IPL cricket database

Abstract:

The motive of this assignment is to build a physical model for IPL cricket database. Raw data is collected from Kaggle as a CSV format and data is audited for null values, cleaned and reformatted to fit the conceptual model.

Data extracted from the Source

1. Match - Match ID, Teams played , Date of match , Season , venue, City name , Country , Toss winner, Match winner , Toss Decision, win type , Outcome, Man of the match ,Win margin
2. Player Role - Match ID, Player ID , Player Name , Date of Birth , batting Hand , Bowling Skill , Country name , Role Description
3. Ball by Ball details - Match ID , Innings ID, Over ID, Ball ID, Runs scored, Extra type, Out type
4. Player Details - Player ID, Player Name, Country
5. Team - Team ID, Team Name

Normalization of Dataset:

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data. In this dataset we are going to perform first three stages of normalization:

1) First normal form:

The dataset that we have downloaded is already in the first normal form because every table has a unique primary key and each attribute contain only atomic values. The ER form of the first normal form is shown below

In [11]:
```
from PIL import Image
from IPython.display import display
display(Image.open('1st Normal Form.png'))
```

2) Second normal form:

In the second normal form , primary key in the role description table is Match ID and player ID but the attributes Player name, Date of birth , Batting hand , Bowling skill is partially dependent on the player ID. So these attributes are moved to the Player table.

The ER form of the first normal form is shown below :

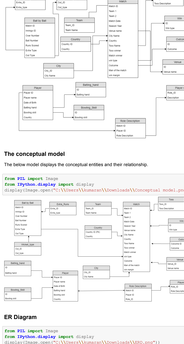
In [14]:
```
from PIL import Image
from IPython.display import display
display(Image.open('2nd Normal Form.png'))
```

3) Third normal form

In the third normal form, the attributes which has transitive functional dependencies are divided.

The ER form of the third normal form is shown below

In [15]:
```
from PIL import Image
from IPython.display import display
display(Image.open('3rd Normal Form.png'))
```



The conceptual model

The below model displays the conceptual entities and their relationship.

In [3]:
```
from PIL import Image
from IPython.display import display
display(Image.open("C:\\Users\\Ammaran\\Downloads\\Conceptual model.png"))
```

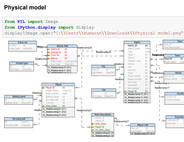

ER Diagram

In [4]:
```
from PIL import Image
from IPython.display import display
display(Image.open("C:\\Users\\Ammaran\\Downloads\\ERD.png"))
```

Relationship between the entities

Physical model

In [5]:
```
from PIL import Image
from IPython.display import display
display(Image.open("C:\\Users\\Ammaran\\Downloads\\Physical model.png"))
```



Questions about the Conceptual model

1) What are the ranges, data types and format of all of the attributes in your entities?

1. Ball by Ball:

Match ID : INT , Innings ID : INT , Over number : INT , Ball number : INT , Runs scored : INT , Extra_type : INT , Out_type : INT ,

2. Extra_Runs

Extra_ID : INT , Extra_type : VarChar(45)

3. Wicket_type

Out_ID : INT , Out_type :VarChar(45)

4. Match

Match ID : INT , Team 1 :Varchar(45) , Team 2 : Varchar(45) , Match Date : DATE , Season year INT , Venue name INT ,City name INT , Country INT , Toss name INT ,Toss Winner INT , Match winner INT , Win type INT ,Outcome INT ,

Win margin INT

5. Toss

Toss_ID : INT , Toss Description : Char(45)

6. Win

ID : INT , Win type :CHar(45)

7. Outcome

ID : INT , Outcome :char(45)

8. Venue

ID : INT ,

Venue name : char(45)

9. Team

Team_ID : INT , Team name : char(45)

10. Country

Country ID : INT , Country : char(45)

11. City

City_ID : INT , City name : char(45)

12. Player

Player ID : INT , Player name : char(45) , Date of birth : DATE , Batting hand : INT , Bowling skill : INT , Country : INT ,

13. Batting hand

ID : INT , Batting hand : char(45) ,

14. Bowling_Skill

ID : INT , Bowling skill : char(45)

15. Role_Description

Match ID : INT , Player ID : char(45) , Role Description : char(45) ,

16. Player_Role

Role_ID : INT , Description : char(45)

2) When should you use an entity versus attribute? (Example: address of a person could be modeled as either)

Entity is the object and attribute is the property of the Object. When there are certain characteristic for an object we will be considering that Object as the entity and its characteristic as attribute.

3) When should you use an entity or relationship, and placement of attributes? (Example: a manager could be modeled as either)

Entity is the Object which has characteristic as attributes. Relationship is formed where is an dependency between two entities.

4) How did you choose your keys? Which are unique?

Keys were choosed based on the uniqueness of the attribute which can identify a particular record. in Match table , Match_ID is unique which will identify a particualar record.

5) Did you model hierarchies using the "ISA" design element? Why or why not?

6) Were there design alternatives? What are their tradeoffs: entity vs. attribute, entity vs. relationship, binary vs. ternary relationship?

There should be no design alternative because the data can be linked only with Match_ID and Played_ID.

7) Where are you going find real-world data populate your model?

Real world data was found on Kaggle.

Questions you must answer about your physical model:

1) Are all the tables in 1NF?

Yes, every value in each table is atomic.

Are all the tables in 2NF?

The tables are in 2NF because there are partial dependency

Are all the tables in 3NF?

The tables are in 3NF because transitive dependency are eliminated by splitting the table.

Report

Files used : Match.csv , Ball by Ball.csv , Player details.csv , Player role.csv

The above files was already audited for null values and cleaned. So there is no specific need to perform auditing for null values again.

Normalization was carried out on the dataset to reduce redundancy ,insertion anomaly , deletion anomaly and updation anomaly.

Conclusion

Data downloaded from Kaggle had lots of redundancy. Normalization techniques were carried out to reduce redundancy ,insertion anomaly , deletion anomaly and updation anomaly.Also, Conceptual, physical and ERD diagram for the dataset is created.

Contribution

Self efforts : 50%

External source: 20%

Guidance by the Professor : 10%

Guidance by Teaching Assistant: 20%

Citations

Data source : https://www.kaggle.com/raghu007/ipl-data-till-2017

LICENSE:

In [ ]:
```
```