فاز دوم پروژه: ساخت پایگاه داده بسکتبال

1. مقدمه

هدف از این پروژه، ساخت یک پایگاه داده‌ی رابطه‌ای برای مدیریت اطلاعات بازیکنان، تیم‌ها، فصل‌ها و جوایز مرتبط با بسکتبال بود. داده‌های اولیه به‌صورت فایل‌های CSV و Excel در اختیار قرار داشتند که ساختار آن‌ها خام و غیرنرمال بود (مثلاً پوزیشن‌ها به‌صورت لیست متنی در یک ستون ذخیره شده بودند یا نام بازیکنان در فایل‌های مختلف با فرمت‌های متفاوت نوشته شده بود).

برای رسیدن به یک دیتابیس ساخت‌یافته، مراحل استخراج، پردازش و بارگذاری (ETL) روی داده‌ها انجام شد.

2. ابزارها و کتابخانه‌های استفاده‌شده

1. اتصال به پایگاه داده(MySQL Connector, SQLAlchemy)

- با استفاده از mysql.connector یک دیتابیس به نام Basketball_Reference ساخته شد.
- سپس با SQLAlchemy جداول به‌صورت کلاس‌های ORM تعریف شدند تا ایجاد و مدیریت دیتابیس راحت‌تر و استانداردتر انجام شود.

2. پردازش داده‌ها(Pandas)

- برای خواندن فایل‌های CSV و Excel و انجام عملیات پاک‌سازی (Drop ، Merge، Normalization)استفاده شد.
- Pandasبه ما امکان داد که داده‌های خام را به DataFrame تبدیل کنیم و سپس خروجی نهایی را به Excel و دیتابیس منتقل کنیم.

3. تبدیل رشته به لیست(ast.literal_eval)

- پوزیشن‌ها در فایل CSV به شکل متن لیستی مثل ['Center', 'Forward'] ذخیره شده بودند.
- برای اینکه بتوانیم روی این داده‌ها حلقه بزنیم، با این ماژول به لیست واقعی Python تبدیل شدند.

4. یکنواخت‌سازی نام‌ها(unicodedata, rapidfuzz)

- چون اسامی بازیکنان در فایل‌های مختلف با نگارش‌های متفاوت وجود داشت (مثلاً فاصله، حروف بزرگ/کوچک یا حروف خاص)، نیاز بود نام‌ها به یک شکل استاندارد ذخیره شوند.
- ابتدا با unicodedata نام‌ها نرمال‌سازی شدند (حذف کاراکترهای خاص و تبدیل به lowercase).
- سپس با rapidfuzz و الگوریتم fuzzy matching مطابقت نام‌ها بررسی شد تا هر بازیکن درست به رکورد خودش در دیتابیس وصل بشود.

5. ساخت جداول نهایی(Excel + SQLAlchemy ORM)

- داده‌های نهایی قبل از بارگذاری در MySQL به‌صورت فایل‌های Excel جداگانه ذخیره شدند.
- این کار کمک کرد که داده‌ها مرحله به مرحله بررسی شوند و امکان Debug راحت‌تر وجود داشته باشد.

6. برای کار با تاریخ‌ها(datetime)

7. برای امن‌سازی رمز عبور در اتصال به دیتابیس(urllib.parse.quote_plus)

3. مراحل پردازش داده‌ها

الف) ایجاد دیتابیس

- با استفاده از mysql.connector یک دیتابیس جدید به نام Basketball_Reference ساخته شد.

ب) پردازش داده‌های بازیکنان

- فایل players_info.csv خوانده شد.
- ستون‌های اضافی حذف شدند.(Positions, Current_team)
- یک PlayerID یکتا برای هر بازیکن ایجاد شد.

ج) ساخت جداول میانی

- Positions:استخراج همه پوزیشن‌های یکتا بازیکنان و ساخت جدول آن‌ها.

