University of Milano-Bicocca
School of Science
Department of Informatics, System and Communication
Master degree in Data Science

Data Management

Analisi del mondo anime attraverso AnimeUnity, AnimeWorld e MyAnimeList

Roberto Ferrari [852220], Samir Doghmi [897358], Stefano Quartuccio [851599]

14/06/2023 - academic year 2022-2023

Contents

1 Introduzione

Il progetto si propone di individuare gli anime più popolari in Italia e all'estero e di valutarne le caratteristiche comuni e le differenze. Per anime si intende un film o una serie televisiva d'animazione animata e ispirata allo stile giapponese. Sono destinati a un pubblico di tutte le età e di tutte le fasce sociali, ma generalmente si rivolgono agli shonen, cioè agli adolescenti di età compresa tra i 12 e i 18 anni.

Gli anime si sono affermati in tutto il mondo come fenomeno globale per il realismo e i dettagli delle ambientazioni suggestive dei mondi immaginari creati, che permettono allo spettatore di immergersi in un'esperienza di intrattenimento e di fuga dalla routine ordinaria.

Ciò che li distingue è la versatilità nell'affrontare una vasta gamma di temi, che spaziano dal fantasy al romanticismo passando per argomenti più borderline. Questa eterogeneità rende gli anime un prodotto di consumo accessibile a chiunque. Non da ultimo, l'espressività e la carica emotiva che trasmettono sono riusciti a far diventare gli anime un prodotto di portata mondiale.

1.1 Obiettivi

La popolarità degli anime può variare a seconda del pubblico e della regione geografica. Il nostro obiettivo è identificare eventuali differenze nella popolarità degli anime tra gli utenti globali e quelli italiani, con particolare focus sulla Top 100 per ciascuna fonte. Le domande centrali a cui vogliamo rispondere sono le seguenti:

- un titolo amato all'estero è apprezzato anche in Italia?
- quali sono le caratteristiche che rendono un anime più popolare tra il pubblico?

Occorre però precisare che le due sorgenti italiane, sottoposte ad analisi, presentano un repertorio di contenuti che potrebbero essere considerati discutibili, in quanto non hanno acquisito i necessari diritti di copyright. Tuttavia, abbiamo deciso di utilizzarli dal momento che, essendo piattaforme di streaming gratuite, sono molto visitate. Di conseguenza, possono essere ritenuti sufficientemente rappresentative dei gusti degli utenti italiani. I due siti in discussione sono AnimeUnity e AnimeWorld. Rispettivamente online dall'26 Settembre 2021 e dall'8 Febbraio 2017.
Le possibili spiegazioni di questa tendenza possono essere una questione di natura economica e di convenienza. Gli anime, talvolta, non sono disponibili in alcuni paesi o presentano stagioni su piattaforme differenti, il che fa lievitare il costo dell'abbonamento. Questo può indurre le persone a cercare sorgenti alternative per guardare i loro anime preferiti e questi siti offrono generalmente una maggiore flessibilità. Inoltre, la gran parte dei contenuti viene usufruita in lingua originale con i sottotitoli, rendendo i titoli in streaming più accessibili a un vasto bacino di utenti e riducendo i costi per le piattaforme stesse in termini di traduzione e adattamento.

Per la controparte internazionale, invece, è stato considerato MyAnimeList, un'enciclopedia online e social network di anime e manga gestito da volontari dal 2005. Si tratta di una comunità di appassionati dove è possibile leggere le news riguardante ogni anime, interagire con altri utenti e recensire gli anime. Secondo un report della piattaforma stessa, nel 2020 ha registrato 18 milioni di utenti attivi mensilmente.

1.2 Metriche di valutazione

L'idea di confrontare due piattaforme di streaming con un database online nasce a seguito dell'impossibilità di avere un repertorio italiano che svolga la stessa funzione e sia anche popolare. Le metriche di valutazione utilizzate per confrontare i diversi titoli sono:

- `rating`: si riferisce al punteggio numerico che un utente assegna a un anime che ha guardato. La scala di valutazione va da 1 a 10, con 10 che rappresenta il punteggio più alto. Gli utenti valutano un anime in base al loro gradimento complessivo della serie. Pertanto, gli episodi dello stesso titolo mostreranno punteggi identici.
- `numero di visualizzazioni`: indicano il numero di visite o visualizzazioni della pagina del titolo;

Riguardo a MyAnimeList abbiamo ottenuto tramite scraping l'attributo "Most Popular" che ordina gli anime in base al numero di visualizzazioni da parte degli utenti e l'attributo " Top Ranking" che classifica i titoli in base al punteggio assegnato su una scala da 1 a 10, di cui definiremo la formula nel capitolo successivo.

Per quanto riguarda AnimeUnity, formuliamo una condizione di filtro che considera sia il numero di visualizzazioni sia il punteggio attribuito dagli utenti a ogni anime. Per ottenere gli anime più popolari, estraiamo gli anime in cui le visualizzazioni sono maggiori o uguali alla media delle visualizzazioni per anime, specificando che questi titoli debbano avere anche un punteggio maggiore dello score medio. La formula può essere espressa come segue:

$$(visualizzazioneAnimeUnity \geq 0.5 \cdot \text{AVG}(visualizzazioneAnimeUnity))$$
$$\text{AND}(avg\_score \geq \text{AVG}(avg\_score))$$

Mentre AnimeWorld riporta solo le visualizzazioni e le valutazioni date dagli utenti a ciascun anime. Questo sito produce una classifica chiamata "Top Anime" basata esclusivamente sul numero delle visualizzazioni. Di fronte a questa situazione, abbiamo preferito concentrarci sugli anime con più views per ottenere un confronto coerente con altre sorgenti dati.

L'utilizzo di queste metriche fornisce una panoramica delle valutazioni degli utenti, della popolarità e dell'apprezzamento di un anime. Oltre a individuare eventuali analogie nelle tendenze tra utenti italiani e non, il nostro intento è anche quello di identificare possibili fattori che possono contribuire a far sì che un titolo di anime sia più apprezzato di altri.

1.3 Workflow

Nel processo di sviluppo del nostro database per la valutazione degli anime, sono state seguite diverse fasi.Di seguito elenchiamo le fasi essenziali:

- **Raccolta dei dati**: Il primo passo consiste nella raccolta dei dati relativi agli anime. Sono state considerate diverse fonti, come piattaforme di streaming e database online. Sono state adottate diverse metodologie di acquisizione dei dati, come le API e lo scraping.
- **Preparazione dei dati**: é stato necessario fare qualche operazione di pulizia per rimuovere eventuali errori o informazioni incosistenti. Inoltre alcune covariate come la data di uscita sono state trasformate in un formato standard per tutti e tre i set di dati.
- **Progettazione del database**: è stato scelto un RDBMS database con MySQl. Sono stati definiti, inoltre, tutte le tabelle e che le chiavi interne ed esterne epr per organizzare e gestire in modo efficiente le informazioni riguardandi gli anime.
- **Data cleaning e data quality**: riduciamo le incongruenze tra i dati, gestiamo i valori mancanti e i dati incorretti. Inoltre, verifichiamo la qualità dei dati con metriche come l'accuracy e consistency.
- **Analisi dei risultati**: Una volta realizzato il database, procediamo ad effettuare una serie di interrogazioni per fornire le risposte alle domande formulate nei paragrafi precedenti.

Viene riportato un workflow sintetico che mostra tutti i passaggi che sono stati appena affrontati dalla raccolta dati alla formulazione delle interrogazioni.

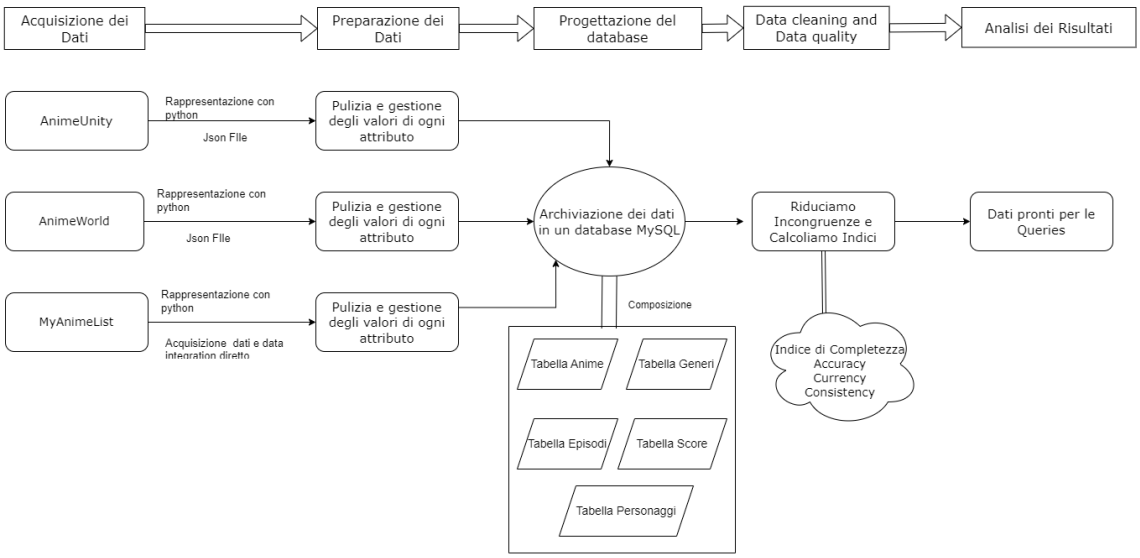

Figure 1: Mappa concettuale fasi progetto

2 Acquisizione dei dati

In questa sezione analizziamo nel dettaglio le tre sorgenti di dati e le informazioni acquisite per ciascuna. Per ogni fonte, elenchiamo le caratteristiche che abbiamo considerato essenziali per raggiungere il nostro obbiettivo.

