BUSINESS INTELLIGENCE

ETL AND DATA WAREHOUSE DESIGN USING KAGGLE RETAIL DATASET
(Northwind Traders)

Supervisor:

Endah Septa Sintiya, S.Pd., M.Kom.

Disusun oleh:

Antonius Kaharap Kautsar	2341720067
Charellino Kalingga S	2341720205
Erwan Majid	2341720064
Hammam Abdullah S B G	2341720203
Ridho Anfa’al	2341720222

INFORMATION TECHNOLOGY DEPARTMENT

DIPLOMA IV IN INFORMATICS ENGINEERING STUDY PROGRAM

MALANG STATE POLYTECHNIC

2025

Table of Contents

CHAPTER 1
INTRODUCTION

1.1 Background

A company's strategic decision-making relies heavily on data. Northwind Traders (a fictional company) wants to implement a Data Warehouse (DW) to analyze its transaction processes, specifically to calculate Key Performance Indicators (KPIs) such as revenue.

Northwind Traders operational data (OLTP) is often scattered across various systems and unstructured for comprehensive analysis. Therefore, an Extract, Transform, and Load (ETL) process is required to integrate the raw data into a centralized, organized structure.

This project uses the AdventureWorks Retail Kaggle dataset, which contains sales, product, customer, and region data. This dataset will be processed through an ETL process using MySQL to produce a Star Schema model ready to support business analysis and KPI visualization.

1.2 Problem Statement

The problem statement in this project is as follows:

1. How to design a Star Schema consisting of fact tables and dimension tables to represent Northwind sales data?
2. How is the ETL process applied to integrate raw CSV data (orders, order_details, products)?
3. How to perform analysis to calculate KPIs like Total Sales and Top Selling Products?

1.3 Project Objectives

The objectives of this project are:

1. Design a star schema based on the Northwind dataset.
2. Build an ETL pipeline to extract, transform, and load clean data.
3. Produce a Data Warehouse structure ready for sales KPI analysis

1.4 Scope of Problem

The scope of the project includes:

1. The dataset used includes the uploaded CSV files: orders, order_details, products, categories, customers, employees, and shippers.
2. Analysis focuses on sales data and supporting dimensions (customer, product, time).
3. Tools ETL yang digunakan adalah Pentaho Data Integration (PDI).
4. Data warehouse dibangun menggunakan skema bintang sederhana

1.5 Project Benefits

The benefits obtained include:

1. Understanding how to integrate various data sources into a single analysis model.
2. Developing the ability to design star schemas based on business needs.
3. Improving skills in performing ETL using professional tools.
4. Producing a data warehouse that can support analysis and decision making.
5. Providing a basis for creating sales KPI dashboards.

CHAPTER 2
CASE STUDY AND DATA DESCRIPTION

2.1 Case Study Selection

The Northwind Traders dataset was selected as the case study because it represents a classic relational database scenario widely used in database literature. It offers a perfect balance of complexity—containing typical ERP (Enterprise Resource Planning) entities like employees, customers, orders, and suppliers—making it an ideal candidate for demonstrating ETL principles and Data Warehouse design

2.2 General Description of Case Study

Northwind Traders is a fictional company that manages orders for specialty foods from around the world. The workflow involves:

- Customers placing Orders.
- Orders containing multiple Line Items (Order Details).
- Products belonging to specific Categories (e.g., Beverages, Condiments).
- Employees facilitating these transactions. The goal is to transform this transactional flow into an analytical model to answer questions like "Who is the most profitable customer?" or "Which category drives the most revenue in Q4?"

2.3 Dataset Source

- Source: Kaggle - Northwind Traders
- URL: https://www.kaggle.com/datasets/jeetahirwar/northwind-traders
- Format: CSV (Comma Separated Values)

2.4 Dataset Structure

Summary of the dataset structure used in this project:

File Name (Table)	**Main Columns**	**Description**

orders.csv	orderID, customerID, employeeID, orderDate, freight	Records transaction headers and dates.
order_details.csv	orderID, productID, unitPrice, quantity, discount	Records transaction details (items sold).
products.csv	productID, productName, categoryID, unitPrice	Stores complete information about every product.
categories.csv	categoryID, categoryName, description	List of main product categories.
customers.csv	customerID, companyName, contactName, country	Stores customer demographic profiles.
employees.csv	employeeID, lastName, firstName, title	Stores information about the employees handling orders.

2.5 Reasons for Case Study Selection

This dataset was chosen because:

1. Referential Integrity: The dataset maintains strong consistency between Foreign Keys (e.g., all productIDs in order_details exist in products), ensuring a smooth Join process.
2. Rich Dimensionality: It allows for slicing data by multiple dimensions: Time (OrderDate), Geography (Customer Country), and Product Hierarchy (Category).
3. Calculated Metrics: It requires data transformation to derive actual sales values (Unit Price × Quantity - Discount), which is a core ETL task.

CHAPTER 3
DATA WAREHOUSE DESIGN (STAR SCHEMA)

3.1 Identification of Fact Tables

The main focus is **Fact Sales** to analyze revenue.

Fact Table	**Source**	**Measurement Metrics**	**Foreign Keys**
Fact Sales	Join of orders & order_details	quantity, unitPrice, discount, LineSales (Calculated)	productID, customerID, employeeID, orderDate

3.2 Dimension Table Identification

Dimension tables provide the "context" for the facts. We denormalized the raw data into four primary dimensions:

1. Dim_Product: Created by joining products and categories. This allows analysis at both the Product level (e.g., "Chai") and Category level (e.g., "Beverages").
2. Dim_Customer: Contains companyName, contactName, city, country. Useful for regional sales analysis.
3. Dim_Employee: Contains firstName, lastName, title. Used to evaluate staff performance.
4. Dim_Time: Derived from orderDate. Attributes include Year, Month, Quarter, DayOfWeek. Essential for time-series analysis.

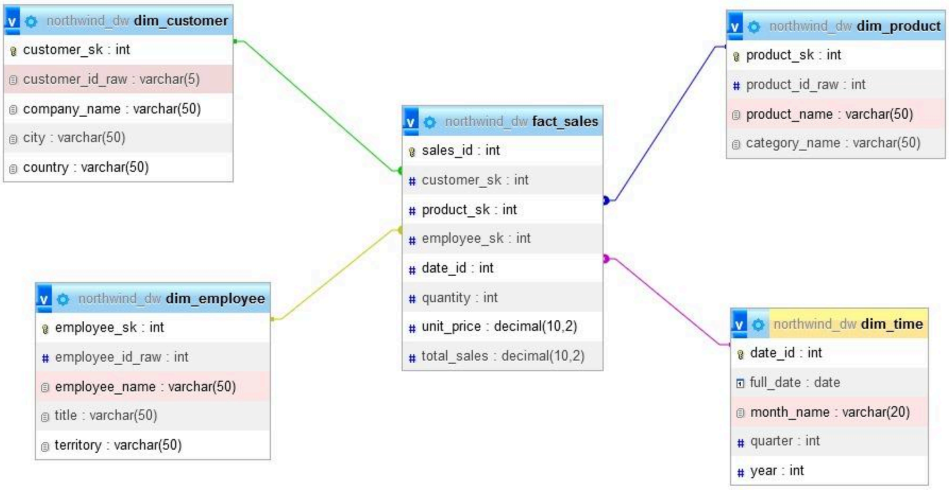
3.3 Main Dataset Structure for Star Schem

Here is a summary of the data that will be mapped into the Star Schema:

File Name (Table)	**Main Column Contents**	**Role in Star Schema**
Adventure Works Sales 2015-2017 (Gabungan)	OrderDate, ProductKey, CustomerKey, TerritoryKey, Order Quantity, SalesAmount	Fact Sales (Tabel Fakta)

Adventure Works Products	ProductKey, ProductName, ProductCost, ProductPrice	Dim Product (Tabel Dimensi)
Adventure Works Customers	CustomerKey, FirstName, Gender, AnnualIncome	Dim Customer (Tabel Dimensi)
Adventure Works Territories	Sales TerritoryKey (dipetakan ke TerritoryKey), Region, Country, Continent	Dim Territory (Tabel Dimensi)
AdventureWorks Calendar	Date, MonthName, Year	Dim Date (Tabel Dimensi)

3.4 Diagram Star Schema



3.5 Inter-Table Relationships (Relational Mapping)

All Star Schema relationships between Dimensions and Facts are set as mandatory to ensure referential integrity and data quality in the Data Warehouse, ensuring that each revenue metric is connected to a valid context of Product, Customer, Region, and Time, in accordance with the requirements for filtering incomplete data at the ETL stage.

CHAPTER 4
ETL PROCESS DESIGN

4.1 General Overview of ETL

The ETL process is the core mechanism for populating the Data Warehouse. Using Pentaho Data Integration (PDI), the team implemented a coordinated Job (Job uas finish.kjb) that orchestrates five distinct Transformations (Transformation 1-5 uas.ktr). The process starts by ensuring the target tables are ready (SQL step), loads all dimension tables first, and finally loads the large fact table using the generated surrogate keys.

4.2 The ETL Job Orchestration

1. The main job sequences the transformations to enforce dependencies:
   Start & Initialization (SQL Step): Executes SQL to ensure the target DW tables are created or truncated, preparing the database (northwind_dw) for the load process.

2. Load Dimensions (Transformation 1-4):
   - Transformation 1 uas.ktr (Dim Employee)
   - Transformation 2 uas.ktr (Dim Customer)
   - Transformation 3 uas.ktr (Dim Product)
   - Transformation 4 uas.ktr (Dim Time)

3. Load Fact Table (Transformation 5):
   Transformation 5 uas.ktr (Fact Sales) - This step relies entirely on the successful completion of the dimension loads to perform Surrogate Key Lookups.

4.3 Transformation Phase Details

4.3.1 Data Cleaning

- Source: Raw CSV files or OLTP staging tables.
- Cleaning/Type Conversion: Ensures data types match the DW schema (e.g., converting text-based dates to MySQL DATE type).
- Integration (Dim_Product): Joins the products and categories tables to create a single denormalized dimension entity.
- Key Generation: Assigns the auto-incrementing _sk (Surrogate Key) for each dimension record upon loading.

4.3.2 Data Transformation

The Transformation 5 uas.ktr is the most complex step, combining multiple sources and performing critical calculations:

- Data Source Integration: Joins the core transactional tables (orders and order_details).

- Surrogate Key Lookup: The original relational keys (customerID, productID, orderDate, etc.) are looked up against the newly loaded dimension tables (dim_customer, dim_product, dim_employee, dim_time) to retrieve the corresponding Surrogate Keys (_sk or date_id).
- Measure Calculation: Calculates the total_sales (revenue) for each line item using the formula:

Total Sales= Quantity x Unit Price x (1 - Discount)

4.3.3 Data Aggregation

This step is implemented using the Calculator step in Pentaho Data Integration (PDI). It transforms raw operational numbers into financial metrics that business users can aggregate (Sum/Count) later in their reports.

- Input Data: The process takes the raw columns from the order_details CSV:
  - Quantity (e.g., 10)
  - UnitPrice (e.g., $20.00)
  - Discount (e.g., 0.05 or 5%)

- The Transformation Logic: A formula is applied to every single row to determine the actual revenue generated by that specific line item.
  - Formula: Line Total = Quantity * UnitPrice * (1 - Discount)

- Output Data: A new field named total_sales is created.
  - Example Result: 10 * 20 * (1 - 0.05) = $190.00

- Why this is "Aggregation": While technically a row-level calculation, this step prepares the Atomic Data (lowest level of detail) so that the Data Warehouse can perform rapid aggregations (Summing total_sales by Year, by Country, or by Product) without having to recalculate the math for every query.

4.3.4 Data Integration

- Each transformation uses a Table Output step in PDI to connect to the target northwind_dw database.
- The loading process first populates the dimension tables with unique, cleaned records.
- The final step loads the fact_sales table using the collected foreign keys (customer_sk, product_sk, employee_sk, date_id) and the calculated measures.

CHAPTER 5
ETL PIPELINE IMPLEMENTATION

5.1 Software Architecture

This sub-section describes the main tools and database configurations used to build and execute the ETL pipeline.

5.1.1 Software Architecture

The ETL process implementation in this project utilized the following combination of software:

Role	Software	Detail / Version
ETL Tool	Pentaho Data Integration (PDI) / Kettle	Version 10.2
Database Management System	MySQL	Used as both the Source (OLTP) and Target (Data Warehouse) DBMS
Source/Staging Schema	oltp_adventureworks	The database schema containing the raw OLTP data.
Target/DW Schema	dw_adventureworkss	The database schema containing the Star Schema (dimensions and facts).

5.1.2 Database Connection Configuration

In PDI, two separate database connections were configured to clearly distinguish between the source (OLTP) and the target (DW), even though both reside on the same DBMS (MySQL). Both connections utilize the Native (JDBC) driver

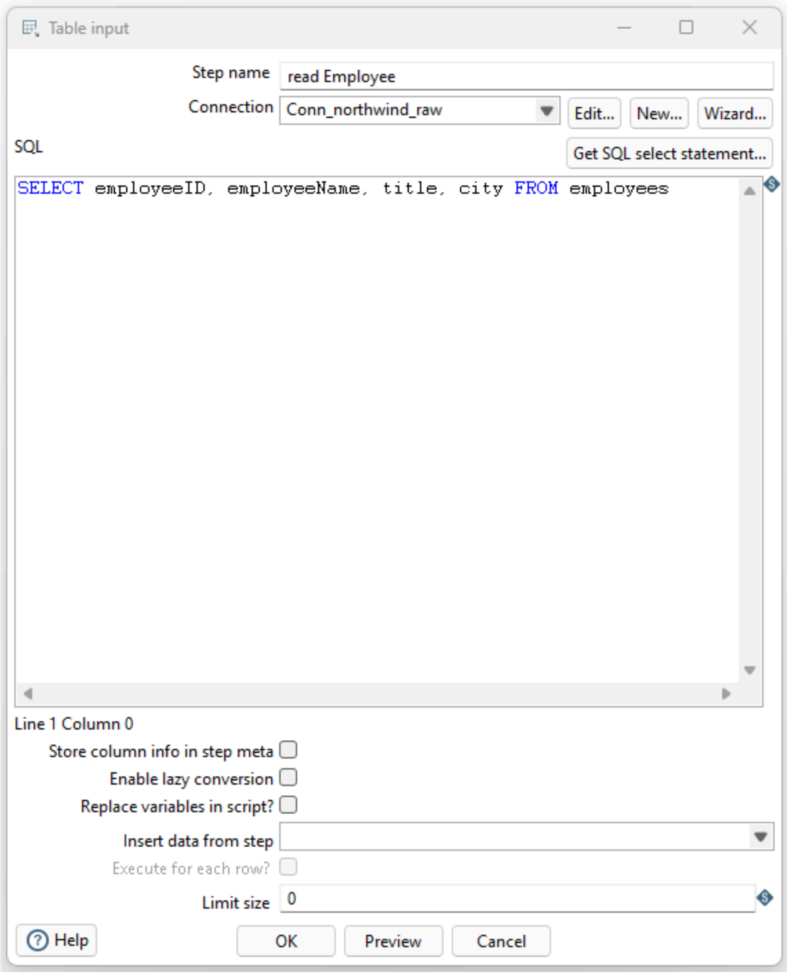
Detail	OLTP Connection	DW Connection
PDI Connection Name	conn_oltp_adventureworks	conn_dw_adventureworkss
Database Type	MySQL	MySQL

Host	localhost	localhost
Schema Name	oltp_adventureworks	dw_adventureworkss
Driver	Native (JDBC)	Native (JDBC)

5.2 Implementation of Dimensional Transformation

5.2.1 ETL_Employee

read Employee



Step name: read Employee

Connection: Conn_northwind_raw

```
SELECT employeeID, employeeName, title, city FROM employees
```

