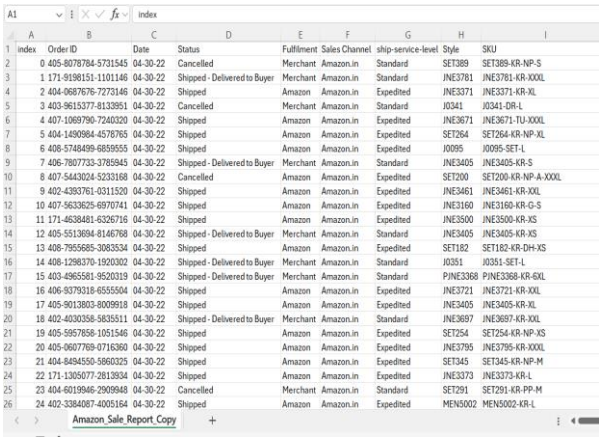
Extracting Data from CSV file and saving it into another CSV file using Pandas:

```
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))

# Save 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}")
```

```
C:\Users\jyoth\AppData\Local\Temp\ipykernel_8864\3919254776.py:3: DtypeWarning: Columns (23) have mixed types. Specify dtype option on import or set low_memory=False.
  sales_df = pd.read_csv(sales_csv)
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	sh
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	INR	647.62	M
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	INR	406.00	BENG

Result Set:

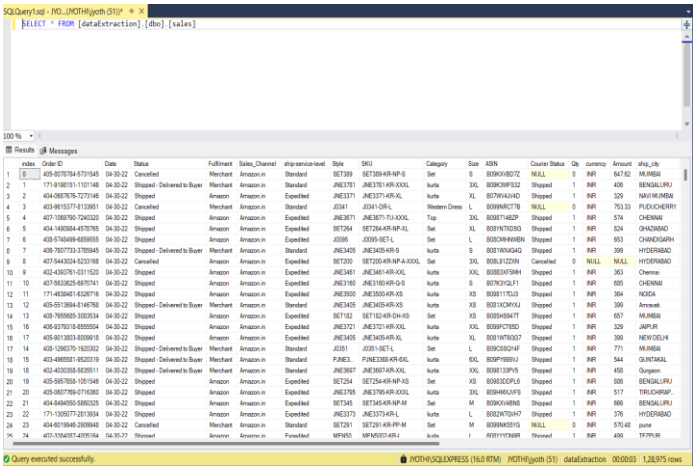


index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU
0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S
1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL
2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL
3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L
4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL
5	404-1490984-4576765	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET264	SET264-KR-NP-XL
6	408-5748499-6859555	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0095	J0095-SET-L
7	406-7807733-3785945	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-S
8	407-5443024-5233168	04-30-22	Cancelled	Amazon	Amazon.in	Expedited	SET200	SET200-KR-NP-A-XXXL
9	402-4393761-0311520	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3461	JNE3461-KR-XXL
10	407-5633625-6970741	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3160	JNE3160-KR-G-S
11	171-4638481-6326716	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-XS
12	405-5513694-8146768	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-XS
13	408-7955685-3083534	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET182	SET182-KR-DH-XS
14	408-1298370-1920302	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	J0351	J0351-SET-L
15	403-4965581-9520319	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	PJNE3368	PJNE3368-KR-6XL
16	406-9379318-6555504	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3721	JNE3721-KR-XXL
17	405-9013803-8009918	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3405	JNE3405-KR-XL
18	402-4030358-5835511	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3697	JNE3697-KR-XXL
19	405-5957858-1051546	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET254	SET254-KR-NP-XS
20	405-0607769-0716360	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3795	JNE3795-KR-XXXL
21	404-8494550-5860325	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET345	SET345-KR-NP-M
22	171-1305077-2813934	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3373	JNE3373-KR-L
23	404-6019946-2909948	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET291	SET291-KR-PP-M
24	402-3384087-4005164	04-30-22	Shipped	Amazon	Amazon.in	Expedited	MEN5002	MEN5002-KR-L

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:///?odbc_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(f"Error inserting data into SQL Server: {e}")
```

Result Set:



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

```python
import pyodbc
import pandas as pd

# SQL Server connection details
server = 'JYOTHI\SQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}'

# Establish connection with Windows Authentication
conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')

# SQL Query to extract data
query = "SELECT * FROM sales"

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

# Save extracted data to a CSV file
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/extracted_data.csv", index=False)

# Close the connection
conn.close()

print("Data successfully extracted and saved to extracted_data.csv")
```

```
C:\Users\jyoth\AppData\Local\Temp\ipykernel_11620\2801874350.py:16: UserWarning: pandas only supports SQLAlchemy connectabl
(engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consid
using SQLAlchemy.
  df = pd.read_sql(query, conn)
Data successfully extracted and saved to extracted_data.csv
```

Result Set:

index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU
0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S
1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL
2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL
3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L
4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL
5	404-1490984-4578765	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET264	SET264-KR-NP-XL
6	408-5748499-6859555	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0095	J0095-SET-L
7	406-7807733-3785945	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-S
8	407-5443024-5233168	04-30-22	Cancelled	Amazon	Amazon.in	Expedited	SET200	SET200-KR-NP-A-XXXL
9	402-4393761-0311520	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3461	JNE3461-KR-XXL
10	407-5633625-6970741	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3160	JNE3160-KR-G-S
11	171-4638481-6326716	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-XS
12	405-5513694-8146768	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-XS
13	408-7955685-3083534	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET182	SET182-KR-DH-XS
14	406-9379318-6555504	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3721	JNE3721-KR-XXL
15	405-9013803-8009918	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3405	JNE3405-KR-XL
16	408-1298370-1920302	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	J0351	J0351-SET-L
17	403-4965581-9520319	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	PJNE3368	PJNE3368-KR-6XL
18	402-4030358-5835511	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3697	JNE3697-KR-XXL
19	405-5957858-1051546	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET254	SET254-KR-NP-XS
20	405-0607769-0716360	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3795	JNE3795-KR-XXXL
21	404-8494550-5860325	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET345	SET345-KR-NP-M
22	171-1305077-2813934	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3373	JNE3373-KR-L
23	404-6019946-2909948	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET291	SET291-KR-PP-M
24	402-3384087-4005164	04-30-22	Shipped	Amazon	Amazon.in	Expedited	MEN5002	MEN5002-KR-L

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

```python
import pyodbc
import pandas as pd

# SQL Server connection details
server = 'JYOTHI\SQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}'  # Ensure you have the right driver installed

# Establish connection with Windows Authentication
conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')

# SQL Query to extract data
query = """Select * From dbo.sales WHERE
    Status = 'Cancelled'
    AND Sales_Channel  = 'Amazon.in'
    AND ship_city = 'MUMBAI'
    ORDER BY
    Date DESC;"""

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

# Save extracted data to a CSV file
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/extracted_data1.csv", index=False)

# Close the connection
conn.close()

print("Data successfully extracted and saved to extracted_data.csv")
```

```
C:\Users\jyoth\AppData\Local\Temp\ipykernel_11620\2887544763.py:21: UserWarning: pandas only supports SQLAlchemy connectable
(engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please conside
using SQLAlchemy.
  df = pd.read_sql(query, conn)
Data successfully extracted and saved to extracted_data.csv
```

Filtered result set :

index	Order ID	Date	Status	Fulfilment	Sales_Channel	ship-service-level	Style	SKU
91877	406-8712400-6972359	06-28-22	Cancelled	Merchant	Amazon.in	Standard	JNE3801	JNE3801-KR-XS
91932	406-5949929-4979542	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3761	JNE3761-KR-XL
92270	405-3838899-0456309	06-28-22	Cancelled	Merchant	Amazon.in	Standard	JNE2009	JNE2009-KR-310-L
92453	407-3557879-2436304	06-28-22	Cancelled	Merchant	Amazon.in	Standard	SET357	SET357-KR-NP-XL
92458	408-9829339-1007560	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	SET357	SET357-KR-NP-XXXL
92655	405-6127694-9861144	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	JNE2009	JNE2009-KR-310-L
92733	404-0619388-6009964	06-28-22	Cancelled	Merchant	Amazon.in	Standard	SET144	SET144-KR-NP-M
92785	405-4034297-6023559	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	J0011	J0011-LCD-M
92806	405-0278233-6386749	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	J0011	J0011-LCD-M
93150	407-9693313-2113110	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	SET305	SET305-KR-DPT-L
93191	407-3703598-0813902	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-XS
93594	404-9483397-5401156	06-27-22	Cancelled	Merchant	Amazon.in	Standard	J0230	J0230-SKD-XXXL
93757	405-5954187-9969122	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3487	JNE3487-KR-XXL
93847	404-0715658-3889120	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	SET400	SET400-KR-NP-XL
93924	402-7612561-6511527	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3518	JNE3518-KR-M
94390	406-5487952-2361935	06-26-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3613	JNE3613-KR-L
94512	405-8618104-5398729	06-26-22	Cancelled	Amazon	Amazon.in	Expedited	J0391	J0391-TP-S
94866	406-1538451-4421913	06-26-22	Cancelled	Merchant	Amazon.in	Standard	JNE3359	JNE3359-KR-XL
94701	408-9086955-3385132	06-26-22	Cancelled	Merchant	Amazon.in	Standard	SET375	SET375-KR-NP-L
95308	407-8887961-2531509	06-26-22	Cancelled	Merchant	Amazon.in	Standard	JNE3797	JNE3797-KR-S
95876	407-0310640-3383509	06-25-22	Cancelled	Merchant	Amazon.in	Standard	J0151	J0151-KR-XS
96364	405-2581875-4881140	06-25-22	Cancelled	Amazon	Amazon.in	Expedited	J0285	J0285-SKD-XXXL
95589	405-3150466-9020338	06-25-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3634	JNE3634-KR-L
97325	407-5573427-2716328	06-24-22	Cancelled	Merchant	Amazon.in	Standard	JNE3476	JNE3476-KR-XXL
96632	171-7393804-1209152	06-24-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3797	JNE3797-KR-XXXL

