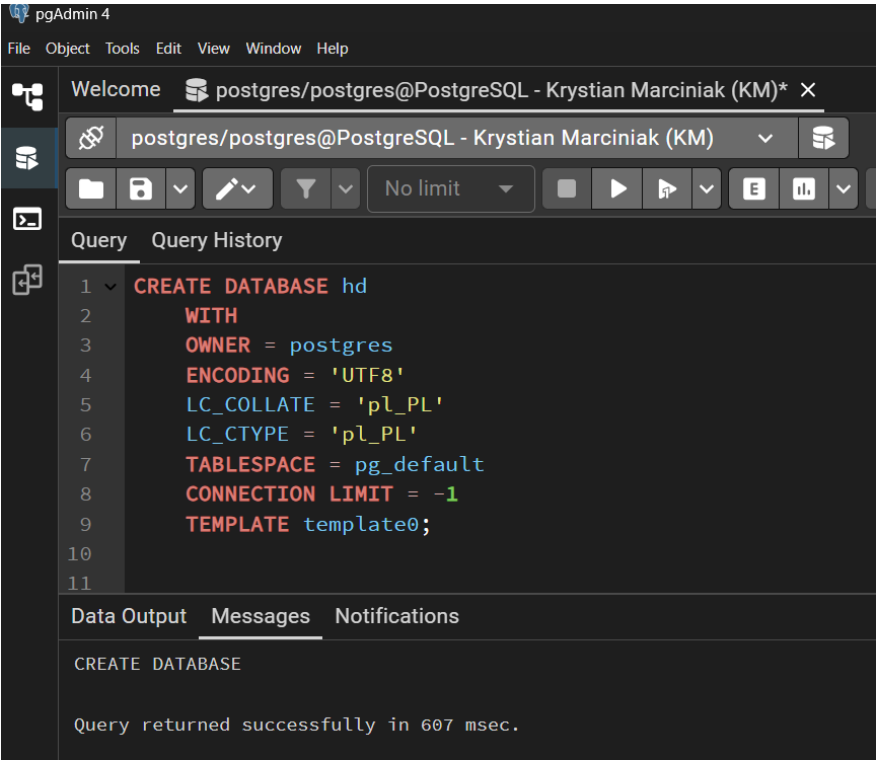


ETAP I – Projektowanie Schematów (Modelowanie)

1. Utwórz pustą bazę danych HD w PostgreSQL.

CREATE DATABASE hurtownia  
WITH  
OWNER = postgres  
ENCODING = 'UTF8'  
LC\_COLLATE = 'pl\_PL'  
LC\_CTYPE = 'pl\_PL'  
TABLESPACE = pg\_default  
CONNECTION LIMIT = -1  
TEMPLATE template0;


































The screenshot shows the pgAdmin 4 interface. The top bar indicates the user is 'postgres/postgres@PostgreSQL - Krystian Marciniak (KM)'. The main pane displays a SQL query: `CREATE DATABASE hd WITH OWNER = postgres ENCODING = 'UTF8' LC_COLLATE = 'pl_PL' LC_CTYPE = 'pl_PL' TABLESPACE = pg_default CONNECTION LIMIT = -1 TEMPLATE template0;`. The 'Messages' tab at the bottom shows the confirmation: 'CREATE DATABASE' and 'Query returned successfully in 607 msec.'

| Składniki polecenia i ich znaczenie: |   |
|--------------------------------------|---|
| CREATE DATABASE hurtownia            | Tworzy nową bazę danych o nazwie <b>hurtownia</b> .   |
| WITH                                 | Wprowadza opcje konfiguracyjne dla tworzonej bazy.  |
| OWNER = postgres                     | Określa właściciela bazy danych - użytkownika PostgreSQL, który będzie miał pełne uprawnienia (np. <b>postgres</b> ). |
| ENCODING = 'UTF8'                    | Ustawia kodowanie znaków w bazie danych. UTF8 pozwala przechowywać znaki z różnych języków (w tym polskie litery).    |
| LC_COLLATE = 'pl_PL'                 | Określa reguły sortowania tekstu według języka polskiego (np. <b>litera "ś" po "s"</b> ).                             |
| LC_CTYPE = 'pl_PL'                   | Określa <b>klasyfikację znaków</b> (czy znak to litera, wielka litera, cyfra itd.), też wg języka polskiego.          |
| TABLESPACE = pg_default              | Określa domyślne miejsce przechowywania danych na dysku – tu pg_default, czyli standardowy katalog.                   |
| CONNECTION LIMIT = -1                | Oznacza <b>brak ograniczenia liczby połączeń</b> do tej bazy danych (może się łączyć dowolna liczba użytkowników).    |
| TEMPLATE template0                   | Tworzy bazę <b>na podstawie czystego szablonu</b> template0, który umożliwia zmianę LC_COLLATE i LC_CTYPE.            |

1. Przeanalizuj plik CSV i zaprojektuj architekturę gwiazdy (Star Schema) obejmującą wszystkie dane:
- a Wyodrębnij tabelę faktów (np. sprzedaż, zamówienia) oraz tabele wymiarów (np. produkt, klient, czas, region).
- Tabele wymiarów (z kluczami sztucznymi - surrogate keys)
3. Dla tabel wymiarów stosuj sztuczne klucze (surrogate keys) typu INT lub SERIAL.

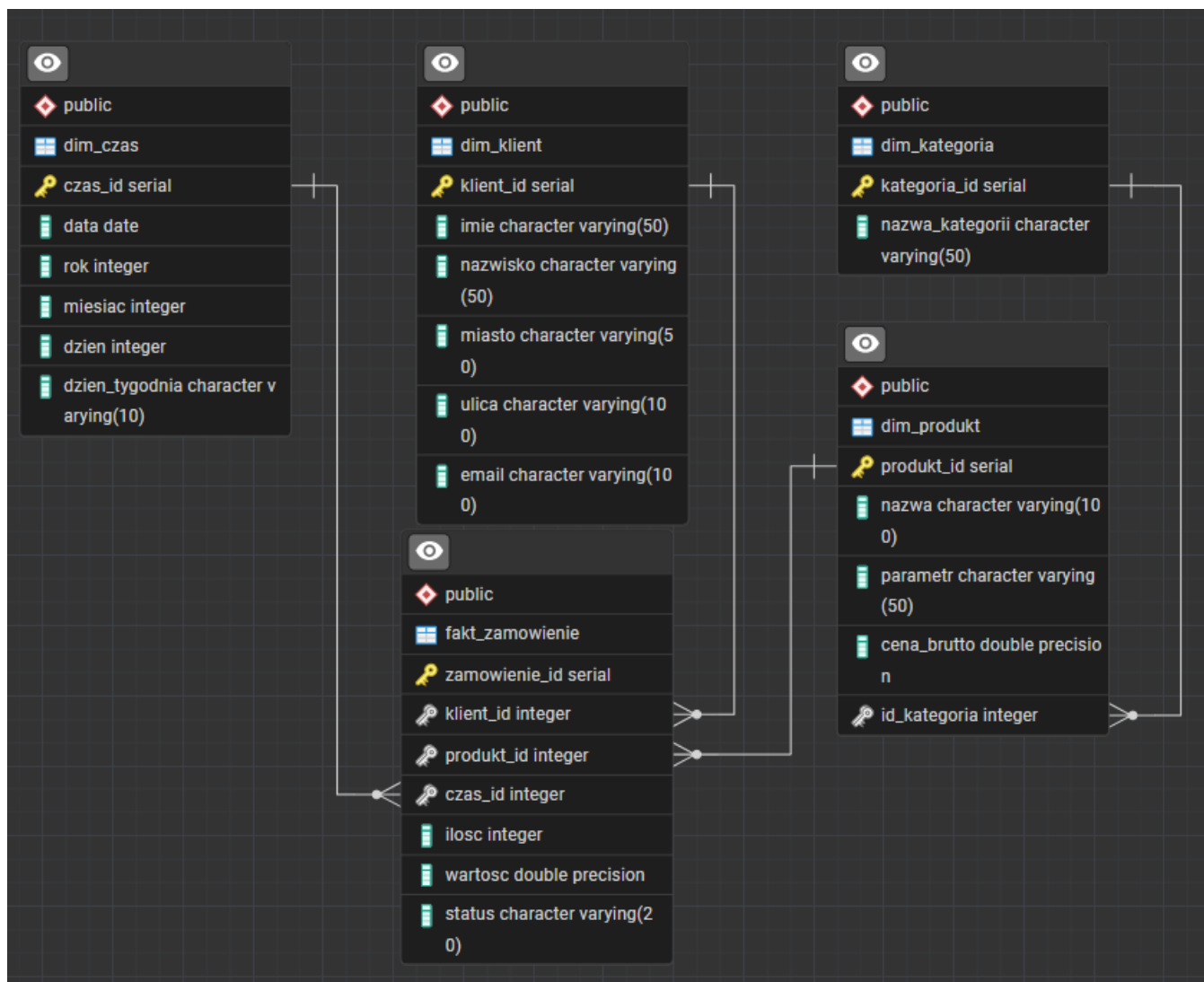
| -- Tabele wymiarów (z kluczami sztucznymi - surrogate keys)  |  |
|--|--|
| CREATE TABLE dim_klient (<br>klient_id SERIAL PRIMARY KEY,<br>imie VARCHAR(50),<br>nazwisko VARCHAR(50),<br>miasto VARCHAR(50),<br>ulica VARCHAR(100),<br>email VARCHAR(100)<br>); | <pre>CREATE TABLE dim_klient (<br/>  klient_id SERIAL PRIMARY KEY,<br/>  imie VARCHAR(50),<br/>  nazwisko VARCHAR(50),<br/>  miasto VARCHAR(50),<br/>  ulica VARCHAR(100),<br/>  email VARCHAR(100)<br/>);</pre> |
| CREATE TABLE dim_produkt (<br>produkt_id SERIAL PRIMARY KEY,<br>nazwa VARCHAR(100),<br>parametr VARCHAR(50),<br>cena_brutto FLOAT,<br>id_kategoria INT<br>);                       | <pre>1 CREATE TABLE dim_produkt (<br/>2   produkt_id SERIAL PRIMARY KEY,<br/>3   nazwa VARCHAR(100),<br/>4   parametr VARCHAR(50),<br/>5   cena_brutto FLOAT,<br/>6   id_kategoria INT<br/>7 );</pre>            |
| CREATE TABLE dim_kategoria (<br>kategoria_id SERIAL PRIMARY KEY,<br>nazwa_kategorii VARCHAR(50)<br>);  | <pre>1 CREATE TABLE dim_kategoria (<br/>2   kategoria_id SERIAL PRIMARY KEY,<br/>3   nazwa_kategorii VARCHAR(50)<br/>4 );</pre>  |
| CREATE TABLE dim_czas (<br>czas_id SERIAL PRIMARY KEY,<br>data DATE,<br>rok INT,<br>miesiac INT,<br>dzien INT,<br>dzien_tygodnia VARCHAR(10)<br>);                                 | <pre>1 CREATE TABLE dim_czas (<br/>2   czas_id SERIAL PRIMARY KEY,<br/>3   data DATE,<br/>4   rok INT,<br/>5   miesiac INT,<br/>6   dzien INT,<br/>7   dzien_tygodnia VARCHAR(10)<br/>8 );</pre>                 |

|  |  |   |  |
|--|--|---|--|
|                                   |                                   |                    |   |
|  public                          |  public                          |  public           |  public                                |
|  dim_klient                     |  dim_produk                     |  dim_kategoria   |  dim_czas                             |
|  klient_id serial               |  produkt_id serial              |  kategoria_id    |  czas_id serial                       |
|  imie character varying(50)     |  nazwa character varying(100)   |  nazwa_kategorii |  data date                            |
|  nazwisko character varying(50) |  parametr character varying(50) |   |  rok integer                          |
|  miasto character varying(50)   |  cena_brutto double precision   |   |  miesiac integer                      |
|  ulica character varying(100)   |  id_kategoria integer           |   |  dzien integer                        |
|  email character varying(100)   |  |   |  dzien_tygodnia character varying(10) |

**SERIAL (typ pseudodanych) -> SERIAL** to wygodna konstrukcja PostgreSQL, która:

- automatycznie tworzy **sekwencję** (SEQUENCE)
- przypisuje ją do kolumny jako DEFAULT nextval(...)
- działa jak auto\_increment w MySQL

| -- Tabela faktów (zamówienia)   |   |  |
|---|---|--|
| CREATE TABLE fakt_zamowienie<br>(<br>zamowienie_id SERIAL<br>PRIMARY KEY,<br>klient_id INT REFERENCES<br>dim_klient(klient_id),<br>produkt_id INT REFERENCES<br>dim_produk(produkt_id),<br>czas_id INT REFERENCES<br>dim_czas(czas_id),<br>ilosc INT NOT NULL,<br>wartosc FLOAT NOT NULL,<br>status VARCHAR(20)<br>); | <pre>CREATE TABLE fakt_zamowienie (<br/>  zamowienie_id SERIAL PRIMARY KEY,<br/>  klient_id INT REFERENCES dim_klient(klient_id),<br/>  produkt_id INT REFERENCES dim_produk(produkt_id),<br/>  czas_id INT REFERENCES dim_czas(czas_id),<br/>  ilosc INT NOT NULL,<br/>  wartosc FLOAT NOT NULL,<br/>  status VARCHAR(20)<br/>);</pre> |  |
| ALTER TABLE dim_produk<br>ADD FOREIGN KEY<br>(id_kategoria)<br>REFERENCES<br>dim_kategoria(kategoria_id);   | <pre>ALTER TABLE dim_produk<br/>  ADD FOREIGN KEY (id_kategoria)<br/>  REFERENCES dim_kategoria(kategoria_id);</pre>  |  |



A.

### Kolejność tworzenia tabel – problem zależności

Jeśli utworzone zostało dim\_produkty przed dim\_kategoria, wtedy odwołanie do dim\_kategoria(kategoria\_id) jeszcze nie zadziało - bo ta tabela nie istnieje. Rozdzielenie deklaracji pozwala:

- utworzyć najpierw wszystkie tabele
- potem dodać zależności między nimi

B. Czystość modelowania -> Projektanci czasem oddzielają strukturę tabeli od relacji logicznych, aby zachować przejrzystość:

CREATE TABLE -> tylko struktura i kolumny

ALTER TABLE -> relacje między tabelami (czyli logika)

C. Etapowe budowanie schematu

W projektach z kilkoma etapami (np. najpierw modelowanie, potem implementacja relacji) często zostawia się FOREIGN KEY na później

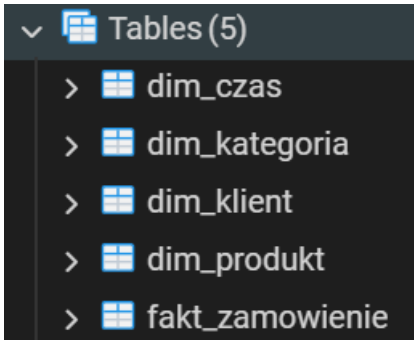
REFERENCES definiuje klucz obcy (foreign key)

Chroni dane przed odniesieniem do nieistniejących rekordów

- Tworzy powiązania

Ułatwia zapytania
- między tabelą faktów i tabelami wymiarów

np. JOIN, GROUP BY



Eksport danych

| Properties Processes     |                                     |       |             |                  |                                  |                  |
|--------------------------|-------------------------------------|-------|-------------|------------------|----------------------------------|------------------|
| Search                   |                                     |       |             |                  |                                  |                  |
|                          |                                     | PID   | Type ^      | Server           | Object                           | Start Time       |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 11960 | Export Data | PostgreSQL - ... | hurtownia/public.fakt_zamowienie | 6.06.2025, 20:06 |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 5704  | Export Data | PostgreSQL - ... | hurtownia/public.dim_produkt     | 6.06.2025, 20:17 |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 10100 | Export Data | PostgreSQL - ... | hurtownia/public.dim_klient      | 6.06.2025, 20:18 |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 2704  | Export Data | PostgreSQL - ... | hurtownia/public.dim_kategoria   | 6.06.2025, 20:19 |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 20976 | Export Data | PostgreSQL - ... | hurtownia/public.dim_czas        | 6.06.2025, 20:20 |

5. Utwórz skrypt SQL schema\_hd.sql zawierający wszystkie polecenia DDL.

```
File Edit Selection View Go Run ...
schema_hd.sql X
C: > Users > kryst > Documents > 41 -- Bazy - Danych > Temat 9 Projekt --
1  -- schema_hd.sql  architektura gwiazdy
2
3  CREATE TABLE dim_klient (
4      klient_id SERIAL PRIMARY KEY,
5      imie VARCHAR(50),
6      nazwisko VARCHAR(50),
7      miasto VARCHAR(50),
8      ulica VARCHAR(100),
9      email VARCHAR(100)
10 );
11
12 CREATE TABLE dim_produkt (
13     produkt_id SERIAL PRIMARY KEY,
14     nazwa VARCHAR(100),
15     parametr VARCHAR(50),
16     cena_brutto FLOAT,
17     id_kategoria INT
18 );
19
```

```
File Edit Selection View Go Run ... ← →
schema_hd.sql X
C: > Users > kryst > Documents > 41 -- Bazy - Danych > Temat 9 Proj
19
20 CREATE TABLE dim_kategoria (
21     kategoria_id SERIAL PRIMARY KEY,
22     nazwa_kategorii VARCHAR(50)
23 );
24
25 CREATE TABLE dim_czas (
26     czas_id SERIAL PRIMARY KEY,
27     data DATE,
28     rok INT,
29     miesiac INT,
30     dzien INT,
31     dzien_tygodnia VARCHAR(10)
32 );
```

```
File Edit Selection View Go Run ... ← → Search
schema_hd.sql X
C: > Users > kryst > Documents > 41 -- Bazy - Danych > Temat 9 Projekt -- Hurtownia > schema_hd.sql
32 ),
33
34 -- Tabela faktów (zamówienia)
35
36 CREATE TABLE fakt_zamowienie (
37     zamowienie_id SERIAL PRIMARY KEY,
38     klient_id INT REFERENCES dim_klient(klient_id),
39     produkt_id INT REFERENCES dim_produkt(produkt_id),
40     czas_id INT REFERENCES dim_czas(czas_id),
41     ilosc INT NOT NULL,
42     wartosc FLOAT NOT NULL,
43     status VARCHAR(20)
44 );
45
46 -- Indeksy wspomagające i dodatkowe klucze obce
47 ALTER TABLE dim_produkt
48     ADD FOREIGN KEY (id_kategoria) REFERENCES dim_kategoria(kategoria_id);
```

-- schema\_hd.sql –

```
CREATE TABLE dim_klient (
    klient_id SERIAL PRIMARY KEY,
    imie VARCHAR(50),
    nazwisko VARCHAR(50),
    miasto VARCHAR(50),
```

```
        ulica VARCHAR(100),  
        email VARCHAR(100)  
    );
```

```
CREATE TABLE dim_produkt (  
    produkt_id SERIAL PRIMARY KEY,  
    nazwa VARCHAR(100),  
    parametr VARCHAR(50),  
    cena_brutto FLOAT,  
    id_kategoria INT  
);
```

```
CREATE TABLE dim_kategoria (  
    kategoria_id SERIAL PRIMARY KEY,  
    nazwa_kategorii VARCHAR(50)  
);
```

```
CREATE TABLE dim_czas (  
    czas_id SERIAL PRIMARY KEY,  
    data DATE,  
    rok INT,  
    miesiac INT,  
    dzien INT,  
    dzien_tygodnia VARCHAR(10)  
);
```

-- Tabela faktów (zamówienia)

```
CREATE TABLE fakt_zamowienie (  
    zamowienie_id SERIAL PRIMARY KEY,  
    klient_id INT REFERENCES dim_klient(klient_id),
```

```
produkt_id INT REFERENCES dim_produkt(produkt_id),  
czas_id INT REFERENCES dim_czas(czas_id),  
ilosc INT NOT NULL,  
wartosc FLOAT NOT NULL,  
status VARCHAR(20)  
);  
  
-- Indeksy wspomagające i dodatkowe klucze obce  
ALTER TABLE dim_produkt  
    ADD FOREIGN KEY (id_kategoria) REFERENCES dim_kategoria(kategoria_id);
```

## ETAP II – Implementacja Procesu ETL i Zasilenie Hurtowni

1. Przygotuj transformacje zasilające tabele wymiarów i tabelę faktów.



Hi Krystian,

Thank you for your interest in the Pentaho+ platform. We can't wait to demonstrate how our cutting-edge data intelligence solutions will take your business's data strategy to the next level!

Trusted by 73% of Fortune 100 companies, Pentaho+ is designed to help organizations like yours achieve unparalleled data performance and efficiency.

Here's what to expect during your demo:

- An overview of Pentaho+ key features and capabilities.
- A demonstration of how our platform can address your specific data challenges.
- An opportunity to ask questions and discuss your data strategy needs.

A Pentaho expert will contact you shortly to schedule a convenient time for your demo. If you don't hear from us soon, please respond to this email with any immediate questions.

To your data-fitness,  
The Pentaho Team

# Pentaho



Copyright © 2024 Hitachi Vantara  
Our address is 2535 Augustine Drive, Santa Clara, CA 95054, United States

If you do not wish to receive future email, [click here](#).

[Click here](#) to view this message in a browser window.



# Pentaho

Spoon - Transformation 1

File Edit View Action Tools Help

View Design

Search

Input

- CSV file input
- Data grid
- De-serialize from file
- ESRI shapefile reader
- Email messages input
- Fixed file input
- GZIP CSV input
- Generate random credit card number
- Generate random value
- Generate rows
- Get data from XML
- Get file names
- Get files rows count
- Get repository names
- Get subfolder names
- Get system info
- Get table names
- Google Analytics
- HL7 input
- JSON input
- LDAP input

Welcome! Transformation 1

190%

CSV file input

Table output

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

2025/06/12 18:44:46 - Spoon - Transformation opened.

2025/06/12 18:44:46 - Spoon - Launching transformation [Transformation 1]...

2025/06/12 18:44:46 - Spoon - Started the transformation execution.

2025/06/12 18:44:46 - Transformation 1 - Dispatching started for transformation [Transformation 1]

2025/06/12 18:44:46 - Table output.0 - Connected to database [PostgreSQL\_Hurtownia] (commit=1000)

2025/06/12 18:44:46 - CSV file input.0 - Header row skipped in file 'C:\HDprojekt\dim\_klient\_full.csv'

2025/06/12 18:44:46 - CSV file input.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)

2025/06/12 18:44:46 - Table output.0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)

2025/06/12 18:44:46 - Spoon - The transformation has finished!!

CSV file input

Step name

CSV file input

Filename

C:\HDprojekt\dim\_klient\_full.csv

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

☒

Header row present?

☒

Add filename to result

☐

The row number field name (optional)

Running in parallel?

☐

New line possible in fields?

☐

Format

mixed

File encoding

UTF-8

| # | Name     | Type   | Format | Length | Precision | Currency | Decimal | Group | Trim type |
|---|----------|--------|--------|--------|-----------|----------|---------|-------|-----------|
| 1 | imie     | String |        | 12     |           | zł       | ,       |       | none      |
| 2 | nazwisko | String |        | 13     |           | zł       | ,       |       | none      |
| 3 | miasto   | String |        | 10     |           | zł       | ,       |       | none      |
| 4 | ulica    | String |        | 13     |           | zł       | ,       |       | none      |
| 5 | email    | String |        | 19     |           | zł       | ,       |       | none      |

Help

OK

Get Fields

Preview

Cancel

Select a File

Select a File

C:\HDprojekt

| Name                | Type | Modified          |
|---------------------|------|-------------------|
| dim_klient_full.csv | File | 06/06/25 11:37 PM |

File type: \*.csv, \*.txt (\*.csv,\*.txt)

Open

Cancel

Database Connection

General

Advanced

Options

Pooling

Clustering

Connection name:

PostgreSQL\_Hurtownia

Connection type:

PostgreSQL

Redshift

Remedy Action Request System

SAP ERP System

SQLite

Snowflake

Sybase

SybaseIQ

Teradata

UniVerse database

Vertica

Vertica 5+

Access:

Native (JDBC)

ODBC

JNDI

Settings

Host Name:

localhost

Database Name:

hurtownia

Port Number:

5432

Username:

postgres

Password:

Test

Feature List

Explore

OK

Cancel

## Execution Results

| Logging  | Execution History | Step Metrics | Performance Graph | Metrics | Preview data |
|--|-------------------|--------------|-------------------|---------|--------------|
|  |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Spoon - Transformation opened.   |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Spoon - Launching transformation [Transformation 1]...                           |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Spoon - Started the transformation execution.                                    |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Transformation 1 - Dispatching started for transformation [Transformation 1]     |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Table output.0 - Connected to database [PostgreSQL_Hurtownia] (commit=1000)      |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - CSV file input.0 - Header row skipped in file 'C:\HDprojekt\dim_klient_full.csv' |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - CSV file input.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)          |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Table output.0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)           |                   |              |                   |         |              |
| 2025/06/12 18:44:46 - Spoon - The transformation has finished!!  |                   |              |                   |         |              |

TRUNCATE TABLE fakt\_zamowienie, dim\_czas RESTART IDENTITY CASCADE;

CSV file input

Step name

dim\_czas\_full.csv

Filename

C:\HDprojekt\dim\_czas\_full.csv

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

☒

Header row present?

☒

Add filename to result

☐

The row number field name (optional)

Running in parallel?

☐

New line possible in fields?

☐

Format

mixed

File encoding

UTF-8

| # | Name           | Type    | Format     | Length | Precision | Currency | Decimal | Group | Trim type |
|---|----------------|---------|------------|--------|-----------|----------|---------|-------|-----------|
| 1 | czas_id        | Integer | #          | 15     | 0         | zł       | ,       |       | none      |
| 2 | data           | Date    | yyyy-MM-dd |        |           | zł       | ,       |       | none      |
| 3 | rok            | Integer | #          | 15     | 0         | zł       | ,       |       | none      |
| 4 | miesiac        | Integer | #          | 15     | 0         | zł       | ,       |       | none      |
| 5 | dzien          | Integer | #          | 15     | 0         | zł       | ,       |       | none      |
| 6 | dzien_tygodnia | String  |            | 9      |           | zł       | ,       |       | none      |

Help

OK

Get Fields

Preview

Cancel

Table output

Step name

dim\_czas

Connection

PostgreSQL\_Hurtownia

Edit...

New...

Wizard...

Target schema

public

Browse...

Target table

dim\_czas

Browse...

Commit size

1000

Truncate table

Ignore insert errors

Specify database fields

Main options

Database fields

Partition data over tables

Partitioning field

Partition data per month

Partition data per day

Use batch update for inserts

Is the name of the table defined in a field?

Field that contains name of table:

Store the tablename field

Return auto-generated key

Name of auto-generated key field

Help

OK

Cancel

SQL

Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

2025/06/12 20:27:56 - Spoon - Transformation opened.

2025/06/12 20:27:56 - Spoon - Launching transformation [Transformation 1]...

2025/06/12 20:27:56 - Spoon - Started the transformation execution.

2025/06/12 20:27:56 - Transformation 1 - Dispatching started for transformation [Transformation 1]

2025/06/12 20:27:56 - dim\_klient.0 - Connected to database [PostgreSQL\_Hurtownia] (commit=1000)

2025/06/12 20:27:56 - dim\_czas.0 - Connected to database [PostgreSQL\_Hurtownia] (commit=1000)

2025/06/12 20:27:56 - dim\_kilent\_full.csv.0 - Header row skipped in file 'C:\HDprojekt\dim\_klient\_full.csv'

2025/06/12 20:27:56 - dim\_czas\_full.csv.0 - Header row skipped in file 'C:\HDprojekt\dim\_czas\_full.csv'

2025/06/12 20:27:56 - dim\_kilent\_full.csv.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)

2025/06/12 20:27:56 - dim\_czas\_full.csv.0 - Finished processing (I=366, O=0, R=0, W=365, U=0, E=0)

2025/06/12 20:27:56 - dim\_klient.0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)

2025/06/12 20:27:56 - dim\_czas.0 - Finished processing (I=0, O=365, R=365, W=365, U=0, E=0)

2025/06/12 20:27:56 - Spoon - The transformation has finished!!

```
--select * from dim_kategoria ;

--TRUNCATE TABLE fakt_zamowienie, dim_kategoria RESTART IDENTITY CASCADE;
```



Table output

Step name

dim\_kategoria.csv

Connection

PostgreSQL\_Hurtownia

Edit...New...Wizard...

Target schema

public

Browse...

Target table

dim\_kategoria

Browse...

Commit size

1000

Truncate table☐

Ignore insert errors☐

Specify database fields☐

Main options

Database fields

Partition data over tables☐

Partitioning field

Partition data per month☒

Partition data per day☐

Use batch update for inserts☒

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

Field that contains name of table:

Store the tablename field☒

Return auto-generated key☐

Name of auto-generated key field

Help

OK

Cancel

SQL

Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

2025/06/12 20:43:32 - Spoon - Transformation opened.

2025/06/12 20:43:32 - Spoon - Launching transformation [Transformation 1]...

2025/06/12 20:43:32 - Spoon - Started the transformation execution.

2025/06/12 20:43:32 - Transformation 1 - Dispatching started for transformation [Transformation 1]

2025/06/12 20:43:32 - dim\_kategoria.csv.0 - Connected to database [PostgreSQL\_Hurtownia] (commit=1000)

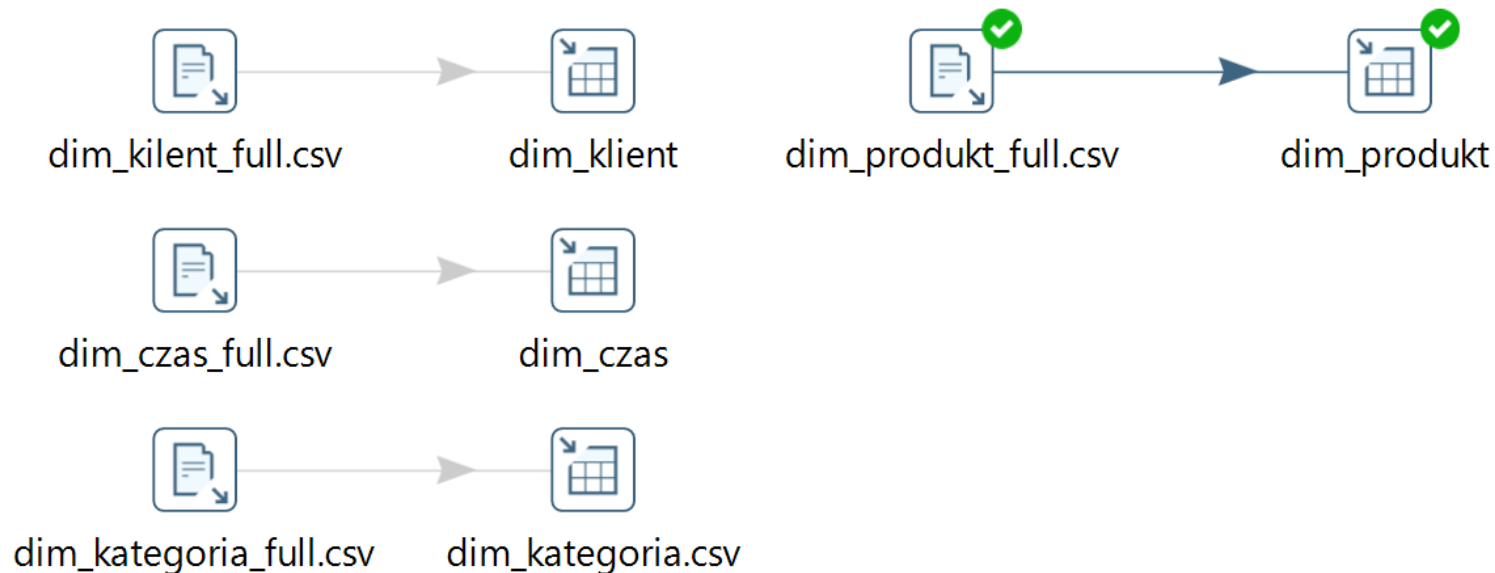
2025/06/12 20:43:32 - dim\_kategoria\_full.csv.0 - Header row skipped in file 'C:\HDprojekt\dim\_kategoria\_full.csv'

2025/06/12 20:43:32 - dim\_kategoria\_full.csv.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)

2025/06/12 20:43:32 - dim\_kategoria.csv.0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)

2025/06/12 20:43:32 - Spoon - The transformation has finished!!

dim\_produk



CSV file input

Step name

dim\_produkkt\_full.csv

Filename

C:\HDprojekt\dim\_produkkt\_full.csv

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

☒

Header row present?

☒

Add filename to result

☐

The row number field name (optional)

Running in parallel?

☐

New line possible in fields?

☐

Format

mixed

File encoding

UTF-8

| # | Name         | Type    | Format | Length | Precision | Currency | Decimal | Group | Trim type |
|---|--------------|---------|--------|--------|-----------|----------|---------|-------|-----------|
| 1 | nazwa        | String  |        | 28     |           | zł       | ,       |       | none      |
| 2 | parametr     | String  |        | 44     |           | zł       | ,       |       | none      |
| 3 | cena_brutto  | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
| 4 | id_kategoria | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
|   |              |         |        |        |           |          |         |       |           |
|   |              |         |        |        |           |          |         |       |           |
|   |              |         |        |        |           |          |         |       |           |
|   |              |         |        |        |           |          |         |       |           |
|   |              |         |        |        |           |          |         |       |           |

Help

OK

Get Fields

Preview

Cancel



Table output

Step name

dim\_produk

Connection

PostgreSQL\_Hurtownia

Edit...

New...

Wizard...

Target schema

public

Browse...

Target table

dim\_produk

Browse...

Commit size

1000

Truncate table

Ignore insert errors

Specify database fields

Main options

Database fields

Partition data over tables

Partitioning field

Partition data per month

Partition data per day

Use batch update for inserts

Is the name of the table defined in a field?

Field that contains name of table:

Store the tablename field

Return auto-generated key

Name of auto-generated key field

Help

OK

Cancel

SQL

## Execution Results

| Logging   | Execution History | Step Metrics | Performance Graph | Metrics | Preview data |
|---|-------------------|--------------|-------------------|---------|--------------|
|   |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - Spoon - Transformation opened.  |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - Spoon - Launching transformation [Transformation 1]...                                |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - Spoon - Started the transformation execution.   |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - Transformation 1 - Dispatching started for transformation [Transformation 1]          |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - dim_produk .0 - Connected to database [PostgreSQL_Hurtownia] (commit=1000)            |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - dim_produk_full.csv.0 - Header row skipped in file 'C:\HDprojekt\dim_produk_full.csv' |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - dim_produk_full.csv.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)          |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - dim_produk .0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)                 |                   |              |                   |         |              |
| 2025/06/12 21:03:04 - Spoon - The transformation has finished!!   |                   |              |                   |         |              |



dim\_klient\_full.csv



dim\_klient



dim\_produk



dim\_produk



dim\_czas\_full.csv



dim\_czas



fakt\_zamowienie\_full.csv



dim\_zamowienie



dim\_kategoria\_full.csv



dim\_kategoria.csv

CSV file input

Step name

fakt\_zamowienie\_full.csv

Filename

C:\HDprojekt\fakt\_zamowienie\_full.csv

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

☒

Header row present?

☒

Add filename to result

☐

The row number field name (optional)

Running in parallel?

☐

New line possible in fields?

☐

Format

mixed

File encoding

UTF-8

| # | Name       | Type    | Format | Length | Precision | Currency | Decimal | Group | Trim type |
|---|------------|---------|--------|--------|-----------|----------|---------|-------|-----------|
| 1 | klient_id  | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
| 2 | produkt_id | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
| 3 | czas_id    | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
| 4 | ilosc      | Integer | #      | 15     | 0         | zł       | ,       |       | none      |
| 5 | wartosc    | Number  | #,.    | 15     | 0         | zł       | .       | ,     | none      |
| 6 | status     | String  |        | 12     |           | zł       | ,       |       | none      |

Help

OK

Get Fields

Preview

Cancel

Table output

Step name

dim\_zamowienie

Connection

PostgreSQL\_Hurtownia

Edit...

New...

Wizard...

Target schema

public

Browse...

Target table

fakt\_zamowienie

Browse...

Commit size

1000

Truncate table

☐

Ignore insert errors

☐

Specify database fields

☐

Main options

Database fields

Partition data over tables

☐

Partitioning field

Partition data per month

☐

Partition data per day

☐

Use batch update for inserts

☒

Is the name of the table defined in a field?

☐

Field that contains name of table:

Store the tablename field

☒

Return auto-generated key

☐

Name of auto-generated key field

Help

OK

Cancel

SQL

Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

⊖

🗑

⚙

2025/06/12 21:08:20 - Spoon - Transformation opened.

2025/06/12 21:08:20 - Spoon - Launching transformation [Transformation 1]...

2025/06/12 21:08:20 - Spoon - Started the transformation execution.

2025/06/12 21:08:20 - Transformation 1 - Dispatching started for transformation [Transformation 1]

2025/06/12 21:08:20 - dim\_zamowienie.0 - Connected to database [PostgreSQL\_Hurtownia] (commit=1000)

2025/06/12 21:08:20 - fakt\_zamowienie\_full.csv.0 - Header row skipped in file 'C:\HDprojekt\fakt\_zamowienie\_full.csv'

2025/06/12 21:08:20 - fakt\_zamowienie\_full.csv.0 - Finished processing (I=51, O=0, R=0, W=50, U=0, E=0)

2025/06/12 21:08:20 - dim\_zamowienie.0 - Finished processing (I=0, O=50, R=50, W=50, U=0, E=0)

2025/06/12 21:08:20 - Spoon - The transformation has finished!!