sql queries for ai cricket project

1. Total Matches by Season

```
SELECT season, COUNT(*) AS total_matches
FROM matches
GROUP BY season
ORDER BY season DESC;
```

2. Most Successful Teams by Wins

```
SELECT o.winner, COUNT(*) AS total_wins
FROM outcome o
GROUP BY o.winner
ORDER BY total_wins DESC
LIMIT 10;
```

3. Average Runs per Match per Venue

```
SELECT m.venue, ROUND(AVG(b.runs_off_bat + b.extras), 2) AS avg_runs
FROM ball_by_ball b
JOIN matches m ON b.match_id = m.match_id
GROUP BY m.venue
ORDER BY avg_runs DESC
LIMIT 10;
```

4. Highest Total Runs in a Match

```
SELECT b.match_id, b.batting_team, SUM(b.runs_off_bat + b.extras) AS total_runs
FROM ball_by_ball b
GROUP BY b.match_id, b.batting_team
ORDER BY total_runs DESC
LIMIT 5;
```

5. Top 10 Batsmen by Total Runs

```
SELECT striker AS batsman, SUM(runs_off_bat) AS total_runs
FROM ball_by_ball
GROUP BY striker
ORDER BY total_runs DESC
LIMIT 10;
```

6. Top 10 Bowlers by Wickets

```
SELECT bowler, COUNT(*) AS total_wickets
FROM ball_by_ball
WHERE wicket_type IS NOT NULL
GROUP BY bowler
ORDER BY total_wickets DESC
LIMIT 10;
```

7. Fielders with Most Catches

```
SELECT fielder_name, COUNT(*) AS catches
FROM ball_fielders
WHERE involvement_type = 'caught'
GROUP BY fielder_name
ORDER BY catches DESC
LIMIT 10;
```

8. Matches Played at Each Venue by Season

```
SELECT season, venue, COUNT(DISTINCT match_id) AS total_matches
FROM matches
GROUP BY season, venue
ORDER BY season DESC, total_matches DESC;
```

9. Top 10 Most Common Wicket Types

```
SELECT wicket_type, COUNT(*) AS occurrences
FROM ball_by_ball
WHERE wicket_type IS NOT NULL
GROUP BY wicket_type
ORDER BY occurrences DESC;
```

10. Total Extras by Team

```
SELECT batting_team, SUM(extras) AS total_extras
FROM ball_by_ball
GROUP BY batting_team
ORDER BY total_extras DESC
LIMIT 10;
```

11. Top 10 Matches with Most Sixes

```
SELECT match_id, COUNT(*) AS sixes
FROM ball_by_ball
WHERE runs_off_bat = 6
GROUP BY match_id
```

```
ORDER BY sixes DESC
LIMIT 10;
```

12. Best Batting Partnerships (Striker–Non-striker)

```
SELECT striker, non_striker, SUM(runs_off_bat) AS partnership_runs
FROM ball_by_ball
GROUP BY striker, non_striker
ORDER BY partnership_runs DESC
LIMIT 10;
```

13. Bowler Economy Rates

```
SELECT bowler,
      ROUND(SUM(runs_off_bat + extras) / (COUNT(ball) / 6.0), 2) AS
economy_rate
FROM ball_by_ball
GROUP BY bowler
HAVING COUNT(ball) > 30
ORDER BY economy_rate ASC
LIMIT 10;
```

14. Number of Overs per Match

```
SELECT match_id, MAX(CAST(SUBSTR(ball, INSTR(ball, '.') + 1) AS INTEGER))
AS max_ball
FROM ball_by_ball
GROUP BY match_id
ORDER BY max_ball DESC
LIMIT 10;
```

15. Players Who Took Both Wickets and Scored Runs

```
SELECT p.player_name
FROM players p
WHERE p.player_name IN (
   SELECT bowler FROM ball_by_ball WHERE wicket_type IS NOT NULL
)
AND p.player_name IN (
   SELECT striker FROM ball_by_ball WHERE runs_off_bat > 0
)
GROUP BY p.player_name
LIMIT 20;
```

16. Team Winning Percentage

```
WITH total_matches AS (
   SELECT team_name, COUNT(*) AS matches_played
   FROM teams
   GROUP BY team_name
),
total_wins AS (
   SELECT winner AS team_name, COUNT(*) AS matches_won
   FROM outcome
   GROUP BY winner
)
SELECT t.team_name,
      COALESCE(w.matches_won, 0) AS wins,
      t.matches_played,
      ROUND(100.0 * COALESCE(w.matches_won, 0) / t.matches_played, 2) AS
win_percent
FROM total_matches t
LEFT JOIN total_wins w ON t.team_name = w.team_name
ORDER BY win_percent DESC;
```

17. Fielder Involvement by Type

```
SELECT involvement_type, COUNT(*) AS total_involvements
FROM ball_fielders
GROUP BY involvement_type
ORDER BY total_involvements DESC;
```

18. Most Common Dismissal Combination (Bowler + Fielder)

```
SELECT b.bowler, f.fielder_name, COUNT(*) AS dismissals
FROM ball_by_ball b
JOIN ball_fielders f
ON b.match_id = f.match_id AND b.ball = f.ball
WHERE b.wicket_type = 'caught'
GROUP BY b.bowler, f.fielder_name
ORDER BY dismissals DESC
LIMIT 10;
```

19. Top Run Scorers Per Season

```
SELECT m.season, b.striker, SUM(b.runs_off_bat) AS total_runs
FROM ball_by_ball b
JOIN matches m ON b.match_id = m.match_id
GROUP BY m.season, b.striker
ORDER BY m.season, total_runs DESC;
```

20. Average Score by Team When Winning Toss

```
SELECT t.toss_winner, ROUND(AVG(b.runs_off_bat + b.extras), 2) AS avg_score
FROM toss t
JOIN ball_by_ball b ON t.match_id = b.match_id
```

```
GROUP BY t.toss_winner
ORDER BY avg_score DESC;
```

21. Top 10 Matches by Total Wickets

```
SELECT match_id, COUNT(*) AS total_wickets
FROM ball_by_ball
WHERE wicket_type IS NOT NULL
GROUP BY match_id
ORDER BY total_wickets DESC
LIMIT 10;
```

22. Players Dismissed the Most

```
SELECT player_dismissed, COUNT(*) AS dismissals
FROM ball_by_ball
WHERE player_dismissed IS NOT NULL
GROUP BY player_dismissed
ORDER BY dismissals DESC
LIMIT 10;
```

23. Top Venues by Boundaries (4s and 6s)

```
SELECT m.venue,
    SUM(CASE WHEN b.runs_off_bat = 4 THEN 1 ELSE 0 END) AS fours,
    SUM(CASE WHEN b.runs_off_bat = 6 THEN 1 ELSE 0 END) AS sixes,
    COUNT(*) AS total_boundaries
FROM ball_by_ball b
JOIN matches m ON b.match_id = m.match_id
GROUP BY m.venue
ORDER BY total_boundaries DESC
LIMIT 10;
```

24. Longest Innings (Most Balls Faced by a Batsman)

```
SELECT striker, match_id, COUNT(*) AS balls_faced
FROM ball_by_ball
GROUP BY striker, match_id
ORDER BY balls_faced DESC
LIMIT 10;
```

25. Player of the Match Frequency

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

Bonus Tip:

You can run all these in:

DB Browser for SQLite

VS Code SQLite Explorer

Or inside Python:

```
import sqlite3
conn = sqlite3.connect("cricket_matches.db")
pd.read_sql_query("SELECT * FROM ball_by_ball LIMIT 5;", conn)
```