FIFA 21

GROUP2

2019312486 김윤지 | 2021719185 Oscar Benoit

Outline

· Introduction

· DB design and modeling

· Demo of UI and functions

· Lessons Learned

Introduction

- Topic: FIFA 21 - Sports (football)

- Target Users: **Football fans** or people interested in football

- Users can use as a **search engine** to find players

  based on characteristics, nationality, name, contract information, etc.

Introduction

- Main function: Searching players matching specific features

    Update database with "Delete" button

- Language: Python

- Packages: Python pandas for importing and cleaning data
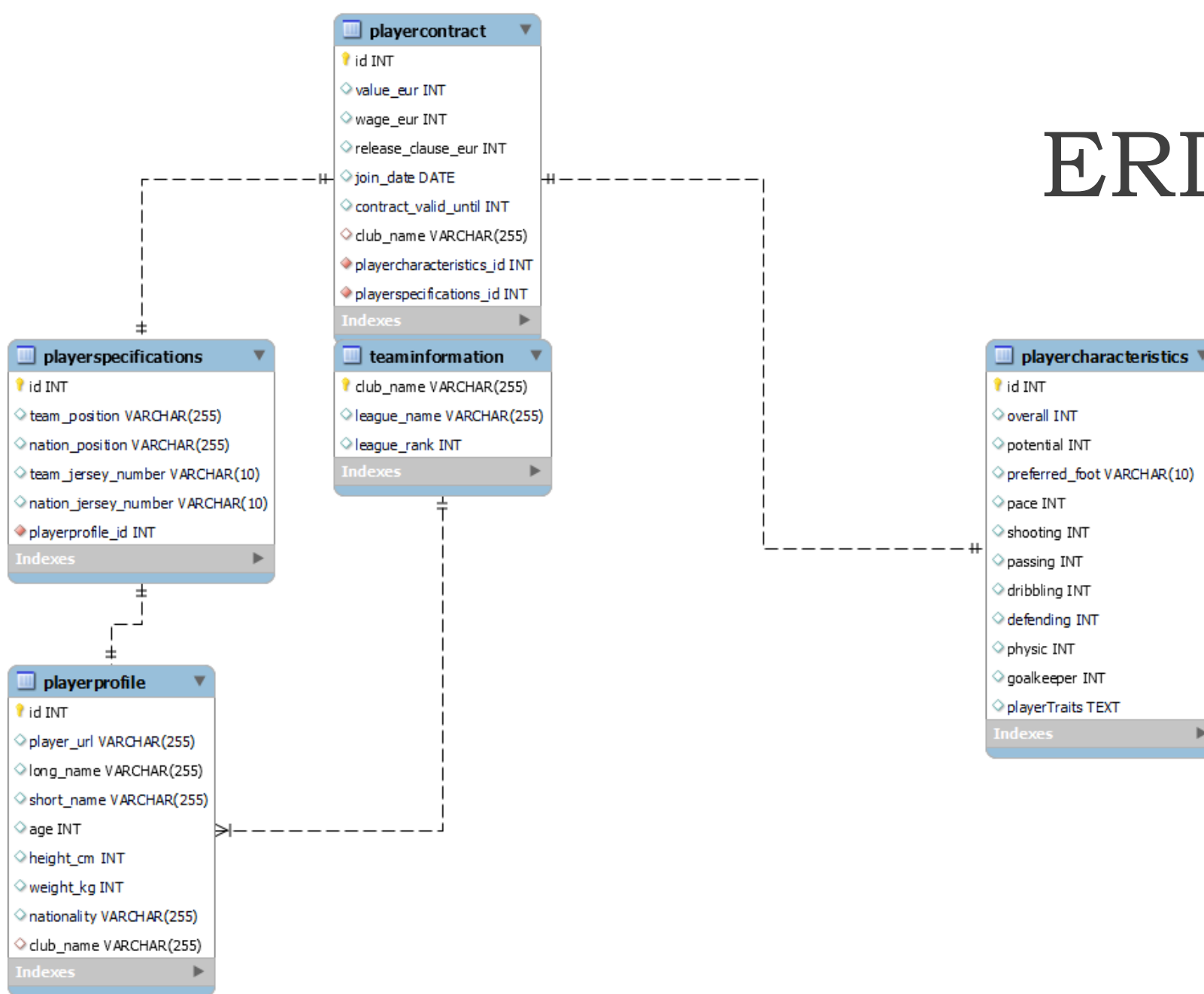
    Tkinter for GUI

