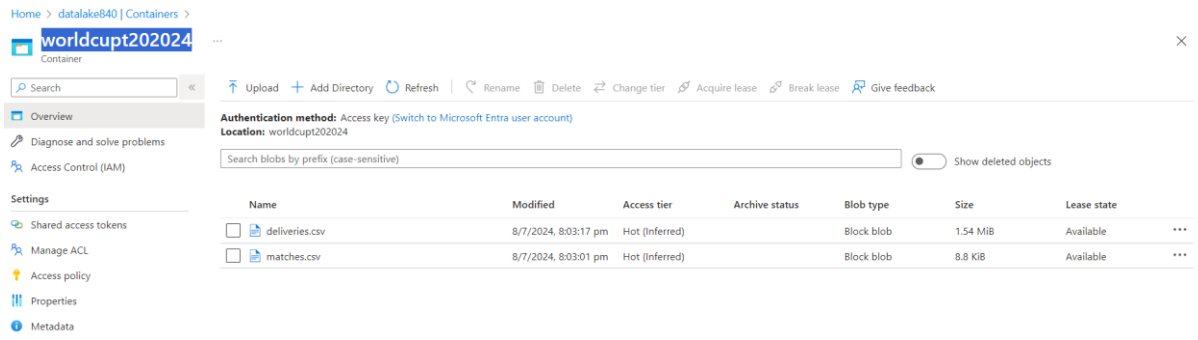
World-cup-T20 2024

Skills: Azure Data Lake, DataBrick, Pyspark, Pysql, DataBrick Dashboard, SQL

Objective:

This project aims to analyze the T20 World Cup 2024 using Databricks to create an interactive and insightful dashboard. This dashboard will leverage the power of PySpark and PySQL to process, analyze, and visualize data. The goal is to provide comprehensive insights into team performances, player statistics, match outcomes, and tournament trends, enabling users to explore and understand the dynamics of the T20 World Cup 2024.

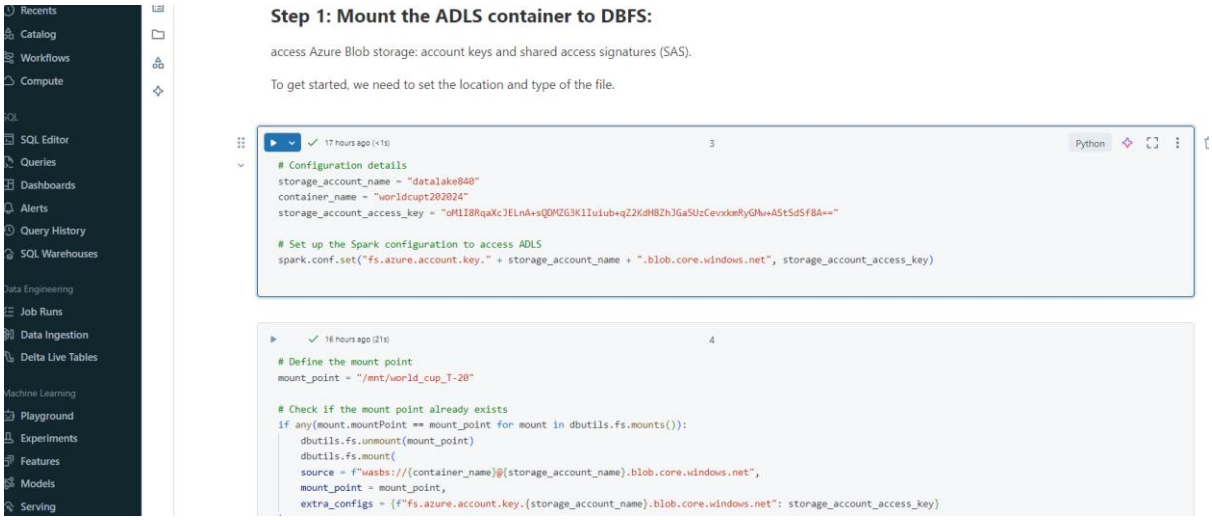
1. Data Collection and Ingestion:
    a. Create an Azure blob storage account, inside the 'worldcupt202024' container, upload the source CSV file.



