Assignment 3 Solution

1. SELECT DISTINCT city, venue

FROM matches

WHERE team1 = 'Royal Challengers Bangalore' OR team2 = 'Royal Challengers Bangalore';

2. WITH batsman\_stats AS (

SELECT batsman,

COUNT(DISTINCT match\_id) AS total\_innings,

SUM(batsman\_runs) AS total\_runs,

ROUND(CAST(SUM(batsman\_runs) AS FLOAT) / COUNT(DISTINCT match\_id), 2) AS batting\_average

FROM deliveries

GROUP BY batsman

)

SELECT batsman, total\_innings, total\_runs, batting\_average

FROM batsman\_stats

ORDER BY total\_runs DESC

LIMIT 1;

3. WITH centuries AS (

SELECT batsman, COUNT(\*) AS hundreds

FROM deliveries

WHERE batsman\_runs >= 100

GROUP BY batsman

),

fifties AS (

SELECT batsman, COUNT(\*) AS fifties

FROM deliveries

WHERE batsman\_runs >= 50 AND batsman\_runs < 100

GROUP BY batsman

)

SELECT c.batsman, f.fifties, c.hundreds

FROM centuries c

LEFT JOIN fifties f ON c.batsman = f.batsman

ORDER BY c.hundreds DESC, f.fifties DESC, c.batsman

LIMIT 10;

4. SELECT season, winner

FROM matches

WHERE id IN (

SELECT MAX(id)

FROM matches

GROUP BY season

)

ORDER BY season;

5. WITH bowling\_figures AS (

SELECT match\_id, bowler,

SUM(CASE WHEN dismissal\_kind IS NOT NULL THEN 1 ELSE 0 END) AS wickets,

SUM(total\_runs) AS runs\_conceded

FROM deliveries

GROUP BY match\_id, bowler

)

SELECT m.season, m.city, m.venue, b.bowler,

CONCAT(b.wickets, '-', b.runs\_conceded) AS best\_figures

FROM bowling\_figures b

JOIN matches m ON b.match\_id = m.id

ORDER BY b.wickets DESC, b.runs\_conceded ASC

LIMIT 1;

6. WITH season\_runs AS (

SELECT m.season, d.batsman, d.batting\_team, SUM(d.batsman\_runs) AS total\_runs

FROM deliveries d

JOIN matches m ON d.match\_id = m.id

GROUP BY m.season, d.batsman, d.batting\_team

)

SELECT season, batsman, batting\_team AS team, total\_runs

FROM season\_runs

WHERE (season, total\_runs) IN (

SELECT season, MAX(total\_runs)

FROM season\_runs

GROUP BY season

)

ORDER BY season;

7. SELECT season, winner,

CASE

WHEN win\_by\_runs > 0 THEN CONCAT(win\_by\_runs, ' Runs')

WHEN win\_by\_wickets > 0 THEN CONCAT(win\_by\_wickets, ' Wickets')

END AS winning\_type

FROM matches

WHERE (season, GREATEST(win\_by\_runs, win\_by\_wickets \* 10)) IN (

SELECT season, MAX(GREATEST(win\_by\_runs, win\_by\_wickets \* 10))

FROM matches

GROUP BY season

)

ORDER BY season;

8. WITH match\_counts AS (

SELECT team1 AS team, COUNT(\*) AS total\_matches

FROM matches

GROUP BY team1

UNION ALL

SELECT team2 AS team, COUNT(\*) AS total\_matches

FROM matches

GROUP BY team2

),

wins AS (

SELECT winner AS team, COUNT(\*) AS wins

FROM matches

GROUP BY winner

)

SELECT m.team AS team\_name,

SUM(m.total\_matches) AS total\_matches,

COALESCE(w.wins, 0) AS won,

SUM(m.total\_matches) - COALESCE(w.wins, 0) AS lost,

ROUND(COALESCE(w.wins::float, 0) / SUM(m.total\_matches) \* 100, 2) AS win\_ratio

FROM match\_counts m

LEFT JOIN wins w ON m.team = w.team

GROUP BY m.team, w.wins

ORDER BY win\_ratio DESC;

9. WITH hat\_tricks AS (

SELECT match\_id, bowler, COUNT(\*) AS hat\_trick

FROM (

SELECT match\_id, bowler,

LAG(player\_dismissed, 1) OVER (PARTITION BY match\_id, bowler ORDER BY over\_number, ball) AS prev1,

LAG(player\_dismissed, 2) OVER (PARTITION BY match\_id, bowler ORDER BY over\_number, ball) AS prev2,

player\_dismissed

FROM deliveries

WHERE player\_dismissed IS NOT NULL

) sub

WHERE player\_dismissed IS NOT NULL AND prev1 IS NOT NULL AND prev2 IS NOT NULL

GROUP BY match\_id, bowler

HAVING COUNT(\*) >= 3

)

SELECT m.season, h.match\_id AS id, h.bowler, m.team2 AS opponent, m.venue

FROM hat\_tricks h

JOIN matches m ON h.match\_id = m.id

ORDER BY m.season, h.match\_id;