SNOWFLAKE

ASSIGNMENT 4

A diagram of a software project

Description automatically generated

Loaded all 3 JSON files into S3 bucket

A screenshot of a computer

Description automatically generated

User created with Access key ID and Access key: AKIA47CRXTAY3TSGQCHK – Active

|  |
| --- |
| Access key ID: |
| **AKIA47CRXTAY3TSGQCHK**   |  | | --- | | Secret access key: | | **53vJxO+V9pZ0y5erwx+UIpkF4SB5nvFBtlJQrXVW** | |

A screenshot of a computer

Description automatically generated

Changing the user role to ACCOUNTADMIN:

-- Using Role Accountadmin

use role accountadmin;

-- Create Databse Chess for futher tasks

create or replace database CHESS;

A screenshot of a computer

Description automatically generated

Creating the file format:

-- Make a file format

create or replace file format CHESS.public.json\_data

type = 'json'

strip\_outer\_array=true;

A screenshot of a computer program

Description automatically generated

--creating AWS S3 external stage using Access Key and AWS Key ID

CREATE OR REPLACE STAGE customer.PUBLIC.S3\_JSON

URL='s3://datapipeline3/json\_data/'

CREDENTIALS=(AWS\_KEY\_ID='AKIA47CRXTAY3TSGQCHK' AWS\_SECRET\_KEY='53vJxO+V9pZ0y5erwx+UIpkF4SB5nvFBtlJQrXVW')

file\_format=json\_data ;

A screenshot of a computer

Description automatically generated

-- list the content of stage

list @customer.PUBLIC.S3\_JSON;

A screenshot of a computer

Description automatically generated

-- Create source list table to load json data this will act as stage table list table

CREATE OR REPLACE TABLE list\_source(v variant);

-- Create stream on above table

CREATE or REPLACE STREAM list\_source\_stream ON TABLE list\_source;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

-- Create source info table to load json data this will act as stage table for info table

CREATE OR REPLACE TABLE info\_source (v variant);

-- Create stream on above table

CREATE or REPLACE STREAM info\_source\_stream ON TABLE info\_source;

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

-- Create source stats table to load json data this will act as stage table for stats table

CREATE OR REPLACE TABLE stats\_source(v variant);

-- Create stream on above table

CREATE or REPLACE STREAM stats\_source\_stream ON TABLE stats\_source;

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Pipe creation to copy data from list\_file.json to list\_source stage table**

-- pipe creation to copy data from list\_file.json to list\_source stage table

CREATE or REPLACE pipe chess\_list\_snowipe

auto\_ingest=true

as copy into list\_source

from @CUSTOMER.PUBLIC.S3\_JSON/list\_file.json

file\_format=(format\_name='json\_data')

on\_error = CONTINUE;



-- describe pipe to genetate SQS code to insert in AWS event notifiation for trigger pipe in snowflake

desc PIPE chess\_list\_snowipe ;

A screenshot of a computer

Description automatically generated

**Pipe creation to copy data from info\_file.json to info\_source stage table**

-- pipe creation to copy data from info\_file.json to info\_source stage table

create or replace pipe chess\_info\_snowipe

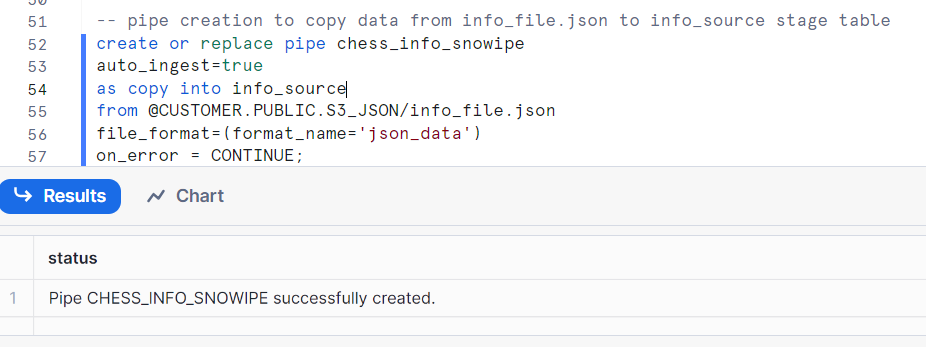
auto\_ingest=true

as copy into info\_source

from @CUSTOMER.PUBLIC.S3\_JSON/info\_file.json

file\_format=(format\_name='json\_data')

on\_error = CONTINUE;



