Consulting Project

Design and Implementation of a Data Warehouse

Report 2 - Logical Design and Physical Design of the Data Warehouse Schema

Prepared By:
Sri Sai Kowshik Reddy Boyalla
Mithilesh Menakuru
Prajakta Ingle
Varshitha Ravikumar

Index

Section 1: Introduction

The logical design for the Dominick Fine Food data warehouse will be designed using Ralph Kimball's dimensional modeling principles since it is optimized for query performance and ease of use. Then we will create a star schema for the data marts relating to our business questions with a central fact table surrounded by dimension tables. Fact tables will display quantitative information like sales amount, and quantity sold with foreign keys to the dimension tables. Dimension tables on the other hand will display descriptive attributes like time and store dimensions. To solve the business questions each fact table will be designed at an appropriate grain level.

We'll implement Kimball's bus architecture which will be explained in the later sections. Here we will create data marts that can be seamlessly integrated guaranteeing a single version of truth for the enterprise. This will allow us for incremental development and deployment of data marts. We can also prioritize based on the business questions that we identify to be critical.

The logical design will consist of a staging area to execute ETL processes with staging tables. The data quality checks will be performed at the staging area to ensure data integrity before the presentation layer. The presentation layer will contain a denormalized star schema with appropriate indexing and aggregate tables to improve the query response time.

Finally, a robust metadata layer will be designed to justify business definitions and dimensions. By following this detailed logical design approach, we aim to create a scalable, flexible data warehouse that will help DFF to analyze historical data and get insights for decision-making.

Section 2: Overview of Kimball’s Methodology

Kimball's methodology, also known as the Kimball Lifecycle Methodology, is a widely accepted approach to designing data warehouses. It emphasizes building data marts that are designed around business processes and later integrating them into an enterprise-wide data warehouse. Here’s a breakdown of each step and the importance of following this methodology to create independent data marts.

Steps in Kimball's Methodology

1. **Business Requirements Definition**

   Description: The first step involves gathering requirements from business stakeholders to understand the key business processes, the metrics (facts) they need to track, and the dimensions (context) they need to analyze. This step focuses on identifying the specific business questions the data warehouse needs to answer.

   Importance: Understanding business needs ensures the data warehouse is designed to provide actionable insights and meet user expectations.

2. **Data Source Identification and Prioritization**

   Description: After gathering requirements, identify the data sources that will provide the necessary information. These sources can be internal systems (e.g., ERP, CRM) or external data feeds. Prioritize the most critical data sources based on the requirements.

   Importance: Identifying and prioritizing data sources ensures the data warehouse pulls accurate and reliable information, while also managing the complexity of data integration.

3. **Dimensional Modeling (Star Schema Design)**

   Description: This is the heart of Kimball’s approach. In this step, business processes are modeled using fact tables (containing quantitative data) and dimension tables (providing context to the facts). Each data mart is designed as a star schema (a fact table linked to several dimension tables).

   Importance: Dimensional modeling helps ensure that the data is easy to query and interpret. It also ensures scalability, as new data marts can be added and linked to common dimensions.

4. **Physical Design**

   Description: This step involves defining how the data will be physically stored in the database, including setting up indexing strategies, partitioning, and optimizing the performance of the data warehouse.

   Importance: Proper physical design is critical to ensuring performance, particularly for complex queries and large volumes of data. The goal is to optimize storage and query speed.

5. **ETL Design and Development**

   Description: Extract, Transform, Load (ETL) processes are designed to extract data from source systems, transform it to fit the dimensional model, and load it into the data warehouse. This process includes cleaning, integrating, and transforming data to ensure quality and consistency.

   Importance: A robust ETL process is crucial to ensure that the data in the warehouse is accurate, consistent, and up-to-date. Poor ETL design can lead to unreliable data and slow performance.

6. **Data Mart Delivery**

   Description: Once the data is loaded, the individual data mart is delivered to the business users. This includes setting up reporting tools, dashboards, and OLAP cubes to help users analyze the data.

   Importance: Delivering the data mart to users with intuitive tools ensures that they can easily access and analyze the data without needing to know the technical intricacies of the warehouse.

7. **Iterate and Enhance**

   Description: The data warehouse is not static. After the initial release, additional iterations are made to add new data marts, refine ETL processes, and expand functionality based on feedback and evolving business needs.

   Importance: Iteration ensures the data warehouse remains aligned with the changing needs of the business. It also allows for continuous improvement and adaptation to new business processes.

