SELECT

bowler,

COUNT(\*) AS total\_wickets

FROM all\_ipl\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler

ORDER BY total\_wickets DESC

LIMIT 10;

SELECT

bowler,

COUNT(\*) AS total\_wickets

FROM cricket\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler

ORDER BY total\_wickets DESC

LIMIT 10;

WITH innings\_totals AS (

SELECT

match\_id,

inning, -- ✅ correct column name here

SUM(total\_runs) AS total\_score

FROM all\_ipl\_deliveries\_summary

GROUP BY match\_id, inning

)

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

m.result

FROM all\_ipl\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_ipl\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

m.result,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE NULL -- in case of tie/no result

END AS winner

FROM all\_ipl\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2

),

team\_stats AS (

SELECT team AS team\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN team = winner THEN 1 ELSE 0 END) AS wins

FROM (

SELECT team1 AS team, winner FROM match\_results

UNION ALL

SELECT team2 AS team, winner FROM match\_results

) all\_matches

GROUP BY team

)

SELECT

team\_name,

matches\_played,

wins,

ROUND((wins \* 100.0 / matches\_played), 2) AS win\_percentage

FROM team\_stats

ORDER BY win\_percentage DESC;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_ipl\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE 'Tie'

END AS winner

FROM all\_ipl\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2

)

SELECT

team AS team\_name,

opponent AS opponent\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) AS wins,

SUM(CASE WHEN winner = opponent THEN 1 ELSE 0 END) AS losses,

ROUND(100.0 \* SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) / COUNT(\*), 2) AS win\_percentage

FROM (

SELECT team1 AS team, team2 AS opponent, winner FROM match\_results

UNION ALL

SELECT team2 AS team, team1 AS opponent, winner FROM match\_results

) t

GROUP BY team, opponent

ORDER BY team\_name, win\_percentage DESC;

SELECT

bowler,

batsman,

COUNT(\*) AS wickets

FROM all\_ipl\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler, batsman

ORDER BY wickets DESC

LIMIT 20;

SELECT

bowler,

batsman,

SUM(batsman\_runs) AS runs\_conceded,

COUNT(\*) AS balls\_bowled,

ROUND(SUM(batsman\_runs)\*6.0/COUNT(\*), 2) AS economy\_rate -- runs per over

FROM all\_ipl\_deliveries\_summary

GROUP BY bowler, batsman

ORDER BY runs\_conceded DESC

LIMIT 20;

SELECT

batsman,

bowler,

SUM(batsman\_runs) AS runs\_scored,

COUNT(\*) AS balls\_faced,

ROUND(SUM(batsman\_runs)\*100.0/COUNT(\*), 2) AS strike\_rate

FROM all\_ipl\_deliveries\_summary

GROUP BY batsman, bowler

ORDER BY runs\_scored DESC

LIMIT 20;

SELECT

bowler,

batsman,

COUNT(CASE WHEN LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped') THEN 1 END) AS wickets,

SUM(batsman\_runs) AS runs\_conceded,

COUNT(\*) AS balls\_bowled,

ROUND(SUM(batsman\_runs) \* 6.0 / COUNT(\*), 2) AS economy\_rate -- runs per over

FROM all\_ipl\_deliveries\_summary

GROUP BY bowler, batsman

ORDER BY wickets DESC, runs\_conceded ASC

LIMIT 30;

SELECT

batsman,

bowler,

SUM(batsman\_runs) AS runs\_scored,

COUNT(\*) AS balls\_faced,

ROUND(SUM(batsman\_runs) \* 100.0 / COUNT(\*), 2) AS strike\_rate

FROM all\_ipl\_deliveries\_summary

GROUP BY batsman, bowler

ORDER BY runs\_scored DESC, strike\_rate DESC

LIMIT 30;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_ipl\_deliveries\_summary

GROUP BY match\_id, inning

),

scores AS (

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs

FROM all\_ipl\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2

)

SELECT

match\_id,

team1,

team2,

team1\_runs,

team2\_runs,

CASE

WHEN team1\_runs > team2\_runs THEN team1

WHEN team2\_runs > team1\_runs THEN team2

ELSE 'Draw'

