Milestone 2

CRICKET STATISTICS

TEAM NAME: dmql\_ks

TEAM MEMBERS:

Member 1: Karan Shah UB id: kshah23

Member 2: Saqlain Naveed Ahmed UB id: sahmed34

Problem Statement:

Cricket is a game whose popularity has grown leaps and bounds in the past few decades. Naturally with each passing year the game keeps evolving and has brought much interest in the field of statistical analysis and machine learning. Keeping this in mind we propose to make a database system which consists of statistics related to the three formats of the game namely, Test, One-Day International and Twenty-20 which could be used in the field of research in the game. The volume of data that we are considering for the database is huge as it includes not only the current players but also the players of past generations. To use this large volume of data, the usage of excel sheets would not be efficient and would also become unmanageable. Also, we plan to combine data from multiple tables and gather statistics on them through joins which would have been an arduous task through excel sheets. The database that we propose considers the batting and bowling statistics separately for all the three formats for which we have six tables to represent respectively. We also have a table that only includes the details of the player and the country he represents to tackle redundancy. We propose to run a complex set of queries on the database that involves joins, aggregations and group-by clauses. The data updates happen on each day when the game is played and the records of the players who have played gets updated.

Target User:

The database will primarily be used by cricket enthusiasts and people who want to conduct research in the field of sports or create models through machine learning to predict future outcomes. The database administrator will be an official from cricket’s international governing body ICC (International Cricket Council)

Designing the Database schema:

To represent a relational schema, we first present E/R diagrams

Diagram, schematic

Description automatically generated

Each Entity from the above diagram is represented below with its attributes.

Diagram

Description automatically generated

We have made few changes in primary keys for the following relations:

1.ODI\_Bowling

2.Test\_Bowling

3.T20\_Bowling

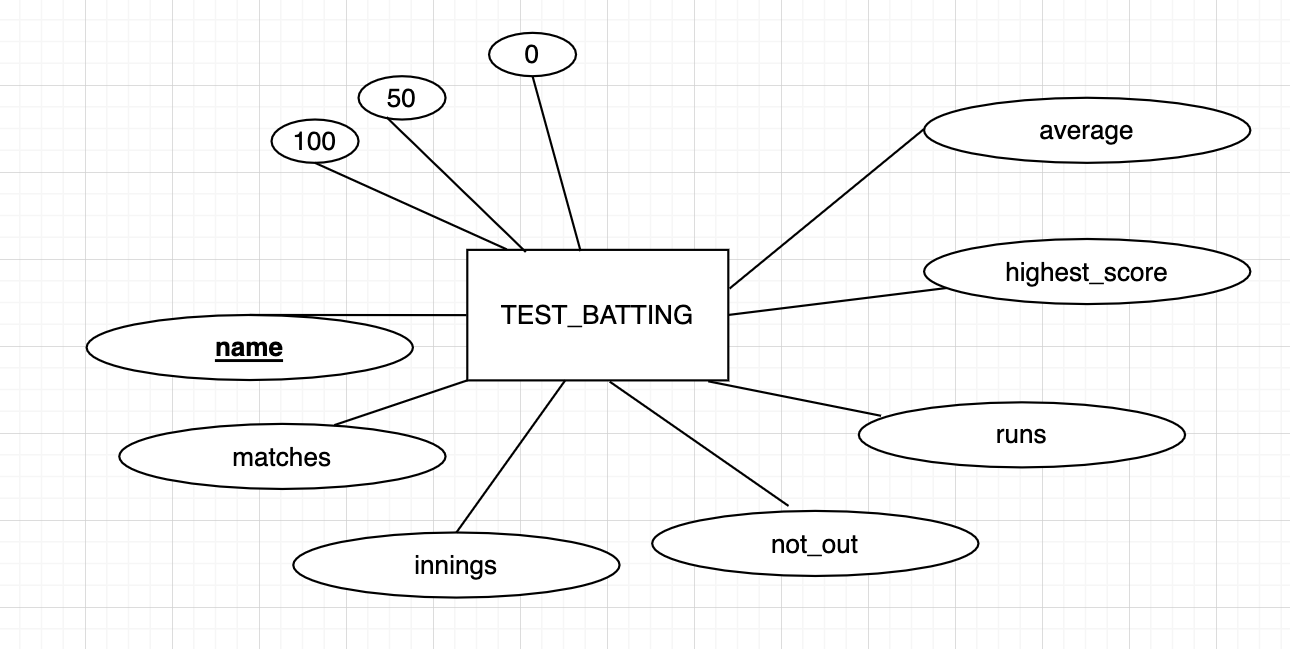
4.ODI\_Batting

5.Test\_Batting

6.T20\_Batting

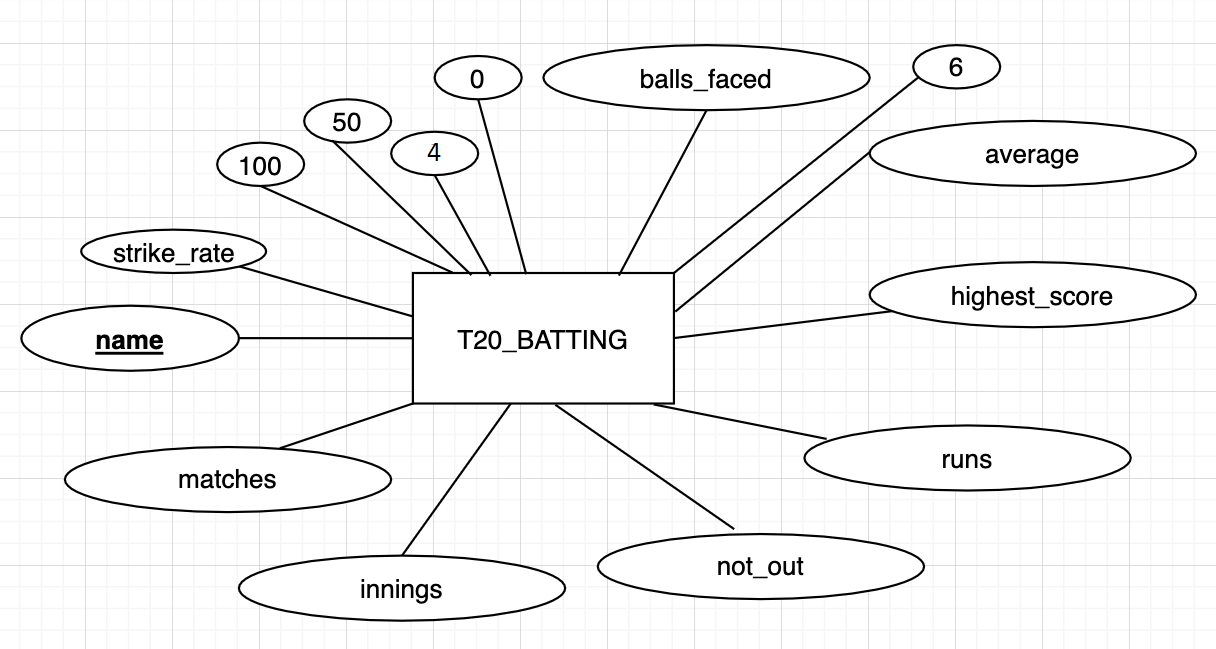
We changed primary key from name to (name. matches) for every above relation.

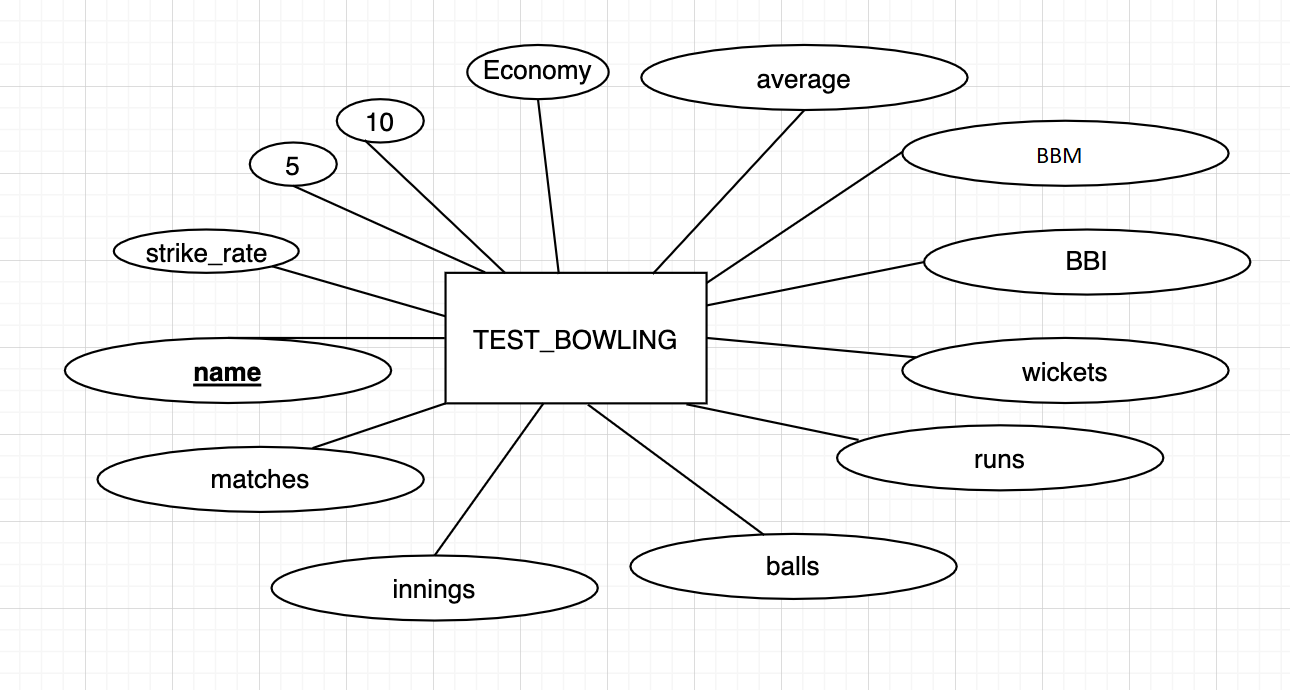
We made this change as we observed many players can have same name. So, name of player and matches together form a robust primary key. As we changed the primary key, following are the changes in ER diagram.



