Project Report

Real time data warehouse using Kimball model on Superstore Sales data

INTRODUCTION

The project was extended from the term paper title "*Inmon vs Kimball: A Comparative Study on Datawarehouse Modelling"* . In the project work, we create a Star Schema and generate the data warehouse according to principles given by Kimball, using an open source data integration and analysis platform Pentaho. Our project is divided into three major steps of creating the data warehouse by integrating it and performing some preprocessing on data. Then creating the OLAP Cubes and finally creating a dashboard, showing some simple analytics using the generated cube. Below we have highlighted the important steps in all the three parts.

PART 1 : *Data warehouse generation using Pentaho Data integration tool*

(Pentaho data integration tool link - https://sourceforge.net/projects/pentaho/)

PDI tool provides a great option for ETL process to create/manage a data warehouse with dimensions and fact tables linked in star schema. We have built our Data warehouse in a local database, PostgreSQL. The following are the details of how the data was.

Order Code	Ship Mode	Region	Product Container
Order Date	Profit	Customer Segment	Product Base Margin
Order Priority	Unit Price	Product Code	Ship Date
Order Quantity	Shipping Cost	Product Category	Row ID
Sales	Customer Code	Product Sub-Category	Province
Discount	Customer Name	Product Name	

CREATING DIMENSIONS

There are a total of 23 columns in the original data. Now below, we run through an example of creating a dimension, using the original data. Here we have shown how the customer dimension has been created.

First a staging area is created for Customer columns, that is a table is generated by selecting required columns (Customer Code;Customer Name;Province;Region;Customer Segment) from main data. Then a process is created which generates a

new table, named customer_dim which has columns as above and a primary key that identifies a row uniquely. Like this all other dimensions (date_dim, order_dim, product_dim) are created.

