↳ for developing → Distributed System → want to decouple two Services

↳ Also messages uses a standard format

Plan to use Cloud PubSub

What Schema type are supported by cloud PubSub

- Avro
- Protocol Buffer
  - Thrift is alternative to this
    - ↳ but this is not supported

Paraquite → is open source file format

↳ used in Hadoop.

↳ Cloud firestor → Need to regularly Create → Backup

run in the Background

Backup file need to be store in Cloud storage

What CMD we can use

→ Gcloud data store Export gs://firestore-backup --async

↓

is the Cmd for → to save data to Cloud storage Bucket

↳ Gsutil → Only use to manage cloud storage

↳ not Cloud datastore

↳ Materialized View → hai → Bigquery ka → And got to know higher charges

As per Aspect

↳ fuquency of Material View refresh
↳ Data Store

→ These 2 thing ↑ Cost

Number of users → reading Data → from → Materialized View

Will not Effect Cost

But total amount of data Scanned

↳ Migrating a data → from data warehouse → On-Premises to google Cloud.

use of data Ware house

↳ BigQuery Bi → BiEngine → is an → fast → in memory analysis Service

↳ Cloud Memory Stor → is a cache and better suited → to Storing Key-value data for application

which needs low latency access to data

There is no bigtable Bi Engine Service

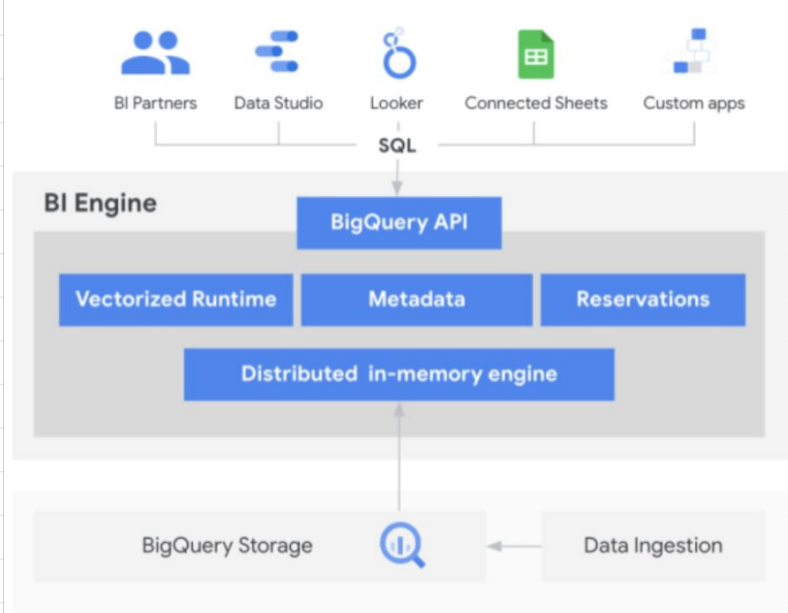
↳ BiEngine Can accelarate Sql queries from any source Including those

Written by data Visualization tools

Can managed Cached tables

↓

for On going optimization



↳ Bi Engine is useful in the following Case

↳ Certain tables that are queried most frequently.

Bi Engine might Not fit for

- Wildcards of queries
- Rely heavily on BigQueries feature with BI Engine doesn't support.

Here an unsupported features

↳ Including External tables

↳ Non Sql user-defined functions

↳ New workload → deployed → to → Cloud Dataproc

↳ configured with an autoscaling Policy.

got

FetchFailedException

↓

It happen when → Suffle Data is lost
↳ Nodes are decomissioned.

To avoid hot-spotting → in your Bigtable Cluster

↓

we are using the prefix of UUID prefix

↓

issue will be not working as we Expected

← There is hot-spotting writing Data to Bigtable

← Cause Cache

→ UUID → that are sequentially genrated

↓

It should use Version 4 and genrated random number genrator

Column families do not accept hotSpotting

Cost of Running → some data pipeline → for large Batch Jobs

↓

its resource Scheduling ← Data flow flux RS

↓

which Reduce Batch Processing Cost → will Reduce cost

↳ By using → Scheduling technique

↓

Preemitable VMs along with regular VMs

Dataflow Streaming Engine :→ is for stream processing.

Dataflow Shuffle → Provides faster Execution of Batch Jobs

↳ But doesn't reduce necessarily cost

Apach Beam → runner → require more management.

↳ Migration Several data warehouse → BigQuery

↓

Cloud storage for → Machine learning Data

ML Engineer ←

Data Analyst ←

are having difficult finding data set

↳ Cloud data Catlog → Can automatically Extract metadata

↳ from sources including

→ Cloud storage
→ Big Query
→ Cloud Bigtable
→ Cloud Pub/Sub
→ Google Sheet.

Cloud logging → used for recording data about Events

↳ not best way to Collect metadata.

Cloud fusion → is an ETL tool not a metadata

↳ Extraction tool

Monitoring System → hai → To → Apache Beam Runner Pe hai

↳ Temperature → received over past hour is analyzed

↳ If any temprature reading is more than Standard deviation

It Consistent time Interval

So it is Sliding Window / hopping Window

Consistent time interval in a Stream.

