|  |
| --- |
| SRH University of Applied Sciences |
| Advanced Information System Project Cricket use case |
| Using Neo4j Graph database |

|  |
| --- |
| Anusha Nagaraj, Shunmuga Prabhu, Vinay Kumar, Sachin, Manoj  2/17/2017 |

**CRICKET**  
Advanced Information System  
17/01/2017  
*Authors:*

*Vinay Kumar, Anusha Nagaraj, Shunmuga Prabhu  
Sachin, Manoj*

Table of Contents

[1 INTRODUCTION 2](#_Toc475084008)

[1.1 Introduction to the project 2](#_Toc475084009)

[1.2 Organisation 2](#_Toc475084010)

[2 PROJECT USE CASE 3](#_Toc475084011)

[2.1 Use Case 3](#_Toc475084012)

[2.2 User Stories 5](#_Toc475084013)

[3 FUNDAMENTALS 5](#_Toc475084014)

[3.1 Technology used 5](#_Toc475084015)

[3.2 Datamodel 6](#_Toc475084016)

[3.3 CRUD operations 6](#_Toc475084017)

[4 PROJECT DATA-MODEL 7](#_Toc475084018)

[4 Implementation 10](#_Toc475084019)

[4.1 Implementation of the data-model 10](#_Toc475084020)

[4.2 Implementation of user-stories and queries 12](#_Toc475084021)

[**4.2.1** **Use Case** 12](#_Toc475084022)

[**4.2.2** **User Stories:-** 19](#_Toc475084023)

[4.3 Queries 22](#_Toc475084024)

[4.3.1 User Stories:- 28](#_Toc475084025)

[5 CONCLUSION 31](#_Toc475084026)

[6 LIST OF REFERENCES 32](#_Toc475084027)

[7 Appendix A 33](#_Toc475084028)

# INTRODUCTION

## Introduction to the project

Cricket is an International Sport, which is played between two teams and each team consists of 11 players. ICC International Cricket Council is the international governing body of Cricket. ICC Conducts the World cup every 4 years, it includes the warm up matches, Knockout stage, Quarter-finals, Semi-finals and Finals. The game contains huge statistics regarding world cups, Teams, Matches and Players contribution to the team.

In this project we are trying to introduce a new online system that can replace administrative paperwork and make huge amount of data available to the users. The users can be players, teams, coaches, media and supporters or fans of cricket. The main Application of the Database is to access high velocity data at high speed to the end user located in any part of the world.

The database will be having history data of each events from match’s played to players involved. This can be used for analysis and can help the teams for taking appropriate decisions for coming matches and for the users and cooperate sectors for different purpose.

Here we are building a database using the NEO4J database.Neo4j is a graph based database i.e. it orders and represents data as a graph. It stores the data in nodes and edges. Neo4j is able to store, process, and query connections efficiently.

## Organisation

The project kick started after the team decided to go ahead with cricket use case. The work was divided equally based on the individual skills. The agenda of the meeting was circulated amongst the team members before hand via WhatsApp and phone calls. Meetings were planned in a way to discuss the work updates and plan the next steps. In order to felicitate the communication within the team we used Google drive and Trello. The data were initially gathered and analyzed using excel.

Task distribution:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **USER  STORIES** | **INSERT** | **QUERIES** | **REPORT** |
| **VINAY** | 2 | **NODES**-2 Team;Refree;Match;Players as per teams assigned **RELATIONSHIPS:**-Player-bowler and batsman as per teams assigned,Player-Team as per team assigned,Team-Match,WorldCup-Match,Stadium-Match,Umpire-Match,Match-Player as per match assigned | 4 | Implementation, Conclusion |
| **ANUSHA  NAGARAJ** | 2 | **NODES:**-2 Teams,Country,Stadium,Players as per teams assigned **REALATIONSHIPS:**-Player-bowler and batsman as per teams assigned,Player-Team as per teams assigned,Country-Stadium,WorldCup-Country,Match-Player as per match assigned | 4 | Implementation and merging of document |
| **PRABHU** | 2 | **NODES:**-WorldCup,Players as per teams assigned **REALATIONSHIPS:**-Player-bowler and batsman as per teams assigned,Player-Team as per teams assigned,Match-Players as per match assigned | 4 | CRUD operations and Project data model |
| **MANOJ** | 2 | **NODES:**-WorldCup,Players as per teams assigned **REALATIONSHIPS:**-Player-bowler and batsman as per teams assigned,Player-Team as per teams assigned,Match-Players as per match assigned | 4 | Introduction and Project Use case |
| **SACHIN** | 2 | **NODES:**-Coach,Players as per teams assigned **REALATIONSHIPS:**-Player-bowler and batsman as per teams assigned,Player-Team as per teams assigned,Match-Players as per match assigned | 4 | Technology used and Data model |

Meetings held:-

|  |  |  |
| --- | --- | --- |
| **DAY** | **HOURS** | **TASK** |
| 27/01/17 | 3 | Merge all User Stories and chose best 5 out of it |
| 30/01/17 | 3 | Data Model Discussion |
| 2/2/2017 | 2 | Created initial Data Model |
| 3/2/2017 | 3 | Data Model enhancement |
| 4/2/2017 | 2 | Data Model enhancement |
| 7/2/2017 | 4 | Merge data to create final data model |
| 14/02/17 | 3 | Merge Queries |
| 16/02/17 | 4 | Report and PPT work |

# PROJECT USE CASE

## Use Case

1. **How many world cups where held? The answer should also provide the information of where they were played, when they were played and who won the cup.**

Requirement:-  
 It is required to find the number world cups held and information about these world cups such as the the place the matches were played and the winner of every worldcup.

1. **Who won the recent world cup? Display all the matches of the winning team including their results.**

Requirement:-

The details of the team who won the recent worldcup is required and the details of match played by them such as match name, date and result.

1. **Which teams were out of quarterfinals in the past 5 world cups? List the teams including against whom they lost and when the match took take place.**

Requirement:-

We need the teams that were out of quarter finals and the details of those teams such as against whom they lost and when the match took place.

1. **List the teams whose captain has changed from cups to cups and the details of the changes.**

Requirement:-

It is required to give the details of each team, its captain and the year of captaincy for which the captain has changed

1. **Which teams have won two consecutive world cups? The answer should include when the world cups took place, against whom they won the finals, the scores of the final, the names of the players, the coach and the captain.**

Requirement:-

It is required to find out the teams that have won 2 consecutive worldcups and the details asked in the use case.

1. **Who holds the highest record of maximum runs, maximum wickets, man of the match till date?**

Requirement:-

It is required to find the players who has scored maximmum runs, maximum wickets and man of the match.

1. **List top 5 players information who have played maximum matches in ICC world Cup.**

Requirement:-

We will need to find the top 5 players who have played maximmum matches in the ICC world cup.The output will have the player’s names and the number of matches they have played.

1. **List top 5 bowlers who are have scored a hat-trick. Give details of the match.**

Requirement:-

We need to find those 5 bowlers who have taken hat-trick in a match and also provide the details of the match in which they took hat-trick

1. **Name all coaches who have changed the teams in series of world cups.**

Requirement:-

The coach who have trained more than one team in the world cup till date are considered and the teams he has coached is expected with the worldcup year.

1. **Which teams have scored maximum 6 and 4 in a match (not sum of all match but single match where maximum hits have been recorded). Give the details of the match where it was scored.**

Requirement:-

It is required to find the team that has hit maximum sixes and fours in a match where maximum fours and sixes were recorded.

1. **Show the average run rate of top 5 batsman and to which team they belong.**

Requirement:-

We need top 5 batsman and their run rate and their respective team and details.

1. **Which 5 team has the highest run chase display by grouping it in response to world cup year.**

Requirement:-

We should come up with 5 teams with highest run chase display in each world cup and the details such as worldcup, chaser and the scores are expected.

1. **List the top 10 players who are blowers but could score the highest runs in world cup (group by world cup year)**

Requirement:-

We should give the top 10 bowlers with the highest runs score in every worldcup.

1. **What was the schedule of semi-finals of and scores of the matches and name of the referee (do it for recent 2 world cups)**

Requirement:-

We should list the semi-final matches with date as in when it took place and the details regarding those matches along with who umpired those matches.

1. **How many times has a player received the MotM(Man of the Match).**

Requirement:-

It is required to give a count of Man of the Match obtained by the players.

## User Stories

1. **Head to Head:- The Media wants to come up with the previous head on record of the two teams that are going to compete in the upcoming match to give a clear picture of the team who is having the upper hand in terms of confidence going into next match**.

Requirement:- The head on output should have the number of matches won and lost and also the win and lose percentage of both the teams.

1. **Top 11 Players:-The International Cricket Council wants to come up with a "World XI" team consisting of 11 best performers among all teams. To accomplish this, the council determines the best players as per individual criteria such as Best pace bowler Best spinner Best openers Best middle order batsmen Best all-rounders etc."**

Requirement:- We need to come up with best bowlers, best batsmen among all teams based on their performance in the world cup. There will be different criteria used in deciding the players.

1. **Team Ranking:- The International Cricket Council wants to release the latest team rankings considering the previous 5 WC's data in hand.**

Requirement:-It is required to come up with the countries holding the top 5 ranks, number of matches won, matches played and their win percentage .

1. **Game Coaching:- The Indian team's coach wants to focus on areas of improvement of a player for which he considers the available history data of a player's performance in WC and determines as what all areas the improvement should be made.**

Requirement:- It is required to provide the Coach necessary information of a player such as number of times a batman has got out in different fashion, his Strike rate etc and Economy, wickets taken etc when it comes to bowler so that coach can take necessary step for improvement.

