

|  |
| --- |
| Business Template  **Iowa liquor sales** |
| **Logo / Image** |

Contents

[1 Business Description 3](#_Toc155614186)

[1.1 Business background 3](#_Toc155614187)

[1.2 Problems because of poor data management 3](#_Toc155614188)

[1.3 Benefits from implementing a Data Warehouse 3](#_Toc155614189)

[1.4 DATASETS DESCRIPTION 3](#_Toc155614190)

[1.5 GRAIN / DIM / FACT 4](#_Toc155614191)

[2 Business Layer 3NF 4](#_Toc155614192)

[3 Business Layer Dimensional Model 4](#_Toc155614193)

[4 Logical Scheme 4](#_Toc155614194)

[5 Data Flow 4](#_Toc155614195)

[6 Fact Table Partitioning Strategy 4](#_Toc155614196)

# 

# Business Description

## Business background

Our Iowa Liquor Company is all about providing great drinks for good times. We know people have different tastes, and we want to offer a unique range of spirits to make every occasion special. The liquor business is competitive, so we pay close attention to what people like and the trends in the market. Understanding what our customers want helps us stand out and be successful in the world of drinks.

## Problems because of poor data management

Right now, our Iowa Liquor Company faces some challenges because of poor data management. Here's what we're dealing with:

Limited Insights:

We don't have a clear picture of what our customers prefer and the trends in the liquor market. This lack of information makes it difficult to make informed decisions about which drinks to offer and how to stay competitive.

Uncertain Pricing Strategies:

Poor data management means we're not sure how to price our drinks effectively. Understanding what prices customers are comfortable with and how they relate to different types of drinks is crucial for our business success.

Missed Opportunities:

Without good data, we might be missing out on opportunities. We could be unaware of emerging trends, popular preferences, or new market segments that could boost our sales and reputation.

Inefficient Marketing:

We might not be reaching our customers effectively. Without proper data, it's challenging to tailor marketing strategies to specific customer preferences and behaviors. This can lead to ineffective advertising and promotions.

Lack of Adaptability:

In the dynamic world of the liquor business, being adaptable is key. Poor data management makes it harder for us to adjust quickly to changing customer demands and market trends, potentially putting us at a disadvantage.

Improving our data management practices will not only address these challenges but also pave the way for more informed, effective, and successful business operations at Iowa Liquor Company.

## Benefits from implementing a Data Warehouse

Implementing a data warehouse at Iowa Liquor Company comes with several advantages that will significantly improve our business operations:

Informed Decision-Making:

A data warehouse provides us with a centralized and organized source of information. This means we can make better decisions about which drinks to offer, how to price them, and how to market them effectively.

Understanding Customer Preferences:

With a data warehouse, we can analyze customer data to understand their preferences. This helps us tailor our product offerings to match what our customers love, ensuring higher satisfaction and loyalty.

Optimized Pricing Strategies:

We can use the data warehouse to analyze pricing dynamics, helping us determine optimal price points for our drinks. This ensures that our pricing is competitive while still meeting the expectations of our customers.

Identifying Trends and Opportunities:

The data warehouse allows us to spot trends early and identify new opportunities in the market. This means we can stay ahead of the competition and explore new avenues for growth.

Targeted Marketing Campaigns:

Armed with insights from the data warehouse, we can create more targeted and effective marketing campaigns. This ensures that our promotions reach the right audience at the right time, maximizing their impact.

Improved Inventory Management:

Understanding consumer behavior and preferences helps us manage our inventory more efficiently. We can stock the right amount of each product, reducing waste and ensuring we always have what our customers want.

Enhanced Adaptability:

The data warehouse equips us with real-time information, making it easier to adapt to changes in the market and customer demands. This agility is crucial for staying competitive in the dynamic liquor industry.

Predictive Analytics:

Utilizing advanced analytics from the data warehouse allows us to predict future trends. This foresight enables us to plan ahead, ensuring we are well-prepared for shifts in customer preferences and market dynamics.

Implementing a data warehouse is not just a technological upgrade; it's a strategic investment that will empower Iowa Liquor Company to thrive in a competitive market by making data-driven decisions and staying attuned to the needs and preferences of our valued customers.

## DATASETS DESCRIPTION

Our company has online and offline sales. For this reason we have two datasets, one for online sales and the second for offline – in store sales.

First dataset is about online sales and contain information about:

Products:

Category: The category of the drinks.

Subcategory: Further classification of the drinks within the category.

Sales Information:

Date: The date of the sale.

Retail Price: The retail price of the bottle.

Sales Price: The actual selling price of the bottle.

Quantity Sold: The number of units sold.

Customer Information:

Gender: The gender.

Age: The age range or demographic information of the customer.

Location: The geographical location of the customer.

Personal information: The passport number, contact information and etc.

Employee Information:

Gender: The gender.

Age: The age range or demographic information of the customer.

Location: The geographical location of the customer.

Personal information: The passport number, contact information and etc.

Position: Includes details about position of the employee.

Warehouse Information:

Name: Includes warehouse names

Address: Includes address information of the warehouse

gender Information:

Name: Includes gender names

The second dataset, which is about offline sales contains the following information:

Products:

Category: The category of the drinks.

Subcategory: Further classification of the drinks within the category.

Sales Information:

Date: The date of the sale.

Retail Price: The retail price of the bottle.

Sales Price: The actual selling price of the bottle.

Quantity Sold: The number of units sold.

Employee Information:

Gender: The gender.

Age: The age range or demographic information of the customer.

Location: The geographical location of the customer.

Personal information: The passport number, contact information and etc.

Position: Includes details about position of the employee.

Store Information:

Address: information about address of the stores like city, zip code and etc.

Name: name of the store

Gender Information:

Name: Includes gender names

The datasets provide a comprehensive overview of sales, allowing for analysis and exploration of trends, sales performance, customer preferences, and more within the beverage industry.

## GRAIN / DIM / FACT

We applied Kimball’s model of business process which contains 4 steps of modelling.

1. Picking a business process to the model

So, in this step we asked questions and thought what we want to take from our data. What insights will be necessary for our business to grow.