END AS winner

FROM scores;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_ipl\_deliveries\_summary

GROUP BY match\_id, inning

),

scores AS (

SELECT

m.match\_id,

m.team1,

m.team2,

m.toss\_winner,

m.toss\_decision,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE 'Draw'

END AS winner,

-- Determine batting order

CASE

WHEN m.toss\_decision = 'bat' THEN m.toss\_winner

ELSE CASE

WHEN m.toss\_winner = m.team1 THEN m.team2

ELSE m.team1

END

END AS team\_batting\_first,

CASE

WHEN m.toss\_decision = 'field' THEN m.toss\_winner

ELSE CASE

WHEN m.toss\_winner = m.team1 THEN m.team2

ELSE m.team1

END

END AS team\_batting\_second

FROM all\_ipl\_match\_summary m

JOIN innings\_totals i1 ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2 ON m.match\_id = i2.match\_id AND i2.inning = 2

),

results AS (

SELECT

team\_batting\_first AS team,

1 AS matches\_batting\_first,

CASE WHEN winner = team\_batting\_first THEN 1 ELSE 0 END AS wins\_batting\_first,

0 AS matches\_batting\_second,

0 AS wins\_batting\_second

FROM scores

UNION ALL

SELECT

team\_batting\_second AS team,

0 AS matches\_batting\_first,

0 AS wins\_batting\_first,

1 AS matches\_batting\_second,

CASE WHEN winner = team\_batting\_second THEN 1 ELSE 0 END AS wins\_batting\_second

FROM scores

)

SELECT

team,

SUM(matches\_batting\_first) AS total\_matches\_batting\_first,

SUM(wins\_batting\_first) AS wins\_batting\_first,

ROUND(100.0 \* SUM(wins\_batting\_first) / NULLIF(SUM(matches\_batting\_first),0), 2) AS win\_percentage\_batting\_first,

SUM(matches\_batting\_second) AS total\_matches\_batting\_second,

SUM(wins\_batting\_second) AS wins\_batting\_second,

ROUND(100.0 \* SUM(wins\_batting\_second) / NULLIF(SUM(matches\_batting\_second),0), 2) AS win\_percentage\_batting\_second

FROM results

GROUP BY team

ORDER BY team;

SELECT

bowler,

COUNT(\*) AS balls\_bowled,

SUM(extras\_runs) AS total\_extras\_conceded

FROM all\_ipl\_deliveries\_summary

GROUP BY bowler

ORDER BY total\_extras\_conceded DESC

LIMIT 5;

SELECT \* from all\_oneday\_match\_summary;

SELECT \* from all\_oneday\_deliveries\_summary;

SELECT

bowler,

COUNT(\*) AS total\_wickets

FROM all\_oneday\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler

ORDER BY total\_wickets DESC

LIMIT 10;

SELECT

batsman,

SUM(batsman\_runs) AS total\_runs

FROM all\_oneday\_deliveries\_summary

GROUP BY batsman

ORDER BY total\_runs DESC

LIMIT 10;

WITH innings\_totals AS (

SELECT

match\_id,

inning, -- ✅ correct column name here

SUM(total\_runs) AS total\_score

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

)

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

m.result

FROM all\_oneday\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

m.result,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE NULL -- in case of tie/no result

END AS winner

FROM all\_oneday\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2

),

team\_stats AS (

SELECT team AS team\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN team = winner THEN 1 ELSE 0 END) AS wins

FROM (

SELECT team1 AS team, winner FROM match\_results

UNION ALL

SELECT team2 AS team, winner FROM match\_results

) all\_matches

GROUP BY team

)

SELECT

team\_name,

matches\_played,

wins,

ROUND((wins \* 100.0 / matches\_played), 2) AS win\_percentage

FROM team\_stats

ORDER BY win\_percentage DESC;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE 'Tie'

END AS winner

FROM all\_oneday\_match\_summary m

JOIN innings\_totals i1

ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2

ON m.match\_id = i2.match\_id AND i2.inning = 2

)

SELECT

team AS team\_name,

opponent AS opponent\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) AS wins,