    b. Mount the ADLS container to DBFS using access key:

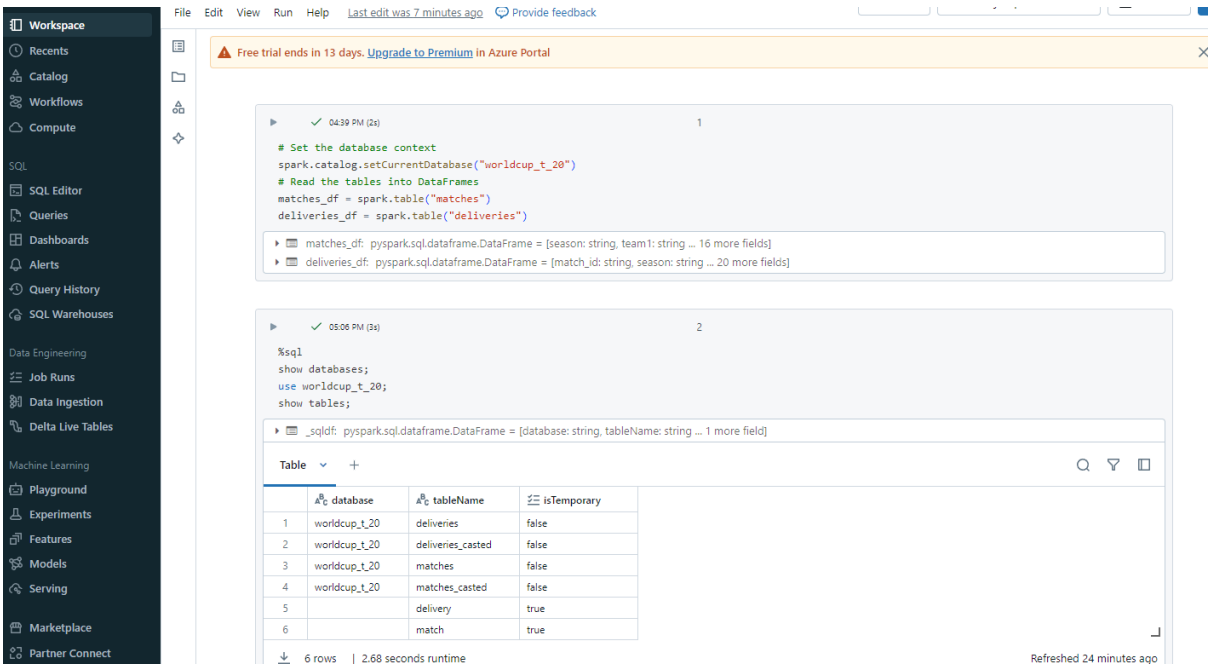
    c. Create the dataframe and upload it on database as table.

    d. Codes for this task mentioned in 'Import from Azure Blob Storage' Notebook



2. Data Transformation and Data cleaning

a. Created a new notebook to transform and clean data ('T20Worldcup_Data Transformation')

