ETL Project

Group 4 - Josh, Kaumudi, Kelly, Sam

What is our data?

Video Games Rating/Sales Data

We decided upon a scenario in which we are working with a budding video game company, poised to release a few different games into market.

The idea is that we find which game to release first based upon our findings.

We set out to find of the most popular games released between 1994 and 2016, and which ESRB rating gave way to higher volumes of sales.

The CSVs

Ratings.csv

game_title

other_platforms

developer

description

ESRB_rating

metascore

userscore

critic_positive

critic_negative

user_mixed

URL

platform

publisher

release_date

genre

multiplayer

num_meta_reviews

num_user_reviews

critic_mixed

user_positive

user_negative

https://www.kaggle.com/tyedwardse/metacritic-game-scores?select=metascrape.csv

Sales.csv

Ranking

Name

Platform

Year

Genre

NorthAmerica_Sales

Eurpean_Sales

Japan_Sales

Other_Sales

Global_Sales

https://www.kaggle.com/atharvaingle/video-games-dataset

What was our process?

Transforming and Cleaning our Data

We sliced out our desired columns from our datasets for use in our analysis

```
#Select columns from ratings_df for analysis
ratings_sliced=ratings_df[['game_title','platform','ESRB_Rating','metascore','userscore']]

#Rename some columns to clarify what the data is
ratings_renamed=ratings_sliced.rename(columns={'ESRB_Rating':'esrb_rating',
                                               'userscore':'user_score'})
ratings_renamed.head()
```

	game_title	platform	esrb_rating	metascore	user_score
0	Farming Simulator 22	PlayStation_5	E	77	5.9
1	Sherlock Holmes: Chapter One	PlayStation_5	M	75	7.5
2	Battlefield 2042	PlayStation_5	M	68	3.0
3	Grand Theft Auto: The Trilogy - The Definitive...	PlayStation_5	NaN	56	0.9
4	The Elder Scrolls V: Skyrim Anniversary Edition	PlayStation_5	M	74	3.2

```
#Select columns from sales_df for analysis
sales_sliced=sales_df[['Name','Platform','Global_Sales']]

sales_renamed=sales_sliced.rename(columns={'Name':'game_title',
                                           'Platform':'platform',
                                          'Global_Sales':'global_sales'})

sales_renamed.head()
```

	game_title	platform	global_sales
0	Wii Sports	Wii	82.74
1	Super Mario Bros.	NES	40.24
2	Mario Kart Wii	Wii	35.82
3	Wii Sports Resort	Wii	33.00
4	Pokemon Red/Pokemon Blue	GB	31.37

Transforming and Cleaning our Data

We merged our final dataframes, and dropped the duplicates to clean the merged dataset.

```
#Merge the two final dataframes
merged_df=pd.merge(ratings_final,sales_final,on=['game_title','atform'])
merged_df.head()
```

	game_title	platform	esrb_rating	metascore	user_score	global_sales
0	Grand Theft Auto V	Xbox_One	M	97	7.8	5.08
1	Grand Theft Auto V	Xbox_One	M	97	7.8	5.08
2	Call of Duty: Advanced Warfare	Xbox_One	M	81	5.5	5.13
3	Mario & Sonic at the Olympic Games	Wii	E	67	7.7	8.06
4	Mario Kart 64	Nintendo_64	E	83	8.6	9.87

```
#Drop duplicates to clean the merged dataset
merged_cleaned_df = merged_df.drop_duplicates()
merged_cleaned_df.head()
```

	game_title	platform	esrb_rating	metascore	user_score	global_sales
0	Grand Theft Auto V	Xbox_One	M	97	7.8	5.08
2	Call of Duty: Advanced Warfare	Xbox_One	M	81	5.5	5.13
3	Mario & Sonic at the Olympic Games	Wii	E	67	7.7	8.06
4	Mario Kart 64	Nintendo_64	E	83	8.6	9.87
5	Mario Kart 8	Wii_U	E	88	8.8	6.96

What does our database look like?

SQL Database

```
-- Create tables for raw data to be loaded into
CREATE TABLE ratings (
Game_Title TEXT,
Platform TEXT,
ESRB_Rating TEXT,
Metascore INT,
User_Score DECIMAL
);

CREATE TABLE sales (
Game_Title TEXT,
Platform TEXT,
Global_Sales TEXT
);
```

We created 2 tables based on the relevant columns.

