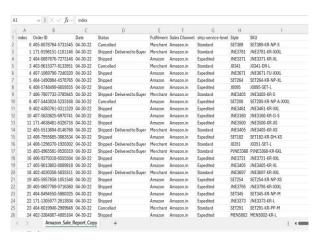
## Extracting Data from CSV file and saving it into another CSV file using Pandas:

```
import pandas as pd
sales_csv =('C:/Users/jyoth/Downloads/Amazon Sale Report.csv/Amazon Sale Report.csv')
sales_df = pd.read_csv(sales_csv)
#print(sales_df)
display(sales_df)
display(sales_df.head(50))
# Sowe Data to Another CSV File
output_csv = 'C:/Users/jyoth/Downloads/Amazon_Sale_Report_Copy.csv'
sales_df.to_csv(output_csv, index=False, encoding="utf-8")
print(f" Data saved to {output_csv}")
```



## Extracting Data from CSV file and saving it into SQL Server Database:

```
import pandas as pd
import pyodbc
from sqlalchemy import create_engine

# Load Data from CSV
sales_csv = 'C:/Users/jyoth/Downloads/Amazon Sale Report.csv/Amazon Sale Report.csv'
sales_df = pd.read_csv(sales_csv, encoding="utf-8", low_memory=False)

# SQL Server Connection Details

db_server = 'JYOTHI\SQLEXPRESS' # SQL Server instance
db_name = 'dataExtraction' # SQL Server database name

# Connection string for SQL Server using pyodbc

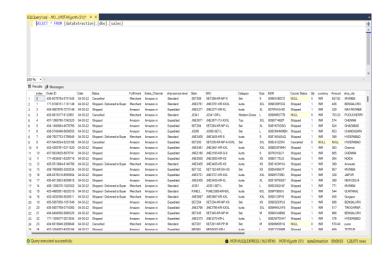
conn_str = f'DRIVER={{SQL Server}}; SERVER={db_server}; DATABASE={db_name}; Trusted_Connection=yes;'

try:
    # Create SQLALchemy engine using pyodbc for SQL Server
    engine = create_engine(f'mssql+pyodbc://fodbc_connect=(conn_str)')

# Connect to SQL Server and insert the DataFrame

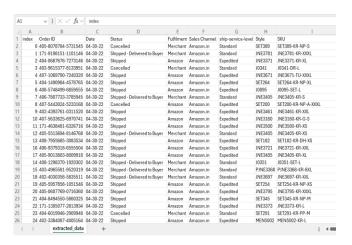
with engine.connect() as conn:
    sales_df.to_sql('sales', conn, if_exists='replace', index=False)
    print("Data inserted into SQL Server database successfully!")

except Exception as e:
    print("Firor inserting data into SQL Server: {e}")
```



## Extracting Data from SQL Server database table and saving it as CSV file:

### Result Set:



# Extracting a result set by querying the table and saving the result set as CSV file:

```
import pyodbc
import pandas as pd

# SQL Server connection details
server = "YVONITSQLEPRESS"
database = 'databatractain"
driver = '(CODE Driver 17 for SQL Server)' # Ensure you have the right driver installed

# Establish connection with Mindows Authentication
com = pyodbc.connect(f'DELVER-(driver)):#ENVER-(server):DATABASE=(database);Trusted_Connection-yes;')

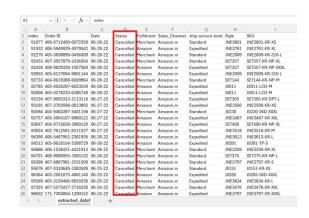
# SQL Query to extract data
query = ""Solb.chamil = 'Amazon.in'
ADD Sales Chammil = 'Amazon.in'
ODIT BY
Date DESG;""

# Execute query and load data into a Pandas DataFrame
df = pd.read_sql(query, comn)

# Some extracted data to a CSV file
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/extracted_datal.csv", indexsFalse)

# Class the connection
conn.close()
print("Data successfully extracted and saved to extracted_data.csv")
```

### Filtered result set:



Extracting data from Web API and saving the data as a table in SQL Server database and as a CSV file:

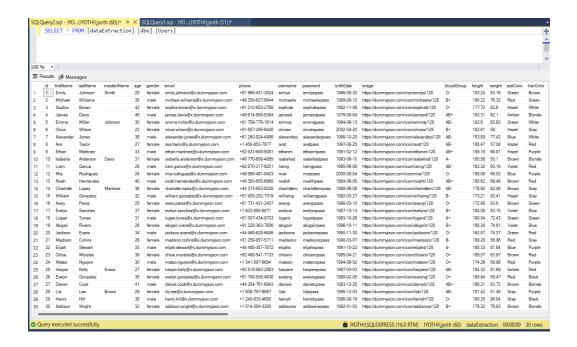
```
import requests
import pyodbc
       import pandas as pd
     # Database connection details
server = 'JYOTHI\\SQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}'
       # Establish SQL Server Connection
     conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')
cursor = conn.cursor()
     # Fetch full data from API
api_url = "https://dummyjson.com/users"
response = requests.get(api_url)
     if response.status_code == 200:
                              users = response.json()["users"] # Extract full users list
                         print("Failed to fetch data from API")
# Create Table for All Fields (Dynamically)

cursor.execute("""

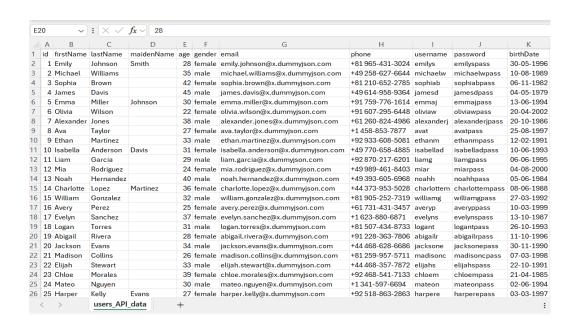
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Users' AND xtype='U')

CREATE TABLE Users (
    id INT PRIMARY KEY,
    firstName NVARCHAR(100),
    maidenName NVARCHAR(100),
    age INT,
    gender NVARCHAR(100),
    email NVARCHAR(20),
    email NVARCHAR(255),
    phone NVARCHAR(30),
    password NVARCHAR(10),
    password NVARCHAR(255),
    birthDate DATE,
    image NVARCHAR(25),
    bloodforoup NVARCHAR(10),
    height FLOAT,
    weight FLOAT,
    weight FLOAT,
    eyeColor NVARCHAR(50),
    hairType NVARCHAR(50),
    hairType NVARCHAR(50),
    ip NVARCHAR(30),
    sacAddress NVARCHAR(50),
    state NVARCHAR(100),
    state Code NVARCHAR(100),
    stateCode NVARCHAR(100),
    stateCode NVARCHAR(200),
    cardNumber NVARCHAR(200),
    cardType NVARCHAR(50),
    cardType NVARCHAR(50),
    cardType NVARCHAR(50),
    cardType NVARCHAR(200),
    company NVARCHAR(205),
    company NVARC
                           jobTitle NVARCHAR(255),
ein NVARCHAR(50),
ssn NVARCHAR(50),
coin NVARCHAR(50),
wallet NVARCHAR(255),
pathropk NVARCHAR(50)
                              network NVARCHAR(50),
role NVARCHAR(50)
     # Insert Full API Data into SQL Server
for user in users:
    cursor.execute("""
INSER INTO Users (
    id, firstName, lastName, maidenName, age, gender, email, phone,
    username, password, birthDate, image, bloodGroup, height, weight,
    eyeColor, hairColor, hairType, ip, macAddress, university,
    city, state, stateCode, postalCode, country, cardNumber,
    cardType, currency, iban, company, jobTitle, ein, ssn, coin,
    wallet, network, role
)
                               user["ad"], user["arsName"], user["alsName"], user["age"],
user["agede"], user["amil"], user["ahme"], user["age"],
user["agede"], user["amil"], user["ahme"], user["assword"],
user["birthDate"], user["ain"], user["birodoFoup"], user["height"],
user["acolon"], user["hain"]["colon"], user["ain"]["sype"], user["bir]],
user["address"], user["ain"], user["address"]["cit"],
user["address"]["stateCode"],
user["address"]["state"], user["address"]["condhame]["cardType"],
user["address"]["country"], user["bank"]["cardMumber"], user["bank"]["cardType"],
user["company"]["title"],
user["company"]["title"],
user["company"]["title"],
user["company"]["title"],
user["company"]["title"],
user["company"]["name"],
user["name"],
use
  conn.commit()
       # Step 4: Fetch Data and Save to CSV
df = pd.read_sql("SELECT * FROM Users", conn)
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/users_API_data.csv", index=False)
       conn.close()
       print("Data successfully extracted and stored in SQL Server & CSV!")
Data successfully extracted and stored in SQL Server & CSV!
```

