DATABASE MAMAGEMENT SYSTEM PROJECT ON INDIAN PREMIER LEAGUE 2020

MSc DATA SCIENCE 2020-2022
UNIVERSITY OF KALYANI

TEAM

Name: Asmita Bose

Roll No: 96/DTS 200006

Name: Sriejan Bagchi

Roll No: 96/DTS 200016

Content

Sl No	Subject
1	Acknowledgement
2.	Introduction
3.	Objective
4.	Datatype and its description
5.	Data Requirements
6.	Data Model
7.	1. Entity Relationship Model
8.	2. Relational Model
9.	ER Model of IPL 2020
10.	Notations Used for Diagram
13.	Conversion of E-R Model to Relational Model
14.	Relationship Model of IPL 2020
15.	Conclusion

Acknowledgement

In the accomplishment of this project successfully, many people have best owned upon us their blessings and the heart pledged support, this time we're utilizing to thank all people who have been concerned with this project.

Primarily, we would thank God for being able to complete this project with success. Then we will thank Prof. Utpal Biswas and Prof. Riman Mondal, under whose guidance we learned a lot about this project. Their suggestions and directions have helped in the completion of this project.

We are thankful to my parents as well. We were able to successfully complete this project with the help of their guidance and support. Finally, we want to thank all our dear friends as well.

Introduction

The **Indian Premier League** (**IPL**), also officially known as **TATA IPL** for sponsorship reasons, is a professional men's Twenty20 cricket league, contested by ten teams based out of seven Indian cities and three Indian states. The league was founded by the Board of Control for Cricket in India (BCCI) in 2007. It is usually held between March and May of every year and has an exclusive window in the ICC Future Tours Programme.

The IPL is the most-attended cricket league in the world and in 2014 was ranked sixth by average attendance among all sports leagues.In 2010, the IPL became the first sporting event in the world to be broadcast live on YouTube.The brand value of the IPL in 2019 was ₹47,500 crore (US$5.9 billion), according to Duff & Phelps.According to BCCI, the 2015 IPL season contributed ₹1,150 crore (US$140 million) to the GDP of the Indian economy.The 2020 IPL season set a massive viewership record with 31.57 million average impressions and with an overall consumption increase of 23 per cent from the 2019 season.

There have been fifteen seasons of the IPL tournament. The current IPL title holder franchise is Gujarat Titans, winning the 2022 season.

IPL Management System Project Overview It is a **multi-role application project** i.e. Admin and User, where Admin will have the primary control over the system. This is a complete web application where users can view information about players & IPL teams.

Objective

It keeps track of all the information about the matches of season 2020, their venue, point table, team, team owner, umpire. The viewers will in it easy in this automated system rather than using the manual writing system. The system contains a database where all the information will be stored safely.

DATA TYPES AND ITS DESCRIPTION

➢ **INT (size)**: A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The size parameter specifies the maximum display width (which is 255).

➢ **CHAR (size):** A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1.

➢ **DATE:** A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'.

➢ **FLOAT (size, d):** A floating point number. The total number of digits is specified in size. The number of digits after decimal point id specified in d parameter.

DATA REQUIREMENTS

Entities

• Match

• Point Table

• Team

• Owner

• Venue

• Umpires

Attributes

❖ Match

• MID

• MDATE

• MVENUE

• MRESULT

• M_UMP

❖ POINT TABLE

• P_T_TEAM

• WINS

• LOSS

• TOTAL

• PTS

• NRR

❖ **TEAM**

• TID

•TNAME

• TCAP

• T_STATE

❖ **OWNER**

• OID

• O_TEAM

❖ **VENUE**

• VID

• VNAME

• CITY

❖ **UMPIRE**

• UID

• UNAME

RELATIONSHIPS – CARDINALITY

❖ MATCH played at VENUE (1 – 1)

❖ MATCH played by TEAM (M – N)

❖ MATCH decided by UMPIRE (1 – 2)

❖ POINT TABLE keeps record of MATCH (1 – M)

❖ POINT TABLE ranked TEAM (1 – M)

❖ TEAM owns by OWNER (1 – 1)

DATA MODEL

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific; hence they were prone to introduce lots of duplication and update anomalies.

