**Creating database**

Create or replace database chess;

**Creating Schema**

create or replace schema chess.Assignment\_4;

**Creating Info Table**

CREATE OR REPLACE TABLE chess.Assignment\_4.info\_table (

username varchar

,followers numeric

,country varchar

,joined date

,location varchar

,name varchar

,player\_id varchar

,status varchar

,title varchar

);

**Creating List Table**

CREATE or replace TABLE list\_table(

username VARCHAR

,is\_live boolean);

**Creating 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 );

**Creating Stage**

CREATE OR REPLACE STAGE chess.Assignment\_4.MY\_S3\_STAGE

URL='s3://sneha-s3-bucket/Snowflake\_Data/'

CREDENTIALS=(AWS\_KEY\_ID='AKIA6XXXXXXXXxxxxx' AWS\_SECRET\_KEY='4gYj4mT2FozwYwUDXXXXXXXXXXxxxxx/6')

file\_format = my\_file\_format;

**Listing Stage**

list @ chess.Assignment\_4.MY\_S3\_STAGE;

**Create pipe for Info Table**

CREATE OR REPLACE PIPE chess.Assignment\_4.info\_table\_pipe

AUTO\_INGEST = TRUE AS

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

FROM (

SELECT $1:username::varchar as username,

$1:followers::numeric as followers,

$1:country::varchar as country,

$1:joined::date as joined,

$1:location::varchar as location,

$1:name::varchar as name,

$1:player\_id::varchar as player\_id,

$1:status::varchar as status,

$1:title::varchar as title

FROM @ chess.Assignment\_4.MY\_S3\_STAGE/Info\_file.json

)

FILE\_FORMAT = (FORMAT\_NAME = ' chess.PUBLIC.my\_file\_format')

ON\_ERROR = 'CONTINUE';

**Refreshing pipe of info\_table**

alter pipe info\_table\_pipe refresh;

**Create pipe for List Table**

CREATE OR REPLACE PIPE chess.Assignment\_4.list\_table\_pipe

AUTO\_INGEST = TRUE AS

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

FROM (

SELECT $1:username::varchar as username,

$1:is\_live::boolean as is\_live,

FROM @ chess.Assignment\_4.MY\_S3\_STAGE/list\_file.json

)

FILE\_FORMAT = (FORMAT\_NAME = ' chess.PUBLIC.my\_file\_format')

ON\_ERROR = 'CONTINUE';

**refresh pipe for List Table**

alter pipe list\_table\_pipe refresh;

CREATE OR REPLACE PIPE chess.Assignment\_4.stats\_table\_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\_rapidd::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.Assignment\_4.MY\_S3\_STAGE/stats\_table.json

)

FILE\_FORMAT = (FORMAT\_NAME = ' chess.PUBLIC.my\_file\_format')

ON\_ERROR = 'CONTINUE';

alter pipe chess.Assignment\_4.stats\_table\_pipe refresh;

select \* from stats\_table;

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

-- Best Blitz Player

SELECT username

FROM players

JOIN games ON players.player\_id = games.player\_id

WHERE games.game\_type = 'blitz'

GROUP BY players.username

ORDER BY AVG(games.rating) DESC

LIMIT 1;

-- Best Bullet Player

SELECT username

FROM players

JOIN games ON players.player\_id = games.player\_id

WHERE games.game\_type = 'bullet'

GROUP BY players.username

ORDER BY AVG(games.rating) DESC

LIMIT 1;

-- Best Classical Chess Player

SELECT username

FROM players

JOIN games ON players.player\_id = games.player\_id

WHERE games.game\_type = 'classical'

GROUP BY players.username

ORDER BY AVG(games.rating) DESC

LIMIT 1;

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

SELECT COALESCE(full\_name, username) AS player\_name, fide\_elo

FROM players

ORDER BY fide\_elo DESC

LIMIT 1;

• Average elo of premium, staff and basic players

SELECT membership\_type, AVG(fide\_elo) AS average\_elo

FROM players

GROUP BY membership\_type;

• Number of professional players and their elo

select count(\*) as num\_professional\_players, avg(cs.fide) as average\_elo  
from chess\_info ci  
join chess\_stats cs  
on ci.primary\_key = cs.primary\_key  
where title = 'gm' or title = 'im' or title = 'fm';

• Average FIDE elo by their professional FIDE elo

-- • average fide elo by their professional fide elo  
select ci.title, cs.avg(fide) as average\_elo  
from chess\_info ci  
join chess\_stats cs;  
on ci.primary\_key = cs.primary\_key  
group by title;

-- • best player currently on live  
select ci.username, cs.fide  
from chess\_info ci  
join chess\_stats cs  
on ci.primary\_key = cs.primary\_key  
join chess\_list cl  
on ci.username = cl.username  
where cl.is\_live = true  
order by cs.fide desc  
limit 1;