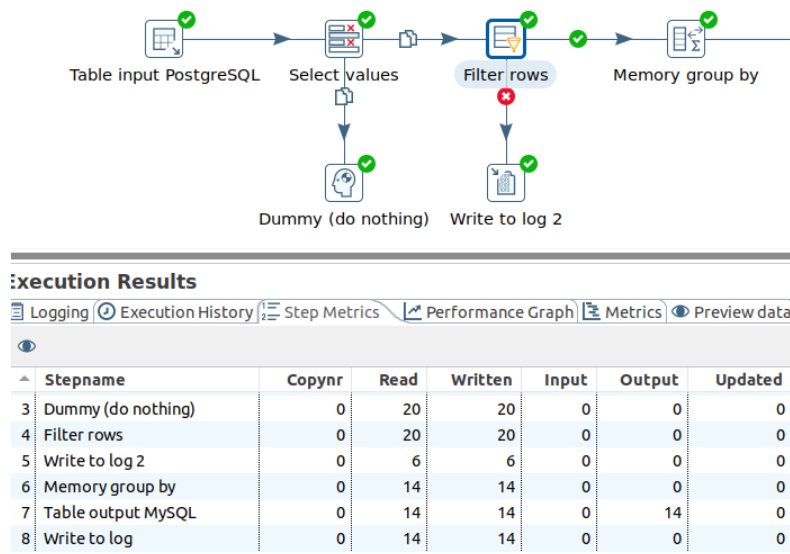


Рисунок 16. Результат фильтрации

Как видно, они отфильтрованы и из 20 в базу выгружены 14.

Задание 4. Суммы платежей по периодам

Memory group by

Step name

☐ Always give back a result row

The fields that make up the group:

Group field

1 id

2 customer_id

3 amount

4 date

5 method

Get Fields

Aggregates :

Name	Subject	Type
1 sum_amount	amount	Sum

Get lookup fields

Рисунок 17. Расчет суммы платежей

Table output

Step name: **Table output MySQL**

Connection: MySQL

Target schema: mgpu_ico_etl_10

Target table: payment_analytics

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options / Database fields

Fields to insert:

Table field	Stream field
1 id	id
2 customer_id	customer_id
3 amount	amount
4 date	date
5 method	method
6 sum_amount	sum_amount

Get fields

Enter field mapping

Рисунок 18. Настройка коннектора для выгрузки данных в таблицу MySQL

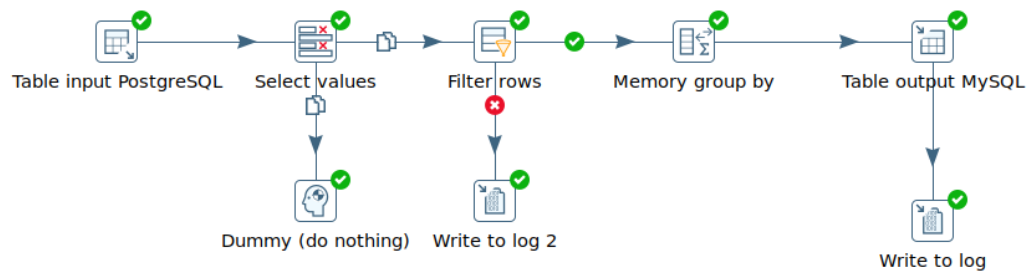


Рисунок 19. Схема итоговой трансформации

information_schema
mgpu_ico_etl_10
New
analis_discounts
bank_segment
customers
orders
payment_analytics
Columns
New
amount (decimal)
customer_id (int)
date (date)
id (PRI, int)
method (varchar)
sum_amount (decimal)

SELECT * FROM `payment_analytics`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key:

Extra options

	id	customer_id	amount	date	method	sum_amount
<input type="checkbox"/> Edit Copy Delete	1	1002	22.26	2025-01-04	debit_card	22.26
<input type="checkbox"/> Edit Copy Delete	3	1009	52.60	2025-01-23	credit_card	52.60
<input type="checkbox"/> Edit Copy Delete	4	1001	55.91	2024-12-29	cash	55.91
<input type="checkbox"/> Edit Copy Delete	6	1009	215.56	2025-01-26	cash	215.56
<input type="checkbox"/> Edit Copy Delete	7	1005	406.62	2024-11-30	debit_card	406.62
<input type="checkbox"/> Edit Copy Delete	8	1007	176.72	2024-12-19	debit_card	176.72
<input type="checkbox"/> Edit Copy Delete	9	1006	60.08	2025-01-17	credit card	60.08

Рисунок 20. Данные успешно выгружены и загружены в базу

Transformation

Entry Name:

Postgre to Mysql

Transformation:

\${Internal.Entry.Current.Directory}/Postgre_to_M... Browse...