DB Design and Modeling

- Fifa Database found [here](#) => 106 columns, 18.9k rows

- Many features, we had to clean => Pandas DataFrame

- NA values, cleaning was necessary

- Cleaning finished : 31 columns, 15959 rows => push the pd.DataFrame to MySQL Workbench

Code example to push the DB on Workbench

```python
mycursor.execute("USE footballplayersdb;")

mycursor.execute('CREATE TABLE IF NOT EXISTS teamInformation (club_name varchar(255) primary key, league_name varchar(255)
cnx.commit()


for row in teamTable.itertuples():
    mycursor.execute('''
                INSERT INTO teamInformation (club_name, league_name, league_rank)
                VALUES (%s,%s,%s)
                ''',
                (row.club_name,
                 row.league_name,
                 row.league_rank)
                )
cnx.commit()
```

ERD Diagram

playercontract
- id INT
- value_eur INT
- wage_eur INT
- release_clause_eur INT
- join_date DATE
- contract_valid_until INT
- club_name VARCHAR(255)
- playercharacteristics_id INT
- playerspecifications_id INT
- Indexes

playerspecifications
- id INT
- team_position VARCHAR(255)
- nation_position VARCHAR(255)
- team_jersey_number VARCHAR(10)
- nation_jersey_number VARCHAR(10)
- playerprofile_id INT
- Indexes

teaminformation
- club_name VARCHAR(255)
- league_name VARCHAR(255)
- league_rank INT
- Indexes

playercharacteristics
- id INT
- overall INT
- potential INT
- preferred_foot VARCHAR(10)
- pace INT
- shooting INT
- passing INT
- dribbling INT
- defending INT
- physic INT
- goalkeeper INT
- playerTraits TEXT
- Indexes

playerprofile
- id INT
- player_url VARCHAR(255)
- long_name VARCHAR(255)
- short_name VARCHAR(255)
- age INT
- height_cm INT
- weight_kg INT
- nationality VARCHAR(255)
- club_name VARCHAR(255)
- Indexes

Requesting on the database

- Creation of a function « search » linked to the search button

```python
26  def search():
27      filledCorrectly=True
28      clear()
29      whereStatement=True
30      querySQL = str("SELECT playerprofile.id, short_name, nationality FROM playerprofile
31  #PLAYER PROFILE PAGE
32
33      if name_textbox.get()!="":
34          querySQL = querySQL+" AND long_name LIKE '%"+name_textbox.get()+"%'"
35
36      if age_textbox.get() !="":
37          if age_textbox.get().isnumeric():
38              if whereStatement:
39                  querySQL = querySQL+" AND age="+age_textbox.get()
40          else:
41              filledCorrectly=False
42              messagebox.showerror("Error", "Wrong age")
```

Demo of UI and functions

5 tabs for entering information + 1 tab for showing Results + Search button on each tabs

Function I. Searching: Enter Info → Message box comes out → Can see the results on "Results" tab.

① Player Profile Tab: For basic information of players

● ex) Searching players who are 25 years old and their height is 170cm.

