SQL Interview Question

Deloitte 2nd Round Interview

"Generate a list of all **unique possible fixtures** (matchups) for a tournament from a list of teams."

📥 Input Teams

TeamName
CSK
DC
GT

📤 Expected Output

Fixtures
CSK VS DC
CSK VS GT
DC VS GT

The Core Pattern

Self-Join for Combinations

1

Self-Join Teams

Join the teams table with itself to get all possible pairs

```
FROM teams t1 JOIN teams t2
```

2

Remove Duplicates

Filter to get only unique matchups (no same team, no reversed pairs)

```
WHERE t1.TeamName < t2.TeamName
```

3

Format Output

Concatenate team names with "VS" separator

```
CONCAT(t1.TeamName, ' VS ',
t2.TeamName)
```

4

Mathematical Formula

For n teams: $nC_2 = n(n-1)/2$ unique fixtures

$$nC_2 = n(n-1)/2$$

> 💡 **The Self-Join Pattern**
>
> This is a classic combinatorial problem solved with a self-join. The key insight is using **t1.TeamName < t2.TeamName** to eliminate duplicates and prevent teams from playing against themselves.

Complete SQL Solution

MySQL 8.0+

```
-- Generate all unique fixtures using self-join pattern
SELECT
    CONCAT(
        t1.TeamName,
        '  VS  ',
        t2.TeamName
    ) AS Fixtures
FROM
    teams t1
JOIN
    teams t2
ON
    t1.TeamName < t2.TeamName
ORDER BY
    t1.TeamName,
    t2.TeamName;
```

🎯 **KEY PATTERN**

The Magic Line: `ON t1.TeamName < t2.TeamName`

Why it works: Using `<` instead of `≠` ensures:

1. No team plays against itself
2. No duplicate fixtures (CSK vs DC and DC vs CSK)
3. Natural alphabetical ordering

Visual Explanation

How Self-Join Creates Combinations

🎲 All Possible Pairs (Before Filtering)

t1.TeamName	t2.TeamName	t1 < t2?	Keep?
CSK	CSK	CSK < CSK? = FALSE	❌ **(Same team)**
CSK	DC	CSK < DC? = TRUE	✅ **(Unique matchup)**
CSK	GT	CSK < GT? = TRUE	✅ **(Unique matchup)**
DC	CSK	DC < CSK? = FALSE	❌ **(Reverse of CSK vs DC)**
DC	DC	DC < DC? = FALSE	❌ **(Same team)**
DC	GT	DC < GT? = TRUE	✅ **(Unique matchup)**
GT	CSK	GT < CSK? = FALSE	❌ **(Reverse of CSK vs GT)**
GT	DC	GT < DC? = FALSE	❌ **(Reverse of DC vs GT)**
GT	GT	GT < GT? = FALSE	❌ **(Same team)**

Master SQL Patterns

Think in Combinations

The Self-Join Combination Pattern

Master once, use it anywhere

```
t1.col_name < t2.col_name
```

✅ Sports: Tournament fixtures

✅ E-commerce: Product bundles

✅ Social: Connection networks

✅ Finance: Stock correlations

✅ Healthcare: Drug interactions

🔄 **Repost this to show your SQL practice has become a habit**

💡 Pro Tip: Remember the pattern **t1.col_name < t2.col_name** for all combination problems!