Product Performance Analysis for E-commerce Success

Nitesh Padidam
vpadidam@buffalo.edu
50495795
vpadidam

Manoj Paladi
manojpal@buffalo.edu
50496792
manojpal

Thrivikrama Rao Kavuri
thrivikr@buffalo.edu
50496382
thrivikr

I. Problem Statement

The aim of the project is to identify our top-performing and under-performing products by analyzing sales, returns i.e Product Performance Analysis and Seasonal trends in sales i.e Seasonal Demand and customer segmentation analysis from the data.

A. Discuss the background of the problem leading to your objectives. Why is it a significant problem?

Product performance analysis :

Identifying how certain products or categories of products are performing in terms of sales, refunds, and customer satisfaction requires conducting a product performance analysis. It assists in assessing the effectiveness of the product's offerings and locating potential improvement areas.

Seasonal Demand Analysis:

As numerous companies suffer shifts in revenue based on seasons, holidays, and other events, seasonal demand analysis is essential. Inventory management and marketing strategies entail an understanding of these trends.

Customer Segmentation :

Through customer segmentation, it is possible to classify clients according to their behavior, demographics, and preferences. It is crucial to customize product offers, customer service, and marketing to diverse customer segments.

B. Explain the potential of your project to contribute to your problem domain. Discuss why this contribution is crucial?

These analyses contribute to good business strategies and optimise cost expenditure.

Based on Product performance analysis, Data driven decisions can be taken on which products to continue or discontinue based on their performances.

Based on return data , quality of product can be enhanced by identifying the product defects.

Utilising Seasonal demand analysis, inventory levels can be optimised by reducing overstocking and stockouts, supply chain can be planned ensuring smooth deliveries and reduced costs.

From Customer data Segmentation, marketing campaigns of products can be planned to the customer segment resulting in higher conversion and new products can be designed aligning the specific segment.

C. Why do you need a database instead of an Excel?

Database is needed instead of excel to address below issues:

1) Scalability: As time passes, volume of data increases, and excel cannot handle that whereas a database handles large volumes of data efficiently.
2) Data Integrity: The database can enforce integrity and consistency based on defined constraints in relations whereas Excel needs manual intervention.
3) MultiUser Access: The database is designed so that multiple users can update/use the data at the same time whereas Excel cannot handle this
4) Security: Database has the ability to restrict access on the user level which is difficult on Excel
5) Data Retrieval: In databases, SQL can be used to retrieve complex aggregations, and filters which cannot be done in Excel.

II. Target User

In real life scenarios, for E-commerce businesses, the database is a critical component. It is used by:

Developers to build and maintain website.

Data Analysts to gather insights from sales trend, customer behaviour etc.

Business Analyst to report financial performance, customer satisfaction.

Customer support team to access customer data and order information to solve the problems related to that.

Business Owners access database make strategic decisions, track overall performance and plan for future scope.

Database Administrators are the team that will administer the database and are responsible for design, implementation and maintenance. Responsibilities also include managing user access and permissions, optimizing database performance, and handling data migrations and updates.

III. Exploring the Data

A. How will you use the data?

In the domain of E-Commerce, understanding data plays a major role. To perform this effectively, the ability to query and update the data is crucial. Hence, a Database is used. Data can be used to find out total revenue, best-selling products, sales by category, sales by time period (daily, monthly, seasonal), and

sales by region. Customer Behaviour can be analyzed based on data further used for customer segmentation

B. What kind of queries do you want to ask?

Selection queries are used to retrieve specific information like information about an order or customer.
Aggregation queries can be asked to get statistics of sales by week, month and most returned product.
Filtering queries can be asked to get information of customers whose bill value is above certain range or product details which are sold over a certain number.
Join Queries can be used to combine sales or return data with product and customer tables

C. How is the data updated?

This particular domain has data updates frequently as new products enter market and new customers order hence increasing sale data.
These updates into database tables are done with the help of predefined constraints in the database schema using procedures or functions ensuring the data integrity.

D. Your application should support both queries and updates?

The database being used is PostgreSQL which can accommodate queries and updates ensuring data integrity and security and is also compatible with linking for web applications, cloud services, and ETL.

