Business Template

KIDS TOYS

CONTENTS

1 BUSINESS DESCRIPTION

1.1 BUSINESS BACKGROUND

Toys are what every family with children has at home. Toys are what they buy for the holidays for their nephews and children of friends. Over time, not only entertaining toys appear, but also games for the mind and development. The opportunity to captivate a child and at the same time allow his brain to develop is an important aspect when choosing toys lately.

More toys were bought globally in 2021 than ever before, with the global market growing 8.5% to $104.2 billion. This business attracts many companies, and in order to enter this market and control your presence in it, you need to have information about your business: the amount of costs and sales, the situation by region and loyal customers.

1.2 PROBLEMS BECAUSE OF POOR DATA MANAGEMENT

Lack of information about certain aspects of the business or incorrect information can affect the long-term strategy of the business. If the owner does not know how to collect and visualize information, this has a negative result. The situation by region must be accurate and reliable in order to adjust development strategies. Also, information about personal selling is important for motivating employees.

1.3 BENEFITS FROM IMPLEMENTING A DATA WAREHOUSE

Based on the results of the implementation of DWH, the business owner will receive answers to the following questions:

- What categories of toys sell best and worst?
- At what time of the year sales fall or increase, will be able to analyze the difference in sales in different periods?
- Which regions are doing the best/worst sales?
- How do its employee’s work? and possible other conclusions.

1.4 DATASETS DESCRIPTION

There are two datasets. Each of them contains the following information.
Product Information:
Name: The name of toy.
Category: The category of the toy (art, games, electronic, et al)
Manufacturer: The name of manufacturer (Lego, Hasbro, Disney, et al)
Age limit: The minimum recommended age for using the toy.
Cost: Toy production costs
Sales Information:
Date: The date of toy sale.
Quantity Sold: The number of units sold.
Sales Price: The actual selling price of the toy.

Payment method: Card/cash

Store Information:

Name: The name of store.

Address: The address of toy’s store.

Customer Information:

Gender: The gender of customer (F/M).

Age: The age range information of the customer.

Employee Information:

Name: The name of employee.

Surname: The surname of employee.

Gender: The gender of employee.

Additional Attributes:

Discount: The percentage of discount applied to the sales price.

Material: The material used in the manufacture of toys.

Datasets differ:

Method of payment for toys: in the first – cash, in the second – a card.

Customer gender: first – man, second – female.

Datasets provide complete information about the sale of children's toys, which allows you to analyze sales by region, category, material, and other indicators.

Grain description:

Dataset should to provide information in each individual row about what product, from what source, at what time, in what quantity, at what price, by what payment method to which customer, which employee in which store sold.

Dimensions description.

Dim Products contains information about product name, category, age limit for toy, material from which the toy is made and manufacturer.

Column name	Description	Data Type
PRODUCT_NAME	Product’s name	VARCHAR
PRODUCT_CATEGORY_NAME	Product’s category	VARCHAR
AGE_LIMIT	Age limit for toys	VARCHAR
MATERIAL	Toy’s main material	VARCHAR
MANUFACTURER	Toy’s manufacturer name	VARCHAR

Example with filled data

PRODUCT_ NAME	PRODUCT_ CATEGORY_NAME	AGE_LIMIT	MATERIAL	MANUFACTURER
Magic sand	Art & Crafts	3+	Textile	Mattel

Dim Employees contains information about name, surname and gender of employees.

Column name	Description	Data Type
EMP_FIRST_NAME	Employee's name	VARCHAR
EMP_SURNAME	Employee's surname	VARCHAR
EMP_GENDER	Gender: M, F	BOOLEAN

Example with filled data

EMP_FIRST_NAME	EMP_SURNAME	EMP_GENDER
Helen	Adaro	F

Dim Customers contains information about customer gender and birthdate.

Column name	Description	Data Type
CUSTOMER_GENDER	Gender: M, F	BOOLEAN
CUSTOMER_CARD	Number of card	VARCHAR
CUSTOMER_BIRTHDATE	Customer age	DATE

Example with filled data

CUSTOMER_GENDER	CUSTOMER_CARD	CUSTOMER_BIRTHDATE
M	AS15248524AD	1991-05-06

Dim Stores contains information about store name and address (including country, city, street with building).

Column name	Description	Data Type
STORE_NAME	Store's name	VARCHAR
STORE_ADDRESS	Full store address	VARCHAR

Example with filled data

STORE_NAME	STORE_ADDRESS
Barbie-Hanoi	Vietnam; Hanoi; 1630 Hicks st

Dim Time contains information about time of sales.

Column name	Description	Data Type
DATA_VALUE	Whole date	DATE
YEAR_DESC	Number of years	INT
QUARTER_DESC	Number of quarters	INT
MONTH_DESC	Number of months	INT
MONTH_NAME	Name of months	VARCHAR
WEEK_DESC	Number of weeks	INT
DAY_DESC	Number of days	INT
DAY_NAME	Name of days	VARCHAR
DAY_NUMBER_OF_WEEK	Number of weeks	INT

Example with filled data

DATA_ VALUE	YEAR_ DESC	QUARTER_ DESC	MONTH_ DESC	MONTH_ NAME	WEEK_ DESC	DAY_ DESC	DAY_ NAME	NOW
2023-01-01	2023	1	1	2021	1	0	Sunday	1

Dim Payments contains information about a possible method of payment for toys.

Column name	Description	Data Type
PAYMENT_NAME	Cash / Card	VARCHAR

Example with filled data

PAYMENT_NAME
Cash

Dim Channels contains information about the different sources where customers come from.

Column name	Description	Data Type
CHANNEL _NAME	Source’s name	VARCHAR
CHANNEL _TYPE	Source’s type	VARCHAR

Example with filled data

CHANNEL _NAME	CHANNEL _TYPE
Instagram	Ad

Dim Suppliers contains information about the suppliers.

Column name	Description	Data Type
SUPPLIER _NAME	Supplier’s name	VARCHAR
SUPPLIER_PHONE	Supplier’s phone	VARCHAR
SUPPLIER_MAIL	Supplier’s mail	VARCHAR

Example with filled data

SUPPLIER _NAME	SUPPLIER_PHONE	SUPPLIER_MAIL
Funzone Toys	+777 888 999	funzone_toys2@ex.com

2 BUSINESS LAYER 3NF

At this stage, changes are presented to the above dimension tables in the 3NF. Columns from Source Triplet (SOURCE_SYSTEM, SOURCE_ENTITY, SOURSE_ID) have been added to each needed dimension.

