Google BigQuery Analysis Report

# A. Detailed Description of Google Big Query

Google Big Query is a fully managed, serverless data warehouse that enables scalable analysis over petabytes of data. It is a Platform as a Service (PaaS) that supports querying using ANSI SQL. It also has built-in machine learning capabilities and is part of the Google Cloud Platform.  
  
BigQuery is designed to process read-heavy workloads and is particularly adept at running large-scale data analytics. Users can perform SQL queries on massive datasets with fast performance and high concurrency. It integrates with various data processing and machine learning tools, enabling a wide range of data-driven applications.  
  
BigQuery’s serverless architecture allows users to focus on analyzing data to find meaningful insights using SQL and does not require any infrastructure management. It's a powerful tool for businesses that need to analyze large volumes of data quickly and efficiently.

Key features and characteristics of Google Big Query include:

Serverless Architecture: Big Query is a serverless platform, meaning you don't have to manage infrastructure. Google takes care of provisioning, scaling, and maintenance.

SQL-Based Querying: It supports SQL, making it accessible to users familiar with SQL syntax. Users can write SQL queries to analyze and manipulate data stored in BigQuery.

High Performance: Big Query is optimized for high-speed querying and can execute complex analytical queries rapidly. It uses a distributed architecture for parallel query processing.

Data Warehousing: It serves as a data warehousing solution, allowing users to store, organize, and analyze structured data in tables.

Integration with Google Services: Big Query seamlessly integrates with other Google Cloud services, enabling data import/export, visualization, and machine learning integration.

Security and Access Control: Google Cloud IAM provides robust access control for datasets and tables. Data is encrypted at rest and in transit for security.

Real-Time Data Streaming: Big Query supports real-time data streaming for the ingestion and analysis of streaming data.

Machine Learning Integration: Big Query ML allows users to build and deploy machine learning models directly within the platform.

Geospatial and GIS Support: Native support for geospatial data and powerful tools for geospatial analysis.

Big Data and Analytics: It can handle both big data and complex analytical workloads, making it suitable for business intelligence, data analytics, data warehousing, and reporting.

Cost Management: Google Cloud offers flexible pricing options, including on-demand and flat-rate pricing. Budgets and cost alerts are available.

Data Governance and Compliance: Big Query supports various data governance and compliance standards, making it suitable for organizations with regulatory requirements.

Pricing of Google Big Query:  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# B. Detailed Description of the Dataset ‘sales\_data’ and ‘customer\_data’

KDD Nuggets referenced data: -

We used a dataset from Kaggle. It's a collection of sales and customer data that's ripe for analysis. With Big Query, we can sift through this dataset to discover patterns and insights that can help drive our business decisions.

Dataset: 🡪 <https://www.kaggle.com/datasets/dataceo/sales-and-customer-data?select=customer_data.csv>

Below is a detailed description of the `customer\_data` and `sales\_data` datasets:

Customer Data:

Customer data is a dataset that contains information about customers. It includes the following fields:

-customer\_id: This field is used to uniquely identify each customer. In this example, the customer is identified as "C241288."

- gender: This field represents the gender of the customer. In this case, the customer is categorized as "Female."

- age: Age is not provided in this example, but it could be a field used to store the customer's age.

- payment\_method: This field indicates the payment method preferred by the customer. In this instance, the customer uses "Credit Card" as their payment method.

Sales Data:

sales\_data is a dataset that captures sales transactions. It includes the following fields:

- invoice\_no: This field serves as a unique identifier for each sales transaction. For this example, an invoice is labeled as "I138884."

- customer\_id: It associates each sale with a specific customer. In this case, the sale is attributed to customer "C241288."

- category: This field represents the category of the product being sold. The product is categorized as "Clothing" in this transaction.

- quantity: It denotes the quantity of the product sold in this transaction. Here, five units of the product were sold.

- price: This field records the price or cost of the product. In this sale, the product costs 1500.4 currency units.

- invoice\_date: This field records the date of the sales transaction. The transaction occurred on May 8, 2022 (05-08-2022).

- shopping\_mall: It indicates the location or shopping mall where the sale took place. In this example, the transaction occurred at "Kanyon."

These datasets can be used for various purposes, such as tracking customer behavior, analyzing sales trends, and generating reports to make data-driven business decisions. Additionally, data analysis and reporting can be performed using SQL queries and analytics tools with these datasets.

# C. Product Classification: Analytical

Google BigQuery, when used for analytical purposes, is designed as an analytical database or data warehousing solution. It is optimized for handling and analyzing large volumes of data and complex queries, making it a suitable choice for analytical workloads. Below is a detailed description of Google BigQuery:

Google BigQuery for Analytical Purposes:

Data Warehousing: Google BigQuery is structured as a data warehousing solution. It is designed to store, organize, and manage structured data in a way that facilitates efficient analysis. The structured data is typically stored in tables with a clearly defined schema.

Analytical Query Performance: The architecture of Google BigQuery is optimized for high-speed querying. It can execute complex analytical queries rapidly, making it well-suited for tasks that involve aggregations, filtering, and data summarization.

Complex Analytics: Google BigQuery is built to handle complex analytics. It can efficiently process analytical workloads that require joining multiple tables, applying various analytical functions, and handling large datasets.

Support for Large Datasets: Analytical work often involves large volumes of data. Google BigQuery is designed to handle big data, making it a suitable platform for businesses that need to analyze vast datasets.

Data Aggregation and Summarization: The design of Google BigQuery supports data aggregation and summarization. It allows users to create views and tables that provide aggregated or summarized information, making it easier to derive insights and generate reports.

Data Visualization: Google BigQuery is often used in conjunction with data visualization tools and business intelligence platforms. Its analytical design makes it compatible with tools that help users create interactive charts, dashboards, and reports.

Why Google BigQuery's Design is Analytical:

The design of Google BigQuery is inherently analytical because it focuses on the following characteristics and features that are crucial for analytical work:

Columnar Storage: Data is stored in a columnar format, which is well-suited for analytical workloads. This storage design enhances query performance, especially when specific columns need to be scanned or aggregated.

Parallel Query Processing: BigQuery's architecture allows it to parallelize query processing. This means that it can split queries into smaller tasks and process them concurrently, significantly reducing query execution time.

Data Clustering and Partitioning: BigQuery supports data clustering and partitioning, enabling users to organize and structure data in ways that improve query performance. Clustering and partitioning help optimize data access and filtering.

Aggregation Functions: It offers a wide range of aggregation functions (e.g., SUM, AVG, COUNT) that are essential for analytical tasks. These functions simplify the process of summarizing and aggregating data.

Join Capabilities: BigQuery has robust join capabilities, allowing users to combine data from multiple tables to perform complex analytics, such as market basket analysis, cohort analysis, and customer segmentation.

Integration with BI Tools: Google BigQuery can be integrated with various business intelligence tools and data visualization platforms, enabling users to create interactive reports and dashboards for in-depth analysis.

Scalability: BigQuery's serverless architecture allows it to scale dynamically, ensuring that it can handle large and growing analytical workloads without manual intervention.

Overall, the analytical design of Google BigQuery makes it a powerful tool for businesses and data professionals to perform data analysis, generate insights, and support data-driven decision-making. Its robust features and optimized architecture align with the needs of analytical workloads and the extraction of valuable insights from data.

# D. Product Data Structures: Analytical

In Google BigQuery, which is often used for analytical purposes, data structures typically differ from traditional relational databases that use primary and foreign keys for data integrity. In an analytical context, data structures prioritize efficient data analysis, aggregation, and reporting over enforcing referential integrity through primary and foreign keys. Below is a detailed description of Google BigQuery for analytical purposes and the provided data:

Analytical Data Structures in Google BigQuery:

Customer Data:

-customer\_data contains information about customers.

-customer\_id: A unique identifier for each customer, allowing for segmentation and analysis.

-gender: The customer's gender, a categorical attribute used for segmentation and analysis.

-age: The customer's age, which can be used for demographic analysis.

-payment\_method: The preferred payment method for each customer, used for payment analysis and segmentation.

Sales Data:

-sales\_data captures sales transactions for analytical purposes.

-invoice\_no: A unique identifier for each sales transaction, facilitating tracking and analysis.

-customer\_id: Links each sale to a specific customer, enabling customer-specific analysis.

-category: Categorizes products into different categories for product analysis and reporting.

-quantity: Records the quantity of each product sold in a transaction, allowing for sales analysis.

-price: The price or cost of each product sold, crucial for revenue analysis.

-invoice\_date: Records the date of each sales transaction, which is essential for date-based analysis.

-shopping\_mall: Indicates where each sale took place, enabling location-based analysis.

Data Mart Design:  
A screenshot of a table

Description automatically generated

Analytical Data Design:

The design of analytical data structures in Google BigQuery prioritizes ease of analysis, query performance, and flexibility. It does not rely on primary and foreign keys, which are more common in transactional databases, to enforce data integrity. Instead, the focus is on data availability, schema-on-read flexibility, and query optimization to support complex data analysis and reporting.