We decided that our main goal is to grow our sales and retain our customers by analyzing their pains and expectations. Also, we want to make predictions about our future sales depending on our historical data.

1. Decide what will our grain be

The grain is the most atomic part of the fact table. So, in our case grain will be each line in our invoice. For example, if we have order with 3 items in it, our grain will describe this separately in the fact table. So we will have 3 rows in the fact table for this order.

1. Choosing dimensions for the fact table

Dimensions give us opportunity to analyze data in fact table in the context of the dimensions. So, we have to ask questions like how, why, who and etc.

In our model, we have these dimensions:

* Date
* Products
* Customers
* Employees
* Warehouses
* Genders
* Stores

1. Identifying numeric values for our fact table

In this step, we decided on the numeric measure of our grain. We will store information about our prices and quantities in the fact table. This will help us to calculate almost everything we might be interested in.

### Description of our dimension tables:

Products dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| product\_id | Identifies products uniquely | BIGINT |
| product\_name | Product name | VARCHAR |
| product\_category\_id | Identifies category uniquely | BIGINT |
| product\_category\_name | Name of product category | VARCHAR |
| Bottle\_price\_retail | Price in stores per bottle | Float |
| Bottle\_price | Price for manufacturing per bottle | Float |
| Volume\_of\_the\_bottle | Bottle volume in ml | Float |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| Product\_ID | Product\_name | Product\_category\_ID | Product\_category\_name |
| 1 | Bacardi gold rum | 123 | Puerto Rico & Virgin Islands |

|  |  |  |
| --- | --- | --- |
| Bottle\_price\_retail | Bottle\_price | Volume\_of\_bottle |
| 10.0 | 8.0 | 1000 |

Customers dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| customer\_id | Identifies customer uniquely | BIGINT |
| customer\_first\_name | Name of customer | Varchar |
| customer\_last\_name | Surname of customer | Varchar |
| customer\_passport\_id | Passport unique number | varchar |
| customer\_gender\_id | Gender of customer ID | bigint |
| customer\_email | Email address of customer | Varchar |
| customer\_phone | Phone number of customer | Varchar |
| customer\_birth\_date | Date of birth | Date |
| customer\_address\_id | Unique identifier of address | bigint |
| customer\_address | Street address | Varchar |
| customer\_city\_id | Unique identifier of city | bigint |
| customer\_city | Name of the city | Varchar |
| customer\_zip\_code | Zip code | varchar |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| Customer\_id | Customer\_first\_name | Customer\_last\_name | Customer\_passport\_id |
| 1 | Nino | Alavidze | AD10056 |

|  |  |  |  |
| --- | --- | --- | --- |
| Customer\_gender | Customer\_email | Customer\_phone | Customer\_birth\_date |
| 2 | [ninoalavidze@gmail.com](mailto:ninoalavidze@gmail.com) | 9999-999-999 | 1995-12-06 |

|  |  |  |  |
| --- | --- | --- | --- |
| Customer\_address\_id | Customer\_address | Customer\_city\_id | Customer\_city |
| 2 | Someone’s street. 45 | 10 | Vienna |

|  |
| --- |
| Customer\_zip\_code |
| 55090 |

Employees dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| employee\_id | Identifies employee uniquely | BIGINT |
| employee \_first\_name | Name of employee | Varchar |
| employee \_last\_name | Surname of employee | Varchar |
| employee \_passport\_id | Passport unique number | Varchar |
| employee \_gender\_id | Gender of employee id | bigint |
| employee \_email | Email address of employee | Varchar |
| employee \_phone | Phone number of employee | Varchar |
| employee \_birth\_date | Date of birth | Date |
| employee\_address\_id | Unique identifier of address | bigint |
| employee \_address | Street address | Varchar |
| employee\_city\_id | Unique identifier of city | bigint |
| employee \_city | Name of the city | Varchar |
| employee \_zip\_code | Zip code | Varchar |
| position\_start\_date | Date when employee start working on this position | Date |
| position\_name | Name of position | varchar |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| employee\_id | employee \_first\_name | employee \_last\_name | Employee\_passport\_id |
| 2 | Marya | Razmadze | 10020120 |

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_gender\_id | Employee\_email | Employee\_phone | Employee\_birth\_date |
| 2 | [marya@gmail.com](mailto:marya@gmail.com) | 9999-999-990 | 1995-12-06 |

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_address\_id | Employee\_address | Employee\_city\_id | Employee\_city |
| 4 | street 5 | 3 | [London](mailto:ninoalavidze@gmail.com) |

|  |  |  |
| --- | --- | --- |
| Employee\_zip\_code | Position\_start\_date | Position\_name |
| 5090P | 2000-12-06 | agent |

Stores dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Store\_id | Identifies store uniquely | BIGINT |
| Store\_name | Name of store | Varchar |
| Store\_address\_id | Unique identifier of address | bigint |
| Store\_address | Street address | Varchar |
| Store\_city\_id | Unique identifier of city | bigint |
| Store\_city | City of the store location | Varchar |
| Store\_zip\_code | Zip code of the store address | varchar |

Exmaples filled with data:

|  |  |  |  |
| --- | --- | --- | --- |
| Store\_id | Store\_name | Store\_address\_id | Store\_address |
| 1 | Store 1 | 3 | Street 3 |

|  |  |  |
| --- | --- | --- |
| Store\_city\_id | Store\_city | Store\_zip\_code |
| 11 | Cologne | 50890 |

Warehouses dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Warehouse\_id | Identifies warehouse uniquely | BIGINT |
| Warehouse\_name | Name of warehouse | Varchar |
| Warehouse\_address\_id | Unique identifier of address | Bigint |
| Warehouse\_address | Street address | Varchar |
| Warehouse\_city\_id | Unique identifier of city | bigint |
| Warehouse\_city | City of the warehouse location | Varchar |
| Warehouse\_zip\_code | Zip code of the warehouse address | varchar |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| warehouse\_id | warehouse \_name | Warehouse\_address\_id | warehouse \_address |
| 1 | WH-1 | 3 | Street 3 |

|  |  |  |
| --- | --- | --- |
| Warehouse\_city\_id | warehouse \_city | Warehouse\_zip\_code |
| 11 | Cologne | 50890 |

