Business Template

BIG STORE SALE DATA

Contents

1 Business Description

1.1 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.

1.2 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.

1.3 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**.

1.4 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.

reorder_point: Minimum stock level before reordering.

reorder_quantity: Quantity to reorder when stock is low.

Customer Information

customer_id: Unique identifier for each customer.

customer_loyalty_level: Loyalty program level for the 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

transaction_id: Unique identifier for each transaction.

transaction_date: Date and 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_age: Age of the customer.

customer_gender: Gender of the customer.

customer_income: Income level of the customer.

customer_loyalty_level: Loyalty program level for the customer

Suppliers

supplier_id: Unique identifier for the supplier providing the product.

supplier_lead_time: Time taken by the supplier to deliver products.

Additional Features

promotion_applied: Whether a promotion was applied to the transaction.

promotion_type: Type of promotion used (e.g., Discount, Free Item).

weather_conditions: Weather conditions on the transaction date.

holiday_indicator: Whether the transaction date was a holiday.

weekday: Day of the week for the transaction date.

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., age, 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.

1.5 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.

QUARTER: the quarter of the transaction (WHICH WILL BE ADDED LATER)

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_NAME: the name of the customer.

CUSTOMER_AGE: the age of the 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_age	INT	Age 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
customer_address	INT (FK)	Links to the Address table

Address

Column Name	Data Type	Description
address_id	INT (PK)	Unique identifier for the address
state	VARCHAR(20)	State of the address
street	VARCHAR(50)	Street name
postal_n	VARCHAR(10)	Postal code
appart_n	VARCHAR(10)	Apartment number

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.

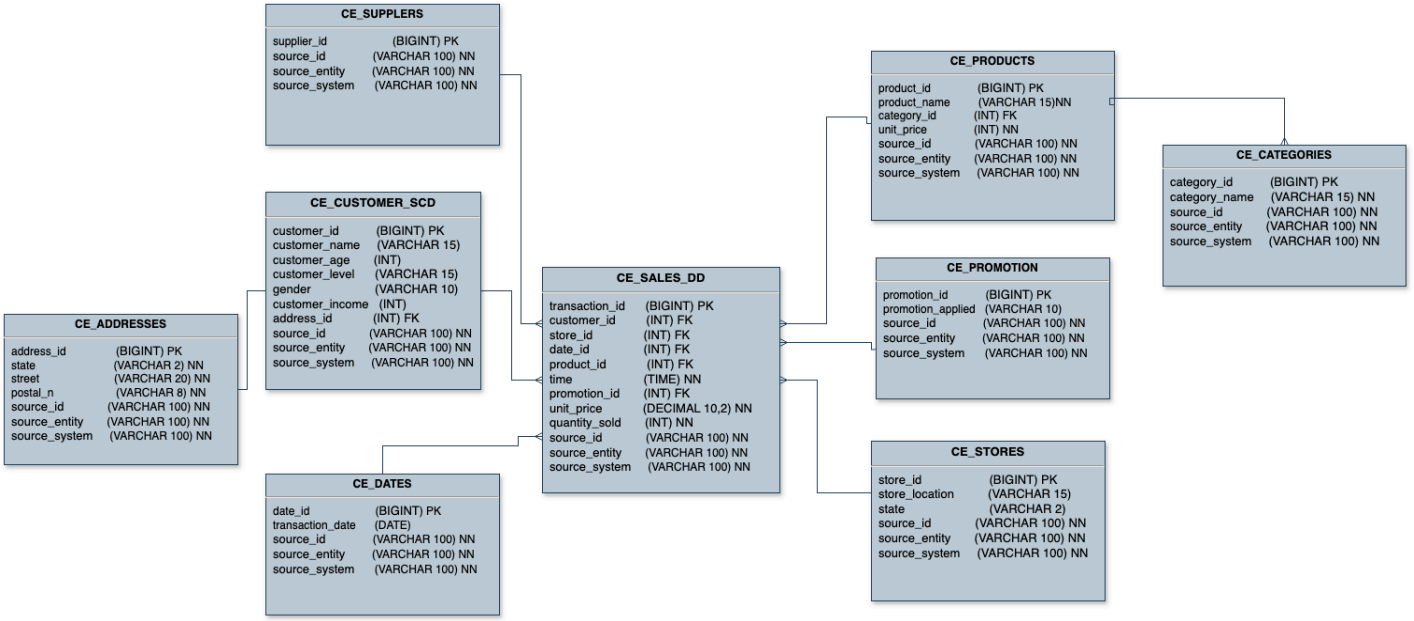
Sales Forecasting: I can look at trends and predict future sales.

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.

