**--creating table iplball**

create table ipl\_ball (id int,inning int,over int,ball int,batsman varchar(100),non\_striker varchar(100),bowler varchar(100),batsmen\_runs int,extra\_run int,total\_runs int,is\_wicket int,dismissal\_kind varchar(100),pyaler\_dismissal varchar(100),fielder varchar(100),extras\_type varchar(100),batting\_team varchar(100),bowling\_team varchar(100) );

**--copy data from csv in ipl ball**

copy ipl\_ball from 'C:\Program Files\PostgreSQL\16\data\copy data\IPL\_Ball.csv' delimiter ',' csv header;

**--making table of batsmen stats data**

create table batsmen\_stats as select count(distinct (id)) as matches\_played ,batsman,count(ball) as ball\_played ,sum(batsmen\_runs) as total\_runs , sum(is\_wicket) as times\_out,

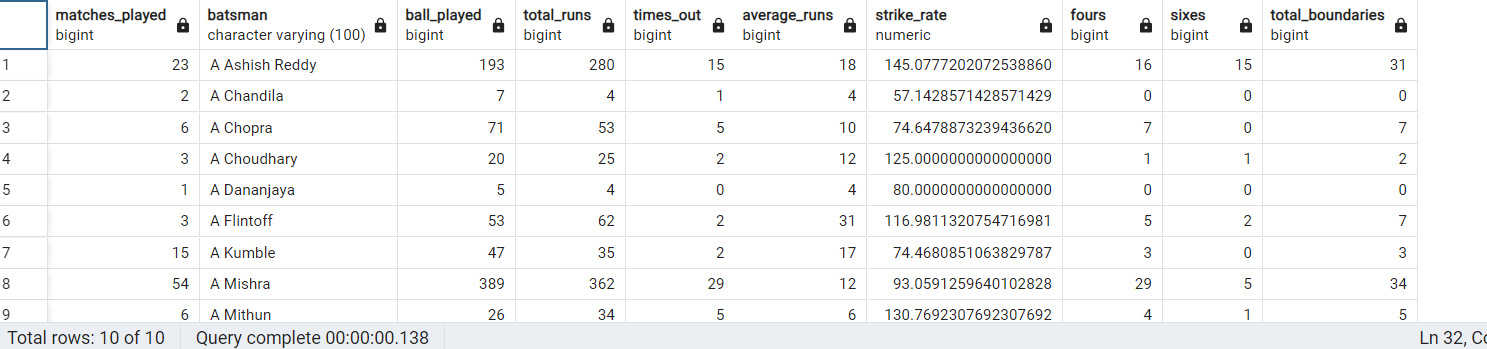
case when sum(is\_wicket)>0 then (SUM(batsmen\_runs) / NULLIF(SUM(is\_wicket), 0)) else SUM(batsmen\_runs) end AS average\_runs,

(SUM(batsmen\_runs) \* 100.0) / NULLIF( COUNT(\*) , 0) AS strike\_rate , SUM(CASE WHEN batsmen\_runs = 4 THEN 1 ELSE 0 END) AS fours,

SUM(CASE WHEN batsmen\_runs = 6 THEN 1 ELSE 0 END) AS sixes,SUM(CASE WHEN batsmen\_runs IN (4, 6) THEN 1 ELSE 0 END) AS total\_boundaries from ipl\_ball where not extras\_type= 'wides' group by batsman;

**--showing demo data of batsmen stats**

select \* from batsmen\_stats ;



**-- creating table from bowler stats**

create table bowler\_stats as SELECT bowler,

COUNT(DISTINCT id) AS matches\_played,

COUNT(ball) AS balls\_delivered,SUM(batsmen\_runs + CASE WHEN extras\_type NOT IN ('byes', 'legbyes') THEN extra\_run ELSE 0 END) as runs\_conseed,

(SUM(batsmen\_runs + CASE WHEN extras\_type NOT IN ('byes', 'legbyes') THEN extra\_run ELSE 0 END) / (COUNT(\*) / 6.0)) AS economy,

