**Department of Computer Science and Engineering**

**Course Code: CSE-304**

**Database Management Systems**

**III B-Tech - 5th Semester**

Logo

Description automatically generated

**Cricket Management System**

**By**

**Group (13)**

Kanulla Venkata Satish Babu (AP20110010004)

Thadimarri Sameer (AP20110010028)

G. Bala Krishna (AP20110010033)

Madire Siva Kumar (AP20110010047)

Kesani Kavya (AP20110010704)

TABLE OF CONTENTS

AIM…………………………………………………………………………..3

INTRODUCTION…………………………………………………………3PROJECT DESCRIPTION……………………………………………5-7

LIST OF ENTITES AND ATTRIBUTES…………………………….7

ER DIAGRAM…………………………………………………………….10

SCHEMA……………………………………………………………………12

NORMALIZATION……………………………………………………13

SQL IMPLMENTATION…………………………………………14-23

CONCLUSION……………………………………………………………24

**Aim:**

Our DBMS project is based on Cricket Management System. It provides various information about the various teams participating in the World Cup, in which all the major countries participate. It also provides us with information about the various players participating in the tournament. The database contains details of players, coaches, and umpires among others. All the useful information about the entire World Cup can have found here.

**Introduction:**

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update, and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked, and modified and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

**Description**

**Entities:**

1. **Team** is an entity type which has many attributes like Team Name which uses the data type varchar. Every team has been given a Team ID which is the primary key which is of data type varchar. Team Ranking, Number of Batsmen and Number of Bowlers are of the data type number. There is another attribute - Wicketkeeper which is of multivalued type and accepts varchar data type. Primary key cannot have null value.
2. **Players** is an entity type which has an attribute – Player Name which is of the data type varchar. It has a primary key, Player ID, which cannot have null value. It has a foreign key, Team ID which is the primary key of the entity, Team. There is a complex attribute, Number of matches played, which comprises of Number of Test Matches, Number of T20 Matches, Number of World Cup Matches and Number of ODIs.
3. **Batsman** is an entity type which has the attributes – Number of sixes hit, Number of Fours hit, the batting average, and the total runs scored. All of these attributes are of the data type number.
4. **Bowler** is an entity type which has the attribute – type of batsman with varchar data type. It also includes number of wickets and economy which are of the data type number.
5. **Umpire** is an entity type which has the attributes name and country of origin of data type varchar. The primary key of this is Umpire Id which is of varchar data type. It also has an attribute Number of matches of data type number.
6. **Coach** is an entity type with a foreign key, Team ID, which is a primary key of entity type, Team. It has a primary key, Coach ID, of data type varchar. It also has another attribute of data type varchar, Name.
7. **Captain** is an entity type with a primary key, Captain ID of data type varchar. It has two foreign keys, i) Player id from table Players and ii) Team ID from table Team. Number of years of captaincy and Number of wins are also attributes of this table of data type number.
8. **Matches** is an entity type with a primary key, match ID, of varchar data type. It has attributes like Team1 Name, Team2 Name, Stadium, Winner Team and Loser Team of data type varchar. Match date is an attribute which uses the datatype date. Match time is an attribute which is of the data type time.

**Relations:**

**Cricket player plays in team (N-1)**

A cricket player can play in only one team but a team can have many players in it but a team must have players in it. So, the relationship becomes (N-1).

**Coach manages team(1-N)**

Coach can manage a single team, but each team can have many coaches (like batting coach, fielding coach, bowling coach). But it is compulsory for a team to have a coach. So, the relationship is 1-N

**Team plays match(M-N)**

Team can play many matches and a match can be played by two teams. So, the relationship is M-N.

**Matches are umpired by Umpire(M-N)**

An umpire can umpire in many matches and a match can have two umpires. So, the relationship is M-N.

**Team headed by a Captain (1-1)**

A team has 1 captain, and a captain is from single team only. So, the relationship is 1-1

**Functional Requirements:**

1. **VIEWER**

System must allow users to login if they enter the correct login id and password. The users must be able to see the player details of each player in the database. Scores of each match must be visible. Match date and venue should be displayed on the login if the users seek for it.

System should display the complete roster of a team including the captain and the players playing in the top 11 and the current rank of the team. The details of the coach must also be available to the users. Referees and their details are also important as the players and the viewers want to see the best referees managing their team’s match. Each player’s statistics should also be available like total runs, number of matches played etc.

System should display data on each match which has been scored in the duration of the entire tournament. System should allow fixtures to be searched and the date should also be available.

**BASIC ANALOGY:**

* View the website with a browser.
* Login to the website.
* View all teams.
* View all players of a team.
* View all batsmen in the tournament.
* View all bowlers in the tournament.
* View all match reports in a season.
* View statistics of a player (all time).
* View coach details.
* View umpire details.
* View Match details.
* View ranking of each team

**View Player information per match:**

a) Number of matches

b) Total runs

c) Total wickets

**View all match details:**

1. Team 1
2. Team 2
3. Umpire
4. Winner
5. Date
6. Time
7. Stadium
8. Rank of teams after match
9. **ADMINISTRATOR**

Administrator is in charge of creating the website which is used to access the database. Administrator has all the privileges of the user but has the authority to add and remove data from the database which the user cannot do.

Administrator is responsible for creating different user accounts and assigning the id and password. Administrators are the one who generate the fixtures and update them in the database. They should be allowed to enter the team’s name of home and away teams. He should have the authority to enter and modify the match details like time and venue in case the need to be changed.

If any player has been punished for bad behaviour or other reasons and cannot play in the World Cup anymore the administrator should be able to delete the data from the database. The rank of every team must keep being modified after each match. After a team is eliminated or disqualified the administrator should be able to delete the entire team’s record.

**BASIC ANALOGY:**

* + Create website.
  + Generate login ID for viewer.
  + Design website.
  + Display different menus.
  + Create World Cup.
  + Display Team Name.
  + Display Team Captain.
  + Display Team Squad.

**View Player information per match:**

a) Number of matches

b) Total runs

c) Total wickets

**REMOVAL OF OLD DATA:**

* 1. If any team gets disqualified, then their data needs to be removed from the database.
  2. If a player gets injured during the World Cup and is unable to play further, then their data needs to be removed from the database.
  3. If any match gets cancelled due to unforeseen circumstances, then the particular match details should be removed.

**MODIFICATION OF DATA:**

1. After every match the existing ranks of every team should be modified.
2. After every match, the statistics of every player should be updated.

iii. Due to unfavourable weather conditions, a match might get delayed. Hence, the match timings need to be changed.

**RETRIEVAL OF DATA:**

***i)* View information of every Team:**

Before the start of a new match, we have to retrieve the Team record like:

1. Team Name
2. Number of Batsman
3. Number of Bowlers
4. Wicketkeeper
5. Number of Wins
6. Number of Losses
7. Names of Players

***ii)* View information of every Match:**

After every match, we have to retrieve the Match details like:

1. First Team Name
2. Second Team Name
3. Umpire
4. Winner
5. Loser

***iii)* View Score Board:**

After every match, we have to retrieve the ranking order of teams:

1. Rank of each team
2. Team name

***iv)* View the captain of each team:**

During the toss, we need to retrieve the data of the captain

1. Name of captain
2. Number of wins under his captaincy
3. Years of captaincy

Entity Relation (ER) MODEL

N

M

MATCHES

plays

TEAM

M

Umpired by

1

Mentored

by

N

UMPIRE

N

1

Headed by

1

COACH

1

CAPTAIN

has

N

PLAYERS

BOWLER

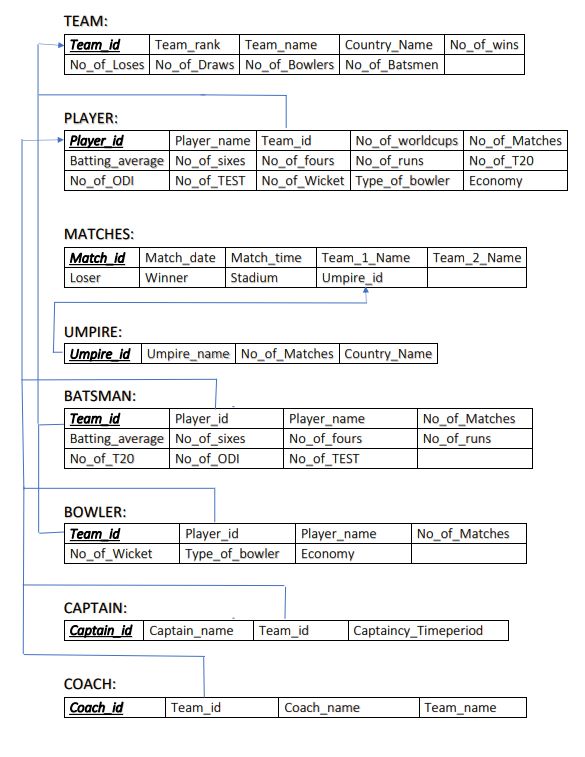
BATSMAN

TABLE DESCRIPTION:

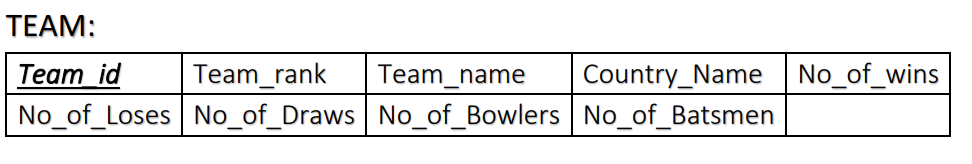
**TABLES**

* + Team
  + Umpire
  + Player
  + Coach
  + Captain
  + Matches

SCHEMA REPRESENTATION:



**NORMALIZATION:**



Candidate key set = {Team\_id}

Non-key Attributes set = {Team\_rank, Team \_name, Country\_Name, No\_of\_wins, No\_of\_Loses, No\_of\_Draws, No\_of\_Bowlers, No\_of\_Batsmen}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “TEAM” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “TEAM” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

Since, Table “TEAM” satisfies the above two conditions, we can say that Table “TEAM” is already in 2nd Normal Form.

**3rd Normal Form:**

* The table “TEAM” is in 2nd Normal Form.

The table “TEAM” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get No\_of\_wins, No\_of\_losses using Team\_name which is a non-prime attribute.

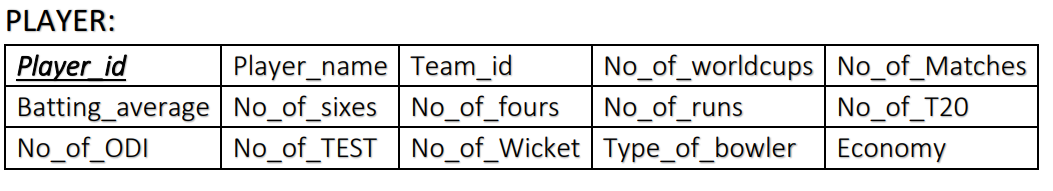
So, to remove transitive dependency, we decompose the table “TEAM”.

Decomposed tables are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Team\_id*** | Team\_rank | Team\_name | Country\_Name |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Team\_id*** | No\_of\_wins | No\_of\_Loses | No\_of\_Draws | No\_of\_Bowlers | No\_of\_Batsmen |

Now, the table “TEAM” is in 3rd Normal Form.



Candidate key set = {Player\_id, Team\_id}

Non-key Attributes set = {Player\_name, No\_of\_worldcups, No\_of\_Matches, Batting\_average, No\_of\_T20, No\_of\_ODI, No\_of\_TEST, No\_of\_Wicket, No\_of\_sixes, No\_of\_fours, No\_of\_runs, Type\_of\_bowler, Economy}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “PLAYER” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “PLAYER” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

There is partial dependency in this table, that is, any non-prime attribute’s values can be retrieved using any one of the attribute from the candidate key set.

Eg: Non-prime attributes like No\_of\_worldcups, No\_of\_matches etc can be retrieved using either Player\_id or Team\_id.

So, to remove Partial dependency, we decompose the table “PLAYER”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Player\_id*** | Team\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | No\_of\_worldcups | No\_of\_Matches | Player\_name |
| Batting\_average | No\_of\_sixes | No\_of\_fours | No\_of\_runs |
| No\_of\_T20 | No\_of\_ODI | No\_of\_TEST | No\_of\_wickets |
| Type\_of\_bowlers | Economy |  |  |

Now, the table “PLAYER” is in 2nd Normal Form.

**3rd Normal Form:**

* The table “PLAYER” is in 2nd Normal Form.

The table “PLAYER” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get No\_of\_runs using No\_of\_sixes or No\_of\_fours which are a non-prime attributes.

