Project Proposal

Title : Curating and Integrating Solo Queue and International Tournament Data in League of Legends Online Game

Student : Juhwan Song

Date: September 15, 2025

1. Overview

The purpose of this project is to curate and integrate **solo queue match data** from regular users and **professional match data** from international tournaments in League of Legends (LoL). The outcome will be a **reproducible and reusable curated dataset** along with an automated workflow.

Professional match data is relatively well-structured through platforms such as Oracle's Elixir or Leaguepedia. In contrast, solo queue data obtained directly from Riot's Match-V5 API is less curated, containing missing values, schema inconsistencies, and duplicated or invalid records. By standardizing and harmonizing these two sources into a unified schema, the project will enable:

- Direct **comparisons between professional and everyday players**, highlighting differences in champion selection, build choices, and objective participation.
- A curated, documented dataset that other researchers can reuse and reproduce with transparency.

This project addresses the full **data lifecycle**: acquisition, modeling, quality assessment, cleaning, metadata, workflow automation, provenance, and dissemination.

2. Plan

2.1 Data Acquisition

- **Solo Queue Data:** Retrieved from Riot Games Developer API (Match, Summoner) using accounts from specific regional servers (e.g., South Korea, United States of America).
- **Tournament Data:** Obtained from Oracle's Elixir, which provides CSV/SQL professional match statistics, and complemented with Leaguepedia API for schedules, patch versions, and team metadata.

2.2 Data Modeling and Quality Assessment

- **Unified Schema:**
    - Matches: matchId, gameVersion, queueType, duration, patch, tournament(optional)

  - Participants: playerId/hashed puuid, championId, role, kills, deaths, assists, gold, cs, damage, items[], runes[], outcome

- **Quality Assessment:**
  - Solo queue: remove invalid games, handle missing values.
  - Tournament: cross-check with multiple sources to identify inconsistencies.

2.3 Cleaning and Transformation

- Map champion/item/rune IDs to human-readable labels.
- Align tournament statistics with the solo queue schema.
- Generate derived metrics: Gold per Minute (GPM), Damage per Minute (DPM), Objective Participation.

2.4 Metadata and Documentation

- Provide metadata using **schema.org/Dataset** in JSON format.
- Create a **data dictionary** that defines all variables, their units, and rules for handling missing or derived values.

2.5 Workflow Automation and Provenance

- Automate the pipeline with Python from acquisition to output.
- Capture provenance by documenting API endpoints, timestamps, processing logs, and code commits.

2.6 Packaging and Dissemination

- Comply with Riot's policy by sharing only limited sample or synthetic data, while allowing others to reproduce the dataset using their own API keys.

3. Data Sources

- **Solo Queue Data:** Riot Games Developer API (Match, Summoner). https://developer.riotgames.com
- **Tournament Data:** Oracle's Elixir – Professional match statistics. https://oracleselixir.com
- **Supplementary Metadata:** Leaguepedia API – tournament schedules, teams, and contextual information. https://lol.fandom.com/wiki/Leaguepedia_API

4. Team

This is an **individual project**.

5. Timeline

Date	Subjects
Sept 15	Proposal submission
~Oct 10	Data acquisition (solo queue + tournament) and schema design
~ Oct 25	Data cleaning, integration, and derived metrics genration
~ Oct 27	Progress report submission
~ Nov 25	Workflow automation, metadata creation, environment packaging
~ Dec 10	Final packaging, Documentation, Testing, Final submission

6. Constraints

- **API Rate Limits:** Riot API enforces strict request limits, requiring caching and backoff strategies.
- **Structural Differences:** Tournament datasets and API-based solo queue data vary in schema, requiring careful mapping.
- **Data Sharing Policy:** Riot restricts redistribution of full raw datasets, so only synthetic/sample data can be shared.

7. Gaps

- License terms for Oracle’s Elixir and Leaguepedia must be reviewed.
- Additional practice with DataCite and schema.org metadata standards may be needed.
- Optimizing workflows for large-scale solo queue data is an open challenge.

8. References

- Riot Games. (n.d.). *Riot Developer Portal*. Riot Games. Retrieved September 15, 2025, from https://developer.riotgames.com
- Oracle’s Elixir. (n.d.). *Professional League of Legends match data*. Retrieved September 15, 2025, from https://oracleselixir.com/
- Leaguepedia. (n.d.). *Leaguepedia API*. Retrieved September 15, 2025, from https://lol.fandom.com/wiki/Leaguepedia_API
- schema.org. (n.d.). *Dataset*. Retrieved September 15, 2025, from https://schema.org/Dataset
- W3C. (2013). *PROV-DM: The PROV Data Model*. World Wide Web Consortium. Retrieved September 15, 2025, from https://www.w3.org/TR/prov-dm/
- DataCite Metadata Working Group. (2021). *DataCite Metadata Schema for the Publication and Citation of Research Data (Version 4.4)*. DataCite e.V. https://schema.datacite.org