INFO216

Group Project – Spring 2021

Building an Esports Knowledge Graph
for the Overwatch League

By Jonatan Berg Romundgard, Sebastian Einar Røkholt

and Ole Einar Markeset

Word count: 1488

Table of Contents

Introduction and Project Rationale

Professionally organized videogame competitions, widely known as electronic sports or *esports*, is an international phenomenon that has grown considerably over recent years *(Green Man Gaming, 2019*; *Willingham, A., 2018; Grand View Research, 2020)*. According to the entertainment analytics company *Newzoo,* this trend is expected to continue as global esports revenues are estimated to reach $1,084 million in 2021 and $1,617 million in 2024 *(Newzoo, 2021).* Though esports is still new when compared to more traditional sports, the covid-19 pandemic seems to have accelerated the convergence rate between esports and more traditional sports as viewership rates for videogame streaming surge towards mainstream popularity *(*Stubbs, M., 2020*; Reyes, M. S., 2021)*.

While traditional sports enjoy the benefits from a massive sports analytics industry, esports have been limited to a few organizations that keep track of player, match and tournament statistics. There is a significant untapped market in the esports analytics industry, especially considering that collecting data from videogame tournaments is much easier than in traditional sports due to the inherent digitization *(Rubleske et. al, 2020).* Esports data can be collected directly from the source and is therefore both more accurate and granular *(Studholme, B. 2021)*. The domain of esports is complex, consisting of an endless number of different types of data that are suitable for analysis. Data is also generated in real time, which means that any tool that can process and distill this information quickly will be great asset for team staff, players, betting services, gamblers and other types of esports enthusiasts.

With this project we aimed to create an analytics tool for esports leagues using semantic web technologies. Existing solutions such as *Liquipedia.net*, *The Overwatch Statslab* and various esports betting platforms such as *unikrn.com* and *topbettingesports.com* are limited in terms of finding meaningful patterns in the large amounts of data. Statistics are mainly presented through tables or news articles, with limited options for filtering or closer investigation. Each instance of data is isolated, and in the case of the videogame *Overwatch,* there is no centralized repository for all esports statistics. Other existing analytics tools may improve upon the problems listed here, but they are generally expensive (*Newzoo, 2021*), and they do not provide semantic highlighting or linking of data.

Earlier this semester, we speculated that esports would be the perfect application for semantic web technologies. Esports is a complex domain where the relationships between players, teams and matches can provide valuable insight which otherwise would have been difficult or impossible to obtain. With the purpose of addressing the problems with existing analytics tools, we have created a semantic knowledge graph for the annual esports tournament *The Overwatch League*. We have also built a simple and lightweight ontology for esports to accompany our graph, due to the lack of suitable open-source ontologies and vocabularies published in the Linked Data format.

By importing our knowledge graph to Blazegraph, we were able to run queries against our knowledge graph that gave valuable information about relationships between players, in-game objects and teams. For example, we were able to query our knowledge graph to produce a table with the players who had selected a given Overwatch hero ("Sigma") for more than 20% of all matches they had played in the Overwatch League, including data about these players' age and nationality.

```sparql
PREFIX ex: <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX DBpedia: <http://dbpedia.org/resource/>
PREFIX dbp_o: <http://dbpedia.org/ontology/>
PREFIX FOAF: <http://xmlns.com/foaf/0.1/>

# Which players have ex:playedHeroes contains "Sigma" with a pick rate above 20%

SELECT ?player ?hero ?pickRate ?age ?nationality
WHERE {
  # get hero pick rate for south korean players
  ?player ex:playedHeroes/rdf:rest*/rdf:first ?heroPickRate ;
          dbp_o:nationality ?nationality ;
          FOAF:age ?age .

  # replace string from ?heroPickRate with regex and bind to new variables
  BIND( REPLACE(?heroPickRate, ".*\\s|%$", "") AS ?tempPickRate )
  BIND( REPLACE(?heroPickRate, "\\s.*", "") AS ?hero )

  # divide pick rate by 100 and get 2 decimals
  BIND( xsd:float(?tempPickRate)/xsd:integer(100) AS ?floatPickRate )
  BIND( SUBSTR(str(?floatPickRate), 0, 5) AS ?pickRate )

  # filter out heroes that are not "Sigma" and pick rates below 0.2
  FILTER ( ?hero = "Sigma" && ?floatPickRate > 0.2 )
}

ORDER BY DESC(?pickRate) DESC(?age)
```

Fig. 1: A SPARQL query

player	hero	pickRate	age	nationality
<http://example.org/hawk>	Sigma	0.35	19	<http://dbpedia.org/resource/United_States>
<http://example.org/brussen>	Sigma	0.33	20	<http://dbpedia.org/resource/Netherlands>
<http://example.org/punk>	Sigma	0.28	19	<http://dbpedia.org/resource/Australia>
<http://example.org/cr0ng>	Sigma	0.26	22	<http://dbpedia.org/resource/South_Korea>
<http://example.org/qoq>	Sigma	0.26	20	<http://dbpedia.org/resource/South_Korea>
<http://example.org/toyou>	Sigma	0.26	19	<http://dbpedia.org/resource/South_Korea>
<http://example.org/hanbin>	Sigma	0.26	19	<http://dbpedia.org/resource/South_Korea>
<http://example.org/showcheng>	Sigma	0.25	20	<http://dbpedia.org/resource/China>
<http://example.org/bernar>	Sigma	0.25	20	<http://dbpedia.org/resource/South_Korea>
<http://example.org/mcgravy>	Sigma	0.23	24	<http://dbpedia.org/resource/United_States>
<http://example.org/gargoyle>	Sigma	0.23	22	<http://dbpedia.org/resource/South_Korea>
<http://example.org/piggy>	Sigma	0.22	21	<http://dbpedia.org/resource/South_Korea>
<http://example.org/frdwnr>	Sigma	0.22	21	<http://dbpedia.org/resource/United_States>
<http://example.org/bianca>	Sigma	0.21	19	<http://dbpedia.org/resource/South_Korea>

Fig. 2: Blazegraph output

Though it may seem trivial, these kinds of patterns are very difficult – if not outright impossible – to visualize without using semantic technologies. We also gain the benefit of being able to quickly locate related resources, such as the DBpedia page for the country of a player's nationality. Keep in mind that this graph only contains a miniscule subset of publicly available esports data.

Methodology and Tools

We have used a wide array of different tools for this project. We began by manually downloading datasets from the Overwatch Stats Lab[1] and importing them to Python with the Pandas[2] library. We then supplemented these datasets with data from Liquipedia.net[3] by writing a Python script to access Liquipedia's API[4]. The data was then imported, processed and condensed in Python.

The main reason why we decided to use Overwatch Stats Lab is that it is maintained by Blizzard, the company that developed Overwatch and hosts the Overwatch League every year. This means that the data is reliable and gets updated automatically after every match in the league. We used Liquipedia because it is the single largest repository for information about the Overwatch League. It is a community-driven wiki where thousands of contributors keep the information updated *(Liquipedia, 2021; Discord, 2021).* Alternative sources for information, such as *omnicmeta.com* or *winstonslab.com,* generally contained much less information and were a less reputable and reliable source.

We generated the triples for the knowledge graph with Python using the RDFLib[5] library. Early on, we discovered that there was a limited number of ontologies about sports events on the Linked Open Data cloud, as well as vocabularies that were suitable for our project. We considered using the *Match* and *Competition* classes from the *BBC Sport Ontology*, but they were very limited and would require a lot of custom extensions anyway. Additionally, the ontology had no *Player* or *Map* classes, so we would have to create those ourselves. We therefore decided to develop our own ontology using *example.org* as a placeholder namespace. We used OWL and RDFS to create classes and properties for players, matches, tournaments and in-game locations ("maps"), while a few appropriate terms from the DBpedia, Schema and FOAF vocabularies were used to extend them. We also implemented the existing SportsTeam class from DBpedia to represent the competing teams.