- Player_Positions: ایجاد رابطه بین بازیکنان و پوزیشن‌ها (بازیکن ممکن است چند پوزیشن داشته باشد).
- Teams : استخراج تیم‌های یکتا از بازیکنان و ایجاد جدول تیم‌ها.
- Player_Teams: رابطه بین بازیکنان و تیم‌ها ساخته شد.

تمام این دیتاها در فایل‌های Excel ذخیره شدند:

- Players.xlsx
- Positions.xlsx
- Player_Positions.xlsx
- Teams.xlsx
- Player_Teams.xlsx

د) پردازش اطلاعات تیم‌ها

- داده‌های تیم‌ها از دو منبع ادغام شدند:
  - Teams.xlsx
  - team_infos.csv
- جدول نهایی در Teams_Final.xlsx ذخیره شد.

ه) پردازش قهرمان‌ها

- فایل champions.csv خوانده شد.
- جدول Seasons ساخته شد (فصل‌های یکتا).
- جدول Champions_ID ساخته شد(شامل ChampionID و TeamID)
- رابطه بین فصل‌ها، قهرمان‌ها و بازیکنان قهرمان ساخته شد و در Champions_Players.xlsxذخیره شد.

و) پردازش رنکینگ بازیکنان

- فایل players_ranked.csv خوانده شد.
- با نرمال‌سازی اسم‌ها وfuzzy matching ، بازیکنان به PlayerID نگاشت شدند.

- خروجی در فایل Player_Ranks.xlsx ذخیره شد.

ز) پردازش جایزه Michael Jordan Trophy

- فایل Michael_Jordan_Trophy.csv خوانده شد.
- مشابه رنکینگ، بازیکنان match شدند.
- خروجی در فایل Michael_Jordan_Trophy.xlsx ذخیره شد.

4. تعریف جداول دیتابیس با SQLAlchemy

با استفاده از ORM در SQLAlchemy جداول زیر ساخته شدند:

1. players: شامل اطلاعات اصلی بازیکنان (نام، تاریخ تولد، قد، وزن، ملیت، امتیاز و ...).
2. positions: پوزیشن‌های مختلف بازی.
3. player_positions : جدول رابطه بین بازیکن و پوزیشن.
4. teams: اطلاعات تیم‌ها (نام، موقعیت، تاریخ تأسیس، تعداد قهرمانی و حضور در پلی‌آف).
5. player_teams: جدول رابطه بین بازیکن و تیم.
6. seasons: فصل‌های بازی.
7. champions: تیم‌های قهرمان هر فصل.
8. champion_players: بازیکنانی که در تیم قهرمان حضور داشتند.
9. player_ranks: رنکینگ بازیکنان در هر فصل.
10. michael_jordan_trophy: جدول مخصوص جایزه Michael Jordan Trophy .

5. بارگذاری داده‌ها در دیتابیس

- داده‌های نهایی از فایل‌های Excel خوانده شدند.
- با متد to_sql() وارد MySQL شدند.
- همه جداول نهایی با داده پر شدند.

6. چرایی جداسازی داده‌ها(Normalization)

یکی از مهم‌ترین اصول طراحی دیتابیس رابطه‌ای نرمال‌سازی (Normalization) است.
یعنی:

- هر موجودیت (Entity) در یک جدول مستقل ذخیره بشود.
- داده‌های تکراری و وابستگی‌های غیرضروری حذف شوند.
- بین جداول ارتباط منطقی (Relation) ایجاد بشود.

