

University of Maryland's Softball Database Management System

This project was taken with the aim of helping the University of Maryland set up a database management system for collecting, storing, and analyzing the Softball team's data.

This project contains theoretical as well as practical implementation in Lucidchart, Microsoft SQL Server, and Tableau.

Pre-requisites

Microsoft SQL Server, Tableau, Lucidchart

Contents

1. Description
2. Sources & References
3. Basic Structure
 - Functionalities
 - ER Diagram
 - Relational Schema
4. Implementation
 - Drop Tables
 - Creating Tables
 - Inserting Values
 - Testing
5. SQL Queries
 - Queries & Outputs
6. Tableau Visualization

Description

We created the database management system for the softball to achieve:

1. Centralization of Information: Streamlining data management by providing a centralized platform for all team-related information.
2. Efficiency and Accessibility: Enhancing efficiency and accessibility by allowing coaches and staff to easily access player statistics, schedules, and records.
3. Improved Decision Making: Facilitating data-driven decision-making, aiding in strategy development and player performance analysis.
4. Communication and Coordination: Serving as a vital tool for communication and coordination among team members, coaches, and administration.
5. Record Keeping and Compliance: Ensuring accurate record-keeping for compliance with university and athletic conference regulations.

Sources & References

Sources:

1. [UMD Softball Schedule Page](#)
2. [UMD Softball Player Roster Page](#)

References:

1. [BUDT703 – Lecture Slides, In-class Activities, Homework Assignments](#)
2. Modern Database Management (13 th Edition), Authors: Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, Publisher: Pearson

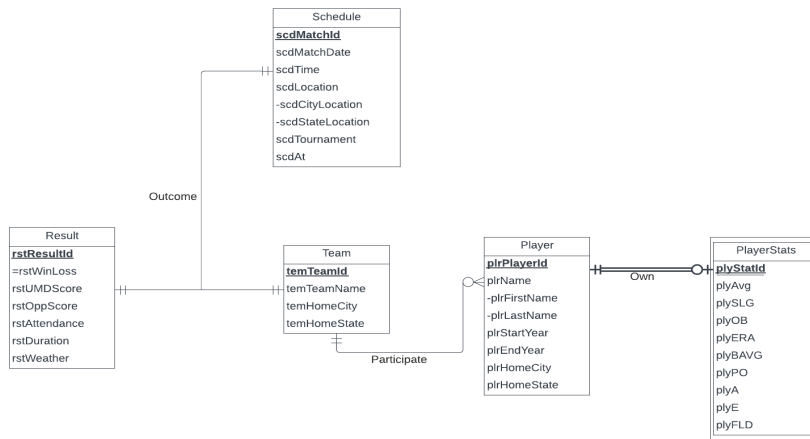
Basic Structure

Functionalities

- All the teams registered under the **“Team”** table with their respective information and any new team introduced will be added to the same.
- Any new player enlisted in the UMD’s softball team will be added to the **“Player”** table along with the descriptive details.
- Post a season/ during a current season the individual statistics will be fed into the **“PlayerStats”** Table.
- The Schedule details including date, location, type of venue of any particular season will be stored in the **“Schedule”** table.
- The results are then fed into the **“Result table”** that contains data related to the attendance, weather, duration, win/loss.

All this data stored can be accessed by Players, Coaches and UMD Softball Management team to study the data and efficiently strategize for their games and season as a whole.

ER Diagram



Project_0507_15_Proposal

Relational Schema

Team (temTeamId, temTeamName, temHomeCity, temHomeState)

Player (plrTeamId, plrPlayerId, plrFirstName, plrLastName, plrStartYear, plrEndYear, plrHomeCity, plrHomeState)

PlayerStats (plrPlayerId, plyStatId, plyAvg, plySLG, plyOB, plyERA, plyBAVG, plyPO, plyA, plyE, plyFLD)

Schedule (scdMatchId, scdMatchDate, scdTime, scdCityLocation, scdStateLocation, scdTournament, scdAt)

Result (rstResultId, rstUMDScore, rstOppScore, rstAttendance, rstDuration, rstWeather)

Outcome (scdMatchId, oppTeamId, rstResultId)

Implementation

Drop Statements:

To drop the tables, we need to consider the foreign key dependencies and use the drop statements in the following order mentioned below:

Step 1: Use the following USE statement to make the changes effective in the right database.

```
USE BUDT703_Project_0507_15;
```

Step 2: Use the drop the statements in the below order.

(Note: The IF EXISTS clause allows us to affect the statement only if the tables have been created and if they don't exist it won't return an error)

```
DROP TABLE IF EXISTS Outcome;  
DROP TABLE IF EXISTS Schedule;  
DROP TABLE IF EXISTS Result;  
DROP TABLE IF EXISTS PlayerStats;  
DROP TABLE IF EXISTS Player;  
DROP TABLE IF EXISTS Team;
```

You can use "Project_0507_15_DROP.sql" as well.