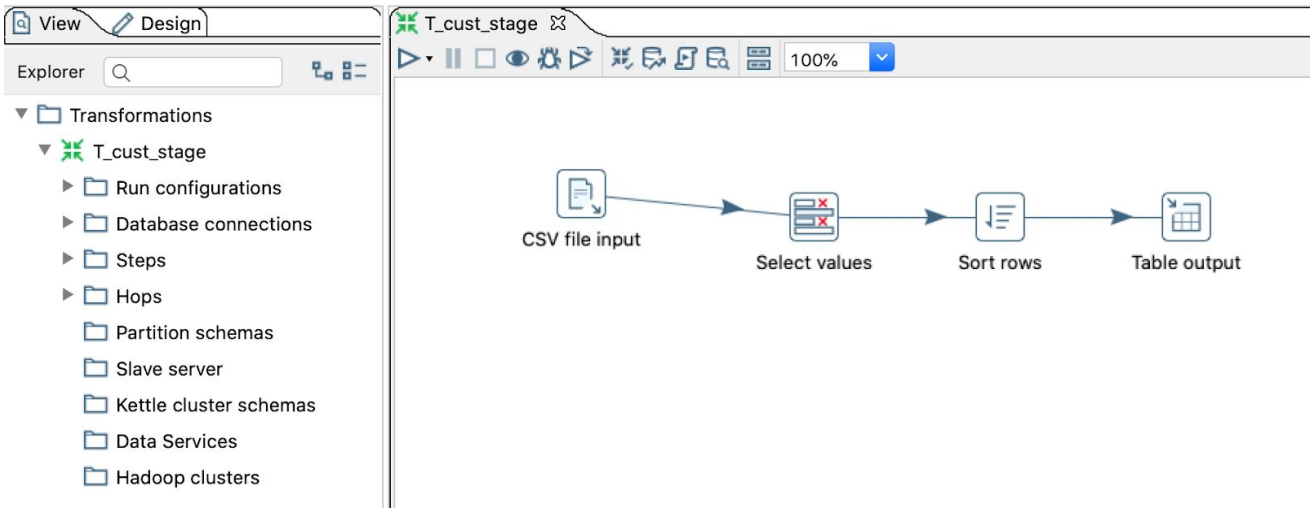


Figure 1 : Staging area for customer dimension in Data integration tool

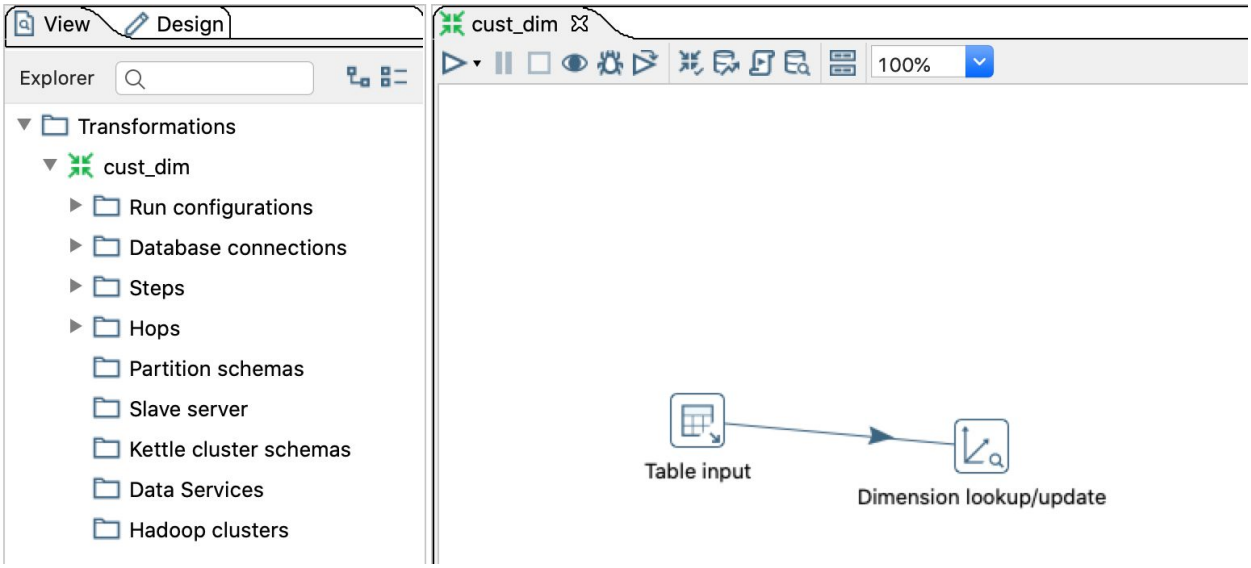


Figure 2 - Customer dimension generation

CREATING A FACT TABLE

A staging area is created, that is a table with selected values as columns from main data (Order Code, Order Date, Order Quantity, Sales, Discount, Profit, Unit Price, Shipping Cost, Customer Code, Product Code, Product Base Margin). Then a process is created to match each row from the above table to PK of dimensions and add all to a new fact_table.

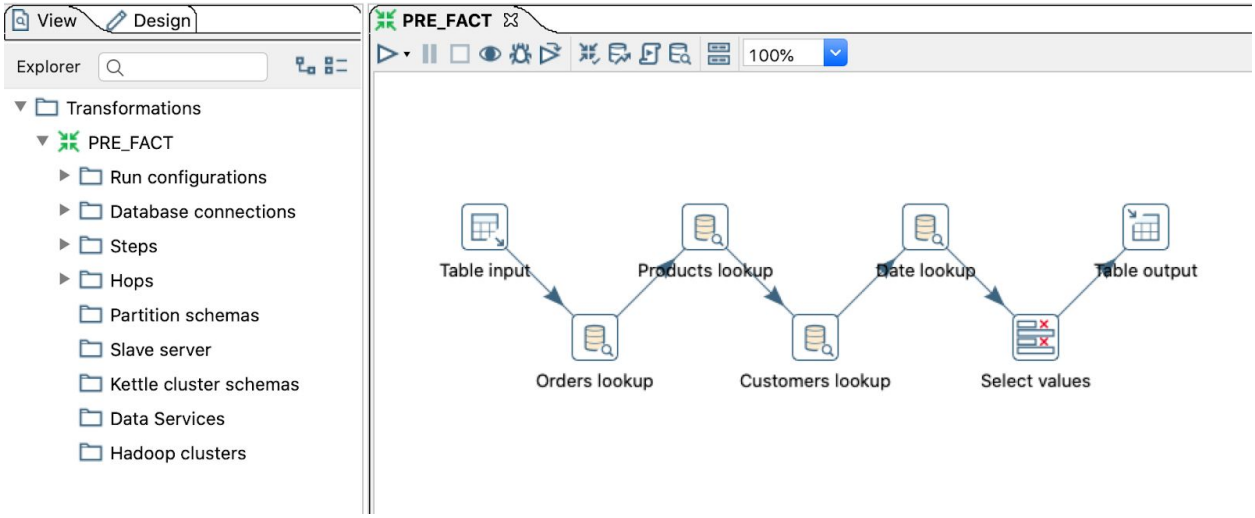


Figure 3 - Fact_table generation

A star schema is now ready in a local database, as shown in figure 4.

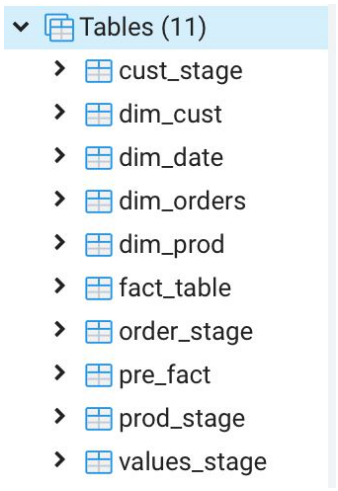


Figure 4 - Tables in local database (Postgre SQL)

This completes step 1. Now using this schema, we create OLAP cubes, as discussed in the next section.

