

|  |
| --- |
| Describtion  **SUPERSTORE DATASET** |
| **Logo / Image** |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background

The Superstore dataset represents sales transactions across multiple geographic locations and product categories. The database supports tracking of orders, customers, products, and shipping details.

## Problems. Current Situation

The same information (customer details, product information, etc.) is stored in multiple places, leading to inconsistencies.

Without proper constraints and validation, data errors are common.

Generating reports requires manual data extraction from multiple sources.

The current structure makes it difficult to perform complex analysis on sales trends, customer behavior, or product performance.

The current system becomes increasingly difficult to maintain.

Different departments maintain their own data, leading to fragmented information and inconsistent understanding.

## the Benefits of implementing a database. Project Vision

A normalized database ensures that each piece of information is stored exactly once, eliminating redundancy and inconsistencies.Proper constraints and relationships ensure data accuracy and validity.

Easy access to related data allows for more comprehensive and automated reporting.

The structured data enables complex analytics to discover sales patterns, customer preferences, and market trends.The database can grow with the business without compromising performance.

# Model description

## Definitions & Acronyms

Definitions & Acronyms

PK: Primary Key

FK: Foreign Key

M2M: Many-to-Many relationship (resolved via junction tables).

## Logical Scheme

## Objects

Table Description

<description>

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field Name | Field Description | Data Type |
| category | category\_id (PK) | Unique identifier for product categories | BIGINT |
|  | category\_name | Name of the category (e.g., Furniture) | VARCHAR(100) |
| sub\_category | sub\_category\_id (PK) | Unique identifier for sub-categories | BIGINT |
|  | sub\_category\_name | Name of the sub-category | VARCHAR(100) |
|  | category\_id (FK) | References category.category\_id | BIGINT |
| product | product\_id (PK) | Unique identifier for products | VARCHAR(20) |
|  | product\_name | Name of the product | VARCHAR(100) |
|  | sub\_category\_id (FK) | References sub\_category.sub\_category\_id | BIGINT |
| order | order\_id (PK) | Unique order identifier | VARCHAR(20) |
|  | order\_date | Date the order was placed | DATE |
|  | ship\_date | Date the order was shipped | DATE |
|  | ship\_mode\_id (FK) | References ship\_mode.ship\_mode\_id | VARCHAR(20) |
|  | customer\_id (FK) | References customer.customer\_id | VARCHAR(20) |
| customer | customer\_id (PK) | Unique customer identifier | VARCHAR(20) |
|  | customer\_name | Name of the customer | VARCHAR(100) |
|  | segment\_id (FK) | References segment.segment\_id | BIGINT |
|  | city\_id (FK) | References city.city\_id | BIGINT |
| ship\_mode | ship\_mode\_id (PK) | Unique identifier for shipping modes | BIGINT |
|  | ship\_mode\_name | Name of the shipping mode | VARCHAR(100) |
| segment | segment\_id (PK) | Unique identifier for customer segments | BIGINT |
|  | segment\_name | Name of the segment | VARCHAR(100) |
| region | region\_id (PK) | Unique identifier for regions | BIGINT |
|  | region\_name | Name of the region | VARCHAR(100) |
| country | country\_id (PK) | Unique identifier for countries | BIGINT |
|  | country\_name | Name of the country | VARCHAR(100) |
| state | state\_id (PK) | Unique identifier for states | BIGINT |
|  | state\_name | Name of the state | VARCHAR(50) |
|  | country\_id (FK) | References country.country\_id | BIGINT |
|  | region\_id(FK) | Reference to region | BIGINT |
| city | city\_id (PK) | Unique identifier for cities | BIGINT |
|  | city\_name | Name of the city | VARCHAR(100) |
|  | state\_id (FK) | References state.state\_id | BIGINT |
| order\_product | order\_product\_id(PK) | Unique identifier for line item, PK | BIGINT |
|  | order\_id(FK) | Reference to order | VARCHAR(20) |
|  | product\_id(FK) | Reference to product | VARCHAR(20) |
|  | quantity | Number of units ordered | BIGINT |
|  | sales | Total sales amount for this item | DECIMAL |
|  | profit | Total profit for this item | DECIMAL |

Example with data

Orders table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| order\_id | order\_date | ship\_date | ship\_mode\_id | customer\_id | sales | quantity | profit |
| CA-2020-123 | 2020-01-01 | 2020-01-05 | 12234 | 2134 | 261.96 | 2 | 41.9136 |

# Model Description

This section describes the steps taken to design the database schema:

Goals:

Normalize data to eliminate redundancy (e.g., geographic hierarchies stored as text).

Establish relational integrity between entities (e.g., orders, products, customers).

Enable efficient querying of hierarchical relationships (e.g., region → country → state).

Approach:

Entity Identification: Identified core entities (e.g., orders, product, customer, region) based on the Superstore dataset’s requirements.

Normalization:

Split denormalized fields (e.g., "City," "State") into separate tables (city, state, country, region).

Created junction tables (e.g., m2m\_order\_product) to resolve M2M relationships.

Relationship Mapping:

Linked product → sub\_category → category for hierarchical product categorization.

Established geographic hierarchy: city → state → country → region.

Outcomes:

A fully normalized schema compliant with PostgreSQL.

Clear relationships between tables, enabling complex queries (e.g., sales by region, customer segmentation).