# Information Visualization

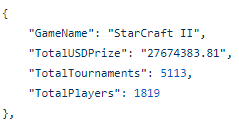
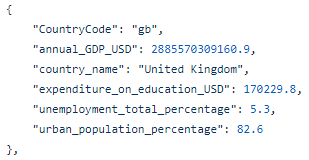
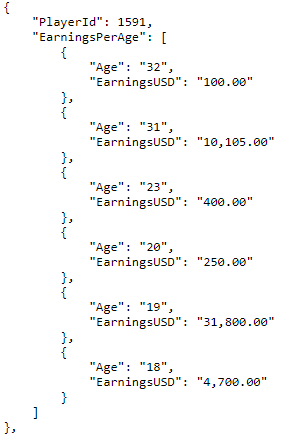
# CHECKPOINT II: Data cleaning and processing

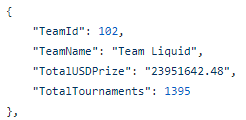
G13-A

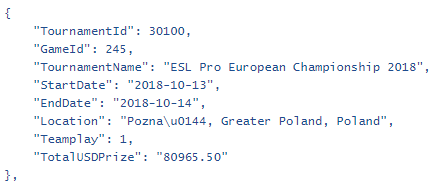
**1. Initial Dataset**

Static tables from esportsearnings.com and worldbank.org’s APIs, and scraping from esportsearnings.com.

Samples:







**2. Selected/Derived Data**

We joined the players and earningsByAge tables by the playerId using Pentaho DI.

We left the rest of the data as we got it from the APIs.

**3. Data abstraction**

Players: static table players.json, containing a player’s ID, their handle (“nickname”), country and total earnings. Attributes:

* playerId, currentHandle: nominal; both identify a player.
* countryCode: nominal; two-letter code identifying the player’s nationality.
* totalUSDPrize: quantitative, ratio; player’s total earnings.
* earningsByAge: sub table with:
  + age: quantitative, ratio; player’s past or present age.
  + earningsUSD: quantitative, ratio; player’s earnings when he was the above age.

Games: static table games.json, containing a game’s name, its earnings (total prize), tournaments and players. Attributes:

* gameName: nominal; identifies a game.
* totalUSDPrize: quantitative, ratio; the game’s total prize money.
* totalTournaments: quantitative, ratio; number of tournaments for that game.
* totalPlayers: quantitative, ratio; number of players for that game.

Teams: static table teams.json, containing a team’s ID, name, tournaments and earnings. Attributes:

* teamId, teamName: nominal; both identify a team/organisation.
* totalTournaments: quantitative, ratio; number of tournaments the team participated in.
* totalUSDPrize: quantitative, ratio; team’s total earnings.

Countries: static table countries.json, containing a country’s name and code, annual GDP, and unemployment/education/urban population metrics. Attributes:

* countryName, countryCode: nominal; identifies a country.
* annualGDP\_USD: quantitative, ratio; country’s annual Gross Domestic Product.
* expenditureOnEducation\_USD: quantitative, ratio; country’s education expenditures.
* urbanPopulationPercentage: quantitative, ratio; country’s urban population.
* unemploymentTotalPercentage: quantitative, ratio; country’s unemployment.

Tournaments: static table tournaments.json, containing game and tournament IDs, start and end dates, and the total prize. Attributes:

* gameId, tournamentId: nominal; they identify the game played and the tournament respectively.
* startDate, endDate: quantitative, hierarchical; start and ending dates for the tournament
* totalUSDPrize: quantitative, ratio; tournament’s total prize pool.

**4. Dataset processing**

The data for games, teams and players was obtained directly from the esportsearnings.com API.

Player earnings by age data was scraped from the same site (using a node.js script to go to each player’s “Tournaments won by age” page and making a .json file from it.)

Country data was obtained from the worldbank.org API in .xlsx format and converted to .json afterwards, with the use of a Python script.

* + - 1. **5. Mapping (Data sample / Questions)**
* **What countries have the highest earnings?**

Group players by countries and get the sum of their earnings.

* **What is the age at which players earn the most?**

Compare each player’s earnings by age.

* **What organizations earned the most?**

Sort the teams table by earnings.

* **What games have the most earnings?**

Sort the games table by earnings.

* **What months are the most active in esports?**

Use the dates from the tournaments table.