SQL – II PROJECT

IPL DATASET

Q1

Show the percentage of wins of each bidder in the order of highest to lowest percentage.

```sql
SELECT bidder_id, bidder_name,
IFNULL(Win_count, 0) AS Win_Count,
IFNULL(Total_count, 0) AS Total_Count,
IFNULL(Win_Percentage, 0) AS Win_Percentage FROM ipl_bidder_details
LEFT JOIN
(Select *, ROUND((Win_count/Total_count*100), 2) AS Win_Percentage FROM
(SELECT Bidder_id, COUNT(*) AS Win_count FROM ipl_bidding_details
WHERE bid_status = "Won"
GROUP BY bidder_id) AS t1
JOIN
(SELECT bidder_id, COUNT(*) AS Total_count FROM ipl_bidding_details
GROUP BY bidder_id) AS t2
USING(bidder_id)) AS t3
USING(bidder_id)
ORDER BY Win_Percentage DESC;
```

Insights:
Bidder, Megaduta Dheer has highest win percentage i.e., 100%. There are total 30 bidders in that 3 bidders have not bided even for a single match that is approximately 10% of the total bidders. From this we can infer that in a set of bidders there are at least 10% people who just enroll but don't bid in matches.

Q2

Display the number of matches conducted at each stadium with the stadium name and city.

```sql
SELECT stadium_id, stadium_name, city, No_of_matches_played FROM
(SELECT Stadium_id, COUNT(match_id) AS No_of_matches_played FROM ipl_match_schedule
GROUP BY stadium_id) AS t1
JOIN ipl_stadium
USING(stadium_id);
```

Insights:
By the output we can say that top matches that is 18 matches have been played in Wankhede Stadium which is in Mumbai and the least matches have been played in Hyderabad and Pune that is 7 matches each.

Q3

In a given stadium, what is the percentage of wins by a team that has won the toss?

```sql
SELECT *, ROUND(Won_toss_and_match/No_of_matches_played * 100, 2) AS Percentage_of_won FROM
(SELECT stadium_id, stadium_name, city, No_of_matches_played FROM
(SELECT Stadium_id, COUNT(match_id) AS No_of_matches_played FROM ipl_match_schedule
GROUP BY stadium_id) AS t1
JOIN ipl_stadium
USING(stadium_id)) AS t3
JOIN
(SELECT stadium_id, COUNT(*) AS Won_toss_and_match FROM
(SELECT match_id FROM ipl_match
WHERE toss_winner = match_winner) AS t2
JOIN ipl_match_schedule
USING(match_id)
GROUP BY stadium_id) AS t4
USING(stadium_id);
```

Insights:
From the data we can tell that the top win percentage is 70% and that percentage has been brought by stadium Swame Mansingh Stadium which is located in Jaipur. In that stadium 10 matches have been played. In those 10 matches 7 matches have been won by the team and the same team has won the toss in that respective matches too. And the least we can tell that is 14.29% that is in Rajogan International Stadium which is located in Hyderabad. In that 7 matches have been played and those 7 matches only one match have been won by same team which won the match and the toss.

Q4

Show the total bids along with the bid team and team name.

```sql
SELECT Team_name, bid_team, Total_bids FROM
(SELECT bid_team, COUNT(bidder_id) as Total_bids FROM ipl_bidding_details
GROUP BY bid_team) AS t1
JOIN ipl_team
ON ipl_team.team_id = t1.bid_team;
```

Insights:
From the data we can tell that highest number of bids have been placed on Sunrisers Hyderabad team 32 bids have been placed on that team and the least has been placed on 3 teams i.e. Chennai Super Kings, Kolkata Knight Riders and Mumbai Indians On each team 22 bids have been placed From this data we can infer that people are pretty much confident enough on Sunrisers Hyderabad that the team may win or lose and people are very doubtful about the winning or losing on CSK, Kolkata Knight Riders and Mumbai Indians

Q5

Show the team ID who won the match as per the win details.

```
SELECT team_id, team_name, win_details, match_id FROM
(SELECT match_id, IF(match_winner = 1, team_id1, team_id2) AS team_id, win_details FROM
ipl_match) AS t1
JOIN ipl_team
USING(team_id);
```

Insights:

from the output given by the above query we can tell that if in windytales if it is mentioned that team won by 7 wickets then we can tell that that respective team has opted bowling first and later they did batting suppose if the team has like team won by 35 runs then we can tell that that respective team has opted batting first and later on they have did bowling and they won the match

Q6

Display the total matches played, total matches won and total matches lost by the team along with its team name.

```
CREATE VIEW v2 AS
(SELECT team_id1 FROM ipl_match
UNION ALL
SELECT team_id2 FROM ipl_match);

SELECT * FROM v2;

SELECT *, (Total_matches_palyed - Total_matches_won) AS Total_matches_lost FROM
(WITH t3 AS
(SELECT *, COUNT(*) AS Total_matches_won FROM
(SELECT IF(match_winner = 1, team_id1, team_id2) AS team_id FROM ipl_match) AS t2
GROUP BY team_id)

SELECT team_id,team_name,
COUNT(team_id1) AS Total_matches_palyed,
Total_matches_won FROM ipl_team
JOIN v2
ON v2.team_id1 = ipl_team.team_id
JOIN t3 USING(team_id)
GROUP BY team_id) AS t4;
```

Insights:

Highest number of matches has been played by Mumbai Indians and Rajasthan Royals and the least number of matches in this tournament has been played by Kolkata Knight Riders and the highest number of matches won is by Rajasthan Royals which is 20 and the least number of matches which has been won is by Delhi Daredevils, Kolkata Knight Riders and Royal Challengers Bangalore apart from these things one interesting finding is that Mumbai Indians which has played the highest matches in this tournament 32 but it has lost 17 matches in this tournament and won only 15 of the matches which have been played by Mumbai Indians

Q7

Display the bowlers for the Mumbai Indians team.

```sql
SELECT player_id, player_name, player_role FROM ipl_team_players
JOIN ipl_player USING(player_id)
WHERE team_id = 5 AND player_role = "Bowler";
```

Insights:

There are total 9 bowlers in Mumbai Indians team.

Q8

How many all-rounders are there in each team, Display the teams with more than 4 all-rounders in descending order.

```sql
SELECT team_id, team_name, Count_of_allrounders FROM ipl_team
JOIN
(SELECT team_id, COUNT(*) AS Count_of_allrounders FROM ipl_team_players
WHERE player_role = "All-Rounder"
GROUP BY team_id
HAVING Count_of_allrounders > 4
ORDER BY Count_of_allrounders DESC) AS T1
USING(team_id);
```

Insights:

By executing the above query, we get the result, and the result tells that there are total 5 teams which have more than 4 all-rounders, and those teams are Delhi Daredevils, Kings XI Punjab, KKR, Rajasthan Royals, and Sunrisers. In these teams, there are 7 all-rounders in 2 teams i.e. Delhi Daredevils and Kings XI Punjab.

Q9

Write a query to get the total bidders' points for each bidding status of those bidders who bid on CSK when they won the match in M. Chinnaswamy Stadium bidding year-wise.

```sql
SELECT MIN(YEAR(bid_date)), bid_status, SUM(total_points) FROM ipl_match
JOIN ipl_match_schedule USING(match_id)
JOIN ipl_stadium USING(stadium_id)
JOIN ipl_bidding_details USING(schedule_id)
JOIN ipl_bidder_points USING(bidder_id)
WHERE (team_id1 = 1 OR team_id2 = 1) AND match_winner = 1 AND stadium_name = "M. Chinnaswamy Stadium"
GROUP BY bid_status
ORDER BY SUM(total_points) DESC;
```

Insights:

By executing the above query we get the results which tells us that in 2017 the total points which has been scored by people who have bidded on CSK are 32 and who has just bidded scored 8 points.

Q10

Extract the Bowlers and All-Rounders that are in the 5 highest number of wickets.

```sql
SELECT player_id, CONVERT(wickets_took, DECIMAL) AS wickets_took, player_role FROM ipl_team_players
JOIN
(SELECT player_id, wickets_took FROM
((SELECT *, SUBSTRING(performance_dtls, p1, p2-p1) AS Wickets_took FROM
(SELECT Player_id, performance_dtls, POSITION("wkt" IN performance_dtls) + 4 AS p1 FROM ipl_player) AS t1
JOIN
(SELECT Player_id, POSITION(" Dot" IN performance_dtls) p2 FROM ipl_player) AS t2
USING(player_id))) AS t3) AS t4
USING(player_id)
WHERE player_role = "bowler" OR player_role = "all-rounder"
ORDER BY player_role, wickets_took DESC;
```

Insights:

By executing the above query, we can get the details of top 5 wicket takers in all-rounder and bowler category, in that one player is from out of India and remaining 4 players are from India.