PART 2: *OLAP cubes from warehouse using Mondrian Schema Workbench*

Once the star schema was created, Mondrian Schema workbench was used to access the database for creating the OLAP cubes from it. The following figure shows the different dimensions and measures that were created. Each dimension has a different level of hierarchy that will be used for drilling down, for example, Customer has Customer type at highest level and names in each type as next level. A dimension can also have two different hierarchies. For examples,

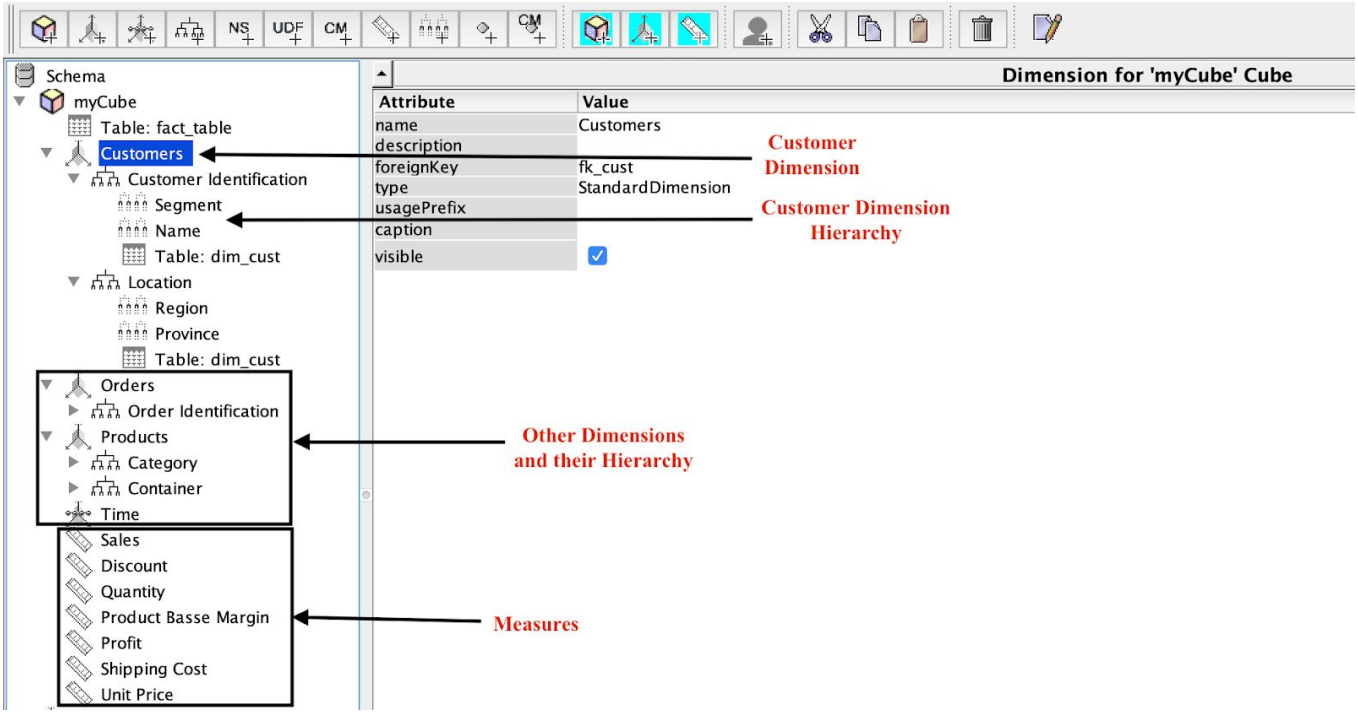


Figure 5: OLAP Cube in Mondrian

PART 3 : *Querying cubes using Pentaho Business Analytics & Pivot4J tool*

Finally when the Cube was ready, it was processed for different operations, using Pivot4J plugin, in Pentaho BI Server. Pivot4J is an analytics tool, which queries the cube using MDX query. It's a simple, easy-to-use plugin, also having drag and drop feature. In the cube, we have measures: Sales, Discount, Quantity, Product Base Margin, Profit, Shipping Cost, Unit Price. And four dimensions: Customers, Orders, Products, and Time.