SUM(CASE WHEN winner = opponent THEN 1 ELSE 0 END) AS losses,

ROUND(100.0 \* SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) / COUNT(\*), 2) AS win\_percentage

FROM (

SELECT team1 AS team, team2 AS opponent, winner FROM match\_results

UNION ALL

SELECT team2 AS team, team1 AS opponent, winner FROM match\_results

) t

GROUP BY team, opponent

ORDER BY team\_name, win\_percentage DESC;

WITH innings\_totals AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

scores AS (

SELECT

m.match\_id,

m.team1,

m.team2,

m.toss\_winner,

m.toss\_decision,

i1.total\_score AS team1\_runs,

i2.total\_score AS team2\_runs,

CASE

WHEN i1.total\_score > i2.total\_score THEN m.team1

WHEN i2.total\_score > i1.total\_score THEN m.team2

ELSE 'Draw'

END AS winner,

-- Determine batting order

CASE

WHEN m.toss\_decision = 'bat' THEN m.toss\_winner

ELSE CASE

WHEN m.toss\_winner = m.team1 THEN m.team2

ELSE m.team1

END

END AS team\_batting\_first,

CASE

WHEN m.toss\_decision = 'field' THEN m.toss\_winner

ELSE CASE

WHEN m.toss\_winner = m.team1 THEN m.team2

ELSE m.team1

END

END AS team\_batting\_second

FROM all\_oneday\_match\_summary m

JOIN innings\_totals i1 ON m.match\_id = i1.match\_id AND i1.inning = 1

JOIN innings\_totals i2 ON m.match\_id = i2.match\_id AND i2.inning = 2

),

results AS (

SELECT

team\_batting\_first AS team,

1 AS matches\_batting\_first,

CASE WHEN winner = team\_batting\_first THEN 1 ELSE 0 END AS wins\_batting\_first,

0 AS matches\_batting\_second,

0 AS wins\_batting\_second

FROM scores

UNION ALL

SELECT

team\_batting\_second AS team,

0 AS matches\_batting\_first,

0 AS wins\_batting\_first,

1 AS matches\_batting\_second,

CASE WHEN winner = team\_batting\_second THEN 1 ELSE 0 END AS wins\_batting\_second

FROM scores

)

SELECT

team,

SUM(matches\_batting\_first) AS total\_matches\_batting\_first,

SUM(wins\_batting\_first) AS wins\_batting\_first,

ROUND(100.0 \* SUM(wins\_batting\_first) / NULLIF(SUM(matches\_batting\_first),0), 2) AS win\_percentage\_batting\_first,

SUM(matches\_batting\_second) AS total\_matches\_batting\_second,

SUM(wins\_batting\_second) AS wins\_batting\_second,

ROUND(100.0 \* SUM(wins\_batting\_second) / NULLIF(SUM(matches\_batting\_second),0), 2) AS win\_percentage\_batting\_second

FROM results

GROUP BY team

ORDER BY team;

SELECT

bowler,

COUNT(\*) AS balls\_bowled,

SUM(extras\_runs) AS total\_extras\_conceded

FROM all\_oneday\_deliveries\_summary

GROUP BY bowler

ORDER BY total\_extras\_conceded DESC

LIMIT 5;

SELECT

bowler,

batsman,

COUNT(\*) AS wickets

FROM all\_oneday\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler, batsman

ORDER BY wickets DESC

LIMIT 20;

SELECT

bowler,

batsman,

SUM(batsman\_runs) AS runs\_conceded,

COUNT(\*) AS balls\_bowled,

ROUND(SUM(batsman\_runs)\*6.0/COUNT(\*), 2) AS economy\_rate -- runs per over

FROM all\_oneday\_deliveries\_summary

GROUP BY bowler, batsman

ORDER BY runs\_conceded DESC

LIMIT 20;

SELECT

batsman,

bowler,

SUM(batsman\_runs) AS runs\_scored,

COUNT(\*) AS balls\_faced,

ROUND(SUM(batsman\_runs)\*100.0/COUNT(\*), 2) AS strike\_rate

FROM all\_oneday\_deliveries\_summary

GROUP BY batsman, bowler

ORDER BY runs\_scored DESC

LIMIT 20;

