Establish Ingestion and Extraction patterns using Azure and Snowflake. (For Batch-time and Real-time)

Guided by- Amit Malik

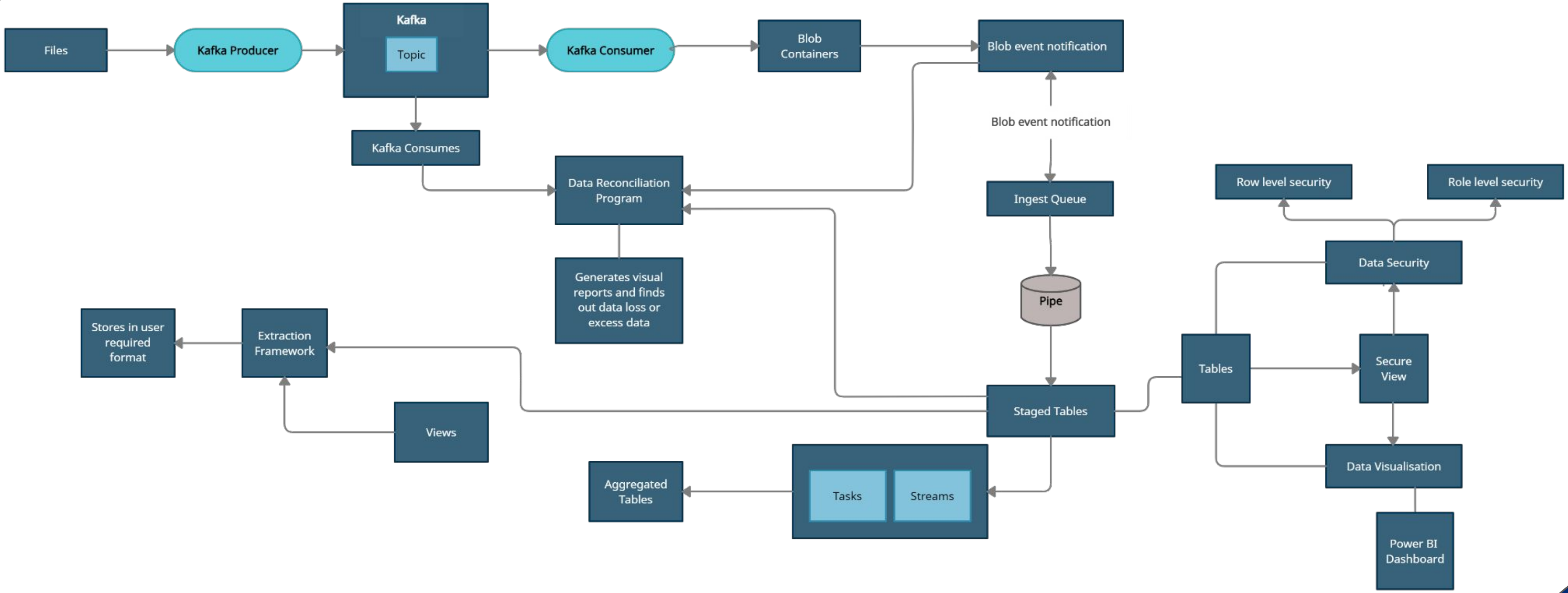
Team Members-

Ashwin Nair

Rajat Mishra

Nehal Tiwari

Work-flow:

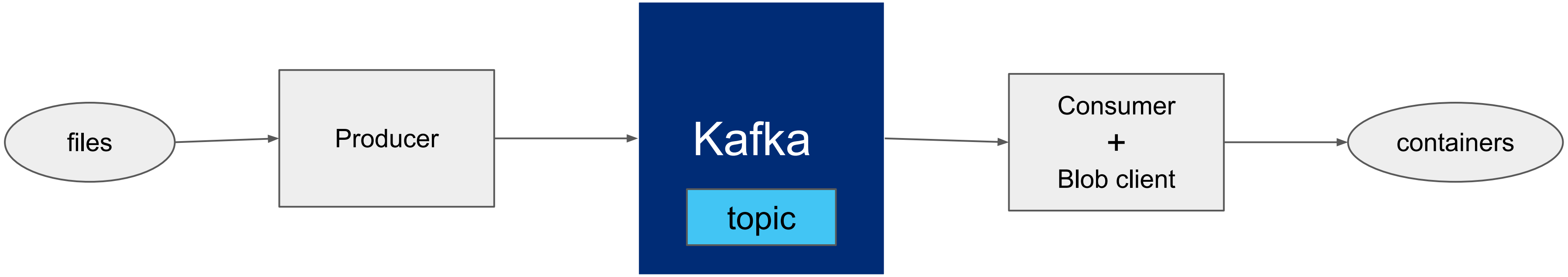


Files
Kafka Producer
Kafka
Topic
Kafka Consumer
Blob Containers
Blob event notification
Kafka Consumes
Data Reconciliation Program
Blob event notification
Ingest Queue
Row level security
Role level security
Generates visual reports and finds out data loss or excess data
Pipe
Data Security
Stores in user required format
Extraction Framework
Tables
Secure View
Views
Staged Tables
Aggregated Tables
Tasks
Streams
Data Visualisation
Power BI Dashboard

Data Ingestion

kafka to azure

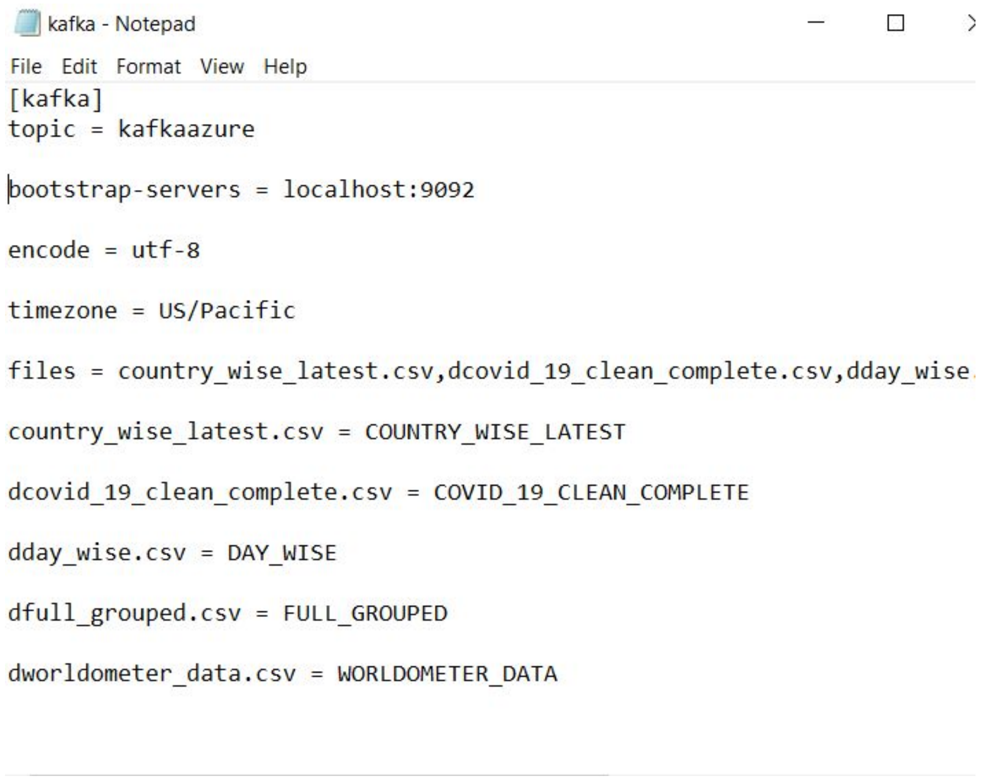
Azure blob containers



- Each kafka topic is mapped to a single azure storage which is in turn mapped to a snowflake database.

Data Ingestion

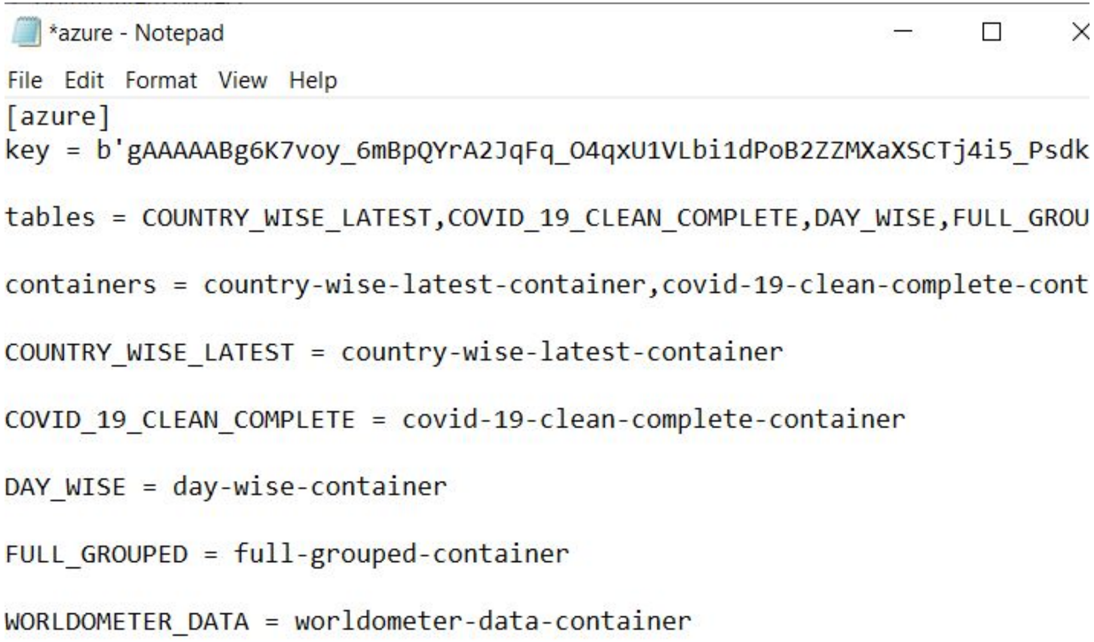
Kafka config specifies topic and maps each file to table



```
[kafka]
topic = kafkaazure

bootstrap-servers = localhost:9092

encode = utf-8

timezone = US/Pacific

files = country_wise_latest.csv,dcovid_19_clean_complete.csv,dday_wise

country_wise_latest.csv = COUNTRY_WISE_LATEST

dcovid_19_clean_complete.csv = COVID_19_CLEAN_COMPLETE

dday_wise.csv = DAY_WISE

dfull_grouped.csv = FULL_GROUPED

dworldometer_data.csv = WORLDOMETER_DATA
```