Genders dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Gender\_id | Identifies gender uniquely | BIGINT |
| Gender\_name | Name of gender | Varchar |

Example with filled data:

|  |  |
| --- | --- |
| gender\_id | gender \_name |
| 1 | Male |

Date dim:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Date\_id | Identifies date uniquely, is date itself | BIGINT |
| Day\_of\_week | Day of week | varchar |
| Day\_of\_month | Day of month | Smallint |
| Day\_of\_year | Day\_of\_year | Smallint |
| Week\_of\_year | Week\_of\_year | Smallint |
| Month\_of\_year | Month of year | Varchar |
| Quarter\_of\_year | Quarter of the year | Varchar |
| Year | Year | Smallint |
| day\_of\_fiscal\_month | Fiscal month day | Smallint |
| Month\_of\_fiscal\_year | Fiscal year month | Varchar |
| Quarter\_of\_Fiscal\_year | Fiscal year quarter | Varchar |
| Fiscal\_year | Fiscal year | smallint |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| Date\_id | Day\_of\_week | Day\_of\_month | Day\_of\_year |
| 20240122 | 2024-01-22 | 22 | Monday |

|  |  |  |  |
| --- | --- | --- | --- |
| Week\_of\_year | Month\_of\_year | Quarter\_of\_year | year |
| 3 | January | 1 | 2024 |

|  |  |  |  |
| --- | --- | --- | --- |
| Day\_of\_fiscal\_month | Month\_of\_fiscal\_year | Quarter\_of\_fisacl\_year | Fiscal\_year |
| 1 | January | 1 | 2024 |

Sales fact table:

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| Invoice\_id | Invoice identifier | Varchar |
| Event\_dt | Date identifier | bigint |
| Product\_id | Product identifier | Bigint |
| Customer\_id | Customer identifier | Bigint |
| Employee\_id | Employee identifier | Bigint |
| Warehouse\_id | Warehouse identifier | Bigint |
| Store\_id | Store identifier | Bigint |
| Quantity\_of\_bottle\_sold | Sold bottles | Int |
| Price | Whole price for the one order | float |

Example with filled data:

|  |  |  |  |
| --- | --- | --- | --- |
| Invoice\_id | Date\_id | Product\_id | Customer\_id |
| Inv2000190 | 20240122 | 1 | 2 |

|  |  |  |
| --- | --- | --- |
| Employee\_id | warehouse\_id | Store\_id |
| 1 | 5 | 10 |

|  |  |
| --- | --- |
| Quantity\_of\_bottles\_sold | Price |
| 2 | 20.0 |

# Business Layer 3NF

To reduce redundancy and make maintenance easier we made our database in third normal form (3NF).

Here is our 3NF model of database:



We have 12 tables connecting each other in different ways.

Our tables are:

* CE\_CUSTOMERS
* CE\_EMPLOYEES\_SCD
* CE\_STORES
* CE\_GENDERS
* CE\_WAREHOUSES
* CE\_VENDORS\_TO\_WAREHOUSES
* CE\_ADDRESSES
* CE\_CITIES
* CE\_PRODUCTS
* CE\_PRODUCT\_CATEGORIES
* CE\_SALES
* CE\_DATES

**CE\_CUSTOMERS** table contains customer personal information, such as name, surname, phone, email, birth date, address\_id.

Table has two connecting tables: CE\_ADDRESSES and CE\_INVOICES.

CE\_CUSTOMERS and CE\_ADRESSES have one-to-many relationship as one customer can have only one address, but one address from CE\_ADDRESSES can have several customer on the same address.

CE\_CUSTOMER and CE\_INVOICES have many-to-one relationship as one customer can have many invoices but one invoice can only have one customer.

**CE\_EMPLOYEES\_SCD** table contains personal information about employees plus start date of the position and position name they work on.

SCD on the end of the table name means that it is slowly changing dimension. Which means that we are collecting historical data about employees.

CE\_EMPLOYESS and CE\_ADRESSES have one-to-many relationship as one employee can have only one address, but one address from CE\_ADDRESSES can have several employees on the same address.

CE\_EMPLOYEES and CE\_INVOICES have many-to-one relationship as one employee can have many invoices but one invoice can only have one employee.

**CE\_STORES** contain information about store names and locations.

This table has 3 connections: CE\_ADDRESSES, CE\_INVOICES, CE\_WAREHOUSES.

CE\_ADRESSES is connecting by one-to-one relationship. So, one store can have one address and one address can have only one store.

CE\_INVOICES has one-to-many relationship, as one store can have many invoices, but one invoice can only include one store in it.

CE\_WAREHOUSES has one-to-many relationship, as one store can only one warehouse, but one warehouse can be for several stores and online shopping.

**CE\_GENDERS** identifies gender of the employee and customer. It is connected with CE\_EMPLOYEES and CE\_CUSTOMERS with one-to-many relationship, as one gender can include many people and one person can have only one gender.

**CE\_WAREHOUSES** is a table about warehouses and collect information like warehouse name and location.

There are 3 connections within the table: CE\_ADDRESSES, CE\_STORES.

CE\_ADDRESSES is connected like one warehouse can have only one address and one address can have only one warehouse. So it is one-to-one relationship.

CE\_STORES connects like, one store can have one warehouse and one warehouse can have several store. So connection type is many-to-one.

**CE\_ADDRESSES** this is table with addresses and zip codes on it. It connect previously mentioned tables with many-to-one relationship and also connects to CE\_CITIES table.

**CE\_CITIES** table contains information about cities and is connected with CE\_ADDRESSES with many-to-one relationship. As one city can have many addresses, but one address can have only one city.

**CE\_PRODUCTS** include information about products, its type and specifications. It is connected with CE\_PRODUCT\_CATEGORIES and CE\_SALES.

One product can have only one category but one category can include many products in it. So that’s why the relationship type is one-to-many.

