ABSTRACT

We are working on a NBA and trying to build a NoSQL database. So We will be extracting data from 3 diffrent data sources and then will be using them together to create a consistent database. (these part has been done in previous assignments and stored in sql database) We will perform several operations over the dataset extracted to make the data clean and error free and consistent. After that we will be developing a database from using the extracted source data and display.

First task is to migrate our data we collected in previous assignments to build sql databse to no sql database

Importing Libraries necessary

In [2]:
```
import pandas as pd
import json
import requests
```

Migrating data to mongoDB

Populating the mongoDB database with data we got in previous assignments

In [3]:
```
# bring the data from previous assignments
players = pd.read_csv('./data/Players.csv')
teams = pd.read_csv('./data/Teams.csv')
player_statistics = pd.read_csv('./data/PlayersStats.csv')
team_statistics = pd.read_csv('./data/TeamsStats.csv')

players.head()
```

Out[3]:

	PlayerID	FirstName	LastName	Height	Weight	Team	BirthDate
0	20000439	Nene	Hilario	83	250	HOU	1982-09-13T00:00:00
1	20000441	Bradley	Beal	75	207	WAS	1993-06-28T00:00:00
2	20000442	John	Wall	76	210	WAS	1990-09-06T00:00:00
3	20000443	Otto	Porter	80	198	CHI	1993-06-03T00:00:00
4	20000452	Garrett	Temple	77	195	BKN	1986-05-08T00:00:00

In [4]:
```
player_statistics.head()
```

Out[4]:

	SCORING	GP	GS	MPG	PPG	PPS	HIGH
0	1 Embiid, Joel Embiid, J. C	34	34	30.8	23.0	1.44	38
1	2 Harris, Tobias Harris, T. SF	50	50	34.4	19.3	1.23	35
2	3 Simmons, Ben Simmons, B. PG	48	48	36.2	16.7	1.45	34
3	4 Richardson, Josh Richardson, J. SG	38	38	31.5	15.0	1.19	32
4	5 Horford, Al Horford, A. C	45	45	31.1	12.5	1.1	32

Player stats has column name wrong so we need to rename it and also need to audit the data because names are given with comma seperated values

In [5]:
```
player_statistics = player_statistics.rename(columns = {"SCORING":"player"})
player_statistics.head()
```

Out[5]:

	player	GP	GS	MPG	PPG	PPS	HIGH
0	1 Embiid, Joel Embiid, J. C	34	34	30.8	23.0	1.44	38
1	2 Harris, Tobias Harris, T. SF	50	50	34.4	19.3	1.23	35
2	3 Simmons, Ben Simmons, B. PG	48	48	36.2	16.7	1.45	34
3	4 Richardson, Josh Richardson, J. SG	38	38	31.5	15.0	1.19	32
4	5 Horford, Al Horford, A. C	45	45	31.1	12.5	1.1	32

Player stats table cleaning

removing multiple names in names coloums

In [6]:
```
p_stats_list = player_statistics.to_json(orient='records')
p_stats_list = json.loads(p_stats_list)

for player in p_stats_list:
    player['player'] = player['player'].split(",")[1].strip()

p_stats_list[0]
```

Out[6]:
```
{'player': 'Joel Embiid',
 'GP': 34,
 'GS': 34,
 'MPG': '30.8',
 'PPG': '23.0',
 'PPS': '1.44',
 'HIGH': 38}
```

In [7]:
```
players_list = players.to_json(orient='records')
players_list = json.loads(players_list)
# players_list[0]

teams_list = teams.to_json(orient='records')
teams_list = json.loads(teams_list)
teams_list[0]
```

Out[7]:
```
{'abbrevation': 'ATL', 'teamName': 'Atlanta Hawks'}
```

Cleaning team stats table for inserting in mongoDB

In [8]:
```
team_statistics.head()
```

Out[8]:

	SCORING	GP	PPG	PPS	PTS/POSS
0	1 Milwaukee MIL	49	120.0	1.32	1.14
1	2 Houston HOU	49	118.7	1.30	1.14
2	3 Dallas DAL	49	116.4	1.29	1.17
3	4 Washington WAS	48	115.7	1.26	1.12
4	5 Los Angeles LAC	49	115.4	1.28	1.13

team stats has column name wrong so we need to rename it and also need to audit the data because names are given with id team name and team abbrevation seperated by space

In [9]:
```
team_statistics = team_statistics.rename(columns = {"SCORING":"team"})
team_statistics.head()
```

Out[9]:

	team	GP	PPG	PPS	PTS/POSS
0	1 Milwaukee MIL	49	120.0	1.32	1.14
1	2 Houston HOU	49	118.7	1.30	1.14
2	3 Dallas DAL	49	116.4	1.29	1.17
3	4 Washington WAS	48	115.7	1.26	1.12
4	5 Los Angeles LAC	49	115.4	1.28	1.13

In [10]:
```
t_stats_list = team_statistics.to_json(orient='records')
t_stats_list = json.loads(t_stats_list)

for team in t_stats_list:
    team['team'] = team['team'].split(" ")[1].strip()

t_stats_list[0]
```

Out[10]:
```
{'team': 'Milwaukee', 'GP': 49, 'PPG': 120.0, 'PPS': 1.32, 'PTS/POSS': 1.1
4}
```

In [11]:
```
from pymongo import MongoClient

def get_db(db_name):
    # For local use
    from pymongo import MongoClient
    client = MongoClient('localhost:27017')
    db = client[db_name]
    return db

# Getting nba_database from mongoDB
nba_Database= get_db("nba_Database")

# Getting collections from nba database
players_collection = nba_Database.players
teams_collection = nba_Database.teams
player_stats_collection = nba_Database.player_stats
team_stats_collection = nba_Database.team_stats
```

In [12]:
```
for player in players_list:
    players_collection.insert_one(player)
```

In [13]:
```
for team in teams_list:
    teams_collection.insert_one(team)
```

In [14]:
```
for p_stat in p_stats_list:
    player_stats_collection.insert_one(p_stat)
```

In [15]:
```
for t_stat in t_stats_list:
    team_stats_collection.insert_one(t_stat)
```

Now that we have inserted data into MongoDB Check MongoCompass to see if data is inserted

Getting Data from social media (twitter) for Players

In [16]:
```
# importing libraries required for downloading data
import tweepy
import twitter

# keys for accesing twitter api
consumerKey = 'lsDkpS786UbLVbxkYOONbeik5'
consumerSecret = 'BhSSMMpwmc6KtFPXWVbzVQezJ1osNthgQHaNDxgrg6TzQhSNUy'
ACCESS_TOKEN = '2483851159-GSH3yLT4Ilon3fD6lfpAYZPRZCaGjP30iAlOQS3'
ACCESS_SECRET = 'j6WQUKvxVSNkKsPMoKv9zrqDvuERqD0sVloCBS1gOT5Vn'

auth = tweepy.OAuthHandler(consumer_key=consumerKey, consumer_secret=consumer
#Connect to the Twitter API using the authentication
api = tweepy.API(auth)
```

In [17]:
```
## trying to get player names to populate tweets abt them in database
for player in p_stats_list:
    player['player'] = ''.join(e for e in player['player'] if e.isalnum())

# checking if our player is made useful to search as tag
p_stats_list[0]['player']
```

Out[17]: 'JoelEmbiid'

In [28]:
```
results = []

try:
    #Get the first 5000 items based on the search query
    for player in p_stats_list:
        search_q = '%#'+ player['player']
        for tweet in tweepy.Cursor(api.search, q=search_q, since='2019-04-21'
            results.append(tweet)
except tweepy.error.TweepError:
    raise

# Verify the number of items returned print
len(results)
```

