

| Business Template  **BIG STORE SALE DATA** |
| --- |

**Contents**

[1 Business Description 3](#_heading=h.1fob9te)

[1.1 Business background 3](#_heading=h.3znysh7)

[1.2 Problems because of poor data management 3](#_heading=h.2et92p0)

[1.3 Benefits from implementing a Data Warehouse 3](#_heading=h.tyjcwt)

[1.4 DATASETS DESCRIPTION 4](#_heading=h.3dy6vkm)

[1.5 GRAIN / DIM / FACT 6](#_heading=h.1t3h5sf)

[2 Business Layer 3NF 10](#_heading=h.4d34og8)

[3 Business Layer Dimensional Model 11](#_heading=h.2s8eyo1)

[4 Logical Scheme 11](#_heading=h.17dp8vu)

[5 Data Flow 11](#_heading=h.3rdcrjn)

[6 Fact Table Partitioning Strategy 11](#_heading=h.26in1rg)

# 

# Business Description

## Business background

In today’s world, businesses rely on data to make important decisions. Whether it's a retailer, manufacturer, or service provider, companies use data to understand their customers, products, and operations. This helps them stay competitive and grow.

For example, a big store might sell products both online and in physical stores. They collect a lot of information—sales data, customer details, stock levels, and supplier information. If this data isn’t managed well, it becomes difficult to understand what’s going on. That’s why many companies use tools like data warehouses to bring all their data together in one place for easier analysis.

## Problems because of poor data management

When businesses don’t manage their data properly, they face a lot of problems:

* Disorganized Data:

Data is stored in different places (e.g., separate systems for online and offline sales), making it hard to see the full picture.

* Bad Decisions:

If data is incomplete or incorrect, reports and decisions based on it will also be wrong. For example, it might be hard to figure out which products sell best or which customers to focus on.

* Wasted Time and Money:

When data isn’t organized, employees spend too much time trying to find and combine information manually. This costs the company money and slows down decision-making.

* Missed Opportunities:

Without quick and accurate data, businesses can’t react to trends or customer needs in time. They may lose customers to competitors.

* Stock Problems:

Poor inventory management can lead to overstocking (wasting money) or stockouts (angry customers), both of which hurt the business.

In short, bad data management leads to mistakes, wasted resources, and lost business opportunities.

## Benefits from implementing a Data Warehouse

A data warehouse solves many of these problems by organizing all the data in one place. Here’s how it helps:

* Everything in One Place:

A data warehouse combines data from different sources, like online and offline sales, so businesses can access everything in one system.

* Better Decisions:

Clean and well-organized data makes it easier to see trends, such as which products sell the most or which customers spend the most.

* Time and Cost Savings:

With automated data collection and reporting, employees can focus on solving problems instead of wasting time on manual tasks.

* Accurate Reports to track KPI:

A data warehouse ensures that data is consistent and correct, so businesses can trust their reports and analysis.

* Spot Trends and Plan Ahead:

Businesses can analyze patterns, like seasonal sales or customer behavior, and use this information to plan for the future.

* Stay Ahead of Competitors:

With quick access to insights, businesses can adapt faster, improve customer service, and make smarter choices to beat their competition.

In summary, a data warehouse helps businesses get the most out of their data. It saves time, reduces mistakes, and **gives companies the tools to make better decisions and grow faster**.

## DATASETS DESCRIPTION

The online database primarily captures data related to e-commerce transactions, focusing on customer and product interactions within an online store environment.

**Transactions Information**

transaction\_id: Unique identifier for each transaction.

transaction\_date: Date and time of the transaction.

store\_id: Identifier for the online store.

store: Name of the online store.

store\_location: Generalized location associated with the online store

**Products Information**

product\_id: Unique identifier for each product.

product\_name: Name of the product.

category: Category to which the product belongs.

quantity\_sold: Number of units sold in the transaction.

unit\_price: Price per unit at the time of sale.

**Customer Information**

customer\_id: Unique identifier for each customer.

**Promotions**

promotion\_applied: Whether a promotion was applied to the transaction.

**Suppliers**

supplier\_id: Unique identifier for the supplier providing the product.

Offline Database

t\_id: Unique identifier for each transaction.

transaction\_date: Date of the transaction.

time: time of the transaction.

store\_id: Identifier for the store where the transaction occurred.

store\_location: Physical location of the store.