Each of these measures and dimensions can be dragged and dropped to form an analysis table. For ex: Drag Sales measure on column structure and Customer dimension + Product dimension on rows structure, a table is automatically generated as shown below:

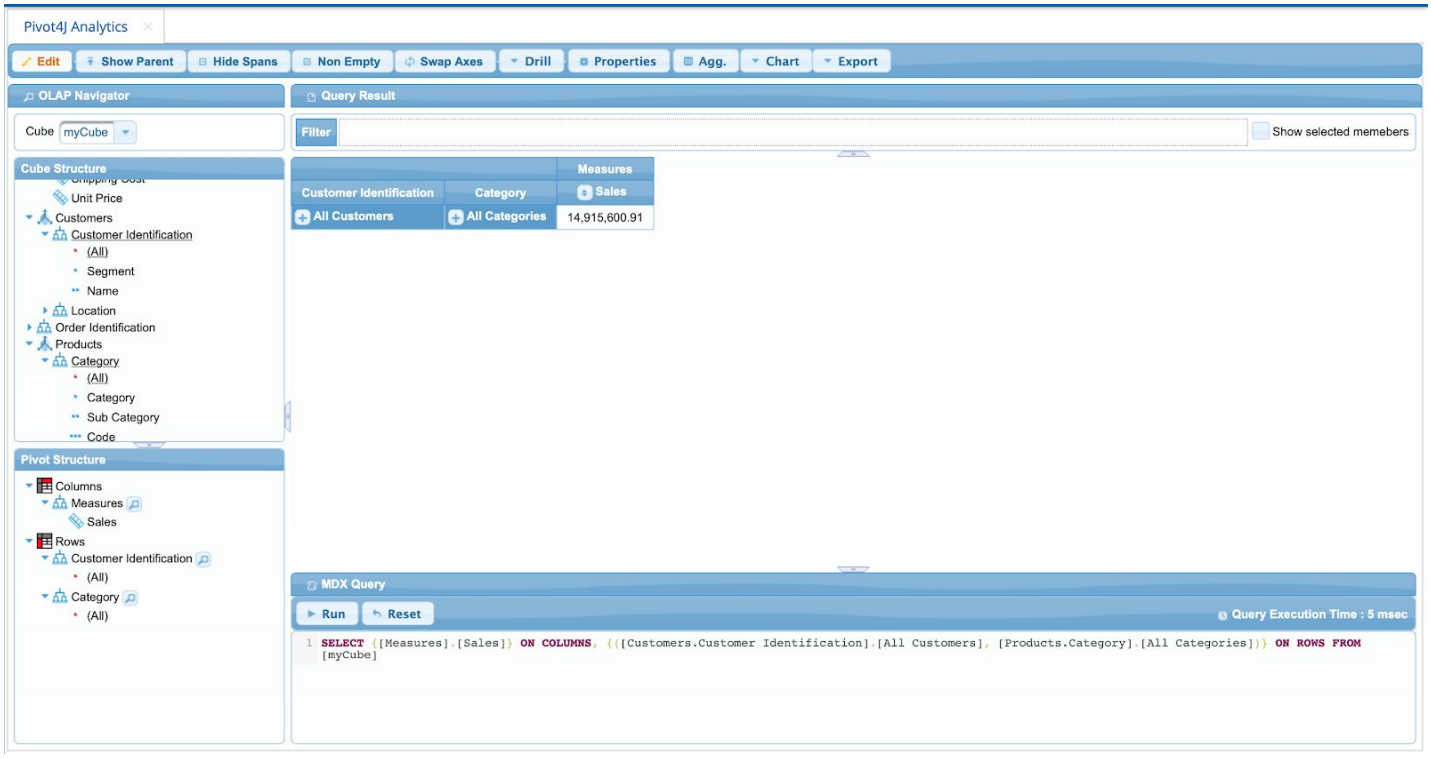


Figure 6: Pivot4J analysis on Cube

Here Total sales is shown with respect to all customers and all product categories they bought. Customer dimension can be drilled down to customer type by clicking on "+":

		Measures
Customer Identification	Category	Sales
All Customers	All Categories	14,915,600.91
Consumer	All Categories	3,063,611.1
Corporate	All Categories	5,498,904.85
Home Office	All Categories	3,564,763.97
Small Business	All Categories	2,788,320.99

Figure 7: Drill down on Customer segment

It can be further drilled down to product category type and sub category:

		Measures
Customer Identification	Category	Sales
All Customers	All Categories	14,915,600.91
Consumer	All Categories	3,063,611.1
	Furniture	1,128,807.21
	Bookcases	184,419.73
	Chairs & Chairmats	374,635.08
	Office Furnishings	127,840.03
	Tables	441,912.37
	Office Supplies	691,382.23
	Technology	1,243,421.66
Corporate	All Categories	5,498,904.85
Home Office	All Categories	3,564,763.97
Small Business	All Categories	2,788,320.99

Figure 8: Drilled down on customer type and product type

Automatic mdx query is also generated while drag and drop is executed:

MDX Query

Run Reset Query Execution Time : 37 msec

```
SELECT NON EMPTY {[Measures].[Sales]} ON COLUMNS, NON EMPTY Hierarchize(Union(Union({{[Customers.Customer Identification].[All Customers],
[Products.Category].[All Categories]}}, CrossJoin({[Customers.Customer Identification].[All Customers].Children, {[Products.Category].[All
Categories]}})), Union(CrossJoin({[Customers.Customer Identification].[Consumer]}, [Products.Category].[All Categories].Children),
CrossJoin({[Customers.Customer Identification].[Consumer]}, [Products.Category].[Furniture].Children)))) ON ROWS FROM [myCube]
```

Figure 9 : MDX query of figure 8 table on Cube

Also different charts can be prepared for current snapshot of the table:

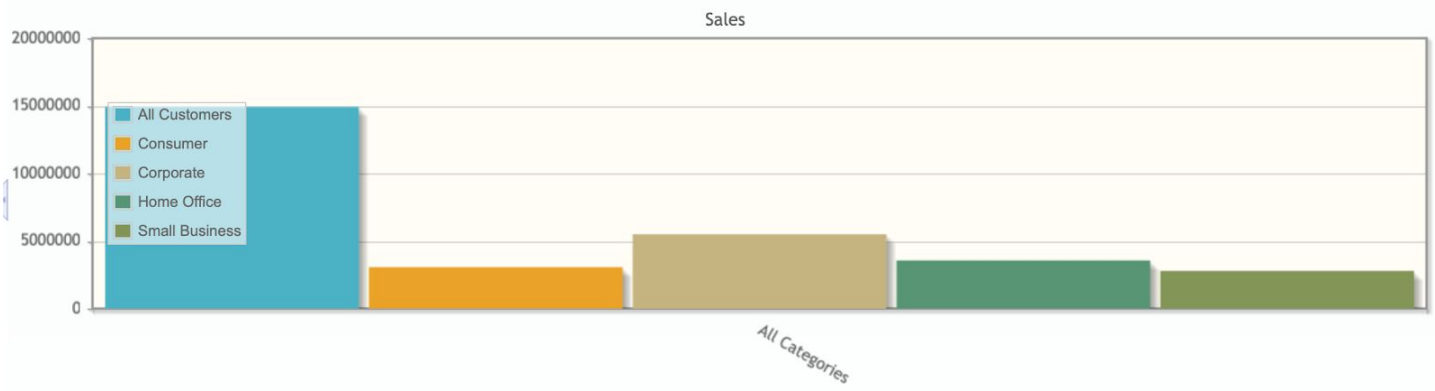


Figure 10 : Bar chart for figure 8

CONCLUSION

In conclusion, by this project work, we integrated a huge amount of data from a Super Store, using Pentaho, an open source data integration and analytics tool. The data was further processed to create 4 dimensions, viz. Product, Order, Customer and Time. Using these dimensions, we have also created a fact table, with 7 measures viz. Sales, Discount, Quantity, Product Base Margin, Profit, Shipping Cost, Unit Price. Further we convert this star schema into an OLAP cube for doing analysis. The OLAP cube was created in Mondrian. Finally, various operations like drill down, creating Bar charts were carried out on this OLAP cube. Thus, we were successfully able to create a data warehouse using the Kimball Model.

Inmon vs Kimball
A Comparative Study on Datawarehouse Modelling

Rachit Rathore
Department of Computer Science and Information Systems
Birla Institute of Technology and Science
Pilani, India
h20150600@pilani.bits-pilani.ac.in

Aditya Mehta
Department of Computer Science and Information Systems
Birla Institute of Technology and Science
Pilani, India
h20150808@pilani.bits-pilani.ac.in

***Abstract*—We are living in the era of a digital revolution, and more and more companies are recognizing that they need to leverage their digital resources to lead or survive efficiently. The data warehouse plays an even more critical role in this scenario, owing to its unique approach as the integrated business data repository. There are two influential types of architecture used today to build a data warehouse: architecture in Inmon and architecture in Kimball. This paper aims to compare and contrast the advantages and drawbacks of each architectural style and to suggest the style to undertake based on various important factors.**

***Index Terms*—Kimball, Inmon, data warehouse, ETL**

I. Introduction

Nowadays, businesses face many challenges, especially in terms of leveraging and analyzing the operational data that they keep in their heterogeneous sources. The ultimate aim of these activities is to acquire valuable decision-making knowledge. Business Intelligence is an integral part of turning raw data into usable and meaningful expertise that supports the decisions of corporate leaders. An information warehouse is a central element in a decision-making framework. This is the crucial component of the information system aimed at processing operational data from various sources and delivering it to users for analysis in different formats. Building a data warehouse includes a modeling approach that takes into account all technical aspects such as data engineering, project management, risk management, implementation, and many other vital elements. Two methods in data processing for data warehouses have existed for many years: the Inmon Model [1] and the Kimbal Model [2]. Both view the data warehouse as the company's central data center, and both use ETL to load the data warehouse. The critical difference is how to model, load, and store data structures in the data warehouse. This architectural disparity impacts the data warehouse's initial delivery period and the ability to handle potential ETL design changes. In a broad sense, a standardized data model is first designed in Bill Inmon's enterprise data warehouse approach, which is also known as the top-down design, then the dimensional data marts are created from the data warehouse, which contains data required for specific business processes or departments. The data marts supporting reports and analyses are first generated in Ralph Kimball's design approach, which is why it is called the bottom-up style, and these are then combined to create a comprehensive data warehouse.

