European Basketball Statistics

BLG 317E - Database Systems - Semester Project

Made by KickStats

WHO ARE WE AND WHAT WE ARE RESPONSIBLE FOR?

Mohamed Ahmed Abdelsattar Mahmoud

Student ID:
150210926

Email:
mahmoud21@itu.edu.tr

Muhammed Yusuf Mermer

Student ID:
150220762

Email:
mermer22@itu.edu.tr

MHD Kamal Rushdi

Student ID:
150210907

Email:
rushdi21@itu.edu.tr

Muhammed Can Özkurt

Student ID:
820220710

Email:
ozkurtm22@itu.edu.tr

OVERVIEW

The project focuses on creating an interactive and feature-rich platform that utilizes a relational database to handle and visualize a complex dataset.

The interaction in our case is the one between the Admin and the user, as will be demonstrated later.

FEATURES OF OUR WEB APP

01

Relational Database Design

A fully normalized database with optimized tables and queries.

02

CRUD Operations

Comprehensive Create, Read, Update, and Delete functionalities.

03

RESTful API

Serves as the communication layer between the backend and frontend for smooth data transfer.

04

Data Validation

Ensures the integrity and consistency of the data before it's stored in the database.

FEATURES OF OUR WEB APP

05

Dynamic Web Interface

Built using modern frameworks, offering a user-friendly and visually appealing UI.

06

Data Visualization

Presents complex data through charts, tables, and other graphical representations.

07

Interactive Elements

Includes clickable buttons, navigational menus, and modals for enhanced interactivity.

Tools used to achieve compatability

Frontend	React js – Next js – Recharts – Css modules
Backend	Flask
Database	MySQL

ABOUT OUR CHOSEN DATASET

Our dataset focuses on basketball games and team performance in tournaments like EuroLeague and EuroCup (2007–2023). It includes detailed game statistics, offensive and defensive metrics, and player contributions, organized seasonally for trend analysis.

Why This Dataset?
Rich, diverse, and perfectly aligned with our project's goal to compare and visualize team performance dynamically.

HOW WE DID UTILIZE IT

Handling empty Integers and corrupted values

Dorsal to -1

Winner column

HOW WE DID UTILIZE IT

Handling empty Integers and corrupted values

```
# Replace empty cells in the specified columns with 0
for column in columns_to_process:
    df[column] = df[column].fillna(0)

# Compare and update values for specific conditions
# Compare score_extra_time_1_a with score_quarter_4_a
df['score_extra_time_1_a'] = df.apply(
    lambda row: 0 if row['score_extra_time_1_a'] == row['score_quarter_4_a'] else row['score_extra_time_1_a'], axis=1
)

# Compare score_extra_time_1_b with score_quarter_4_b
df['score_extra_time_1_b'] = df.apply(
    lambda row: 0 if row['score_extra_time_1_b'] == row['score_quarter_4_b'] else row['score_extra_time_1_b'], axis=1
)
```

HOW WE DID UTILIZE IT

Dorsal to -1

```
# Load the dataset
file_path = folder_path + file_name
df = pd.read_csv(file_path)

# Replace "TOTAL" in the "dorsal" column with -1
df['dorsal'] = df['dorsal'].replace("TOTAL", -1)

# Save the modified DataFrame to a new file with "_filtered" suffix
filtered_file_path = folder_path + file_name.replace('.csv', '_filtered.csv')
df.to_csv(filtered_file_path, index=False)
```

HOW WE DID UTILIZE IT

Winner column

```python
# Load the dataset
file_path = folder_path + file_name
df = pd.read_csv(file_path)

# Create the 'winner' column based on the conditions
df['winner'] = df.apply(
    lambda row: 'team_a' if row['score_a'] > row['score_b'] else ('team_b' if row['score_a'] < row['score_b'] else 'draw'),
    axis=1
)

# Save the modified DataFrame to a new file with "_with_winner" suffix
output_path = folder_path + file_name.replace('.csv', '_with_winner.csv')
df.to_csv(output_path, index=False)
```

