Week - 3 LAB

Topic: “Introduction to Python & ETL”

Tumma Karthikeyan

M.S Analytics

Information Retrieval

Lead Instructor: Tim Toennies

Co- Instructor: Tim Hough

AA-5100-209. **Tumma Karthikeyan**

Date: 23 April 2023

Python & ETL

Introduction:

The purpose of this assignment is to import data about Disney movies from a CSV file into a SQL Server database using Python and Pandas. The data includes information such as the movie title, release date, genre, MPAA rating, IMDB rating, running time, budget, and box office. The objective is to create a table in the SQL Server database and insert the data from the CSV file into this table using Python and Pandas.

In this assignment, we will be using ODBC to establish a connection to the SQL Server database and create a table in it. We will then use Pandas to read the data from the CSV file and insert it into the table. Finally, we will use Python to view the first 25 records in the table.

This assignment will involve assumptions testing and breaking out the results by questions being answered. Tables with results will also be presented. The discussion will summarize what was found and provide recommendations based on the results.

Overall, this assignment will provide hands-on experience with importing data into SQL Server using Python and Pandas and analysing the data using SQL queries.

Step: 1

```
import pandas as pd

# load csv file

rawdata = pd.read_csv (r'C:\Users\student\Downloads\disney_movies_2.csv')

dataframe = pd.DataFrame(rawdata)
```

The code you provided is using the Pandas library to import data from a CSV file into a Pandas DataFrame.

The first line imports the pandas library and aliases it as 'pd'.

The second line reads the CSV file 'disney_movies_2.csv' using the read_csv() function provided by Pandas. The file path is specified as a raw string (r'disney_movies.csv') to avoid any issues with escape characters. The data from the CSV file is loaded into a pandas object called 'rawdata'.

The third line creates a new Pandas DataFrame called 'dataframe' from the 'rawdata' object. The purpose of this step is to create a tabular format of the data that can be easily manipulated and analyzed using Pandas. The DataFrame has a row for each record in the CSV file and a column for each field in the record.

Overall, this code imports data from a CSV file into a Pandas DataFrame, which can be used to perform further analysis and manipulation using Python and Pandas.

Step: 2

```
# print dataframe

print(dataframe.head())
```

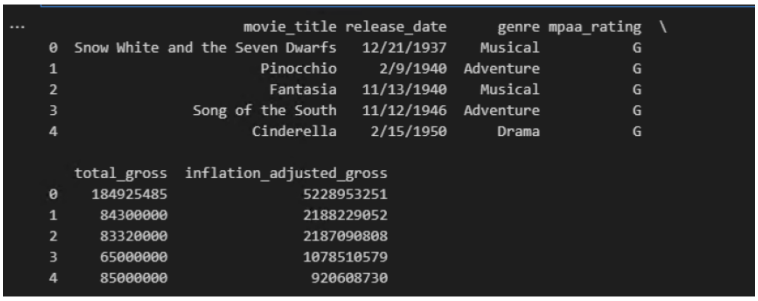
The code you provided is using the Pandas library to print the first 5 rows of the 'dataframe' object that was created in the previous code block.

The 'head()' function is a method provided by Pandas that returns the first n rows of a DataFrame. In this case, the code is using the default value of n=5, so the first 5 rows of the DataFrame are printed to the console.

The 'print()' function is a built-in Python function that outputs the argument to the console. In this case, the argument is the DataFrame returned by the 'head()' function.

Overall, this code block prints the first 5 rows of the 'dataframe' object to the console, allowing you to quickly inspect the imported data and verify that it was loaded correctly.

Output:



```
...                     movie_title release_date      genre mpaa_rating  \
   0  Snow White and the Seven Dwarfs   12/21/1937    Musical           G
   1                        Pinocchio     2/9/1940  Adventure           G
   2                         Fantasia   11/13/1940    Musical           G
   3                Song of the South   11/12/1946  Adventure           G
   4                       Cinderella    2/15/1950      Drama           G

     total_gross  inflation_adjusted_gross
   0   184925485                5228953251
   1    84300000                2188229052
   2    83320000                2187090808
   3    65000000                1078510579
   4    85000000                 920608730
```

Step:3

```
#to avoid issues with the import when objects are blank or null

dataframe = dataframe.fillna('Unkonwn')
```

The code snippet is filling any null or blank values in a pandas DataFrame with the string "Unknown".

In data analysis, missing data can sometimes be problematic as it can lead to errors or incorrect results in computations. To avoid such issues, it is common to replace any missing or blank values with a default value or a value that makes sense in the context of the data.

Here, the fillna() function is being used to replace any null or blank values in the DataFrame with the string 'Unknown'. The DataFrame object is being modified in place, meaning that the original object is being changed rather than creating a new object. This can be useful when dealing with large datasets to save memory and avoid unnecessary copying of data.

By replacing missing values with the string 'Unknown', the code is essentially creating a new category or label to represent any missing or unknown data, which can be useful for later analysis or visualization.

Step 4:

```
# Connect to SQL server using pyodbc driver

import pyodbc
```

```
# create database connection

conn = pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};"Server=EC2AMAZ-MSOSK73;"Database=Adventureworks2019;"Trusted_Connection=yes;')

cursor = conn.cursor()
```

The code you provided is using the pyodbc library to connect to a SQL Server database using Windows Authentication.

The first line imports the pyodbc library, which provides an interface for connecting to various databases using Python.

The second line creates a database connection using the connect() function provided by pyodbc. The function takes a connection string as an argument, which specifies the database driver, server name, database name, and authentication method. In this case, the connection string specifies the following:

- Driver={ ODBC Driver 17 for SQL Server }This specifies that the SQL Server driver should be used to connect to the database.
- Server= EC2AMAZ-MSOSK73: This specifies the name of the SQL Server instance to connect to.
- Database= Adventureworks2019: This specifies the name of the database to connect to.
- Trusted_Connection=yes: This specifies that Windows Authentication should be used to authenticate the connection.

The third line creates a cursor object, which can be used to execute SQL statements on the database connection. The cursor object is obtained from the connection object using the cursor() method.

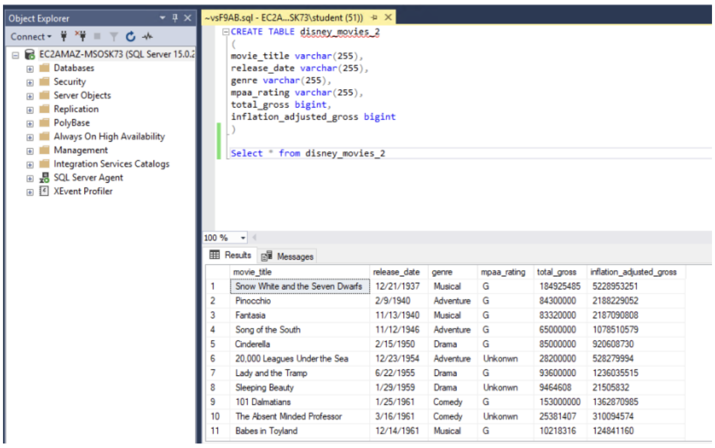
Overall, this code block establishes a connection to a SQL Server database using pyodbc and creates a cursor object that can be used to execute SQL statements on the database.

Step 5:

# create table using connection above(SQL script)

#cursor.execute('''CREATE TABLE disney_movies (movie_title nvarchar(255), release_date DATE, genre nvarchar(50), mpaa_rating nvarchar(25),total_gross int, inflation_adjusted_gross int)''')