1. **The Quiz:- "The media conducts a quiz for entertainment to the cricket fans based on the history of WC data. It comes up with 5 questions and the corresponding options obtained by querying the history data.**

Requirement:- The Quiz involves question for which the answer should be obtained through querying our DB. Our requirement here is to provide answers and media will be able to organize a quiz competition among the cricket fans

# FUNDAMENTALS

## Technology used

NoSQL is a non relational database management system. It is used for distributed , storing and accessing large volume of data, for example:-Google, Facebook , Amazon, which collects terabytes of data every day. The data stored does not always require a fixed schema which helps in evolving continuously as the requirement changes without the penalty of expensive schema changes and migrations.

In this project we are using NEO4J which is a graph based database. It stores data in the form of nodes and relationships. Data is store as properties inside nodes and relationships. Relationships in NEO4J are the vital parts of the database. They help in connecting related nodes and thus helps in finding related nodes.

The implementation of a data model in NEO4J is same as the model that can be designed on a white board. The entities of the model can be represented as nodes and the relationships in the model as relationships in NEO4J. This allows the model to be consistent throughout conception, design, implementation, storage of domain .

The ICC contains huge amount of data of all the matches, players, umpires, coaches, stadiums , teams etc. The data in the ICC are also highly related. For every world cup the data increases and may also need change in schema from time to time. Fetching data from a worldcup database involves extracting information from multiple entities which are related. Since the factors such as huge amount of data , highly related data, flexible schema and retrieval of information for analytical purpose involves access of multiple nodes, it makes NEO4J a very appropriate database for storing and retrieval of the data for our implementation.

## Datamodel

NEO4J is referred to as Prorerty Graph Model as it stores data in nodes and relationship and both can have properties.

The fundamental units of NEO4J:-

**Nodes:-** They are used to represent entities.They can be labeled with one or more labels.They can have properties.

**Relationships:-** They are used to represent relationships of a model.They help in connecting and hence organizing the nodes.They can have properties.

**Properties:-** Properties are the string that are supported by values of different data types such as numeric,string,boolean,list etc.

**Labels:-** Labels are assigned to nodes, which helps in grouping nodes into sets.i.e all nodes labeled with the same label belongs to one set.This helps in the queries of the database to work only on that set when working with a node’s label. They are also used in adding indexes to the properties and defining constraints as well.

**Traversal:-** Answering a query might require accessing multiple nodes and the relationships between them .Such navigation from one node to another to implement a logic is called traversal.

**Paths:-** It is one or more nodes with its connected relationships, typically used during traversals.

## CRUD operations

The Create, Remove, Update and Delete Operations that can be performed in neo4j are tabulated

below.

|  |  |
| --- | --- |
| **TASK** | **QUERY** |
| To Create a Node: | CREATE (m:MATCH) RETURN m |
| To Create a Property in a Node: | CREATE (m:MATCH{id:"M1"})return m |
| To Create a Relationship between two Node: | MATCH (m:MATCH{id:"M1"}),(n:MATCH{id:"M2"}) CREATE (m)-[r:SEMIFINAL]->(n)return m,n,r |
| To Create a Property in a Relationship: | MATCH (m:MATCH{id:"M1"}),(n:MATCH{id:"M2"}) CREATE (m)-[r:SEMIFINAL{Played\_In:”SOUTH AFRICA”]->(n)return m,n,r |
| To Remove a Property in a Node: | MATCH (m:MATCH{id:"M1"}) REMOVE m.name return m |
| To Remove a property in a Relationship: | MATCH (m:MATCH{id:"M1"})-[r:SEMIFINAL]->(n:MATCH{id:"M2"})REMOVE r.Year return r,m,n |
| To Update a Property values: | MATCH (m:MATCH{id:"M1"})-[r:SEMIFINAL]->(n:MATCH{id:"M2"}) SET m.name="IND vs PAK" return m,n,r |
| To Update a new Property to a Node or Relationship: | MATCH (m:MATCH{id:"M1"})-[r:SEMIFINAL]->(n:MATCH{id:"M2"}) SET r.Year="2007" return m,n,r |
| To Delete a Node: | MATCH (m:MATCH{id:"M3"}) DELETE m |
| To Delete a Relationship: | MATCH (m:MATCH{id:"M1"})-[r:SEMIFINAL]->(n:MATCH{id:"M2"}) DELETE r |
| To Delate Node and Relationship: | MATCH (m:MATCH{id:"M1"})-[r:SEMIFINAL]->(n:MATCH{id:"M2"}) DELETE r,m,n |

# 4 PROJECT DATA-MODEL

In neo4j database, data can be stored in node and also in relationship. This section explains about the nodes and relationships used to facilitate our cricket use cases.

**WorldCup:** WorldCup is a node which contains two labels and seven properties. First label is identical for all the WorldCup nodes and the second label is used to differentiate between different world cup types. The seven properties are **id** which contains unique id, **wonBy** is used to store the team’s full name which won the world cup, **numberOfParticipants** stores the total number of teams taken part in that particular world cup. **MOS** stores the player name who received the man of the series award, **mostRuns** property is used to store the player name who scored the highest runs in that particular world cup and the **mostWickets** is used to store the player name who has taken maximum wickets of that particular world cup, **year** contains the year in which the world cup was held.

**Match:** Match is a node in our database and it contains two labels to differentiate between quarterfinal, semifinal and final. This node has six properties which helps us to store the match details.They are **ID**, **name** which tells us between which team this match was held, **date** on which the match was held, **toss** gives the team which won the toss and its preference, **MOM** (Man of the Match) stores the player name who received that award and **result** property stores the winning team name and its won by details.

**Team:**

Team node has one label and two properties. The two properties are **id** and **name** which stores the team name short form representation of it (for example: India as IND) and its full name.

**Player:**

Player node contains one label and four properties. Player label name as Player is identical for all the nodes. Its properties are **id** which contains unique id for each player, **name** stores the player name, **DOB** is used to store the date of birth of the player and **medicalIssues** is used to store player’s medical history.

**Country:**

Country node has one label and one property. Label is identical for all country nodes and the **name** property stores the country name.

**Umpire:**

Umpire node contains on label and three properties. The properties of umpire node are **id**, **name** and **nationality** stores the unique umpire id, name of the umpire and his/her nationality.

**Coach:**

Coach is a node with one label and three properties. It is similar to umpire node and it contains the same properties as of umpire node.

**Stadium:**

Stadium node contains one label and three properties such as **id**, **name** and **capacity** where id stores the unique id of stadium, name contains the stadium’s name and capacity stores the number of audience it can accommodate.

**Batsman,Bowler and Hatrick:**

Batsman,Bowler and Hatrick are single nodes in our database and has no property.

**Relationships:-**

**STATS\_FOR:**

STATS\_FOR is a relationship which is used to relate the Player node and Match node. In neo4j, Relationship (edge) can contain only one label name but it can also have properties. This relationship is used to store the player’s statistics associated with that particular match.

**Depending upon the player’s statistics, the properties of this relationship changes**. If a player has no statistics associated with that particular match then the there is no property. If a player has batted but no bowled then the properties are **runsScored**, **ballsTaken**, **sixes**, **fours** and **dismissal**. If a player has only bowled then the properties are **oversBowled**, **runsConceded**, **wickets** and **maiden**. If a player has batted and bowled then both the above properties apply.

**Umpired:**

This relationship is used to relate Umpire node and Match node. It contains no property. This relationship is used to connect the umpire node to the Match node in with he/she has umpired.

**IS\_A:**

IS\_A relationship is used to relate the Player to the Batsman/ Bowler node. **Its properties differ depending upon the end node**. If the end node is **Batsman** then its properties are **type** and **position** where type is used to store whether the player is a right hand or left hand batsman, position is used to store his/her batting order.

In case of **Bowler** being the end node, the properties are **type** and **bowlType**. Property named type stores where he is a right or left arm bowler and bowlType property stores the bowling style of that particular player.

If a player is a batsman and a bowler, then there will be two outward relationships from that player one connecting to bowler node and the other connecting to batsman node.

**CONDUCTED:**

This relationship relates between WorldCup node and Match node and it does not contain any property. It is used to connect from WorldCup node to Match node to show in which world cup this match was held.

**HOSTED:**

HOSTD is a relationship with no property. It relates Country and WorldCup node. This relationship is used to show which country hosted that particular world cup.

**TOOK:**

This relationship has no property and it relates Player to Hatrick node. It contains two properties matchId and batsmanOut which stores the Match unique id and the player’s unique ids in an array. This relationship is used to connect the Player who has taken consecutive three wickets in an over to the Hatrick node.

**COACHED:**

Coached is a relationship with one property called **wcid** and it relates between Coach node and Team node. wcId property stores unique ids of the world cup in an array in which that particular coach has coached the team.

**HELD\_IN:**

This relationship does not have any property and this helps to relate Match to Stadium to show in which stadium that match was held.

**PLAYED\_FOR\_THE:**

This relationship contains single property which stores the world cup id in an array. This relationship help to relate Player to Team node to show in which team that player played and the property value shows in which all world cups he has played.

**KEPT\_WICKETS:**

This relationship is like PLAYED\_FOR\_THE and the only difference is the label name to differentiate that this player has played in that particular team as wicket keeper.