HOW WE DID UTILIZE IT

Merging Columns

Adding Zeros

Team Data Table Creation

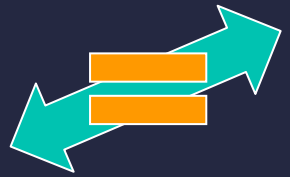
Merging Columns

CUP_PLAY_BY_PLAY:

game_play_id	player_id	game_id
U2007_001_003	U2007_P000168	U2007_001
U2007_001_004	U2007_P000168	U2007_001
U2007_001_005	U2007_P000643	U2007_001

CUP_PLAY_BY_PLAY:

game_play_id	game_player_id
U2007_001_003	U2007_001_P000168
U2007_001_004	U2007_001_P000168
U2007_001_005	U2007_001_P000643



CUP_BOX_SCORE:

game_player_id
U2007_001_P000168
U2007_001_P000168
U2007_001_P000643

Adding Zeros

CUP_POINTS:
CUP_PLAY_BY_PLAY:

game_player_id	game_play_id
U2007_001_P000168	U2007_1_4
U2007_001_P000317	U2007_1_262
U2007_001_PBAT	U2007_1_80

game_player_id	game_play_id
U2007_001_P000168	U2007_001_004
U2007_001_P000317	U2007_001_262
U2007_001_PBAT	U2007_001_080

CUP_PLAY_BY_PLAY:

game_play_id
U2007_001_004
U2007_001_262
U2007_001_080

Team Data Table Creation

Search Team Data

Collect all team names and abbreviations, then create a CSV file.

Match Logos

Check for matching folder names and add a logo URL column where applicable.

Create & Load Table

Create the MySQL table and load the CSV data into it.

SQL Data Setup for Before Website

01
CREATE
Define tables with columns, no relationships yet.

02
LOAD
Import data from .CSV files.

03
ELIMINATE
Remove problematic rows.

04
RELATE
Add relationships between tables.

WEBSITE PURPOSE

ADMIN

Provides direct UI for the ADMIN to edit the data inside the website

USER

Allows the user to visualize the data without the need to read complex tables

Functionalities

Admin View

CRUD

Uses SQL clauses and operators (e.g WHERE, ORDER BY)

Filtering, sorting, and column selecting

Pagination

User view

No CRUD

Uses complex queries (e.g JOIN)