Data Ingestion

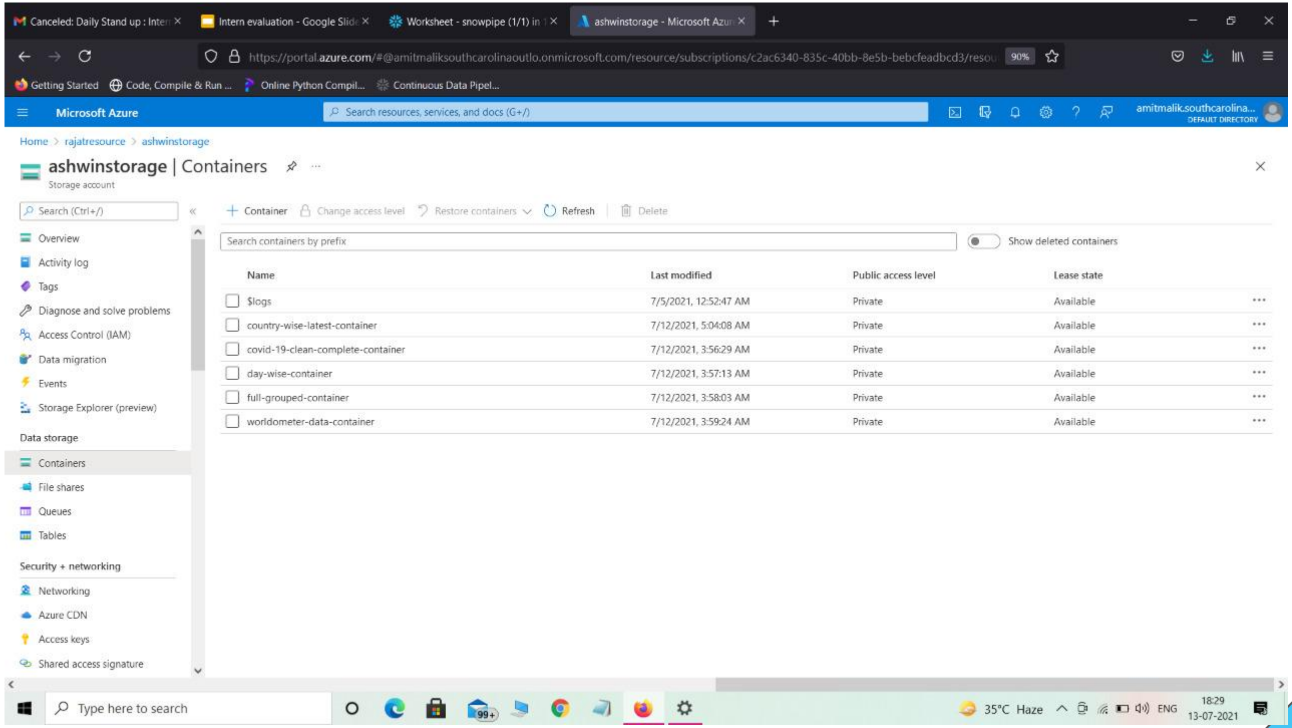
Setting up azure config file for consumer code to connect to Azure



```
[azure]
key = b'gAAAABg6K7voy_6mBpQYrA2JqFq_O4qxU1VLbi1dPoB2ZZMXaXSCTj4i5_Psdk

tables = COUNTRY_WISE_LATEST,COVID_19_CLEAN_COMPLETE,DAY_WISE,FULL_GROU

containers = country-wise-latest-container,covid-19-clean-complete-cont

COUNTRY_WISE_LATEST = country-wise-latest-container

COVID_19_CLEAN_COMPLETE = covid-19-clean-complete-container

DAY_WISE = day-wise-container

FULL_GROUPED = full-grouped-container

WORLDOMETER_DATA = worldometer-data-container
```

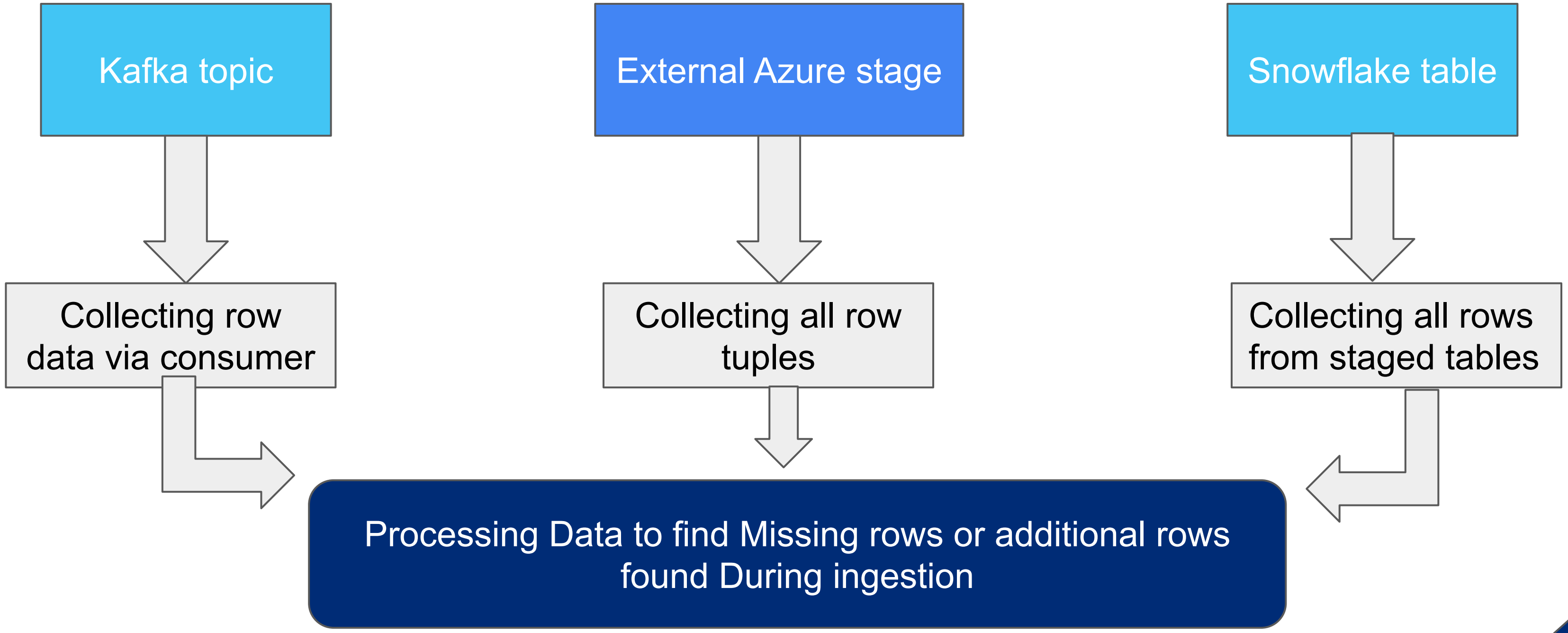
Data Ingestion

Image depicting the mapped containers inn azure storage


Canceled: Daily Stand up : Inter
Intern evaluation - Google Slide
Worksheet - snowpipe (1/1) in
ashwinstorage - Microsoft Azur
https://portal.azure.com/#@amitmaliksouthcarolinaoutlo.onmicrosoft.com/resource/subscriptions/c2ac6340-835c-40bb-8e5b-bebcfeadbcd3/resou
90%
Getting Started
Code, Compile & Run ...
Online Python Compil...
Continuous Data Pipel...
Microsoft Azure
Search resources, services, and docs (G+/)
amitmalik.southcarolina...
DEFAULT DIRECTORY
Home > rajatresource > ashwinstorage
ashwinstorage | Containers
Storage account
Search (Ctrl+/)
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage Explorer (preview)
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Azure CDN
Access keys
Shared access signature
Container
Change access level
Restore containers
Refresh
Delete
Search containers by prefix
Show deleted containers
Type here to search
35°C Haze
ENG
18:29
13-07-2021

Name	Last modified	Public access level	Lease state
$logs	7/5/2021, 12:52:47 AM	Private	Available
country-wise-latest-container	7/12/2021, 5:04:08 AM	Private	Available
covid-19-clean-complete-container	7/12/2021, 3:56:29 AM	Private	Available
day-wise-container	7/12/2021, 3:57:13 AM	Private	Available
full-grouped-container	7/12/2021, 3:58:03 AM	Private	Available
worldometer-data-container	7/12/2021, 3:59:24 AM	Private	Available

Data Reconciliation

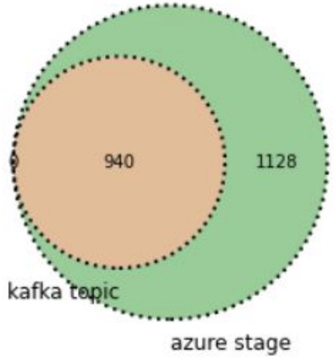


Kafka topic
External Azure stage
Snowflake table
Collecting row data via consumer
Collecting all row tuples
Collecting all rows from staged tables
Processing Data to find Missing rows or additional rows found During ingestion

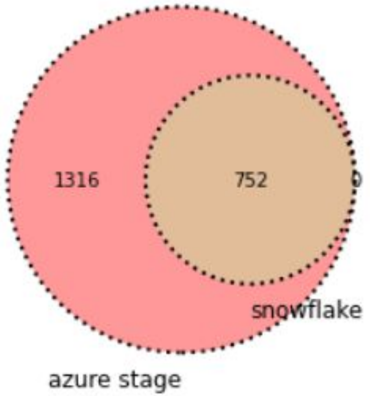
Data Reconciliation

Accessible visuals and dataframes in python program

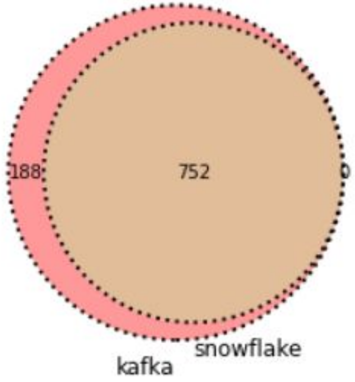
```
data reconciliation for DAY_WISE
0 940 1128
```



```
1316 752 0
```