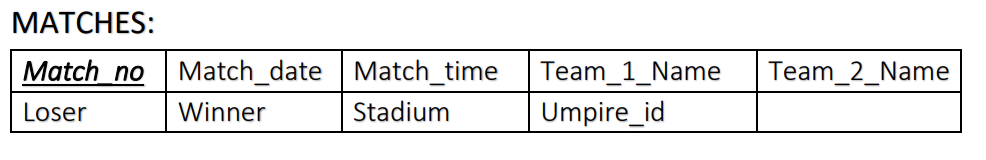
So, to remove transitive dependency, we decompose the table “PLAYER”.

Decomposed tables are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | No\_of\_Matches | Player\_name | No\_of\_runs |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | No\_of\_worldcups | Batting\_average | No\_of\_sixes |
| No\_of\_fours | No\_of\_T20 | No\_of\_ODI | No\_of\_TEST |
| No\_of\_wickets | Type\_of\_Bowler | Economy |  |

Now, the table “PLAYER” is in 3rd Normal Form.



Candidate key set = {Match\_id}

Non-key Attributes set = {Match\_date, Match\_Time, Team\_1\_Name, Team\_2\_Name, Loser, Winner, Stadium, Umpire\_id}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “MATCHES” is already in 1st Normal Form.

**2nd Normal Form:**

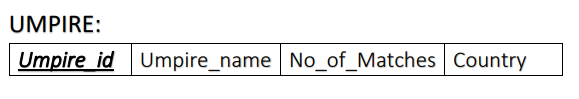
* The table “MATCHES” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

Since, Table “MATCHES” satisfies the above two conditions, we can say that Table “MATCHES” is already in 2nd Normal Form.

**3rd Normal Form:**

* The table “MATCHES” is in 2nd Normal Form.
* The table “MATCHES” should not have transitive dependency which means any non-prime attribute should not be dependent on any other non-prime attribute.

By looking at the table “MATCHES”, we can say that table “MATCHES” is in 2nd normal form and has no transitive dependency, we can say that table “MATCHES” is in 3rd Normal Form.



Candidate key set = {Umpire\_id}

Non-key Attributes set = {Umpire\_name, No\_of\_matches, Country}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “UMPIRE” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “UMPIRE” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

Since, Table “UMPIRE” satisfies the above two conditions, we can say that Table “UMPIRE” is already in 2nd Normal Form.

**3rd Normal Form:**

* The table “UMPIRE” is in 2nd Normal Form.

The table “UMPIRE” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get No\_of\_Matches using Umpire\_name which is a non-prime attribute.

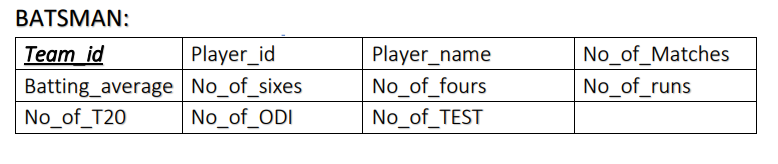
So, to remove transitive dependency, we decompose the table “UMPIRE”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Umpire\_id*** | Umpire\_name |

|  |  |  |
| --- | --- | --- |
| ***Umpire\_id*** | No\_of\_Matches | Country |

Now, the table “UMPIRE” is in 3rd Normal Form.



The table “BATSMAN” is derived from the table “PLAYER”. This process in DBMS is called Specialisation.

Candidate key set = {Team\_id, Player\_id}

Non-key Attributes set = {Player\_name, No\_of\_Matches, Batting\_average, No\_of\_T20, No\_of\_ODI, No\_of\_TEST, No\_of\_sixes, No\_of\_fours, No\_of\_runs}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “BATSMAN” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “BATSMAN” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

There is partial dependency in this table, that is, any non-prime attribute’s values can be retrieved using any one of the attributes from the candidate key set.

Eg: Non-prime attributes like No\_of\_matches etc can be retrieved using either Player\_id or Team\_id.

So, to remove Partial dependency, we decompose the table “BATSMAN”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Player\_id*** | Team\_id |

|  |  |  |
| --- | --- | --- |
| ***Player\_id*** | No\_of\_Matches | Player\_name |
| Batting\_average | No\_of\_fours | No\_of\_runs |
| No\_of\_T20 | No\_of\_TEST | No\_of\_sixes |
| No\_of\_ODI |  |  |

Now, the table “BATSMAN” is in 2nd Normal Form.

**3rd Normal Form:**

* The table “BATSMAN” is in 2nd Normal Form.

The table “BATSMAN” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get No\_of\_runs using No\_of\_sixes or No\_of\_fours which are a non-prime attributes.

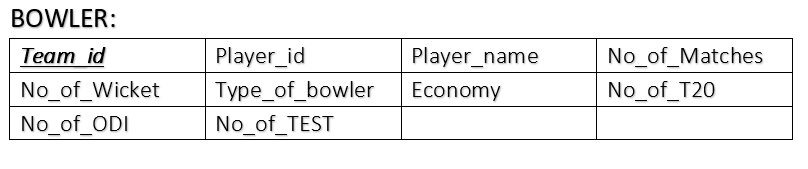
So, to remove transitive dependency, we decompose the table “BOWLER”.

Decomposed tables are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | No\_of\_Matches | Player\_name | No\_of\_runs |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | Batting\_average | No\_of\_sixes | No\_of\_fours |
| No\_of\_T20 | No\_of\_ODI | No\_of\_TEST |  |

Now, the table “BATSMAN” is in 3rd Normal Form.



The table “BOWLER” is derived from the table “PLAYER”. This process in DBMS is called Specialisation.

Candidate key set = {Team\_id, Player\_id}

Non-key Attributes set = {Player\_name, No\_of\_Matches, , No\_of\_T20, No\_of\_ODI, No\_of\_TEST, No\_of\_Wicket, Type\_of\_bowler, Economy}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “BOWLER” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “BOWLER” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

There is partial dependency in this table, that is, any non-prime attribute’s values can be retrieved using any one of the attributes from the candidate key set.

Eg: Non-prime attributes like No\_of\_matches etc can be retrieved using either Player\_id or Team\_id.

So, to remove Partial dependency, we decompose the table “BOWLER”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Player\_id*** | Team\_id |

|  |  |  |
| --- | --- | --- |
| ***Player\_id*** | No\_of\_Matches | Player\_name |
| No\_of\_Wicket | Type\_of\_bowler | Economy |
| No\_of\_T20 | No\_of\_TEST | No\_of\_sixes |
| No\_of\_ODI |  |  |

Now, the table “BOWLER” is in 2nd Normal Form.