SELECT

bowler,

batsman,

COUNT(CASE WHEN LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped') THEN 1 END) AS wickets,

SUM(batsman\_runs) AS runs\_conceded,

COUNT(\*) AS balls\_bowled,

ROUND(SUM(batsman\_runs) \* 6.0 / COUNT(\*), 2) AS economy\_rate -- runs per over

FROM all\_oneday\_deliveries\_summary

GROUP BY bowler, batsman

ORDER BY wickets DESC, runs\_conceded ASC

LIMIT 30;

SELECT

batsman,

bowler,

SUM(batsman\_runs) AS runs\_scored,

COUNT(\*) AS balls\_faced,

ROUND(SUM(batsman\_runs) \* 100.0 / COUNT(\*), 2) AS strike\_rate

FROM all\_oneday\_deliveries\_summary

GROUP BY batsman, bowler

ORDER BY runs\_scored DESC, strike\_rate DESC

LIMIT 30;

WITH batsman\_scores AS (

SELECT

match\_id,

batsman,

SUM(batsman\_runs) AS runs\_scored,

MAX(CASE WHEN dismissal\_kind IS NOT NULL THEN 1 ELSE 0 END) AS is\_out

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, batsman

)

SELECT

match\_id,

batsman,

runs\_scored,

CASE

WHEN runs\_scored = 0 AND is\_out = 1 THEN 'Out for 0'

WHEN runs\_scored >= 100 THEN 'Century'

WHEN runs\_scored >= 50 THEN 'Half-Century'

ELSE 'Other'

END AS performance

FROM batsman\_scores

WHERE runs\_scored = 0

OR runs\_scored >= 50

ORDER BY match\_id, runs\_scored DESC;

WITH batsman\_scores AS (

SELECT

match\_id,

batsman,

SUM(batsman\_runs) AS runs\_scored,

MAX(CASE WHEN dismissal\_kind IS NOT NULL THEN 1 ELSE 0 END) AS is\_out

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, batsman

),

batsman\_performance AS (

SELECT

batsman,

CASE

WHEN runs\_scored = 0 AND is\_out = 1 THEN 'Duck'

WHEN runs\_scored >= 100 THEN 'Century'

WHEN runs\_scored >= 50 THEN 'Half-Century'

ELSE 'Other'

END AS performance

FROM batsman\_scores

)

SELECT

batsman,

SUM(CASE WHEN performance = 'Duck' THEN 1 ELSE 0 END) AS total\_ducks,

SUM(CASE WHEN performance = 'Half-Century' THEN 1 ELSE 0 END) AS total\_half\_centuries,

SUM(CASE WHEN performance = 'Century' THEN 1 ELSE 0 END) AS total\_centuries

FROM batsman\_performance

GROUP BY batsman

ORDER BY total\_centuries DESC, total\_half\_centuries DESC;

WITH batsman\_scores AS (

SELECT

match\_id,

batsman,

SUM(batsman\_runs) AS runs\_scored,

MAX(CASE WHEN dismissal\_kind IS NOT NULL THEN 1 ELSE 0 END) AS is\_out

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, batsman

),

batsman\_performance AS (

SELECT

batsman,

match\_id,

runs\_scored,

CASE

WHEN runs\_scored = 0 AND is\_out = 1 THEN 'Duck'

WHEN runs\_scored >= 100 THEN 'Century'

WHEN runs\_scored >= 50 THEN 'Half-Century'

ELSE 'Other'

END AS performance

FROM batsman\_scores

)

SELECT

batsman,

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

SUM(CASE WHEN performance = 'Duck' THEN 1 ELSE 0 END) AS total\_ducks,

SUM(CASE WHEN performance = 'Half-Century' THEN 1 ELSE 0 END) AS total\_half\_centuries,

SUM(CASE WHEN performance = 'Century' THEN 1 ELSE 0 END) AS total\_centuries

FROM batsman\_performance

GROUP BY batsman

ORDER BY total\_centuries DESC, total\_half\_centuries DESC, total\_ducks ASC;

use Cric\_Analysis;

