|  |
| --- |
| Business Template  **SD POINT SEAFOOD RETAIL SUPPLIER** |
| **Logo / Image** |
|  |

Contents

[**SD POINT SEAFOOD RETAIL SUPPLIER** 1](#_Toc147614425)

[1 Business Description 2](#_Toc147614426)

[1.1 Business background 2](#_Toc147614427)

[1.2 Problems because of poor data management 2](#_Toc147614428)

[1.3 Benefits from implementing a Data Warehouse 2](#_Toc147614429)

[1.4 DATASETS DESCRIPTION 2](#_Toc147614430)

[2 Task 2: Grain / DIM / Fact 4](#_Toc147614431)

[3 Task 3: 3NF MODEL 11](#_Toc147614432)

[4 Task 4: DM MODE 12](#_Toc147614433)

[5 Task 5: LOGICAL SCHEMA AND DATA FLOW 17](#_Toc147614434)

[6 Task 5: data loading from sources to dimension layer 23](#_Toc147614435)

# Business Description

## Business background

SD Point is one the major sea food supplier across Latvia, and Lithuania. Head office is situated in Riga Latvia. Their main customers are restaurants, small scale business, and individual people. Main category of products are fish, Shellfish, Mollusks. They have been working in Latvia for last ten years and have monopoly in Latvian market. They opened branch in Lithuania 4 years before. Since the market in Lithuania is very competitive company decoded to give discounts for most of their products. So, database sources in this project will be Latvian sales and Lithuanian sales operational databases. Main difference between data sources will be Lithuanian sales dataset will contain extra column discount and values of employees, store data will be different.

## Problems because of poor data management

Poor data management doesn’t let to do successful business because of insufficient information about what should you do next. If you don’t use instruments which can give you information for analysis and which can help you to come up with a business strategy you won’t be competitive in this or that field.

## Benefits from implementing a Data Warehouse

Using of data warehouse can help you with the problems described above. Implementing a data warehouse can answer you the following questions:

* Which products have highest demand?
* Which store brings us more income?
* Who is the most valuable customer?
* Which employee made more sales?

Further processing data would also let you:

* Which combination of products sells most?
* Do we have trend in sales per season?
* And many other.

## DATASETS DESCRIPTION

1.The first dataset contains the following information about sales on the Latvian market.

Product\_ID:

This attribute gives the ID of which product sold.

Customer\_First\_Name:

This attribute gives the first name of customer who made the sale.

Customer\_Last\_Name:

This attribute gives the last name of customer who made the sale.

Customer\_ID:

This attribute gives the ID of customer who made the sale.

EMP\_First\_Name:

This attribute gives the first name of employee who made the sale.

EMP\_Last\_Name:

This attribute gives the last name of employee who made the sale.

EMP\_ID:

This attribute gives the ID of employee who made the sale.

Payment\_Method:

This attribute gives details of payment method used for sales.

Channel\_Type:

This attribute gives details of through which channel sales made.

Store\_ID:

This attribute gives store ID of that particular sales occurred.

Shipper\_ID:

This attribute gives the ID of company that responsible for delivery of the product.

Sale\_Date:

This attribute gives the details of the date in which sale occurred.

Quantity\_Kg:

This attribute gives the details of quantity the product in Kg.

Total\_Amount:

This attribute gives the details of total amount paid by the customer.

2. The second dataset contains the following information about sales on the Latvian market.

Product\_ID:

This attribute gives the ID of which product sold.

Customer\_First\_Name:

This attribute gives the first name of customer who made the sale.

Customer\_Last\_Name:

This attribute gives the last name of customer who made the sale.

Customer\_ID:

This attribute gives the ID of customer who made the sale.

EMP\_First\_Name:

This attribute gives the first name of employee who made the sale.

EMP\_Last\_Name:

This attribute gives the last name of employee who made the sale.

EMP\_ID:

This attribute gives the ID of employee who made the sale.

Payment\_Method:

This attribute gives details of payment method used for sales.

Channel\_Type:

This attribute gives details of through which channel sales made.

Store\_ID:

This attribute gives store ID of that particular sales occurred.

Shipper\_ID:

This attribute gives the ID of company that responsible for delivery of the product.

Sale\_Date:

This attribute gives the details of the date in which sale occurred.

Quantity\_Kg:

This attribute gives the details of quantity the product in Kg.

Discount:

This attribute gives the details of the discount applied for the sale.

Total\_Amount:

This attribute gives the details of total amount paid by the customer.

# Task 2: Grain / DIM / Fact

* Grain: In this project I will have a transaction grain fact table. Grain level is one sale made by one customer per one kind of product in one store. In this sea food business company sell products only in kg, so quantity measurement will be in KG.

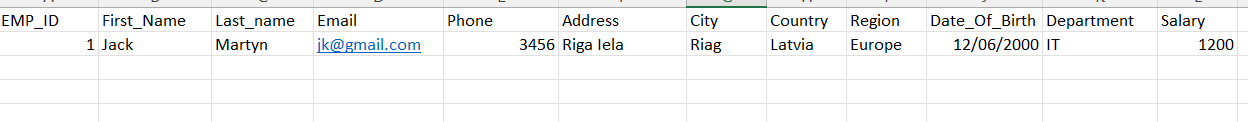
Dim Tables:

Employees

Give employee details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Emp\_ID | Represents ID of each employee, It’s the unique identifier of the table, PK | Bigint |
| First\_Name | First name of employee. | Text |
| Last\_Name | Last name of employee. | Text |
| Email | Email of each employee | Text |
| Phone | Phone number of each employee | Text |
| Address | Address of each employee | Text |
| City | City name of employee | Text |
| Country | Country name of employee | Text |
| Region | Region of employee | Text |
| Date\_Of\_Birth | Date of birth of each employee | Date |
| Department | Name of the department employee belongs to | Text |
| Salary | Salary of each employee | Decimal(10,2) |

Example with filled data

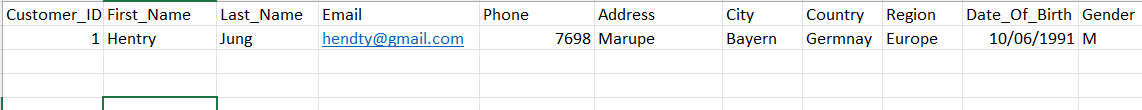


Customers

Give employee details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Customer\_ID | Represents ID of each customer, It’s the unique identifier of the table, PK | Bigint |
| First\_Name | First name of customer. | Text |
| Last\_Name | Last name of customer. | Text |
| Email | Email of each customer | Text |
| Phone | Phone number of each customer. | Text |
| Address | Address of each customer | Text |
| City | City name of customer | Text |
| Country | Country name of customer | Text |
| Region | Region of customer | Text |
| Date\_Of\_Birth | Date of birth of each customer | Date |
| Gender | Gender of each customer | Text |