Line 1 Column 0

Store column info in step meta □

Enable lazy conversion □

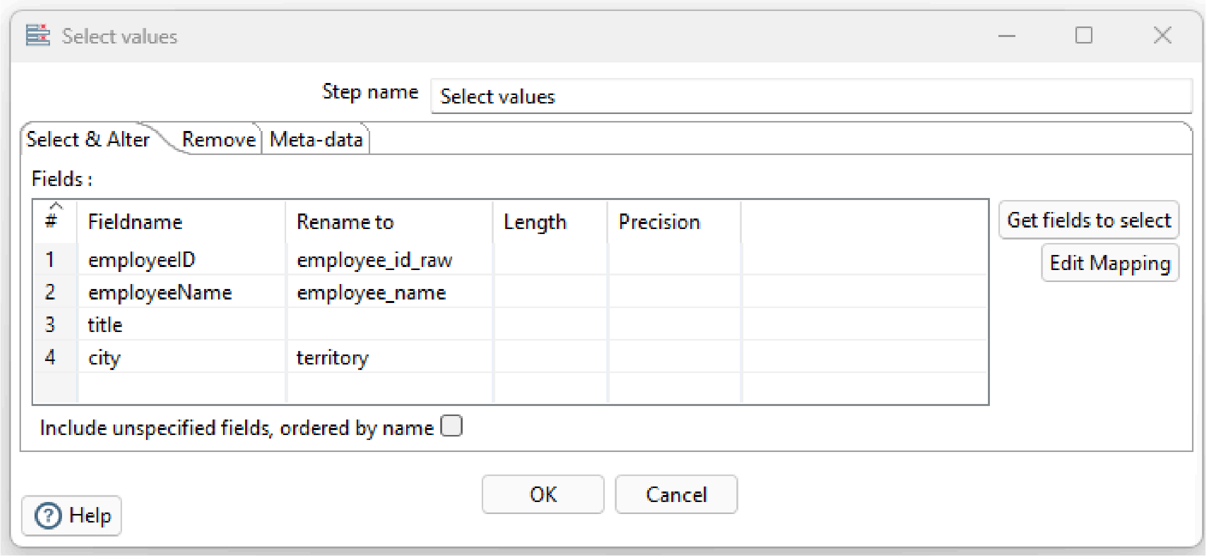
Replace variables in script? □

Insert data from step:

Execute for each row? □

Limit size: 0

Select values



Step name: Select values

Fields :

#	Fieldname	Rename to	Length	Precision
1	employeeID	employee_id_raw		
2	employeeName	employee_name		
3	title			
4	city	territory		

Include unspecified fields, ordered by name □

Load dim_employee

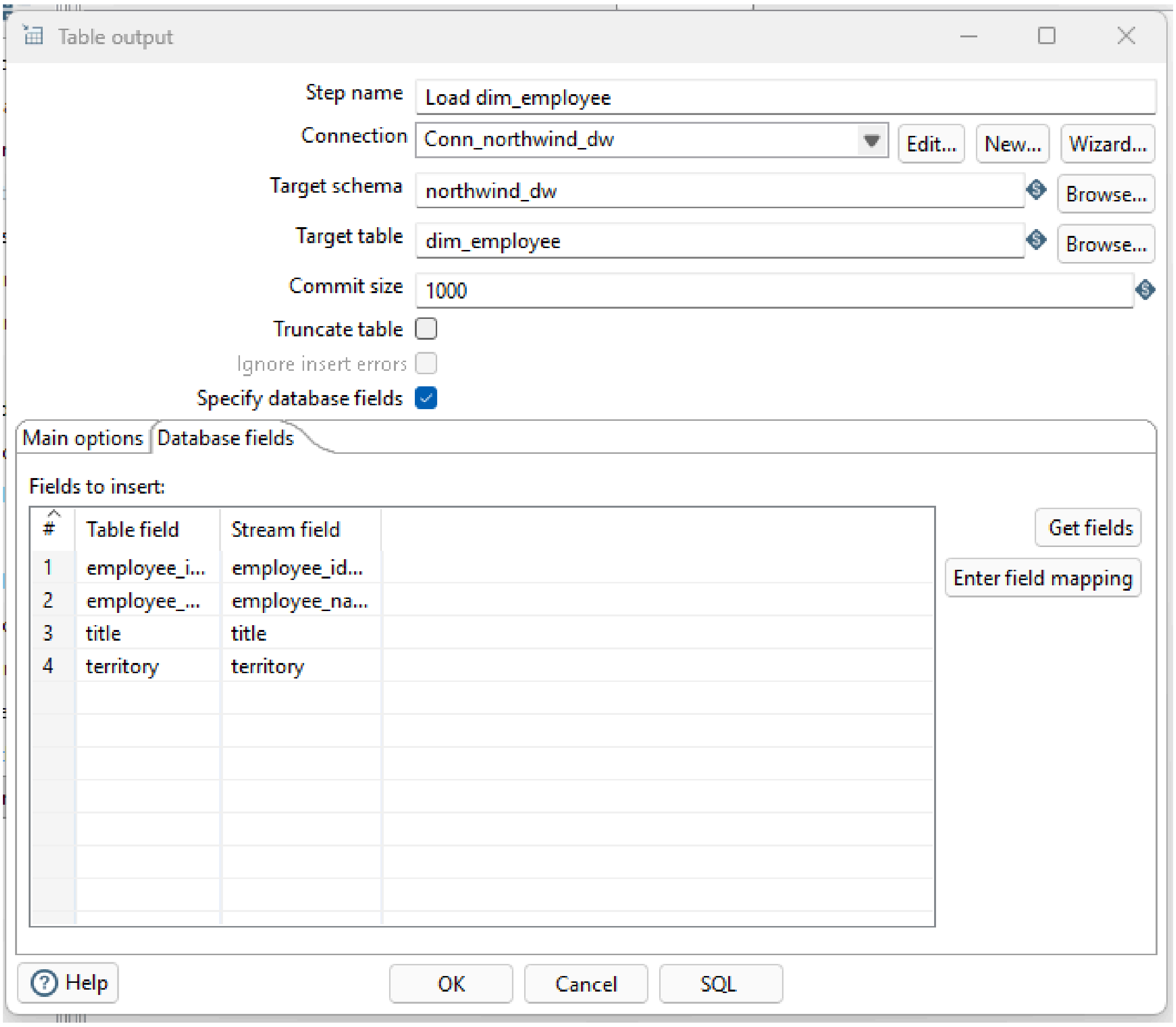


Table output

Step name: Load dim_employee

Connection: Conn_northwind_dw

Target schema: northwind_dw

Target table: dim_employee

Commit size: 1000

Truncate table □

Ignore insert errors □

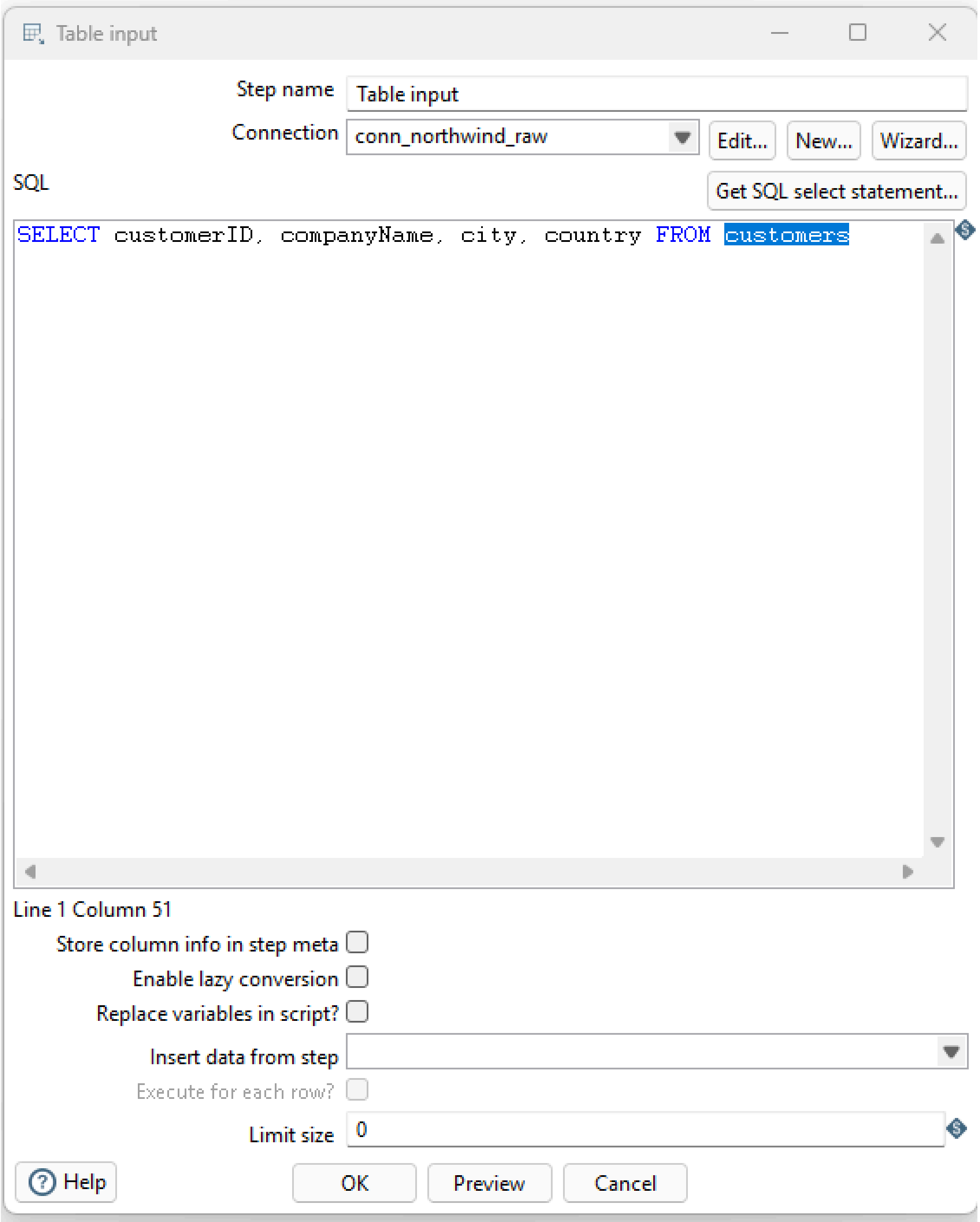
Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1	employee_i...	employee_id...
2	employee_...	employee_na...
3	title	title
4	territory	territory

5.2.2 ETL_Customers

Table input



Step name: Table input

Connection: conn_northwind_raw

```
SELECT customerID, companyName, city, country FROM customers
```

Line 1 Column 51

Store column info in step meta □

Enable lazy conversion □

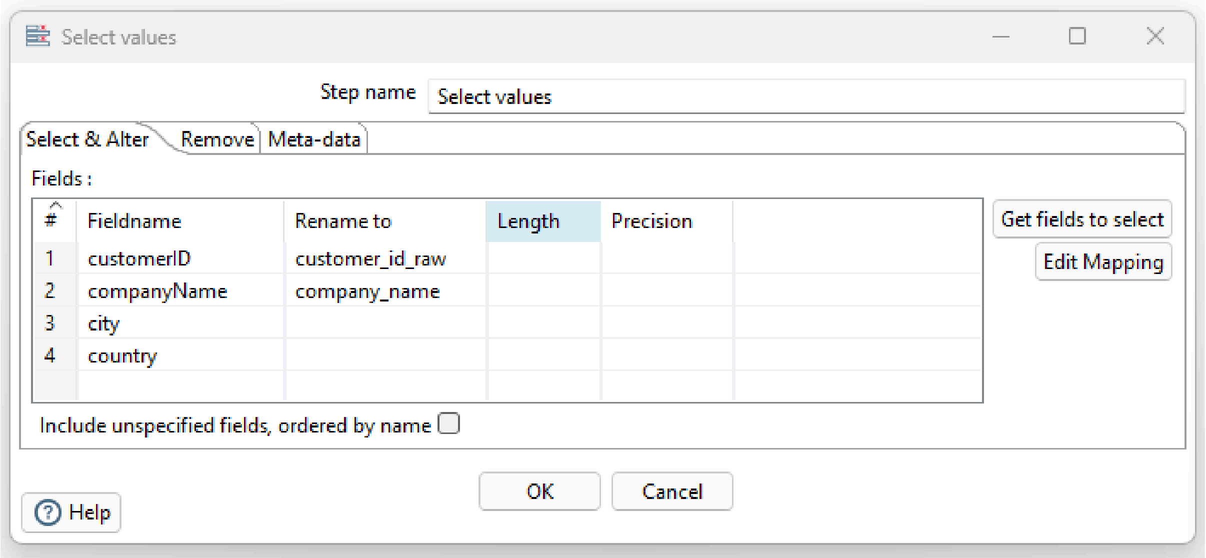
Replace variables in script? □

Insert data from step:

Execute for each row? □

Limit size: 0

Select values



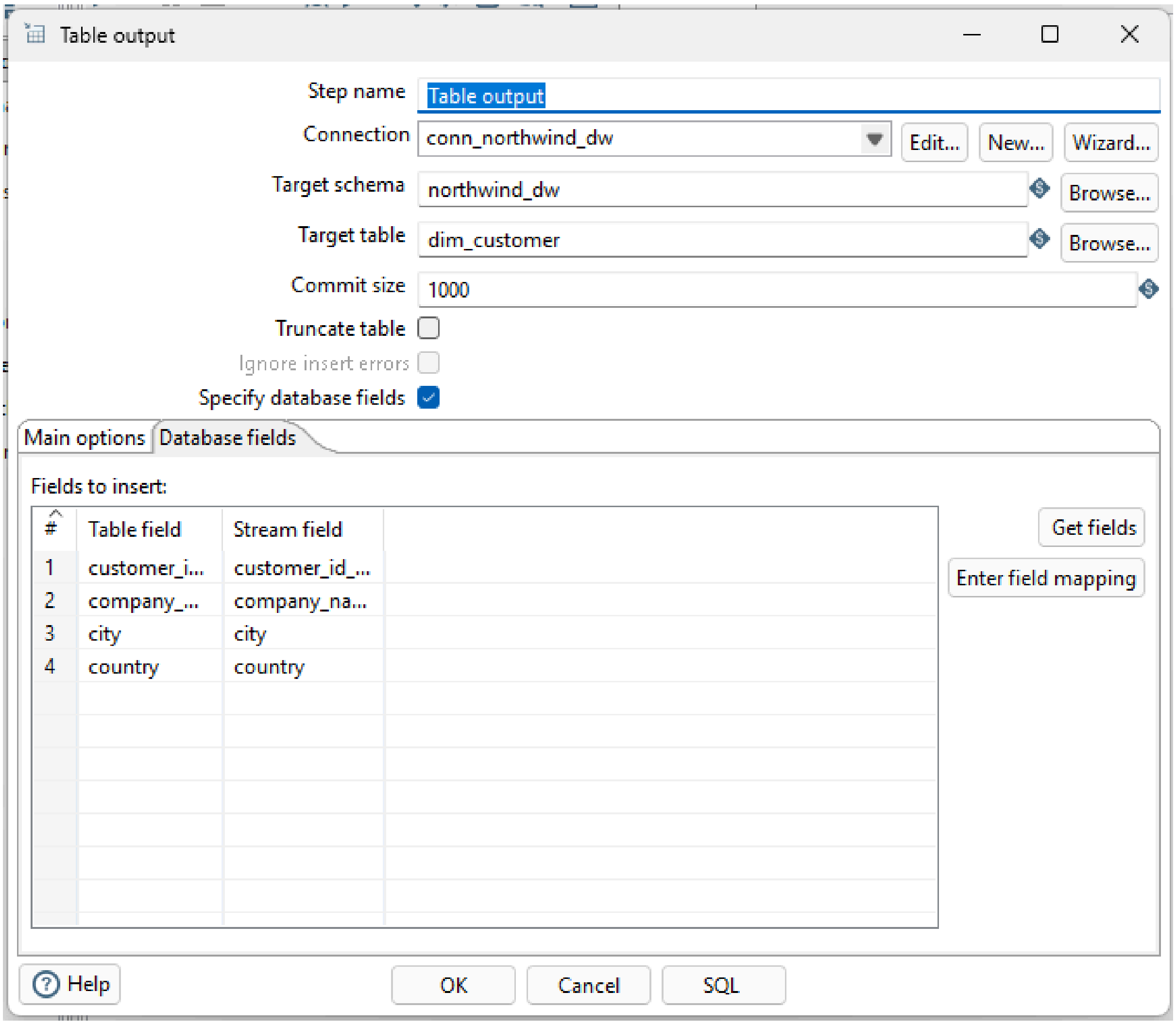
Step name: Select values

Fields :

