Faculty of Media Engineering and Technology

German University in Cairo

Dr. Nourhan Ehab

Milestone 2: FPL Companion *(10%)*

Deadline: 2nd of December at 23:59

After completing **Milestone 1**, you now have a solid understanding of the FPL dataset. Its structure, the different categories of features, how player statistics evolve across gameweeks, and the connections between players, teams, and match events. You also explored common pitfalls like missing and incorrect data, which can significantly distort predictions if historical data is not handled carefully.

Building on this foundation, **Milestone 2** shifts toward constructing a **Knowledge Graph (KG)** using Neo4j that semantically represents the interconnected world of Fantasy Premier League. The core goal of this milestone is to transform raw tabular FPL data into a rich, graph-based model that captures the true relationships between **players, teams, fixtures, gameweeks, positions, and performance metrics**. By representing the FPL universe as an interconnected graph, we lay the groundwork for a more intelligent, explainable, and context-aware system that can support applications such as recommendation engines, matchup analysis, squad planning tools, and predictive assistants.

A major enhancement in Milestone 2 is the addition of **scoring-rule semantics**, which are central to the FPL ecosystem. These rules define how different positions earn points from various actions, such as goals, assists, clean sheets, and saves, adding a crucial layer of reasoning to the graph. By integrating these rules into the KG, the system becomes capable of understanding not just what happened in a match, but **why players earned the points they did**, enabling more accurate insights, smarter recommendations, and more interpretable analyses for FPL managers.

Faculty of Media Engineering and Technology

German University in Cairo

Dr. Nourhan Ehab

The Dataset:

In this milestone, you are provided with a **smaller, more focused subset of the FPL dataset** that you explored in Milestone 1. Specifically, we concentrate on the **last two seasons:** 2021-22 and 2022-23, which allows you to construct a cleaner and more manageable Knowledge Graph while still capturing modern FPL dynamics. To support a more precise representation of match structure, two new columns have been introduced to explicitly distinguish between the **home team** and the **away team**, removing any ambiguity in match interpretation.

Unlike Milestone 1, **no preprocessing or exploratory data analysis is required at this stage**. Instead, the emphasis shifts entirely toward modeling—your task is to use this curated dataset to **unify the schema of the Knowledge Graph**, ensuring consistent node labels, relationship types, and property definitions across the entire FPL ecosystem. This streamlined dataset gives you everything you need to focus on semantic modeling rather than data cleaning, allowing you to design a coherent, well-structured KG that reflects the true complexity and interconnectedness of Fantasy Premier League.

Milestone 2 Requirements:

This milestone requires you to construct an **FPL Knowledge Graph** using Neo4j that accurately represents the domain according to the specified schema. The knowledge graph will capture the complex relationships between **players, teams, fixtures, gameweeks, positions, and performance metrics**, creating a rich semantic structure that supports advanced querying and FPL-driven insights.

Your primary tasks include:

1. **Knowledge Graph Construction**

   Implement a script (using Python and the Neo4j driver) that reads the provided CSV files and constructs the knowledge graph according to the exact schema specification provided below. This involves creating nodes for Seasons, Gameweeks, Fixtures, Teams, Players, Positions, and establishing the relationships between them as defined in the schema.

2. **Schema Compliance**

   It is critical that you strictly adhere to the provided schema. You are **not permitted to modify the schema in any way**, including:
   – Changing the type of any node (e.g., you cannot rename “Player” to “Footballer” or “Team” to “Club”)
   – Changing the type of any relationship (e.g., you cannot rename “:PLAYED_IN” to “:APPEARED_IN” or “:HAS_GW” to “:CONTAINS”)
   – Removing, or modifying any properties specified for nodes
   – Removing any node types or relationship types from the schema

3. **Implementing The Given Scoring Rule**

   You need to utilize the scoring rules provided below in order to be able **to detect players which have been incorrectly assigned two positions**. These players will have 2 positions linked to them, one of which is a defender. Your task is to find said players, and provide details of only the fixtures in which they were **mistaken** as a defender. Keep in mind that there are general scoring for any position, and specific scoring per position.

4. **Cypher Query Development**

   After constructing the knowledge graph, you will develop Cypher queries to answer a comprehensive set of analytical questions. These queries will test your understanding of graph traversal, aggregation, filtering, and pattern matching in Cypher. The queries range from simple filtering operations to complex multi-hop traversals that require careful consideration of relationship directions and node properties.