After creating the very basic esports ontology, we implemented automatic semantic highlighting of data retrieved from Liquipedia with DBpedia's Spotlight[6] tool. This enabled us to easily add

---

[1] https://overwatchleague.com/en-us/statslab-teamfights
[2] https://pandas.pydata.org
[3] https://liquipedia.net/
[4] https://liquipedia.net/overwatch/api.php
[5] https://rdflib.readthedocs.io/en/stable/
[6] https://www.dbpedia-spotlight.org/

triples with ontology terms and resources from DBpedia[7] and Schema.org[8] to supplement graph entities suchas players or teams.

The final generated graph was written to a Turtle file, which was then imported into Blazegraph[9]. We considered running the SPARQL queries in Python but found the Blazegraph interface to be easier to work with. The queries were generally very quick to run and allowed us to realize the knowledge graph's full potential.

A flow chart describing the data pipeline is presented below, detailing the processing steps from source (left) to Blazegraph (right).

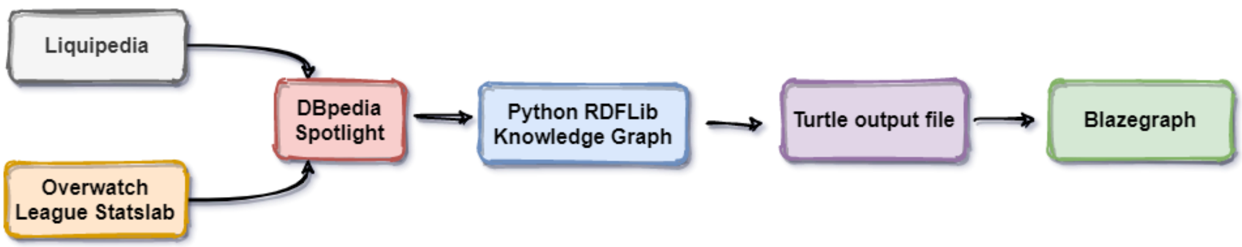

Fig. 3: Flowchart describing the processing steps (created with Draw.io)

Finally, we used *WebVowl*[10] to visualize the ontology we built for our graph by importing the Turtle file to the WebVOWL platform. Each class is represented as a dark blue node, while the edges representing the properties between classes. Yellow boxes represent an object where the range of possible values is restricted to literals, such as strings, integers or dates. In cases where multiple classes share the same property, a light blue circle is used to specify a disjoint relationship between the property's different domains and ranges. The sizes of the nodes for each term represent their importance in the ontology in terms of node modularity.