COUNT(ball) / NULLIF(SUM(is\_wicket), 0) AS strike\_rate,

SUM(is\_wicket) AS wickets\_taken

FROM

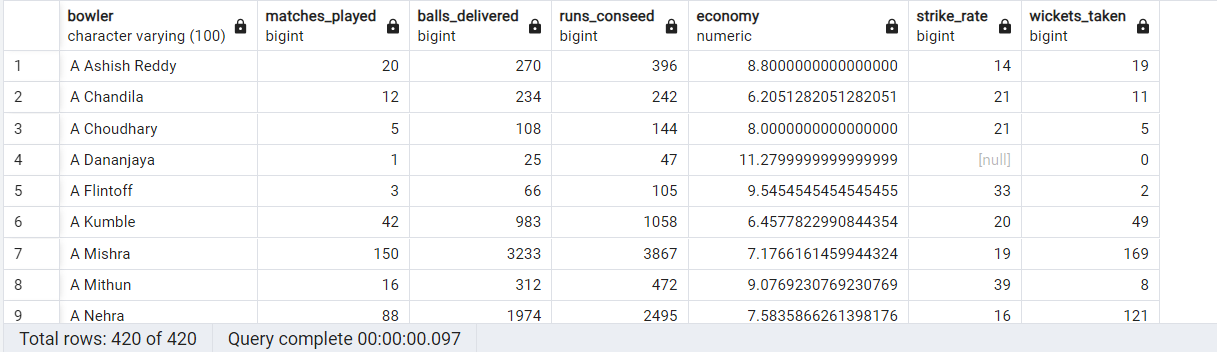
IPL\_Ball

GROUP BY

bowler;

**--showing demo of table bowler\_stats**

select \* from bowler\_stats ;



**/\* Your first priority is to get 2-3 players with high S.R who have faced at least 500 balls.**

**And to do that you have to make a list of 10 players you want to bid in the auction so that when**

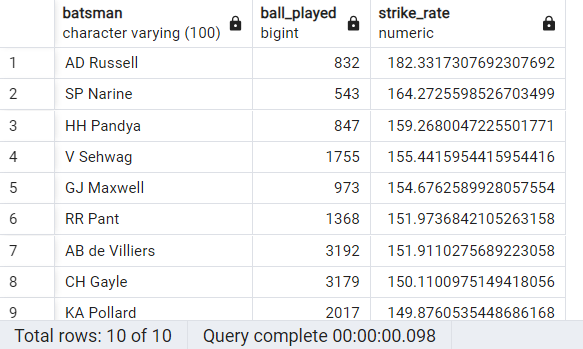
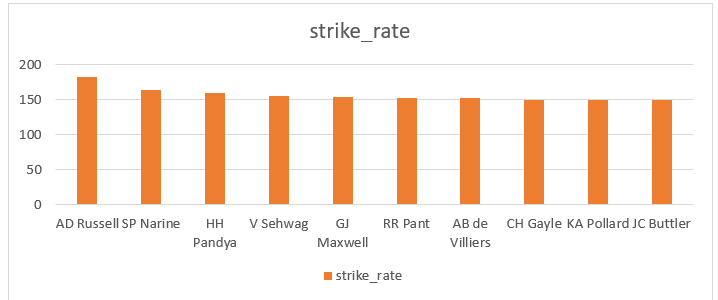
**you try to grab them in auction you should not pay the amount greater than you have in the purse**

**for a particular player.\*/**

select batsman ,ball\_played,strike\_rate from batsmen\_stats where ball\_played>500

order by strike\_rate desc

limit 10;

**/\* Now you need to get 2-3 players with good Average who have played more the 2 ipl seasons.**

**And to do that you have to make a list of 10 players you want to bid in the auction so that**

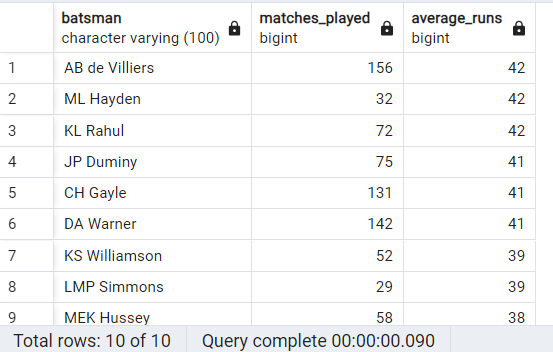
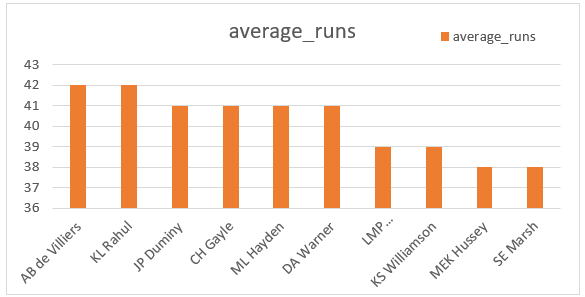
**when you try to grab them in auction you should not pay the amount greater than you have in**

