**CSCI-6370 Database Management**

**Final Project Report**

**“Pro-Kabaddi League Analysis”**

|  |  |  |
| --- | --- | --- |
| Thakkar Vatsal | Deval Aditya | Kandula Punith |

The University of Georgia

**Introduction:**

The Pro Kabaddi League, founded in 2014, is a professional Kabaddi league based in India. It has gained immense popularity not just within the country, but also in several other nations. In the first season alone, it attracted 43.5 crores (435 million) viewers, second only to the 2014 Indian Premier League's viewership of 55.2 crores (552 million). Surprisingly, no one has previously worked on analyzing the Kabaddi league data considering all seasons. However, our team has taken the initiative to develop a web application for our final term project that showcases various analyses of Pro-Kabaddi League data for all seasons. Our analysis includes match-specific and player-specific statistics, as well as identifying the factors that influence match outcomes.

**Dataset & Normalization:**

As part of this project, we are gathered data from reliable sources such as Kaggle. This includes valuable information about matches, players, and seasons, as well as specific events that took place during the games. The dataset covers 7 seasons, beginning in 2014 and continuing through 2021, with comprehensive details included for each.

You can find the dataset at,  
[*https://www.kaggle.com/datasets/sripaadsrinivasan/prokabadiseassion17stats*](https://www.kaggle.com/datasets/sripaadsrinivasan/prokabadiseassion17stats)

It mainly consists of four tables,

1. Ds\_team.csv (34 Columns, ~1400 Rows)
2. Ds\_player.csv (31 Columns, ~17000 Rows)
3. Ds\_match.csv (17 Columns, 684 Rows)
4. Ds\_events.csv (41 Columns, ~67,000 Rows)

Upon review, we found that the tables were not in a normalized state and contained duplicate data, making it difficult to analyze. To address this, we applied several normalization techniques to the data and separated it into five primary tables and two supporting tables. Additionally, the data set included other data, such as world cup match data, and numerous null values, which we cleaned and preprocessed to make it suitable for our application's specific needs. You can find the code for this and file.

Our final dataset consists following tables,

1. Team (3 Columns, 12 Rows)
2. Match (8 Columns, ~600 Rows)
3. Player (2 Columns, ~500 Rows)
4. Season (3 Columns, 7 Rows)
5. Venue (4 columns , ~20 Rows)
6. Team\_Match\_Stats (23 Columns, ~1200 Rows)
7. Player\_Match\_stats (20 Columns, ~15000 Rows)

To help us better understand our database, we created a preliminary ER Diagram.

**The Preliminary ER Diagram of our database:**

A picture containing diagram, line, design

Description automatically generated

Furthermore, we also developed a final EER diagram of our Pro-kabaddi League (PKL) database schema.

**EER Diagram of PKL schema (Created in MySQL workbench):**

A screenshot of a computer

Description automatically generated with low confidence

Note: You can also install this database in your system by running the python script provided with the code.

**Queries and Impact of Project:**

To maximize the potential of the data set, we have formulated analytical queries that concentrate on match-specific and player-specific metrics. Additionally, we aim to pinpoint the factors that influence the outcome of a match, such as toss and venue. This information can help us make better decisions and improve overall performance of teams and players. We also strive to identify the factors that affect match outcomes, such as toss and venue, to make better decisions and improve overall team and player performance. By analyzing these factors, teams can uncover patterns and trends to optimize their strategies and gain a competitive advantage. This type of analysis can also assist team staff in developing better strategies for player auctions, ultimately leading to greater success and efficient goal attainment.

Following is the list of queries:

|  |  |
| --- | --- |
| 1. | What is the success rate of the raiders who have attempted the most raids across all seasons? |
| 2. | a season-wise comparison of the maximum and average raid points, tackle points, and total points? |
| 3. | How does each team's win percentage compare between home matches, neutral venues, and away matches? |
| 4. | What is the win rate of a team when they win the toss compared to when they don't? |
| 5. | Who are the defenders with the highest success rate in tackling across all seasons? |
| 6. | What are the success and failure rates of teams in knockout or decider matches? |  |

To display the results of this queries we designed a web application using Servlet and JSP.

The procedure to run the web application is provided into the README.md file which is provided with the code. There are 4 major components of the web application.

1. ***Index.jsp*** *file* : which serves as the default home page or landing page for a web application that is built using Java Server Pages (JSP) technology.
2. ***IndexServlet.java*** *file*: a Java-based server-side technology that runs on a application server and is used to process and respond to client requests over HTTP or HTTPS protocols.
3. ***DBConnector.java*** file: which establishes connection to and interact with databases using SQL queries.
4. ***ProjectQueries***.***java*** file: Contains all the queries which is going to be executed.

**Queries in SQL:**

**Query-1:** What is the success rate of the raiders who have attempted the most raids across all seasons?

SELECT player\_name AS 'Player Name',

ROUND(SUM(Player\_successful\_raids)/SUM(player\_total\_number\_of\_raids) \* 100, 2) AS 'Raider Success Rate(%)'

FROM player\_match\_stats

JOIN Player ON player.player\_id = player\_match\_stats.player\_id

WHERE player\_match\_stats.player\_id in

(SELECT player\_id

FROM

(SELECT player\_id,

SUM(player\_total\_number\_of\_raids) AS total\_raids

FROM player\_match\_stats

GROUP BY player\_id

ORDER BY Total\_raids DESC

LIMIT 10) AS T)

GROUP BY player\_match\_stats.player\_id ;

**Query-2**: A season-wise comparison of the maximum and average raid points, tackle points, and total points?

SELECT season\_name AS 'Season Name',

ROUND(AVG(total\_raid\_points), 0) AS 'Average Raid Points',

MAX(total\_raid\_points) AS 'Maximum Raid Points in a Match',

ROUND(AVG(total\_tackle\_points), 0) AS 'Average Tackle Points',

MAX(total\_tackle\_points) AS 'Maximum Tackle Points in a Match',

ROUND(AVG(score), 0) AS 'Average Total Points',

MAX(score) AS 'Maximum Total Points in a match'

FROM team\_match\_stats AS TMS

JOIN pkl.match AS M ON TMS.match\_id = M.match\_id

JOIN season AS S ON M.season\_id = S.season\_id

GROUP BY M.season\_id;

**Query-3:** How does each team's win percentage compare between home matches, neutral venues, and away matches?

SELECT team\_name AS 'Team Name',

ROUND(Numeber\_of\_home\_matches\_won/Total\_home\_matches \* 100, 2) AS 'Win Percentage Home',

ROUND(number\_of\_neutral\_matches\_won/Total\_neutral\_matches \* 100, 2) 'Win Percentage Neutral',

ROUND(Number\_of\_away\_matches\_won/Total\_away\_matches \* 100, 2) 'Win Percentage Away'

FROM

(SELECT team\_id,

COUNT(CASE

WHEN team\_id = venue\_type

AND team\_id = match\_winner\_id THEN 1

END) AS Numeber\_of\_home\_matches\_won,

COUNT(CASE

WHEN team\_id = venue\_type THEN 1

END) AS Total\_home\_matches,

COUNT(CASE

WHEN venue\_type='Neutral'

AND team\_id = match\_winner\_id THEN 1

END) AS number\_of\_neutral\_matches\_won,

COUNT(CASE

WHEN venue\_type='Neutral' THEN 1

END) Total\_neutral\_matches,

COUNT(CASE

WHEN team\_id != venue\_type

AND venue\_type!='Neutral'

AND team\_id = match\_winner\_id THEN 1

END) Number\_of\_away\_matches\_won,

COUNT(CASE

WHEN team\_id != venue\_type

AND venue\_type!='Neutral' THEN 1

END) Total\_away\_matches

FROM team\_match\_stats AS TMS3

JOIN

(SELECT M.match\_id,

match\_winner\_id,

team\_1,

team\_2,

CASE

WHEN team\_1 = v.team\_id THEN team\_1

WHEN team\_2 = v.team\_id THEN team\_2

ELSE 'Neutral'

END AS venue\_type

FROM pkl.match AS M

JOIN

(SELECT TMS.team\_id AS team\_1,

TMS2.team\_id AS team\_2,

TMS.match\_id

FROM team\_match\_stats AS TMS

JOIN

(SELECT \*

FROM team\_match\_stats AS T1

ORDER BY team\_id DESC) AS TMS2 ON TMS.match\_id = TMS2.match\_id

AND TMS.team\_id <> TMS2.team\_id

AND TMS.team\_id < TMS2.team\_id) AS T ON M.match\_id = T.match\_id

JOIN venue AS V ON M.venue\_id = V.venue\_id) AS T3 ON T3.match\_id = TMS3.match\_id

GROUP BY tms3.team\_id) AS T4

JOIN team ON T4.team\_id = team.team\_id;

**Query-4:** What is the win rate of a team when they win the toss compared to when they don't?

SELECT team\_name AS 'Team Name',

Round(COUNT(CASE

WHEN TMS.team\_id = match\_winner\_id

AND TMS.team\_id = toss\_winner\_id THEN 1

END) / COUNT(CASE

WHEN TMS.team\_id = toss\_winner\_id THEN 1

END)\*100, 2) AS 'Win Rate (Toss Won)(%)',

Round(COUNT(CASE

WHEN TMS.team\_id = match\_winner\_id

AND TMS.team\_id != toss\_winner\_id THEN 1

END) /COUNT(CASE

WHEN TMS.team\_id != toss\_winner\_id THEN 1

END) \*100, 2) AS 'Win Rate (Toss Not Won)(%)'

FROM team\_match\_stats AS TMS

JOIN pkl.match AS M ON TMS.match\_id = M.match\_id

JOIN team AS T ON TMS.team\_id = T.team\_id

GROUP BY TMS.team\_id;

**Query-5:** Who are the defenders with the highest success rate in tackling across all seasons?

SELECT player\_name AS 'Player Name',

ROUND(SUM(player\_successful\_tackles)/SUM(player\_total\_number\_of\_tackles) \* 100, 2) AS 'Tackle Success Rate(%)'

FROM player\_match\_stats

JOIN Player ON player.player\_id = player\_match\_stats.player\_id

WHERE player\_match\_stats.player\_id in

(SELECT player\_id

FROM

(SELECT player\_id,

SUM(player\_total\_number\_of\_tackles) AS total\_tackles

FROM player\_match\_stats

GROUP BY player\_id

ORDER BY total\_tackles DESC

LIMIT 10) AS T)

GROUP BY player\_match\_stats.player\_id ;

**Query-6:** What are the success and failure rates of teams in knockout or decider matches?

SELECT team\_name AS 'Team Name',

ROUND(matches\_lost/matches\_played\*100, 2) 'Losing Rate(%)',

ROUND(matches\_won/matches\_played\*100, 2) 'Winning Rate(%)',

matches\_played AS 'Total Deciders Played'

FROM

(SELECT team\_id,

COUNT(CASE

WHEN team\_id != match\_winner\_id THEN 1

END) matches\_lost,

COUNT(CASE

WHEN team\_id= match\_winner\_id THEN 1

END) matches\_won,

COUNT(team\_id) matches\_played

FROM pkl.match AS M

JOIN team\_match\_stats AS TMS ON M.match\_id = TMS.match\_id

WHERE match\_number NOT LIKE 'MATCH%'

GROUP BY team\_id) AS T1

JOIN team ON T1.team\_id = team.team\_id;

**Results of query and Snapshot of web application:**

*Home Page:*

*A screenshot of a computer

Description automatically generated with medium confidence*

*Result of Query-1:*

*A screenshot of a computer screen

Description automatically generated with low confidence*

*Result of Query-2:*

*A screenshot of a computer

Description automatically generated with medium confidence*

*Result of Query-3:*

*A screenshot of a computer

Description automatically generated with low confidence*

*Result of Query-4:*

*A screenshot of a computer

Description automatically generated with low confidence*

*Result of Query-5:*

*A screenshot of a computer screen

Description automatically generated with low confidence*

*Result of Query-6:*

*A screenshot of a score

Description automatically generated with low confidence*

**Conclusion:**

In conclusion, analyzing trends and patterns within the data set can unlock new opportunities for success and help teams achieve their objectives more effectively. This information can also assist team personnel in devising better strategies for player auctions.