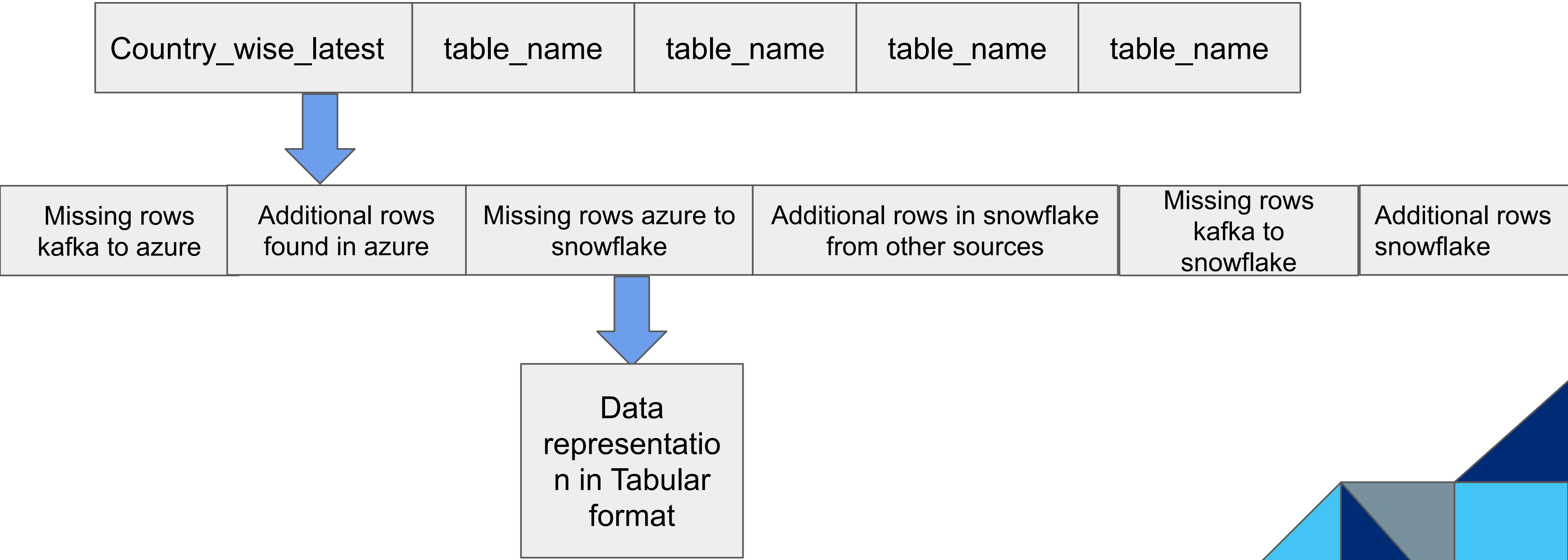


```
188 752 0
```



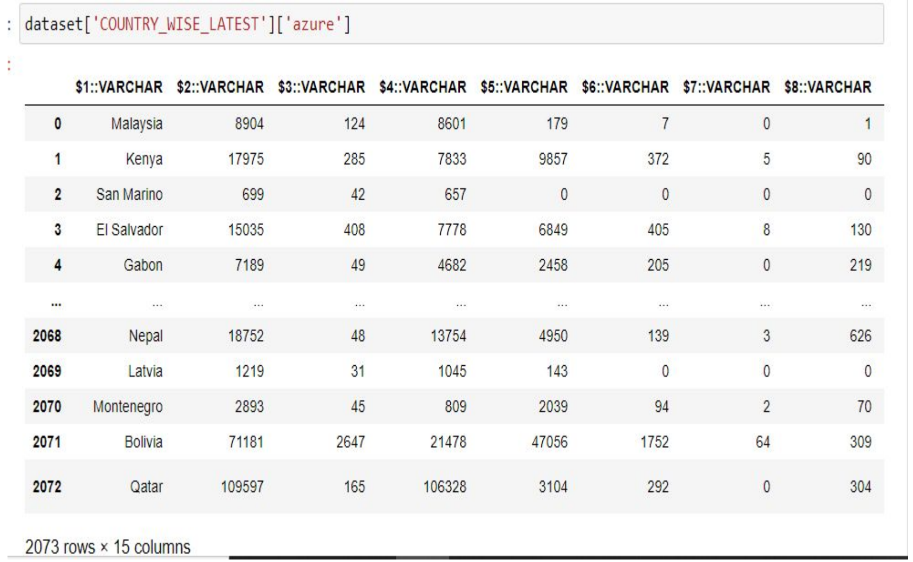
Data Reconciliation

The program provides a dictionary of dictionary of tables with missing Data


Country_wise_latest
table_name
table_name
table_name
table_name
Missing rows kafka to azure
Additional rows found in azure
Missing rows azure to snowflake
Additional rows in snowflake from other sources
Missing rows kafka to snowflake
Additional rows snowflake
Data representation in Tabular format

Data Reconciliation

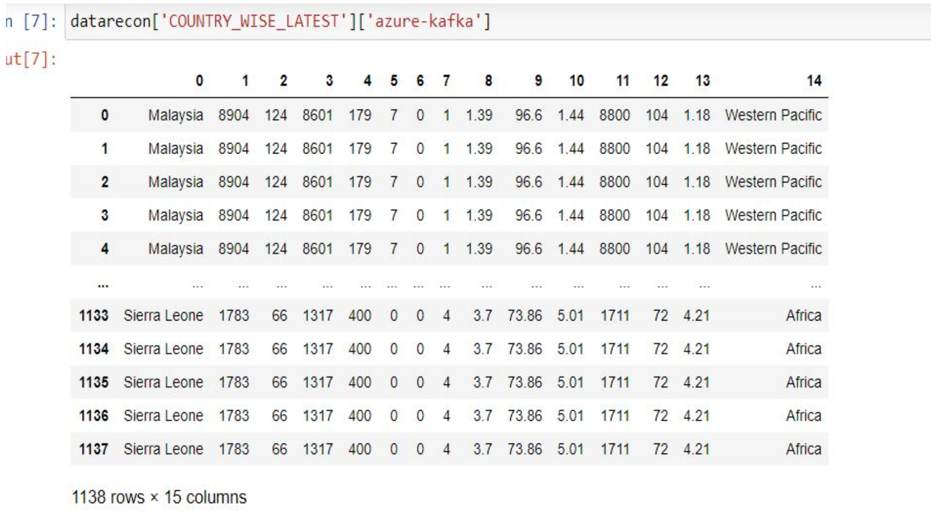
Identification of Missing Records



```
: dataset['COUNTRY_WISE_LATEST']['azure']
:
```

	$1::VARCHAR	$2::VARCHAR	$3::VARCHAR	$4::VARCHAR	$5::VARCHAR	$6::VARCHAR	$7::VARCHAR	$8::VARCHAR
0	Malaysia	8904	124	8601	179	7	0	1
1	Kenya	17975	285	7833	9857	372	5	90
2	San Marino	699	42	657	0	0	0	0
3	El Salvador	15035	408	7778	6849	405	8	130
4	Gabon	7189	49	4682	2458	205	0	219
...	...	...	...	...	...	...	...	...
2068	Nepal	18752	48	13754	4950	139	3	626
2069	Latvia	1219	31	1045	143	0	0	0
2070	Montenegro	2893	45	809	2039	94	2	70
2071	Bolivia	71181	2647	21478	47056	1752	64	309
2072	Qatar	109597	165	106328	3104	292	0	304

2073 rows × 15 columns



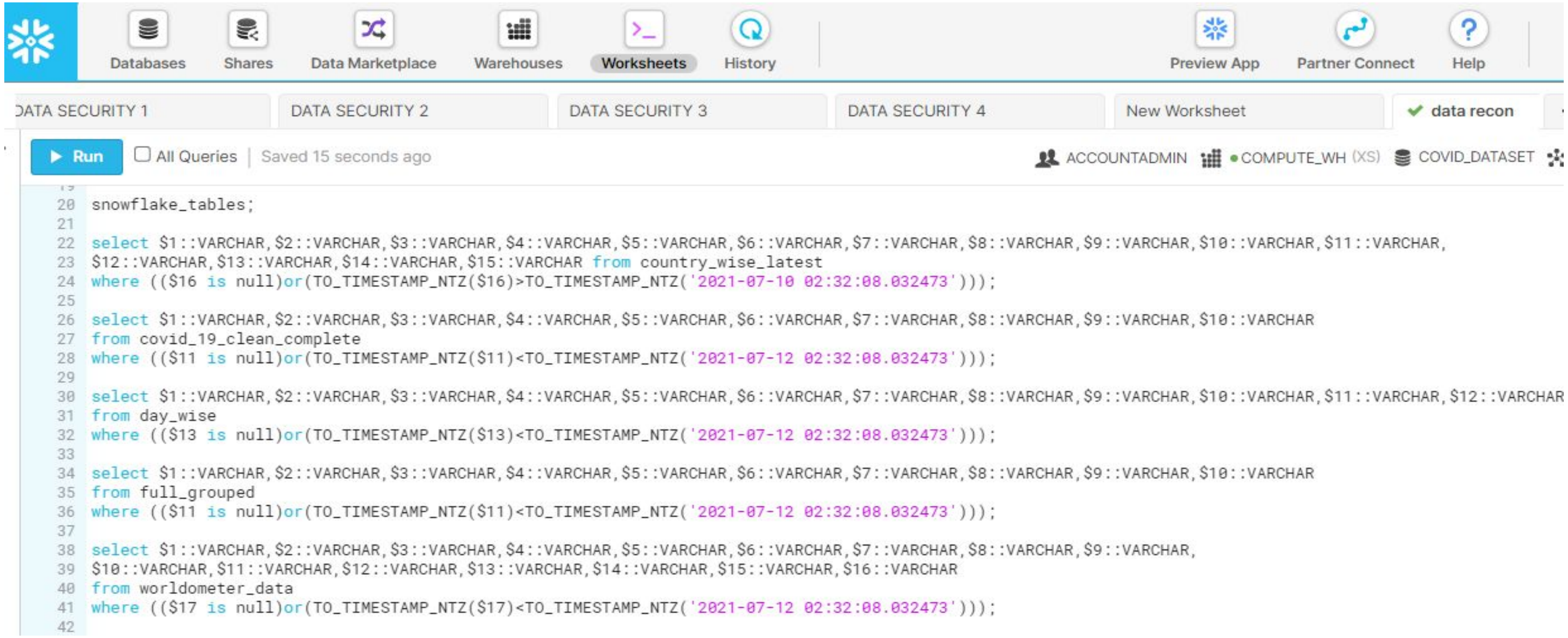
```
n [7]: datarecon['COUNTRY_WISE_LATEST']['azure-kafka']
ut[7]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	Malaysia	8904	124	8601	179	7	0	1	1.39	96.6	1.44	8800	104	1.18	Western Pacific
1	Malaysia	8904	124	8601	179	7	0	1	1.39	96.6	1.44	8800	104	1.18	Western Pacific
2	Malaysia	8904	124	8601	179	7	0	1	1.39	96.6	1.44	8800	104	1.18	Western Pacific
3	Malaysia	8904	124	8601	179	7	0	1	1.39	96.6	1.44	8800	104	1.18	Western Pacific
4	Malaysia	8904	124	8601	179	7	0	1	1.39	96.6	1.44	8800	104	1.18	Western Pacific
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1133	Sierra Leone	1783	66	1317	400	0	0	4	3.7	73.86	5.01	1711	72	4.21	Africa
1134	Sierra Leone	1783	66	1317	400	0	0	4	3.7	73.86	5.01	1711	72	4.21	Africa
1135	Sierra Leone	1783	66	1317	400	0	0	4	3.7	73.86	5.01	1711	72	4.21	Africa
1136	Sierra Leone	1783	66	1317	400	0	0	4	3.7	73.86	5.01	1711	72	4.21	Africa
1137	Sierra Leone	1783	66	1317	400	0	0	4	3.7	73.86	5.01	1711	72	4.21	Africa

1138 rows × 15 columns

Data Reconciliation

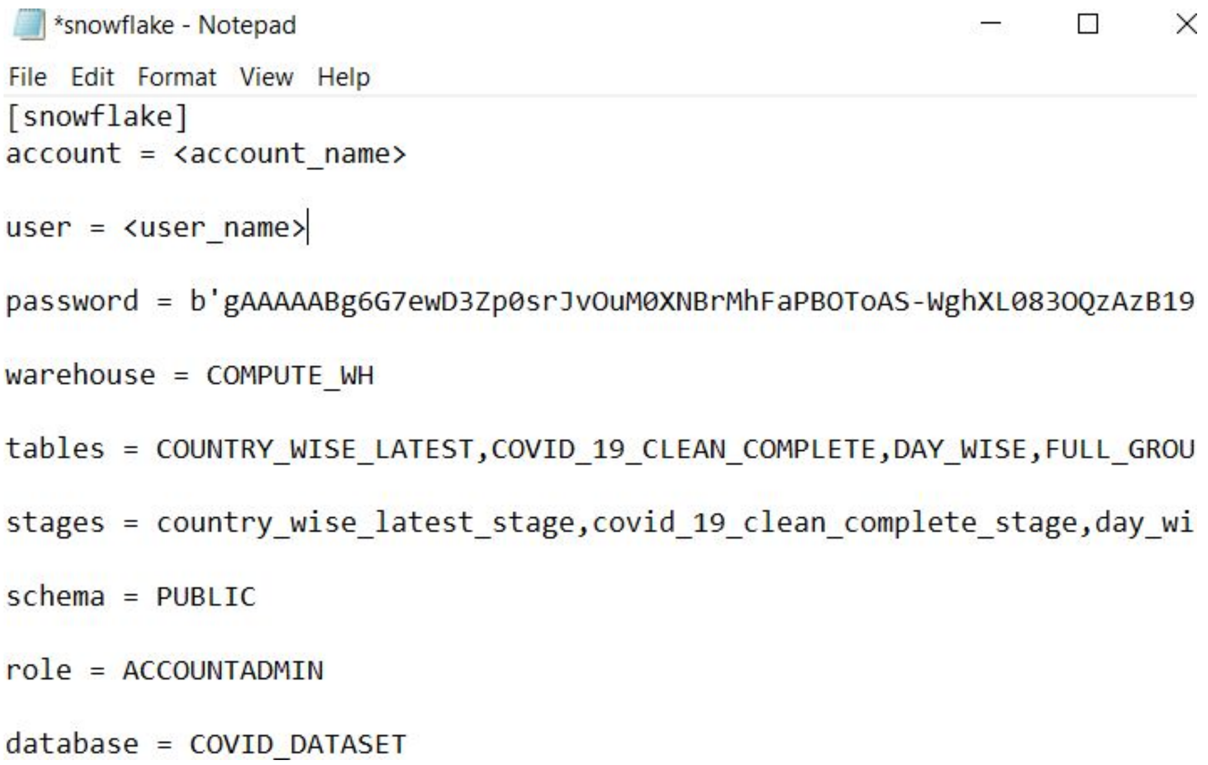
Sql code used by python to access table data from snowflake



