Table of Contents

[Queries 1](#_Toc159004727)

[1. Create a table named ‘matches’ with appropriate data types for columns 1](#_Toc159004728)

[2. Create a table named ‘deliveries’ with appropriate data types for columns 2](#_Toc159004729)

[3. Select the top 20 rows of the *deliveries*table after ordering them by id, inning, over, ball in ascending order. 3](#_Toc159004730)

[4. Fetch data of all the matches played on 2nd May 2013 from the *matches*table.. 3](#_Toc159004731)

[5. Fetch data of all the matches where the result mode is ‘runs’ and margin of victory is more than 100 runs. 4](#_Toc159004732)

[6. Fetch data of all the matches where the final scores of both teams tied and order it in descending order of the date 4](#_Toc159004733)

[7. Get the count of cities that have hosted an IPL match. 5](#_Toc159004734)

[8. Write a query to fetch the total number of boundaries and dot balls from the *deliveries\_v02*table. 5](#_Toc159004735)

[9. 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. 6](#_Toc159004736)

[10. 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. 7](#_Toc159004737)

[11. Write a query to fetch the total number of dismissals by dismissal kinds where dismissal kind is not NA 8](#_Toc159004738)

[12. Write a query to get the top 5 bowlers who conceded maximum extra runs from the *deliveries*table 8](#_Toc159004739)

[*13.* 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* 9](#_Toc159004740)

[14. Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored. 10](#_Toc159004741)

[15. 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. 10](#_Toc159004742)

[16. Get unique team1 names from the *matches*table, you will notice that there are two entries for*Rising Pune Supergiant* one with *Rising Pune Supergiant* and another one with *Rising Pune Supergiant****s***.  Your task is to create a *matches\_corrected*table with two additional columns *team1\_corr*and *team2\_corr*containing team names with replacing *Rising Pune Supergiant****s*** with *Rising Pune Supergiant*. Now analyse these newly created columns. 11](#_Toc159004743)

[17. Create a new table deliveries\_v04 with the first column as ball\_id containing information of match\_id, inning, over and ball separated by ‘-’ (For ex. 335982-1-0-1 match\_id-inning-over-ball) and rest of the columns same as deliveries\_v03) 11](#_Toc159004744)

[18. Compare the total count of rows and total count of distinct ball\_id in deliveries\_v04 11](#_Toc159004745)

# Queries

## Create a table named ‘matches’ with appropriate data types for columns

-- Table: public.deliveries

-- DROP TABLE IF EXISTS public.deliveries;

CREATE TABLE IF NOT EXISTS public.deliveries

(

id integer NOT NULL,

inning integer NOT NULL,

over integer,

ball integer,

batsman character varying COLLATE pg\_catalog."default",

non\_striker character varying COLLATE pg\_catalog."default",

bowler character varying COLLATE pg\_catalog."default",

batsman\_runs integer,

extra\_runs integer,

total\_runs integer,

is\_wicket integer,

dismissal\_kind character varying COLLATE pg\_catalog."default",

player\_dismissed character varying COLLATE pg\_catalog."default",

fielder character varying COLLATE pg\_catalog."default",

extras\_type character varying COLLATE pg\_catalog."default",

batting\_team character varying COLLATE pg\_catalog."default",

bowling\_team character varying COLLATE pg\_catalog."default",

CONSTRAINT "FK\_MATCH\_ID" FOREIGN KEY (id)

REFERENCES public.matches ("ID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.deliveries

OWNER to postgres;

## Create a table named ‘deliveries’ with appropriate data types for columns

-- Table: public.matches

-- DROP TABLE IF EXISTS public.matches;