CREATE TABLE all\_test\_match\_summary (

match\_id INT PRIMARY KEY NOT NULL,

match\_date DATE,

venue VARCHAR(350),

team1 VARCHAR(150),

team2 VARCHAR(150),

toss\_winner VARCHAR(150),

toss\_decision VARCHAR(30),

result VARCHAR(200),

player\_of\_match VARCHAR(200)

);

CREATE TABLE all\_testmatch\_deliveries\_summary (

match\_id INTEGER,

inning INTEGER,

match\_over INTEGER,

ball INTEGER,

batsman TEXT,

bowler TEXT,

non\_striker TEXT,

batsman\_runs INTEGER,

extras\_runs INTEGER,

total\_runs INTEGER,

player\_dismissed TEXT,

dismissal\_kind TEXT,

fielder TEXT,

FOREIGN KEY (match\_id) REFERENCES all\_test\_match\_summary(match\_id));

SELECT \* from all\_test\_match\_summary;

SELECT \* from all\_testmatch\_deliveries\_summary;

SELECT

player\_of\_match,

COUNT(\*) AS total\_awards

FROM

all\_ipl\_match\_summary

WHERE

player\_of\_match IS NOT NULL

GROUP BY

player\_of\_match

ORDER BY

total\_awards DESC;

SELECT

player\_of\_match,

COUNT(\*) AS total\_awards

FROM

all\_oneday\_match\_summary

WHERE

player\_of\_match IS NOT NULL

GROUP BY

player\_of\_match

ORDER BY

total\_awards DESC;

WITH team\_scores AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score,

SUM(extras\_runs) AS extras

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

t1.match\_id,

t1.total\_score AS team1\_score,

t2.total\_score AS team2\_score,

t1.extras AS team1\_extras,

t2.extras AS team2\_extras,

CASE

WHEN t1.total\_score > t2.total\_score THEN 'Team1'

WHEN t2.total\_score > t1.total\_score THEN 'Team2'

END AS winner,

ABS(t1.total\_score - t2.total\_score) AS winning\_margin

FROM team\_scores t1

JOIN team\_scores t2

ON t1.match\_id = t2.match\_id

AND t1.inning = 1

AND t2.inning = 2

)

SELECT \*

FROM match\_results

WHERE

(winner = 'Team1' AND team2\_extras > winning\_margin)

OR

(winner = 'Team2' AND team1\_extras > winning\_margin);

WITH team\_scores AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score,

SUM(extras\_runs) AS extras

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

s.match\_id,

m.team1,

m.team2,

t1.total\_score AS team1\_score,

t2.total\_score AS team2\_score,

t1.extras AS team1\_extras,

t2.extras AS team2\_extras,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team1

WHEN t2.total\_score > t1.total\_score THEN m.team2

END AS winner,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team2

WHEN t2.total\_score > t1.total\_score THEN m.team1

END AS loser,

ABS(t1.total\_score - t2.total\_score) AS winning\_margin

FROM team\_scores t1

JOIN team\_scores t2

ON t1.match\_id = t2.match\_id

AND t1.inning = 1

AND t2.inning = 2

JOIN ipl\_match\_summary m

ON t1.match\_id = m.match\_id

JOIN all\_oneday\_deliveries\_summary s

ON t1.match\_id = s.match\_id

GROUP BY s.match\_id, m.team1, m.team2, t1.total\_score, t2.total\_score, t1.extras, t2.extras

)

SELECT

match\_id,

winner,

loser,

winning\_margin,

CASE WHEN winner = team1 THEN team2\_extras ELSE team1\_extras END AS extras\_conceded\_by\_loser

FROM all\_oneday\_deliveries\_summary

WHERE extras\_conceded\_by\_loser > winning\_margin

ORDER BY extras\_conceded\_by\_loser - winning\_margin DESC;

WITH team\_scores AS (

SELECT

match\_id,

inning,

SUM(total\_runs) AS total\_score,

SUM(extras\_runs) AS extras

FROM all\_oneday\_deliveries\_summary

GROUP BY match\_id, inning

),

match\_results AS (

SELECT

m.match\_id,

m.match\_date,

m.venue,

m.team1,

m.team2,

t1.total\_score AS team1\_score,

t2.total\_score AS team2\_score,

t1.extras AS team1\_extras,

t2.extras AS team2\_extras,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team1