Q11

Show the percentage of toss wins of each bidder and display the results in descending order based on the percentage

```sql
CREATE VIEW v1 AS
SELECT bidder_id, bid_team, IF(toss_winner = 1, team_id1, team_id2) AS toss_won_team FROM ipl_bidder_details
JOIN ipl_bidding_details USING(bidder_id)
JOIN ipl_match_schedule USING(schedule_id)
JOIN ipl_match USING(match_id);

SELECT * FROM v1;

SELECT *, ROUND(Count_toss_won/Count_total * 100, 2) AS Percentage  FROM
(SELECT bidder_id, COUNT(*) AS Count_toss_won FROM v1
WHERE bid_team = toss_won_team
GROUP BY bidder_id
ORDER BY bidder_id) AS t1
JOIN
(SELECT bidder_id, COUNT(*) AS Count_total FROM v1
GROUP BY bidder_id
ORDER BY bidder_id) AS t2
USING(bidder_id)
ORDER BY percentage DESC;
```

Insights:

the above query will return 29 rows and there are no specific insights which we can notice from the result

Q12

Find the IPL season which has a duration and max duration.

```sql
SELECT tournmt_id, tournmt_name, DATEDIFF(to_date, from_date) AS Duration FROM ipl_tournament
GROUP BY tournmt_id
ORDER BY duration DESC;
```

Insights:

From the above query we can tell that highest number of days has been played in 2012 and 2013 and least number of days has been played in 2009. From this output we can say that 2008, 9 and 10 these are the least top 3 matches as per the duration

Q13

Write a query to display to calculate the total points month-wise for the 2017 bid year. sort the results based on total points in descending order and month-wise in ascending order.

```sql
SELECT bidder_id, bidder_name, MIN(YEAR(bid_date)) AS bid_year, MONTH(bid_date) AS bid_month, SUM(Total_points) AS total_points FROM ipl_bidder_details
JOIN ipl_bidding_details USING(bidder_id)
JOIN ipl_bidder_points USING(bidder_id)
WHERE YEAR(bid_date) = 2017
GROUP BY bidder_id,bidder_name, bid_month
ORDER BY total_points DESC, bid_month;
```

Insights:

By running the above query, we can conclude that Aryabhatta Parichuri has topped the list consecutively for 2 months i.e., 4th month and 5th month. By this, we can tell that he has a good amount of experience in this field. In the list, total points has been scored by Ronaldo de Sousa and Ganga Pandey and Krishna Valimbi. Their total points is zero. There may be two reasons for this score. Either they have not bidded at all or else they have bidded but lost the bid.

Q14

Write a query for the above question using sub-queries by having the same constraints as the above question.

Insights:

Same as above question.

Q15

Write a query to get the top 3 and bottom 3 bidders based on the total bidding points for the 2018 bidding year.

```
SELECT * FROM
(SELECT bidder_id, bidder_name, total_points,
ROW_NUMBER() OVER(ORDER BY total_points) AS Least_Ranking,
ROW_NUMBER() OVER(ORDER BY total_points DESC) AS Highest_Ranking FROM ipl_bidder_points
JOIN ipl_bidder_details USING(bidder_id)) As t1
WHERE Least_Ranking <= 3
UNION
SELECT * FROM
(SELECT bidder_id, bidder_name, total_points,
ROW_NUMBER() OVER(ORDER BY total_points) AS Least_Ranking,
ROW_NUMBER() OVER(ORDER BY total_points DESC) AS Highest_Ranking FROM ipl_bidder_points
JOIN ipl_bidder_details USING(bidder_id)) As t1
WHERE Highest_Ranking <= 3;
```

Insights:

By running the above query, we get the result, it fetches 6 records and there are no any meaningful insights which we can draw from the above result.

Q16

You need not insert the records separately into both tables rather Create a trigger in such a way that it should insert the details into the student back table when you insert the student details into the student table automatically.

```
CREATE TABLE student_details
(Student_id INT,
Student_name VARCHAR(50),
mail_id VARCHAR(50),
mobile_no INT);

ALTER TABLE student_details
MODIFY mobile_no BIGINT;

CREATE TABLE student_details_backup
(Student_id INT,
Student_name VARCHAR(50),
mail_id VARCHAR(50),
mobile_no INT);

ALTER TABLE student_details_backup
MODIFY mobile_no BIGINT;

INSERT INTO student_details
VALUES(152, "Dileep", "Dileep@student.com", 9676150614);

SELECT * FROM student_details;

DELETE FROM student_details_backup;

DROP TRIGGER insert_student_backup4;

DELIMITER $$
CREATE TRIGGER insert_student_backup4
AFTER UPDATE ON Student_details
```

```sql
FOR EACH ROW
BEGIN
    INSERT INTO student_details_backup
      VALUES (OLD.student_id, OLD.student_name, OLD.mail_id, OLD.mobile_no);
END$$
DELIMITER ;


UPDATE student_details
SET mobile_no = 9676150611
WHERE student_id = 152;

SELECT * FROM student_details;
SELECT * FROM student_details_backup;
```