CREATE TABLE IF NOT EXISTS public.matches

(

id integer NOT NULL,

city character varying COLLATE pg\_catalog."default",

date date,

player\_of\_match character varying COLLATE pg\_catalog."default",

venue character varying COLLATE pg\_catalog."default",

neutral\_venue integer,

team1 character varying COLLATE pg\_catalog."default",

team2 character varying COLLATE pg\_catalog."default",

toss\_winner character varying COLLATE pg\_catalog."default",

toss\_decision character varying COLLATE pg\_catalog."default",

winner character varying COLLATE pg\_catalog."default",

result character varying COLLATE pg\_catalog."default",

result\_margin integer,

eliminator "char",

method character varying COLLATE pg\_catalog."default",

umpire1 character varying COLLATE pg\_catalog."default",

umpire2 character varying COLLATE pg\_catalog."default",

CONSTRAINT "IPL\_MATCH\_pkey" PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.matches

OWNER to postgres;

## Select the top 20 rows of the *deliveries*table after ordering them by id, inning, over, ball in ascending order.

select

\*

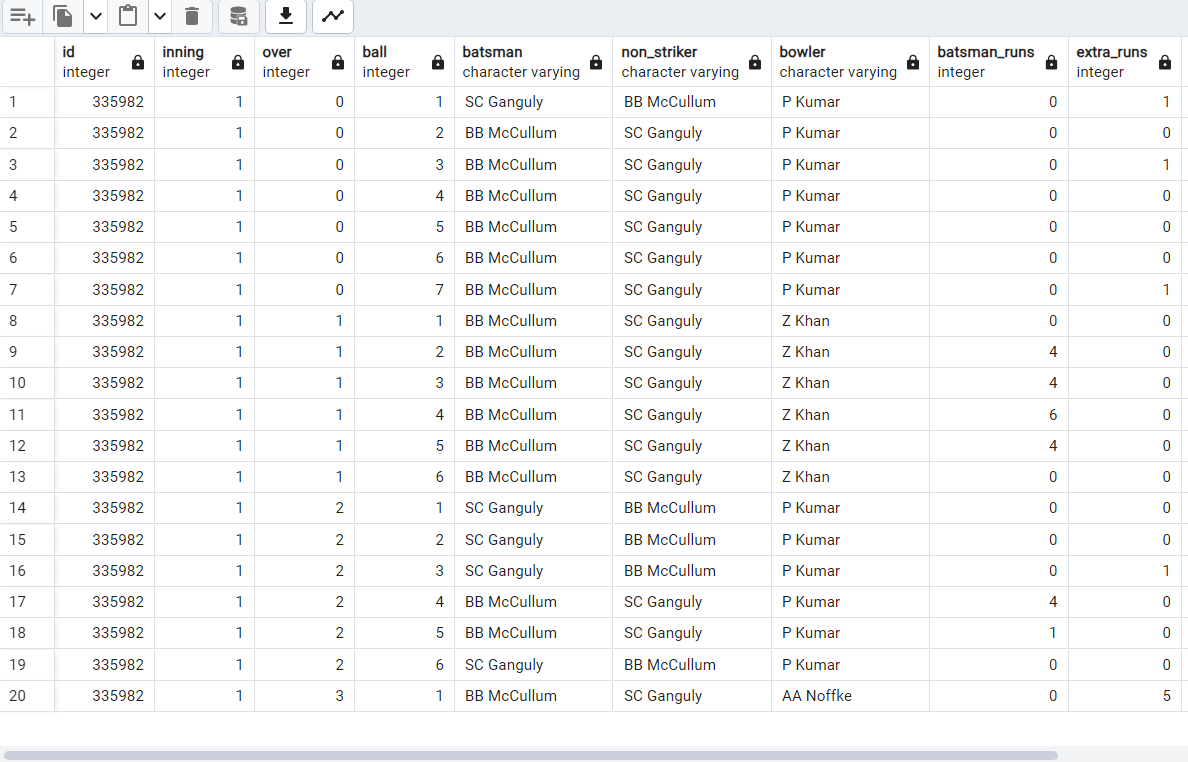
from

deliveries

order by

id, inning, over, ball asc

limit 20;