#	Fieldname	Rename to	Length	Precision
1	customerID	customer_id_raw		
2	companyName	company_name		
3	city			
4	country			

Include unspecified fields, ordered by name □

Table output



Step name: Table output

Connection: conn_northwind_dw

Target schema: northwind_dw

Target table: dim_customer

Commit size: 1000

Truncate table □

Ignore insert errors □

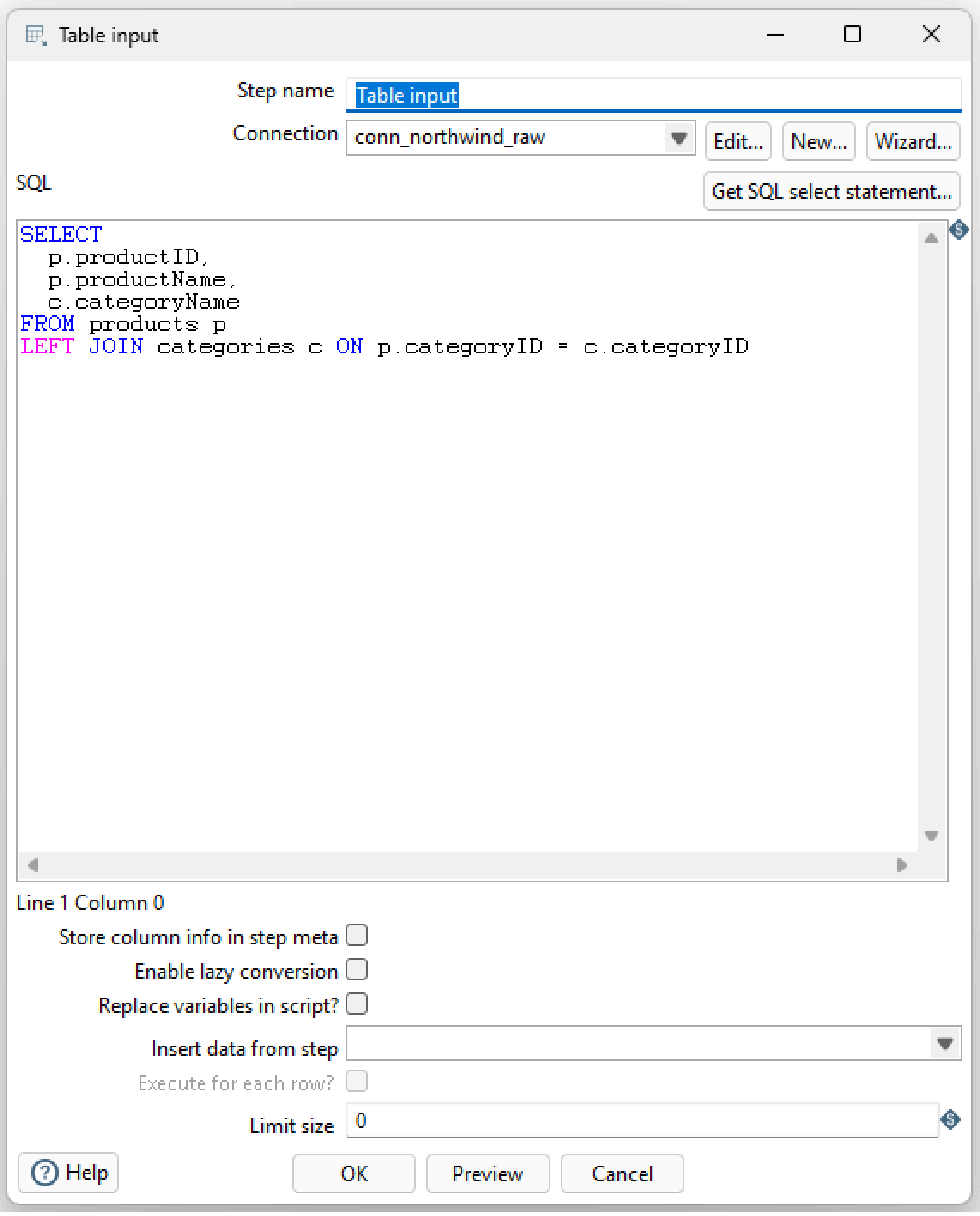
Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1	customer_i...	customer_id_...
2	company_...	company_na...
3	city	city
4	country	country

5.2.3 ETL_Categories

Table input



Step name: Table input

Connection: conn_northwind_raw

```
SELECT
  p.productID,
  p.productName,
  c.categoryName
FROM products p
LEFT JOIN categories c ON p.categoryID = c.categoryID
```

Line 1 Column 0

Store column info in step meta □

Enable lazy conversion □

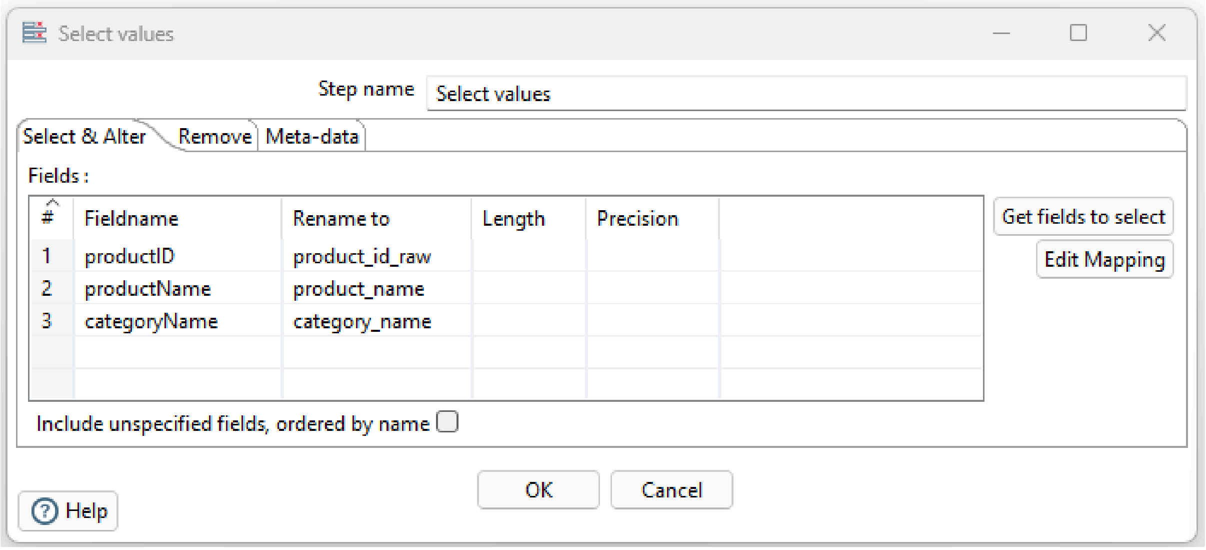
Replace variables in script? □

Insert data from step

Execute for each row? □

Limit size 0

Select values



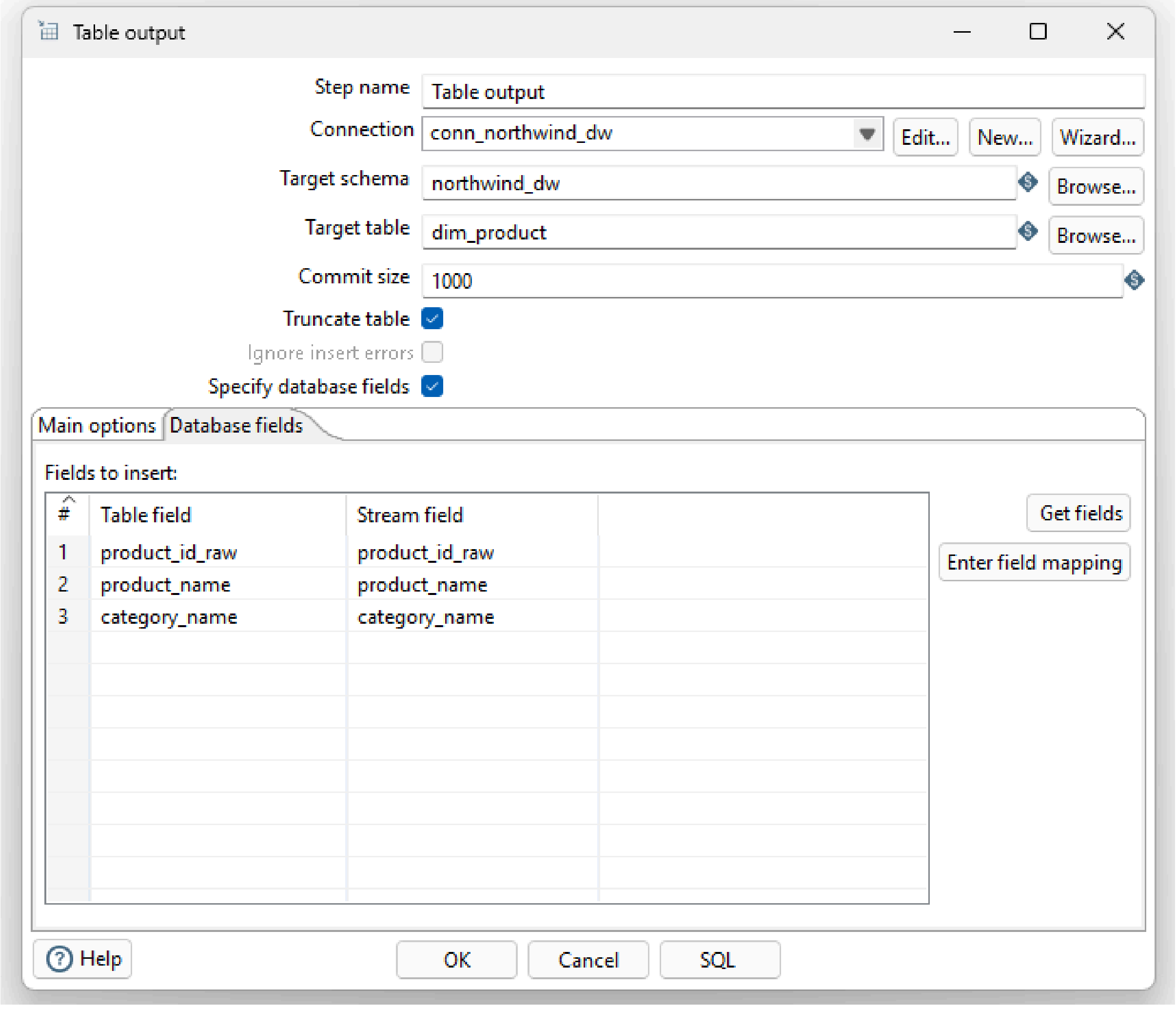
Step name: Select values

Fields :

#	Fieldname	Rename to	Length	Precision
1	productID	product_id_raw		
2	productName	product_name		
3	categoryName	category_name		

Include unspecified fields, ordered by name □

Table output



Step name: Table output

Connection: conn_northwind_dw

Target schema: northwind_dw

Target table: dim_product

Commit size: 1000

Truncate table ☑

Ignore insert errors □

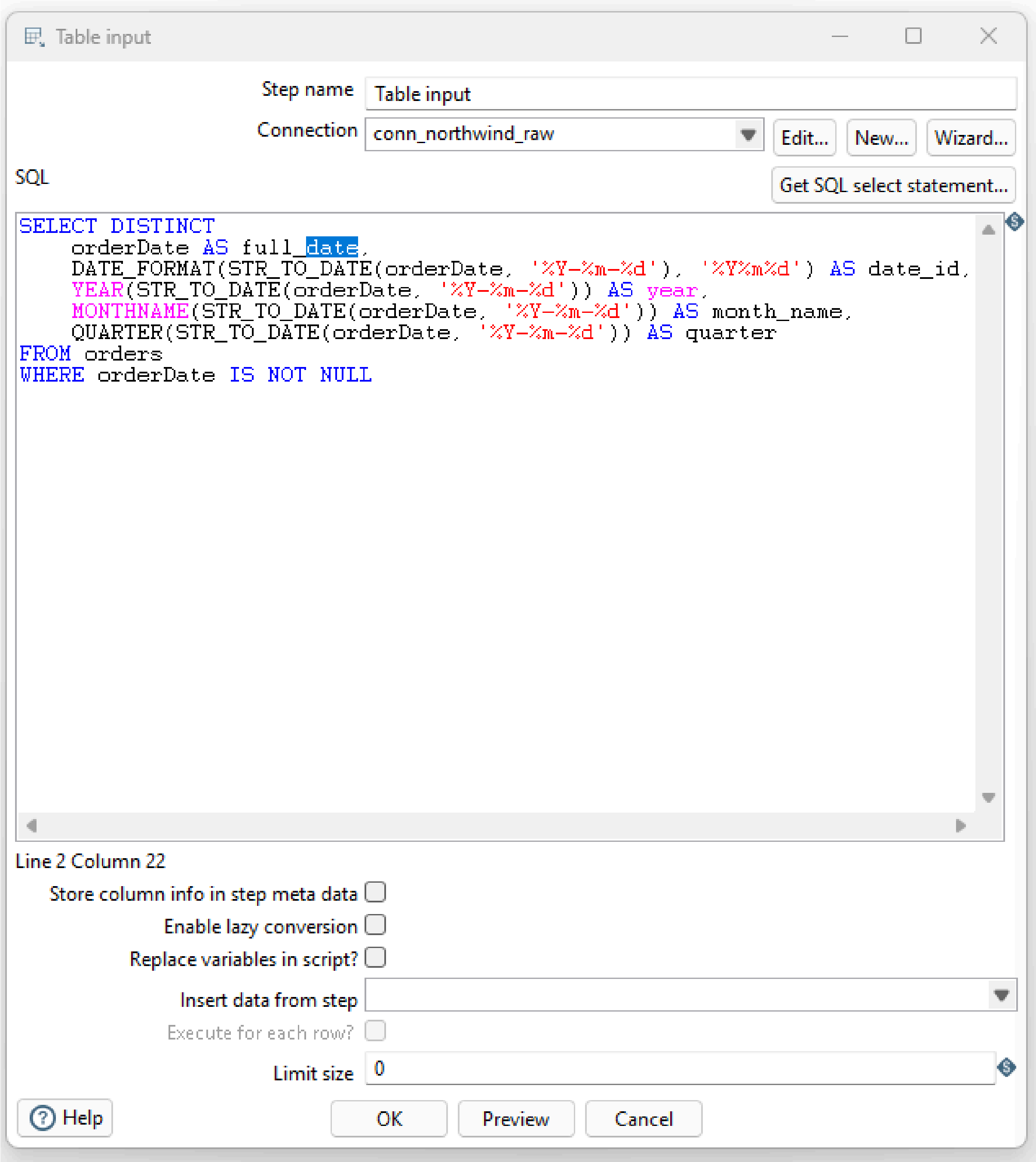
Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1	product_id_raw	product_id_raw
2	product_name	product_name
3	category_name	category_name

5.2.4 ETL_Date

Table input



Step name: Table input

Connection: conn_northwind_raw

```
SELECT DISTINCT
    orderDate AS full_date,
    DATE_FORMAT(STR_TO_DATE(orderDate, '%Y-%m-%d'), '%Y%m%d') AS date_id,
    YEAR(STR_TO_DATE(orderDate, '%Y-%m-%d')) AS year,
    MONTHNAME(STR_TO_DATE(orderDate, '%Y-%m-%d')) AS month_name,
    QUARTER(STR_TO_DATE(orderDate, '%Y-%m-%d')) AS quarter
FROM orders
WHERE orderDate IS NOT NULL
```