**VICE\_CAPTIONED:**

This relationship is like PLAYED\_FOR\_THE and the only difference is the label name to differentiate that this player has played in that particular team as a vice caption.

**TV\_UMPIRED:**

This relationship does not have any property and it is used to show that the umpire has TV umpired for that particular match.

**CAPTIONED:**

This relationship is similar to PLAYED\_FOR\_THE and the only difference is the label name to differentiate that this player has played in that particular team as a caption.

**IS\_IN:**

This relationship has no property and is used to relate from Country to Stadium node. This relationship helps to find the country in which that particular stadium is located.

**PARTICIPATED\_IN:**

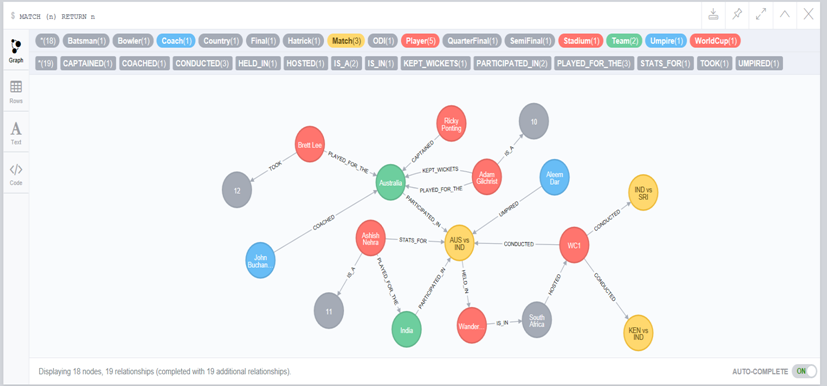
This relationship is used to relate from Team to Match node and has three properties and they are **score**, **wickets** and **overs**. This relationship is used to connect team to a match node which played in that particular match. Properties in this relationship stores each team’s final score, number of overs played and the wickets lost by them.

**REFEREED:**

This relationship has no property and it relates Umpire to Match. This relationship is used to show which all umpires where umpired in that particular match.

**Unique Constraint:**

id property of all the nodes used in this data model applies unique constraint.



# Implementation

## Implementation of the data-model

**Nodes:-**  
 11 Nodes are created with their respective properties that holds the data of each node.

|  |  |  |
| --- | --- | --- |
| **NODES** | **CREATE QUERY** | **DETAILS** |
| **WorldCup** | CREATE (n:WorldCup:ODI{id : "WC1", wonBy : "Australia", numberOfParticipants : 14, MOS : "Sachin Tendulkar", mostRuns : "Sachin Tendulkar", mostWickets : "Chaminda Vaas", year : 2003 }) | Considering 2 WorldCup’s data, WorldCup nodes are  created with the labels “WorldCup“ and “ODI“ and the  details of the WorldCup are stored as its properties |
| **Match** | CREATE (n:Match:Final{id : "WC2M7", name : "AUS vs SRI", date : "28-Apr-2007", toss : "AUS bats", MOM : "Adam Gilchrist", result : "AUS won by 53 runs" }) | A match can be a Quarter Final, Semi Final or a Final.  Labels are used to differenciate the types of match .  The details of match are store as its property. |
| CREATE (n:Match:QuarterFinal{id : "WC1M2", name : "IND vs SRI", date : "10-Mar-2003", toss : "SRI bowls", MOM : "Javagal Srinath", result : "IND won by 183 runs" }) |
| CREATE (n:Match:SemiFinal{id : "WC1M5", name : "KEN vs IND", date : "20-Mar-2003", toss : "IND bats", MOM : "Saurav Ganguly", result : "IND won by 91 runs" }) |
| **Team** | CREATE (n:Team{id : "IND", name : "India"}) | Nodes are created for all the teams participated in all the  worldcups and their details are stored as properties. |
| **Player** | CREATE (n:Player{id : "RSA\_09", name : "Graeme Smith", DOB : "01-Feb-1981", medicalIssues : "Groin Injury"}) | Nodes are created for All the Players of worldcup and  their details are store as their properties. |
| **Batsman Bowler Hatrick** | CREATE (n:Batsman) | A node with label Hatrick/Batsman/Bowler is created  with no properties. |
| CREATE (n:Bowler) |
| CREATE (n:Hatrick) |
| **Umpire** | CREATE (n:Umpire{id : "U2", name : "Asoka de Silva", nationality : "Sri Lanka"}) | Nodes are created for all the umpires who have umpired in the worldcups till date and their details are store as the  properties |
| **Coach** | CREATE (n:Coach{id : "C2", name : "Duncan Fletcher", nationality : "Zimbabwe" }) | Nodes are created for all the coaches who have coached  the cricket teams of worldcups till date and their details  are store as the properties. |
| **Country** | CREATE (n:Country{name : "Kenya"}) | Nodes are created for all the countries that has  participated in the worldcups till date and their details are  store as the properties. |
| **Stadium** | CREATE (n:Stadium{id : "S2", name : "Wanderers Stadium", capacity : 34000}) | Nodes are created for all the stadiums that has participated  in the worldcups till date and their details are store as  the properties. |

|  |  |  |
| --- | --- | --- |
| **RELATIONSHIP** | **CREATE QUERY** | **DETAILS** |
| **UMPIRED** | MATCH (a:Umpire), (b:Match) WHERE a.id = "U8"  AND b.id = "WC2M1" CREATE (a)-[r:UMPIRED]->(b) | A relationships is created between Umpire and  Match in which they have umpired. |
| **CONDUCTED** | MATCH (a:WorldCup),(b:Match) WHERE a.id = "WC1"  AND b.id contains "WC1" CREATE (a)-[r:CONDUCTED]->(b) | A relationships is created between WorldCup and Match  which shows that the matches have been conducted in the  respective worldcup. |
| **PLAYED\_FOR\_THE VICE\_CAPTAINED CAPTAINED** | MATCH (a:Player),(b:Team) WHERE b.id = "NZ" AND a.id = "NZ\_04"  CREATE (a)-[r:PLAYED\_FOR\_THE { wcId : [WC1"]}]->(b)  MATCH (a:Player),(b:Team) WHERE b.id = "NZ" AND a.id = "NZ\_05"  CREATE (a)-[r:VICE\_CAPTAINED { wcId : [WC1","WC2"]}]->(b)  MATCH (a:Player),(b:Team) WHERE b.id = "NZ" AND a.id = "NZ\_05"  CREATE (a)-[r:VICE\_CAPTAINED { wcId : [WC1","WC2"]}]->(b) | A relationships is created between Player and Team as  PLAYED\_FOR\_THE/VICE\_CAPTAINED/CAPTAINED ,  depending on their role in the match. |
| **COACHED** | MATCH (a:Coach),(b:Team) WHERE b.id = "RSA" AND a.id = "C9" CREATE (a)-[r:COACHED { wcId : ["WC1"]}]->(b) | A relationships is created between COACH and TEAM in  which they have coached. |
| **HOSTED** | MATCH (a:Country),(b:WorldCup) WHERE a.name = "Kenya" AND b.id = "WC1" CREATE (a)-[r:HOSTED]->(b) | A relationships is created between COUNTRY and  WORLDCUP which says that the country has hosted  the worldcup. |
| **IS\_A** | MATCH (a:Player),(b:Batsman) WHERE a.id = "PAK\_10" CREATE (a)-[r:IS\_A { type : "Right hand", position : "Opener"}]->(b)  MATCH (a:Player),(b:Bowler) WHERE a.id = "AUS\_08" CREATE (a)-[r:IS\_A { type : "Right arm", bowlType : "Fast medium"}]->(b) | A relationships is created between Player and Batsman  which holds the properties such as his batting style  and the position.  A relationships is created between Player and Bowler  which holds the properties such as his batting style. |
|
| **PARTICIPATED\_IN** | MATCH (a:Team),(b:Match) WHERE a.id = "WI" AND b.id = "WC2M2" CREATE (a)-[r:PARTICIPATED\_IN { score :190, wickets :10, overs :44.3}]->(b) | A relationships is created between Team and Match in  which they have participated.This holds the team scores  and records for that match. |
| **HELD\_IN** | MATCH (a:Match),(b:Stadium) WHERE a.id = "WC1M3" AND b.id = "S3" CREATE (a)-[r:HELD\_IN]->(b) | A relationships is created between MATCH and  STADIUM in which the match was conducted. |
| **IS\_IN** | MATCH (a:Country),(b:Stadium) WHERE a.name = "West Indies" AND b.id = "S8" CREATE (b)-[r:IS\_IN]->(a) | A relationships is created between COUNTRY  and STADIUM in which it is situated in. |
| **TOOK** | MATCH (a:Player),(b:Hatrick) WHERE a.id = "AUS\_07" CREATE (a)-[r:TOOK { matchId : "WC1M1", batsmenOut : ["NZ\_20","NZ\_08","NZ\_10"]}]->(b) RETURN r | A relationship is created between Player and Hatrick  which consists of the batsman id whose wickets have  been taken |
| **STATS\_FOR** | MATCH (a:Player),(b:Match) WHERE a.id = "NZ\_03" AND b.id = "WC2M5" CREATE (a)-[r:STATS\_FOR{ runsScored :30, ballsTaken :38, sixes : 0, fours : 2, dismissal : "not out", Overs : 9, runsConceded : 46, wickets :2, maiden : 1}]->(b) | A relationships is created between PLAYER and  MATCH which holds the stats of the batsman/bowler  for that match. |