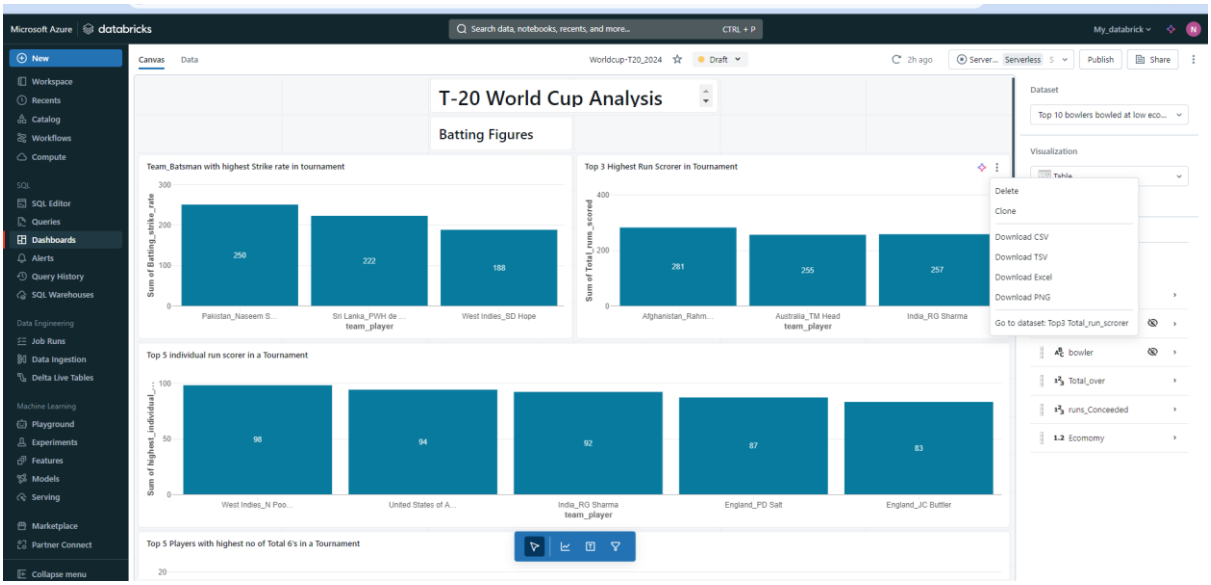
b. 

c. Store the clean data in database as a table.

3. Dashboard Creation

a. Using data tab on Databrick dashboard select database and tables required to create the

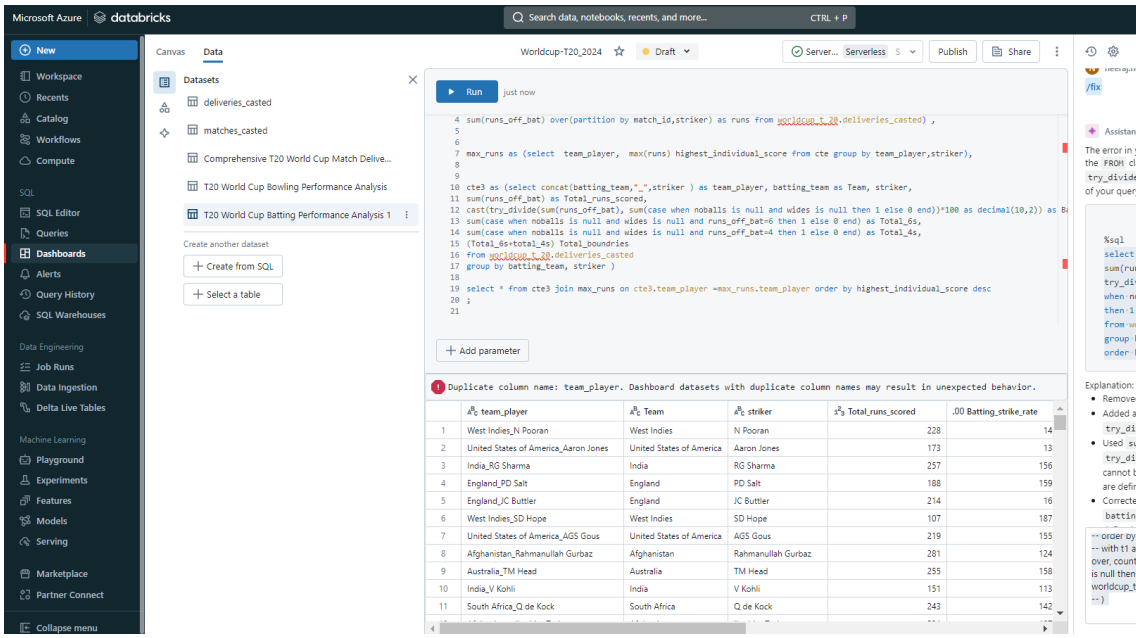
dashboard.