Why It’s Important to Follow Kimball's Methodology to Create Independent Data Marts

1. **Modular and Incremental Development**

   Kimball's methodology advocates creating data marts incrementally, with each one focusing on a specific business process (e.g., sales, finance, marketing). These data marts are designed independently but use conformed dimensions (shared dimension tables across multiple data marts). This approach makes it easier to build the data warehouse in smaller, manageable pieces that deliver immediate business value while avoiding the complexity of trying to build an entire enterprise data warehouse in one go.

2. **Business-Centric Approach**

   By focusing on business processes and requirements from the start, Kimball's methodology ensures that the data warehouse is closely aligned with the organization's business needs. Each data mart is designed to answer specific business questions, making the data warehouse more valuable and user-friendly.

3. **Simplified Integration**

   One of the key aspects of Kimball's approach is the concept of conformed dimensions, which allows for easy integration of data marts into a larger enterprise data warehouse. Even though the data marts are built independently, they can be linked together because they share common dimensions (e.g., time, customer, product). This makes it possible to create a unified data warehouse over time without needing to redesign or rework the data marts.

4. **Reduced Complexity**

   Building data marts independently following Kimball's methodology reduces the initial complexity of building an enterprise-wide data warehouse. Each data mart is a self-contained, manageable project that focuses on a specific business process. Once several data marts are in place, they can be integrated to form the larger data warehouse. This approach prevents large-scale failures that could occur if you attempt to build everything at once.

5. **Scalability and Flexibility**

   Kimball's methodology is designed to be flexible. As business needs evolve, new data marts can be added to the warehouse with minimal disruption to the existing infrastructure. The modular nature of the data marts, combined with the use of conformed

dimensions, makes it easy to expand the data warehouse to accommodate new business processes or additional data sources.

6. **User Empowerment and Adoption**

   The business-driven nature of Kimball's methodology results in data marts that are highly relevant to end-users. When users can directly see the value of the data mart in answering their questions, they are more likely to adopt the system. Kimball's approach also emphasizes delivering solutions quickly, allowing users to benefit from data insights earlier, rather than waiting for an entire warehouse to be completed.