**CE\_PRODUCT\_CATEGORIES** include information about categories of products and connect with products table.

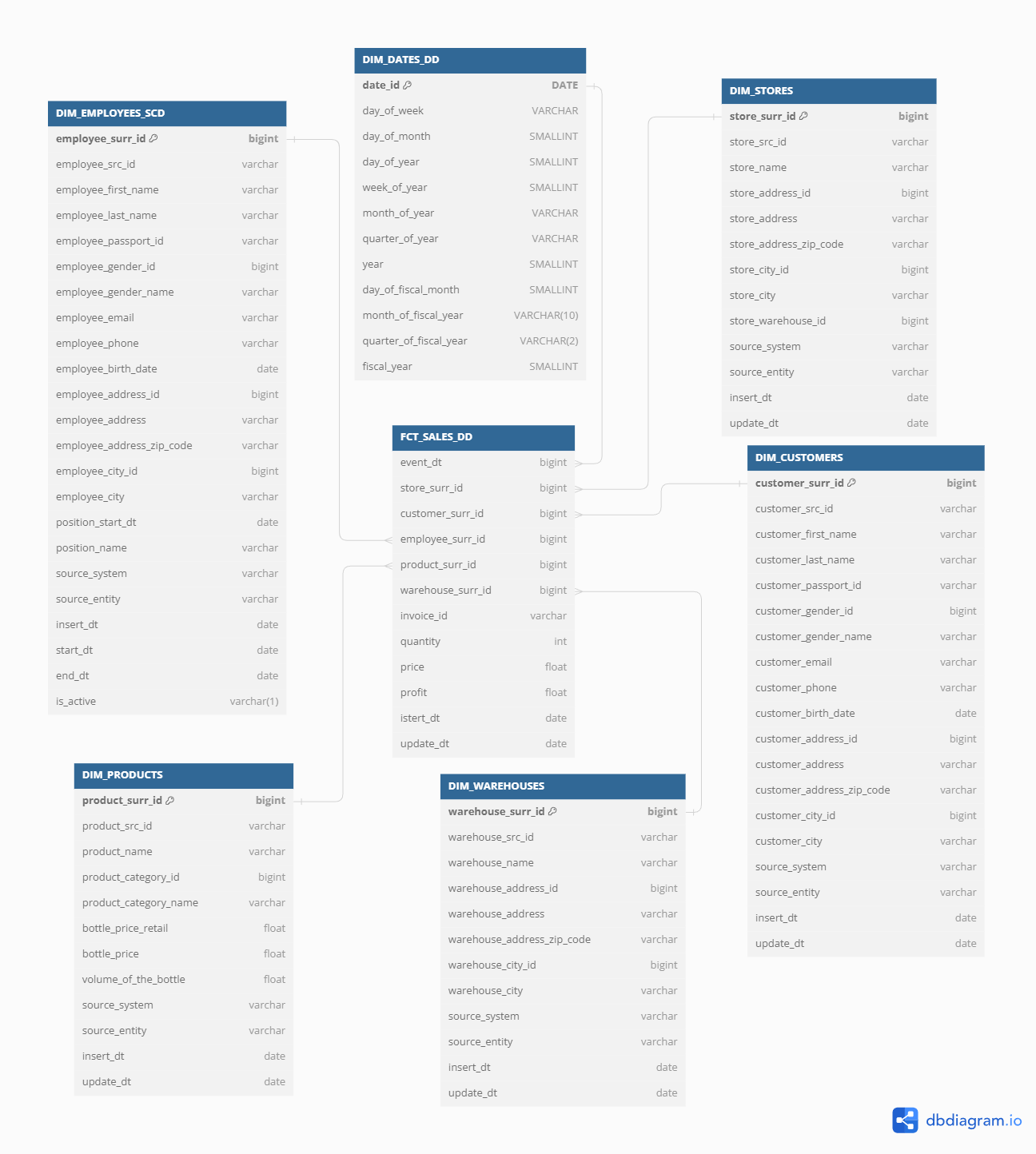
**CE\_DATES** include information about dates like number of day in a week, quarter number, information about holidays and weekdays.

**CE\_SALES** is table which include all the information about sales like customers, employees, invoices, stores, quantities and prices. It is connected almost all the previously mentioned tables. So we can analyze sales in a broad view.

In the CE\_SALES table there is a column named **invoice\_id,** which is Degenerate Dimension. This column itself has a value for our fact table, but there is no other attributes for it. So we inserted it in the fact table only.

# Business Layer Dimensional Model

Here is our dimensional model.



We have 6 dimensions and 1 fact.

All the dimensions are connected with fact table with many-to-one relationship, as each row in fact table is unique.

In the fact table I have price which it defined by multiplying quantity on bottle\_price\_retail from DIM\_PRODUCTS.

Also, there is calculated column named profit which is our profit per row. So we calculate it like: (DIM\_PRODUCTS.bottle\_price\_retail – DIM\_PRODUCTS.bottle\_price) \* FCT\_SALES.quantity.

Each dimension table has its own purpose.

So, DIM\_DATES give us chance to analyze our sales by time periods, like how was our sales changed in the specific time period.

DIM\_WAREHOUSES give us chance to decide which warehouse has more traffic, so we can make a bigger or smaller stock there.

DIM\_EMPLOYEES\_SCD will give us chance to analyze employees, their positions, their working experience and etc.

DIM\_CUSTOMERS will help us to control our customer info, give us knowledge about their age segments, their geographical data and etc.