IV. ER Diagram

*A. **Relational Schemas***

1) **Territory**(SalesTerritoryKey, Region, Country)
   • SalesTerritoryKey – Uniqueness, Not null and irreducibility of this attribute made it choose as a primary key
   • Country – No action when primary key is deleted
2) **Continent** (Country, Continent)
   • Country – Uniqueness, irreducibility and Immutability of this attribute made it choose as a primary key
3) **Returns** (ReturnDate, TerritoryKey, ProductKey, ReturnQuantity)
   • Return date – Not null, Meaningful and to maintain foreign key relationships this attribute made it choose as a primary key
   • TerritoryKey - Uniqueness, Not null and to maintain foreign key relationships this attribute made it chosen as a primary key
   • ProductKey – Selectivity and to maintain foreign key relationships this attribute made it chosen as a primary key
   • ReturnQuantity – Uniqueness and to maintain foreign key relationships this attribute made it chosen as a primary key
4) **Calendar** (Date)
   • Date – Selectivity and uniqueness of this attribute made it choose as a primary key
5) **Product_Subcategory**(ProductSubcategoryKey, SubcategoryName)
   • ProductSubcategoryKey – Stability, Meaningful and to maintain foreign key relationships this attribute made it choosen as a primary key
   • Subcategoryname – Subcategory name foreign key is defined along with cascade as to ensure that deletion of primary key will result in removal of this subcategory name data.
6) **Category_subcategory**(SubcategoryName, ProductCategoryKey)
   • SubcategoryName – Uniqueness, meaningful and irreducibility of this attribute made it chosen as a primary key
   • ProductCategorykey – Product Category Foreign Key is defined along with cascade to ensure that deletion of the primary key will result in removal of this subcategory
7) **Product_Category**(ProductCategoryKey,CategoryName)
   • ProductCategoryKey – Uniqueness, selectivity and to maintain foreign key relationships this attribute made it chosen as a primary key
8) **Customers**(CustomerKey, Prefix, FirstName, LastName, BirthDate, MaritalStatus, Gender, EmailAddress, AnnualIncome, TotalChildren,, EducationLevel, Occupation, HomeOwner)
   • Customerkey - Uniqueness, selectivity, and maintaining foreign key relationships this attribute made it chosen as a primary key
   • Prefix – Is set default to null
   • Annual Income – Is set default to 0
   • TotalChildren – Is set default to 0
   • Educationlevel – Is set default to Null
   • Occupation – Is set default to Null
   • HomeOwner – Is set default to Null
   • MaritalStatus - Is set default to Null
   • ProductKey – Irreducibility, Uniquness and to maintain foreign key relationships this attribute made it choose as a primary key
9) **Products**(ProductKey, ProductSubcategoryKey, ProductCost, ProductPrice)
   • ProductKey – Irreducibility, Uniquness and to maintain foreign key relationships this attribute made it choose as a primary key
   • ProductSubCategoryKey - foreign key is defined along with cascade to ensure that deletion of the primary key will result in removal of this subcategory.
10) **Product_Name**(ProductKey, ProductName)
    ProductKey - Irreducibility, Uniquness and maintaining foreign key relationships this attribute made it chosen as a primary key
    • ProductName - foreign key is defined along with cascade to ensure that deletion of primary key will result in removal of this product.
11) **Products_description**(ProductSKU,ProductName, ModelName, ProductDescription, ProductColor,

ProductSize, ProductStyle)

• ProductName – Meaningful, Not Null, and irreducibility of this attribute made it choose as a primary key

12) **Sales**(OrderDate,StockDate, OrderNumber,ProductKey,CustomerKey,TerritoryKey, OrderLine Item, OrderQuantity)

• OrderNumber – Uniqueness, irreducibility of this attribute made it choose as a primary key

• ProductKey - Irreducibility, Uniquness and to maintain foreign key relationships this attribute made it choose as a primary key

• CustomerKey - foreign key is defined along with cascade as to ensure that deletion of primary key will result in removal of this customer key in sales table even.

• TerritoryKey - foreign key is defined along with cascade as to ensure that deletion of primary key will result in removal of this territory data in sales table.

• Orderdate - foreign key is defined along with no action as to ensure that deletion of primary key will result in removal of this orderdate of this table.

• OrderQuantity – Is set default to 0.

B. Relationships between tables:

Many to One:

This relationship in a database or data model involves multiple records in one entity (referred to as the "many" side) being linked to a single record in another entity (referred to as the "one" side). It is essentially a "one-to-many" relationship seen from a different angle.

