DAX QUERIES

1. Total Centuries :

```
Centuries =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonData =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason
    )

VAR BatterRuns =
    SUMMARIZE(
        SeasonData,
        ball_by_ball_data[match_id],
        ball_by_ball_data[batter],
        "Total Runs", SUM(ball_by_ball_data[batter_runs])
    )

VAR PlayersWithCentury =
    FILTER(
        BatterRuns,
        [Total Runs] >= 100
    )

RETURN
COUNTROWS(PlayersWithCentury)
```

2. Total Half Centuries :

```
Half Centuries =
VAR SlectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonData = FILTER(
    ball_by_ball_data,
    RELATED(ipl_matches_data[season]) = SlectedSeason
)

VAR BattetRuns =
    SUMMARIZE(
        SeasonData,
        ball_by_ball_data[match_id],
        ball_by_ball_data[batter],
        "TotalRuns", SUM(ball_by_ball_data[batter_runs])
    )
```

```
VAR CenturyCount = FILTER(BattetRuns, [TotalRuns] >= 50 && [TotalRuns] < 100 )

RETURN COUNTROWS(CenturyCount)
```

3. IPL Season :

```
IPL Season = SELECTEDVALUE(ipl_matches_data[season])
```

4. Matches Lost :

```
Matches Lost =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR Team1LostMatches =
    CALCULATE(
        COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team1], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        NOT ISBLANK(ipl_matches_data[match_winner]),
        ipl_matches_data[match_winner] <> ipl_matches_data[team1]
    )

VAR Team2LostMatches =
    CALCULATE(
        COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team2], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        NOT ISBLANK(ipl_matches_data[match_winner]),
        ipl_matches_data[match_winner] <> ipl_matches_data[team2]
    )

RETURN Team1LostMatches + Team2LostMatches
```

5. Total Matches Played :

```
Matches Played =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR Team1Matches =
    CALCULATE(
        COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team1], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20"
    )

VAR Team2Matches =
    CALCULATE(
```

```
        COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team2], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20"
    )

RETURN Team1Matches + Team2Matches
```

6. Matches Won :

```
Matches Won =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])
VAR CurrentTeam = SELECTEDVALUE(teams_data[team_name])
RETURN
    CALCULATE(
        COUNTROWS(ipl_matches_data),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_winner] = CurrentTeam,
        ipl_matches_data[match_type] = "T20"
    )
```

7. No Result :

```
No Result Played =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR Team1Matches =
    CALCULATE(COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team1], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        ipl_matches_data[result] = "no result")
VAR Team2Matches =
    CALCULATE(COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team2], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        ipl_matches_data[result] = "no result")

RETURN Team1Matches + Team2Matches
```

8. Orange Cap Holder :

```
Orange Cap Holder =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])
VAR SeasonDataOnly =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason
    )
```

```
VAR RunSummary =
    SUMMARIZE(
        SeasonDataOnly,
        ball_by_ball_data[batter],
        "Total Runs", SUM(ball_by_ball_data[batter_runs])
    )
VAR MaxRuns = MAXX(RunSummary, [Total Runs])
VAR TopScorer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(RunSummary, [Total Runs] = MaxRuns)
    )
RETURN MAXX(TopScorer, ball_by_ball_data[batter])
```

9. Orange Cap Holder Image :

```
Orange Cap Image =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonDataOnly =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason
    )

VAR RunSummary =
    SUMMARIZE(
        SeasonDataOnly,
        ball_by_ball_data[batter],
        "Total Runs", SUM(ball_by_ball_data[batter_runs])
    )

VAR MaxRuns = MAXX(RunSummary, [Total Runs])

VAR TopScorer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(RunSummary, [Total Runs] = MaxRuns)
    )

RETURN
    LOOKUPVALUE(
        'players-data-updated'[player_image],
        'players-data-updated'[player_name],
        MAXX(TopScorer, ball_by_ball_data[batter])
    )
```

10. Orange Cap Holder's Total Runs :

```
Orange Cap Runs =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonDataOnly =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason
    )

VAR RunSummary =
    SUMMARIZE(
        SeasonDataOnly,
        ball_by_ball_data[batter],
        "Total Runs", SUM(ball_by_ball_data[batter_runs])
    )

VAR MaxRuns = MAXX(RunSummary, [Total Runs])

RETURN MaxRuns
```

