2



|  |
| --- |
| Business Template |
| **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

In the dynamic world of retail, our business operates at the intersection of online and offline sales channels, offering a diverse range of products to meet customer needs. The modern retail landscape is complex and competitive, requiring sophisticated data management and analytical capabilities to understand customer behavior, optimize sales strategies, and make informed business decisions.

## Problems because of poor data management

Traditional retail businesses often struggle with fragmented data sources, making it difficult to gain comprehensive insights. Without a robust data management system, retailers face significant challenges:

* Inability to track sales performance across different channels
* Limited understanding of customer preferences and buying patterns
* Difficulty in identifying cross-selling and upselling opportunities
* Inefficient inventory management
* Challenges in personalizing customer experiences

## Benefits from implementing a Data Warehouse

A comprehensive Data Warehouse solution can transform these challenges into strategic advantages:

* Unified view of sales across online and offline channels
* Detailed customer segmentation and behavior analysis
* Real-time performance tracking
* Predictive analytics for inventory and sales forecasting
* Enhanced decision-making through comprehensive data insights

## DATASETS DESCRIPTION

Online Store Dataset: online store dataset captures digital transactions with rich, multidimensional data:

* Comprehensive sales tracking
* Detailed customer information
* Product categorization
* Geographic distribution
* Digital transaction specifics (delivery method, payment types)

Offline Store Dataset: Complementing the online data, offline store dataset provides:

* In-store sales transactions
* Local market insights
* Physical store performance metrics
* Alternative customer interaction points

Unique Features:

* Two distinct yet complementary sales channels
* Comprehensive tracking of 500,000 transactions per channel
* Historical data covering two years
* Multiple dimensional perspectives (customer, product, geographic, temporal)

Hierarchical Relationships:

* Geographic Hierarchy: Country → Region → State/Province → City
* Product Hierarchy: Category → Sub-Category → Product Name
* Time Hierarchy: Year → Quarter → Month → Week → Day

Fact Table:

* Name: FCT\_SALES\_DD
* Attributes:
  + SALE\_SURR\_ID (Primary Key)
  + CUSTOMER\_SURR\_ID (FK)
  + PRODUCT\_SURR\_ID (FK)
  + ORDER\_SURR\_ID (FK)
  + EMPLOYEE\_SURR\_ID (FK)
  + GEOGRAPHIC\_SURR\_ID (FK)
  + DATE\_SURR\_ID (FK)
  + SALES\_TOT\_AMT
  + QUANTITY\_CNT
  + DISCOUNT\_PCT
  + PROFIT\_TOT\_AMT

Dimension Tables:

1. Customer Dimension (DIM\_CUSTOMERS):
   * CUSTOMER\_SURR\_ID
   * CUSTOMER\_NAME
   * CUSTOMER\_SEGMENT\_NAME
   * COUNTRY\_NAME
   * CITY\_NAME
   * POSTAL\_CODE\_NO
2. Product Dimension (DIM\_PRODUCTS):
   * PRODUCT\_SURR\_ID
   * PRODUCT\_NAME
   * PRODUCT\_CATEGORY\_NAME
   * PRODUCT\_SUBCATEGORY\_NAME
   * PRODUCT\_DESC
3. Order Dimension (DIM\_ORDERS):
   * ORDER\_SURR\_ID
   * EVENT\_DT
   * SHIP\_DT
   * SHIP\_MODE\_NAME
   * ORDER\_TYPE\_NAME
4. Geographic Dimension (DIM\_GEOGRAPHY):
   * GEOGRAPHIC\_SURR\_ID
   * COUNTRY\_NAME
   * REGION\_NAME
   * STATE\_NAME
   * CITY\_NAME
   * POSTAL\_CODE\_NO
5. Employee Dimension (DIM\_EMPLOYEES):
   * EMPLOYEE\_SURR\_ID
   * EMPLOYEE\_NAME
   * EMPLOYEE\_SURNAME
   * DEPARTMENT\_NAME
   * ROLE\_NAME

Date Dimension (DIM\_TIME\_DAY):

* DATE\_SURR\_ID
* FULL\_DT
* YEAR\_NO
* QUARTER\_NO
* MONTH\_NO
* MONTH\_NAME
* WEEK\_NO
* DAY\_NO
* DAY\_NUMBER
* ISWEEKEND

## GRAIN / DIM / FACT

Using the four-step dimensional design process:

Step 1- Select the Business Process

The business process being measured is sales transactions, where the organization tracks details about product sales, customers, employees involved, order details, and geographic locations.  
This process captures key performance metrics, such as total sales amount, quantity sold,discount applied, and profit earned, along with descriptive context provided by the companion dimensions

Step 2- Declare the Grain

The grain of the fact table is defined as one row per sale transaction, capturing the most granular level of detail.

