Business Template

## WOMEN SHOE RETAIL SUPPLIER

## 1 Business Description

This is a description of a stores that sells women's shoes. They want to make sure they have the right shoes in stock and that their customers are happy. To do this, they're going to start using a computer system that helps them keep track of what shoes are selling the most and what colors are popular. They'll also use this system to figure out what prices work best and where to advertise. They have a lot of information about their shoes and they want to use it to make their customers happy!

## 1.1 Business background

Happy Feet is a women's footwear retailer operating in the competitive market of the United States. With headquarters located in major cities in USA that specialize in offering a diverse range of high-quality women's shoes. Company specializes in offering a wide array of high-quality women's shoes, with a focus on satisfying the demands of fashion enthusiasts, individuals seeking trendy footwear, and those prioritizing both style and comfort. As a leading player in the footwear industry, company have established a strong presence in the market over the past years. The company's commitment to delivering top-notch products at affordable prices has allowed it to establish a unique position in the fashion footwear sector.

## 1.2 Problems because of poor data management

Poor data management can indeed pose challenges for successful business operations, as it can lead to:

1. Insufficient information about customer preferences, which can hinder the ability to tailor products and services to meet their needs.

2. Inefficient stock management, resulting in overstocking or understocking of certain products, leading to lost sales or missed opportunities.

3. Inability to identify and respond to market trends, potentially causing the business to fall behind its competitors.

Recognizing the importance of effective data management in the women's shoe industry, we understand that poor data practices can be an obstacle for successful business operations. Insufficient information about customer preferences, stock management, and market trends can impede our ability to make informed decisions and stay competitive in the women's shoe market.

## 1.3 Benefits from implementing a Data Warehouse

To address these challenges, we are implementing a comprehensive data warehouse system. This strategic move will enable us to analyze crucial aspects of our business, answering essential questions such as:

1. Which shoe categories have the highest demand?
2. What colors are trending among our customers?
3. Which stores contribute the most to our revenue?
4. Who are our most valuable customers?

Furthermore, by processing and analyzing the data, we aim to gain insights into:

1. Seasonal sales trends
2. Popular materials and styles
3. Optimal pricing strategies
4. Effective marketing channels

## 1.4 DATASETS DESCRIPTION

Dataset 1: Sales Information

Product\_ID: This attribute gives the ID of which product sold.

Shoe\_Brand: This attribute gives the name of the shoe brand.

Category: This attribute specifies the category of the product (e.g., heels, sneakers, boots).

Manufacturer: The company or entity that manufactured the shoe.

Store\_Name: The name of the store where the sale occurred.

Store\_Address: The address of the store.

Colors: The colors available for the shoe product.

Sale\_Date: The date on which the sale occurred.

Quantity\_Sold: The quantity of the product sold.

Customer\_First\_Name: The first name of the customer who made the sale.

Customer\_Last\_Name: The last name of the customer who made the sale.

Customer\_ID: This attribute gives the ID of the customer who made the sale.

Total\_Amount: The total amount paid by the customer for the purchase.

Dataset 2: Sales Information with Discounts

Additional\_Attribute: Discount

All attributes from Dataset 1

Additional Business-Specific Information:

Brands

Seasonal Information

Material Details

Product Descriptions

Available Sizes

Price Information (Min, Max, Average)

# 1.5: BUSINESS TEMPLATE – Grain

Declare the Grain

The grain of shoe payment fact table would be **transactional**. This is because each row in the fact table represents a single transaction, specifically a sale transaction, with details about the sale such as the date of the sale, the quantity of shoes sold, the total amount paid, and any discount applied. This grain allows for detailed analysis of sales transactions, such as calculating total sales, average discount rates, and sales trends over time.

Identify the Dimensions

Dimensions are the key descriptive attributes of the fact table that provide context to the measures. In this step, we can identify the following dimensions:

Employees dimension: contains data about the employees who sell the stuff, their first and last name, email, and address, city, country, code, country name, region, economic region.

Customers dimension: provides information about the individual customers, their first and last name and address, city, country, code, country name, region, economic region.

Product dimension: includes details about the product as product description, material, size, colour and category.

Store dimension: includes details about store location, address, country, zipcode, phone number.

Identify the Facts

Facts are the numerical measurements that represent the business process.

the key fields in the fact table (payments) that uniquely identify a record are payment\_id, customer\_id, employee\_id, payment\_id, amount paid per customer and update\_dt. These fields collectively represent a specific payment transaction.

**Define the Level of Detail**

The level of detail is determined by the combination of key fields. In this case, the level of detail is defined by a unique combination of customer\_id, employee\_id, sale\_id, and the date (update\_dt). Each record at this level represents an individual payment made by a specific customer for a specific sale, facilitated by a particular employee, and occurring on a specific date.

**Justify the Selection**

The selection of this particular grain is justified by the need to capture detailed information about individual payment transactions. This level of granularity allows for a comprehensive analysis of sales performance, customer behavior, and employee effectiveness at the transactional level.

