NBA Player Statistics ETL Pipeline

In [10]:
```
# Import Necessary Classes
import sqlalchemy
from bs4 import BeautifulSoup
import requests
import pandas as pd
import time
import schedule
```

In [29]:
```
# TODO: EXTRACT
# Data Scraping
def automate():
    url = 'https://www.basketball-reference.com/leagues/NBA_2025_per_game.html'
    page = requests.get(url)
    soup = BeautifulSoup(page.text, features="html.parser")

    # print(soup)

    table = soup.find(table=soup.find('table', {'id': 'per_game_stats'}))

    headers = table.findAll('th')
```

In [30]:
```
    # print(headers)

    '''
    ran into an issue when extracting data. essentially
    the rank column was throwing everything off so I decided to delete it.
    '''
    # filter unnecessary column headers out
    column_names_tables = [i.text.strip() for i in headers if i.text.strip() and no
    # now delete rank column
    column_names_tables = column_names_tables[1:]
```

In [31]:
```
    # print(column_names_tables)

    # TODO: TRANSFORM
    # Data Cleaning

    df = pd.DataFrame(columns=column_names_tables)

    column_data = table.findAll('tr')

    # print(column_data)

    for row in column_data[1:]:
        row_data = (row.find_all('td'))
        individual_row_data = [i.text.strip() for i in row_data]

        length = len(df)
        df.loc[length] = individual_row_data
```

In [32]:

```
# create rank column
df.insert(0, 'Rank', df.index + 1)

# fix names
df.loc[df['Player'] == 'Nikola JokiÄ', 'Player'] = 'Nikola Jokic'
df.loc[df['Player'] == 'Luka DonÄiÄ', 'Player'] = 'Luka Doncic'
df.loc[df['Player'] == 'Nikola VuÄeviÄ ', 'Player'] = 'Nikola Vucevic'
df.loc[df['Player'] == 'Kristaps PorziÅÄ£is', 'Player'] = 'Kristaps Porzingis'
df.loc[df['Player'] == 'Dennis SchrÃ¶der', 'Player'] = 'Dennis Schroder'
df.loc[df['Player'] == 'Jonas ValanÄiÅ«nas', 'Player'] = 'Jonas Valanciunas'
df.loc[df['Player'] == 'Bogdan BogdanoviÄ', 'Player'] = 'Bogdan Bogdanovic'
df.loc[df['Player'] == 'Nikola JoviÄ', 'Player'] = 'Nikola Jovic'
df.loc[df['Player'] == 'Jusuf NurkiÄ', 'Player'] = 'Jusuf Nurkic'
df.loc[df['Player'] == 'Tidjane SalaÃ¼n', 'Player'] = 'Tidjane Salaun'
df.loc[df['Player'] == 'Moussa DiabatÃ©', 'Player'] = 'Moussa Diabaté'
df.loc[df['Player'] == 'Dario Å ariÄ', 'Player'] = 'Dario Saric'
df.loc[df['Player'] == 'Vlatko ÄanÄar', 'Player'] = 'Vlatko Canchar'
df.loc[df['Player'] == 'Lester QuiÃ±ones', 'Player'] = 'Lester Quinones'
df.loc[df['Player'] == 'Armel TraorÃ©', 'Player'] = 'Armel Traore'
df.loc[df['Player'] == 'Karlo MatkoviÄ', 'Player'] = 'Karlo Matkovic'

print(df)

# save to csv
df.to_csv(r'C:\Users\Anthony Arthur\OneDrive - Cleveland State University\nba20
```

```
     Rank                  Player Age Team Pos   G  GS    MP    FG   FGA  \
0       1    Giannis Antetokounmpo  30  MIL  PF  27  27  35.2  12.7  21.2
1       2            Nikola Jokic  29  DEN   C  31  31  37.1  12.0  21.7
2       3  Shai Gilgeous-Alexander  26  OKC  PG  35  35  34.5  11.0  21.1
3       4             LaMelo Ball  23  CHO  PG  23  23  33.7  10.3  24.4
4       5          Paolo Banchero  22  ORL  PF   5   5  36.4   9.6  19.4
..    ...                     ...  ..  ...  ..  ..  ..   ...   ...   ...
525   526             Mac McClung  26  ORL  SG   1   0   5.0   0.0   0.0
526   527           Justin Minaya  25  POR  SF   1   0   6.0   0.0   0.0
527   528             Riley Minix  24  SAS  SF   1   0   7.0   0.0   1.0
528   529             Cole Swider  25  DET  SF   2   0   6.5   0.0   2.5
529   530          League Average

     ...  ORB  DRB   TRB  AST  STL  BLK  TOV   PF   PTS Awards
0    ...  2.0  9.6  11.6  5.9  0.8  1.5  3.4  2.4  32.3
1    ...  3.5  9.5  13.0  9.7  1.7  0.6  3.3  2.1  31.5
2    ...  0.9  4.7   5.6  6.1  2.0  1.1  2.6  2.1  31.3
3    ...  0.9  4.3   5.2  7.3  1.3  0.2  4.0  3.7  29.8
4    ...  2.4  6.4   8.8  5.6  0.6  0.8  2.2  2.6  29.0
..   ...  ...  ...   ...  ...  ...  ...  ...  ...   ...    ...
525  ...  0.0  1.0   1.0  2.0  0.0  0.0  0.0  0.0   0.0
526  ...  0.0  1.0   1.0  0.0  0.0  0.0  0.0  1.0   0.0
527  ...  0.0  2.0   2.0  0.0  0.0  0.0  0.0  0.0   0.0
528  ...  0.0  1.0   1.0  0.5  0.0  0.0  0.0  0.5   0.0
529  ...

[530 rows x 31 columns]
```

In [33]:
```
    # TODO: LOAD
    engine = sqlalchemy.create_engine('mysql+pymysql:/[redacted]3

    df.to_sql(
        name="nbapergame",
        con=engine,
        index=False,
        if_exists='replace'
    )
```

Out[33]: 530

In [ ]:
```
schedule.every().monday.at("06:00").do(automate)

run = True

while run:
    schedule.run_pending()
    time.sleep(1)
```