Source system (SA) – prepared sets of data.

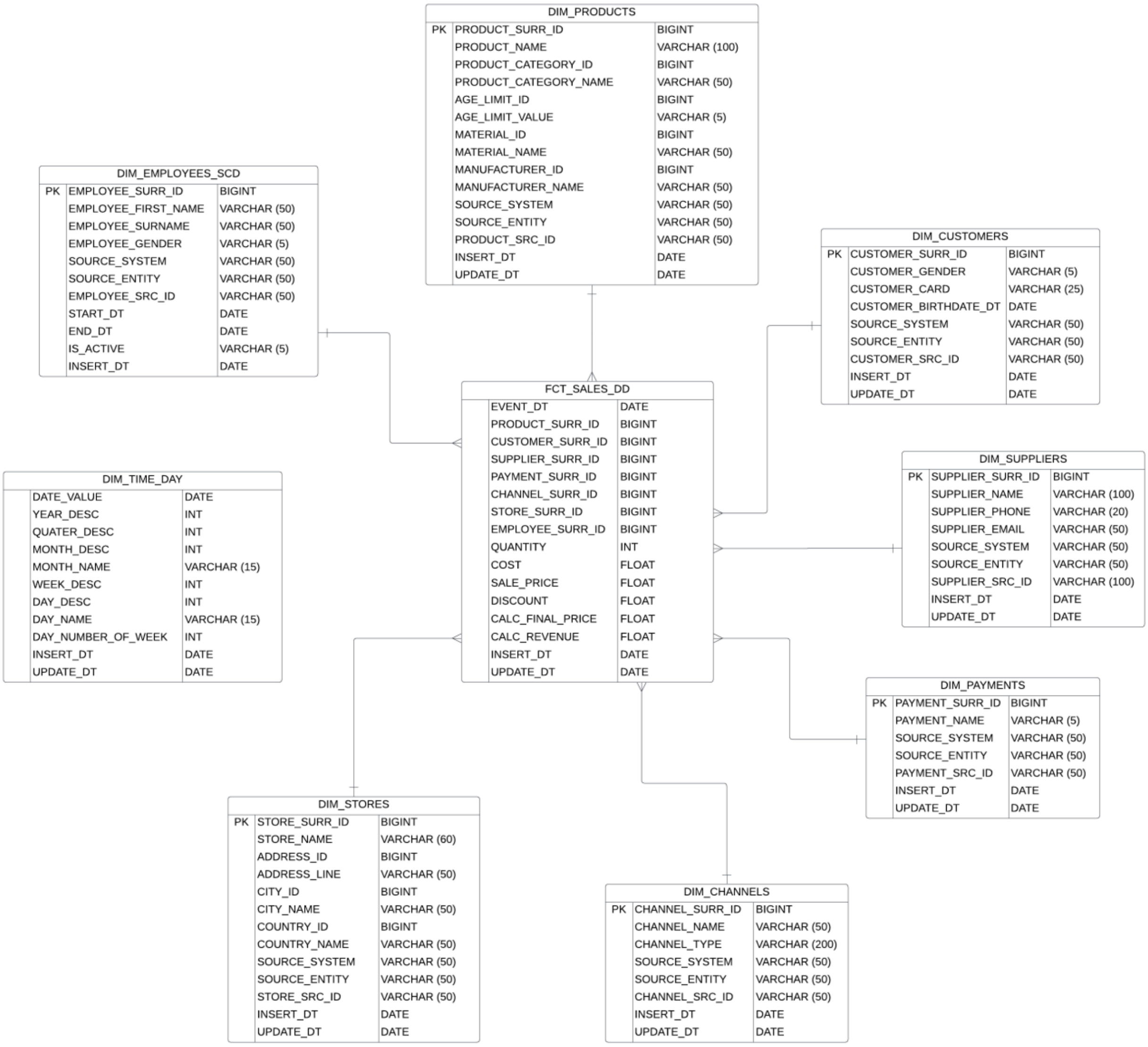
Source entity (SRC) – prepared tables.

Source id (NAME_SRC_ID) – id from source system.

Each dimension has a surrogate key (generated by sequence) and natural key (id from source table).

The data types are described in each table, SOURCE TRIPLET is always VARCHAR type.

Dimension CE_EMPLOYEES is chosen to be brought into SCD2 type. This table was chosen because in this model it is important to keep track of employee information (first name, surname) to maintain data consistency in the system without losing previous data (in cases of marriage, divorce, name change and other situations). Dimension CE_EMPLOYEES has composite PK (EMPLOYEE_ID+START_DT).



DIM_PRODUCTS		
PK	PRODUCT_SURR_ID	BIGINT
	PRODUCT_NAME	VARCHAR (100)
	PRODUCT_CATEGORY_ID	BIGINT
	PRODUCT_CATEGORY_NAME	VARCHAR (50)
	AGE_LIMIT_ID	BIGINT
	AGE_LIMIT_VALUE	VARCHAR (5)
	MATERIAL_ID	BIGINT
	MATERIAL_NAME	VARCHAR (50)
	MANUFACTURER_ID	BIGINT
	MANUFACTURER_NAME	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	PRODUCT_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_EMPLOYEES_SCD		
PK	EMPLOYEE_SURR_ID	BIGINT
	EMPLOYEE_FIRST_NAME	VARCHAR (50)
	EMPLOYEE_SURNAME	VARCHAR (50)
	EMPLOYEE_GENDER	VARCHAR (5)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	EMPLOYEE_SRC_ID	VARCHAR (50)
	START_DT	DATE
	END_DT	DATE
	IS_ACTIVE	VARCHAR (5)
	INSERT_DT	DATE

DIM_CUSTOMERS		
PK	CUSTOMER_SURR_ID	BIGINT
	CUSTOMER_GENDER	VARCHAR (5)
	CUSTOMER_CARD	VARCHAR (25)
	CUSTOMER_BIRTHDATE_DT	DATE
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	CUSTOMER_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