Debates over which one is better and more productive have been going on for years. However, since all philosophies have their benefits and differentiating factors, a straightforward solution has never been arrived at, so businesses tend to use any of these. However, in making BI decisions, it is crucial to understand the strengths and weaknesses of the two.

In our term paper, we have organized the study as follows. First we discuss in brief, few of the important concepts that are required to understand data warehouse modelling. In next two sections we cover the various aspects of Inmon and Kimball techniques of data warehouse modeling in depth. Following that we summarize the major differences and similarities between the two approaches and finally highlight the key factors that should be considered for choosing one of the approach.

II. Prior Study

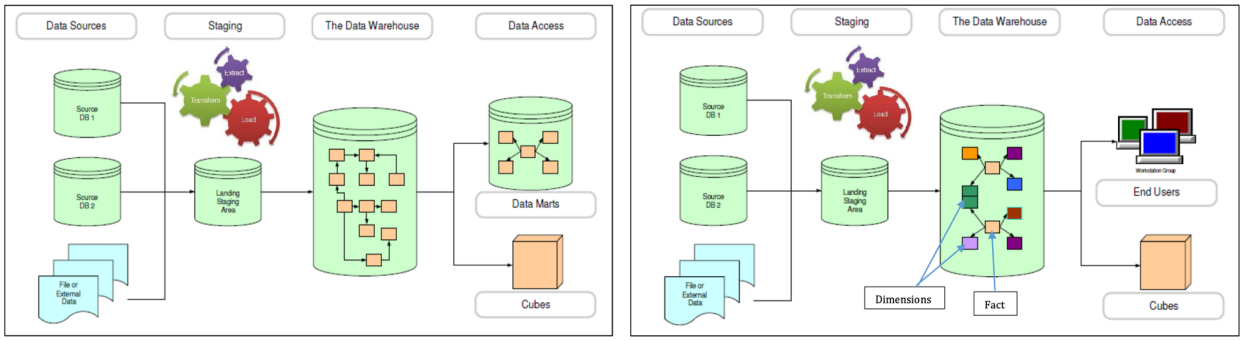
a) Transaction Systems: The enterprise is assisted by operational systems [2] or transaction systems. ransaction systems are used to capture and store data from business transactions such as sales, marketing, etc. in different forms, including relational data, hierarchical data, or spreadsheets. These are also referred to as data source systems that provide data to the warehouse.

b) ETL Process: A method called ETL [2] is used to carry data from the transaction system. ETL stands for extract, transform and load. ETL process consolidates data, transforms it into a specific standard format and loads it into a single repository called a data warehouse.

c) Data Marts: Data marts [2] are departmental views of the subject-oriented knowledge. Data marts collect data from the data storage center. To store data, the data marts use dimensional architecture (Star Schema or Snowflake Schema). Some analytics systems or business intelligence applications explicitly access data from data marts, rather than corporate data warehouse.

III. The Inmon Model

The Inmon approach [2] [4] to developing a data warehouse starts with the creation of a data model that defines main



(a) The Inmon Model

(b) The Kimball Model

Fig. 1: Data Warehouse Modelling Approaches [3]

focus areas, such as Customer, product, supplier, etc. and significant entities for each. For example, a model is built for Customer with all the details pertaining to that entity, and under Customer, there could be ten different entities. The detailed model captures all of the data, including business keys, characteristics, dependencies, engagement, and relationships. This structure of the entity is built in a normalized form; thus, duplication of data is avoided as far as possible. ETL tools are used to extract all these data from different sources and store them into 'atomic' data units, which are the lowest level of representation, that passes through the staging or landing field. The next step is to construct the physical model, which is often standardized. This is a 'data warehouse.' This hierarchical model makes it less complicated to load the data, but it is hard to use this framework to query, as it requires several tables and joins. And data marts are designed exclusively for departments; hence data warehouse is at the Corporate Information Factory (CIF) center. The data marts should be designed specifically for Accounting, Marketing, etc. and the data marts will have de-normalized data to assist in reporting. The data center is the only source of data for the numerous data marts and the enterprise-wide centralized repository. This ensures the quality and accuracy of data across the enterprise is kept intact. This is known as the Top-Down Approach, where 'Atomic' data (data at the lowest level of detail) is first processed in the data warehouse, using ETL tools and dimensional data marts that have all the data required for the specific process, or different divisions are generated from the data warehouses.