b. Create own customised SQL queries for dashboard

i. T20 World Cup Bowling Performance Analysis.sql

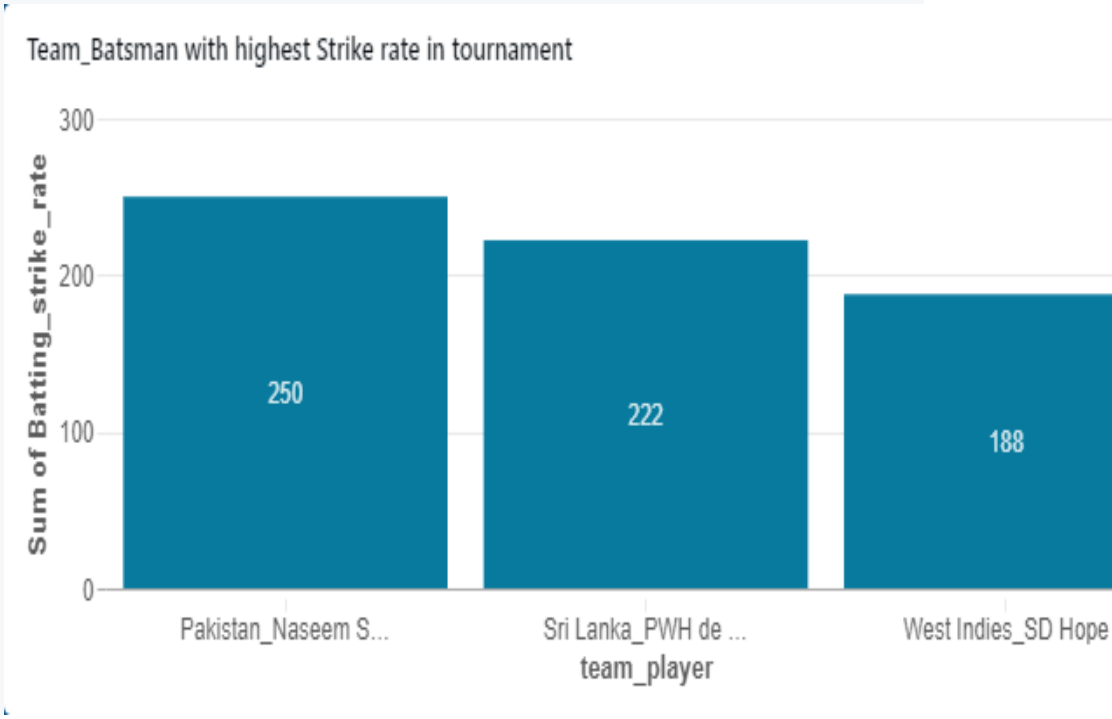
ii. T20 World Cup Batting Performance Analysis.sql



iii. Below mentioned queries and dashboard derived:

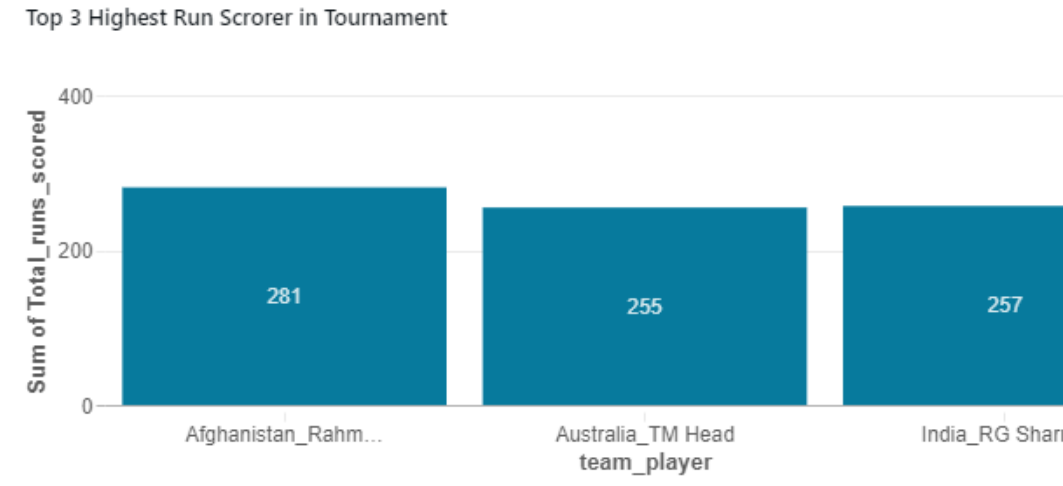
1. Team_Batsman with highest Strike rate in the tournament

```
select concat(batting_team,"_",striker ) as team_player,
batting_team as Team, striker,

cast(try_divide(sum(runs_off_bat), sum(case when noballs is
null and wides is null then 1 else 0 end))*100 as
decimal(10,2)) as Batting_strike_rate
from worldcup_t_20.deliveries_casted
group by batting_team, striker order by Batting_strike_rate
desc limit 3;
```



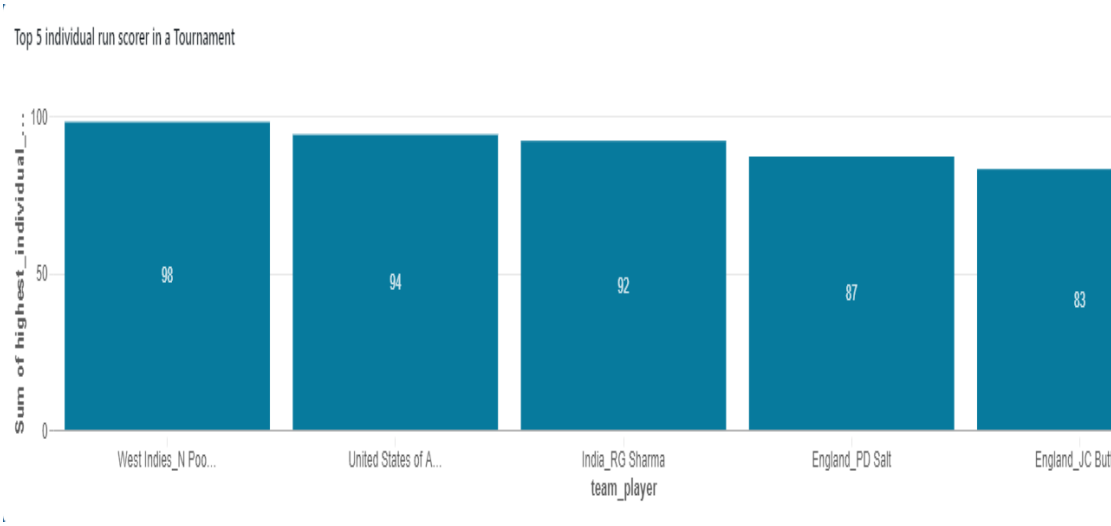
2. Top 3 Highest Run Scorer in Tournament

```
select concat(batting_team,"_",striker ) as team_player,
batting_team as Team, striker,
sum(runs_off_bat) as Total_runs_scored

from worldcup_t_20.deliveries_casted
group by batting_team, striker  order by Total_runs_scored
desc limit 3;
```