FCT_SALES_DD		
	EVENT_DT	DATE
	PRODUCT_SURR_ID	BIGINT
	CUSTOMER_SURR_ID	BIGINT
	SUPPLIER_SURR_ID	BIGINT
	PAYMENT_SURR_ID	BIGINT
	CHANNEL_SURR_ID	BIGINT
	STORE_SURR_ID	BIGINT
	EMPLOYEE_SURR_ID	BIGINT
	QUANTITY	INT
	COST	FLOAT
	SALE_PRICE	FLOAT
	DISCOUNT	FLOAT
	CALC_FINAL_PRICE	FLOAT
	CALC_REVENUE	FLOAT
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_TIME_DAY		
	DATE_VALUE	DATE
	YEAR_DESC	INT
	QUATER_DESC	INT
	MONTH_DESC	INT
	MONTH_NAME	VARCHAR (15)
	WEEK_DESC	INT
	DAY_DESC	INT
	DAY_NAME	VARCHAR (15)
	DAY_NUMBER_OF_WEEK	INT
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_SUPPLIERS		
PK	SUPPLIER_SURR_ID	BIGINT
	SUPPLIER_NAME	VARCHAR (100)
	SUPPLIER_PHONE	VARCHAR (20)
	SUPPLIER_EMAIL	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	SUPPLIER_SRC_ID	VARCHAR (100)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_PAYMENTS		
PK	PAYMENT_SURR_ID	BIGINT
	PAYMENT_NAME	VARCHAR (5)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	PAYMENT_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_STORES		
PK	STORE_SURR_ID	BIGINT
	STORE_NAME	VARCHAR (60)
	ADDRESS_ID	BIGINT
	ADDRESS_LINE	VARCHAR (50)
	CITY_ID	BIGINT
	CITY_NAME	VARCHAR (50)
	COUNTRY_ID	BIGINT
	COUNTRY_NAME	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	STORE_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_CHANNELS		
PK	CHANNEL_SURR_ID	BIGINT
	CHANNEL_NAME	VARCHAR (50)
	CHANNEL_TYPE	VARCHAR (200)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	CHANNEL_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

Pic. 1 – 3NF layer

3 BUSINESS LAYER DIMENSIONAL MODEL

At this stage, the 3NF schema is processed into a DIM model. To do this, the tables are denormalized, followed by the creation of a fact table and its dimensions. Each dimension has a Source Triplet (SOURCE_SYSTEM, SOURCE_ENTITY, SOURSE_ID).

Source system (SA) – describes the source this row is loaded from.

Source entity (SRC) – describes the entity this row is loaded from.

Source id (NAME_SRC_ID) – describe how the row can be identified in source entity.

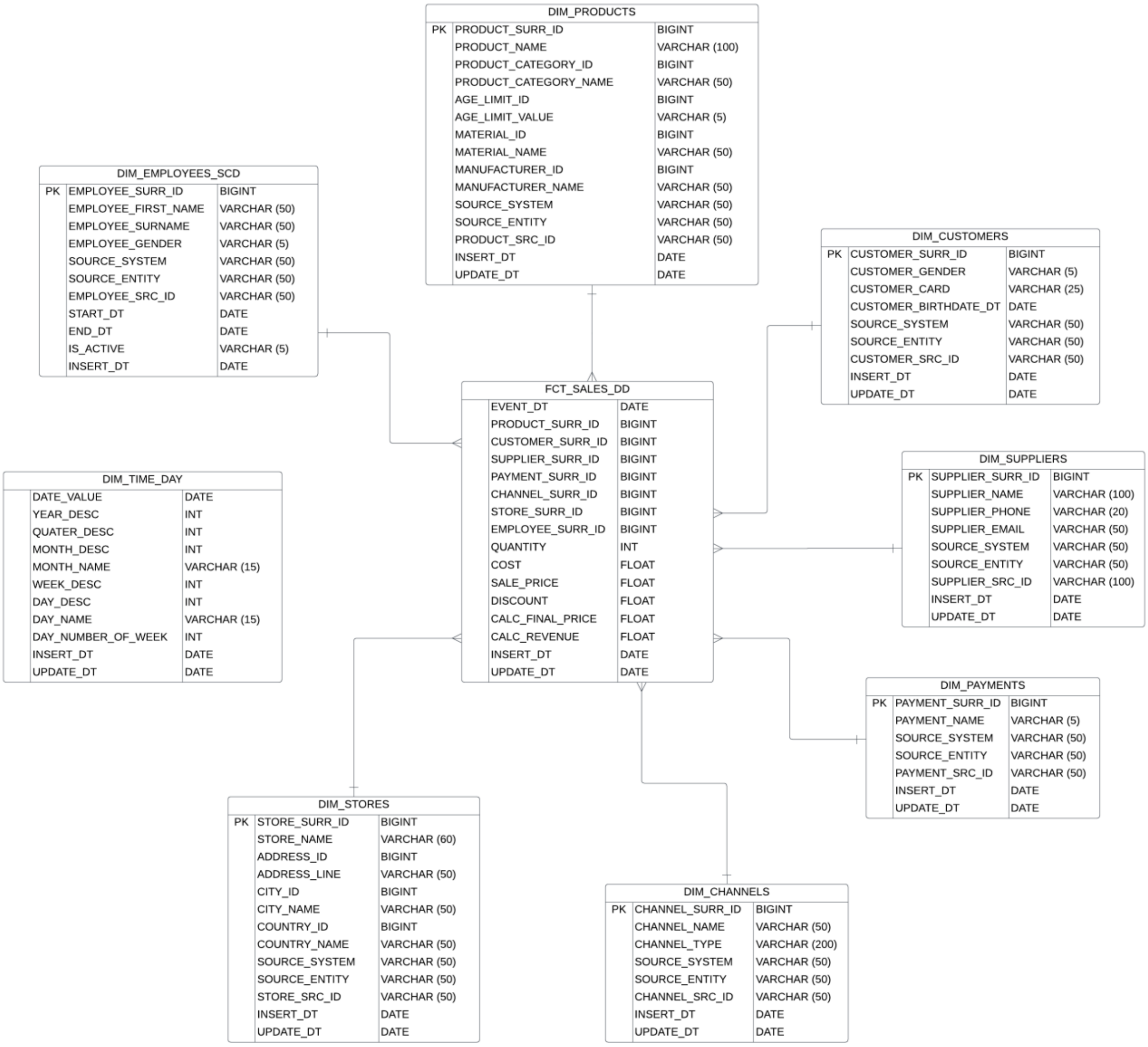
<Name>_SURR_ID – new generated by sequences id for each dimensions.

