Data Analytics with BigQuery

Target Complete Date	@December 9, 2022
Status	Completed
Reviewed	Reviewed
# Time to complete (Hours)	1.5
Type	Cloud Guru

Introduction

- In Memory BI Engine
- Machine-learning capabilities
- Support for geospatial data storage and processing

Key features

1. High availability
2. Supports Standard SQL
3. Federated Data
    a. Data from outside BigQuery
4. Automatic Backups
5. Governance and Security
6. Separation of storage and compute

Managing data

Dataset = Database

Native tables: Standard Bigquery tables (held within bigquery storage)

External tables: Data held outside bigquery

- Schema inside BQ

Data ingestion

1. Real time events
    a. Streaming
2. Batch source
    a. Bulk load
    b. Data pushed to Cloud storage, Cloud dataflow picks it up. Processes it and pushes it to BigQuery

BigQuery SQL Dialects

- Legacy SQL
    - Non-standard SQL dialect
    - Migration to standard SQL is recommended
- Standard SQL

    - Preferred dialect

Using data in BigQuery

- BI Tools
- Cloud Datalab
- Export to sheets or storage
- Colleagues
- GCP Big Data Tools

Using BigQuery

Jobs

> ✏️ Job: Action that is run in BigQuery on your behalf (asynchronously)

Job types:

- Load
- Export
- Query
- Copy

Query job priority:

- Interactively (default)
    - Query is executed as soon as possible
    - Concurrent query running at any time + daily limit
    - Query results are saved to temporary table or permanent table
- Batch

    - Queued. It is executed when there are idle resources available in the shared job pool

Table storage in BigQuery

- Capacitor columnar data format
- Tables can be partitioned
- Individual records exist as rows
- Table schemes specified at load or at creation

Capacitor storage system

- Proprietary columnar data storage that supports semi-structured data (nested and repeated fields)
    - Data is converted from input format to capacitor format on load
- Each value stored together with a repetition level and a definition level

Denomalisation

- BigQuery performance optimised when data is denormalised appropriately
- Nested and repeated columns
- Maintain data relationships in an efficient manner
- Record (Struct) Data type

BigQuery allows for many input types including most main format types.

BigQuery views

> 💡 A view is a virtual table defined by a SQL query. A view can be accessed the same way you access a table. However it is unmaterialized meaning the underlying query is executed each time the view is accessed.

Benefits of views

- Control access to data
- Reduce query complexity
- Constructing logical tables
  - Organize similar information from different physical tables in BQ
- Ability to create authorized views

Limitations

- Cannot export data from a view
- Cannot combined standard and legacy SQL
- Cannot retrieve data from a view
- No user defined functions
- No wildcard table references

External (federated data source)

- Support for querying data from cloud Bigtable, cloud storage, google drive

Use cases:

- Load and clean your data in one pass
- Small, frequently changing data joined with other tables

Limitations

- No guarantee of consistency
- Lower query performance

- Cannot use tabledatalist API
- Cannot run export jobs on external data
- Cannot reference in wildcard table query
- Cannot query parquet or ORC formats
- Query results not cached
- Limited to 4 concurrent queries

Data Transfer service

Allows you to bulk transfer data from other data services.

Partitioning and clustering

Partitioning

Partitioning tables, allows you to break up a large table into many smaller tables. The different table partitions are stored separately at the physical level. Data is generally partitioned based on a single column. This improves query performance.

Ingestion time partitioned tables

Partition by load or arrival date. Data automatically loaded into date-based partitions (daily). Use _Partitiontime in queries to limit partitions scanned.

Partitioned tables

> 💡 Partition based on a specific column which has type, timestamp or date. Data partitioned based on value supplied in partitioning column. Use partitioning column in queries.

- __ null __
- __ unpartitioned __
- You must declare a table is a partition table during creation

Why use partitioning

> 💡 Using partitions means less data need to be read and processed. Limiting the data processed, by selecting specific partitions lowers the costs. Query performance is also improved.

Clustering

> 💡 Clustering is similar to creating an index in a table. It is only supported on partitioned table in bigquery. You should use clustering to improve aggregation results. The data associated with a cluster key is generally stored together.

- Ordering of cluster column is important. It is the order in which you should access information

Limitations

- Only supported for partitioned tables
- Standard SQL only for querying clustered tables
- Specify clustering columns only when table is created
- You can specify one to 4 cluster columns

BigQuery best practices

Slot

> 💡 Unit of computational capacity used to execute queries. It determines pricing and resource allocation.