Line 2 Column 22

Store column info in step meta data □

Enable lazy conversion □

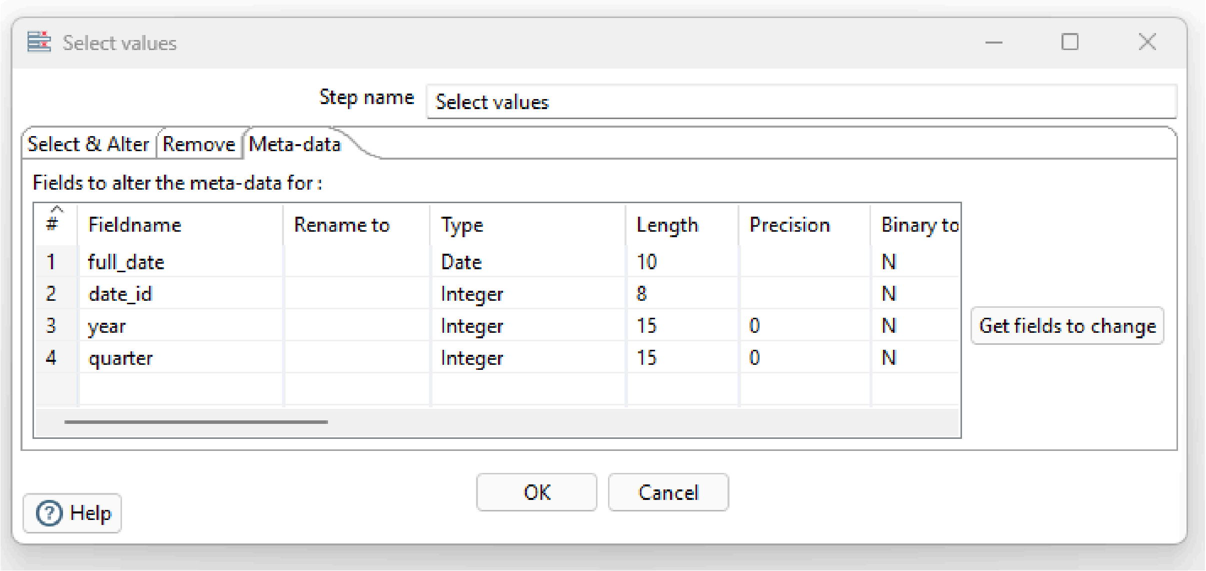
Replace variables in script? □

Insert data from step

Execute for each row? □

Limit size: 0

Select values

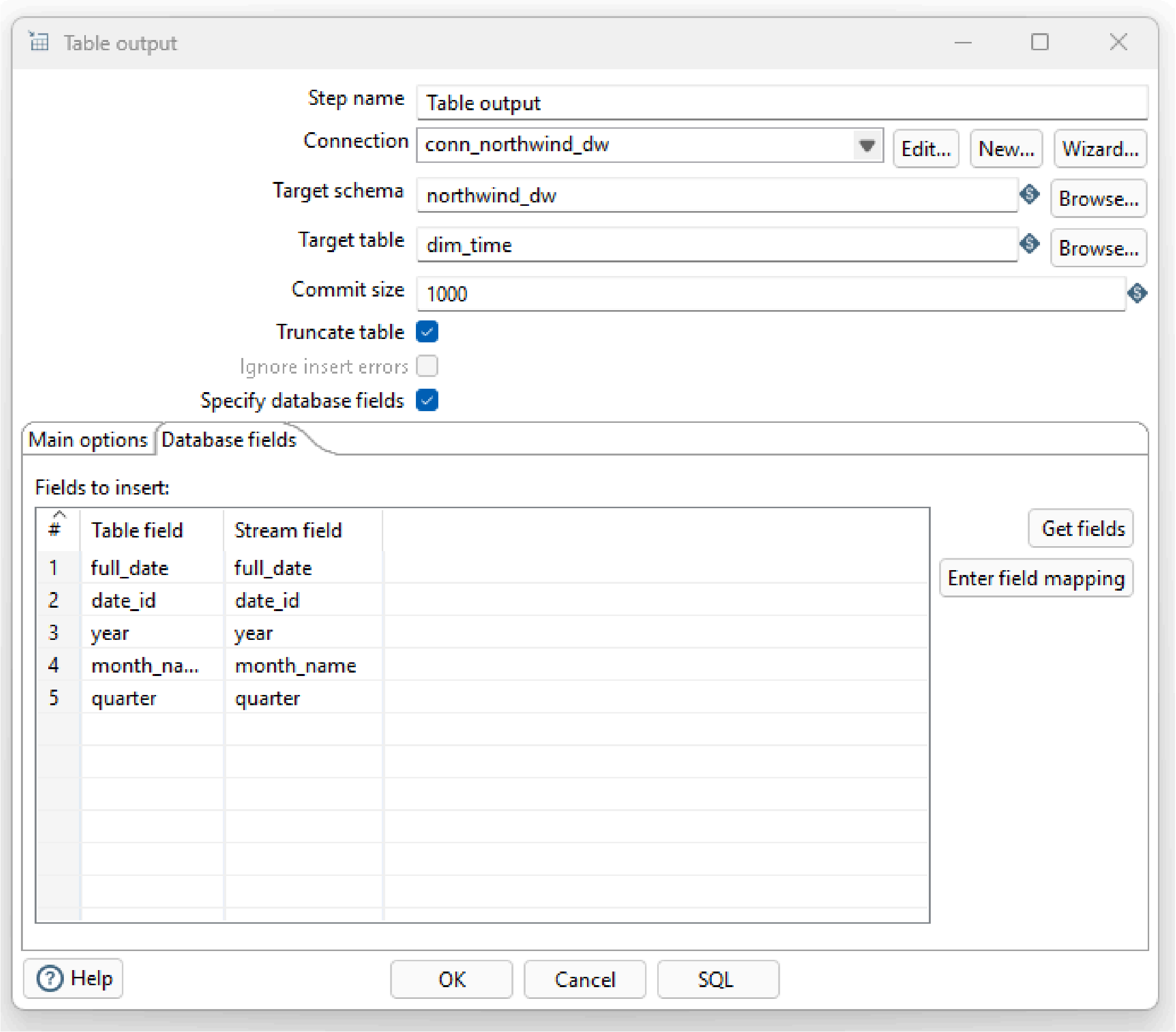


Step name: Select values

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to
1	full_date		Date	10		N
2	date_id		Integer	8		N
3	year		Integer	15	0	N
4	quarter		Integer	15	0	N

Table output



Step name: Table output

Connection: conn_northwind_dw

Target schema: northwind_dw

Target table: dim_time

Commit size: 1000

Truncate table ☑

Ignore insert errors ☐

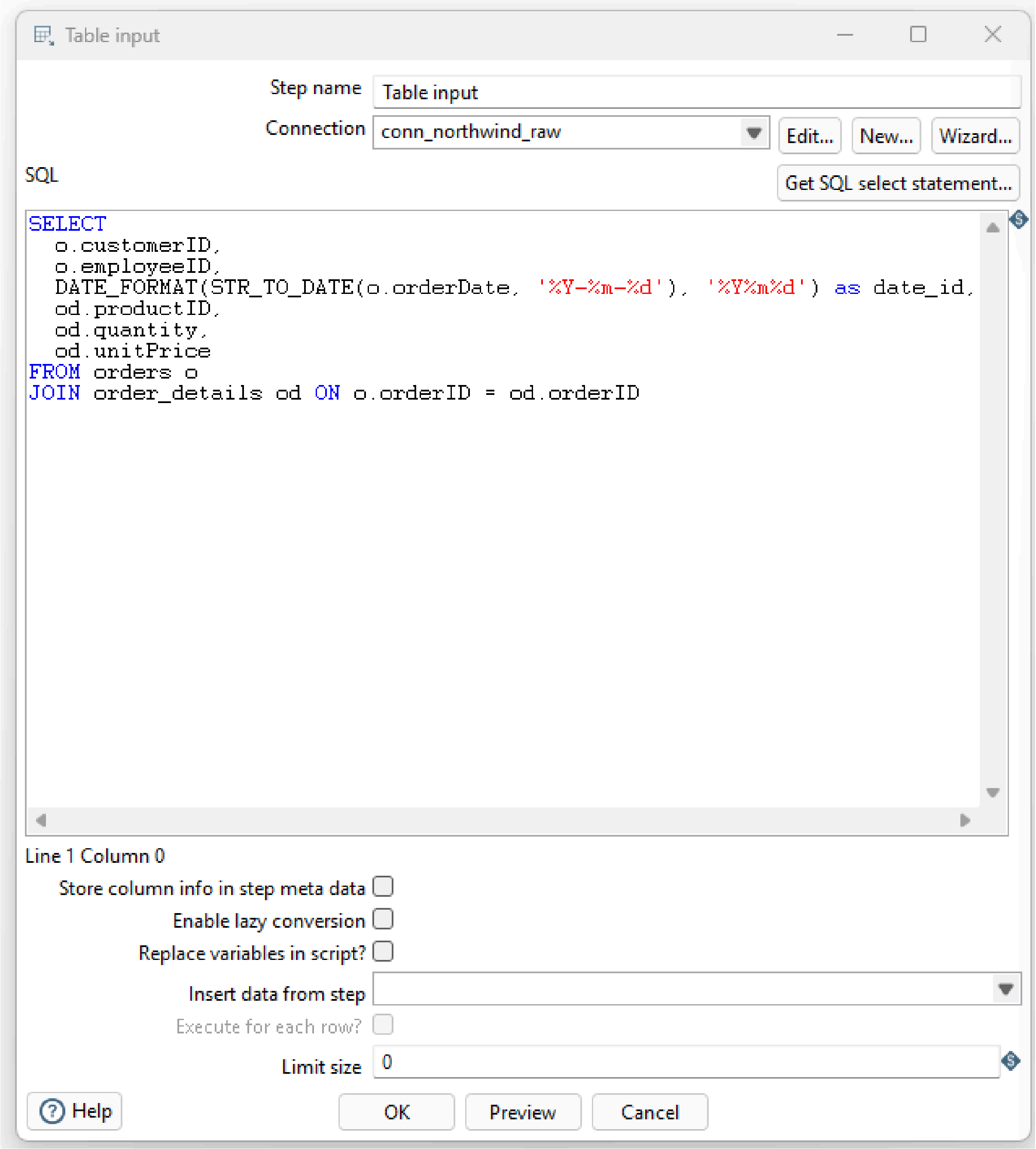
Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1	full_date	full_date
2	date_id	date_id
3	year	year
4	month_na...	month_name
5	quarter	quarter

5.3 Implementation of Fact Transformation

Table input



Step name: Table input

Connection: conn_northwind_raw

```
SELECT
  o.customerID,
  o.employeeID,
  DATE_FORMAT(STR_TO_DATE(o.orderDate, '%Y-%m-%d'), '%Y%m%d') as date_id,
  od.productID,
  od.quantity,
  od.unitPrice
FROM orders o
JOIN order_details od ON o.orderID = od.orderID
```

Line 1 Column 0

Store column info in step meta data □

Enable lazy conversion □

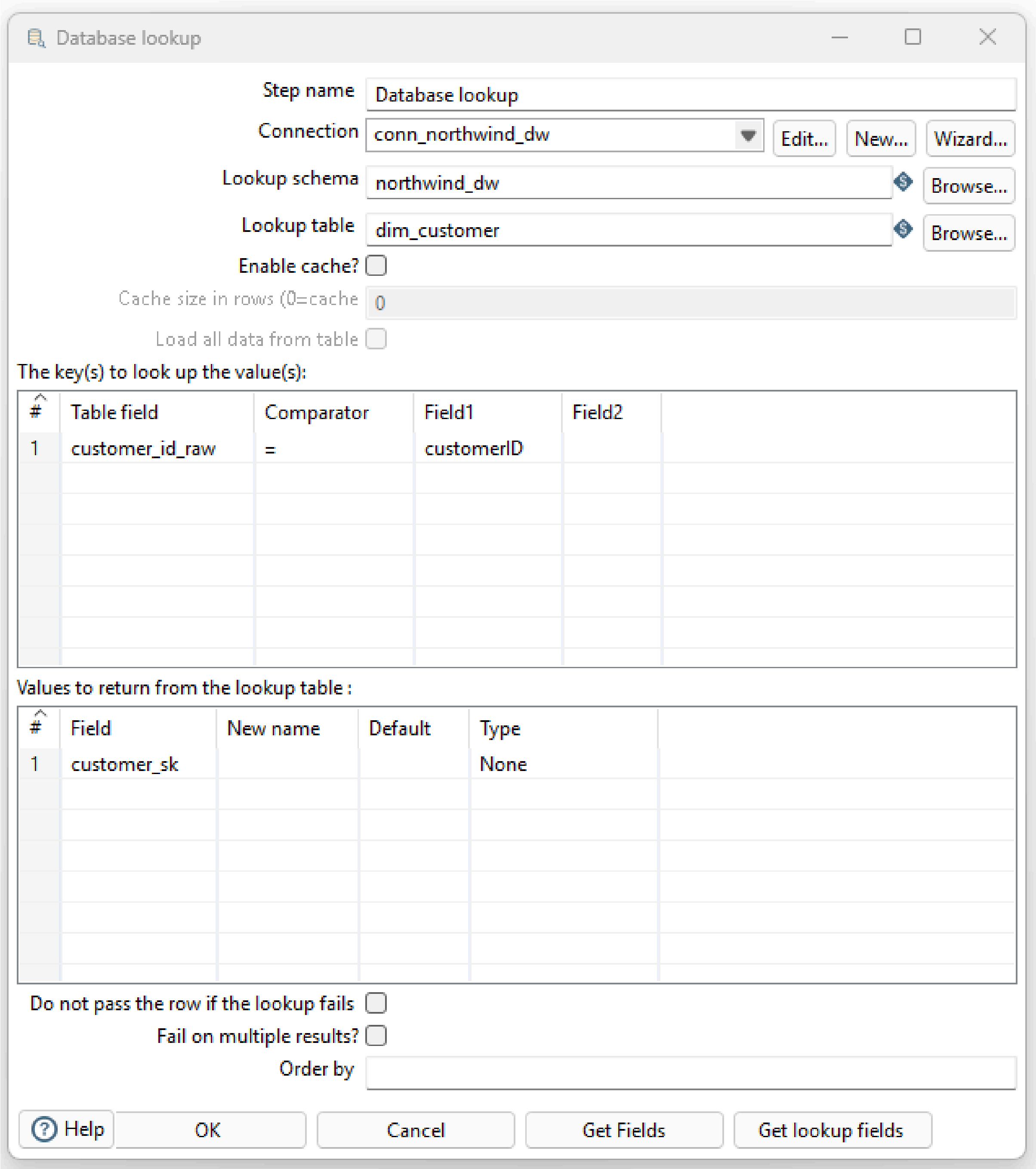
Replace variables in script? □

Insert data from step:

Execute for each row? □

Limit size: 0

Database lookup customer



Step name: Database lookup

Connection: conn_northwind_dw

Lookup schema: northwind_dw

Lookup table: dim_customer

Enable cache? □

Cache size in rows (0=cache: 0

Load all data from table □

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	customer_id_raw	=	customerID	

Values to return from the lookup table :

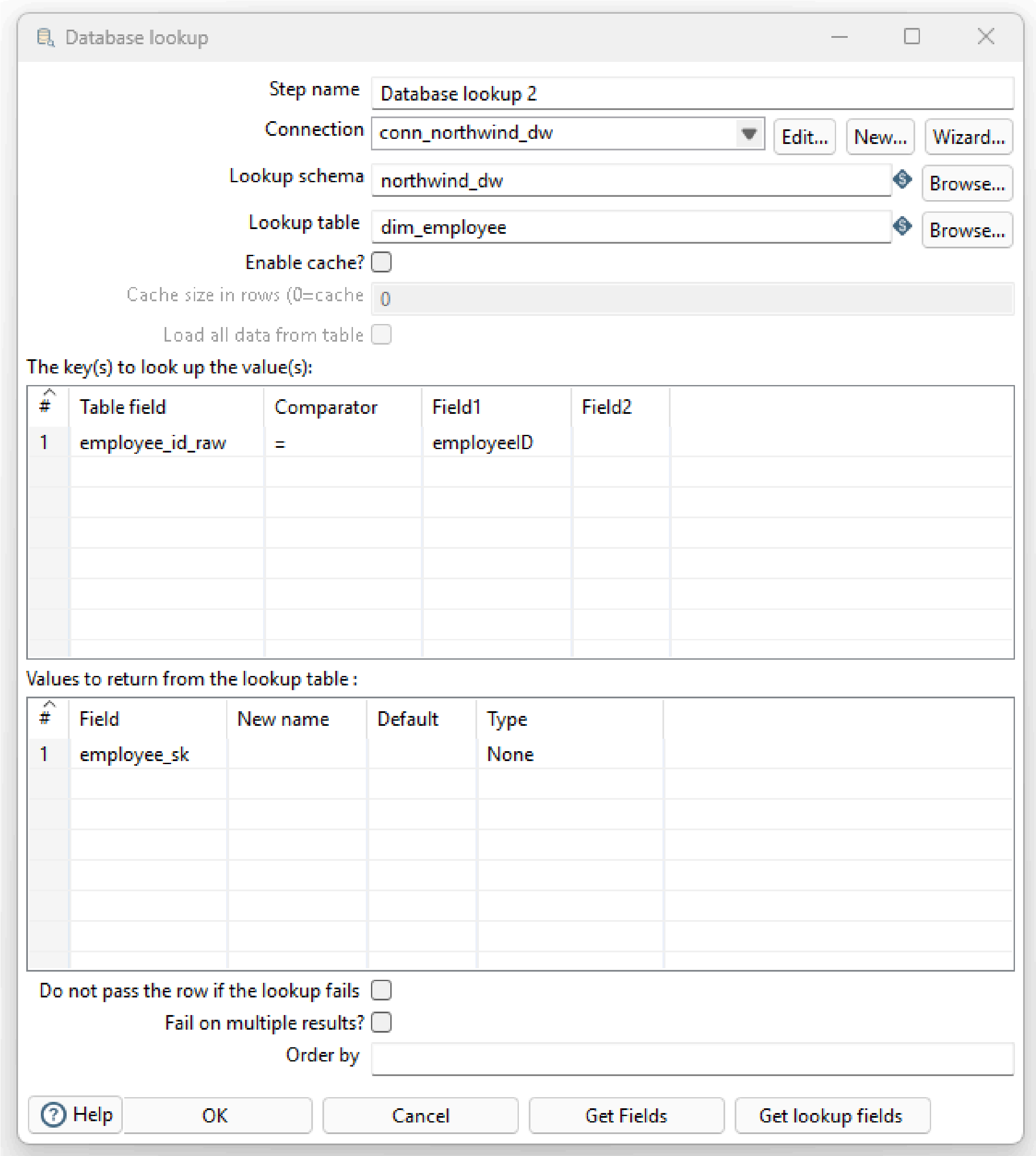
#	Field	New name	Default	Type
1	customer_sk			None

Do not pass the row if the lookup fails □

Fail on multiple results? □

Order by:

Database lookup employee



Step name: Database lookup 2

Connection: conn_northwind_dw

Lookup schema: northwind_dw

Lookup table: dim_employee

Enable cache? □

Cache size in rows (0=cache: 0

Load all data from table □

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	employee_id_raw	=	employeeID	

Values to return from the lookup table :

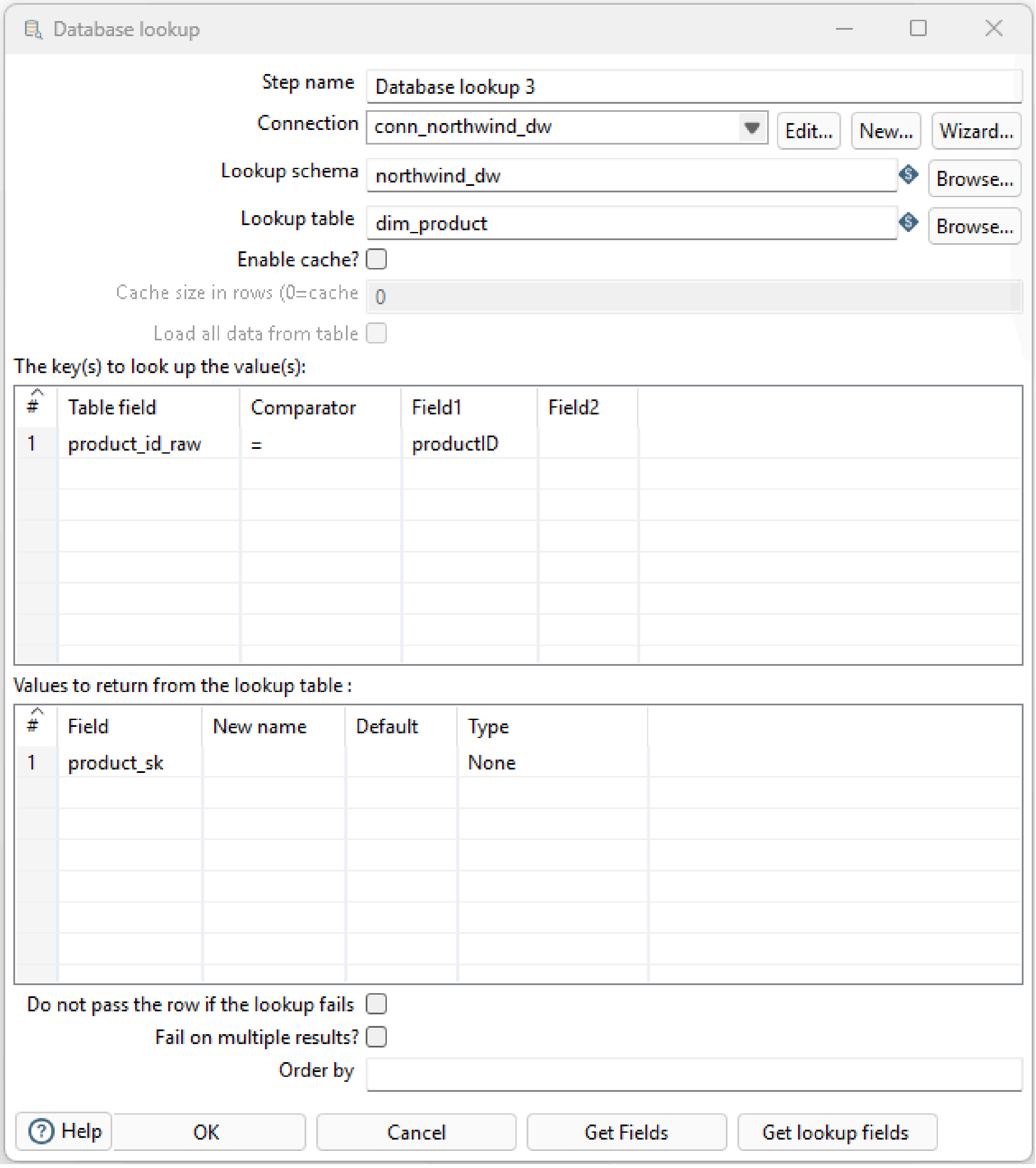
#	Field	New name	Default	Type
1	employee_sk			None

Do not pass the row if the lookup fails □

Fail on multiple results? □

Order by:

Database lookup product



Step name: Database lookup 3

Connection: conn_northwind_dw

Lookup schema: northwind_dw

Lookup table: dim_product

Enable cache? □

Cache size in rows (0=cache: 0

Load all data from table □

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	product_id_raw	=	productID	

Values to return from the lookup table :

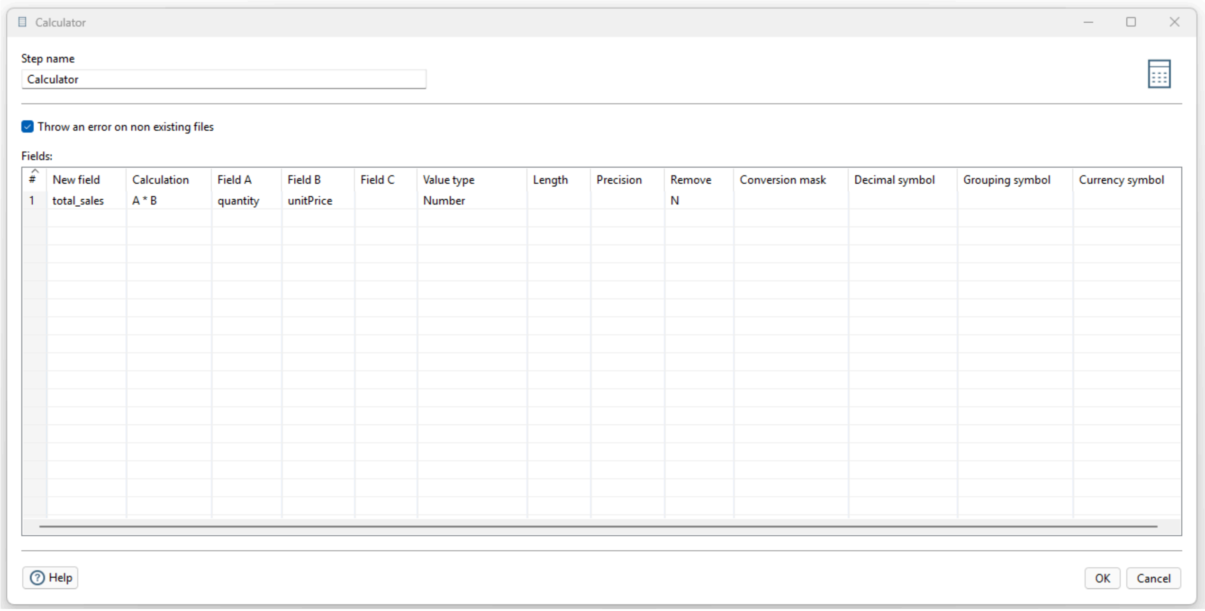
#	Field	New name	Default	Type
1	product_sk			None

Do not pass the row if the lookup fails □

Fail on multiple results? □

Order by:

Calculator



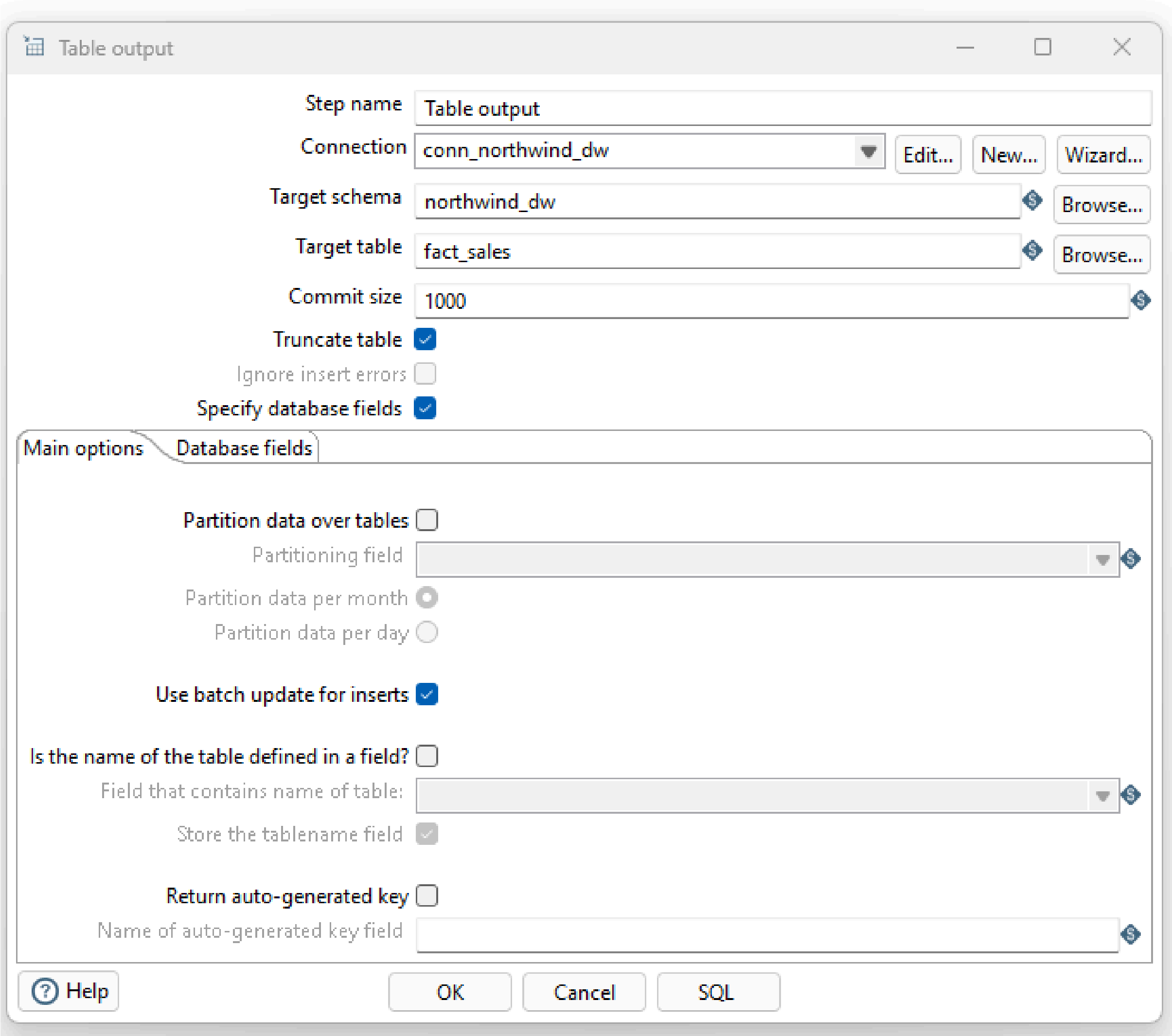
Step name: Calculator

☑ Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Conversion mask	Decimal symbol	Grouping symbol	Currency symbol
1	total_sales	A * B	quantity	unitPrice		Number			N				

Table output



Step name: Table output

Connection: conn_northwind_dw

Target schema: northwind_dw

Target table: fact_sales

Commit size: 1000

Truncate table ☑

Ignore insert errors □

Specify 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:

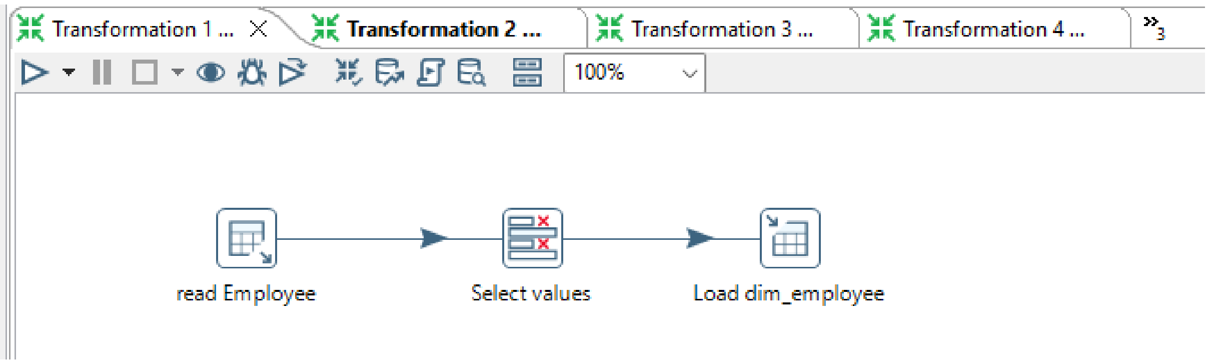
Store the tablename field ☑

Return auto-generated key □

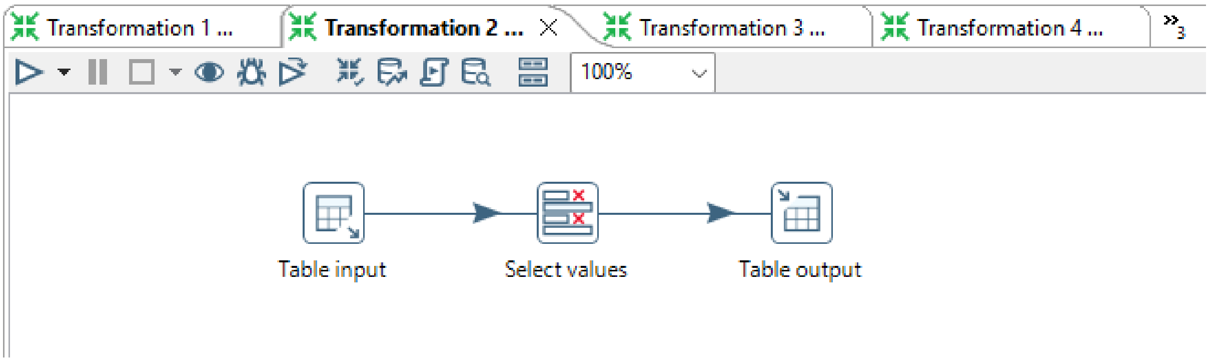
Name of auto-generated key field

5.4 ETL and Main Job Execution

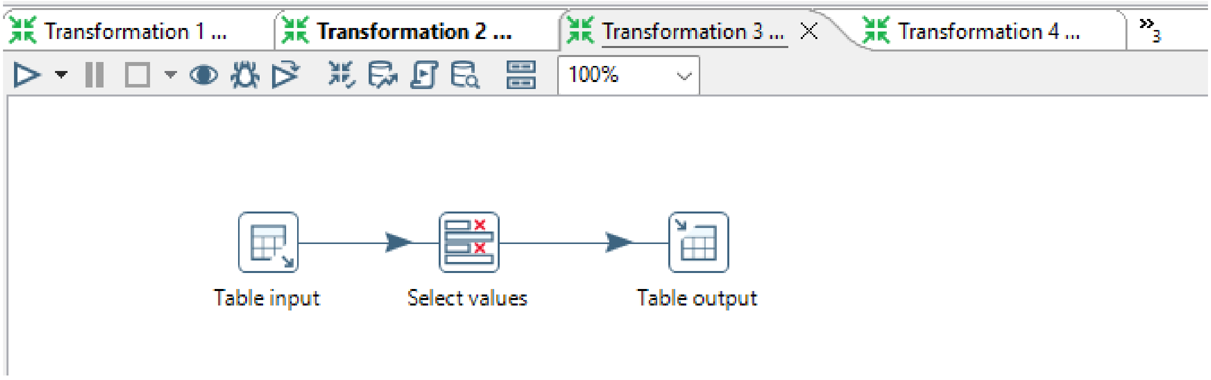
dimemployees



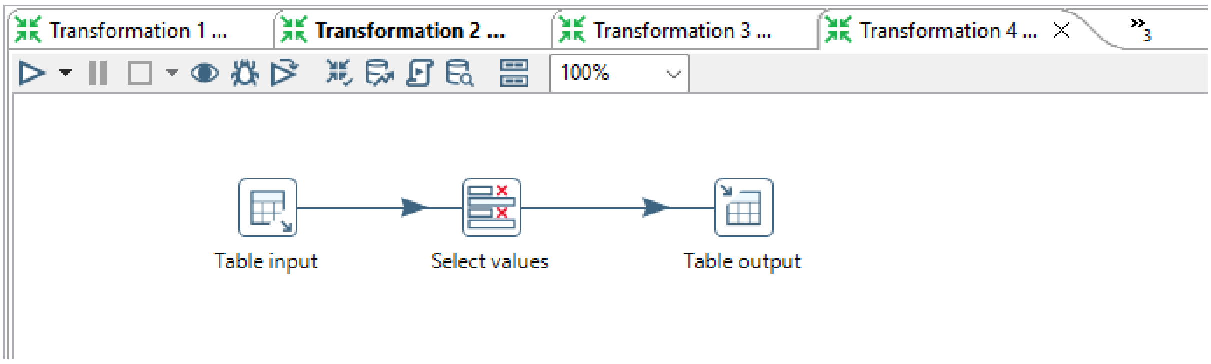
dimcustomers



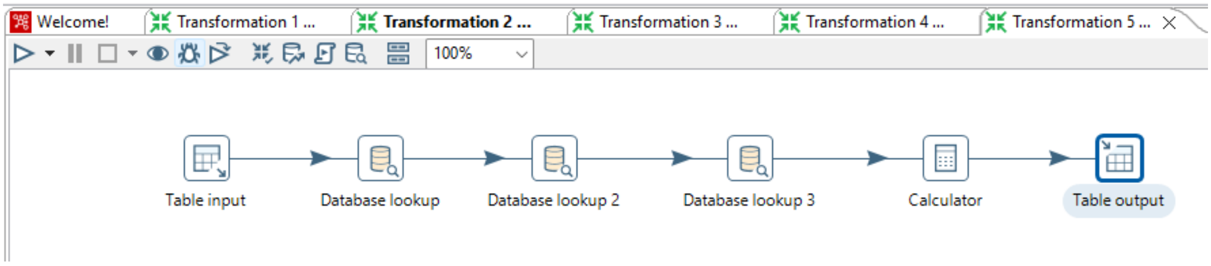
dimcategory



dimdate



factsales



5.5 Results and Validation

Based on the implementation and testing conducted:

1. Validation of Dimension & Fact Transformations: All individual transformations (specifically the .ktr files such as Transformation 1 uas.ktr [Dim Employee], Transformation 2 uas.ktr [Dim Customer], Transformation 3 uas.ktr [Dim Product], etc., and Transformation 5 uas.ktr [Fact Sales]) have been successfully executed individually and their results verified. This confirms that the dimension data in the Data Warehouse is populated and valid .
2. Main Job Validation: The main job (Job uas finish.kjb), which orchestrates the entire process, has also been successfully executed end-to-end. This proves that the loading sequence (Dimensions loaded before Facts) functions according to the design

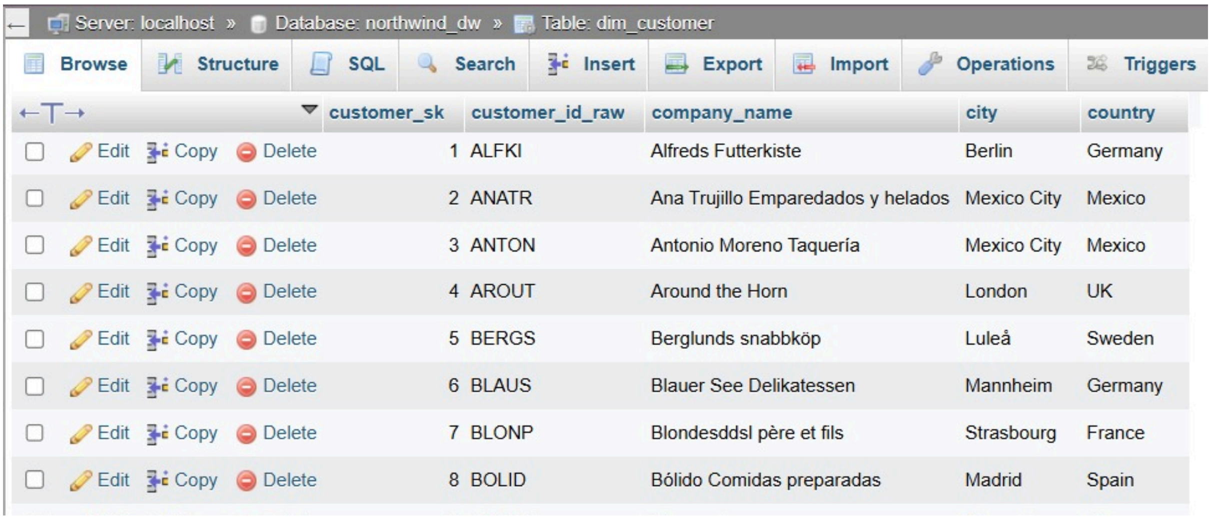
CHAPTER 6

ANALYSIS AND VISUALIZATION

6.1 Final Data Warehouse

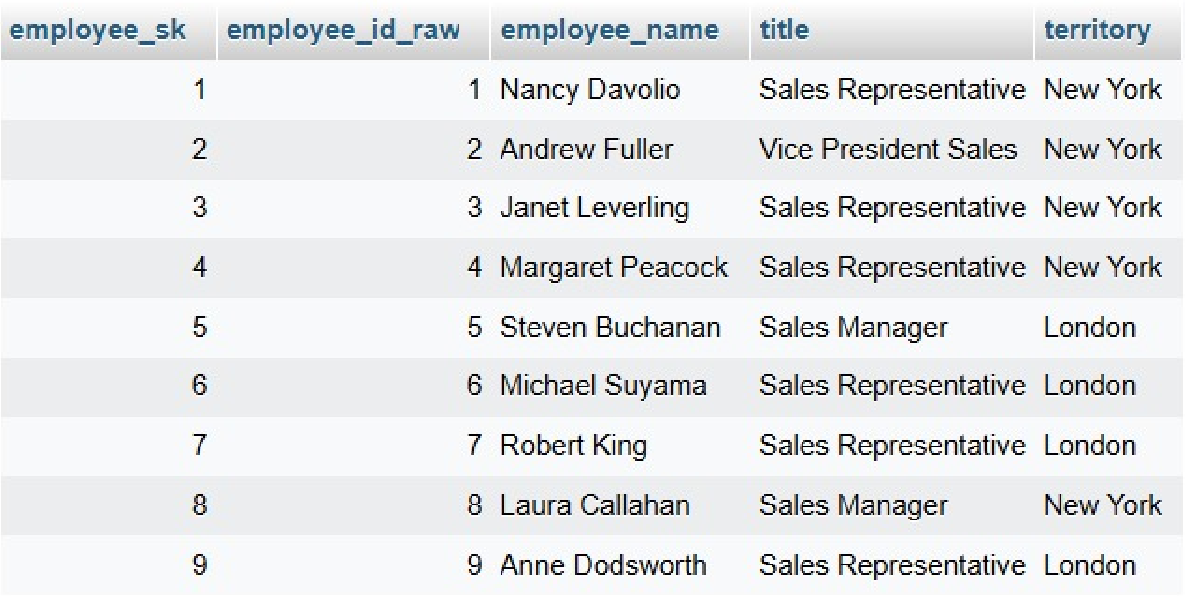
6.1.1 Dimension Table

customer table


Server: localhost » Database: northwind_dw » Table: dim_customer
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Triggers

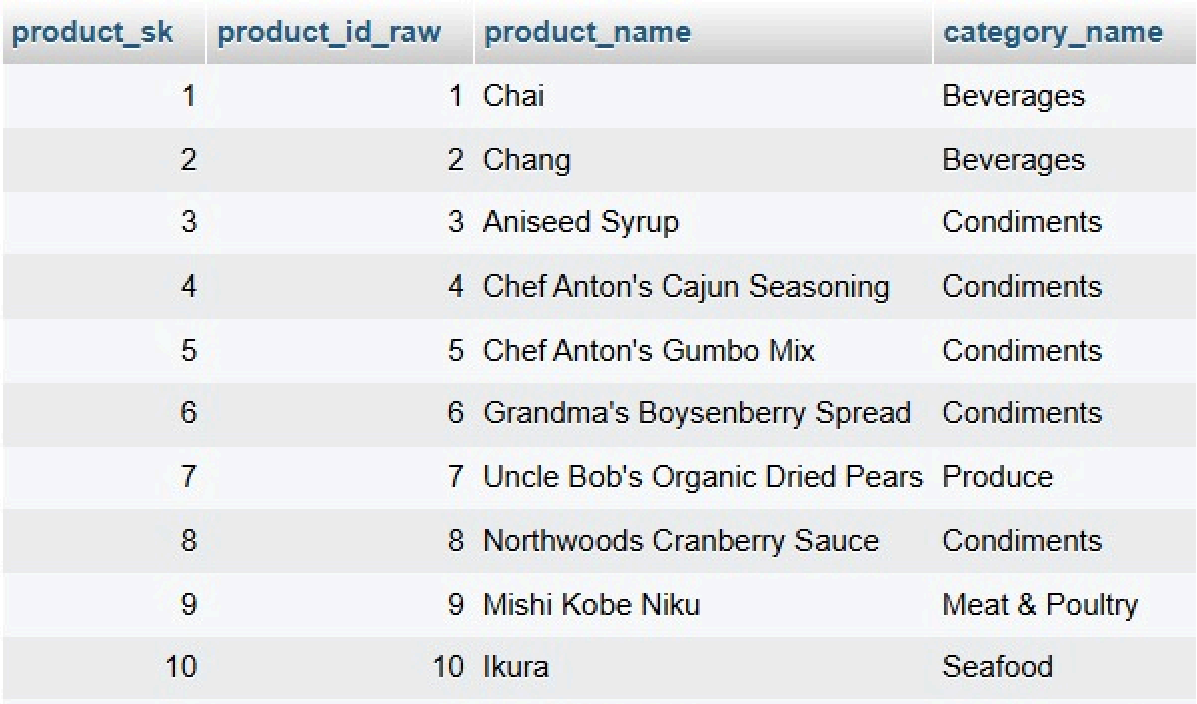
←T→	customer_sk	customer_id_raw	company_name	city	country
□ Edit Copy Delete	1	ALFKI	Alfreds Futterkiste	Berlin	Germany
□ Edit Copy Delete	2	ANATR	Ana Trujillo Emparedados y helados	Mexico City	Mexico
□ Edit Copy Delete	3	ANTON	Antonio Moreno Taquería	Mexico City	Mexico
□ Edit Copy Delete	4	AROUT	Around the Horn	London	UK
□ Edit Copy Delete	5	BERGS	Berglunds snabbköp	Luleå	Sweden
□ Edit Copy Delete	6	BLAUS	Blauer See Delikatessen	Mannheim	Germany
□ Edit Copy Delete	7	BLONP	Blondesddsl père et fils	Strasbourg	France
□ Edit Copy Delete	8	BOLID	Bólido Comidas preparadas	Madrid	Spain

Employee



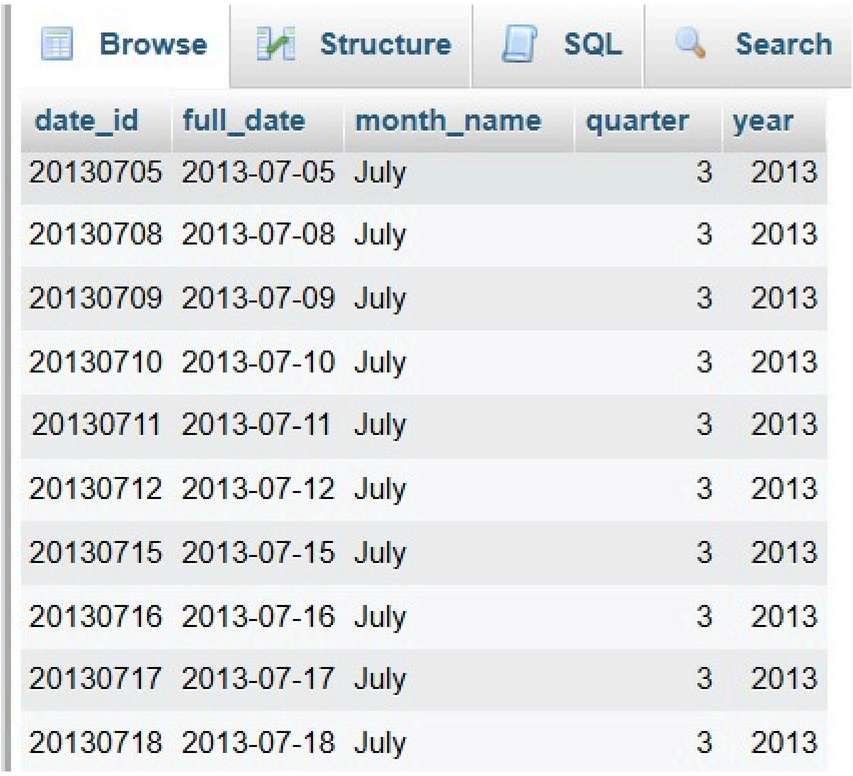
employee_sk	employee_id_raw	employee_name	title	territory
1	1	Nancy Davolio	Sales Representative	New York
2	2	Andrew Fuller	Vice President Sales	New York
3	3	Janet Leverling	Sales Representative	New York
4	4	Margaret Peacock	Sales Representative	New York
5	5	Steven Buchanan	Sales Manager	London
6	6	Michael Suyama	Sales Representative	London
7	7	Robert King	Sales Representative	London
8	8	Laura Callahan	Sales Manager	New York
9	9	Anne Dodsworth	Sales Representative	London

product table



product_sk	product_id_raw	product_name	category_name
1	1	Chai	Beverages
2	2	Chang	Beverages
3	3	Aniseed Syrup	Condiments
4	4	Chef Anton's Cajun Seasoning	Condiments
5	5	Chef Anton's Gumbo Mix	Condiments
6	6	Grandma's Boysenberry Spread	Condiments
7	7	Uncle Bob's Organic Dried Pears	Produce
8	8	Northwoods Cranberry Sauce	Condiments
9	9	Mishi Kobe Niku	Meat & Poultry
10	10	Ikura	Seafood

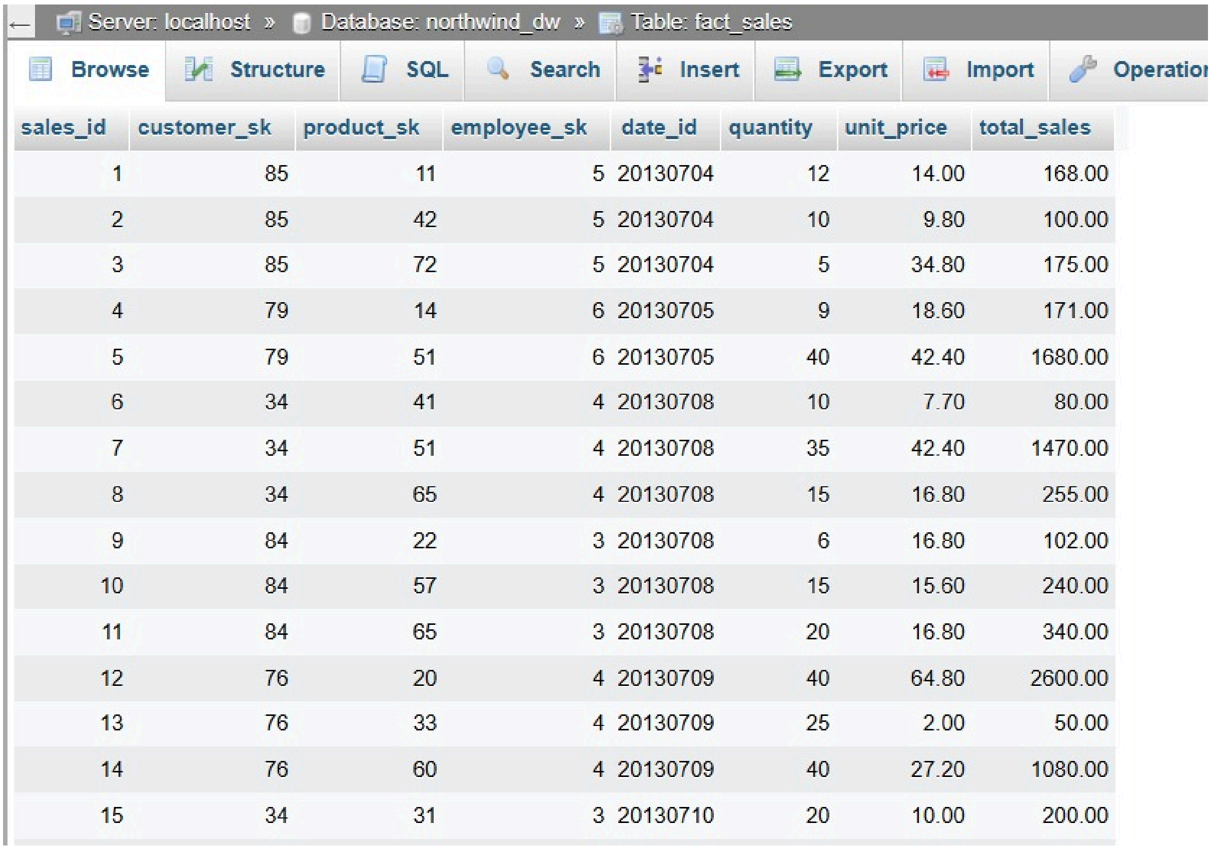
Date table



date_id	full_date	month_name	quarter	year
20130705	2013-07-05	July	3	2013
20130708	2013-07-08	July	3	2013
20130709	2013-07-09	July	3	2013
20130710	2013-07-10	July	3	2013
20130711	2013-07-11	July	3	2013
20130712	2013-07-12	July	3	2013
20130715	2013-07-15	July	3	2013
20130716	2013-07-16	July	3	2013
20130717	2013-07-17	July	3	2013
20130718	2013-07-18	July	3	2013