**the purse for a particular player. \*/**

select batsman,matches\_played,average\_runs from batsmen\_stats where matches\_played>28

order by average\_runs desc

limit 10;

**/\* Now you need to get 2-3 Hard-hitting players who have scored most runs in boundaries and have**

**played more the 2 ipl season. To do that you have to make a list of 10 players you want to bid in**

**the auction so that when you try to grab them in auction you should not pay the amount greater than**

**you have in the purse for a particular player.\*/**

SELECT

batsman,matches\_played,

total\_runs,fours,sixes,

(fours \* 4 + sixes \* 6) AS total\_boundary\_runs,

CASE

WHEN total\_runs = 0 THEN 0

ELSE ((fours \* 4 + sixes \* 6) / total\_runs::float) \* 100

END AS boundary\_percentage

FROM

batsmen\_stats

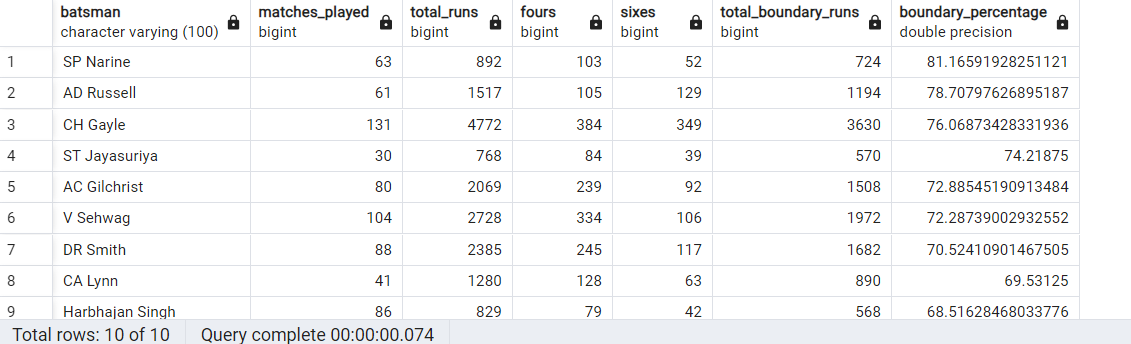
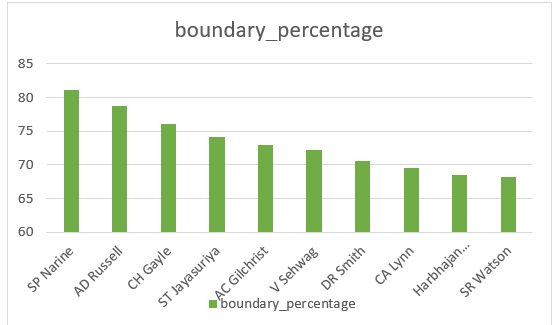
where

matches\_played>28

ORDER BY

boundary\_percentage DESC

limit 10;

**/\* Your first priority is to get 2-3 bowlers with good economy who have bowled at least 500 balls**

**in IPL so far.To do that you have to make a list of 10 players you want to bid in the auction so**

**that when you try to grab them in auction you should not pay the amount greater than you have in**

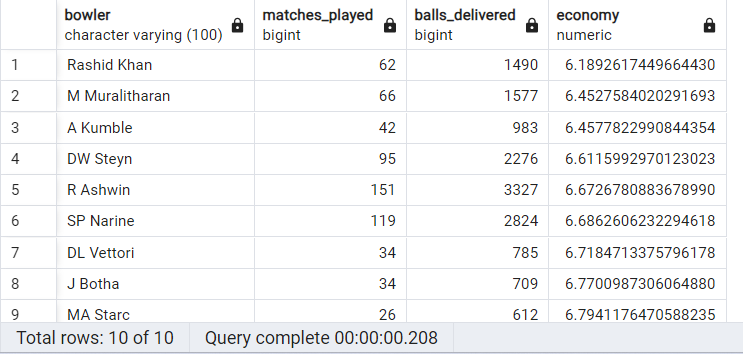
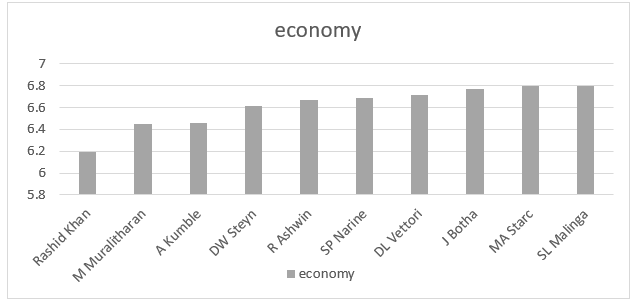
**the purse for a particular player.\*/**