In [38]:

```
def toDataFrame(tweets):

    DataSet = pd.DataFrame()
    DataSet['tweetID'] = [tweet.id for tweet in tweets]
    DataSet['tweetText'] = [tweet.text.encode('utf-8') for tweet in tweets]
    DataSet['tweetRetweetCt'] = [tweet.retweet_count for tweet in tweets]
    DataSet['tweetFavoriteCt'] = [tweet.favorite_count for tweet in tweets]
    DataSet['tweetSource'] = [tweet.source for tweet in tweets]
    DataSet['tweetCreated'] = [tweet.created_at for tweet in tweets]
    DataSet['hashTags'] = [tweet.entities.get('hashtags') for tweet in tweets

    return DataSet

#Pass the tweets list to the above function to create a DataFrame
socialMediaData = toDataFrame(results)
```

In [39]:

```
socialMediaData.head()
```

Out[39]:

	tweetID	tweetText	tweetRetweetCt	tweetFavoriteCt	tweetSource	tweetC
0	1246204330463682560	b'RT @JeffSkversky: \xf0\x9f\x91\x8f Sixers st...	9	0	Twitter for Android	2020 22
1	1246189799326842888	b'RT @JeffSkversky: \xf0\x9f\x91\x8f Sixers st...	9	0	Twitter for iPhone	2020 21
2	1246169961527005185	b'@JoelEmbiid is quickly becoming my favorite ...	0	0	Twitter Web App	2020 20
3	1246162090211098625	b'RT @JeffSkversky: \xf0\x9f\x91\x8f Sixers st...	9	0	Twitter for iPhone	2020 19
4	1246161802494447626	b'RT @JeffSkversky: \xf0\x9f\x91\x8f Sixers st...	9	0	Twitter for Android	2020 19

In [40]:

```
def turndf_to_json(df):
    json_list = df.to_json(orient='records')
    json_list = json.loads(json_list)
    return json_list

socialMediaData = turndf_to_json(socialMediaData)
```

In [41]:
```
# testing if data is converted to json
socialMediaData[0]
```

Out[41]:
```
{'tweetID': 1246204330463682560,
 'tweetText': 'RT @JeffSkversky: 👏 Sixers star Joel Embiid, 76ers Managing
Partner Josh Harris, and Co-Managing Partner David Blitzer Join Forces to P
rovi…',
 'tweetRetweetCt': 9,
 'tweetFavoriteCt': 0,
 'tweetSource': 'Twitter for Android',
 'tweetCreated': 1585953240000}
```

Entering socialMedia data into mongodb database

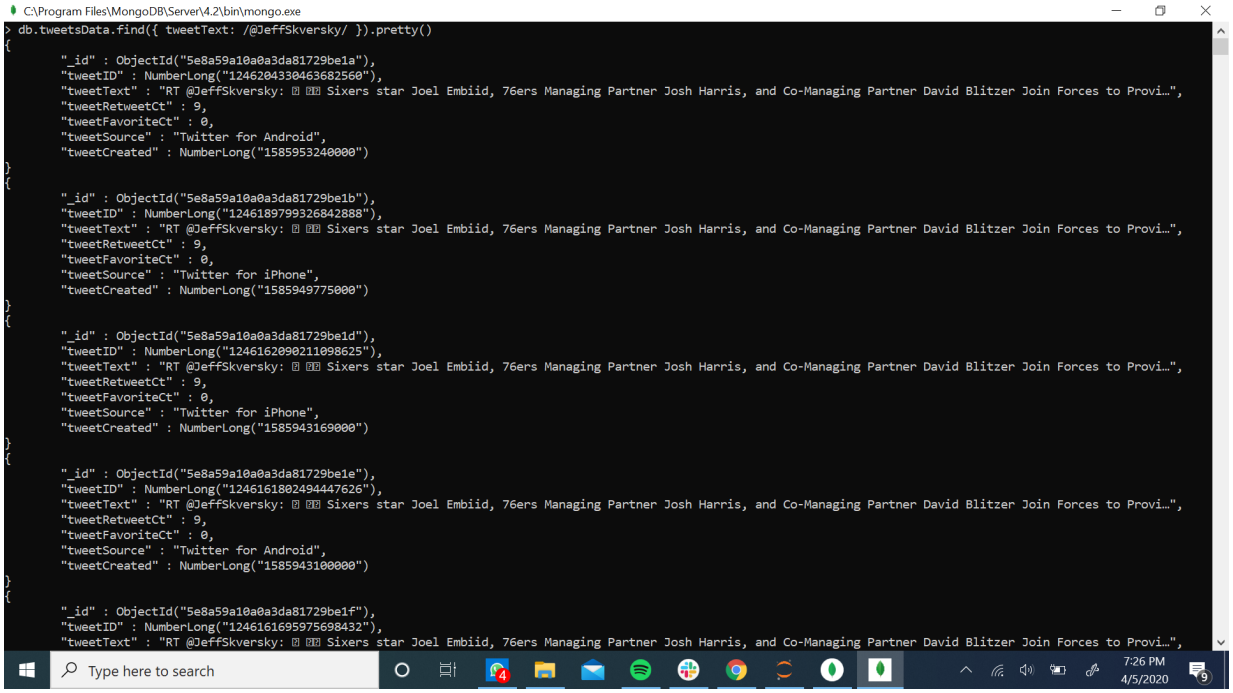
In [42]:
```
socialMediaCollection = nba_Database.tweetsData

for data in socialMediaData:
    socialMediaCollection.insert_one(data)
```

