Cloud-based Data Warehouse Model Creation, Loading, and Performance Evaluation: A Comparative Analysis

[1]Pooja D. Kavishwar (Research Scholar)
Department of Computer Science Shivaji Science College,
Congress Nagar, Nagpur, Maharashtra, India

[2]Dr. S. R. Pande (HOD)
Department of Computer Science, Shivaji Science College,
Congress Nagar, Nagpur, Maharashtra, India

Abstract:- This research paper aims to investigate the process of creating and loading a data warehouse model on cloud platforms and evaluate its performance. With the increasing adoption of cloud computing, organizations are leveraging cloud platforms to store and process large volumes of data for analytics purposes. By examining the data warehouse model creation, loading procedures, and performance metrics on different cloud platforms, this study aims to provide insights into the strengths and weaknesses of various platforms in supporting efficient and scalable data warehousing solutions.

Keywords:- *Data Warehouse models; Conceptual Data Models; Physical Data Models; Analytics Query; Query Response Time.*

I. INTRODUCTION

Data warehousing is a process of collecting, organizing, and storing large volumes of structured and/or unstructured data from various sources to support decision-making and business intelligence activities.

Cloud-based data warehousing refers to the practice of hosting and managing data warehouses on cloud computing platforms.Cloud platforms offer many features like data scalability and flexibility to handle varying data volumes. Cloud-based data warehousing reduces infrastructure costs, as they pay for resources on a usage basis.[1] Cloud providers employ robust security measures to protect data, including encryption, access controls, and data backups. Cloud platforms offer integration capabilities with various data sources and analytics tools, allowing seamless data ingestion and integration. [2]

Data warehouse models serve as the foundation for decision support and business intelligence activities. Accurate and well-designed data warehouse models, combined with efficient loading processes and optimal performance, enable organizations to extract meaningful insights and make data-driven decisions more effectively. Evaluating performance on cloud platforms ensures that decision-makers can access timely and accurate information for analysis.[3]

II. METHODOLOGY

A. *Selection of Cloud Platforms for Evaluation*

➢ *Google Big Query :*

It is a serverless, highly scalable, and cost-effective cloud data warehouse service from Google Cloud Platform (GCP). BigQuery uses the Google File System (GFS) for storage and the BigQuery Query Engine for querying data. Big Query is a good choice for businesses that need to analyze large amounts of data quickly and easily.[4]

➢ *Snowflake :*

It is a cloud-based data warehouse that is designed to be highly scalable, secure, and easy to use. Snowflake uses a unique architecture that separates storage from compute, which allows it to scale horizontally and provide high performance. Snowflake is a good choice for businesses that need to analyze large amounts of data quickly and easily.[5]

B. *Data Warehouse Model Creation Process*

The foremost step to start working on data warehouse modeling is to understand the business for which the data warehouse is to be created and used for.[6] Business and business requirements play a vital role in data modeling.[7] The process of Building data-driven decision-making and performance analysis analytical systems to support a range of business queries is stated in this research paper. The business domain selected for the experimental work is Retail business domain. The research paper focuses on analytics like checking products orders status, local supplier revenue generation, Predicting Revenue generation, checking shipping Volume. Thus addressing these requirements, a scheme for the warehouse model is to be designed.

The unstructured data which was considered for the study had following aspects present in it.

➢ *Sales Data:*

Products details, highest purchase consumer details, Revenue and profit margins over a product, over the time sales performance data.

➢ *Consumer Segmentation Details:*

Consumers Demographics, Purchasing Behavior, and Preferences.

- *Inventory Data*:

Total available products, its quality information, supplier details, demographics data etc.

- *Supplier Data:*

Details of the vendor with its delivery times, product quality, and pricing.

- *Order Data:*

Consumers Purchase Data Details. Product Sold Its Details and the Supplier which Sold those Product Details.

I.

- *Objectives of Designing a Schema of a Data Warehouse Model.*

Based on the above unstructured form of data following analysis can be made. These analysis criteria are the objectives of designing a schema of a data warehouse model.

