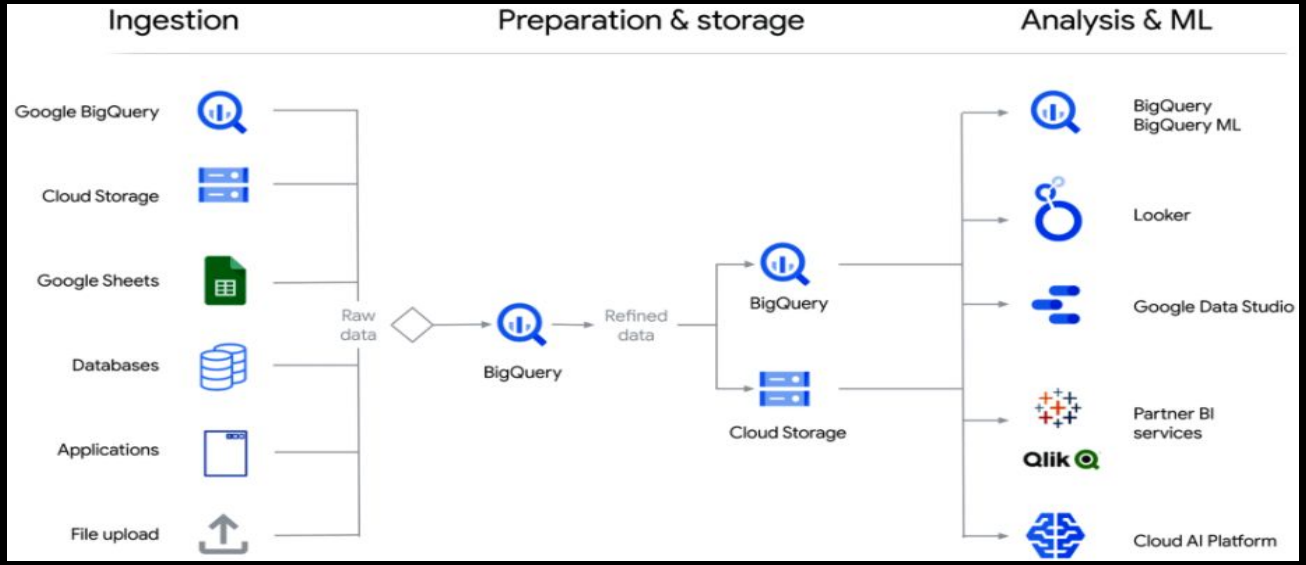
BigQuery

What is BigQuery?

BigQuery is a fully managed enterprise data warehouse that helps manage and analyze data with built-in features like machine learning, geospatial analysis and business intelligence.

BigQuery's serverless architecture lets us use SQL queries to answer organization's biggest questions with zero infrastructure management. It's scalable, cost effective, distributed analysis engine lets us query terabytes in seconds and petabytes in minutes.



- BigQuery is a fully-managed, server less data warehouse that helps you ingest, store, analyze and visualize data with ease up to petabyte scale. (In layman terms, it is just a **BIG** database).

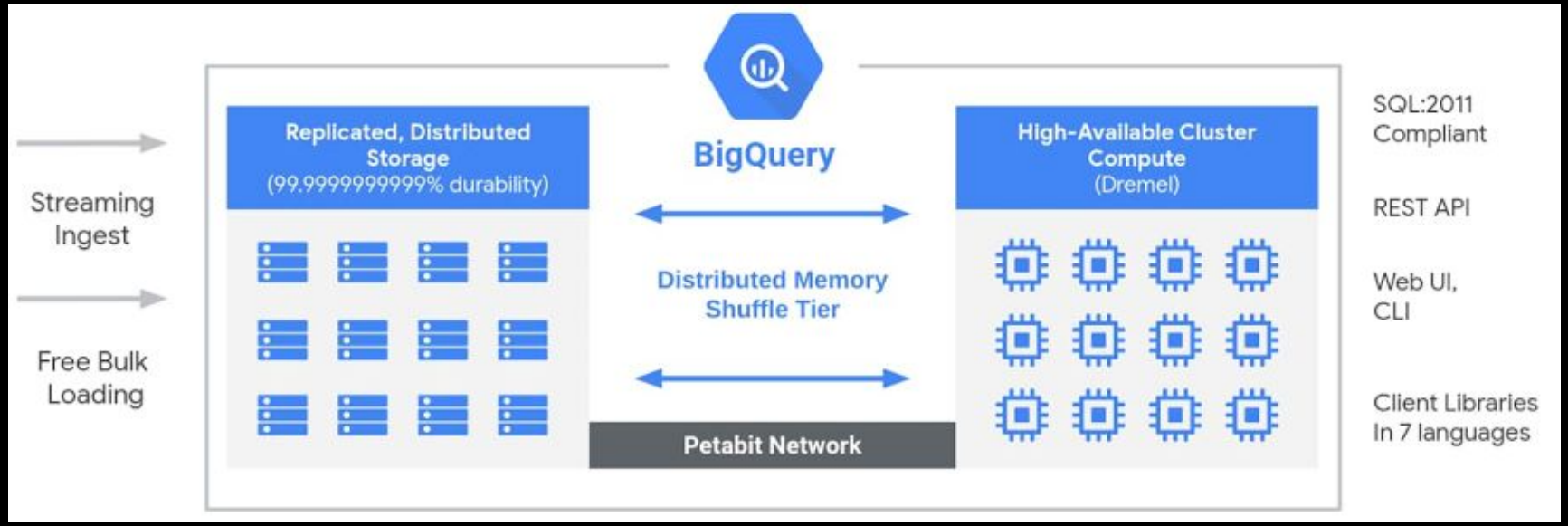
- It is a platform as a service data (PaaS) warehouse.

- Uses standard SQL dialect that is ANSI compliant.

- BigQuery comes with a built-in query engine that is capable of running SQL queries on terabytes of data in a matter of seconds, and petabytes in only minutes!

- It also has built-in machine learning capabilities.

BigQuery Architecture

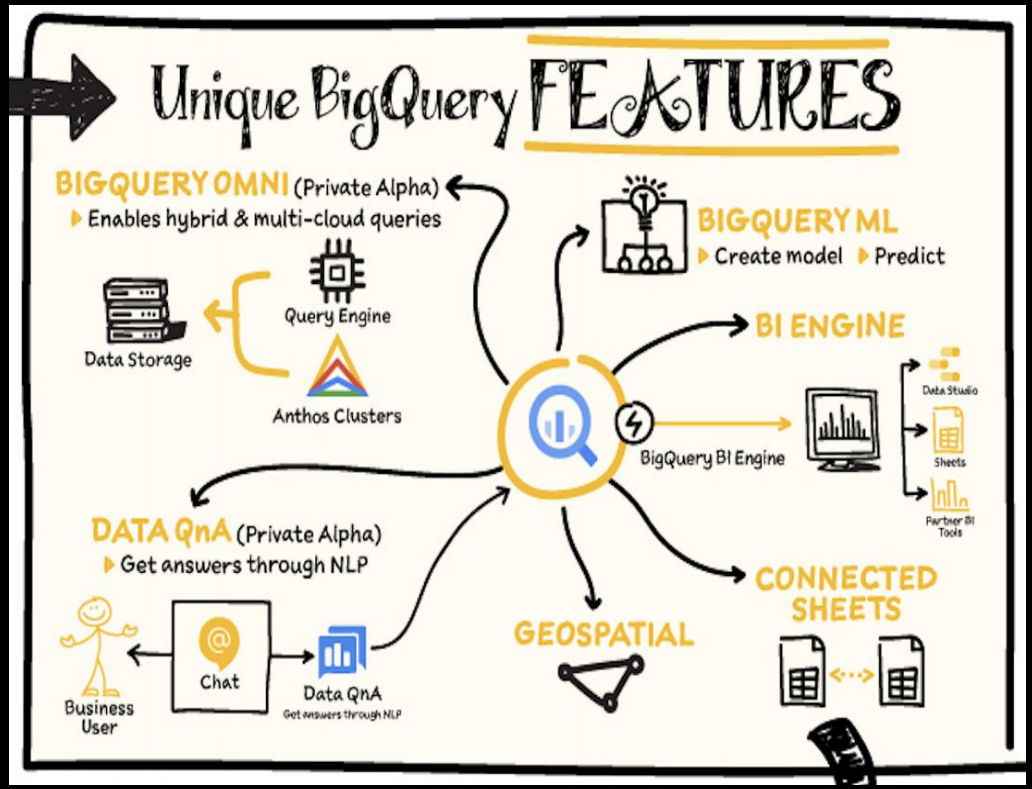


- BigQuery’s serverless architecture decouples storage and compute and allows them to scale independently on demand. This structure offers both immense flexibility and cost controls for customers because they don’t need to keep their expensive compute resources up and running all the time.

- Its different from traditional node-based cloud data warehouse solutions or on-premise massively parallel processing (MPP) systems. This approach allows customers of any size to bring their data into the data warehouse and start analyzing their data using Standard SQL without worrying about database operations and system engineering.

Features of BigQuery

- Multi Cloud Functionality
- **Built-in ML Integration**
- Foundation for BI
- **Geospatial Analysis**
- Automated Data Transfer
- Free Access
- **Partitioning and Clustering**

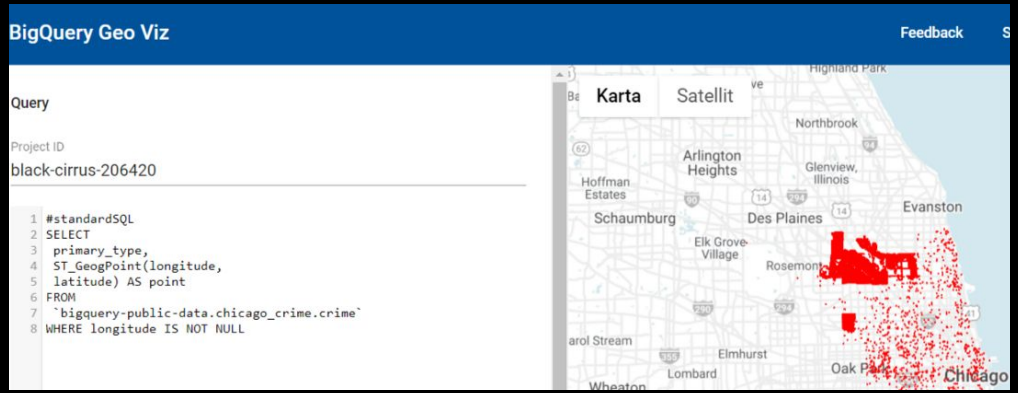


What is Geospatial Analysis ?

Geospatial Analytics highlights historical changes and current shifts by collecting, displaying, and manipulating Imagery and Geographic Information System (GIS) Data related to a specific location. GIS provides information about location and mapping. It functions by converting latitudes and longitudes columns into geographical points.

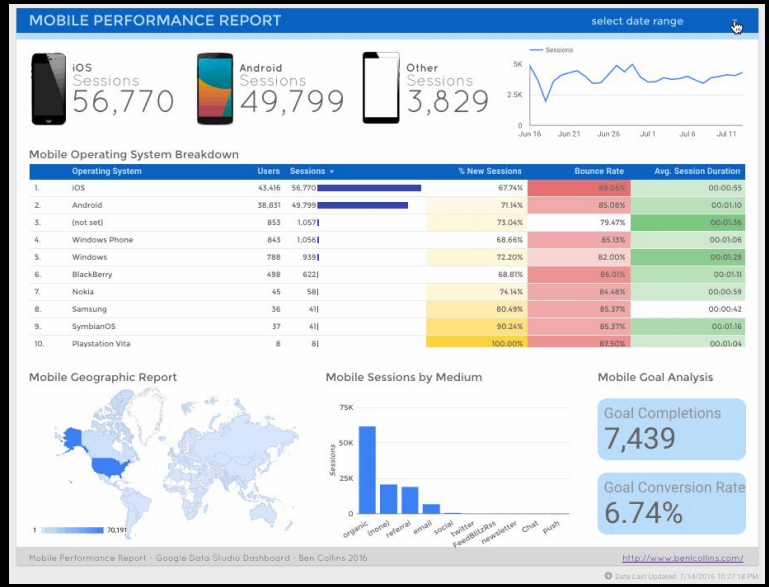
The gathered information helps create Data Visualizations, such as Graphs, Maps, Stats, and Cartograms. These reports help the human brain to understand the distance, proximity, and contiguity not visible in large datasets.

Tools Supported by BigQuery for Geospatial Analytics - BigQuery GeoViz, Google Data Studio, Google Earth Engine and Jupyter Notebooks.



BigQuery Geo Viz

Google data studio



Machine learning using BigQuery

Google BigQuery ML is a set of tools and extensions that enables users to create, train and execute Machine Learning models in Google BigQuery using standard SQL queries. It eliminates the need for data movement by allowing users to directly create Machine Learning models into the Data Warehouse.

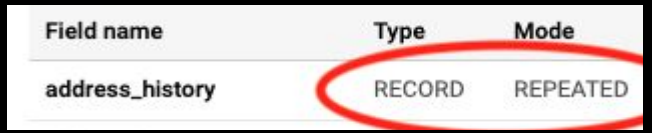
Key Features of Google BigQuery ML - Automatically Generate ML Models, Eliminates Data Transfer and Encrypted Models

Steps to Create Models using Google BigQuery ML - Setting Up the Environment, Creating Machine Learning Model, Evaluating the Model, Predicting Using the Model

Array and Structs using BigQuery

Arrays in BigQuery, like in any other language, are a collection of elements of the same data type. Struct is a data type that has attributes in key-value pairs, just like a dictionary in Python. It can have more attributes, each with its own value, related to one key/ID.

