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 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.

Questions Involving at least Two JOINs

Question 3: Which players have scored a half-century and taken at least two wickets in the same T20 international match?

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BattingCard ON Player.PlayerID = BattingCard.PlayerID
JOIN BowlingCard ON Player.PlayerID = BowlingCard.PlayerID AND
BattingCard.MatchID = BowlingCard.MatchID
WHERE BattingCard.RunsScored >= 50 AND BowlingCard.WicketsTaken >= 2;
```

Justification for JOINs

This query utilizes INNER JOIN to find players who have both batted and bowled effectively in the same match. It ensures that only players who meet both criteria scoring a half century and taking 2 or more wickets, are selected.

Question 4: Which players have played in the most T20 international matches?

SQL Query:

```
SELECT Player.PlayerName, COUNT(MatchPlayers.MatchID) AS
NumberOfMatches
FROM Player
JOIN MatchPlayers ON Player.PlayerID = MatchPlayers.PlayerID
JOIN Match ON MatchPlayers.MatchID = Match.MatchID
GROUP BY Player.PlayerName
ORDER BY NumberOfMatches DESC;
```

Justification for JOINs

The query uses two INNER JOIN operations to correlate players with their participation in matches. An INNER JOIN is appropriate as we want to count only the matches that each player actually played in. The GROUP BY clause aggregates the number of matches per player, and the ORDER BY clause ranks the players by their number of appearances.

Question 5: List all players who have taken a wicket and scored runs in the same match.

SQL Query:

```
SELECT DISTINCT Player.PlayerName
FROM Player
JOIN BattingCard ON Player.PlayerID = BattingCard.PlayerID
JOIN BowlingCard ON Player.PlayerID = BowlingCard.PlayerID AND
BattingCard.MatchID = BowlingCard.MatchID
WHERE BattingCard.RunsScored > 0 AND BowlingCard.WicketsTaken > 0;
```

Justification for JOINs

The INNER JOIN is utilized twice in this query to integrate data from the Player, BattingCard, and BowlingCard tables. The first join connects the Player table with BattingCard to filter players who have scored runs. The second join with BowlingCard further refines the selection to those players who have also taken wickets in the same match:

- The WHERE clause is pivotal as it enforces the criteria of scoring runs and taking wickets, thereby focusing on players who have demonstrated all-round capabilities in a single game.
- The DISTINCT keyword ensures that players are listed once, even if they have met the criteria in multiple matches.

Conclusion

The strategic implementation of UNION and JOIN operations within SQL queries facilitates the extraction of fine data from a T20 Internationals Database. It showcases the ability to synthesize discrete data points and draw correlations across diverse datasets. Applying these SQL constructs allows for a meticulous analysis of player performance, match outcomes, and officiating patterns, offering a granular view into the fabric of T20 cricket statistics. Through these queries, we can unveil patterns, identify exceptional performances, and understand the dynamics of match venues, contributing to the rich tapestry of cricket analytics.