I think it is the grain because it ensures that every sale event is uniquely represented and also it supports detailed analysis by combining other dimension tables.

Step 3- Identify the Dimensions

Customer Dimension (DIM\_CUSTOMERS): Who made the purchase?

Product Dimension (DIM\_PRODUCTS): What product was sold?

Order Dimension (DIM\_ORDERS): What were the details of the order (e.g., shipping, date)?

Geography Dimension (DIM\_GEOGRAPHY): Where did the transaction occur?

Employee Dimension (DIM\_EMPLOYEES**)**: Who facilitated the sale?

Date Dimension (DIM\_TIME\_DAY): When did the transaction happen?

Step 4-Identify the facts

SALES\_TOT\_AMT: Total revenue from the sale.

QUANTITY\_CNT: Quantity of products sold.

DISCOUNT\_PCT: Discount applied to the sale.

PROFIT\_TOT\_AMT: Total profit generated from the sale.

Fact Table: FCT\_SALES\_DD

Description: Captures each unique sales transaction at the most granular level.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| SALE\_SURR\_ID | Unique identifier for each sale transaction | Int |
| CUSTOMER\_SURR\_ID | Foreign key to the customer dimension | int |
| PRODUCT\_SURR\_ID | Foreign key to the product dimension | int |
| ORDER\_SURR\_ID | Foreign key to the order dimension | Int |
| EMPLOYEE\_SURR\_ID | Foreign key to the employee dimension | int |
| GEOGRAPHIC\_SURR\_ID | Foreign key to the geographic dimension | Int |
| DATE\_SURR\_ID | Foreign key to the time dimension | Date |
| SALES\_TOT\_AMT | Total sales amount for the transaction | Numeric(12,2) |
| QUANTITY\_CNT | Quantity of products sold | Int |
| DISCOUNT\_PCT | Discount percentage applied | Numeric(5,2) |
| PROFIT\_TOT\_AMT | Total profit from the transaction | Numeric(12,2) |

Example data

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SALE\_SURR\_ID | CUSTOMER\_SURR\_ID | PRODUCT\_SURR\_ID | ORDER\_SURR\_ID | EMPLOYEE\_SURR\_ID | GEOGRAPHIC\_SURR\_ID | DATE\_SURR\_ID | SALES\_TOT\_AMT | QUANTITY\_CNT | DISCOUNT\_PCT | PROFIT\_TOT\_AMT |
| 1 | 101 | 201 | 301 | 401 | 501 | 601 | 1500 | 3 | 10% | 300 |

Dimension Table: DIM\_CUSTOMERS

Description: Contains customer details, such as name, segment, and location.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| CUSTOMER\_SURR\_ID | Unique identifier for each customer | Int |
| CUSTOMER\_NAME | Full name of the customer | Text |
| CUSTOMER\_SEGMENT\_NAME | Segment the customer belongs to | Text |
| COUNTRY\_NAME | Country of the customer | Text |
| CITY\_NAME | City of the customer | Text |
| POSTAL\_CODE\_NO | Customer's postal code | Text |

Example data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CUSTOMER\_SURR\_ID | CUSTOMER\_NAME | CUSTOMER\_SEGMENT\_NAME | COUNTY\_NAME | CITY\_NAME | POSTAL\_CODE\_NO |
| 101 | John doe | Corporate | USA | New-york | 10001 |

Dimension Table: DIM\_PRODUCTS

Description: Contains details about the products sold.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| PRODUCT\_SURR\_ID | Unique identifier for each product | Int |
| PRODUCT\_NAME | Name of the product | Text |
| PRODUCT\_CATEGORY\_NAME | Category of the product | Text |
| PRODUCT\_SUBCATEGORY\_NAME | Subcategory of the product | |  | | --- | |  |  |  | | --- | | Text | |
| PRODUCT\_DESC | Description of the product | Text |