```sql
snowflake_tables;

select $1::VARCHAR,$2::VARCHAR,$3::VARCHAR,$4::VARCHAR,$5::VARCHAR,$6::VARCHAR,$7::VARCHAR,$8::VARCHAR,$9::VARCHAR,$10::VARCHAR,$11::VARCHAR,
$12::VARCHAR,$13::VARCHAR,$14::VARCHAR,$15::VARCHAR from country_wise_latest
where (($16 is null)or(TO_TIMESTAMP_NTZ($16)>TO_TIMESTAMP_NTZ('2021-07-10 02:32:08.032473')));

select $1::VARCHAR,$2::VARCHAR,$3::VARCHAR,$4::VARCHAR,$5::VARCHAR,$6::VARCHAR,$7::VARCHAR,$8::VARCHAR,$9::VARCHAR,$10::VARCHAR
from covid_19_clean_complete
where (($11 is null)or(TO_TIMESTAMP_NTZ($11)<TO_TIMESTAMP_NTZ('2021-07-12 02:32:08.032473')));

select $1::VARCHAR,$2::VARCHAR,$3::VARCHAR,$4::VARCHAR,$5::VARCHAR,$6::VARCHAR,$7::VARCHAR,$8::VARCHAR,$9::VARCHAR,$10::VARCHAR,$11::VARCHAR,$12::VARCHAR
from day_wise
where (($13 is null)or(TO_TIMESTAMP_NTZ($13)<TO_TIMESTAMP_NTZ('2021-07-12 02:32:08.032473')));

select $1::VARCHAR,$2::VARCHAR,$3::VARCHAR,$4::VARCHAR,$5::VARCHAR,$6::VARCHAR,$7::VARCHAR,$8::VARCHAR,$9::VARCHAR,$10::VARCHAR
from full_grouped
where (($11 is null)or(TO_TIMESTAMP_NTZ($11)<TO_TIMESTAMP_NTZ('2021-07-12 02:32:08.032473')));

select $1::VARCHAR,$2::VARCHAR,$3::VARCHAR,$4::VARCHAR,$5::VARCHAR,$6::VARCHAR,$7::VARCHAR,$8::VARCHAR,$9::VARCHAR,
$10::VARCHAR,$11::VARCHAR,$12::VARCHAR,$13::VARCHAR,$14::VARCHAR,$15::VARCHAR,$16::VARCHAR
from worldometer_data
where (($17 is null)or(TO_TIMESTAMP_NTZ($17)<TO_TIMESTAMP_NTZ('2021-07-12 02:32:08.032473')));
```

Data Reconciliation

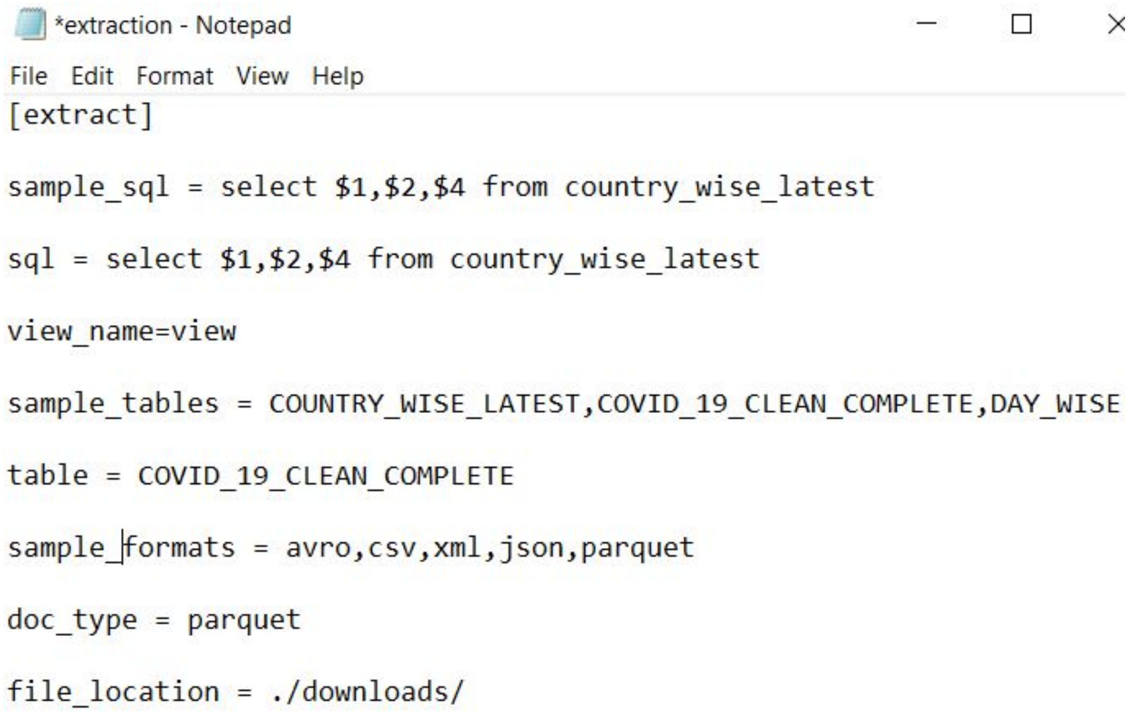
Setting snowflake config file to connect python code to snowflake



```
[snowflake]
account = <account_name>

user = <user_name>

password = b'gAAAAABg6G7ewD3Zp0srJvOuM0XNBrMhFaPBOToAS-WghXL083OQzAzB19

warehouse = COMPUTE_WH

tables = COUNTRY_WISE_LATEST,COVID_19_CLEAN_COMPLETE,DAY_WISE,FULL_GROU

stages = country_wise_latest_stage,covid_19_clean_complete_stage,day_wi

schema = PUBLIC

role = ACCOUNTADMIN

database = COVID_DATASET
```

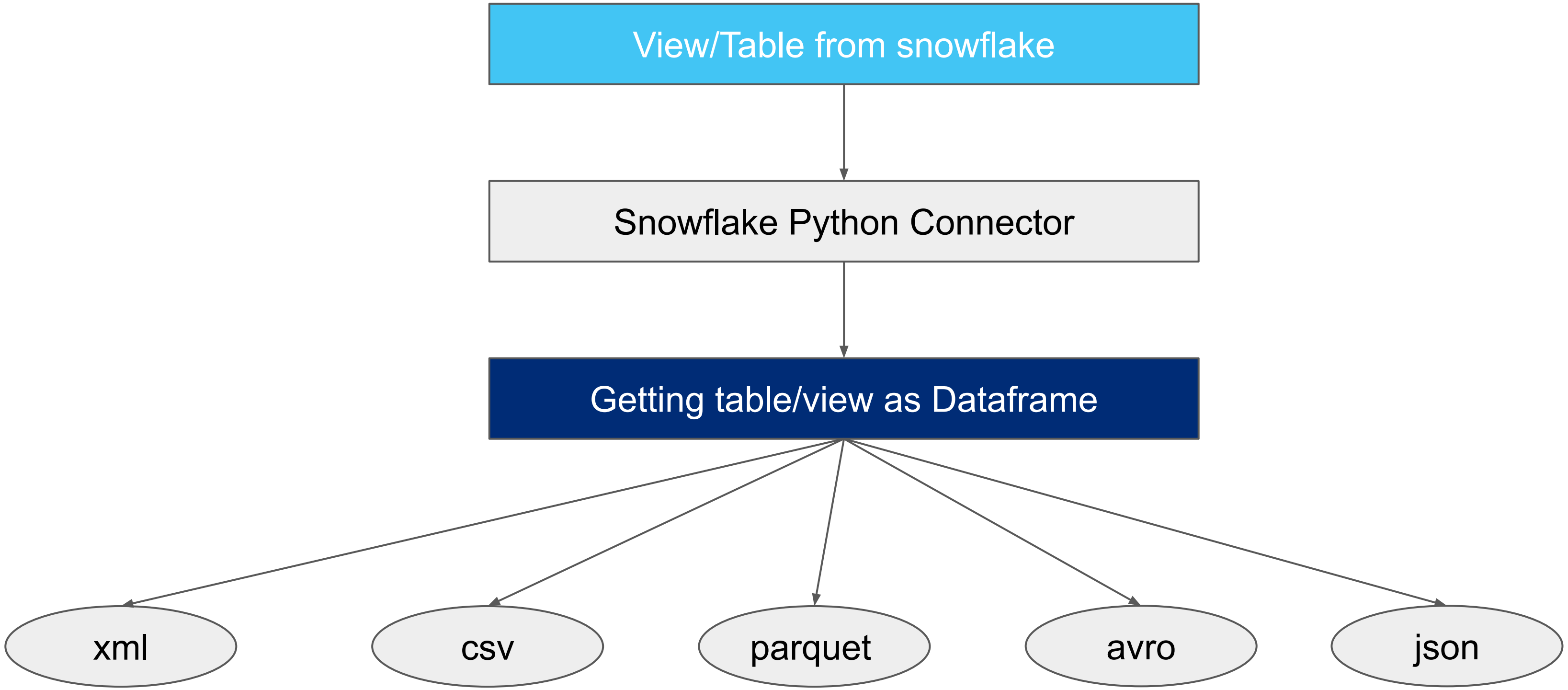
Data Extraction

Setting the config file to download a view or table in the required file format