Relationships :

Returns —◇→ Calendar
Sales —◇→ Calendar
Returns —◇→ Products
Sales —◇→ Products
Sales —◇→ Customers
Sales —◇→ Territory
Territory —◇→ Continent
Category_Subcategory —◇→ Product_Category
Product_subcategory —◇→ products

One to One :

In a one-to-one relationship within a database or data model, each record in one entity is intricately linked to precisely one record in the other entity, and the same holds true in reverse. This type of relationship is characterized by the essential features of uniqueness and exclusivity in the connection between these two entities.

Relationships :

Territory ←◇→ Customers
Products_description ←◇→ Product_Name
Category_Subcategory ←◇→ Product_Subcategory
Products ←◇→ Product_Name

C. ER Diagrams

Initial ER Diagram -

ER diagram for the tables created from raw data i.e non normalised database is plotted as shown below in Fig-1.

Final ER Diagram -

ER diagram for the relations after normalization are shown in below Fig-2.

V. Loading Data

Source of raw data of the database are in csv formats. Data from these files are brought into the PostgreSQL Database. Below are the steps :

• Inspecting the data from csv , DDL of tables are created with respective data format except the Date column; it is also imported as varchar initially.

• Primary key and foreign Key are declared in DDL. NOT NULL constraint and default value of attributes are also declared.

• Furtherly , data for each table is directly imported from csv with UTF8 and LATIN1 encoding.

• All the attributes with the date field are varchar and are corrected to date format YYYY-MM-DD and then the data type of these attributes are also converted to DATE.

• As this is initial data, to ensure smooth start, data is directly imported instead of insert queries. Further down the process , ETL will be designed to update data ensuring no manual intervention.

