STEPS TO SCRAPE & QUERY TRANSFERMARKT DATA FOR THE BCSG ROUND 2 DATA ENGINEER PROJECT

Pranav Natarajan

WHY TRANSFERMARKT?

1. Website tailored to obtain & maintain player valuation data for football across leagues.
2. Quantitative approach to valuation, backed by a qualitative discussion from the `Transfermarkt Community`[1,2].
3. Contains all the data we need for this project-both for teams and players!

[1] HTTPS://WWW.TRANSFERMARKT.CO.IN/TRANSFERMARKT-MARKET-VALUE-EXPLAINED-HOW-IS-IT-DETERMINED-/VIEW/NEWS/385100

[2] HTTPS://WWW.NYTIMES.COM/2021/08/12/SPORTS/SOCCER/SOCCER-FOOTBALL-TRANSFERMARKT.HTML

PROJECT STEPS

Step 1: Understand the Website Layout

Upon inspecting the HTML code for the Transfermarkt.us website(s) for LaLiga Clubs and players for 22/23, I noticed that all the data required for this project were in *tables*. I specifically decided to use the compact player data tables for each club in the 22/23 season as it had all the data required to complete the project, and thus would result in minimal data storage.

This, along with the specifications meant that I could use the ***requests***, ***bs4***, ***pandas*** & ***sqlite3*** modules on python to complete this project.

Step 2: Scrape & Format Club data table as a pandas dataframe

I then scraped the data from the LaLiga Clubs Page on Transfermarkt.us, storing the ***Name***, ***Squad*** Size, ***Avg. Squad Age***, No. of ***Foreigners***, ***Avg. Market Value ($)*** for a player on the team, and the ***Total Market Value ($)*** for the team as a pandas dataframe.

I then used pandas apply functions to reformat the market values from string to float, and changing ***Squad, Avg. Squad Age*** & ***Foreigners*** to a numeric data type.

PROJECT STEPS

Step 3: Scrape & Format compact player data tables as a pandas dataframe

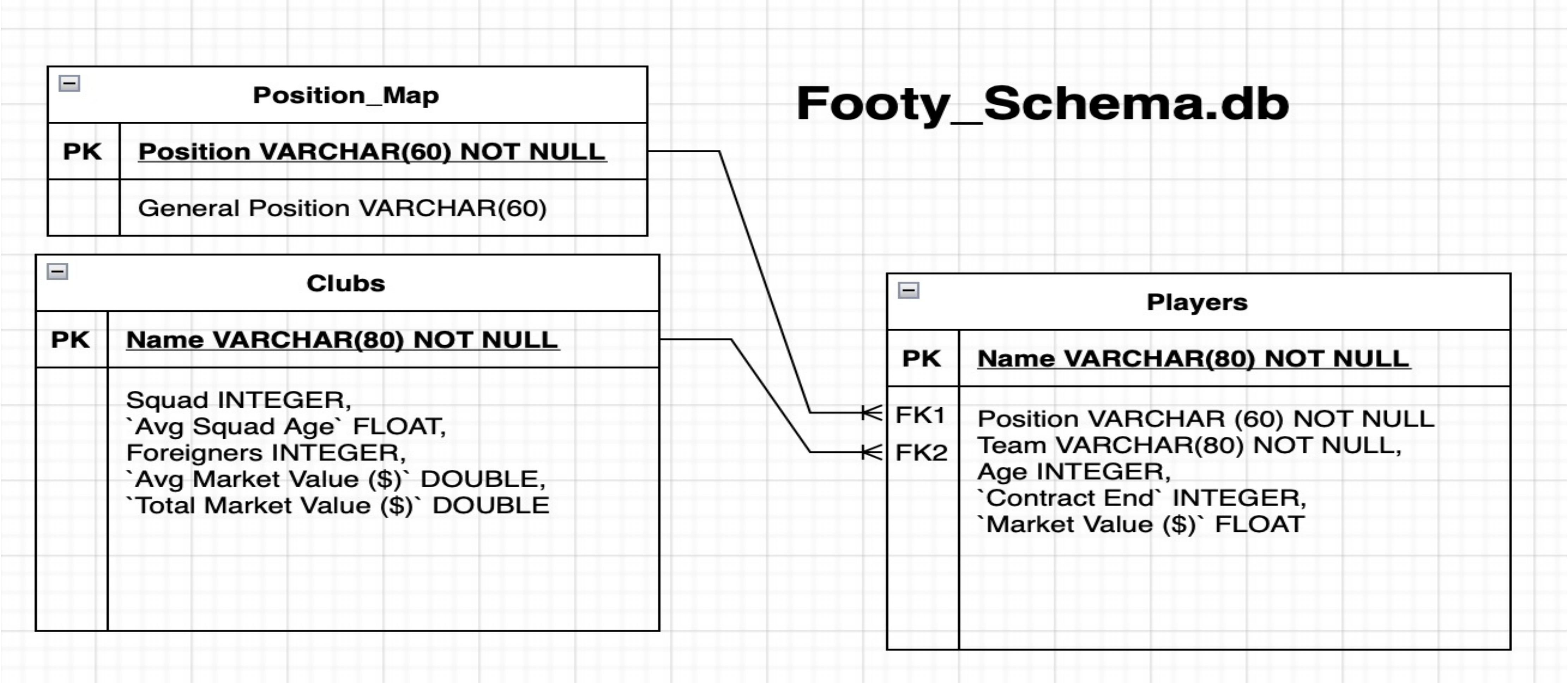
I then scraped the data from the respective Teams' pages on Transfermarkt.us, reformatting a list of lists to store the ***Name***, ***Position***, ***Age***, ***Contract End*** date, ***Market Value ($)***, and the ***Team (as a foreign key reference to the Clubs table)*** as a pandas dataframe.

I then used pandas apply functions to reformat the market value from string to float, and changed ***Age*** to a numeric data type.

Step 4: Create the Position Map pandas dataframe

I then created a pandas dataframe, with each ***Position*** being a primary key, while the more general positions (Goalkeepers, Defenders, Midfielders, Attackers) were the attribute.

The general positions were created using an apply function.



Footy_Schema.db

Position_Map	
PK	Position VARCHAR(60) NOT NULL
	General Position VARCHAR(60)

Clubs	
PK	Name VARCHAR(80) NOT NULL
	Squad INTEGER, `Avg Squad Age` FLOAT, Foreigners INTEGER, `Avg Market Value ($)` DOUBLE, `Total Market Value ($)` DOUBLE

Players	
PK	Name VARCHAR(80) NOT NULL
FK1 FK2	Position VARCHAR (60) NOT NULL Team VARCHAR(80) NOT NULL, Age INTEGER, `Contract End` INTEGER, `Market Value ($)` FLOAT

STEP 5: CREATE THE TABLES ON THE SCHEMA FOOTY_SCHEMA.DB ON SQLITE

I used 3 tables, keeping in mind the normalization of data. Data was bulk inserted using INSERT INTO after converting their pandas dataframes to a list of row tuples using list(pd.itertuples())

PROJECT STEPS

Step 6: Write the query to get the total market value for each team by general position

I used *sqlite3* to write the query above, getting a 80 x 3 table. I then proceeded to save the results on the database and as a csv file called ”*query_results.csv*”.

Step 7: Create pure code file from ipynb notebook, and push relevant files to `BCSG_DataEngineerR2` repo on github

I created a .py file of all the code for this project from the notebook I completed this project in, and pushed the code files, footy_schema.db, query_results.csv, and this powerpoint presentation onto the repo.