Entity-Relationship Model

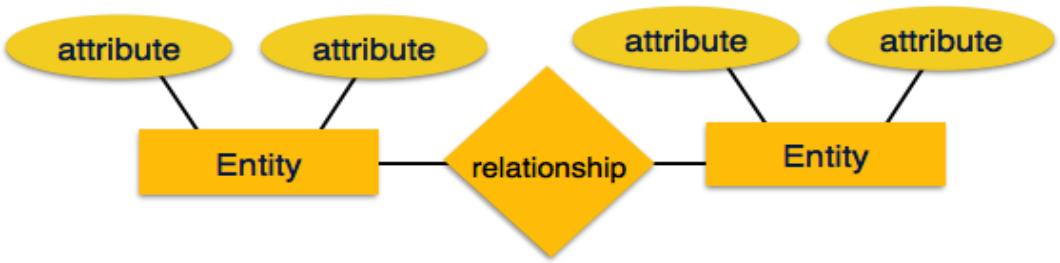
Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on –

- **Entities** and their *attributes*.
- **Relationships** among entities.

These concepts are explained below.

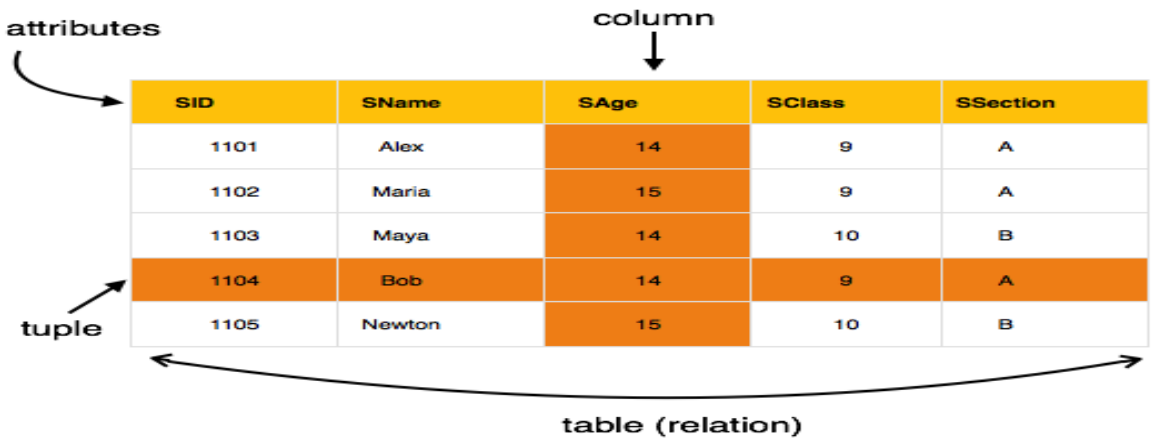


- **Entity** – An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship** – The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

- Mapping cardinalities –
    - one to one
    - one to many
    - many to one
    - many to many

Relational Model

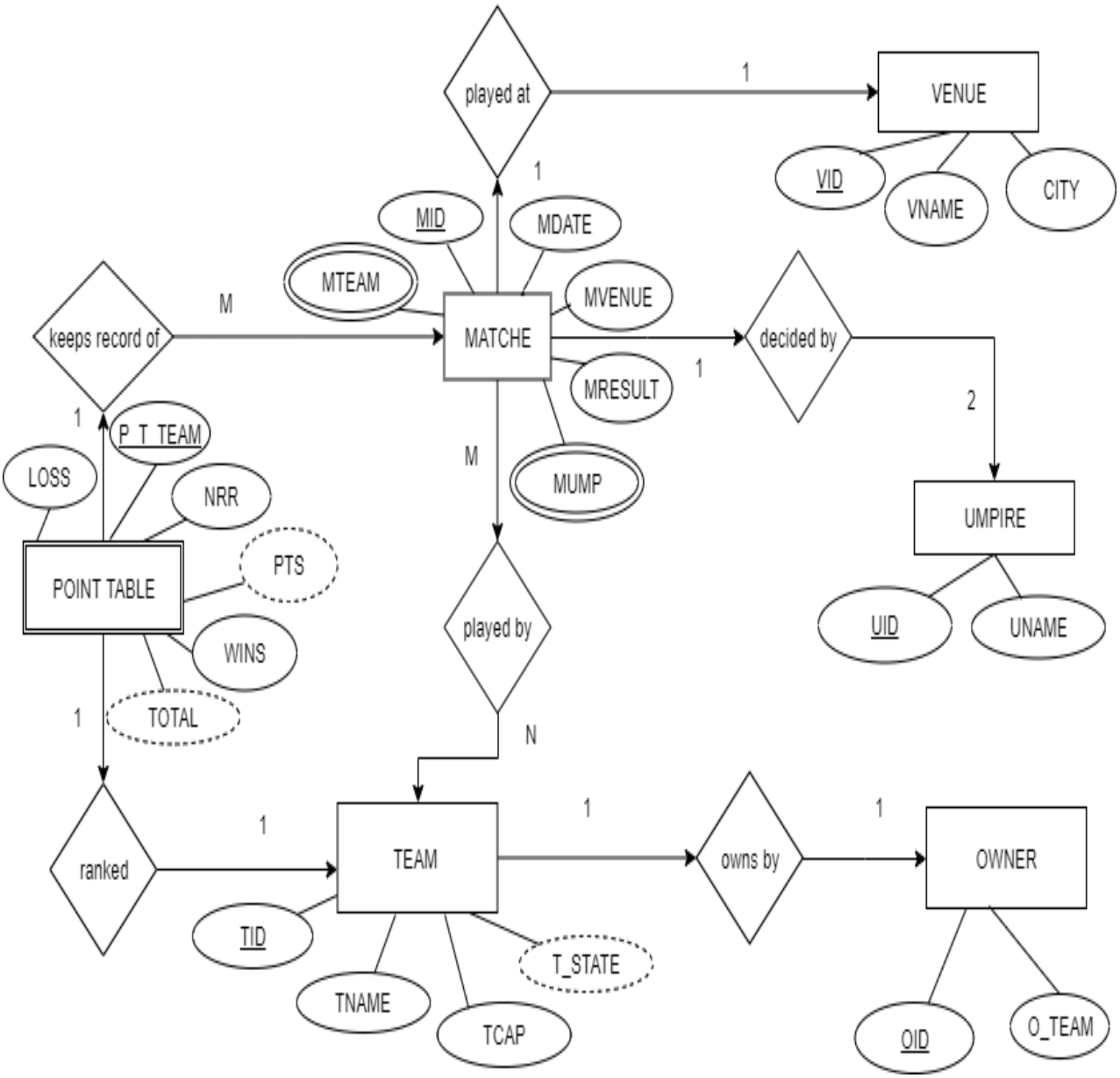
The most popular data model in DBMS is the Relational Model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an **n-ary relation**.



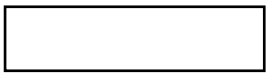
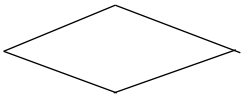
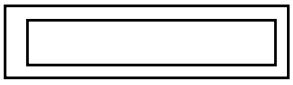
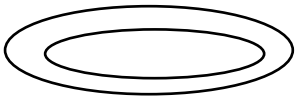
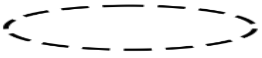
The main highlights of this model are –

- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

E-R DIAGRAM OF IPL 2020



NOTATIONS USED FOR E-R DIAGRAM

E-R ELEMENT	SYMBOL	SYMBOL NAME	FUNCTION
Entity		Rectangle	Represent entity sets.
Attribute		Ellipse	Represent attributes.
Relationship		Diomand	Represents relationship set among Entity set.
Link		Line	Link attributes to entity sets & entty sets to relationshio sets.
Week Entity		Double Rectangle	Represents weak entity sets
Multivariate Attribute		Double Ellipse	Represents multivalued attributes
Derived Attribute		Dashed Ellipse	Represent derived attributes
Key Attribute		Underlined Ellipse	Represents primary key attributes

CONVERSION OF ER MODEL TO RELATIONAL MODEL

Since E-R diagram gives us the good knowledge about the requirement and the mapping of the entities in it, we can easily convert them as tables and columns, I.e., using E-R diagrams one can easily create relational data model, which is nothing but the logical view of the database.

There are various steps involves in converting it into tables and columns. Each type of entity, attribute and relationship in the diagram takes their own depiction here. In this section our object is to convert the E-R diagram of Ipl 2020 into tables, columns and mappings.

The basic rules for converting the E-R diagram into tables are:

- **Convert all the Entities in the diagram to tables**

All the entities represented in the rectangular box in the E-R diagram become independent tables in the database. In the given diagram MATCHE, VENUE, POINT TABLE, TEAM, OWNER, UMPIRE from individual tables.

- **All single valued attributes of one entity are converted to a column of the table**

All the attributes, whose value at any instance of time is unique, are considered as columns of that table. In the MATCHE Entity, M_Id, M_DATE from the columns of MATCHE Table. Similarly, V_ID, V_NAME from the columns of VENUE table and so on.

- **Key attribute in the E-R diagram becomes the Primary key of the table**

In diagram, M_ID, V_ID, U_ID, T_ID, O_ID are the key attributes of the entities. Hence, we consider them as the primary keys of the respective tables.

- **Declare the foreign key column, if applicable**

In the diagram, attribute M_VENUE, MRESULT, MTEAM< P_T_TEAM are the foreign keys. MVENUE in MATCHE entity is from VENUE entity. Hence, add MVENUE in MATCHE table and assign it foreign key constraint. Hence, by declaring the foreign key constraints, mapping between the tables is established.

- **Any multi-valued attributes are converted into new table**

MTEAM in MATCHE table is a multivariate attribute. In one match always two team participates. So, we cannot represent multiple values in a single column of MATCHE table. We need to store it separately, so that we can store any number of teams, adding/ removing/ deleting MTEAM should not create any redundancy or anomalies in the system. Hence, we create a separate table MATCHD with M_ID and MTEAM and MUMP as its column (since MUMP is also a multivariate attribute). We create a composite key using both the columns.

- **One can ignore derived attribute, since it can be calculated at any time**

In TEAM table, TSTATE can be predicted at any point of time by considering the TNAME. Again, in POINT TABLE PTS can be derived at any point of time by multiplying WINS by 2. Hence, we need not create a column for this attribute. It reduces the duplicity in the database.

These are the very basic rules of conerting E-R diagram into tables and columns and assigning the mapping between the tables.

Lets consider some of the special cases:

- **Converting Weak Entity**

  Weak entity is also represented as table. All the attributes of the weak entity form the column of the table. But the key attribute represented in the diagram cannot form the primary key of this table. We have to add a foreign key column, which would be the primary key column of its strong entity. This foreign key column along with its key attribute column from the primary key of the table.

In our diagram, POINT TABLE is the weak entity. Hence, we create a table for it. Its attributes P_T_TEAM, WINS, LOSS, TOTAL, NRR, PTS forms the column of the table. Although P_T_TEAM is represented as the key attribute in the diagram, it cannot be considered as primary key. IN order to add primary key to the column, we have to find the foreign key first. TEAM is the strong entity relaed to POINT TABLE. Hence, the primary key T_ID pf TEAM is added to POINT TABLE as the foreign key. Now, we can create a composite primary key out of T_ID and P_T_TEAM.

- **Representing 1:1 relationship**

In our diagram, we have MATCHE played at VENUE relation. It is a 1:1 relation I.e. one match is conducted at one venue at a time. We can represent this case in two ways:

1. Create table for both MATCHE and VENUE. Add the primary key of VENUE in MATCHE table as foreign key. It implies the venue for that particular match.

2. Create table for both MATCHE and VENUE. Add the primary key of MATCHE in VENUE table as foreign key. It implies the match for that particular venue.

In both the cases, meaning is same. Foreign key column can be added in either of the table, depending on developer's choice.