Check MongoClient to verify if data is populated

Search for tweets with tag JeffSkversky

Query: First move to nba_Database

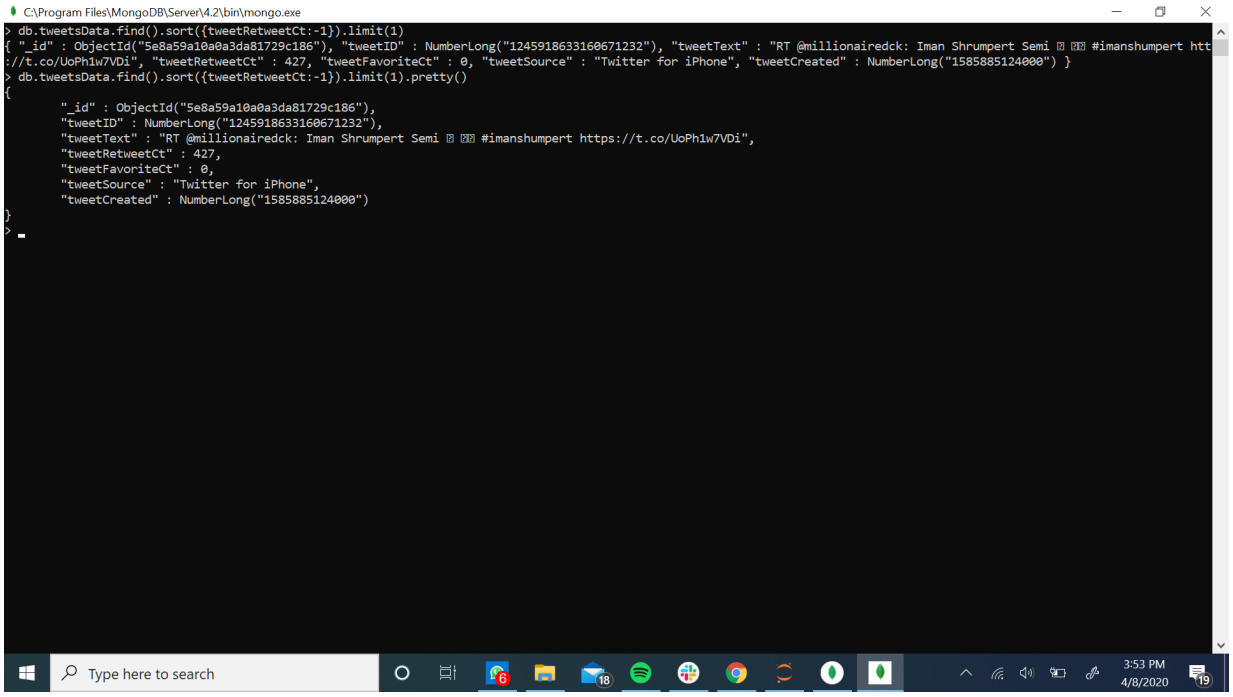
- use nba_Database
- db.tweetsData.find({ tweetText: /@JeffSkversky/ }).pretty()