If we want to store multiple *Structs* against each key/ID, we can use *Array of Structs.*

Array	
Struct	
Array of Structs	

Difference between Array and Struct

ARRAY	STRUCTS
❑ An array is an ordered list containing values of the same data type.	❑ Structs are flexible containers of ordered fields each with a type (required) and a name (optional).
❑ It is used when we want to store repeated values in a single row such as those below scenarios. • An employee has one or multiple skills. • A restaurant listed on Zomato has one or multiple cuisine labels such as Italian, Pizza, Pasta, Casual. • A customer's sales order includes one or multiple items.	❑ Contrasting with arrays, you can store multiple data types in a Struct, even Arrays. In Google BigQuery, a Struct is a parent column representing an object that has multiple child columns. For example, • A restaurant has a location represented by different fields such as address, city, state, postal code. • An employee has a qualification associated with different fields such as university, degree, start date and end date.

Partitioning and Clustering in BigQuery

A partitioned table is a special table that is divided into segments, called partitions, that make it easier to manage and query your data. Dividing a large table into smaller partitions, improves query performance and control costs by reducing the number of bytes read by a query. BigQuery tables can be partitioned by - Time-unit column, ingestion time, integer range.

Ingestion time	_PARTITIONTIME	Partition (hourly)
2021-05-07 17:22:00	2021-05-07 17:00:00	2021050717
2021-05-07 17:40:00	2021-05-07 17:00:00	2021050717

Ingestion Time Partitioning

Partitioning is used under the following circumstances:

- You want to know query costs before a query runs. Partition pruning is done before the query runs, so you can get the query cost after partitioning pruning through a dry run. Cluster pruning is done when the query runs, so the cost is known only after the query finishes.
- You need partition-level management. For example, you want to set a partition expiration time, load data to a specific partition, or delete partitions.
- You want to specify how the data is partitioned and what data is in each partition. For example, you want to define time granularity or define the ranges used to partition the table for integer range partitioning.

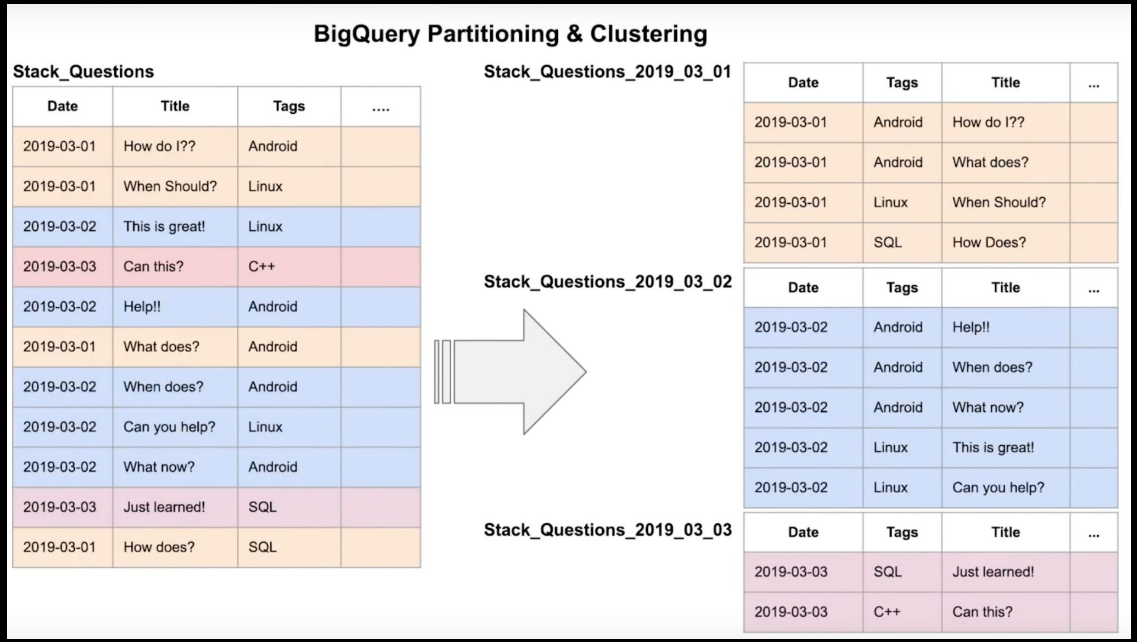
Clustering is another way of organizing data which stores one next to the other all those rows that share similar values in the chosen clustering columns. This process increases the query efficiency and performances. Note that BigQuery supports this feature only on partitioned tables.

Clustering should be used under the following circumstances -:

- We don't need strict cost guarantees before running the query.
- We need more granularity than partitioning alone allows. To get clustering benefits in addition to partitioning benefits, we can use the same column for both partitioning and clustering.
- Queries commonly use filters or aggregation against multiple particular columns.
- The cardinality of the number of values in a column or group of columns is large.

Prefer clustering over partitioning under the following circumstances:

- Partitioning results in a small amount of data per partition (approximately less than 1 GB).
- Partitioning results in a large number of partitions beyond the limits on partitioned tables.
- Partitioning results in your mutation operations modifying most partitions in the table frequently (for example, every few minutes).



Thank you!