Example data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRODUCT\_SURR\_ID | PRODUCT\_NAME | PRODUCT\_CATEGORY\_NAME | PRODUCT\_SUBCATEGORY\_NAME | PRODUCT\_DESC |
| 201 | Laptop | Electronics | Computers | High-performance laptop |

Dimension Table: DIM\_ORDERS

Description: Contains information about customer orders.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| ORDER\_SURR\_ID | Unique identifier for each order | Int |
| EVENT\_DT | Date the order was placed | Date |
| SHIP\_DT | Date the order was shipped | Date |
| SHIP\_MODE\_NAME | Shipping mode (e.g., Standard, Express) | Text |
| ORDER\_TYPE\_NAME | Type of order (e.g., Online, In-Store) | Text |

Example data:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ORDER\_SURR\_ID | EVENT\_DT | |  | | --- | |  |  |  | | --- | | **SHIP\_DT** | | SHIP\_MODE\_NAME | ORDER\_TYPE\_NAME |
| 301 | 2025-01-20 | 2025-01-25 | Standard | Online |

Dimension Table: DIM\_GEOGRAPHY

Description: Captures geographic information related to sales transactions.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| GEOGRAPHIC\_SURR\_ID | Unique identifier for geographic data | Int |
| COUNTRY\_NAME | Country of the sale | Text |
| REGION\_NAME | Region of the sale | Text |
| STATE\_NAME | State of the sale | Text |
| CITY\_NAME | City of the sale | Text |
| POSTAL\_CODE\_NO | Postal code of the sale | Text |

Example data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GEOGRAPHIC\_SURR\_ID | COUNTRY\_NAME | REGION\_NAME | STATE\_NAME | CITY\_NAME | POSTAL\_CODE\_NO |
| 501 | USA | East coast | NY | New York | 10001 |

Dimension Table: DIM\_EMPLOYEES

Description: Contains information about employees involved in sales transactions.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| EMPLOYEE\_SURR\_ID | Unique identifier for each employee | Int |
| EMPLOYEE\_NAME | First name of the employee | Text |
| EMPLOYEE\_SURNAME | Last name of the employee | Text |
| DEPARTMENT\_NAME | Department the employee belongs to | Text |
| ROLE\_NAME | Role of the employee | Text |

Example data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_SURR\_ID | EMPLOYEE\_NAME | EMPLOYEE\_SURNAME | DEPARTMENT\_NAME | ROLE\_NAME |
| 401 | Alice | Johnson | Sales | Manager |

Dimension Table: DIM\_TIME\_DAY

Description: Captures time-related details.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| DATE\_SURR\_ID | Unique identifier for each date | Date |
| FULL\_DT | Full date (YYYY-MM-DD) | Date |
| YEAR\_NO | Year of the date | Int |
| QUARTER\_NO | Quarter of the year (1-4) | Int |
| MONTH\_NO | Month of the year (1-12) | Int |
| WEEK\_NO | Week number of the year | Int |
| DAY\_NO | Day of the month | Int |

Example data:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATE\_SURR\_ID | FULL\_DT | YEAR\_NO | QUARTER\_NO | MONTH\_NO | WEEK\_NO | DAY\_NO | DAY\_NAME | ISWEEKEND |
| 601 | 2025-01-20 | 2025 | 1 | 1 | 3 | 20 | Monday | No |

# Business Layer 3NF

1. Reiterate the core concept of Slowly Changing Dimensions (SCD).

SCDs are dimension tables that handle historical changes in their attribute values over time. There are several types:

* Type 0: Original values never change
* Type 1: Overwrite the old value with the new value (no history kept)
* Type 2: Add a new row with the changed values (preserves history)
* Type 3: Add new columns to track old and new values