**3rd Normal Form:**

* The table “BOWLER” is in 2nd Normal Form.

The table “BOWLER” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get No\_of\_Matches using No\_of\_T20, No\_of\_ODI, No\_of\_TEST which are a non-prime attributes.

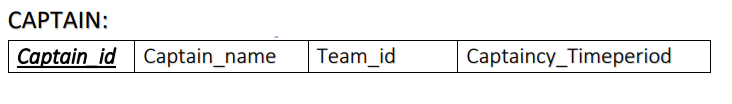
So, to remove transitive dependency, we decompose the table “BOWLER”.

Decomposed tables are shown below:

|  |  |  |
| --- | --- | --- |
| ***Player\_id*** | No\_of\_Matches | Player\_name |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Player\_id*** | No\_of\_Wicket | Type\_of\_bowler | Economy |
| No\_of\_T20 | No\_of\_ODI | No\_of\_TEST |  |

Now, the table “BOWLER” is in 3rd Normal Form.



Candidate key set = {Captain\_id, Team\_id}

Non-key Attributes set = {Captain\_name, Captaincy\_Timeperiod}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “CAPTAIN” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “CAPTAIN” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

There is partial dependency in this table, that is, any non-prime attribute’s values can be retrieved using any one of the attributes from the candidate key set.

Eg: Non-prime attributes like Captaincy\_Timeperiod can be retrieved using either Captain\_id or Team\_id.

So, to remove Partial dependency, we decompose the table “CAPTAIN”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Captain\_id*** | Team\_id |

|  |  |  |
| --- | --- | --- |
| ***Captain\_id*** | Captain\_name | Captaincy\_Timeperiod |

Now, the table “CAPTAIN” is in 2nd Normal Form.

**3rd Normal Form:**

* The table “CAPTAIN” is in 2nd Normal Form.

The table “CAPTAIN” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get Captaincy\_Timeperiod using Captain\_name which is a non-prime attribute.

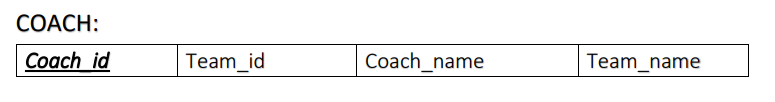
So, to remove transitive dependency, we decompose the table “CAPTAIN”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Captain\_id*** | Captain\_name |

|  |  |
| --- | --- |
| ***Captain\_id*** | Captaincy\_Timeperiod |

Now, the table “CAPTAIN” is in 3rd Normal Form.



Candidate key set = {Coach\_id, Team\_id}

Non-key Attributes set = {Coach\_name, Team\_name}

**1st Normal Form:**

The table has no multivalued attributes; hence, we can say that table “COACH” is already in 1st Normal Form.

**2nd Normal Form:**

* The table “COACH” is in 1st Normal Form.
* Every non-prime attribute should be fully dependent on the candidate key set.

There is partial dependency in this table, that is, any non-prime attribute’s values can be retrieved using any one of the attributes from the candidate key set.

Eg: Non-prime attributes like Team\_name can be retrieved using either Coach\_id or Team\_id.

So, to remove Partial dependency, we decompose the table “COACH”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Coach\_id*** | Team\_id |

|  |  |  |
| --- | --- | --- |
| ***Coach\_id*** | Coach\_name | Team\_name |

Now, the table “COACH” is in 2nd Normal Form.

**3rd Normal Form:**

* The table “COACH” is in 2nd Normal Form.

The table “COACH” has transitive dependency that is, we can get the details of non-prime attributes using other non-prime attributes as well.

Eg: we can get Team\_name using Coach\_name which is a non-prime attribute.

So, to remove transitive dependency, we decompose the table “COACH”.

Decomposed tables are shown below:

|  |  |
| --- | --- |
| ***Coach\_id*** | Couch\_name |

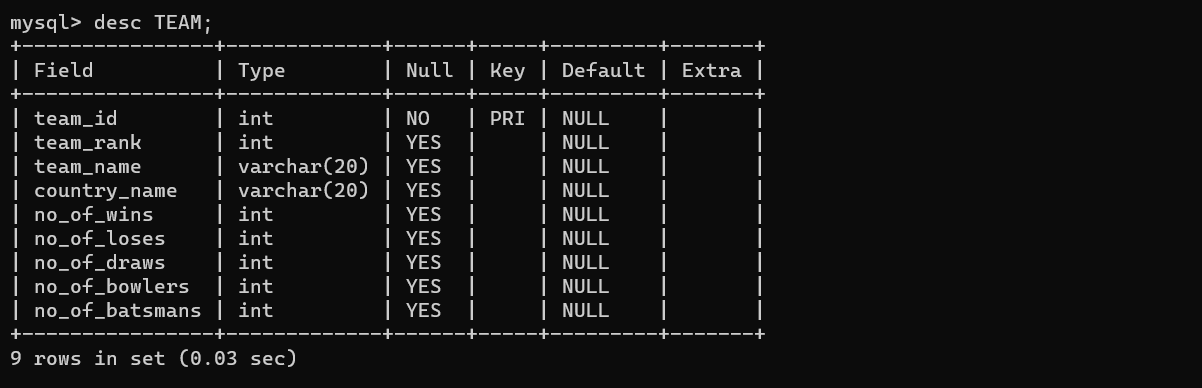
|  |  |
| --- | --- |
| ***Coach\_id*** | Team\_name |

Now, the table “COACH” is in 3rd Normal Form.