	game_title text	platform text	esrb_rating text	metascore integer	user_score numeric	global_sales text
1	Mario Kart Wii	Wii	E	82	8.4	35.82
2	New Super Mario Bros. Wii	Wii	E	87	8.3	28.62
3	Grand Theft Auto V	PlayStation_3	M	97	8.3	21.4
4	Call of Duty: Modern Warfare 3	PlayStation_3	M	88	3.3	13.46
5	Super Smash Bros. Brawl	Wii	T	93	8.8	13.04
6	Grand Theft Auto V	PlayStation_4	M	97	8.4	11.98
7	Super Mario 64	Nintendo_64	E	94	9.1	11.89
8	Super Mario Galaxy	Wii	E	97	9.1	11.52
9	Gran Turismo	PlayStation	E	96	8.6	10.95
10	Gran Turismo 5	PlayStation_3	E	84	7.8	10.77
11	Call of Duty: Modern Warfare 2	PlayStation_3	M	94	6.6	10.69
12	Grand Theft Auto IV	PlayStation_3	M	98	7.8	10.57
13	Just Dance 3	Wii	E10+	74	8.0	10.26
14	Mario Kart 64	Nintendo_64	E	83	8.6	9.87
15	Final Fantasy VII	PlayStation	T	92	9.1	9.72
16	Call of Duty: Ghosts	PlayStation_3	M	71	2.8	9.59
17	Just Dance 2	Wii	E10+	74	7.2	9.52
18	Halo 2	Xbox	M	95	8.7	8.49
19	Mario Party 8	Wii	E	62	6.5	8.42
20	FIFA Soccer 13	PlayStation_3	E	88	6.6	8.24
21	The Sims 3	PC	T	86	7.8	8.11
22	GoldenEye 007	Nintendo_64	T	96	9.0	8.09
23	Mario & Sonic at the Olympic Games	Wii	E	67	7.7	8.06
24	Final Fantasy VIII	PlayStation	T	90	8.7	7.86
25	Super Mario Galaxy 2	Wii	E	97	9.1	7.69
26	The Legend of Zelda: Ocarina of Time	Nintendo_64	E	99	9.1	7.6
27	The Legend of Zelda: Twilight Princess	Wii	T	95	9.0	7.31
28	Just Dance	Wii	E10+	49	7.8	7.27
29	Battlefield 3	PlayStation_3	M	85	7.5	7.23

	game_title text	platform text	global_sales text
1	Wii Sports	Wii	82.74
2	Super Mario Bros.	NES	40.24
3	Mario Kart Wii	Wii	35.82
4	Wii Sports Resort	Wii	33.0
5	Pokemon Red/Pokemon Blue	GB	31.37
6	New Super Mario Bros.	DS	30.01
7	Wii Play	Wii	29.02
8	New Super Mario Bros. Wii	Wii	28.62
9	Duck Hunt	NES	28.31
10	Nintendogs	DS	24.76
11	Mario Kart DS	DS	23.42
12	Pokemon Gold/Pokemon Silver	GB	23.1
13	Wii Fit Plus	Wii	22.0
14	Grand Theft Auto V	PlayStation_3	21.4
15	Super Mario World	SNES	20.61
16	Brain Age: Train Your Brain in Minutes a Day	DS	20.22
17	Pokemon Diamond/Pokemon Pearl	DS	18.36
18	Super Mario Land	GB	18.14
19	Super Mario Bros. 3	NES	17.28
20	Grand Theft Auto V	Xbox_360	16.38
21	Pokemon Ruby/Pokemon Sapphire	GBA	15.85
22	Pokemon Black/Pokemon White	DS	15.32
23	Brain Age 2: More Training in Minutes a Day	DS	15.3
24	Call of Duty: Modern Warfare 3	Xbox_360	14.76
25	Pokémon Yellow: Special Pikachu Edition	GB	14.64
26	Call of Duty: Black OPlayStation	Xbox_360	14.64
27	Pokemon X/Pokemon Y	3DS	14.35
28	Call of Duty: Black OPlayStation 3	PlayStation_4	14.24
29	Call of Duty: Black OPlayStation II	PlayStation_3	14.03

Connecting between Python and SQL

In [14]:
```
#Import config file with password, username and database name from Postgres
from config import password,username,database
```

In [15]:
```
#Connect to database
rds_connection_string = f"{username}:{password}@localhost:5432/{database}"
engine = create_engine(f'postgresql://{rds_connection_string}')
```

In [16]:
```
#Check for tables
engine.table_names()
```

```
<ipython-input-16-3a4280413f14>:2: SADeprecationWarning: The Engine.table_names() method is deprecated and will be removed in a future rele
ase.  Please refer to Inspector.get_table_names(). (deprecated since: 1.4)
  engine.table_names()
```

Out[16]:
```
['ratings', 'sales', 'merged']
```

In [17]:
```
#Insert ratings data into Database
ratings_final.to_sql(name='ratings', con=engine, if_exists='append', index=False)
```

In [18]:
```
#Insert sales data into Database
sales_final.to_sql(name='sales', con=engine, if_exists='append', index=False)
```

In [19]:
```
#Insert merged data into Database
merged_cleaned_df.to_sql(name='merged', con=engine, if_exists='append', index=False)
```

Finally, this transformed database can be utilised by anyone within the company in their future analysis which is the advantage of transformation of data, loading and arriving at a new database.

Thanks for listening