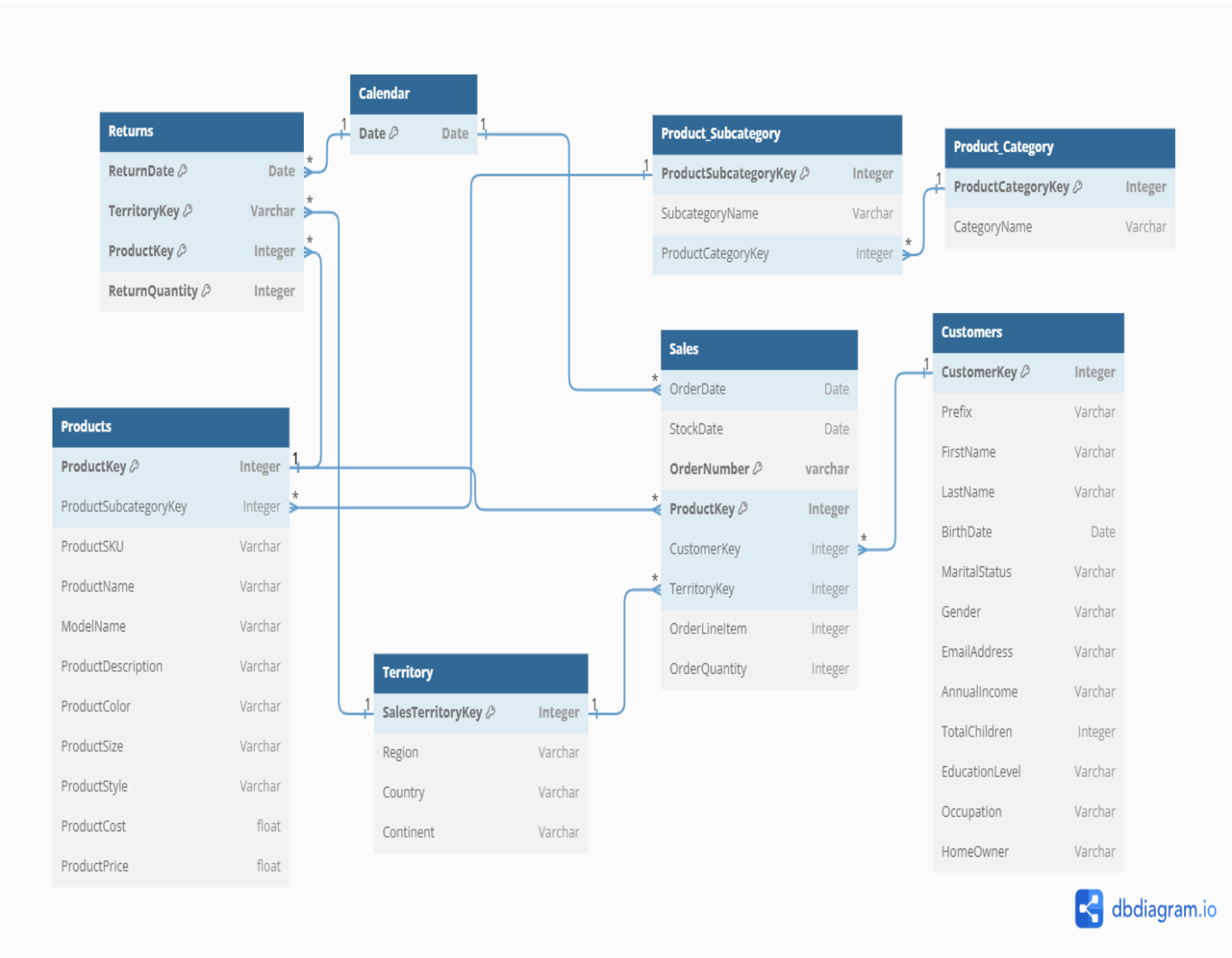

Fig. 1. Initial ER diagram

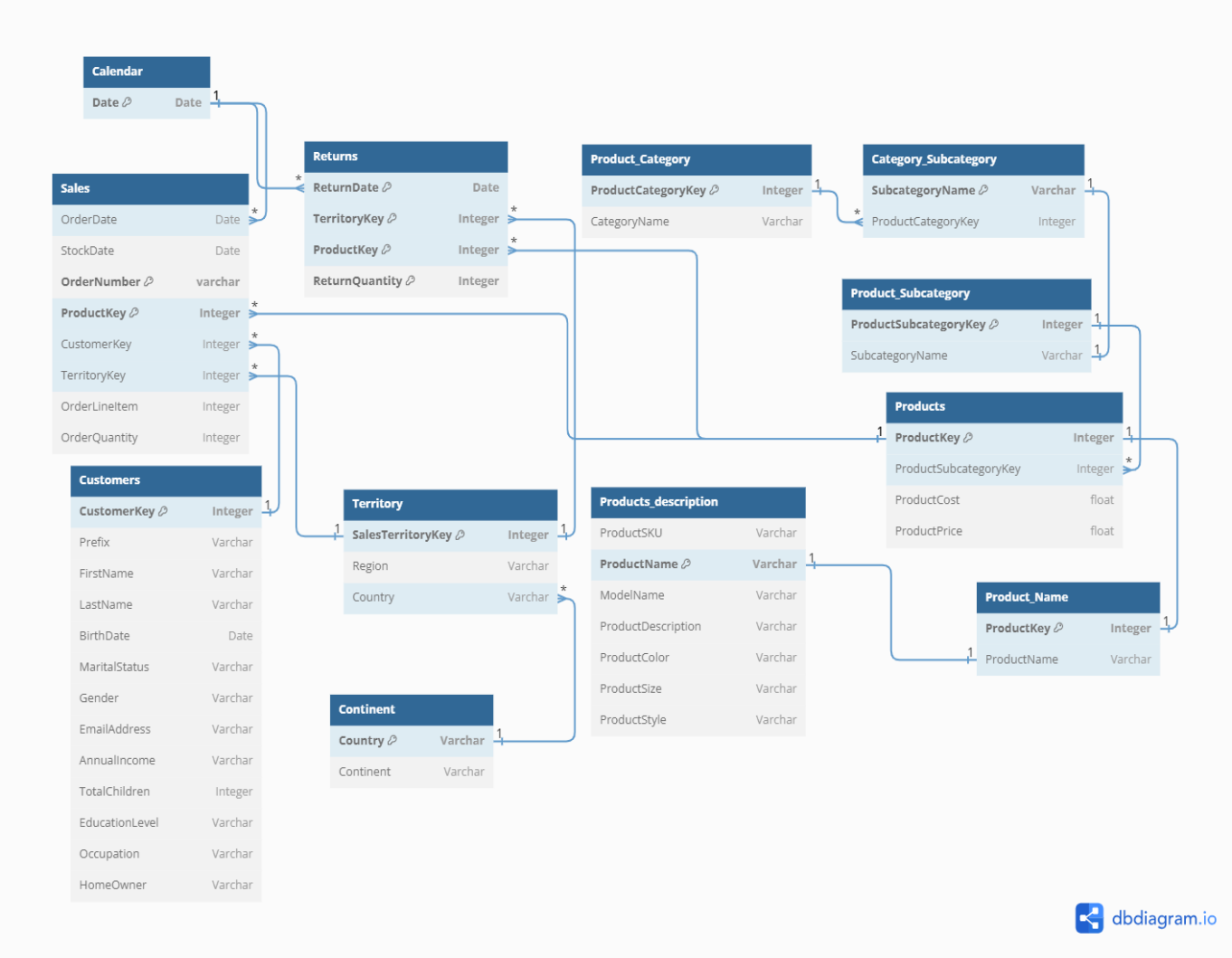

Fig. 2. Final ER diagram

VI. Table Descriptions

TABLE I
TERRITORY

Attributes	Description	Type	Constraint
SalesTerritoryKey	Key to identify a Territory	Integer	Primary Key Not Null
Region	Which part the World Map	Varchar	Not Null
Country	Country	Varchar	Not Null

Functional Dependencies: SalesTerritoryKey $\rightarrow Region, Country$

TABLE II
CONTINENT

S.No	Attributes	Description	Type	Constraint
1	Country	Country	Varchar	Primary Key Not Null
2	Continent	Continent	Varchar	Not Null

Functional Dependencies: Country $\rightarrow Continent$

TABLE III
CALENDAR

S.No	Attributes	Description	Type	Constraint
1	Date	Date	Date	Primary Key Not Null

Functional Dependencies: Date $\rightarrow Date$

TABLE IV
RETURNS

S.No	Attributes	Description	Type	Constraint
1	ReturnDate	Date of Return	Date	Primary Key Not Null
2	TerritoryKey	TerritoryKey	Varchar	Primary Key Not Null
3	ProductKey	ID for each product	Integer	Primary Key Not Null
4	ReturnQuantity	quantity returned	Integer	Primary Key Not Null

All Attributes $\rightarrow AllAttributes$

TABLE V
PRODUCT_SUBCATEGORY

S.No	Attributes	Description	Type	Constraint
1	ProductSubcategory Key	ID for subcategory product	Integer	Primary Key Not Null
2	SubcategoryName	Name of the category	Varchar	Not Null

Functional Dependencies: ProductSubcategoryKey $\rightarrow SubcategoryName$

TABLE VI
CATEGORY_SUBCATEGORY