Demo of UI and functions

② Player Characteristics Tab: Skills of players

-Placed sliders to choose numbers within the range

● ex) Searching players with Potential over 65, Overall over 80
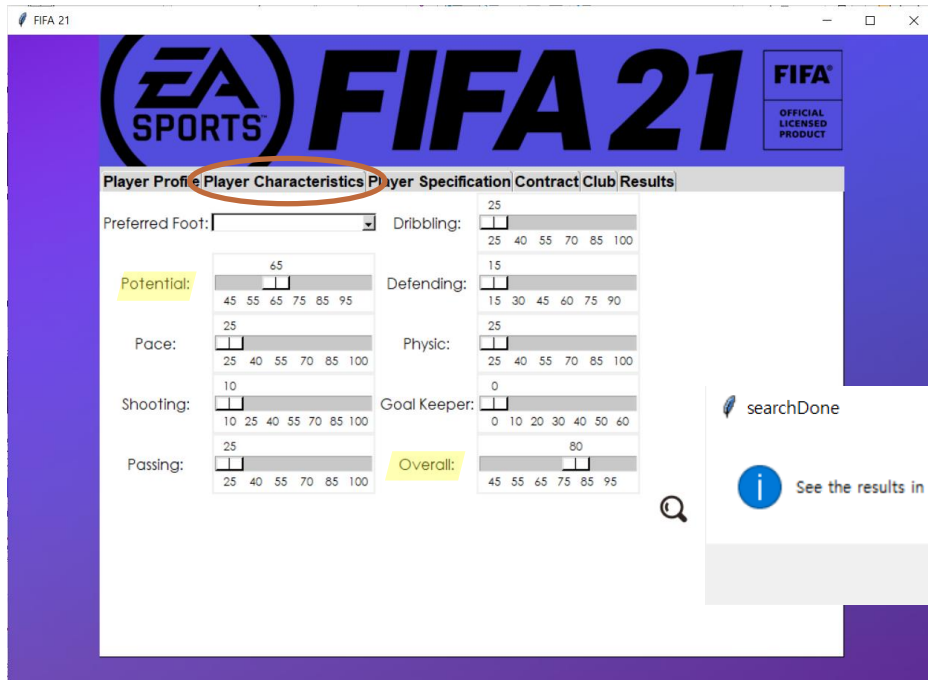
Demo of UI and functions

③ Player Specification Tab: Positions, Jersey number

● ex) Searching players who have same position both in team and nation as "ST"

Demo of UI and functions

④ Contract Tab: For contract information

  -Placed sliders as well & Calendar to choose date

● ex) Searching players who joined on July 9th,2020 and their contract valid year is 2024.

In order to get the SQL date format, which different from the Tkinter calendar, we created a function to convert the date : getSQLDate()

Demo of UI and functions

⑤ Club Tab: Club, League information

- ex) Searching players whose club name is "Paris Saint-German

Demo of UI and functions

Function II. Update:

Delete button → Message box → Can see player deleted from the result → Next player from the DB will be shown