Example with filled data

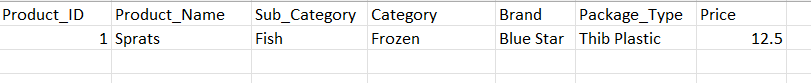


Products

Gives the details of each product of the company

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Product\_ID | Represents ID of each product, It’s the unique identifier of the table, PK | Bigint |
| Product\_Name | Name of each product | Text |
| Sub\_Category | Name of sub category each product belongs | Text |
| Category | Name of category each product belongs | Text |
| Brand | Brand name of product | Text |
| Package\_Type | Package name of product | Text |
| Price | Price of each product | Decimal(10,2) |

Example with filled data

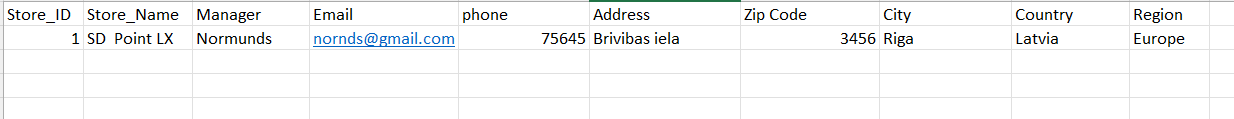


Stores

Give information about all stores of the company

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Store\_ID | Represents ID of each store, It’s the unique identifier of the table, PK | Bigint |
| Store\_Name | Name of each store | Text |
| Manager | Name of manager of particular store | Text |
| Email | Store email id | Text |
| Phone | Store phone number | Text |
| Address | Store address | Text |
| Zip Code | Zip code where store belongs | Text |
| City | Price of each product | Text |
| Country | Country where store belongs | Text |
| Region | Region where store belongs | Text |

Example with filled data

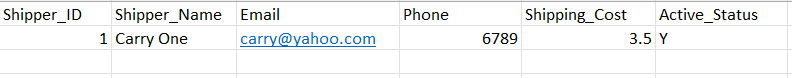


Shippers

Give details of shipping company who are responsible for delivering products

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Shipper\_ID | Represents ID of each shipping company, It’s the unique identifier of the table, PK | Bigint |
| Shipper\_Name | Name of shipping company | Text |
| Email | Contact email id of shipping company | Text |
| Phone | Contact phone of shipping company | Text |
| Shipping\_Cost | Cost of shipping per delivery | Decimal(10,2) |
| Active\_Status | Flag indicating whether the shipper is active or not | Text |

Example with filled data

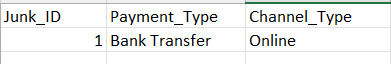


Junks

Give the details of low cardinality attributes

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Junk\_ID | Represents ID of each junk attrubutes, It’s the unique identifier of the table, PK | Bigint |
| Payment\_Type | Details of payment type by customer | Text |
| Channel\_Type | Details of channel type in which sale occuared | Text |

Example with filled data

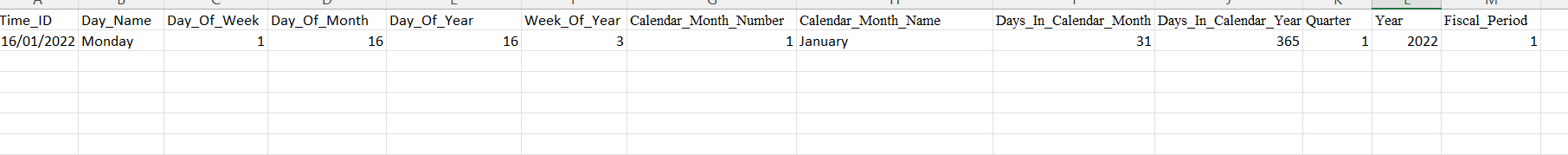


Times

Give details of various attributes of time.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Time\_ID | Each ID represents specific calendar date, PK | Date |
| Day\_Name | Name of the day for corresponding date | Text |
| Day\_Of\_Week | Day of the week corresponding to the date | Int |
| Day\_Of\_Month | Day of the month corresponding to the date | Int |
| Day\_Of\_Year | Day of the year corresponding to the date | Int |
| Week\_Of\_Year | Week of the year corresponding to the date | Int |
| Calendar\_Month\_Number | Calendar Month number corresponding to the date | Int |
| Calendar\_Month\_Name | Calendar Month name corresponding to the date | Text |
| Days\_In\_Calendar\_Month | Number of days in calendar month corresponding to the date | Int |
| Days\_In\_Calendar\_Year | Number of days in calendar year corresponding to the date | Int |
| Quarter | Quarter of the year corresponding to the date | Int |
| Year | Year corresponding to the date | Int |
| Fiscal\_Period | Fiscal period associated with the date | Int |

Example with filled data



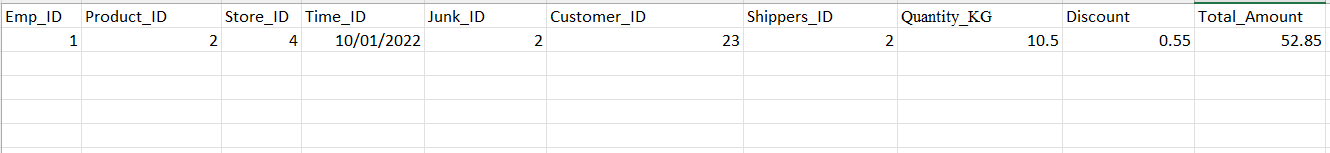
Fact\_Table

Employees

Give employee details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Emp\_ID | ID of the employee for the corresponding sale, FK | Bigint |
| Product\_ID | ID of the product sold, FK | Bigint |
| Store\_ID | ID of the store where sale occurred, FK | Bigint |
| Time\_ID | Time when sale occurred, FK | Bigint |
| Junk\_ID | ID of the junk attributes in corresponding sale, FK | Bigint |
| Customer\_ID | ID of the employee for the corresponding sale, FK | Bigint |
| Shippers\_ID | ID if shipper who will deliver product for corresponding sale, FK | Bigint |
| Quantity\_KG | Quantity of the product sold in Kg | Decimal(5,2) |
| Discount | Discount for the corresponding sale. | Decimal(5,2) |
| Amount | Total amount paid per each sale. | Decimal(10,2) |