**Products Information**

product\_id: Unique identifier for each product.

product\_name: Name of the product.

category: Category to which the product belongs.

quantity\_sold: Number of units sold in the transaction.

unit\_price: Price per unit at the time of sale.

**Customer Information**

customer\_id: Unique identifier for each customer.

customer\_gender: Gender of the customer.

customer\_income: Income level of the customer.

**Suppliers**

supplier\_id: Unique identifier for the supplier providing the product.

supplier\_lead\_time: Time taken by the supplier to deliver products.

The online database focuses on capturing data from e-commerce transactions, while the offline database deals with sales data from physical stores. These two databases are designed to manage different types of operations but share some similarities and key differences.

The online database contains data relevant to digital transactions, such as customer activity, product sales, and promotions used in online stores. It often has fewer details about customers and operational factors like inventory or weather but is focused on tracking customer loyalty, product categories, and sales performance in a digital environment.

In contrast, the offline database provides more comprehensive information about in-store transactions, including customer demographics (e.g., gender, income), inventory levels, supplier performance, and external factors such as weather or holidays. This allows businesses to analyze the impact of these factors on sales and plan inventory or promotions accordingly.

## GRAIN / DIM / FACT

The grain of my fact table is at the transaction level. This means each row in the fact table shows one transaction, including details like what was sold, how much, and to whom:

TRANSACTION\_ID: a unique ID for each transaction.

QUANTITY\_SOLD: how many products were sold in this transaction.

UNIT\_PRICE: the price of the unit product

PRODUCT\_ID: links to the product that was sold.

CUSTOMER\_ID: links to the customer who made the purchase.

STORE\_ID: links to the store where the transaction happened.

DATE\_ID: links to the date of the transaction.

PROMOTION\_ID: links to any promotion used during the transaction.

Dimensions

TIME TABLE (DIMENSION)

DATE\_ID: a unique ID for each date (WHICH WILL BE ADDED LATER))

YEAR: the year of the transaction.

MONTH: the month of the transaction.

DAY: the specific day of the transaction.

PRODUCT TABLE (DIMENSION)

PRODUCT\_ID: a unique ID for each product.

PRODUCT\_NAME: the name of the product.

CATEGORY: the category the product belongs to (like Electronics, Appliances).

SUPPLIER\_ID: links to the supplier of the product.

UNIT\_PRICE: the unit price of the product.

CUSTOMER TABLE (DIMENSION)

CUSTOMER\_ID: a unique ID for each customer.

CUSTOMER\_LEVEL: the loyalty level of the customer (like Gold, Silver).

GENDER: the gender of the customer.

CUSTOMER\_INCOME: the income of the customer.

STORE TABLE (DIMENSION)

STORE\_ID: a unique ID for each store.

STORE\_LOCATION: the location of the store.

STATE: the state where the store is located.

PROMOTION TABLE (DIMENSION)

PROMOTION\_ID: a unique ID for each promotion.

PROMOTION\_APPLIED: tells if a promotion was used.

Fact Table: Sales

| **Attribute** | **Data Type** | **Description** |
| --- | --- | --- |
| TRANSACTION\_ID | INT (PK) | Unique identifier for each transaction |
| QUANTITY\_SOLD | INT | Number of items sold in the transaction |
| UNIT\_PRICE | DECIMAL(10, 2) | Price per unit sold |
| PRODUCT\_ID | INT (FK) | Links to Product table |
| CUSTOMER\_ID | INT (FK) | Links to Customer table |
| STORE\_ID | INT (FK) | Links to Store table |
| DATE\_ID | INT (FK) | Links to Date table |
| PROMOTION\_ID | INT (FK) | Links to Promotion table |

Customer

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| customer\_id | INT (PK) | Unique identifier for each customer |
| customer\_name | VARCHAR(50) | Name of the customer |
| customer\_level | VARCHAR(20) | Loyalty level of the customer |
| gender | VARCHAR(10) | Gender of the customer |
| customer\_income | DECIMAL(15,2) | Annual income of the customer |

Address

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| address\_id | INT (PK) | Unique identifier for the address |
| state | VARCHAR(20) | State of the address |

Date

| Column Name | Data Type | Description |
| --- | --- | --- |
| date\_id | INT (PK) | Unique identifier for each date |
| transaction\_date | DATE | Full date of the transaction |