select bowler,matches\_played , balls\_delivered ,economy from bowler\_stats

where balls\_delivered>500

order by economy

limit 10;

**/\* Now you need to get 2-3 bowlers with the best strike rate and who have bowled at least 500 balls**

**in IPL so far.To do that you have to make a list of 10 players you want to bid in the auction so that**

**when you try to grab them in auction you should not pay the amount greater than you have in the purse**

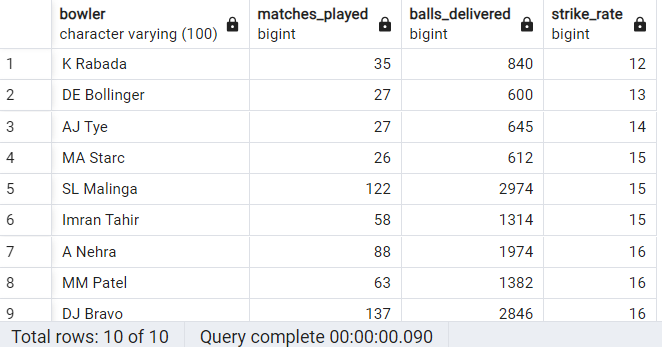
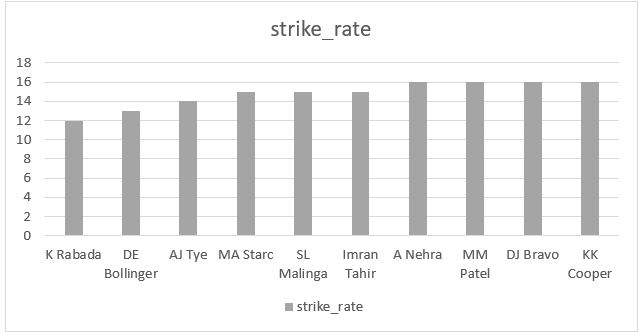
**for a particular player.\*/**

select bowler,matches\_played,balls\_delivered,strike\_rate from bowler\_stats

where balls\_delivered>500

order by strike\_rate

limit 10;

**/\* Now you need to get 2-3 All\_rounders with the best batting as well as bowling strike rate and who**

**have faced at least 500 balls in IPL so far and have bowled minimum 300 balls.To do that you have**

**to make a list of 10 players you want to bid in the auction so that when you try to grab them in**

**auction you should not pay the amount greater than you have in the purse for a particular player.\*/**

**-- creating table for allrounders\_stats**

create table allrounder\_stats as select a.matches\_played,a.batsman as allrounder,a.ball\_played,a.total\_runs,a.average\_runs,a.strike\_rate as batting\_strikerate

,a.fours,a.sixes,b.balls\_delivered,b.runs\_conseed,b.economy,b.strike\_rate as bowling\_strikerate,b.wickets\_taken

from batsmen\_stats as a join bowler\_stats as b

on a.batsman=b.bowler;

**--showig data of all rounder**

select \* from allrounder\_stats;

**-- selecting allrounder for team**

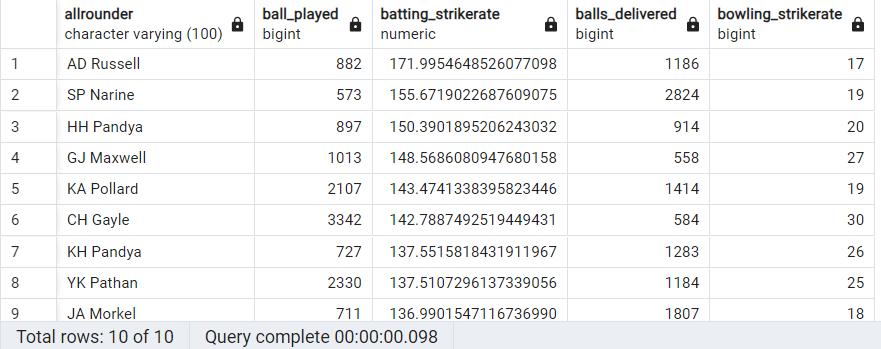
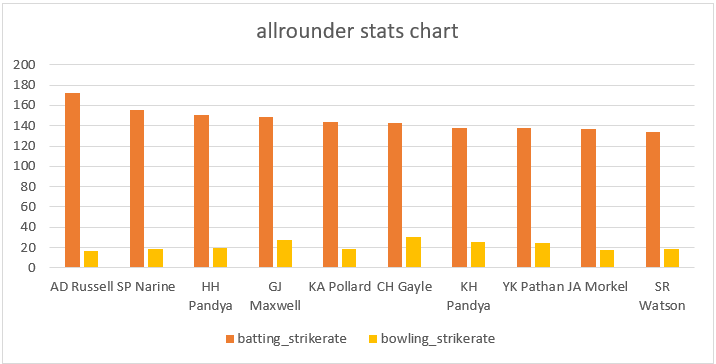
select allrounder,ball\_played,batting\_strikerate,balls\_delivered, bowling\_strikerate

from allrounder\_stats

where ball\_played>499 and balls\_delivered>299

order by batting\_strikerate desc,bowling\_strikerate desc

limit 10;

**/\* selecting a wicketkeeper for a team\*/**

**Explaination –** I am selecting a wicket keeper on the bases of the catches taken by a wicket keeper which is a most important factor of wicketkeeper as data given and I also used their strike rate.

So that the wicket keeper should contribute to team total runs with a good strike rate which help to score quick runs in t20 . I have taken top 30 data of catch taken and filter wicket

Wicket keeper using wicket keeper name list and then I take the strike rate data in order to get top 5 of result for target. Luckily I have 3 of them already in target list so I mention them as

Wicket keeper batsmen and remaining 2 I mentioned as wicket keeper.

select a.matches\_played,a.batsman,a.strike\_rate,b.catches\_taken from batsmen\_stats as a join (select fielder,count(dismissal\_kind) as catches\_taken

from ipl\_ball where fielder in (select distinct(fielder) from ipl\_ball where dismissal\_kind='caught')

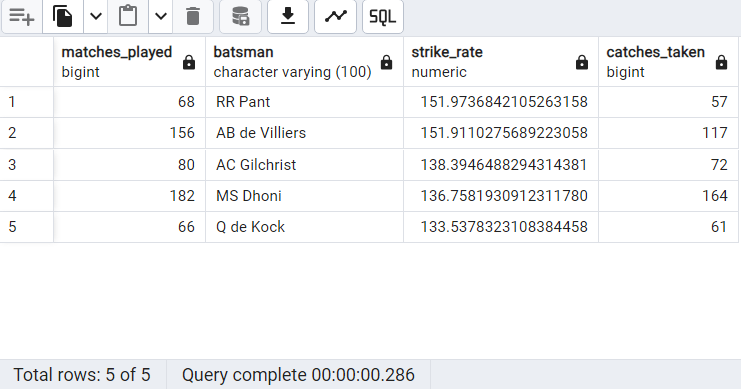
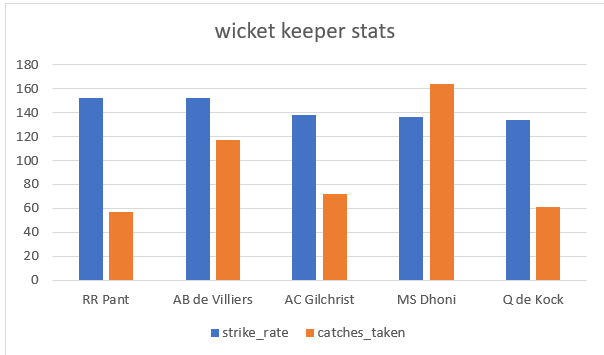
group by fielder

order by catches\_taken desc limit 30

) as b on b.fielder=a.batsman

where a.batsman in ('MS Dhoni','KD Karthik','RV Uthappa','AB de Villiers','PA Patel','WP Saha','NV Ojha','AC Gilchrist','Q de Kock','RR Pant','KC Sangakkara')

order by strike\_rate desc limit 5;