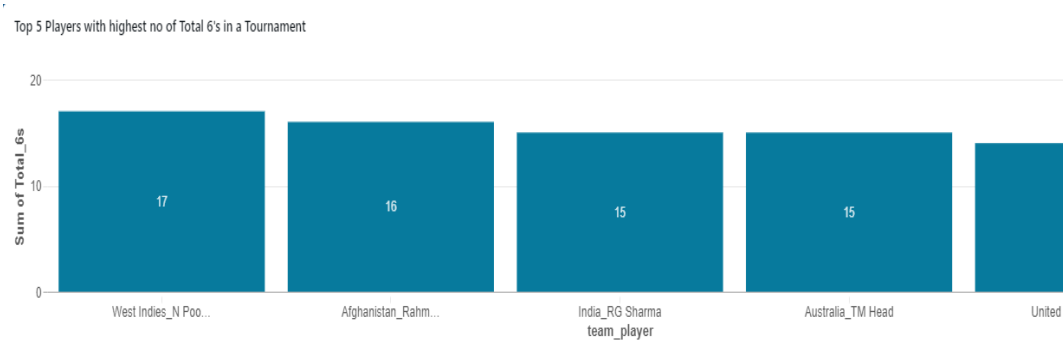
3. Top 5 individual run scorer in a Tournament

```
with cte as(select distinct concat(batting_team,"_",striker )
as team_player,match_id, batting_team as Team, striker,
sum(runs_off_bat) over(partition by match_id,striker) as runs
from worldcup_t_20.deliveries_casted)


select  team_player,  max(runs) highest_individual_score from
cte group by team_player,striker order by
highest_individual_score desc limit 5;
```



4. Top 5 Players with highest no of Total 6's in a Tournament

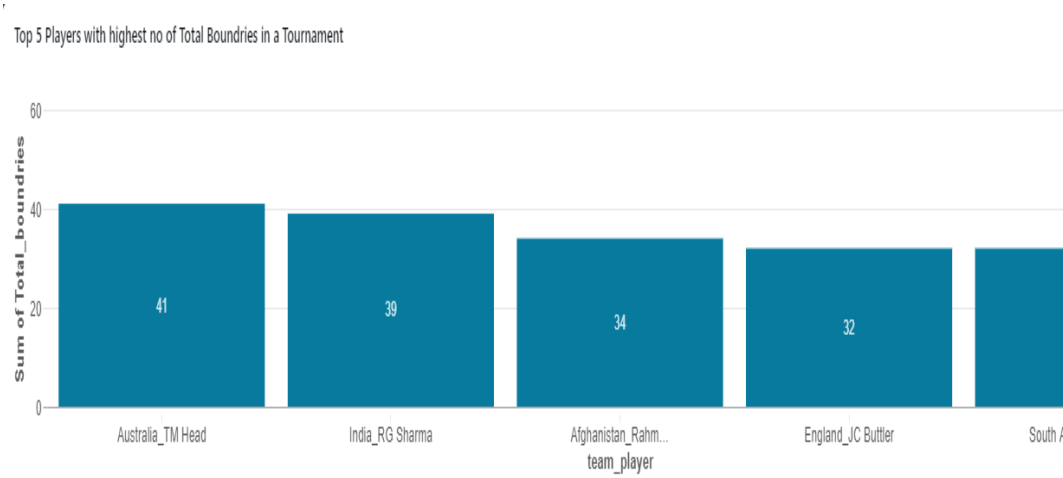
```
select concat(batting_team,"_",striker ) as team_player,
batting_team as Team, striker,
sum(runs_off_bat) as Total_runs_scored,
sum(case when noballs is null and wides is null and
runs_off_bat=6 then 1 else 0 end) as Total_6s,
sum(case when noballs is null and wides is null and
runs_off_bat=4 then 1 else 0 end) as Total_4s
from worldcup_t_20.deliveries_casted
group by batting_team, striker
order by   Total_6s desc limit 5;
```



5. Top 5 Players with highest no of Total Boundries in a Tournament

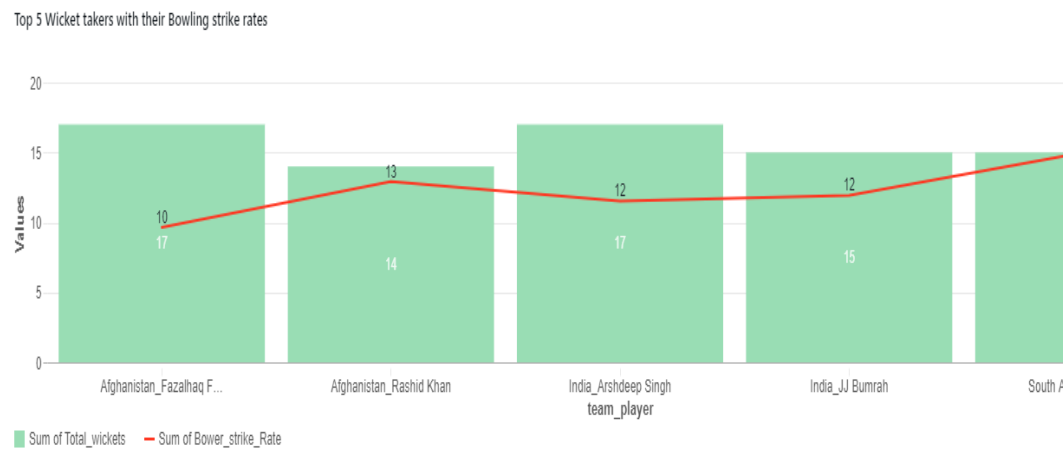
```
select concat(batting_team,"_",striker ) as team_player,
batting_team as Team, striker,
sum(runs_off_bat) as Total_runs_scored,
```

```
sum(case when noballs is null and wides is null and
runs_off_bat=6 then 1 else 0 end) as Total_6s,
sum(case when noballs is null and wides is null and
runs_off_bat=4 then 1 else 0 end) as Total_4s,
(Total_6s+total_4s) Total_boundries
from worldcup_t_20.deliveries_casted
group by batting_team, striker
order by  Total_boundries desc limit 5;
```



6. Top 5 Wicket takers with their Bowling strike rates

```
select concat(bowling_team,"_",bowler ) as
team_player,
bowling_team as Team, bowler,
cast(try_divide(sum(case when noballs != null or wides
!= null  then 0 else 1 end),sum(case when wicket_type
!= 'run out' then 1 else 0 end))as decimal(10,2)) as
Bower_strike_Rate,

sum(case when wicket_type != 'run out' then 1 else 0
end) as Total_wickets

from worldcup_t_20.deliveries_casted
group by bowling_team,bowler order by Total_wickets
desc limit 5;
```



7. **Top 10 Bowlers Bowled with lowest economy Rate with alteast 12 over bowles in Tournament**

```
select concat(bowling_team,"_",bowler ) as team_player,
bowling_team as Team, bowler,
count(distinct int(ball),match_id) as Total_over,
sum(runs_off_bat)+sum(extras) runs_Conceeded,
cast(try_divide(runs_Conceeded,Total_over) as decimal(10,2)) as
Ecomomy
from worldcup_t_20.deliveries_casted
group by bowling_team,bowler having Ecomomy is not
null  and  Total_over>=12
order by Ecomomy limit 10;
```

Top 10 Bowlers Bowled with lowest economy Rate with alteast 12 over bowles in Tournament

team_player	Total_over	runs_Conceeded	Ecomomy
New Zealand_TG Southee	12	39	3.25
New Zealand_TA Boult	16	64	4
New Zealand_LH Ferguson	16	66	4.13
India_JJ Bumrah	30	138	4.6
South Africa_OEG Baartman	19	98	5.16
Pakistan_Mohammad Amir	17	92	5.41
West Indies_RL Chase	15	82	5.47
New Zealand_MJ Santner	14	78	5.57
Bangladesh_Mustafizur Rahman	26	145	5.58
Netherlands_PA van Meekeren	15	85	5.67

iv. **Publish and schedule the baseboard to automate the data refresh**.

4. **Finally schedule the workflow to automate the data flow, in this case it is not requires but if source data is streaming or live than we can automate the workflow**

using job run option in databrick.