Dim Tables:

* EMPLOYEES
* Give EMPLOYEES table details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| EMPLOYEE\_ID | Represents ID of each employee, It’s the unique identifier of the table, PK | BIGINT |
| FIRST\_NAME | First name of employee | VARCHAR(255) |
| LAST\_NAME | Last name of employee | VARCHAR(255) |
| PHONE | Phone | BIGINT |
| ADDRESS | Address | VARCHAR(255) |
| CITY | City name | VARCHAR(255) |
| COUNTRY | Country | VARCHAR(255) |
| ZIPCODE | Country code | VARCHAR(255) |
| REGION | Region | VARCHAR(255) |
| ECONOMIC REGION | Economic Region | VARCHAR(255) |
| INSERT\_UP | Date | DATE |
| UPDATE\_ID | Date | DATE |

Example with filled data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE  \_ID | FIRST  \_NAME | LAST  \_NAME | ADDRESS | CITY |
| 1 | John | Smith | 123 Main St, Apt 45 | New York |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COUNTRY\_CODE | COUNTRY\_NAME | REGION | ECONOMIC\_REGION | UPDATE | INSERT |
| 34758 | USA | East Coast | North America | 2024-01-15 | 2024-01-15 |

Customers

Give CUSTOMERS table 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 | VARCHAR(255) |
| LAST\_NAME | Last name of customer | VARCHAR(255) |
| PHONE | Phone | VARCHAR(255) |
| ADDRESS | Address | VARCHAR(255) |
| CITY | City name | VARCHAR(255) |
| ZIPCODE | Country code | VARCHAR(255) |
| COUNTRY | Country name | VARCHAR(255) |
| REGION | Region of | VARCHAR(255) |
| ECONOMIC\_REGION | Economic Region | VARCHAR(255) |
| UPDATE\_ID | Date | DATE |
| INSERT\_DT | Date | DATE |

Example with filled data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CUSTOMER\_ID | FIRST\_NAME | LAST\_NAME | EMAIL\_CUST | ADDRESS | CITY |
| 101 | John | Johnson | mary.j@email.com | 123 Main St, Apt 45 | New York |

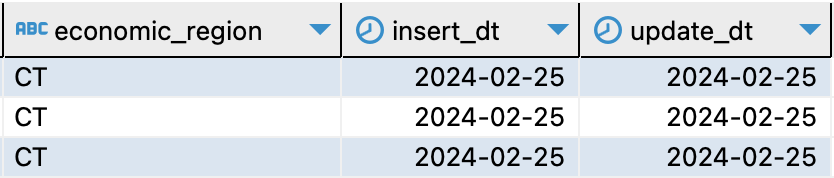
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COUNTRY\_CODE | COUNTRY\_NAME | REGION | ECONOMIC\_REGION | INSERT\_DT | UPDATE |
| 34758 | USA | East Coast | North America | 2024-01-15 | 2024-01-15 |

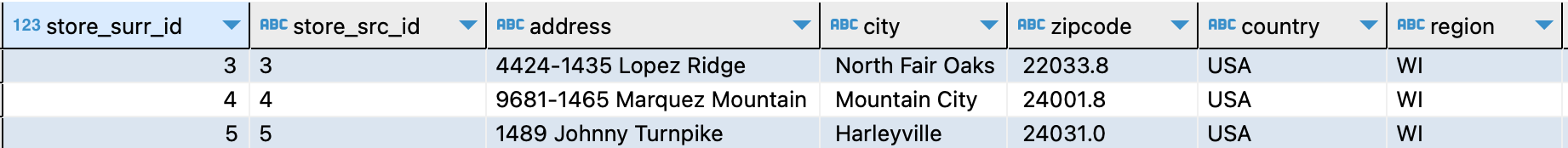
Stores

Give STORE table details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Store\_ID | Represents ID of each customer, It’s the unique identifier of the table, PK | BIGINT |
| PHONE | Phone | VARCHAR(255) |
| ADDRESS | Address | VARCHAR(255) |
| CITY | City name | VARCHAR(255) |
| ZIPCODE | Country code | VARCHAR(255) |
| COUNTRY | Country name | VARCHAR(255) |
| REGION | Region of | VARCHAR(255) |
| ECONOMIC\_REGION | Economic Region | VARCHAR(255) |
| UPDATE\_ID | Date | DATE |
| INSERT\_DT | Date | DATE |

Example: with filed data:



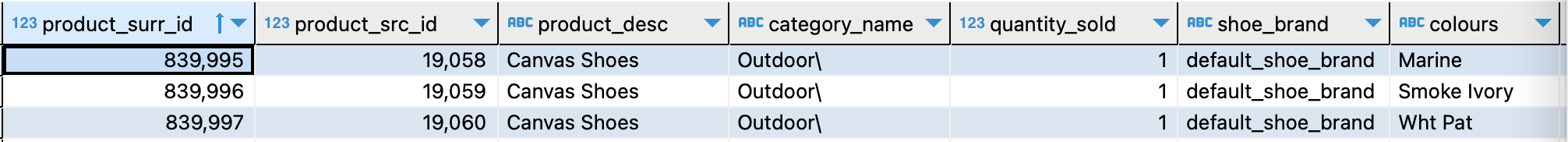


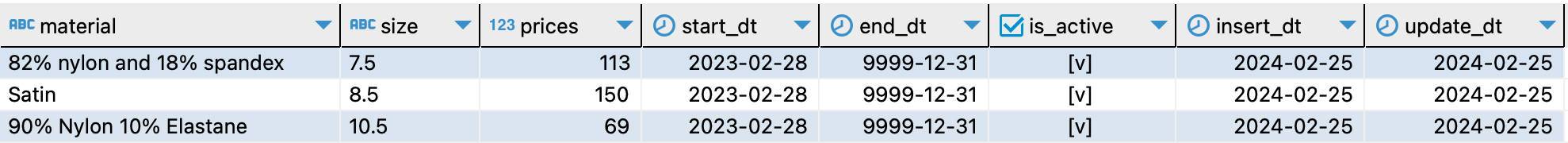
Products

Give PRODUCTS table details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| product\_ID | Represents ID of each sale, It’s the unique identifier of the table, PK | BIGINT |
| PRODUCT\_SRC\_ID | Product ID | VARCHAR(255) |
| PRODUCT\_DESC | Product description | VARCHAR(255) |
| CATEGORY\_NAME | Category Name | VARCHAR(255) |
| QUATITY\_SOLD | Quantity units sold | INTEGER |
| SHOE BRAND | Shoe brand | VARCHAR(255) |
| COLOURS | Colour of the shoes | VARCHAR(255) |
| MATERIAL | Material of the shoes | VARCHAR(255) |
| SIZE | Size of the shoes | INTEGER |
| PRICES | Price of the shoes | INETEGR |
| START\_DT | DATE | DATE |
| END\_DT | DATE | DATE |
| IS\_ACTIVE | TRUE or FALSE | BOOLEAN |
| INSERT\_DT | DATE | DATE |
| UPDATE\_ID | Update date | DATE |

Example with filled data



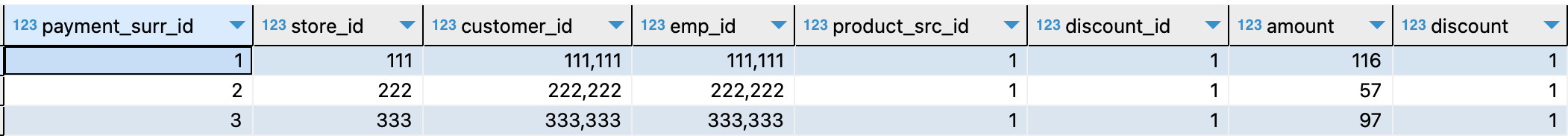


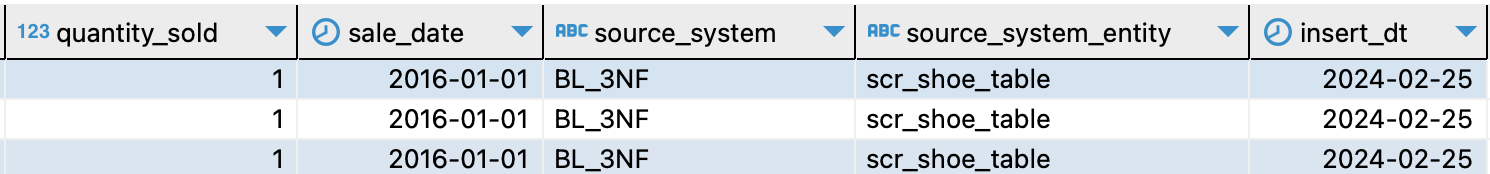
Payments

Give PAYMENTS table details

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| PAYMENT\_ID | Represents ID of each payment, It’s the unique identifier of the table, PK | BIGINT |
| CUSTOMER\_SURR\_ID | Customer ID of each customer, FK | BIGINT |
| EMPLOYEE\_SURR\_ID | Employee ID, FK | BIGINT |
| PRODUCT\_SURR\_ID | Represents ID of each product, FK | BIGINT |
| STORE\_SURR\_ID | Represents ID of each store, FK | BIGINT |
| SALE\_DATE | Represents date of sale | DATE |
| QUANTITY\_SOLD | Quantity sold in units | INETGER |
| DISCOUNT | Price discount | INETEGER |
| AMOUNT | Amount of payment | NUMERIC |
| TIME\_ID | DATE | DATE |
| UPDATE\_ID | Update date | DATE |

Example with filled data



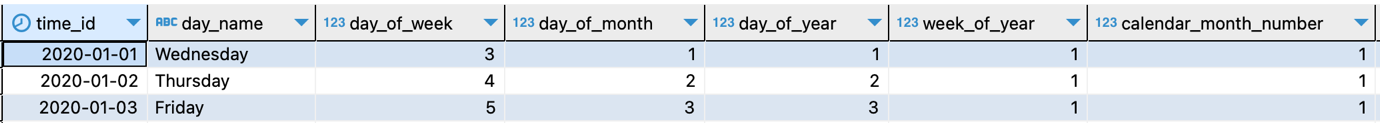


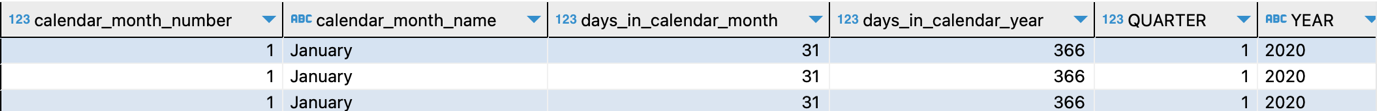
Times

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





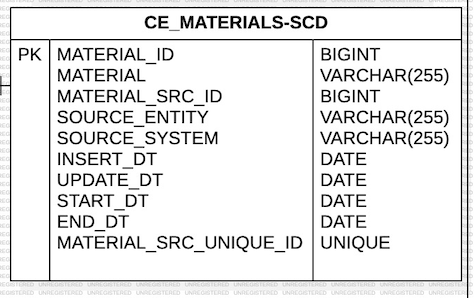
# Task 3: 3NF MODEL

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 the relationship between tables by defining foreign keys.