Create Statements:

To create the tables, we need to consider the foreign key dependencies and create the statements in the following order mentioned below:

Step 1: Use the following USE statement to make the changes effective in the right database.

```
USE BUDT703_Project_0507_15;
```

Step 2: Use the create the statements in the below order to create the tables, 'Team', 'Player', 'PlayerStats', 'Schedule', 'Result', 'Outcome' respectively in the mentioned order.

```
CREATE TABLE Team (  
    temTeamId CHAR(4) NOT NULL,  
    temTeamName VARCHAR(50),  
    temHomeCity Varchar(50),  
    temHomeState CHAR(2)  
    CONSTRAINT pk_Team_teamId PRIMARY KEY (temTeamId)  
);
```

```
CREATE TABLE Player (  
    plrTeamId CHAR (4),
```

```

plrPlayerId CHAR (4) NOT NULL,
plrFirstName VARCHAR (50),
plrLastName VARCHAR (50),
plrStartYear SMALLINT,
plrEndYear SMALLINT,
plrHomeCity VARCHAR (50),
plrHomeState VARCHAR (50),
CONSTRAINT pk_Player_playerId PRIMARY KEY (plrPlayerId),
CONSTRAINT fk_Player_teamId FOREIGN KEY (plrTeamId)
    REFERENCES Team (temTeamId)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE PlayerStats (
    plyPlayerId CHAR (4),
    plyStatId CHAR (5) NOT NULL,
    plyAvg FLOAT,
    plySLG FLOAT,
    plyOB FLOAT,
    plyERA FLOAT,
    plyBAVG FLOAT,
    plyPO SMALLINT,
    plyA SMALLINT,
    plyE SMALLINT,
    plyFLD FLOAT
    CONSTRAINT pk_PlayerStats_playerId_playerstatId PRIMARY KEY
    (plyPlayerId, PlyStatId),
    Constraint fk_PlayerStats_playerId FOREIGN KEY (plyPlayerId)
        REFERENCES Player (plrPlayerID)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE Schedule (
    scdMatchId CHAR(4) NOT NULL,
    scdMatchDate DATE,
    scdTime TIME,
    scdCityLocation VARCHAR(30),
    scdStateLocation VARCHAR(30),
    scdTournament VARCHAR(50),
    scdAt VARCHAR(7)
    CONSTRAINT pk_Schedule_scdMatchId PRIMARY KEY (scdMatchId)
);

```

```

CREATE TABLE Result (
    rstResultId CHAR(4) NOT NULL,
    rstUMDScore SMALLINT,
    rstOppScore SMALLINT,
    rstAttendance INTEGER,
    rstDuration FLOAT,
    rstWeather VARCHAR(30)
    CONSTRAINT pk_Result_rstResultId PRIMARY KEY (rstResultId)
);

```

```
);
```

```
CREATE TABLE Outcome (  
    scdMatchId CHAR(4) NOT NULL,  
    oppTeamID CHAR(4) NOT NULL,  
    rstResultId CHAR(4) NOT NULL,  
    Constraint pk_Outcome_scdMatchId_oppTeamId_rstResultId PRIMARY KEY  
    (scdMatchId, oppTeamId, rstResultId),  
    Constraint fk_Outcome_scdMatchId FOREIGN KEY (scdMatchId)  
        References Schedule (scdMatchId)  
        ON DELETE NO ACTION ON UPDATE CASCADE,  
    Constraint fk_Outcome_oppTeamId FOREIGN KEY (oppTeamId)  
        References Team (temTeamId)  
        ON DELETE NO ACTION ON UPDATE CASCADE,  
    Constraint fk_Outcome_rstResultId FOREIGN KEY (rstResultId)  
        References Result (rstResultId)  
        ON DELETE NO ACTION ON UPDATE CASCADE  
);
```

You can use "Project_0507_15_CREATE.sql" as well.

INSERT Statements:

You can use "Project_0507_15_INSERT.sql" to insert the values into the tables we created above.

Below are the snips for reference.

Team:

```
--Insert statement for the table Team:-  
INSERT [dbo].[Team] ([temTeamId], [temTeamName], [temHomeCity], [temHomeState])  
VALUES ('T001', 'University Of Maryland', 'College Park', 'MD'),  
       ('T002', 'Oregon', 'Eugene', 'OR'),  
       ('T003', 'BYU', 'Provo', 'UT'),  
       ('T004', 'California Baptist', 'Riverside', 'CA'),  
       ('T005', 'Oklahoma State', 'Stillwater', 'OK'),  
       ('T006', 'North Dakota State University', 'Fargo', 'ND'),
```