Output:



```
CREATE TABLE disney_movies_2
(
movie_title varchar(255),
release_date varchar(255),
genre varchar(255),
mpaa_rating varchar(255),
total_gross bigint,
inflation_adjusted_gross bigint
)

Select * from disney_movies_2
```

	movie_title	release_date	genre	mpaa_rating	total_gross	inflation_adjusted_gross
1	Snow White and the Seven Dwarfs	12/21/1937	Musical	G	184925485	5228953251
2	Pinocchio	2/9/1940	Adventure	G	84300000	2188229052
3	Fantasia	11/13/1940	Musical	G	83320000	2187090808
4	Song of the South	11/12/1946	Adventure	G	65000000	1078510579
5	Cinderella	2/15/1950	Drama	G	85000000	920608730
6	20,000 Leagues Under the Sea	12/23/1954	Adventure	Unkonwn	28200000	528279994
7	Lady and the Tramp	6/22/1955	Drama	G	93600000	1236035515
8	Sleeping Beauty	1/29/1959	Drama	Unkonwn	9464608	21505832
9	101 Dalmatians	1/25/1961	Comedy	G	153000000	1362870985
10	The Absent Minded Professor	3/16/1961	Comedy	Unkonwn	25381407	310094574
11	Babes in Toyland	12/14/1961	Musical	G	10218316	124841160

The code you provided creates a new table named 'disney_movies' in the SQL Server database using the cursor object created in the previous code block.

The 'execute()' method of the cursor object is used to execute a SQL statement on the database connection. In this case, the SQL statement is a CREATE TABLE statement that creates a new table with the following columns:

- movie_title: a string column with a maximum length of 255 characters, used to store the title of each movie
- release_date: a date column, used to store the release date of each movie
- genre: a string column with a maximum length of 50 characters, used to store the genre of each movie
- mpaa_rating: a string column with a maximum length of 25 characters, used to store the MPAA rating of each movie
- total_gross: an integer column, used to store the total gross earnings of each movie
- inflation_adjusted_gross: an integer column, used to store the inflation-adjusted gross earnings of each movie

Overall, this code block creates a new table in the database with the specified columns, which can be used to store the data from the Disney movies CSV file.

Step 6:

```
# import csv data to table in database

for row in dataframe.itertuples():

    cursor.execute('''

                INSERT INTO disney_movies (movie_title, release_date, genre,
mpaa_rating,total_gross, inflation_adjusted_gross)

                VALUES (?,?,?,?,?,?)''',row.movie_title, row.release_date,row.genre,
row.mpaa_rating, row.total_gross, row.inflation_adjusted_gross

                )

conn.commit()
```

The code you provided imports the data from the Disney movies CSV file into the 'disney_movies' table in the SQL Server database using the cursor object created in the previous code block.

The 'itertuples()' method of the dataframe object returns an iterator that yields each row of the dataframe as a named tuple. The 'for' loop iterates over each row in the dataframe, and for each row, the 'execute()' method of the cursor object is used to execute an INSERT INTO statement on the database connection.

The INSERT INTO statement specifies the table name and column names, and uses question marks (?) as placeholders for the actual values that will be inserted into the table. The values to be inserted are passed as arguments to the 'execute()' method, with each argument corresponding to a question mark in the INSERT INTO statement. The values are accessed from the named tuple using dot notation.

Finally, the 'commit()' method of the database connection is called to commit the changes to the database. This ensures that the data is permanently saved to the database.

Overall, this code block inserts each row of data from the CSV file into the 'disney_movies' table in the database, which allows the data to be queried and analyzed using SQL.

Step 7:

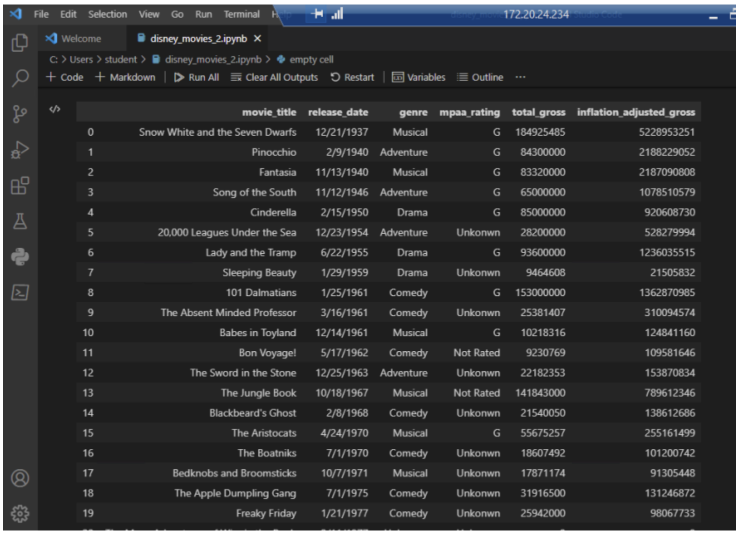
```
# Select the first 25 records from a table
sql = "SELECT  * FROM disney_movies_2"
```

