IIOAB JOURNAL

ARTICLE

DATA MODELING TECHNIQUES FOR DATA WAREHOUSE

Rohith Bharath*

Project Manager, Infosys, Chennai, INDIA

ISSN 0976-3104

ABSTRACT

The Entity-Relationship (ER) model is widely utilized in relational database environments for database design, focusing on day-to-day operations. On the other hand, Multidimensional (MD) data modeling plays a vital role in designing data warehouses aimed at supporting managerial decision-making. It facilitates decision-making by enabling users to delve deeper into detailed information, view summarized information through roll-up operations, select specific items of interest by slicing and dicing dimensions, and reorient the view of MD data through pivoting. When creating a MD model, regardless of whether it follows a star or snowflake schema, the process involves identifying facts, dimensions, and measure attributes. This paper aims to investigate how the Multidimensional model can serve as the primary framework for data warehouse design instead of the ER Model.

KEY WORDS
Entity-Relationship Model; Multidimensional Model; Fact; Dimensions; Attributes

Received: 19 May 2023
Accepted: 04 July 2023
Published: 15 July 2023

***Corresponding Author**
E-mail:
rohithbharath1977@gmail.com

INTRODUCTION

Data modeling is the process of creating a conceptual representation of data, including the relationships between different data entities and the rules that govern them. This representation, called a data model, can be used to design and implement a database [1, 2]. By using data Modeling, organizations can visualize the different types of data used, the connections between pieces of information, and how data is structured and organized. Data Modeling is a method for enhancing data to streamline information flow throughout businesses for varied work purposes [1]. There are several data modeling techniques, in this article we will be focusing on following two models:

- Entity-Relationship (ER) Data Modeling
- Multidimensional Data Modeling

ER MODELING

Entity

A person, place, object, or event of interest to a company or organization is referred to as an entity. A class of objects, or items in the actual world that can be seen and categorized according to their traits and attributes, is represented by an entity [2, 5]. In the [Fig. 1] Customer and Branch are entities.

Relationship

It illustrates the structural relationship and interaction between the model's elements. It can be used to define the connection between two entities [2, 5]. The most occurrences of one entity that can be connected to one instance in another table and vice versa. In the [Fig. 1] Withdraw and Loan are relationships.

Attribute

The traits and properties of an entity are described by its attributes [2, 3, 4]. An entity's attribute names ought to be distinctive and self-explanatory. The minimal cardinality of an attribute is zero when an instance has no value for it, indicating that it is either nullable or optional. If an attribute's maximum cardinality in ER Modeling is more than 1, the modeler will attempt to normalize the entity before elevating the attribute to another entity. As a result, an attribute's maximum cardinality is often 1. In the [Fig. 1] Name, Address and Application are attributes.

DIMENSIONAL DATA MODELING

Dimensional Modeling (DM) is a data structured method designed specifically for data warehouse storage. Dimensional Modeling is used to enhance databases for quicker data retrieval. The "fact" and "dimension" tables that make up the Dimensional Modeling idea were created by Ralph Kimball [2]. A dimensional model is a tool used in data warehouses to read, summarize, and analyze numerical data such as values, balances, counts, weights, etc. Relational models, on the other hand, are designed for the insertion, updating, and deletion of data in an online transaction system that is live [4].

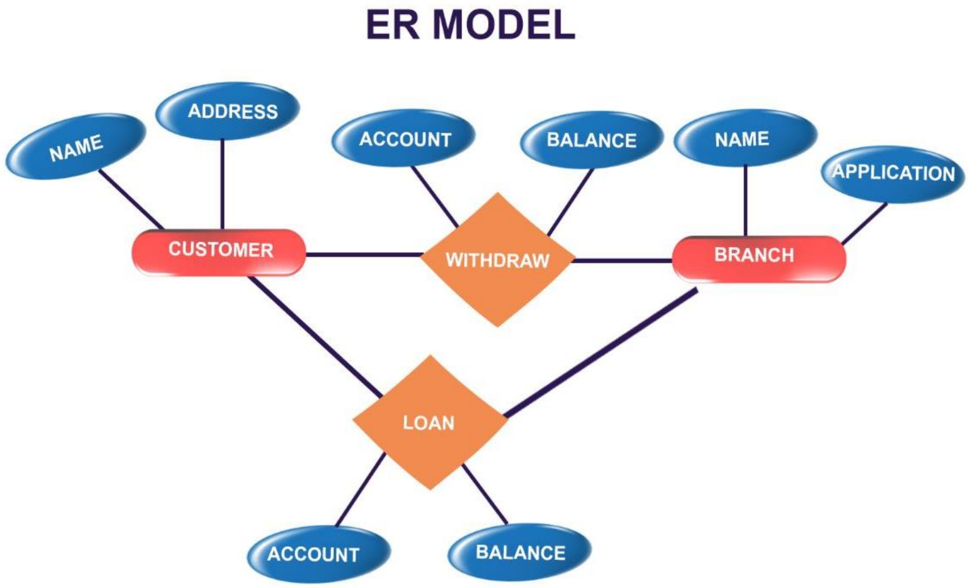

Fig. 1: Entity relationship model.

Dimension

A dimension comprises a set of members or units that share the same type of characteristics [2, 6, 13]. Typically, a dimension is depicted as an axis in a diagram. Within a Multidimensional model, each data point in the fact table is linked to a single member from each of the multiple dimensions. In other words, dimensions provide the contextual framework for the facts. Numerous analytical procedures are employed to measure the influence of dimensions on the facts. Eg. Customer, Location, Sku etc.

Types of Dimensions

In a data warehouse, the dimensions are divided into various categories based on their usage and behavior. Knowing the kind of each dimension before constructing a table in a data warehouse can help you make the best choices. There are various types of dimensions. Major three of them are shown below: conformed dimensions, junk dimensions and slowly changing dimensions.

Conformed Dimensions

Different fact tables can be associated with a conformed dimension while still retaining the same meaning. Conformed dimensions enable cross-domain queries in constellation-type data warehouse systems with several fact tables [7]. The date is an instance of a conformed dimension. Each fact table has the same meaning. The majority of data warehouses contain a single date dimension that is shared by all fact tables as a result. As they must guarantee consistency across several domains, other conformed parameters, which are less evident than the date, may present design difficulties.

Junk Dimensions

In a data warehouse, facts frequently have indicator characteristics like flags, Boolean values, or any other set of values that, due to their low cardinalities, do not make sense as dimensions [8]. A junk dimension is frequently built to combine all of these qualities into one table in order to avoid generating small dimensions for each of these variables and unnecessarily expanding the number and sizes of the fact tables. Instead of keeping the value of each of the attributes in the fact table, it is sufficient to include a single foreign key to the junk dimension table.

Slowly Changing Dimensions

Slowly Changing Dimensions are basically those dimensions whose key value will remain static but description might change over the period of time [9]. For example, the product id in a companies, product line might remain the same, but the description might change from time to time, hence, product dimension is called slowly changing dimension.

Fact

A fact encompasses a set of interconnected data elements, comprising both measures and contextual data. Generally, each fact represents a business entity, a business transaction, or an event that can be utilized in analyzing the business or its processes [10]. Within a data warehouse, facts are incorporated into the central tables that store all the numerical data.
Eg. Sales, Amount, Price

Types of Facts

There are three types of facts:

- ***Additive:*** Additive facts are facts that can be summed up through all of the dimensions in the fact table [2, 11]. Eg. Sales amount.

- ***Semi-Additive:*** Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others [2, 11]. Eg. Current balance.

- ***Non-Additive:*** Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table [2, 11]. Eg. Profit margin

Hierarchy

A dimension's elements can be arranged into one or more hierarchies. There may be several levels in each hierarchy. Not all members of a dimension locate on the same hierarchy [2, 11]. Eg. Product,Date,Location [Fig. 2].