- *Analysis can be made in identifying target markets, developing personalized marketing strategies, and improving consumer retention.*
- *Assessing supplier performance based on criteria such as delivery times, product quality, and pricing. This analysis can help in vendor selection, negotiation, and maintaining healthy supplier relationships.*
- *Order Fulfillment Analysis can be done to improve consumer satisfaction by reducing delivery times and ensuring timely order processing.*
- *Financial analysis by examining revenue, expenses, and profitability across different dimensions such as products, regions, or time periods. This analysis can provide insights into cost structures, profitability drivers, and financial trends*

- *Based on these Objectives Conceptual Design can be Drawn*

- *Conceptual Model of the Data Warehouse Model*

- ✓ *Consumer*: This entity represents information about the consumers, including their unique identifier, name, address, contact details, and demographic information.
- ✓ *Supplier*: This entity captures details about the suppliers, such as their identifier, name, address, and contact information.
- ✓ *Product*: This entity represents the products offered by the business. It includes attributes like a unique identifier, product name, description, price, and other relevant product details.
- ✓ *Order*: This entity represents consumer orders. It includes attributes such as an order identifier, order date, consumer identifier, and additional information related to the order.
- ✓ *OrderItem*: This entity captures individual line items within an order. It includes attributes such as the order identifier, product identifier, quantity, price, and other details specific to each line item.
- ✓ *Region*: This entry represents regions of the globe. It includes only attributes as key and name of the region,
- ✓ *Nation*:This entry represents names of the nations in the respective regions. It has attributes as key and name of nation with region key.
- ✓ The relationships between these entities can be represented as follows:
- ✓ Consumers can place multiple orders, so there is a one-to-many relationship between the consumer and Order entities.
- ✓ Orders can contain multiple line items, so there is a one-to-many relationship between the Order and OrderItem entities.
- ✓ Products can be associated with multiple line items, so there is a one-to-many relationship between the Product and OrderItem entities.
- ✓ Suppliers can provide multiple products, so there is a one-to-many relationship between the Supplier and Product entities.
- ✓ Region entries have many Nations entries.

- *Logical Data Model*

The logical data model for the snowflake schema would further refine the conceptual data model by specifying the attributes, data types, and relationships between entities. Here is a representation of the logical data model for the snowflake schema:

- *Entities:*

- ✓ Consumer {ConsumerID (Primary Key), Name, Address, City, State, Zip, Phone, …}
- ✓ Supplier { SupplierID (Primary Key), Name, Address, City, State, Zip, Phone, …}
- ✓ Product { ProductID (Primary Key), Name, Description, Price, …}
- ✓ ProdSupp {ProductID (Foreign Key),SupplierID (Foreign Key),AvailQty, Supplycost}
- ✓ Order { OrderID (Primary Key), consumerID (Foreign Key), OrderDate, …}
- ✓ OrderItem { OrderID (Foreign Key), ProductID (Foreign Key), Quantity, Price}
- ✓ Region {RegionKey(Primary Key), Name}
- ✓ Nation {NationKey(Primary Key) ,RegionKey(Foreign Key), Name}

- *Relationships:*

- ✓ One Consumer can place multiple Orders (One-to-Many)
- ✓ Many Supplier can supply multiple Products (Many-to-Many)
- ✓ One Product can be included in multiple OrderOrderItems (One-to-Many)
- ✓ One Order can have multiple OrderItems (One-to-Many)

- *Physical Model*

A physical model for the snowflake schema refers to the way data is stored and organized on disk in a database system to optimize query performance. It involves several components, including tables, indexes, and partitions.[8] The model consists of a set of tables that represent different entities in a decision support system, such as Consumer, orders, order items, suppliers, and more. These tables store the actual data that is used to generate the analytics queries. Each table has a defined schema with columns that correspond to specific attributes or characteristics of the entity it represents. Indexes are data structures that help speed up query processing by allowing faster access to specific data based on selected columns.[9] Keys are the indexes used in this model. Partitioning improves query performance by reducing the amount of data that needs to be scanned for a particular query. Orders table is partitioned into two tables orders and order items where order items store individual order details. The physical model involves designing and configuring these components in a way that optimizes query execution and resource utilization. [10]