The code you provided executes a SQL query on the 'disney_movies_2' table in the 'movies' database, which selects the first 25 records from the table.

The SQL query is constructed as a string, with the SELECT statement selecting all columns (*) from the table, and the TOP clause limiting the number of rows returned to 25. The 'FROM' keyword specifies the table to be queried and the 'movies.disney_movies' notation specifies the fully-qualified table name, with the 'movies' database and 'disney_movies' table separated by a dot (.).

This code block prepares the SQL query to be executed on the database connection in the next code block. The result of executing the query will be the first 25 rows of the 'disney_movies' table, which will be returned as a pandas dataframe.

Output:



	movie_title	release_date	genre	mpaa_rating	total_gross	inflation_adjusted_gross
0	Snow White and the Seven Dwarfs	12/21/1937	Musical	G	184925485	5228953251
1	Pinocchio	2/9/1940	Adventure	G	84300000	2188229052
2	Fantasia	11/13/1940	Musical	G	83320000	2187090808
3	Song of the South	11/12/1946	Adventure	G	65000000	1078510579
4	Cinderella	2/15/1950	Drama	G	85000000	920608730
5	20,000 Leagues Under the Sea	12/23/1954	Adventure	Unkonwn	28200000	528279994
6	Lady and the Tramp	6/22/1955	Drama	G	93600000	1236035515
7	Sleeping Beauty	1/29/1959	Drama	Unkonwn	9464608	21505832
8	101 Dalmatians	1/25/1961	Comedy	G	153000000	1362870985
9	The Absent Minded Professor	3/16/1961	Comedy	Unkonwn	25381407	310094574
10	Babes in Toyland	12/14/1961	Musical	G	10218316	124841160
11	Bon Voyage!	5/17/1962	Comedy	Not Rated	9230769	109581646
12	The Sword in the Stone	12/25/1963	Adventure	Unkonwn	22182353	153870834
13	The Jungle Book	10/18/1967	Musical	Not Rated	141843000	789612346
14	Blackbeard's Ghost	2/8/1968	Comedy	Unkonwn	21540050	138612686
15	The Aristocats	4/24/1970	Musical	G	55675257	255161499
16	The Boatniks	7/1/1970	Comedy	Unkonwn	18607492	101200742
17	Bedknobs and Broomsticks	10/7/1971	Musical	Unkonwn	17871174	91305448
18	The Apple Dumpling Gang	7/1/1975	Comedy	Unkonwn	31916500	131246872
19	Freaky Friday	1/21/1977	Comedy	Unkonwn	25942000	98067733

Step 8:

# load query results to python dataframe

```
query_results = pd.read_sql(sql, conn)
```

The code you provided uses the pd.read_sql() function from the pandas library to execute the SQL query you constructed in the previous code block and load the result set into a pandas dataframe.

The pd.read_sql() function takes two arguments: the SQL query to execute, and the database connection to use for the query. In this case, the SQL query is represented by the sql variable that you defined earlier, and the connection is represented by the conn variable.

When the pd.read_sql() function is called, it sends the SQL query to the database using the specified connection, and retrieves the results as pandas data frame. The resulting dataframe is assigned to the variable query_results.

At this point, query_results contains the first 25 rows of the disney_movies table in the movies database, as specified by the SQL query.

