

# Methodology

## The Challenge

The raw dataset lists matches sequentially. To answer the question, I needed to 'freeze time' on December 25th for every single year since 1993, while simultaneously calculating the final season results. For that i also need to make a points table.

	season	match_date	hometeam	awayteam	fthg	ftag	ftr	hthg	htag	
1	1993-94	1993-08-14 00:00:00.000000	Arsenal	Coventry	0	3	A	NA	NA	NF
2	1993-94	1993-08-14 00:00:00.000000	Aston Villa	QPR	4	1	H	NA	NA	NF
3	1993-94	1993-08-14 00:00:00.000000	Chelsea	Blackburn	1	2	A	NA	NA	NF
4	1993-94	1993-08-14 00:00:00.000000	Liverpool	Sheffield Weds	2	0	H	NA	NA	NF
5	1993-94	1993-08-14 00:00:00.000000	Man City	Leeds	1	1	D	NA	NA	NF

## The Solution: CTEs (Common Table Expressions)

I used CTEs to break the problem into logical modular steps. This avoided complex sub-queries and made the logic readable and reusable.

### CTE 1

to calculate points in every match

skill highlights

CASE...WHEN  
UNION ALL

```
WITH team_points AS (
    SELECT season, match_date, hometeam AS team,
           CASE WHEN ftr = 'H' THEN 3 WHEN ftr = 'D' THEN 1 ELSE 0 END AS point
    FROM matches
    UNION ALL
    SELECT season, match_date, awayteam AS team,
           CASE WHEN ftr = 'A' THEN 3 WHEN ftr = 'D' THEN 1 ELSE 0 END AS point
    FROM matches
),
```

### CTE 2&3

Using match point data ranking teams based on their total points. two tables for before xmas and in the end of the season

skill highlights

Window Function  
CAST  
RANK()  
MAKE\_DATE()

```
xmas_standing AS (
    SELECT season, team, sum(point) AS xmas_point,
           rank() OVER (PARTITION BY season ORDER BY sum(point) DESC) as ranking
    FROM team_points
    WHERE match_date < MAKE_DATE(CAST(LEFT(season, 4) AS INT), 12, 25)
    GROUP BY season, team
),
final_standing AS (
    SELECT season, team, sum(point) AS final_point,
           rank() OVER (PARTITION BY season ORDER BY sum(point) DESC) as ranking
    FROM team_points
    GROUP BY season, team
)
```

NEXT 