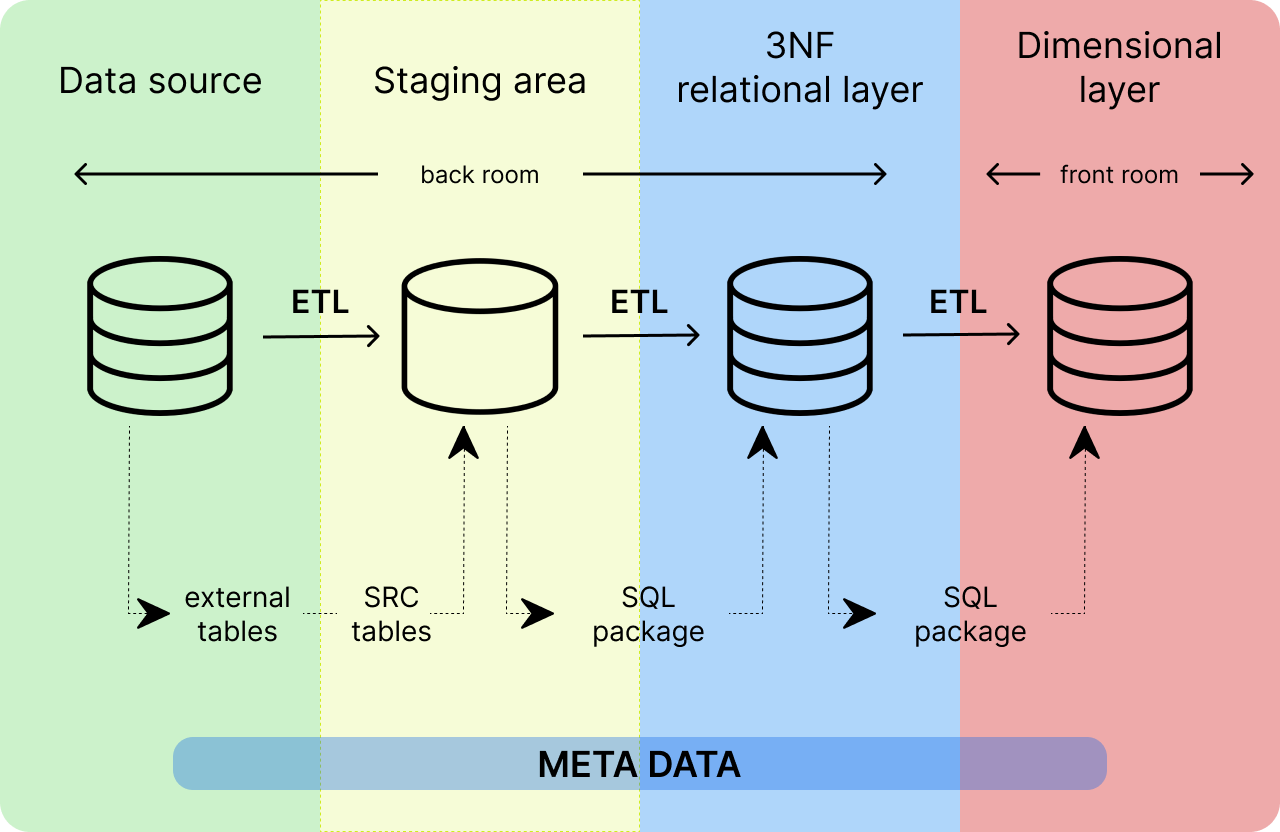
DIM\_STORES will give us a clue about sales by stores, for example.

DIM\_PRODUCTS will be one of the usable dimension of our data. It will help us to see which products are in demand more.

Our fact table FCT\_SALES\_DD include some metrics which I mentioned earlier and give us possibility to speak on the end of the facts.

Thus, all of this dimensions will give us chance to analyze our historical data, make some predictions and make our business better.

# Logical Scheme



Data Warehouse (DWH) load process:

1. Data Source: This is the initial stage of DWH load process. Data is extracted from two sources which contains data of company divided as online and offline sales.

2. Staging Area: The data from the sources is then loaded into the staging area. The staging area is an intermediate layer where the data is prepared for loading into the 3NF and dimensional layer.

3. 3NF Relational Layer: After the data is transformed in the staging area, it is loaded into the 3NF layer. This layer represents a normalized view of the data, where each table represents a unique entity. The 3NF layer reduces data redundancy and improves data integrity.

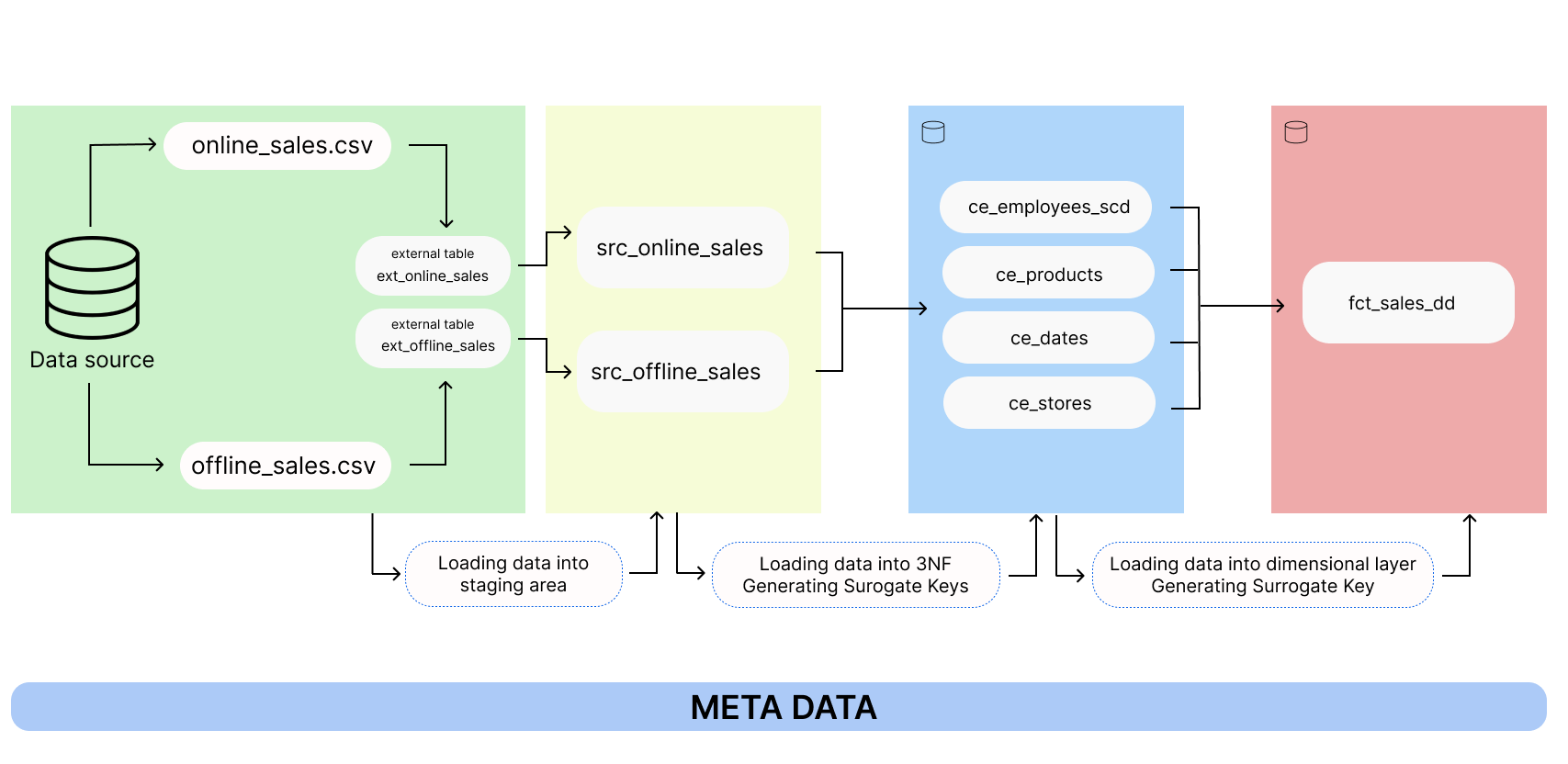
4. Dimensional Layer: Finally, the data is loaded into the dimensional layer. The dimensional layer is a denormalized view of the data that is optimized for querying and reporting. This layer contains fact tables that are surrounded by dimension tables.