Columns INSERT_DT, UPDATE_DT, START_DT, END_DT, IS_ACTIVE, INSERT_DT depending on type (SCD TYPE1 or SCD TYPE2).

A separate DIM_TIME_DAY dimension is also created for which information to fill is generated using an SQL script. This dimension represents itself a set of dates from 2020-01-01 to 2023-07-01 and has a logical relationship with the fact table (by column EVENT_DT). DATE_VALUE stores data information in YYYY-MM-DD format.

A fact table is a collection of foreign keys of related dimensions and metrics.
QUANTITY – the quantity of toys sold on a specific day.
COST – the production cost of one toy.
SALE_PRICE – the manufacturer's selling price of one toy.
DISCOUNT – the discount applied to the toy's selling price.
CALC_FINAL_PRICE – the calculated metric final price of one toy after applying the discount: SALE_PRICE - DISCOUNT.
CALC_REVENUE: the calculated metric total revenue from toy sales on a specific day: CALC_FINAL_PRICE * QUANTITY.

The stores dimension includes all information about the store and its location, the product dimension includes all product information (category, age limit, material and manufacturer).



DIM_PRODUCTS		
PK	PRODUCT_SURR_ID	BIGINT
	PRODUCT_NAME	VARCHAR (100)
	PRODUCT_CATEGORY_ID	BIGINT
	PRODUCT_CATEGORY_NAME	VARCHAR (50)
	AGE_LIMIT_ID	BIGINT
	AGE_LIMIT_VALUE	VARCHAR (5)
	MATERIAL_ID	BIGINT
	MATERIAL_NAME	VARCHAR (50)
	MANUFACTURER_ID	BIGINT
	MANUFACTURER_NAME	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	PRODUCT_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_EMPLOYEES_SCD		
PK	EMPLOYEE_SURR_ID	BIGINT
	EMPLOYEE_FIRST_NAME	VARCHAR (50)
	EMPLOYEE_SURNAME	VARCHAR (50)
	EMPLOYEE_GENDER	VARCHAR (5)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	EMPLOYEE_SRC_ID	VARCHAR (50)
	START_DT	DATE
	END_DT	DATE
	IS_ACTIVE	VARCHAR (5)
	INSERT_DT	DATE

DIM_CUSTOMERS		
PK	CUSTOMER_SURR_ID	BIGINT
	CUSTOMER_GENDER	VARCHAR (5)
	CUSTOMER_CARD	VARCHAR (25)
	CUSTOMER_BIRTHDATE_DT	DATE
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	CUSTOMER_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

FCT_SALES_DD		
	EVENT_DT	DATE
	PRODUCT_SURR_ID	BIGINT
	CUSTOMER_SURR_ID	BIGINT
	SUPPLIER_SURR_ID	BIGINT
	PAYMENT_SURR_ID	BIGINT
	CHANNEL_SURR_ID	BIGINT
	STORE_SURR_ID	BIGINT
	EMPLOYEE_SURR_ID	BIGINT
	QUANTITY	INT
	COST	FLOAT
	SALE_PRICE	FLOAT
	DISCOUNT	FLOAT
	CALC_FINAL_PRICE	FLOAT
	CALC_REVENUE	FLOAT
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_SUPPLIERS		
PK	SUPPLIER_SURR_ID	BIGINT
	SUPPLIER_NAME	VARCHAR (100)
	SUPPLIER_PHONE	VARCHAR (20)
	SUPPLIER_EMAIL	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	SUPPLIER_SRC_ID	VARCHAR (100)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_TIME_DAY		
	DATE_VALUE	DATE
	YEAR_DESC	INT
	QUATER_DESC	INT
	MONTH_DESC	INT
	MONTH_NAME	VARCHAR (15)
	WEEK_DESC	INT
	DAY_DESC	INT
	DAY_NAME	VARCHAR (15)
	DAY_NUMBER_OF_WEEK	INT
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_PAYMENTS		
PK	PAYMENT_SURR_ID	BIGINT
	PAYMENT_NAME	VARCHAR (5)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	PAYMENT_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

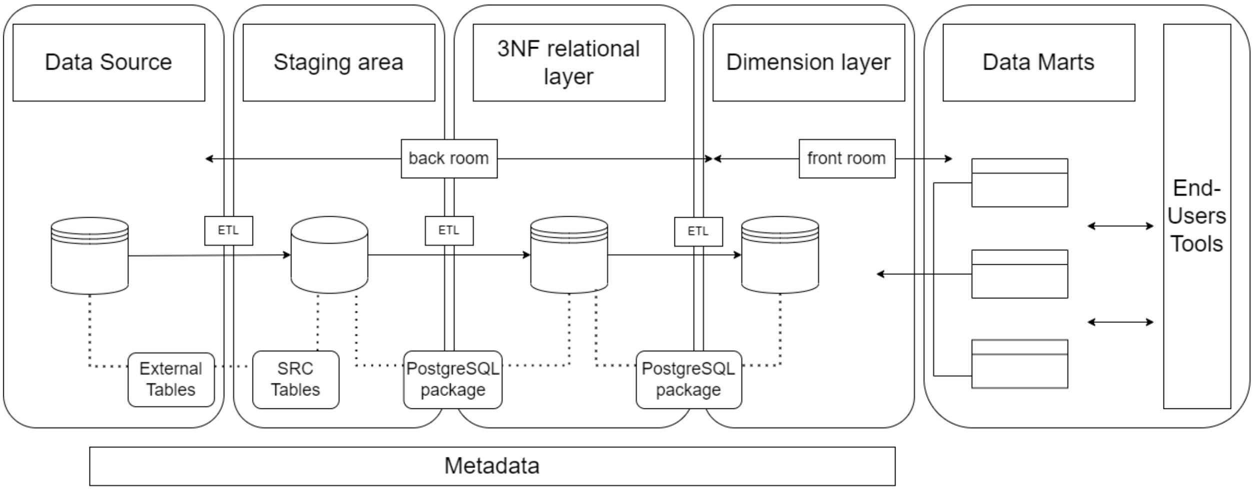
DIM_STORES		
PK	STORE_SURR_ID	BIGINT
	STORE_NAME	VARCHAR (60)
	ADDRESS_ID	BIGINT
	ADDRESS_LINE	VARCHAR (50)
	CITY_ID	BIGINT
	CITY_NAME	VARCHAR (50)
	COUNTRY_ID	BIGINT
	COUNTRY_NAME	VARCHAR (50)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	STORE_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