Example with filled data

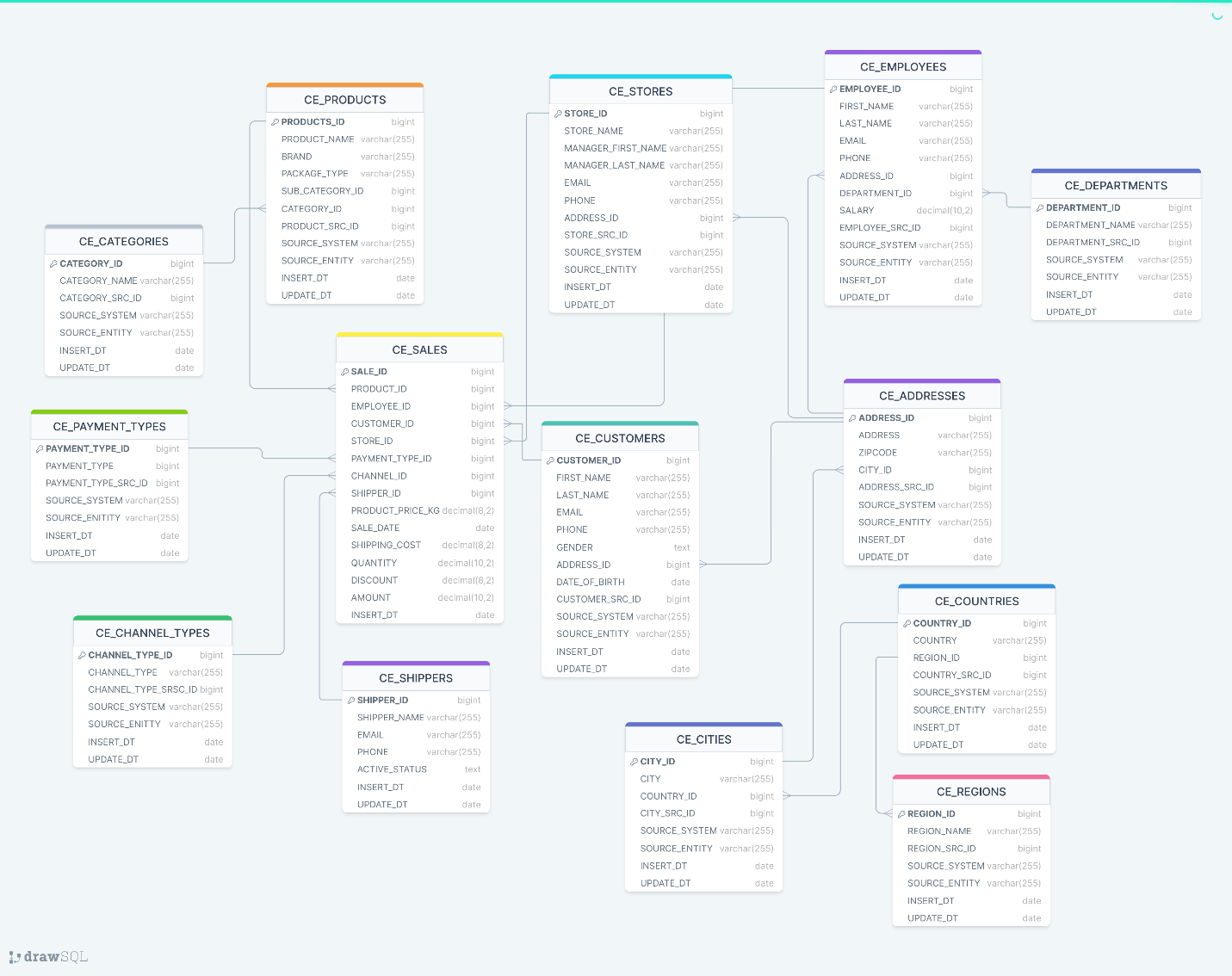


# Task 3: 3NF MODEL

Design Process:

1. Determine the data source that will be used to populate data ware house, and understand the structure and contents of the source data, including columns, datatype, etc.
2. Identify the specific requirements and goals of the data warehouse. Determine the key entities, relationships, and attributes that need to be captured in data warehousing.
3. Identify main entities, their attributes and relationships between them.
4. Create logical model using normalization techniques to achieve 3NF model.
5. Assign primary keys to each table to ensure uniqueness and assign source triplets as per the requirement. Assign source\_id which will be primary key from the source entity. Establish relationship between tables by defining foreign keys.

3NF model is uploaded in GitHub also.



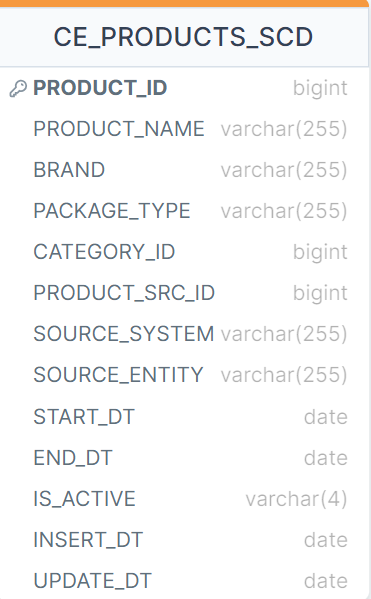
# Task 4: DM MODE

Updated SCD2 in 3NF model.