به همین دلیل داده‌ها از حالت خام به چند فایل/جدول مستقل تبدیل شدند:

1. Players.xlsx : شامل اطلاعات پایه بازیکنان مثل نام، تاریخ تولد، ملیت، قد و وزن.
2. Positions.xlsx : شامل لیست یکتای پوزیشن‌ها.(Guard, Forward, Center, ...)
3. Player_Positions.xlsx : رابطه بین بازیکن و پوزیشن(ارتباط Many-to-Many).
4. Teams.xlsx : لیست تیم‌ها.
5. Player_Teams.xlsx : رابطه بازیکنان با تیم‌ها(ارتباطMany-to-Many).
6. Seasons.xlsx : فصل‌های مختلف مسابقات.
7. Champions_ID.xlsx : لیست تیم‌های قهرمان همراه با شناسه یکتا.
8. Champions_Players.xlsx : بازیکنانی که در تیم‌های قهرمان حضور داشتند.
9. Player_Ranks.xlsx : رتبه‌بندی بازیکنان در هر فصل.
10. Michael_Jordan_Trophy.xlsx : اطلاعات بازیکنانی که این جایزه را دریافت کردند.

مزایای جداسازی

- حذف تکرار داده‌ها: اطلاعات هر بازیکن فقط یک بار در جدول Players ذخیره می‌شود.
- انعطاف‌پذیری: اضافه یا حذف کردن اطلاعات راحت‌تر است.
- یکپارچگی داده‌ها: تغییر در یک جدول (مثلاً نام تیم) به‌طور خودکار در همه روابط اثر می‌گذارد.
- تحلیل‌پذیری بهتر: می‌توان بین جداول Join زد و گزارش‌های متنوع گرفت.

7. منطق کلی پروژه(ETL)

این پروژه در واقع یک ETL (Extract, Transform, Load) است:

1. Extract : داده‌ها از CSV وExcel های خام خوانده شدند.
2. Transform : داده‌ها تمیز شدند، نرمال‌سازی شدند، جداول میانی ساخته شدند.
3. Load : داده‌های نهایی وارد دیتابیس MySQL شدند.

8. جمع‌بندی

این پروژه نشان می‌دهد که چگونه می‌توان داده‌های خام و نامرتب را با استفاده از Python و MySQL به یک دیتابیس استاندارد رابطه‌ای تبدیل کرد.

نکات کلیدی:

- استفاده از Pandas و RapidFuzz برای پردازش و یکپارچه‌سازی داده‌ها.
- طراحی اصولی جداول بر اساس Normalization .
- پیاده‌سازی روابط چندبه‌چند با جداول میانی.
- جداسازی فایل‌های Excel برای خوانایی، نگهداری و پردازش راحت‌تر.
- بارگذاری نهایی داده‌ها در MySQL برای گزارش‌گیری و تحلیل‌های بعدی.

9. فرآیند اتصال، بازیابی و تحلیل داده‌ها در پایتون

در این بخش از پروژه، از زبان پایتون و کتابخانه‌های mysql.connector و pandas برای اتصال به پایگاه داده و استخراج داده‌ها استفاده شده است. منطق کد به صورت زیر است:

مدیریت خطا :(Error Handling)
کل عملیات در یک بلوک try/except قرار گرفته است. در صورت بروز خطا (مانند عدم دسترسی به پایگاه داده یا خطای کوئری)، پیام خطا در خروجی چاپ می‌شود.

ایجاد اتصال به پایگاه داده:
با استفاده از دستور

with mysql.connector.connect(**config) as cnxn :

اتصال به پایگاه داده برقرار می‌شود. پارامتر config شامل اطلاعات کاربری و مشخصات اتصال (مانند نام سرور، نام کاربری، رمز عبور و نام پایگاه داده) است. استفاده از ساختار with باعث می‌شود که اتصال پس از اتمام عملیات یا در صورت بروز خطا به صورت خودکار بسته شود

انتقال داده‌ها به پانداس:

با استفاده از دستور

df_top50_height = pd.read_sql(query, cnxn)

خروجی کوئری مستقیماً در قالب یک DataFrame پانداس ذخیره می‌شود. این کار امکان تحلیل و پردازش راحت‌تر داده‌ها در محیط پایتون را فراهم می‌کند .

10. نمایش نتایج:

در نهایت، داده‌های استخراج‌شده با دستور display(df_top50_height) به صورت جدول در محیط اجرا نمایش داده می‌شوند . اطلاعات بدست آمده به فاز سه میرن برای تحلیل آماری .