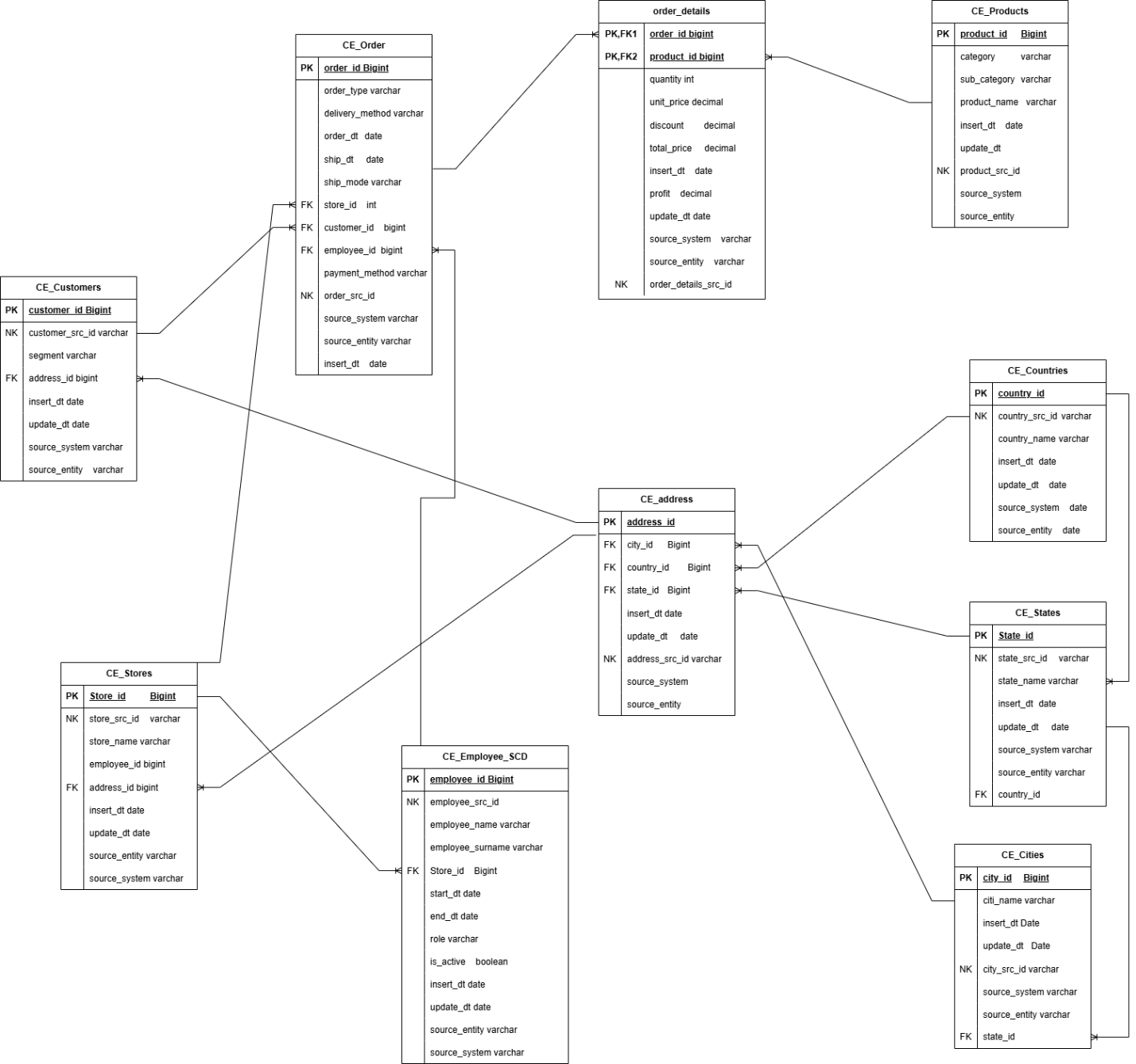
2.

I created 3NF schema based on my dataset. I keep everything what they have in common but I added some entities also. At first for offline dataset we need information about employees and also store where this transaction happened. So I added them in the schema, everything other is unchanged. I thought employee entity would be good example for scd2, since employee can change its role we can see what was his previous role in the store before becoming for example “manager” and for that I added start\_date, end\_date and role to the employee table. there start\_date and employee\_ID is the composite primary key, just like it works for scd2.

