ETL Project - Data Vis Bootcamp

Snehitha Soma, Karissa Malseed, Hamish Cocks

Introduction

This project required us to use the Extract Transform Load strategy to process and import data into a usable database. For our project we chose to analyse information regarding the cast, crew, and ratings of every movie released by the extremely popular Walt Disney Company from its beginnings to the present day.

Our aim was to find varying types of freely available online data sources that were to then be cleaned using the pandas Python library, and subsequently load these cleaned dataframes into a PostgreSQL database ready for exploration.

Extract

The data sources used in the project were extracted from:

1. Disney movie data from kaggle.com.au as a json file: https://www.kaggle.com/sooaaib/walt-disney-movies?select=disney_movies.json

2. MPAA/Age Rating data from kaggle.com.au as csv files: https://www.kaggle.com/prateekmaj21/disney-movies

3. The Disney movie characters and voice actors sources from dataworld.com.au as csv files: https://data.world/kgarrett/disney-character-success-00-16/workspace/file?filename=disney-characters.csv

Transform

The extracted data was retrieved into jupyter notebooks for a series of transformations, clean ups.

The specials characters(/n) were removed from the movie title columns.

```
#Formatting the movie_title Column in Characters_df to remove '\n' Characters
characters_df['movie_title']=  characters_df['movie_title'].str.replace('\r\n', '')
characters_df.head()
```

	movie_title	release_date	hero	villian	song
0	\nSnow White and the Seven Dwarfs	December 21, 1937	Snow White	Evil Queen	Some Day My Prince Will Come
1	\nPinocchio	February 7, 1940	Pinocchio	Stromboli	When You Wish upon a Star
2	\nFantasia	November 13, 1940	NaN	Chernabog	NaN
3	Dumbo	October 23, 1941	Dumbo	Ringmaster	Baby Mine
4	\nBambi	August 13, 1942	Bambi	Hunter	Love Is a Song

The Release date column was cleaned by extracting the first index element from the list.

	Title	Release_Date	Director	IMDB Rating	Run Time(mins)	Budget (USD in Millions)
0	Academy Award Review of	[May 19, 1937]	NaN	7.2	41 minutes (74 minutes 1966 release)	NaN
1	Snow White and the Seven Dwarfs	[December 21, 1937 (Carthay Circle Theatre , ...	[David Hand (supervising), William Cottrell, W...	7.6	83 minutes	1490000.0
2	Pinocchio	[February 7, 1940 (Center Theatre), February...	[Ben Sharpsteen, Hamilton Luske, Bill Roberts,...	7.4	88 minutes	2600000.0
3	Fantasia	[November 13, 1940]	[Samuel Armstrong, James Algar, Bill Roberts, ...	7.8	126 minutes	2280000.0
4	The Reluctant Dragon	[June 20, 1941]	[Alfred Werker, (live action), Hamilton Luske,...	6.9	74 minutes	600000.0

```
#Formatting the Release Date column
movies_df['Release_Date' ] = movies_df['Release_Date'].str[0]
new_df  = movies_df["Release_Date"].str.split("(", n = 1, expand = True)
movies_df["Release_Date"] = new_df[0]
movies_df.head()
```

	Title	Release_Date	Director	IMDB Rating	Run Time(mins)	Budget (USD in Millions)
0	Academy Award Review of	May 19, 1937	NaN	7.2	41 minutes (74 minutes 1966 release)	NaN
1	Snow White and the Seven Dwarfs	December 21, 1937	[David Hand (supervising), William Cottrell, W...	7.6	83 minutes	1490000.0
2	Pinocchio	February 7, 1940	[Ben Sharpsteen, Hamilton Luske, Bill Roberts,...	7.4	88 minutes	2600000.0
3	Fantasia	November 13, 1940	[Samuel Armstrong, James Algar, Bill Roberts, ...	7.8	126 minutes	2280000.0
4	The Reluctant Dragon	June 20, 1941	[Alfred Werker, (live action), Hamilton Luske,...	6.9	74 minutes	600000.0

All the movie tiles with improper release dates were dropped.

```
whole_df = whole_df.drop_duplicates(subset=['Release_Date'])
whole_df = whole_df.reset_index(drop=True)
basic_df = whole_df[['Title', 'Release_Date', 'Director','Run Time(mins)','mpaa_rating', 'genre' ]]
basic_df = basic_df.sort_values(by=['Title'])
basic_df = basic_df[(basic_df['Release_Date'].str.len() > 9)]
basic_df = basic_df.reset_index(drop=True)
basic_df = basic_df.reset_index(drop=False)
basic_df = basic_df.rename(columns = {"index":"id", "Title":"title", "Release_Date":"release_date", "Director":"director",
                                      "Run Time(mins)":"run_time", 'mpaa_rating':"age_rating", "genre":"genre"})
basic_df
```

	id	title	release_date	director	run_time	age_rating	genre
0	0	101 Dalmatians	November 27, 1996	Stephen Herek	103 minutes	G	Comedy
1	1	102 Dalmatians	November 22, 2000	Kevin Lima	100 minutes	G	Comedy
2	2	20,000 Leagues Under the Sea	December 23, 1954	Richard Fleischer	127 minutes	NaN	Adventure
3	3	A Bug's Life	November 20, 1998	John Lasseter	95 minutes	G	Adventure
4	4	A Far Off Place	March 12, 1993	Mikael Salomon	108 minutes	PG	Adventure
...	...	...	...	...	...	...	...
209	209	Wild Hearts Can't Be Broken	May 24, 1991	Steve Miner	88 minutes	NaN	Drama
210	210	Winnie the Pooh	April 6, 2011	[Stephen J. Anderson, Don Hall]	69 minutes	G	Adventure
211	211	Wreck-It Ralph	October 29, 2012	Rich Moore	101 minutes	PG	Adventure
212	212	Zokkomon	22 April 2011	Satyajit Bhatkal	109 minutes	PG	Adventure
213	213	Zootopia	February 13, 2016	[Byron Howard, Rich Moore]	108 minutes	PG	Adventure

The cleaned data frames were then merged accordingly to create four tables to hold relevant data.

The ‘Basic’ table holds brief information about each movie.

```
basic_df
```

	id	title	release_date	director	run_time	age_rating	genre
0	0	101 Dalmatians	November 27, 1996	Stephen Herek	103 minutes	G	Comedy
1	1	102 Dalmatians	November 22, 2000	Kevin Lima	100 minutes	G	Comedy
2	2	20,000 Leagues Under the Sea	December 23, 1954	Richard Fleischer	127 minutes	NaN	Adventure
3	3	A Bug's Life	November 20, 1998	John Lasseter	95 minutes	G	Adventure
4	4	A Far Off Place	March 12, 1993	Mikael Salomon	108 minutes	PG	Adventure
...	...	...	...	...	...	...	...
209	209	Wild Hearts Can't Be Broken	May 24, 1991	Steve Miner	88 minutes	NaN	Drama
210	210	Winnie the Pooh	April 6, 2011	[Stephen J. Anderson, Don Hall]	69 minutes	G	Adventure
211	211	Wreck-It Ralph	October 29, 2012	Rich Moore	101 minutes	PG	Adventure
212	212	Zokkomon	22 April 2011	Satyajit Bhatkal	109 minutes	PG	Adventure
213	213	Zootopia	February 13, 2016	[Byron Howard, Rich Moore]	108 minutes	PG	Adventure

214 rows × 7 columns

The ‘Ratings’ table holds data about review/rating of each movie.

```
ratings_df.head(20)
```

	title	release_date	imdb	rotten_tomatoes
0	101 Dalmatians	November 27, 1996	5.7	41%
1	102 Dalmatians	November 22, 2000	4.9	31%
2	20,000 Leagues Under the Sea	December 23, 1954	7.2	89%
3	A Bug's Life	November 20, 1998	7.2	92%
4	A Far Off Place	March 12, 1993	6.6	45%
5	A Goofy Movie	April 7, 1995	6.8	58%
6	A Kid in King Arthur's Court	August 11, 1995	4.7	5%
7	African Cats	April 22, 2011	7.6	73%
8	Aladdin	May 8, 2019	8.0	95%
9	Aladdin	November 25, 1992	8.0	95%
10	Alice Through the Looking Glass	May 10, 2016	6.2	29%
11	Alice in Wonderland	July 26, 1951	6.4	51%
12	Alice in Wonderland	February 25, 2010	6.4	51%
13	Aliens of the Deep	January 28, 2005	6.4	84%
14	Angels in the Outfield	July 15, 1994	6.2	33%
15	Around the World in 80 Days	June 13, 2004	5.9	32%
16	Atlantis: The Lost Empire	June 3, 2001	6.9	49%
17	Babes in Toyland	December 14, 1961	6.3	36%
18	Bears	April 18, 2014	7.4	90%
19	Beauty and the Beast	February 23, 2017	8.0	94%

The ‘Crew’ table holds information about various technicians worked on every movie.

```
crew_df.head(10)
```

	title	release_date	composer	cinematographer	editor	screenplay
0	Snow White and the Seven Dwarfs	December 21, 1937	[Frank Churchill, Paul Smith, Leigh Harline]	NaN	NaN	NaN
1	Pinocchio	February 7, 1940	[Leigh Harline, Paul J. Smith]	NaN	NaN	NaN
2	Fantasia	November 13, 1940	See program	James Wong Howe	NaN	NaN
3	Song of the South	November 12, 1946	[Edward Plumb, Daniele Amfitheatrof, Paul J. S...	Gregg Toland	William M. Morgan	[Live action:, Morton Grant, Maurice Rapf, Dal...
4	Cinderella	February 13, 2015	[Oliver Wallace, Paul J. Smith]	NaN	Donald Halliday	NaN
5	Cinderella	February 15, 1950	[Oliver Wallace, Paul J. Smith]	NaN	Donald Halliday	NaN
8	Alice in Wonderland	July 26, 1951	Oliver Wallace	NaN	Lloyd Richardson	NaN
9	Alice in Wonderland	February 25, 2010	Oliver Wallace	NaN	Lloyd Richardson	NaN
12	20,000 Leagues Under the Sea	December 23, 1954	[Paul Smith, Joseph S. Dubin, (orchestration)]	Franz Planer	Elmo Williams	Earl Felton
13	Lady and the Tramp	June 22, 1955	Oliver Wallace	NaN	Don Halliday	NaN

The ‘Cast’ table shows various characters in the movie and voice actors of those characters along with the character names of main leads(Hero and Villain).

```
cast_df
```

	title	release_date	hero	villain	character	voice_actor
0	Atlantis: The Lost Empire	June 3, 2001	Milo Thatch	Commander Rourke	Audrey Ramirez	Jacqueline Obradors
1	Atlantis: The Lost Empire	June 3, 2001	Milo Thatch	Commander Rourke	Commander Lyle Rourke	James Garner
2	Atlantis: The Lost Empire	June 3, 2001	Milo Thatch	Commander Rourke	Cookie	Jim Varney
3	Atlantis: The Lost Empire	June 3, 2001	Milo Thatch	Commander Rourke	Dr. Joshua Sweet	Phil Morris
4	Atlantis: The Lost Empire	June 3, 2001	Milo Thatch	Commander Rourke	Fenton Q. Harcourt	David Ogden Stiers
...	...	...	...	...	...	...
174	Wreck-It Ralph	October 29, 2012	Ralph	Turbo	Swizzle "The Swizz" Malarkey	None
175	Wreck-It Ralph	October 29, 2012	Ralph	Turbo	Taffyta Muttonfudge	Mindy Kaling
176	Wreck-It Ralph	October 29, 2012	Ralph	Turbo	Vanellope von Schweetz	Sarah Silverman
177	Wreck-It Ralph	October 29, 2012	Ralph	Turbo	Wreck-It Ralph	John C. Reilly
178	Wreck-It Ralph	October 29, 2012	Ralph	Turbo	Wynchel	Adam Carolla

179 rows × 6 columns

Load

The PostgreSQL table schema was defined in PgAdmin using an SQL query file based on the Entity Relationship Diagram found in the ‘schema.sql’ file. These tables reflected the finalised dataframes; ‘Basic’, ‘Ratings’, ‘Cast’, and ‘Crew’, outlined above. A connection to PostgreSQL was set up within the Jupyter Notebook using the SQLAlchemy function ‘create_engine’. The pandas function ‘to_sql’ was then used to read the dataframes into the afore-mentioned schemas of the SQL database. Another pandas function, ‘read_sql_query’, can then be utilised to carry out SQL queries on the schemas. PostgreSQL was the best option for us to construct this database as we are competent in SQLAlchemy and pandas. It also provides an excellent framework for one to query databases.

```
In [91]: rds_connection_string = f'postgres:{password}@localhost:5432/etl_project'
         engine = create_engine(f'postgresql://{rds_connection_string}')

In [92]: engine.table_names()
Out[92]: ['rating', 'casting', 'crew', 'basic']

In [93]: basic_df.to_sql(name='basic', con=engine, if_exists='append', index=False)

In [94]: pd.read_sql_query('select * from basic', con=engine).head()
```

Out[94]:

	id	title	release_date	director	run_time	age_rating	genre
0	0	101 Dalmatians	November 27, 1996	Stephen Herek	103 minutes	G	Comedy
1	1	102 Dalmatians	November 22, 2000	Kevin Lima	100 minutes	G	Comedy
2	2	20,000 Leagues Under the Sea	December 23, 1954	Richard Fleischer	127 minutes	None	Adventure
3	3	A Bug's Life	November 20, 1998	John Lasseter	95 minutes	G	Adventure
4	4	A Far Off Place	March 12, 1993	Mikael Salomon	108 minutes	PG	Adventure

NOTE: users must alter the ‘config.py’ file within the Github repository with their own PostgreSQL password in order for the connection to be established.

Conclusions

We were successful in our goal of employing ETL to a set of data. We did, however, run into some challenges and roadblocks along the way and can share the following observations:

- Picking a suitable dataset turned out to be more demanding than initially thought. Much of the data surrounding movies in general was often massive, and handling such big datasets seemed infeasible for this project. Furthermore we wanted to prioritise merging datasets from different sources rather than being handed one massive set with everything we need.

- As you can see from the above Transform section it was evident that the data was not in an ideal condition. Many of the columns were not consistent in their formatting, for example we ran into 3 or 4 entries within the ‘release_date’ column that were not in the same format as the rest of the entries. While it was an easy fix we did not pick it up right away and consequently had to backtrack. It was a good learning experience nonetheless as we were too trusting of the data source.

- During our efforts to load the dataframes into PostgreSQL we struggled with the SQL syntax and realised that SQL automatically renders all column and table names to lowercase which consequently did not match up with the dataframe column names. Similarly we had trouble defining a primary key within the database. Initially we defined Title as the primary key, but as there were remakes of the movies included in our tables we had to switch to Release Date as the primary key.

In conclusion it is evident that ETL is an extremely important method in developing databases. Generating a user-friendly and intuitive database is often the key to many companies' inner workings and therefore gaining competence in ETL should be high on the list of any budding data analyst.