

# **COMP 3380 Project**

Part 4 - Project Report

Group 26

December 23, 2021

Drafted by members

Aneesh Makkar

Mansimar Singh Bhasin

Nouran Bileha

Tanish Sood

Zenish Yopinbhai Patel

Note: we updated our ER diagram, relational model, and normalization. We basically just removed one entity (Umpire) and one relationship (referee) because it was not consistent with our database. The new documents are attached in this submission and can be found in part 1 folder.

## **A summary of the data**

The data that we have worked on is about a cricket league.

### **1) Why was it chosen?**

Our group wanted to find a database that is of interest to us. We like the cricket game, and it is a very popular game played in a few countries. We know a lot about the game, so we thought it would be fun, easy and interesting to build a database on the topic in which we have in-depth knowledge in order to come up with interesting queries.

### **2) What does it consist of?**

It consists of data such as all the players, their batting skills and bowling skills, all matches that were played from the year 2008 to 2016, the location where the games were played.

### **3) How large is it? (File size, number of records)**

File size: 580 KB (a total of 14 tables)

Total number of records across all tables: 17,465

## **A discussion of the data model**

### **1) Why was it broken down into those tables?**

We focused on normalization that is why our data was broken down into 14 tables in order to avoid redundancy and lossiness. When doing our normalization, we aimed for having BCNF in all of our tables, and we luckily achieved that. That is why our data include 4 extra tables that resulted from normalization.

### **2) Did students face any difficult choices when deciding on how to set up the model?**

Finding data that interests us all was challenging. We wanted to make sure that we understand our data in order to make good queries. When we found a database, it was also a challenge to get approval from the owner of the data. In addition, deciding total or partial participation was easy for some relationships, but we had to do some research to make sure about total participation such as match and man of because we wanted to see if every match in the season over the past years had a man of the match.

## **A summary of the database**

List each of the final tables, along with its cardinality and arity:

Competing: 1154 rows and 2 columns (TeamID, MatchID)

Designation: 540 rows and 2 columns (ID, RoleID)

Location: 29 rows and 3 columns (CityID, CityName, Country)

Match: 577 rows and 3 columns (MatchID, SID, VenueID)

MatchInfo: 577 rows and 2 columns (MatchID, CityID)

Player: 469 rows and 2 columns (MatchID, ID)

PlayerInfo: 469 rows and 2 columns (PlayerName and DOB)

PlayerMatch: 12694 rows and 2 columns (MatchID, ID)

Position: 4 rows and 2 columns (RoleID, RoleName)

Season: 9 rows and 2 columns (SID, Syear)

Skills: 426 rows and 3 columns (ID, BattingStyle, BowlingStyle)

Stadium: 35 rows and 3 columns (VenueID, VName, CityID)

Team: 13 rows and 2 columns (TeamID, TeamName)

TeamInfo: 469 rows and 3 columns (ID, TeamID, PlayerName)

## **A list of the queries implemented in Part 2**

On top of the queries mentioned below, we also display our data using our interface.

1. A query that will give the number of players that play in a specific team.
2. Players with same DOB
3. Players who played at least 1 match in “DELHI” (capital city of IPL).
4. City in which the maximum number of matches were held in descending order.
5. Looking for a captain for the teams? Here we go!
6. Players who played a match in Chennai (or any other city) as a wicketkeeper (or any other role).
7. Number of cricket stadium in each country
8. Player who played the most number of matches.

9. Cricketers who played in UAE(or any other country) and were right arm medium pacers(or any other batting/bowling skills)
10. Which team played the maximum number of matches in all cities

## **A summary of the interface**

We made the interface using netbeans. Firstly, we made a connection with sqlite in netbeans, so we can use our database to make a working graphical user interface. After that, we decided how to insert the query to get meaningful results. After considering the factor that no one should be able to do sql injection on our database, we used prepared statements where the user's input was required. Moreover, we added a combo box wherever user input is required. The languages that were used in development of this project are sql and java. External libraries are imported in the project. The Names of libraries are Prepared Statement Library, Result set Library, IOExceptions library. PrintWriter was also imported for printing of data in csv files.

It is an easy to use interface, but we were unable to make an exe file due to a problem in the java development kit. So, we submitted the net beans project. To use it, the first person has to add a jdbc driver file (jar file provided) to the libraries folder of the project. After that, a person can open Frame.java present in the javaApplication1 package. Open the 'source' window where the code is written and right click on the pane where the code is written and then run the file using the menu which opens when a person right clicks on the pane. We also provided a video link in case the marker is unable to open the project.

Video Link for our interface:

<https://youtu.be/w9hCok-SX8A>