**Dimensional Model Documentation**

**1. Introduction:**

During this part of the FlipKart Data Warehouse project, I created and set up a dimensional model using the standardized database that I had already made. The dimensional model uses a star schema design pattern, which is very helpful for business data and analytical processing. I turned the normalized structure into a model that works best for complicated queries, reporting, and data analysis by making smart design choices.

**2. Dimensional Model Overview:**

There are six dimension tables and two fact tables in the dimensional model. The facts have the business measures and foreign keys to the dimensions. The dimensions hold the main business units and descriptive attributes. This layout makes it easy to find your way around the data and runs queries quickly.

A screenshot of a computer

AI-generated content may be incorrect.

2.1 Verification of Fact Tables and Dimensions

**3. Star Schema Design:**

For this project, I chose a star schema method because it achieves the best balance between query speed and ease of understanding. In this plan:

* The fact tables are in the middle of the model.
* Like the points on a star, dimension tables go around the facts.
* Surrogate keys show how facts and measurements are related to each other.
* Dimensions are denormalized to improve the speed of queries.

The following database diagram makes the star system design clear:

A screenshot of a computer flow chart

AI-generated content may be incorrect.

3.1 Star Schema Diagram with Dimensions and Fact tables

A diagram of a data warehouse

AI-generated content may be incorrect.

3.2 Visualized Star Schema Diagram

**4. Dimension Tables Creation and Design:**

It was carefully planned that each dimension table would support analytical queries while keeping the purity of the data. Through ETL processes like data transformation, enrichment, and denormalization, the dimensions were generated from the normalized database tables.

a. Customer Dimension:

The Customer dimension has information about the types of customers and derived attributes that can be used for business research.

A screenshot of a computer

AI-generated content may be incorrect.

4.1 Verifying Customer Dimension

We achieved the following:

* SCD Type 2 was put in place to keep track of changes to customer data over time.
* Based on trust scores, the derived CustomerSegment attribute sorts customers into three groups: Premium, Standard, and Basic.
* Keeps the original key (CustomerID) but adds a substitute key (CustomerSK).
* Added all the necessary customer information from the original standardized Customers table

b. Product Dimension:

Information about products and categories are put together in a denormalized way in the Product dimension.

A screenshot of a computer

AI-generated content may be incorrect.

4.2 Verifying Product Dimension

We achieved the following:

* Set up SCD Type 2 to keep track of changes to products
* Category data from the different Categories table that has been denormalized
* Added a derived StockStatus attribute to mark the amount of inventory
* Both the product key and the category key were saved for future use.

c. Location Dimension:

The Location dimension gives you a way to look at sales by area based on their location.

A screenshot of a computer

AI-generated content may be incorrect.

4.3 Verifying Location Dimension

We achieved the following:

* Set up SCD Type 2 to keep track of changes in position
* Added the derived Region property to group locations by
* Maintained the order of geography (Country > State > City)

d. Date Dimension:

For time-based analysis, the Date dimension gives you characteristics that are based on time.

A screenshot of a computer

AI-generated content may be incorrect.

4.4 Verifying Date Dimension

We achieved the following:

* Made a full range of dates to help with study of the past and the future
* Date traits like day of the week, month, quarter, and year have been added.
* Added special date flags for weekends and holidays
* Used a surrogate key based on dates to make joins go quickly.

Similarly, we created other 2 Dimensions: **Payment Method Dimension** and Shipping Method Dimension.

A screenshot of a computer

AI-generated content may be incorrect.

4.5 Verifying Payment Method and Shipping Method Dimension

**5. Fact Tables Creation and Design:**

I made two fact tables with different analytical needs in mind. Each one connects to more than one dimension using "surrogate keys."

a. Sales Fact Table:

There are transaction-level data with measures that are useful for sales research in the Sales fact table.

A screenshot of a computer

AI-generated content may be incorrect.

5.1 Verifying Fact Sales Tables

We achieved the following:

* Set up a transaction grain (the most specific level)
* Measurements were given for number, sales amount, and discount.
* Connected to all factors for a full analysis
* Used more than one date (order date and ship date)

b. Customer Purchase Behavior Fact Table:

This fact table shows how customers usually buy different types of products.

A screenshot of a computer

AI-generated content may be incorrect.

5.2 Verifying Customer Purchase Behavior Fact Table

We achieved the following:

* Set up a customer grain (one row for each customer).
* Added measures of spending by type of product
* Extra steps were added for various methods of purchase
* Linked to measurements that were agreed upon (Customer and Date)

**6. Explanation of How Dimensions Were Derived from Normalized Sources:**

In the dimensional modeling step, the process of getting dimensions from normalized source tables was a big change. This wasn't an easy one-to-one mapping; it involved a lot of complicated changes, like making a surrogate key, denormalizing, deriving attributes, and putting in place a slowly changing dimension methodology. Let me go over this process in more depth for each measure.

A. Customer Dimension Derivation

One of the most difficult parts of our dimensional model is the Customer dimension. It mostly came from the standardized DB\_FlipKart.dbo.Customers table, but it was changed in important ways:

A screenshot of a computer

AI-generated content may be incorrect.