## Implementation of user-stories and queries

### **Use Case**

1. How many world cups where held? The answer should also provide the information of where they were played, when they were played and who won the cup.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Count of World cup nodes is found for the number of worldcups and data stored in the world cup node and its related nodes and subnodes such as Country and Stadium respectively give us the information related to the world cup. | **QUERY:-**  Match(w:WorldCup)return count(w) as No\_Of\_WorldCup\_Held  MATCH (c1:Country)-->(wc:WorldCup)-->(m:Match)-->(s:Stadium)-->(c2:Country) RETURN distinct wc.id as World\_Cup,wc.year as Worldcup\_Year,collect(distinct c1.name) as Hosted\_By,collect(distinct c2.name) as Matches\_Conducted\_In,wc.wonBy as Won\_By ORDER BY World\_Cup ASC |

1. Who won the recent world cup? Display all the matches of the winning team including their results.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Each Match node is connected to its corresponding WorldCup node. The winner information is present in the WorldCup node and the matches played by that team in that worldcup is fetched by traversing the Worldcup and the Match nodes. | **QUERY:-**  MATCH (wc:WorldCup)  WITH wc.wonBy as won\_by,wc.year as wc\_year  ORDER BY wc\_year DESC  LIMIT 1  MATCH (wc:WorldCup{year:wc\_year})-->(m:Match)<-[r2:PARTICIPATED\_IN]-(t:Team)  WHERE t.name=won\_by  RETURN wc\_year as Recent\_Worldcup,won\_by as Winner,m.name as Match,m.date as Date,m.result as Result |

1. Which teams were out of quarterfinals in the past 5 world cups? List the teams including against whom they lost and when the match took take place.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Every match is a node and is labeled as QuarterFinals, SemiFinals and Finals in order to group different types of matches. As, each match is connected to Team nodes that have participated.A collection of teams participated in SemiFinal are deducted from the ones participated in the QuarterFinals which will result in the teams that were out of QuarterFinals. The details of whom they lost against is fetched by traversing the graph from team to team nodes connected by a Match node in which they have lost. The details of when the match took place is available in the Match node. | **QUERY:-**  match (wc:WorldCup)-->(qf:SemiFinal)<--(t:Team)  WITH wc.id as wc\_id,collect(t.id) as sf\_teams  MATCH (wc:WorldCup)-->(qf:QuarterFinal)<--(t:Team)  WHERE wc.id=wc\_id AND not(t.id in sf\_teams)  WITH t.id as Teams\_out\_of\_QF,wc\_id  match (wc:WorldCup{id:wc\_id})-->(qf:QuarterFinal)<--(t1:Team{id:Teams\_out\_of\_QF}),(qf)<--(t2:Team)  where qf.result contains t2.id  RETURN wc\_id as WorldCup ,Teams\_out\_of\_QF,t2.id as Lost\_Against,qf.date as Conducted\_On,qf.result |

1. List the teams whose captain has changed from cups to cups and the details of the changes.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** CAPTAINED is label given to the edge that connects team and player with which we see whether he has captained the team in all world cups held and if he is, we will be filtering out such teams. Rest are considered and team with changes in captaincy are found. The team and the undergone changes in captaincy are mentioned in query results | **QUERY:-**  MATCH (p:Player) -[c:CAPTAINED]-> (t:Team) WITH t,p,c, LENGTH(c.wcId) as size  MATCH(wc:WorldCup) WITH t,p,c, size, COUNT(wc) as worldCupCount  WHERE size <> worldCupCount  MATCH (woc:WorldCup) WITH t,p,c,woc WHERE woc.id IN c.wcId WITH t,p, COLLECT(woc.year) as years  WITH t, p, FILTER(x in years WHERE NOT (x-4) IN years) as finalYears WITH t.name as Team,  COLLECT({Player\_Name:p.name, Captaincy\_Year:finalYears}) as Change\_Details  RETURN Team, Change\_Details ORDER BY Team DESC |

1. Which teams have won two consecutive world cups? The answer should include when the world cups took place, against whom they won the finals, the scores of the final, the names of the players, the coach and the captain.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** Team with not more than 1 world cup is filtered initially and then teams with more than one world cup are considered and checked whether the world cups the won were of consecutive or not. If they are consecutive, then such team and years of win, team details, scores of final match are displayed in the query results | **QUERY:-**  MATCH (wc:WorldCup) WITH wc.wonBy as Winner, COLLECT(wc.year) as years, count(wc.year) as winnersCount  WHERE winnersCount > 1 WITH FILTER(x in years WHERE (x + 4) IN years) as arrayOfResults, Winner  WHERE LENGTH(arrayOfResults) > 0  MATCH (w:WorldCup) -[c:CONDUCTED]-> (m:Final) <-[part:PARTICIPATED\_IN]- (t:Team)  WITH part, m, t, Winner, arrayOfResults, w.wonBy as Winner1, w.year as winYear, w.id as wcId  WHERE Winner = Winner1 AND ( w.year IN arrayOfResults OR LENGTH(FILTER(x in arrayOfResults WHERE x + 4 = w.year)) > 0 )  WITH winYear, Winner1, part, m, t, wcId  MATCH (ma:Final)<-[st:STATS\_FOR]-(player:Player)  WHERE m.id = ma.id AND player.id CONTAINS t.id  WITH wcId, winYear, Winner1, part, m, t, COLLECT(DISTINCT player.name) as Players\_Played  MATCH (cap:Player) -[capt:CAPTAINED]-> (teamCap:Team) <-[co:COACHED]- (coach:Coach)  WHERE cap.id contains t.id AND LENGTH(FILTER(x in capt.wcId WHERE x=wcId)) > 0  AND LENGTH(FILTER(x in co.wcId WHERE x=wcId)) > 0  WITH coach.name as Coach, cap.name as Captain, t, wcId, part, Winner1, winYear, m.name as Match\_Name, m.result as Match\_Result,Players\_Played ORDER BY t.name  RETURN DISTINCT Winner1 as Winner, winYear as Year\_Won, Match\_Name, Match\_Result as Result,  COLLECT({Team:t.name,Score:part.score +"/" + part.wickets + " (" + part.overs + ")", Players:Players\_Played, Captain:Captain, Coach: Coach}) as Match\_Details |

1. Who holds the highest record of maximum runs, maximum wickets, man of the match till date?

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  To calculate maximmum runs , every player’s total runs are taken by summing the runs stored in the relationship between player and the match node.The player with highest runs scores is extracted from this.To calculate maximum wickets, every player’s total wickets taken are taken by summing the wickets taken property stored in the relationship between player and the match node.The player with highest number of wickets taken is extracted from this.To calculate maximum man of the match, the match node holds a property called manofthe match which is used to calculate number of man of the match each player has got and the one with highest man of the match is extracted from it. | **QUERY:-**  MATCH (p:Player) -[s:STATS\_FOR]-> (m:Match) WITH p.name as name, sum(s.runsScored) as TotalRuns ORDER BY TotalRuns DESC LIMIT 1  RETURN name,collect({runs:TotalRuns,title:"max runs"}) as result  UNION  MATCH (p:Player) -[s:STATS\_FOR]-> (m:Match) WITH p.name as name, sum(s.wickets) as TotalWickets ORDER BY TotalWickets DESC LIMIT 1  RETURN name,collect({runs:TotalWickets,title:"max wickets"}) as result  UNION  MATCH (m:Match) WITH m.MOM as name, count(m.MOM) as ManOfTheMatchCount ORDER BY ManOfTheMatchCount DESC LIMIT 1  RETURN name,collect({runs:ManOfTheMatchCount,title:"max Man of the match"}) as result |

1. List top 5 players information who have played maximum matches in ICC world Cup.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Each player node is connected to the Match nodes in which he has participated. A count on this relationship is done for each player and is sorted in descending order on the number of matches played to pick up the top 5 players information. | **QUERY:-**  match (p:Player)-[r:STATS\_FOR]->(m:Match)  RETURN p.name as player\_name,count(r) as matches\_played  ORDER BY matches\_played DESC  LIMIT 5 |

1. List top 5 bowlers who are have scored a hat-trick. Give details of the match.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** Hatrick is a node that is connected to the Player node who has taken it. Using this and the match where he has taken d hat-trick we can display the results such as name of the player who has taken hat-trick, against which team, score of the match and batsmen whom he got out etc. | **QUERY:-**  MATCH (h:Hatrick) <-[t:TOOK]- (p:Player)-[play:PLAYED\_FOR\_THE|CAPTAINED|VICE\_CAPTAINED|KEPT\_WICKETS]->(team:Team)  MATCH (tea:Team)<-[pll:PLAYED\_FOR\_THE|CAPTAINED|VICE\_CAPTAINED|KEPT\_WICKETS]- (pl : Player) -[s:STATS\_FOR]-> (m:Match)  WHERE pl.id IN t.batsmenOut AND m.id = t.matchId  return DISTINCT pl.name as Wickets, tea.name as Batting\_Team,  s.dismissal as Dismissal, p.name as Bowler,  team.name as Bowling\_Team, s.runsScored as Runs\_Scored, m.name as Match, m.result as Result |

1. Name all coaches who have changed the teams in series of world cups.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  The relationship COACHED contains the worldcup ids for which the coach has coached the team.The Coach node having more than one relationship with the teams are taken. | **QUERY:-**  match (coach:Coach)-[r:COACHED]->(team:Team)  WITH count(r) as count,coach as coach\_node  match (coach:Coach)-[r:COACHED]->(team:Team)  where count>1 AND coach.name=coach\_node.name  return coach.name as Coach ,team.name as Team , r.wcId as WorldCup |

