

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

## **ОТЧЕТ**

по дисциплине «Проектный практикум по разработке ETL-решений»  
Лабораторная работа 1.1.: Установка и настройка ETL-инструмента.  
Создание конвейеров данных  
Направление подготовки – 38.03.05 «Бизнес-информатика».  
профиль подготовки – «Аналитика данных и эффективное управление»

**Выполнила:**

студентка группы АДЭУ-211  
st92

---

**Руководитель:**

Кандидат технических наук, доцент

---

Москва  
2025 год

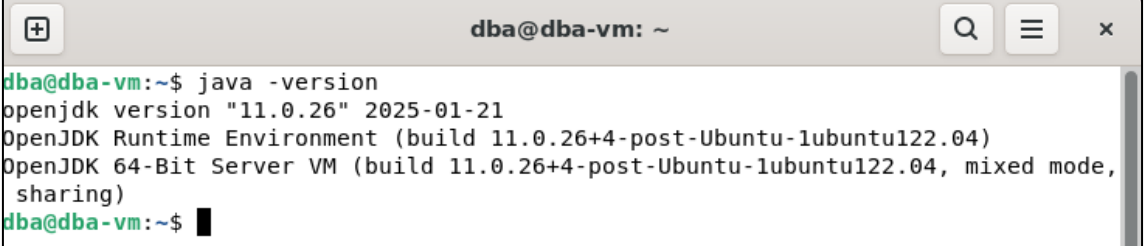
**Цель работы:** изучение основных принципов работы с ETL-инструментами на примере Pentaho Data Integration (PDI), настройка конвейера обработки данных, фильтрация и замена значений в Excel-файле, а также выгрузка обработанных данных в базу данных MySQL/PostgreSQL.

### Задачи

1. Настроить среду для работы с Pentaho Data Integration (PDI):
  - a. Запуск виртуальной машины с Ubuntu 22.04 в VirtualBox.
  - b. Проверка установки Java и WebKitGTK.
  - c. Развертывание Pentaho Data Integration.
2. Создать ETL-конвейер:
  - a. Загрузить данные из CSV-файла.
  - b. Очистить, преобразовать и отфильтровать данные.
  - c. Выполнить замену значений.
  - d. Выгрузить обработанные данные в MySQL или PostgreSQL.
3. Проверить корректность обработки:
  - a. Выполнить SQL-запросы для проверки результата.
  - b. Подготовить отчет с описанием проделанных шагов

### Ход работы:

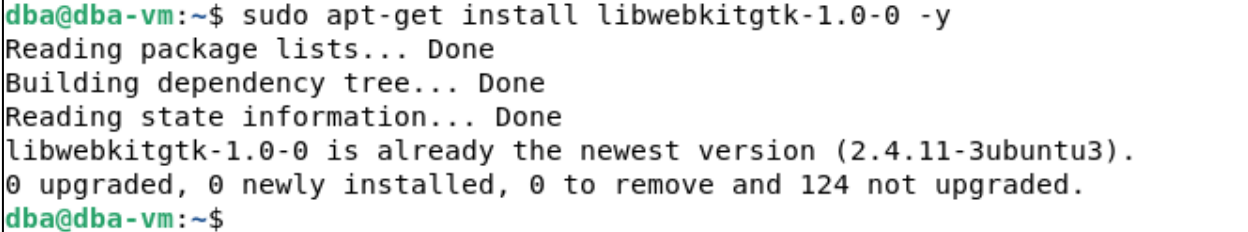
#### Проверка установки Java



```
dba@dba-vm: ~  
dba@dba-vm:~$ java -version  
openjdk version "11.0.26" 2025-01-21  
OpenJDK Runtime Environment (build 11.0.26+4-post-Ubuntu-1ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 11.0.26+4-post-Ubuntu-1ubuntu122.04, mixed mode,  
sharing)  
dba@dba-vm:~$
```