Product

| Column Name | Data Type | Description |
| --- | --- | --- |
| product\_id | INT (Pk) | Unique identifier for each product |
| product\_name | VARCHAR(50) | Name of the product |
| supplier\_id | INT (FK) | Links to the supplier information |
| unit\_price | DECIMAL(10,2) | Standard price of the product |

Category

| Column Name | Data Type | Description |
| --- | --- | --- |
| category\_id | INT (Primary Key) | Unique identifier for each category |
| category\_name | VARCHAR(50) | Name of the category |

Store

| Column Name | Data Type | Description |
| --- | --- | --- |
| store\_id | INT (Primary Key) | Unique identifier for each store |
| store\_location | VARCHAR(50) | Location of the store |
| state | VARCHAR(20) | State where the store is located |

Promotion

| Column Name | Data Type | Description |
| --- | --- | --- |
| promotion\_id | INT (Primary Key) | Unique identifier for each promotion |
| promotion\_applied | VARCHAR(10) | Whether a promotion was applied |

This grain lets me see detailed information about each transaction. I can know what products were sold, how many, at what price, and who bought them and what income they have

Benefits of the Grain

Detailed Insights: I can see every transaction and understand customer behavior.

Inventory Management: Helps me keep track of product demand and stock.

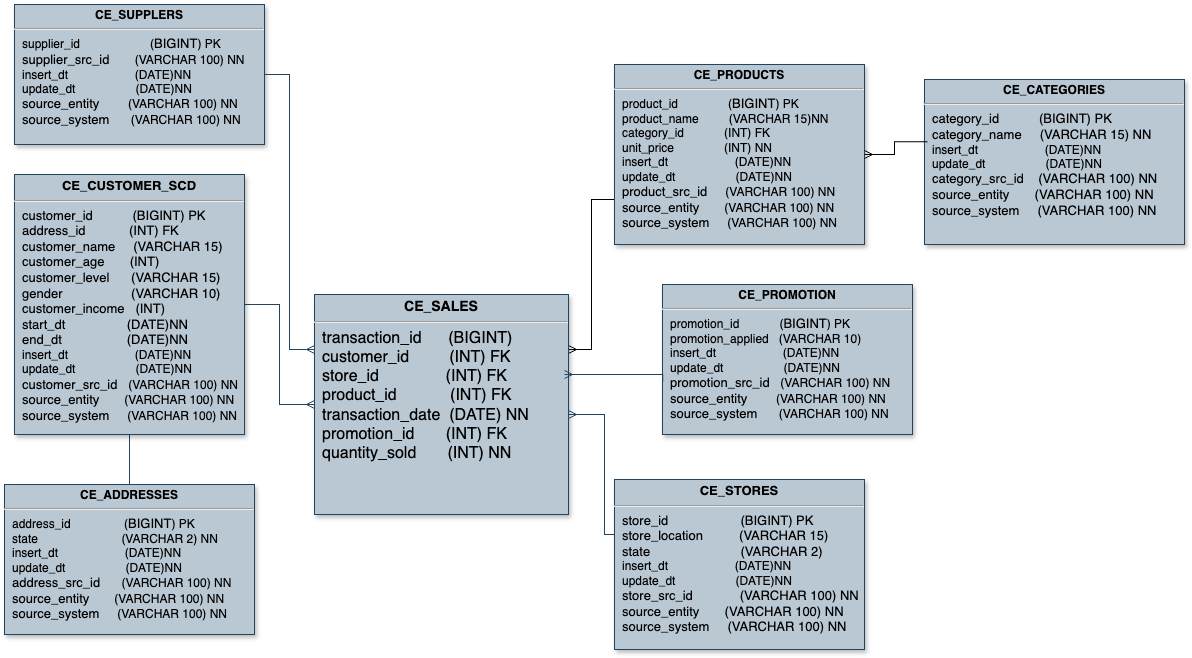
Sales Forecasting: I can look at trends and predict future sales.

Better Marketing: The data helps create more targeted promotions and offers.

Customer Understanding: I can learn more about customer preferences and their demographics.

With this grain and dimensions, I can analyze data in detail and use it to make smart business decisions.

