Writeup

August 5, 2022

Basketball Props

Motivation

In 2018 the federal ban on sports betting was overturned, this has led to sports gambling becoming an entirely new industry. Making smart and informed decisions with sports betting is incredibly important, and researching which decisions are worth it can help give gamblers an advantage that prevents them from throwing money away. This database aims to provide an easy way for users to research NBA data to help efficiently make those smarter decisions, specifically on NBA game results and player props. Player Props being the total of a specific statistic (such as Nikola Jokic's rebounds in a game). In the past researching which bets had a history of hitting more often required users to have dozens of tabs open and clicking different windows to bring up multiple results at the same time. This database and the queries that I provide will give users a simple way to research which bets have a history of happening more often.

Project Description

This project contains three python notebooks in addition to this one.

tablesetup.ipynb handles the creation of the sql tables, triggers, and indices, it then loads in data from external CSV files and inserts them into the SQL Tables

This file contains the following actions:

Load in our required libraries and then connect to our provided database

Create our Tables, after the creation we add some constraints

Create appropriate Triggers for our Tables

Create appropriate Indices for our Tables

Load in external CSV files into Pandas Dataframe Objects

Remove Unneccessary columns and nan values from our Dataframes

Convert each Dataframe to a Dictionary

Insert the values from each Dictionary to the SQL Tables

tablemodifications.ipynb handles the testing of constraints, the testing of triggers through deletions, the testing of triggers a duplicate value, the testing of inserting a duplicate value

This file contains the following actions:

Load in the required libraries and connect to the database

Test that the constraints introduced in table hold by checking a record, and trying a duplicate insertion expecting a failure

Test each trigger that was created in tablesetup.ipynb

queries.ipynb handles the queries that are executed against the database

This file contains the following queries and is the main purpose of the project:

The following queries are executed

The Team Roster for a Specified Team in each season

Career Averages for Highest Scoring Players

Players who Averaged 10,5,5

Best Individual Seasons

Season Averages for a Specified Player

Player Props against a Specific Team

Win Percentage of one Specific Team against another Specific Team

Required Items Checklist

The following items were required for this project, a brief description of how and where that requirement is met is provided below:

Multiple Tables - tablesetup.ipynb, four tables were created for this project Teams, Players, Games, Games_Details

Relationships between Tables - tablesetup.ipynb, in the creation of tables each table other than Teams references another table

Show SQL Statments for table creation, insertion of initial data, updates and queries - tablesetup.ipynb contains table creation and insertion of initial data with acccompanying code, tablemodification.ipynb contains updates of data, queries.ipynb contains sql queries against the tables

Table Creation - tablesetup.ipynb, the four tables are created

Constraints - tablesetup.ipynb, Teams has a primary key, Games_Details and Games are both given composite primary keys tablemodifications.ipynb contains testing of these constraints

Indexes - tablesetup.ipynb, contains indices that were created to assist with the more common queries

Queries - queries.ipynb, filled entirely with queries

Joins between Tables - queries.ipynb, utilizes joins between tables in most of its queries

Grouping Results - queries.ipynb, groups results in most of its queries

Updates - tablesetup.ipynb, update triggers are implemented to update Games_Details when Players is updated, this trigger is tested in tablemodifications.ipynb

Deleting Items that are Foreign Keys in other tables - tablesetup.ipynb, deletion triggers are implemented here, those triggers are testested in tablemodifications.ipynb

Main Queries

This project computes a multitude of queries, from player season averages, to career averages, and much more. Each individual query is discussed more in-depth in the queries.ipynb notebook. Here I'd like to go over the two main queries that handle the motiviation behind the project

The first query Player Props was the main motivation behind the project. This query returns the player performances of a specific player against a specific team.