Popular and Trending Hashtags

Query: First move to nba_Database

- use nba_Database
- db.tweetsData.find().sort({ tweetRetweetCt: -1 }).limit(1).pretty()



Query: First move to nba_Database

- use nba_Database
- db.tweetsData.find().sort({ tweetFavoritesCt: -1 }).limit(1).pretty()

```
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.tweetsData.find().sort({tweetFavoriteCt:-1}).limit(1).pretty()
{
        "_id" : ObjectId("5e8a59a20a0a3da81729c1d4"),
        "tweetID" : NumberLong("1245723924614135808"),
        "tweetText" : "□ □ □ □ □ \n#IND\n#IndianaPacers\n#VictorOladipo\n#MylesTurner\n#DomantasSabonis\n#MalcolmBrogdon\n#TJwarren\n#nbaart\n#artwork… https://t.co/TCd4kYg
4fR",
        "tweetRetweetCt" : 11,
        "tweetFavoriteCt" : 93,
        "tweetSource" : "Twitter for iPhone",
        "tweetCreated" : NumberLong("1585838702000")
}
>
```

Query: In order to find which users are similar to other users we jus select the tweets related to particular hashtag so that we can get to know who also have same hashtag First move to nba_Database

- use nba_Database
- db.tweetsData.find({ tweetText: /#nba/ }).pretty()

```
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.tweetsData.find({ tweetText: /#nba/ }).pretty()
{
        "_id" : ObjectId("5e8a59a10a0a3da81729be25"),
        "tweetID" : NumberLong("1245660943016554500"),
        "tweetText" : "RT @Dee_Black_MMA_: Tobias looks to be coming into his own with Embiid out. #nba #nbatv #basketball #tobiasharris #philly #76ers #sixers #p…",
        "tweetRetweetCt" : 30,
        "tweetFavoriteCt" : 0,
        "tweetSource" : "Twitter for iPhone",
        "tweetCreated" : NumberLong("1585823686000")
}
{
        "_id" : ObjectId("5e8a59a10a0a3da81729be26"),
        "tweetID" : NumberLong("1244484732995461121"),
        "tweetText" : "RT @Dee_Black_MMA_: When Philly is 100% healthy, they can beat anyone - 2.12.20. #nba #nbatv #basketball #philadelphia #76ers #sixers #phil…",
        "tweetRetweetCt" : 18,
        "tweetFavoriteCt" : 0,
        "tweetSource" : "Twitter for iPhone",
        "tweetCreated" : NumberLong("1585543256000")
}
{
        "_id" : ObjectId("5e8a59a10a0a3da81729be2f"),
        "tweetID" : NumberLong("1245743181452509184"),
        "tweetText" : "Ben Simmons Sister: NBA Coach Is 'Downlow' - 'He Cheated On Me w/ Man' #bensimmons #entertainmentnews #nbacoach\nhttps://t.co/qRM9mEi6Je",
        "tweetRetweetCt" : 0,
        "tweetFavoriteCt" : 0,
        "tweetSource" : "FS Poster",
        "tweetCreated" : NumberLong("1585843293000")
}
{
        "_id" : ObjectId("5e8a59a10a0a3da81729be30"),
        "tweetID" : NumberLong("1245732644614221826"),
        "tweetText" : "#kyrieirving #joelembiid #bensimmons #76ers #hoops #nba #basketball #espn #jellyfam□ □□ #zionwilliamson #kevindurant… https://t.co/U5pQrWnkjB",
        "tweetRetweetCt" : 0,
        "tweetFavoriteCt" : 0,
        "tweetSource" : "Twitter for Android",
        "tweetCreated" : NumberLong("1585840781000")
}
{
        "_id" : ObjectId("5e8a59a10a0a3da81729be43"),
        "tweetID" : NumberLong("1244904111851569157"),
        "tweetText" : "RT @Dee_Black_MMA_: Milk Shake Milton is ballin' @sixers #sixers #shakemilton #nba #nbatv #basketball #philadelphiaseventysixers #philly #7…",
```