This design allows analysts and data professionals to explore and analyze data without the constraints of traditional relational databases, making it well-suited for tasks such as market analysis, trend analysis, and generating insights from large datasets.

# E. CRUD Operations or ETL Process

ETL (Extract, Transform, Load) is a crucial data processing method, and when applied in the context of Google BigQuery, it plays a significant role in data integration, preparation, and analysis. Below is a detailed description of the ETL process in Google BigQuery:

1. Extract (E):

-Data Sources: The ETL process begins with data extraction from various sources. Data sources can include external systems, cloud storage, databases, or other applications. In Google BigQuery, data can be extracted from sources like Google Cloud Storage, Google Sheets, external databases, and more.

-Data Formats: Data extracted from these sources can be in various formats, such as CSV, JSON, Avro, or Parquet. Google BigQuery provides tools and functions to handle different data formats efficiently.

-Data Transfer: The data extraction can involve data transfer processes, such as copying data from one storage location to Google Cloud Storage or directly to BigQuery.

-Authentication and Authorization: Ensuring secure access to data sources is essential. Google Cloud IAM (Identity and Access Management) is used to manage authentication and authorization to access external data.

A diagram of a data processing process

Description automatically generated

2. Transform (T):

-Data Cleansing: Data cleansing is a critical part of the transformation process. It involves identifying and handling missing or incorrect data, duplicate records, and outliers. Data quality and consistency are improved during this stage.

-Data Structuring: Data is structured to match the desired format and schema. This may involve reshaping data, renaming columns, and changing data types to align with the analysis and reporting requirements.

-Data Aggregation: Aggregations are applied to transform raw data into summarized or aggregated forms, which are useful for analytical tasks. Aggregated data helps in generating insights from large datasets.

-Data Enrichment: Additional data can be added to the dataset from other sources to enhance its value. This may include joining data from different tables or enriching with external data sources.

-Deriving Calculated Fields: New fields or calculated values can be derived from existing data. For example, calculating total sales, growth rates, or customer segments.

-Data Validation: Data is validated to ensure that it adheres to the defined business rules and data quality standards.

3.Load (L):

-Loading to BigQuery: Transformed data is loaded into Google BigQuery, a fully managed and scalable data warehouse. BigQuery provides an optimized environment for storing and querying structured data, making it suitable for analytical workloads.

-Table Creation: During the load process, tables in BigQuery may be created or updated with the transformed data. Tables can be partitioned or clustered to optimize query performance.

-Incremental Loading: In cases of ongoing data ingestion, an incremental loading strategy may be employed. This involves adding new data to existing datasets without reloading the entire dataset.

-Data Scheduling: ETL jobs can be scheduled to run at specific intervals to keep data up to date, ensuring that analysis and reporting are based on the latest data.

ETL Workflow Automation:

- Google Cloud provides tools and services for automating ETL workflows. Cloud Composer (based on Apache Airflow) is often used to orchestrate and automate ETL tasks and workflows in a managed and scalable way.

Scalability and Cost Management:

- Google BigQuery's serverless architecture allows for scaling ETL processes as data volumes grow. Cost management is also supported through different pricing models, including on-demand and flat-rate pricing.

Security and Compliance:

- Google Cloud's security and compliance measures help ensure that data remains secure and compliant during the ETL process.

From the above data set we have created a data mart without primary and foreign keys, and we have created business questions as well. Below we can see the screenshots of the Results.

A screenshot of a computer

Description automatically generated

A graph of blue squares

Description automatically generated

A screenshot of a computer

Description automatically generated

Google Big Query does not enforce keys and other constraints, it has a function Generate\_UUID() which can be used as a primary key because it generates a unique key for the complete row as you can see above.

In summary, the ETL process in Google BigQuery is a comprehensive set of tasks and procedures for extracting, transforming, and loading data into BigQuery for analytical purposes. It plays a fundamental role in data integration and preparation, making data available and actionable for analysis and reporting. The design of Google BigQuery and its integration with other Google Cloud services make it a powerful platform for executing ETL workflows efficiently and securely.

Teams Recording Link:

https://usfedu-my.sharepoint.com/:v:/r/personal/thulasisaimediga\_usf\_edu/Documents/Recordings/Call%20with%20Reddy%20and%202%20others-20231108\_215557-Meeting%20Recording.mp4?csf=1&web=1&e=1VQ6HA&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZyIsInJlZmVycmFsQXBwUGxhdGZvcm0iOiJXZWIiLCJyZWZlcnJhbE1vZGUiOiJ2aWV3In19