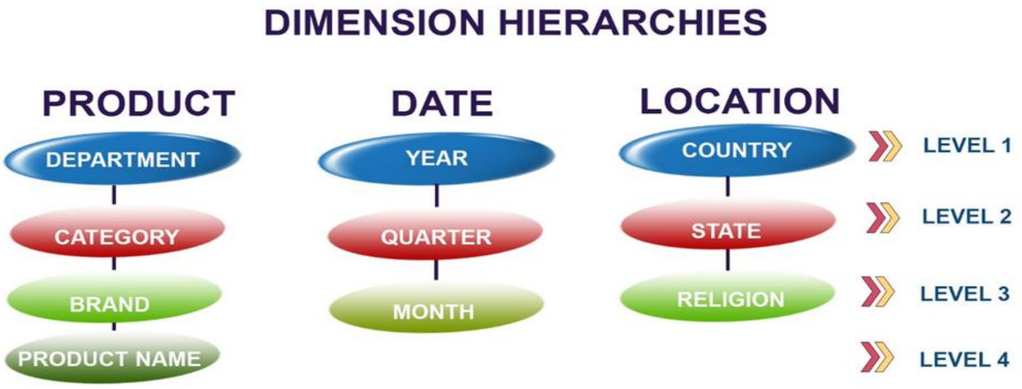

Fig. 2: Product, date and location hierarchies.

Measure

A measure refers to a numerical characteristic of a fact, indicating the performance or behavior of the business in relation to the dimensions [2, 12]. The specific numeric values associated with measures are referred to as variables. A measure is determined based on combinations of dimension members and is situated within the facts themselves. Eg. Sales revenue, Sales volume, Quantity supplied.

TECHNIQUES

Dimensional modeling employs two fundamental models: star schema and snowflake schema.

Star Schema

The Star Schema has the most straightforward structure of all the Schemas. A set of Dimensions Tables are surrounded by the Fact Table in a Star Schema [2, 16]. There are several missing normalizations in these Dimension Tables. The Dimension Tables in this Schema will include a list of characteristics that characterize the Dimension. Additionally, they have foreign keys that are connected to the Fact Table to produce results [Fig. 3].

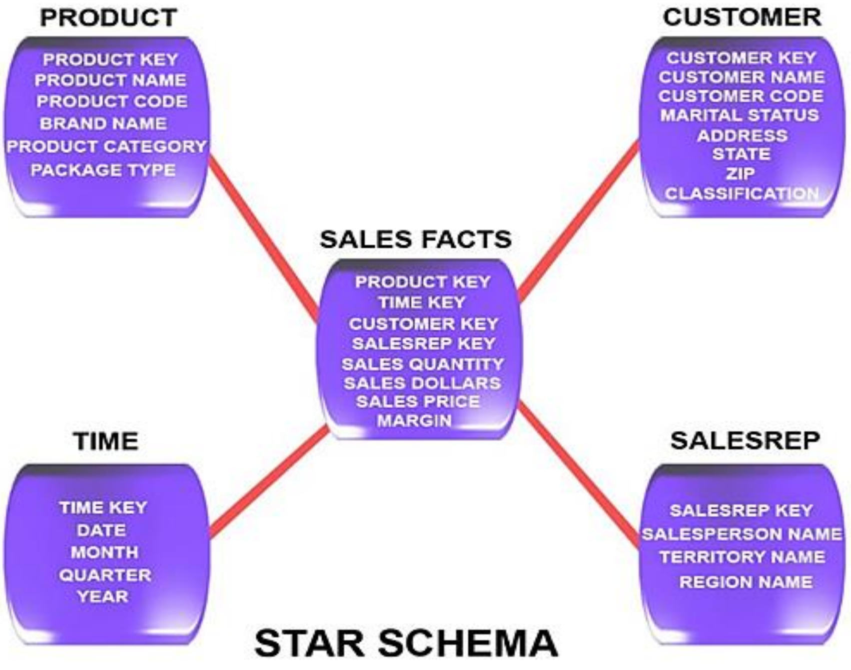

Fig. 3: Star schema.

Snowflake Schema

The Dimensions are fully normalized and divided into additional tables, unlike a Star Schema. Because the data has already been normalized, this Schema takes up less storage space [2, 17]. This Schema's efficient structure makes it simple to add Dimensions, and it also reduces data redundancy [Fig. 4].