Added columns to CE\_MATERIALS table:

* START\_DT: This column represents the beginning of the validity period for that particular product. When a new version of that particular product is introduced, for example, if the product is changed, the material, a new record will be created 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 the future for example (31-dec-9999). When that particular record becomes 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 is generally indicated as an indicator (“Active” or “Not Active”) or a Boolean value. Due to the presence of multiple records in SCD2 dimensions, we can use the IS\_ACTIVE column to identify the current or active version of that dimension.



TASK 4. Metrics

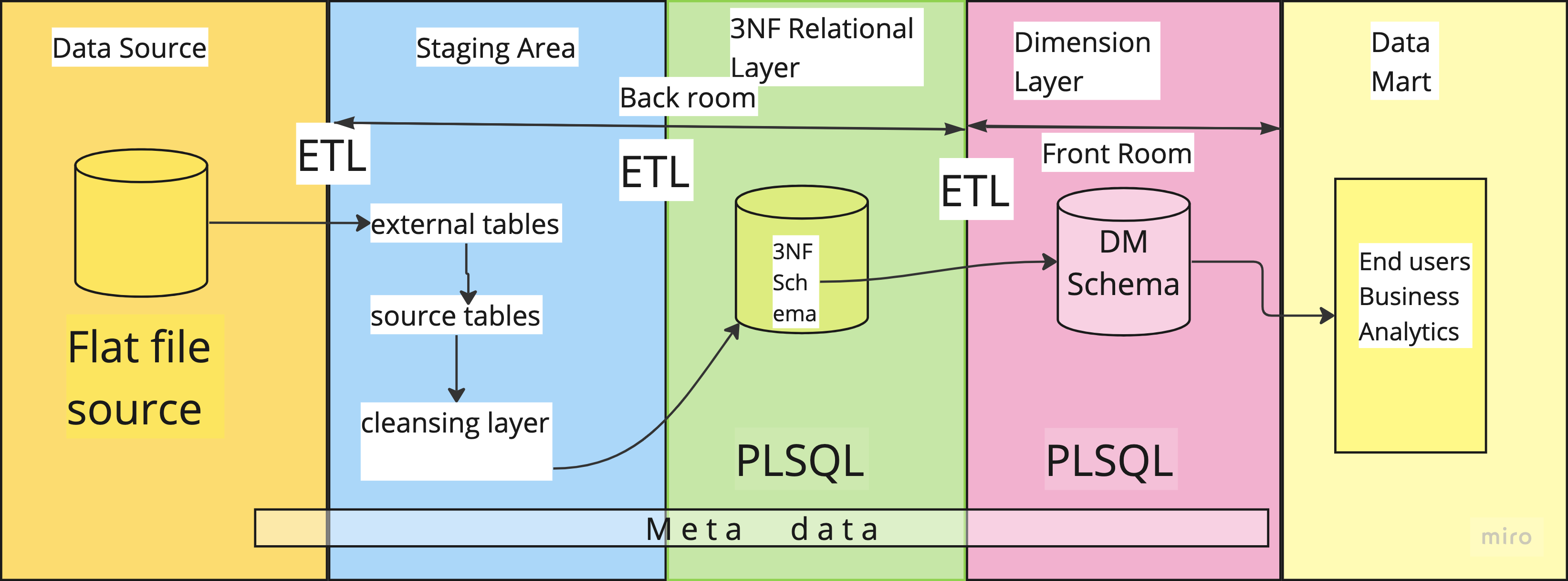
Metrics that are applicable to a shoe business:

**Quantity**: This metric would represent the number of shoe pairs sold per sale. In a shoe business, this could be measured in pairs of shoes, boxes, or any other unit of measurement that makes sense for the shoe business. The quantity metric identifies which styles, materials, brands of shoes are selling well.

**Discount**: This metric would represent the discount applied to each shoe sold. In a shoe business, this could be a flat discount applied to all shoes, or it could vary depending on factors like the brand, style, or size of the shoes. The discount metric evaluates the effectiveness of sales promotions.

**Amount**: This metric would represent the total amount paid per customer per sale. In a shoe business, this would be the total price of the shoes sold to the customer, minus any discounts applied. The amount metric can help to track overall sales revenue and profitability.

# Task 5: LOGICAL SCHEMA AND DATA FLOW



Data Source

This is where the raw data originates. In general, it could be transactional databases, operational systems, external data feeds, or any other source of data. In this project I have 2 external tables which act as data sources for the data warehouse.

Staging Area

The staging area serves as an intermediate storage area where the raw data from the source systems is copied or extracted. In the staging area, data is typically transformed, cleansed, and standardized before being loaded into the data warehouse. It acts as a transitional zone between the source layer and the data warehouse itself. This ensures that the data is consistent and ready for further processing. It is an important stage for data transformation and data integration.

3NF Relational Layer

The third normal form (3NF) relational layer represents the normalized data model within the data warehouse. In this layer, data is organized into multiple tables, each representing a specific entity or fact. Relationships between tables are established through primary keys and foreign keys. This layer helps maintain data integrity and reduces redundancy.

The goal of this layer is to reduce data redundancy and ensure data integrity. We use the PLSQL package to load data from the staging area to the 3Nf layer.

Dimension Layer

The dimension layer contains dimension tables that provide context and descriptive attributes for the data in the fact tables. Dimension tables typically include attributes such as customer details, product information, time dimensions, and other reference data.

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 a fast and effective query. We access data from 3NF layer to the dimension layer using PLSQL package or through foreign key relationships.

Back Room

Back room is responsible for the processes like data storage, integration, transformation, and preparation carry out in a data warehouse. It may include extract, cleanse, integration, transform and loading of data. The back room includes components such as ETL tools, data integration platforms, data warehouses, data lakes, and other storage and processing infrastructure. Backend operations involve activities like extracting data from various source systems, cleansing and standardizing the data to ensure consistency, integrating data from multiple sources, transforming data into a format suitable for analytics, and loading it into the data warehouse.

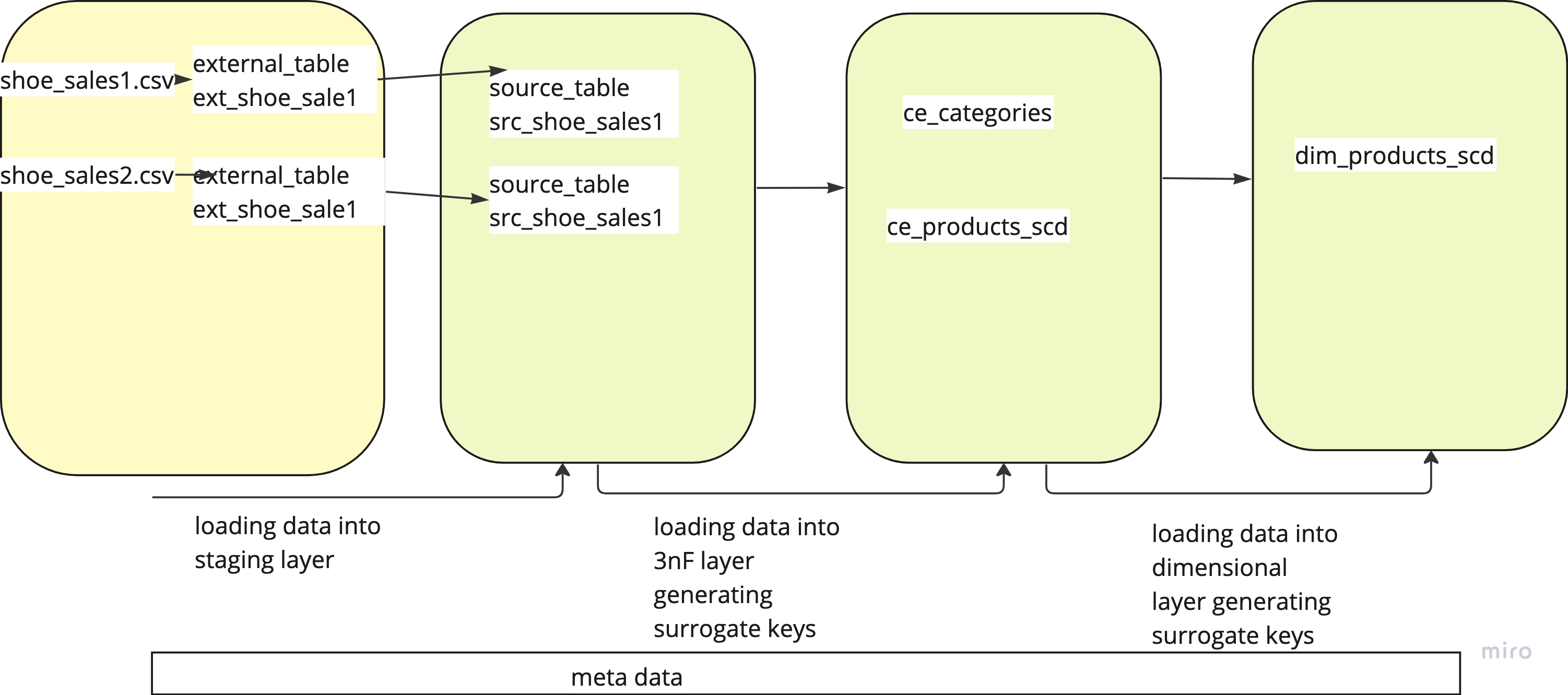
Front Room

This is the user-facing area of the data warehouse. This is the accessible part of the data warehouse where end users like data analysts, managers, and decision-makers interact with the data and gain insights from it. In the front room users can query the data warehouse, retrieve relevant data, and perform various analytical operations. The front room includes components such as BI tools, reporting tools, data visualization platforms, dashboarding software, and other user interface elements that facilitate data exploration and analysis

