**Course – Data Modeling Project**

**Title**- Data Co Supply Chain Datawarehouse Bus Architecture

***Program*** *- Business Intelligence and Systems Infrastructure*

***College*** *- Algonquin College of Arts & Technology* Ottawa, Ontario

**Team Members**

*Abdul Haseeb Mohammed  
Md Irteza Chowdhury*

*Mohammed Abdel Salam*

*John Tacco-Melendez*

*Adhiraj Kedar Pustake*

**Abstract:**

Supply Chain Management involves managing multiple business process to ensure smooth functioning of the business operations by tracking processes such as Payment, Shipment, managing inventory level, making decisions based on product performance and simple sales tracking and management. By creating a unified data warehouse bus architecture on this process, a larger view can be established on the running of the business.

**Data Source:**

An open-source dataset has been selected named “DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS” to create a data model for reporting and analysis on Supply chain management published on the site Mendeley Data by Fabian Constante, Fernando Silva, and António Pereira.

**Data Dictionary:**

|  |  |
| --- | --- |
| FIELDS | DESCRIPTION |
| Type | : Type of transaction made |
| Days for shipping (real) | : Actual shipping days of the purchased product |
| Days for shipment (scheduled) | : Days of scheduled delivery of the purchased product |
| Benefit per order | : Earnings per order placed |
| Sales per customer | : Total sales per customer made per customer |
| Delivery Status | : Delivery status of orders: Advance shipping , Late delivery , Shipping canceled , Shipping on time |
| Late\_delivery\_risk | : Categorical variable that indicates if sending is late (1), it is not late (0). |
| Category Id | : Product category code |
| Category Name | : Description of the product category |
| Customer City | : City where the customer made the purchase |
| Customer Country | : Country where the customer made the purchase |
| Customer Email | : Customer's email |
| Customer Fname | : Customer name |
| Customer Id | : Customer ID |
| Customer Lname | : Customer lastname |
| Customer Password | : Masked customer key |
| Customer Segment | : Types of Customers: Consumer , Corporate , Home Office |
| Customer State | : State to which the store where the purchase is registered belongs |
| Customer Street | : Street to which the store where the purchase is registered belongs |
| Customer Zipcode | : Customer Zipcode |
| Department Id | : Department code of store |
| Department Name | : Department name of store |
| Latitude | : Latitude corresponding to location of store |
| Longitude | : Longitude corresponding to location of store |
| Market | : Market to where the order is delivered : Africa , Europe , LATAM , Pacific Asia , USCA |
| Order City | : Destination city of the order |
| Order Country | : Destination country of the order |
| Order Customer Id | : Customer order code |
| order date (DateOrders) | : Date on which the order is made |
| Order Id | : Order code |
| Order Item Cardprod Id | : Product code generated through the RFID reader |
| Order Item Discount | : Order item discount value |
| Order Item Discount Rate | : Order item discount percentage |
| Order Item Id | : Order item code |
| Order Item Product Price | : Price of products without discount |
| Order Item Profit Ratio | : Order Item Profit Ratio |
| Order Item Quantity | : Number of products per order |
| Sales | : Value in sales |
| Order Item Total | : Total amount per order |
| Order Profit Per Order | : Order Profit Per Order |
| Order Region | : Region of the world where the order is delivered : Southeast Asia ,South Asia ,Oceania ,Eastern Asia, West Asia , West of USA , US Center , West Africa, Central Africa ,North Africa ,Western Europe ,Northern , Caribbean , South America ,East Africa ,Southern Europe , East of USA ,Canada ,Southern Africa , Central Asia , Europe , Central America, Eastern Europe , South of USA |
| Order State | : State of the region where the order is delivered |
| Order Status | : Order Status : COMPLETE , PENDING , CLOSED , PENDING\_PAYMENT ,CANCELED , PROCESSING ,SUSPECTED\_FRAUD ,ON\_HOLD ,PAYMENT\_REVIEW |
| Product Card Id | : Product code |
| Product Category Id | : Product category code |
| Product Description | : Product Description |
| Product Image | : Link of visit and purchase of the product |
| Product Name | : Product Name |
| Product Price | : Product Price |
| Product Status | : Status of the product stock :If it is 1 not available , 0 the product is available |
| Shipping date (DateOrders) | : Exact date and time of shipment |
| Shipping Mode | : The following shipping modes are presented : Standard Class , First Class , Second Class , Same Day |

**Executive Summary:**

**Sales Process:**

**A screenshot of a computer

Description automatically generated**

Fact Sales

The Sales Process is a transaction fact table with order id as granularity. Each record corresponds to one order. The original dataset has multiple records for one order, so it was grouped by order id to find summarized metrics for each order. As for Categorical features the first value was taken for each order except for order status the final order status was retained for Sales Process.

A screenshot of a computer

Description automatically generated

**Sales Process Dimensions**

The Sales Process has seven dimensions some of which are conformed dimensions with other business process. One piece of information to be pointed out is the location table has location of Customer at the time of transaction, location of supplier company, Location of department and location of delivery. Though the sales process only has a relationship with customer location and department location and to find market where the order is for an additional relationship with delivery can be considered. Finally, a prefix of O was attached to order id, C to customer id, S to supplier id and D to department ID as these id’s have been stored in location table.