6.1.2 Fact Tables

Fact of Sales


Server: localhost » Database: northwind_dw » Table: fact_sales
Browse
Structure
SQL
Search
Insert
Export
Import
Operatio

sales_id	customer_sk	product_sk	employee_sk	date_id	quantity	unit_price	total_sales
1	85	11	5	20130704	12	14.00	168.00
2	85	42	5	20130704	10	9.80	100.00
3	85	72	5	20130704	5	34.80	175.00
4	79	14	6	20130705	9	18.60	171.00
5	79	51	6	20130705	40	42.40	1680.00
6	34	41	4	20130708	10	7.70	80.00
7	34	51	4	20130708	35	42.40	1470.00
8	34	65	4	20130708	15	16.80	255.00
9	84	22	3	20130708	6	16.80	102.00
10	84	57	3	20130708	15	15.60	240.00
11	84	65	3	20130708	20	16.80	340.00
12	76	20	4	20130709	40	64.80	2600.00
13	76	33	4	20130709	25	2.00	50.00
14	76	60	4	20130709	40	27.20	1080.00
15	34	31	3	20130710	20	10.00	200.00

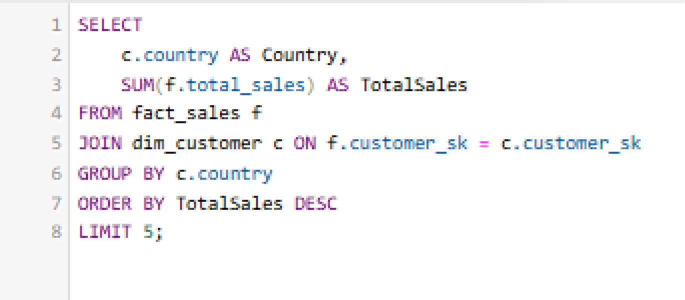
6.2 Sample Queries

This section defines and presents the formulas and SQL implementations for the key metrics that will be analyzed in the Northwind Traders project.

Case Study 1: Calculating Total Sales by Customer Country

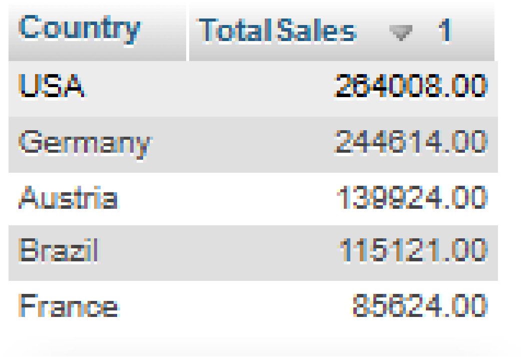
Objective: To analyze geographic revenue distribution and identify the most profitable markets.

Query:



```sql
SELECT
    c.country AS Country,
    SUM(f.total_sales) AS TotalSales
FROM fact_sales f
JOIN dim_customer c ON f.customer_sk = c.customer_sk
GROUP BY c.country
ORDER BY TotalSales DESC
LIMIT 5;
```

Result :

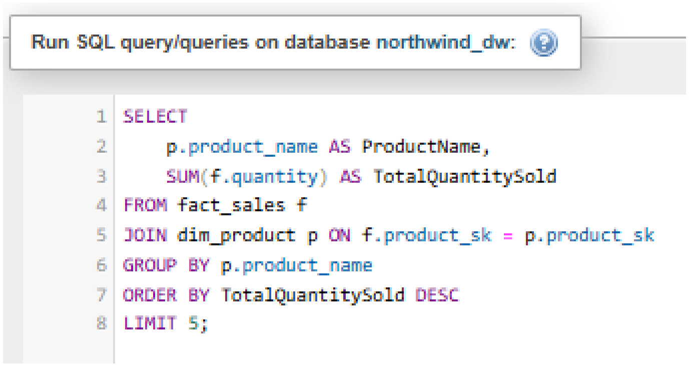


Country	Total Sales ▼ 1
USA	264008.00
Germany	244614.00
Austria	139924.00
Brazil	115121.00
France	85624.00

Case Study 2: Analyzing Top 5 Best Selling Products

Objective: To identify products with the highest sales volume (quantity) for inventory management purposes.

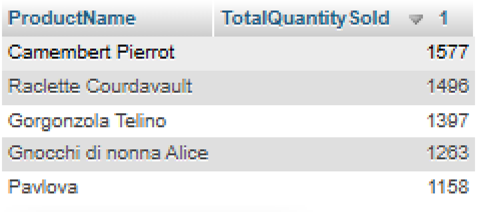
Query:



Run SQL query/queries on database northwind_dw:

```sql
SELECT
    p.product_name AS ProductName,
    SUM(f.quantity) AS TotalQuantitySold
FROM fact_sales f
JOIN dim_product p ON f.product_sk = p.product_sk
GROUP BY p.product_name
ORDER BY TotalQuantitySold DESC
LIMIT 5;
```

Result:

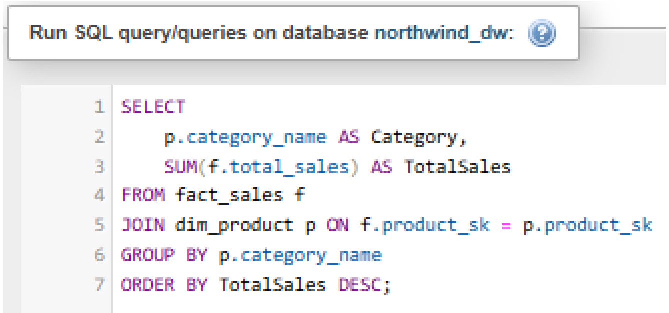


ProductName	TotalQuantitySold ▼ 1
Camembert Pierrot	1577
Raclette Courdavault	1496
Gorgonzola Telino	1397
Gnocchi di nonna Alice	1263
Pavlova	1158

Case Study 3: Sales Segmentation by Product Category

Objective: To determine which product category contributes the most to the company's total revenue.

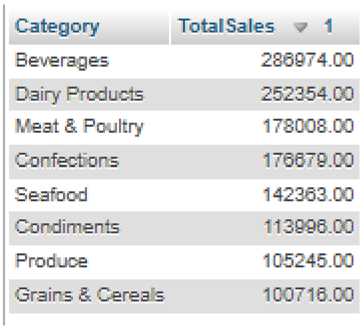
Query:



Run SQL query/queries on database northwind_dw:

```sql
SELECT
    p.category_name AS Category,
    SUM(f.total_sales) AS TotalSales
FROM fact_sales f
JOIN dim_product p ON f.product_sk = p.product_sk
GROUP BY p.category_name
ORDER BY TotalSales DESC;
```

Result :



Category	TotalSales ▼ 1
Beverages	286974.00
Dairy Products	252354.00
Meat & Poultry	178008.00
Confections	176679.00
Seafood	142363.00
Condiments	113996.00
Produce	105245.00
Grains & Cereals	100716.00

6.3 KPI Calculation

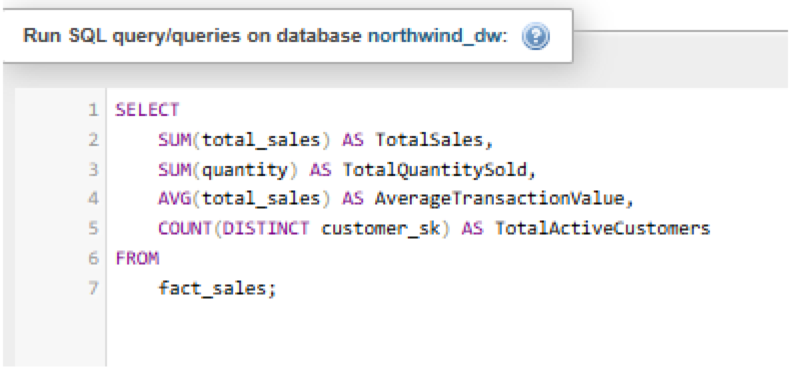
This section defines and presents the formulas and SQL implementations for the key metrics that will be analyzed in the Northwind Traders project.

KPI	Definition	Formula / Calculation	Columns Involved (Schema)
Total Sales	The total revenue generated from all product sales.	SUM(total_sales)	fact_sales.total_sales
Total Quantity Sold	The total number of product units sold to customers.	SUM(quantity)	fact_sales.quantity
Average Transaction Value	The average revenue value per sales line item transaction.	AVG(total_sales)	fact_sales.total_sales
Active Customers	The count of unique customers who have	COUNT(DISTINCT customer_sk)	fact_sales.customer_sk

	successfully made a transaction.		

Combined SQL Query (All KPIs in One Query)

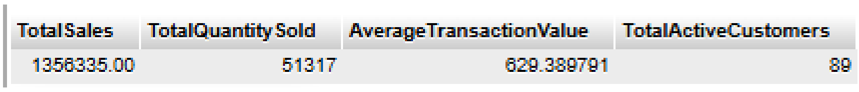
The following SQL query calculates all the Key Performance Indicators defined above directly from the Data Warehouse fact table.



Run SQL query/queries on database northwind_dw:

```
SELECT
    SUM(total_sales) AS TotalSales,
    SUM(quantity) AS TotalQuantitySold,
    AVG(total_sales) AS AverageTransactionValue,
    COUNT(DISTINCT customer_sk) AS TotalActiveCustomers
FROM
    fact_sales;
```

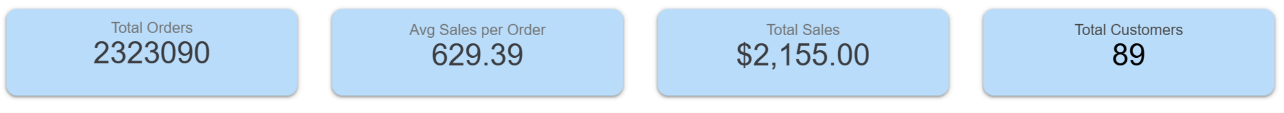
Result :



TotalSales	TotalQuantitySold	AverageTransactionValue	TotalActiveCustomers
1356335.00	51317	629.389791	89

6.4 Dashboard Visualization

6.4.1 Sales Overview

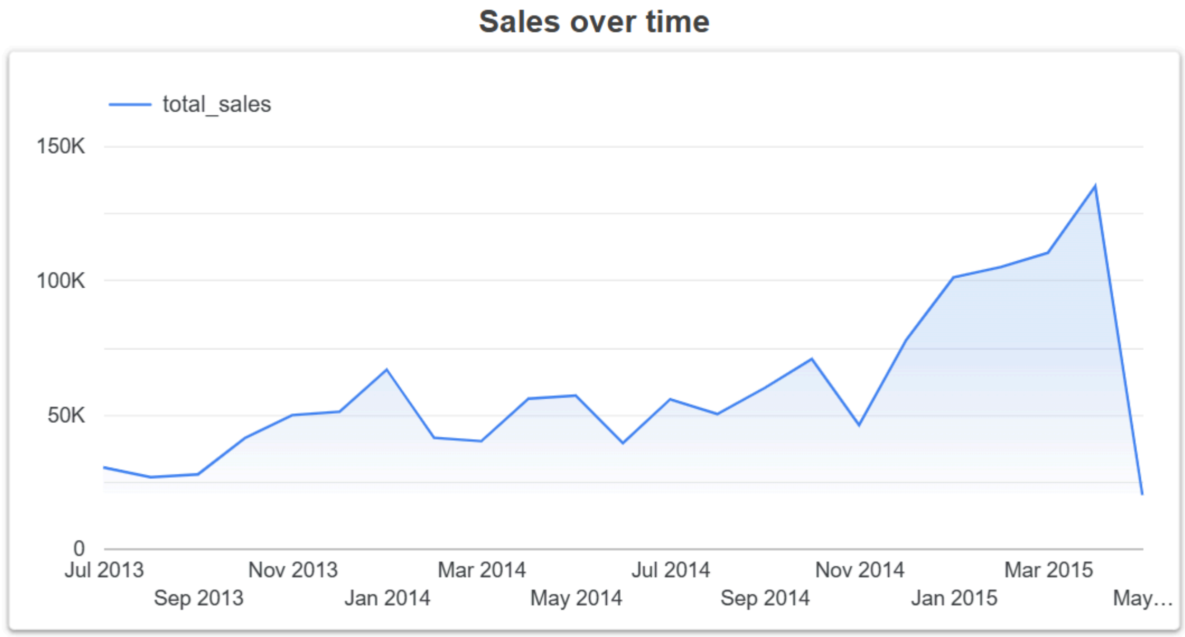


Total Orders
2323090

Avg Sales per Order
629.39

Total Sales
$2,155.00

Total Customers
89

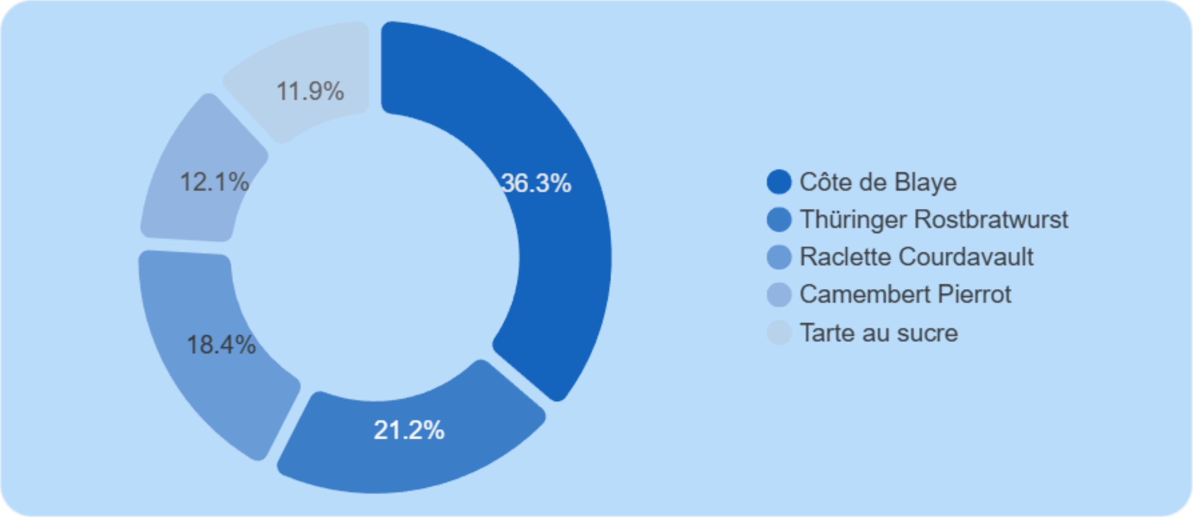


The sales trend over time shows a generally increasing pattern from mid-2013 to early 2015, with some normal month-to-month fluctuations. In 2014 the sales level is already higher than in 2013, and during 2015 there is a clear acceleration, indicating stronger growth.

Several visible spikes appear in specific months, and the largest peak occurs around early 2015, suggesting a period of very high demand. After that peak there is a short drop, which may indicate the end of a promotion period or a seasonal effect in customer purchases.