DIM_CHANNELS		
PK	CHANNEL_SURR_ID	BIGINT
	CHANNEL_NAME	VARCHAR (50)
	CHANNEL_TYPE	VARCHAR (200)
	SOURCE_SYSTEM	VARCHAR (50)
	SOURCE_ENTITY	VARCHAR (50)
	CHANNEL_SRC_ID	VARCHAR (50)
	INSERT_DT	DATE
	UPDATE_DT	DATE

Pic. 2 – Dimensional model

4 LOGICAL SCHEME

Logical model for the DWH load process. The model includes Data Source, Staging area, 3NF relational layer, Dimension layer and Data marts which provide information to users.



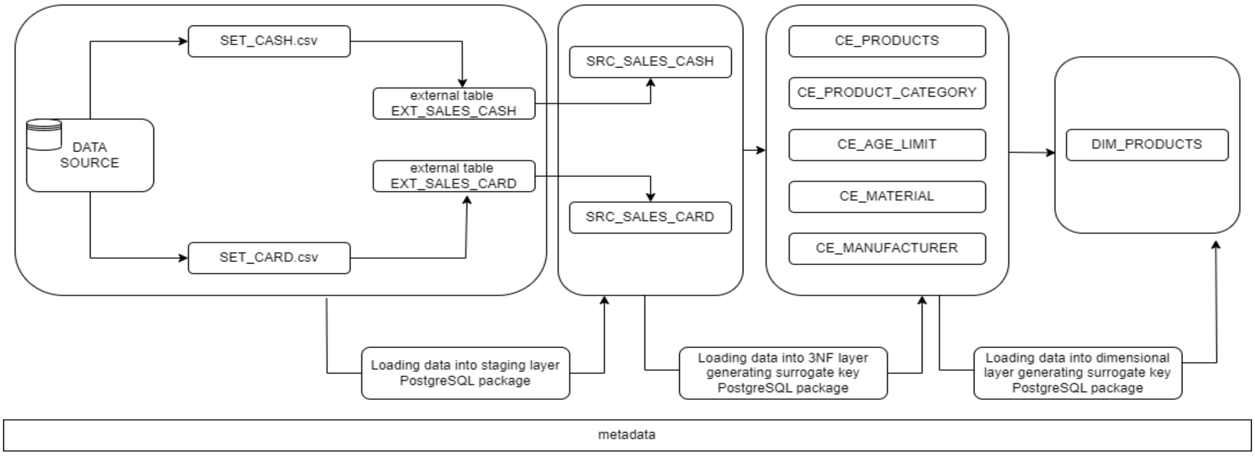
Pic. 3 – Logical schema

5 DATA FLOW

Data flows diagrams represent a high-level view of the DWH load process and describe the important data sources, transformations and destinations involved in the process.

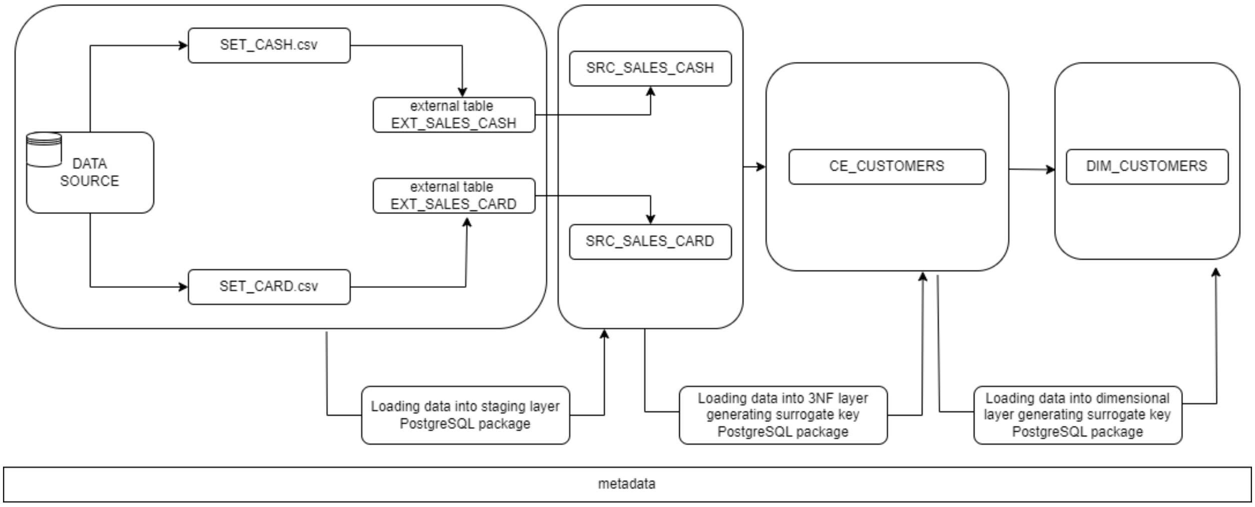
On each diagram was represented two sets (SET_CASH.csv, SET_CARD.csv), external tables (EXT_SALES_CASH, EXT_SALES_CARD), staging layer tables (SRC_SALES_CASH, SRC_SALES_CARD) and different tables from 3 NF layer and dimensional layer. Data was loaded to new step by PostgreSQL script.

When data loaded from 3NF to dimensional layer in case of DIM_PRODUCTS data was chosen from five tables from 3NF layer (CE_PRODUCTS, CE_PRODUCT_CATEGORY, CE_AGE_LIMIT, CE_MATERIAL, CE_MANUFACTURER).