**Data Model for Sales Process:**

**A screenshot of a computer

Description automatically generated**

**Sales Process Dashboard:**

**A screenshot of a computer

Description automatically generated**

**Payment Tracking Process:**

**A screenshot of a computer

Description automatically generated**

**Fact Payment Tracking**

The Fact Payment Tracking table is an accumulated fact table where each record corresponds to one state of a payment. The original dataset had the order date. A python script was used to generate records showing multiple states of a payment. This was accomplished by using random time difference in the flow of transactions from one state to another having a new record. Like the sales process, the first value was taken for all the categorical features. Additionally, for accurate visualization and reporting one measure was created for each state of payment. Two approaches are possible to find number of payments in each state first is using a Latest State Flag calculated column which finds the latest records for each payment using order id and date. The latest state would be the most recent record for each order which can be found using max date for each order id. The second approach is to find the difference in number of payments in the next most state and current state. For instance, the pending. To find the number of payments in the pending state a count of records in pending state can be found and a difference can be found from count of records in processing state which would give number of payments currently in pending state. The measure columns are given a green background to discern them from table columns.

A screenshot of a computer

Description automatically generated

**Dimensions for Payment Tracking**

The payment tracking has multiple conformed dimensions and like sales process a prefix was attached to ID columns used in location table.

**Data Model for Payment Tracking:**

A screenshot of a computer

Description automatically generated

**Payment Tracking Dashboard:**

A screenshot of a computer

Description automatically generated

**Shipment Tracking Process:**

A screenshot of a logistic

Description automatically generated

**Fact Shipment Tracking**

The Fact Shipment Tracking table is an accumulated fact table where each record corresponds to one state of shipment. The original dataset had the order date and ship date. A python script was used to generate records for multiple states of shipment. The logic used for generating accurate dates is generating dates between order date and ship date. Additionally, when the shipping mode is the same day the time difference is split appropriately within a day. And when the shipping mode is not the same day the time difference is simply split randomly using the number of states for a shipment and between order date and ship date. Like the sales process, the first value was taken for all the categorical features. Additionally, for accurate visualization and reporting one measure was created for each state of shipment. Two approaches are possible to find number of shipments in each state first is using a Latest State Flag calculated column which finds the latest records for each shipment using order id and date. The latest state would be the most recent record for each shipment which can be found using max date for each order id. The second approach is to find the difference in number of shipments in the next most state and current state. For instance, the Processing state. To find the number of shipments in the processing state a count of records in processing state can be found and a difference can be found from count of records in transit state which would give number of shipments currently in processing state. The measure columns are given a green background to discern them from table columns.

A screenshot of a computer

Description automatically generated

**Dimensions for Shipment Tracking**

The shipment tracking has multiple conformed dimensions and like sales process a prefix was attached to ID columns used in location table.

**Data Model for Shipment Tracking:**

**A screenshot of a computer

Description automatically generated**

**Shipment Tracking Dashboard:**

A screenshot of a computer

Description automatically generated

**Product Performance Process:**

A screenshot of a computer

Description automatically generated

**Fact Product Performance**

The Product Performance fact table is a periodic snapshot with monthly granularity. To explain clearly, the grouping let me explain the business assumptions. We have one store with multiple departments where each department sells products. For instance, in the data we have a Fitness department which would sell products for customers who buy fitness-related products. The product performance fact table was constructed by grouping the dataset using Product ID, Department ID, Year and month. This way every record in product performance fact table would have monthly metrics for a product in different departments if its sold in different departments. The main metrics of this fact table are Units sold, amount discounted, Total Sales (After Discount), Total Sales (Before Discount), Order Count [which is number of times a product was involved in different orders], and total profit.

A screenshot of a product list

Description automatically generated

**Dimensions for Fact Product Performance**

**Data Model for Product Performance:**

**A screenshot of a computer

Description automatically generated**

**Product Performance Dashboard:**

**A screenshot of a computer

Description automatically generated**

**Inventory Management Process:**

**A screenshot of a receipt

Description automatically generated**

**Fact Inventory Management**

The Inventory Management fact table is a periodic snapshot with daily granularity. To explain clearly, the grouping let me explain the business assumptions. We have one store with multiple departments where each department sells products and tracks daily inventory levels. The Inventory Management fact table was constructed by grouping the dataset using Product ID, Department ID, and Date first, then an initial in stock quantity was generated for each product within each department and using quantity of items ordered the in-stock amount is updated and when in stock amount goes below a certain threshold a supplier would bring new stock in a random amount. For instance, in the data we have a Fitness department which would sell products to customers who buy fitness-related products and also track in stock quantity of each product in the inventory, quantity which is how many units of product go out of the inventory along with replenish amount that is units of product brought by the supplier and a flag column called New Stock Arrival is used to indicate when new stock arrives. This way every record in the Inventory Management fact table would have daily metrics for a product in different departments that is in stock quantity, replenish amount, quantity of units of products going out of the inventory. Also, a dedicated supplier is assigned to each department. The suppliers are from different countries across the world.

**A screenshot of a document

Description automatically generated**

**Dimensions for Fact Inventory Management**

**Data Model for Fact Inventory Management:**

**A screenshot of a computer

Description automatically generated**

**Fact Inventory Management Dashboard:**

**A screenshot of a computer

Description automatically generated**

**Final Data Model for Supply Chain Management using Data warehouse Bus Architecture:**

**A screenshot of a computer screen

Description automatically generated**

**Conclusion:**

In conclusion, The Data warehouse bus architecture and the individual dashboard would help managing the various business processes by reporting accurate information.

**Recommendations:**

The product performance process can be further enhanced by adding more metrics such as product performance in different locations.

**References:**

<https://data.mendeley.com/datasets/8gx2fvg2k6/5>

[1] F. Constante, F. Silva, and A. Pereira, “DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS,” *data.mendeley.com*, vol. 5, Mar. 2019, doi: https://doi.org/10.17632/8gx2fvg2k6.5.

‌