# Data Flow

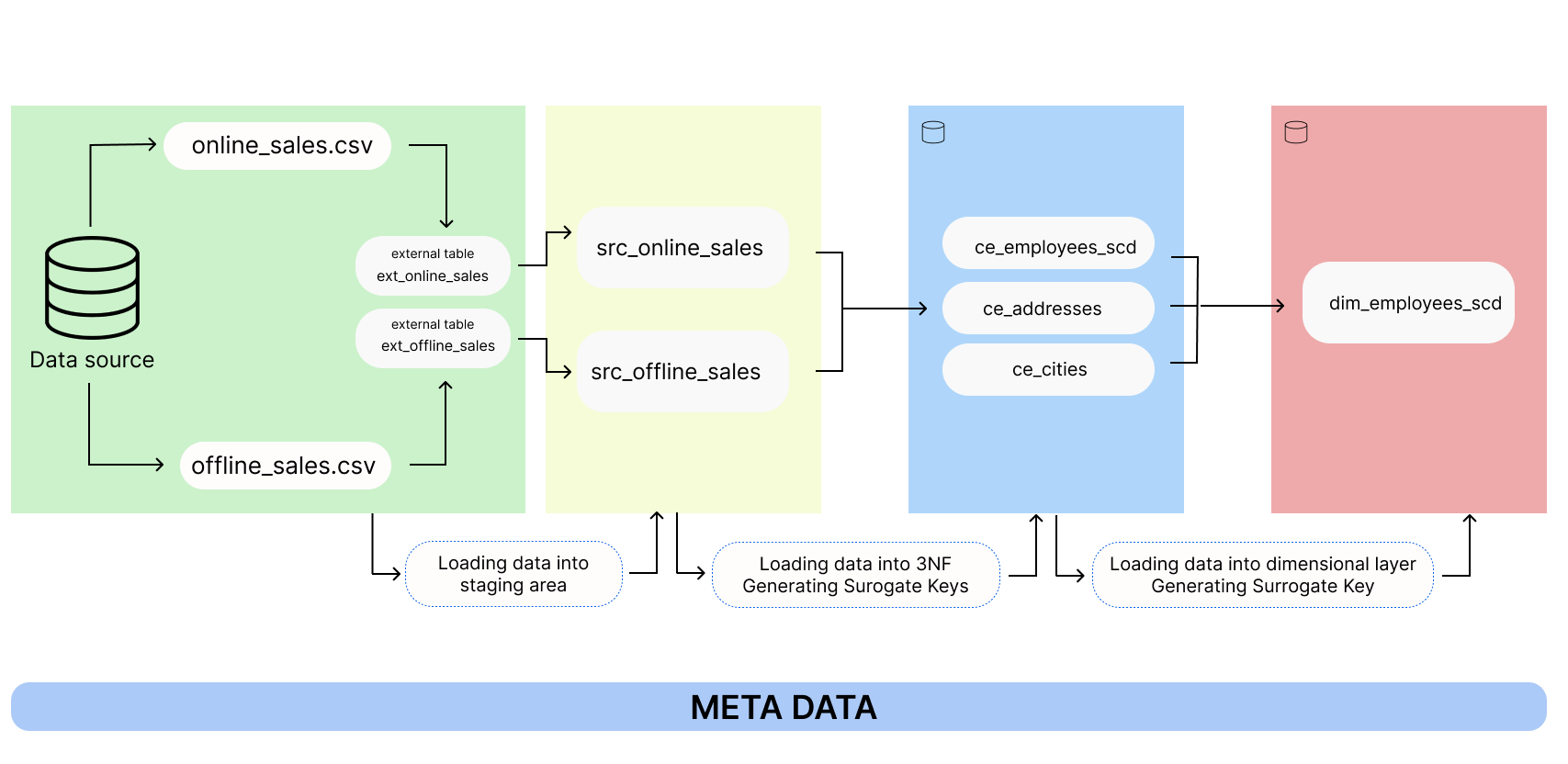
Data flow for DWH:

First we have two data sources, which are linked as external (ext) tables. Then these external tables are materialized as source (src) tables. After that data is loaded into 3NF, where each table contains only unique data. The last stage is loading data into dimensional layer, which is a denormalized view of the 3NF layer.