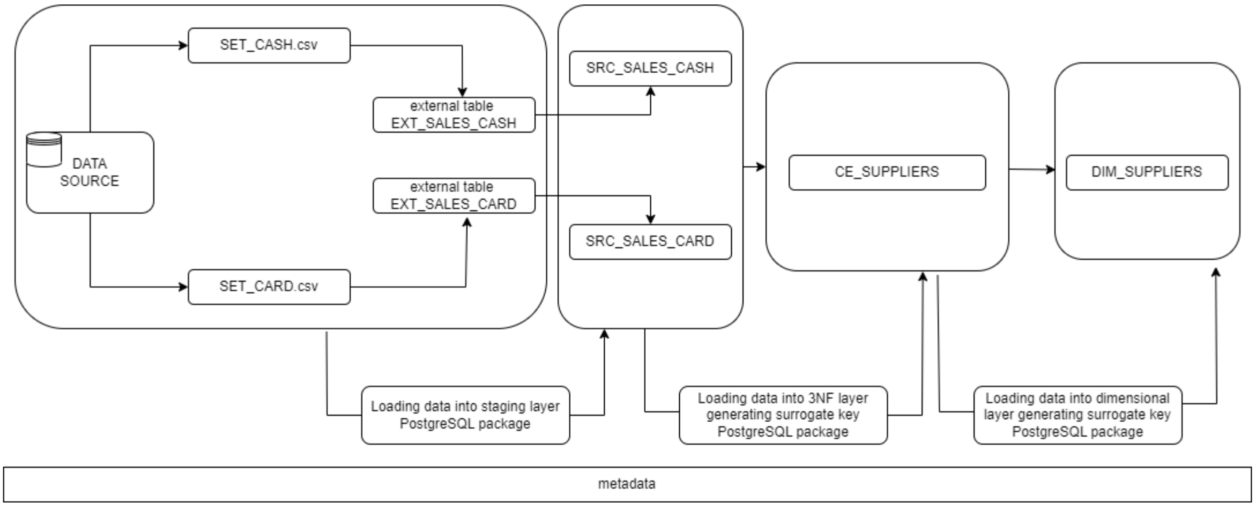

DATA SOURCE
SET_CASH.csv
SET_CARD.csv
external table EXT_SALES_CASH
external table EXT_SALES_CARD
SRC_SALES_CASH
SRC_SALES_CARD
CE_PRODUCTS
CE_PRODUCT_CATEGORY
CE_AGE_LIMIT
CE_MATERIAL
CE_MANUFACTURER
DIM_PRODUCTS
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

Pic.4 – Data flow for Products

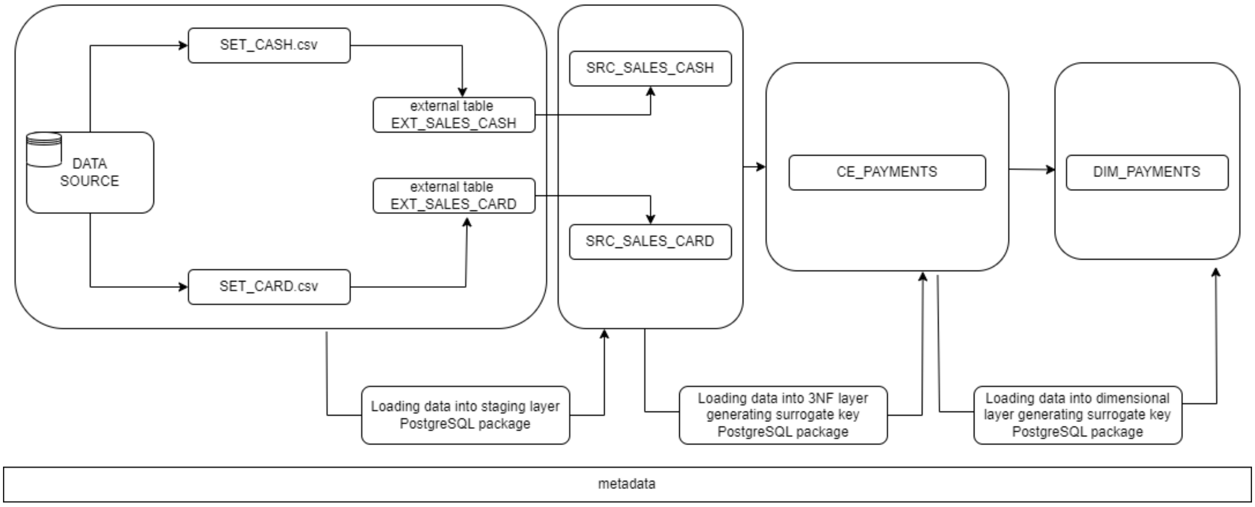
For DIM_CUSTOMERS, DIM_SUPPLIERS, DIM_PAYMENTS, DIM_CHANNELS, DIM_EMPLOYEES_SCD data was chosen from CE_CUSTOMERS, CE_ SUPPLIERS, CE_ PAYMENTS, CE_ CHANNELS, CE_ EMPLOYEES_SCD tables on 3NF layer accordingly.


SET_CASH.csv
external table EXT_SALES_CASH
DATA SOURCE
external table EXT_SALES_CARD
SET_CARD.csv
SRC_SALES_CASH
SRC_SALES_CARD
CE_CUSTOMERS
DIM_CUSTOMERS
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