S.No	Attributes	Description	Type	Constraint
1	SubcategoryName	Name of the category	Varchar	Primary Key Not Null
2	ProductCategoryKey	ID of product category	Varchar	Not Null

Functional Dependencies: SubcategoryName $\rightarrow ProductCategoryKey$

TABLE VII
CUSTOMERS

S.No	Attributes	Description	Type	Constraint
1	CustomerKey	Name of the category	integer	Primary Key Not Null
2	Prefix	ID of product category	Varchar	Not Null
3	FirstName	FirstName	Varchar	Not Null
4	LastName	LastName	Varchar	Not Null
5	BirthDate	date of birth	Date	Not Null
6	MaritalStatus	Married or un married	Varchar	Not Null
7	Gender	Gender	Varchar	Not Null
8	EmailAddress	EmailAddress	Varchar	Not Null
9	AnnualIncome	yearly income	Varchar	Not Null
10	TotalChildren	TotalChildren	Integer	Not Null
11	EducationLevel	level of education	Varchar	Not Null
12	Occupation	Occupation	Varchar	Not Null
13	HomeOwner	Owner or not	Varchar	Not Null

CustomerKey $\rightarrow remaining - all - attributes$

TABLE VIII
PRODUCT_CATEGORY

S.No	Attributes	Description	Type	Constraint
1	ProductCategoryKey	ID of the category	Integer	Primary Key Not Null
2	CategoryName	Name of product category	Varchar	Not Null

ProductCategoryKey $\rightarrow CategoryName$

TABLE IX
PRODUCTS

S.No	Attributes	Description	Type	Constraint
1	ProductKey	Name of the category	Varchar	Primary Key Not Null
2	ProductSubcategory Key	ID of product category	Varchar	Not Null
3	ProductCost	cost price of product	integer	Not Null
4	ProductPrice	selling price of product	integer	Not Null

