All T20 Internationals Dataset (2005 - 2023) | SQL Joins and Union for In-Depth Cricket Data Analysis | Unit 6

Avinash Bunga

Information Systems and Business Analytics, Park University CIS622DLAF2P2023 Data Architecture for Business Analytics Professor: Gulnoza Khakimova

Nov 26, 2023

SQL Joins and Union for In-Depth Cricket Data Analysis

Introduction

In database management for sporting events, particularly T20 International cricket matches, the ability to deftly retrieve and analyze data is paramount. SQL queries become the tools through which valuable insights are garnered from the underlying data. Employing UNION and JOIN operations in SQL allows us to aggregate data from different tables and to collate information based on specific conditions. This discourse focuses on formulating SQL queries for a T20 Internationals Dataset, utilizing UNION to merge distinct lists and JOIN to extract related data across multiple tables.

Questions Necessitating Two JOINs and a UNION

Question 1: List all players who have either bowled more than 5 overs in a match or batted in the top 5 positions in any T20 international.

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BowlingCard ON Player.PlayerID = BowlingCard.PlayerID
WHERE BowlingCard.OversBalled > 5
UNION
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BattingCard ON Player.PlayerID = BattingCard.PlayerID
WHERE BattingCard.Innings <= 5;
```

Justification for UNION

A UNION is required to compile a list of players based on two distinct criteria—bowling and batting—which are not directly related. It merges the results from separate queries into a single list, ensuring no player is counted twice, even if they meet both conditions.

Question 2: Identify players who have participated as a captain or won the Man of the Match award in any T20 international.

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN TeamCaptain ON Player.PlayerID = TeamCaptain.PlayerID
UNION
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN Match ON Player.PlayerName = Match.ManOfTheMatch;
```

Justification for UNION

This query employs UNION to create a combined list of players who have served as captains and those who have won the Man of the Match award. Since these two roles do not overlap by necessity and can occur in different matches, UNION is necessary to amalgamate the two separate lists into a single unique list of player names. A JOIN would not be appropriate as it would imply that we are only interested in cases where a player has both captained and won the Man of the Match in the same game, which is not the requirement here.

Question 3: List all venues that have hosted either the opening match or the final match of a T20 international series.

SQL Query:

```
SELECT DISTINCT Venue.VenueStadium
FROM Venue
JOIN Match ON Venue.VenueID = Match.VenueID
JOIN Series ON Match.SeriesID = Series.SeriesID
WHERE Match.MatchNo = 1
UNION
SELECT DISTINCT Venue.VenueStadium
FROM Venue
JOIN Match ON Venue.VenueID = Match.VenueID
JOIN Series ON Match.SeriesID = Series.SeriesID
WHERE Match.MatchNo = (
  SELECT MAX(MatchNo) FROM Match WHERE Match.SeriesID =
Series.SeriesID
);
```

Justification for UNION

A UNION is used to merge two sets of venue data: one set that hosted the first match of a series (MatchNo = 1) and another that hosted the final match of a series (the maximum MatchNo within a series). The UNION is appropriate because the two conditions represent separate match instances that don't have a direct relational overlap.

Question 4: List players who have either scored over 100 runs or taken 5 or more wickets in a single T20 international match.

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BattingCard ON Player.PlayerID = BattingCard.PlayerID
WHERE BattingCard.RunsScored > 100
UNION
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BowlingCard ON Player.PlayerID = BowlingCard.PlayerID
WHERE BowlingCard.WicketsTaken >= 5;
```

Justification for UNION

The UNION is used to compile a list of standout individual performances in different disciplines: batting and bowling. The two lists are exclusive as they represent two different forms of cricketing achievements, hence requiring UNION to merge them into a single list of players who have had either batting or bowling excellence.

Question 5: Find all players who have either hit more than 3 sixes in a match or bowled more than 10 overs in any T20 international.

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BattingCard ON Player.PlayerID = BattingCard.PlayerID
WHERE BattingCard.Sixes > 3
```

```
UNION
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BowlingCard ON Player.PlayerID = BowlingCard.PlayerID
WHERE BowlingCard.OversBalled > 10;
```

Justification for UNION

This query leverages UNION to combine two separate performance achievements: batting prowess and bowling stamina. The first part of the union gathers players who have hit more than 3 sixes in a match, indicative of aggressive batting. The second part selects players who have delivered more than 10 overs, showcasing endurance in bowling. As these are distinct feats that don't correlate directly, UNION is the ideal choice to amalgamate the results, ensuring that players who have achieved either milestone are represented without repetition.

Conclusion

The sophisticated interplay between JOIN and UNION operations within SQL queries is instrumental in dissecting and understanding complex data relationships in a T20 Internationals Dataset. When queries necessitate the simultaneous use of two JOINs, it allows for a deeper linkage between related data points across multiple tables, affording a more detailed and nuanced data extraction. These JOIN operations are critical when the information needed depends on the intersection of multiple criteria, such as player performances in specific matches or the roles individuals play in the broader context of the game.

Moreover, the strategic use of UNION in conjunction with JOINs enhances the capability to compile comprehensive lists from disparate yet related categories of data. This is particularly

useful in scenarios where the datasets are mutually exclusive or when there is a need to consolidate results from different queries into a singular output without redundancy.

These SQL operations empower data analysts to create robust, multifaceted queries that cater to intricate questions. They showcase the dynamism and flexibility of SQL in handling complex data retrieval tasks, making it an invaluable tool for generating actionable insights in sports analytics and beyond. The ability to seamlessly combine multiple JOINs with a UNION operation underscores the agility of SQL in navigating relational databases, ensuring that users can access a full spectrum of information with precision and ease.

Reference:

Taylor, A. G. (2019, March 31). *How to use the SQL union join*. Dummies. https://www.dummies.com/article/technology/programming-web-design/sql/how-to-use-the-sql-union-join-260866/