Player:

```
--Insert statement for the table Player:-  
INSERT [dbo].[Player] ([plrTeamId], [plrPlayerId], [plrLastName], [plrFirstName], [plrStartYear], [plrEndYear], [plrHomeCity], [plrHomeState])  
VALUES ('T001', 'P001', 'Williams', 'Diamond', 2023, 2023, 'Augusta', 'GA'),  
       ('T001', 'P002', 'Murphy', 'Bailey', 2023, 2023, 'Chesapeake', 'VA'),  
       ('T001', 'P003', 'Bucher', 'Keira', 2023, 2023, 'San Diego', 'CA'),  
       ('T001', 'P004', 'Coenwell', 'Caitly', 2023, 2023, 'Pasadena', 'MD'),  
       ('T001', 'P005', 'Runya', 'Madiso', 2023, 2023, 'West', 'TX'),
```

PlayerStats:

```
--Insert statement for the table PlayerStats:-
INSERT [dbo].[PlayerStats] ([plyPlayerId], [plyStatId], [plyAvg], [plySLG], [plyOB], [plyERA], [plyBAVG], [plyPO], [plyA], [plyE], [plyFLD])
VALUES ('P001', 'P1001', CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.00 AS Decimal(4, 2)), CAST(0.000 AS Decimal(4, 3)), 0, 0, 0, CAST(0.000 AS Decimal(4, 3))),
('P002', 'P1002', CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.00 AS Decimal(4, 2)), CAST(0.000 AS Decimal(4, 3)), 0, 0, 0, CAST(0.000 AS Decimal(4, 3))),
('P003', 'P1003', CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(2.35 AS Decimal(4, 2)), CAST(0.254 AS Decimal(4, 3)), 3, 18, 0, CAST(1.000 AS Decimal(4, 3))),
('P004', 'P1004', CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.000 AS Decimal(4, 3)), CAST(0.00 AS Decimal(4, 2)), CAST(0.000 AS Decimal(4, 3)), 0, 0, 0, CAST(0.000 AS Decimal(4, 3))),
('P005', 'P1005', CAST(0.320 AS Decimal(4, 3)), CAST(0.480 AS Decimal(4, 3)), CAST(0.393 AS Decimal(4, 3)), CAST(0.00 AS Decimal(4, 2)), CAST(0.000 AS Decimal(4, 3)), 26, 31, 2, CAST(0.966 AS Decimal(4, 3))),
```

Schedule:

```
--Insert statement for the table Schedule:-
INSERT [dbo].[Result] ([rstResultId], [rstUMDScore], [rstOppScore], [rstAttendance], [rstDuration], [rstWeather])
VALUES ('R001', 7, 3, 547, 2.42, 'Clear'),
('R002', 10, 2, 203, 2.23, 'Sunny'),
('R003', 1, 2, 203, 2.23, NULL),
('R004', 11, 6, 342, 3.34, 'Sunny'),
('R005', 4, 2, 500, 2.15, NULL),
```

Result:

```
--Insert statement for the table Result:-
INSERT [dbo].[Result] ([rstResultId], [rstUMDScore], [rstOppScore], [rstAttendance], [rstDuration], [rstWeather])
VALUES ('R001', 7, 3, 547, 2.42, 'Clear'),
('R002', 10, 2, 203, 2.23, 'Sunny'),
('R003', 1, 2, 203, 2.23, NULL),
('R004', 11, 6, 342, 3.34, 'Sunny'),
('R005', 4, 2, 500, 2.15, NULL),
```

Outcome:

```
--Insert statement for the table Outcome:-
INSERT [dbo].[Outcome] ([scdMatchId], [oppTeamID], [rstResultId])
VALUES ('M001', 'T002', 'R001'),
('M002', 'T003', 'R002'),
('M003', 'T004', 'R003'),
('M004', 'T005', 'R004'),
('M005', 'T006', 'R005'),
```

Testing:

In order to test the data that has been inserted into the table, use the following queries:

```
SELECT * FROM Team;
```

	temTeamId	temTeamName	temHomeCity	temHomeState
1	T001	University Of Maryland	College Park	MD
2	T002	Oregon	Eugene	OR
3	T003	BYU	Provo	UT
4	T004	California Baptist	Riverside	CA
5	T005	Oklahoma State	Stillwater	OK
6	T006	North Dakota State ...	Fargo	ND
7	T007	Virginia	Charlottesv...	VA
8	T008	Texas A&M Commerce	Commerce	TX
9	T009	Baylor	Waco	TX
10	T010	Minnesota	Minneapolis	MN
11	T011	Saint Joseph's	Philadelphia	PA

doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 154 rows

SELECT * FROM Player;

	plrTeamId	plrPlayerId	plrFirstName	plrLastName	plrStartYear	plrEndYear	plrHomeCity	plrHomeState
1	T001	P001	Diamind	Williams	2023	2023	Augusta	GA
2	T001	P002	Bailey	Murphy	2023	2023	Chesapeake	VA
3	T001	P003	Keira	Bucher	2023	2023	San Diego	CA
4	T001	P004	Caitly	Coenwell	2023	2023	Pasadena	MD
5	T001	P005	Madiso	Runya	2023	2023	West	TX
6	T001	P006	Sydney	Lewis	2023	2023	Prosper	TX
7	T001	P007	Delaney	Reefe	2023	2023	Fredrick	MD
8	T001	P008	Sam	Bea	2023	2023	Rockwall	TX
9	T001	P009	Grace	Carrington	2023	2023	Orange	CT
10	T001	P010	Gracely	Solarz	2023	2023	Riva	MD
11	T001	P011	Maize	Macfarlane	2023	2023	Rockli	CA