Faculty of Media Engineering and Technology

German University in Cairo

Dr. Nourhan Ehab

The Knowledge Graph Schema:

1. Nodes:

1. Season: [season_name]
2. Gameweek: [season, GW_number]
3. Fixture: [season, fixture_number], kickoff_time
4. Team: [name]
5. Player: [player_name, player_element]
6. Position: [name]

```
*** [ ] represent unique identifiers. (If more than one attribute
is inside the bracket, then the node is uniquely identified by
all attributes inside the square brackets.
```

2. Relationships:

- (Season) - [:HAS_GW]-> (Gameweek)
- (Gameweek) - [:HAS_FIXTURE]-> (Fixture)
- (Fixture) - [:HAS_HOME_TEAM]-> (Team)
- (Fixture) - [:HAS_AWAY_TEAM]-> (Team)
- (Player) - [:PLAYS_AS]-> (Position)
- (Player) - [:PLAYED_IN]-> (Fixture);
    - Properties*: minutes, goals_scored, assists, total_points, bonus, clean_sheets, goals_conceded, own_goals, penalties_saved, penalties_missed, yellow_cards, red_cards, saves, bps, influence, creativity, threat, ict_index, form*

Faculty of Media Engineering and Technology

German University in Cairo

Dr. Nourhan Ehab

Knowledge Graph Verification Queries

NOTE ON VERIFICATION: These are example questions & answers for initial structure confirmation. Create queries that answer the following questions. If your queries return the same answers as below, then the structure of your knowledge graph is sound. The final assessment of the graph's design and robustness will involve a separate set of private queries to ensure the model is flexible and not hardcoded to these specific examples.

1. How many gameweeks were played during the seasons provided?

   Answer: **75**

2. How many player nodes were created (note: the same player can have different nodes representing him in each season)

   Answer: **1513**

3. What teams were qualified for the 2021-22 season but not for the 2022-23 season?

   Answer: **`["Burnley", "Watford", "Norwich"]`**

4. What was the biggest total_point value scored in one gameweek in the 2022-23 season and by who was it scored?

   Answer: **23 points by Erling Haaland**

5. How many matches did player Mohamed Elneny actually participate in?

   Answer: **19**

   **Hint: Sometimes players had not only different elements but also different names for each season 🙂

Scoring Rules:

Fantasy points in FPL are derived from a combination of universal actions and position-specific rules. While some events contribute points regardless of a player's role, others vary depending on whether the player is a goalkeeper, defender, midfielder, or forward. The following tables summarize these scoring rules.

General:

Action	Points Awarded
Goal Assist	3
Playing up to 60 minutes	1
Playing 60 minutes or more	2
Penalty Miss	-2
Own Goal	-2
Yellow Card	-1
Red Card	-3

Position Specific:

Action	Goalkeeper (GK)	Defender (DEF)	Midfielder (MID)	Forward (FWD)
Goal Scored	10	6	5	4
Clean Sheet***	4	4	1	0

Goals Conceded	**-1** (for every 2 goals conceded)	**-1** (for every 2 goals conceded)	**0**	**0**
Penalty Save	**5**	**0**	**0**	**0**
Saves	**1** (for every 3 saves)	**0**	**0**	**0**

*** Clean sheet points are only included if the player has played more than 60 minutes.

During the data entry of some fixtures, certain players were recorded with inconsistent positional labels. To validate these cases, we aim to display the affected players and verify whether the assigned **defender** position is indeed incorrect. This evaluation will rely on the **position-specific scoring rules** outlined in the scoring tables, as they provide a reliable benchmark for detecting positional inconsistencies. We also want to display the player's name, total number of incorrect fixtures per player, total number of fixtures played, and the incorrect fixtures' details. The details include:

- **`def_score`** -> which is what the score should have been if the player's position was indeed a defender.
- **`season`** -> the season that contains the incorrect fixture.
- **`fixture_number`** -> the fixture with the incorrect position.
- **`matches_def`** -> a variable which contains whether the fixture position matches the defender position or not
- **`total_points`** -> the actual score of the player during this fixture.

Faculty of Media Engineering and Technology

German University in Cairo

Dr. Nourhan Ehab

This is an example of a player who has an incorrect position with the expected output:

player	not_def_fixtures	total_fixtures	fixture_details
“Boubacar Traoré”	**2**	**32**	`[ { "def_score": 0.0, "season": "2022-23", "fixture_number": 160, "matches_def": false, "total_points": 1 }, { "def_score": 4.0, "season": "2022-23", "fixture_number": 150, "matches_def": false, "total_points": 5 } ]`

Deliverables:

You are required to submit the following by filling out the form:

1. **Create_kg.py**
   - Python script that creates the Knowledge Graph in Neo4j.
   - Should read CSV files located in the same directory
   - Should follow the exact schema specification.
   - The script should use the config.txt file with Neo4j credentials. Not doing so might result in 0.
2. **config.txt**

```
URI=neo4j://localhost:7687

USERNAME=neo4j

PASSWORD=your_password
```

3. **rule.txt**

A text file containing the implemented rule for the scoring task.

4. **queries.txt**

   A text file containing the queries for the above questions with their numbers.

```
//1. Query 1

Match(...)

Return (....)

//2. Query 2

Match(..)

Return (....)

Etc..
```