Рисунок 1 – Выполнение команды просмотра версии Java

#### Проверка установки WebKitGTK



```
dba@dba-vm:~$ sudo apt-get install libwebkitgtk-1.0-0 -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
libwebkitgtk-1.0-0 is already the newest version (2.4.11-3ubuntu3).  
0 upgraded, 0 newly installed, 0 to remove and 124 not upgraded.  
dba@dba-vm:~$
```

Рисунок 2 – Выполнение команды загрузки webkitgtk

## Развернем Pentaho Data Integration

```
dba@dba-vm:~$ cd ~/Downloads/data-integration/  
dba@dba-vm:~/Downloads/data-integration$ chmod +x spoon.sh  
dba@dba-vm:~/Downloads/data-integration$ ./spoon.sh  
Gtk-Message: 16:34:14.449: Failed to load module "canberra-gtk-module"
```

Рисунок 3 – Выполнение команды для развертывания Pentaho Data Integration

Создадим Трансформацию и добавим в рабочую область CSV file input:

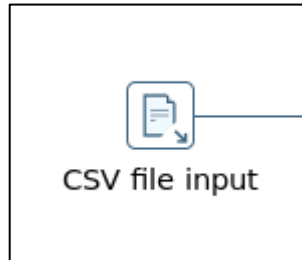


Рисунок 4 – Добавление объекта CSV file input

Укажем путь к файлу, который был предварительно загружен из Kaggle

The screenshot shows the 'CSV file input' configuration window. The 'Step name' is 'CSV file input'. The 'Filename' is '\$(Internal.Entry.Current.Directory)/supply\_chain\_data.csv'. The 'Delimiter' is ','. The 'Enclosure' is '"'. The 'NIO buffer size' is '50000'. The 'Lazy conversion?' checkbox is checked. The 'Header row present?' checkbox is checked. The 'Add filename to result' checkbox is unchecked. The 'The row number field name (optional)' field is empty. The 'Running in parallel?' checkbox is unchecked. The 'New line possible in fields?' checkbox is unchecked. The 'Format' is 'mixed'. The 'File encoding' is 'UTF-8'. Below the configuration fields is a table with 8 columns: Name, Type, Format, Length, Precision, Currency, Decimal, Group, and Trim type. The table contains 6 rows of data.

Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1 Product type	String		9		\$	.	,	none
2 SKU	String		5		\$	.	,	none
3 Price	Number	#,.	18	16	\$	.	,	none
4 Availability	Integer	#	15	0	\$	.	,	none
5 Number of products sold	Integer	#	15	0	\$	.	,	none
6 Revenue generated	Number	#,.	18	13	\$	.	,	none

Рисунок 5 – Настройки объекта CSV file input

Очистим, преобразуем и отфильтруем данные

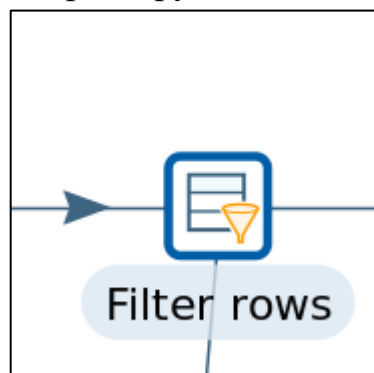


Рисунок 6 – Добавление объекта Filter rows

Очистим нулевые значения в столбцах Product type, Shipping times, Lead times

Filter rows

Step name: Filter rows

Send 'true' data to step: Value mapper

Send 'false' data to step: Write to log

The condition:

```
(  
  OR  
  OR  
  )  
AND  
  Product type IS NULL  
  Shipping times IS NULL  
  Lead times IS NULL
```

Buttons: Help, OK, Cancel

Рисунок 7 – Настройки объекта Filter rows

Реализуем трансформацию полей Customer demographics для стандартизации

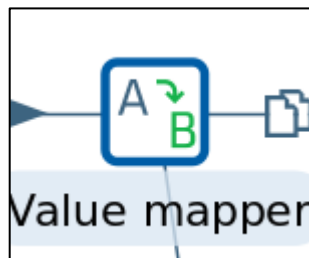


Рисунок 8 – Добавление объекта Value mapper

Выполним соответствующие настройки

Value mapper

Step name : value mapper

Fieldname to use : Customer demographics

Target field name (empty=overwrite) :