✓ Query executed succe... | doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 88 rows

SELECT * FROM PlayerStats;

Results		Messages									
	plyPlayerId	plyStatId	plyAvg	plySLG	plyOB	plyERA	plyBAVG	plyPO	plyA	plyE	plyFLD
1	P001	PI001	0	0	0	0	0	0	0	0	0
2	P002	PI002	0	0	0	0	0	0	0	0	0
3	P003	PI003	0	0	0	2.35	0.254	3	18	0	1
4	P004	PI004	0	0	0	0	0	0	0	0	0
5	P005	PI005	0.32	0.48	0.393	0	0	26	31	2	0.966
6	P006	PI006	0.216	0.309	0.3	0	0	99	3	0	1
7	P007	PI007	0	0	0	0	0	0	0	0	0
8	P008	PI008	0	0	0	0	0	0	0	0	0
9	P009	PI009	0	0	0	0	0	0	0	0	0
10	P010	PI010	1	1	1	0	0	0	0	0	0
11	P011	PI011	0	0	0	0	0	0	0	0	0

Query executed succe... | doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 88 rows

SELECT * FROM Schedule;

Results Messages

	scdMatchId	scdMatchDate	scdTime	scdCityLocation	scdStateLocation	scdTournament	scdAt	
1	M001	2023-02-09	21:30:00.0000000	Puerto Vallarta	Mexico	Puerto Vallarta College Challenge	Neutral	
2	M002	2023-02-10	16:00:00.0000000	Puerto Vallarta	Mexico	Puerto Vallarta College Challenge	Neutral	
3	M003	2023-02-11	11:00:00.0000000	Puerto Vallarta	Mexico	Puerto Vallarta College Challenge	Neutral	
4	M004	2023-02-12	12:00:00.0000000	Puerto Vallarta	Mexico	Puerto Vallarta College Challenge	Neutral	
5	M005	2023-02-12	14:30:00.0000000	Puerto Vallarta	Mexico	Puerto Vallarta College Challenge	Neutral	
6	M006	2023-02-17	15:00:00.0000000	Chapel Hill	NC	ACC/Big Ten Challenge	Neutral	
7	M007	2023-02-18	12:30:00.0000000	Chapel Hill	NC	ACC/Big Ten Challenge	Neutral	

Query executed successfully. | doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 546 rows

SELECT * FROM Result;

Results		Messages				
	rstResultId	rstUMDScore	rstOppScore	rstAttendance	rstDuration	rstWeather
1	R001	7	3	547	2.42	Clear
2	R002	10	2	203	2.23	Sunny
3	R003	1	2	203	2.23	NULL
4	R004	11	6	342	3.34	Sunny
5	R005	4	2	500	2.15	NULL
6	R006	2	1	145	2.05	Cloudy
7	R007	6	4	387	1.43	Sunny

Query executed successfully. | doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 546 rows

SELECT * FROM Outcome;

Results			
	scdMatchId	oppTeamID	rstResultId
1	M001	T002	R001
2	M002	T003	R002
3	M003	T004	R003
4	M004	T005	R004
5	M005	T006	R005
6	M006	T007	R006
7	M007	T007	R007

Query executed successfully. | doitsqlx.rhsmith.umd.edu,97... | AD\jagan11 (113) | BUDT703_Project_0507_15 | 00:00:00 | 546 rows

SQL Queries and Output:

To understand our analysis use the Project_0507_15_SELECT.sql file to run the following queries and get the respective outputs

Query 1:

--What are the 10 biggest victory Margin for UMD and against who?