```
[extract]

sample_sql = select $1,$2,$4 from country_wise_latest

sql = select $1,$2,$4 from country_wise_latest

view_name=view

sample_tables = COUNTRY_WISE_LATEST,COVID_19_CLEAN_COMPLETE,DAY_WISE

table = COVID_19_CLEAN_COMPLETE

sample_formats = avro,csv,xml,json,parquet

doc_type = parquet

file_location = ./downloads/
```

Data Extraction


View/Table from snowflake
Snowflake Python Connector
Getting table/view as Dataframe
xml
csv
parquet
avro
json

Establish Ingestion and Extraction patterns using Azure and Snowflake. (For Batch-time and Real-time)

Guided by- Amit Malik

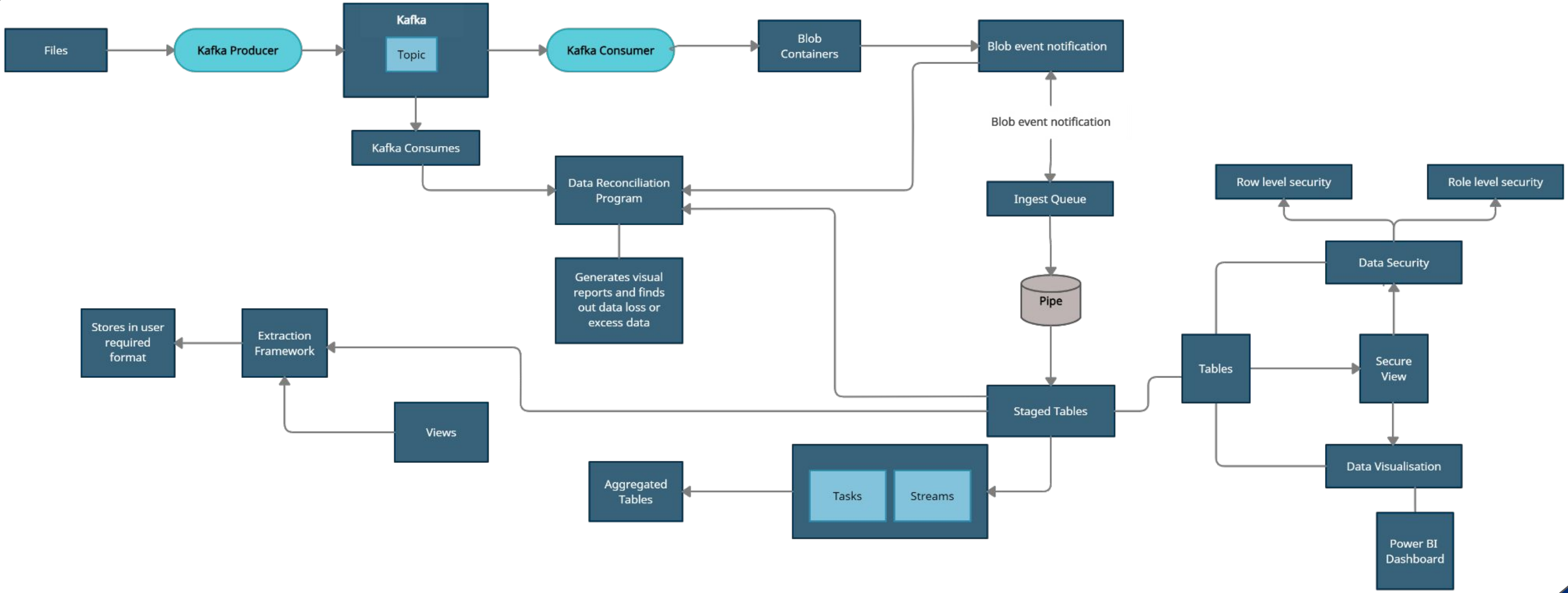
Team Members-

Ashwin Nair

Rajat Mishra

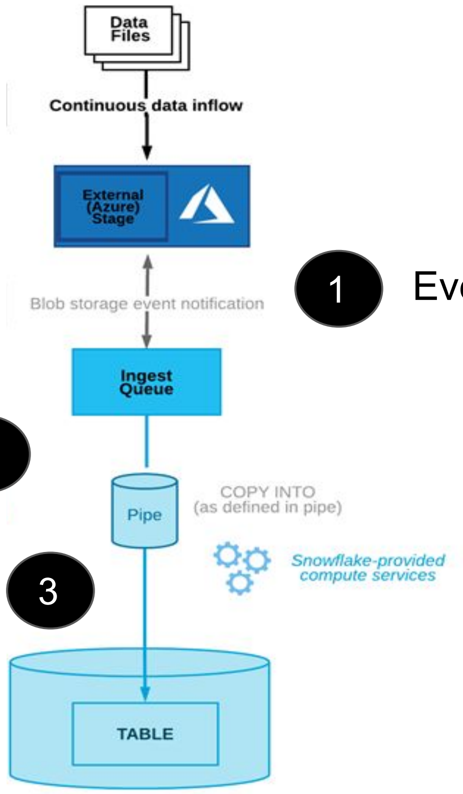
Nehal Tiwari

Work-flow:


Files
Kafka Producer
Kafka
Topic
Kafka Consumer
Blob Containers
Blob event notification
Kafka Consumes
Data Reconciliation Program
Blob event notification
Ingest Queue
Row level security
Role level security
Generates visual reports and finds out data loss or excess data
Pipe
Data Security
Stores in user required format
Extraction Framework
Tables
Secure View
Staged Tables
Views
Data Visualisation
Aggregated Tables
Tasks
Streams
Power BI Dashboard

Data Ingestion

Snowpipe's auto-ingest feature


Data Files
Continuous data inflow