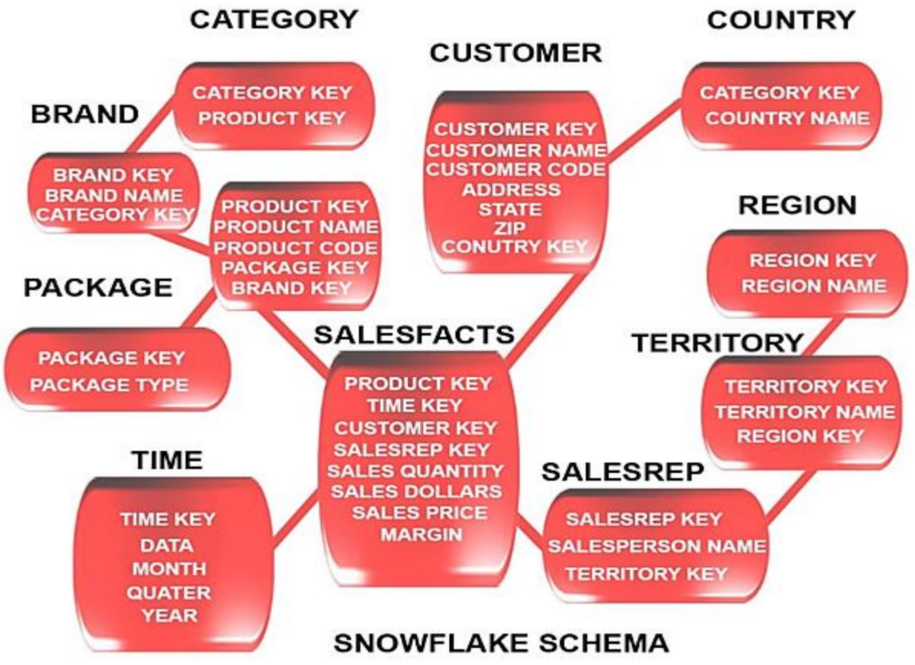

Fig. 4: Snowflake schema.

WHY ER IS NOT SUITABLE FOR DATA WAREHOUSES

The ER Model is not easily comprehensible or memorable for end users. It lacks navigation capabilities, and there is no graphical user interface (GUI) that can transform a general ER diagram into a user-friendly format[8, 9]. ER modeling is not well-suited for complex, ad-hoc queries as it primarily focuses on optimizing repetitive and narrow queries. The use of ER modeling techniques undermines one of the key advantages of data warehousing, which is intuitive and high-performance data retrieval. This is because ER modeling often leads to highly normalized relational tables, which can hinder efficient data retrieval.

BENEFITS OF DIMENSIONAL MODEL

Dimensional tables are simpler to interpret than normalized models. The business can easily understand the dimensional model. In order for the business to understand what each fact, dimension, or characteristic implies, this model is built on business terminology. Denormalized and streamlined dimensional models are used for quick data querying. This paradigm is recognized by many relational database platforms, which then optimize query execution plans to improve performance. In a data warehouse, dimensional Modeling produces a high-performance-optimized schema. It results in fewer joins and lessens data redundancy. Models with dimensions can easily adapt to change. More columns can be added to dimension tables without having an impact on currently running business intelligence applications that use these tables.

ER VS MULTIDIMENSIONAL MODEL

The following Table-1 describes the differences between ER and multidimensional modeling.

Table 1: Comparison of ER and dimensional model

ER Modeling	Dimensional Modeling
Entities and Relationships.	Fact Tables and Dimension Tables.
Few levels of granularity.	Multiple levels of granularity.
Real-time information.	Historical information.
It eliminates redundancy.	It plans for redundancy.
OLTP Application	OLAP Application
High transaction volumes using few records at a time.	Low transaction volumes using many records at a time.
Highly Volatile data.	Non-volatile data.
Normalization is suggested.	De-Normalization is suggested.