- Number slots for query
  - Query size
  - Query complexitiy
    - Amount of information shuffled during query
- BigQuery automatically manages your slots quota
- Flat rate pricing available - purchase fixed number of slots
- View slot usage using stackdriver

Query plan execution

- Diagnostic query plan and execution timeline
- BQ is so fast because it leverages heavily distributed parallel architecture
- Declarative SQL statement
  - Query stages
  - Execution steps
- Query stage information
  - Stage overview
  - Step information
  - Stage timing classification
  - timeline metadata

BigQuery Best Practices

Three main topics

- Controlling costs
- Query performance
- Optimizing storage

Cost controls

- Avoid using Select *
    - Use preview options to sample data
        - Don’t pay to preview
- Price queries before executing
    - Convert bytes to price
- Using LIMIT does not affect costs
- View costs using a dashboard and query your audit logs
- Partition by date
- Materialise query results in stages
- Consider the cost of large result sets
- Use streaming inserts with caution
    - Go with bulk

Query performance

Dimensions

1. Input data and data sources
2. Shuffling
3. Query computation

4. Materialisation
5. SQL anti-patterns

Input data and data sources

- Prune partitioned queries
- Denormalise data whenever possible
- Use external data sources appropriately
- Avoid excessive use of wild card tables

- [ ] What is wild card

Query computation

- Avoid repeatedly transforming data via SQL queries
    - Use materialisation
- Avoid JavaScript user defined functions
- Order query operations to maximise performance
- Optimise JOIN patterns

SQL anti-patterns

- Avoid self-joins
- Avoid data skew
- Avoid unbalanced joins
- Avoid joins that generate more outputs than inputs
- Avoid DML statements that update or insert single row

Optimising storage

- Use expiration settings (automaticaly deleted at expiration)
- Take advantage of long-term storage
    - Lower monthly charges for cold storage

- Use the pricing calculator to estimate storage costs

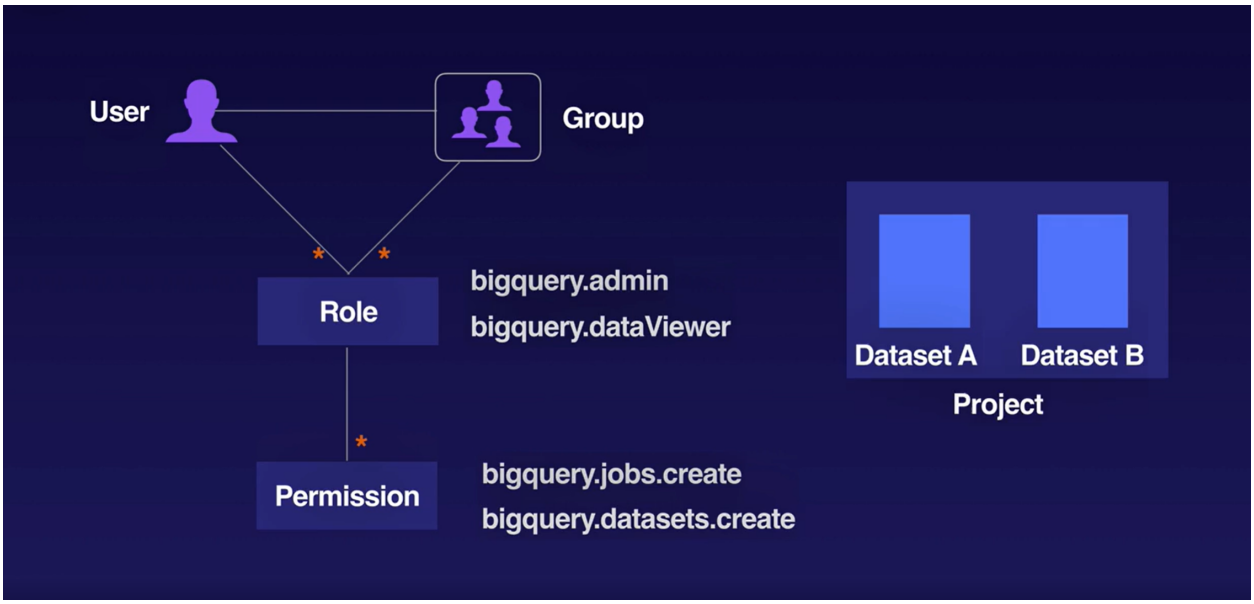
Securing BigQuery

Roles

> 💡 Access to data and functionality are controlled through roles

- Primitive roles: Defined at project level
  - Owner, Editor, Viewer
- Predefined roles
  - Granular access
  - Service specific
  - GCP managed
- Custom roles: User managed
  - Create and assign privileges to roles

