Scraping & DataBase

“ Le foot c’est mieux quand il fait beau! ”

Elaboré par:

- Michael KRYSZTOFIAK
- Nina SMIRNOVA
- Hugo FUGERAY
- Nouha EL ABED

SOMMAIRE

Structure de la présentation:

- Objectif du projet
- Ressources et Outils utilisés
- Gestion projet et la répartition des tâches
- Avancement du projet
- Points notables : difficultés, astuces…
- Points d'amélioration

Objectif de project

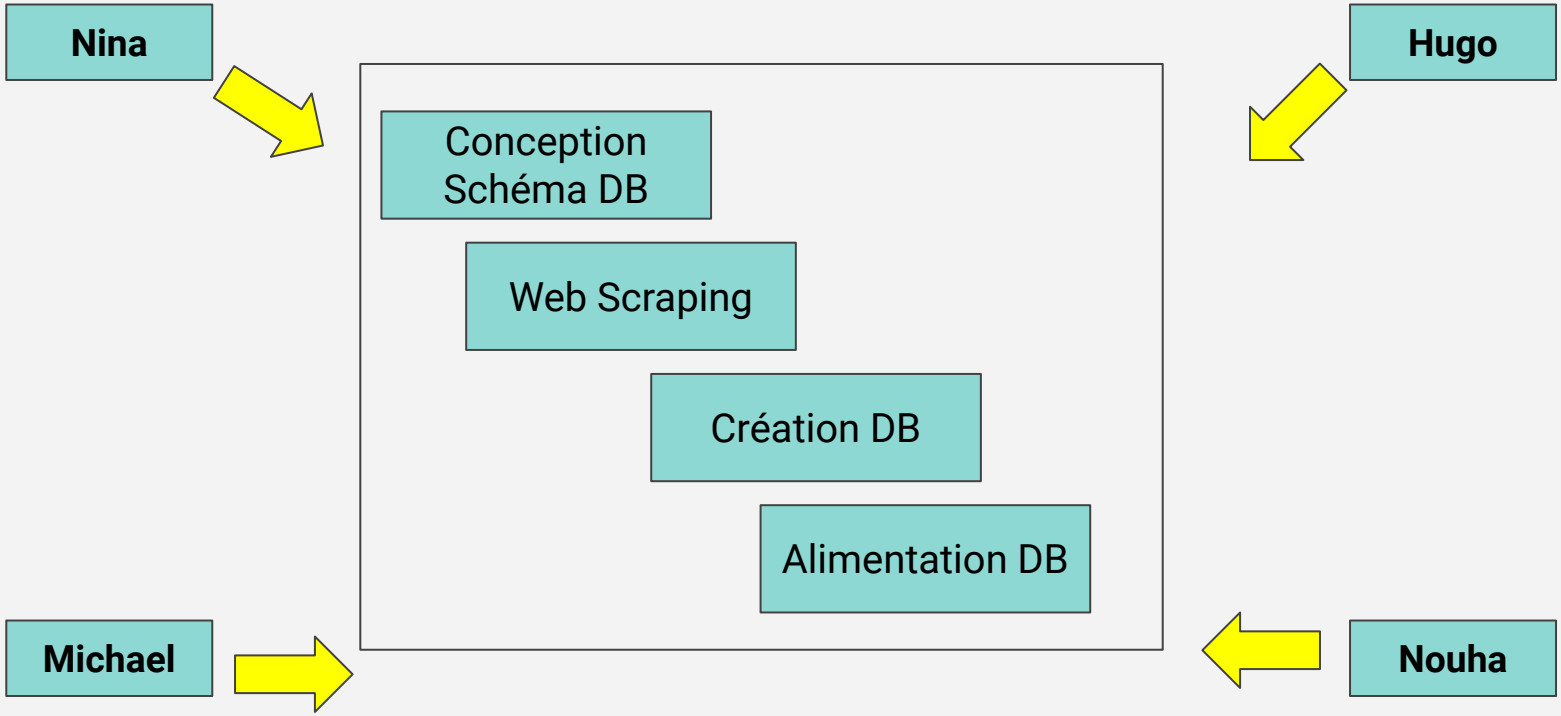
Conception et création d'une base de données relationnelle incluant un maximum de données sur:

- Un championnat de football
- La météo associée à chaque rencontre

Les ressources et outils utilisés

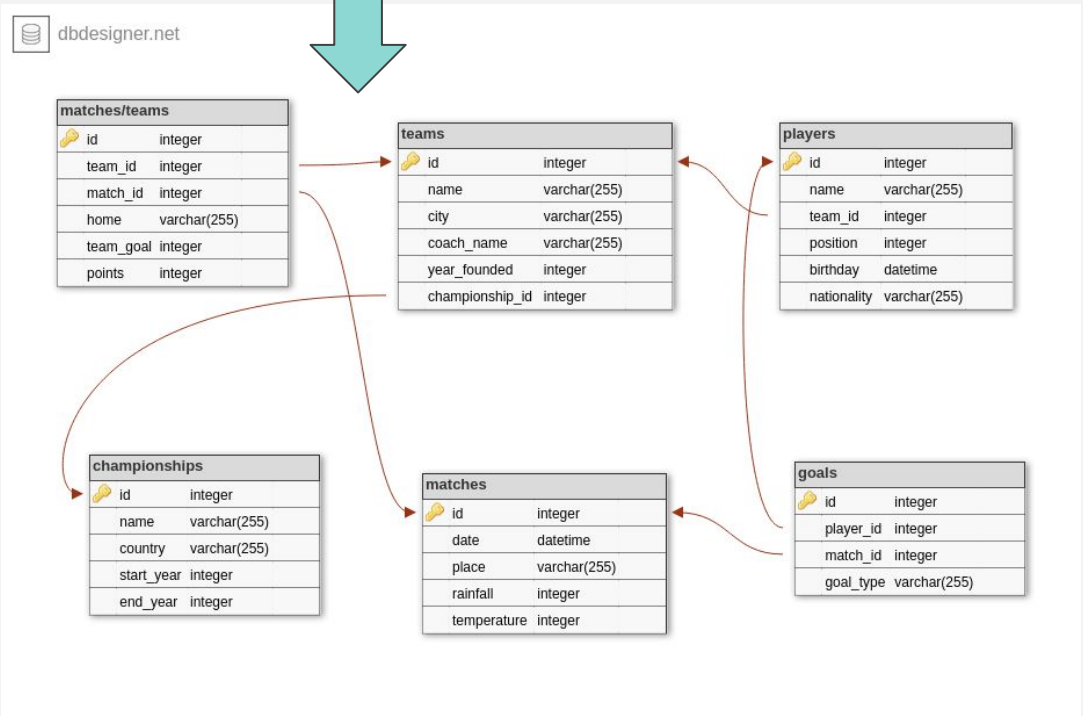
1. Les données ont été extraites de sites suivants:
    - Données de championnats: L'ÉQUIPE
    - Données de météo: Historique-Météo.net quel temps faisait-il ce jour-là ?
2. Architecture de la base de données :  DBdesigner
3. Gestion de projet: GitHub
4. Technologies : Jupyter , sqlite
5. requests, BeautifulSoup, sqlite3, pandas, numpy

Gestion et répartition des tâches



Avancement du projet

Conception de la base de données | Web Scraping | Création de la base de donnée | Alimentation la base de données



Difficultés :

- Choix des tables pivots
- Foreign Keys

Résolutions ::

- Partage en commun
- Live Code

Avancement du projet

Conception de la base de données

Web Scraping

Création de la base de donnée

Alimentation la base de données

```
url = "https://www.lequipe.fr/Football/ligue-1/saison-2020-2021/page-calendrier-resultats"

response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

json_list = [(i['value']) for i in soup.find(class_='SelectNav__select')]

dico = {
    'date': [],
    'lieu': [],
    'team_domicile': [],
    'team_exterieur': [],
    'score_domicile': [],
    'score_exterieur': [],
    'lien_detail': []
}

for journee in json_list:

    response = requests.get(journee)
    data = json.loads(response.text)

    for day in data['items'][:-1]:
        for game in day['items']:

            dico["lieu"].append(game['event']['lieu']['ville'])
            dico["date"].append(
                datetime.datetime.strptime(game['date'].split('+')[0],
                                           "%Y-%m-%dT%H:%M:%S"))
            dico["team_domicile"].append(
                game['event']['specifics']['domicile']['equipe']['nom'])
            dico["team_exterieur"].append(
                game['event']['specifics']['exterieur']['equipe']['nom'])
            dico["score_domicile"].append(
                game['event']['specifics']['score']['domicile'])
            dico["score_exterieur"].append(
                game['event']['specifics']['score']['exterieur'])
            dico["lien_detail"].append(game['event']['lien_web'])

df_matches = pd.DataFrame(dico)
```

Difficultés :

- Découverte de la structure du site
- Données réparties sur de nombreuses pages différentes
- Pas de connexion à API météo gratuite

Résolution:

- Boucle multipage pour le scrapping
- utilisation d'un fichier CSV contenant les données météo pour chaque match

Avancement du projet

Conception de la base de données | Web Scraping | Création de la base de donnée | Alimentation la base de données

Difficultés :

- Certain Type de données n'existe pas au format sqlite(booleen, timestamp)

Résolution :

- Encodage booleen en integer
- timestamp au format date
- pas de prise en compte des type de goals (CSC, etc.)

Create Tables

functions for tables creation ¶

In [11]:

```
#create tables
#return None

def create_tables(cursor):

    #championships
    cursor.execute("""
    CREATE TABLE championships(
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(255),
    country VARCHAR(255),
    start_year INTEGER,
    end_year INTEGER
    )
    ;""")

    conn.commit()

    #matches
    cursor.execute("""
    CREATE TABLE matches(
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    date DATE,
    place VARCHAR(255),
    rainfall REAL,
    temperature REAL
    )
    ;""")
    conn.commit()
```

Avancement du projet

Conception de la base de données | Web Scraping | Création de la base de donnée | Alimentation la base de données

Difficultés :

- Certains Df nécessitent d' être retravaillés
- Certains buteurs ont été transférés en cours d'année
- Protection du code vis à vis de l'injection SQL

Résolutions :

- Restructuration des df sous pandas
- Saisie manuelle des buteurs transférés
- Structuration des requêtes

fill players

Entrée [21]:

```
#fill players
for i in range(len(df_players)):

    cursor.execute("""
    INSERT INTO players(
    first_name,
    last_name,
    team_id,
    position,
    birthday,
    nationality)
    VALUES (?,?,?,?,?,?)
    """,(
        None,
        df_players.iloc[i]['Nom'],
        int(df_players.iloc[i]['id_teams']),
        df_players.iloc[i]['Pos.'],
        df_players.iloc[i][2],
        df_players.iloc[i]['Pays']
    ))

    conn.commit()
```

executed in 2.53s, finished 17:37:12 2021-06-26

Améliorations

- Déclinaison de la structure sur plusieurs Championnats (from Russia import Nina)
- Mise en production de la DB (from Merignac import Hugo)
- Renforcement de la structure de la DB avec une table meteo reliée à une API(from Tunisia import Nouha)
- Scrapper directement les joueurs sur la fiche du joueur, mieux préparer le scrapping en fonction de la structure des tables à alimenter (from Poland import Michael)