IV. The Kimball Model

The Kimball approach [2] [4] to data warehouse construction starts with defining the essential business processes and evaluating and recording the primary data sources (operational systems) for the data warehouse. ETL tool is used to fetch and load various types of data formats from multiple sources into a staging area. From here, data is loaded into a dimensional model, which is not normalized yet. The star-schema is the basic principle of dimensional modeling. There is usually a fact table in the star schema, which contains numeric data and is surrounded by several dimensions, which is the reference information supporting of truth. The fact table has all the related measures for the subject area, and it also has foreign keys from the various dimensions that surround the fact. The measurements are fully denormalized, allowing the user to drill up and drill down without joining to another table. It will develop multiple star schemes to meet various reporting requirements. 'Conformed dimensions' achieve integration in the dimensional model. The key aspects, including customer and product shared across different facts, will be built once and will be used by all other variables. This means that across the facts, one thing or idea is used in the same way. This is how data marts which contain dimensions and facts are created. The single data mart also models a particular business area such as "sales" or "output." Such data marts offer a thin view of the organizational data and can be integrated into more massive data warehouses known as the bus architecture as and when necessary. It is essentially an implementation of "the bus," a set of conformed dimensions, and conformed facts, which are dimensions shared between facts belonging to two or more data marts. This data warehouse design methodology is known as bottom-up, where data marts are first built to provide reporting and analytical capabilities for specific business processes.

V. Comparative Study

A. Similarities between Inmon and Kimball

1) The data warehouse is considered as a central repository that supports business reporting in both approaches.
2) Both approaches use time stamped data. In Inmon the time attribute is called as the "time element" whereas Kimball it is called as "date dimension".
3) Extract transform and load (ETL) process is common in both. The data is first extracted from operational system databases, then transformed to meet the standards of data warehousing architecture and loaded into centralized Data warehouse, in case of Inmon's or Data mart, in Kimball's approach.
4) Data attributes in both are similar and both gives same end user query results.

TABLE I: Difference between Inmon and Kimball modelling approach [5] [6]

Description	Inmon	Kimball
	Development Methodologies	
Approach	Top Down approach	Bottom Up approach
Architecture	Enterprise-wide data warehouse is used to generate department-wise specific data marts.	Data marts are first designed for specific individual business processes and through integration enterprise-wide consistency is achieved.
Complexity	Complex to create enterprise-wide data warehouse comprising of all the key business process.	Fairly simple to construct and later on integrate data marts for each single business processes.
	Data Modelling	
Data Alignment	Subject driven	Process
Data Redundancy	Data is normalised (mostly 3NF) hence redundancy is low.	Highly redundant as data is in denormalized form.
Modelling	ER Model is used for modelling data in enterprise data warehouse, and then data marts are created from it afterwards.	Dimensional Model is used to organize all the data in dimensional data warehouse.
Tools	Relational tools (Entity Relationship Diagrams)	Dimensional tools (Star Schemas or Snowflakes)
Data Updates	Due to low redundancy data update irregularities are minimal and this makes ETL process simpler and less prone to failure.	Data update anomalies are caused over time due to redundant data.
Data Retrieval	Complex because of normalised data. More tables means more number of joins to perform for data retrieval.	Fast data retrieval as data is denormalised and divided into facts and dimensions
	Philosophical	
End User Accessibility	Low as data is in normalised form, hence complex structure, therefore high number of table joins which is not feasible for end users to query directly.	High as data marts for each specific processes are present to be queried directly by analytical systems in reasonable response time.
Audience	IT professionals	End users
	Influential Factors	
Building	Time consuming	Comparatively less time to implement.
Cost	Incur high development costs initially, but the subsequent development costs are low due to low maintenance.	Incurs a low cost initially as we only need to plan and architect the data warehouse and the cost for subsequent phases remain constant.
Maintenance	Easy as low data redundancy.	Difficult because there are redundant data and revisions require additional cost.
Skill Requirement	Specialized team with high technical skills, in large numbers.	Generalized small team is required.
Data Integration	Enterprise wide	Individual business or department wide.
Query Performance	The query performance is slow as data is in 3NF.	Comparatively high response time as data is denormalized hence less number of tables and joins.
Flexible to changes	Very flexible because if there is change in business requirements or source data, it is easy to update it in data warehouse as all are in one location. Overall model architecture is also unaffected by above modifications.	Hard to update data warehouse if there is a change in business requirements or source data as whole dimensional model design is affected. Adding columns to the fact or dimension tables causes performance issues. Also, extra cost to handle redundancy.