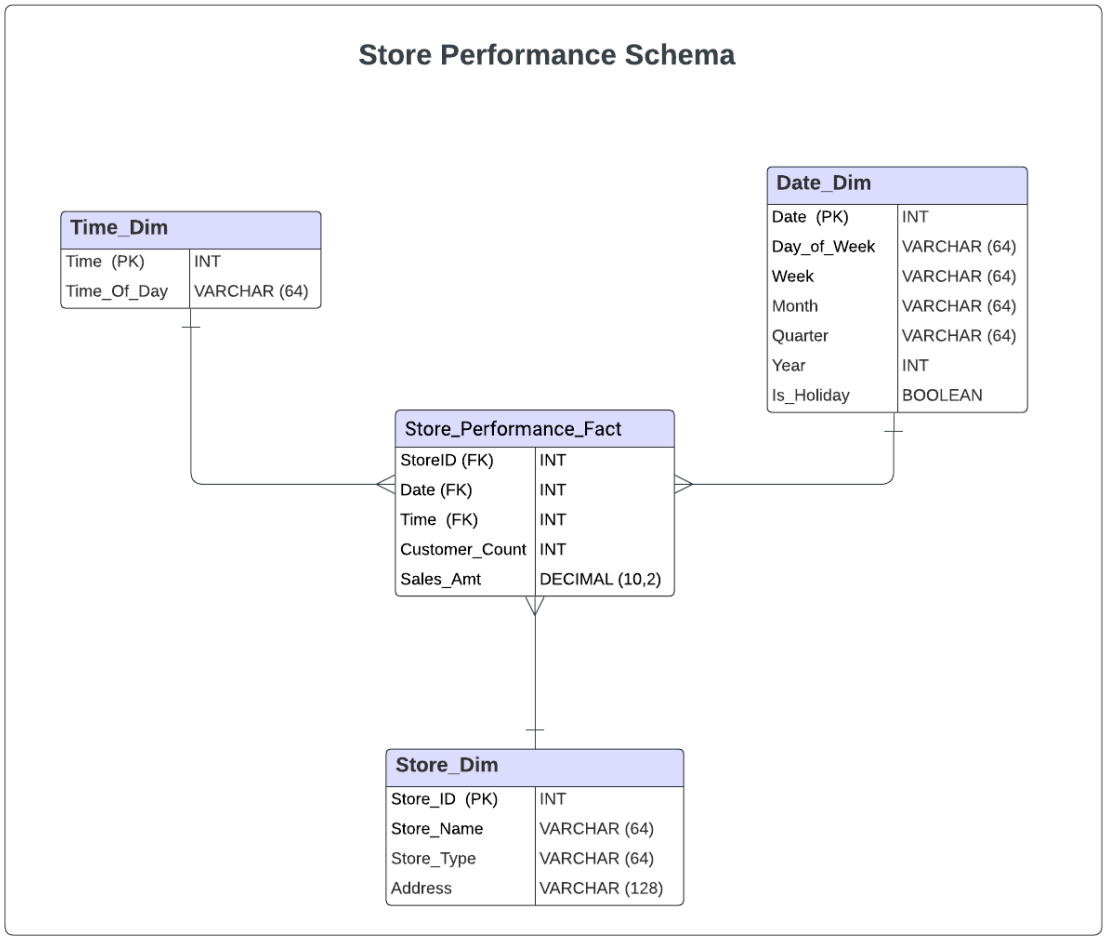
Conclusion

Kimball's methodology offers a practical, business-focused, and iterative approach to building data warehouses. By following this methodology, organizations can build independent data marts that provide immediate business value while maintaining the flexibility to integrate them into a larger, enterprise-wide data warehouse. This approach ensures the data warehouse remains aligned with business needs and can grow incrementally as new processes and data requirements emerge.

Section 3: DW Logical Design (Star Schema Design)

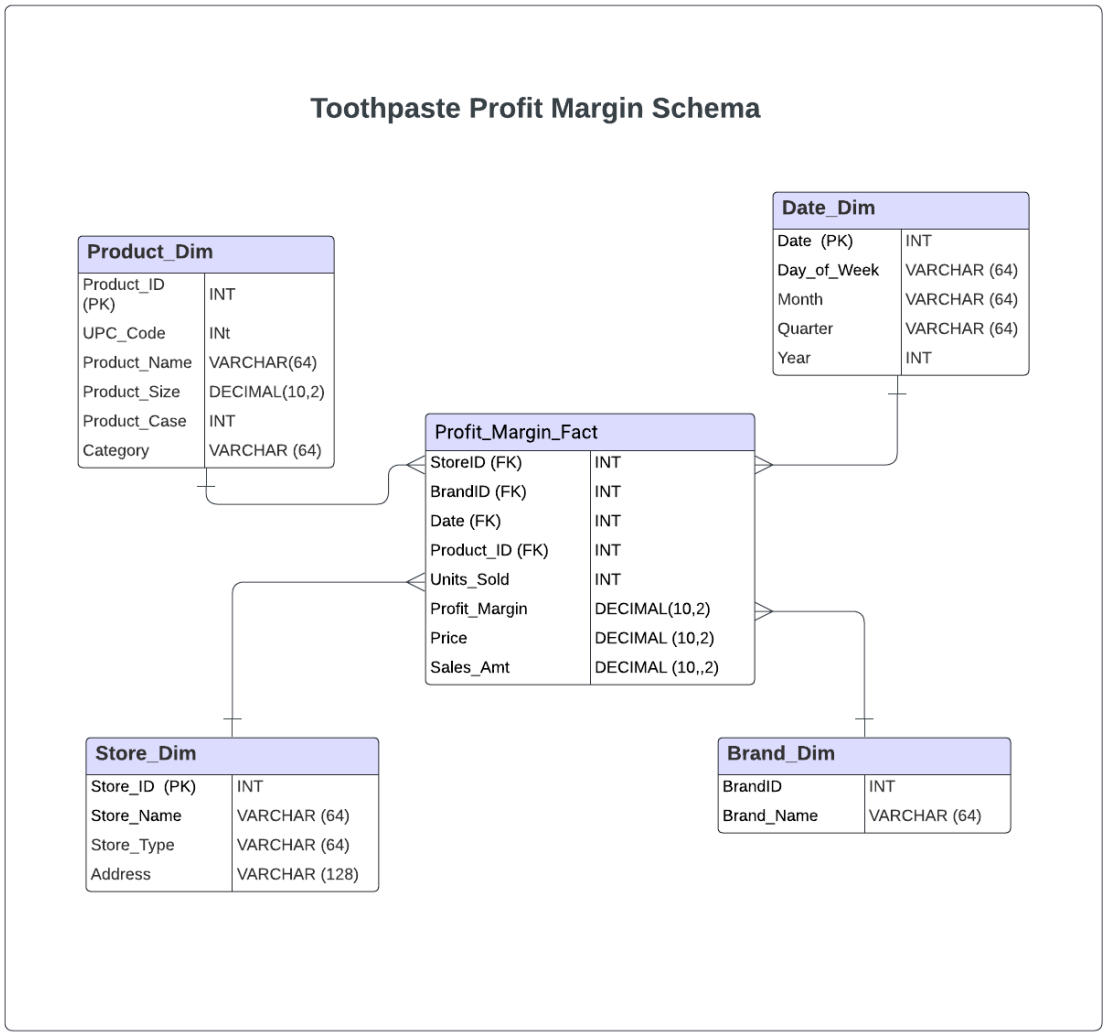
Store Performance Schema

This Schema addresses the business question **"Which store has the highest store traffic for the last quarter of 1994?".** The schema includes a fact table that contains customer count data linked to specific stores and dates. The data can be further filtered to identify the customer count in the last quarter of 1994 across all stores. The store_dim helps us identify the store with the highest traffic during the last quarter of 1994 and provides insights into store performance and customer behavior.



