

```
/**
```

```
STEPS:
```

- 1.CREATE DATABASE AND USE IT
- 2.IMPORT BOTH CSV FILE
- 3.ANALYSING DATA
- 4.WORING ON DIFFERENT QUESTIONS***/

```
--SETUP DATABASE
```

```
CREATE DATABASE IPL_DATABASE_PROJECT;
```

```
USE IPL_DATABASE_PROJECT;
```

```
/******SOME DATA ANALYSIS*****/
```

```
--SEEING IMPORTED TABLES
```

```
SELECT * FROM IPL_BALL;
```

```
SELECT * FROM IPL_MATCHES;
```

```
--NUMBER OF ROWS IN TABLES
```

```
SELECT COUNT(*) AS NO_OF_ROWS  
FROM IPL_MATCHES;
```

```
SELECT COUNT(*) AS NO_OF_ROWS  
FROM IPL_BALL;
```

```
--NUMBER OF COLUMNS IN TABLES
```

```
SELECT COUNT(*) as NO_OF_COLUMNS FROM information_schema.columns WHERE  
table_name = 'IPL_MATCHES';
```

```
SELECT COUNT(*) as NO_OF_COLUMNS FROM information_schema.columns WHERE  
table_name = 'IPL_BALL';
```

--ANALYSIS QUESTION

--Q.1) How many players have won player of the match award at least once

```
SELECT COUNT(DISTINCT(player_of_match)) AS PLAYER_OF_MATCH_COUNT FROM  
IPL_MATCHES;
```

--Q.2) Get details of top 5 matches which were won by maximum number of runs/
result_margin.

--FIRST UPDATE NULL VALUES OF THIS COLUMN

```
UPDATE IPL_MATCHES  
SET result_margin='0'  
WHERE result_margin IS NULL;
```

--CHECKING TOP 5 MATCHES

```
SELECT TOP 5 * FROM IPL_MATCHES  
ORDER BY result_margin DESC
```

--Q.3) Order the rows by city in which the match was played

```
SELECT * FROM IPL_MATCHES  
ORDER BY city;
```

--Q.4) Find venue of 10 most recently played matches

```
SELECT TOP 10 venue FROM IPL_MATCHES  
ORDER BY date DESC;
```

--Q.5) Return a column with comment based on total_runs

```
SELECT *,  
CASE  
WHEN total_runs=0 THEN 'DOT'  
WHEN total_runs=1 THEN 'SINGLE'  
WHEN total_runs=2 THEN 'DOUBLE'  
WHEN total_runs=3 THEN 'THREE'  
WHEN total_runs=4 THEN 'FOUR'  
WHEN total_runs=5 THEN 'FIVE'  
WHEN total_runs=6 THEN 'SIX'  
END AS RUN_COMMENT  
FROM IPL_BALL;
```

--Q.6) Create table deliveries_v02 with all the columns of deliveries and an additional column ball_result containing value boundary, dot or other depending on the total_run (boundary for >= 4, dot for 0 and other for any other number)

--FIRST INSERT NEW COLUMN IN IPL_BALL DATASET

```
ALTER TABLE IPL_BALL
ADD ball_result_label VARCHAR(50);
```

--UPDATE NEW COLUMN WITH VALUES

```
UPDATE IPL_BALL
SET    ball_result_label=
      CASE total_runs
        WHEN 0 THEN 'DOT'
        WHEN 1 THEN 'SINGLE'
        WHEN 2 THEN 'DOUBLE'
        WHEN 3 THEN 'THREE'
        WHEN 4 THEN 'FOUR'
        WHEN 5 THEN 'FIVE'
        WHEN 6 THEN 'SIX'
      END
WHERE total_runs IN (0,1,2,3,4,5,6);
```

--MAKE NEW TABLE deliveries_v02 FROM OLD TABLE IPL_BALL

```
SELECT * INTO deliveries_v02
FROM IPL_BALL;
```

--CHECKING TABLE IS CREATED OR NOT

```
SELECT *FROM deliveries_v02;
```

--Q.7) Write a query to fetch the total number of boundaries and dot balls.

```
SELECT ball_result_label,COUNT(*) AS COUNT_
FROM deliveries_v02
WHERE ball_result_label='SIX' OR ball_result_label='DOT' OR
      ball_result_label='FOUR'
GROUP BY ball_result_label
```

--Q.8) What is the highest runs by which any team won a match.

```
SELECT MAX(result_margin) AS WINNER_RUN_MARGIN FROM IPL_MATCHES;
```

--Q.9) On an average, teams won by how many runs in ipl.

```
SELECT ROUND(AVG(result_margin),2) AS AVERAGE_RUN_MARGIN FROM IPL_MATCHES;
```

--Q.10) How many extra runs were conceded in ipl by SK Warne

```
SELECT SUM(extra_runs) AS RUNS_CONCEDED FROM IPL_BALL
WHERE bowler='SK Warne';
```

--Q.11) How many boundaries (4s) and (6s) have been hit in ipl

```
SELECT total_runs,COUNT(total_runs) AS COUNT_
FROM IPL_BALL
WHERE total_runs=4 OR total_runs=6
GROUP BY total_runs;
```

--Q.12) How many balls did SK Warne bowl to batsman SR Tendulkar.

```
SELECT COUNT(*) AS COUNT_OF_BALL
FROM IPL_BALL
WHERE batsman='SR Tendulkar' AND bowler='SK Warne';
```

--Q.11) How many matches were played in the March and June

```
SELECT COUNT(*) AS COUNT_OF_MATCHES FROM IPL_MATCHES
WHERE MONTH(date)=03 OR MONTH(date)=06;
```

--Q.12) Add column season in IPL_MATCHES dataset and update this column taking year of match from date column. [↗](#)

--ADDING COLUMN

```
ALTER TABLE IPL_MATCHES
ADD season int;
```

--UPDATE COLUMN WITH YEAR

```
UPDATE IPL_MATCHES
SET season =YEAR(date)
```

--Q.13) Name the players who won player of match only once in this period.

```
SELECT player_of_match, COUNT(player_of_match) FROM IPL_MATCHES
GROUP BY player_of_match
HAVING COUNT(player_of_match)=1
```

--Q.14) Fetch data of all the matches played on 2nd May 2013

```
SELECT *FROM IPL_MATCHES
WHERE date= '2013-05-02';
```

--Q.15) Fetch data of all the matches where the margin of victory is more than 100 runs [↗](#)

```
SELECT *FROM IPL_MATCHES
WHERE result_margin>100;
```

--Q.16) 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 IPL_MATCHES
WHERE result='tie'
ORDER BY date DESC;
```

--Q.17) Get the count of cities that have hosted an IPL match

```
SELECT COUNT(DISTINCT city) AS COUNT_OF_CITIES
FROM IPL_MATCHES;
```

--Q.18) Write a query to fetch the total number of dismissals by dismissal kinds [↗](#)

```
SELECT dismissal_kind, COUNT(dismissal_kind)AS DISMISSAL_COUNT
FROM DELIVERIES
WHERE dismissal_kind != 'NA'
GROUP BY dismissal_kind
ORDER BY DISMISSAL_COUNT DESC;
```

--Q.19) Write a query to get the top 5 bowlers who conceded maximum extra runs

```
SELECT TOP 5 bowler, COUNT(extra_runs) AS COUNT_OF_EXTRA_RUNS
FROM IPL_BALL
GROUP BY bowler
ORDER BY COUNT_OF_EXTRA_RUNS DESC;
```

--Q.20) Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored.

```
SELECT M.venue, COUNT(B.total_runs) AS TOTAL_RUNS
FROM IPL_MATCHES AS M
JOIN
IPL_BALL AS B
ON M.id=B.id
GROUP BY M.venue
ORDER BY TOTAL_RUNS DESC;
```

--Q.21) 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 M.venue, YEAR(M.date) AS YEAR_OF_MATCH, COUNT(B.total_runs) AS TOTAL_RUNS
FROM IPL_MATCHES AS M
JOIN
IPL_BALL AS B
ON M.id=B.id
WHERE M.venue='Eden Gardens'
GROUP BY M.venue, YEAR(M.date)
ORDER BY TOTAL_RUNS DESC;
```

--Q.22) Create table deliveries_v03 with all columns of deliveries_v02 and an additional column for row number partition over id.

--CREATING VIEW AND ADD COLUMN AND SAME TIME USED PARTITION BY

```
WITH CTE AS
(
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY id ORDER BY id ) AS r_nums
    FROM deliveries_v02
)
UPDATE CTE
SET r_num = r_nums;
```

--CREATING TABLE FROM OLD TABLE

```
SELECT * INTO deliveries_v03
FROM deliveries_v02;
```

--Q.23) Use the r_num created in deliveries_v03 to identify instances where id is repeating .

```
SELECT *
FROM deliveries_v03
WHERE r_num = 2
```

--Q.24) Use subqueries to fetch data of all the ball_id which are repeating.

```
SELECT *
FROM deliveries_v03
WHERE id in ( SELECT id
              FROM deliveries_v03
              WHERE r_num=2 )
ORDER BY id
```