2.1 AnimeUnity API

Abbiamo utilizzato come pagina di riferimento l'archivio in cui si trovano tutti gli anime presenti nel sito web. L'archivio contiene 4234 risultati che si dividono in serie doppiate (dub ITA) e serie sottotitolate (sub ITA).
Per rendere gli anime distinguibili, consideriamo le covariate Titolo e Dub. In questo modo risulta più semplice la gestione dei duplicati e il confronto tra gli anime.

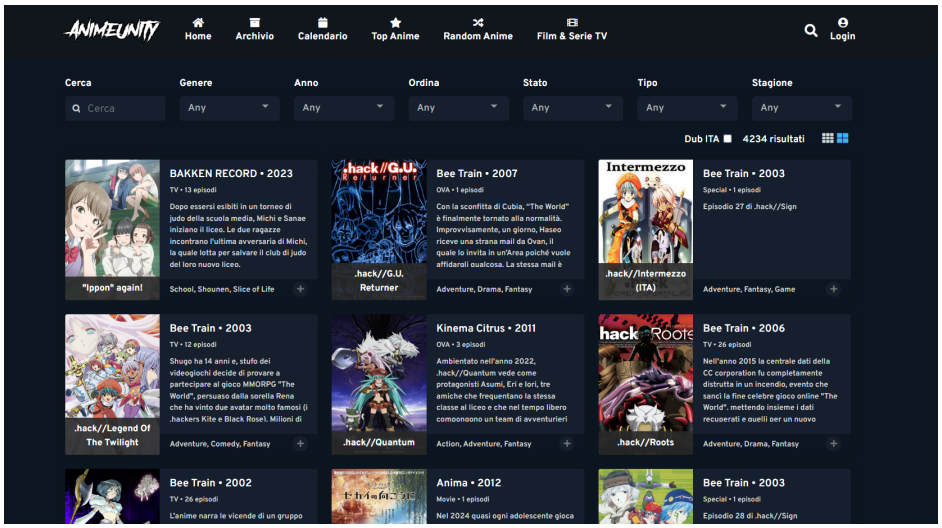

Figure 2: Schermata archivio AnimeUnity

Contattando i responsabili del sito, siamo riusciti a ottenere i token necessari per poter utilizzare l'Api come metodo di acquisizione dei dati. Come anticipato, la nostra attenzione si è rivolta alla sezione archivio, riportata nella figura qui sopra, in cui sono memorizzati tutti i titoli di nostro interesse. La disposizione degli elementi non è cruciale, poichè L'Api permette di accedere ai dati in modo strutturato e affidabile. È stato quindi inizializzata una lista vuota, chiamata result. Successivamente è stato definito un ciclo for che esegua 129 iterazioni partendo dall'elemento 0 e incrementando di 30 ad ogni passo fino a ottenere l'ultimo elemento dell'archivio.
A questo punto, il contenuto della nostra richiesta viene convertito in for-

mato Json utilizzando la funzione dumps. Di questi 4234 titoli, ne sono stati presi in considerazione 3802, poiché 428 presentavano dati incompleti.

Le proprietà considerate di ogni anime sono:

- `Id`: è un valore unico assegnato a ciascun anime per identificarlo in modo univoco;
- `Titolo`: il titolo dell'anime rappresenta il nome dell'opera;
- `Stagione`: la stagione si riferisce alla stagione in cui l'anime è stato trasmesso per la prima volta;
- `Tipo di anime`: si riferisce al formato o categoria dell'anime che può essere TV, Special, Ova, Ona o Film;
- `Data di uscita`: la data di uscita si riferisce all'anno e stagione in cui l'anime è stato trasmesso per la prima volta;
- `Numero di episodi`: si riferisce al numero totale di episodi dell'anime;
- `Studio`: indica la casa di produzione che ha realizzato l'anime;
- `Visualizzazioni`: si riferisce al numero totale di visualizzazioni dell'anime;
- `Plot`: il plot rappresenta una breve descrizione della trama dell'anime;
- `Score`: è rappresentato dalla combinazione degli attributi favoriti e visualizzazioni. Lo score rappresenta anche il ranking stesso degli anime;
- `Favoriti`: il numero di favoriti si riferisce al numero di utenti che hanno aggiunto l'anime ai propri preferiti;
- `Dub`: si tratta di una variabile binaria in cui 0 indica che l'anime ha solo una versione sottotitolata in italiano, mentre un valore di 1 indica che l'anime è stato doppiato;
- `LinkAnimeUnity`: si riferisce al link di ogni episodio;
- `visiteAnimeUnity`: corrisponde al numero di visualizzazioni per ciascun episodio di ogni anime.

2.2 AnimeWorld Scraping

Per quanto riguarda invece la seconda fonte italiana abbiamo implementato la tecnica dello scraping, quindi di recupero di dati 'grezzi' direttamente dalla pagina web: a questo scopo abbiamo utilizzato come web scraper la piattaforma *Zyte*, leader nel servizio di web scraping, per ricavare le informazione di nostro interesse dal sito web di AnimeWorld.

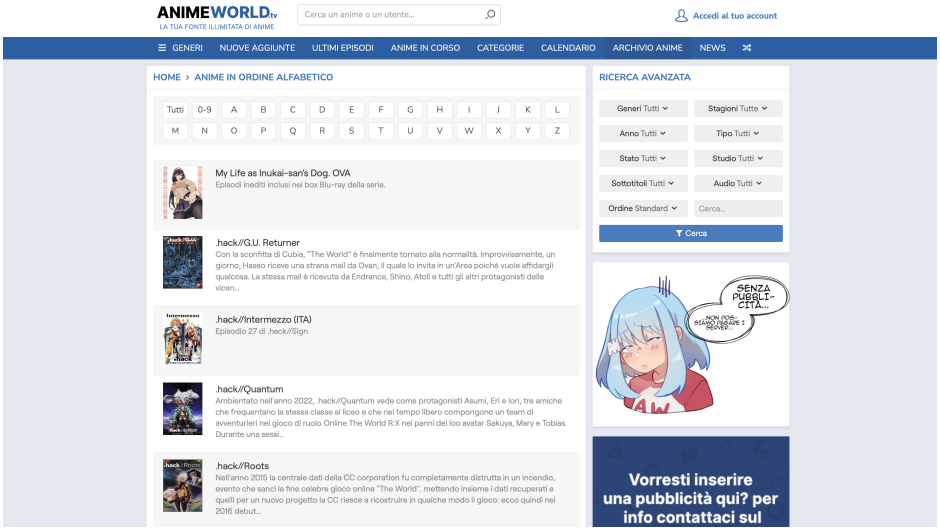

Figure 3: Schermata archivio AnimeWorld

L'operazione di scraping è stata effettuata, anche in questo caso, prendendo come punto di riferimento la sezione "Archivio Anime" presente nella pagina, che raccoglie tutti i titoli presenti in ordine alfabetico a gruppi di 25, esclusa l'ultima pagina, per 177 pagine, contando un totale di più di 4400 titoli. Va tenuto in considerazione, tuttavia, che per alcune serie è disponibile sia la serie sottotitolata che quella doppiata in lingua italiana, perciò il numero reale di titoli è inferiore a quello nominale.

Per ottenere le informazioni di nostro interesse abbiamo sfruttato il pacchetto *scrapy* di Python e, nello specifico, della classe *Spiders* che permette di definire in che modo verranno estratti i dati da un sito web, ovvero la modalità di scraping. Infatti per ogni classe di *spider* è possibile determinare diverse funzioni finalizzate allo scraping, come il nome dello *spider*, i domain autorizzati e l'url da cui partire per l'estrazione.

Così facendo si è stabilito come link di partenza per lo scraper la pagina dell'archivio anime e, attraverso cicli successivi, abbiamo ricavato tutte le serie presenti nell'archivio.

```python
# -*- coding: utf-8 -*-
from typing import Any, Union

import scrapy

class ListAnimeWorld(scrapy.Spider):
    name = "listAnimeWorld"
    allowed_domains = ["www.animeworld.tv"]
    start_urls = [
        'https://www.animeworld.tv/az-list',
    ]

    def parse(self, response):
        for anime_link in response.css(".items .item a ::attr(href)").extract():
            yield scrapy.Request(response.urljoin(anime_link), callback=self.parse_list_anime_page)

        next_page = response.css("#go-next-page.disabled").extract_first()
        if not next_page:
            next_page = response.css("#go-next-page ::attr(href)").extract_first()
            yield scrapy.Request(response.urljoin(next_page), callback=self.parse)

    def parse_link_url(self, response):
        p = response.css("#player video source").get()
        return p
```

Figure 4: Script per l'estrazione dei dati dalle pagine dell'archivio

Per ciascuno di questi titoli è possibile, poi, ottenere informazioni più dettagliate attraverso un'apposita pagina, che contiene voci come *categoria*, *data di uscita*, *studio*, *voto*, *visualizzazioni*, oltre che la descrizione della serie.

Figure 5: Esempio di informazioni fornite per ciascuna serie

Il passo successivo, perciò, è stato quello di definire nello *spider* l'analisi di queste pagine dedicate a ciascun anime, stabilendo tutti gli *item* da estrarre, ovvero le proprietà sopracitate, oltre che la lista degli episodi.

Le proprietà tenute in considerazione in seguito all'operazione di scraping sono le seguenti:

- `Data_uscita`: fa riferimento alla data in cui l'anime è stato trasmesso per la prima volta, riportando giorno, mese e anno;
- `MyAnimeList`: proprietà che indica il link della serie ottenuta dal sito web della fonte mondiale *MyAnimeList* allo scopo di assicurare l'identità tra gli anime, utile soprattutto nella successiva fase di *querying*;
- `Stagione`: la stagione si riferisce alla stagione in cui l'anime è stato trasmesso per la prima volta;
- `Categoria`: si riferisce al formato dell'anime che può essere TV, Special, Ova, Ona o Film;
- `Rating`: esprime la valutazione media dell'anime assegnata dagli utenti;
- `Visualizzazioni`: si riferisce al numero totale di visualizzazioni dell'anime;
- `Descrizione`: questa proprietà descrive la trama principale dell'anime;
- `Titolo`: indica il titolo della serie.

2.3 MyAnimeList Scraping

Infine, così come per AnimeWorld, anche le informazioni relative alla fonte mondiale, MyAnimeList, sono state ricavate tramite un'operazione di *scraping*, e anche questa fonte presenta una sezione che archivia tutte le serie anime, facilitando così le operazioni di estrazione.

A differenza del precedente script relativo all'acquisizione dei dati da *AnimeWorld*, in questo caso si è optato per un'operazione congiunta di acquisizione e successivo immagazzinamento dei dati nel database, attraverso la libreria *mysql.connector* di Python che permette, tramite un'API rilasciata da MySql, di comunicare direttamente con database di tipo MySql. Per quanto riguarda le operazioni di acquisizione, ancora una volta si è utilizzato il pacchetto *scrapy* di Python data la flessibilità dello strumento *spider*, specialmente nel poter definire ad hoc le caratteristiche dello scraping.

In questo modo, perciò, abbiamo ricavato le informazioni di nostro interesse partendo dalla sezione all'interno di MyAnimeList che archivia, in maniera analoga alle precedenti fonti, le serie anime: così come per lo script di AnimeWorld, questa sezione di archivio ha rappresentato il link di partenza dello *spider*; tuttavia durante questa fase di acquisizione non si presenterà la problematica della 'doppia' serie per ciascun anime, non essendo presente il doppiaggio italiano.

Successivamente si sono definti gli attributi da acquisire durante la fase di *parsing* dell'archivio: accanto ad attributi che abbiamo ottenuto in precedenza come il `titolo`, il `genere`, il `punteggio`, o i `favoriti`, la fonte mondiale *MyAnimeList* offre un livello di informazione ulteriore che abbiamo ritenuto opportuno sfruttare per arricchire maggiormente i dati a nostra disposizione.

A questo scopo sono stati estratti attributi che indicano i texttt nomi dei `personaggi` e i `loro ruoli`, che possono essere *main* o *supporting*, oltre che attributi come `ranked`, il quale descrive il posizionamento di un titolo all'interno della community di MyAnimeList, ottenuto attraverso il confronto tra i punteggi assegnati alle serie secondo questa formula:

$$WeightedScore = \frac{v}{(v+m)} * S + \frac{m}{(v+m)} * C$$

dove:

- S = Punteggio medio assegnato all'anime;
- v = Numero di utenti che ha valutato un dato anime, a condizione che abbiano visionato almeno 1/5 della serie, altrimenti il punteggio assegnato non viene conteggiato;
- m = Numero minimo di utenti che devono valutare affinchè venga calcolato il punteggio;
- C = Punteggio medio attraverso l'intero database anime.

Altri attributi che hanno arricchito ulteriormente il nostro database sono la `popularity`, che indica, appunto, la popolarità della serie, sulla base del numero di persone che l'ha vista, la sta guardando o è in lista, e `demographic`, il quale offre un dettaglio informativo ulteriore, poichè separa il dato che esprime unicamente il genere del titolo, quindi ad esempio *azione*, *avventura*, dal dato che indica in modo più specifico il target verso cui è maggiormente indirizzata la serie.

3 Database

3.1 Scelta del Database

La natura dei nostri dati è piuttosto semplice: ciascun anime è rappresentato come entità indipendente e associato a diverse caratteristiche come il tipo, lo studio di produzione e il numero di visualizzazioni. Per facilitare la comprensione delle relazioni tra di essi, riteniamo che il modello relazionale possa rappresentare la scelta ideale.

Il modello relazionale ci permette di organizzare i dati in tabelle, in cui ogni tabella corrisponde a una determinata relazione. Questa impostazione ci permette di mantenere una struttura coerente e organizzata, rendendo più intuitive le associazioni tra i dati.

Inoltre, disponiamo di informazioni chiare sul significato di ciascun attributo e riteniamo che questi dati non cambieranno significativamente nel tempo. Nonostante la rigidità dello schema possa essere un limite e la prospettive di aggiungere nuove istanze possa minare il formato scelto, ipotizziamo che le modifiche future saranno molto minime. Di fatto, la frequenza settimanale di rilascio degli episodi, se si considera che una singola stagione di un anime conta in media 12 episodi, non rende necessario dotarsi di un database che raccolga, elabori e memorizzi i dati in tempo reale.

Un esempio di attributo che abbiamo trovato in MyAnimeList , ma che non era presente negli altri due siti di streaming sono i dati demografici che indicano il target di utenti alla quale è consigliata la visione di questi anime. L'utilizzo di un database più complesso per archiviare dati di questa tipologia potrebbe richiedere un sforzo eccessivo che sovrastima l'effettiva necessità e scopo del nostro database. Mentre i Database NoSQL tendono a dare la priorità alla tolleranza delle partizioni. Al contrario, noi siamo più interessati a mantenere i dati consistenti. Pertanto, abbiamo definito dei vincoli di integrità come l'utilizzo di chiavi esterne che garantiscono le relazioni tra le tabelle e le chiavi interne assicurano l'unicità di ogni riga inserita. In questo modo è possibile fare joints multi-tabella e ottenere risultati complessi, rendendo più efficiente l'estrazione di informazioni dettagliate in tempi brevi.

Tuttavia, molti database RDBMS possono soffrire di gravi problemi di prestazioni all'aumentare delle loro dimensioni, ma dal momento che abbiamo un set di dati 60MB che raccoglie gli anime dal 1966 ai primi inizi del 2023 e che in media vengono aggiunti 200 titoli all'anno, non prevediamo particolari cali di prestazioni.

Tra i sistemi di database disponibili, abbiamo deciso di affidarci a MySQL. Si tratta di un database relazionale open source, supportato da una vasta comunità e con una solida reputazione. Supporta anche l'utilizzo di SQL come linguaggio di interrogazione offrendo numerosi vantaggi. Trattandosi di un linguaggio di alto livello, presenta una sintassi immediata che lo rende accessibile anche ad utenti meno esperti.

3.2 Modellazione Database

Le istanze principali da gestire riguardano gli anime stessi, i personaggi, gli episodi, gli score e i generi per ogni titolo disponibile. L'approccio alla definizione di 5 tabelle si basa sul presupposto di avere molteplici valori per ogni istanza all'interno di ogni anime.

Considerando la tabella `Anime`, è stato necessario identificare una coppia di chiavi primarie, nel nostro caso "anime id" e "dub", per riconoscere in modo univoco ogni tupla nel database. Oltre alle principali informazioni, sono state aggiunte anche due covariate corrispondenti alla versione italiana e originale del titolo. C'è da dire che queste due colonne presentano molti valori identici, simili o che differiscono tra loro. La gestione di queste due istanze verrà approfondita nel capitolo successivo.

Le chiavi primarie appena definite nella tabella `Anime`, sono fondamentali come chiave esterne in quanto rappresentano il vincolo relazionale con le altre tabelle. Pertanto, la tabella `Personaggi` permette di trattare le relazioni tra i personaggi e gli anime. Ogni riga nella tabella rappresenta un personaggio specifico e la chiave esterna "anime" lega il personaggio all'anime a cui appartiene. Questo schema semplifica l'associazione dei personaggi ai rispettivi anime e consente di eseguire interrogazioni specifiche sui personaggi.

La tabella `Episodi` include i dettagli relativi agli episodi come i numeri degli episodi, i link per lo streaming con riferimento alle due piattaforme di riproduzione video e le visualizzazioni per ogni episodio. Quest'ultima informazione è stata reperibile solamente per AnimeUnity. Inoltre, ciascuna riga rappresenta un episodio specifico e le chiavi esterne dub e id anime consentono di associarlo alla versione e all'anime corrispondente.

Viceversa, la tabella `Score` raccoglie le valutazioni degli utenti per tutte e tre le fonti, mentre la tabella `Genere` contiene tutti i generi per ogni titolo. Viene seguito lo stesso ragionamento effettuato precedentemente per poter identificare ogni anime in modo univoco.

Nel complesso, La struttura adottata per il nostro database permette di semplificare le funzioni di interrogazione, offrendo una grande flessibilità. La struttura corrente ci permette di aggiungere facilmente nuovi record nelle rispettive tabelle senza dover modificare l'intero schema.

Viene riportato un diagramma riassuntivo che mostra le istanze di ciascuna tabella che compongono la struttura del database appena descritto:

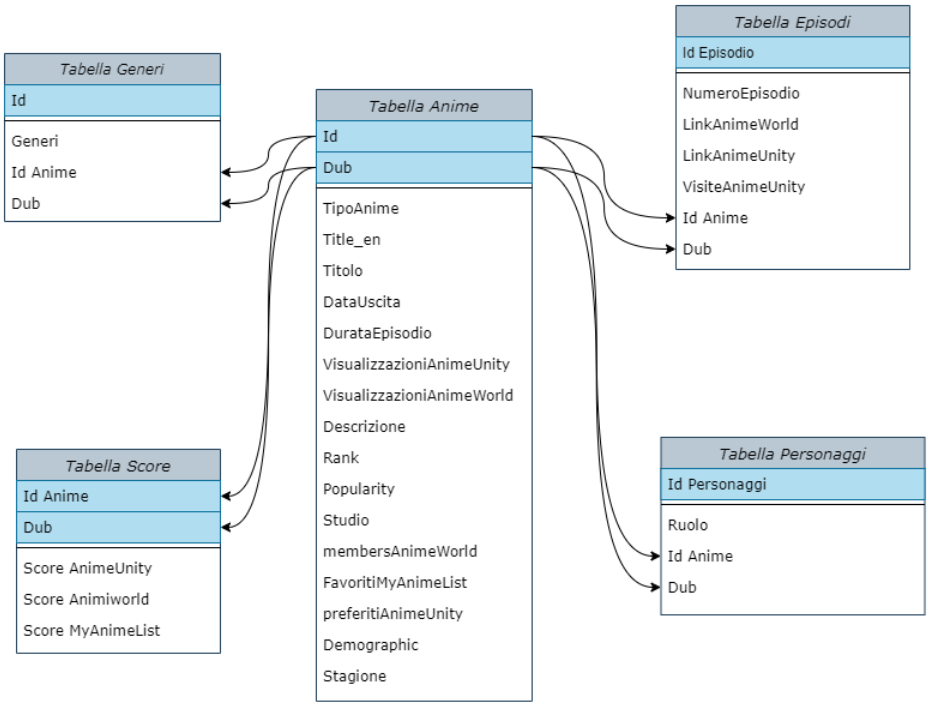

Figure 6: Le caselle con lo sfondo in blu celeste indicano le chiavi primarie mentre le freccie si riferiscono alle chiavi esterne

4 Data cleaning and Data integration

4.1 Data Cleaning

La pulizia dei dati è un processo che mira ad identificare e correggere gli errori, le inconsistenze e le anomalie presenti nei dati raccolti. Gli errori possono derivare da diverse fonti, tra cui errori di input umano, errori di misurazione o problemi tecnici, e da ciò derivano una serie di attività, come la rimozione di valori mancanti, la risoluzione di inconsistenze e duplicati, l'eliminazione di outliers e la standardizzazione dei formati. Queste attività richiedono l'applicazione di algoritmi e regole specifiche, nonché l'uso di strumenti software adeguati per automatizzare il processo e ridurre al minimo gli errori umani.

Nel nostro caso, considerando che abbiamo applicato complessivamente le stesse operazioni per tutte e tre le fonti, di seguito descriviamo le operazioni di pulizia dei dati specificatamente eseguite per il sito streaming AnimeWorld, che si è rivelato essere la sorgente che ci ha impegnato più tempo. Di seguito riportiamo alcuni esempi:

- Nel sito non era compresa la colonna Dub; pertanto, per stabilire se l'anime fosse in lingua italiana, abbiamo usato il titolo come parametro di riferimento.

  ```
  def contains_ita(string):
     return 1 if "(ITA)" in string else 0
  ```

  Dal codice descritto, verifichiamo se la stringa contiene il testo "(ITA)". Se la condizione viene soddisfatta, viene restituito il valore 1, altrimenti viene visualizzato il valore 0. Il risultato viene memorizzato come dato binario nella colonna `dub`.

- Abbiamo pensato che fosse interessante poter fare delle query considerando separatamente la stagione e l'anno. A questo scopo, con la seguente finestra di codice, abbiamo definito l'espressione regolare da estrarre. Con la funzione re.search, ogni volta che viene trovata una corrispondenza tra la colonna `AnnoUscita` e l'espressione appena definita, il valore risultante viene assegnato alla covariata `Anno`.

```
regex = r"<dd>\d+\s+\w+\s+(\d+)</dd>"
annoUscita = anime["Data_uscita"]
match = re.search(regex, annoUscita)
anno = 0
if match:
anno = match.group(1)
```

- Se il numero è considerato di tipo float piuttosto che intero, il separatore delle migliaia può creare non pochi problemi. Per evitare questo conflitto, abbiamo rimosso il simbolo del punto sostituendolo con una stringa vuota e convertendo il valore risultante in un numero intero.

```
int(visualizzazioni.replace(".","")
```

- In AnimeWord, non era presente alcuna istanza che indicasse l'id di ogni anime. Tuttavia, esaminando l'insieme delle cifre che seguono /anime/ all'interno dell'URL di ciascun titolo, e confrontando l'id visualizzato in MyAnimeList, è stato possibile verificare che la composizione alfanumerica si riferiva all'id dell'anime. Dunque, con la funzione strip e split, abbiamo ripulito la selezione da spazi vuoti ed estratto il primo elenco di elementi che seguivano la dicitura (/anime/).Il valore ottenuto è stato assegnato alla colonna `IdAnime`.

```
idAnime = anime["MyAnimeList"]
idAnime = idAnime.split("/anime/",1)[1].strip()
```

4.2 Data Integration

L'integrazione dei dati, d'altra parte, riguarda la combinazione di dati provenienti da diverse fonti, che possono essere eterogenee per natura, struttura e formato. L'obiettivo dell'integrazione dei dati è creare una vista coerente e unificata dell'informazione, in modo da consentire l'analisi e l'elaborazione dei dati provenienti da più origini. Durante il processo di integrazione, è necessario affrontare sfide come la risoluzione dei conflitti tra i valori dei dati, l'allineamento dei formati e degli schemi, nonché la gestione della ridondanza delle informazioni.

Dato che abbiamo applicato le stesse operazioni a tutte e tre le fonti, riportiamo di seguito le operazioni di controllo effettuate per il sito di streaming AnimeWorld:

- Per ogni titolo nel dataset viene recuperato l'`idAnime`. Il primo controllo che effettuiamo è verificare se l'id è nullo, stampando un riga vuota. Se l'`IdAnime` è presente, vengono eseguite operazioni di aggiornamento dei dati. Questi includono l'aggiunta di visualizzazioni, punteggi, descrizioni e altri attributi all'anime esistente. Diversamente, se l'id dell'anime risulta assente, viene effettuata un'operazione di inserimento, creando un nuovo record nella tabella.

```
for anime in dataset["records"]:
    idAnime = anime["MyAnimeList"]

    if(idAnime is None):
        print("")
    else:
        idAnime = idAnime.split("/anime/",1)[1].strip()
        print(idAnime)
        visualizzazioni = anime["Visualizzazioni"]
        scoreAnimeWorld = anime["Rating"]
        titolo = anime["Title"]
        dub = str(contains_ita(titolo))
        descrizione = anime["Description"]
        stagione = anime["Stagione"]
        annoUscita = anime["Data_uscita"]
```

- Nel codice riportato, nel caso in cui l'anime non è contenuto nella lista "records" , viene eseguita un'operazione di inserimento dei valori quali `Stagione`, `DataUscita` e `descrizione` nella tabella `Anime`.

Al contrario, se l'anime è contenuto, viene eseguita un 'operazione di aggiornamento vincolato attraverso i valori `IdAnime`, `dub` e `titolo`. In questo modo vengono aggiornati solo i records che soddisfano i vincoli appena definiti.

```
if(records == []):
    sql = "INSERT INTO anime (id, dub, titolo,
        stagione, dataUscita,
        visualizzazioneAnimeWorld, descrizione) VALUES
        (%s, %s, %s, %s, %s, %s, %s)"
    val = (idAnime, dub, titolo, stagione, anno,
        int(visualizzazioni.replace(".","")),
        descrizione)
    cursor.execute(sql, val)
else:
    sql = "UPDATE anime SET visualizzazioneAnimeWorld
        = %s WHERE id = %s and dub = %s"
    val = (int(visualizzazioni.replace(".","")),
        idAnime, dub)
    cursor.execute(sql, val)

     sql = "UPDATE anime SET visualizzazioneAnimeWorld
         = %s, titolo=%s WHERE id = %s and dub = %s and
         titolo IS NULL"
     val = (int(visualizzazioni.replace(".","")),
         titolo, idAnime, dub)
    cursor.execute(sql, val)
```

- In questa sezione di codice viene riproposta la stessa procedura descritta in precedenza, con la differenza che le operazioni di aggiornamento e inserimento vengono eseguite per la tabella Episodi. I valori sono aggiornati vincolando le operazioni all' `IdAime` , al valore `dub` e al numero dell'episodio. Viene inoltre aggiunto l'attributo linkAnimeWorld.

```
for episodios in anime["Link-ep"]:
    for episodio in episodios:
        linkAnime = episodios[episodio]
        cursor.execute("select * from episodi where
            idAnime = " + str(idAnime) + " AND dub = " +
            dub + " AND numeroEpisodio = " + episodio)
        records = cursor.fetchall()
```

```
        if(records == []):
            sql = "INSERT INTO episodi (numeroEpisodio,
                idAnime, dub, linkAnimeWorld) VALUES (%s,
                %s, %s, %s)"
            val = (episodio, int(idAnime), dub,
                linkAnime)
            cursor.execute(sql, val)
        else:
            sql = "UPDATE episodi SET linkAnimeWorld =
                %s WHERE idAnime = %s and dub = %s and
                numeroEpisodio=%s"
            val = (linkAnime, int(idAnime), dub,
                episodio)
            cursor.execute(sql, val)
```

- Nella tabella `Score` viene adottata la stessa procedura descritta in alto. L'operazione di aggiornamento viene eseguita utilizzando come vincolo la combinazione dei valori `Idanime` e `Dub`, che consente di identificare in modo univoco ogni titolo della tabella.

```
cursor.execute("select * from score WHERE id = " +
    str(idAnime) + " and dub = " + dub)
records = cursor.fetchall()

if(records == []):
    sql = "INSERT INTO score (id, dub, animeworld)
        VALUES (%s, %s, %s)"
    val = (idAnime, dub, scoreAnimeWorld)
    cursor.execute(sql, val)
else:
    sql = "UPDATE score SET animeworld = %s WHERE id =
        %s AND dub = %s"
    val = (scoreAnimeWorld, idAnime, dub)
    cursor.execute(sql, val)
```

Come risultato dell'integrazione dei dati, per ogni attributo riportiamo le fonti utilizzate:

Attributi	Fonti
Id	AnimeUnity, AnimeWorld, MyAnimeList
Dub	AnimeUnity, AnimeWorld
TipoAnime	AnimeUnity
Title_en	MyAnimeList
Titolo	AnimeUnity, AnimeWorld
DataUscita	AnimeUnity, AnimeWorld
DurataEpisodio	AnimeUnity
VisualizzazioniAnimeUnity	AnimeUnity
VisualizzazioniAnimeWorld	VisualizzazioniAnimeWorld
Descrizione	AnimeUnity, AnimeWorld
Rank	MyAnimeList
Popularity	MyAnimeList
Studio	AnimeUnity
membersAnimeWorld	MyAnimeList
FavoritiMyAnimeList	MyAnimeList
preferitiAnimeUnity	AnimeUnity
Generi	MyAnimeList
NumeroEpisodio	AnimeUnity
LinkAnimeWorld	AnimeWorld
LinkAnimeUnity	AnimeUnity
visiteAnimeUnity	visiteAnimeUnity
Ruolo	MyAnimeList
Score AnimeUnity	AnimeUnity
Score AnimeWorld	AnimeWorld
Score MyAnimeList	MyAnimeList
Stagione	AnimeUnity, AnimeWorld

5 Data Quality

Le metriche prese in considerazione per valutare la data quality sono: *accuracy*, *completeness*, *currency* e *consistency*.

5.1 Indice di Accuracy

Secondo la definizione di *accuracy*, un dato è accurato quando rappresenta correttamente un fenomeno della realtà.
Nel nostro caso le fonti dalle quali abbiamo fatto riferimento non si possono definire ufficiali, quindi le informazioni disponibili potrebbero essere soggette ad imprecisioni; tuttavia fonti più attendibili, che potrebbero essere rappresentate dai siti web ufficiali dei diversi studios, non possiedono tutte quelle informazioni legate all'apprezzamento delle serie, come punteggi, visualizzazioni, membri, ma costituiscono una semplice "vetrina" attraverso la quale annunciano le loro produzioni.
Tuttavia, il titolo riportato è stato talvolta adattato alla versione inglese e italiana o mantenuto nella lingua originale. Poiché non sapevamo quale covariata considerare, abbiamo mantenuto entrambi i titoli, archiviati così nelle colonne `title_en` e `titolo`.
Successivamente, abbiamo utilizzato la funzione SequenceMatcher per calcolare la corrispondenza tra le due colonne. L'indice di similarità ricavato è pari all'86,82%.
Di seguito, viene mostrato il confronto tra alcune righe del DataFrame.

	dub	title_en	titolo	similarity_index
2616	0	One Punch Man 2nd Season	One Punch Man 2	76.923077
4098	0	Lupin III: Part 6	LUPIN THE 3rd PART 6	32.432432
4334	0	My Home Hero	My Home Hero	100.000000
2668	0	Kekkai Sensen & Beyond	Blood Blockade Battlefront & Beyond	49.122807
540	1	Bakuten Shoot Beyblade 2002	Bakuten Shoot Beyblade 2002 (ITA)	90.000000
2187	0	ChÃ¤oS;Child	Chaos;Child	78.260870
3521	0	Fairy Gone Part 2	Fairy Gone 2	82.758621
3361	1	Koukaku Kidoutai: SAC_2045	Ghost in the Shell: Stand Alone Complex 2045 (...	34.210526
2236	0	Akagami no Shirayuki-hime 2nd Season	Akagami no Shirayuki-hime 2	85.714286
782	0	Naisho no Tsubomi	Naisho no Tsubomi	100.000000
3798	0	Majutsushi Orphen Hagure Tabi: Kimluck-hen	Sorcerous Stabber Orphen 2	29.411765
224	0	Shiritsu Araiso Koutougakkou Seitokai Shikkoubu	Araiso Private High School Student Council Exe...	23.853211

5.2 Indice di Completeness

Si tratta di una misura della corrispondenza tra il mondo reale e il set di dati specifico. Rappresenta la percentuale di dati presenti rispetto a quelli attesi o desiderati. Questo indice viene calcolato a livello di singolo attributo all'interno di ciascuna tabella.

- Per la tabella `Anime`, su un totale di 3927 righe:

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
Stagione	10	0,25
TipoAnime	57	1,45
`Title_en`	4	9,14
Titolo	78	1,78
DataUscita	42	1,07
NumeroEpisodi	1.540	39,22
VisualizzazioneAnimeUnity	0	0,00
VisualizzazioneAnimeWorld	0	0,00
Descrizione	0	0,00
Rank	0	0,00
Popularity	4	0,09
Studio	1.532	34,96
Members	4	0,09
Favoriti	25	0,63
PreferitiAnimeUnity	1.575	40,11
Demographic	2708	68,96

Table 1: Tabella Anime

  La percentuale di completezza è pari al 89,27%. Ciò significa che il 10.72% delle righe presenta valori mancanti.

- Per quanto riguarda la tabella `episodi` il numero delle righe è 67.307 e la percentuale di completezza è dell'89,8%.

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
NumeroEpisodio	0	0,00
Id_Anime	0	0,00
LinkAnimeWorld	6.058	9.14
LinkAnimeUnity	20.208	30.50
VisiteAnimeUnity	20.208	30.50

Table 2: Tabella Episodi

il valore considerevole dei missing value è dovuto dalla presenza della coviarata dub; Questa circostanza si verifica con i titoli archiviati su AnimeUnity che presentano unicamente la versione in lingua Italiana: in tal modo, viene generato il doppio delle righe corrispondenti a un determinato numero di episodi. In queste condizioni, il numero di valori Nan per gli attributi LinkAnimeUnity e VisitAnimeUnity è uguale.

- Per la tabella `genere`, su un totale di 18.552 righe, la percentuale di completezza è del 100%.

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
Genere	0	0,00
Id_Anime	0	0,00

Table 3: Tabella Genere

- Situazione identica per la tabella `Personaggi`, composta da 89.990 righe, in cui il coefficiente di completezza è del 100%.

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
Ruolo	0	0,00
Id_Anime	0	0,00

Table 4: Tabella Personaggi

- Per quanto riguarda la tabella `Score`, costituita da 4.519 tuple, il tasso di completezza si attesta al 90,38%. Per l'attributo LinkAnime-Unity, riscontriamo la stessa situazione della tabella Episodi.

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
AnimeWorld	451	9,98
AnimeUnity	1.695	37,51
MyAnimeList	27	0,60

Table 5: Tabella Score

5.3 Indice di Currency

L'indice di *currency* misura la frequenza con la quale i dati sono aggiornati: nel nostro caso i dati cambiano continuamente, sia in termini di nuove serie anime che vengono prodotte, che in termini di nuovi episodi relativi a serie in corso, o ancora serie che vengono doppiate in lingua italiana.

Per assicurare la *currency*, il nostro database dovrebbe essere aggiornato almeno settimanalmente, al fine di ottenere l'aggiunta di nuovi episodi che, tendenzialmente, si verifica ogni settimana per le serie in corso. Tuttavia le fonti utilizzate non sono ufficiali, quindi la frequenza con la quale i dati vengono aggiornati è soggetta a cambiamenti e non sempre è costante.

L'operazione di aggiornamento del database si è interrotta dopo qualche settimana di scraping, con lo scopo di iniziare la fase di analisi dei risultati.

5.4 Indice di Consistency

Questo indice mostra la consistenza di diverse rappresentazioni dello stesso oggetto della realtà nel nostro database. Nel nostro caso abbiamo verificato che gli attributi estratti, che presentano il medesimo significato, e successivamente integrati nel database, fossero rappresentati dallo stesso formato di dato.

Nello specifico abbiamo verificato la consistenza di alcuni attributi più significativi della tabella `Anime`, constatando che la tipologia di dato per ciascun

attributo si è rivelata piuttosto coerente e costante tra le diverse fonti, ad eccezione dell'attributo che fa riferimento alla data di uscita della serie anime: in questo caso la differenza tra le fonti AnimeUnity e AnimeWorld risiede nel fatto che per la prima si è estratto direttamente l'anno, immediatamente disponibile come `integer`, mentre per la seconda si è dovuto effettuare una pulizia del dato, disponibile nel formato `string`, eliminando il giorno e il mese in quanto giudicati superflui per il nostro scopo.

Per questioni di semplicità, nella tabella sottostante si sono abbreviati i termini che indicano le diverse fonti: "AnimeU" fa riferimento a AnimeUnity, "AnimeW" si riferisce a AnimeWorld e infine "MAL" è l'acronimo per MyAnimeList.

Attributo AnimeU	**AnimeU DType**	**Attributo AnimeW**	**AnimeW DType**	**Attributo MAL**	**MAL DType**
Id	`integer`	Id	`integer`	Id	`integer`
Dub	`boolean`	Dub	`string`	-	-
Stagione	`string`	Stagione	`string`	-	-
TipoAnime	`string`	TipoAnime	`string`	-	-
Titolo	`string`	Titolo	`string`	Title	`string`
Data_uscita	`integer`	Data di uscita	`string`	-	-
Plot	`string`	Description	`string`	-	-
-	-	-	-	Ranked	`integer`
-	-	-	-	Popularity	`integer`
Studio	`string`	-	-	-	-
Members	`integer`	-	-	Members	`integer`
Studio	`string`	-	-	-	-

Table 6: Tabella di Consistency

6 Analisi dei risultati

6.1 Confronto tra Top100 AnimeUnity e Top100 AnimeWorld

Una volta ottenuto il dataset completo, possiamo procedere all'analisi della Top 100 degli anime presenti sui due siti di streaming italiani. Confrontiamo la formula per individuare gli anime più visti in Italia. Lo scopo di questa analisi è identificare le tendenze e i gusti prevalenti del pubblico italiano.

Listing 1: 1 query

```
SELECT e.id, e.dub, d.titolo, d.animeunity, d.myanimelist,
d.visualizzazioneAnimeUnity, d.visualizzazioneAnimeWorld
FROM anime e
INNER JOIN (
  SELECT a1.id, b.dub, a1.titolo, s.animeunity, b.myanimelist,
  a1.visualizzazioneAnimeUnity, b.visualizzazioneAnimeWorld
  FROM anime a1
  INNER JOIN score s ON a1.id = s.id AND a1.dub = s.dub
  INNER JOIN (
    SELECT b.titolo, b.id, b.dub,
    b.visualizzazioneAnimeWorld, p.myanimelist
    FROM anime b
    INNER JOIN score p ON b.id = p.id AND b.dub = p.dub
    ORDER BY b.visualizzazioneAnimeWorld DESC
    LIMIT 100
  ) AS b ON b.dub = a1.dub AND b.id = a1.id
  WHERE a1.visualizzazioneAnimeUnity >= (
    SELECT 0.5 * AVG(a2.visualizzazioneAnimeUnity)
    FROM anime a2
  )
  GROUP BY a1.id, b.dub, a1.titolo,
  b.visualizzazioneAnimeWorld,
  b.myanimelist, a1.visualizzazioneAnimeUnity
  HAVING AVG(s.animeunity) >= (
    SELECT AVG(avg_score)
    FROM (
      SELECT AVG(animeunity) AS avg_score
      FROM score
      GROUP BY id
    ) AS subquery
  )
  ORDER BY AVG(s.animeunity) DESC
```

```
  LIMIT 100
) AS d ON e.id = d.id AND e.dub = d.dub;
```

Abbiamo eseguito un'operazione di join interna degli anime più visti sulle due piattaforme e abbiamo ottenuto un totale di 89 titoli in comune.

id	dub	titolo	avg_score	myanimelist	visualizzazioneAnimeUnity	visualizzazioneAnimeWorld
21	0	One Piece	9.41	8.69	35.224.643	24.550.641
1.735	0	Naruto: Shippuuden	9.34	8.26	11.059.425	11.349.667
21	1	One Piece (ITA)	9.25	8.69	11.910.145	15.971.310
1.735	1	Naruto: Shippuuden (ITA)	9.24	8.26	15.910.229	12.809.491
5.114	0	Hagane no Renkinjutsushi: Fullmetal Alchemist	9.22	9.1	343.263	1.027.205
11.061	0	Hunter x Hunter (2011)	9.18	9.04	5.421.380	7.451.319
40.776	0	Haikyuu!!: To the Top 2	9.1	8.54	1.183.836	816.915
813	1	Dragon Ball Z (ITA)	9.06	8.16	4.300.157	3.308.871
47.778	0	Kimetsu no Yaiba: Yuukaku-hen	9.06	8.8	1.356.295	1.705.933
48.583	0	Shingeki no Kyojin: The Final Season Part 2	9.05	8.77	954.920	1.455.321
235	0	Meitantei Conan	9.04	8.17	8.080.783	5.253.122
32.935	0	Haikyuu!! 3	9.04	8.78	591.842	747.532
39.551	0	Tensei shitara Slime Datta Ken 2	9.02	8.39	785.147	1.009.288
41.487	0	Tensei shitara Slime Datta Ken 2 Part 2	9	8.33	575.718	897.884
40.028	0	Shingeki no Kyojin: The Final Season	8.98	8.8	183.666	1.306.682
35.972	0	Fairy Tail: Final Series	8.98	7.57	1.016.618	1.663.218
38.883	0	Haikyuu!! to the top	8.96	8.36	974.990	1.023.431
263	0	Fighting Spirit	8.95	8.76	541.539	699.323
269	1	Bleach (ITA)	8.93	7.91	1.806.960	1.281.987
37.991	0	JoJo no Kimyou na Bouken Part 5: Ougon no Kaze	8.93	8.58	237.101	1.104.673
5.231	1	Inazuma Eleven (ITA)	8.92	7.69	2.959.245	1.577.921
20	0	Naruto	8.91	7.98	2.762.365	2.197.291
20	1	Naruto (ITA)	8.88	7.98	8.959.749	9.053.289

Figure 7: Inner join tra Top 100 Animeworld e Top 100 AnimeUnity

Tra questi, rileviamo 15 anime che presentano l'attributo dub=1, indicando che si tratta di una versione doppiata. In generale, si osserva che la versione doppiata presenta meno visualizzazioni rispetto alla controparte in lingua originale. Ciò nonostante, serie come Naruto:Shippuden e Dragonball, mostrano una tendenza opposta. Si può ipotizzare che la differenza potrebbe essere dovuta dalla preferenza degli utenti.

Per quanto riguarda le caratteristiche dell'insieme, eseguiamo alcune query per estrarre le informazioni più rappresentative. Analizzando i dati ottenuti da una precedente interrogazione mediante un semplice conteggio per attributo, possiamo osservare che tra questi, sono stati rilasciati 36 anime nel periodo compreso tra il 2019 e il 2021, con il 2019 che risulta l'anno con più titoli popolari. L'anno di rilascio degli anime dell'insieme varia invece dal 1986 al 2022 , coprendo un ampio range temporale. In relazione alla stagione,l'autunno con 32 titoli risulta essere il periodo più indicato per l'uscita di una nuova serie.

Con la seguente interrogazione contiamo i titoli raggruppando per lunghezza degli episodi e e tipo di anime.

Listing 2: 2 query

```
SELECT e.durataEpisodi, e.tipoAnime,
COUNT(e.id) AS numero_anime
FROM anime e
INNER JOIN (
  SELECT a1.id, b.dub, a1.titolo, s.animeunity,
  b.myanimelist, a1.visualizzazioneAnimeUnity,
  b.visualizzazioneAnimeWorld
  FROM anime a1
  INNER JOIN score s ON a1.id = s.id AND a1.dub = s.dub
  INNER JOIN (
    SELECT b.titolo, b.id, b.dub,
    b.visualizzazioneAnimeWorld, p.myanimelist
    FROM anime b
    INNER JOIN score p ON b.id = p.id AND b.dub = p.dub
    ORDER BY b.visualizzazioneAnimeWorld DESC
    LIMIT 100
  ) AS b ON b.dub = a1.dub AND b.id = a1.id
  WHERE a1.visualizzazioneAnimeUnity >= (
    SELECT 0.5 * AVG(a2.visualizzazioneAnimeUnity)
    FROM anime a2
  )
  GROUP BY a1.id, b.dub, a1.titolo,
  b.visualizzazioneAnimeWorld, b.myanimelist,
  a1.visualizzazioneAnimeUnity
  HAVING AVG(s.animeunity) >= (
    SELECT AVG(avg_score)
    FROM (
      SELECT AVG(animeunity) AS avg_score
      FROM score
      GROUP BY id
    ) AS subquery
  )
  ORDER BY AVG(s.animeunity) DESC
  LIMIT 100
) AS d ON e.id = d.id AND e.dub = d.dub
GROUP BY e.durataEpisodi, e.tipoAnime;
```

Si può notare che il campione è rappresentato per il 98,9% da anime distribuiti in TV, mentre il restante si riferisce agli ONA (Original Net Animation). Si tratta di titoli realizzati e distribuiti esclusivamente per la visione sul web e solitamente consititono in episodi molto brevi con una durata variabile. Mentre la durata media di un episodio TV anime si attesta sui 24

minuti.

durataEpisodi	tipoAnime	numero_anime
24	TV	66
23	TV	11
25	TV	10
26	TV	1
7	ONA	1

Figure 8: I titoli sono stati raggruppati a seconda del tipo di anime e durata dell'episodio

Analizzando i generi di maggior successo singolarmente, possiamo osservare che il genere Action compare 49 volte, seguito dal Fantasy con 36 e Adventure con 28 titoli. Se invece consideriamo l'insieme dei generi per ogni anime, emerge che la combinazione Action, Adventure e Fantasy è la più apprezzata dal pubblico italiano.

Listing 3: 3 query

```
SELECT g.tot_generi, COUNT(e.titolo) AS tot
FROM anime e
INNER JOIN (
  SELECT a1.id, b.dub, a1.titolo, s.animeunity,
  b.myanimelist, a1.visualizzazioneAnimeUnity,
  b.visualizzazioneAnimeWorld
  FROM anime a1
  INNER JOIN score s ON a1.id = s.id AND a1.dub = s.dub
  INNER JOIN (
    SELECT b.titolo, b.id, b.dub,
    b.visualizzazioneAnimeWorld, p.myanimelist
    FROM anime b
    INNER JOIN score p ON b.id = p.id AND b.dub = p.dub
    ORDER BY b.visualizzazioneAnimeWorld DESC
    LIMIT 100
  ) AS b ON b.dub = a1.dub AND b.id = a1.id
  WHERE a1.visualizzazioneAnimeUnity >= (
    SELECT 0.5 * AVG(a2.visualizzazioneAnimeUnity)
    FROM anime a2
  )
  GROUP BY a1.id, b.dub, a1.titolo,
  b.visualizzazioneAnimeWorld, b.myanimelist,
  a1.visualizzazioneAnimeUnity
```

```
  HAVING AVG(s.animeunity) >= (
    SELECT AVG(avg_score)
    FROM (
      SELECT AVG(animeunity) AS avg_score
      FROM score
      GROUP BY id
    ) AS subquery
  )
  ORDER BY AVG(s.animeunity) DESC
  LIMIT 100
) AS d ON e.id = d.id AND e.dub = d.dub

LEFT JOIN (
  SELECT idAnime, dub,
  GROUP_CONCAT(genere SEPARATOR ',␣') AS tot_generi
  FROM generi_anime
  GROUP BY idAnime, dub
) AS g ON e.id = g.idAnime AND e.dub = g.dub
WHERE e.dub=0
GROUP BY g.tot_generi
ORDER BY tot desc;
```

tot_generi	tot
Action, Adventure, Fantasy	13
Sports	8
Action, Adventure, Supernatural	6
Action, Adventure, Comedy, Fantasy	5
Action, Fantasy	4
Action	4
Action, Drama	2
Action, Award Winning, Fantasy	2
Drama, Fantasy, Ecchi	2
Romance	2

Figure 9: I titoli sono stati raggruppati a seconda del genere

Se consideriamo le case produttrici, lo studio David Production si distingue per aver realizzato 7 anime di successo, totalizzando 9,4 millioni di views tra le due piattaforme streaming. Tenendo conto unicamente del titolo, si tratta delle saghe di "Enen no Shouboutai" e "Jojo". Al contrario, se consideriamo lo studio con il maggiore numero di anime differenti di successo,

troviamo lo studio Perrot, Madhouse e TMS Entertainment ciascuno con 4 anime. Riguardo le visualizzazioni, lo studio Toei Animation si posiziona come lo studio con più views con oltre 62 millioni di visualizzazioni, prendendo in considerazione unicamente le versioni con dub = 0. Va anche detto che tra gli Anime prodotti da questo studio figura "One Piece", una serie in corso dal 1999 con 1044 episodi al momento dello scraping.

Listing 4: 4 query

```
SELECT e.studio, COUNT(d.titolo) AS tot_anime,
GROUP_CONCAT(d.titolo SEPARATOR ',') AS nomi_anime,
sum(d.visualizzazioneAnimeUnity),
sum(d.visualizzazioneAnimeWorld)
FROM anime  e
INNER JOIN (
  SELECT a1.id, b.dub, a1.titolo, s.animeunity,
  b.myanimelist, a1.visualizzazioneAnimeUnity,
  b.visualizzazioneAnimeWorld
  FROM anime a1
  INNER JOIN score s ON a1.id = s.id AND a1.dub = s.dub
  INNER JOIN (
    SELECT b.titolo, b.id, b.dub,
    b.visualizzazioneAnimeWorld, p.myanimelist
    FROM anime b
    INNER JOIN score p ON b.id = p.id AND b.dub = p.dub
    ORDER BY b.visualizzazioneAnimeWorld DESC
    LIMIT 100
  ) AS b ON b.dub = a1.dub AND b.id = a1.id
  WHERE a1.visualizzazioneAnimeUnity >= (
    SELECT 0.5 * AVG(a2.visualizzazioneAnimeUnity)
    FROM anime a2
  )
  GROUP BY a1.id, b.dub, a1.titolo,
  b.visualizzazioneAnimeWorld, b.myanimelist,
  a1.visualizzazioneAnimeUnity
  HAVING AVG(s.animeunity) >= (
    SELECT AVG(avg_score)
    FROM (
      SELECT AVG(animeunity) AS avg_score
      FROM score
      GROUP BY id
    ) AS subquery
  )
  ORDER BY AVG(s.animeunity) DESC
```

```
  LIMIT 100
) AS d ON e.id = d.id AND e.dub = d.dub
WHERE e.dub=0
GROUP BY e.studio
ORDER BY tot_anime DESC
```

studio	tot_anime	nomi_anime	sum(d.visualizzazioneAnimeUnity)	sum(d.visualizzazioneAnimeWorld)
David Production	7	JoJo no Kimyou na Bouken Part 4: Diamond wa ...	2.670.536	6.727.364
Production I.G	6	Kuroko's Basketball 2, Haikyuu!!: To the Top 2, ...	5.297.511	5.498.801
Bones	5	Hagane no Renkinjutsushi: Fullmetal Alchemist,...	2.692.191	7.802.520
Studio Pierrot	5	Bleach, Black Clover, Naruto: Shippuuden, Naru...	26.110.954	33.049.668
A-1 Pictures	4	Kaguya-sama wa Kokurasetai: Tensai-tachi no ...	4.812.495	5.283.825
MADHOUSE	4	Hunter x Hunter (2011), Fighting Spirit, The Irr...	7.079.051	9.857.167
MAPPA	4	Jujutsu Kaisen, Shingeki no Kyojin: The Final S...	4.967.311	7.492.579
TMS Entertainment	4	Dr. Stone, Kanojo, Okarishimasu, Fruits Basket ...	10.066.451	8.554.222
CloverWorks	3	Rascal Does Not Dream of Bunny Girl Senpai, S...	2.259.078	2.702.298
Toei Animation	3	World Trigger, Dragon Ball Super, One Piece	36.326.171	26.139.422
White Fox	3	Akame ga Kill!, Re:Zero kara Hajimeru Isekai S...	1.584.779	2.358.806
8bit	2	Tensei shitara Slime Datta Ken 2, Tensei shitar...	1.360.865	1.907.172
J.C. Staff	2	Is It Wrong to Try to Pick Up Girls in a Dungeon...	479.267	1.755.016

Figure 10: I titoli e il numero di visualizzazioni di entrambe le fonti sono stati aggregati in base allo studio

6.2 Confronto con MyAnimeList

Come metrica, abbiamo adottato la "Most Popular" che classifica gli anime in base agli utenti che aggiungono il titolo alla propria lista.

6.2.1 Fase esplorativa

In questa prima fase esploriamo alcune caratteristiche di tutti gli anime disponibili.

Listing 5: 5 query

```
SELECT a.id, titolo, popularity, members, ranks, s.myanimelist
FROM anime a
LEFT JOIN score s ON a.id= s.id AND a.dub= s.dub
WHERE a.dub = 0
ORDER BY members DESC;
```

Dall'interrogazione emerge che gli anime più popolari in termini di visualizzazioni non sono necessariamente quelli che hanno ottenuto uno score medio più elevato. Questo suggerisce l'assenza di una forte associazione tra il numero di views e lo score medio ottenuto.

id	titolo	popularity	members	ranks	myanimelist
16.498	Shingeki no Kyojin	1	3.723.767	107	8.54
5.114	Hagane no Renkinjutsushi: Fullmetal Alchemist	3	3.158.223	1	9.1
30.276	One Punch Man	4	3.041.495	128	8.5
11.757	SAO	5	2.938.798	3.051	7.2
31.964	Boku no Hero Academia	6	2.866.953	742	7.89
38.000	Demon Slayer: Kimetsu no Yaiba	7	2.782.386	122	8.51
20	Naruto	8	2.702.446	614	7.98
22.319	Tokyo Ghoul	9	2.686.161	943	7.79
11.061	Hunter x Hunter (2011)	10	2.638.223	10	9.04
32.281	Kimi no Na wa	11	2.580.898	28	8.85
25.777	Shingeki no Kyojin 2	12	2.550.857	129	8.5
33.486	Boku no Hero Academia 2	14	2.398.508	446	8.11
1.735	Naruto: Shippuuden	15	2.334.552	280	8.26

Figure 11: I titoli sono visualizzati in base al numero di membri in ordine decrescente

Se consideriamo lo studio, la casa di produzione più prolifica dal 1986 al 2022 è A-1 Pictures con 99 titoli rilasciati, seguita da "Studio Denn" con 99 e "J.C: Staff" con 73 anime. Approfondendo ulteriormente , con la seguente interrogazione abbiamo calcolato la media dei membri e lo score medio per anime.

Listing 6: 6 query

```
SELECT a.studio, COUNT(a.id) AS tot_anime,
ROUND(AVG(a.members),2) AS media_membri,
ROUND(AVG(d.myanimelist), 2) AS score_medio
FROM anime a
LEFT JOIN score d ON a.dub = d.dub AND a.id = d.id
WHERE a.dub = 0 AND a.studio IS NOT NULL
GROUP BY a.studio
ORDER BY media_membri DESC;
```

Un dato interessante è che l'anime che ha ottenuto il punteggio più alto secondo la nostra classifica deriva da una collaborazione tra lo studio "Trigger" e lo studio "CloverWorks".Invece, se prendiamo in considerazione le case di produzione con almeno 10 titoli prodotti, "Wit Studio" registra una media di 922 mila membri per anime e uno score medio di 7.97.

studio	tot_anime	media_membri	score_medio
Trigger, CloverWorks	1	1.557.341.0	7.21
CoMix Wave Films	3	1.468.786.67	8.24
MAPPA, Tezuka Productions	1	1.138.080.0	8.24
Wit Studio	16	921.779.06	7.97
SILVER LINK., Nexus	1	904.082.0	7.45
Studio DEEN, Marvy Jack	1	687.264.0	6.46
Studio Bind	3	681.923.0	8.33
White Fox	19	649.197.47	7.73
Bones	46	630.216.78	7.61
CloverWorks	14	592.293.5	7.6
ufotable	23	566.337.91	7.86
Orange	3	549.228.67	8.01
Ashi Productions	1	524.335.0	6.15
Trigger	9	521.480.89	7.59
A-1 Pictures	99	461.073.8	7.46

Figure 12: Dopo aver raggruppato gli anime in base allo studio di produzione, abbiamo calcolato il punteggio medio e il numero medio di membri

6.2.2 Top 100 MyAnimeList

Nella comparazione tra la fonte internazionale e le fonti italiane prendiamo in considerazione solo gli anime sottotitolati, in modo da non avere titoli duplicati ma un risultato più coerente. Questa decisione è stata presa dopo aver verificato se i titoli, con la versione italiana, siano o meno presenti nella top 100 di MyAnimeList. Di seguito sono riportati i primi 100 anime.

Listing 7: 7 query

```
SELECT a.id, a.titolo, a.popularity
FROM anime a
inner JOIN (
SELECT id,dub, studio
FROM anime
WHERE popularity <= 100 AND dub= 0
) AS b ON  a.id=b.id AND a.dub=b.dub
ORDER BY a.popularity asc;
```

Otteniamo cosi un'elenco di 83 righe. la mancata presenza degli anime restanti è dovuta alle modalità di scraping utilizzate durante il processo di raccolta dei dati.

id	titolo	popularity
16.498	Shingeki no Kyojin	1
5.114	Hagane no Renkinjutsushi: Fullmetal Alchemist	3
30.276	One Punch Man	4
11.757	SAO	5
31.964	Boku no Hero Academia	6
38.000	Demon Slayer: Kimetsu no Yaiba	7
20	Naruto	8
22.319	Tokyo Ghoul	9
11.061	Hunter x Hunter (2011)	10
32.281	Kimi no Na wa	11
25.777	Shingeki no Kyojin 2	12
33.486	Boku no Hero Academia 2	14
1.735	Naruto: Shippuuden	15
19.815	No Game No Life	16
40.748	Jujutsu Kaisen	17

Figure 13: Top 100 MyAnimelist in base alla popolarità

Dall'interrogazione, si può notare come i primi posti siano occupati da anime diversi rispetto alle considerazioni svolte sulle piattafrome italiane.
Un esempio importante è l'anime "Shingeki no Kyojin", più comunemente conosciuto come "Attack on Titan", che si trova in cima alla lista di MyAnimeList ma che ha ottenuto poche visualizzazioni in altre fonti. Questo potrebbe essere spiegato dal fatto che Prime Video, un servizio di streaming incluso nell'abbonamento Prime di Amazon, ha acquistato i diritti dell'anime e ha reso disponibili gli episodi sottotitolati subito dopo la messa in onda in Giappone, consentendo agli utenti di seguire la serie in tempo reale, ad un prezzo molto contenuto. Inoltre la popolarità di questa serie può essere dovuta anche all'uscita della sua stagione finale, rilasciata in più parti, tra l'agosto del 2022 e quello futuro del 2023.
Se invece consideriamo i centri di produzione, lo studio "A-1 Pictures" ha ottenuto un notevole successo, con ben 10 anime posizionati tra i più popolari. Tra di essi, figurano titoli come "Fairy Tail", "SAO" e "Nanatsu no Taizai".
Con quest'ultima interrogazione facciamo un inner join tra le classifiche Top 100 anime di tutte e tre le fonti.

Listing 8: 8 query

```
SELECT e.id, e.dub, d.titolo, v.popularity,
v.myanimelist, d.animeunity, d.animeworld
FROM anime e
INNER JOIN (
  SELECT a1.id, b.dub, a1.titolo,s.animeworld,
```

```
  s.animeunity, b.myanimelist, a1.visualizzazioneAnimeUnity
  , b.visualizzazioneAnimeWorld
  FROM anime a1
  INNER JOIN score s ON a1.id = s.id AND a1.dub = s.dub
  INNER JOIN (
    SELECT b.titolo, b.id, b.dub,
    b.visualizzazioneAnimeWorld, p.myanimelist
    FROM anime b
    INNER JOIN score p ON b.id = p.id AND b.dub = p.dub
    WHERE p.dub = 0
    ORDER BY b.visualizzazioneAnimeWorld DESC
    LIMIT 100
  ) AS b ON b.dub = a1.dub AND b.id = a1.id
  WHERE b.dub = 0 AND a1.visualizzazioneAnimeUnity >= (
    SELECT 0.5 * AVG(a2.visualizzazioneAnimeUnity)
    FROM anime a2
  )
  GROUP BY a1.id, b.dub, a1.titolo, b.visualizzazioneAnimeWorld,
  b.myanimelist, a1.visualizzazioneAnimeUnity
  HAVING AVG(s.animeunity) >= (
    SELECT AVG(avg_score)
    FROM (
      SELECT AVG(animeunity) AS avg_score
      FROM score
      GROUP BY id
    ) AS subquery
  )
  ORDER BY AVG(s.animeunity) DESC
  LIMIT 100
) AS d ON e.id = d.id AND e.dub = d.dub
INNER JOIN (
  SELECT r.id, titolo, popularity, r.dub, r.myanimelist
  FROM anime
  INNER JOIN score r ON r.id = anime.id AND r.dub = anime.dub
  WHERE popularity <= 100 AND r.dub = 0
) AS v ON v.dub = e.dub AND v.id = e.id
ORDER BY v.popularity ASC;
```

Il risultato ottenuto è una lista di 32 anime. Dal momento che myAnimelist includeva solo 83 titoli, le tre fonti (MyAnimeList, AnimeUnity e AnimeWorld) condividono il 38,5% dei titoli delle rispettive Top 100.

id	dub	titolo	popularity	myanimelist	animeunity	animeworld
5.114	0	Hagane no Renkinjutsushi: Fullmetal Alchemist	3	9.1	9.22	8.78
38.000	0	Demon Slayer: Kimetsu no Yaiba	7	8.51	8.75	8.56
20	0	Naruto	8	7.98	8.91	8.55
11.061	0	Hunter x Hunter (2011)	10	9.04	9.18	8.71
33.486	0	Boku no Hero Academia 2	14	8.11	8.34	8.2
1.735	0	Naruto: Shippuuden	15	8.26	9.34	8.63
40.748	0	Jujutsu Kaisen	17	8.64	8.85	8.63
21	0	One Piece	20	8.69	9.41	8.38
31.240	0	Re:Zero kara Hajimeru Isekai Seikatsu	25	8.24	8.47	8.16
36.456	0	Boku no Hero Academia 3	26	8.04	8.39	8.17
22.199	0	Akame ga Kill!	29	7.47	8.25	8.18
23.755	0	Nanatsu no Taizai	31	7.67	8.23	8.2
24.833	0	Ansatsu Kyoushitsu	34	8.09	8.42	8.37
20.583	0	Haikyuu!!	36	8.44	8.88	8.64
9.919	0	Ao no Exorcist	39	7.5	8	8
269	0	Bleach	40	7.91	8.78	8.27
40.028	0	Shingeki no Kyojin: The Final Season	41	8.8	8.98	8.48
6.702	0	Fairy Tail	49	7.57	8.45	8.02

Figure 14: Dopo aver ottenuto la Top 100 da tutte e tre le fonti, è stata effettuata una join interna

La valutazione media degli anime è pari all'8,11% per MyAnimeList, all'8,64% per AnimeUnity e all'8,47% per AnimeWorld. Dallo studio "A1-Picutres" emerge la maggiore consistenza nel creare anime di successo, con 4 diversi titoli che hanno riscosso un apprezzamento a livello mondiale. La combinazione dei generi Azione, Avventura e Fantasy si conferma la scelta migliore per attirare una vasta platea di utenti.

7 Conclusioni e sviluppi futuri

In conclusione, abbiamo creato un database relazionale per le serie di "anime" da tre fonti di dati ricavate dallo scraping di AnimeWorld, AnimeUnity e MyAnimeList. Il database cosi ottenuto, contiene tutti i personaggi e gli episodi delle serie, con attributi utili come i rating, lo studio di produzione e il genere, consentendoci di rispondere ai numerosi interrogativi che ci siamo posti.

Il database può essere arricchito e ampliato ogni volta che vengono aggiornati anime, episodi o personaggi. Siamo soddisfatti che il flusso di lavoro possa essere quasi completamente automatizzato.

Per gli sviluppi futuri, l'automazione è sicuramente interessante e ci permetterebbe di approfondire altri aspetti. La possibile aggiunta di nuovi attributi, come il nome del doppiatore o il sesso dei personaggi principali, potrebbe permetterci di effettuare confronti più complessi per la tabella personaggi, ad esempio osservando quali doppiatori, associati a determinati personaggi, compaiono in serie di successo, o ancora quante volte un doppiatore è la voce di un personaggio principale, o di supporto, di una serie anime.