DATABASE AND APPLICATION

PROJECT PHASE 1

SRS Documentation For Cricket Tournament

Team Name:
A_2
Team members:
Kushagra Agarwal (2018113012)
Shreeya Pahune (2018113011)

Introduction

This is an SRS Documentation for a DBMS which keeps track of a mini-world for any Cricket Tournament handling the most major functions.

Purpose

The following document describes a database/archive for a Cricket Tournament. As an overview it allows the creation, maintenance and modifying of details about the team and players along with the matches played over the course of the tournament.

Users

Cricket enthusiasts can look into the DB for current stats and the data can also be used by Sports websites or any Sports reporters. Furthermore it acts as an archive. Coaches of teams who wish to track the performances of their players as well as opponents they need to be wary of.

What users do with DBMS

This DBMS allows efficient and reliable retrieval of information by the users. Finding out match statistics and analysing the performance of different players. This can be helpful to book tickets in advance for future matches by looking up the day they are going to be played and the venue. Also the user can follow his favourite players' performances in the tournament. It also helps to readily obtain the best performers and performances of the tournament.

List of all Entities, their attributes and a description:

- ***Teams:***

  Contains all the teams participating in the tournament along with description given by these attributes:

  - *TeamID:* Primary Key, INTEGER, NOT NULL (ID of the Team)
  - *CountryCode:* INTEGER, NOT NULL (Code of the country the team represents)
  - *Coach:* VARCHAR(20) (Name of the coach the team)
- ***Players:***

  Contains all the players participating in the tournament along with description given by these attributes:

  - *PID:* Primary Key, INTEGER , NOT NULL (ID of the Player)
  - *TeamID:* INTEGER, NOT NULL (Team for which the player plays)
  - *Fname:* VARCHAR(20), NOT NULL (First Name of Player)
  - *Lname:* VARCHAR(20), NOT NULL (Last Name of Player)
  - *CaptainID:* INTEGER, NOT NULL (PID of the player that is the captain of the team)
  - *Age:* INTEGER, NOT NULL (Age of Player)
  - *JerseyNo:* INTEGER, NOT NULL *(*Number on the player's Jersey)
  - *JColour:* VARCHAR(20), NOT NULL (Jersey colour)
- ***Matches***

  Contains all the matches played in the tournament along with description given by these attributes:

  - *TID1:* Foreign Key, INTEGER, NOT NULL (ID of participating team)
  - *TID2:* Foreign Key, INTEGER, NOT NULL (ID of other participating team)
  - *Day:* Partial Key, INTEGER, NOT NULL (Day of Tournament on which match was played)

- *VenueID:* INTEGER, NOT NULL (ID of venue where match was played)
- *Won by:* VARCHAR(20) (TID of Winning Team)
- *Man of the Match:* INTEGER (PID of the Man of the Match)
- *Umpire:* VARCHAR(20), NOT NULL (Name of the Umpire)

- ***Bowlers:***

  Details of Bowlers (subclass of Players)

  - *BowlerID:* Primary Key, INTEGER, NOT NULL (ID of the Player)
  - *Matches:* INTEGER, NOT NULL (Number of matches played by player)
  - *RunsGvn:* INTEGER, NOT NULL (Total Runs given by bowler)
  - *Wickets:* INTEGER, NOT NULL (Total wickets taken by player)
  - *Best Figure:* INTEGER, NOT NULL (Best Figure of Player)
  - *BowlAverage:* Derived Attribute, INTEGER, NOT NULL (RunsGvn/Wickets)

- ***Batsman:***

  Details of Batsman(subclass of Players)

  - *BatID:* Primary Key, INTEGER, NOT NULL (ID of the Player)
  - *Innings:* INTEGER, NOT NULL (Number of innings played by player)
  - *Runs:* INTEGER, NOT NULL (Total runs made by player)
  - *High Score:* INTEGER, NOT NULL (Highest Score of Player)
  - *Innings not out:* INTEGER, NOT NULL (Number of innings played by player for which he wasn't out)
  - *BatAverage:* Derived Attribute, INTEGER, NOT NULL (Runs/[Innings-Innings not out])