ProductKey $\rightarrow ProductSubcategoryKey, ProductCost,$ *ProductPrice*

TABLE X
PRODUCT_NAME

S.No	Attributes	Description	Type	Constraint
1	ProductKey	Name of the category	Varchar	Primary Key Not Null
2	ProductName	ID of product category	Varchar	Not Null

ProductKey $\rightarrow ProductName$

TABLE XI
PRODUCTS_DESCRIPTION

S.No	Attributes	Description	Type	Constraint
1	ProductName	Name of the category	Varchar	Primary Key Not Null
2	ProductSKU	ID of product category	Varchar	Not Null
3	ModelName	ModelName	Varchar	Not Null
4	ProductDescription	ProductDescription	Varchar	Not Null
5	ProductColor	ProductColor	Varchar	Not Null
6	ProductSize	ProductSize	Varchar	Not Null
7	ProductStyle	ProductStyle	Varchar	Not Null

ProductName $\rightarrow ProductSKU, ModelName, ProductDescription,$ *ProductColor, ProductSize, ProductStyle*

TABLE XII
SALES

S.No	Attributes	Description	Type	Constraint
1	OrderNumber	Name of the category	Varchar	Primary Key Not Null
2	ProductKey	ID of product category	Varchar	Primary Key Not Null
3	OrderDate	date of order	date	Not Null
4	StockDate	date of stock availability	date	Not Null
5	CustomerKey	ID of customer	Integer	Not Null
6	TerritoryKey	TerritoryKey	Integer	Not Null
7	OrderLineItem	OrderLineItem	Integer	Not Null
8	OrderQuantity	Quantity ordered	Integer	Not Null

OrderNumber, ProductKey $\rightarrow OrderDate, StockDate, CustomerKey,$ *TerritoryKey, OrderLineItem, OrderQuantity*

VII. DATABASE NORMALISATION

Normalization process results in a more efficient, maintainable, and reliable database. It helps in reducing data redundancy, easier maintenance and faster query performance.

Below discussed are tables that need to be normalized in the current data.

A. ***Territory*** *(SalesTerritoryKey, Region, Country, Continent)*

- Here are the following Functional Dependencies
- SalesTerritoryKey →Region, Country
- Country →Continent
- Transitive dependencies exits so we divide the relations into

1) ***Territory*** *(SalesTerritoryKey, Region, Country) :*

2) ***Continent*** *(Country, Continent)*

: Now the above two tables are in BCNF

B. ***Product_Subcategory*** *(ProductSubcategoryKey, SubcategoryName, ProductCategoryKey)*

- Here the following Functional Dependencies
- ProductSubcategoryKey →SubcategoryName
- SubcategoryName → ProductCategoryKey
- Transitive dependencies exits so we divide the relations into

1) ***Product_Subcategory****(ProductSubcategoryKey, SubcategoryName):*

2) ***Category_subcategory****(SubcategoryName, ProductCategoryKey)*

: Now the above two tables are in BCNF

C. ***Products*** *(ProductKey, ProductSubcategoryKey, ProductCost, ProductPrice, ProductName, ProductSKU, ProductName, ModelName, ProductDescription, ProductColor, ProductSize, ProductStyle)*

- Here are the following functional dependencies for above relation
- ProductKey →ProductSubcategoryKey, ProductCost, ProductPrice, ProductName
- ProductName → ProductSubcategoryKey, ProductCost, ProductPrice
- ProductName →ProductSKU, ProductName, ModelName, ProductDescription, ProductColor, ProductSize, ProductStyle
- Here transitive dependencies exist
- Dividing it into three tables as follows

1) Products (ProductKey, ProductSubcategoryKey, ProductCost, ProductPrice):

2) Product_Name(ProductKey, ProductName):

3) Products_description(ProductSKU, ProductName, ModelName, ProductDescription, ProductColor, ProductSize, ProductStyle)

: Now all the three tables are in BCNF.

Remaining all the tables are in BCNF and their respective functional dependencies are mention in **Section VI**

VIII. REFERENCES

1) Data Source
2) Lecture Slides
3) DATABASE SYSTEMS The Complete Book Second Edition by Hector Garcia-Molina,Jeffrey D. Ullman,Jennifer Widom