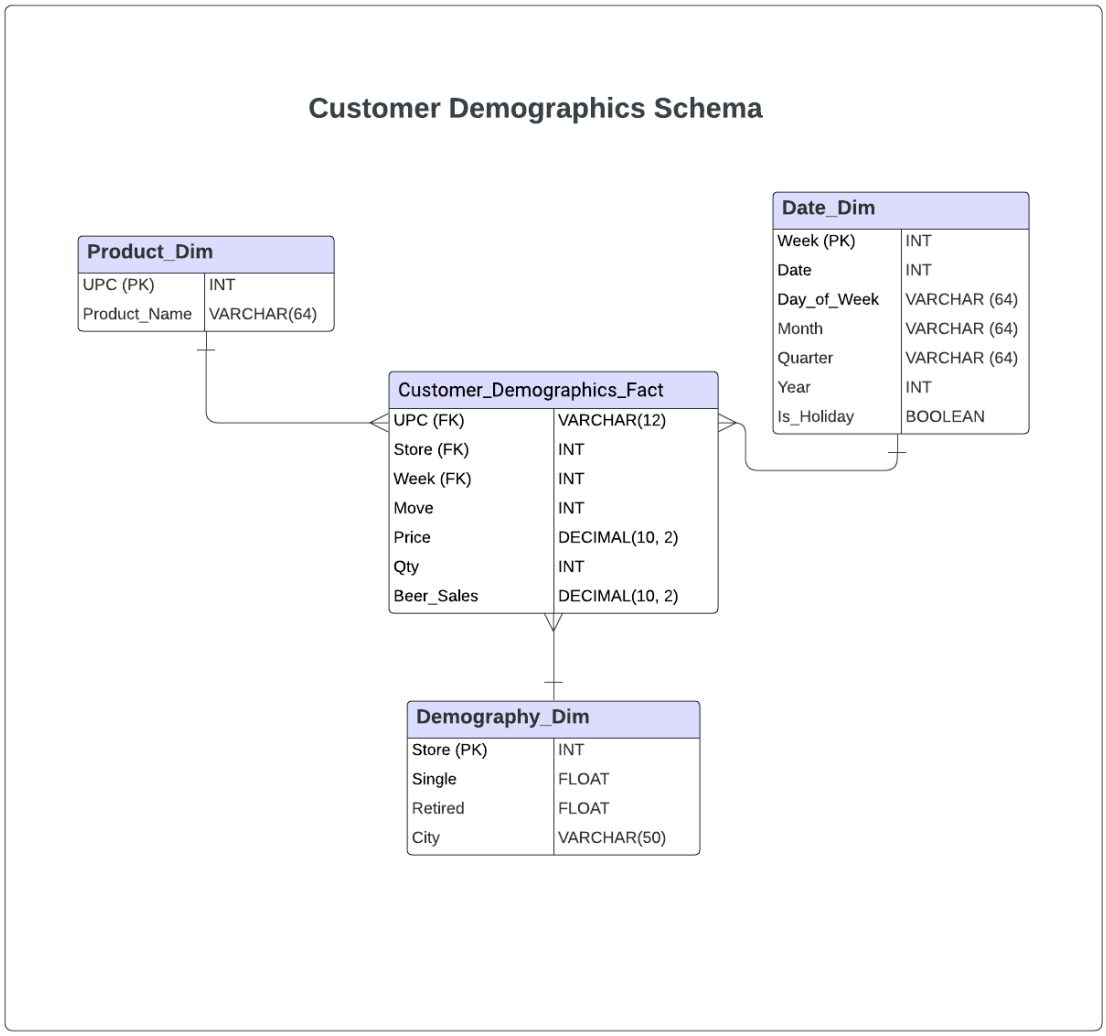
Toothpaste Profit Margin Schema

This Schema addresses the business question **“How does the profit margin of toothpaste vary by brand?”.** The schema links the products with their brands through the product_dim table. This allows us to analyze the difference in profit margins across different toothpaste brands. The insights gathered through this analysis is essential to adjust the pricing strategies and assessing the performance based on the brands in each category