- ex) want to delete last player

Demo of UI and functions

III. Error Handling

① Entering wrong data type : Gives warning message

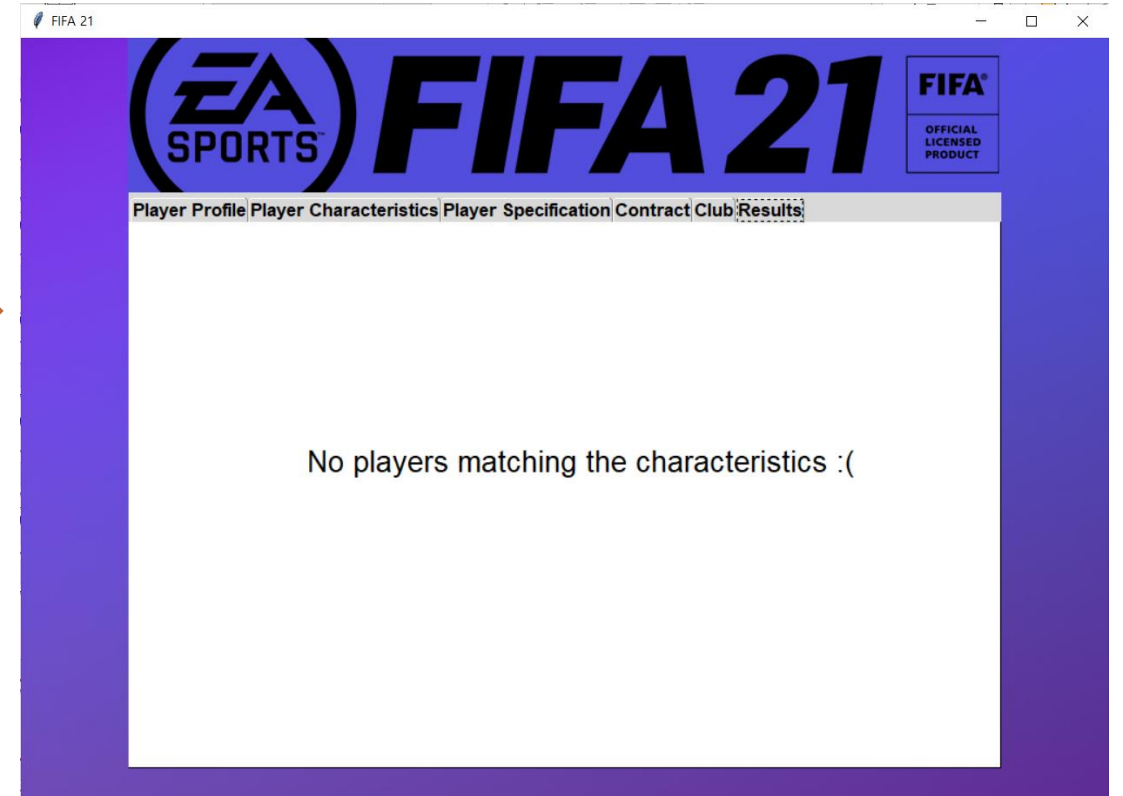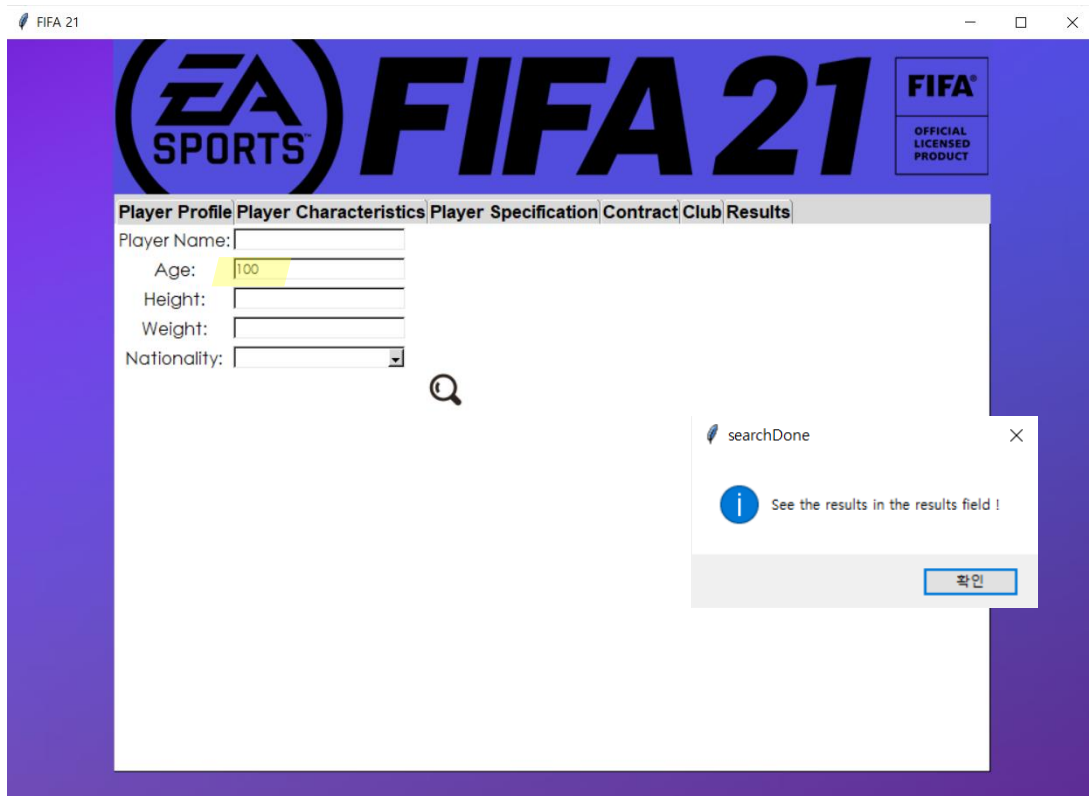● ex) entering character for weight on Player Profile tab.

Demo of UI and functions

III. Error Handling

② When there is no players matching

● ex) Searching players whose age is 100.

Lessons Learned

- Creating DB: The NA values and the number of columns to clean.

  Also, finding the method how to clean and push on MySQL Workbench

- GUI: hard to align the widgets appropriately,

  tried several times using grid, place, and pack to find suitable method

  ⇒ grid was convenient , have rows/columns so easier to think about their locations.

THANK YOU