**IPL Batters SQL Queries**

**Syed Ali Ashraf.**

**Dataset – Kaggle**

**Tech-Stack – MySQL workbench 8.0**

**Imported data from csv file.**

**List all players along with the team they play for.**

select player\_name , team from batters;

**Show players who have scored more than 500 runs.**

select player\_name ,runs

from batters

where runs > 500;

**Display the player with the highest individual score (HS).**

select player\_name , high\_score

from batters;

**Retrieve players who have scored at least one century (100s).**

select player\_name , hundreds

from batters

where Hundreds >= 1;

**Show all players with an average (AVG) above 50.**

select player\_name , avg\_

from batters

where avg\_ > 50

order by avg\_;

**List players who haven’t scored any fifty (50s**).

select player\_name ,fiftys

from batters

where fiftys = 0;

**Count how many players belong to each team.**

select team , count(player\_name) as No\_of\_Players

from batters

group by team

order by No\_of\_Players desc;

**Retrieve the top 10 players based on strike rate (SR).**

select player\_name , sr

from batters

order by sr desc limit 10;

**Find players who have played more than 10 matches**.

select player\_name , matches

from batters

where matches > 10

order by matches desc;

Get the list of players whose highest score is greater than 100.

select player\_name , high\_score

from batters

where high\_score >=100;

**Show players with a strike rate between 90 and 150.**

select player\_name , sr

from batters

where sr between 90 and 150;

**Retrieve players with more than 50 fours (4s) and more than 10 sixes (6s).**

select player\_name ,fours,sixes

from batters

where fours>50 and sixes > 6;

**Display the top 5 players with most runs from a particular team (e.g., KKR).**

select team , player\_name , runs

from batters

where team = 'KKR'

order by runs desc limit 5;

**Find the average strike rate for each team.**

select team , avg(sr)

from batters

group by team ;

**Identify the team with the highest average runs per player.**

select team , avg(runs)/count(player\_name) as Average\_runs\_per\_player

from batters

group by team

order by Average\_runs\_per\_player desc ;

**Count total centuries scored across all players.**

select sum(hundreds) as Total\_Hundreds from batters;

**Find the total balls faced (BF) by each team.**

select team , sum(Balls\_Faced) as Balls\_Faced

from batters

group by team;

**Get the number of players who have a batting average over 70 in fewer than 20 innings.**

select count(player\_name) as No\_of\_Players

from batters

where avg\_>50 and innings <=10;

select player\_name as No\_of\_Players

from batters

where avg\_>50 and innings <=10;

**Rank players within each team based on runs scored.**

select team , player\_name , runs,

rank() over(partition by team order by runs desc ) as rank\_in\_team

from batters;

**List top 3 players from each team by average.**

with ranked as (

select team , player\_name , avg\_,

rank() over(partition by team order by avg\_ desc ) as avg\_in\_team

from batters)

select team , player\_name , avg\_ , avg\_in\_team from ranked

where avg\_in\_team <=3;

**Retrieve the player with the second-highest runs in the dataset.**

select player\_name, runs from batters

order by runs desc

limit 1 offset 1;

**Find the player with the lowest strike rate who has scored 500+ runs.**

select player\_name , runs , sr

from batters

where runs >500

order by sr limit 1;

**Sort all players by number of matches played in descending order.**

select player\_name , matches

from batters

order by matches desc ;

**Show top 5 players who have hit the most sixes.**

select player\_name , sixes

from batters

order by sixes desc limit 5;

**Identify the most consistent players: those with no ducks and high AVG**.

select player\_name , high\_score , avg\_

from batters

where high\_score <> 0

order by avg\_ desc limit 1;

**Create a summary table for each team: total runs, total sixes, and total matches.**

select team , sum(matches) as Total\_matches , sum(runs) as Total\_runs , sum(sixes) as Total\_sixes

from batters

group by team

order by team desc ;

**List players who’ve scored more than 1 century AND 5 fifties.**

select player\_name , hundreds , fiftys

from batters

where Hundreds =1 and fiftys >5;

**Compute rank of players by AVG within each team.**

select team, player\_name , avg\_ ,

rank() over ( partition by team order by avg\_ desc)

from batters;

**Find percentage contribution of each player's runs to their team's total.**

select team,player\_name,runs ,sum(runs) over (partition by team) as total\_runs,

round((runs\*100)/sum(runs) over ( partition by team),2) as percent\_contribution

from batters;

**Get difference between a player’s AVG and team’s average AVG.**

select team , player\_name , avg\_ as Batters\_Avg, avg(avg\_) over (partition by team ) as Teams\_avg,

round(avg\_-avg(avg\_) over ( partition by team),2) as avg\_difference

from batters;

**Identify each team’s top scorer using window functions.**

with rr as(

select team , player\_name , runs,

rank() over ( partition by team order by runs desc ) as Runs\_Rank

from batters)

select \* from rr

where runs\_rank=1;

**Compute percentile rank of players based on SR**.

select player\_name , sr ,

round(percent\_rank() over ( order by sr desc)\*100 ,2)as Percentile\_rank

from batters;.

**Use a CTE to show top performers who have more than 300 BF and SR > 130**

with performers as

(

select player\_name , balls\_faced , sr

from batters

where Balls\_faced>300 and SR>130 )

select \* from performers;

**List players who outperform team averages in both AVG and SR.**

select team , avg(avg\_) as team\_average , avg(sr) as team\_sr\_average

from batters

group by team)

select b.player\_name,b.team,a.team\_average, a.team\_sr\_average , b.avg\_,b.sr from

batters b join team\_avg a on b.team=a.team

where b.avg\_ > a.team\_average

and b.sr > a.team\_sr\_average;