Step 9:

```
# print data frame
dataframe.head(25)
```

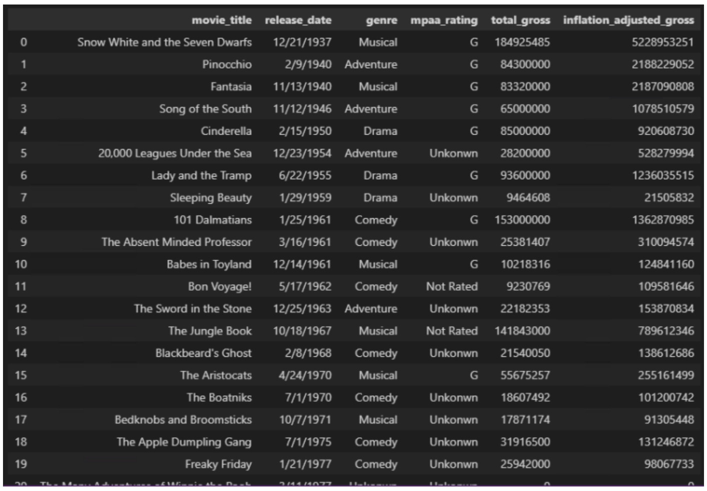
The code dataframe.head(25) is used to display the first 25 rows of a dataframe.

In this code, dataframe is the name of the dataframe that we want to display. The head() function is a method in Pandas library that can be used to return the first n rows of a dataframe, where n is the number specified within the parentheses.

Therefore, dataframe.head(25) will return the first 25 rows of the dataframe. This can be useful for inspecting the data and getting a quick overview of the data structure, column names, and the data types of the columns.

Note that if the dataframe has less than 25 rows, then all the rows will be displayed. Additionally, if no argument is passed to the head() function, it will return the first 5 rows by default.

Output:



	movie_title	release_date	genre	mpaa_rating	total_gross	inflation_adjusted_gross
0	Snow White and the Seven Dwarfs	12/21/1937	Musical	G	184925485	5228953251
1	Pinocchio	2/9/1940	Adventure	G	84300000	2188229052
2	Fantasia	11/13/1940	Musical	G	83320000	2187090808
3	Song of the South	11/12/1946	Adventure	G	65000000	1078510579
4	Cinderella	2/15/1950	Drama	G	85000000	920608730
5	20,000 Leagues Under the Sea	12/23/1954	Adventure	Unkonwn	28200000	528279994
6	Lady and the Tramp	6/22/1955	Drama	G	93600000	1236035515
7	Sleeping Beauty	1/29/1959	Drama	Unkonwn	9464608	21505832
8	101 Dalmatians	1/25/1961	Comedy	G	153000000	1362870985
9	The Absent Minded Professor	3/16/1961	Comedy	Unkonwn	25381407	310094574
10	Babes in Toyland	12/14/1961	Musical	G	10218316	124841160
11	Bon Voyage!	5/17/1962	Comedy	Not Rated	9230769	109581646
12	The Sword in the Stone	12/25/1963	Adventure	Unkonwn	22182353	153870834
13	The Jungle Book	10/18/1967	Musical	Not Rated	141843000	789612346
14	Blackbeard's Ghost	2/8/1968	Comedy	Unkonwn	21540050	138612686
15	The Aristocats	4/24/1970	Musical	G	55675257	255161499
16	The Boatniks	7/1/1970	Comedy	Unkonwn	18607492	101200742
17	Bedknobs and Broomsticks	10/7/1971	Musical	Unkonwn	17871174	91305448
18	The Apple Dumpling Gang	7/1/1975	Comedy	Unkonwn	31916500	131246872
19	Freaky Friday	1/21/1977	Comedy	Unkonwn	25942000	98067733

Reference:

- Anthony, T. C. (2019). Python Programming for Beginners: Learn Python in One Day. Independently published. ISBN-10: 1070694229, ISBN-13: 978-1070694229.

- VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media. ISBN-10: 1491912057, ISBN-13: 978-1491912058.

- Lecture videos from Week-5