-- describe pipe to genetate SQS code to insert in AWS event notifiation for trigger pipe in snowflake

desc PIPE chess\_list\_snowipe ;

A screenshot of a computer

Description automatically generated

**Pipe creation to copy data from stats\_file.json to stats\_source stage table**

-- pipe creation to copy data from stats\_file.json to stats\_source stage table

create or replace pipe chess\_stats\_snowipe

auto\_ingest=true

as copy into stats\_source

from @CUSTOMER.PUBLIC.S3\_JSON/stats\_file.json

file\_format=(format\_name='json\_data')

on\_error = CONTINUE;

A screenshot of a computer code

Description automatically generated

-- describe pipe to genetate SQS code to insert in AWS event notifiation for trigger pipe in snowflake

desc PIPE chess\_list\_snowipe ;

A screenshot of a computer

Description automatically generated

-- create temporary list table to load data from stream. This table will help to flatten data using lateral flatten.

CREATE OR REPLACE TABLE list\_temp(v variant);

-- create temporary info table to load data from stream. This table will help to flatten data using lateral flatten.

CREATE OR REPLACE TABLE info\_temp (v variant);

-- create temporary stats table to load data from stream. This table will help to flatten data using lateral flatten.

CREATE OR REPLACE TABLE stats\_temp(v variant);

A screenshot of a computer program

Description automatically generated

**Creating task to load data from stream to temporary table**

-- create task to load data from stream to temporary table

CREATE or replace TASK process\_list

WAREHOUSE = COMPUTE\_WH

SCHEDULE = '1 MINUTE'

AS

INSERT INTO list\_temp

SELECT parse\_json($1)

FROM list\_source\_stream;

A screenshot of a computer program

Description automatically generated

-- create task to load data from stream to temporary table

CREATE or replace TASK process\_info

WAREHOUSE = COMPUTE\_WH

SCHEDULE = '1 MINUTE'

AS

INSERT INTO info\_temp

SELECT parse\_json($1)

FROM info\_source\_stream;

A screen shot of a computer program

Description automatically generated

-- create task to load data from stream to temporary table

CREATE or replace TASK process\_stats

WAREHOUSE = COMPUTE\_WH

SCHEDULE = '1 MINUTE'

AS

INSERT INTO stats\_temp

SELECT parse\_json($1)

FROM stats\_source\_stream;

A screenshot of a computer program

Description automatically generated

ALTER TASK process\_list RESUME;

ALTER TASK process\_info RESUME;

ALTER TASK process\_stats RESUME;

-- create list table with required schema

CREATE OR REPLACE TABLE list\_table (username varchar, is\_live boolean);

A screen shot of a computer

Description automatically generated

-- create info table with required schema

CREATE OR REPLACE TABLE info\_table

(username varchar, followers numeric, country varchar, joined date, location varchar, name varchar, player\_id numeric, status varchar,

title varchar, primary\_key numeric );

A screenshot of a computer

Description automatically generated

-- create stats table with required schema

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);

A screenshot of a computer

Description automatically generated

-- Insert json data to list table

insert into list\_table

select PARSE\_JSON($1):username,

PARSE\_JSON($1):is\_live

from list\_temp;

-- insert json data to info table

insert into info\_table

select PARSE\_JSON($1):username::varchar ,

PARSE\_JSON($1):followers::numeric,

PARSE\_JSON($1):country::varchar,

PARSE\_JSON($1):joined::date,

PARSE\_JSON($1):location::varchar,

PARSE\_JSON($1):name::varchar,

PARSE\_JSON($1):player\_id::numeric,

PARSE\_JSON($1):status::varchar,

PARSE\_JSON($1):title::varchar,

PARSE\_JSON($1):primary\_key::numeric

from info\_temp;