1. Which teams have scored maximum 6 and 4 in a match (not sum of all match but single match where maximum hits have been recorded). Give the details of the match where it was scored.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  To calculate this, Match where the maximum sixes and fours were hit is found by summing each player's fours and sixes in a match and also finding the maximum among those summed results. Once, Match has been found, the team with maximum fours and sixes is found by grouping the summed result by team. The data such as number of fours and sixes are being stored in the STATS\_FOR relationship between Match and Player node that are considered as the source for the query | **QUERY:-**  MATCH (p:Player) -[st:STATS\_FOR]-> (m:Match) WITH m.id as mi, sum(st.fours+st.sixes) as Total ORDER BY Total DESC LIMIT 1  MATCH (pl) -[s:STATS\_FOR]-> (mt:Match),(t:Team) WHERE mt.id = mi and pl.id contains t.id WITH t, mt, sum(s.sixes+s.fours) as tt  return t.name as Team, tt as Total\_Hits, mt.name as Match, mt.result as Result |

1. Show the average run rate of top 5 batsman and to which team they belong.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Each player is connected to the Batsman node and the property position says if the player is a batsman or a bowler.  Therefore, the players whose position is Middle order or Opner are considered and the total runs scored by them is taken from the relationship property between Player and the match. Total matches in which the player has got out are considered and division of the total runs calculated and matches played is done to calculate the average run rate of a player. To do this traversals from nodes Batsman, Player, Match and the Team is done. | **QUERY:-**  MATCH (b:Batsman) <-[i:IS\_A]- (p:Player) -[s:STATS\_FOR]-> (m:Match), (t:Team)  WHERE i.position IN ["Middle order", "Opener"] AND p.id contains t.id  WITH t,p.name as Player, COLLECT(s.dismissal) as dis-missals, sum(s.runsScored) as Total\_Runs  WITH t,Player, Total\_Runs, Length(FILTER(x in dismissals WHERE x <> "not out")) as Matches\_Played  WHERE Matches\_Played > 0 RETURN t.name as Team, Player, Total\_Runs, Matches\_Played,  Total\_Runs / Matches\_Played as Batting\_Average Order by Total\_Runs DESC LIMIT 5 |

1. Which 5 team has the highest run chase display by grouping it in response to world cup year.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  To fetch the chasers and their details , toss information stored in the match is taken and a decision is made on who the chasing team is . The matches in which the chasers have won are considered and their run scores are taken. Its done for every worldcup and a collection of teams ranked as per their runs scored is prepared for every worldcup and top 5 teams are considdered in every collection. To accomplish the above logic traversals are done from nodes WorldCup,Match and teams. | **QUERY:-**  MATCH (wc:WorldCup)--(m:Match)  WITH wc.id as wc\_id,m.id as match\_id,m.name as match\_name,m.toss as toss,rtrim(left(m.toss,3)) as country1,m.result as result  WITH  CASE  WHEN rtrim(left(match\_name,3))=country1  THEN ltrim(right(match\_name,3))  ELSE rtrim(left(match\_name,3)) END AS country2,country1,toss,match\_id,wc\_id,result  WITH  CASE  WHEN toss contains "bats"  THEN country2  ELSE country1 END as chasing\_country,match\_id,wc\_id,result  WHERE result contains chasing\_country  WITH chasing\_country as chaser,match\_id,wc\_id  MATCH (wc:WorldCup{id:wc\_id})--(m:Match{id:match\_id})<-[r2:PARTICIPATED\_IN]-(t:Team{id:chaser})  WITH chaser,match\_id,wc\_id,r2.score as score  ORDER BY r2.score DESC  WITH wc\_id as wc\_id,collect({chase\_score:score,chaser:chaser})[0..5] as collection  UNWIND collection as team\_details  RETURN wc\_id as WorldCup,team\_details as Team\_Details  ORDER BY wc\_id DESC |

1. List the top 10 players who are blowers but could score the highest runs in world cup (group by world cup year)

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  To come up with this , we will be considering the Player node connected to node Batsman and the relationship property position being the bowlers.  The runs scored by them is summed from the property in the relationship between Player and the matche. The scores are sorted and the players with the runs scored are collected into a collection for every worldcup. The first 10 elements in collection gives us the details required. | **QUERY:-**  MATCH (b:Batsman) <-[i:IS\_A{position:"Tail end"}]- (p:Player) -[s:STATS\_FOR] -> (m:Match) <-[c:CONDUCTED]- (wc:WorldCup)  WITH p,wc.year as year, sum(s.runsScored) as totalRuns  ORDER BY year, totalRuns DESC  WITH year, COLLECT({Player:p.name, Total\_Runs:totalRuns })[..10] as topRunsScorer  RETURN year,topRunsScorer |

1. What was the schedule of semi-finals of and scores of the matches and name of the referee (do it for recent 2 world cups)

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** Match node with SemiFinal label is used to gather than information of all the semi-finals played in the recent world cups and also scores of those matches are displayed along with umpires who umpired the match | QUERY:- Match (wc:WorldCup)  WITH distinct(wc.year) as wc\_year  order by wc\_year desc  Limit 2  match (s:Stadium)--(match:SemiFinal)--(wc:WorldCup),(t:Team)--(match)--(u:Umpire)  where wc.year=wc\_year  WITH wc\_year as World\_Cup,match.date as match\_date,match.name as match,match.result as Result,match.id as id,s.name as venue,collect(DISTINCT u.name) as Refree\_Names,match.toss as toss,rtrim(left(match.toss,3)) as country1  WITH  CASE  WHEN rtrim(left(match,3))=country1  THEN ltrim(right(match,3))  ELSE rtrim(left(match,3)) END as country2,country1,id as match\_id,toss,World\_Cup,match\_date,match ,venue, Refree\_Names, Result  WITH  CASE  WHEN toss contains "bats"  THEN country1  ELSE country2 END as innings1,country2,country1,match\_id,toss,World\_Cup,match\_date,match ,venue, Refree\_Names, Result  WITH  CASE  WHEN country1=innings1  THEN country2  ELSE country1 END as innings2, innings1,match\_id,toss,World\_Cup,match\_date,match ,venue, Refree\_Names, Result  match (t1:Team{id:innings1})-[r1:PARTICIPATED\_IN]->(m:SemiFinal{id:match\_id})<-[r2:PARTICIPATED\_IN]-(t2:Team{id:innings2})  RETURN World\_Cup,match\_id,match,match\_date,venue,Refree\_Names,Result,r1.score as innings1\_score,r2.score as innings2\_score  ORDER BY match\_id |

1. How many times has a player received the MotM(Man of the Match)

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  Grouping MotM property stored in the World cup by Player and counting it will give the number of MotM obtained. | **QUERY:-**  MATCH (m:Match)  RETURN m.MOM as Player,count(m.MOM) as Man\_of\_the\_Match  ORDER BY Player ASC |

### **User Stories:-**

1. "The Media wants to come up with the previous head on record of the two teams that are going to compete in the upcoming match to give a clear picture of the team who is having the upper hand in terms of confidence going into next match"

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  The statistics of the matches are stored in the Match node and the PARTICIPATED\_IN relationship between Match and Team. Counting the number of Wins and Lost and grouping theM by team give us the required information about the results between them. Also, Win percentage and Loss percentage is calculated by dividing the count of wins by count of matches and count of losses by count of matches | **QUERY:-**  MATCH (m:Match) <-[p:PARTICIPATED\_IN]- (t:Team) WHERE m.name contains "IND" and m.name contains "AUS" WITH t,  COLLECT(m.result) as result WITH t, result, LENGTH(FILTER(x in result WHERE x contains t.id and x contains "won")) as Win,  LENGTH(FILTER(x in result WHERE not(x contains t.id) OR not(x contains "won"))) as Lost, LENGTH(FILTER(x in result WHERE x contains "Tied" OR x contains "Drawn" ))  as No\_Result WITH No\_Result,Win, Lost, result, t, Win/size(result) as Win\_Percent, Lost/size(result) as Loss\_Percent RETURN t.name as Team, Win, Lost, No\_Result,  Win\_Percent \* 100 as Win\_Percentage, Loss\_Percent \* 100 as Lost\_Percentage, result |

1. "The International Cricket Council wants to come up with a "World XI" team consisting of 11 best performers among all teams. To accomplish this, the council determines the best players as per individual criteria such as Best pace bowler Best Spinner, Best openers, Best middle order batsmen etc."