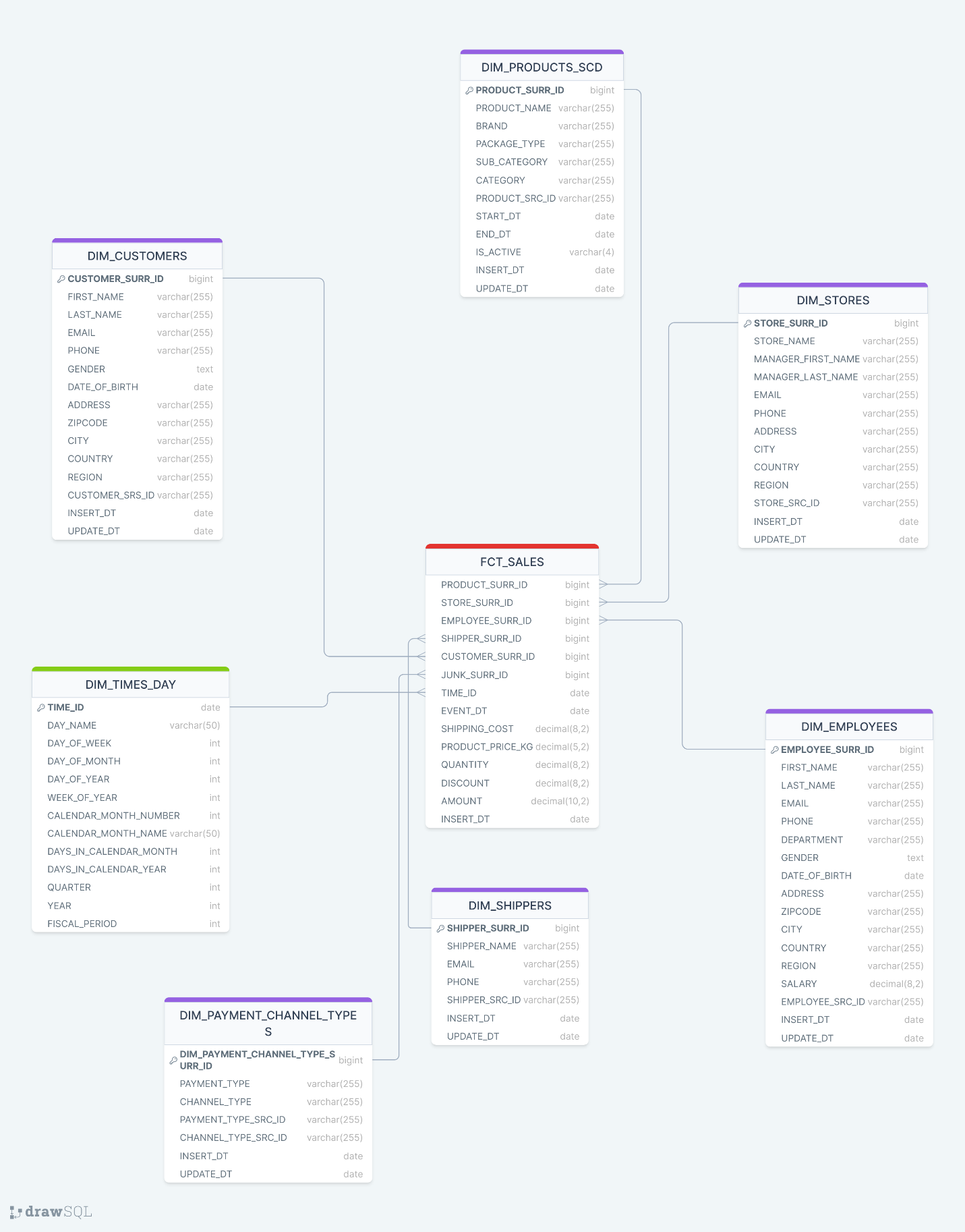
CE\_PRODUCTS

Added columns to CE\_PRODUCTS table:

* START\_DT: This column represents beginning of the validity period for that particular product. When a new version of that particular product is introduced, for example if package type of the product is changed a new record will create with the corresponding start date.
* END\_DT: This column represents the end of the validity period for that particular product. END\_DT for an active product will be a date in future for example (31-dec-9999). When that particular record became inactive END\_DT will change accordingly.
* IS\_ACTIVE: This column is used to represent whether a record is currently active or not active. This column generally indicated as an indicator (“Active” or “Not Active”) or a Boolean value. Due to presents of multiple records present in SCD2 dimensions we can use IS\_ACTIVE column to identify current or active version of that dimension.



1. BUSINESS LAYER DIMENSIONAL MODEL



Matrices:

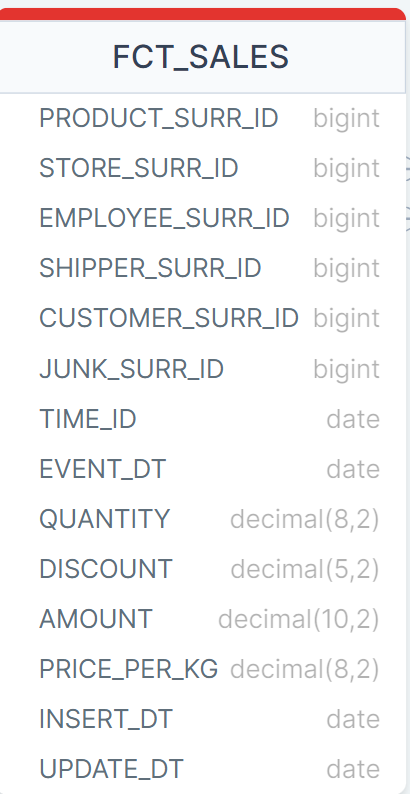
1. QUANTITY: Quantity of product per sale. It is measured in Kg.

2. DISCOUNT: Discount for each product

3. AMOUNT: Amount paid per customer per sale.

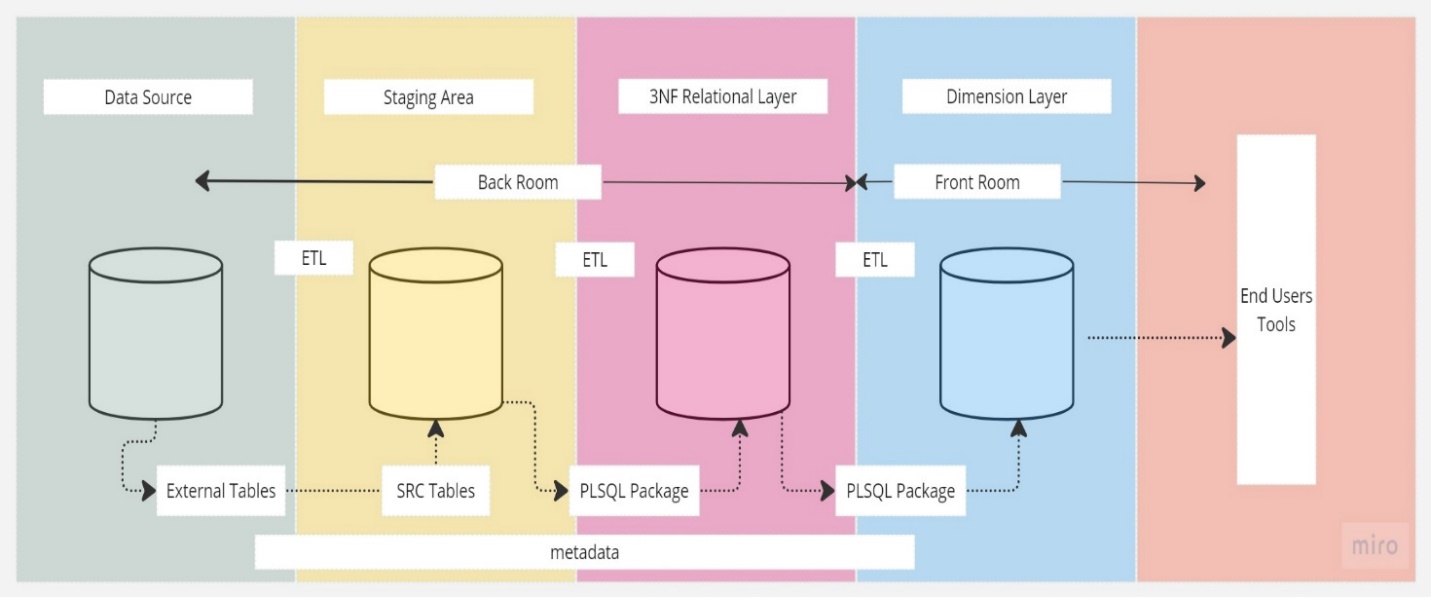
Calculated metric:

PRICE\_PER\_KG = AMOUNT / QUANTITY



# Task 5: LOGICAL SCHEMA AND DATA FLOW

* LOGICAL SCHEMA



Data Source

This layer consists of source data in the form of external tables. In this project I have two external tables which act as data source for the data ware house.

Staging Area

Data in external tables from data source layer will load to staging area. This layer act as source tables for the data warehouse. It acts as a transitional zone between source layer and data warehouse itself. This layer plays important role in data transformation and data integration.

3NF Relational Layer

This layer act as third normalization layer in data ware house architect. The goal of this layer is to reduce data redundancy and unsure data integrity. We use PLSQL package to load data from staging area to 3Nf layer.

Dimension Layer

This layer is responsible for maintaining the integrity and consistency of the dimension tables. This layer organizes and manages the descriptive attributes of the data, allowing for efficient and effective analysis and reporting. Mostly the tables in this layer will be de normalized which ensures fast and effective query. We access data from 3NF layer to dimension layer using PLSQL package.

Back Room

Back room is responsible for the process like data storage, integration, transformation, and preparation carry out in a data warehouse. It may include extract, cleanse, integration, transform and loading of data.

Front Room

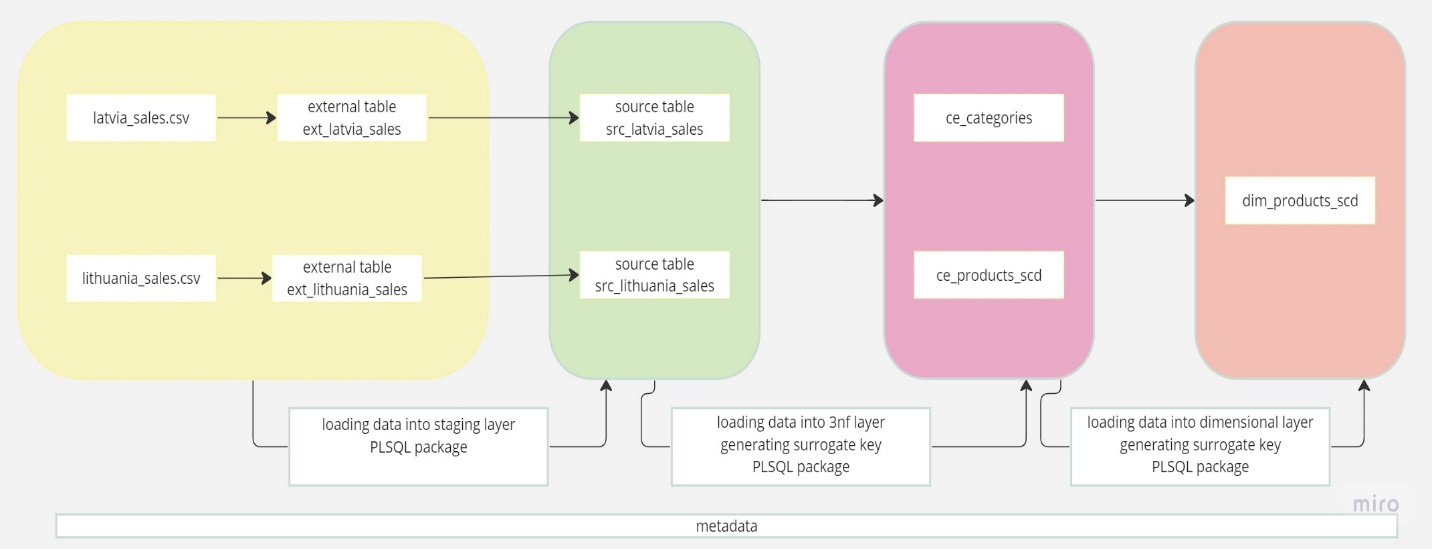
This is the user facing area of the data warehouse. This is the accessible part of data warehouse where end users like data analysists, managers, decision makers interact with the data and gain insights from it. In the front room users can query data warehouse, retrieve relevant data and perform various analytical operations.