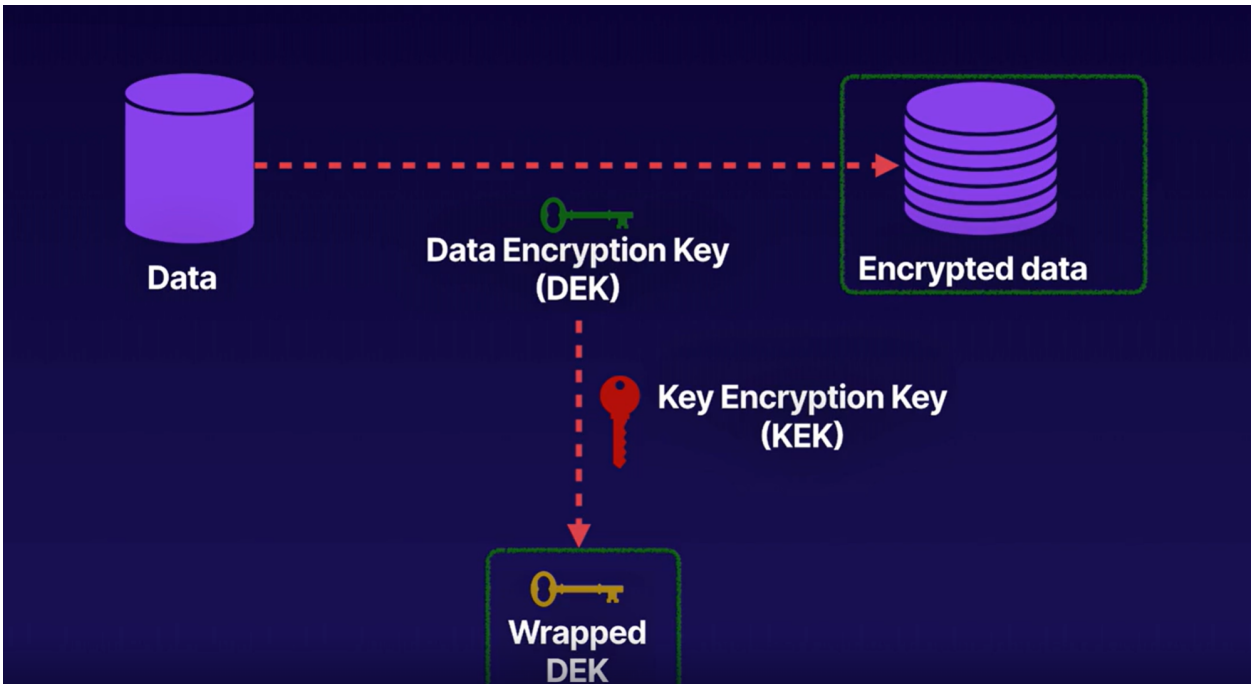
> ✏️ Permission: Very granular level of access.



Handling sensitive data

- Credit card numbers
- Medical information
- Social security numbers
- People’s names
- Address information

Cloud Data Loss Prevention API provides a fully managed service that automatically identifies sensitive information.



BigQuery Logging and Monitoring

> 💡 You should always be logging resources events and monitoring them. In general you use stackdriver to do your BigQuery monitoring. In Stackdriver you can undertake BigQuery monitoring. You can also set up Stackdriver alerts.

- Cloud audit logs are collections of logs offered by GCP to allow insights into the services.
- Look at Bigquery Audit Metadata logs. They are much more alligned with the state of BigQuery resources.

Audit logs

Divided into streams:

- Admin
- System events
- Data access

Machine Learning with BigQuery ML

Allows you to undertake Machine Learning in BigQuery using SQL.

- Jupyter notebooks
- GUI

Types of models:

- Linear regression
- Binary logistic regression
- Multi-class logistic regression
- K-means clustering
    - Number of points that can be seperated into clusters

Benefits of BigQuery ML

- Democratising ML
    - More people can use it
- Models trained and evaluated using SQL
- Speed and agility
- Simplicity
- Avoid regulatory restrictions
    - Moving data in cloud can result in contravening reg processes

ML process

1. Prepare data
2. Create and train the model - CREATE MODEL
3. Evaluate the model - ML.EVALUATE
4. use model for predictions - ML.PREDICT

Exam tips

- Understand good organizational design
  - Understand how different teams should be able to access data
  - Learn how you would grant access to authorized roles
- Cost controls
  - Optimization may come up in exam
- Partition tables appropriately
  - Clustering and partitioning
- Optimize query operations and JOINs
  - You should know how to write optimized SQL statements

☑️ ~~Look into SQL anti-patterns~~

Lab: Working with BigQuery

> ⚙️ Data can be ingested using JSON. It can also be exported as a CSV file. Saving results can send them to google sheets files.

Lab: Advanced BigQuery features

> ⚙️ This lab illustrated the benefit of partitioning data on efficiency and also created partition expiry.

- Why would you want to expire the partition?
  - Present a view from only a certain amount of days based on a partition
  - Data older than 60 days will expire
- You can share views with others

- You can also share specific views and datasets

Quiz

☑️ ~~Look into ACID compliance~~

IAM

bigquery.jobs.create