View statistics, match results, and players performance in each season

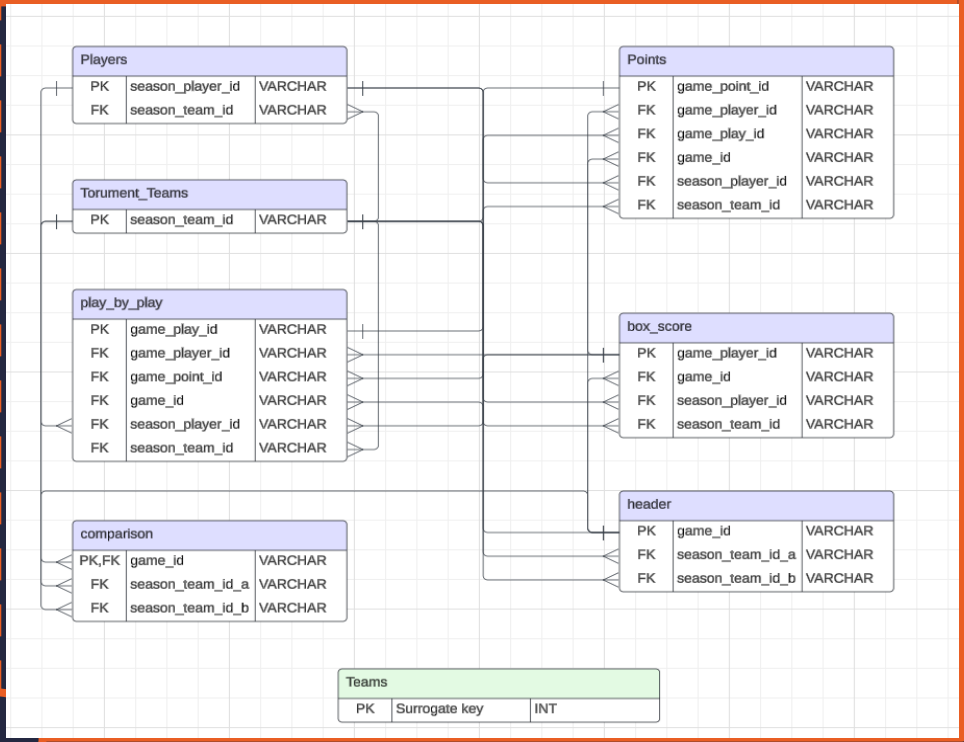
ER Diagram

Master Key

All of our keys started by season (e.g E2007)

This concept was used throughout all of our user view features

It allows the user to select the season data they want to visualize



DATABASE DESIGN

2 X Relations

We used both tables of Euroleague and Eurocup for completeness

Logo Table

Added Logos table for the user view design

Data cleaning

Cleaned the data using scripts (e.g removing duplicate PKs in some tables)

Table dependency

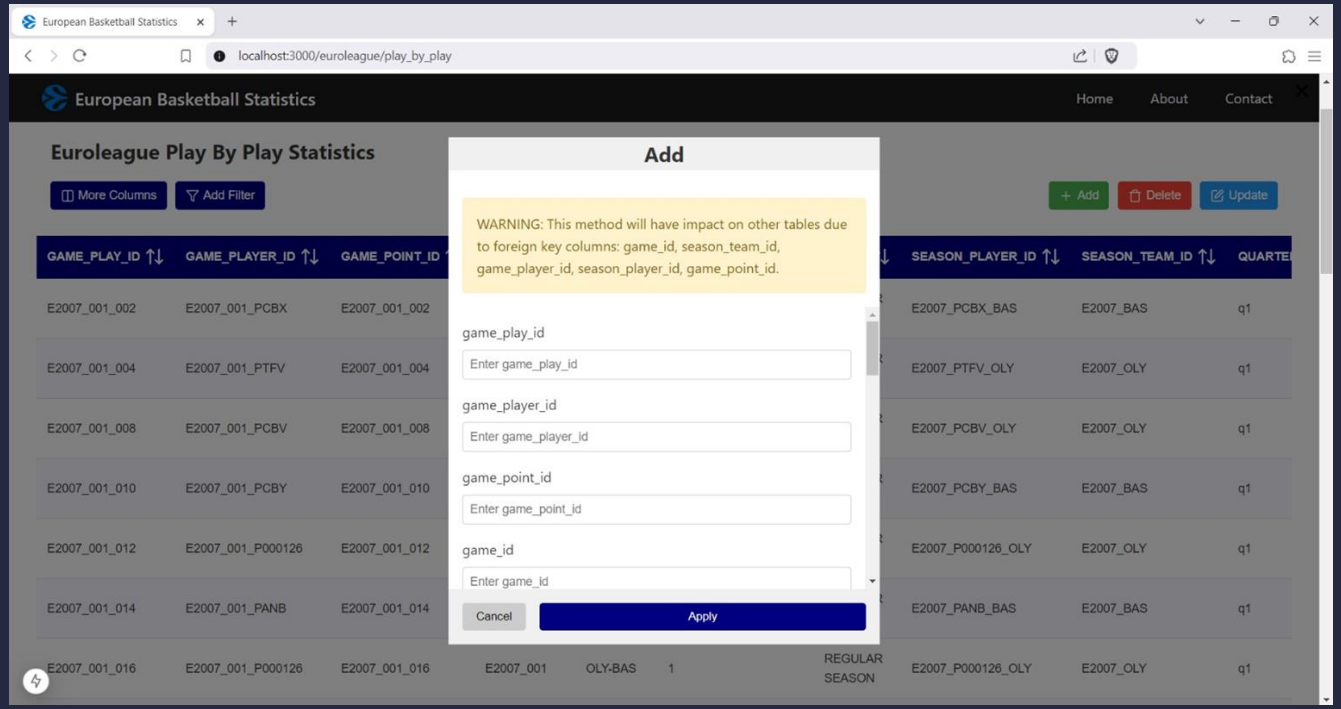
Made header table as main and comparison as secondary due to them using the same PK