* Selected players to target in auction

|  |  |
| --- | --- |
| **Players name** | **Cateogery** |
| AD Russell | Allrounder |
| SP Narine | Allrounder |
| HH Pandya | Allrounder |
| GJ Maxwell | Allrounder |
| KA Pollard | Allrounder |
| CH Gayle | Allrounder |
| Harbhajan Singh | Allrounder |
| SR Watson | Allrounder |
| DJ Bravo | Allrounder |
| KH Pandya | Allrounder |
| YK Pathan | Allrounder |
| JA Morkel | Allrounder |
| JC Buttler | Batsman |
| KL Rahul | Batsman |
| V Sehwag | Batsman |
| JP Duminy | Batsman |
| ML Hayden | Batsman |
| DA Warner | Batsman |
| LMP Simmons | Batsman |
| KS Williamson | Batsman |
| MEK Hussey | Batsman |
| SE Marsh | Batsman |
| ST Jayasuriya | Batsman |
| DR Smith | Batsman |
| CA Lynn | Batsman |
| Rashid Khan | Bowler |
| M Muralitharan | Bowler |
| A Kumble | Bowler |
| DW Steyn | Bowler |
| R Ashwin | Bowler |
| DL Vettori | Bowler |
| J Botha | Bowler |
| MA Starc | Bowler |
| SL Malinga | Bowler |
| K Rabada | Bowler |
| DE Bollinger | Bowler |
| AJ Tye | Bowler |
| Imran Tahir | Bowler |
| A Nehra | Bowler |
| MM Patel | Bowler |
| KK Cooper | Bowler |
| AB de Villiers | Wicket keeper and batsmen |
| AC Gilchrist | Wicket keeper and batsmen |
| RR Pant | Wicket keeper and batsmen |
| MS Dhoni | wicket keeper |
| Q de Kock | wicket keeper |

Note\*- I donated a wicket keeper and batsmen because they are already selected in team but I get them in wicket

Table result also

-- Additional questions

**--creating table deliveries**

create table deliveries (id int,inning int,over int,ball int,batsman varchar(100),non\_striker varchar(100),bowler varchar(100),batsmen\_runs int,extra\_run int,total\_runs int,is\_wicket int,dismissal\_kind varchar(100),pyaler\_dismissal varchar(100),fielder varchar(100),extras\_type varchar(100),batting\_team varchar(100),bowling\_team varchar(100) );

**--copy data from csv in ipl ball**

copy deliveries from 'C:\Program Files\PostgreSQL\16\data\copy data\IPL\_Ball.csv' delimiter ',' csv header;

**--showing data of iplball**

select \* from deliveries;

**--creating table ipl maches**

create table matches (id int,city varchar(100),date date,player\_of\_match varchar(100),venue varchar(200),neutral\_vanue int,team1 varchar(100),team2 varchar(100),toss\_winnwe varchar(100),toss\_decision varchar(100),winner varchar(100),result varchar(100),result\_margin int,eliminator varchar(100),method varchar(100),umpire1 varchar(100),umpire2 varchar(100));

**--copy data from csv in ipl maches**

copy matches from 'C:\Program Files\PostgreSQL\16\data\copy data\IPL\_matches.csv' delimiter ',' csv header;

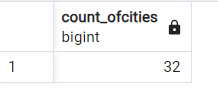
**--showing data of ipl\_maches**

select \* from matches;

**Q1- Get the count of cities that have hosted an IPL match?**

select count(distinct city) as count\_ofcities from matches

where NOT city ='NA';