WITH VictoryMargins AS (

SELECT t.temTeamName AS 'Team Name', MAX(r.rstUMDScore - r.rstOppScore) AS 'Victory Margin'

FROM Result r, Outcome o, Team t, Schedule s

WHERE r.rstResultId = o.rstResultId

AND o.oppTeamID = t.temTeamId

AND r.rstUMDScore > r.rstOppScore

GROUP BY t.temTeamName)

SELECT TOP(10) v.*

FROM VictoryMargins v

ORDER BY v.[Victory Margin] DESC;

Output:

	Team Name	Victory Margin
1	Iowa	19
2	Howard	15
3	Rhode Island	15
4	Lafayette	14
5	Indiana	13
6	Fairfield	11
7	Michigan State (DH)	11
8	Monmouth	11
9	Dartmouth	10
10	East Carolina	10

Query 2:

-- Which 10 teams did UMD lose against the most?

```
WITH Losses AS (  
    SELECT t.temTeamName AS 'Opponent', COUNT(r.rstResultId) AS '# of times Lost'  
    FROM Result r, Outcome o, Team t  
    WHERE r.rstResultId = o.rstResultId  
    AND o.oppTeamID = t.temTeamId  
    AND r.rstUMDScore < r.rstOppScore  
    GROUP BY t.temTeamName)  
SELECT TOP(10) l.*  
FROM Losses l  
ORDER BY l.[# of times Lost] DESC;
```

Output:

	Opponent	# of times Lost
1	Michigan State	34
2	Ohio State	18
3	Indiana	16
4	Penn State	13
5	Minnesota	12
6	Nebraska	11
7	Rutgers	10
8	Illinois	9
9	Iowa	6
10	NorthWestern	6

Query 3:

--Who are the Top players to be chosen for 2024 year?

```
WITH AverageStats AS  
(SELECT AVG(plyAvg) 'plyAverageAvg'  
FROM PlayerStats  
where plyAvg < 1 And  
    plyAvg > 0),
```

```
AboveAvgPlayer AS (  
SELECT plyPlayerId, plyAvg FROM PlayerStats  
WHERE plyAvg > (SELECT plyAverageAvg FROM AverageStats) AND  
    plyAvg < 1  
)
```

```
SELECT CONCAT(p.plrFirstName , ' ', p.plrLastName) AS 'Player Full Name',  
    a.plyAvg AS 'Player Average'
```

```

FROM Player p, AboveAvgPlayer a
WHERE p.plrPlayerId = a.plyPlayerId AND
      p.plrStartYear >= 2021
ORDER BY a.plyAvg DESC;

```

Output:

	Player Full Name	Player Average
1	Jennifer Bran	0.333
2	Jaeda McFarland	0.331
3	Mackense Greico	0.328
4	Madiso Runya	0.32
5	Kiley Goff	0.292
6	Mega Mikami	0.281
7	Michaela Jones	0.271
8	Sammi Woods	0.255

Query 4:

--Top 10 Win ratio for tournaments where there played atleast 5 matches?

WITH TournamentOutcome AS

(SELECT

```

CASE WHEN rstUMDScore > rstOppScore THEN 'Win'
      WHEN rstUMDScore < rstOppScore THEN 'Loss'
      when (rstUMDScore=0 and rstOppScore = 0) Then 'Cancelled'
      ELSE 'Draw' END AS 'MatchOutcome',
s.scdTournament

```

FROM Result r,

Outcome o,

Schedule s

WHERE r.rstResultId = o.rstResultId And

o.scdMatchId = s.scdMatchId And

s.scdTournament is not null),

TotalMatches AS (

SELECT scdTournament,

```

Cast(count(CASE WHEN MatchOutcome = 'Win' or

```

```

      MatchOutcome = 'Loss' or

```

```

      MatchOutcome = 'Draw' THEN MatchOutcome END ) AS

```

FLOAT) 'MatchesCountByTournament',

```

Cast(Count(CASE

```

```

      WHEN MatchOutcome ='Win'

```

```

        THEN MatchOutcome END) AS FLOAT) AS 'WinMatchCount'
FROM TournamentOutcome
WHERE MatchOutcome != 'Cancelled'
GROUP BY scdTournament
HAVING COUNT(MatchOutcome) > 3
),
TournamentWinRatio AS (
SELECT scdTournament,
        ROUND((WinMatchCount/MatchesCountByTournament), 2) as 'WinRatio'
FROM TotalMatches
WHERE MatchesCountByTournament >=5
)
SELECT TOP 10 * FROM TournamentWinRatio
ORDER BY WinRatio DESC;

```

Output:

	scdTournament	WinRatio
1	East Carolina Tournament	1
2	Houston Classic	0.8
3	Chanticleer Showdown	0.8
4	Mary Nutter Classic	0.8
5	Panther Invitational	0.8
6	Puerto Vallarta College Challenge	0.8
7	Capital City Classic	0.71
8	ACC/Big Ten Challenge	0.63
9	Amy S.Harrison Classic	0.6
10	Coastal Carolina Kicki'Chicken Classic	0.6

Query 5:

--Which stadium was most visited in terms of total audience all years combined?

```

WITH BaseTable AS (
SELECT t.temTeamName,
        r.rstAttendance,
        s.scdCityLocation
FROM Team t,
        Result r,
        schedule s,
        Outcome o
WHERE t.temTeamId = o.opppTeamID AND
        o.rstResultId = r.rstResultId AND
        o.scdMatchId = s.scdMatchId
),