External (Azure) Stage
Blob storage event notification
Ingest Queue
Pipe
COPY INTO (as defined in pipe)
Snowflake-provided compute services
TABLE

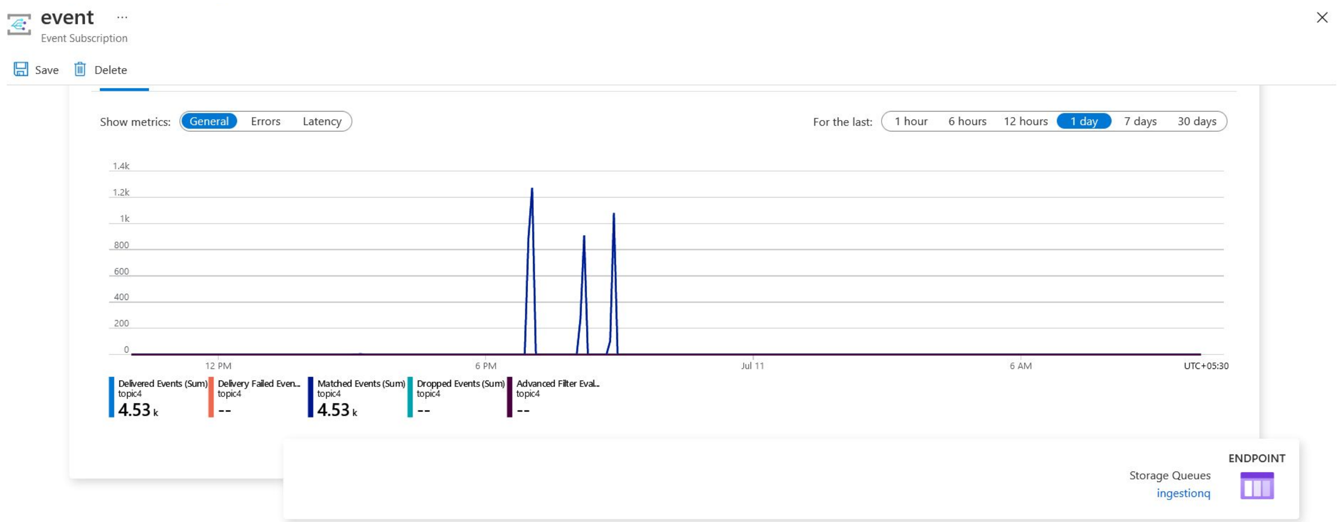
1 Event Subscription

Notification Integration 2

Snowpipe 3

Data Ingestion

- **Connecting Azure Blob Containers with Ingestion Queue via Event Subscription.**



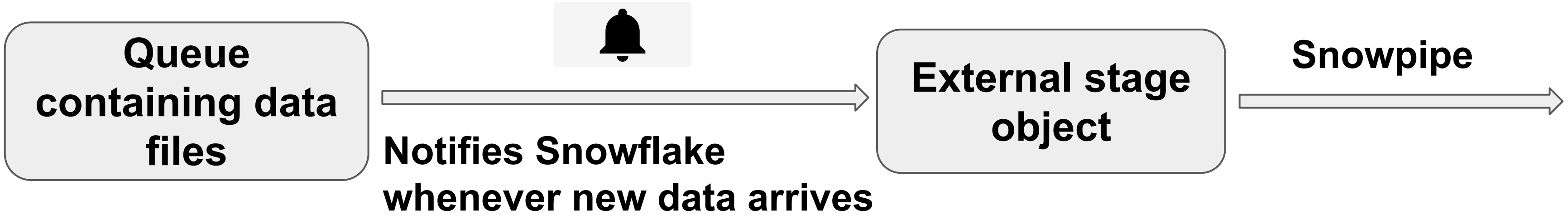
event
Event Subscription
Save
Delete
Show metrics: General Errors Latency
For the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days
1.4k
1.2k
1k
800
600
400
200
0
12 PM
6 PM
Jul 11
6 AM
UTC+05:30
ENDPOINT

Delivered Events (Sum) topic4	Delivery Failed Even... topic4	Matched Events (Sum) topic4	Dropped Events (Sum) topic4	Advanced Filter Eval... topic4
4.53 k	--	4.53 k	--	--

Storage Queues
ingestionq

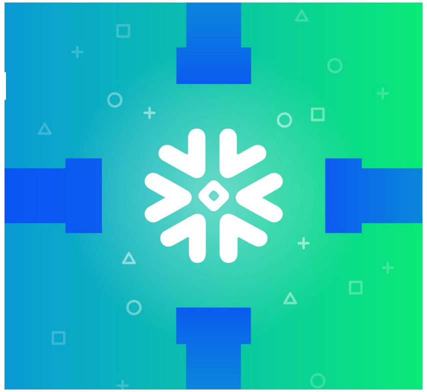
Data Ingestion

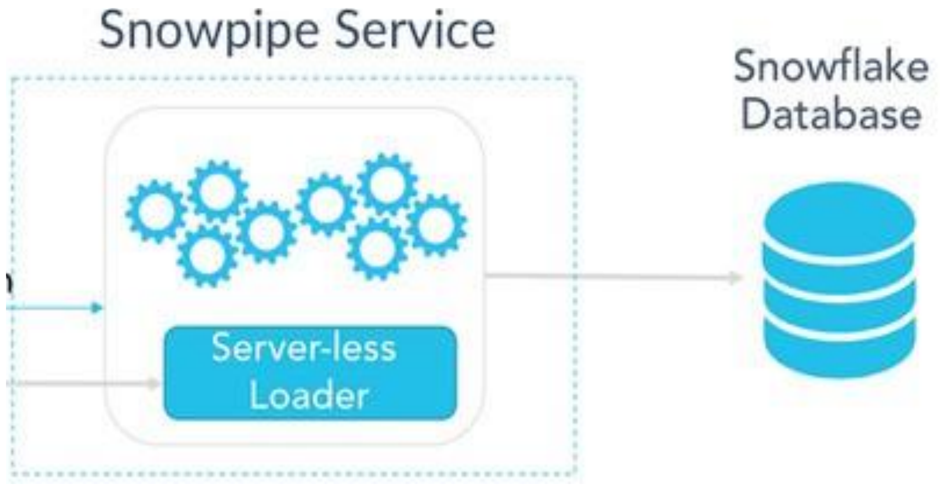
- **Connecting the Ingestion Queue with Snowflake Stage(external) via Notification Integration.**


Queue containing data files
Notifies Snowflake whenever new data arrives
External stage object
Snowpipe

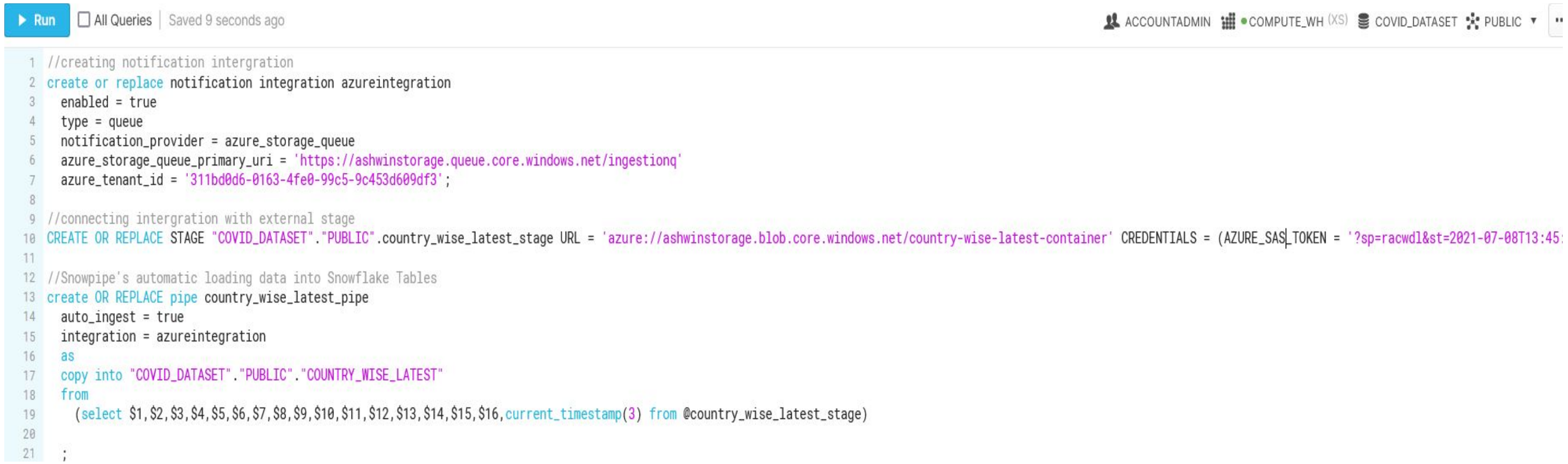
Data Ingestion

- **Snowpipe Automatically loading data into Snowflake Table.**





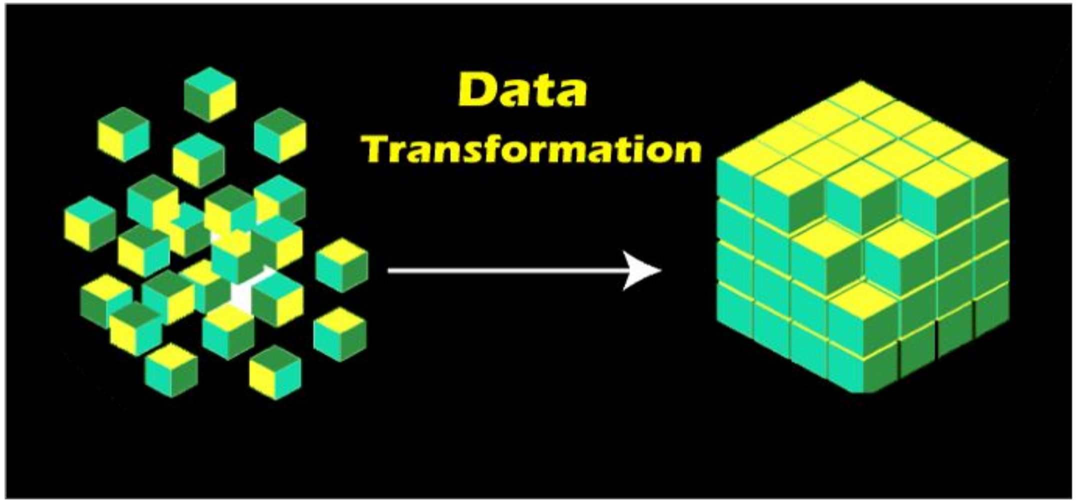
Code Snippets



```sql
//creating notification intergration
create or replace notification integration azureintegration
  enabled = true
  type = queue
  notification_provider = azure_storage_queue
  azure_storage_queue_primary_uri = 'https://ashwinstorage.queue.core.windows.net/ingestionq'
  azure_tenant_id = '311bd0d6-0163-4fe0-99c5-9c453d609df3';

//connecting intergration with external stage
CREATE OR REPLACE STAGE "COVID_DATASET"."PUBLIC".country_wise_latest_stage URL = 'azure://ashwinstorage.blob.core.windows.net/country-wise-latest-container' CREDENTIALS = (AZURE_SAS_TOKEN = '?sp=racwdl&st=2021-07-08T13:45

//Snowpipe's automatic loading data into Snowflake Tables
create OR REPLACE pipe country_wise_latest_pipe
  auto_ingest = true
  integration = azureintegration
  as
  copy into "COVID_DATASET"."PUBLIC"."COUNTRY_WISE_LATEST"
  from
    (select $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,current_timestamp(3) from @country_wise_latest_stage)

  ;
```

Data Transformation

- **Why Data Transformation?**



Data is **transformed** to make it better-organized.**Transformed data** would be easier for both humans and computers to use.It acts as a power booster for the **Data analytics process**.

Data Transformation

- **What are Streams and Tasks?**

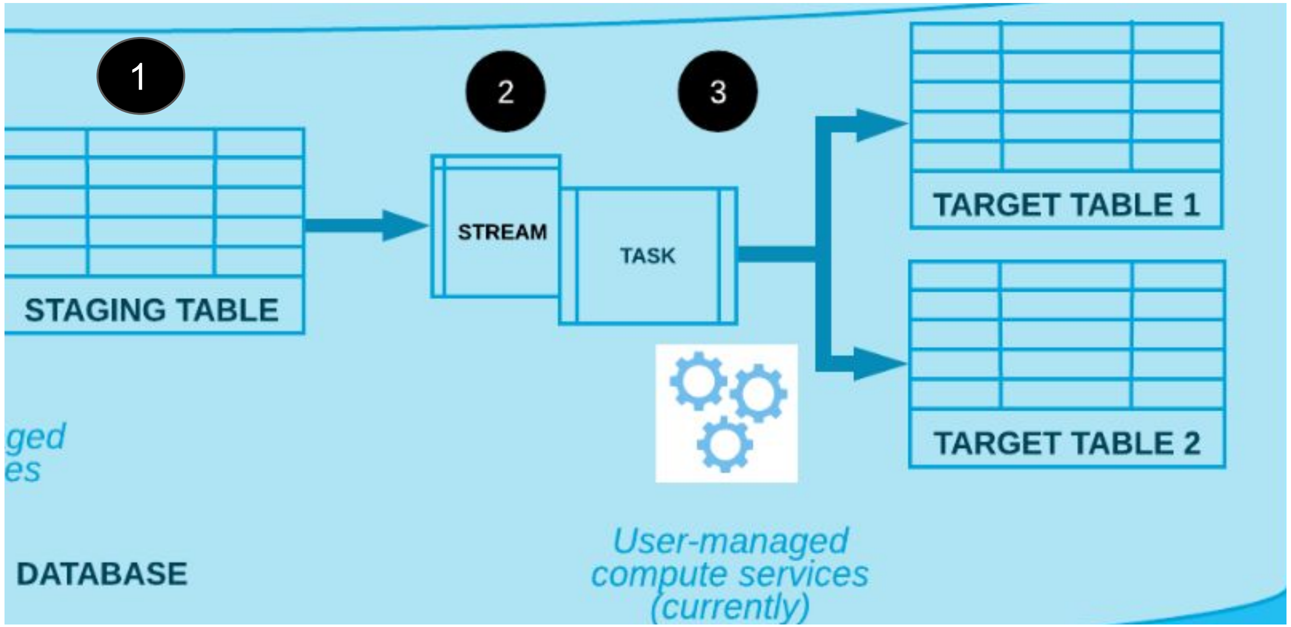
snowflake
STREAM
TASK
CDC

A **stream** is a **Snowflake** object type that provides change data capture (CDC).
A **Task** consumes these streams to run a scheduled query.

Data Transformation

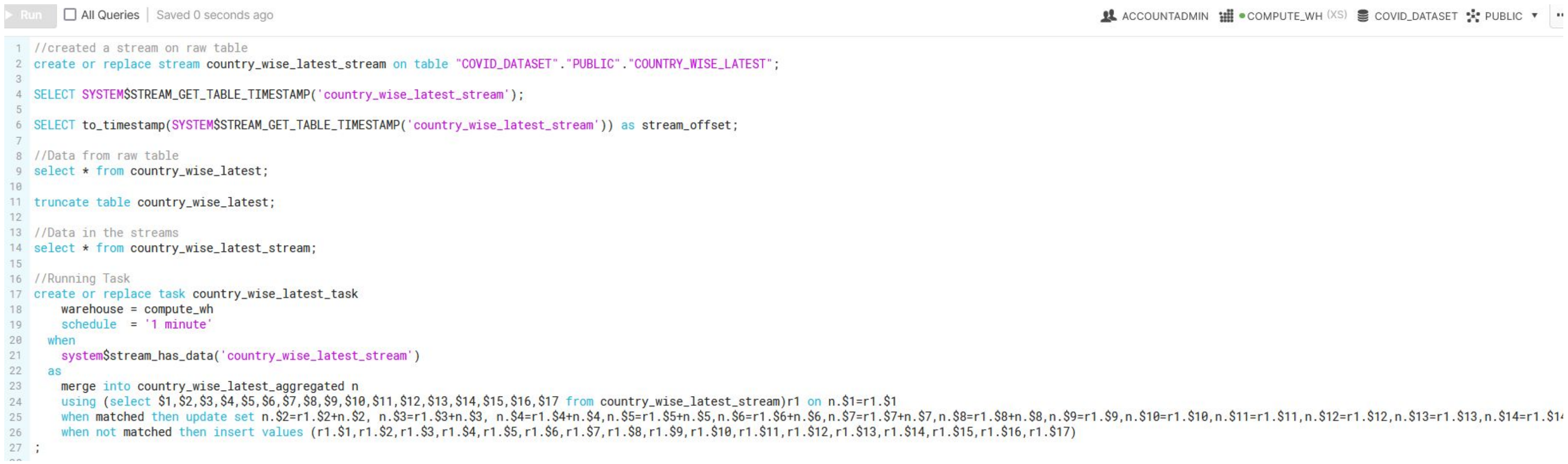
- **How Do Streams and Tasks work?**

Raw Data ⇨



Aggregated Data

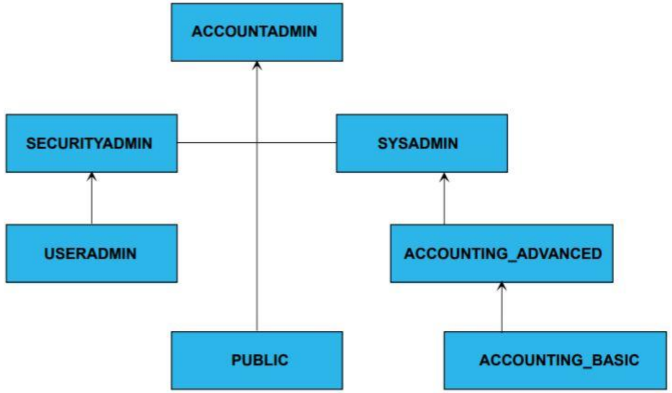
Code Snippets

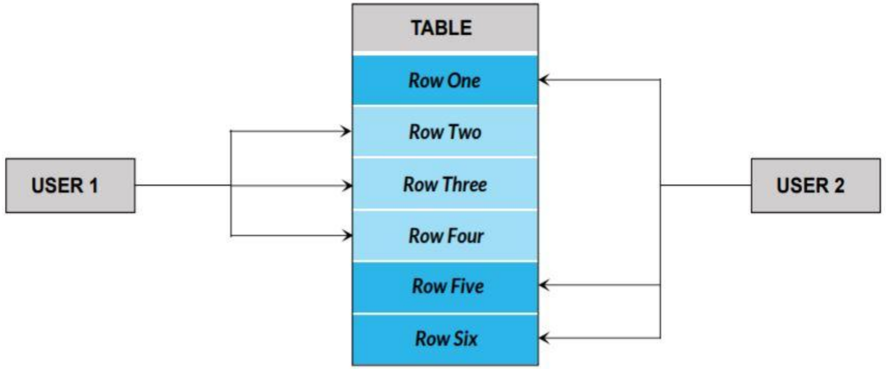


```sql
//created a stream on raw table
create or replace stream country_wise_latest_stream on table "COVID_DATASET"."PUBLIC"."COUNTRY_WISE_LATEST";

SELECT SYSTEM$STREAM_GET_TABLE_TIMESTAMP('country_wise_latest_stream');

SELECT to_timestamp(SYSTEM$STREAM_GET_TABLE_TIMESTAMP('country_wise_latest_stream')) as stream_offset;

//Data from raw table
select * from country_wise_latest;

truncate table country_wise_latest;

//Data in the streams
select * from country_wise_latest_stream;

//Running Task
create or replace task country_wise_latest_task
    warehouse = compute_wh
    schedule  = '1 minute'
  when
    system$stream_has_data('country_wise_latest_stream')
  as
    merge into country_wise_latest_aggregated n
    using (select $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17 from country_wise_latest_stream)r1 on n.$1=r1.$1
    when matched then update set n.$2=r1.$2+n.$2, n.$3=r1.$3+n.$3, n.$4=r1.$4+n.$4,n.$5=r1.$5+n.$5,n.$6=r1.$6+n.$6,n.$7=r1.$7+n.$7,n.$8=r1.$8+n.$8,n.$9=r1.$9,n.$10=r1.$10,n.$11=r1.$11,n.$12=r1.$12,n.$13=r1.$13,n.$14=r1.$1
    when not matched then insert values (r1.$1,r1.$2,r1.$3,r1.$4,r1.$5,r1.$6,r1.$7,r1.$8,r1.$9,r1.$10,r1.$11,r1.$12,r1.$13,r1.$14,r1.$15,r1.$16,r1.$17)
;
```

Data Security

- Most important aspect of data handling.
- Role level and row level access.
- 4 roles created coherent with the data set used.





We have created four roles aligning to the dataset we have been using:

WHO Role

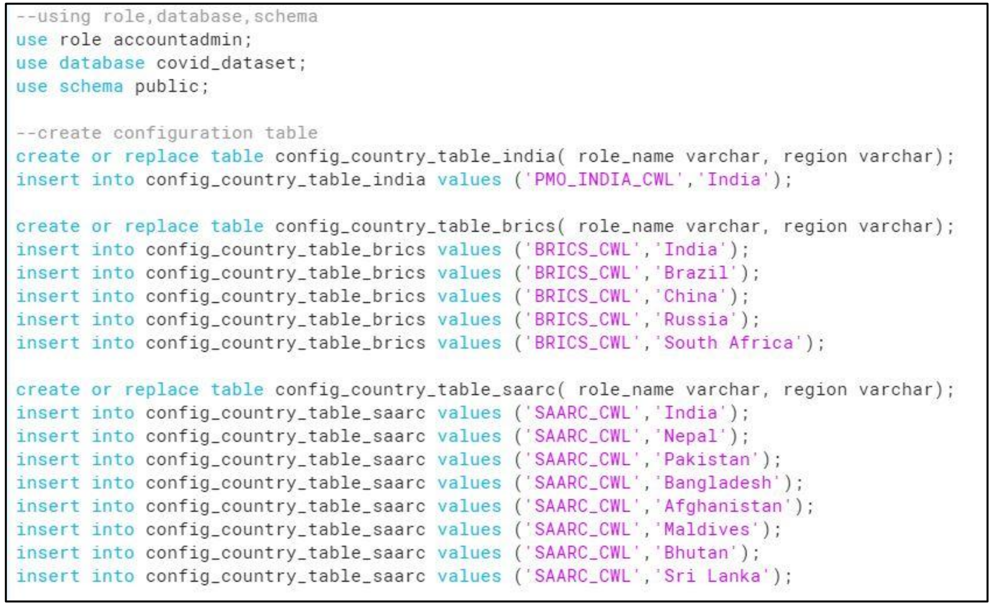
PMO India Role

BRICS Role

SAARC Role

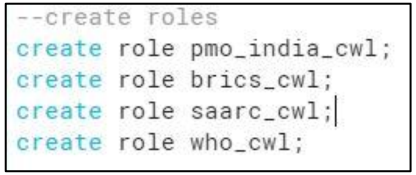
Code overview

1. Creating Configuration tables:



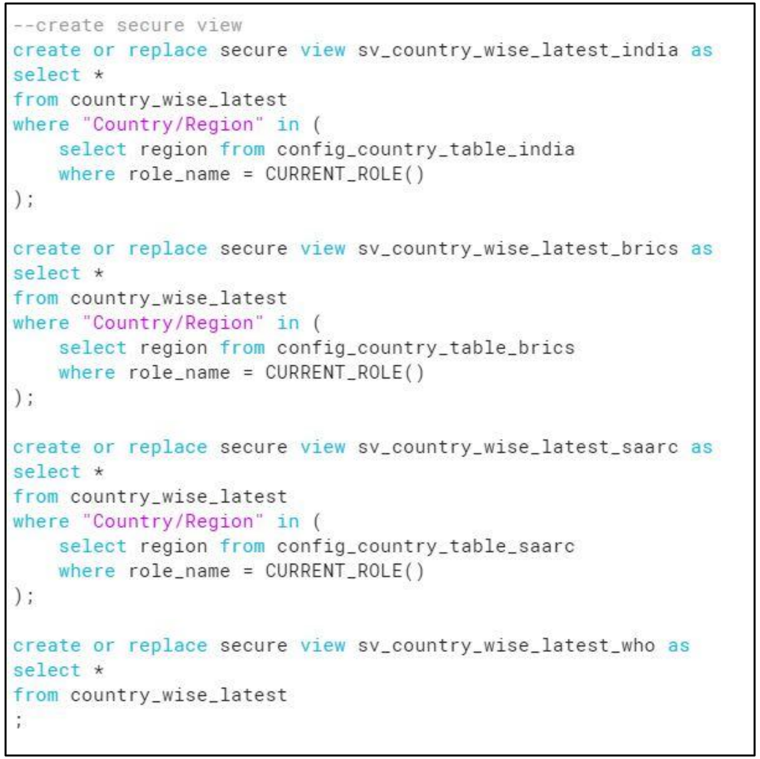
```sql
--using role,database,schema
use role accountadmin;
use database covid_dataset;
use schema public;