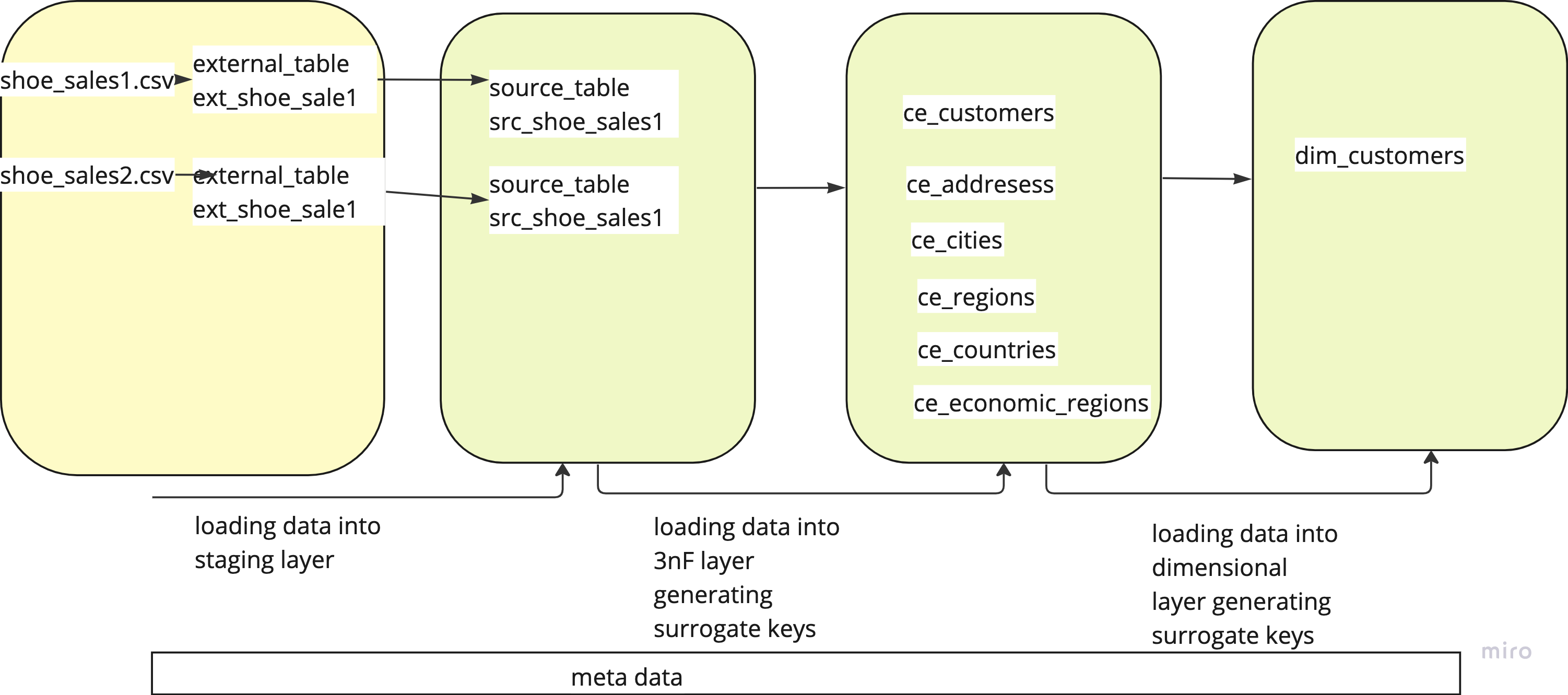
Metadata

In a 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 utilize the data assets in the data warehouse.