-- insert into stats table

insert into stats\_table

select cbz.value:last.rating as last\_blitz::numeric,

cbz.value:record.draw as draw\_blitz::numeric,

cbz.value:record.loss as loss\_blitz::numeric,

cbz.value:record.win as win\_blitz::numeric,

cbt.value:last.rating as last\_bullet::numeric,

cbt.value:record.draw as draw\_bullet::numeric,

cbt.value:record.loss as loss\_bullet::numeric,

cbt.value:record.win as win\_bullet::numeric,

cbr.value:last.rating as last\_rapid::numeric,

cbr.value:record.draw as draw\_rapid::numeric,

cbr.value:record.loss as loss\_rapid::numeric,

cbr.value:record.win as win\_rapid::numeric,

v.value:fide as fide::numeric,

v.key as username::varchar

from stats\_temp

inner join lateral flatten(input => v) v

inner join lateral flatten(input => v.value:chess\_blitz) cbz

inner join lateral flatten(input => v.value:chess\_bullet) cbt

inner join lateral flatten(input => v.value:chess\_rapid) crd;

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

create table best\_blitz as (

select info\_table.username, stats\_table.last\_blitz as best\_blitz

from stats\_table join info\_table

on stats\_table.username= info\_table.username

where last\_blitz = (

select max(last\_blitz) from stats\_table));

create table best\_bullet as (

select info\_table.username, stats\_table.last\_bullet as best\_bullet

from stats\_table join info\_table

on stats\_table.username = info\_table.username

where last\_bullet = (

select max(last\_bullet) from stats\_table));

create table best\_rapid as (

select info\_table.username, stats\_table.last\_rapid as best\_rapid

from stats\_table join info\_table

on stats\_table.username = info\_table.username

where last\_rapid = (

select max(last\_rapid) from stats\_table));

select \*

from best\_blitz join best\_bullet join best\_rapid;

--------------------------------------------------------------------------------------------------------------------------------------------------

-- full name or username of best player by country

select

ifnull(info\_table.name, info\_table.username) as "Name",

best.best\_blitz,

stats\_table.FIDE

from info\_table

join stats\_table

on info\_table.username = stats\_table.username

join (

select max(stats\_table.last\_blitz) as best\_blitz

from stats\_table join info\_table

on stats\_table.username = info\_table.username) as best

on stats\_table.last\_blitz = best.best\_blitz

order by best.best\_blitz desc;

------------------------------------------------------------------------------------------------------------------------------------------------------

-- avg elo by status (premium, staff, basic)

select info\_table.status,

round(avg(stats\_table.last\_blitz),2) as average\_blitz,

round(avg(stats\_table.last\_bullet),2) as average\_bullet,

round(avg(stats\_table.last\_rapid),2) as average\_rapid,

count(info\_table.country) as number\_of\_streamers

from info\_table right join stats\_table

on info\_table.primary\_key = stats\_table.primary\_key

group by info\_table.status

having average\_blitz is not null

or average\_bullet is not null

or average\_rapid is not null

order by number\_of\_streamers desc, average\_blitz desc, average\_bullet desc, average\_rapid desc;

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- Number of professional players and their elo

select count(info\_table.title)

from info\_table;

select info\_table.username, info\_table.title, stats\_table.FIDE

from stats\_table join info\_table

on stats\_table.primary\_key = info\_table.primary\_key

where stats\_table.FIDE is not null

order by stats\_table.FIDE desc;

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- Average FIDE elo by their professional FIDE elo

select info\_table.title, avg(stats\_table.FIDE) as average\_FIDE

from stats\_table join info\_table

on stats\_table.username = info\_table.username

where stats\_table.FIDE is not null and (info\_table.title is not null)

group by info\_table.title

order by average\_FIDE desc;

----------------------------------------------------------------------------------------------------------------------------------------------------------

-- Best player currently on live

select info\_table.username, stats\_table.last\_blitz, stats\_table.last\_bullet, stats\_table.last\_rapid

from info\_table

join stats\_table

on info\_table.primary\_key = stats\_table.primary\_key

join list\_table

on lower(info\_table.username) = lower(list\_table.username)

where list\_table.is\_live = TRUE and (last\_blitz is not null or last\_bullet is not null or last\_rapid is not null)

order by last\_blitz desc, last\_bullet desc, last\_rapid desc;

--------------------------------------------------------------------------------------------------------------------------------------------------------