**Q2- Create table deliveries\_v02 with all the columns of the table ‘deliveries’ and an additional**

**column ball\_result containing values boundary, dot or other depending on the total\_run**

**(boundary for >= 4, dot for 0 and other for any other number)**

**(Hint 1 : CASE WHEN statement is used to get condition based results)**

**(Hint 2: To convert the output data of the select statement into a table, you can use a**

**subquery. Create table table\_name as [entire select statement].**

CREATE TABLE DELIVERIES\_V02 AS

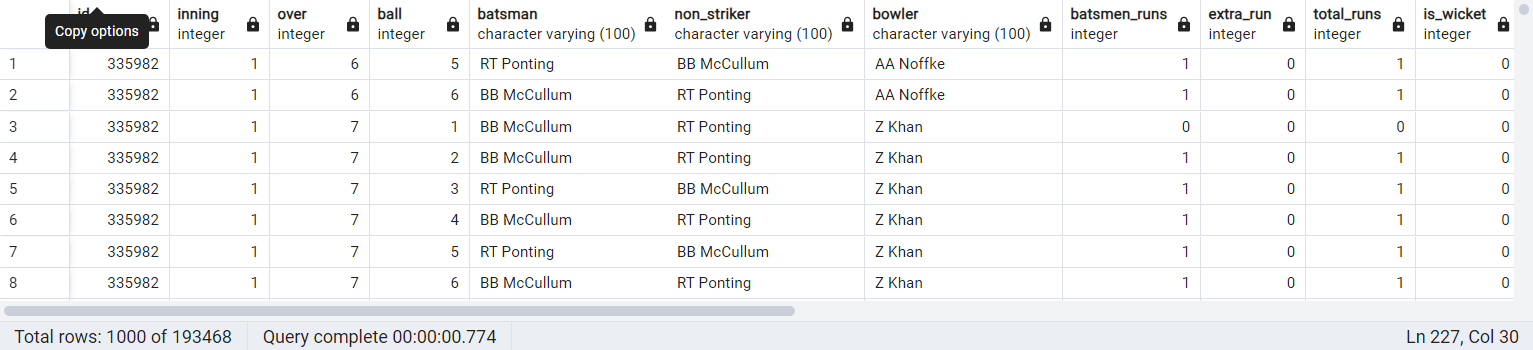
SELECT id,inning,over,ball,batsman,non\_striker,bowler,batsmen\_runs,extra\_run,total\_runs,is\_wicket,dismissal\_kind,

pyaler\_dismissal as player\_dissmisal,fielder,extras\_type,batting\_team,bowling\_team ,CASE WHEN total\_runs >= 4 THEN 'boundary' WHEN total\_runs = 0 THEN 'dot' ELSE 'other' END AS ball\_result

FROM DELIVERIES;

**--selecting deliveries v02 data**

select \* from DELIVERIES\_V02;

 note\*- other columns are not able to capture in this screen shot

**Q3- Write a query to fetch the total number of boundaries and dot balls from the**

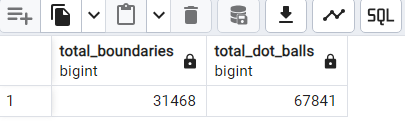
**deliveries\_v02 table**

SELECT

SUM(CASE WHEN ball\_result = 'boundary' THEN 1 ELSE 0 END) AS total\_boundaries,

SUM(CASE WHEN ball\_result = 'dot' THEN 1 ELSE 0 END) AS total\_dot\_balls

FROM deliveries\_v02;



**Q4-Write a query to fetch the total number of boundaries scored by each team from the**

**deliveries\_v02 table and order it in descending order of the number of boundaries**

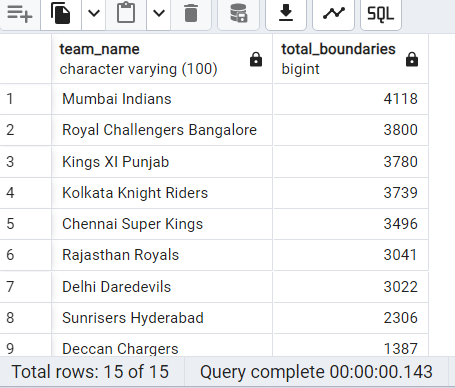
**scored**

select distinct(batting\_team) as team\_name , SUM(CASE WHEN ball\_result='boundary' then 1 else 0 END) as total\_boundaries

from DELIVERIES\_V02

GROUP BY batting\_team

order by total\_boundaries desc;



**Q5 -Write a query to fetch the total number of dot balls bowled by each team and order it in**

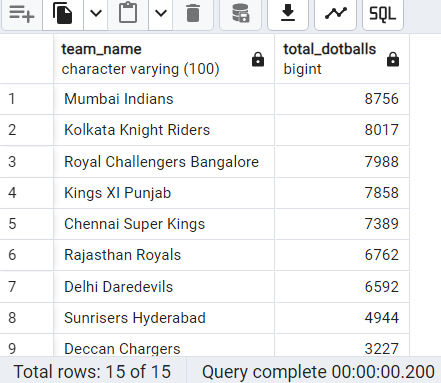
**descending order of the total number of dot balls bowled**.

