

Database Normalization

30th June, 2022

Submitted by: Akshay Lad

Need for Normalization

- To minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- This is achieved by dividing the larger tables into smaller ones and linking them using relationships.

Types of Normalization

1. 1NF

It says that columns in DB should have atomic value. Such that one column should not have multiple values.

All the available data is single valued (ie. its atomic) . Thus, the dataset is in 1NF.

2. 2NF

It says all non-key attributes should be dependent on key attributes. In our dataset,

We remove these attributes like Venue, Umpire, Player to form their own entity. Thus, we create a table of their own with their unique id which maps with current data through a foreign key.

3. 3NF and 3.5NF (BCNF)

It is used to reduce data duplication. It says for a relation $X \rightarrow Y$, X should be a super key and there should be no transitive relation between non-prime attributes.

In our dataset,

In the Matches table, “City” attribute tells about the city where the match is played. “Venue_ID” links the details of the venue (stadium) where the match is being played. Now, Functional dependency is as follows:

Match_ID \rightarrow Venue_ID, city Venue_ID \rightarrow City

Also, Match_ID \rightarrow city

Thus, the Matches table is not in 3NF. To resolve this, We move the “city” attribute in the Venues table along with stadium details.

In the Deliveries table,

The Delivery_ID (Primary key) depends on Extra_runs and Extra_type but also they depend on each other. Dismissal_type and Dismissed_player_id depend on is_wicket but also on delivery_ID.

Functional dependency is as follows:

Delivery_ID \rightarrow Extra_type, Extra_runs

Extra_type \rightarrow Extra_runs

The players dismissal, wicket, Dismissal_type are functional dependency is as follows:

Delivery_ID \rightarrow is_wicket, Dismissal_Type, Dismissed_player_id, Fielder_id

Is_wicket \rightarrow Dismissed_player_id, Dismissal_Type

We decompose the Deliveries into Extra_Runs and Dismissal_Type

This ensures all the tables are upto 3.5NF.

4. 4NF

In 4NF, we check for multi-valued dependencies between attributes and primary keys. All tables are already in 4NF form.

5. Junction Tables

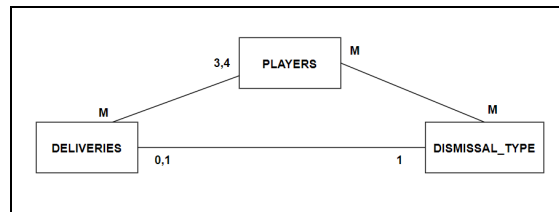


Figure: Many-to-Many relationship table

Players and Dismissal_Type have a many to many relationship. Such that each player can be dismissed by many types. A dismissal_type can have multiple players.

Junction tables are used to avoid MANY-TO-MANY relations.

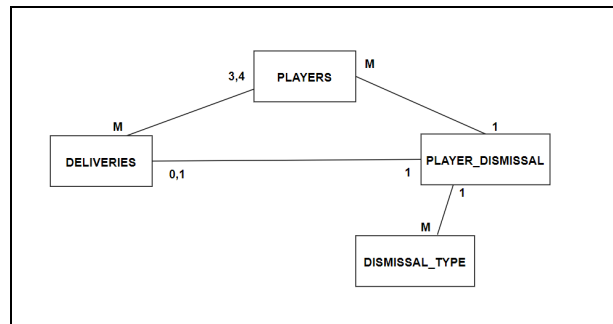


Figure: Junction table implemented