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** (Pace bowler) – For the best 3 medium pace bowlers, we consider the Players who plays in the Tail end position which is a property stored in the IS\_A relation between Player and Batsman nodes and the bowlType = Fast medium property stored in . We then sum their wickets taken in the entire world cup. The best 3 among the leading wicket takers are selected  (Off spin bowler) – For the best Off spin bowler, we consider the Players who plays in the Tail end position which is a property stored in the IS\_A relation between Player and Batsman nodes and the bowlType = Off spin property stored in relationship connecting Player and Bowler. We then sum their wickets taken in the entire world cup. The best 3 among the leading wicket takers are selected  (Leg spin bowler) – For the best Leg spin bowler, we consider the Players who plays in the Tail end position which is a property stored in the IS\_A relation between Player and Batsman nodes and the bowlType = Leg spin property stored in relationship connecting Player and Bowler. We then sum their wickets taken in the entire world cup. The best 3 among the leading wicket takers are selected  (Opener) – For the best 2 Openers, we consider the Players who plays in the Opener position which is a property stored in the IS\_A relation between Player and Batsman nodes. We then sum their runs scored in the entire world cup. The best 3 players who are averaging > 35 and having the best strike rate among the leading runs scorer are selected  (Middle order) – For the best 4 middle order batsmen, we consider the Players who plays in the Middle order position which is a property stored in the IS\_A relation between Player and Batsman nodes. We then sum their runs scored in the entire world cup. The best 3 among the leading average run scorer are selected | **QUERY:-**  MATCH (b:Batsman) <-[i:IS\_A{position:"Tail end"}]- (p:Player) -[s:STATS\_FOR]-> (m:Match), (t:Team) WHERE p.id contains t.id  MATCH (bow:Bowler) <-[is:IS\_A{bowlType:"Fast medium"}]- (p) return t.name as Team, p.name as Player,sum(s.wickets)  as Total\_Wickets ORDER BY Total\_Wickets DESC LIMIT 3  MATCH (b:Batsman) <-[i:IS\_A{position:"Tail end"}]- (p:Player) -[s:STATS\_FOR]-> (m:Match), (t:Team) WHERE p.id contains t.id  MATCH (bow:Bowler) <-[is:IS\_A{bowlType:"Leg spin"}]- (p) return t.name as Team, p.name as Player,sum(s.wickets)  as Total\_Wickets ORDER BY Total\_Wickets DESC LIMIT 1  MATCH (b:Batsman) <-[i:IS\_A{position:"Tail end"}]- (p:Player) -[s:STATS\_FOR]-> (m:Match), (t:Team) WHERE p.id contains t.id  MATCH (bow:Bowler) <-[is:IS\_A{bowlType:"Off spin"}]- (p) return t.name as Team, p.name as Player,sum(s.wickets)  as Total\_Wickets ORDER BY Total\_Wickets DESC LIMIT 1  MATCH (b:Batsman) <-[i:IS\_A{position:"Middle order"}]- (p:Player) -[s:STATS\_FOR]-> (m:Match), (t:Team) WHERE p.id contains t.id  WITH t, p.name as Player, COLLECT(s.dismissal) as dismissals, sum(s.runsScored) as Total\_Runs  WITH t, Player, Total\_Runs, Length(FILTER(x in dismissals WHERE x <> "not out")) as Matches\_Played  WHERE Matches\_Played > 0 RETURN t.name as Team, Player, Total\_Runs, Matches\_Played,  Total\_Runs / Matches\_Played as Batting\_Average Order by Total\_Runs DESC LIMIT 4  MATCH (b:Batsman) <-[i:IS\_A{position:"Opener"}]- (p:Player) -[s:STATS\_FOR]-> (m:Match) , (t:Team) WHERE p.id contains t.id  WITH t, p.name as Player, COLLECT(s.dismissal) as dismissals, sum(s.runsScored) as Total\_Runs, sum(s.ballsTaken) as Delivery\_Faced  WITH t, Player, Delivery\_Faced, Total\_Runs, Length(FILTER(x in dismissals WHERE x <> "not out")) as Matches\_Played  WHERE Matches\_Played > 0 WITH t, Player, Matches\_Played, Delivery\_Faced, Total\_Runs, (Total\_Runs / Matches\_Played) as Batting\_Average  WHERE Batting\_Average > 35 RETURN t.name as Team, Player, Total\_Runs, Matches\_Played, Batting\_Average,  (Total\_Runs \* 100)/Delivery\_Faced as Strike\_Rate Order by Strike\_Rate DESC LIMIT 2 |

1. The International Cricket Council wants to release the latest team rankings considering the previous 5 WC's data in hand.

|  |  |
| --- | --- |
| **IMPLEMENTATION:-** We consider the team and the count of matches won by the team to rank them among other teams. This data is present in the Match node and PARTICIPATED\_IN relation that connects Match and Team nodes. We provide Team, win percentage, matches played, matches won in query results | **QUERY:-**  MATCH (t:Team)-[r:PARTICIPATED\_IN]->(m:Match)  with count(r) as matches\_played,t.name as team\_name  MATCH (t:Team)-[r:PARTICIPATED\_IN]->(m:Match)  where m.result contains t.id and t.name=team\_name  with m.result as result,t.name as team\_name,matches\_played  with count(team\_name) as matches\_won,team\_name,matches\_played  return collect(team\_name) as Team, (matches\_won\*100)/matches\_played as Win\_Percentage, matches\_played as Matches\_Played, matches\_won as Matches\_Won  ORDER BY Win\_Percentage DESC |

1. "The Indian team's coach wants to focus on areas of improvement of a player for which he considers the available history data of a player's performance in WC and determines as what all areas the improvement should be made"

|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  For a Player as a Bowler, Total wickets taken, runs conceded, Overs Bowled and Economy should be provided. Data for providing this stat are stored in the relationship STATS\_FOR between Match and Player. Summing of wickets, runsConceded and overBowled of each match give us the total wickets, runs and overs respectively where as runsConceded per overBowled give us the Economy For a Player as a Batsman, Total Runs scored, Total balls taken in scoring those runs, Strike rate and also number of times he has got out with a particular fashion such as lbw, run-out, caught etc. has to be provided Relationship STATS\_FOR between nodes Match and Player hold the data for the above results.  Summing up the runsScored, ballsTaken give us the Total runs scored and total balls faced respectively where as runsScored per ball give us the Strike rate. For the dismissal information about the player, count of different ways a player has got out is found and dismissal property  as part of the STATS\_FOR relationship holds this information. | **QUERY:-**  (Bowler)  match (p:Player{id:"IND\_03"})-[r:STATS\_FOR]-(m:Match) where r.oversBowled IS NOT NULL  return p.name as Player\_Name,sum(r.runsConceded) as Total\_Runs\_Conceded, sum(r.oversBowled) as Overs\_Bowled,  sum(r.wicketsTaken) as Total\_Wickets\_Taken,tofloat(sum(r.runsConceded))/sum(r.oversBowled) as Economy  (Batsman)  match (p:Player{id:"SRI\_08"})-[r:STATS\_FOR]->(m:Match) where r.dismissal or r.runsScored is not null  return p.name as Player\_Name, sum(r.runsScored) as Total\_Runs\_Scored, sum(r.ballsTaken) as Balls\_Faced,  tofloat(sum(r.runsScored)) / sum(r.ballsTaken)\*100 as Strike\_Rate  match (p:Player{id:"SRI\_09"})-[r:STATS\_FOR]->(m:Match) with p,collect(r) as STATS\_FOR  return p.name as Player\_Name,size(filter(r in STATS\_FOR where r.dismissal =~ 'cau.\*')) as caught,  size(filter(r in STATS\_FOR where r.dismissal =~ 'lbw.\*')) as lbw,  size(filter(r in STATS\_FOR where r.dismissal =~ 'run.\*')) as run\_out,  size(filter(r in STATS\_FOR where r.dismissal =~ 'bowled.\*')) as bowled |

1. "The media conducts a quiz for entertainment to the cricket fans based on the history of WC data. It comes up with 5 questions and the corresponding options obtained by querying the history data."  
   Questions for quiz
2. Who Umpired the WC final?
3. Where was the WC final 2003 held?
4. Who was the Coach for India in 2003 world cup?
5. Which team scored the least in an innings in 2003 WC?

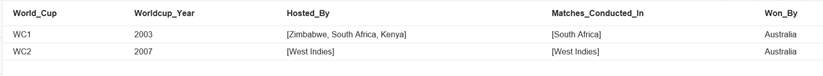
|  |  |
| --- | --- |
| **IMPLEMENTATION:-**  In this case, finding the umpire who umpired 2003 WC final requires the data stored in World cup node, Final match node and Umpire node. Filter out the 2003 World cup with year mentioned in World cup and retrieve the Umpire related to the final match of the world cup. Finding where the 2003 WC final was held is found by filtering the world cup by year 2003 and then fetching the name of the stadium where the final match was held.Finding who coached the team India during 2003 World cup is obtained by filtering the team India among all teams and finding the related coach node to retreive the coach's name. Finding the team that scored the least in 2003 WC requires PARTICIPATED edge between match and team that holds the individual team score. At first, World cup is filtered and then minimum of the individual score of the team is retrieved. | **QUERY:-**  1. Who Umpired the WC final?  MATCH (W:WorldCup)-[r:CONDUCTED]->(M:Final)<-[r1:UMPIRED]-(U:Umpire) where W.year=2003 return M.name as Match, COLLECT(U.name) AS Umpires UNION  MATCH (W:WorldCup)-[r:CONDUCTED]->(M:Match)<-[r1:UMPIRED]-(U:Umpire) return M.name as Match, COLLECT(U.name) AS Umpires LIMIT 3  2. Where was the WC final 2003 held?  MATCH (W:WorldCup)--(M:Final)-[r]->(S:Stadium) WHERE W.year = 2003 return DISTINCT S.name as Stadium, W.year as Year UNION  MATCH (W:WorldCup)--(M:Match)-[r]->(S:Stadium) return DISTINCT S.name as Stadium, W.year as Year LIMIT 4  3. Who was the Coach for India in 2003 world cup?  MATCH (C:Coach)-[r]->(T:Team) where T.name="India" and r.wcId="WC1"  return C.name as Coach,T.name as Team UNION MATCH (C:Coach)-[r]->(T:Team) return C.name as Coach,T.name as Team LIMIT 4  4) Match (W:WorldCup)-[R]->(M:Match)<-[R1]-(T:Team) where W.year=2003 return T.name as Team,min(R1.score) AS Score order by Score limit 4  5. Who has the maximum not-outs in world cup?  MATCH (m:Match) <-[s:STATS\_FOR]- (p:Player)  WHERE s.dismissal = "not out" RETURN p.name,count(s.dismissal) as Not\_Outs ORDER BY Not\_Outs DESC LIMIT 4 |

## Queries

Outputs:-

1. How many world cups where held? The answer should also provide the information of where they were played, when they were played and who won the cup.

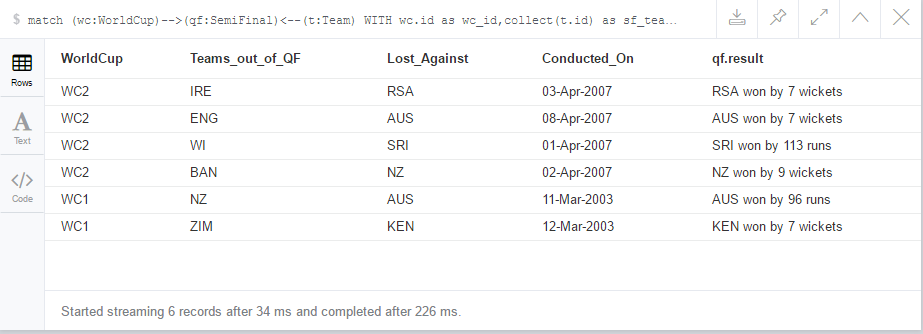




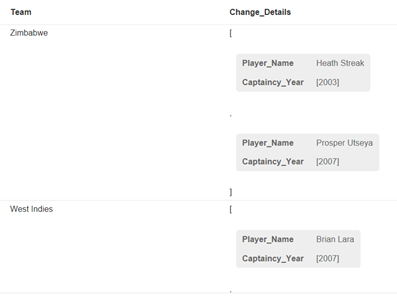
1. Who won the recent world cup? Display all the matches of the winning team including their results.



1. Which teams were out of quarterfinals in the past 5 world cups? List the teams including against whom they lost and when the match took take place.

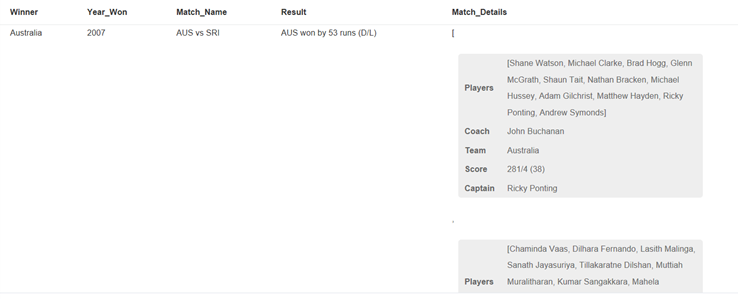


1. List the teams whose captain has changed from cups to cups and the details of the changes.



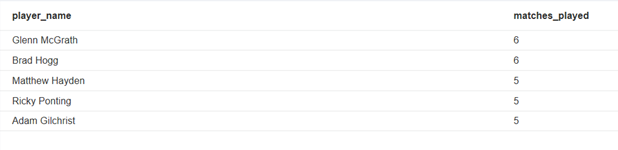
1. List top 5 bowlers who are have scored a hat-trick. Give details of the match.



1. Which teams have scored maximum 6 and 4 in a match (not sum of all match but single match where maximum hits have been recorded). Give the details of the match where it was scored.   
     
   
2. Which teams have won two consecutive world cups? The answer should include when the world cups took place, against whom they won the finals, the scores of the final, the names of the players, the coach and the captain.   
   
3. Who holds the highest record of maximum runs, maximum wickets, man of the match till date?



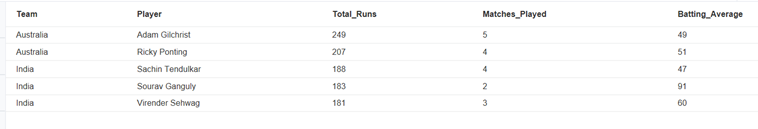
1. List top 5 players information who have played maximum matches in ICC world Cup.



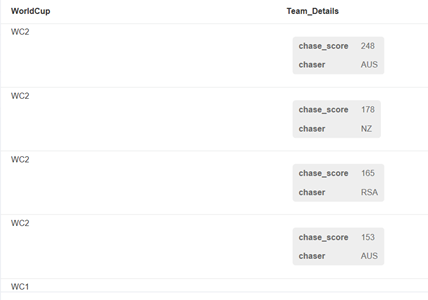
1. Name all coaches who have changed the teams in series of world cups.



1. Show the average run rate of top 5 batsman and to which team they belong.



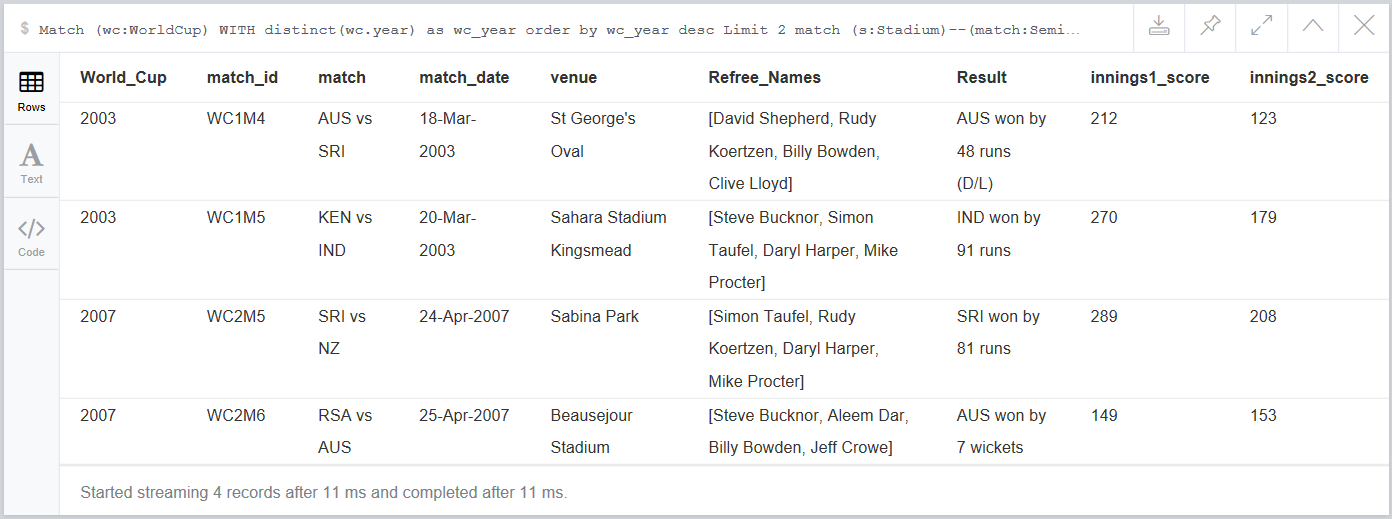
1. Which 5 team has the highest run chase display by grouping it in response to world cup year.



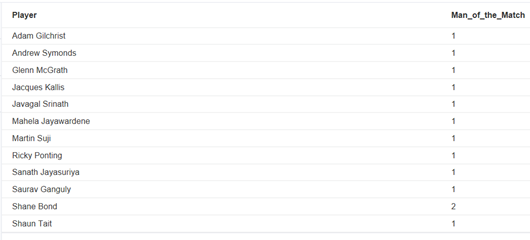
1. List the top 10 players who are blowers but could score the highest runs in world cup (group by world cup year).



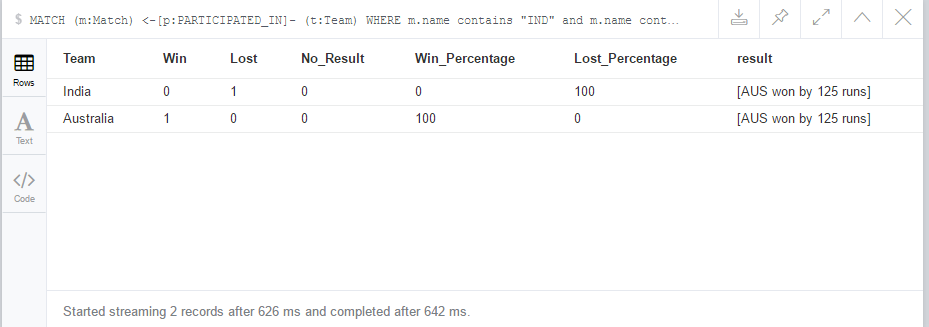
1. What was the schedule of semi-finals of and scores of the matches and name of the referee (do it for recent 2 world cups)

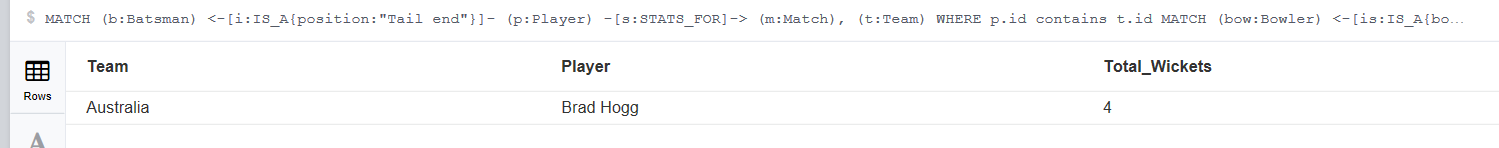


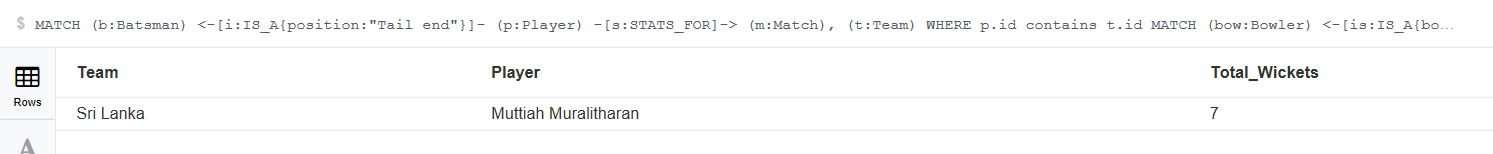
1. How many times has a player received the MoM(Man of the Match) award?

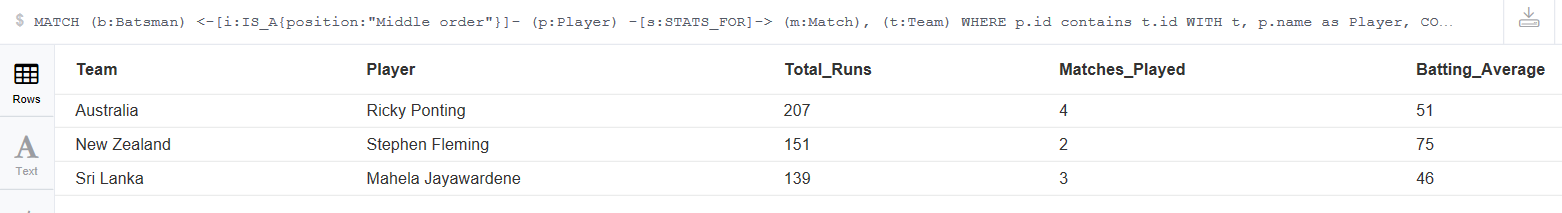


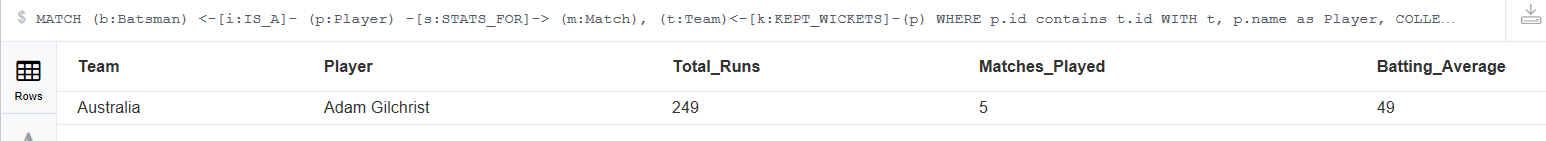
### User Stories:-

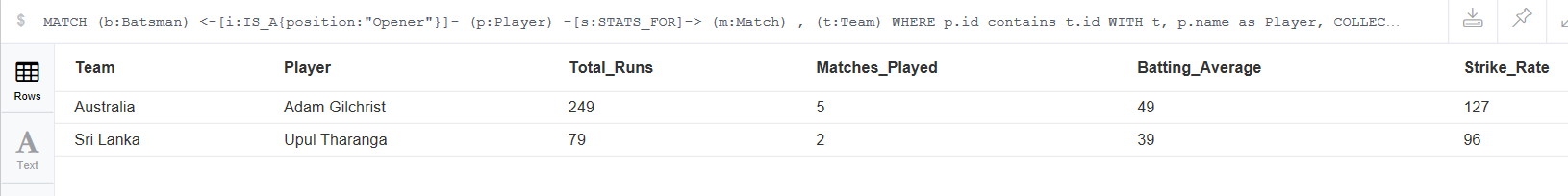
1. Head to head   
   
2. Sg-top 11 players 











1. Srg-team ranking 
2. Game coaching   
   Bowler BatsmanPlayer 

Questions for quiz

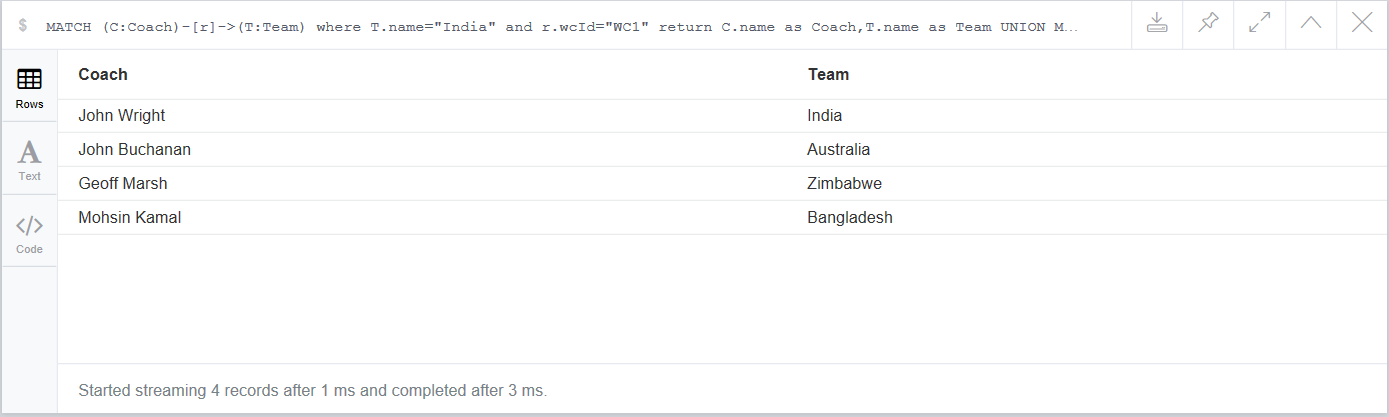
1. Who Umpired the WC final?



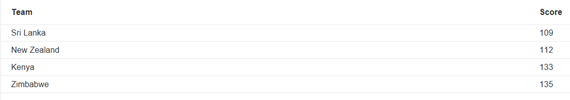
1. Where was the WC final 2003 held?

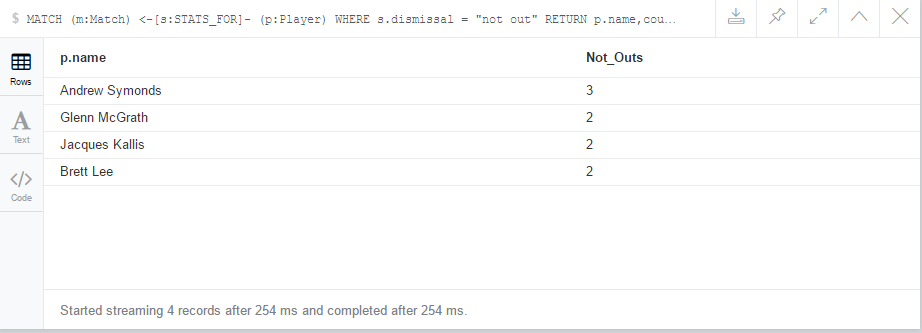


1. Who was the Coach for India in 2003 world cup?



1. Which team scored the least in an innings in 2003 WC?

  
5. Who has the maximum not-outs in world cup?



# CONCLUSION

Neo4j is a graphical database which has allowed us in imagining a real time scenario and modelling it into Nodes and edges.

This very feature has boosted the way of thinking from the traditional SQL Tables structure into more of white board pictorial representations which is a graph.

In this project, enough effort has gone in exploring its huge querying functionalities and still more to be explored. We have come across similar aggregation

functions such as that of SQL like sum(), count(), max(), min(), avg() etc. Some functionalities of SQL are implemented in Neo4j in its own way such as WITH clause to pass on the previously fetched query results onto next query as opposed to Sub-Queries in SQL. Arrays are something that we always thought of while dealing with SQL but has been greatly realized in Neo4j. FILTER and COLLECT are the new founds when it comes to dealing with array data types in Neo4j.

During the project realization, we did stumble upon some scenarios where it felt SQL had the feature which is more direct unlike Neo4j which takes a step more to arrive at the query results. One such scenario is to find the maximum of total runs scored by the batsman. SQL allows using aggregation within aggregation function which Neo4j doesn't. Also, Neo4j doesnt allow adding multiple labels to the relationship which could have been a huge plus point in our scenario of implementation if it had allowed. Player played for the team and also captained the team could have been easily realized adding two different labels as opposed to different relations that we have gone with as of now.

So far, Neo4j has behaved easy to handle DB providing all those features we could have gone for in SQL one way or the other. Exploration never ends here.

# LIST OF REFERENCES

[1] https://blog.synyx.de/2013/09/neo4j-jug-karslruhe/  
[2] https://neo4j.com/  
[3] <https://en.m.wikipedia.org/wiki/Cricket>  
[4] https://en.m.wikipedia.org/wiki/International\_Cricket\_Council  
[5] http://www.espncricinfo.com/  
[6] https://en.m.wikipedia.org/wiki/NoSQL  
[7] http://www.datastax.com/nosql-databases

[8] <http://neo4j.com/docs/developer-manual/current/introduction/graphdb-concepts/>

[9] http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL

# Appendix A

# The file attached here contains the CSV file of this projects dummy data.

# 

Screenshot of our complete data model disgram,