### Result Set saved in SQL Server database:



### Result Set saved in CSV File:



Extracting data from a web page through Web scraping and saving the data as a table in SQL Server database and as a CSV file:

```
from bs4 import BeautifulSoup
 import pandas as pd
import numpy as np
# Function to extract Product Title
def get_title(soup):
     try:
    # Outer Tag Object
    title = soup.find("span", attrs=("id":'productTitle'))
    .
      return title_string
     try:
    rating = soup-find("i", attrs={'class':'a-icon a-icon-star a-star-4-5'}).string.strip()
     except AttributeError:
    try:
        rating = soup.find("span", attrs={'class':'a-icon-alt'}).string.strip()
           except:
rating = ""
   return rating
# Function to extract Number of User Reviews
def get_review_count(soup):
     get_reviem_count_sorp,.
try:
    review_count = soup.find("span", attrs={'id':'acrCustomerReviewText'}).string.strip()
      except AttributeError:
      return review_count
# Function to extract Availability Status
def get_availability(soup):
      get_available.rry;
try:
    available = soup.find("div", attrs=('id':'availability'))
    available = available.find("span").string.strip()
     except AttributeError:
   available = "Not Available"
      return available
if name == ' main ':
      # dad your user agent

HEADERS = (('User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.
      # The webpage URL
URL = "https://www.amazon.ca/s?k=Electronic+devices&crid=18YOR7TERHJOJ&sprefix=electronic+devices%2Caps%2C195&ref=nb_sb_
      # HTTP Request
webpage = requests.get(URL, headers=HEADERS)
      # Soup Object containing all data
soup = BeautifulSoup(webpage.content, "html.parser")
     # Fetch Links as List of Tag Objects
links = soup.find_all("a", attrs={'class':'a-link-normal s-line-clamp-4 s-link-style a-text-normal'})
     d = {"title":[], "rating":[], "reviews":[],"availability":[]}
      # Loop for extracting product details from each link
     for link in links_list:
    new_webpage = requests.get("https://www.amazon.com" + link, headers=HEADERS)
           new_soup = BeautifulSoup(new_webpage.content, "html.parser")
           # Function calls to display all necessary product information
d['title'].append(get_title(new_soup))
d('rating').append(get_rating(new_soup))
d('reviews').append(get_review_court(new_soup))
d['availability'].append(get_availability(new_soup))
     amazon_df = pd.DataFrame.from_dict(d)
     amazon_df['title'].replace('', np.nan, inplace=True)
amazon_df = amazon_df.dropna(subset=['title'])
amazon_df.to_csv("amazon_data.csv", header=True, index=False)
amazon df
```

### Result Set:

	A	В	С	D
1	title	rating	reviews	availability
2	Vasco V4 Language Translator Device   108 La	3.8 out of 5 stars	1,241 ratings	Only 15 left in stock - order soon.
3	Language Translator Device 144 Languages an	3.5 out of 5 stars	287 ratings	In Stock
4	Apple AirTag 4 Pack	4.7 out of 5 stars	136,677 ratings	Not Available
5	VICHYIE 20 in 1 Multifunctional Cleaner Kit for I	4.3 out of 5 stars	1,383 ratings	In Stock
6	Electronic Devices (Conventional Current Vers	4.3 out of 5 stars	166 ratings	Not Available
7	Electronic Devices: Conventional Current Vers	4.6 out of 5 stars	26 ratings	Not Available
8	4 Port USB Charging Station for Multiple Device	4.3 out of 5 stars	1,269 ratings	Not Available
9	Roku Express 4K+   Streaming Player HD/4K/H	4.6 out of 5 stars	2,440 ratings	Not Available
0	Amazon Echo Pop (newest model), Our smalle	4.7 out of 5 stars	77,241 ratings	Not Available
1	70M Laser Measure Device, Mileseey Digital La	4.4 out of 5 stars	1,708 ratings	Not Available
2	Solid-State Electronic Devices: An Introduction	3.7 out of 5 stars	4 ratings	In Stock
3	Electronic Devices (Conventional Current Vers	4.3 out of 5 stars	90 ratings	Not Available
4	Offline Language Voice Translator Device W09	3.9 out of 5 stars	155 ratings	In Stock
5	M9 AI Language Translator Earbuds Open-Ear I	4.0 out of 5 stars	41 ratings	In Stock
6	Translator Earbuds, 3-in-1 Language Translato	2.0 out of 5 stars	1 rating	In Stock
7	Language Translator Device No WiFi Needed Ir	3.8 out of 5 stars	77 ratings	In Stock
8	SooPii 60W 6-Port Charging Station for Multiple	4.7 out of 5 stars	4,414 ratings	In Stock
9	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
0	Unitek Fast Charging Station for Multiple Device	4.3 out of 5 stars	235 ratings	In Stock
1	Skylight Calendar: 15 inch Digital Calendar & C	4.4 out of 5 stars	1,769 ratings	In Stock
2	Amazon Echo (newest model), Alexa speaker v	4.7 out of 5 stars	149,562 ratings	Not Available
3	Power-Bank-Portable-Charger - 40000mAh Po	4.3 out of 5 stars	165 ratings	Not Available
4	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
5		4.4 out of 5 stars	712 ratings	Not Available
6	320W Charging Station for Multiple Devices, Zo	4.6 out of 5 stars	430 ratings	In Stock