Diagram

Description automatically generated





Diagram

Description automatically generated

Diagram

Description automatically generated

**Relation: Players**

Player table has 2 attributes

1. player

2. country

**Functional dependency for Player table is:**

Player 🡪 country

We do not observe any partial and transitive dependency in above relation.

As player is super key for Player table, the relation is already in BCNF.

**Relation: Bowling\_test**

Bowling\_test has 13 attributes

1. name

2. matches

3. Innings

4. Balls

5. Runs\_conceded

6. wickets

7. BBI

8. BBM

9. Average

10. Economy

11. Strike\_rate

12. five\_wicket\_haul

13. ten\_wicket\_haul

**Functional dependency for Bowling\_test table is:**

Name, matches🡪 innings, balls, runs\_conceded, wickets, BBI, BBM, Average, Economy, Strike\_rate, five\_wicket\_haul, ten\_wicket\_haul.

As (name,matches) is super key for Bowling\_test, the relation is already in BCNF. We do not observe any partial and transitive dependency in above relation.

**Relation: Bowling\_ODI**

Bowling\_ODI has 12 attributes

1. name

2. matches

3. Innings

4. Balls

5. Runs\_conceded

6. wickets

7. BBI

8. Average

9. Economy

10. four\_wicket\_haul

11. five\_wicket\_haul

12. Strike\_rate

**Functional dependency for Bowling\_ODI table is:**

(name,matches)🡪 innings, balls, runs\_conceded, wickets, BBI, Average, Economy, four\_wicket\_haul, five\_wicket\_haul, Strike\_rate.

We do not observe any partial and transitive dependency in above relation.

As (name,matches) is super key for Bowling\_ODI, the relation is already in BCNF.

**Relation: Bowling\_T20**

Bowling\_T20 has 12 attributes

1. name

2. matches

3. Innings

4. Balls

5. Runs\_conceded

6. wickets

7. BBI

8. Average

9. Economy

10. four\_wicket\_haul

11. five\_wicket\_haul

12. Strike\_rate

**Functional dependency for Bowling\_T20 table is:**

(name,matches)🡪 matches, innings, balls, runs\_conceded, wickets, BBI, Average, Economy, four\_wicket\_haul, five\_wicket\_haul, Strike\_rate.

We do not observe any partial and transitive dependency in above relation.

As (name,matches) is super key for Bowling\_T20, the relation is already in BCNF.

**Relation: Test\_Batting**

Test\_Batting has 10 attributes

1. name

2. matches

3. Innings

4. Not\_out

5. Runs

6. highest\_score

7. Average

8. Zeroes

9. fifties

10. hundreds

**Functional dependency for Test\_Batting table is:**

(name,matches)🡪 innings, not\_out, runs, highest\_score, average, zeroes, fifties, hundreds

We do not observe any partial and transitive dependency in above relation.

As (name,matches) is super key for Test\_Batting, the relation is already in BCNF.

**Relation: ODI\_Batting**

ODI\_Batting has 12 attributes:

1. name

2. matches

3. Innings

4. Not\_out

5. Runs

6. highest\_score

7. Average

8. balls\_faced

9. Zeroes

10. fifties

11. hundreds

12. Strike\_rate

**Functional dependency for ODI\_Batting table is:**

(name,matches)🡪 innings, not\_out, runs, highest\_score, average,balls\_faced, zeroes, fifties, hundreds,strike\_rate.

We do not observe any partial and transitive dependency in above relation.

As (name,matches) is super key for ODI\_Batting, the relation is already in BCNF.

**Relation: T20\_Batting**

T20\_Batting has 14 attributes:

1. name

2. matches

3. Innings

4. Not\_out

5. Runs

6. highest\_score

7. Average

8. fours

9. sixes

10. balls\_faced

11 Zeroes

12. fifties

13. hundreds

14. Strike\_rate

**Functional dependency for T20\_Batting table is:**

(name,matches)🡪 innings, not\_out, runs, highest\_score, average, fours, sixes, balls\_faced, fifties, hundreds, strike\_rate.

We do not observe any partial and transitive dependency in above relation.

As (name,matches) is super key for T20\_Batting, the relation is already in BCNF.

**Problems faced while handling larger dataset:**

Before loading the data into the table, we cleaned the data.

Some of the data contained redundant data which was of no use. There were attributes with null values too. We replaced those values with zero to maintain the constraint and integrity in database.

We removed the unwanted tuples which were not much informative and were redundant.

At first, we planned to make the player name as primary key but later we found that multiple player can have same name. So, we changed the primary key to (name, matches). This made our primary key robust.