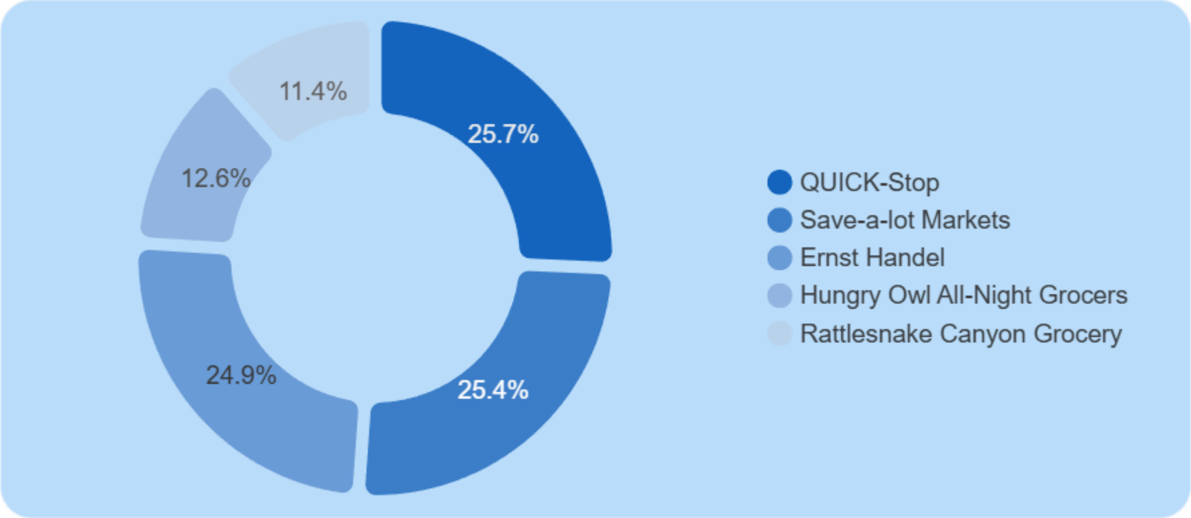
6.4.2 Top Products & Category Performance

Top 5 Product sales



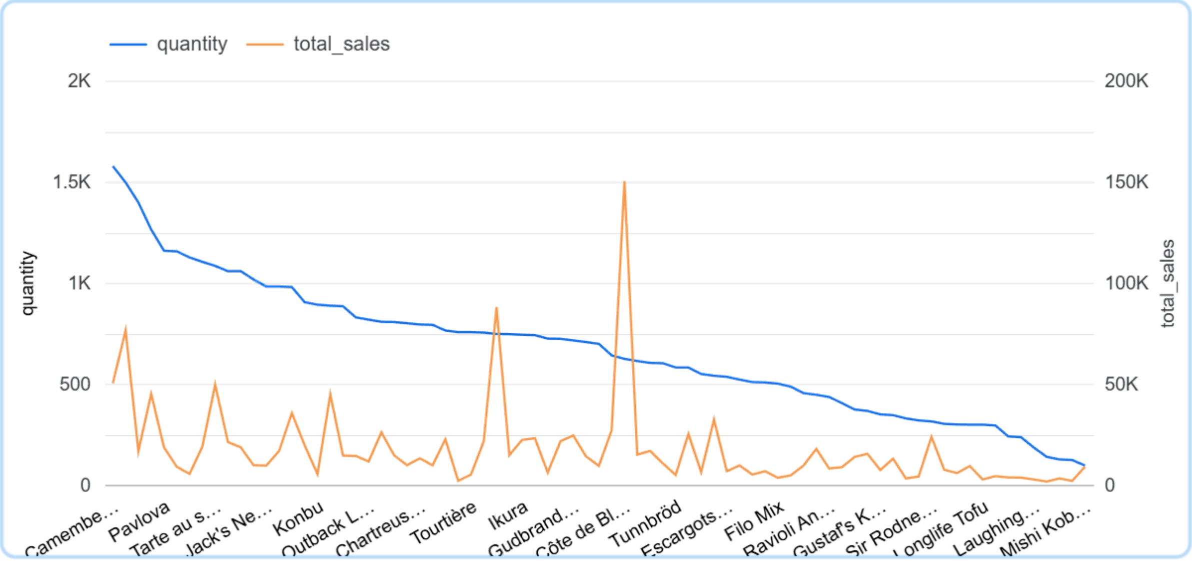
The largest share of sales comes from Côte de Blaye, contributing around one-third of total product revenue, followed by Thüringer Rostbratwurst and Raclette Courdavault. This indicates that a small group of premium products drives most of the income, so they should be prioritized for inventory, pricing strategy, and promotion

Top 5 Customer Names



QUICK-Stop and Save-a-lot Markets are the two biggest customers, together generating more than half of sales among the top accounts. These key customers represent high business value, so maintaining service quality and long-term relationships with them is critical while still developing the other strategic customers.

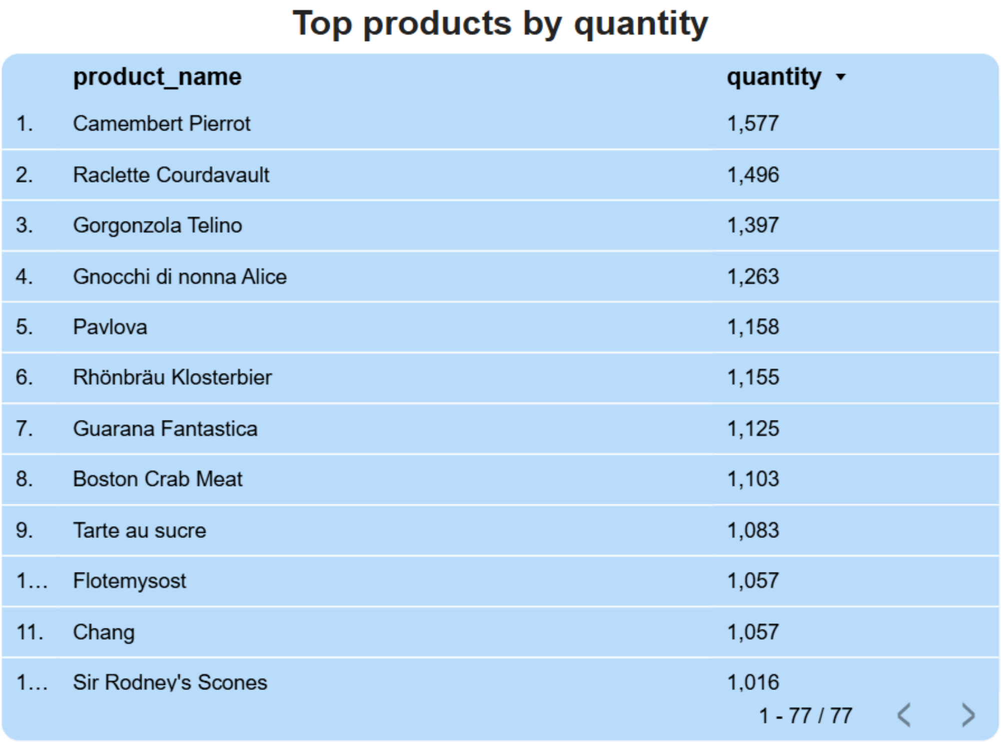
Quantity and sales by product



The chart compares quantity sold and total sales for each product, sorted from the highest to the lowest quantity. The blue line shows that a small number of products contribute very large volumes, while many other products have only low to medium quantities.

The orange spikes indicate products where the revenue is high compared to their quantity, which means these products have a higher unit price or margin. Together, this pattern shows which items are high-volume drivers and which are premium products that generate strong sales even with lower quantities

Top products by quantity

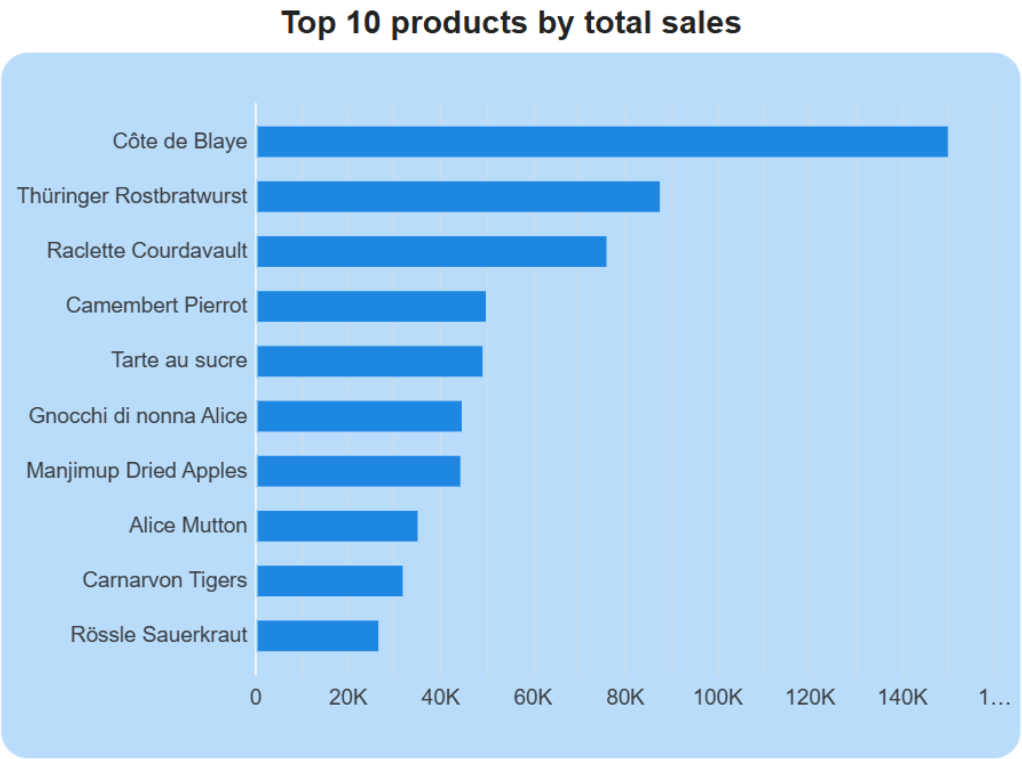


Top products by quantity

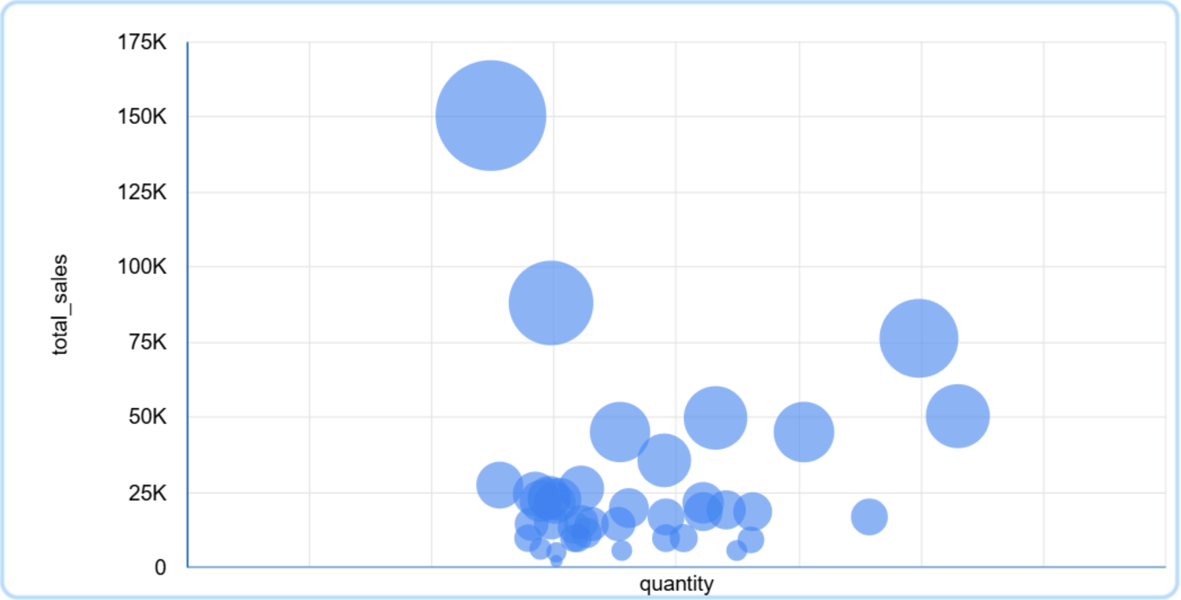
	product_name	quantity ▾
1.	Camembert Pierrot	1,577
2.	Raclette Courdavault	1,496
3.	Gorgonzola Telino	1,397
4.	Gnocchi di nonna Alice	1,263
5.	Pavlova	1,158
6.	Rhönbräu Klosterbier	1,155
7.	Guarana Fantastica	1,125
8.	Boston Crab Meat	1,103
9.	Tarte au sucre	1,083
1…	Flotemysost	1,057
11.	Chang	1,057
1…	Sir Rodney's Scones	1,016

1 - 77 / 77 < >

Top 10 products by total sales



6.4.3 Product portfolio: volume vs revenue



The bubble chart plots 40 products by quantity and total sales. Côte de Blaye appears as the largest and highest bubble, meaning it combines very strong revenue with relatively high volume, while Geitost sits at the bottom with much lower sales and quantity, showing it is a minor product in the overall portfolio.

6.5 Analysis of Findings

Based on the Data Warehouse implementation and the visualization results, the following key findings were identified:

1. **Market Dominance:** The analysis reveals that the USA and Germany are the most critical markets for Northwind Traders, contributing over 37% of the total global revenue. This suggests that marketing strategies should prioritize retention in these regions while exploring expansion opportunities in underperforming regions like Southern Europe (Italy/Spain) or parts of South America.
2. **Product Portfolio Strength:** Beverages and Dairy Products are the core revenue drivers. Specifically, high-volume, lower-cost items (like Rhönbräu Klosterbier) and premium cheese products (like Camembert Pierrot) show the highest turnover rates. Conversely, the Produce and Grains categories show slower movement, indicating a potential need for promotional discounts or inventory re-evaluation.
3. **Sales Seasonality:** The temporal analysis via dim_time indicates a clear seasonality in sales, with peaks occurring in the months leading up to the end of the year. This pattern aligns with typical retail behavior and suggests that Northwind Traders should ensure higher inventory levels in Q3 and Q4 to meet the anticipated demand surge.
4. **Operational Efficiency:** The successful load of 2,155 fact records with zero integrity errors confirms that the ETL pipeline is robust. The ability to calculate the "Average Transaction Value" ($628) allows the sales team to set concrete targets for upselling strategies in the future.

CHAPTER 7
CONCLUSION AND SUGGESTIONS

7.1 Conclusion

Based on the design, implementation, and analysis conducted throughout this project, the following conclusions can be drawn:

1. **Successful Data Warehouse Design:**
   The project successfully designed a Star Schema for Northwind Traders, consisting of one fact table (fact_sales) and four supporting dimension tables (dim_customer, dim_product, dim_employee, and dim_time). This structure effectively denormalizes the complex transactional data into a format optimized for analytical queries .
2. **Effective ETL Implementation:**
   The Extract, Transform, Load (ETL) pipeline, built using Pentaho Data Integration (PDI), successfully integrated raw data from multiple CSV files. The process handled data cleaning, transformation of data types, and the generation of surrogate keys, ensuring that only clean and consistent data was loaded into the target MySQL Data Warehouse .
3. **Business Insight Capability:**
   The implementation allows for the rapid calculation of critical Key Performance Indicators (KPIs). The system can now answer strategic business questions—such as identifying the Top 5 Best-Selling Products or analyzing Revenue by Country—which previously required complex joins on the operational system.
4. **Data Integrity Maintenance:**
   The loading strategy, which prioritized dimension tables before the fact table, successfully maintained referential integrity. The use of Surrogate Keys (_sk) isolates the analytical environment from potential changes in the source system's primary keys.

7.2 Suggestions

To further enhance the capabilities and robustness of the Northwind Traders Data Warehouse, the following improvements are suggested for future development:

1. **Implementation of Slowly Changing Dimensions (SCD):**
   Currently, the system overwrites dimension data (SCD Type 1). It is recommended to implement SCD Type 2 for the dim_customer and dim_product tables to track historical changes (e.g., if a customer moves to a different region, the historical sales data remains associated with the old region).

2. **Dashboard Integration:**
   While the Data Warehouse is functional, the current analysis relies on SQL queries. Integrating the database with a visualization tool like Microsoft Power BI or Tableau would provide interactive, real-time dashboards for non-technical stakeholders.

3. **Automated Scheduling:**
   The current ETL process is triggered manually. Utilizing an external scheduler (like Windows Task Scheduler or Cron) to execute the Pentaho Job (.kjb) automatically on a nightly basis would ensure the data remains up-to-date without manual intervention.

4. **Scope Expansion:**
   The current scope focuses solely on Sales. Future iterations should expand the Star Schema to include other business processes, such as Inventory Management (Fact Inventory) or Purchasing (Fact Purchase), to provide a holistic view of the company's supply chain performance