Default upon non-matching :

Field values:

	Source value	Target value
1	Female	F
2	Male	M
3	Unknown	U
4	Non-binary	NB

Buttons: Help, OK, Cancel

Рисунок 9 – Настройки объекта Value mapper

Рассчитаем общие затраты и прибыль

Добавим первый калькулятор с данными параметрами

Step name

Calculator

☐ Throw an error on non existing files

Fields:

	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	CostsAll	A + B + C	Manufacturing costs	Shipping costs	Costs	None			N

Help OK Cancel

Рисунок 10 – Калькулятор расчета общих затрат

Добавим второй калькулятор с данными параметрами

Step name

Calculator 2

☒ Throw an error on non existing files

Fields:

	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Color
1	Profit	A - B	Revenue generated	CostsAll		None			N	

Help OK Cancel

Рисунок 11 – Калькулятор расчета прибыли

Создадим таблицу в базе данных с помощью скрипта:

```
CREATE TABLE chain_data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    product_type VARCHAR(20) NOT NULL,
    SKU VARCHAR(10),
    price DECIMAL(10,2),
    availability DECIMAL(10,2),
    number_of_products_sold DECIMAL(10,2),
```

```
revenue_generated DECIMAL(10,2),
customer_demographics VARCHAR(2),
stock_levels DECIMAL(10,2),
lead_times DECIMAL(10,2),
order_quantities DECIMAL(10,2),
shipping_times DECIMAL(10,2),
shipping_carries VARCHAR(20),
shipping_costs DECIMAL(10,2),
supplier_name VARCHAR(50),
location VARCHAR(30),
lead_time DECIMAL(10,2),
production_volumes DECIMAL(10,2),
manufacturing_lead_time DECIMAL(10,2),
manufacturing_costs DECIMAL(10,2),
inspection_results VARCHAR(20),
defect_rates DECIMAL(10,2),
transportation_modes VARCHAR(20),
routes VARCHAR(20),
costs DECIMAL(10,2),
profit DECIMAL(10,2)
);
```

Добавим объект Select values и переименуем столбцы в соответствии со значениями столбцов из таблицы

**Select values** [X]

Step name:

Select & Alter Remove Meta-data

Fields :

	Fieldname	Rename to	Length
1	Product type	product_type	
2	SKU		
3	Price	price	
4	Availability	availability	
5	Number of products sold	number_of_products_sold	
6	Revenue generated	revenue_generated	
7	Customer demographics	customer_demographics	
8	Stock levels	stock_levels	
9	Lead times	lead_times	
10	Order quantities	order_quantities	
11	Shipping times	shipping_times	
12	Shipping carriers	shipping_carriers	
13	Shipping costs	shipping_costs	
14	Supplier name	supplier_name	
15	Location	location	
16	Lead time	lead_time	
17	Production volumes	production_volumes	
18	Manufacturing lead time	manufacturing_lead_time	
19	Manufacturing costs	manufacturing_costs	
20	Inspection results	inspection_results	
21	Defect rates	defect_rates	
22	Transportation modes	transportation_modes	
23	Routes	routes	
24	Costs	costs	
25	Profit	profit	

Get fields to select

Edit Mapping

Рисунок 12 – Настройка объекта Select values

Добавим объект Table output для выгрузки таблицы в базу данных

**Table output** [X]

Step name:

Connection:  Edit... New... Wizard...

Target schema:  Browse...

Target table:  Browse...