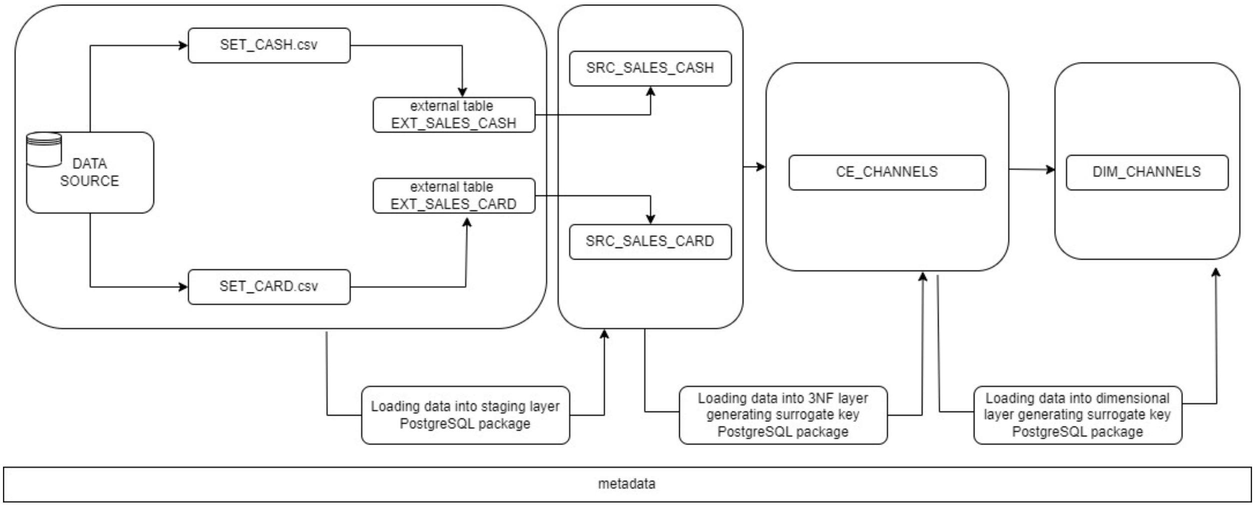
Pic. 5 – Data flow for Customers


SET_CASH.csv
external table EXT_SALES_CASH
DATA SOURCE
external table EXT_SALES_CARD
SET_CARD.csv
SRC_SALES_CASH
SRC_SALES_CARD
CE_SUPPLIERS
DIM_SUPPLIERS
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

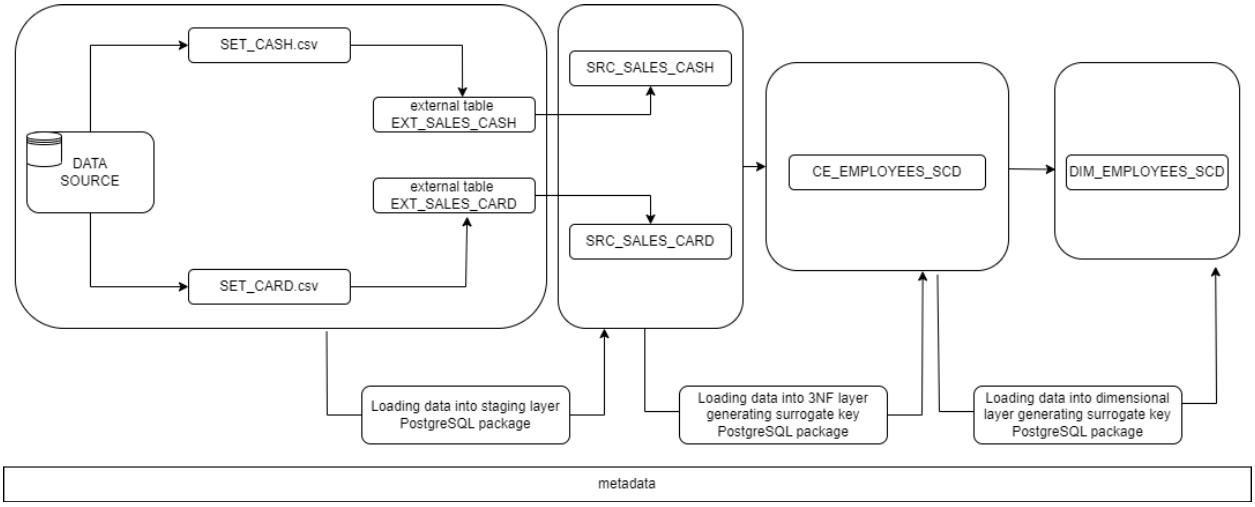
Pic. 6 – Data flow for Suppliers


SET_CASH.csv
external table EXT_SALES_CASH
DATA SOURCE
external table EXT_SALES_CARD
SET_CARD.csv
SRC_SALES_CASH
SRC_SALES_CARD
CE_PAYMENTS
DIM_PAYMENTS
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

Pic. 7 – Data flow for Payments

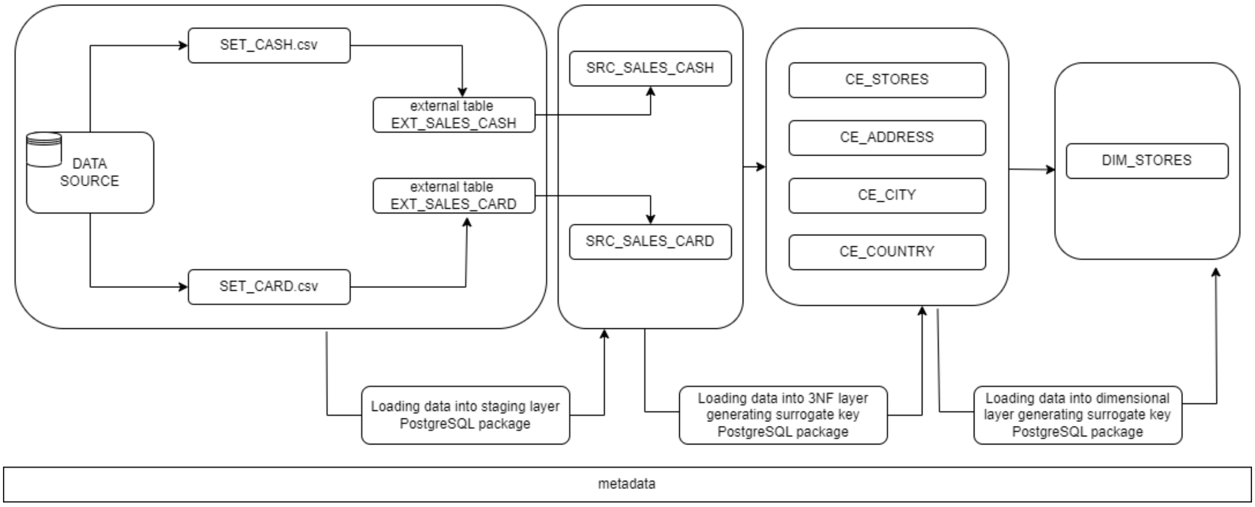

SET_CASH.csv
external table EXT_SALES_CASH
DATA SOURCE
external table EXT_SALES_CARD
SET_CARD.csv
SRC_SALES_CASH
SRC_SALES_CARD
CE_CHANNELS
DIM_CHANNELS
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