Metadata

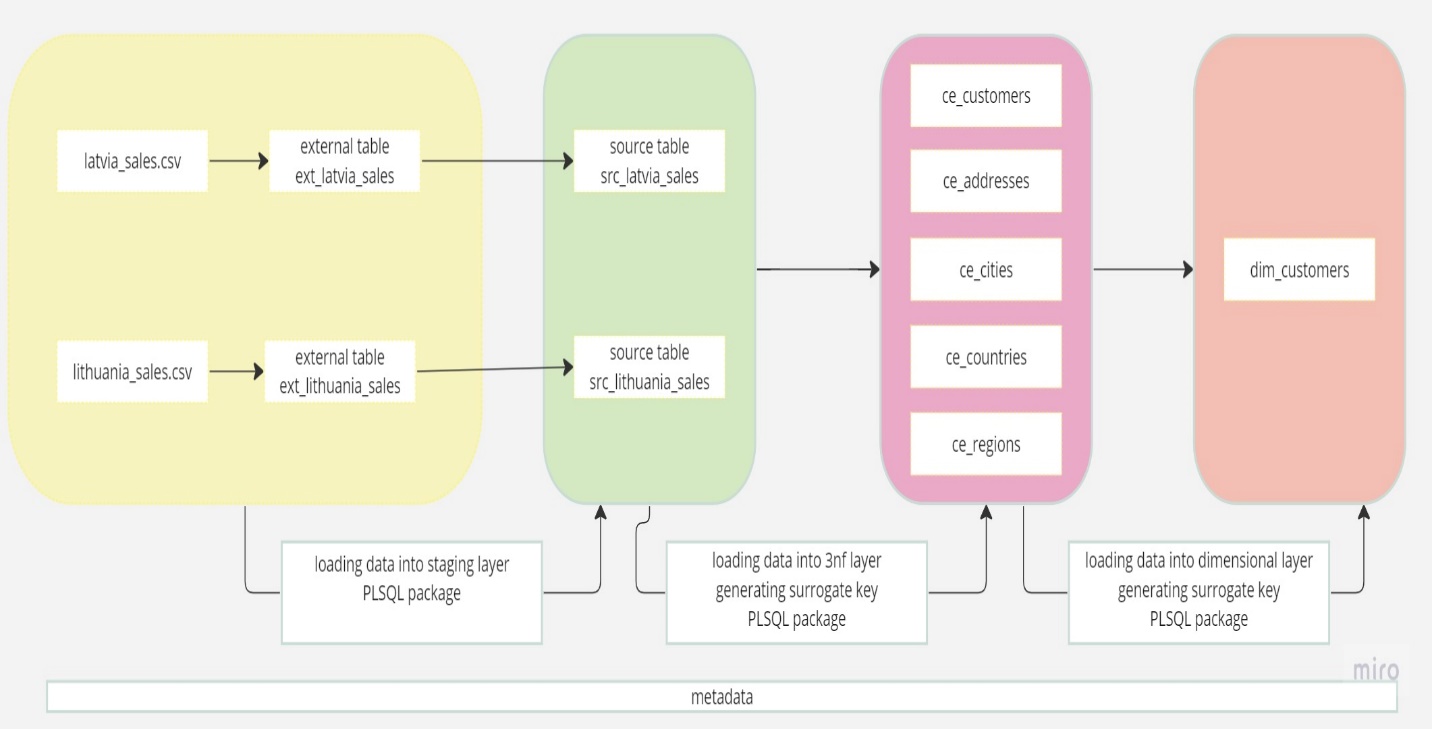
Ina data warehouse metadata refers to the information about the data stored within the data warehouse. It provides description and content to the data, allowing users and systems to understand and effectively utilizes the data assets in the data warehouse.