WHEN t2.total\_score > t1.total\_score THEN m.team2

END AS winner,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team2

WHEN t2.total\_score > t1.total\_score THEN m.team1

END AS loser,

ABS(t1.total\_score - t2.total\_score) AS winning\_margin

FROM team\_scores t1

JOIN team\_scores t2

ON t1.match\_id = t2.match\_id

AND t1.inning = 1

AND t2.inning = 2

JOIN ipl\_match\_summary m

ON t1.match\_id = m.match\_id

)

SELECT

match\_id,

match\_date,

venue,

winner,

loser,

winning\_margin,

CASE

WHEN winner = team1 THEN team2\_extras

ELSE team1\_extras

END AS extras\_conceded\_by\_loser

FROM match\_results

WHERE extras\_conceded\_by\_loser > winning\_margin

ORDER BY extras\_conceded\_by\_loser - winning\_margin DESC;

SELECT

batsman,

SUM(batsman\_runs) AS total\_runs

FROM all\_testmatch\_deliveries\_summary

GROUP BY batsman

ORDER BY total\_runs DESC

LIMIT 10;

SELECT

bowler,

COUNT(\*) AS total\_wickets

FROM all\_testmatch\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY bowler

ORDER BY total\_wickets DESC

LIMIT 10;

WITH innings\_totals AS (

SELECT

match\_id,

CASE

WHEN inning IN (1, 3) THEN 'Team1'

WHEN inning IN (2, 4) THEN 'Team2'

END AS team\_side,

SUM(total\_runs) AS total\_score

FROM all\_testmatch\_deliveries\_summary

GROUP BY match\_id, team\_side

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

t1.total\_score AS team1\_runs,

t2.total\_score AS team2\_runs,

m.result,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team1

WHEN t2.total\_score > t1.total\_score THEN m.team2

ELSE NULL -- draw/tie/no result

END AS winner

FROM innings\_totals t1

JOIN innings\_totals t2

ON t1.match\_id = t2.match\_id

AND t1.team\_side = 'Team1'

AND t2.team\_side = 'Team2'

JOIN all\_test\_match\_summary m

ON m.match\_id = t1.match\_id

),

team\_stats AS (

SELECT team AS team\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN team = winner THEN 1 ELSE 0 END) AS wins

FROM (

SELECT team1 AS team, winner FROM match\_results

UNION ALL

SELECT team2 AS team, winner FROM match\_results

) all\_matches

GROUP BY team

)

SELECT

team\_name,

matches\_played,

wins,

ROUND((wins \* 100.0 / matches\_played), 2) AS win\_percentage

FROM team\_stats

ORDER BY win\_percentage DESC;

WITH innings\_totals AS (

SELECT

match\_id,

CASE

WHEN inning IN (1, 3) THEN 'Team1'

WHEN inning IN (2, 4) THEN 'Team2'

END AS team\_side,

SUM(total\_runs) AS total\_score

FROM all\_testmatch\_deliveries\_summary

GROUP BY match\_id, team\_side

),

match\_results AS (

SELECT

m.match\_id,

m.team1,

m.team2,

t1.total\_score AS team1\_runs,

t2.total\_score AS team2\_runs,

CASE

WHEN t1.total\_score > t2.total\_score THEN m.team1

WHEN t2.total\_score > t1.total\_score THEN m.team2

ELSE 'Tie'

END AS winner

FROM innings\_totals t1

JOIN innings\_totals t2

ON t1.match\_id = t2.match\_id

AND t1.team\_side = 'Team1'

AND t2.team\_side = 'Team2'

JOIN all\_test\_match\_summary m

ON m.match\_id = t1.match\_id

)

SELECT

team AS team\_name,

opponent AS opponent\_name,

COUNT(\*) AS matches\_played,

SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) AS wins,

SUM(CASE WHEN winner = opponent THEN 1 ELSE 0 END) AS losses,

ROUND(100.0 \* SUM(CASE WHEN winner = team THEN 1 ELSE 0 END) / COUNT(\*), 2) AS win\_percentage