- ***Country:***

  Countries playing in the Tournament

  - *CID:* Primary Key, INTEGER, NOT NULL ID of the Country
  - *CName:* Name of Country

- ***Venue:***

  Description of the venue

  - *VID:* Primary Key, INTEGER, NOT NULL ID of the Venue
  - *Venue:* Name of Venue
- ***Player type:***

  Description of the each player's type

  - *PID:* Foreign Key, INTEGER, NOT NULL (ID of the Player)
  - *Type:* Partial Key, VARCHAR(20) IN {Batsman, Bowler} (what the type is: {Batsman, Bowler})

Database Requirements:

Entity types:

1. Teams
2. Players
3. Batsman, Bowlers
4. Matches
5. Country, Venue
6. Player Type

Weak Entity types:

1. Schedule:
   - Foreign Key: TID1, TID2
   - Partial Key: Day
2. Player Type:
   - Foreign Key: PID
   - Partial Key: Type

Relationship types:

1. Team REPRESENTS Country: 1-1 Relationship (Binary)
2. Player PLAYS FOR Team: N-1 Relationship (Binary)
3. Player CAPTAIN OF Players: 1-N Relationship (Unary)

4. Team PLAYS Match: N:M Relationship (Binary)
5. Matches PLAYED ON Venue: N-1 Relationship (Binary)

n>3 Relationships:

1. At a Venue A, a certain Match B is played, with one of the playing teams being C, having a player D, whose high score is E

   [Venue] -> [Matches] -> [Teams] -> [Players] -> [Batsman]

2. Matches, Players, Teams, Venues are linked by the relationship Plays in a Tournament. Players of Teams play tournaments Matches at Venues.

```
                         [Matches]
                             |
[Venues]  --  <Plays in a Tournament>  --  [Teams]
                             |
                         [Players]
```

Subclass:

1. Batsman is subclass of Players
2. Bowlers is subclass of Players

Attribute types:

1. Composite Attributes:
   - *Name attribute in Player* can be broken down as FName and LName
   - *Jersey attribute in Player* can be broken down as JNo and JColour
2. Multivalued Attributes:
   - *Type attribute in Player Type*, here each player can take multiple values of Type
3. Derived Attributes:
   - *BatAverage* attribute in Batsman

    - *BowlAverage* attribute in Bowlers

Functional Requirements

Insertion:

1. Insertion of a New Player if he is selected into the team
2. Insertion of a New Team if it qualifies for the Tournament
3. Insertion of a New Bowler record if a Batsman starts bowling
4. Insertion of a New Batsman record if a Bowler starts batting
5. Insertion of a New Venue if a new stadium is constructed
6. Inserting the Man of the Match entry after a match is over
7. Inserting the Won By entry after the match is over

Updation:

1. Updating the Records of a Batsman after every match
2. Updating the Records of a Bowler after every match
3. Updating the Venue for a Match if due to unforeseen circumstances the match is relocated
4. Update the Age of a Player to -1 if he dies due to unforeseen circumstances

Deletion:

1. Deleting a Venue record if it is demolished

Report:

1. Calculating the Number of WIns for each team: We can do this by adding all the times a particular team's id appears in the attribute (Won by) in the Matches entity
2. Finding the Best Batsman of the tournament: Sort the Runs attribute in the Batsman entity and take the first entry
3. Finding the Best Bowler of the tournament: Sort the Wickets attribute in the Bowler entity and take the first entry
4. Finding the Highest Individual score of the tournament: Sort the Highest Score attribute of the Batsman entity and take the first entity

5. Finding the Best Bowling Figures in the tournament: Sort the Best Figures attribute of the Bowler entity and take the first entity
6. Finding the Number of Players who are All Rounders, i.e, both Batsman and Bowler: In the entity Player Type we calculate the number of entries which are multi-valued, i.e, have both the types batsman and bowler assigned to them.