## Fetch data of all the matches played on 2nd May 2013 from the *matches*table..

select

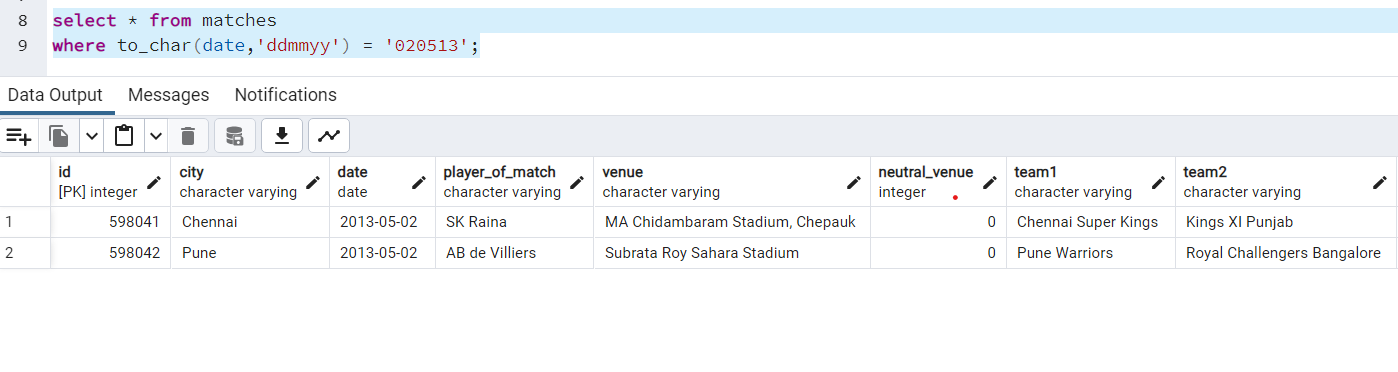
\*

from

matches

where

to\_char(date,'ddmmyy') = '020513';



## Fetch data of all the matches where the result mode is ‘runs’ and margin of victory is more than 100 runs.

select

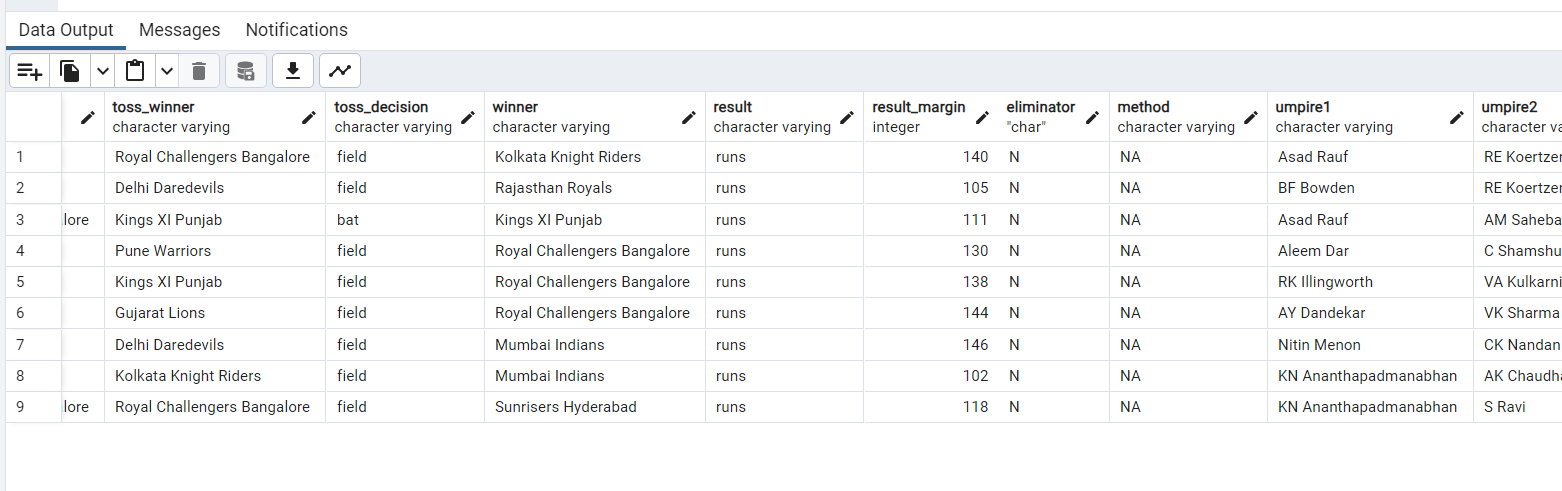
\*

from

matches

where

result = 'runs' and result\_margin > 100;



## Fetch data of all the matches where the final scores of both teams tied and order it in descending order of the date

select

\*

from

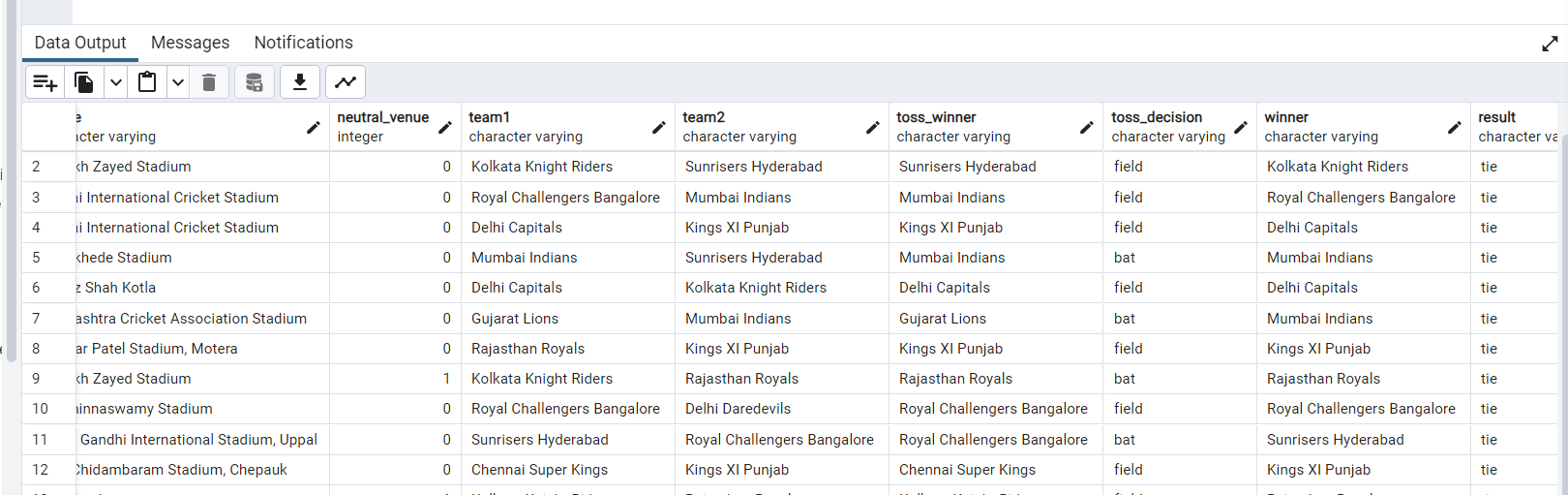
matches

where

result = 'tie'

order by

date desc;



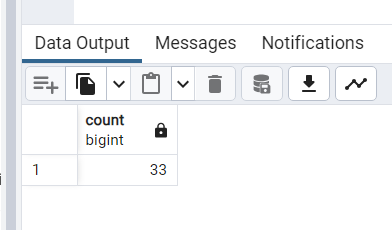
## Get the count of cities that have hosted an IPL match.

select

count(distinct City)

from

matches;



## Write a query to fetch the total number of boundaries and dot balls from the *deliveries\_v02*table.

select

ball\_result,count(\*)

from

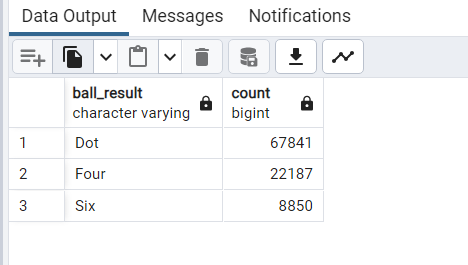
deliveries\_v02

where

ball\_result in ('Dot','Four','Six')

group by

ball\_result;



## 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

batting\_Team,ball\_result,count(\*) as Boundaries

from

deliveries\_v02

where

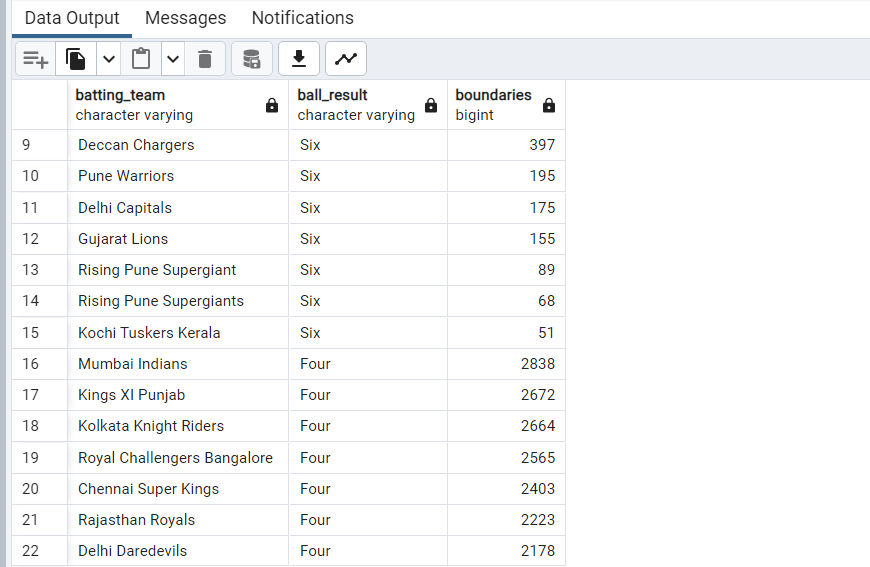
ball\_result in ('Four','Six')

group by

batting\_Team,ball\_result

order by

Ball\_result Desc, Boundaries desc;



## 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

bowling\_Team,ball\_result,count(\*) as Dots

from

deliveries\_v02

where

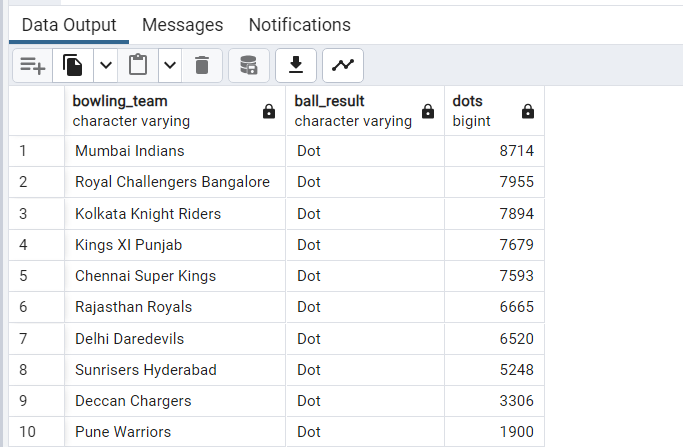
ball\_result in ('Dot')

group by

bowling\_Team,ball\_result

order by

Dots desc;



## Write a query to fetch the total number of dismissals by dismissal kinds where dismissal kind is not NA

select

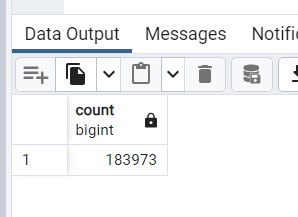
count(\*)

from

deliveries

where

dismissal\_kind = 'NA';



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

select

Bowler, sum(Extra\_Runs) as Total\_Extras

from

deliveries

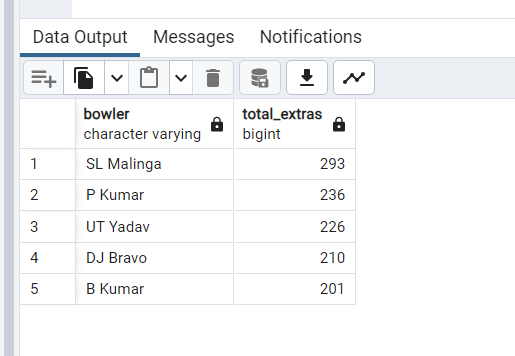
group by

Bowler

order by

Total\_Extras desc

limit 5;



## 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 \* from deliveries\_v02

where 1 = 2

alter table deliveries\_v03

add column venue character varrying;

alter table deliveries\_v03

add column Match\_date Date;

insert into deliveries\_v03

select d.\*,m.venue,m.date

from deliveries\_v02 as d inner join

matches m on m.id = d.id;

## 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)

from

deliveries\_v03

group by

venue

order by

sum(total\_runs) desc;

select

venue, sum(total\_runs)

from

deliveries d inner join matches m

on m.id = d.id

group by

venue

order by

sum(total\_runs) desc;

## 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

to\_char(Match\_date,'yyyy'),sum(total\_runs)

from

deliveries\_v03

where

venue = 'Eden Gardens'

group by

to\_char(Match\_date,'yyyy');

select

to\_char(date,'yyyy'),sum(total\_runs)

from

deliveries d inner join matches m

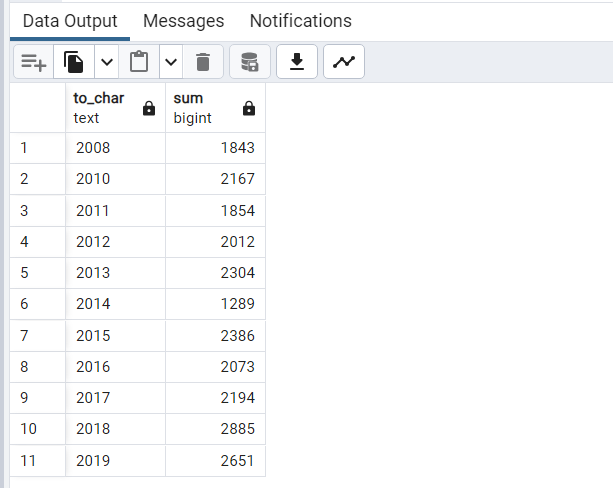
on m.id = d.id

where

venue = 'Eden Gardens'

group by

to\_char(date,'yyyy');



## Get unique team1 names from the *matches*table, you will notice that there are two entries for*Rising Pune Supergiant* one with *Rising Pune Supergiant* and another one with *Rising Pune Supergiant****s***.  Your task is to create a *matches\_corrected*table with two additional columns *team1\_corr*and *team2\_corr*containing team names with replacing *Rising Pune Supergiant****s*** with *Rising Pune Supergiant*. Now analyse these newly created columns.

update matches

set team1 = 'Rising Pune Supergiant',

team2 = 'Rising Pune Supergiant'

where

team1 = 'Rising Pune Supergiants'

or

team2 = 'Rising Pune Supergiants';

## Create a new table deliveries\_v04 with the first column as ball\_id containing information of match\_id, inning, over and ball separated by ‘-’ (For ex. 335982-1-0-1 match\_id-inning-over-ball) and rest of the columns same as deliveries\_v03)

insert into deliveries\_v04

select \*, concat(id,'-',inning,'-',over,'-',ball)

from deliveries\_v03;

## Compare the total count of rows and total count of distinct ball\_id in deliveries\_v04

select count(\*) total\_count ,count(distinct ball\_id) ballid\_count

from deliveries\_v04;