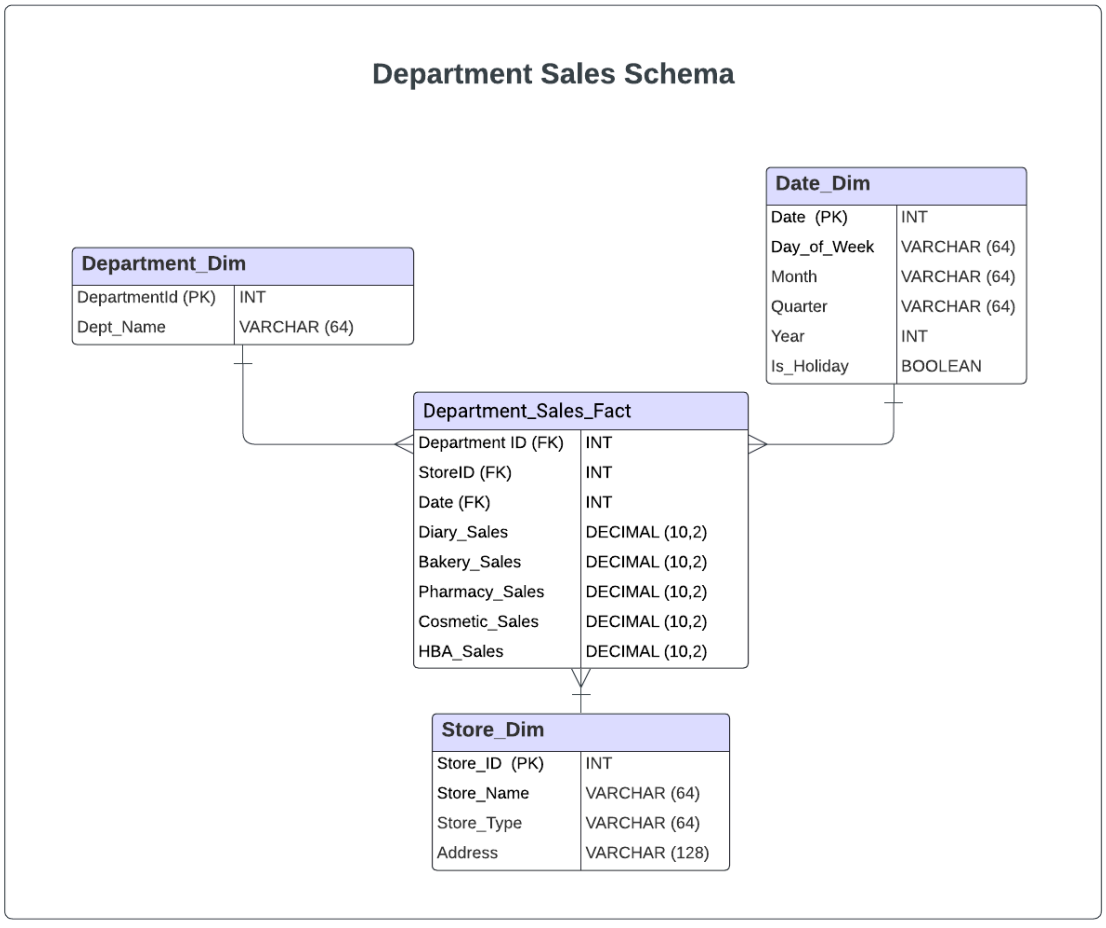
Customer Demographics Schema

This Schema addresses the business question **"What were the highest sales contributions from single and retired individuals in the Buffalo Grove stores for BUDWEISER BEER N.R.B during the Thanksgiving week of 1993?".** This Schema links the demographics data with sales data. The data can be filtered through particular demographics categories in demography_dim and link it with specific products in product_dim table. This allows us to analyze the sales from a particular demographic group which is essential for targeted marketing and product placement.



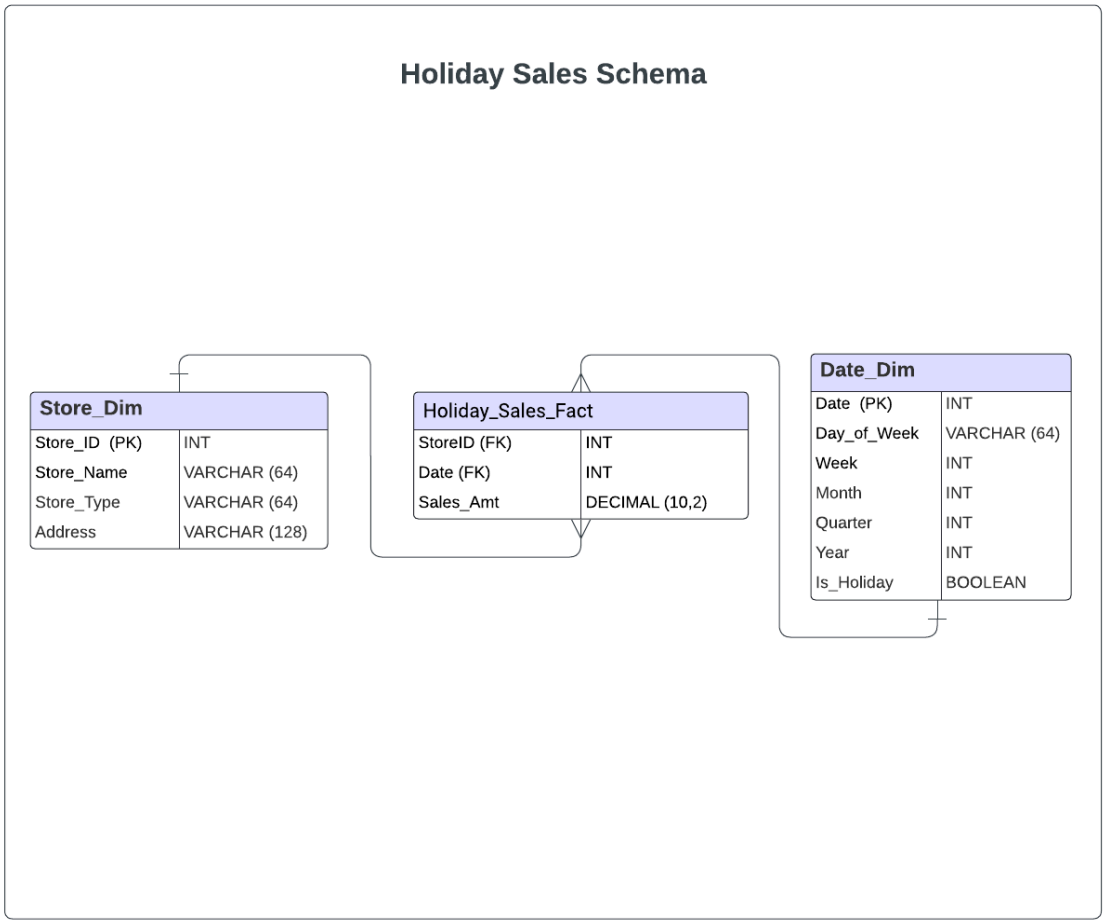
Department Sales Schema

This Schema addresses the business question **"What are the monthly average sales amounts for the BAKERY, DAIRY, PHARMACY, COSMETIC, and HABA departments across Dominick's Fine Food stores located in Naperville and Schaumburg during the year 1994, and how do these averages compare among the different departments within these locations?".** The schema captures the sales data specific to each department. The sales amount can be analyzed by filtering through specific departments and locations using stores_dim table, it can then be aggregated by month using date_dim table to calculate monthly average sales for each department. This analysis allows us to compare the performance of different departments within specific stores which aids in the resource allocation and marketing efforts.



Holiday Sales Schema

This Schema addresses the business question **“Which holiday week in 1991 and 1992 saw the highest grocery sales in the low-tier Buffalo Grove stores?”.** The schema captures the sales data during the holiday season. The sales amount is linked with the dates marked as holiday in the date_dim table, the store can be filtered to Buffalo Grove stores in the store_dim table. This will help us get the data regarding the seasonal sales in a particular store which is necessary to optimize the inventory and plan promotion for future.



Section 4: Mapping Tables

4.1 Source to Staging

For BQ 1: ccount.csv -> WEEK, STORE, CUSTCOUN, DATE
For BQ 2: ccount.csv->STORE, DATE, DAY (New Column Needed), GROCERY
For BQ 3: ccount.csv -> STORE, DATE, DIARY, BAKERY, PHARMACY, COSMETIC, HBA, MONTH (Needed a new column - month)
For BQ 4: wtpa.csv -> UPC, PROFIT, MOVE, PRICE, QTY
Upctpa.csv -> UPC, DESCRIP, SIZE, CASE
For BQ 5: WBER.csv -> UPC, STORE, WEEK, MOVE, PRICE, QTY,
DEMO.csv -> STORE, SINGLE, RETIRED, CITY
CCOUNT.csv -> STORE, WEEK, BEER

BQ 1					
Source data	Source data field	Mapping	Staging Table type	Staging Table name	Attribute
CCOUNT.csv	WEEK	Copy	Relation	CCOUNT_staging	Week
CCOUNT.csv	STORE	Copy	Relation	CCOUNT_staging	Store_Num
CCOUNT.csv	CUSTCOUN	Copy	Relation	CCOUNT_staging	Cust_Count
CCOUNT.csv	DATE	Copy	Relation	CCOUNT_staging	Date
BQ 2					
Source data	Source data field	Mapping	Staging Table type	Staging Table name	Attribute
CCOUNT.csv	GROCERY	Copy	Relation	CCOUNT_staging	Grocery_sales
CCOUNT.csv	STORE	Copy	Relation	CCOUNT_staging	Store_Num
CCOUNT.csv	DATE	Copy	Relation	CCOUNT_staging	Date
CCOUNT.csv	DAY (N)	Copy	Relation	CCOUNT_staging	Day

BQ 3					
Source data	Source data field	Mapping	Staging Table type	Staging Table name	Attribute
CCOUNT.csv	DIARY	Copy	Relation	CCOUNT_staging	Diary_sales
CCOUNT.csv	STORE	Copy	Relation	CCOUNT_staging	Store_Num
CCOUNT.csv	DATE	Copy	Relation	CCOUNT_staging	Date
CCOUNT.csv	BAKERY	Copy	Relation	CCOUNT_staging	Bakery_sales
CCOUNT.csv	PHARMACY	Copy	Relation	CCOUNT_staging	Pharmacy_sales
CCOUNT.csv	COSMETIC	Copy	Relation	CCOUNT_staging	Cosmetic_sales
CCOUNT.csv	HBA	Copy	Relation	CCOUNT_staging	HBA_sales
CCOUNT.csv	MONTH (N)	Copy	Relation	CCOUNT_staging	Month
BQ 4					
Source data	Source data field	Mapping	Staging Table type	Staging Table name	Attribute
WPTA.csv	UPC	Copy	Relation	WPTA_staging	Upc
WPTA.csv	PROFIT	Copy	Relation	WPTA_staging	Profit_Margin
WPTA.csv	MOVE	Copy	Relation	WPTA_staging	Move
WPTA.csv	PRICE	Copy	Relation	WPTA_staging	Price
WPTA.csv	QTY	Copy	Relation	WPTA_staging	Qty
WPTA.csv	STORE	Copy	Relation	WPTA_staging	Store_Id
WPTA.csv	TOTAL SALES (N)	Copy	Relation	WPTA_staging	Total_Sales
UPCTPA.csv	UPC	Copy	Relation	UPCTPA_staging	Upc_Id
UPCTPA.csv	DESCRIP	Copy	Relation	UPCTPA_staging	Product_name
UPCTPA.csv	SIZE	Copy	Relation	UPCTPA_staging	Product_size

UPCTPA.csv	CASE	Copy	Relation	UPCTPA_staging	Product_case
BQ 5					
Source data	Source data field	Mapping	Staging Table type	Staging Table name	Attribute
CCOUNT.csv	STORE	Copy	Relation	CCOUNT_staging	Store_Num
CCOUNT.csv	WEEK	Copy	Relation	CCOUNT_staging	Week
CCOUNT.csv	BEER	Copy	Relation	CCOUNT_staging	Beer_Sales
DEMO.csv	STORE	Copy	Relation	DEMO_staging	Store_Id
DEMO.csv	SINGLE	Copy	Relation	DEMO_staging	Single_perct
DEMO.csv	RETIRED	Copy	Relation	DEMO_staging	Retired_perct
DEMO.csv	CITY	Copy	Relation	DEMO_staging	City
WBER.csv	UPC	Copy	Relation	WBER_staging	Upc
WBER.csv	STORE	Copy	Relation	WBER_staging	Store_Id
WBER.csv	WEEK	Copy	Relation	WBER_staging	Week
WBER.csv	MOVE	Copy	Relation	WBER_staging	Move
WBER.csv	PRICE	Copy	Relation	WBER_staging	Price
WBER.csv	QTY	Copy	Relation	WBER_staging	Qty

4.2 Staging to Data marts

BQ 1 [Data Mart 1]					
Source data in staging	Staging table data field	Mapping	Data Mart Table type	Table name	Attribute
CCOUNT_staging	Week	Copy	Dimension Table	Date_Dim	Week

CCOUNT_staging	Store_Num	Copy	Dimension Table	Store_Dim	Store_ID
CCOUNT_staging	Cust_Count	Copy	Fact Table	Store_Performance_Fact	Customer_Count
CCOUNT_staging	Date	Copy	Dimension Table	Date_Dim	Date
BQ 2 [Data Mart 2]					
Source data in staging	Staging table data field	Mapping	Data Mart Table type	Table name	Attribute
CCOUNT_staging	Grocery_sales	Copy	Fact Table	Holiday_Sales_Fact	Sales_Amt
CCOUNT_staging	Store_Num	Copy	Dimension Table	Store_Dim	Store_ID
CCOUNT_staging	Day	Copy	Dimension Table	Date_Dim	Day_of_Week
CCOUNT_staging	Date	Copy	Dimension Table	Date_Dim	Date
BQ 3 [Data Mart 3]					
Source data in staging	Staging table data field	Mapping	Data Mart Table type	Table name	Attribute
CCOUNT_staging	Diary_sales	Copy	Fact Table	Department_Sales_Fact	Diary_Sales
CCOUNT_staging	Bakery_sales	Copy	Fact Table	Department_Sales_Fact	Bakery_Sales
CCOUNT_staging	Pharmacy_sales	Copy	Fact Table	Department_Sales_Fact	Pharmacy_Sales
CCOUNT_staging	Cosmetic_sales	Copy	Fact Table	Department_Sales_Fact	Cosmetic_Sales
CCOUNT_staging	HBA_sales	Copy	Fact Table	Department_Sales_Fact	HBA_Sales
CCOUNT_staging	Month	Copy	Dimension Table	Date_Dim	Month

CCOUNT_staging	Store_Num	Copy	Dimension Table	Store_Dim	Store_ID
CCOUNT_staging	Date	Copy	Dimension Table	Date_Dim	Date
BQ 4 [Data Mart 4]					
Source data in staging	Staging table data field	Mapping	Data Mart Table type	Table name	Attribute
WPTA_staging	Upc	Copy	Dimension Table	Product_Dim	UPC_Code
WPTA_staging	Profit_Margin	Copy	Fact Table	Profit_Margin_Fact	Profit_Margin
WPTA_staging	Move	Copy	Fact Table	Profit_Margin_Fact	Move
WPTA_staging	Price	Copy	Fact Table	Profit_Margin_Fact	Price
WPTA_staging	Qty	Copy	Fact Table	Profit_Margin_Fact	Units_Sold
WPTA_staging	Total_Sales	Copy	Fact Table	Profit_Margin_Fact	Sales_Amt
WPTA_staging	Store_Id	Copy	Dimension Table	Store_Dim	Store_ID
UPCTPA_staging	Upc_Id	Copy	Dimension Table	Product_Dim	UPC_Code
UPCTPA_staging	Product_name	Copy	Dimension Table	Product_Dim	Product_Name
UPCTPA_staging	Product_size	Copy	Dimension Table	Product_Dim	Product_Size
UPCTPA_staging	Product_case	Copy	Dimension Table	Product_Dim	Product_Case
BQ 5 [Data Mart 5]					
Source data in staging	Staging table data field	Mapping	Data Mart Table type	Table name	Attribute

CCOUNT_staging	Week	Copy	Dimension Table	Date_Dim	Week
CCOUNT_staging	Beer_Sales	Copy	Fact Table	Customer_Demographics_Fact	Beer_Sales
DEMO_staging	Store_Id	Copy	Dimension Table	Demography_Dim	Store
DEMO_staging	Single_perct	Copy	Dimension Table	Demography_Dim	Single
DEMO_staging	Retired_perct	Copy	Dimension Table	Demography_Dim	Retired
DEMO_staging	City	Copy	Dimension Table	Demography_Dim	City
WBER_staging	Upc	Copy	Dimension Table	Product_Dim	UPC
WBER_staging	Move	Copy	Fact Table	Customer_Demographics_Fact	Move
WBER_staging	Price	Copy	Fact Table	Customer_Demographics_Fact	Price

Section 5: Physical Design

The logical design for DFF will be based on Ralph Kimball's dimension approach. This is optimized for ease of use and query performance. For the data aggregation, we will create aggregate tables for the commonly searched data to enhance the query performance. These aggregates will be updated incrementally during nonpeak hours to create minimum impact when real-time operations. Next for indexing, we will create a combination of clustered and non-clustered indexes on dimension tables and fact tables. Clustered indexes will be created using the primary keys of the dimension tables and the date column of fact tables. We can use column store indexes on large fact tables in order to enhance query performance.

For data standardization, we can implement an ETL process to ensure that data is consistent across all the data marts. This will include defining a standard format for date, product codes and store identifiers. In order to handle future growth and expansion, we can implement a tiered storage strategy. This involves storing currently and frequently accessed data on high-performance SSDs with historical data being stored to a less expensive storage options. We will use the SQL Server partitioning feature to manage large tables providing an easy means to archive old data. We can also implement a data compression strategy for storage space optimization, especially for historical data.