--create configuration table
create or replace table config_country_table_india( role_name varchar, region varchar);
insert into config_country_table_india values ('PMO_INDIA_CWL','India');

create or replace table config_country_table_brics( role_name varchar, region varchar);
insert into config_country_table_brics values ('BRICS_CWL','India');
insert into config_country_table_brics values ('BRICS_CWL','Brazil');
insert into config_country_table_brics values ('BRICS_CWL','China');
insert into config_country_table_brics values ('BRICS_CWL','Russia');
insert into config_country_table_brics values ('BRICS_CWL','South Africa');

create or replace table config_country_table_saarc( role_name varchar, region varchar);
insert into config_country_table_saarc values ('SAARC_CWL','India');
insert into config_country_table_saarc values ('SAARC_CWL','Nepal');
insert into config_country_table_saarc values ('SAARC_CWL','Pakistan');
insert into config_country_table_saarc values ('SAARC_CWL','Bangladesh');
insert into config_country_table_saarc values ('SAARC_CWL','Afghanistan');
insert into config_country_table_saarc values ('SAARC_CWL','Maldives');
insert into config_country_table_saarc values ('SAARC_CWL','Bhutan');
insert into config_country_table_saarc values ('SAARC_CWL','Sri Lanka');
```

2. Creating roles:



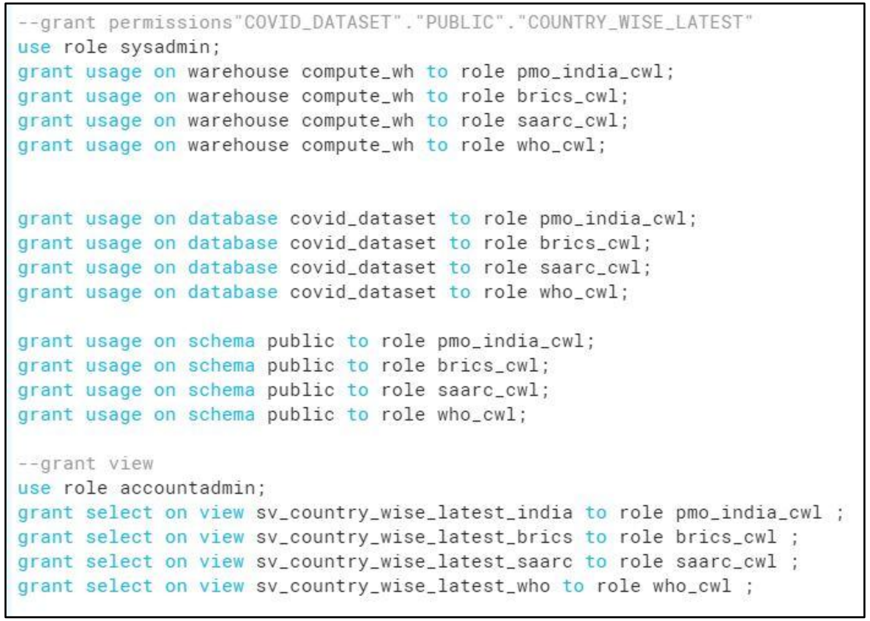
```sql
--create roles
create role pmo_india_cwl;
create role brics_cwl;
create role saarc_cwl;
create role who_cwl;
```

3. Creating secure views:



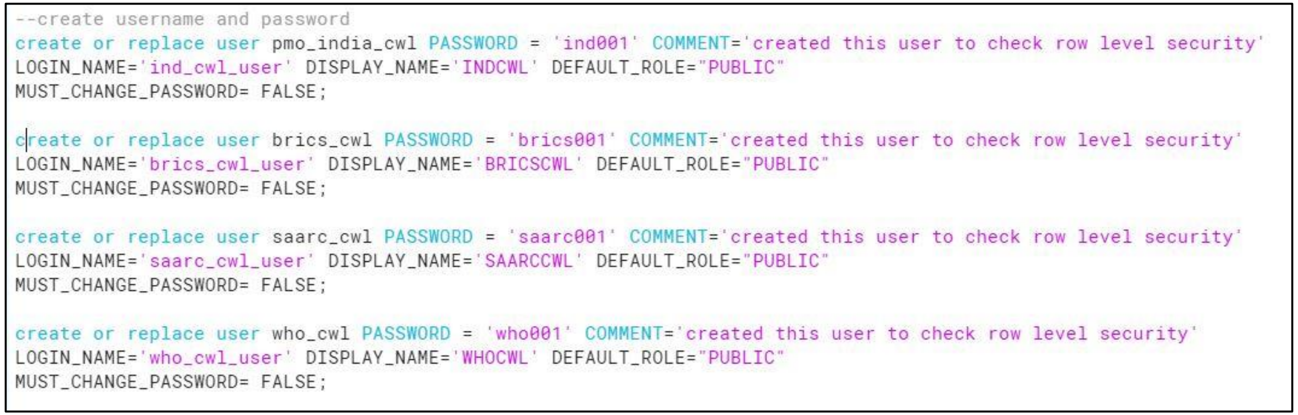
```sql
--create secure view
create or replace secure view sv_country_wise_latest_india as
select *
from country_wise_latest
where "Country/Region" in (
    select region from config_country_table_india
    where role_name = CURRENT_ROLE()
);

create or replace secure view sv_country_wise_latest_brics as
select *
from country_wise_latest
where "Country/Region" in (
    select region from config_country_table_brics
    where role_name = CURRENT_ROLE()
);

create or replace secure view sv_country_wise_latest_saarc as
select *
from country_wise_latest
where "Country/Region" in (
    select region from config_country_table_saarc
    where role_name = CURRENT_ROLE()
);

create or replace secure view sv_country_wise_latest_who as
select *
from country_wise_latest
;
```

4. Granting permissions for usage on warehouse, database, schema and secure view:



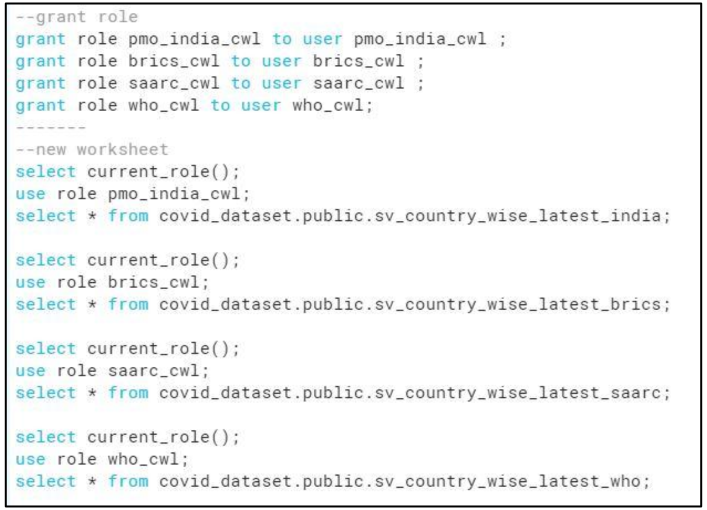
```sql
--grant permissions"COVID_DATASET"."PUBLIC"."COUNTRY_WISE_LATEST"
use role sysadmin;
grant usage on warehouse compute_wh to role pmo_india_cwl;
grant usage on warehouse compute_wh to role brics_cwl;
grant usage on warehouse compute_wh to role saarc_cwl;
grant usage on warehouse compute_wh to role who_cwl;


grant usage on database covid_dataset to role pmo_india_cwl;
grant usage on database covid_dataset to role brics_cwl;
grant usage on database covid_dataset to role saarc_cwl;
grant usage on database covid_dataset to role who_cwl;

grant usage on schema public to role pmo_india_cwl;
grant usage on schema public to role brics_cwl;
grant usage on schema public to role saarc_cwl;
grant usage on schema public to role who_cwl;

--grant view
use role accountadmin;
grant select on view sv_country_wise_latest_india to role pmo_india_cwl ;
grant select on view sv_country_wise_latest_brics to role brics_cwl ;
grant select on view sv_country_wise_latest_saarc to role saarc_cwl ;
grant select on view sv_country_wise_latest_who to role who_cwl ;
```

5. Creating usernames and passwords for each roles:



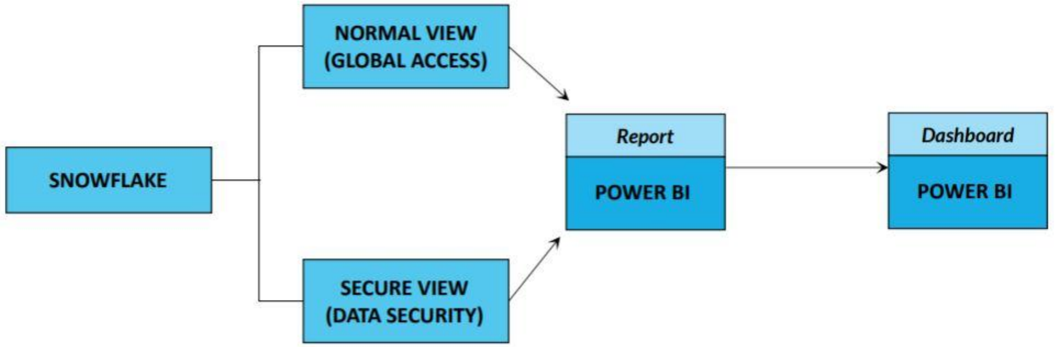
```
--create username and password
create or replace user pmo_india_cwl PASSWORD = 'ind001' COMMENT='created this user to check row level security'
LOGIN_NAME='ind_cwl_user' DISPLAY_NAME='INDCWL' DEFAULT_ROLE="PUBLIC"
MUST_CHANGE_PASSWORD= FALSE;

create or replace user brics_cwl PASSWORD = 'brics001' COMMENT='created this user to check row level security'
LOGIN_NAME='brics_cwl_user' DISPLAY_NAME='BRICSCWL' DEFAULT_ROLE="PUBLIC"
MUST_CHANGE_PASSWORD= FALSE;

create or replace user saarc_cwl PASSWORD = 'saarc001' COMMENT='created this user to check row level security'
LOGIN_NAME='saarc_cwl_user' DISPLAY_NAME='SAARCCWL' DEFAULT_ROLE="PUBLIC"
MUST_CHANGE_PASSWORD= FALSE;

create or replace user who_cwl PASSWORD = 'who001' COMMENT='created this user to check row level security'
LOGIN_NAME='who_cwl_user' DISPLAY_NAME='WHOCWL' DEFAULT_ROLE="PUBLIC"
MUST_CHANGE_PASSWORD= FALSE;
```

6. Granting roles to users for login in secure views:



```
--grant role
grant role pmo_india_cwl to user pmo_india_cwl ;
grant role brics_cwl to user brics_cwl ;
grant role saarc_cwl to user saarc_cwl ;
grant role who_cwl to user who_cwl;
-------
--new worksheet
select current_role();
use role pmo_india_cwl;
select * from covid_dataset.public.sv_country_wise_latest_india;

select current_role();
use role brics_cwl;
select * from covid_dataset.public.sv_country_wise_latest_brics;

select current_role();
use role saarc_cwl;
select * from covid_dataset.public.sv_country_wise_latest_saarc;

select current_role();
use role who_cwl;
select * from covid_dataset.public.sv_country_wise_latest_who;
```

Data Visualisation



- Dashboard for visual representation of data using Power BI.
- Facilitated comparison between countries.
- Visualisation helps in understanding the dataset better.

Takeaways

Rajat Mishra :

- Never Hesitate to ask your doubts with your seniors/mentors/peers. Clarify it before its late.
- Learning your team members and synchronizing with them is a key aspect.

Ashwin Nair :

- Identifying core strengths of team members helps in overall project execution
- Helping your team mates boosts team coordination and team efficiency

Nehal Tiwari :

- No matter how unexplored field is, there is always a way to ace it by learning.
- Coordinating between members of team for a better finished product.

Q
A

THANK YOU!!