This query is demonstrated by searching for Jaylen Brown performances against the Brooklyn Nets

First a CTE is created of all GAME_IDs where the nets were either the home or away team

Next Games_Details is queries to find the specific player-game records involving Jaylen Brown

These two are merged together in one query to find Jaylen Brown's performances against the Brooklyn Nets

[21]:
```
%%sql

    with nets_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Nets") OR
    TEAM_ID_away = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Nets"))

    SELECT * FROM nets_games LIMIT 10;
```

```
 * mysql://jema2714:***@applied-sql.cs.colorado.edu:3306/jema2714
10 rows affected.
```

[21]:

GAME_ID
22100993
22100975
22100960
22100941
22100928
22100726
22100910
22100890
22100464
22100877

[22]:
```
%%sql

    SELECT g.GAME_DATE_EST, gd.PLAYER_NAME, gd.PTS, gd.REB, gd.AST, gd.FG_PCT, gd.FG3_PCT, gd.FT_PCT
    FROM Games_Details gd LEFT JOIN Games g ON gd.GAME_ID = g.GAME_ID
    WHERE gd.PLAYER_NAME = "Jaylen Brown"
    ORDER BY GAME_DATE_EST DESC
    LIMIT 10;
```

```
 * mysql://jema2714:***@applied-sql.cs.colorado.edu:3306/jema2714
10 rows affected.
```

[22]:

GAME_DATE_EST	PLAYER_NAME	PTS	REB	AST	FG_PCT	FG3_PCT	FT_PCT
2022-03-11	Jaylen Brown	22	8	4	0.364	0.125	0.714
2022-03-09	Jaylen Brown	15	5	5	0.375	0.125	0.4
2022-03-06	Jaylen Brown	21	4	5	0.471	0.5	1.0
2022-03-01	Jaylen Brown	2	1	1	1.0	0.0	0.0
2022-02-27	Jaylen Brown	23	4	8	0.625	0.4	0.25
2022-02-26	Jaylen Brown	27	3	0	0.417	0.167	1.0
2022-02-24	Jaylen Brown	18	1	6	0.5	0.2	0.625
2022-02-16	Jaylen Brown	31	2	6	0.571	0.5	0.8
2022-02-15	Jaylen Brown	29	8	3	0.588	0.714	0.667
2022-02-13	Jaylen Brown	17	9	3	0.375	0.333	0.75

[24]:
```
%%sql

    with nets_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Nets") OR
    TEAM_ID_away = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Nets"))

    SELECT g.GAME_DATE_EST, gd.PLAYER_NAME, gd.PTS, gd.REB, gd.AST, gd.FG_PCT, gd.FG3_PCT, gd.FT_PCT
    FROM Games_Details gd
    LEFT JOIN Games g ON gd.GAME_ID = g.GAME_ID
    WHERE gd.PLAYER_NAME = "Jaylen Brown" AND gd.GAME_ID IN (SELECT * FROM nets_games)
    ORDER BY g.GAME_DATE_EST DESC
```

```
 * mysql://jema2714:***@applied-sql.cs.colorado.edu:3306/jema2714
4 rows affected.
```

[24]:

GAME_DATE_EST	PLAYER_NAME	PTS	REB	AST	FG_PCT	FG3_PCT	FT_PCT
2022-03-06	Jaylen Brown	21	4	5	0.471	0.5	1.0
2022-02-24	Jaylen Brown	18	1	6	0.5	0.2	0.625
2022-02-08	Jaylen Brown	22	7	9	0.75	0.333	0.75
2021-11-24	Jaylen Brown	13	5	0	0.333	0.0	1.0

If the above image does not load please see PlayerPropsQuery.png in the images folder for look at the code and results of the Player Props Query

The second main query implemented was to calculate the win percentages for one specified team against another.

This query is demonstrated by finding the percentage of the last ten games that the Chicago Bulls won against the Philadelphia 76ers

First two CTEs are created one for Philadelphia GAME_IDs and one for Chicago GAME_IDs.

Next a CTE of the last ten GAME_IDs where Chicago Played Philadelphia is Created

Next CTEs are calculated for the count of the games where each team won at home and away

Chicago's wins are summed together for home and away, this number is converted into a percentage to represent the percentage of the last 10 games that Chicago played against Philadelphia where Chicago won

```
%%sql

    with chicago_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Bulls") OR
    TEAM_ID_away = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Bulls")),
    philly_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "76ers") OR
    TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "76ers"))


    SELECT *
    FROM Games g
    WHERE g.GAME_ID in (SELECT * FROM philly_games) AND g.GAME_ID in (SELECT * FROM chicago_games)
    ORDER BY g.GAME_DATE_EST DESC
    LIMIT 10;
```

```
 * mysql://jema2714:***@applied-sql.cs.colorado.edu:3306/jema2714
10 rows affected.
```