CONCLUSION

This paper explores the concept of an E-R structured data warehouse without associative entities, specifically fact tables, and discusses its feasibility in light of recent advancements in data warehousing. Several conclusions are derived from the presented arguments. Not all E-R models can be translated into a collection of star schemas that preserve the same information. However, every appropriately designed E-R data warehousing model can indeed be represented as a set of star schemas. Numerous E-R data warehouse models are inadequately constructed because they fail to explicitly acknowledge many-to-many relationships and the necessity to resolve them using associative entities, namely fact tables. Using data warehousing E-R models that only specify atomic data dependency relationships without fact tables can result in poor query response performance in large databases. This, in turn, hinders or even prevents the execution of multi-stage analysis processes. Essentially, it reduces the data warehouse to merely a large staging area for data marts, devoid of its own independent analytical functionality. Considering the emergence of Operational Data Stores (ODSs) and non-queryable centralized staging areas for storing, extracting, cleansing, transforming data, and gathering centralized metadata, the addition of another non-queryable staging area, referred to as a data warehouse, is unnecessary. Instead, what we truly require is a dimensionally modeled data warehouse capable of supporting enterprise-wide Decision Support Systems (DSS). Such a data warehouse should prioritize optimal query response performance and offer advanced OLAP functionality.

CONFLICT OF INTEREST

Authors declare no conflict of interest.

ACKNOWLEDGEMENTS

None.

FINANCIAL DISCLOSURE

None.

REFERENCES

[1] Blakeley JA, N Coburn, PA Larson. [1989] Updating derived relations: Detecting irrelevant and autonomously computable updates. ACM Transactions on Database Systems 14 (3): 369-400.

[2] Kimball R. [1996] The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, New York.

[3] Blakeley JA, Coburn N, Larson P. [1986] Updating derived relations: Detecting irrelevant and autonpmously computable updates. In Proceedings of the 12th International Conference on Very Large Data Buses (Kyoto), 457-466.

[4] Firestone JM. [1998] Dimensional Modeling and E-R Modeling In The Data Warehouse White Paper No. Eight. (https://www.dkms.com/papers/dmerdw.pdf, Accessed on March 03, 2023).

[5] Seenivasan D, [2023] ETL (Extract, Transform, Load) Best Practices, International Journal of Computer Trends and Technology, 71(1):40-44,

[6] Hussain H, Malik SUR, Hameed A, et al. [2013] A survey on resource allocation in high performance distributed computing systems. Parallel Computing 39(11):709-736.

[7] Gour V, Sarangdevot SS, Tanwar GS, Sharma A. [2010] Improve performance of extract, transform and load (ETL) in data warehouse. International Journal on Computer Science and Engineering 2(3):786-789.

[8] Korhonen JJ, Melleri I, Hiekkanen K, Helenius M. [2014] Designing data governance structure: an organizational perspective. GSTF Journal on Computing 2(4): 11-17.

[9] Ranjan J. [2009] Business intelligence: Concepts, components, techniques and benefits. Journal of theoretical and applied information technology 9(1):60-70.

[10] Radhakrishna V, SravanKiran V, Ravikiran K. [2012] Automating ETL process with scripting technology. In 2012 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, IEEE, 1-4.

[11] Seenivasan D. [2023] Exploring Popular ETL Testing Techniques, International Journal of Computer Trends and Technology 71(2):32-39.

[12] Ali SMF, Wrembel R. [2017] From conceptual design to performance optimization of ETL workflows: current state of research and open problems. The VLDB Journal 26(6):777-801.

[13] Seenivasan D. [2023] Improving the Performance of the ETL Jobs, International Journal of Computer Trends and Technology 71(3): 27-33.

[14] https://medium.com/ziegert-group/etl-performance-improvement-c5a9bd65b6af (Accessed on March 12, 2023).

[15] https://blog.devart.com/how-to-optimize-sql-query.html Accessed on March 14, 2023).

[16] http://www.ijmer.com/papers/(NCASG)%20-%202013/24.pdf Accessed on March 12, 2023).

[17] https://dataintegrationinfo.com/improve-etl-performance/ Accessed on Jan 03, 2023).

[18] https://medium.com/@data_analytics/etl-for-data-warehousing-1203dc346a4e Accessed on March 14, 2023).