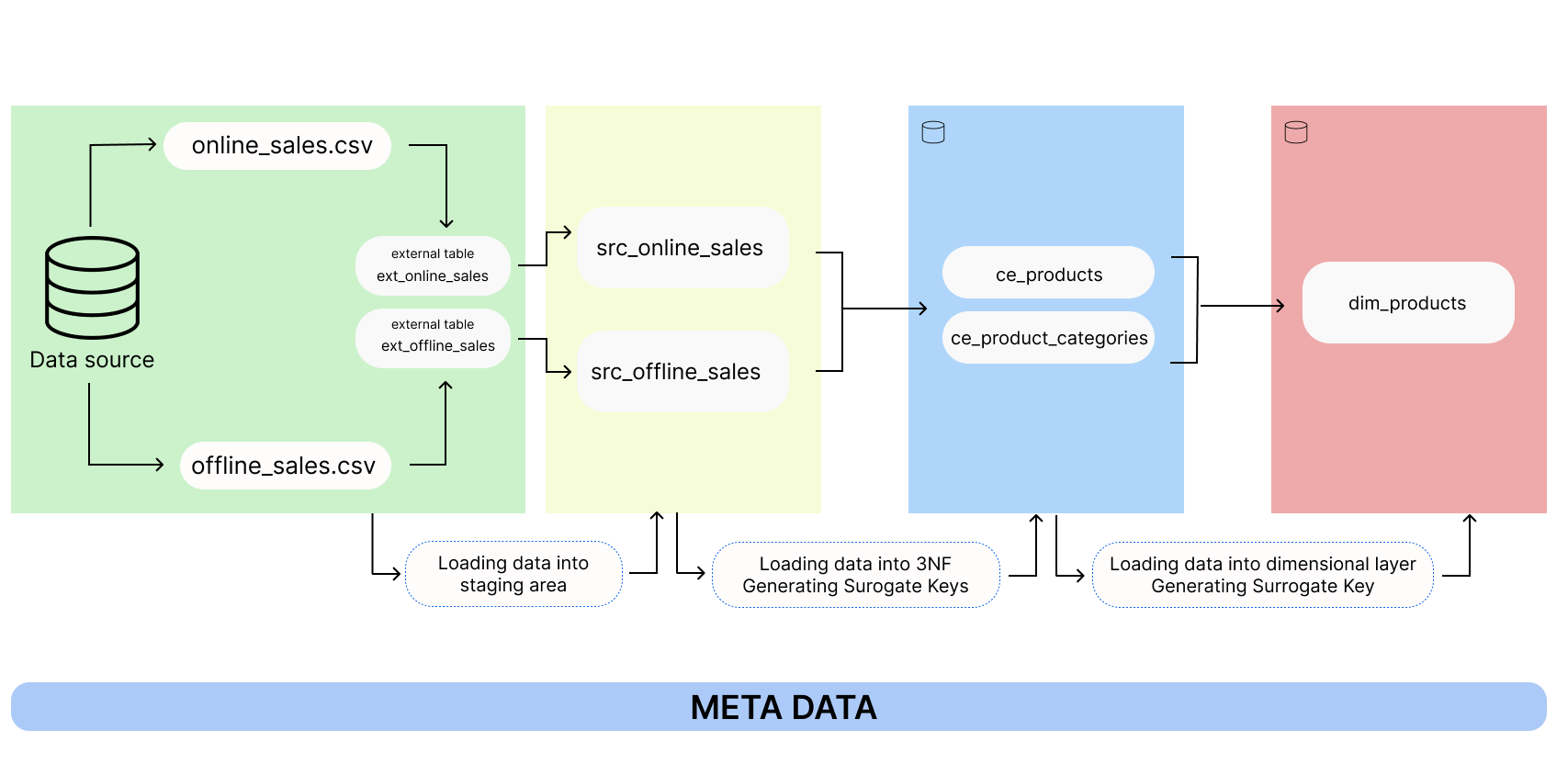
Below there are data flow for each table:



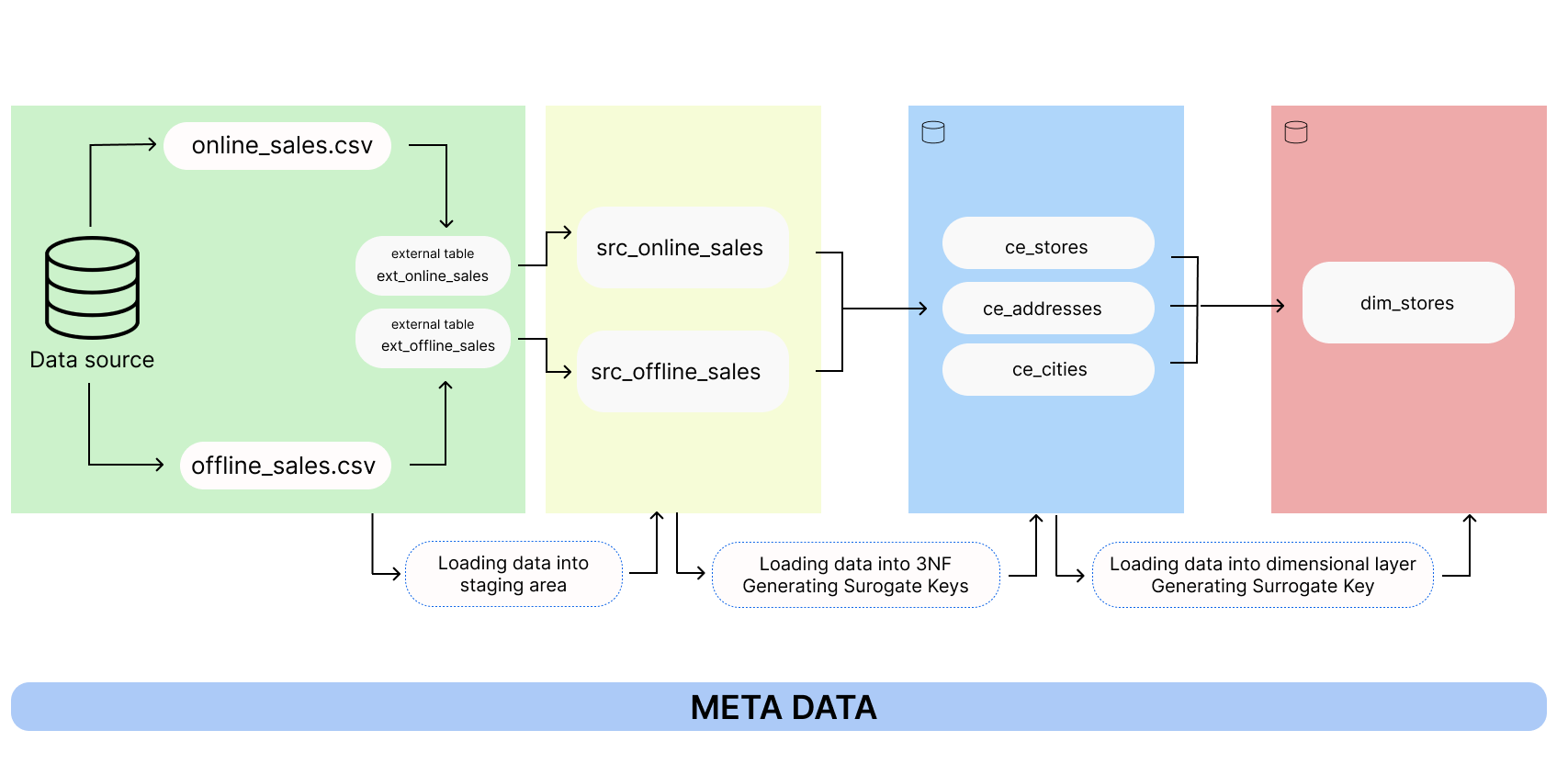
fact\_sales\_dd.



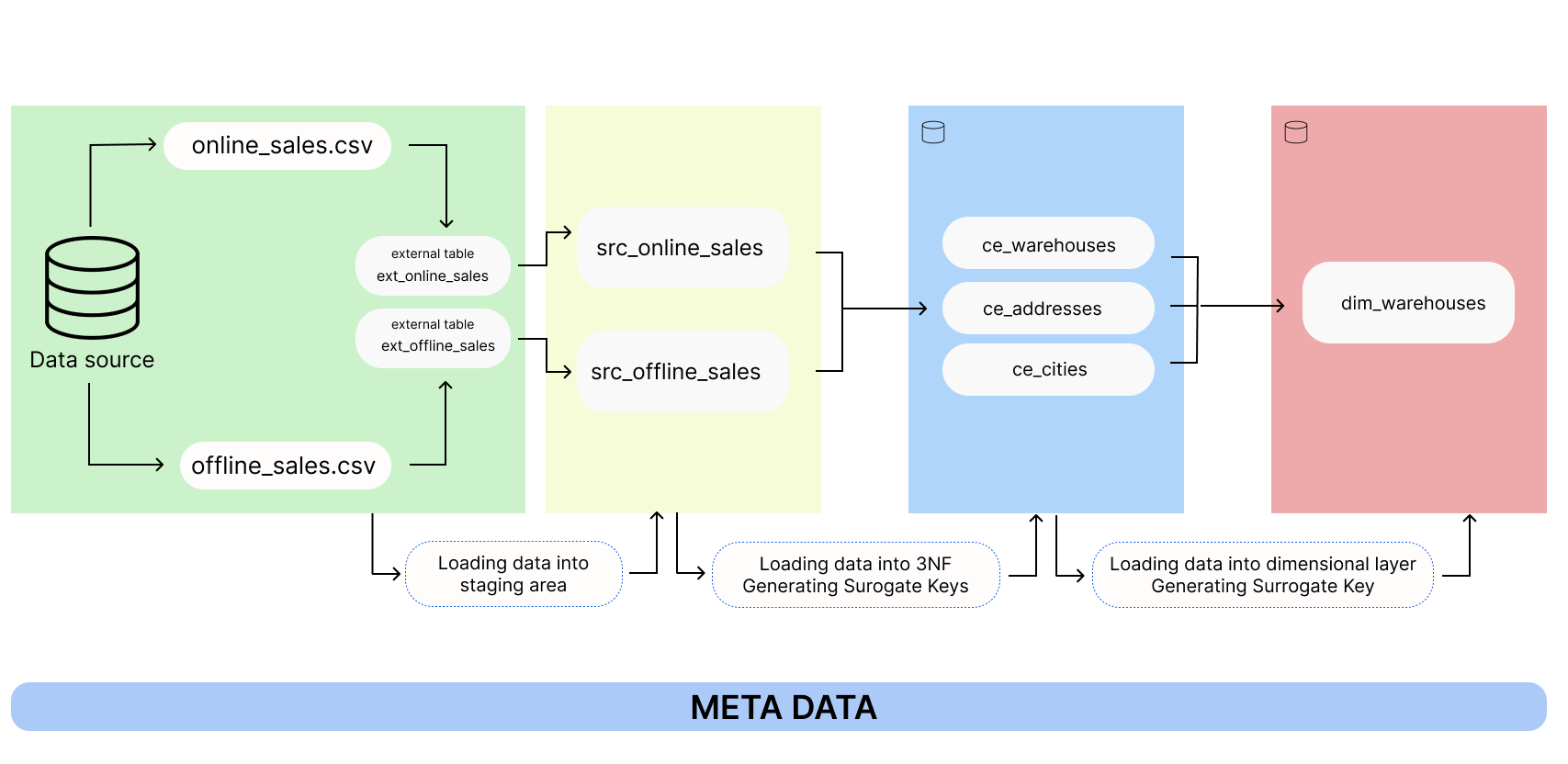
Dim\_employees\_scd



Dim\_products



Dim\_stores

dim\_warehouses

# Fact Table Partitioning Strategy