* DATA FLOW

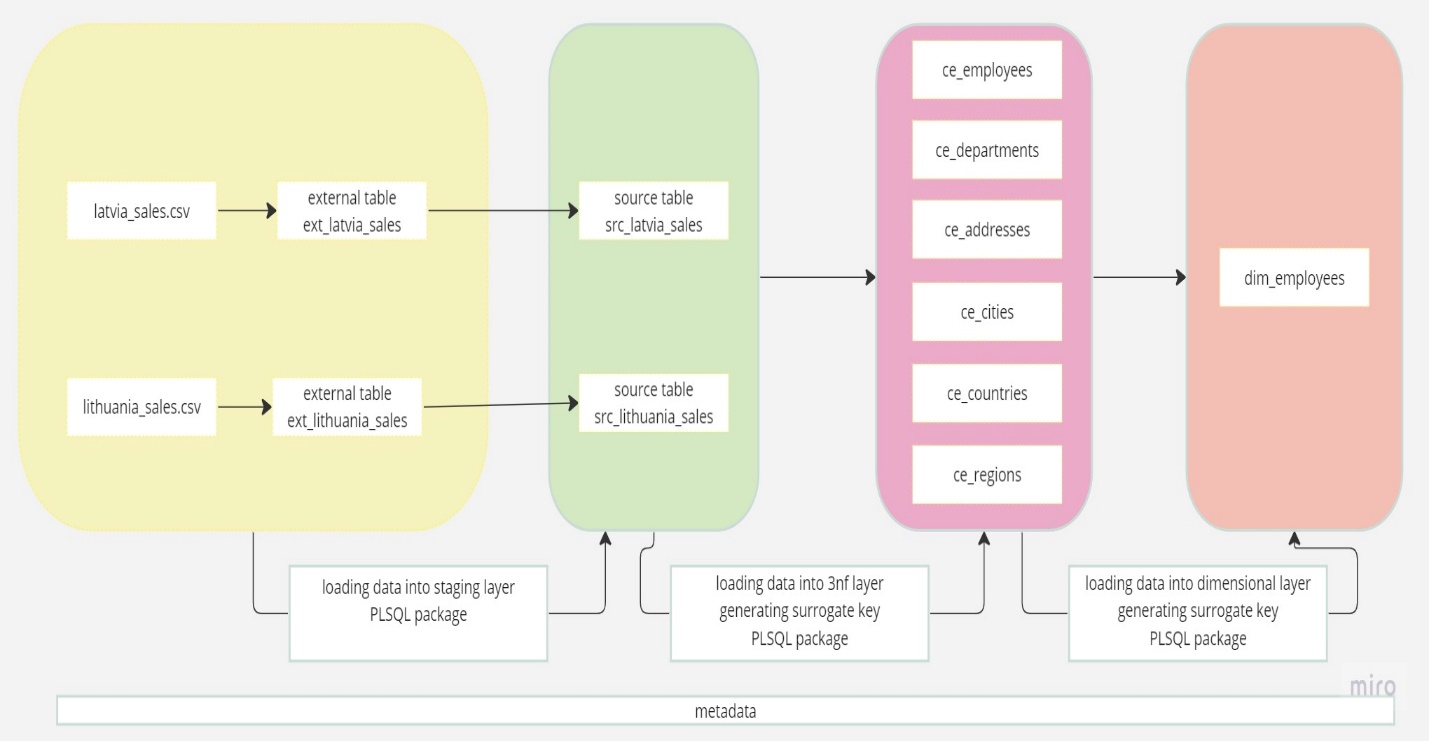
Data flow for dim\_products table



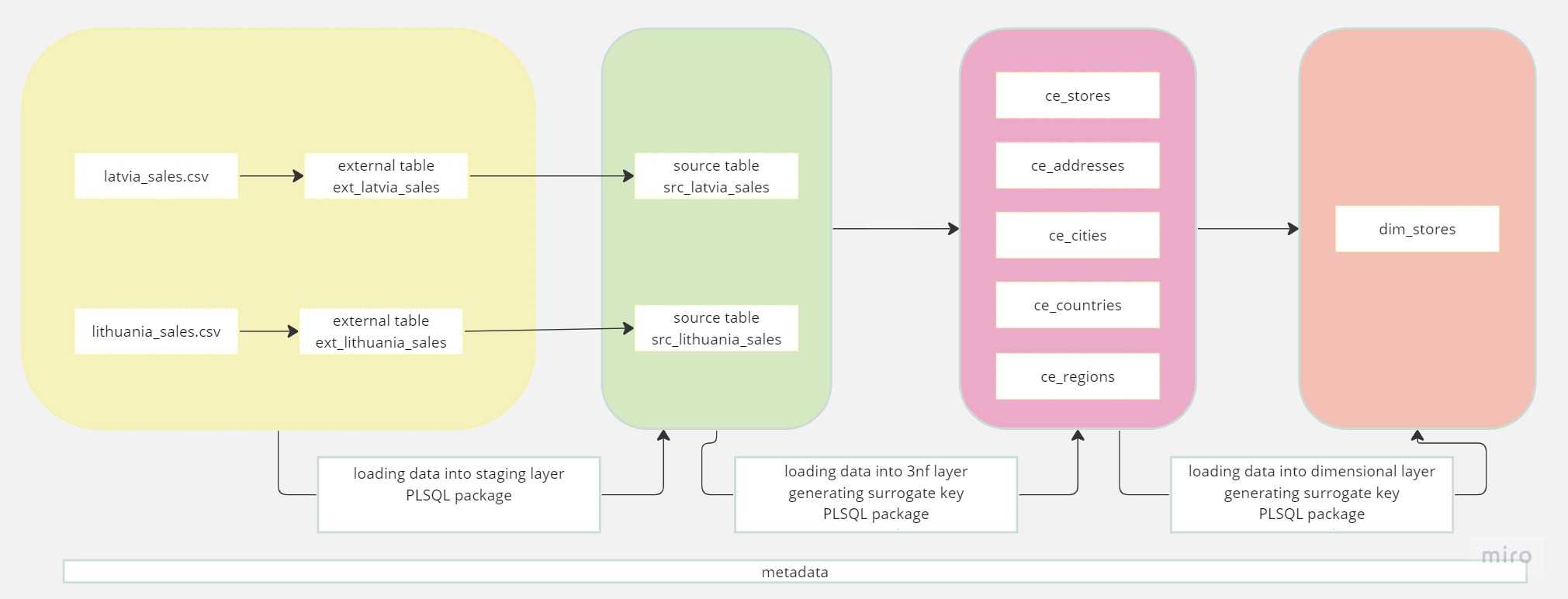
Data flow for dim\_customers table



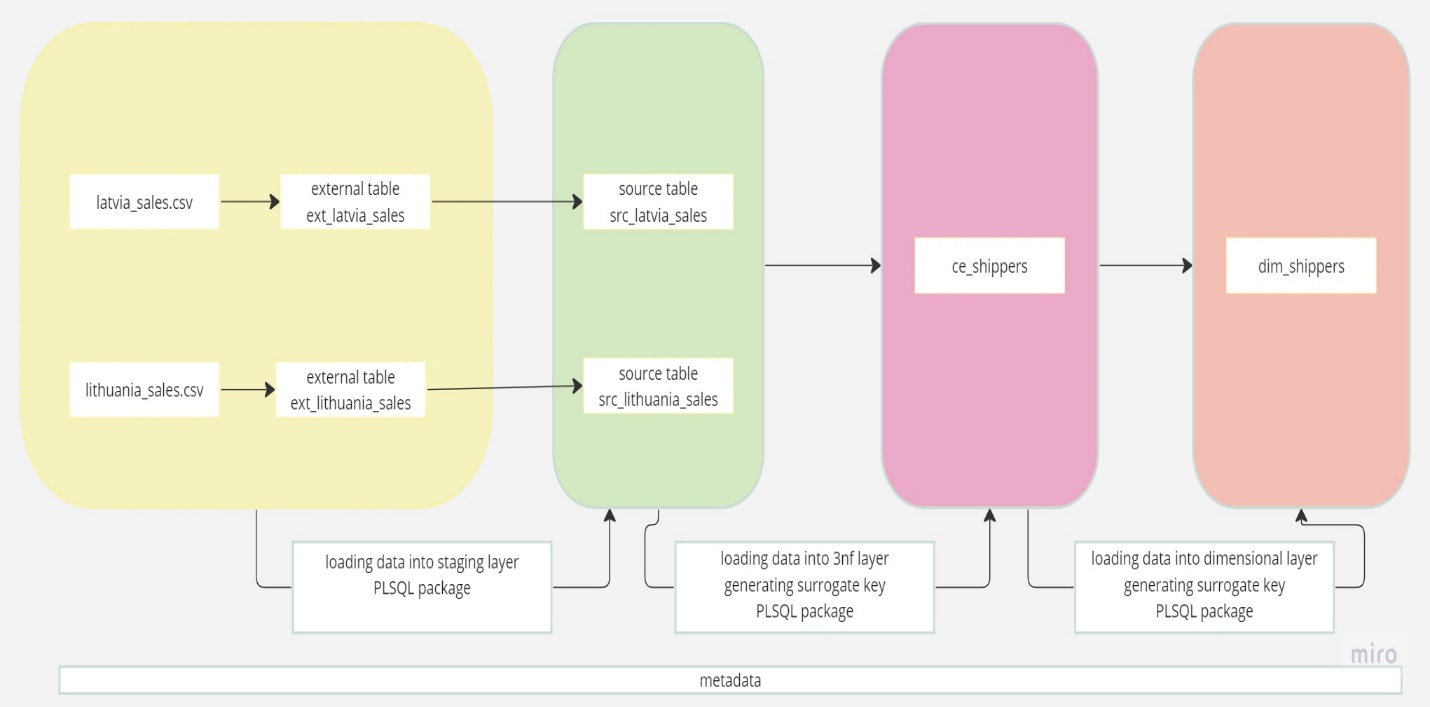
Data flow for dim\_employees table



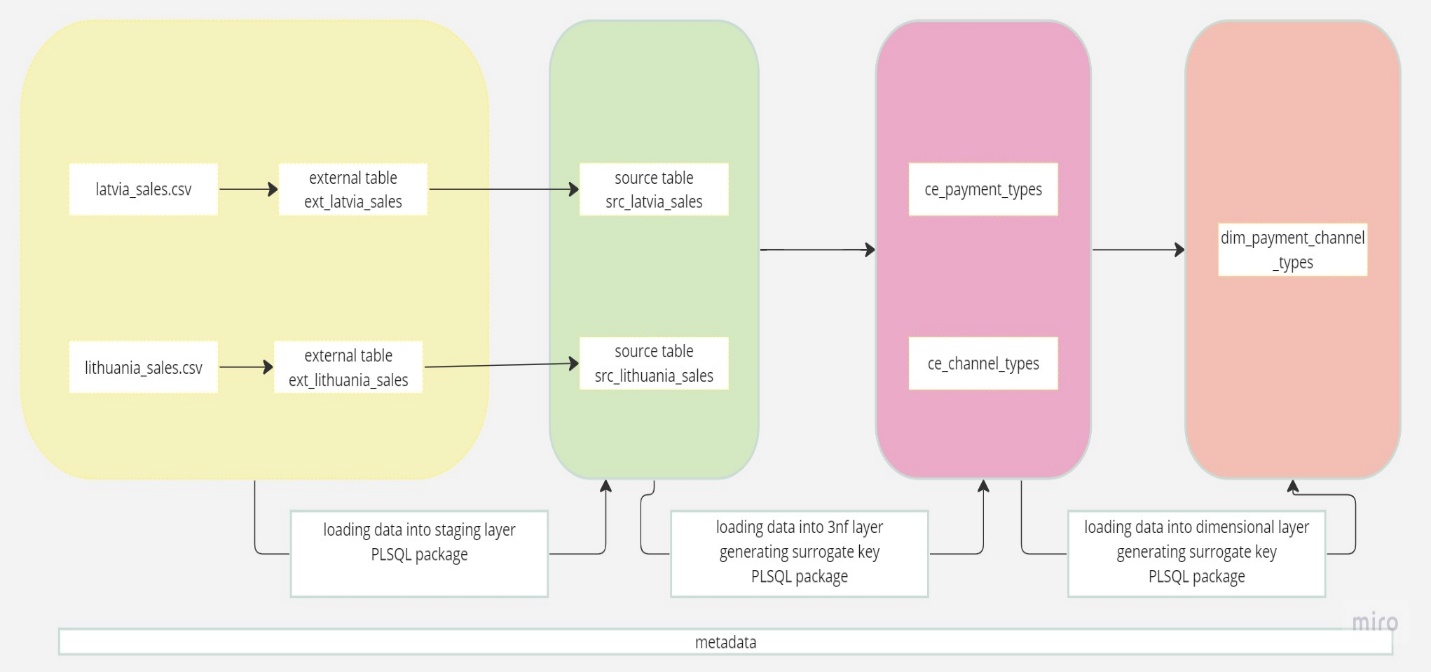
Data flow for dim\_stores



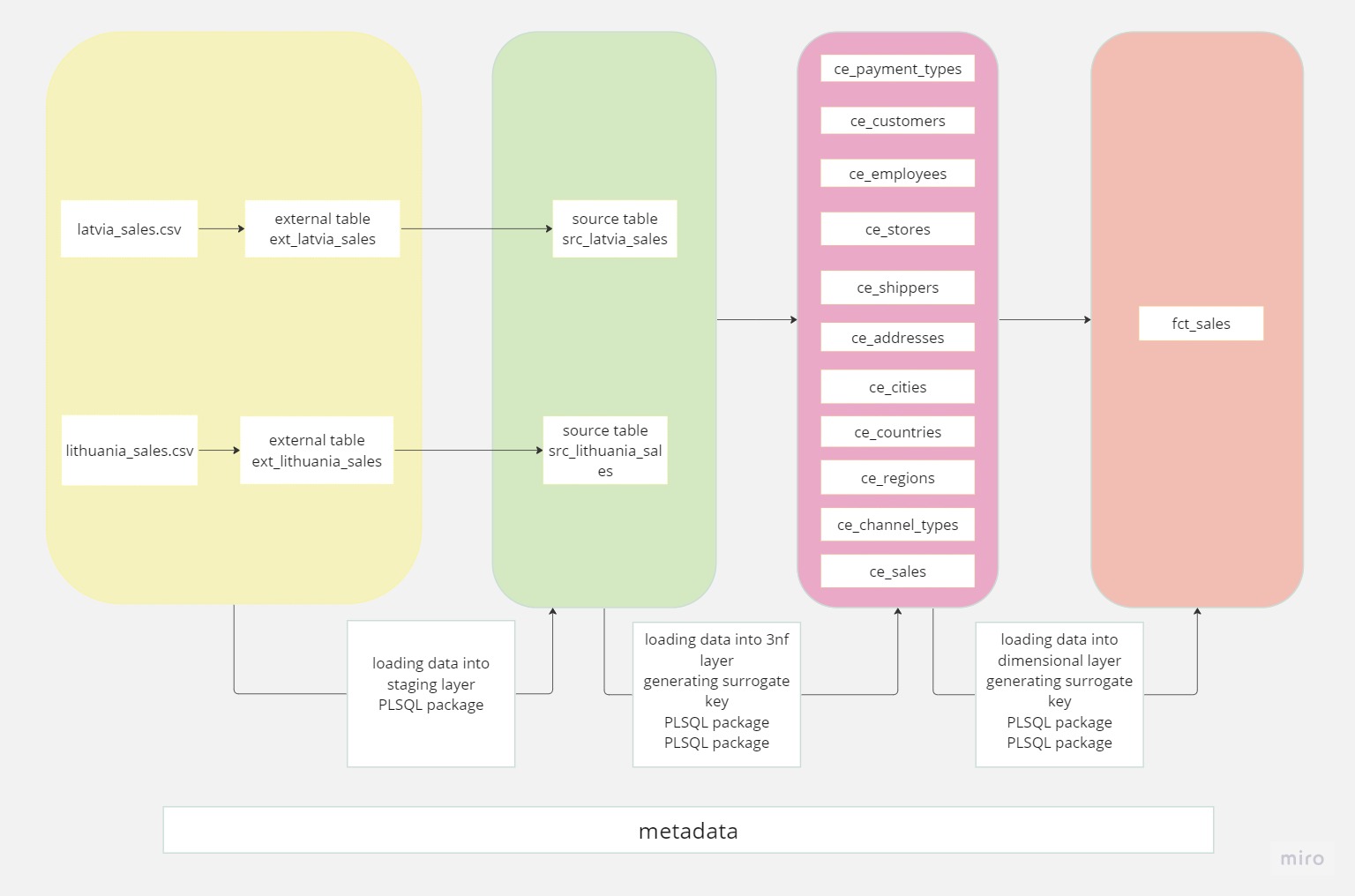
Data flow for dim\_shippers



Data Flow for dim\_payment\_channel\_types



Data flow for fct\_sales



# Task 5: data loading from sources to dimension layer

* In this project I used procedures to load data from one layer to different layers.
* First created two different procedures to load data from foreign tables to physical layer.

Then used meta table with help of view to incrementally load to sales table in 3nf later and one dimension table in 3nf layer. For rest of tables in 3nf layer used different procedure per each table and loaded directly from source tables.

* For each dimensions used separate procedures to load data from 3nf layer. Fact table is loaded using partitions and with the help of procedure.