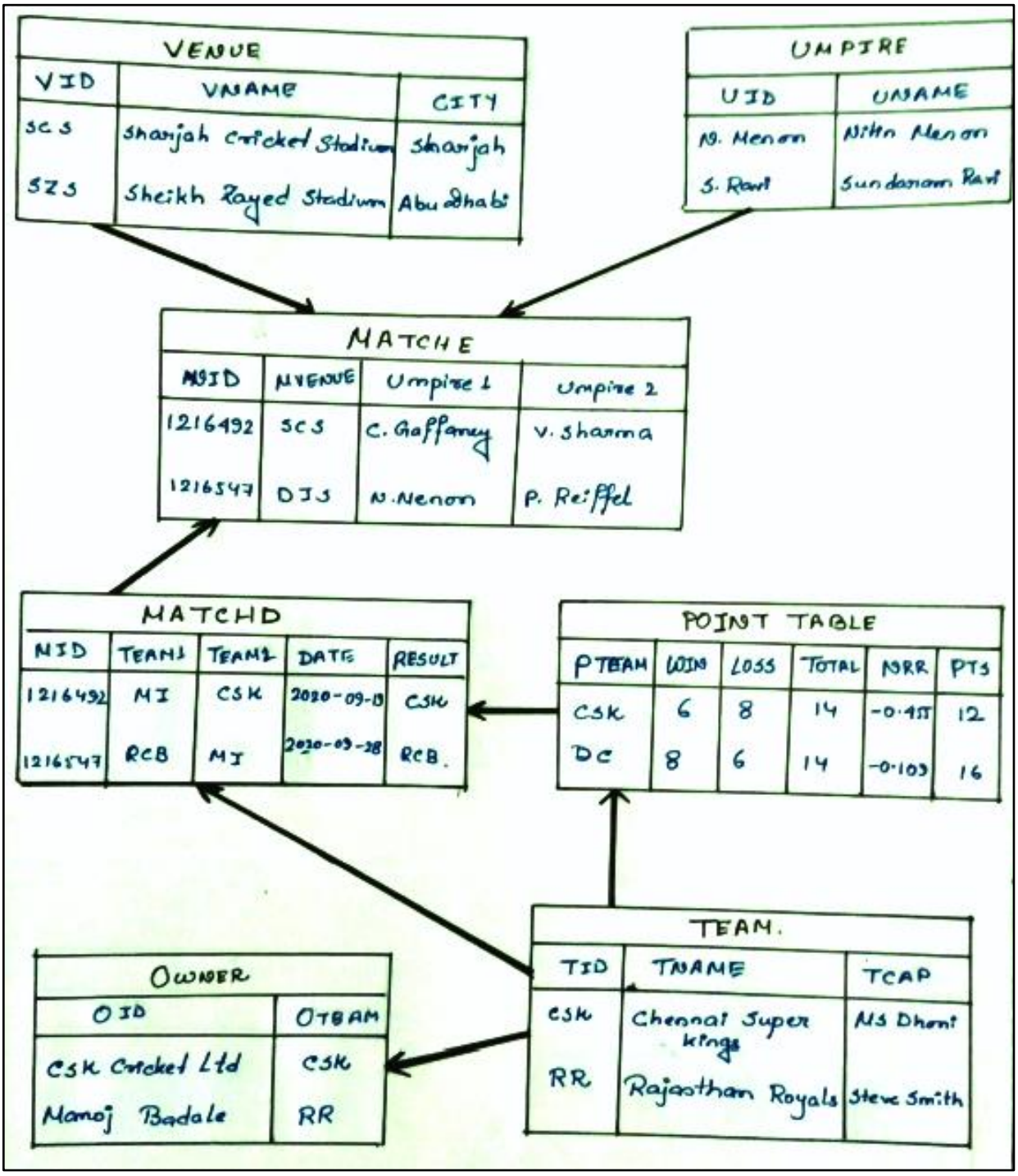
- **Representing 1:N relationship**

Consider POINT TABLE and MATCH relation, where one POINT TABLE keeps records of multiple matches. This is a 1:N relation. In this case, primary key of POINT TABLE is added to MATCHE table as MRESULT I.e. the primary key at 1 cardinality entity is added as foreign key to N cardinality Entity.

- **Representing M:N relationship**

Consider the example, multiple MATCHE played by multiple TEAM, which is a M:N relation. In this case both the participating entities are converted into tables and a new table is created for the relation between them . Primary keys of entity tables are added into new table to form the composite primary key. We can add any additional columns, if present as attribute of the relation in E-R diagram.

RELATIONAL MODEL FOR IPL 2020



CONCLUSION

The project, which was created with PHP and MySQL, is based on the user's requirements and an analysis of the current system, with room for future improvement. Modern software's increased functionality necessitates a suitable methodology for software development. This IPL database management program is intended for users who wish to keep track of various details by entering them into a database. Various information and details concerning the contest quickly expanded. The number of matches will subsequently grow daily as a result. Therefore, it is quite difficult for someone watching the IPL to keep track of upcoming games, as well as to observe the accomplishments of different players and have data at their fingertips. After identifying the shortcomings of the current system, a more user-friendly and GUI-focused computer system is designed that will be compatible with the existing system. Potential Improvements The current project merely involves gathering the data, storing it in the appropriate data tables, expressing it in the various forms necessary, and enabling attribute-based searching. There are several improvements that can be put into practice even further. These are what they are:

- Module that, when the appropriate required item is selected or entered, automatically provides information on various cricket boards, stadiums, and rankings of different IPL teams and players.
- Module that provides information about fully and partially completed information on different details.
- Can design a module that allows users to login and access information through a window.
- Include players in the scope of this project.