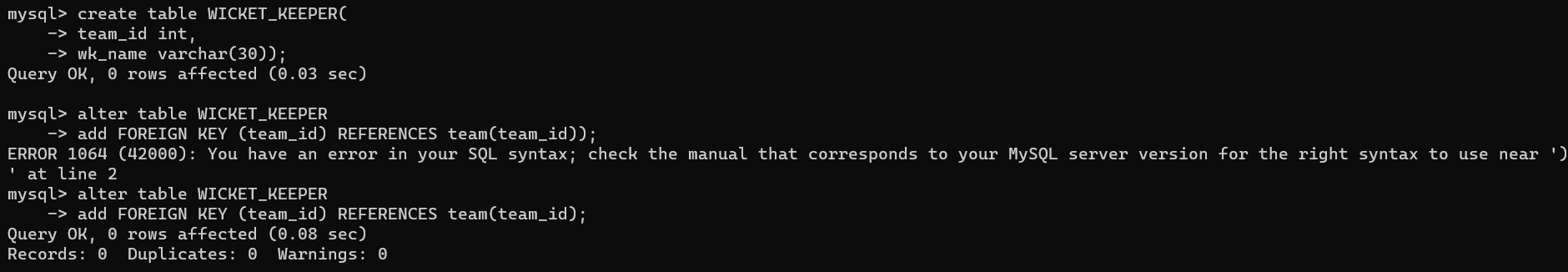
**SQL:**

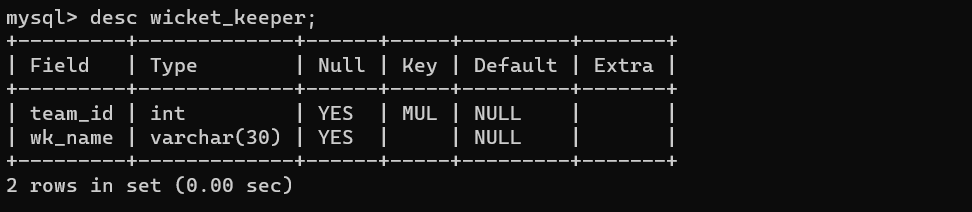
**TABLE TEAM:**



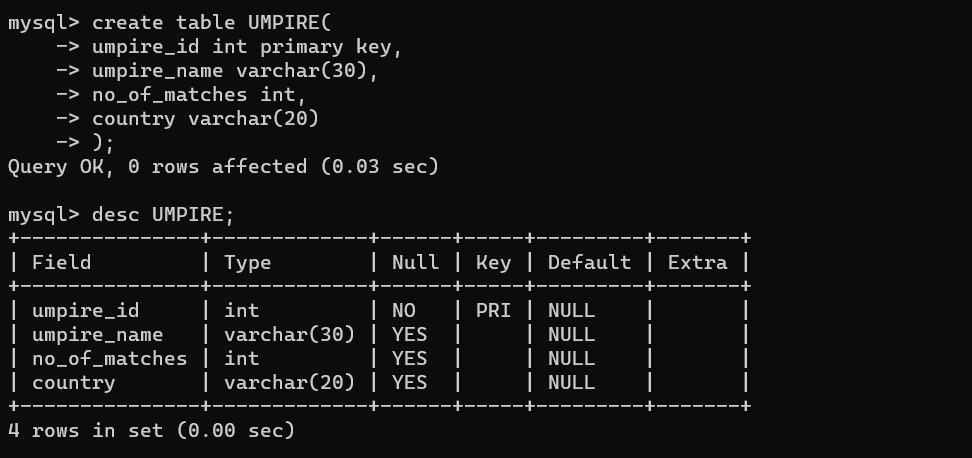


**TABLE WICKET\_KEEPER:**



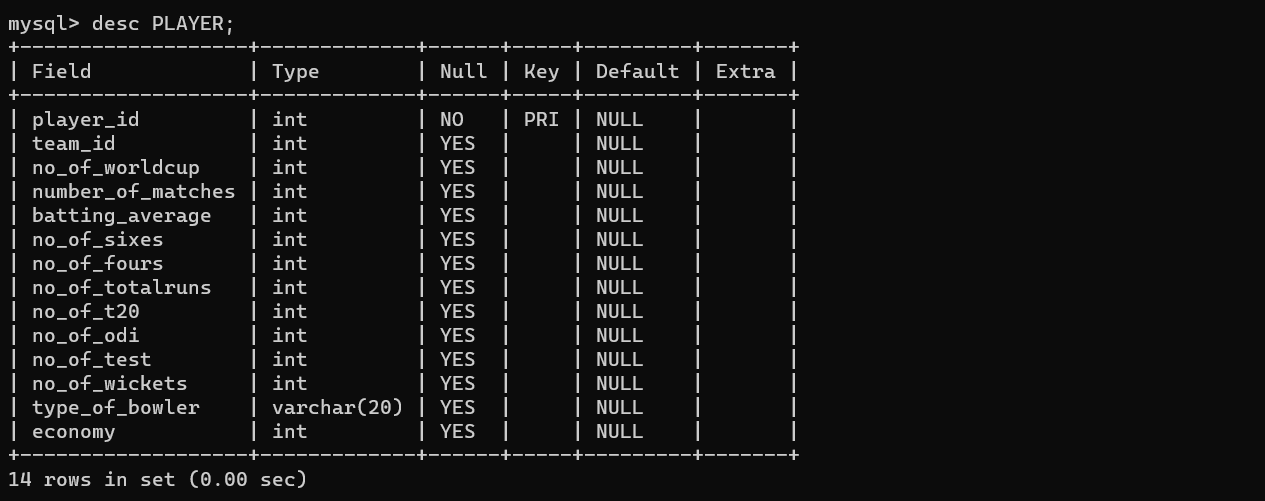


**TABLE UMPIRE:**

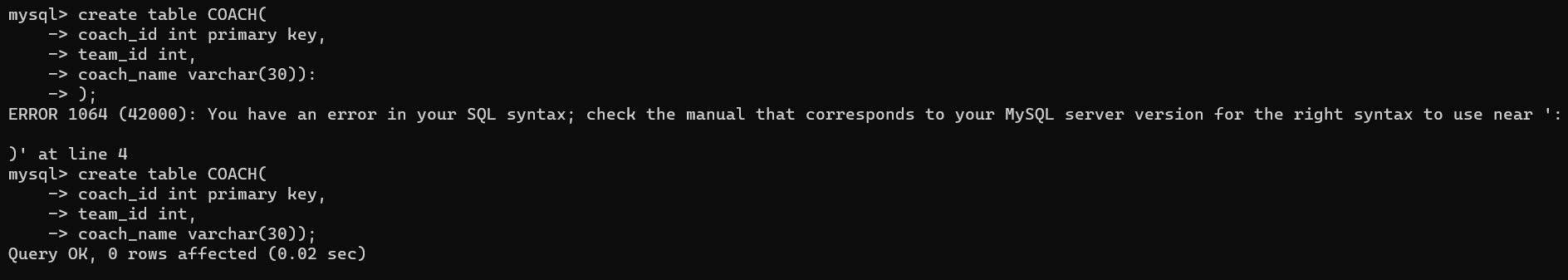


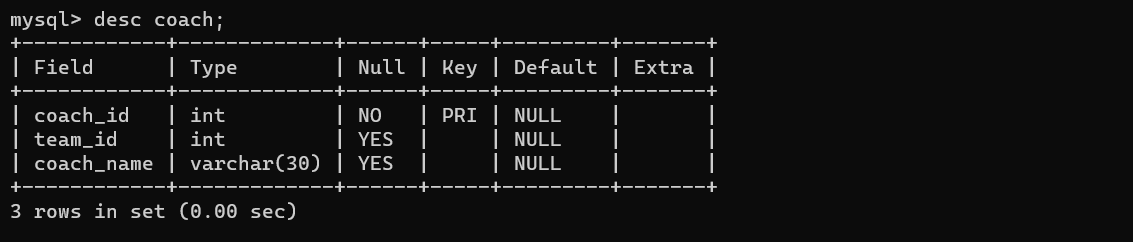
**TABLE PLAYER:**



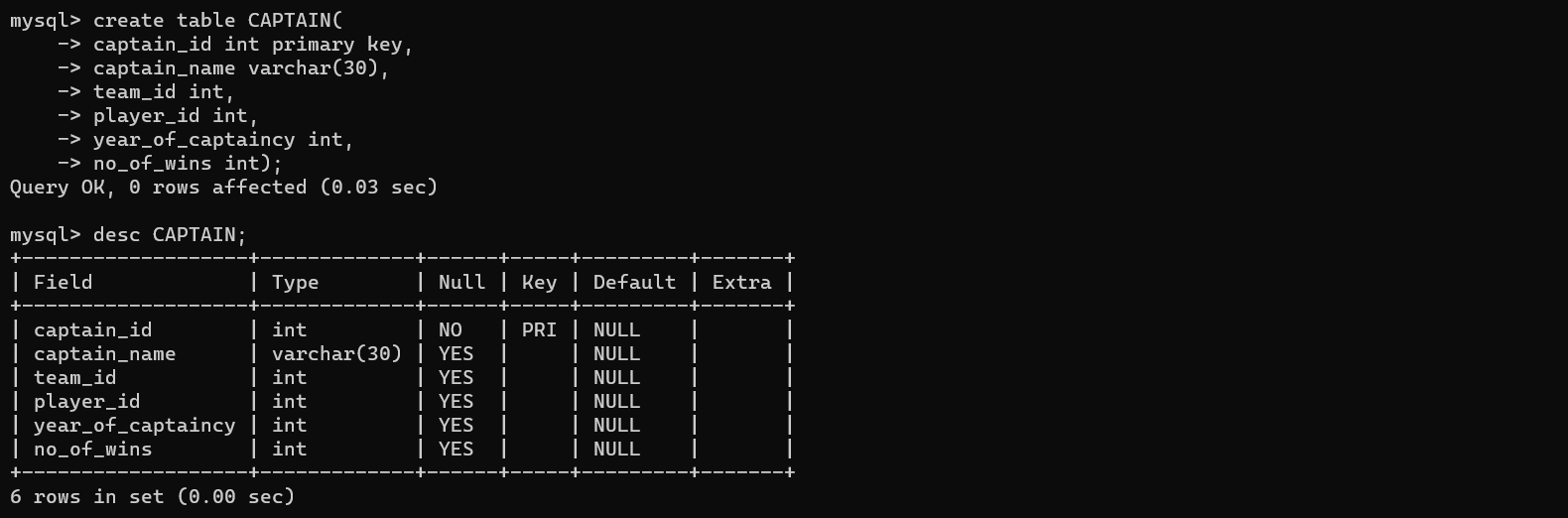


**TABLE COACH:**

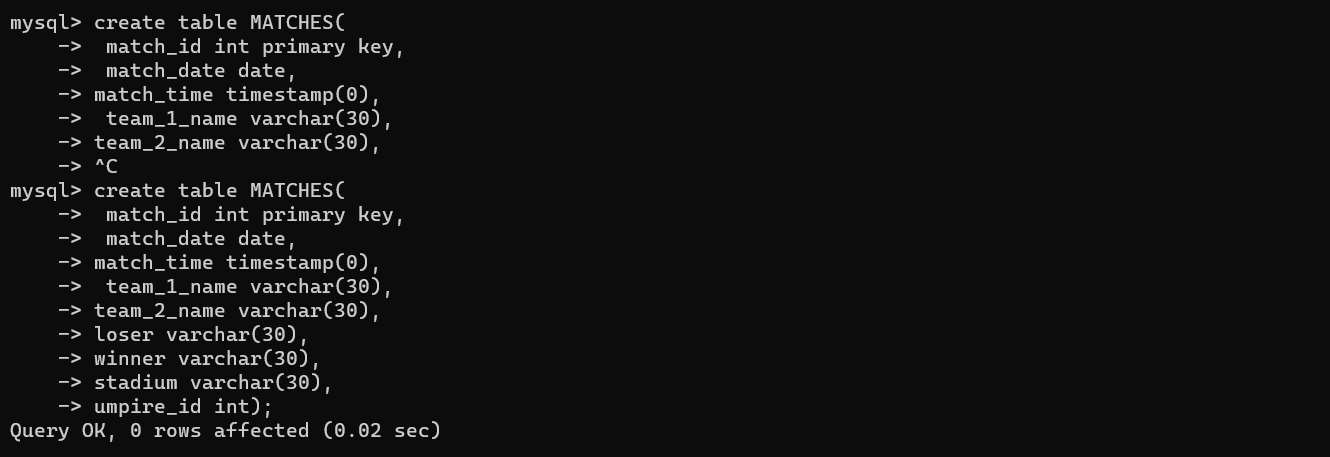




**TABLE CAPTAIN:**

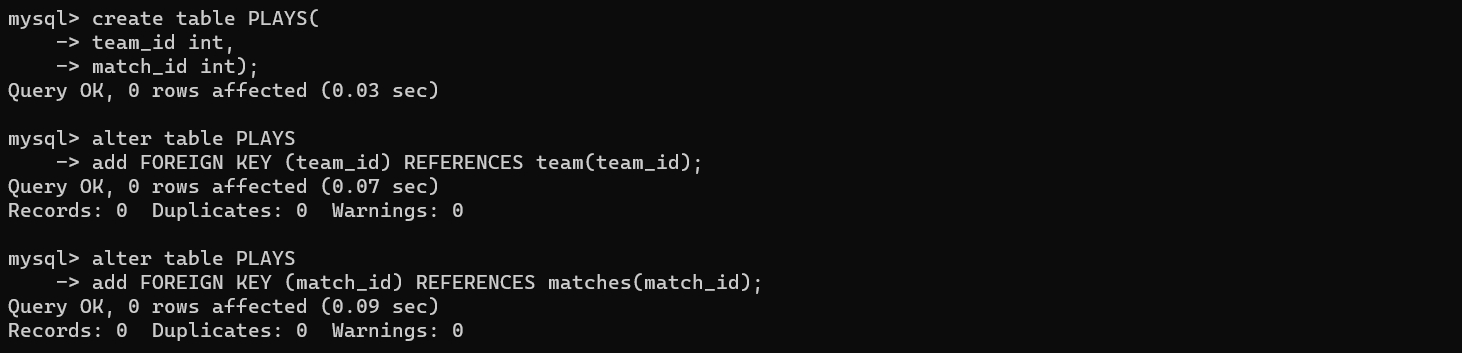


**TABLE MATCHES:**



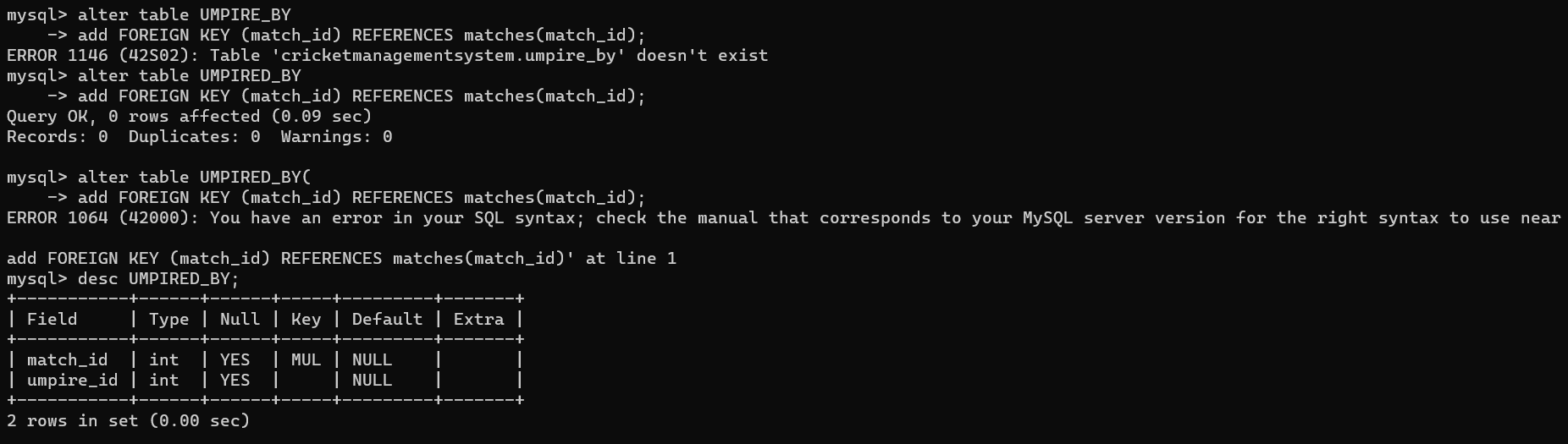