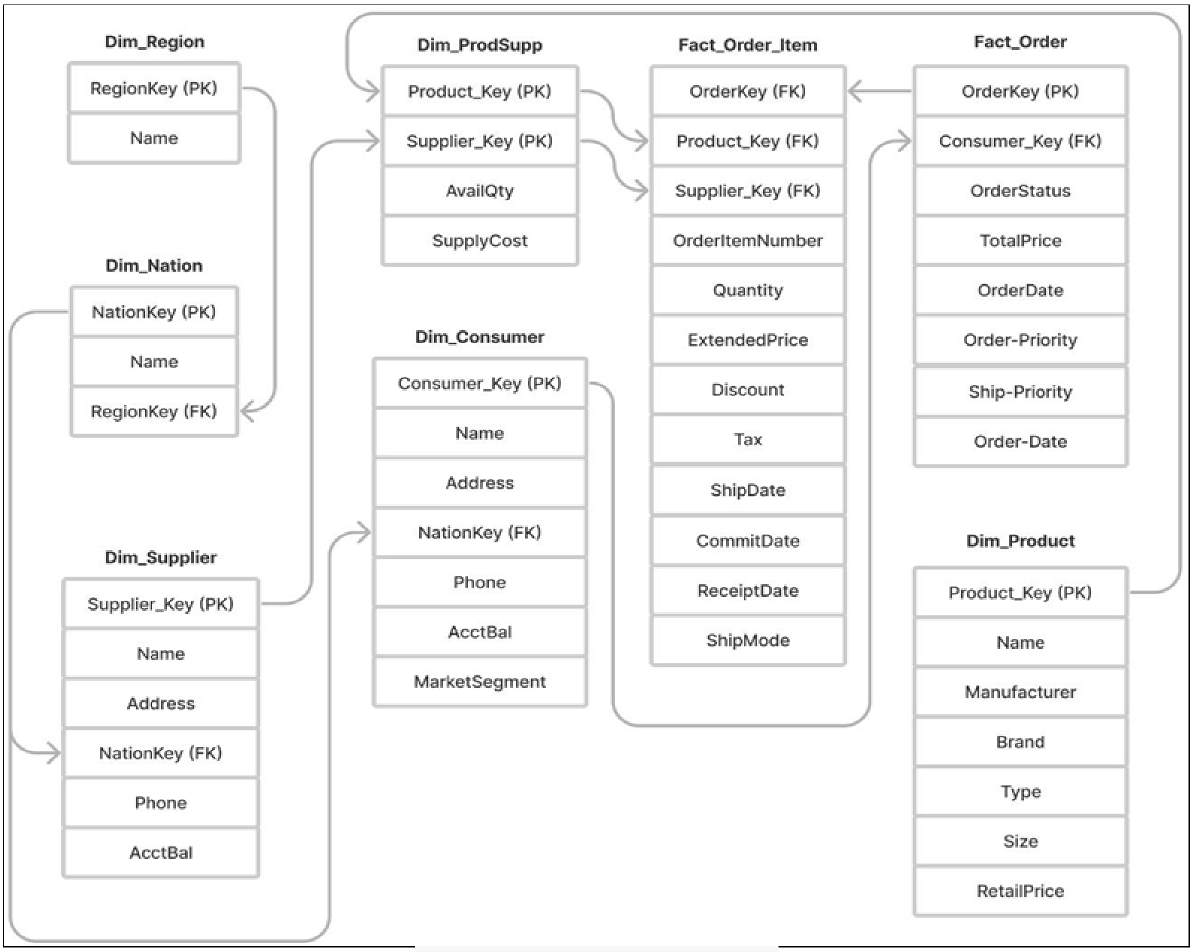

Fig1 Physical Model ER Diagram

C. *Loading Data on to the Snowflake and GCP Cloud Platforms*

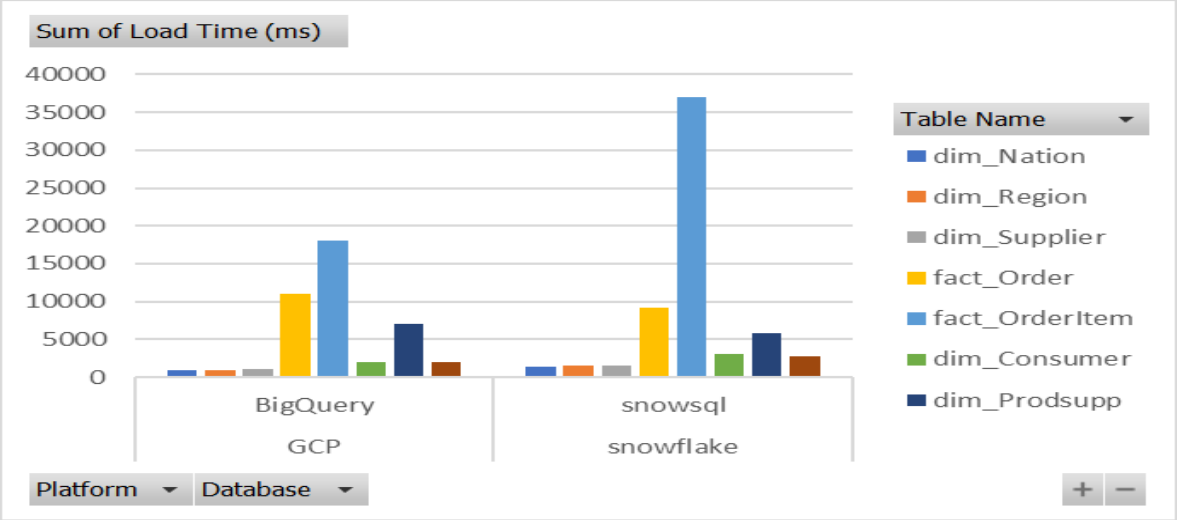

Fig 2 Load Time of Different Tables on the Two Platforms

D. *Analytics Queries Fired on Snow Flake and GCP Cloud Platform*

➢ *Order Priority Checking Query*

The Order Priority Checking Query counts the number of orders ordered in a given quarter of a given year in which at least one Order Item was received by the consumer later than its committed date. The query lists the count of such orders for each order priority sorted in ascending priority order.

Tables used: Orders, Order Item

➢ *Local Supplier Volume Query*

The Local Supplier Volume Query lists for each Nation in a Region the revenue volume that resulted from Order Item transactions in which the Consumer ordering parts and the supplier filling them were both within that nation. Revenue volume for all qualifying Order Items in a particular nation is defined as sum(l_extendedprice * (1 - l_discount)).

Tables used: Consumer, Orders, Order Item, Supplier, Nation, Region

➢ *Forecasting Revenue Change Query*

The Forecasting Revenue Change Query considers all the OrderItems shipped in a given year with discounts between DISCOUNT-0.01 and DISCOUNT+0.01. The potential revenue increase is equal to the sum of [l_extendedprice * l_discount] for all OrderItems with discounts and quantities in the qualifying range.

Tables used: Order Item

➢ *Volume Shipping Query*

The Volume Shipping Query finds, for two given nations, the gross discounted revenues derived from Order Items. The query lists the supplier nation, the consumer nation, the year, and the revenue from shipments that took place in that year.

Tables used: Order Item, Supplier, Order Item, Orders, Consumer, Nation

III. RESULTS

➢ *Presentation of Findings Related to Data Warehouse Analytics Queries Response Time on Different Cloud Platform.*

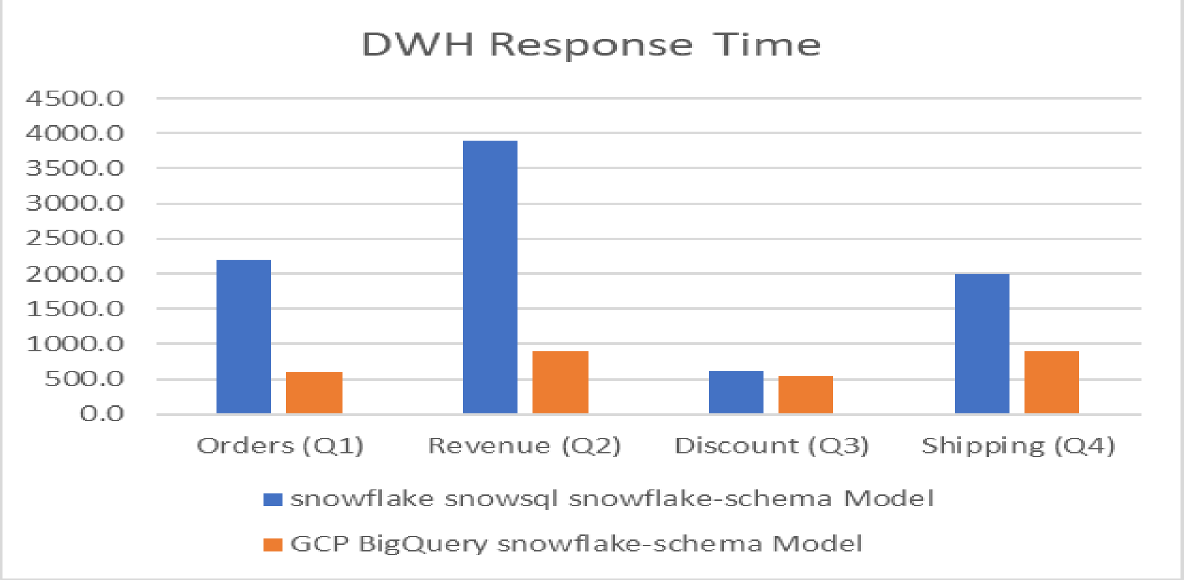

Fig 3 Data ware house Response Time for the Analytics Queries

IV. CONCLUSION

➢ *Comparison of the results across different cloud platforms*

The load time for each table also varies depending on the size and complexity of the table. For example, the fact_OrderItem table is the largest table in the dataset, and it has the longest load time.

The data shows that GCP BigQuery is significantly faster than Snowflake SnowSQL for all tables. This is likely due to the fact that GCP BigQuery is a cloud-based platform, while Snowflake SnowSQL is a self-hosted platform. Cloud-based platforms are typically faster than self-hosted platforms because they have access to more resources and can scale more easily.

REFERENCES

[1] Niu, Y., Ying, L., Yang, J., Bao, M., & Sivaparthipan, C. B. (2021). Organizational business intelligence and decision making using big data analytics. Information Processing & Management, 58(6), 102725.

[2] Díaz, M., Martín, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. Journal of Network and Computer applications, 67, 99-117.

[3] Oketunji, T., & Omodara, O. (2011). Design of Data Warehouse and Business Intelligence System: A case study of Retail Industry.

[4] Bisong, E., & Bisong, E. (2019). Google bigquery. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, 485-517.

[5] Dageville, B., Cruanes, T., Zukowski, M., Antonov, V., Avanes, A., Bock, J., ... & Unterbrunner, P. (2016, June). The snowflake elastic data warehouse. In Proceedings of the 2016 International Conference on Management of Data (pp. 215-226).

[6] Manjunath, T. N., Hegadi, R. S., & Ravikumar, G. K. (2010). Analysis of data quality aspects in datawarehouse systems. International Journal of Computer Science and Information Technologies, 2(1), 477-485.

[7] Simsion, G., & Witt, G. (2004). Data modeling essentials. Elsevier.

[8] Lightstone, S. S., Teorey, T. J., & Nadeau, T. (2010). Physical Database Design: the database professional's guide to exploiting indexes, views, storage, and more. Morgan Kaufmann.

[9] Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. ACM Sigmod record, 26(1), 65-74.

[10] Polyvyanyy, A., Ouyang, C., Barros, A., & van der Aalst, W. M. (2017). Process querying: Enabling business intelligence through query-based process analytics. Decision Support Systems, 100, 41-56.