GAME_DATE_EST	GAME_ID	TEAM_ID_home	SEASON	PTS_home	FG_PCT_home	FT_PCT_home	FG3_PCT_home	AST_home	REB_home	T
2022-03-07	22100969	1610612755	2021	121.0	0.488	0.794	0.4	29	38	
2021-11-03	22100111	1610612755	2021	103.0	0.494	0.714	0.429	24	35	
2021-02-19	22000450	1610612755	2020	112.0	0.471	0.793	0.333	23	53	
2020-02-09	21900788	1610612755	2019	118.0	0.5	0.719	0.407	29	44	
2020-01-17	21900620	1610612755	2019	100.0	0.488	1.0	0.31	30	44	
2019-04-10	21801224	1610612755	2018	125.0	0.559	0.6	0.462	30	47	
2018-10-18	21800014	1610612755	2018	127.0	0.473	0.9	0.333	30	55	
2018-01-24	21700704	1610612755	2017	115.0	0.519	0.75	0.5	31	48	
2017-04-06	21601173	1610612755	2016	90.0	0.363	0.773	0.212	19	50	
2016-11-25	21600229	1610612755	2016	89.0	0.397	0.818	0.409	18	48	

```
%%sql

    with chicago_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Bulls") OR
    TEAM_ID_away = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "Bulls")),
    philly_games as (SELECT GAME_ID FROM Games
    WHERE TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "76ers") OR
    TEAM_ID_home = (SELECT TEAM_ID
    FROM Teams t
    WHERE NICKNAME = "76ers")),
    last_ten_philly_chicago as (
        SELECT * FROM Games g
        WHERE g.GAME_ID in (SELECT * FROM philly_games) AND g.GAME_ID in (SELECT * FROM chicago_games)
        ORDER BY g.GAME_DATE_EST DESC
        LIMIT 10),
    philly_home_wins as (
        SELECT COUNT(HOME_TEAM_WINS) as wins
        FROM last_ten_philly_chicago
        WHERE last_ten_philly_chicago.HOME_TEAM_WINS = 1 and last_ten_philly_chicago.TEAM_ID_home = (SELECT TEAM
    philly_away_wins as (
        SELECT COUNT(HOME_TEAM_WINS) as wins
        FROM last_ten_philly_chicago
        WHERE last_ten_philly_chicago.HOME_TEAM_WINS = 0 and last_ten_philly_chicago.TEAM_ID_away = (SELECT TEAM
    chicago_home_wins as (
        SELECT COUNT(HOME_TEAM_WINS) as wins
        FROM last_ten_philly_chicago
        WHERE last_ten_philly_chicago.HOME_TEAM_WINS = 1 and last_ten_philly_chicago.TEAM_ID_home = (SELECT TEAM
    chicago_away_wins as (
        SELECT COUNT(HOME_TEAM_WINS) as wins
        FROM last_ten_philly_chicago
        WHERE last_ten_philly_chicago.HOME_TEAM_WINS = 0 and last_ten_philly_chicago.TEAM_ID_away = (SELECT TEAM
    )

    SELECT (sum(wins)/10)*100 "Chicago Win Percentage"
    FROM ( SELECT * FROM chicago_away_wins UNION ALL SELECT * FROM chicago_home_wins ) chicago_wins;
```



```
 * mysql://jema2714:***@applied-sql.cs.colorado.edu:3306/jema2714
1 rows affected.
```

Chicago Win Percentage
20.0000

If the above image does not load please see PlayerPropsQuery.png in the images folder for look at the code and results of the Player Props Query

Reminder : To look at these queries and the many more that were used for this project (team rosters, player averages etc.) please see the queries.ipynb file

Note that the database used was unable to store all of the data contained in the CSV files, while the Player Props query and Team Win Percentage query both work as expected, the others do not as they are missing several games and several games_details records. This has been tessted locally and the other queries perform as expected. That can be confirmed in Rough_Draft.ipynb which contains a pandassql version of the queries and operates as expected (only explore it if you want to see those results otherwise the file is not relevant)

[ ]: