FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

LAB ASSIGNMENT
MECD - Data Warehouses
April 2024

Bruno Dias (201907828)
Helena Costa (202302812)
Lara Sá Neves (202007579)
Pedro Tavares (201806358)

Abstract

With an increase in process complexity and the market need for better and faster data analysis, Data Warehouses are becoming more and more inevitable. In this project, a Data Warehouse is designed and loaded for My Anime List, a website for tracking anime. An operational database was first loaded using real data from the website. Then, a dimensional model was built using a star schema and bus matrix. Finally, the facts are decided for each star and the loading process of the data warehouse is extensively covered. The result is a data warehouse with a time dimension and several bridge tables to convey a group structure to many to many relationships. The aggregations allow user focused analysis and anime seasonal statistics.

1. Introduction

In the past few decades, only a few tv shows, movie tracking apps and websites have been designed. Some of them became huge successes, like IMDB and TV Time. In similarity, with the sudden rise of the Anime industry (a worldwide phenomenon that has attracted S&P 500 companies, such as Netflix) another opportunity emerged for a tracking app to fill a clear gap in the market. It was in this context that MyAnimeList, MAL, was created: a tracking website focused on Anime only.

The aim of this project is to build a Data Warehouse for MyAnimeList based on a dataset of csv files with extracted data from the website. It will focus on two different business perspectives. The first one, the main product, focuses on allowing each user to have a list where they can add animes and a corresponding status and score. This is how it is tracked what animes are either watched or being watched, between several other possibilities. The second perspective addresses user recommendations, based on user and anime information, such as demographic data and anime genres.

The chosen dataset [1] aligns with project requirements and complies featuring a central fact table ('UserAnimeList') exceeding one million records, and eight dimensions [including a temporal dimension] (User, Anime, Genre, ...). Aggregations provide semi-additive measures (e.g., users per country), and while the main fact table lacks a purely additive measure, viewership data (e.g, number of views per season) within aggregations offers an equivalent substitute.

2. Planning: dimensional bus matrix, dimensions and facts dictionary

Dimensional Bus Matrix

This section focuses on the planning of the data warehouses through the star model methodology. The bus matrix is presented in Table 1. This data warehouse involves 10 different dimensions and 3 stars. The interaction star has the smaller granularity (one rating or status update) and the other two stars, user_evolution (User Evolution) and anime_stats (Anime Statistics), are aggregations.

	Anime	User	Day	Season	Producer	Licensor	Studio	Genre	Country	Year
user_evolution		x						x	x	x
interaction	x	x	x		x	x	x	x		
anime_stats	x			x						

Table 1 - Dimensional Bus Matrix

Dimensions

The Data Warehouse has 10 dimensions associated. Some of them come naturally from the interaction process: Anime, User and Day. The Producer, Licensor, Studio and Genre dimensions were chosen for two reasons. First, they are interesting areas for conducting studies. Additionally, conveying the many to many relationship through a dimension makes analysis easier than having arrays of values stored. Regarding the time dimensions, the Day was projected into the Year dimension due to the year granularity of the User Evolution. Similarly, the Season dimension emerges from the granularity of Anime Statistics.

The Season dimension could not be inserted into the Days hierarchical dimension as the year. This is because the Winter season belongs to two different years. To overcome this, it was decided to create the Season as a completely independent dimension from the Days, and insert a start date and end date for each of them. This aims to facilitate joins between tables that have a season granularity and day granularity. Another alternative would be to make the start date a date_id from the dimension Days, but there is no clear advantage in doing this.

There is a Dimension Table for each dimension presented. For instance, the Anime Dimension is explained in detail in Table 2. The other Tables can be found in the Appendix Section.

For the Producer, Licensor, Studio and Genre dimensions, the many to many relationship is established through bridge tables. This design groups different entities, allowing analysis per group and understanding how important one entity is in the group. This comes at the cost of forcing this grouping, which is not exactly clear to non-experts in the industry. The bridge tables are further explained in tables similar to the ones made for the dimensions. The bridge table for genre is shown in Table 3, while the remainder are shown in the section Appendix.

Name	*Description*	*SCD*	*Version*	1.0	*Date*	3/26/2024
Anime	A anime present in the platform since it was created	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Anime_id	Anime		PK	Int		
Anime_id_mal	Anime in the operational		UK	Int		
Title	Title of the anime		UK	Text		
Title_japanese	Anime's title in japanese		UK	Text		
Image_url	Link of anime's image		UK	Text		
Anime_type	Type of the anime			Text		
Source	Anime's source			Text		
Episodes	Number of episodes available			Int		
Status	Anime may still be on the air, may not have aired yet, or may have already ended			Text		
Aired_from	Date of the first aired episode			Date		
Aired_to	Date of the last aired episode			Date		
Duration_minutes	An episode's duration in minutes			Int		
Rating	Age gap appropriate			Text		
Broadcast	Regular schedule			Text		
Opening_theme	Opening song		UK	Text		
Ending_theme	Ending song		UK	Text		

Table 2 - Anime: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Bridge_genre	Bridge to aggregate genres into groups	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Genre_group_id	Genre group		PK	Int		
Genre_id	Genre		PK	Int		
Weight	Each genre holds a weight within the anime genre group, with these weights summing to one and reflecting their importance			Double		

Table 3 - Bridge Genre: Dimension

Facts

For the fact table Interactions, there are three different facts: score, status and watched_episodes. The score and status are not additive facts and the watched_episodes is semi-additive (can be summed across the User dimension, but can not be summed across the Day dimension). Please check Table 4 for more detail.

Regarding the fact table User Evolution there are five different facts: age, years_activity, nr_users_per_country, average_users_age_per_country and average_score_per_user_per_year. These facts enable an analysis specialized on a specific user, providing data relevant for recommendations and yearly relevant statistics for users. Please check Table 17 in the Appendix section for more detail.

Concerning the fact table Anime Stats, the season_views (additive measure), the season_ratings (additive measure) and season_popularity_rank focus on the granularity season. The facts score, score_rank, number_views and number_ratings focus on cumulative measures until that season. For practical reasons, it was decided to put a lower limit of 1997 on this analysis. This allowed faster queries while losing very little relevant data, as the emphasis of the analysis is in the past few years. Check Table 18 in the Appendix section for more detail.

Star	Interactions	Version	1.0	Date	26/03/2024
Granularity	Each interaction from a user with an anime. This includes adding it to the user list with a status or rating it with a score				
Dimensions					
User_id	User				
Anime_id	Anime				
Date_id	Date of the interaction				
Genre_group_id	Genre				
Studio_group_id	Studio				
Licensor_group_id	Licensor				
Producer_group_id	Producer				
Measures					
sccore	User rating of the anime - a integer value from 1 to 10				
status	Status of the anime on the user list, for instance "watching", "completed"				
watched_episodes	Number of episodes the user watch of the referenced anime				

Table 4 - Interactions: Fact Table

3. Dimensional data model

Format justification

In the beginning, we were thinking about the basic processes in our case and it was clear that 'watching the anime' and 'evaluating it' were it. However, as we had information about the user's status regarding the anime, the date they started watching it, among other things, we thought it would be more interesting for the 'watch' star to transform into the 'interaction' star to represent not only the anime that the user watched but also those they stopped watching, the ones they want to watch, etc. After some discussion, we questioned why the evaluation of the anime couldn't also belong to the interaction star because, in fact, the evaluation of the anime is an interaction of the user with it. So we ended up with a very complete '**Interaction'** star, which allowed us to create two others to analyze users and anime separately. Initially, we thought of having one star to analyze user records and another with user statistics; however, we decided to combine them into the **'User_evolution**' star, since they had the same dimensions. Additionally, we created the '**Anime_stats**' star to have a seasonal aggregation for an anime. The granularity chosen was by season due to the fact that the anime releases are partitioned into distinct seasons, approximately equivalent to the seasons of the year.

Dimensional Data Warehouse

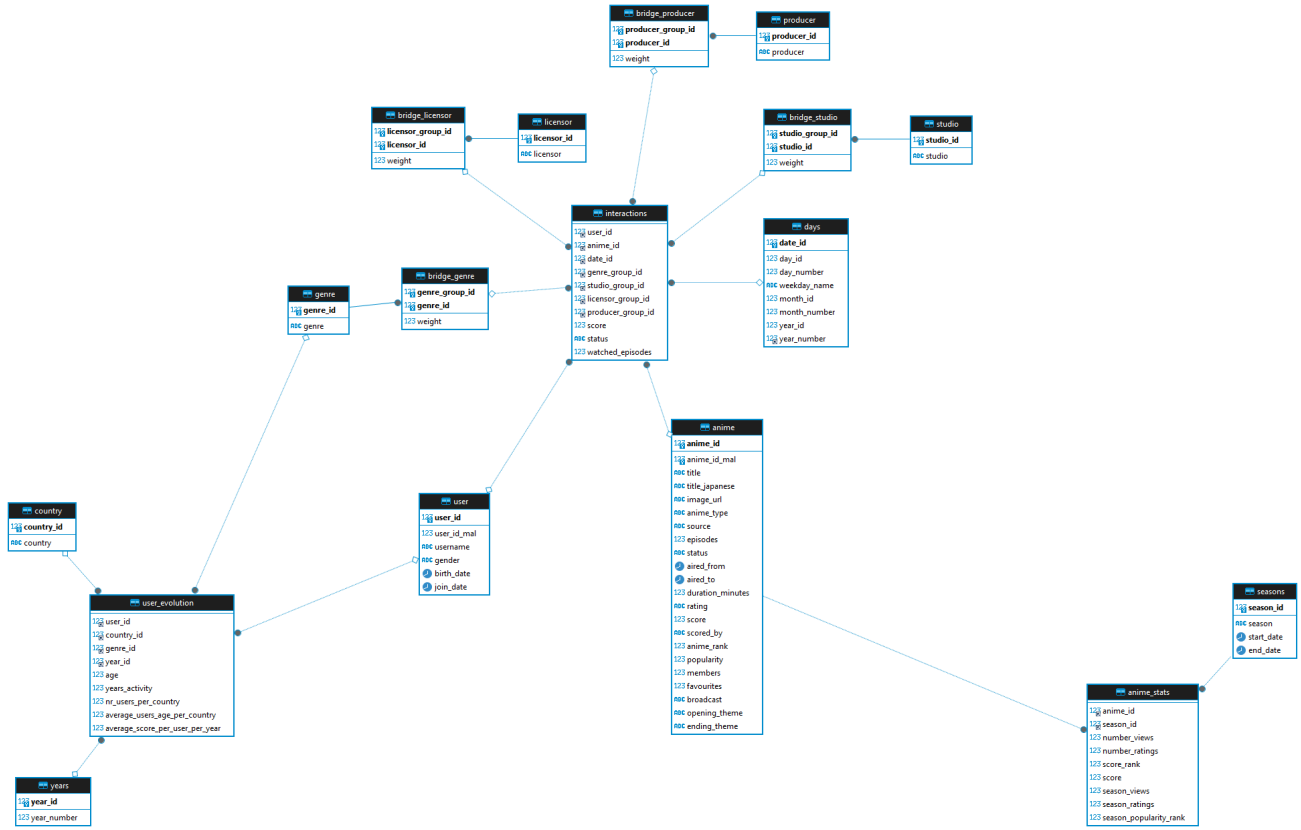


Fig.1 - ER Diagram of the Data Warehouse

After having our operational data warehouse, stars and dimensions defined, we started constructing the dimensional data warehouse by writing the appropriate queries. We decided not to include several columns from the original dataset. Here's a breakdown of these excluded columns and the reasoning behind it. Please note that the corresponding ETL diagrams of the following sections are presented in section 4. For now, Figure 1 shows the ER Diagram of the Data Warehouse constructed after following the ETL.

Star 1: Interactions - Facts

Score: Stores the score a user gave to an anime, extracted directly from our operational model. If null, it means no score was given. The score value is an integer and falls between one and ten.

Status: Captures a user's relationship with an anime (Watching, Completed, On Hold, Dropped, Plan to Watch). This enables insights such as:

- Anime Popularity Prediction: High "Plan to Watch" counts suggest potential viewership surges.
- User Engagement Profiling: Track user activity levels by the variety of anime statuses they have. We'll visualize these patterns to identify highly active vs. casual viewers.

Watched_episodes: Records the number of episodes each user has seen per anime. This allows us to calculate total episodes watched by a user and see in which years the user spent more time watching anime.

Star 1: Interactions - Creation

This star schema integrates the 'User', 'Anime', 'Genre', 'Studio', 'Licensor', and 'Producer' dimensions. To accommodate many-to-many relationships between the fact table and the latter four dimensions, bridge tables were employed. These bridge tables contain a 'group_id' (independent from the anime), an ID linking to each dimension, and a weight (calculated as 1 divided by the number of entities in each group). The process of creation and loading of the bridge tables is explained in more depth in section 4.

After establishing the bridge tables and connecting the 'Anime' and 'User' dimensions to the fact table, data extraction and loading for the facts were easily accomplished from our operational model.

Star 2: User Evolution - Facts|

Age: Stores each user's age (calculated from their birth date). Helps us understand the age distribution of our user base and how behavior might vary across age groups.

Years_activity: Tracks how long a user has been registered (calculated from their join date). Offers insights into user loyalty and how engagement might change over time.

Nr_users_per_country: Provides a geographical breakdown of our users. Enables analysis of the application's popularity in different regions.

Average_users_age_per_country: Calculates the average user age within each country. Helps identify regional trends in user demographics and potential differences in content preferences.

Favorite genre: Records each user's preferred content genre. This topic was created using the weights from the genre bridge table. Crucial for understanding individual tastes and overall genre popularity.

Average_score_per_user_per_year: Tracks the average rating a user gives each year. This allows us to spot years with notably high or low scores, potentially indicating trends in the quality of anime releases.

Star 2: User Evolution - Creation process

This star integrates several dimensions, ‘User’, ‘Year’, ‘Genre’, ‘Country’. The ‘User’ dimension, such as the ‘Interactions’ star, records information on the user and the ‘Year’ is the time dimension used for this star. Then we have 2 dimensions that were of difficult implementation, the ‘Genre’ that separates each genre for the multiple genres each anime can have and the ‘Country’ that records information of possible locations of the users. The work behind the creation of these dimensions are explained in greater detail in section 4 of ETL

Star 3: Anime_stats - Facts

Number_views: The total count of times an anime has been watched from start to finish up until a specific season.

Number_ratings: The total number of ratings submitted by users who have watched an anime in its entirety up to a specific season. Having higher values for both Number_views and Number_ratings suggests a larger, more engaged audience.

Score_rank: The position of an anime within a ranking based on its overall score. This fact demonstrates how an anime compares to others in terms of overall user ratings.

Score: The cumulative average rating of an anime, calculated from all user ratings submitted up to a specific season. Revealing how favorably an anime is perceived by its viewers. An anime with a high Score but relatively low Number_views may be a potential "sleeper hit", deserving of additional promotion.

Season_views: The total number of times an anime has been fully viewed within a particular season.

Season_ratings: The total number of ratings given by users who have completed an anime within a particular season. By identifying spikes or dips in Season_views and Season_ratings, it is possible to measure the impact of new releases or external factors.

Season_popularity_rank: The anime's ranking within a specific season based on the number of times it has been fully watched (season_views).

Star 3: Anime_stats - Creation Process

This data model integrates two dimensions: the 'Anime' dimension and the 'Season' dimension. The 'Anime' dimension functions similarly to those in other models, storing information about each anime series. The unique aspect is the 'Season' dimension, created to address the challenge of anime seasons not fitting into a traditional hierarchical time structure.

To ensure data completeness, we guarantee at least one entry for each anime in every season. Even if it wasn't viewed during that time. This enables us to accurately track trends over time.

The fact table within this star schema stores diverse information. This includes semi-additive measures like 'number_views' and 'number_ratings', which can be summed across certain dimensions (like anime) but not others (like time). Additionally, we track aggregations, which are calculated values like average scores or rankings derived from the data. The work behind the creation of these dimensions are explained in greater detail in section 4 of ETL.

4. ETL process - Extraction, transformation and loading

In this section the ETL, Extraction Transformation and Loading, process is explained. It uses the tools MySQL, Dbeaver, Python and Excel. Excel is used for preprocessing of the raw csv files. Dbeaver and MySQL are used to create the databases and import the data from external files (csv and sql). Finally, Python is used to run MySQL queries to complete the Data Warehouse, DW. Python was chosen to facilitate building the DW using a project with modularity, instead of a huge and complex sql script.

The conceptual design of the ETL process, as an overall view, is shown in Figure 2.

It starts by loading - using Dbeaver restore database tool - a world database from a sql file that can be found in a github repository [4] that contains ISO codes, country code, capital, native language, timezones (for countries), and more.

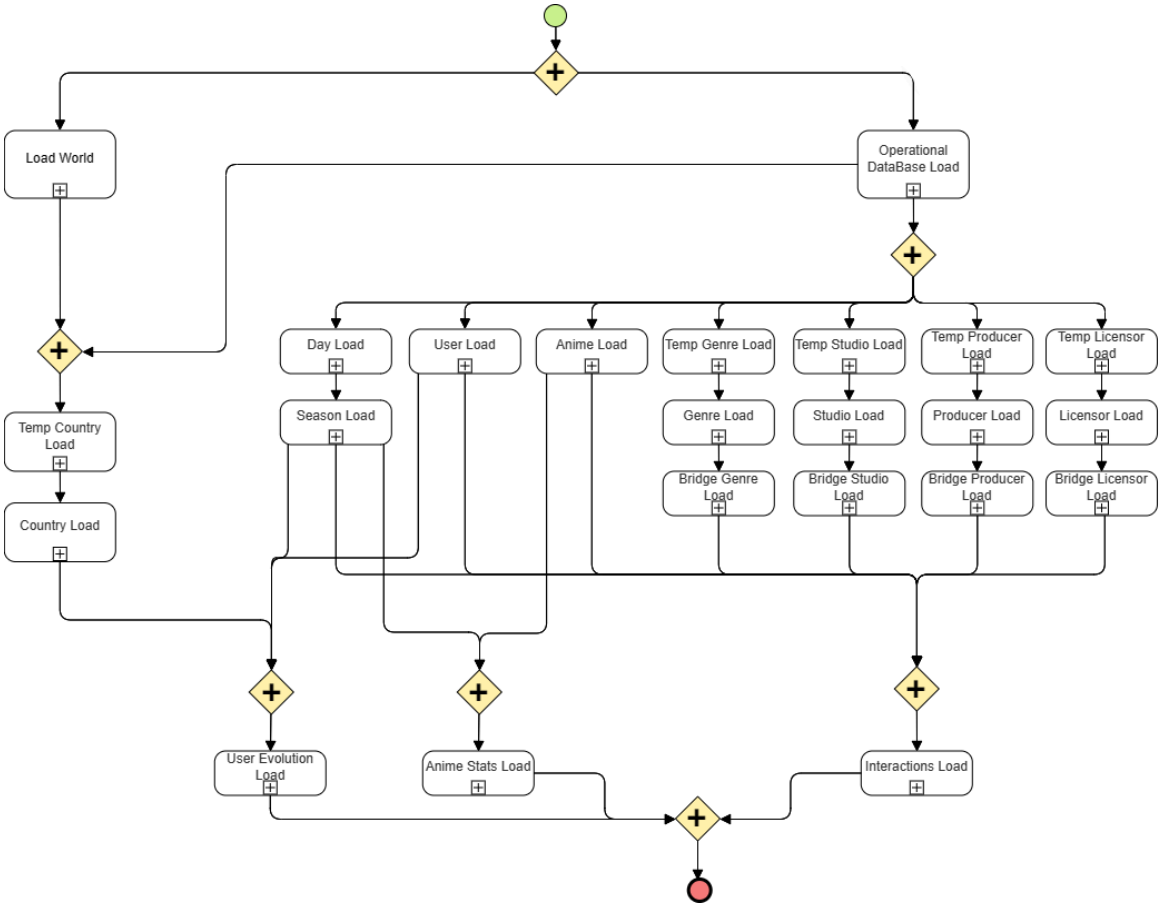


Fig.2 - Conceptual design of the ETL process of building the data warehouse. BPMN methodology was used and it focus on the overall view of the process.

In parallel, the operational database is loaded, performing operations in both Excel and SQL language.

The process is further explained in Figure 3. Each one of the three tables of the operational model is loaded using csv's that can be found in the original source kaggle [1]. However, previously some steps of preprocessing are needed, such as changing data formats, changing column names that belong to native SQL operations - for instance, type - and converting booleans representation from string to

numerical. Regarding loading the csv files into the operational database, most string variables were converted to text instead of varchar to avoid errors due to long strings, making exceptions for keys that needed to be unique. Then, duplicate user_id's are removed from the user table and the user_id column is inserted into the interaction table using the username as mapping. Finally, user_id and anime_id columns are used to create primary keys and foreign keys.

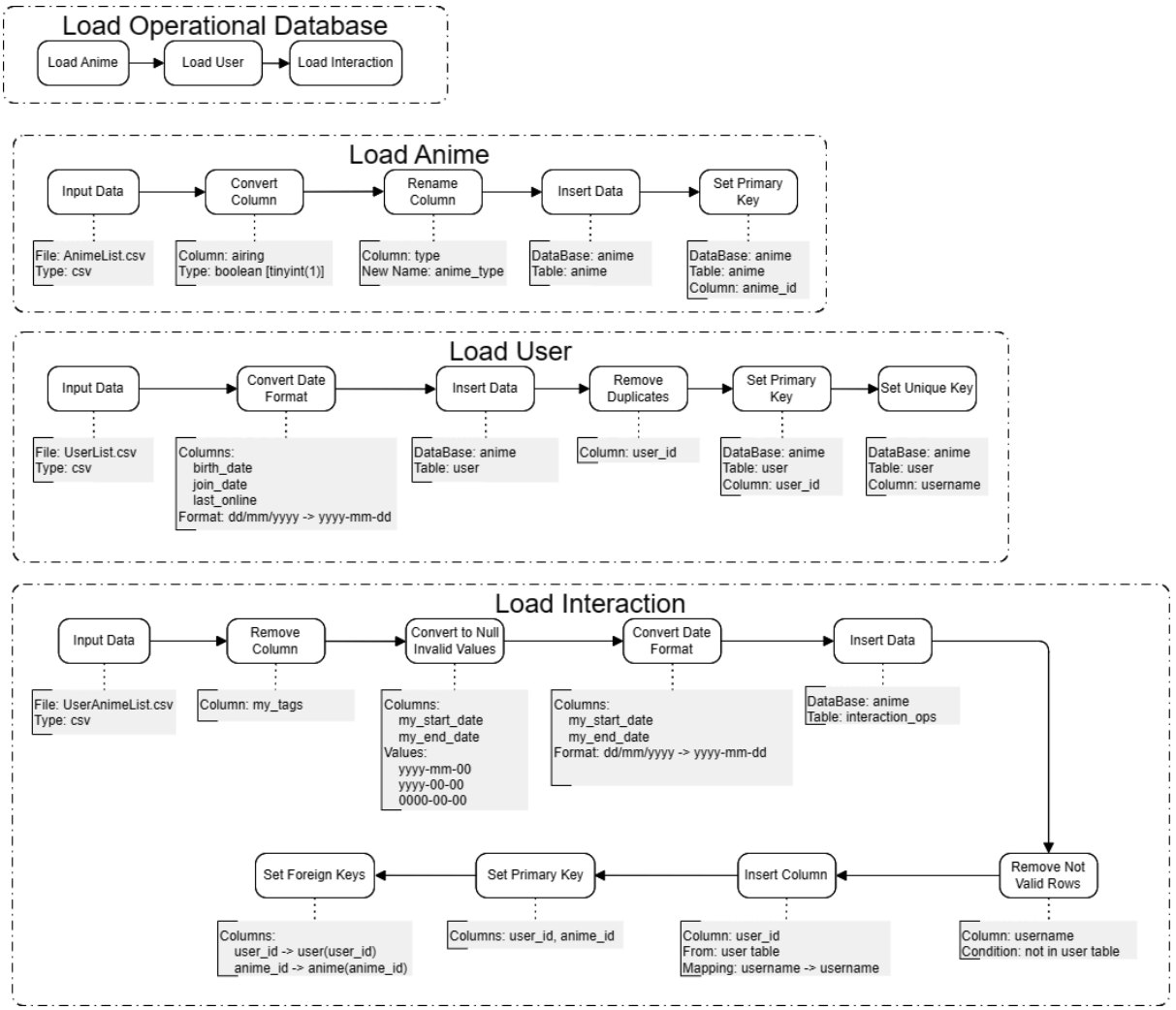


Fig.3 - Conceptual design of the ETL process focusing on the data task of loading the operational database into MySQL.

We also would like to detail some additional ETL processes in our anime dimension, which we consider to be one of the most important ones.

Some columns were deleted due to redundancy: title_english, title_synonym, aired_string, airing, premiered and access_rank. Others were not included for being irrelevant in the scope of the project: background and related. Finally, some were aggregations without any date reference and, thus, not useful: score, popularity, members, scored_by, favorites, anime_rank, stats_mean_score, stats_rewatched and stats_episodes.

Regarding transformations, the duration format was standardized to minutes, the two dates shown in the aired column were splitted to two date type columns (start and end), and the last_online timestamp was divided into date and time columns.

Additionally, using both the operational and world databases, the column location from the user table is prepared, cleaned and inserted in a temporary table in the anime_dimensional database. Figure 4. shows the whole process with finer detail. The aim of this data task is to extract the country of each user from raw data without any precise structure - column location. To do this, it is assumed that each unity of location is delimited by commas - could be something like city, state, country or country, country. Furthermore, it was decided that only country names and state names would be valid. After doing a case insensitive lookup on countries and states, only users with a clear single country are considered to have a country, otherwise they are assigned null value (for the temporary table, these rows are removed).

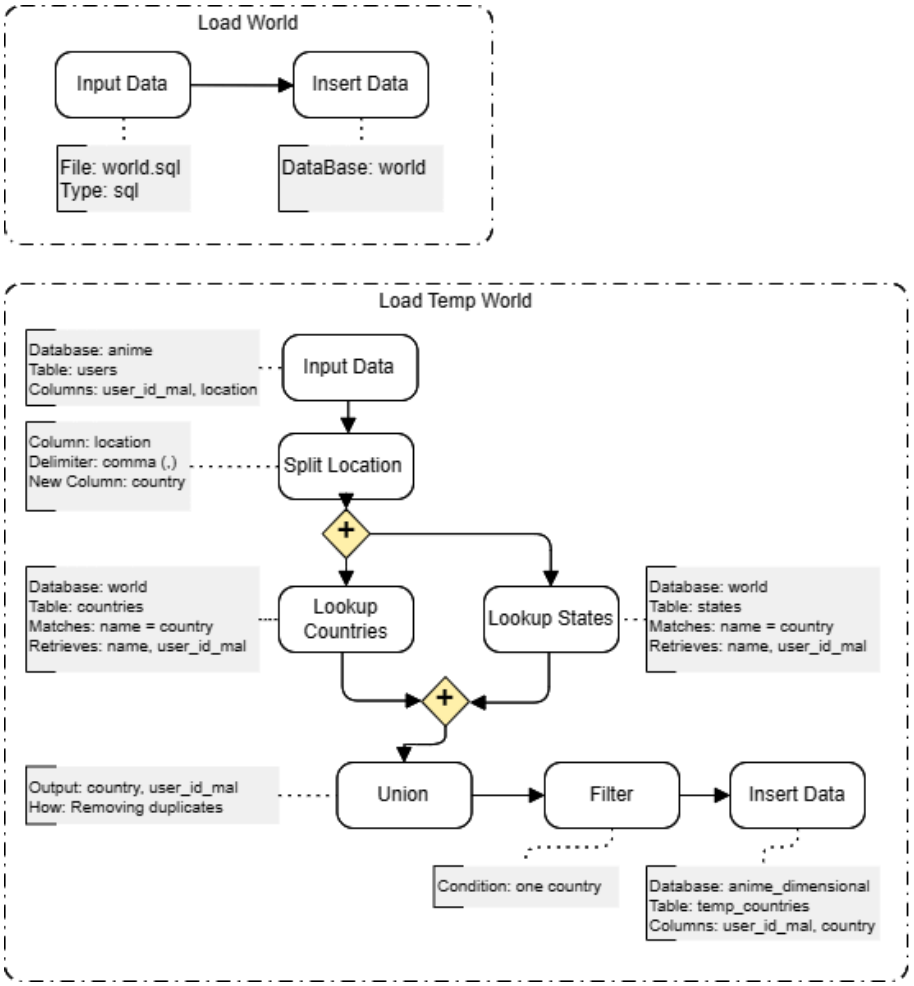


Fig.4 - Conceptual design of the ETL process focusing on the data task of extracting the country of each user from a raw location field without any clear structure.

The rest of the process was constructed doing ad-queries in MySQL and won't be covered in such detail (for more detail check the section Dimensional Model). Nevertheless, it is important to emphasize the division into two phases. The first step loads every dimension in parallel, while the second loads each star in parallel. Notice the existence of temporary tables that are used in both phases.

However, one thing worth explaining in detail is the bridge tables. To build them, first the entities for each anime were splitted from a single string to multiple rows, using a comma as delimiter. Then, using an ordered group_concat, it was possible to add a group id for each anime. Then, the distinct

groups were inserted in the bridge tables, by joining with the first anime entity table, and the temporary tables involved were kept for further usage.

5. Querying and data analysis

5.1 Querying

The best way to understand the concept behind the making of the stars in our data warehouse is by making queries. Query 1 focuses on the 'Interactions' star, specifically examining how users engage with animes from different studios. We analyze animes that users have completed and rated positively. The queries reveal how many users finish anime from each studio, the average number of episodes users complete per studio, and the average user-given score for each studio's anime.

This information helps us identify studios that consistently produce popular, engaging, and well-received content. It can even uncover less-popular studios that create excellent anime.

```
SELECT
    s.studio,
    COUNT(DISTINCT user_id) as num_users,
    avg(i.watched_episodes) as avg_watched_episodes,
    avg(i.score) as avg_score
FROM interactions i
JOIN anime a ON i.anime_id = a.anime_id
JOIN bridge_studio bs ON i.studio_group_id = bs.studio_group_id
JOIN studio s ON bs.studio_id = s.studio_id
WHERE i.status = 'completed' and i.score > 0
GROUP BY s.studio_id, s.studio
ORDER BY num_users DESC, avg_watched_episodes DESC;
```

Query 1

Query 1 utilizes a bridge table to connect to the 'Studio' dimension. This demonstrates how bridge tables handle scenarios where we have multiple associations per interaction (e.g., multiple studios per anime). Our implementation ensures that such complex analyses remain easy to interpret and replicate.

On Query 2, we focus on the ‘User_evolution’ star, delving into the relationship between user age groups and their genre preferences. It aims to categorize users into age brackets and then analyze the popularity of different genres within each group.

Understanding these age-specific preferences can help to provide content recommendations or create more effective, targeted marketing strategies. Additionally, the results could highlight how entertainment preferences shift between generations, offering insights into broader cultural trends.

```sql
SELECT
    CASE
        WHEN age < 18 THEN 'Under 18'
        WHEN age BETWEEN 18 AND 24 THEN '18-24'
        WHEN age BETWEEN 25 and 35 THEN '25-35'
        ELSE '35+'
    END AS age_group,
    g.genre,
    COUNT(DISTINCT user_id) as num_users
FROM user_evolution ue
JOIN genre g on ue.genre_id = g.genre_id
GROUP BY age_group, g.genre
ORDER BY age_group, num_users DESC;
```

Query 2

This analysis was kept simple as it uses information from the age of each user to create distinct age groups for analysis. It also uses the recorded data of the favorite genre of each user, that was previously calculated using the sum of weights in the bridge table. This approach allows for a focused examination of how genre preferences evolve across different age groups.

The Queries 3 and 4 delve into the 'anime_stats' star, with distinct focuses on uncovering overlooked titles and examining viewership trends. The Query 3 looks for anime that are less popular (indicated by a popularity rank greater than 50) but received excellent scores (score rank less than 10). This could reveal high-quality anime that might be overlooked due to lack of mainstream attention.

```sql
SELECT
    a.title,
    s.season,
    ast.number_views,
    ast.score
FROM anime_stats ast
join seasons s on s.season_id = ast.season_id
join anime a on a.anime_id = ast.anime_id
WHERE ast.season_popularity_rank > 50 AND ast.score_rank < 10
ORDER BY s.start_date DESC;
```

Query 3

The Query 4 calculates the increase in viewership between consecutive seasons for each anime. By ordering the results in descending order, it highlights the top 10 anime that experienced the most significant growth in viewership from one season to the next.

```sql
SELECT
    a.title,
    s.season,
    ast.number_views,
    ast.season_views,
    ast.number_views - LAG(ast.number_views, 1) OVER (PARTITION BY ast.anime_id ORDER BY
ast.season_id) AS season_view_growth
FROM anime_stats ast
join seasons s on s.season_id = ast.season_id
join anime a on a.anime_id = ast.anime_id
ORDER BY season_view_growth DESC
LIMIT 10;
```

Query 4

These queries demonstrate how your data warehouse design supports diverse analysis. It highlights the value of storing rankings and scores as pre-calculated facts, and effectively utilizes the 'Seasons' time dimension to track trends.

5.2 Data analysis

Having a deep understanding into our dataset, our primary goal is to present concise and captivating summaries of essential points using Power BI, leveraging the knowledge and concepts acquired during *data preparation and visualization* course. It includes the distribution of users' birth years, total user count, anime count, types of anime, gender distribution among users, top 5 producers and licensors, prevalent genres alongside their frequency (leveraging our bridge tables with weights), and the geographical distribution of users. Employing a color-coded map, countries are depicted with green tones indicating higher user concentrations, red for lower concentrations, and oranges for those in between. Notably, the interactivity of the visualization allows users to access exact user counts by clicking on each country.

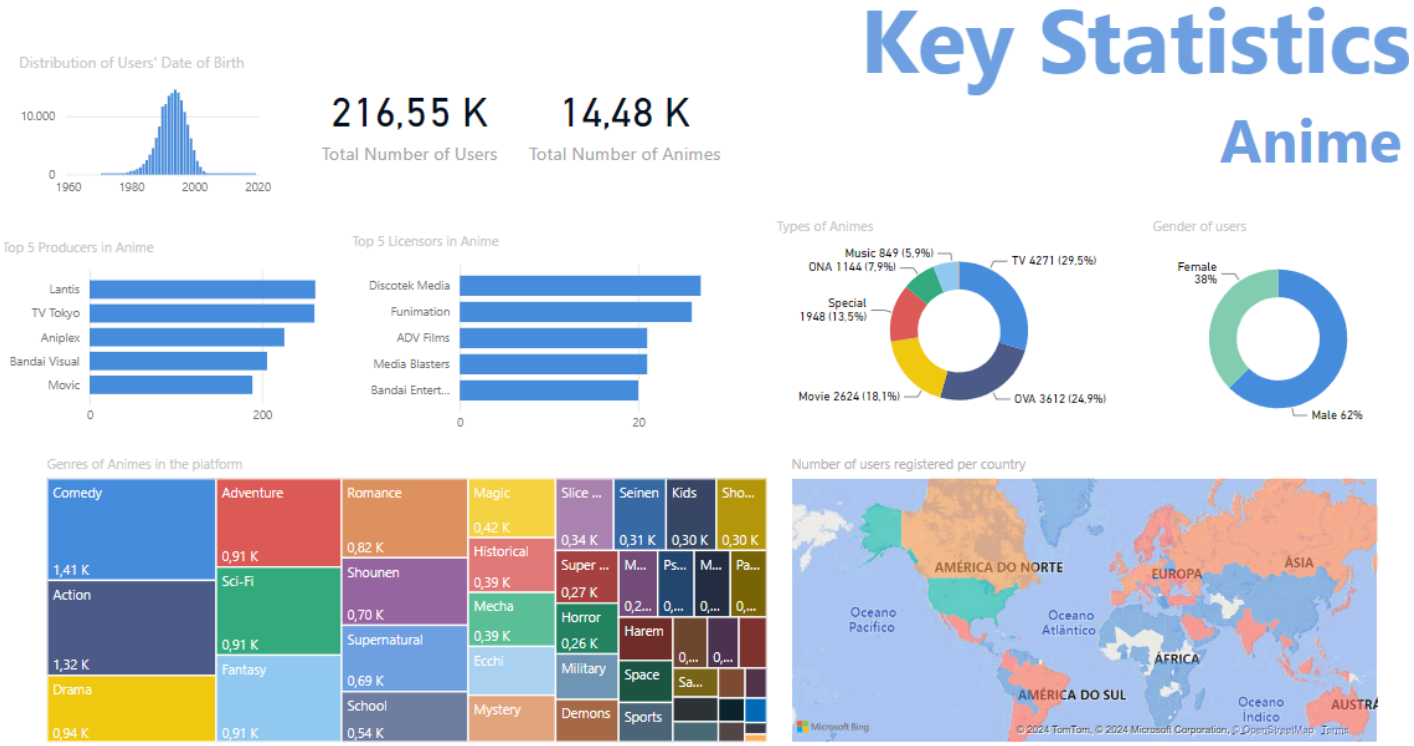


Fig.5 - Visualization in Power BI

We observe a higher representation of men compared to women. Furthermore, comedy and action emerge as the most prevalent types of anime. A majority of these anime belong to the TV category. The United States stands out as the country with the highest number of users.

Additionally, we aimed to gain insight into the most viewed anime and compare its viewership and rating to other anime titles. This analysis provides valuable context by highlighting the popularity and perceived quality of individual anime within our dataset.

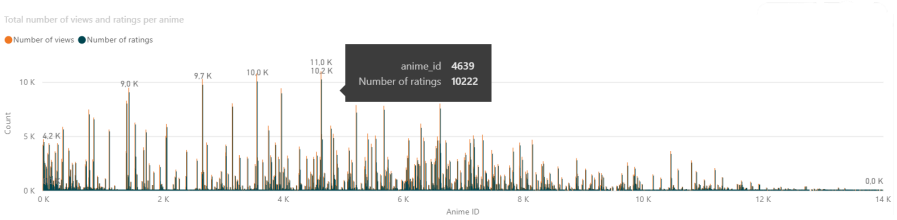
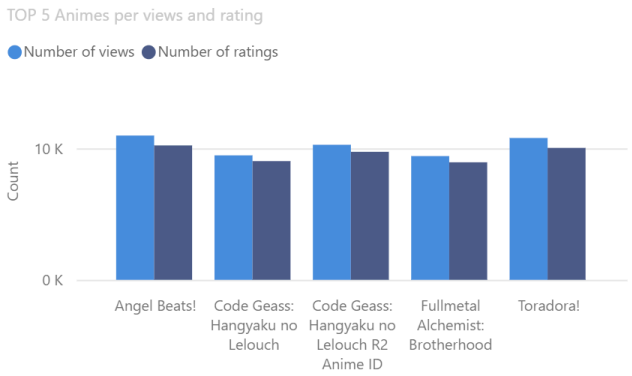


Fig.6 - Total number of views and ratings for all animes_id, in Power BI

However, due to the lack of organization in this visualization, we opted to plot only the top 5 entries. We still see that some anime have really high visualizations but not all of them, as expected. This approach facilitates easier access to both the names and values of the most significant data points.



anime_id, title	Number of views	Number of ratings
4639, Angel Beats!	10972	10222
3567, Toradora!	10787	10033
2659, Code Geass: Hangyaku no Lelouch R2	10270	9736
1432, Code Geass: Hangyaku no Lelouch	9466	9029
3974, Fullmetal Alchemist: Brotherhood	9410	8937

Fig.7 - TOP 5 Animes, in Power BI

6. Critical reflections

The dimensional model designed has several advantages compared to the operational database. First, it focuses on an anime relationship with the user as a product. One of the shortcomings of the operational database is losing past records when the user list is updated. This data warehouse was built in order to also support an operational database that allowed to keep track of each change in the user list along the time. Nevertheless, it has the shortcoming that the dimensional model interaction table could use more storage compared to the operational model interaction table.

The dimensional model also supports analysis from perspectives that would be extremely hard in the operational model. The usage of dimensions and bridge tables for Genre, Studio, Licensor and Produces enable a broad range of analysis with ease. For instance, in our project it is simple to retrieve the favorite genre of a given user, but this would be challenging in the operational database.

Another advantage is how fast it is to study an anime. The Anime Stats allows fast and reliable analysis on the evolution of an anime. This very same analysis on the operational database would take hours and very complex queries. The reason for this is the non continuity in time in the operational database (interactions are pontual actions) and the amount of missing or incorrect information (scores 0 for instance). Furthermore, the addition of a Season dimension promotes natural analysis from a business perspective due to how animes are aired in season intervals.

The dimensional model also allows analysis using the country of the user. This data is very hard to extract from the operational database, since there is no clear structure in how locations are stored in the operational database. However, one of the shortcomings of the dimensional model is that we lost information contained in the operational database location as the data filtering discarded a lot of information like states, cities, etc…

The Data Warehouse comes with some setbacks. The first is how hard it would be to insert data in the dimensional model. The insertion of bridge tables in the design only made it even more challenging. In addition, the dimensional model takes quite a long time (around 1 hour) to create with the current amount of data. With an increase in volume of data, the time to load it could become prohibitive.

7. Conclusion

This project emphasizes the significant effort involved in building a robust data warehouse and its important value for analyzing large datasets. By implementing a data warehouse, we transformed complex data within the anime database into a clear, intuitive structure. This simplifies analysis, particularly for dimensions like studio, genre, licensors, producers, country, and seasons, which were previously difficult to examine.

Additionally, the new structure makes it significantly easier to track changes over time and adapt to future data additions. We gained the ability to conduct granular analysis across years and seasons, a major improvement over the previous day-based limitations.

While teamwork in this context presented challenges, particularly with error correction and version control, a shared data warehouse could streamline collaboration on future projects.

8. Appendix

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
User	Users registered in the platform since it was created	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
User_id	User		PK	Int		
User_id_mal	User in the operational model		UK	Int		
Username	Reference of the user		UK	Int		
Gender	Gender of the user			Text		
Birth_date	Birthdate of the user			Date		
Join_date	Join date of the user in the platform			Date		

Table 5 - User: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Country	Each country	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
Country_id	Country		PK	Int		
Country	Name of the country		UK	Text		

Table 6 - Country: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Years	Years with registrations of users	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
Year_id	Year		PK	Int		
Year_number	Reference of the year		UK	Decimal	4	0

Table 7 - Year: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Days	Time information	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
Date_id	Date		PK	Int		
Day_id	Day		LK	Int		
Day_number	Reference of the day			Int		
Weekday_name	Reference of weekday			Varchar	10	
Month_id	Month surrogate		LK	Int		
Month_number	Reference of the month			Decimal	2	0
Year_id	Year surrogate		LK	Int		
Year_number	Reference of the year			Decimal	4	0

Table 8 - Days: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Genre	Anime's genres	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
Genre_id	Genre		PK	Int		
Genre	Reference of the genre			Varchar	50	

Table 9- Genre: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Licensor	Anime's licensor	type 1	*Hierarchy*			
Attribute	***Description***	***Level***	***Key***	***Type***	***Size***	***Precision***
Licensor_id	Licensor		PK	Int		
Licensor	Reference of the licensor			Varchar	50	

Table 10 - Licensor: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Bridge_licensor	Bridge to aggregate licensors into groups	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Producer_group_id	Licensor group		PK	Int		
Producer_id	Licensor		PK	Int		
Weight	Each licensor holds a weight within the anime licensing group, with these weights summing to one and reflecting their importance			Double		

Table 11 - Bridge Licensor: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Producer	Anime's producer	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Producer_id	Producer		PK	Int		
Producer	Reference of the producer			Varchar	50	

Table 12 - Producer: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Bridge_producer	Bridge to aggregate producers into groups	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Producer_group_id	Producer group		PK	Int		
Producer_id	Producer		PK	Int		
Weight	Each producer holds a weight within the anime production group, with these weights summing to one and reflecting their importance			Double		

Table 13 - Bridge Producer: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Studio	Anime's studio	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Studio_id	Studio		PK	Int		
Studio	Reference of the studio			Varchar	50	

Table 14 - Studio: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Bridge_studio	Bridge to aggregate studios into groups	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Studio_group_id	Studio group		PK	Int		
Studio_id	Studio		PK	Int		
Weight	Each studio holds a weight within the anime production group, with these weights summing to one and reflecting their importance			Double		

Table 15 - Bridge Studio: Dimension

Name	*Description*	*SCD*	*Version*	1.0	*Date*	26/03/2024
Seasons	Season information	type 1	*Hierarchy*			
Attribute	*Description*	*Level*	*Key*	*Type*	*Size*	*Precision*
Season_id	Season		PK	Int		
Season	Reference of the season			Varchar	10	
Start_date	Starting date of the season			Date		
End_date	Ending date of the season			Date	4	0

Table 16 - Seasons: Dimension

Star	User Evolution	Version	1.0	Date	26/03/2024
Granularity	Each user registered in the Anime's Webiste Platform per year of active period				
Dimensions					
User_id	User				
Country_id	Country of registration				
Genre_id	Favorite genre				
Year_id	Year				
Measures					
Age	Current age of the user				
Years_activity	Difference between the date of register in the platform and the date of the last time online in the platform				
Nr_users_per_country	Total n° of users enrolled in the platorm in the country of the user				
Average_users_age_per_country	Average age of all users enrolled in the country of the user				
Average_score_per_user_per_year	Average score of all animes that a user evaluated per year				

Table 17 - User Evolution: Fact Table

Star	Anime Stats	Version	1.0	Date	26/03/2024
Granularity	Seasonal snapshot for a given anime				
Dimensions					
Anime_id	Anime				
Season_id	Season				
Measures					
number_views	Number of times the anime has been fully seen until that season				
number_ratings	Number of ratings of users who have fully seen the anime until that season				
score_rank	Rank of the anime based on the score attribute				
score	Score of the anime considering all user ratings until that season				
season_views	Number of times the anime has been fully seen in that season				
season_ratings	Number of ratings of users who have fully seen the anime in that season				
season_popularity_rank	Rank of the anime based on the season_views_attribute - popularity in that season				

Table 18 - Anime_Stats: Fact Table

References:

[1] MyAnimeList Dataset. (2016). www.kaggle.com. https://www.kaggle.com/datasets/azathoth42/myanimelist

[2] Vaisman, A., & ZimányiE. (2014). Data Warehouse Systems Design and Implementation. Berlin, Heidelberg Springer.

[3] Corr, L., & Stagnitto, J. (2014). Agile data warehouse design : collaborative dimensional modeling, from whiteboard to star schema. Decisionone Press.

[4] Countries Cities States Database. (2024). www.github.com. https://github.com/dr5hn/countries-states-cities-database