### **Table of Contents**

Quer	ies2
1.	Create a table named 'matches' with appropriate data types for columns2
2.	Create a table named 'deliveries' with appropriate data types for columns2
3. aso	Select the top 20 rows of the <i>deliveries</i> table after ordering them by id, inning, over, ball in tending order
4.	Fetch data of all the matches played on 2nd May 2013 from the matches table4
5. tha	Fetch data of all the matches where the result mode is 'runs' and margin of victory is more an 100 runs4
6. de:	Fetch data of all the matches where the final scores of both teams tied and order it in scending order of the date5
7.	Get the count of cities that have hosted an IPL match5
8. the	Write a query to fetch the total number of boundaries and dot balls from e deliveries_v02 table
9.	Write a query to fetch the total number of boundaries scored by each team from e deliveries_v02 table and order it in descending order of the number of boundaries scored 6
10. de:	Write a query to fetch the total number of dot balls bowled by each team and order it in scending order of the total number of dot balls bowled
11. kin	Write a query to fetch the total number of dismissals by dismissal kinds where dismissal d is not NA
12. the	Write a query to get the top 5 bowlers who conceded maximum extra runs from e deliveries table
	Write a query to create a table named <i>deliveries_v03</i> with all the columns <i>deliveries_v02</i> table and two additional column (named <i>venue</i> and <i>match_date</i> ) <i>venue</i> and <i>date</i> from table <i>matches</i>
	Write a query to fetch the total runs scored for each venue and order it in the descending der of total runs scored
15. de:	Write a query to fetch the year-wise total runs scored at <i>Eden Gardens</i> and order it in the scending order of total runs scored
Pu col	tries for <i>Rising Pune Supergiant</i> one with <i>Rising Pune Supergiant</i> and another one with <i>Rising ne Supergiants</i> . Your task is to create a <i>matches_corrected</i> table with two additional umns <i>team1_corr</i> and <i>team2_corr</i> containing team names with replacing <i>Rising Pune</i>
17. of	cergiants with Rising Pune Supergiant. Now analyse these newly created columns
18.	

#### Queries

(

```
1. 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;
   2. 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;
```

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

```
select

*

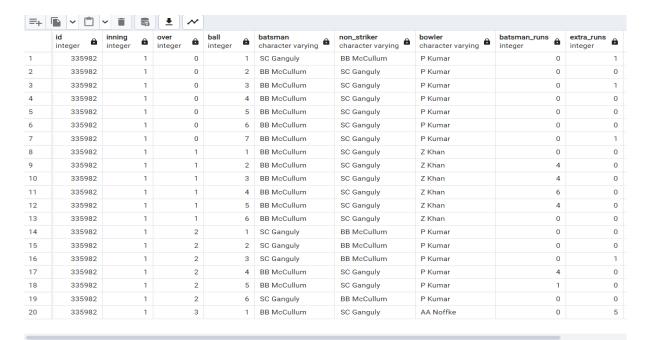
from

deliveries

order by

id, inning, over, ball asc

limit 20;
```



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

select
 \*
from
 matches
where
 to\_char(date,'ddmmyy') = '020513';



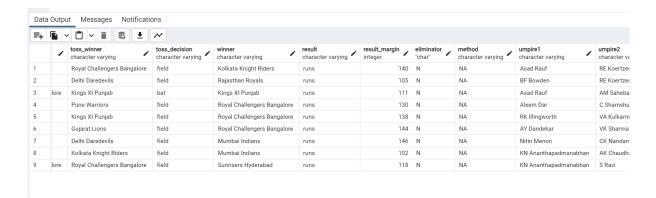
5. Fetch data of all the matches where the result mode is 'runs' and margin of victory is more than 100 runs.

select

\*
from

matches
where

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



6. 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

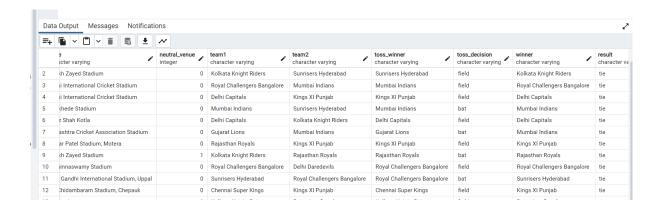
matches

where

result = 'tie'

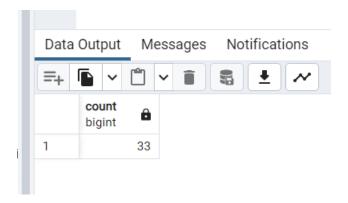
order by

date desc;

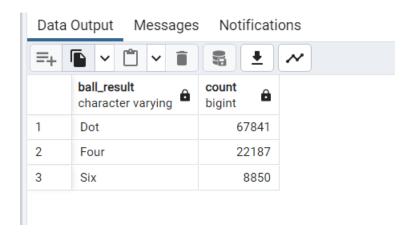


7. Get the count of cities that have hosted an IPL match.

select
count(distinct City)
from
matches;



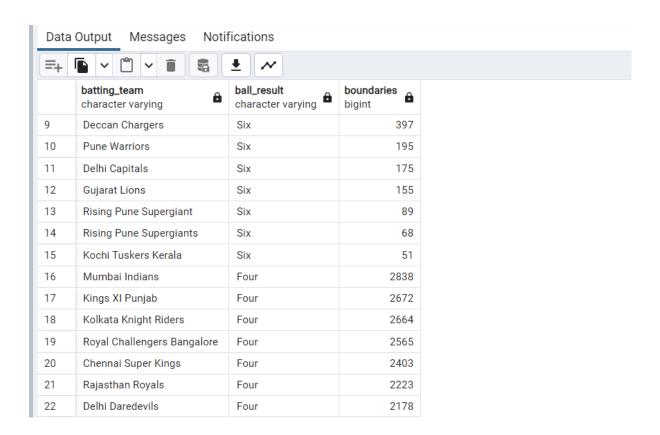
8. Write a query to fetch the total number of boundaries and dot balls from the *deliveries\_v02* table.



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.

```
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;
```



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.

```
select

bowling_Team,ball_result,count(*) as Dots

from

deliveries_v02

where

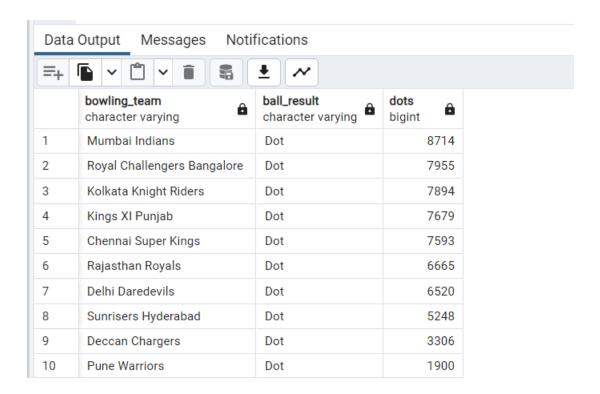
ball_result in ('Dot')

group by

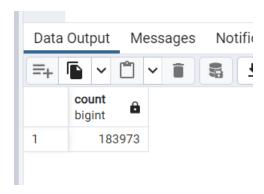
bowling_Team,ball_result

order by

Dots desc;
```



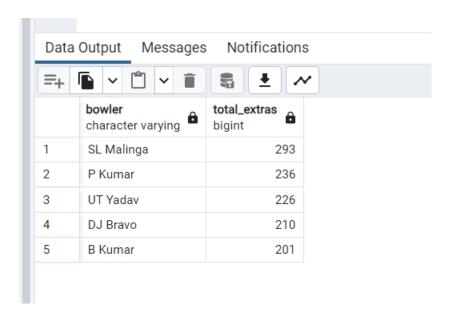
## 11. Write a query to fetch the total number of dismissals by dismissal kinds where dismissal kind is not NA



## 12. 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;



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* 

create table deliveries\_v03 as select \* from deliveries\_v02 where 1 = 2

alter table deliveries\_v03 add column venue character varrying;

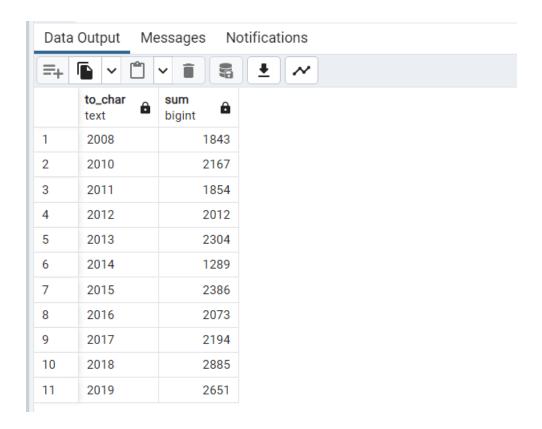
alter table deliveries\_v03 add column Match\_date Date;

insert into deliveries\_v03 select d.\*,m.venue,m.date from deliveries\_v02 as d inner join matches m on m.id = d.id; 14. 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;
```

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.

```
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');
```



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 Supergiants*. 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 Supergiants* 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';
```

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)

select \*, concat(id,'-',inning,'-',over,'-',ball)
from deliveries\_v03;

# 18. 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;