In [ ]:

```
# Create a dictionary
d = dict()

# Saving HashTag name and its count in all the tweets and saving as keys and
for tweet in range(0, len(results)):
    hashTag = results[tweet].entities.get('hashtags')
    for i in range(0, len(hashTag)):
        HashTag = hashTag[i]['text']
        if HashTag in d:
            d[HashTag] = d[HashTag] + 1
        else:
            d[HashTag] = 1

# Dictionary converted to a Dataframe
HashTags = pd.DataFrame(list(d.items()),columns = ['HashTag','Count'])
HashTags
```

In [ ]:

```
#Sorting the dataframe as per the count
HashTags = HashTags.sort_values(by='Count', ascending=False)
HashTags
```

AUDIT VALIDITY/ACCURACY

We say data is accurate only when it is neat and with no junk values. By using various commands like drop, del and lambda functions, all the unwanted junk values were deleted from the above rows and columns which gives valid and accuarate data report.

AUDIT COMPLETNESS

In real world, when a list of teams stats, player stats, player information, team information from a particular Player or Team or season is requested, a list of it will be displayed or presented, similarly when we compare it with above data too, we get proper real time data showing correct information for all the Matches played by teams/players. This can be extended for multiple seasons like which team is popular in that season.

AUDIT CONSISTENCY/UNIFORMITY

The datasets which have been used in this assignment show a uniform relationship between each of the dataset since they are linked to each other by a common attribute.

REPORT

Files used from previous assignments:

1. PlayersTable.csv
2. Teams.csv
3. PlayersStats.csv
4. TeamStats.csv

Description for converting sql to nosql database

Functions :

1. turndf_to_json to turn data from dataframe which is taken from csv file into json since mongodb accepts data as json format we need to convert all the data into json format
2. get_db to connect mongoDB database

Data is reformatted to fit into a conceptual model. Data gathered from different sources Web API, Web scraping, Raw file (from kaggle datasets) and are used to fit into a conceptual model.

Code used:
Step 1. Extraction of Data
3 main methods were used for the extraction of data:

1. Using the API:
   Here the API key for the site was used and libraries like: request to access the website using the URL (https://api.sportsdata.io (https://api.sportsdata.io)) and API key (")
   json to convert the file into json format
   pandas to create data frames from the raw data
2. Using the website to scrap the data
   Here the data was extracted using the sites data directly using the libraries like:
   request to access the website using the URL (https://www.foxsports.com (https://www.foxsports.com))
   BeautifulSoup to scrape the contents of the website
   find() and find_all() methods were used to find the desired content in the system used dynamic links to scrape all the data contained in 28 pages of the website
3. By loading the csv file:
   Here the data was extracted using a csv file on the system using the libraries like:
   Pandas to a read the csv file and load it into data frames
   read_csv method is used to read .csv file

Step 2. Cleaning and Auditing Data
To gain knowledge about the dataset we used various methods like
describe, isnull, any, shape, columns, is_unique, info, iloc, loc, os

CONCLUSION

Primary focus of this assignment is to learn how to get the transfer data from sql to mongoDB. cleaning of data, checking null values present in the data, data munging and to reformat the data to fit a conceptual database schema.

CONTRIBUTION

Your contribution towards project. How much code did you write and how much you took from other site or some other source.

I contributed By Own: 40%
Teammate contribution: 40%
Provided by the prof github projects, w3schools, geeksforgeeks and stackoverflow : 20%

CITATIONS

Sources from where you have gained knowledge or used codes, data. It may include Web links, github links, code taken from somewhere etc.

1. https://www.infoworld.com/article/2608083/do-twitter-analysis-the-easy-way-with-mongodb.html?page=3 (https://www.infoworld.com/article/2608083/do-twitter-analysis-the-easy-way-with-mongodb.html?page=3)
2. https://www.w3schools.com/python/python_mongodb_find.asp (https://www.w3schools.com/python/python_mongodb_find.asp)
3. https://docs.scipy.org/doc/numpy/user/basics.indexing.html (https://docs.scipy.org/doc/numpy/user/basics.indexing.html)

LICENSE

In [ ]: