dbt

classmate
Date
Page

Interview Tech m

Data flow, data proc, redrate in bq, External tables in bq, partition, clustering, challenges in project, real scenarios, queries, primary key in bq, pub/sub, DAT tool, bq slots, data modelling, difference b/w external and internal table. ways to Export from Bq

GCP Services for Data Analytics

Bigquery → It is a serverless and cost-effective enterprise data warehouse that works across the cloud and scales with your data.

Looker → It is a Business Intelligence tool and data analytics platform for data exploration, modelling, transformation and visualization

Dataproc → It is fully managed and highly scalable service for running Apache Hadoop and Apache Spark.

Dataflow → It is a fully managed streaming analytics service that minimize late

processing time, and cost through autoscaling

Dataform → It helps analytics teams transform data in BQ using SQL. With dataform data engineers, data analyst can develop BQ-table definations.

Normalization

It is a process to reduce redundancy and dependency. The primary goal of normalization is to eliminate data anomalies and ensure data integrity.

It which all anomalies are solved in Normalization

① Insertion Anomaly: No incomplete or Contradictory info in insertion.

② Deletion Anomaly: Prevents loss of invaluable info when deleting data.

③ Update Anomaly: Update are made in away to maintain data integrity.

3) Jenkins is a CI/CD Tool that works
a)
Checkout → Build → Test → Deployment
or
Plan → Code → Push → Build → Test → Deploy

Works on cron syntax

Minute, hour, dayofmon, month, dayofweek
0 2 * * *

Means it will trigger daily at 2AM

4 Global variable in python is defined outside all the function that is accessed by all the function.

If you want to modify the value of global variable within a function you need to use 'global' keyword

global variable = 10

Round function syntax

round (number [, ndigits])

Google Cloud logging

It is a service by GCP that allow you to store, search, analyze and monitor logs generated by your applications and infrastructure.

Window function

These function in SQL are used to perform calculation across a specified range of rows related to the current row.

Ranking → Select employee_id, salary, RANK() over (order by salary Desc) As salary_rank from employee;

Running total → Select o_date, o_amt, sum(or_amt) over (order by o_date) As running_total from orders;

Three types of window function are:

1. Aggregate window function
2. Ranking Window function
3. Value Window function

NTILE() → divides the table into n buckets.

Value Window function

- Lag() → To access previous value of row before the current row value.

Syntax: LAG(Revenue, 1, 0) OVER(PARTITION BY org ORDER BY [year])

- LEAD() → It is used to access the subsequent row along with data of current row.

eg: lead(order) OVER(Order by [order])

Benefits of Compute Engine:

- Storage Efficiency.
- Stability.
- Easy Integration.
- Security
- Compute globally as per requirement

- Default bucket location is US region
- Compute Engine let you create VM on google's infrastructure

- The data in VM depends on type of disk we If it is persistant disk the data is retained if instance stops. In case of local SSD the data is not retained

IAM (Identity and Access Management).

- Storage and Data Services.

Object	Relational	Non-Relational (Wareh
Cloud Storage (GCS)	Cloud SQL Cloud Spanner	Cloud Firestore Cloud Bigtable BO

4. Three Nosql database in GCP.
   1. Datastore
   2. Cloud Firestore
   3. Cloud Bigtable

Difference b/w these :-

Count(*), Count(1)

```
select Count(*) from emp;
select Count(1) from emp;
select Cont(distinct 2) from emp;
selet 1 from emp;
```

Deloitte Interview

Object Storage	Relational	Non-Relational	Warehouse
GCS	CloudSQL	Cloud firestone	BQ
	CloudSpanner	Cloud Bigtable	

CloudSQL → This fully managed Relational database service that supports MySQL, Postgresql and SQL Server.

feature → Auto backups, replication of data for high availibily, automated patch management

Cloud Bigtable → Cloud Bigtable is a fully managed highly scalable NoSQL database service that it designed to handle large amount of data with high read write speed.

DataFlow → It is a managed service for executing a wide variety of data processing patterns.

Dataproc → This fully managed service used to run Apace hadoop and spark clusters.

Databricks and dataprocc does the same large-scale data processing

Datalab → It is an interactive tool for analysing, visualizing data on GCP. It is designed to work with big data technologies such as BQ, Apache Spark. It provides Jupyter Notebook environment that integrates with various GCP services.

These Notebook are stored in google source repositories, anyone can collaborate. Virtualization on the cloud, we can install the datalab using "gcloud component install datalab".

Dataprep → It is a cloud-based data preparation platform that helps organisation clean, enrich, and transform raw, messy data into format suitable for analysis and reporting.

Helps in streamlining the data preparation process for technical and non-technical users to work with data.

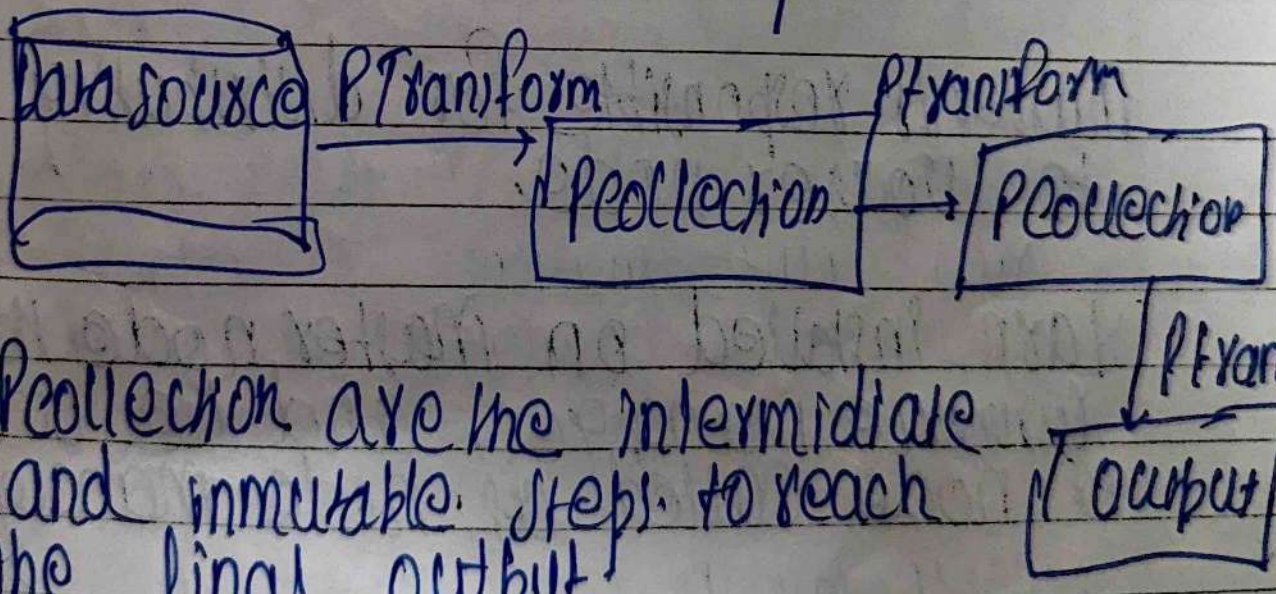
It is managed by Trifacta, and suggests

Intelligent data tranformation methods and helps collaborate with other users

Pub/Sub: It is fully managed cloud, real-time messaging service that allows you to send and receive message b/w independent applications.

Dataflow

It is a managed service to run data processing in a wide variety of patterns with the available default templates.



PCollection are the intermidiate and immutable steps to reach the final output

eg: To read the word count on gcs and post the result on the gcs.

Bigquery: It's fully managed serverless data warehouse and analytics platform provided by google cloud platform. It is used for reporting and analytics purp

Spark Architecture

- Multiple machine connected together called as Cluster.

Work on master-slave architecture, the slave also called as worker node

Master is responsible for distributing the to worker node

Yarn installed on master node that is resource manager and the worker node called as node manager.

Questions from architecture -

- Driver → The main program that create SparkContext, coordinating the distrib

Operations.

Spark Context → The Entry point to any spark functionality.

Cluster manager, Manages resources across spark applications on cluser such as yarn, standalone.

RDD → The Resilient Distributed Dataset is the fundamental data structure in Spark, representing a distributed collection of objects that can be processed in parallel.

Spark SQL This module of Spark for structured data processing. It provides a SQL wrapper and supports various source like Parquet, Avro, JSON etc. Bridges gap b/w RDD and relational tables.

Executor → These are worker nodes responsible for executing the tasks on spark cluster.

Narrow and wide transformations.

- Narrow transformations involve one-to-one mapping and do not require shuffling of data.
- Wide transformation involve shuffling and redistribution of data across partitions.
- Spark driver program is responsible for creating Spark Context (Entry point for spark job), coordinating tasks, managing the overall execution of spark application.
- It run main function and communicates with cluster manager to allocate resources and create tasks.
- Transformations are operations applied to RDDs to create new RDD.

Narrow → filter, union, map, flatmap.
wide → groupby, reduceby, join

Google Transfer Appliance → It provides an easy and secure way to physically ship large amounts of data to the cloud.

It is a briefcase sized appliance shiped from google to the organisation to upload data and return back to google so it can be uploaded to the cloud storage. It comes in 2 size capacity 40TB and 300TB.

Cloud Storage transfer service → It is a managed service of cloud to transfer data from GCP or any other cloud platform to GCS. We can migrate the data from the data transfer service job without writing any code.

We can schedule it as per requirement that is one time or recurring basis.

Bigquery data transfer service → It enables you to move data into BQ from variety of sources. Transfers can be both one time as well as ongoing or scheduled.

Docker is a platform that allows you to pack-age and run applications in containers. A container is like a lightweight, standalone and executable package that contains everything needed to run a piece of software, including the code, runtime, software, libraries. It ensure the code or application runs consistently across different environment. And solves the problem of 'This runs on my machine'.

Kubernetes is like conductor for your software. It helps in managing all the part of complex application, it helps to coordinate and orchestrate your software component ensures they play together seamlessly.

Kubernetes helps us :-

1. Deploy applications
2. Scale Applications
3. handle faliure
4. Update Applications.

Lazy Execution in Spark

Lazy Execution refers to strategy where transformations on RDD (Resilient Distributed Dataframe) are not immediately executed. Instead, they are recorded and the actual computation is deferred until action is invoked.

It is a key optimization feature in spark that helps in improving the efficiency of distributed data processing.

Window Functions Questions

Functions that allow to perform calculations on or across range of rows related to the current row.

- Row_Number()
- Rank()
- Dense_RANK()
- NTILE()
- SUM()
- AVG()
- MIN()
- MAX()

Table Sales

P_id	sale_date	revenue
1	=	=
2		

① Calculate running total of revenue for each product over time.

```
Select P_id, sale_date, revenue,
Sum(revenue) over (partition by p_id order by
sale_date) As total from sales.
```

② Assign a rank to each product based on their total revenue.

```
Select p_id, revenue, RANK() over (order by
Sum (revenue) desc) as revenue rank
from sales group by p_id.
```

③ Row Number assign unique number to each employee based on their salary in descending order.

```
Select e_id, e_name, salary,
Row_NUMBER() over (ORDER BY salary desc)
as Salary_rank from employee.
```

④ give me the rank of employee based on salary in a emp table, using CTE

4 with Ranked Employee As (
Select e_id, ename, salary,
RANK() OVER(Order by salary desc) As salary_rank
from employee
)
4 Select eid, e_name, salary, salary_rank
from Ranked employee.

5 Give 2nd highest salary using Window fun.
with Ranked Employee As (
Select eid, ename salary,
DENSE_RANK() OVER(orderby by salary desc)
as salary_rank from employee)
Select ename, salary from Ranked Employee
where salary_rank = 2.

Date
Page

Optimizing SQL Queries

We need to adress the performance bottleneck, to make the query more efficient.

1. Use Indexing. (Columns in where, join, order by) are indexed.
2. Analyze Execution plan (Explain before the query).
3. limit the result set.
4. Partitioning large table.
5. Avoid Select * as it fetches unnecessary columns.
6. Check for Blocking or locking issues (the queries might be blocked by another transaction).
7. Review and Optimize Join Condition.

classmate

Spark SQL transformations

optimizing Spark SQL Jobs

1. Choose appropriate file formats. as Parquets often more efficient than CSV or JSON.
2. leverage partitioning and bucketing to organize data on the disk as it reduces amount of data to be scanned.
3. Using broadcast join for smaller datasets or dataframes.
4. Using a broadcast variable as a local copy is stored to each worker nodes that reduces data transfers
5. Tuning shuffle operations we can adjust the partitions based on size of data. default is 200. spark.sql.shuffle.partitions.
6. Dynamic partition Pruning. → It is a technique of Spark to push the unnecessary data and pushdown the required data for better data retrival.

Set spark.sql.adaptive.enabled = True

Lifecycle rules

(7) Sufficient hardware resources allocated to cluster

5. Spark UI and Spark history server are used to identify the bottlenecks and optimize your queries accordingly.

6. There is an complex query with multiple subquery and join condition. How can you optimize it

(1) Examine the execution plan.

(2) Ensure Join condions and where clause have indexed column.

(3) Review the subqueries and evaluate if they can be optimized or replaced with Joins or Exists clause.

(4) Limit the result set.

(5) Consider materialized views if the query is having heavy aggregation or computation

(6) Use CTE (Common table Expression).

⑦ Partitioning table

Materialized View → It is a type of view which stores the data to avoid recomputing the results or making the heavy queries easy to execute.

create materalized View Ab as
select pid, sale from sales;

In DWH, we create materialized views a summary table by which we can improve the performance of analytical queries.

Why we use Views → Virtual table

① Views encapsulate complex queries into single name.

② Views can be used to control access to underlying table.

③ Views promotes code reusibility. Instead of writing complex queries again.

Kaa Cadars.

Cloud Composer: It is fully managed apache airflow service by which we can create, schedule, monitor, and manage our work-flows.

In data Engineering, workflow represents a series of tasks for ingesting, transforming, analyzing or utilizing data. Airflow creates workflows using DAG's.

Despite of several connectors available to migrate data, why we don't use in live environment.

1. Downtime Concerns.
2. Data volume and velocity.
3. Network reliability.
4. Custom logic on the transformations
5. Fear of data loss.

Teradata architecture Vs BQ Architecture

Bigquery architecture is decoupled architecture that is storage and compute are separated making it less expensive.

Colossus → It is a successor of GFS (Google file system) that is used for storage.

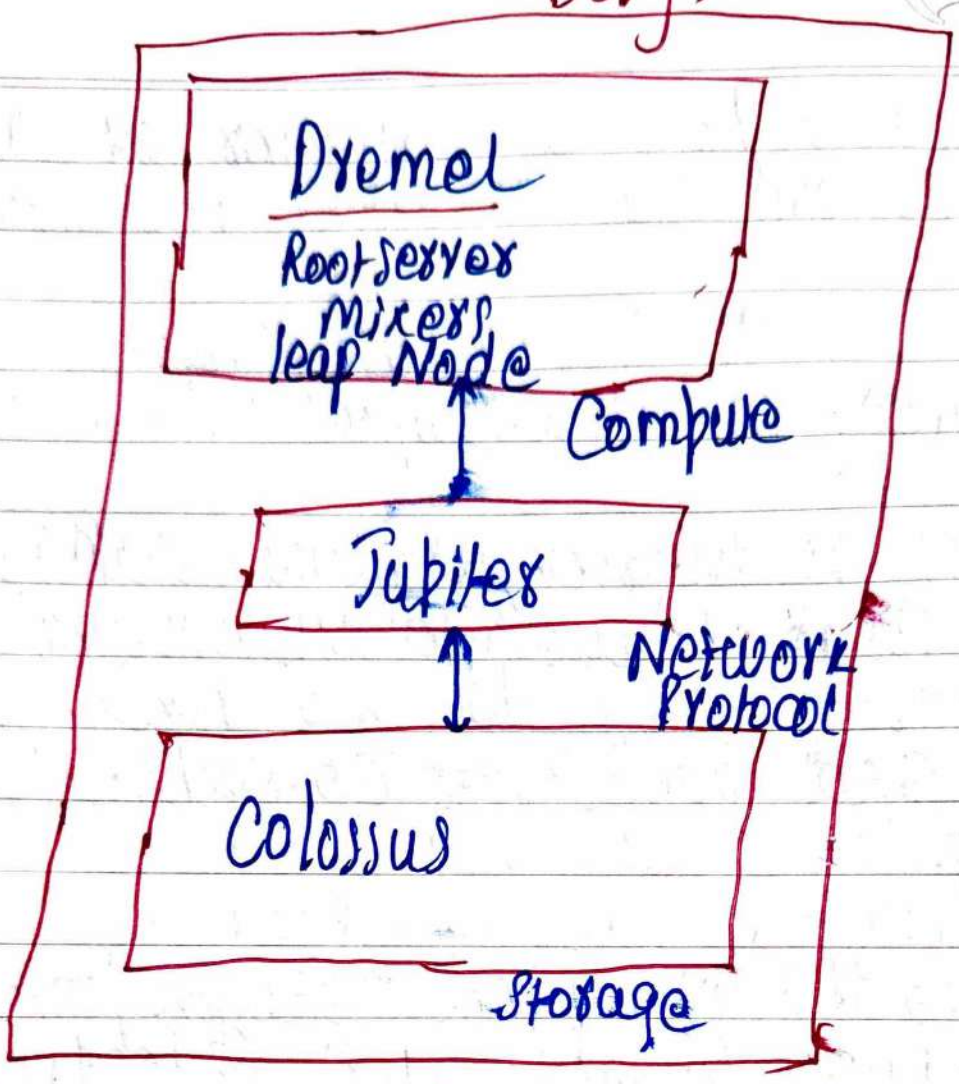
Dremel → It is Compute Engine to process the query to give output.

It has root Server, Mixers, leaf Nodes. where leaf node is used to read data from Colossus. Mixers does the heavy aggregation part and root Server co-ordinates everything.

Jupiter → It is high speed network protocol to connect Dremel and Colossus. called as petabit Network protocol.

And whole architecture is orchestrated by Borg. that co-ordinates and schedules the overall architecture.

BQ is column oriented Database or Storage.



Read a csv, infer schema then filter the record and write to a table

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('my
    getOrCreate().

my_schema = StructType(
    [StructField('id', IntegerType(), True)
)
```

```
df = spark.read.format('csv').option('mode':
    .option('schema': my-schema)
    .load('path')

transformed_df = df.filter(col('id')>100)

transformed:df.write.format('parquet').
    saveAsTable('db.tablename').
```

HSBC

- SQL Transformer in GCP
- life cycle rules in buckets
- Decoupled BQ architecture
- storage class
- Python code to find hobbies starts with T in a dic list of dictionaries
- create a cluster table
- hadoop coupled system
- BQ table is clustered on year, month, day and hour, all are int. find the data in date range (use partition concept).
- find percentage like in table of emp.
- how BQ stores and compute data

PwC

- Introduction
- past Experience
- Questions related to project (Data loading process, how to validate data)
- Volume and size of objects
- SQL query to get the latest and oldest record from SCD table.
- What is SCD and how we apply SCD on table. What is its advantages.
- Python code to find frequency of each word in String.
- SQL to get the 2nd highest of the Salary.
- Apache airflow, basics, dags, RDD.
- BQ architechture
- Python code to read a csv and give the top 5 apps from one dataset of playstore.

Infosys (2 Rounds).

1st Round

- Intro and role in project
- questions on projects
- BQ architecture
- nested and repeated fields in BQ.
- ways to export data from table in BQ.
- Python basic questions, list, dictionary
- what is set and tuple
- a scenario where we can use tuple
- Spark architecture
- Migration experience questions

2nd Round

- Intro and project responsibility
- Python proficiency.
- SQL question on rank and dense_rank()
- GenAI basic knowledge
- Any example of GenAI.

Public Sapient

↳ 9. have an input df with two colum i want to add new colum in the df with True or false.

↳ write full syntax to read a csv file in spark
↳ write syntax to declare the schema in spark.
↳ what is infer schema, header
↳ what if inferschema is false

↳ Input → name, math, english
Sau 70 30

Out → name subject mark
Sau Mam 70
Sau Eng 30

↳ Table join count Questions

↳ bank, grade, loan
icici a 1
hdfc a 3
fnb b 2

out → bank, grade, score, comp-avg score

icici	a	1	2
Hdfc	a	3	2
Pnb	b	2	2

↳ com_avg → competetion average score.

↳ Questions related to project loding data.
↳ how loding was done
↳ many counter question on loding volume
↳ why used manual approach to load dat
↳ Transfers services.