**TABLE PLAYS:**





**TABLE UMPIRED\_BY:**



**CODE TO INSERT VALUES TO TABLE:**

**TABLE TEAM:**

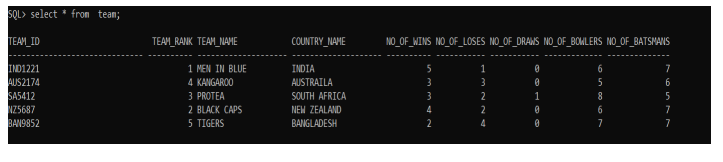
insert into team values ('IND1221', 1, 'MEN IN BLUE','INDIA', 5, 1, 0, 6, 7);

Insert into team values ('AUS2174', 4, 'KANGAROO','AUSTRAILA', 3, 3, 0, 5, 6);

Insert into team values ('SA5412', 3, 'PROTEA','SOUTH AFRICA', 3, 2, 1, 8, 5);

Insert into team values ('NZ5687', 2, 'BLACK CAPS','NEW ZEALAND', 4, 2, 0, 6, 7);

Insert into team values ('BAN9852', 5, 'TIGERS','BANGLADESH', 2, 4, 0, 7, 7);

****

**UMPIRE VALUES:**

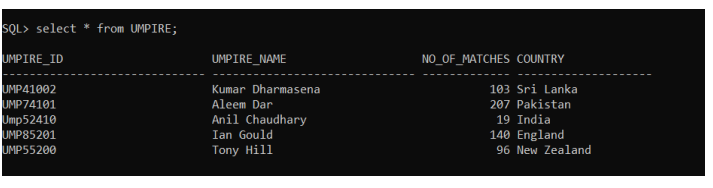
insert into UMPIRE values ( 'UMP41002', 'Kumar Dharmasena', 103, 'Sri Lanka' );

