

|  |
| --- |
| Business Template  **supermarket sales data** |
| 130+ Thousand Supermarket Logo Royalty-Free Images, Stock Photos & Pictures  | Shutterstock |

Contents

[1 Business Description 3](#_Toc202208821)

[1.1 Business background 3](#_Toc202208822)

[1.2 Problems because of poor data management 3](#_Toc202208823)

[1.3 Benefits from implementing a Data Warehouse 3](#_Toc202208824)

[1.4 DATASETS DESCRIPTION 3](#_Toc202208825)

[1.4.1 Dataset 1: Online Orders 3](#_Toc202208826)

[1.4.2 Dataset 2: Offline Orders 3](#_Toc202208827)

[1.5 GRAIN / DIM / FACT 4](#_Toc202208828)

[1.5.1 Fact Table 5](#_Toc202208829)

[1.5.2 Dim Tables 6](#_Toc202208830)

[2 Business Layer 3NF 9](#_Toc202208831)

[3 Business Layer Dimensional Model 9](#_Toc202208832)

[4 Logical Scheme 9](#_Toc202208833)

[5 Data Flow 9](#_Toc202208834)

[6 Fact Table Partitioning Strategy 9](#_Toc202208835)

# 

# Business Description

## Business background

The business operates in the **retail industry**, specifically selling consumer products across various city branches. The organization handles both **online and offline orders**, with a broad product portfolio classified into categories and sold through multiple store locations. The company employs sales staff, offers occasional discounts, and operates in a competitive environment where timely and accurate reporting of sales data is crucial to drive strategic decisions.

## Problems because of poor data management

The company has faced several challenges due to fragmented and inconsistent data management practices. The lack of a unified reporting system has resulted in discrepancies between online and offline sales data, complicating the process of performance evaluation and financial forecasting. Manual changing of records not only delays strategic decision-making but also increases the risk of human error. Also, the business has struggled to gain insights into key performance metrics such as cost efficiency, revenue generation, and discount impact. Without integrated customer and product-level analytics, it is difficult to tailor offerings, track performance, and stay ahead of the competition.

## Benefits from implementing a Data Warehouse

Implementing a data warehouse will significantly enhance the company's ability to collect, store, and analyze data from both operational systems. With structured access to consistent and cleansed sales and cost data, decision-makers will be able to track profitability, identify high-performing products and branches, and monitor employee performance across locations. The warehouse will support time-based, product-based, and geography-based reporting and allow for the exploration of patterns in customer purchases. Hierarchical structures within the data, such as those linking product categories to individual items and time-based groupings like year, month, and day, will provide valuable multidimensional views. Ultimately, the business will benefit from improved forecasting, targeted marketing strategies, and optimized inventory management.

## DATASETS DESCRIPTION

### ****Dataset 1: Online Orders****

Includes all orders placed through the company's e-commerce platform. The dataset contains:

* **Product Information:**

Item Code, Item Name, Category Code, Category Name, Wholesale Price, Loss Rate

* **Sales Facts:**

Date, Time, Quantity Sold, Unit Selling Price, Discount, Total Sales, Cost, Gross Income

* **Organizational Data:**

Employee ID, Branch, City

* **Customer Information:**

Simulated using Customer IDs

* **Other:**

Transaction ID, Source System = "Online", Date hierarchy (Year, Month, Day)

### ****Dataset 2: Offline Orders****

Includes all orders placed in company’s physical branches

* **Product Information:**

Item Code, Category Code, Wholesale Price, Loss Rate

* **Sales Facts:**

Date, Time, Quantity Sold, Unit Selling Price, Discount, Total Sales, Cost, Gross Income

* **Organizational Data:**

Employee ID, Branch, City

* **Customer Information:**

Customer ID

* **Other:**

Transaction ID, Source System = "Offline", Date hierarchy (Year, Month, Day)

In this dataset we have several entities, such as

1. **Product**

Which represents items available for sale. Includes product codes, names, categories, pricing, and expected loss rates. Used to analyze what is being sold.

1. **Customer**

Which represents the buyer.

1. **Date**

Which represents the transaction date and supports hierarchical analysis by day, month, and year. Enables time-based trend reporting.

1. **Employee**

Which represents staff responsible for handling sales (online processing or in-store). Helps assess staff performance and assignment.

1. **Branch**

Which represents physical store locations and helps us to analyze geographic trends and branch-level performance.

1. **Channel**

That distinguishes the source of the sale — either **Online** or **Offline** — helping compare e-commerce vs. physical store performance.

1. **Sales Facts**

That is the central numeric data of each transaction: quantity sold, unit price, discounts, total revenue, cost, and gross income. Tied to all dimensions for complete analysis.

The two datasets differ in several important ways. The online dataset contains complete information for all attributes, while the offline version is limited and that shows the lack of advanced data systems. Additionally, transaction IDs are uniquely prefixed to distinguish between the two sources, and the structure of certain dimensions varies to reflect differing system designs. Despite these differences, both datasets share common dimensions, such as date, customer, and geography, which will allow them to be merged in the data warehouse.

## GRAIN / DIM / FACT

The business process selected is 'Sales Transactions'. This process captures all data generated from both online and offline product sales.  
  
The grain of the fact table is defined at the transaction line-item level, meaning one row represents one product sold in a specific transaction. This ensures the atomic level of detail necessary for flexible and accurate aggregation.  
  
The following dimension tables are identified to provide descriptive context: Product, Customer, Employee, Date, Branch (Geography), and Channel (Online/Offline).  
  
The fact table captures measurable metrics such as Quantity Sold, Unit Price, Total Amount, Discount, Cost, and Gross Income. These values are all additive and useful for analysis across various dimensions.

### Fact Table

FCT\_Sales\_DD

This is the central fact table that stores detailed information about each sales transaction. Each row represents a product sold in a specific transaction, along with measurable data such as quantity sold, sales amount, cost, discount, and gross income. It references all dimension tables through foreign keys.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| EVENT\_DT | Date of the transaction | DATE |
| DATE\_KEY | FK to DIM\_TIME\_DAY | BIGINT |
| PRODUCT\_SURR\_ID | FK to DIM\_PRODUCTS\_SCD | BIGINT |
| CUSTOMER\_SURR\_ID | FK to DIM\_CUSTOMERS | BIGINT |
| EMPLOYEE\_SURR\_ID | FK to DIM\_EMPLOYEES | BIGINT |
| BRANCH\_SURR\_ID | FK to DIM\_BRANCHES | BIGINT |
| CHANNEL\_SURR\_ID | FK to DIM\_CHANNELS | BIGINT |
| PRICE\_SURR\_ID | FK to DIM\_PRODUCT\_PRICES\_SCD | BIGINT |
| QUANTITY\_NO | Quantity sold | INT |
| UNIT\_PRICE\_ACT | Unit selling price | FLOAT |
| DISCOUNT\_ACT | Discount applied | FLOAT |
| AMOUNT\_TOT\_ACT | Total revenue | FLOAT |
| COST\_ACT | Cost of goods sold | FLOAT |
| GROSS\_INCOME\_ACT | Gross profit | FLOAT |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |

Examples:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EVENT\_DT | DATE\_KEY | PRODUCT\_SURR\_ID | PRICE\_SURR\_ID | CUSTOMER\_SURR\_ID | EMPLOYEE\_SURR\_ID | BRANCH\_SURR\_ID | CHANNEL\_SURR\_ID | QUANTITY\_NO | UNIT\_PRICE\_ACT | DISCOUNT\_ACT | AMOUNT\_TOT\_ACT | COST\_ACT | GROSS\_INCOME\_ACT | INSERT\_DT | UPDATE\_DT |
| 2024-05-12 | 20240512 | 101 | 1001 | 501 | 3002 | 5 | 1 | 3 | 49.99 | 10.0 | 139.97 | 100.0 | 39.97 | 2024-05-12 | 2024-05-12 |

### Dim Tables

DIM\_Products\_SCD

This table contains descriptive information about products such as item code, item name, category, wholesale price, and expected loss rate. It allows analysis of sales performance by product or product category.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| PRODUCT\_SURR\_ID | Surrogate key | BIGINT |
| PRODUCT\_SRC\_ID | Item code | VARCHAR(20) |
| PRODUCT\_NAME | Product name | VARCHAR(100) |
| CATEGORY\_CODE | Category code | VARCHAR(20) |
| CATEGORY\_NAME | Category name | VARCHAR(100) |
| LOSS\_RATE\_ACT | Expected loss rate | FLOAT |
| START\_DT | Start of valid period | DATE |
| END\_DT | End of valid period | DATE |
| IS\_ACTIVE | Active flag | CHAR(1) |
| INSERT\_DT | Insert timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Examples:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRODUCT\_SURR\_ID | PRODUCT\_SRC\_ID | PRODUCT\_NAME | CATEGORY\_CODE | CATEGORY\_NAME | LOSS\_RATE\_ACT | START\_DT | END\_DT | IS\_ACTIVE | INSERT\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 101 | A456 | Men's Running Shoes | C001 | Footwear | 0.05 | 2020-01-01 | 9999-12-31 | Y | 2020-01-01 | Online | Products |

DIM\_Customers

This table holds information about customers including ID, name, gender, age, and segment type. Enables customer-based reporting such as segmentation, frequency of purchase, and targeting.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| CUSTOMER\_SURR\_ID | Surrogate key | BIGINT |
| CUSTOMER\_SRC\_ID | Customer ID from source | INT |
| CUSTOMER\_NAME | Customer full name | VARCHAR(100) |
| GENDER | Gender | VARCHAR(15) |
| AGE\_NO | Age | INT |
| SEGMENT\_NAME | Segment (e.g., Premium) | VARCHAR(50) |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Example:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CUSTOMER\_SURR\_ID | CUSTOMER\_SRC\_ID | CUSTOMER\_NAME | GENDER | AGE\_NO | SEGMENT\_NAME | INSERT\_DT | UPDATE\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 501 | 501 | Ana Kordzaia | Female | 29 | Premium | 2023-01-10 | 2023-01-10 | Online | Customers |

DIM\_Employees

This table represents sales employees or staff members. Includes employee IDs, names, roles, and hire dates. Useful for analyzing employee performance and sales by staff.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| EMPLOYEE\_SURR\_ID | Surrogate key | BIGINT |
| EMPLOYEE\_SRC\_ID | Employee ID from source | INT |
| EMPLOYEE\_NAME | Employee full name | VARCHAR(100) |
| ROLE\_NAME | Employee role/title | VARCHAR(50) |
| HIRE\_DT | Hiring date | DATE |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |
| SOURCE\_SYSTEM | Source system name | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Example:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_SURR\_ID | EMPLOYEE\_SRC\_ID | EMPLOYEE\_NAME | ROLE\_NAME | HIRE\_DT | INSERT\_DT | UPDATE\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 3002 | 3002 | David Jashi | Sales Associate | 2021-07-10 | 2021-07-10 | 2021-07-10 | Online | Employees |

DIM\_time\_Date

This table stores calendar-related attributes for each transaction date including year, month, day, and weekday. Enables time-based reporting like monthly sales, year-over-year growth, and seasonal trends.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| DATE\_KEY | Surrogate key (sequence-generated) | BIGINT |
| DATE\_SRC\_ID | Natural source date | DATE |
| YEAR\_NO | Calendar year | INT |
| MONTH\_NO | Month number | INT |
| DAY\_NO | Day of the month | INT |
| WEEKDAY\_NAME | Day of the week | VARCHAR(20) |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Example:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATE\_KEY | DATE\_SRC\_ID | YEAR\_NO | MONTH\_NO | DAY\_NO | WEEKDAY\_NAME | INSERT\_DT | UPDATE\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 20240512 | 2024-05-12 | 2024 | 5 | 12 | Sunday | 2024-05-12 | 2024-05-12 | Online | Orders |

DIM\_Branches

This table provides details about physical store locations. Includes branch ID, name, city, and region. Used to evaluate store-level performance and geographic trends.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| BRANCH\_SURR\_ID | Surrogate key | BIGINT |
| BRANCH\_SRC\_ID | Original Branch ID | INT |
| BRANCH\_NAME | Store name | VARCHAR(100) |
| CITY\_NAME | City | VARCHAR(50) |
| REGION\_NAME | Region or State | VARCHAR(50) |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Example:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BRANCH\_SURR\_ID | BRANCH\_SRC\_ID | BRANCH\_NAME | CITY\_NAME | REGION\_NAME | INSERT\_DT | UPDATE\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 5 | 5 | Liberty Mall | Chicago | Illinois | 2022-03-01 | 2022-03-01 | Offline | Branches |

DIM\_Channels

This table distinguishes between sales made online and offline. This dimension enables comparison of performance between different sales platforms.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| CHANNEL\_SURR\_ID | Surrogate key | BIGINT |
| CHANNEL\_SRC\_ID | Source Channel ID | INT |
| CHANNEL\_NAME | Channel name (Online / Offline) | VARCHAR(25) |
| INSERT\_DT | Insert timestamp | DATE |
| UPDATE\_DT | Last update timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

Example:

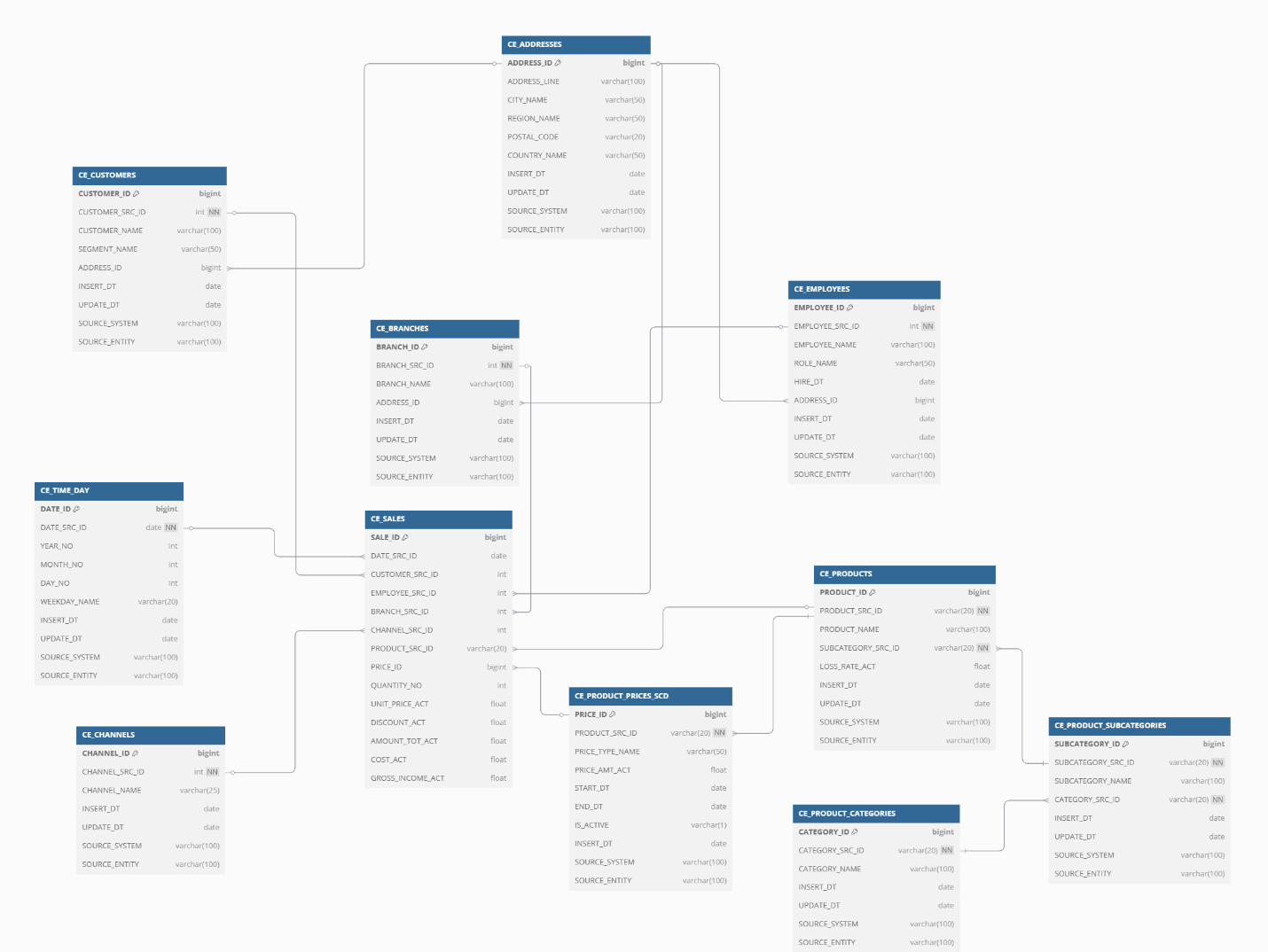
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CHANNEL\_SURR\_ID | CHANNEL\_SRC\_ID | CHANNEL\_NAME | INSERT\_DT | UPDATE\_DT | SOURCE\_SYSTEM | SOURCE\_ENTITY |
| 1 | 1 | Online | 2021-01-01 | 2021-01-01 | System | Channel Table |

DIM\_Product\_Prices\_SCD

This table keeps track of prices changes over time, whether it is caused by inflation or promotion.

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| PRICE\_SURR\_ID | Surrogate key | BIGINT |
| PRODUCT\_SURR\_ID | FK to DIM\_PRODUCTS\_SCD | BIGINT |
| PRICE\_TYPE\_NAME | Price type | VARCHAR(50) |
| PRICE\_AMT\_ACT | Price amount | FLOAT |
| START\_DT | Start of price validity | DATE |
| END\_DT | End of price validity | DATE |
| IS\_ACTIVE | Active flag | CHAR(1) |
| INSERT\_DT | Insert timestamp | DATE |
| SOURCE\_SYSTEM | Source system | VARCHAR(100) |
| SOURCE\_ENTITY | Source table name | VARCHAR(100) |

# Business Layer 3NF



## Overview of the tables

CE\_CUSTOMERS – Core Entity (SCD Type 1)

**This entity represents customer master data consolidated from both online and offline systems.**  
It includes customer ID, name, segment type, and a reference to the customer’s address. Changes (like segment or name updates) overwrite existing data. This is modeled as **SCD Type 1**, and it links directly to the fact table and address dimension.

CE\_EMPLOYEES – Core Entity (**SCD Type 1**)

**This entity stores current employee information such as role, name, and hire date.**  
It includes a reference to the employee’s address. Historical changes like promotions overwrite existing records. This table is modeled as **SCD Type 1** (no versioning) and links to the fact table and address table.

CE\_BRANCHES – Core Entity (SCD Type 1)

**This entity contains information about physical store locations.**  
Includes branch ID, name, and an address reference. Changes to branch names or location details overwrite prior values. This table is **SCD Type 1**, allowing only the latest data to be retained.

CE\_CHANNELS – Core Entity (SCD Type 1 / Static)

**This entity defines whether a sale came from Online or Offline sources.**  
Channel values are mostly static and rarely updated. Modeled as a **Type 0 or Type 1** dimension. Supports cross-channel sales analysis.

CE\_TIME\_DAY – Core Entity (Calendar, SCD Type 1)

**This entity provides a calendar view to support date-level reporting.**  
It includes day, month, year, and weekday fields. This is a **stable SCD Type 1** table, updated only when new dates are loaded.

CE\_ADDRESSES – Core Entity (SCD Type 1)

**This entity centralizes address data for customers, employees, and branches.**  
It includes street, city, region, country, and postal code. Changes to addresses are overwritten in place, so this is modeled as **SCD Type 1**.

CE\_PRODUCT\_CATEGORIES – Core Entity (SCD Type 1)

**This entity stores top-level product classification data.**  
Includes category IDs and names used to group products. It is modeled as **SCD Type 1**, reflecting only the latest category values.

CE\_PRODUCT\_SUBCATEGORIES – Core Entity (SCD Type 1)

**This entity stores mid-level classification between category and product.**  
Each subcategory is linked to a category and is modeled as **SCD Type 1** with no versioning logic.

CE\_PRODUCTS – Core Entity (**SCD Type 1**)

**This entity stores current product-level information.**  
Includes product name, subcategory, and loss rate. It is modeled as **SCD Type 1**, meaning product classification updates are applied in place. Pricing is decoupled into a separate SCD2 table.

CE\_PRODUCT\_PRICES\_SCD – Core Entity (**SCD Type 2**)

**This entity tracks historical changes in product pricing using SCD Type 2 logic.**  
Each row represents a different price version with valid-from and valid-to dates. It includes the product ID, price amount, price type, and SCD metadata (START\_DT, END\_DT, IS\_ACTIVE). This enables full pricing history analysis.

CE\_SALES – Fact Table

**This is the central fact table storing each individual sales transaction at the line-item level.**  
It includes links to all dimensions (product, customer, employee, branch, time, channel) as well as metrics like quantity sold, unit price, discount, cost, and gross income. It uses surrogate IDs and natural keys for lookup joins.

## Steps for creating 3NF form

**Step 1: Analyzed Source Systems**

At the beginning I analyzed both the online and offline sales datasets. Despite structural differences — such as online data containing product names and categories while offline lacked them — shared business entities were identified and unified. These included: **Customer, Product, Employee, Branch, Channel, and Date**. Redundant representations across datasets were merged into single logical entities for consistency and reusability.

**Step 2: Normalized Business Entities (3NF Design)**

Each real-world concept was modeled into a normalized table, ensuring that each entity represents a single subject. Non-key and repeating attributes were separated into distinct tables to eliminate redundancy and ensure referential integrity. For example:

**Sales transactions** were isolated from **product metadata**

**Product hierarchy** was split into **Category → Subcategory → Product**

**Address information** was centralized in CE\_ADDRESSES and referenced by customers, employees, and branches

**Step 3: Applied Naming Conventions**

All objects were named according to EPAM's business layer standards:

All tables were prefixed with CE\_ (Core Entity)

Surrogate keys use the pattern <ENTITY>\_ID

Natural keys from source data use <ENTITY>\_SRC\_ID

Audit timestamps follow the pattern INSERT\_DT, UPDATE\_DT

Slowly Changing Dimension (SCD2) tables include START\_DT, END\_DT, and IS\_ACTIVE

**Step 4: Defined Surrogate and Natural Keys**

Each table was assigned:

A **surrogate key** (e.g., PRODUCT\_ID, EMPLOYEE\_ID) – system-generated and used for joins

A **natural key** (e.g., PRODUCT\_SRC\_ID) – derived from the source system and used in the source triplet

This structure ensures data consistency, flexibility for versioning, and traceability to original systems.

**Step 5: Added Source Triplet and Technical Metadata**

To support **data governance and lineage**, every entity includes:

The **source triplet**: SOURCE\_SYSTEM, SOURCE\_ENTITY, and <ENTITY>\_SRC\_ID

**Audit fields**: INSERT\_DT, UPDATE\_DT to track when data was loaded or modified

These fields ensure full transparency of data origin and update history across all tables.

**Step 6: Implemented Slowly Changing Dimensions (SCD)**

The model distinguishes between SCD types based on business requirements:

CE\_PRODUCT\_PRICES\_SCD was implemented as **SCD Type 2**  
This allows versioning of historical product price changes. It includes START\_DT, END\_DT, and IS\_ACTIVE fields, and supports margin and price trend analysis over time.

**Other tables**, such as CE\_EMPLOYEES, CE\_CUSTOMERS, and CE\_PRODUCTS, were modeled as **SCD Type 1**, meaning only the current version of data is maintained and updated in place.

**Step 7: Modeled Logical Relationships**

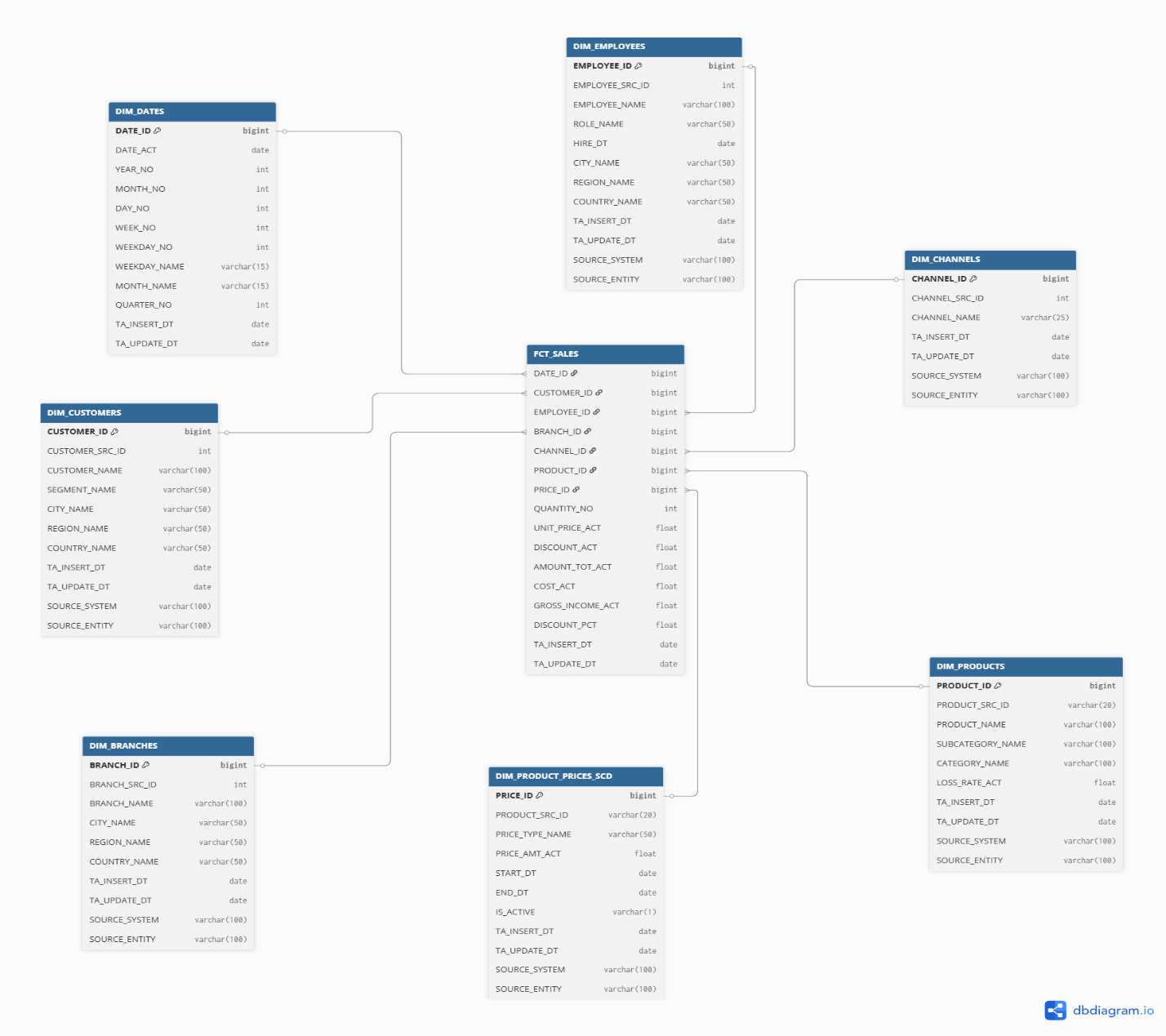
Logical relationships were established through **surrogate keys**, especially between the central fact table (CE\_SALES) and its dimensions:

CE\_SALES references: CUSTOMER\_ID, EMPLOYEE\_ID, BRANCH\_ID, CHANNEL\_ID, DATE\_ID, PRODUCT\_SRC\_ID, and PRICE\_ID

Relationships are defined using natural keys where appropriate (especially for product pricing and time)

For **SCD Type 2 dimensions**, such as CE\_PRODUCT\_PRICES\_SCD, relationships are handled **logically** using business rules and lookup logic (e.g., matching by PRODUCT\_SRC\_ID and SALE\_DATE within a valid price range), rather than enforcing a physical foreign key.

# Business Layer Dimensional Model



## Overview of metrics

**QUANTITY\_NO**

This metric represents the number of units of a product sold in a specific transaction line. It is essential for understanding sales volume and is often used in identifying top-selling products, unit-level trends, and forecasting demand.

**UNIT\_PRICE\_ACT**

This is the actual price per unit that the customer paid at the time of purchase. It may reflect promotions or negotiated pricing. This metric is crucial for price analysis, discount evaluations, and margin comparison across products or channels.

**DISCOUNT\_ACT**

This metric records the total amount discounted from the sale. It may originate from promotional campaigns, loyalty benefits, or manual overrides. Tracking this value helps assess the impact of discounts on overall revenue and promotional effectiveness.

**AMOUNT\_TOT\_ACT**

This metric represents the total revenue collected from the sale after the discount is applied. It is calculated as:  
(QUANTITY\_NO × UNIT\_PRICE\_ACT) - DISCOUNT\_ACT.  
It provides the net sales value and is the most direct input for revenue analysis and gross income calculations.

**COST\_ACT**

This metric holds the cost of goods sold (COGS) for the product in the transaction. It typically comes from the product price dimension (DIM\_PRODUCT\_PRICES\_SCD) and reflects the internal cost incurred to sell one or more units. It is used to determine profitability and margin performance.

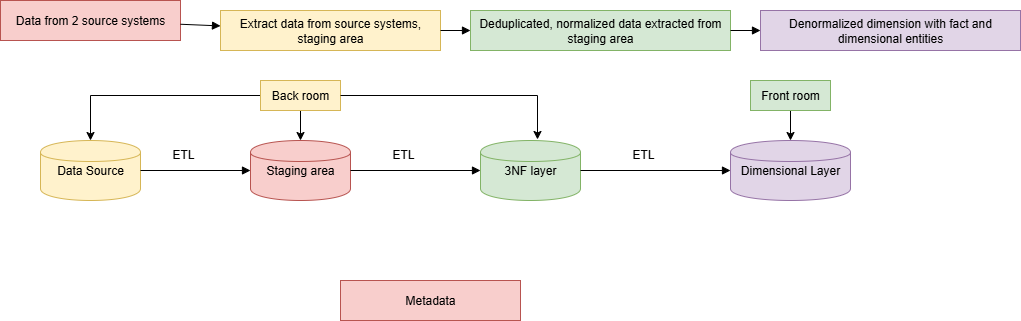
**GROSS\_INCOME\_ACT**

This is a derived metric calculated as:  
GROSS\_INCOME\_ACT = AMOUNT\_TOT\_ACT - COST\_ACT.  
It shows the profit made from the transaction line, before deducting operational or overhead expenses. This metric is critical for profitability reporting and identifying high-margin products or customers.

**DISCOUNT\_PCT** (Calculated Metric)

This metric expresses the discount as a percentage of the full price, using the formula:  
(DISCOUNT\_ACT / (QUANTITY\_NO × UNIT\_PRICE\_ACT)) × 100.  
It helps categorize transactions by discount level, identify over-discounting, and analyze discount strategies by customer segment, employee, or product.

# Logical Scheme



The logical schema for the supermarket data warehouse is composed of several layers that reflect the data transformation flow from source to analytics-ready structures:

**Layers:**

**Source Systems:**

Online Orders Dataset (e-commerce transactions)

Offline Orders Dataset (physical store transactions)

**Staging Layer (SA):**

Raw data is imported via external tables using file\_fdw or flat file ingestion.

Basic validation, null handling, and deduplication occur here.

Table examples: sa\_online.ext\_online\_orders, sa\_offline.ext\_offline\_orders

**Business Layer 3NF (BL\_3NF):**

Entities are normalized and structured in 3NF.

Includes core entities like:

CE\_CUSTOMERS, CE\_EMPLOYEES, CE\_PRODUCTS, CE\_ADDRESSES, CE\_PRODUCT\_PRICES\_SCD, CE\_BRANCHES, CE\_CHANNELS, CE\_TIME\_DAY, CE\_SALES

SCD Type 2 is applied to product prices, all other dimensions are Type 1.

**Dimensional Model (BL\_DM):**

Data is transformed into a star schema for analytical queries.

Central fact table: FCT\_SALES\_DD

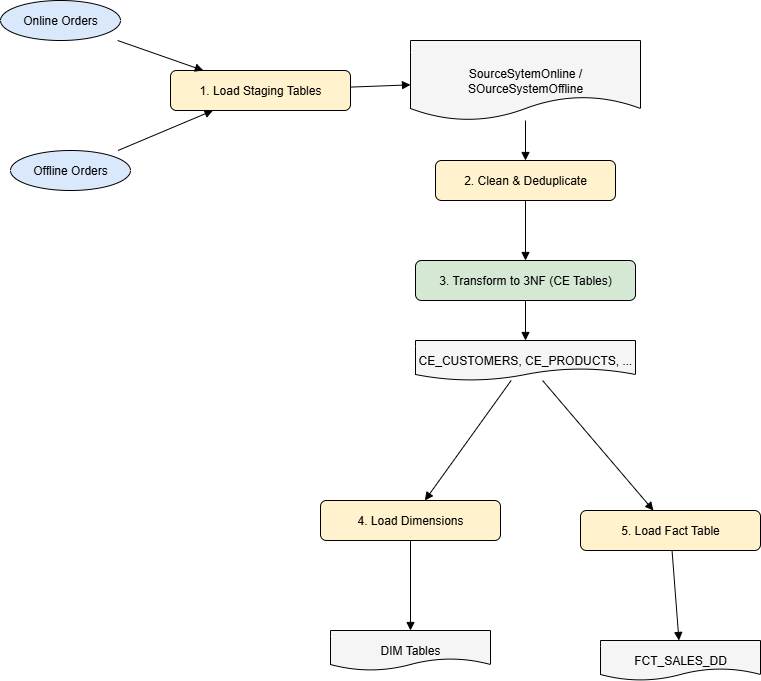
Dimension tables: DIM\_CUSTOMERS, DIM\_EMPLOYEES, DIM\_PRODUCTS\_SCD, DIM\_PRODUCT\_PRICES\_SCD, DIM\_BRANCHES, DIM\_CHANNELS, DIM\_DATES

**Logical Relationships:**

All dimension tables are linked to the fact table via surrogate keys.

DIM\_PRODUCT\_PRICES\_SCD is logically joined using PRODUCT\_SURR\_ID and date ranges from EVENT\_DT.

# Data Flow



# Fact Table Partitioning Strategy