

Cricket Data Analysis with SQL

Database Selection

```
USE cricket_analysis;
```

- Select the cricket analysis database.

Match Tables

```
SELECT * FROM odi_matches;  
SELECT * FROM odi_deliveries;  
SELECT * FROM test_matches;  
SELECT * FROM test_deliveries;  
SELECT * FROM ipl_matches;  
SELECT * FROM ipl_deliveries;  
SELECT * FROM t20_matches;  
SELECT * FROM t20_deliveries;
```

- Display match and delivery records.

Match Count by Format

```
SELECT 'Test' AS format, COUNT(*) AS total_matches FROM test_matches  
UNION  
SELECT 'ODI', COUNT(*) FROM odi_matches  
UNION  
SELECT 'T20', COUNT(*) FROM t20_matches  
UNION  
SELECT 'IPL', COUNT(*) FROM ipl_matches;
```

- Count total matches across formats.

Top Batsmen

```
-- IPL  
SELECT batsman, SUM(runs_batter) AS total_runs  
FROM ipl_deliveries  
GROUP BY batsman
```

```

ORDER BY total_runs DESC
LIMIT 10;

-- ODI
SELECT batsman, SUM(runs_batter) AS total_runs
FROM odi_deliveries
GROUP BY batsman
ORDER BY total_runs DESC
LIMIT 10;

-- Test
SELECT batsman, SUM(runs_batter) AS total_runs
FROM test_deliveries
GROUP BY batsman
ORDER BY total_runs DESC
LIMIT 10;

-- T20
SELECT batsman, SUM(runs_batter) AS total_runs
FROM t20_deliveries
GROUP BY batsman
ORDER BY total_runs DESC
LIMIT 10;

```

- Get top 10 batsmen in each format.

IPL Strike Rates

```

SELECT batsman, SUM(runs_batter) AS runs,
       (SUM(runs_batter) * 100 / COUNT(*)) AS strike_rate
FROM ipl_deliveries
GROUP BY batsman
HAVING COUNT(*) >= 500
ORDER BY strike_rate DESC
LIMIT 10;

```

- Top strike rate batsmen (min 500 balls).

Top Wicket Takers

```

-- Test
SELECT bowler, COUNT(*) AS wickets
FROM test_deliveries WHERE is_wicket = 1
GROUP BY bowler

```

```

ORDER BY wickets DESC
LIMIT 10;

-- IPL
SELECT bowler, COUNT(*) AS wickets
FROM ipl_deliveries WHERE is_wicket = 1
GROUP BY bowler
ORDER BY wickets DESC
LIMIT 10;

-- ODI
SELECT bowler, COUNT(*) AS wickets
FROM odi_deliveries WHERE is_wicket = 1
GROUP BY bowler
ORDER BY wickets DESC
LIMIT 10;

-- T20
SELECT bowler, COUNT(*) AS wickets
FROM t20_deliveries WHERE is_wicket = 1
GROUP BY bowler
ORDER BY wickets DESC
LIMIT 10;

```

- Top 10 bowlers by wickets.

Bowling Economy

```

-- IPL
SELECT bowler, SUM(runs_total) AS runs_conceded, COUNT(*)/6 AS overs_bowled,
       SUM(runs_total) / (COUNT(*) / 6) AS economy
FROM ipl_deliveries
GROUP BY bowler
HAVING COUNT(*) >= 1200
ORDER BY economy ASC
LIMIT 10;

-- ODI
SELECT bowler, SUM(runs_total) AS runs_conceded, COUNT(*)/6 AS overs_bowled,
       SUM(runs_total) / (COUNT(*) / 6) AS economy
FROM odi_deliveries
GROUP BY bowler
HAVING COUNT(*) >= 1200
ORDER BY economy ASC
LIMIT 10;

```

```

-- T20
SELECT bowler, SUM(runs_total) AS runs_conceded, COUNT(*)/6 AS overs_bowled,
        SUM(runs_total) / (COUNT(*) / 6) AS economy
FROM t20_deliveries
GROUP BY bowler
HAVING COUNT(*) >= 1200
ORDER BY economy ASC
LIMIT 10;

-- Test
SELECT bowler, SUM(runs_total) AS runs_conceded, COUNT(*)/6 AS overs_bowled,
        SUM(runs_total) / (COUNT(*) / 6) AS economy
FROM test_deliveries
GROUP BY bowler
HAVING COUNT(*) >= 1200
ORDER BY economy ASC
LIMIT 10;

```

- Top economical bowlers (≥ 1200 balls).

Team Wins

```

-- IPL
SELECT match_winner, COUNT(*) AS wins
FROM ipl_matches
GROUP BY match_winner
ORDER BY wins DESC
LIMIT 10;

-- ODI
SELECT match_winner, COUNT(*) AS wins
FROM odi_matches
GROUP BY match_winner
ORDER BY wins DESC
LIMIT 10;

-- T20
SELECT match_winner, COUNT(*) AS wins
FROM t20_matches
GROUP BY match_winner
ORDER BY wins DESC
LIMIT 10;

-- Test
SELECT match_winner, COUNT(*) AS wins
FROM test_matches

```

```
WHERE match_winner IS NOT NULL
GROUP BY match_winner
ORDER BY wins DESC
LIMIT 10;
```

- Teams with most wins in each format.

Average Win Margin - ODI

```
SELECT AVG(win_by_runs) AS avg_margin
FROM odi_matches
WHERE win_by_runs > 0;
```

- Average margin of runs in ODIs.

Toss Decisions - IPL

```
SELECT toss_decision, COUNT(*) AS count
FROM ipl_matches
GROUP BY toss_decision;
```

- Toss choice stats (bat/field).

Toss vs Match Win - IPL

```
SELECT SUM(CASE WHEN match_winner = toss_winner THEN 1 ELSE 0 END) * 100 /
COUNT(*) AS toss_with_match_win
FROM ipl_matches;
```

- % of matches where toss winner also won.

Player of the Match Awards - IPL

```
SELECT player_of_match, COUNT(*) AS awards
FROM ipl_matches
GROUP BY player_of_match
ORDER BY awards DESC
LIMIT 10;
```

- Top players with most awards.

Chasing vs Defending Wins - IPL

```
SELECT SUM(CASE WHEN win_by_wickets > 0 THEN 1 ELSE 0 END) AS chasing_wins,  
       SUM(CASE WHEN win_by_runs > 0 THEN 1 ELSE 0 END) AS defending_wins  
FROM ipl_matches;
```

- Comparison of chasing vs defending wins.

Conclusion

- SQL queries highlight batting, bowling, team strategies, and outcomes across cricket formats.