What is BigQuery?

BigQuery is a server less data warehouse solution with a built in query engine is available in the Goole cloud platform. The query engine is capable of running SQL queries on terabytes of data in a few second. So when data is huge and in structured format we can use BigQuery to analyse and process the data.

Why to use BigQuery?

If companies are having small amount of data then we can store in a spreadsheet and do operations on data according to our requirements. But if the amount of data grows to gigabytes or terabytes then we will need a more efficient system like a data warehouse to store data. Again if amount of data is huge, then to ask a question and get back the answer takes a long time may be some minutes or hours or a day. For example log data from thousands of Retailors around the world or IOT data from millions of vehicle censors across the globe.

So BigQuery is a fully managed and server less data warehouse that is one does not need to set up or install anything or no need of database administrator. One can log into Google cloud project from browser and get started.

BigQuery focuses on analytics rather than management infrastructure. Google cloud provides a web service where we can assign a read or write permissions to specific users, groups or projects with more important feature, keeps sensitive data as more secured.

Advantage:

* It centralizes the data
* Any variety of data can be stored (log, xml, multimedia, censor data, chat…
  + Stores structured, semi-structured and unstructured data
* Scalability : never bother about the data size
* Can be used for Data analytics, data visualization, machine learning etc..

Types of SQL Statements

* Data query language(DQL)
  + select
* Data definition language(DDL)
  + create, alter, drop
* Data manipulation language(DML)
  + insert, update, delete
* Data control language
  + grant, revoke

**Data definition language(DDL)**

***Create a table***

**Points to check**:

*Number of errors allowed*: Maximum number of rows that containing the errors can be ignored. If number of rows with error exceeds zero, the job will result an invalid message and fails.

Always set to zero value

*Unknown value*: if you defined a schema which contains 3 columns and the file contains 4 columns then if we *enable* (ignore unknown values) then the rows that match with 4th column value will not be import to big query.

*Header rows to skip*: want to skip row from the file then give the number…

*Jagged rows*: The missing values are treated as nulls. If unchecked, records with missing trailing columns are treated as bad records, and if there are too many bad records, an invalid error is returned in the job result.

If checked: missing values are treated as NULL values otherwise columns with missing values will be treated as bad records.

*Create a table*

**Data manipulation language (DML)**

Data manipulation language (DML) statements, such as INSERT, UPDATE, DELETE, and MERGE, enable users to add, modify, and delete data stored in BigQuery.

**Insert Query:**

Insert statement is used to append new records into a table. We can assign multiple number of insert statements to insert new records into a given table.

To create a table….

Case-1

CREATE TABLE optimistic-jet-344105.AdvertisementDataset.customer\_new

(

customer\_id INT64,

name STRING,

location STRING,

ts TIMESTAMP

)

insert into optimistic-jet-344105.AdvertisementDataset.customer\_new (customer\_id,name,location,ts) values (101,'radha','Mangalore','2022-8-03') ;

SELECT distinct \* FROM `optimistic-jet-344105.AdvertisementDataset.customer\_new` LIMIT 10

CREATE OR REPLACE TABLE `optimistic-jet-344105.AdvertisementDataset.customer\_new`

AS SELECT DISTINCT \* FROM `optimistic-jet-344105.AdvertisementDataset.customer\_new`;

Case-2

**Create a table and insert values into it.**

Create a text document(.txt) and enter values(‘radhe’,23,False) …use comma separated values and save it as .txt format.

Go to (+) create a New table and import the .txt file as a CSV file. Select **UPLOAD.**

Follow the document **How to create a table.**

INSERT INTO AdvertisementDataset.Inventory\_new (product, quantity)

VALUES('rice cooker', 10) ;

select \* from optimistic-jet-344105.AdvertisementDataset.Inventory\_new;

Case-3

Create table option and Import the CSV file( instead of text file)

*But there are some limitations that BigQuery have.*

**Limitations**:

BigQuery does not limit the number of inert statements that write to a single table concurrently. First 1500 INSERT statements run concurrently during 24 hours of time period. After this a fixed number of statements run concurrently while other INSERT DML statements over the limit will be in queued to prevent system overload. Once a previous job finishes the next pending job is de-queued and run.

**Update, Delete and Merge Query**

Update, delete and merge statements are called mutating DML statements. The following discussion shows the query statements related to update, delete and merge statements.

**Update statement**

Update statement permits to modify existing rows in a table. Each update statement must include the where clause followed by a condition.

update my-first-project-345607.DatasetFirst.MovieTable\_new1 set Rating=3.8 where id=3987;

**Delete statement**

Delete statement allows to delete rows from a table. We must use where clause followed by a condition.

DELETE name FROM ‘project.dataset. customer’ WHERE id=3;

**Merge statement**

To merge two table contents depending on some conditions.

Suppose we have two tables such as *Inventory* and *NewArrivals*.

We can join target and source tables by using join clause. Depending on the match status the corresponding when clause with join query (inner join, outer join, full outer join) occurs.

|  |  |
| --- | --- |
| Product |  |
| Product | Quantity |
| dishwasher | 30 |
| dryer | 30 |
| font load washer | 20 |
| microwave | 20 |
| oven | 5 |
| Top load washer | 10 |

|  |  |
| --- | --- |
| New Arrivals |  |
| Product | Quantity |
| Dryer | 20 |
| Oven | 30 |
| refrigerator | 25 |
| Top load washer | 10 |

The folloing query explains the use of merge clause. For example

If an item exists in both the tables then quantity field is incremented otherwise items from the

NewArrivals table are included in the Product table.

merge dataset. product p  
using dataset.newarrivals s  
on t.product = s.product  
when matched then  
  update set quantity = p.quantity + s.quantity  
when not matched then  
  insert (product, quantity) values(product, quantity)

**Limitations**

* Mutating DML statements have limitations like BigQuery insert statements. It assigns jobs as ‘PENDING’ if already some DML jobs are still running. When a previously running job finishes, the next pending job is dequeued and run.
* A conflict arises when DML statements try to change or access the same partition.

**SQL statements Questions**

1. Find the total number of rows of a BigQuery table?

SELECT count(\*) FROM `optimistic-jet-344105.MovieDataset.LocalMoviedataset` ;

1. Create a table and copy the contents from old table to New table?

CREATE TABLE  my-first-project-345607.DatasetFirst.MovieTable\_new1 AS (SELECT \* FROM  my-first-project-345607.DatasetFirst.MovieTable WHERE id BETWEEN 3000 and 5000);

select \* from my-first-project-345607.DatasetFirst.MovieTable\_new1;

1. Join statement

**Inner join:** it returns rows that meet the join condition

select N.Movie\_name,N.id from my-first-project-345607.DatasetFirst.MovieTable as O INNER JOIN my-first-project-345607.DatasetFirst.MovieTable\_new1 as N ON O.Year = N.Year;

**Right outer join**

Returns all rows from the table on the right side of JOIN and matched rows from the left side of the JOIN

Select sum(ratings) from new table where movie\_names are same for both the tables.

select N.Movie\_name, sum(N.rating) from my-first-project-345607.DatasetFirst.MovieTable as O INNER JOIN my-first-project-345607.DatasetFirst.MovieTable\_new1 as N ON O.Year = N.Year group by Movie\_name;