# Business Layer 3NF



The CE\_SALES table stores transactional data related to sales. It includes attributes such as transaction\_id (Primary Key), customer\_id (Foreign Key to CE\_CUSTOMER\_SCD), product\_id (Foreign Key to CE\_PRODUCTS), store\_id (Foreign Key to CE\_STORES), promotion\_id (Foreign Key to CE\_PROMOTIONS), quantity\_sold. This table is the core of the schema, capturing detailed sales transaction data.

SCD type 0 (sales transactions can’t be changed once recorded)

The CE\_CUSTOMER\_SCD table stores customer information. It includes attributes such as customer\_id (Primary Key), customer\_level,customer\_name, customer\_age,customer\_level,gender, customer\_income, and address\_id (Foreign Key to CE\_ADDRESSES)start\_dt, end\_dt, insert\_dt, update\_dt, customer\_src\_id, source\_entity, and source\_system. This table provides detailed information about customers involved in transactions.

SCD type 2 (add a new row when customer details change to keep history)

The CE\_PRODUCTS table stores product information. It includes attributes such as product\_id (Primary Key), product\_name, unit\_price, and category\_id (Foreign Key to CE\_CATEGORIES) start\_dt, end\_dt, insert\_dt, update\_dt, product\_src\_id, source\_entity, and source\_system. This table provides detailed information about products involved in transactions.

SCD type 1(overwrite old product details with new ones)

The CE\_PROMOTION table stores promotion information. It includes attributes such as promotion\_id (Primary Key) and promotion\_applied, start\_dt, end\_dt, insert\_dt, update\_dt, promotion\_src\_id, source\_entity, and source\_system This table provides details about promotions applied to transactions.

SCD type 0 (when promotion is used, it can’t be changed)

The CE\_ADDRESSES table stores address information. It includes attributes such as address\_id (Primary Key), state, start\_dt, end\_dt, insert\_dt, update\_dt, addresses\_src\_id, source\_entity, and source\_system. This table provides detailed address information for customers.

SCD type 2 (add a new row when an address changes to keep history)

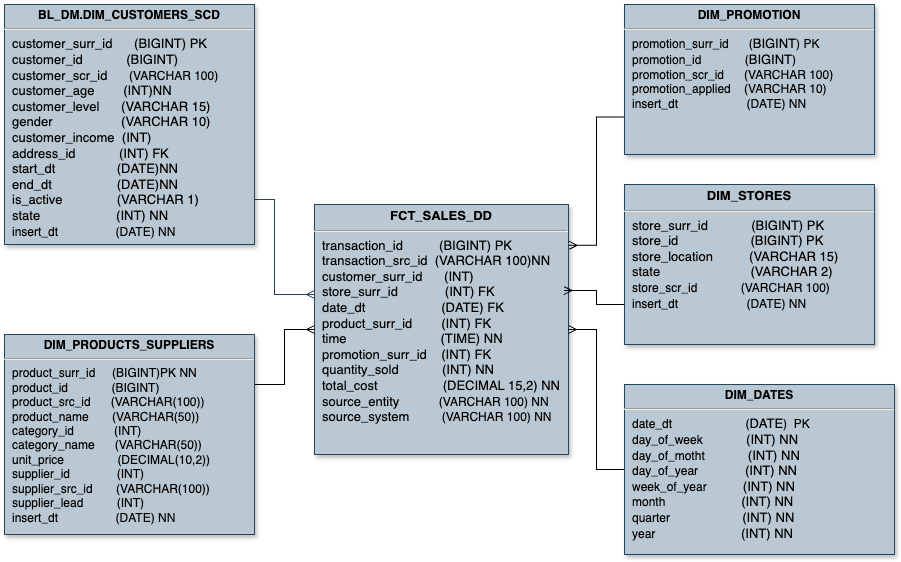
The CE\_CATEGORIES table stores product category information. It includes attributes such as category\_id (Primary Key) and category\_name, start\_dt, end\_dt, insert\_dt, update\_dt, categories\_src\_id, source\_entity, and source\_system. This table provides category details for products.

SCD type 1 (overwrite old category names with new ones)

The CE\_SUPPLIERS table stores supplier information. It includes attributes such as supplier\_id (Primary Key) and supplier\_name, start\_dt, end\_dt, insert\_dt, update\_dt, suppliers\_src\_id, source\_entity, and source\_system. This table provides details about suppliers of products.

SCD type 1 (overwrite old supplier details with new ones)

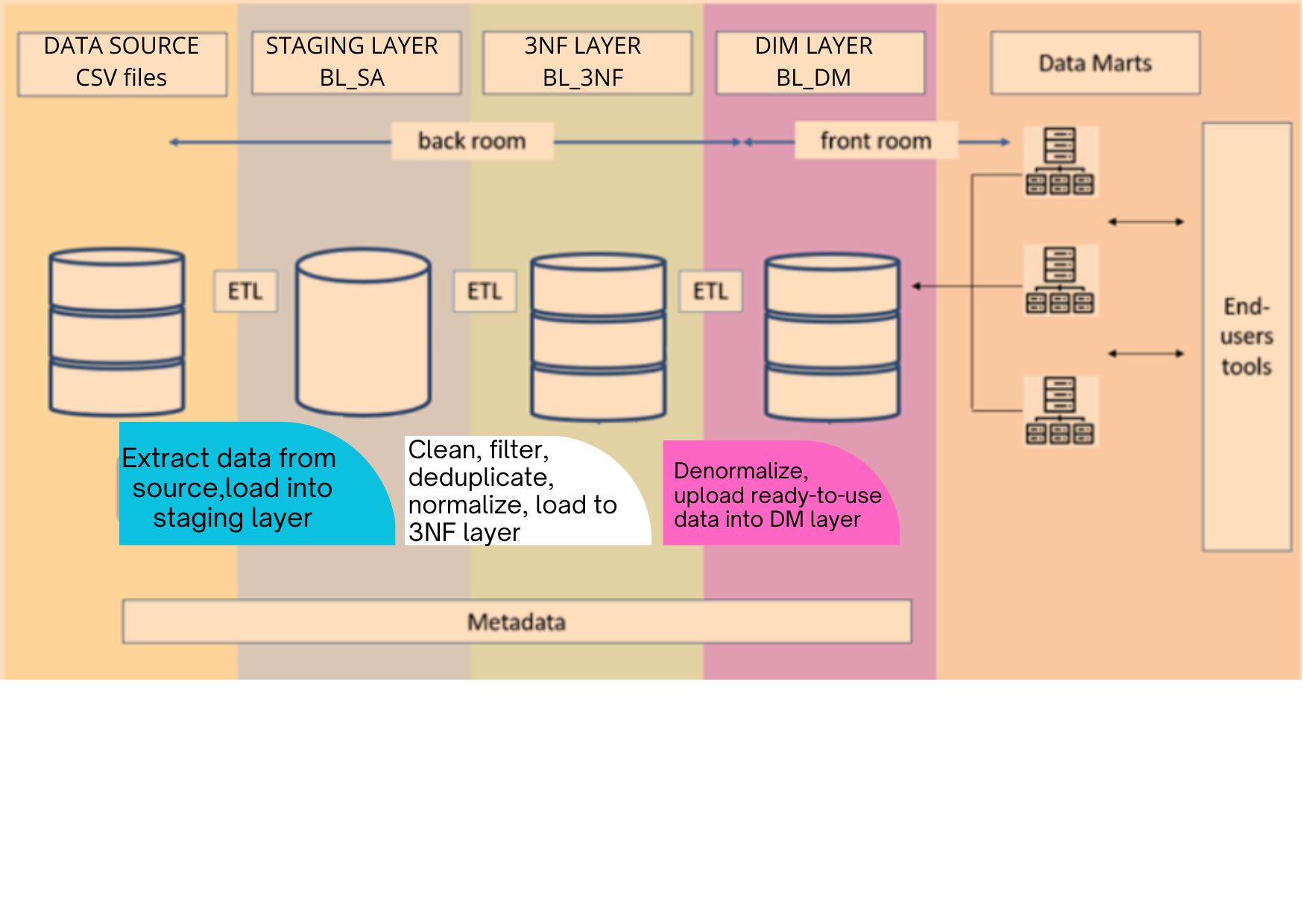
# Business Layer Dimensional Model



In the FCT\_SALES\_DD there is only one metric total\_cost which is generated by quantity\_sold\* unit\_price.

It represents the total amount spent to purchase a specific quantity of a product. This value helps in understanding how much was paid for an order

# Logical Scheme



# Data Flow

# 

# Fact Table Partitioning Strategy