Pic.8 – Data flow for Channels


SET_CASH.csv
external table EXT_SALES_CASH
DATA SOURCE
external table EXT_SALES_CARD
SET_CARD.csv
SRC_SALES_CASH
SRC_SALES_CARD
CE_EMPLOYEES_SCD
DIM_EMPLOYEES_SCD
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

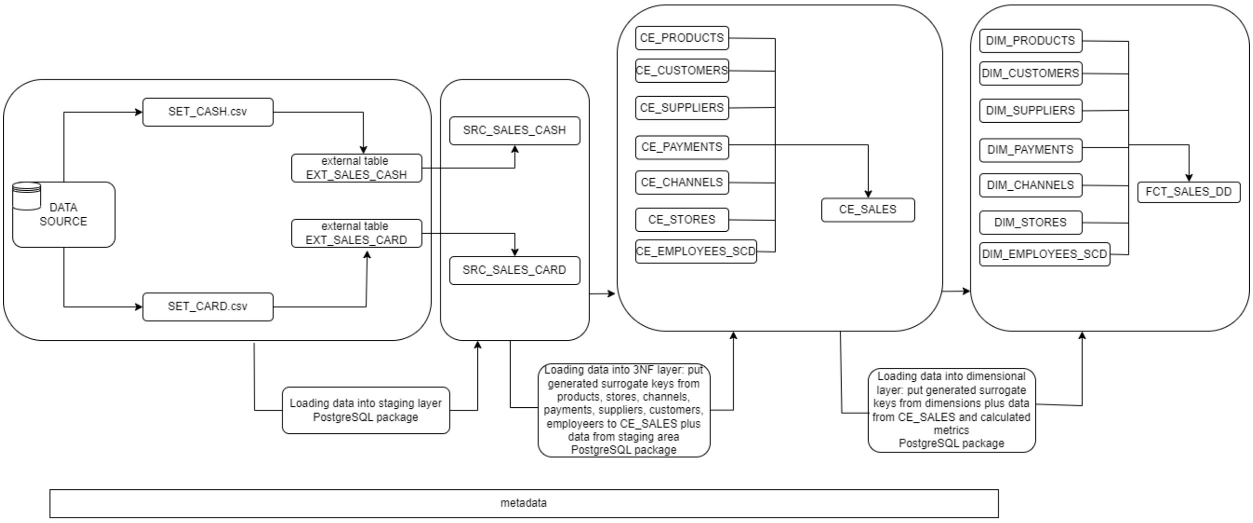
Pic. 9 – Data flow for Employees

For DIM_STORES data was chosen from CE_STORES, CE_ADDRESS, CE_CITY, CE_COUNTRY on 3NF layer.


DATA SOURCE
SET_CASH.csv
SET_CARD.csv
external table EXT_SALES_CASH
external table EXT_SALES_CARD
SRC_SALES_CASH
SRC_SALES_CARD
CE_STORES
CE_ADDRESS
CE_CITY
CE_COUNTRY
DIM_STORES
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer generating surrogate key PostgreSQL package
Loading data into dimensional layer generating surrogate key PostgreSQL package
metadata

Pic.10 – Data flow for Stores

For fact table data are included generated id from products, stores, channels, payments, suppliers, customers, employees tables and data from staging area. This data is put to CE_SALES without generating surrogate id. To FCT_SALES_DD data is loaded from CE_SALES plus added calculated metrics (CALC_FINAL_PRICE, CALC_REVENUE) by PostgreSQL package.


DATA SOURCE
SET_CASH.csv
SET_CARD.csv
external table EXT_SALES_CASH
external table EXT_SALES_CARD
SRC_SALES_CASH
SRC_SALES_CARD
CE_PRODUCTS
CE_CUSTOMERS
CE_SUPPLIERS
CE_PAYMENTS
CE_CHANNELS
CE_STORES
CE_EMPLOYEES_SCD
CE_SALES
DIM_PRODUCTS
DIM_CUSTOMERS
DIM_SUPPLIERS
DIM_PAYMENTS
DIM_CHANNELS
DIM_STORES
DIM_EMPLOYEES_SCD
FCT_SALES_DD
Loading data into staging layer PostgreSQL package
Loading data into 3NF layer: put generated surrogate keys from products, stores, channels, payments, suppliers, customers, employeers to CE_SALES plus data from staging area PostgreSQL package
Loading data into dimensional layer: put generated surrogate keys from dimensions plus data from CE_SALES and calculated metrics PostgreSQL package
metadata

Pic. 11 – Data flow for Sales

6 FACT TABLE PARTITIONING STRATEGY

For the CE_SALES table, an incremental loading method was chosen. The data is loaded at the first load, then incremental loading goes depending on flag and the table is filled with only new data.

Partitioning is a database design strategy that involves dividing large tables into smaller, more manageable pieces called partitions based on certain criteria. The primary goal of partitioning is to improve query performance, data management, and maintenance. A fact table FCT_SALES_DD, the partitioning strategy is designed to organize data efficiently based on the EVENT_DT column, which represents the date of the sales event.

The fact table is partitioned using the RANGE partitioning strategy:

fct_sales_dd_2020: This partition contains data for events that occurred from January 1, 2020, to January 1, 2021.

fct_sales_dd_2021: This partition contains data for events that occurred from January 1, 2021, to January 1, 2022.

fct_sales_dd_2022: This partition contains data for events that occurred from January 1, 2022, to January 1, 2023.

fct_sales_dd_2023: This partition contains data for events that occurred from January 1, 2023, to January 1, 2024.

Each partition stores data for a specific date range, and this partitioning strategy can significantly improve query performance when selecting or aggregating data based on event dates. Additionally, it simplifies data archiving and maintenance tasks, allows to manage data in smaller, more manageable parts.