↳ Data warehouse team is concerned that some data sources may have Poor quality of Data

Dont want to Bring incorrect Data or invalid Data Into Data Warehouse

For this we must to do

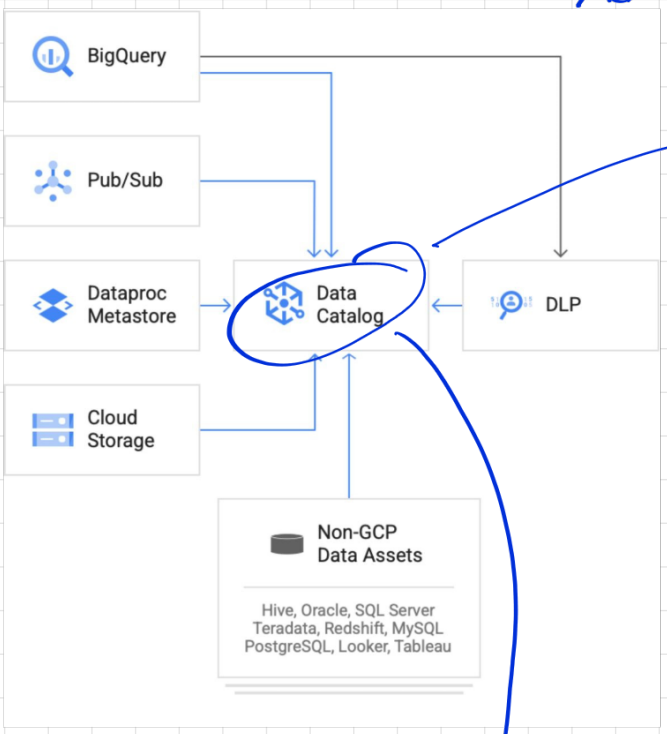
Data quality assessment

↓

From the source Data after it Extracted it from

Source System ←

↳ check of ranges
↳ distribution Values
↳ Attributs Count the number.



↳ Collect all metadata

↓

Provide Power Based Predicate Based Solution

Search Solution

↓

For technical and business metadata Solution

↓ ↓

Data catalog → not index a Data within a Data Entry

↳ Data Catlog Only Indexes the

→ Protect

↳ Data loss Prevention is to → help for → discover → Classify → most Sensitive Data

It will Provide Support to metadata

↳ managment

---

we have Compute Engine → attached GPU

↳ But GPUs not used when you train tensorflow model

→ how to Ensure the GPU can be used by training model

↓

Install GPU Driver

↳ Deploying VM image

↓

which already have installed GPU

⇒ MongoDB replacement → in GCP

↓

Cloud firestore

Bigtable

↳ wide column No Sql database → not good replacement

For mongoDB

Load Data from AVRO → Bigtable

↳ Dataflow → Cloud Storage Bigtable template

↳ Cloud Sql Auth Proxy ↓

Recomended way to Connect to Cloud Sql

↓

For Deploying a Cloud Sql databases to Prodution

Strong Encryption is used to Protect ↓

the Confidential and Integrated of Data

Not to Poform Authentication ←

Type of Data Model

Document Model → Support → Semi Structured Schemas
That frequently Change

Star Schema
Snowflake Schema
→ are denormalized relational Model
↓
used in Data Warehousing

Network Model → is used as → graph-like structure
↓
such as transportation network

* Cloud Pub/Sub → Delivery at least Once

Cloud Audit logs
this contain logs till 30 days
to keep them longer then that → we need to Preserve the logs for
↓
Cloud Storage

Cloud logging → doesn't Keep logs beyond 30 days
↳ doesn't support retention policy

Cloud monitoring → collects and display metrics
↳ it doesn't stores logs

Bigtable → is not a good option for logs
↳ low latency writes at high volumes and provides for key lookup queries.
↓
that requires range scanning

Overfitting and Underfitting → with respect to Data

↓
Poor performance → (ML)

↳ for Example → Supervised learning → (ML) is best understood as approximately target function (f)

$y = f(x)$

↓
Characterization describe → range of Classification
↳ Prediction Problem
↓
Machine algorithm
↑
That can be used to address them.