CHALLENGES & CONCLUSION

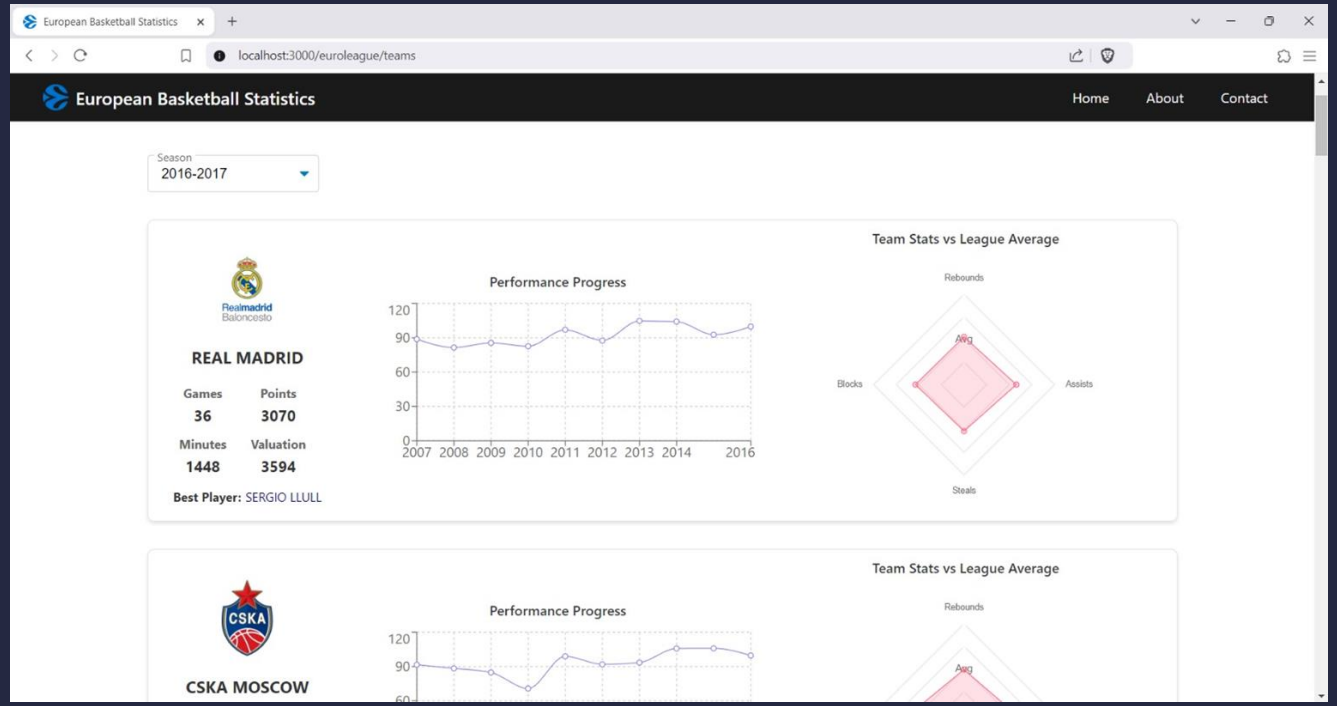
FINDING A DATASET

SQL

ADMIN VIEW PAGE



USER VIEW PAGE




JS
JS
C/S
Pythwipt
CSS

THANK YOU