2 Business Layer 3NF



The CE_SALES_DD 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), date_id (Foreign Key to CE_DATE), promotion_id (Foreign Key to CE_PROMOTIONS), quantity_sold, unit_price, total_amount, source_system, source_entity, and source_id. 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_name, customer_age, customer_level, gender, customer_income, and address_id (Foreign Key to CE_ADDRESSES). 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). This table provides detailed information about products involved in transactions.
SCD type 1(overwrite old product details with new ones)

The CE_PROMOTIONS table stores promotion information. It includes attributes such as promotion_id (Primary Key) and promotion_applied. This table provides details about promotions applied to transactions.
SCD type 0 (when promotion is used, it can't be changed)

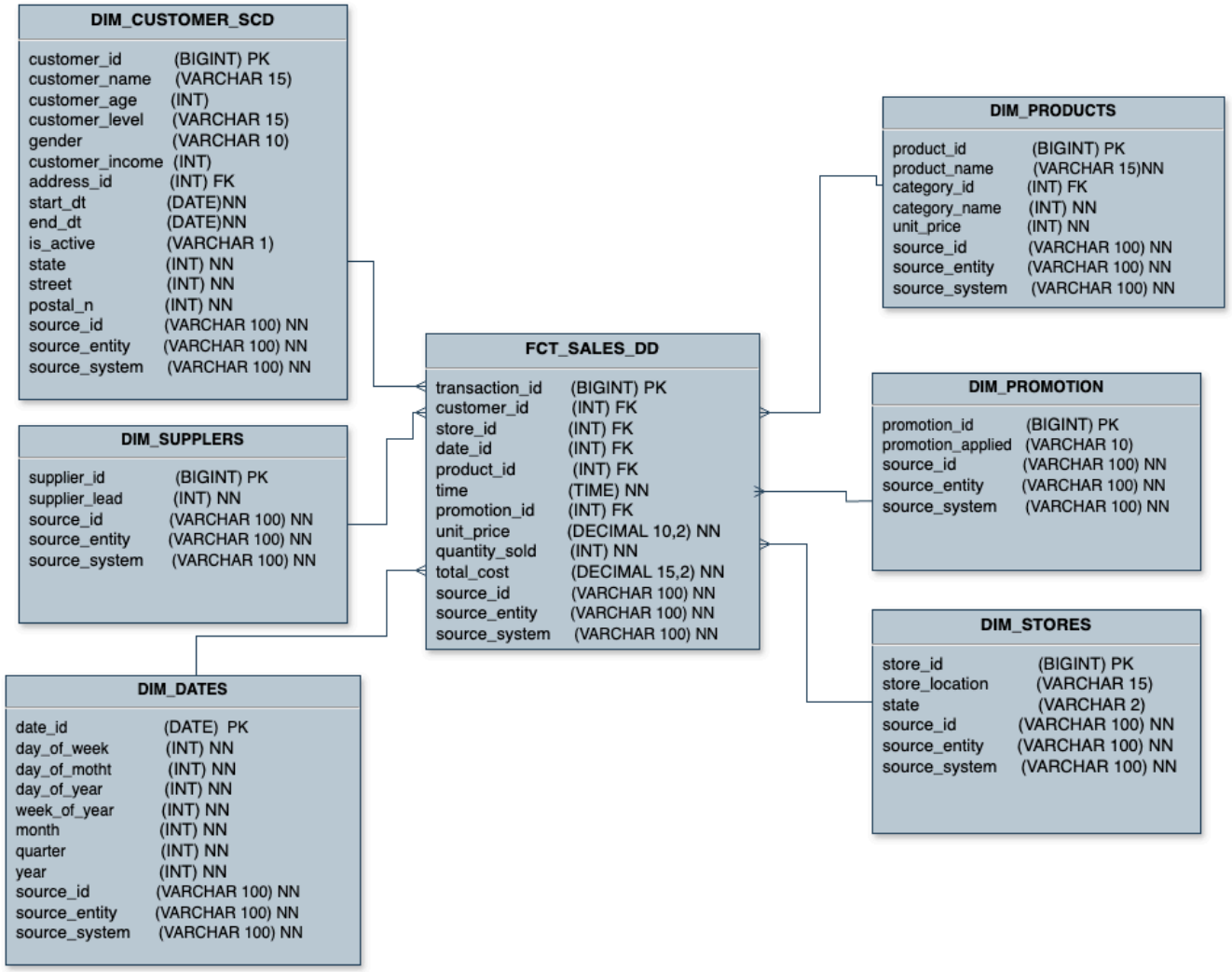
The CE_ADDRESSES table stores address information. It includes attributes such as address_id (Primary Key), state, street, postal_n, and appart_n. This table provides detailed address information for customers.
SCD type 2 (add a new row when an address changes to keep history)

The CE_DATE table stores date information. It includes attributes such as date_id (Primary Key) and transaction_date. This table provides date details for transactions.
SCD type 0 (dates don't change they stay the same forever)

The CE_CATEGORIES table stores product category information. It includes attributes such as category_id (Primary Key) and category_name. 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. This table provides details about suppliers of products.
SCD type 1 (overwrite old supplier details with new ones)

3 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

4 Logical Scheme

5 Data Flow

6 Fact Table Partitioning Strategy