Commit size:

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

	Table field	Stream field
1	product_type	product_type
2	SKU	SKU
3	Price	Price
4	Availability	Availability
5	number_of_p	number_of_products_sold
6	revenue_gen	revenue_generated
7	customer_de	customer_demographics
8	stock_levels	stock_levels
9	lead_times	lead_times

Get fields

Enter field mapping

Help OK Cancel SQL

Рисунок 13 – Настройки объекта Table output

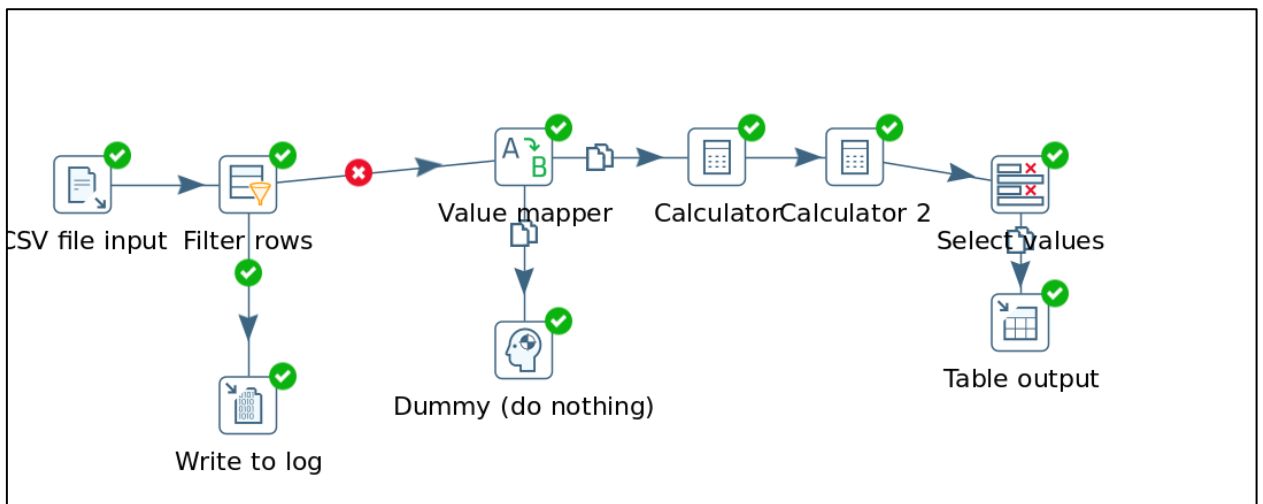


Рисунок 14 – Рабочая область Pentaho

Убедимся, что таблица успешно загружена в базу данных

id	product_type	SKU	price	availability	number_of_products_sold	revenue_generated	customer_demographics	stock_levels	lead_times	order_quantities	shipping_times	shipping_carrier
1	haircare	SKU0	69.81	55.00	802.00	8662.00	NB	50.00	7.00	96.00	4.00	Carrier B
2	skincare	SKU1	14.84	95.00	736.00	7460.90	F	53.00	30.00	37.00	2.00	Carrier A
3	haircare	SKU2	11.32	34.00	8.00	9577.75	U	1.00	10.00	88.00	2.00	Carrier B
4	skincare	SKU3	61.16	68.00	83.00	7766.84	NB	23.00	13.00	59.00	6.00	Carrier C
5	skincare	SKU4	4.81	26.00	871.00	2686.51	NB	5.00	3.00	56.00	8.00	Carrier A
6	haircare	SKU5	1.70	87.00	147.00	2628.35	NB	90.00	27.00	66.00	3.00	Carrier B
7	skincare	SKU6	4.08	48.00	65.00	7823.48	M	11.00	15.00	58.00	8.00	Carrier C
8	cosmetics	SKU7	42.96	59.00	426.00	8496.10	F	93.00	17.00	11.00	1.00	Carrier B
9	cosmetics	SKU8	68.72	78.00	150.00	7517.36	F	5.00	10.00	15.00	7.00	Carrier C
10	skincare	SKU9	64.02	35.00	900.00	4971.15	U	14.00	27.00	83.00	1.00	Carrier A

Рисунок 15 – Таблица chain\_data

## Выводы:

В ходе работы мы изучили основные принципы работы с ETL-инструментами на примере Pentaho Data Integration, настроили конвейера обработки данных, фильтрацию и замену значений в Excel-файле, а также выгрузили обработанные данные в базу данных MySQL/PostgreSQL. Задачи работы выполнены, а цель достигнута.