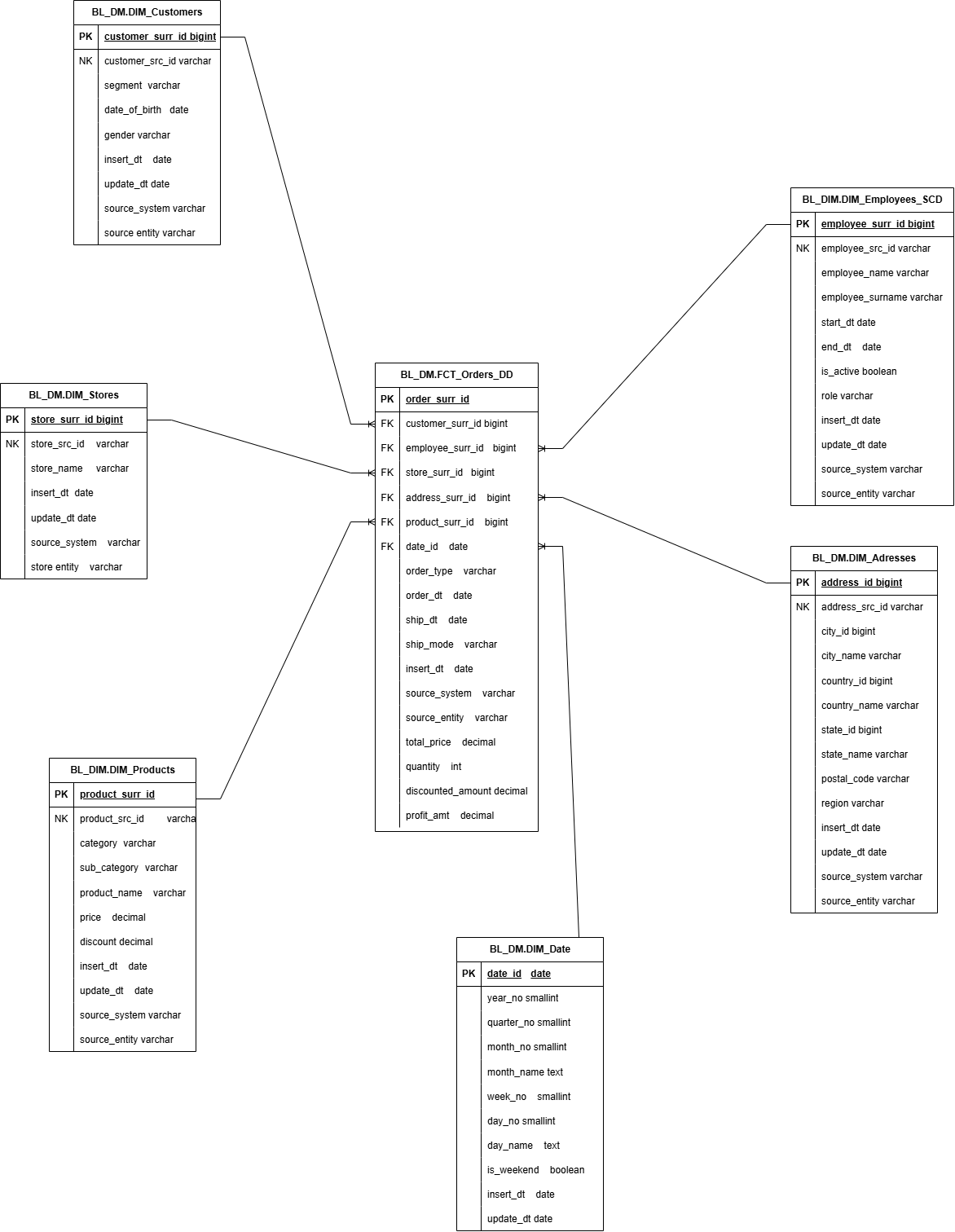
Now I will talk about other entities. Hard part for me was to handle address without violating 3NF rules. So because of that I created 4 entities: country, city, address and state and I established some relationship between them. 1-N relationship between country and city and same goes for country and state. and at last I established connection between city and address entities. Now I will continue with product entity. In the table I have category and sub-category attributes and I thought if that violates 3NF, I left it because I think 1 category can have many sub-categories and some categories may have same sub-categories so there will be no dependency between them, but I’m not sure if my logic is correct. Also I had problem with products table because I didn’t know how to connect them to other tables logically, and I came up with idea that it would be good to connect it with orders table, because with order\_details table we can get information if the purchased item had any discount or how many items customer bought and also the total price is saved there.

We also have payments table and it has 1-N relationship with two other entities customer and employee because 1 customer can have many payment records and one payment record is for exactly one customer. Same logic goes for the employee also. I think everything else is clear for the schema.

(update) I removed address\_id from employee table because I think employees address gives us nothing and changed relationship between address, cities, countries and states. I combined order and payments table and also I want to talk about orders->order\_details->products. Order details act as bridge table between those two and it’s nothing more, at first schema I made a mistake. I think this schema should be now correct. And also in first schema I also had src\_ids but dbdiagram doesn’t marked them as NK



# Business Layer Dimensional Model



Metrics in FCT\_ORDERS\_PAYMENTS:

1. TOTAL\_PRICE

* Formula: QUANTITY \* UNIT\_PRICE
* Description: Gross order amount before discounts

2.DISCOUNTED\_AMOUNT

* Formula: TOTAL\_PRICE \* DISCOUNT\_PERCENTAGE
* Description: Monetary value of discount applied

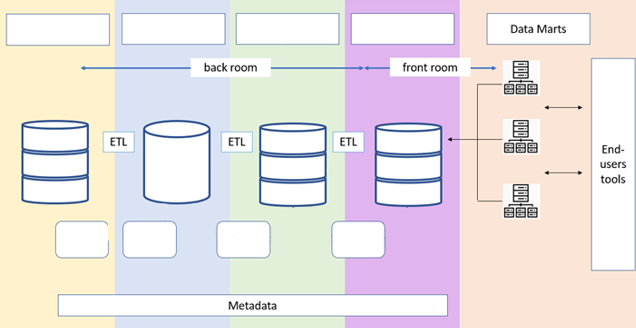
3.PROFIT\_AMT

* Formula: NET\_PRICE - COST\_PRICE
* Where NET\_PRICE = TOTAL\_PRICE - DISCOUNTED\_AMOUNT
* Description: Net profit after costs and discounts

Also I changed one thing in fact table. I removed paymentID because in the task\_3 I thought that if we have orderID already then we can get every information about the payment too, since one order is one payment and I see no need for paymentID and I changed both schemas, this one and past schema too.

I had reason why I added SCD2 in products table, since product price can change by time I thought it would be good to add SCD2 to see its price history but I’m not sure it is good idea or not.

# Logical Scheme



SQL transform

SQL transform

Source tables

External tables

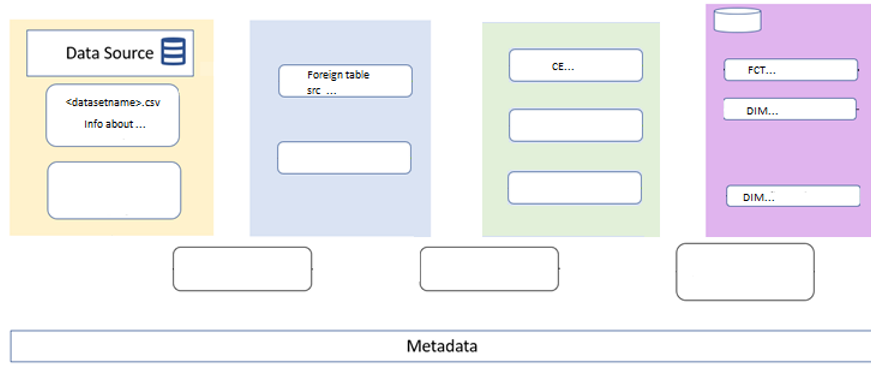
Dimensional layer

Relational layer

Staging area

Data sources

# Data Flow



Dim\_date

Dim\_employees\_scd

Dim\_products

Dim\_stores

Dim\_customers

FCT\_orders\_dd

Ce\_employees\_scd

Ce\_order\_details

Ce\_orders

Ce\_stores

Ce\_states

Ce\_products

Ce\_countries

Ce\_addresses

Ce\_cities

Ce\_customers

Src\_offline\_sales

Src\_online\_sales

offline\_sales\_with\_store\_id.csv

modified\_online\_sales\_with\_stores.csv

# Fact Table Partitioning Strategy