ASSIGNMENT - 4

create database demo\_db;

use warehouse demo\_wh;

use database demo\_db;

--Snowflake chess data loading form S3 bucket

--Implementing Data Warehouse on Snowflake

-- creating integration

create or replace storage integration chess\_integration1

type = external\_stage

storage\_provider = s3

enabled = true

storage\_aws\_role\_arn = 'arn:aws:iam::211125644323:role/chess-asgn4-role'

storage\_allowed\_locations = ('s3://chess-bucket-asgn4/chess/');

desc integration chess\_integration1;

-- updatng trust policy with STORAGE\_AWS\_IAM\_USER\_ARN and STORAGE\_AWS\_EXTERNAL\_ID in AWS role

--creating file format

create or replace file format json\_file\_format

type='json',

strip\_outer\_array=TRUE;

--Create external stage object

create or replace stage chess\_stage1

URL = 's3://chess-bucket-asgn4/chess/'

STORAGE\_INTEGRATION = chess\_integration

file\_format = json\_file\_format;

--create the three different tables for the data

-- Create the list\_table

CREATE OR REPLACE TABLE list\_table (

username VARCHAR,

is\_live BOOLEAN

);

-- Create the info\_table

CREATE OR REPLACE TABLE info\_table (

username VARCHAR,

followers NUMERIC,

country VARCHAR,

joined DATE,

location VARCHAR,

name VARCHAR,

player\_id STRING,

status VARCHAR,

title VARCHAR,

primary\_key NUMERIC

);

-- Create the stats\_table

CREATE OR REPLACE TABLE stats\_table (

last\_blitz NUMERIC,

draw\_blitz NUMERIC,

loss\_blitz NUMERIC,

win\_blitz NUMERIC,

last\_bullet NUMERIC,

draw\_bullet NUMERIC,

loss\_bullet NUMERIC,

win\_bullet NUMERIC,

last\_rapid NUMERIC,

draw\_rapid NUMERIC,

loss\_rapid NUMERIC,

win\_rapid NUMERIC,

FIDE NUMERIC,

primary\_key NUMERIC

);

-- Create a Snowpipe to load data from S3 into the list\_table

CREATE OR REPLACE PIPE list\_pipe AUTO\_INGEST = TRUE AS

COPY INTO list\_table (username, is\_live)

FROM (SELECT

$1:username::STRING,

$1:is\_live::BOOLEAN

FROM @chess\_stage1/list\_file.json)

FILE\_FORMAT = json\_file\_format;

-- Create a Snowpipe to load data from S3 into the info\_table

CREATE OR REPLACE PIPE info\_pipe AUTO\_INGEST = TRUE AS

COPY INTO info\_table (username, followers, country, joined, location, name, player\_id, status, title,primary\_key)

FROM (SELECT

$1:username::VARCHAR,-- Assuming username can be used as primary\_key

$1:followers::NUMERIC,

$1:country::VARCHAR,

$1:joined::DATE,

$1:location::VARCHAR,

$1:name::VARCHAR,

$1:player\_id::STRING,

$1:status::VARCHAR,

$1:title::VARCHAR

$1:primary\_key::NUMERIC

FROM @chess\_stage1/Info\_file.json)

FILE\_FORMAT = json\_file\_format;

-- Create a Snowpipe to load data from S3 into the stats\_table

CREATE OR REPLACE PIPE stats\_pipe AUTO\_INGEST = TRUE AS

COPY INTO stats\_table (last\_blitz, draw\_blitz, loss\_blitz, win\_blitz, last\_bullet, draw\_bullet, loss\_bullet, win\_bullet, last\_rapid, draw\_rapid, loss\_rapid, win\_rapid, FIDE,primary\_key)

FROM (

SELECT

$1:last\_blitz::NUMERIC AS last\_blitz,

$1:draw\_blitz::NUMERIC AS draw\_blitz,

$1:loss\_blitz::NUMERIC AS loss\_blitz,

$1:win\_blitz::NUMERIC AS win\_blitz,

$1:last\_bullet::NUMERIC AS last\_bullet,

$1:draw\_bullet::NUMERIC AS draw\_bullet,

$1:loss\_bullet::NUMERIC AS loss\_bullet,

$1:win\_bullet::NUMERIC AS win\_bullet,

$1:last\_rapid::NUMERIC AS last\_rapid,

$1:draw\_rapid::NUMERIC AS draw\_rapid,

$1:loss\_rapid::NUMERIC AS loss\_rapid,

$1:win\_rapid::NUMERIC AS win\_rapid,

$1:FIDE::NUMERIC AS FIDE,

$1:primary\_key::NUMERIC AS primary\_key

FROM @chess\_stage1/stats\_file.json

)

FILE\_FORMAT = json\_file\_format;

show pipes;

alter pipe list\_pipe refresh;

alter pipe info\_pipe refresh;

alter pipe stats\_pipe refresh;

Select SYSTEM$PIPE\_STATUS('list\_pipe');

Select SYSTEM$PIPE\_STATUS('info\_pipe');

Select SYSTEM$PIPE\_STATUS('stats\_pipe');

select \* from list\_table limit 10;

select username from list\_table group by username having count (username)>2;

select \* from info\_table limit 10;

select \* from stats\_table limit 10;

truncate table list\_table;

truncate table info\_table;

truncate table stats\_table;

--Results

--SQL queries to retrieve information:

--1.Username of the best player by category (blitz, chess, bullet)

-- Best Blitz Player

SELECT username, MAX(last\_blitz) AS max\_blitz\_rating

FROM stats\_table

GROUP BY username

ORDER BY max\_blitz\_rating DESC

LIMIT 1;

-- Best Bullet Player

SELECT username, MAX(last\_bullet) AS max\_bullet\_rating

FROM stats\_table

GROUP BY username

ORDER BY max\_bullet\_rating DESC

LIMIT 1;

-- Best Rapid Player

SELECT username, MAX(last\_rapid) AS max\_rapid\_rating

FROM stats\_table

GROUP BY username

ORDER BY max\_rapid\_rating DESC

LIMIT 1;

--2.Full name (or username if null) of the best player and his FIDE elo

SELECT

COALESCE(p.name, p.username) AS best\_player\_name,

s.FIDE AS fide\_elo

FROM info\_table p

JOIN stats\_table s ON p.username = s.username

ORDER BY s.FIDE DESC

LIMIT 1;

--3.Average elo of premium, staff and basic players

SELECT

status,

AVG(s.FIDE) AS average\_fide\_elo

FROM info\_table p

JOIN stats\_table s ON p.username = s.username

WHERE p.status IN ('premium', 'staff', 'basic')

GROUP BY status;

--4.Number of professional players and their elo

SELECT

COUNT(p.username) AS professional\_player\_count,

AVG(s.FIDE) AS average\_fide\_elo

FROM info\_table p

JOIN stats\_table s ON p.username = s.username

WHERE p.title IS NOT NULL; -- Assuming professional players have a non-null title

--5.Average FIDE elo by their professional FIDE elo

SELECT

title,

AVG(s.FIDE) AS average\_fide\_elo

FROM info\_table p

JOIN stats\_table s ON p.username = s.username

WHERE p.title IS NOT NULL

GROUP BY title;

--6.Best player currently on live

SELECT

l.username,

s.FIDE AS fide\_elo

FROM list\_table l

JOIN stats\_table s ON l.username = s.username

WHERE l.is\_live = TRUE

ORDER BY s.FIDE DESC

LIMIT 1;