Options Logging Arguments Parameters

Run configuration:

Pentaho local

Рисунок 21. Подключение трансформации в job

Showing rows 0 - 12 (13 total, Query took 0.0003 seconds.) [date: 2024-11-30... - 2025-02-15...]

```
SELECT date, AVG(amount) as avg_amount FROM payment_analytics GROUP BY date
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

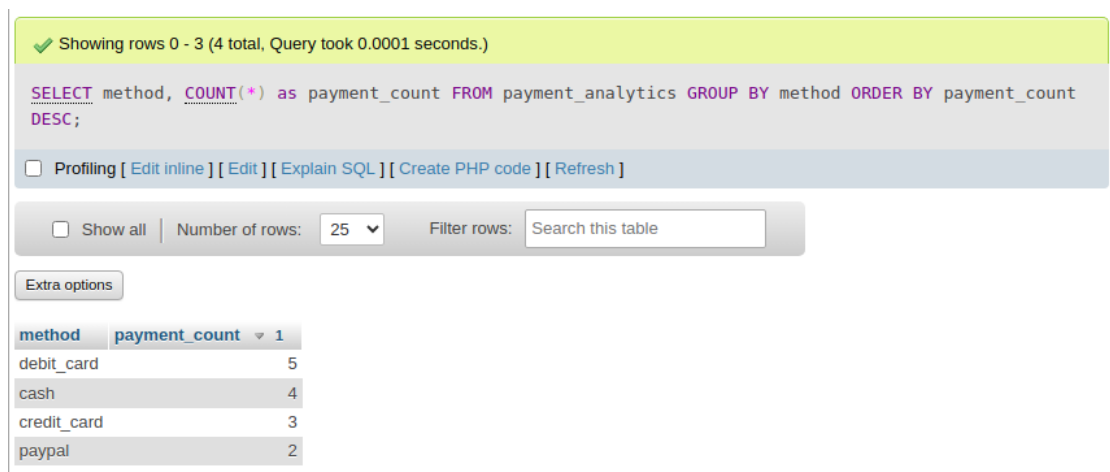
☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

					date	avg_amount
<input type="checkbox"/>	Edit	Copy	Delete		2024-11-30	406.620000
<input type="checkbox"/>	Edit	Copy	Delete		2024-12-10	388.800000
<input type="checkbox"/>	Edit	Copy	Delete		2024-12-19	176.720000
<input type="checkbox"/>	Edit	Copy	Delete		2024-12-21	308.470000
<input type="checkbox"/>	Edit	Copy	Delete		2024-12-29	55.910000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-04	22.260000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-14	89.700000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-15	443.870000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-17	75.075000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-23	52.600000
<input type="checkbox"/>	Edit	Copy	Delete		2025-01-26	215.560000
<input type="checkbox"/>	Edit	Copy	Delete		2025-02-06	367.570000
<input type="checkbox"/>	Edit	Copy	Delete		2025-02-15	425.270000

Рисунок 22. Анализ платежей по дням

Задание 5. Анализ популярных методов



The screenshot shows a database query interface. At the top, a green status bar indicates 'Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)'. Below this, the SQL query is displayed: `SELECT method, COUNT(*) as payment_count FROM payment_analytics GROUP BY method ORDER BY payment_count DESC;`. A toolbar below the query includes a 'Profiling' checkbox and links for 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. Below the toolbar, there are controls for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. An 'Extra options' button is also present. The query results are shown in a table with two columns: 'method' and 'payment_count'. The results are ordered by payment count in descending order.

method	payment_count
debit_card	5
cash	4
credit_card	3
paypal	2

Рисунок 23. Количество платежей по методам оплаты

Вывод:

Созданы исходные таблицы в PostgreSQL с различными наборами данных

Настроены целевые таблицы в MySQL для приема данных.

Разработаны процессы трансформации данных в Pentaho.

Реализованы механизмы обработки ошибок и валидации данных.

Созданы представления для связанных данных.

По результату анализа видно, что больше всего платежей совершают дебетовыми картами.