**Write queries for the following tasks:**

1. Create a table named ‘matches’ with appropriate data types for columns \*
2. Create a table named ‘deliveries’ with appropriate data types for columns \*
3. Import data from csv file ’IPL\_matches.csv’ attached in resources to the table ‘matches’ which was created in Q1 \*
4. Import data from csv file ’IPL\_Ball.csv’ attached in resources to the table ‘deliveries’ which was created in Q2 \*
5. Select the top 20 rows of the *deliveries*table after ordering them by id, inning, over, ball in ascending order. \*
6. Select the top 20 rows of the *matches*table. \*
7. Fetch data of all the matches played on 2nd May 2013 from the *matches*table.. \*
8. Fetch data of all the matches where the result mode is ‘runs’ and margin of victory is more than 100 runs. \*
9. Fetch data of all the matches where the final scores of both teams tied and order it in descending order of the date. \*
10. Get the count of cities that have hosted an IPL match. \*
11. 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 select statement into a table, you can use a subquery. Create table *table\_name*as *[entire select statement].*
12. Write a query to fetch the total number of boundaries and dot balls from the *deliveries\_v02*table.
13. 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.
14. 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.
15. Write a query to fetch the total number of dismissals by dismissal kinds where dismissal kind is not NA
16. Write a query to get the top 5 bowlers who conceded maximum extra runs from the *deliveries*table
17. 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*
18. Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored.
19. 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.
20. 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.
21. 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)
22. Compare the total count of rows and total count of distinct ball\_id in deliveries\_v04;
23. SQL Row\_Number() function is used to sort and assign row numbers to data rows in the presence of multiple groups. For example, to identify the top 10 rows which have the highest order amount in each region, we can use row\_number to assign row numbers in each group (region) with any particular order (decreasing order of order amount) and then we can use this new column to apply filters. Using this knowledge, solve the following exercise. You can use hints to create an additional column of row number.  
    Create table deliveries\_v05 with all columns of deliveries\_v04 and an additional column for row number partition over ball\_id. (HINT : Syntax to add along with other columns,  row\_number() over (partition by ball\_id) as r\_num)
24. Use the r\_num created in deliveries\_v05 to identify instances where ball\_id is repeating. (HINT : select \* from deliveries\_v05 WHERE r\_num=2;)
25. Use subqueries to fetch data of all the ball\_id which are repeating. (HINT: SELECT \* FROM deliveries\_v05 WHERE ball\_id in (select BALL\_ID from deliveries\_v05 WHERE r\_num=2);