FROM (

SELECT team1 AS team, team2 AS opponent, winner FROM match\_results

UNION ALL

SELECT team2 AS team, team1 AS opponent, winner FROM match\_results

) t

GROUP BY team, opponent

ORDER BY team\_name, win\_percentage DESC;

WITH innings\_scores AS (

SELECT

match\_id,

inning,

batsman,

SUM(batsman\_runs) AS runs\_scored

FROM all\_testmatch\_deliveries\_summary

GROUP BY match\_id, inning, batsman

),

classified AS (

SELECT

batsman,

CASE

WHEN runs\_scored = 0 THEN 'duck'

WHEN runs\_scored BETWEEN 50 AND 99 THEN 'fifty'

WHEN runs\_scored >= 100 THEN 'hundred'

ELSE 'other'

END AS score\_type

FROM innings\_scores

)

SELECT

batsman,

SUM(CASE WHEN score\_type = 'duck' THEN 1 ELSE 0 END) AS ducks,

SUM(CASE WHEN score\_type = 'fifty' THEN 1 ELSE 0 END) AS fifties,

SUM(CASE WHEN score\_type = 'hundred' THEN 1 ELSE 0 END) AS hundreds

FROM classified

GROUP BY batsman

ORDER BY hundreds DESC, fifties DESC

LIMIT 15;

WITH innings\_scores AS (

SELECT

match\_id,

inning,

batsman,

SUM(batsman\_runs) AS runs\_scored

FROM all\_testmatch\_deliveries\_summary

GROUP BY match\_id, inning, batsman

),

classified AS (

SELECT

batsman,

match\_id,

CASE

WHEN runs\_scored = 0 THEN 'duck'

WHEN runs\_scored BETWEEN 50 AND 99 THEN 'fifty'

WHEN runs\_scored >= 100 THEN 'hundred'

ELSE 'other'

END AS score\_type

FROM innings\_scores

)

SELECT

batsman,

COUNT(DISTINCT match\_id) AS matches\_played,

SUM(CASE WHEN score\_type = 'duck' THEN 1 ELSE 0 END) AS ducks,

SUM(CASE WHEN score\_type = 'fifty' THEN 1 ELSE 0 END) AS fifties,

SUM(CASE WHEN score\_type = 'hundred' THEN 1 ELSE 0 END) AS hundreds

FROM classified

GROUP BY batsman

ORDER BY hundreds DESC, fifties DESC

LIMIT 15;

WITH wickets\_per\_innings AS (

SELECT

match\_id,

inning,

bowler,

COUNT(player\_dismissed) AS wickets

FROM all\_testmatch\_deliveries\_summary

WHERE LOWER(dismissal\_kind) IN ('caught', 'lbw', 'caught and bowled', 'bowled', 'stumped')

GROUP BY match\_id, inning, bowler

),

five\_wicket\_list AS (

SELECT

bowler,

COUNT(\*) AS five\_wicket\_hauls

FROM wickets\_per\_innings

WHERE wickets >= 5

GROUP BY bowler

),

ten\_wicket\_list AS (

SELECT

bowler,

COUNT(\*) AS ten\_wicket\_matches

FROM (

SELECT

match\_id,

bowler,

SUM(wickets) AS match\_wickets

FROM wickets\_per\_innings

GROUP BY match\_id, bowler

HAVING SUM(wickets) >= 10

) t

GROUP BY bowler

),

matches\_played AS (

SELECT

bowler,

COUNT(DISTINCT match\_id) AS matches\_played

FROM all\_testmatch\_deliveries\_summary

GROUP BY bowler

)

SELECT

m.bowler,

m.matches\_played,

COALESCE(f.five\_wicket\_hauls, 0) AS five\_wicket\_hauls,

COALESCE(t.ten\_wicket\_matches, 0) AS ten\_wicket\_matches

FROM matches\_played m

LEFT JOIN five\_wicket\_list f ON m.bowler = f.bowler

LEFT JOIN ten\_wicket\_list t ON m.bowler = t.bowler

ORDER BY ten\_wicket\_matches DESC, five\_wicket\_hauls DESC

LIMIT 15;