TEAMWORK:  **ANIME ETL**

**Introduction**

We decided to create an anime database using a relational datatable to automatise and make suggestions for anime fans all around the world.This is a similar algorithm that Netflix uses to determine which series or movie the user will enjoy.

**1.Extract**

**a)Data Source**: Kaggle: <https://www.kaggle.com/alancmathew/anime-dataset>

* + Import csv files to notebook:We imported the files from the kaggle anime.cvs that contains all the data from anime including gender, type of movie and tv, start date,end date, id cliente.There is another file with the rating for each client.

**b)Create tables Python:** To make different joins we created the next tables in python.

* + user: table which describe the users cultural and geographical attributes.This table contains the 10 countries, gender and age of the user.
  + gender: This table selects the category or gender for each anime.
  + gender\_id:This table was created to link the animes with their gender.

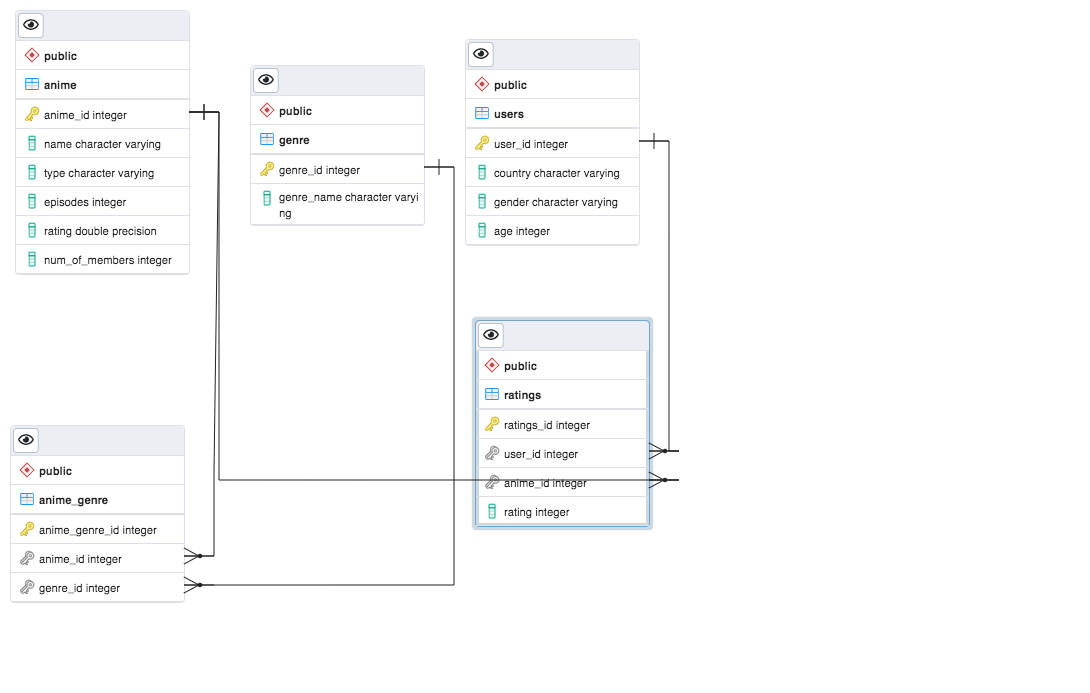
**2. Transformation**

**a) Cleaning**

* + Deleted rating -1: We decided to delete the rating with -1 because this people havnet watch and rate the anime, it wouldn't be fair to take the into consideration.
  + Split Multiple Categories of Genre:We decided to separate each category to measure each category. We create about 40 different categories for each anime and movie.
  + Delet episode column with unknown values: We decided to delete the anime with known values. Because we found out there were many values unknown and many anime shows still running and some anime finished with unknown values.

**b) Joining/ Filtering:We have created 4 tables in sql**

* Anime:Explains how anime is categories
  + anime\_id(integer) primary key: joining with anime\_id in rating table.
  + name (varchar)
  + type (varchar)
  + episodes(integer)
  + rating(integer)
  + num\_members(integers)
* rating: Describe the rating of each user
  + anime\_id(integer) forigen key: joining with anime\_id in rating table and joining with anime\_gender
  + user\_id (integer) forigen key:joining with user table
  + rating(integer)
* users:Describes the user geographical attributes
  + user\_id (integer) forigen key:joining with rating table
  + country character (varchar)
  + gendar character(varchar)
  + age (varchar)
* anime\_gender:Linkes anime table with gender table
  + anime\_id(integer) primary key: joining with anime table.
  + genre\_id(integer) foreign key: joining with gender table.
* gender:Describe the character gender of the anime
  + genre\_id(integer) foreign key: joining with anime\_gender table.
  + genre\_name (varchar)



**c) Load**

The team decided that the type of final production database to load the data would work better into a relational database. This is because the objective of analysing which genre of members prefer a certain "anime\_genre" would be possible by a relational db. In the same way, this kind of DataBase would allow users and Anime makers not only the countries, gender and age of the members; but also to know which are the preferences of type of anime per age and the rating that members are giving to them.

The final tables or collections are used in the production database. Once the DataBase was confirmed in SQLPosgres, the tables and columns were checked one by one. Once this DataBase called "anime\_db" was built. The team used Jupyter Notebook to connect to SQLPostgres and to load the information to the tables.

**Conlcusion:**

By doing this projects the team can highlight the following learnings:Teamwork was crucial to the success of this project! By working closely with your team through all phases of the project to ensure giving different perspectives and knowledge to the project tackling more difficult problems.The process of ETL is entirely iterative requiring constant attention to data quality.