6.1 Verify Customer Dimension Derivation

A screenshot of a computer

AI-generated content may be incorrect.

6.2 Verify Customer segmentation derivation

The transformation involved:

* Holding on to all data information from the source
* Adding tracking fields for SCD Type 2 (EffectiveDate, ExpirationDate, and IsCurrent)
* Using business logic to turn customer loyalty scores into useful groups
* Making a substitute key that is different from the normal key
* Setting up the system to keep track of changes made to customer traits over time

B. Product Dimension Derivation

The Product dimension was made by denormalizing data from two normalized tables, DB\_FlipKart.dbo.Products and DB\_FlipKart.dbo.Categories, and then putting them together. This change got rid of the need for joins in analytical queries while keeping the link in a hierarchy:

A screenshot of a computer

AI-generated content may be incorrect.

6.3 Verification of Product dimension denormalization

A screenshot of a computer

AI-generated content may be incorrect.

6.4 Verify Stock Status derivation

The transformation involved:

* Putting together facts about a product and its category
* Getting rid of the need for a foreign key link during query time
* Using the SCD Type 2 way to keep track of the past
* Making substitute keys for fact table joins that work better

C. Location Dimension Derivation

The Location dimension was Derived from the normalized DB\_FlipKart.dbo.Locations table and shows regional information. This change was mostly about adding useful business classifications to geographic data:

A screenshot of a computer

AI-generated content may be incorrect.

6.5 Verification of Location dimension derivation

A screenshot of a computer

AI-generated content may be incorrect.

6.6 Verifying Location Hierarchy and Region Derivation

Transformation involved:

* Geographical order was kept (Country > State > City).
* Added a derived Region property that is based on how countries are grouped.
* It now has SCD Type 2 traits.
* Made a substitute key for dimension links

D. Date Dimension Creation

The Date dimension wasn't drawn from a normalized source like the others; instead, it was made using a set of steps. This method is often used for date measurements because they need certain calculations and attributes:

A screenshot of a computer

AI-generated content may be incorrect.

6.7 Date Dimension Verification

Changes that were made:

* Date records were made for a certain set of dates.
* The day, month, quarter, and year of the date were calculated.
* Added special colors for holidays and weekends
* Made a substitute key based on dates

E. Payment and Shipping Method Dimensions

The different numbers in the OrderDetails table were used to get these dimensions.

A screenshot of a computer

AI-generated content may be incorrect.

6.8 Verifying Payment Method Derivation

A screenshot of a computer

AI-generated content may be incorrect.

6.9 Verifying Shipping Method Derivation

Transformation involved:

* Getting unique method values
* Made substitute keys
* Original method names were kept.

I successfully changed a normalized database that was best for transaction processing into a dimensional model that was best for analysis queries by using these changes.   
Checking with the verification queries makes sure that all dimensions were correctly obtained from normalized sources while keeping the integrity of the data. Beyond the source data, each dimension now has business-meaningful characteristics that weren't there in the original normalized form. This adds a lot of analytical value.

**7. Dimensional Bus Matrix**

Below is the bus matrix showing the relationship between fact tables and dimensions in our FlipKart data warehouse:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Fact Table/ Dimension** | **Customer \*** | **Product** | **Location** | **Date\*** | **Payment Method** | **Shipping Method** |
| **fact\_Sales** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **fact\_CustomerPurchaseBehavior** | ✓ |  |  | ✓ |  |  |

**Conforming Dimensions(\*):**

1. Customer\*: This is shared between both fact tables so that customer-centered analysis can be done across all business activities.

2. Date\*: This is used by both fact tables to look at data over time and see trends.

**Division-Specific Dimensions**

1. Product—Used only for analyzing sales transactions
2. Location—Only supports sales research by geography
3. Payment Method: This is only used for sales.
4. Shipping Method—Only for sales deals

**Hierarchical Relationships**

1. **Product**: Category → Subcategory → Product
2. **Location**: Region → Country → State → City
3. **Date**: Year → Quarter → Month → Day

This bus matrix is like a plan for our dimensional model. It shows which dimensions can be used for each business process and which dimensions can be used to analyze multiple fact tables together.

**8. Conclusion**

From our normalized database, the dimensional model implementation met all needs and created a complex analytical structure. Six dimension tables (Customer, Product, Location, Date, PaymentMethod, ShippingMethod) and two fact tables (fact\_Sales, fact\_CustomerPurchaseBehavior) were connected by surrogate keys in a star structure.

The model uses Slowly Changing Dimension Type 2 for historical tracking, conformed dimensions for cross-process analysis, and business-value-added derived characteristics. The bus matrix shows dimension-fact relationships, whereas the DDL scripts create the structure. From normalized sources, all dimensions were carefully derived using transformation techniques that kept data integrity and optimized for analytical queries.  
  
The dimension tables include customer segmentation, product stock status, and geographic regions that were not standardized. Referential integrity with dimensions is maintained in fact tables while storing relevant analysis measures. This entire transformation from operational to analytical data model provides a solid platform for business intelligence reporting and analysis of FlipKart e-commerce data.