insert into UMPIRE values ( 'UMP74101', 'Aleem Dar', 207, 'Pakistan' );

insert into UMPIRE values ( 'Ump52410', 'Anil Chaudhary', 19, 'India' );

insert into UMPIRE values ( 'UMP85201', 'Ian Gould', 140, 'England' );

insert into UMPIRE values ( 'UMP55200', 'Tony Hill', 96, 'New Zealand' );

****

**COACH VALUES:**

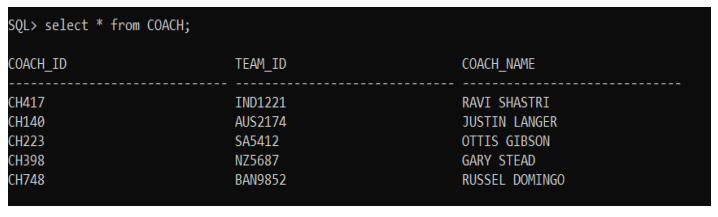
insert into COACH values( 'CH417', 'IND1221', 'RAVI SHASTRI' );

insert into COACH values( 'CH140', 'AUS2174', 'JUSTIN LANGER' );

insert into COACH values( 'CH223', 'SA5412', 'OTTIS GIBSON' );

insert into COACH values( 'CH398', 'NZ5687', 'GARY STEAD' );

insert into COACH values( 'CH748', 'BAN9852', 'RUSSEL DOMINGO' );