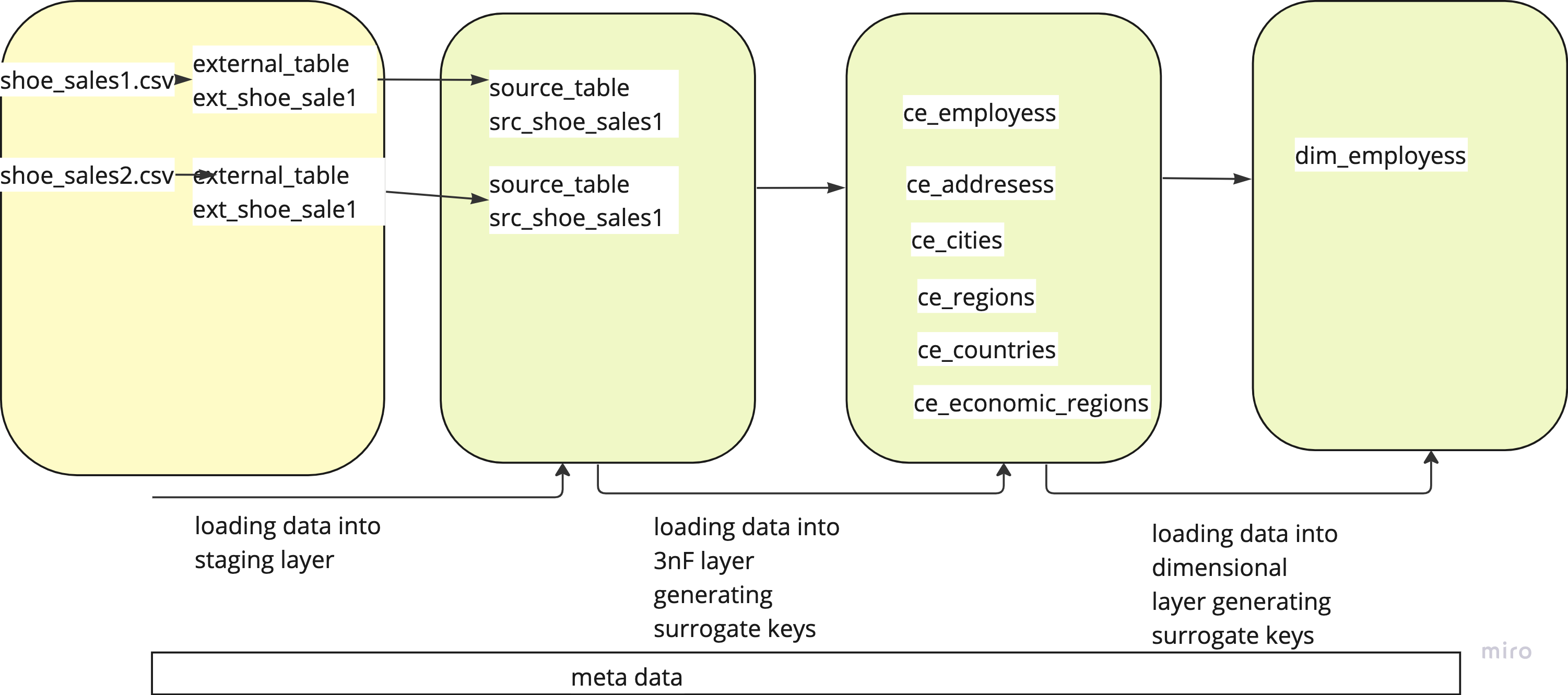
Data flow for dim\_products table



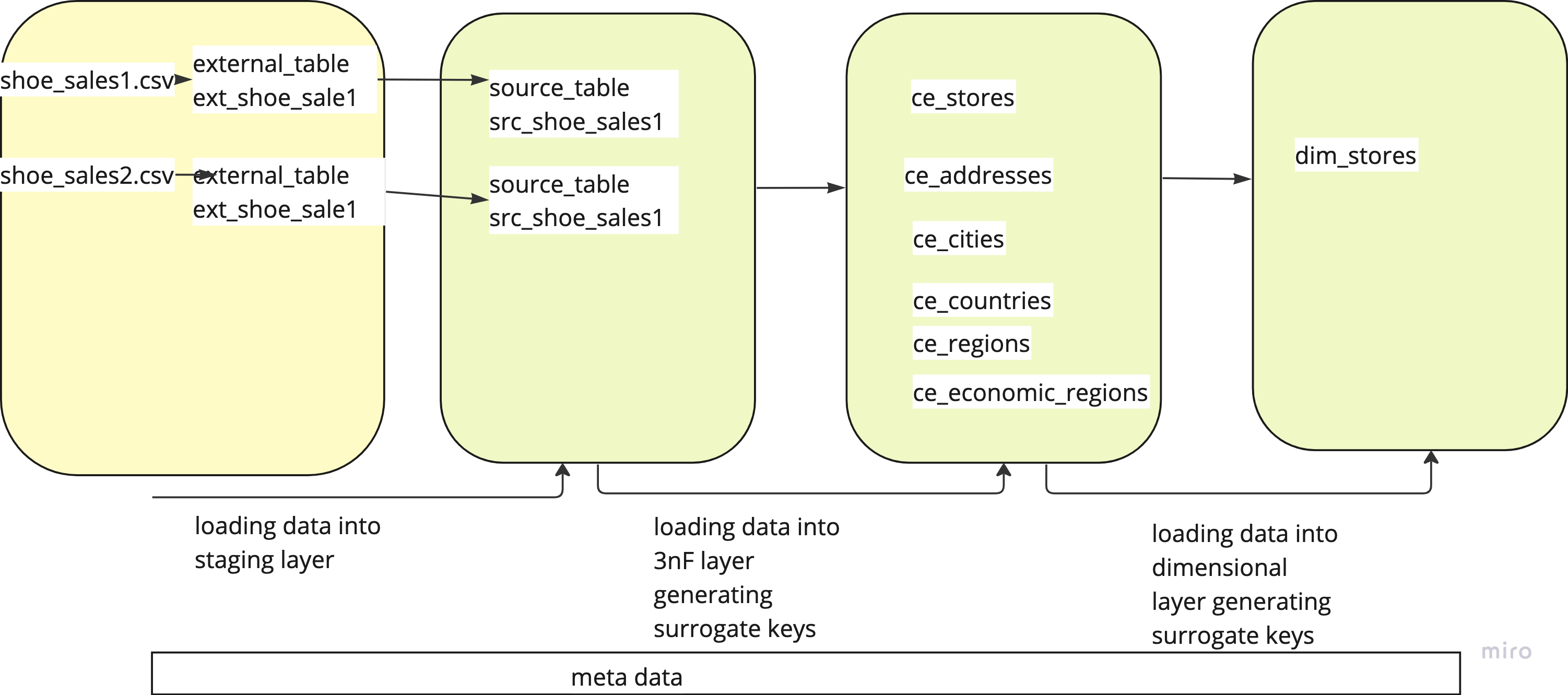
Data flow for dim\_customers table



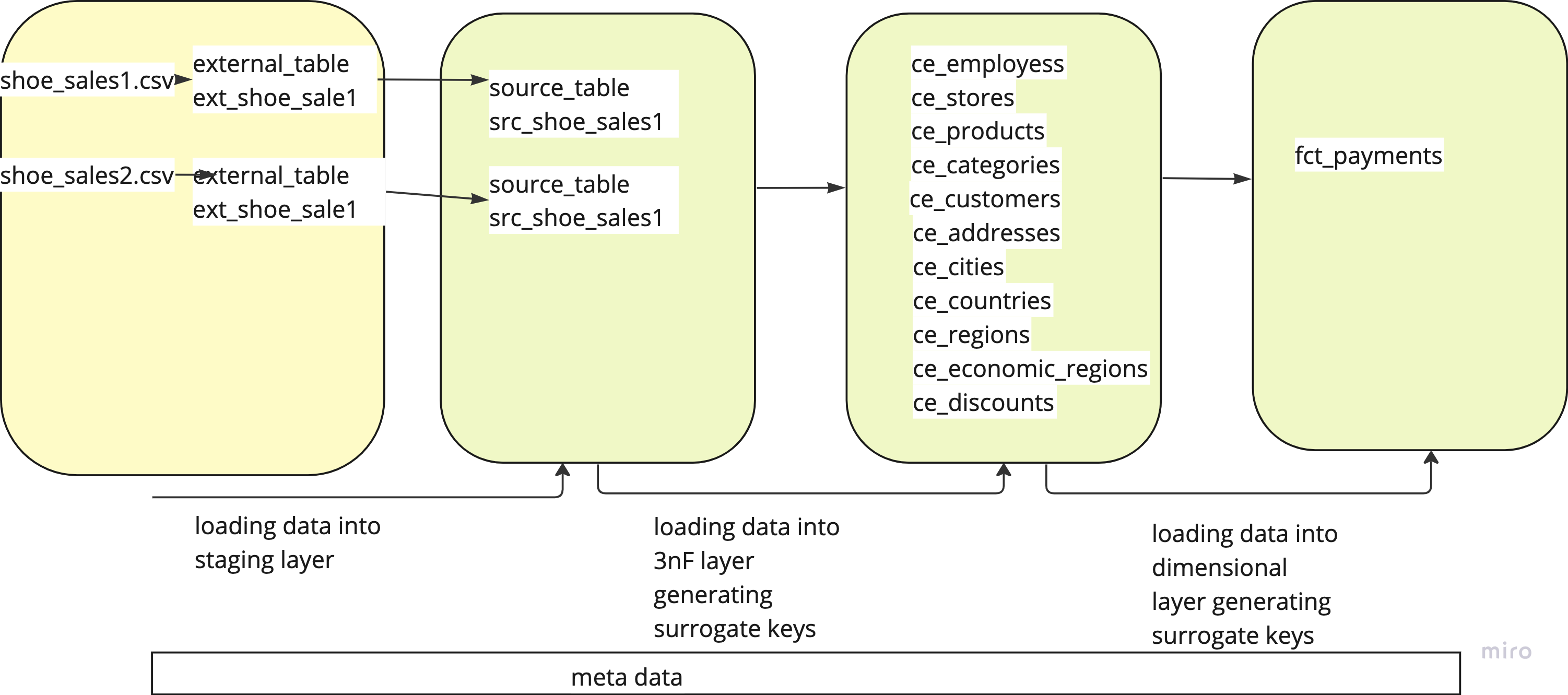
Data flow for dim\_employess table



Data flow for dim\_stores



Data flow for fct\_payments



In this project, I utilized procedures to load data from one external layer to different layers. The first procedure was used to load data from foreign tables to the physical layer. Subsequently, to load data into the 3NF (Third Normal Form) layer, I used different procedures for each table, loading the data directly from the source table. For each dimension, I created separate procedures to load data from the 3NF layer. Finally, the fact table was loaded using the data from the 3NF layer.

**TASK6. PARTITIONING STRATEGY**

Given that your material table is a child table to the product table and will be updated periodically (once in 3-6 months), a suitable partitioning strategy would be to partition the material table based on the update frequency or a relevant attribute that changes with each update. Since the updates are infrequent and occur over a relatively short period, partitioning by a date range that aligns with these update cycles could be effective.

1. **Partition by Date Range**: Since the material table is updated periodically, partitioning can be done by a date range that aligns with these update cycles. For example, if updates typically occur every 3-6 months, partition can be done to update the table by these intervals. This approach allows the database to quickly access the most recently updated materials without scanning the entire table.
2. **Using a Partition Key**: The column that changes with each update cycle as the partition key. The specific material column that is updated or changed with each update, such as an update\_dt column, would be an ideal candidate for the partition key.

Top of Form

Bottom of Form

Top of Form

Bottom of Form