---

[7] https://www.dbpedia.org/

[8] https://schema.org/

[9] http://sandbox.i2s.uib.no/bigdata/#splash

[10] http://www.visualdataweb.de/webvowl/

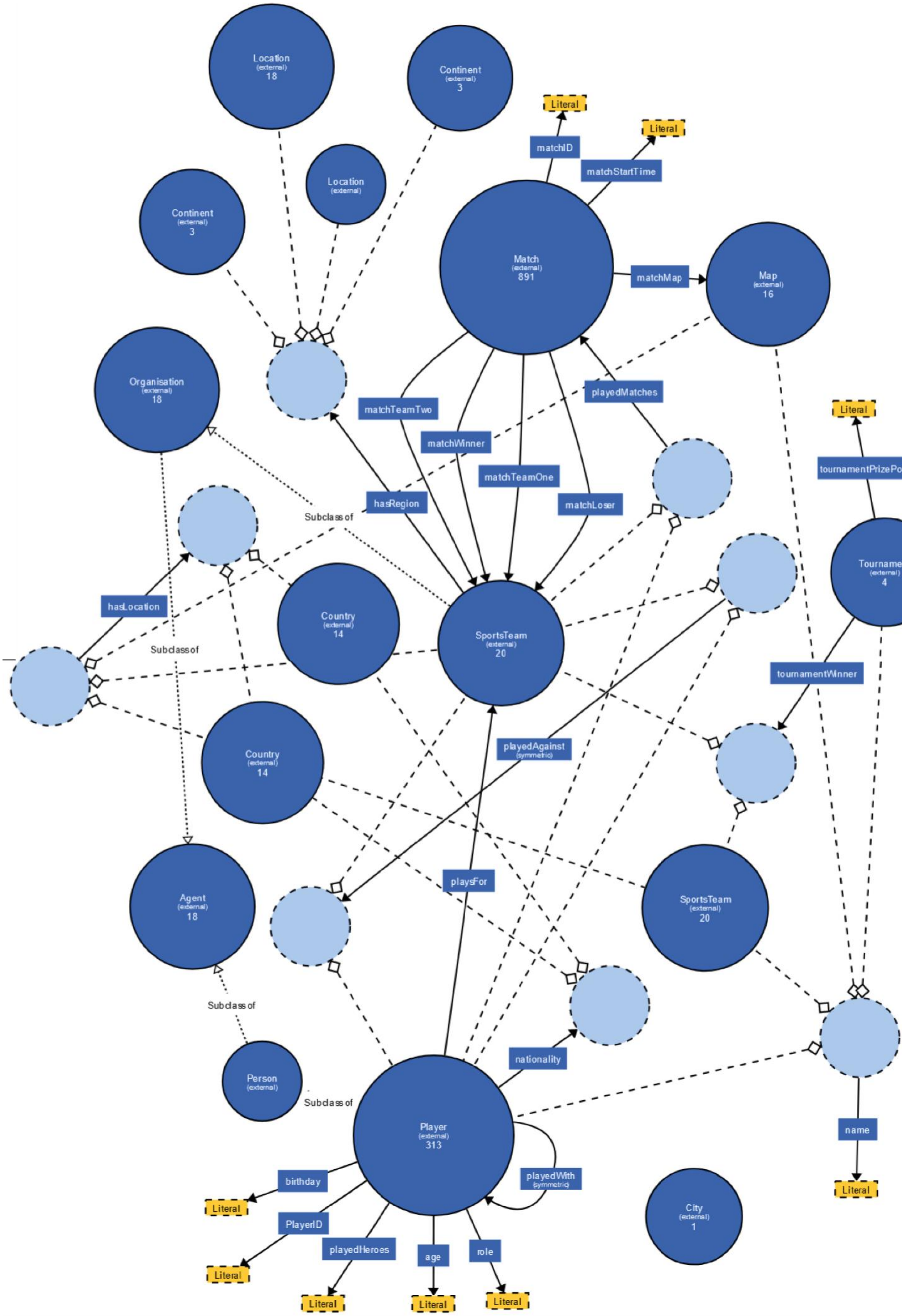

Fig. 4: Visualizing the E-Sports Ontology with WebVOWL

Reflection

In addition to problems with finding a suitable ontology for e-sports, we faced several issues that made the project more challenging than expected. Firstly, Liquipedia had a few inconsistencies and errors that resulted in data missing from our graph. We also discovered that some players had participated in matches under different player IDs, aka. gamertags, which was probably because they changed their ID over the course of the four years of Overwatch League tournaments we had collected data from. These issues were quite difficult to debug and handle, and we realized that we should have queried our graph earlier in the development process to test the quality of the graph's contents.

We also faced issues because of a weakness in DBpedia Spotlight's algorithm for calculating similarity scores between possible terms and the query resource. The weakness resulted in a lower similarity score for some terms when simultaneously querying multiple resources, so we had to query every resource individually instead. We also attempted to use Spotlight to retrieve the DBpedia resource for the players, as many Overwatch League players have their own DBpedia page. However, Spotlight struggled to locate the correct resource because many player IDs were confused with other resources (e.g: one player was named “hydration”).

If we were to redo the project from the beginning, we would have chosen to create our own namespace for the esports ontology. It would also have been wise to frequently test the graph with SPARQL queries during development in order to verify that no triples were missing or contained errors. Finally, we would have liked to obtain data from more sources to supplement our graph further. The ideal source of data would of course be to access in-game resources directly by collaborating directly with Blizzard.

Contributions

The group's members contributed equally to the project.

References

Discord (2021). The Liquipedia contributor's server at https://discord.gg/liquipedia which currently has more than 12,700 members.

Grand View Research (2020). *Esports Market Size, Share & Trends Analysis Report By Revenue Source*. Retrieved 06.05.2021 from: https://www.grandviewresearch.com/industry-analysis/esports-market

Green Man Gaming (2019). *The Money Game.* Retrieved 07.05.2021 from: https://www.greenmangaming.com/the-money-game/

Liquipedia (2021). Retrieved 06.05.2021 from: https://liquipedia.net/

Newzoo (2021). *Global Esports and Live Streaming Report 2021.* Retrieved 06.05.2021 from: https://newzoo.com/products/reports/global-esports-live-streaming-market-report

Newzoo (2021). *Newzoo Expert: The Gold Standard for Games Market Data.* Retrieved 06.05.2021 from: https://newzoo.com/expert

Reyes, M. S. (2021, Jan 5). Esports Ecosystem Report 2021. *Business Insider (2021).* Retrieved 06.05.2021 from: https://www.businessinsider.com/esports-ecosystem-market-report?op=1&r=US&IR=T

Rubleske, J., Fletcher, T. & Westerfeld, B. (2020). *E-Sports Analytics: A Primer and Resource for Student Research Projects and Lesson Plans*. Retrieved 06.05.2021 from: http://files.eric.ed.gov/fulltext/EJ1242004.pdf

Stubbs, M. (2020, August 5). New Report From Fnatic Shows Esports Grew Significantly During The Pandemic. *Forbes.* Retrieved 06.05.2021 from: https://www.forbes.com/sites/mikestubbs/2020/08/05/new-report-from-fnatic-shows-esports-grew-significantly-during-the-pandemic/?sh=2eb14ea47de4

Studholme, B. (2021, April) Making Esports Data Beautiful. *The Esports Journal, 8, 25.*

Willingham, A. (2018, Aug 27). What is eSports? A look at an explosive billion-dollar industry. *CNN.* Retrieved 06.05.2021 from: https://edition.cnn.com/2018/08/27/us/esports-what-is-video-game-professional-league-madden-trnd/index.html

Appendix A: Two more SPARQL Examples

Example query 1: Get the average age of all Swedish players.

```
PREFIX ex: <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX DBpediaOntology: <http://dbpedia.org/ontology/>
PREFIX DBpedia: <http://dbpedia.org/resource/>
PREFIX FOAF: <http://xmlns.com/foaf/0.1/>


SELECT (COUNT(?player) AS ?swedishPlayerCount)
       (AVG(?age) AS ?averageAge)

WHERE {
  ?player rdf:type ex:Player ;
          DBpediaOntology:nationality DBpedia:Sweden ;
          FOAF:age ?age ;
}
```

Fig. A.1: SPARQL query for calculating the average age of all Swedish players

swedishPlayerCount	averageAge
10	24.3

Fig. A.2: Blazegraph output

Example query 2: Get all matches where player `bdosin` played against `tobi` and calculate the winner for each match.

```
PREFIX ex: <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ((?First) AS ?teamOnePlayer) ((?Second) AS ?teamTwoPlayer) ?match ?winningPlayer
WHERE {
  # Get match triples and some data about the matches
  ?match rdf:type ex:Match ;
         ex:matchTeamOnePlayer1 ?teamOnePlayer1 ;
         ex:matchTeamOnePlayer2 ?teamOnePlayer2 ;
         ex:matchTeamOnePlayer3 ?teamOnePlayer3 ;
         ex:matchTeamOnePlayer4 ?teamOnePlayer4 ;
         ex:matchTeamOnePlayer5 ?teamOnePlayer5 ;
         ex:matchTeamOnePlayer6 ?teamOnePlayer6 ;
         ex:matchTeamOne ?teamOne ;
         ex:matchTeamTwoPlayer1 ?teamTwoPlayer1 ;
         ex:matchTeamTwoPlayer2 ?teamTwoPlayer2 ;
         ex:matchTeamTwoPlayer3 ?teamTwoPlayer3 ;
         ex:matchTeamTwoPlayer4 ?teamTwoPlayer4 ;
         ex:matchTeamTwoPlayer5 ?teamTwoPlayer5 ;
         ex:matchTeamTwoPlayer6 ?teamTwoPlayer6 ;
         ex:matchTeamTwo ?teamTwo ;
         ex:matchWinner ?winningTeam .

  OPTIONAL {
    ?match ex:matchTeamOnePlayer7 ?teamOnePlayer7 ;
           ex:matchTeamOnePlayer8 ?teamOnePlayer8 ;
           ex:matchTeamOnePlayer9 ?teamOnePlayer9 .

    ?match ex:matchTeamTwoPlayer7 ?teamTwoPlayer7 ;
           ex:matchTeamTwoPlayer8 ?teamTwoPlayer8 ;
           ex:matchTeamTwoPlayer9 ?teamTwoPlayer9 .
  }

  VALUES ?First  { ex:tobi ex:bdosin } .
  VALUES ?Second { ex:bdosin ex:tobi } .
  FILTER (?First = ex:tobi && ?Second = ex:bdosin || ?First = ex:bdosin && ?Second = ex:tobi)

  # Get match triples where one of the listed predicates has a value from ?First
  ?match (ex:matchTeamOnePlayer1 | ex:matchTeamOnePlayer2 | ex:matchTeamOnePlayer3 |
          ex:matchTeamOnePlayer4 | ex:matchTeamOnePlayer5 | ex:matchTeamOnePlayer6 |
          ex:matchTeamOnePlayer7 | ex:matchTeamOnePlayer8 | ex:matchTeamOnePlayer9 ) ?First .

  # Get match triples where one of the listed predicates has a value from ?First
  ?match (ex:matchTeamTwoPlayer1 | ex:matchTeamTwoPlayer2 | ex:matchTeamTwoPlayer3 |
          ex:matchTeamTwoPlayer4 | ex:matchTeamTwoPlayer5 | ex:matchTeamTwoPlayer6 |
          ex:matchTeamTwoPlayer7 | ex:matchTeamTwoPlayer8 | ex:matchTeamTwoPlayer9 ) ?Second .

  # Define variable ?winningPlayer
  BIND( if(?winningTeam = ?teamOne, ?First, ?Second) AS ?winningPlayer )
}
ORDER BY ?match
```

Fig. A.3: SPARQL query for match results where `bdosin` played vs `tobi`

teamOnePlayer	teamTwoPlayer	match	winningPlayer
<http://example.org/tobi>	<http://example.org/bdosin>	<http://example.org/10596>	<http://example.org/bdosin>
<http://example.org/bdosin>	<http://example.org/tobi>	<http://example.org/10323>	<http://example.org/bdosin>
<http://example.org/tobi>	<http://example.org/bdosin>	<http://example.org/10264>	<http://example.org/bdosin>

Fig. A.4: Blazegraph output part 1 (left side)

TeamOne	TeamTwo	winningTeam
<http://dbpedia.org/resource/Seoul_Dynasty>	<http://dbpedia.org/resource/London_Spitfire>	<http://dbpedia.org/resource/London_Spitfire>
<http://dbpedia.org/resource/London_Spitfire>	<http://dbpedia.org/resource/Seoul_Dynasty>	<http://dbpedia.org/resource/London_Spitfire>
<http://dbpedia.org/resource/Seoul_Dynasty>	<http://dbpedia.org/resource/London_Spitfire>	<http://dbpedia.org/resource/London_Spitfire>

Fig. A.5: Blazegraph output part 2 (right side)