****

**CAPTAIN VALUES:**

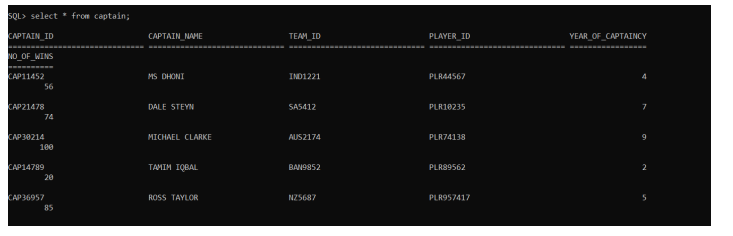
insert into captain values( 'CAP11452', 'MS DHONI', 'IND1221','PLR44567', 4,56 );

insert into captain values( 'CAP21478', 'DALE STEYN', 'SA5412','PLR10235', 7,74 );

insert into captain values( 'CAP30214', 'MICHAEL CLARKE', 'AUS2174','PLR74138', 9,100 );

insert into captain values( 'CAP14789', 'TAMIM IQBAL', 'BAN9852','PLR89562',2,20 );

insert into captain values( 'CAP36957', 'ROSS TAYLOR', 'NZ5687','PLR957417', 5,85 );

****

**MATCHES VALUES:**

insert into MATCHES values( 'MAT101',to\_date('12-03-2011','dd-mmyyyy'),to\_timestamp('15:30','hh24:mi'),'India','Bangladesh','Bangladesh','India','Feroz Shah Kotla','UMP55200' );

insert into MATCHES values( 'MAT201',to\_date('15-03-2011','dd-mmyyyy'),to\_timestamp('9:30','hh24:mi'),'England','Australia','England','Australia','Eden Gardens','UMP41002' );

insert into MATCHES values( 'MAT301',to\_date('21-03-2011','dd-mm-yyyy'),to\_timestamp('11:30','hh24:mi'),'Sri Lanka','Bangladesh','Bangladesh','Sri Lanka','M.A. Chidambaram','UMP74101' );

insert into MATCHES values( 'MAT401',to\_date('23-03-2011','dd-mm-yyyy'),to\_timestamp('15:30','hh24:mi'),'New Zealand','South Africa','South Africa','New Zealand','Sardar Patel','UMP85201' );

insert into MATCHES values( 'MAT501',to\_date('26-03-2011','dd-mmyyyy'),to\_timestamp('8:30','hh24:mi'),'England','India','England','India','Wankhede','Ump524 10' );

****

**PLAYER VALUES:**

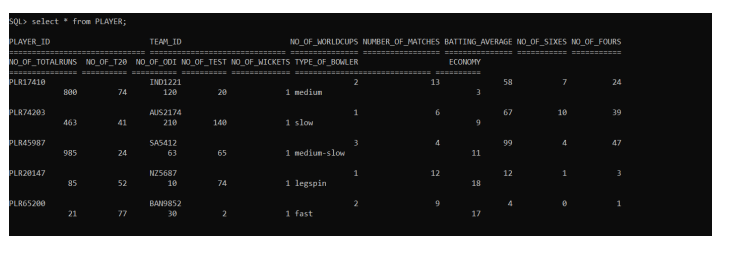
insert into PLAYER values ( 'PLR17410', 'IND1221', 2, 13, 58,7,24,800,74,120,20,1,'medium',3.2 );

insert into PLAYER values ( 'PLR74203', 'AUS2174', 1, 6, 67, 10,39,463,41,210,140 ,1,'slow',8.5 );

insert into PLAYER values ( 'PLR45987', 'SA5412', 3, 4, 99, 4, 47,985, 24,63,65,1,'medium-slow',11.2 );

insert into PLAYER values ( 'PLR20147', 'NZ5687', 1, 12, 12, 1, 3,85, 52 ,10,74,1,'legspin',18.3 );

insert into PLAYER values ( 'PLR65200', 'BAN9852', 2, 9, 4, 0,1 ,21,77, 30,2,1,'fast',17.3 );

****

UMPIRED\_BY VALUES:

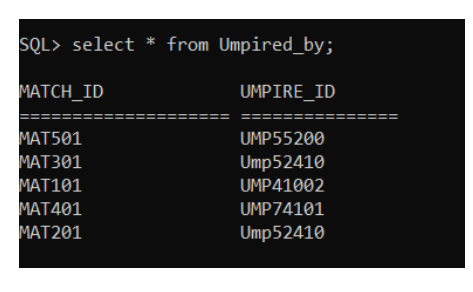
insert into Umpired\_by values( 'MAT501','UMP55200' );

insert into Umpired\_by values( 'MAT301','Ump52410' );

insert into Umpired\_by values( 'MAT101','UMP41002' );

insert into Umpired\_by values( 'MAT401','UMP74101' );

insert into Umpired\_by values( 'MAT201','Ump52410' );

****

**PLAYS VALUES:**

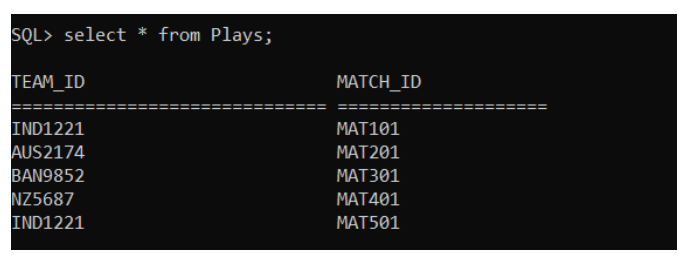
insert into Plays values( 'IND1221','MAT101' );

insert into Plays values( 'AUS2174','MAT201' );

insert into Plays values( 'BAN9852','MAT301' );

insert into Plays values( 'NZ5687','MAT401' );

insert into Plays values( 'IND1221','MAT501' );

****

**WICKET\_KEEPER VALUES:**

insert into WICKET\_KEEPER values( 'IND1221','MS Dhoni' );

insert into WICKET\_KEEPER values( 'IND1221','Dinesh Kartik' );

insert into WICKET\_KEEPER values( 'AUS2174','Tim Lee' );

insert into WICKET\_KEEPER values( 'AUS2174','Peter Hegward' );

insert into WICKET\_KEEPER values( 'AUS2174','Hefer Kingsly' );

****

**Conclusion:**

Cricket management system allows the user to check the profile of any player as the DBMS helps in storing and retrieving the data from the databases. Many operations can be performed from the data available on the database such as analysing the form of a player in recent, past times. It shows how well the player is been with his/her game throughout the selected time period. In real world, during IPL Auctions, Many analysts use data from databases and perform analysis and come to conclusion whether the player is fit for their franchise before actually buying the player in the auction.