B. Differences between Inmon and Kimball

Since differences are a major highlight of this paper, we present a summarized feature-wise comparison of the two models on differentiating factors in Table I

VI. Deciding Factors

As addressed in the previous section, the pros and cons of both Inmon's and Kimball's approaches to constructing a data warehouse, in this section we will illustrate some of the considerations that will help determine which solution is to be used to serve the company better. Some of the important factors are summarized below.

a) Reporting Requirements: If company or organization reporting needs are broad and integrated reporting is needed, then the Inmon model is suitable. The Kimbal model works better if reporting criteria are based on specific business processes or team-oriented, instead.

b) Project Urgency: Developing a normalized model of data is complex and takes longer than a denormalized one. If there is enough time for the company then Inmon model can be followed, otherwise Kimball approach is better.Developing a normalized model of data is complex and takes longer than a denormalized one. If there is enough time for the company then Inmon model can be followed, otherwise Kimball approach is better.

c) Staffing Plan: The higher the complexity of data model is aimed, the more expert skill set is needed. Unless the organization wants to provide a large team of professionals to manage the data center,it better to pursue Kimball model. And if the firm is ready to invest in expert professionals, they can proceed with Inmon model.

d) Frequency of changes: If the reporting criteria are likely to change more rapidly and the data source structures are unpredictable, then the Inmon approach works better, because it is more versatile. If the specifications and data source structures are comparatively stable, then the business can use the Kimball approach.

There are few sectors, which can be analyzed where a pattern is followed for choosing the approach for building a data warehouse. For example, Marketing sector or Insurance sector.

e) Marketing Domain: This is a specialized domain which is precisely looking for a specific area of knowledge for its business. Therefore, enterprise wide data warehouse only will not meet the requirements. It also needs specific data marts and hence Kimball's approach is acceptable.

f) Insurance Domain: It is important to get an overall picture about individual clients, background of claims, mortality rate, agents of clients, demographics, profitability of each scheme, etc. All the topics are inter-related and better suitable for Inmon's approach to data warehouse.

VII. Conclusion

In this paper we presented two modeling methods for building a data warehouse : Inmon and Kimball. Bill Inmon recommends the construction of a data warehouse which follows the top-down approach. This begins with the creation of a large integrated corporate data center where all available data from transaction systems are pooled and structured using ER models, and then data marts are built for departmental analytical needs. Ralph Kimball suggests that the data warehouse be designed which follows the bottom-up approach. It starts with the construction of data marts which serve the needs of specific departments and are directly accessible via analytical systems. Using dimension mapping, data is structured as star schema. The incorporation of these data marts for data consistency (called information bus) then forms a large data warehouse for the entire enterprise. Finally we compared the similarities and differences between them, and deduced some deciding factors that will help to determine the solution to use where it will meet the business needs properly. Data warehouses can be efficiently planned and delivered using both the Inmon and Kimball models. It is not possible to generalize which solution is better, as both have their advantages and disadvantages, and both work well in different circumstances. The data warehouse architect has to select a model based on the different factors such as the business requirement and priorities, the complexity of the business, the time and expense involved, and the degree of dependence between the different functions. Inmon is suitable for businesses that can manage time and expense, where requirements and data sources change frequently as this model do not adjust design with each changing business situation while Kimball is suitable for stable business where requirements don't change frequently. Because of the bottom-up approach, the Kimball model is more flexible and so we can gradually start small and scale-up. On the other hand, the layout of Inmon is more organized as it is simpler to maintain because it is compact (3NF form) although it takes more time to create and it is difficult to query as more joins are required. Implementation of the Kimball data warehouse architecture takes a relatively small amount of time but maintenance is difficult due to redundant data. Implementation of a data warehouse based on Inmon will incur high initial costs. On the other hand, Kimball is incurring low initial costs and the cost for subsequent phases remains the same. A specialized team would be expected to introduce an Inmon-based data warehouse. On the other side, Kimball requires the introduction of a generalist group.

VIII. Acknowledgment

We would like to thank Dr. Avinash Gautam for his supervision and useful guidance in the course of working on this project.

References

[1] W. H. Inmon, *Building the Data Warehouse*. USA: John Wiley Sons, Inc., 1992.

[2] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2nd ed. USA: John Wiley Sons, Inc., 2002.

[3] I. Abramson, "Data warehouse: The choice of inmon versus kimball," *IAS Inc*, 2004.

[4] Zenut, "Kimball vs inmon data warehouse architectures."

[5] M. Breslin, "Data warehousing battle of the giants," *Business Intelligence Journal*, vol. 7, pp. 6–20, 2004.

[6] L. Yessad and A. Labiod, "Comparative study of data warehouses modeling approaches: Inmon, kimball and data vault," in *2016 International Conference on System Reliability and Science (ICSRS)*. IEEE, 2016, pp. 95–99.