select distinct(batting\_team) as team\_name , SUM(CASE WHEN ball\_result='dot' then 1 else 0 END) as total\_dotballs

from DELIVERIES\_V02

GROUP BY batting\_team

order by total\_dotballs desc;



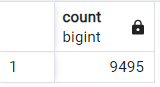
**Q6-Write a query to fetch the total number of dismissals by dismissal kinds where dismissal**

**kind is not NA**

select count(dismissal\_kind)

from deliveries

where NOT dismissal\_kind='NA';



**Q7 Write a query to get the top 5 bowlers who conceded maximum extra runs from the**

**deliveries table**

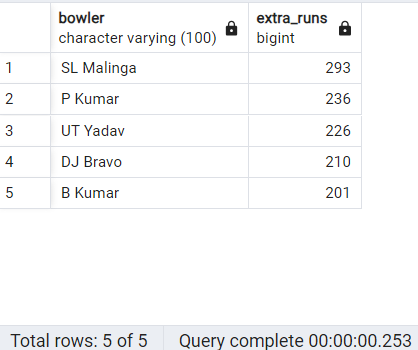
select bowler,sum(extra\_run) as extra\_runs

from deliveries

group by bowler

order by extra\_runs desc

limit 5;



**Q8- Write a query to create a table named deliveries\_v03 with all the columns of**

**deliveries\_v02 table and two additional column (named venue and match\_date) of venue**

**and date from table matches**

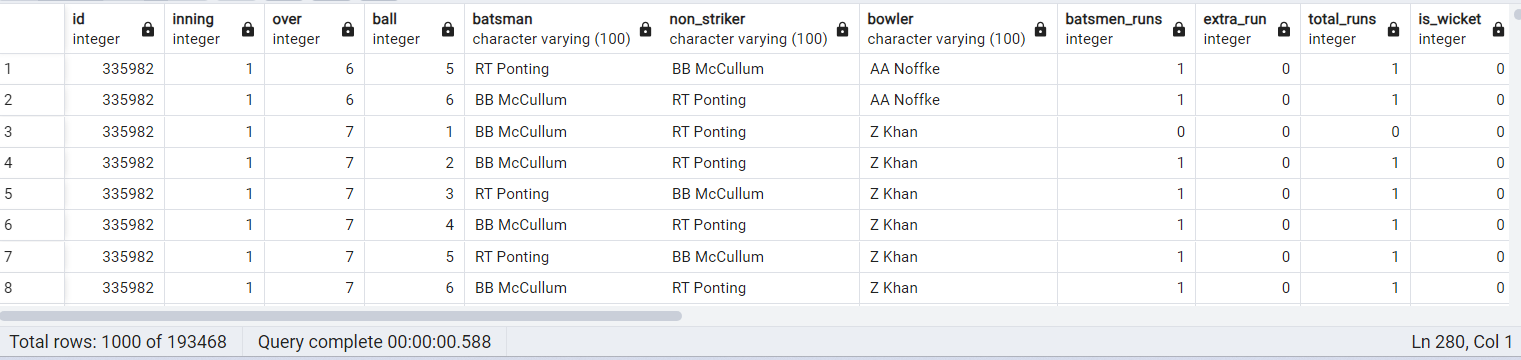
create table deliveries\_v03 as select d.\*,m.venue,

m.date as match\_date

from DELIVERIES\_V02 as d join matches as m

on m.id=d.id;

select \* from deliveries\_v03;

 note\*- other columns are not able to capture in this screen shot

**Q9-Write a query to fetch the total runs scored for each venue and order it in the descending**

**order of total runs scored**

select venue,sum(total\_runs) as total\_runs

from deliveries\_v03

group by venue

order by total\_runs desc;



**Q-10 Write a query to fetch the year-wise total runs scored at Eden Gardens and order it in the**

**descending order of total runs scored**

select extract(year from match\_date) as year ,sum(total\_runs) as total\_runs from deliveries\_v03

where venue = 'Eden Gardens'

group by extract(year from match\_date)

order by total\_runs desc;