Cloud Spanner → Database having Performance issue

Or wo issue is with form of due to Problematic query

Where this is Popup

Query Insight

↳ Set of Prebuilt Dash Board

going to be faster for resolving issue

which makes it very easy to find Spikes in ↑↑ in performances

Query is use more than this Spikes will Pop up

↳ CPU usage

↳ Top query using the CPU in that time window

It will give more insight Such as

How many Sight → Rows Column

Retrived

It Enabled by default

↳ It is worth → Only query → Supported by query insight
↳ DML

Spanner mutation → Not available

↳ Cloud Spanner → update data → Data manipulation Language [DML]
↳ Spanner Mutation

```
List<Mutation> mutations =
    Arrays.asList(
        Mutation.newInsertBuilder( table: "Ledger") Mutation.WriteBuilder
            .set("CustomerId") ValueBinder<Mutation.WriteBuilder>
            .to(CustomerId) Mutation.WriteBuilder
            .set("AccountId") ValueBinder<Mutation.WriteBuilder>
            .to(AccountId) Mutation.WriteBuilder
            .set("TransactionId") ValueBinder<Mutation.WriteBuilder>
            .to(TransactionId) Mutation.WriteBuilder
            .set("Date") ValueBinder<Mutation.WriteBuilder>
            .to(java.time.LocalDate.now().toString()) Mutation.WriteBuilder
            .set("Amount") ValueBinder<Mutation.WriteBuilder>
            .to(Amount) Mutation.WriteBuilder
            .set("Details") ValueBinder<Mutation.WriteBuilder>
            .to(Details) Mutation.WriteBuilder
            .build(),
        Mutation.newUpdateBuilder( table: "Account") Mutation.WriteBuilder
            .set("CustomerId") ValueBinder<Mutation.WriteBuilder>
            .to(CustomerId) Mutation.WriteBuilder
            .set("AccountId") ValueBinder<Mutation.WriteBuilder>
            .to(AccountId) Mutation.WriteBuilder
            .set("Balance") ValueBinder<Mutation.WriteBuilder>
            .to(Balance) Mutation.WriteBuilder
            .set("LastTransactionTime") ValueBinder<Mutation.WriteBuilder>
```

```
public static void createAccount(DatabaseClient dbClient, String CustomerId){
  dbClient
      .readWriteTransaction()
      .run(transaction -> {

        String AccountId = String.valueOf(UUID.randomUUID());
        String CreatedOn = java.time.LocalDate.now().toString();
        String Balance = "0";
        String sql =
            String.format("INSERT INTO Account (CustomerId, AccountId, CreatedOn, Balance) "
                + " VALUES ('%s', '%s', '%s', %s)"
                , CustomerId, AccountId, CreatedOn, Balance);
        long rowCount = transaction.executeUpdate(Statement.of(sql));
        System.out.printf("%d record inserted with AccountId %s.\n", rowCount, AccountId);
        return null;
      });
}
```

DML → Insert / Update / Delete SQL

→ Standard DML
↓
Great for single statement

→ Partition DML
↳ special implementation for bulk data changes

DML → If you change a Data → that change will remain to later Statement
↓
within the same transaction

Mutation → Sequence of { Insert, Updates, delete } → Make Changes table or rows

without using Sql layer

Still made with in ACID transaction

Changes → are → done → Jise order mai specified hai

↳ Buffered in client locally and then sent to spanner when transaction is finished

Due to that Mutation donot support read after write within same transaction

Constrain is → Check at Commit time rather after each mutation

DML → great for OLTP use

Large Change Operation use → Mutation
↳ Partition DML

↳ If you want to check → Read / Change → within same transaction

Or Constrain should be Check after each Statement → use DML

→

DML or Mutations?

Operations	DML	Mutations
Insert Data	Supported	Supported
Delete Data	Supported	Supported
Update Data	Supported	Supported
Read Your Writes (RYW)	Supported	Unsupported
Upsert (insert or update)	Unsupported	Supported
SQL Syntax	Supported	Unsupported
Constraint Checking	After every statement	At commit time

For Example → finacial Bank hai
Jab account create kar rahe hai
Single query hoga
Then DML

```
public static void createAccount(DatabaseClient dbClient, String CustomerId){
  dbClient
      .readWriteTransaction()
      .run(transaction -> {

        String AccountId = String.valueOf(UUID.randomUUID());
        String CreatedOn = java.time.LocalDate.now().toString();
        String Balance = "0";
        String sql =
            String.format("INSERT INTO Account (CustomerId, AccountId, CreatedOn, Balance) "
                + " VALUES ('%s', '%s', '%s', %s)"
                , CustomerId, AccountId, CreatedOn, Balance);
        long rowCount = transaction.executeUpdate(Statement.of(sql));
        System.out.printf("%d record inserted with AccountId %s.\n", rowCount, AccountId);
        return null;
      });
}
```

When → update the account → Mutation
↳ Enter the Details into ledger
As we don't need to

Read the Data after the changes.

DML Statement And Mutation Statement should be matched within the same transaction

↳ Cloud Spanner → like we can use granular instance sizing
- for low use Case
- lower cost

↳ Availability
↳ Consistency
↳ Performance

Spanner → Easily → Manage local → without require complex sharding.
↳ Support Scalability by → Separating → Support
- Storage

↳ BigQuery → Support materialized Views
- Precomputed Views that Periodically cache the result of a query of increased.
- Performance
- Efficiency

Queries that use → Materialized Views → generally faster
↳ Consume fewer resources than queries that retrieve the same data only from base tables.

Materialized Views → Key characteristic → Zero maintenance
↳ Flush Data
↳ Smart tuning

Process get Benifits
↳ from Views → OLAP Online Analitics processing
↳ ETL
↳ BI

Improve query performance → Preaggregate Data :- Aggregation of Streaming Data
↳ Pre-filter Data → Run query only in Particular Subset
↳ Prejoined Data
Query Joins, Espically between Large and small tables