Transfer market management system

<Abstract>

Every Summer and Winter, the FIFA allows football clubs to go through the transfer market. To improve their team, each of the football clubs make a scouting team especially operated during the transfer market. Fortunately, through the Football Manager 2019, I was familiar with the scouting team structure. However, playing with the Football Manager 2019 I realized that the game only lists the players I am interested but does not take full concern of the scouting team structure. Supposing that a real football club is offering for a management system of the scouting team for the upcoming transfer market, the project is aiming to build a simple command line program to manage the team and the scouting process going on between 1JAN20 to 14JAN20.

<Changes>

Compared to the first final project proposal, I decided to exclude tables explaining the relationship between different tables but rather used more foreign keys within the table. The first reason of deciding to use relationship table was to easily manage the relationship between different tables but this made the whole system look more complicated. As I deleted several tables and added foreign keys to depict the relation, the system was easier to manage. Also I excluded sorting some of player's statistics during the season and the tactical analysis as I wanted to focus more on the scouting team management system. Including the stats and the tactic analysis simply enlarged the size of the data I had to deal with while not giving any significant advantages to the project. Rather I added some functions like '—count' and '—list' which may improve the convenience of users.

<Data>

The data of the scouting team members are all made-up but the structure of the team and the relationship between the positions followed those that were suggested in the Football Manager 2019. There are mainly three positions in the team and each of the scouters in the positions has a single staff that assists to schedule the meetings with the players. The important point is that it is the staff who is managing the schedule while players are getting directly getting in touch with the scouters. This a bit weird relation confirmed me the importance of well-organized database management system for effective transfer market. All the tables were created based on the basic information that is concerned in the scouting team. I added the concept of the 'id' for each member to replace the SSN and randomly generated the phone numbers through https://www.randomlists.com/phone-numbers. The data of the real football players were brought from https://sofifa.com/. I included the position of the players to depict the feature that which positions the football clubs are mainly concentrating on for improvement. The schedule of the meetings can be mainly sorted by the date and type(transfer/loan) of the meeting. Again, the schedule is recognized by the staff id while players could be recognized by the scouter id. The

size of the tables follows the regular scouting team size of the football clubs. Yet, I increased the number of players that are interested to get more possible future sorting options.

<Functions>

- Help/Version

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --version
transfermarket_1.0

Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --help
usage: transfermarket.py [-h] [-v]
                         [--search | --list | --schedule | --count | --register]
                         [name] [plid] [plpos] [sid]

This program is written to manage the transfermarket

positional arguments:
  name          name of the player
  plid          player id
  plpos         player position
  sid           scouter id
```

'Help' is the built-in option for the command line program introducing the usage of the program. Added 'Version' to enable users to keep in track of latest updates of the program.

- Search

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --search Ronaldo
(u'Ronaldo', 32658, u'FW', 1)
```

Search the player by name and gives the information in the database.

- List

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --list scouter
(101, u'Solskjaer', u'408-0290', 1)
(102, u'Giggs', u'826-8596', 2)
(103, u'Scholes', u'221-3253', 2)
(104, u'Neville', u'419-5608', 3)
(105, u'Rooney', u'339-4191', 3)
(106, u'Saha', u'233-8520', 3)
```

Provides a list of scouter/staff/player. The user may choose which list to be loaded.

- Schedule

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --schedule 2JAN20
(2901, 17898, u'T', 3)
(6901, 18564, u'L', 6)
```

Takes the date as an input and shows the meeting on the given date, here the list does not provide the player name but only gives the player id as most football clubs keep the exact schedule in secret.

- Count

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --count
T: 15 L: 10
```

Shows the number of transfer and loan meeting going on.

- Register

```
Lenovo@DESKTOP-TPQNTU5 ~/fifa
$ python transfermarket.py --register Min 2 MF 2
player registered
```

Enables a new player to register for the meeting. (This function is frequently causing error so currently kept it as a block comment. The new data can still be loaded simply within the code by following the same format with in the 'session.add')

<Future Objectives>

I would like to make a nice UI for the user rather than keeping this as a command line program. Also adding up some relation with players' agents will improve the overall program for actual use. For each of the command option I would be able to put additional options such as sorting standards scouter/staff search.

<Conclusion>

As my project group went through change, I had to come up with a new project which I could individually work on. Fortunately, while concerning about the topic I could come up with the topic that is closely related to my interest. Before going through the final project, I couldn't easily understand the utility and efficiency of DBMS. But working on this project helped me to understand more about the real-world usage of the technique. Still, I think I need to work on more to figure out methods of finding more efficient database designs. Thank you.

<Reference>

[1] https://micropyramid.com/blog/understand-self-and-__init__-method-in-python-class/

[2] https://galid1.tistory.com/424

[3] https://www.pythoncentral.io/understanding-python-sqlalchemy-session/

[4] https://weicomes.tistory.com/261

[5] https://stackoverflow.com/questions/47735329/updating-a-row-using-sqlalchemy-orm

[6] https://kite.com/python/docs/sqlalchemy.orm.query.Query.update

[7] https://stackoverflow.com/questions/47374679/

[8] Adviced by Ehit Agarwal