```

```

TotalAudience AS (
SELECT temTeamName,

```

```

        sum(rstAttendance) AS 'TotalAttendance'
FROM BaseTable
GROUP BY temTeamName)

```

```

SELECT TOP 10 * FROM TotalAudience
ORDER BY TotalAttendance DESC;

```

Output:

	temTeamName	TotalAttendance
1	Michigan State	34395
2	Penn State	12719
3	Indiana	8934
4	Ohio State	8314
5	Nebraska	6341
6	Iowa	5094
7	Minnesota	4941
8	Wisconsin	3287
9	Clemson	3242
10	Illinois	2982

Query 6:

--How many times has UMD won so far in each weather condition?

```

WITH WeatherOutcome AS
(SELECT rstWeather,
        CASE WHEN rstUMDScore > rstOppScore THEN 'Win' END AS 'MatchOutcome'
FROM Result r
WHERE rstWeather is not NULL
)

SELECT rstWeather,
        count(CASE WHEN MatchOutcome = 'Win' THEN MatchOutcome END ) AS 'Total
Wins'
FROM WeatherOutcome
WHERE MatchOutcome != 'cancelled'
GROUP BY rstWeather, MatchOutcome
ORDER BY rstWeather ASC, MatchOutcome DESC
;

```

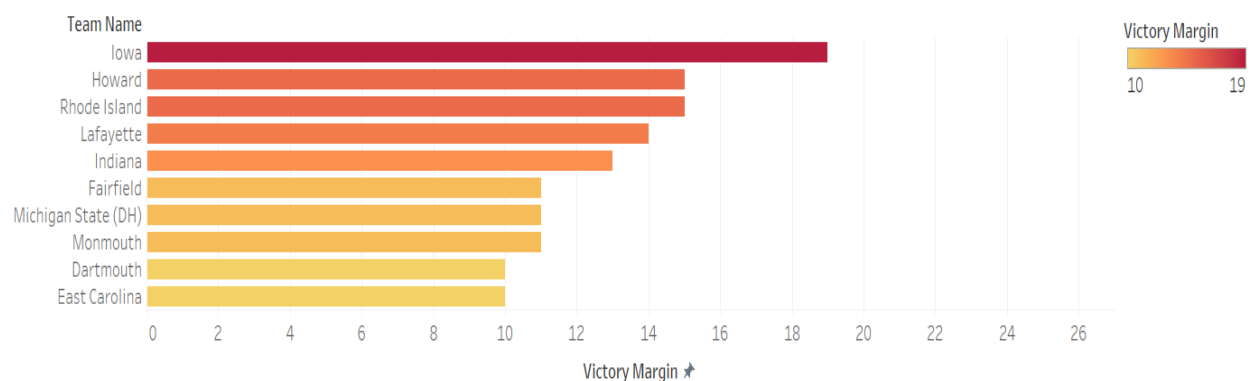
	rstWeather	Total Wins
1	Clear	28
2	Cloudy	36
3	Cold	1
4	Hazy	1
5	Overcast	17
6	Rain	3
7	Sunny	59
8	Wind	5

Tableau Visualization

To further visualize our analysis use the Project_0507_15_VISUALIZATION.twb file to observe the following outputs.

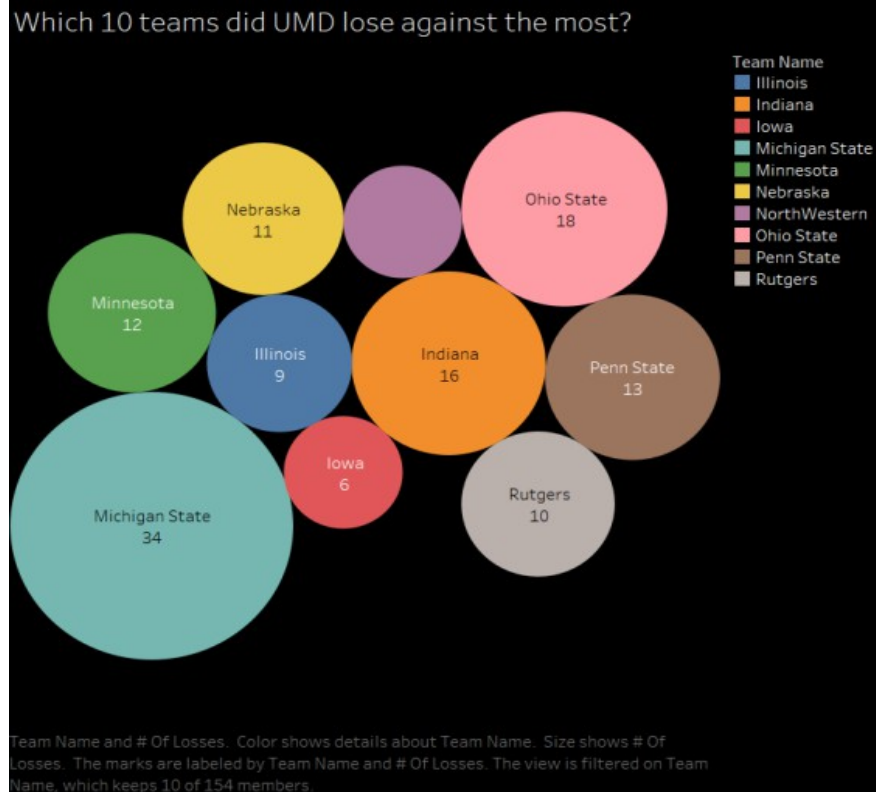
1.

What are the 10 biggest victory Margin for UMD and against who?



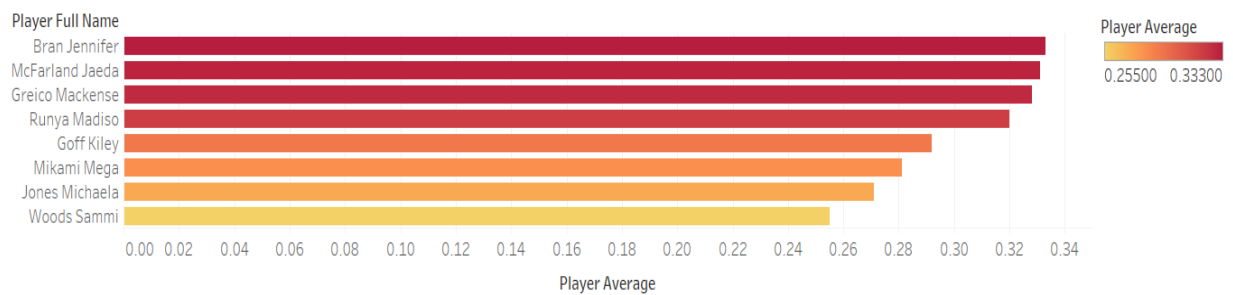
Victory Margin for each Team Name. Color shows Victory Margin. The view is filtered on Team Name, which keeps 10 of 154 members.

2.



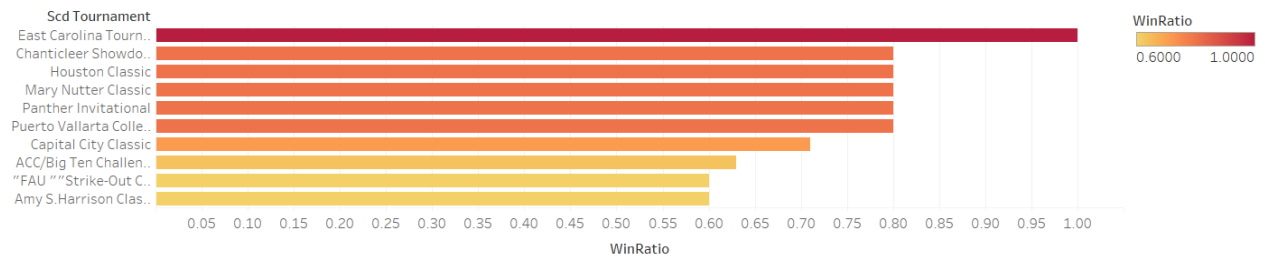
3.

Who are the Top players to be chosen for 2024 year?



4.

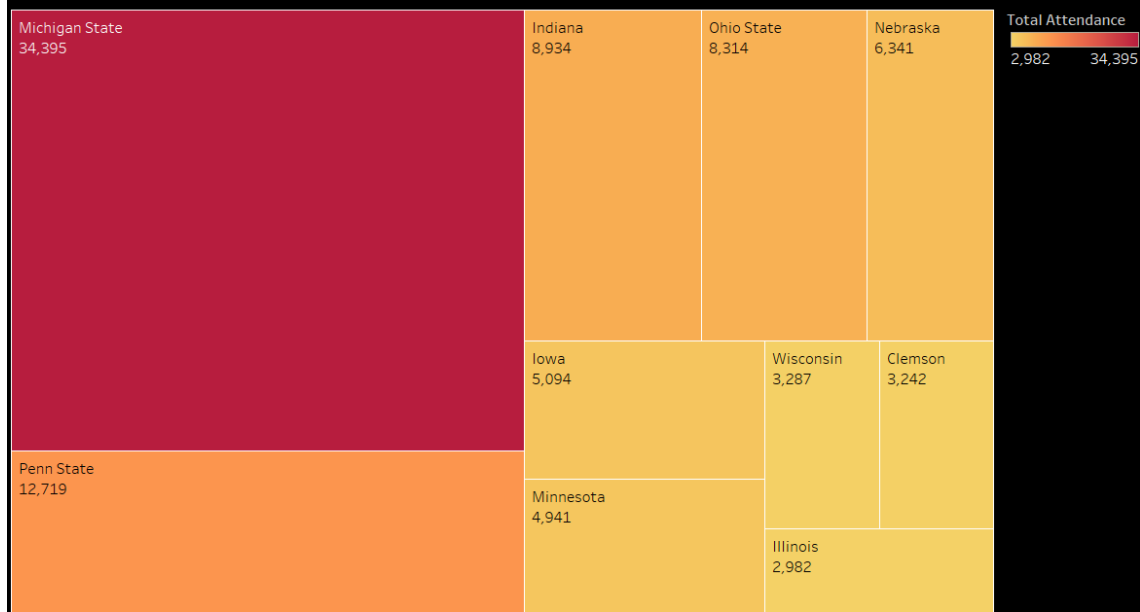
Top 10 Win ratio for tournaments where UMD played atleast 5 matches?



WinRatio as an attribute for each Scd Tournament. Color shows WinRatio as an attribute. The view is filtered on WinRatio as an attribute and Scd Tournament. The WinRatio as an attribute filter keeps non-Null values only. The Scd Tournament filter keeps 10 of 36 members.

5.

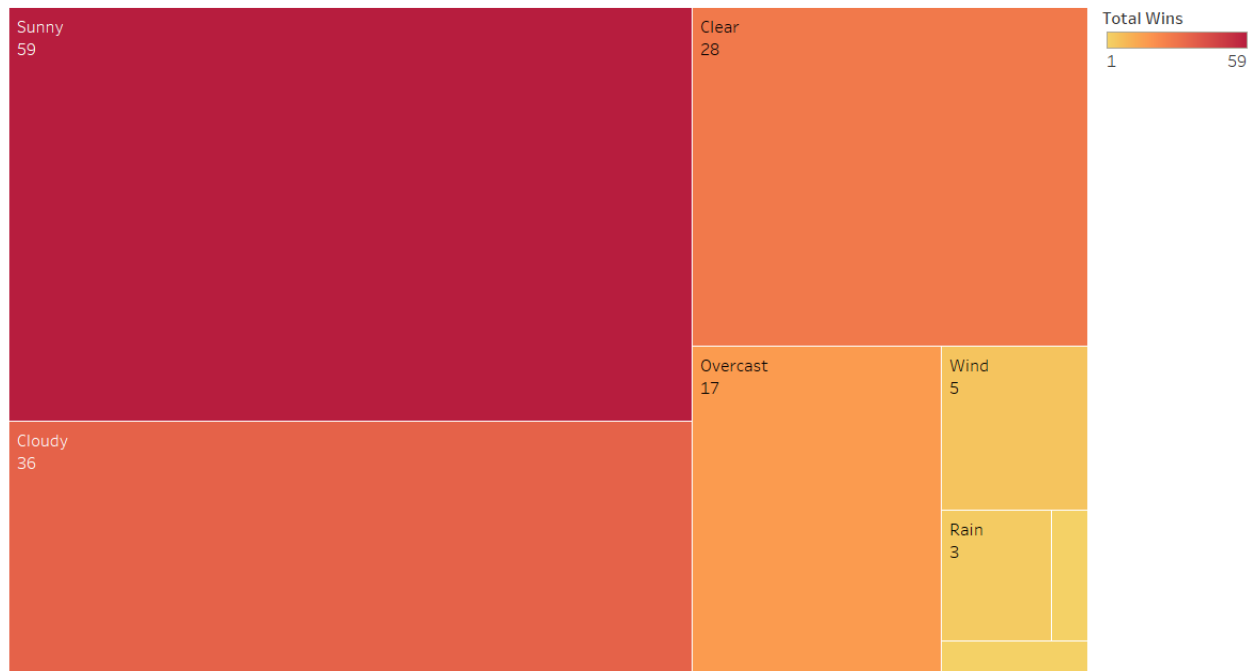
Which team attracted most audience all years combined?



Team Name and Total Attendance. Color shows Total Attendance. Size shows Total Attendance. The marks are labeled by Team Name and Total Attendance. The data is filtered on sum of Rst Attendance, which keeps non-Null values only. The view is filtered on Team Name, which keeps 10 of 154 members.

6.

How many times has UMD won so far in each weather condition?



Rst Weather and sum of Total Wins. Color shows sum of Total Wins. Size shows sum of Total Wins. The marks are labeled by Rst Weather and sum of Total Wins. The view is filtered on sum of Total Wins and Rst Weather. The sum of Total Wins filter keeps non-Null values only. The Rst Weather filter excludes Null.

Contributors:

[Jagadeesh Harinarayanan](#)

[Tejaswini Kandula](#)

[Venkat Nikhil Yerramilli](#)