11. Orange Cap Holder's Team Name :

```
Orange Cap Team Name =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonDataOnly =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason
    )

VAR RunSummary =
    SUMMARIZE(
        SeasonDataOnly,
        ball_by_ball_data[batter],
        "Total Runs", SUM(ball_by_ball_data[batter_runs])
    )

VAR MaxRuns = MAXX(RunSummary, [Total Runs])

VAR TopScorer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(RunSummary, [Total Runs] = MaxRuns)
    )

VAR FullTeamName =
    CALCULATE(
        MAX(ball_by_ball_data[team_batting]),
```

```
        FILTER(
            SeasonDataOnly,
            ball_by_ball_data[batter] = MAXX(TopScorer, ball_by_ball_data[batter])
        )
    )

RETURN
    FullTeamName
```

12. Purple Cap Holder :

```
Purple Cap Holder =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

-- Filter: wickets in selected season, exclude non-bowler dismissals
VAR SeasonWickets =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[is_wicket] = TRUE() &&
        NOT ball_by_ball_data[wicket_kind] IN { "run out", "retired hurt", "obstructing the field",
"retired out" }
    )

-- Summarize bowler and count wickets
VAR WicketSummary =
    SUMMARIZE(
        SeasonWickets,
        ball_by_ball_data[bowler],
        "WicketCount", COUNTROWS(
            FILTER(
                SeasonWickets,
                ball_by_ball_data[bowler] = EARLIER(ball_by_ball_data[bowler])
            )
        )
    )

-- Find highest wicket count
VAR MaxWickets = MAXX(WicketSummary, [WicketCount])

-- Get the bowler(s) with that wicket count
VAR TopBowler =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[bowler]),
        FILTER(WicketSummary, [WicketCount] = MaxWickets)
    )

-- Return the name (if multiple, it picks one alphabetically)
RETURN
    MAXX(TopBowler, ball_by_ball_data[bowler])
```

13. Purple Cap Holder's Image :

```
PurpleCapImage =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

-- Step 1: Filter valid bowler wickets for the selected season
VAR SeasonWickets =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[is_wicket] = TRUE() &&
        NOT ball_by_ball_data[wicket_kind] IN {
            "run out",
            "retired hurt",
            "retired out",
            "obstructing the field",
            "timed out"
        }
    )

-- Step 2: Summarize bowler-wise wicket count
VAR WicketSummary =
    SUMMARIZE(
        SeasonWickets,
        ball_by_ball_data[bowler],
        "WicketCount", COUNTROWS(
            FILTER(
                SeasonWickets,
                ball_by_ball_data[bowler] = EARLIER(ball_by_ball_data[bowler])
            )
        )
    )

-- Step 3: Get the max wicket count
VAR MaxWickets = MAXX(WicketSummary, [WicketCount])

-- Step 4: Get the bowler with max wickets
VAR TopBowler =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[bowler]),
        FILTER(WicketSummary, [WicketCount] = MaxWickets)
    )

-- Step 5: Return image using LOOKUP from player table
RETURN
LOOKUPVALUE(
    'players-data-updated'[player_image],
    'players-data-updated'[player_name], MAXX(TopBowler, ball_by_ball_data[bowler])
)
```

14. Purple Cap Holder's Team :

```
PurpleCapTeam =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

-- Step 1: Filter valid wickets for the selected season
VAR SeasonWickets =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[is_wicket] = TRUE() &&
        NOT ball_by_ball_data[wicket_kind] IN {
            "run out",
            "retired hurt",
            "retired out",
            "obstructing the field"
        }
    )

-- Step 2: Summarize bowler and team, count wickets per bowler
VAR WicketSummary =
    ADDCOLUMNS(
        SUMMARIZE(
            SeasonWickets,
            ball_by_ball_data[bowler],
            ball_by_ball_data[team_bowling]
        ),
        "WicketCount",
        COUNTROWS(
            FILTER(
                SeasonWickets,
                ball_by_ball_data[bowler] = EARLIER(ball_by_ball_data[bowler])
            )
        )
    )

-- Step 3: Max Wickets
VAR MaxWickets = MAXX(WicketSummary, [WicketCount])

-- Step 4: Return full team name of the top wicket-taker
RETURN
MAXX(
    FILTER(WicketSummary, [WicketCount] = MaxWickets),
    ball_by_ball_data[team_bowling]
)
```

15. Purple Cap Holder's Wicket Count :

```
PurpleCapWicketCount =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

-- Step 1: Filter valid bowler wickets in selected season
VAR SeasonWickets =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[is_wicket] = TRUE() &&
        NOT ball_by_ball_data[wicket_kind] IN {
            "run out",
            "retired hurt",
            "retired out",
            "obstructing the field",
            "timed out"
        }
    )
-- Step 2: Summarize wickets per bowler
VAR WicketSummary =
    SUMMARIZE(
        SeasonWickets,
        ball_by_ball_data[bowler],
        "WicketCount", COUNTROWS(
            FILTER(
                SeasonWickets,
                ball_by_ball_data[bowler] = EARLIER(ball_by_ball_data[bowler])
            )
        )
    )

-- Step 3: Get the highest wicket count
VAR MaxWickets = MAXX(WicketSummary, [WicketCount])

RETURN
MaxWickets
```

16. Season Winner :

```
SeasonWinner =
VAR SelectedSeason = SELECTEDVALUE('ipl_matches_data'[season_id])

-- Try to get the final match (explicitly marked)
VAR FinalMatchID_WithStage =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        FILTER(
            'ipl_matches_data',
            'ipl_matches_data'[season_id] = SelectedSeason &&
            TRIM(LOWER('ipl_matches_data'[stage])) = "final"
```

```
        )
    )

-- Fallback: get the last match by match_id for that season
VAR FinalMatchID_Fallback =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        'ipl_matches_data'[season_id] = SelectedSeason
    )

-- Use the "Final" match if it exists; otherwise fallback to the last match
VAR FinalMatchID =
    IF(
        ISBLANK(FinalMatchID_WithStage),
        FinalMatchID_Fallback,
        FinalMatchID_WithStage
    )

-- Now get the winner from that match
VAR FinalMatchWinner =
    CALCULATE(
        MAX('ipl_matches_data'[match_winner]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

RETURN FinalMatchWinner
```

17. Season Runner Up :

```
SeasonRunnerUp =
VAR SelectedSeason = SELECTEDVALUE('ipl_matches_data'[season_id])

-- Try to get the final match (explicitly marked)
VAR FinalMatchID_WithStage =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        FILTER(
            'ipl_matches_data',
            'ipl_matches_data'[season_id] = SelectedSeason &&
            TRIM(LOWER('ipl_matches_data'[stage])) = "final"
        )
    )

-- Fallback: latest match if "Final" isn't marked
VAR FinalMatchID_Fallback =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        'ipl_matches_data'[season_id] = SelectedSeason
    )
```

```
-- Final match ID to use
VAR FinalMatchID =
    IF(
        ISBLANK(FinalMatchID_WithStage),
        FinalMatchID_Fallback,
        FinalMatchID_WithStage
    )

-- Winner of the final
VAR Winner =
    CALCULATE(
        MAX('ipl_matches_data'[match_winner]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Team1 in the final
VAR Team1 =
    CALCULATE(
        MAX('ipl_matches_data'[team1]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Team2 in the final
VAR Team2 =
    CALCULATE(
        MAX('ipl_matches_data'[team2]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Determine runner-up
VAR RunnerUp =
    IF(
        Team1 = Winner,
        Team2,
        Team1
    )

RETURN RunnerUp
```

18. Season Winner Team's Logo :

```
Season Winner Logo =

VAR SelectedSeason = SELECTEDVALUE('ipl_matches_data'[season_id])

-- Try to get the final match (explicitly marked)
VAR FinalMatchID_WithStage =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        FILTER(
```

```
            'ipl_matches_data',
            'ipl_matches_data'[season_id] = SelectedSeason &&
            TRIM(LOWER('ipl_matches_data'[stage])) = "final"
        )
    )

-- Fallback: get the last match by match_id for that season
VAR FinalMatchID_Fallback =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        'ipl_matches_data'[season_id] = SelectedSeason
    )

-- Use the "Final" match if it exists; otherwise fallback to the last match
VAR FinalMatchID =
    IF(
        ISBLANK(FinalMatchID_WithStage),
        FinalMatchID_Fallback,
        FinalMatchID_WithStage
    )

-- Now get the winner from that match
VAR FinalMatchWinner =
    CALCULATE(
        MAX('ipl_matches_data'[match_winner]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

RETURN
LOOKUPVALUE(
    teams_data[image_url],
    teams_data[team_name],FinalMatchWinner)
```

19. Season Runner Up Team's Logo :

```
Season Runner Up Logo =
VAR SelectedSeason = SELECTEDVALUE('ipl_matches_data'[season_id])

-- Try to get the final match (explicitly marked)
VAR FinalMatchID_WithStage =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        FILTER(
            'ipl_matches_data',
            'ipl_matches_data'[season_id] = SelectedSeason &&
            TRIM(LOWER('ipl_matches_data'[stage])) = "final"
        )
    )
```

```
-- Fallback: latest match if "Final" isn't marked
VAR FinalMatchID_Fallback =
    CALCULATE(
        MAX('ipl_matches_data'[match_id]),
        'ipl_matches_data'[season_id] = SelectedSeason
    )

-- Final match ID to use
VAR FinalMatchID =
    IF(
        ISBLANK(FinalMatchID_WithStage),
        FinalMatchID_Fallback,
        FinalMatchID_WithStage
    )

-- Winner of the final
VAR Winner =
    CALCULATE(
        MAX('ipl_matches_data'[match_winner]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Team1 in the final
VAR Team1 =
    CALCULATE(
        MAX('ipl_matches_data'[team1]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Team2 in the final
VAR Team2 =
    CALCULATE(
        MAX('ipl_matches_data'[team2]),
        'ipl_matches_data'[match_id] = FinalMatchID
    )

-- Determine runner-up
VAR RunnerUp =
    IF(
        Team1 = Winner,
        Team2,
        Team1
    )

RETURN
LOOKUPVALUE(teams_data[image_url],teams_data[team_name],RunnerUp)
```

20. Tie Matches :

```
Tie Matches =
VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR Team1Matches =
    CALCULATE(COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team1], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        ipl_matches_data[result] = "tie")

VAR Team2Matches =
    CALCULATE(COUNTROWS(ipl_matches_data),
        USERELATIONSHIP(ipl_matches_data[team2], teams_data[team_name]),
        ipl_matches_data[season] = SelectedSeason,
        ipl_matches_data[match_type] = "T20",
        ipl_matches_data[result] = "tie")

RETURN Team1Matches + Team2Matches
```

21. Top 4's Hitter :

```
Top Fours Count =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonFours =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 4
    )

VAR FourSummary =
    SUMMARIZE(
        SeasonFours,
        ball_by_ball_data[batter],
        "FoursCount",
        COUNTROWS(
            FILTER(
                SeasonFours,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxFours = MAXX(FourSummary, [FoursCount])

RETURN MaxFours
```

22. Top 4's Hitter Image :

```
Top Fours Player Image =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonFours =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 4
    )

VAR FourSummary =
    SUMMARIZE(
        SeasonFours,
        ball_by_ball_data[batter],
        "FoursCount",
        COUNTROWS(
            FILTER(
                SeasonFours,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxFours = MAXX(FourSummary, [FoursCount])

VAR TopFoursPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(FourSummary, [FoursCount] = MaxFours)
    )

RETURN
LOOKUPVALUE(
    'players-data-updated'[player_image],
    'players-data-updated'[player_name], MAXX(TopFoursPlayer, ball_by_ball_data[batter])
)
```

23. Top 4 Hitter's Name :

```
Top Fours Player Name =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonFours =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 4
```

```
    )

VAR FourSummary =
    SUMMARIZE(
        SeasonFours,
        ball_by_ball_data[batter],
        "FoursCount",
        COUNTROWS(
            FILTER(
                SeasonFours,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxFours = MAXX(FourSummary, [FoursCount])

VAR TopFoursPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(FourSummary, [FoursCount] = MaxFours)
    )

RETURN
    MAXX(TopFoursPlayer, ball_by_ball_data[batter])
```

24. Top 4 Hitter's Team Name :

```
Top Fours Player Team Name =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonFours =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 4
    )

VAR FourSummary =
    SUMMARIZE(
        SeasonFours,
        ball_by_ball_data[batter],
        "FoursCount",
        COUNTROWS(
            FILTER(
                SeasonFours,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
```

```
    )

VAR MaxFours = MAXX(FourSummary, [FoursCount])

VAR TopFoursPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(FourSummary, [FoursCount] = MaxFours)
    )

VAR BatterTeam =
    CALCULATE(
        MAX(ball_by_ball_data[team_batting]),
        FILTER(
            SeasonFours,
            ball_by_ball_data[batter] = MAXX(TopFoursPlayer, ball_by_ball_data[batter])
        )
    )

RETURN BatterTeam
```

25. Top 6 Hitter Name :

```
Top Six Player Name =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonSix =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 6
    )

VAR SixSummary =
    SUMMARIZE(
        SeasonSix,
        ball_by_ball_data[batter],
        "SixCount",
        COUNTROWS(
            FILTER(
                SeasonSix,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxSix = MAXX(SixSummary, [SixCount])
```

```
VAR TopSixPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(SixSummary, [SixCount] = MaxSix)
    )

RETURN
    MAXX(TopSixPlayer, ball_by_ball_data[batter])
```

26. Total Sixes :

```
Top Sixes Count =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonSixes =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 6
    )

VAR SixSummary =
    SUMMARIZE(
        SeasonSixes,
        ball_by_ball_data[batter],
        "SixesCount",
        COUNTROWS(
            FILTER(
                SeasonSixes,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxSixes = MAXX(SixSummary, [SixesCount])

RETURN MaxSixes
```

27. Top Six Hitter Image :

```
Top Sixes Player Image =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonSixes =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
```

```
        ball_by_ball_data[batter_runs] = 6
    )

VAR SixSummary =
    SUMMARIZE(
        SeasonSixes,
        ball_by_ball_data[batter],
        "SixesCount",
        COUNTROWS(
            FILTER(
                SeasonSixes,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxSixes = MAXX(SixSummary, [SixesCount])

VAR TopSixesPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(SixSummary, [SixesCount] = MaxSixes)
    )

RETURN
LOOKUPVALUE(
    'players-data-updated'[player_image],
    'players-data-updated'[player_name], MAXX(TopSixesPlayer, ball_by_ball_data[batter])
)
```

28. Top Six Hitter Player Name :

```
Top Sixes Player Team Name =

VAR SelectedSeason = SELECTEDVALUE(ipl_matches_data[season])

VAR SeasonSixes =
    FILTER(
        ball_by_ball_data,
        RELATED(ipl_matches_data[season]) = SelectedSeason &&
        ball_by_ball_data[batter_runs] = 6
    )

VAR SixSummary =
    SUMMARIZE(
        SeasonSixes,
        ball_by_ball_data[batter],
        "SixesCount",
        COUNTROWS(
```

```
            FILTER(
                SeasonSixes,
                ball_by_ball_data[batter] = EARLIER(ball_by_ball_data[batter])
            )
        )
    )

VAR MaxSixes = MAXX(SixSummary, [SixesCount])

VAR TopSixesPlayer =
    CALCULATETABLE(
        VALUES(ball_by_ball_data[batter]),
        FILTER(SixSummary, [SixesCount] = MaxSixes)
    )

VAR BatterTeam =
    CALCULATE(
        MAX(ball_by_ball_data[team_batting]),
        FILTER(
            SeasonSixes,
            ball_by_ball_data[batter] = MAXX(TopSixesPlayer, ball_by_ball_data[batter])
        )
    )

RETURN BatterTeam
```

29. Total 4's :

```
Total 4's =
CALCULATE(
    COUNTROWS(ball_by_ball_data),
    ball_by_ball_data[batter_runs] = 4,
    KEEPFILTERS(VALUES(ipl_matches_data[season]))
)
```

30. Total 6's :

```
Total 6's =
CALCULATE(
    COUNTROWS(ball_by_ball_data),
    ball_by_ball_data[batter_runs] = 6,
    KEEPFILTERS(VALUES(ipl_matches_data[season]))
)
```

31. Total Matches :

```
Total Matches = CALCULATE(DISTINCTCOUNT(ipl_matches_data[match_id]))
```

32. Total Points of Teams :

```
Total Points =
VAR Win = [Matches Won]
VAR NR = [No Result Played]
RETURN (Win * 2) + NR
```

33. Total Teams Participating :

```
Total Teams = CALCULATE(DISTINCTCOUNT(ipl_matches_data[team1]))
```

34. Total Venues :

```
Total Venues =
CALCULATE(DISTINCTCOUNT(ipl_matches_data[venue]))
```

SQL QUERIES

```
-- 1 Create Database

CREATE DATABASE ipl;
GO

-- 2 Use Database

USE ipl;
GO


-- 3 Create Teams Table

CREATE TABLE Teams (
Team_ID INT PRIMARY KEY,
Team_Name VARCHAR(150),
Team_Name_Short VARCHAR(50),
Image_URL VARCHAR(500)
);
GO

-- 4 Create Players Table

CREATE TABLE Players (
Player_ID INT PRIMARY KEY,
Player_Name VARCHAR(150),
Bat_Style VARCHAR(100),
Bowl_Style VARCHAR(100),
Field_Pos VARCHAR(100),
Player_Full_Name VARCHAR(200),
Player_Name 2 VARCHAR(150),
Player_Image VARCHAR(500)
);
GO

-- 5 CREATE MATCHES TABLE

CREATE TABLE Matches (
Match_ID INT PRIMARY KEY,
Season_ID INT,
Balls_Per_Over INT,
City VARCHAR(150),
Match_Date DATE,
Event_Name VARCHAR(150),
Match_Number INT,
Gender VARCHAR(20),
```

```
Match_Type VARCHAR(50),
Format VARCHAR(50),
Overs INT,
Season INT,
Team_Type VARCHAR(50),
Venue VARCHAR(200),
Toss_Winner VARCHAR(150),
Team_1 VARCHAR(150),
Team_2 VARCHAR(150),
Toss_Decision VARCHAR(50),
Match_Winner VARCHAR(150),
Win_By_Runs INT,
Win_By_Wickets INT,
Player_of_Match INT,
Result VARCHAR(100),
Stage VARCHAR(100)
);
GO

-- 6 CREATE BALL BY BALL TABLE

CREATE TABLE BallByBall (
Season_ID INT,
Match_ID INT,
Batter VARCHAR(150),
Bowler VARCHAR(150),
Non_Striker VARCHAR(150),
Team_Batting VARCHAR(150),
Team_Bowling VARCHAR(150),
Over_Number INT,
Ball_Number INT,
Batter_Runs INT,
Extras INT,
Total_Runs INT,
Batsman_Type VARCHAR(50),
Bowler_Type VARCHAR(50),
Player_Out INT NULL,
Fielders_Involved INT NULL,
IS_Wicket BIT,
IS_Wide_Ball BIT,
IS_No_Ball BIT,
IS_Leg_Bye BIT,
IS_Bye BIT,
IS_Penalty BIT,
Wide_Ball_Runs INT,
No_Ball_Runs INT,
Leg_Bye_Runs INT,
Bye_Runs INT,
Penalty_Runs INT,
Wicket_Kind VARCHAR(100) NULL,
```

```
IS_Super_Over BIT,
Innings INT);
GO


-- 7 Bulk Insert Teams Data

BULK INSERT Teams
FROM 'C:\Users\Dell\Desktop\Project\IPL\Teams.csv'
WITH (
   FORMAT = 'CSV',
   FIRSTROW = 2,
   FIELDTERMINATOR = ',',
   ROWTERMINATOR = '0x0A',
   TABLOCK
);
GO


-- 8 Bulk Insert Players Data

BULK INSERT Players
FROM 'C:\Users\Dell\Desktop\Project\IPL\Players.csv'
WITH (
   FORMAT = 'CSV',
   FIRSTROW = 2,
   FIELDTERMINATOR = ',',
   ROWTERMINATOR = '0x0A',
   TABLOCK
);
GO


-- 9 Bulk Insert Matches Data

BULK INSERT Matches
FROM 'C:\Users\Dell\Desktop\Project\IPL\Matches.csv'
WITH (
   FORMAT = 'CSV',
   FIRSTROW = 2,
   FIELDTERMINATOR = ',',
   ROWTERMINATOR = '0x0A',
   TABLOCK
);
GO
```

```
-- 10 Bulk Insert Ball By Ball Data

BULK INSERT BallByBall
FROM 'C:\Users\Dell\Desktop\Project\IPL\BallByBall.csv'
WITH (
   FORMAT = 'CSV',
   FIRSTROW = 2,
   FIELDTERMINATOR = ',',
   ROWTERMINATOR = '0x0A',
   TABLOCK
);
GO
```