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

```python
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
else:
    print("Failed to fetch data from API")
    exit()
```

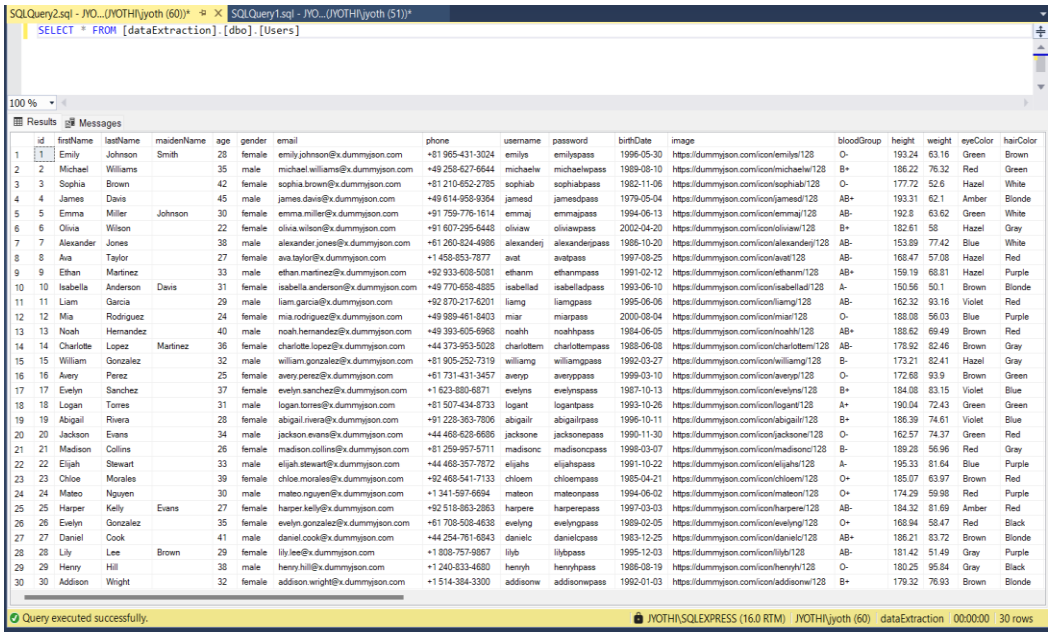
```python
# Create Table for All Fields (Dynamically)
cursor.execute("""
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Users' AND xtype='U')
CREATE TABLE Users (
    id INT PRIMARY KEY,
    firstName NVARCHAR(100),
    lastName NVARCHAR(100),
    maidenName NVARCHAR(100),
    age INT,
    gender NVARCHAR(20),
    email NVARCHAR(255),
    phone NVARCHAR(50),
    username NVARCHAR(100),
    password NVARCHAR(255),
    birthDate DATE,
    image NVARCHAR(255),
    bloodGroup NVARCHAR(10),
    height FLOAT,
    weight FLOAT,
    eyeColor NVARCHAR(50),
    hairColor NVARCHAR(50),
    hairType NVARCHAR(50),
    ip NVARCHAR(50),
    macAddress NVARCHAR(50),
    university NVARCHAR(255),
    city NVARCHAR(100),
    state NVARCHAR(100),
    stateCode NVARCHAR(10),
    postalCode NVARCHAR(20),
    country NVARCHAR(100),
    cardNumber NVARCHAR(50),
    cardType NVARCHAR(50),
    currency NVARCHAR(10),
    iban NVARCHAR(50),
    company NVARCHAR(255),
```

```python
    jobTitle NVARCHAR(255),
    ein NVARCHAR(50),
    ssn NVARCHAR(50),
    coin NVARCHAR(50),
    wallet NVARCHAR(255),
    network NVARCHAR(50),
    role NVARCHAR(50)
)
""")
conn.commit()

# Insert Full API Data into SQL Server
for user in users:
    cursor.execute("""
    INSERT 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
    )
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    """, (
        user["id"], user["firstName"], user["lastName"], user["maidenName"], user["age"],
        user["gender"], user["email"], user["phone"], user["username"], user["password"],
        user["birthDate"], user["image"], user["bloodGroup"], user["height"], user["weight"],
        user["eyeColor"], user["hair"]["color"], user["hair"]["type"], user["ip"],
        user["macAddress"], user["university"], user["address"]["city"],
        user["address"]["state"], user["address"]["stateCode"], user["address"]["postalCode"],
        user["address"]["country"], user["bank"]["cardNumber"], user["bank"]["cardType"],
        user["bank"]["currency"], user["bank"]["iban"], user["company"]["name"],
        user["company"]["title"], user["ein"], user["ssn"],
        user["crypto"]["coin"], user["crypto"]["wallet"], user["crypto"]["network"],
        user["role"]
    ))
```

```python
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)

# Close connection
conn.close()

print("Data successfully extracted and stored in SQL Server & CSV!")
```

```
Data successfully extracted and stored in SQL Server & CSV!
C:\Users\jyoth\AppData\Local\Temp\ipykernel_36224\3776510391.py:99: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  df = pd.read_sql("SELECT * FROM Users", conn)
```

Result Set saved in SQL Server database:

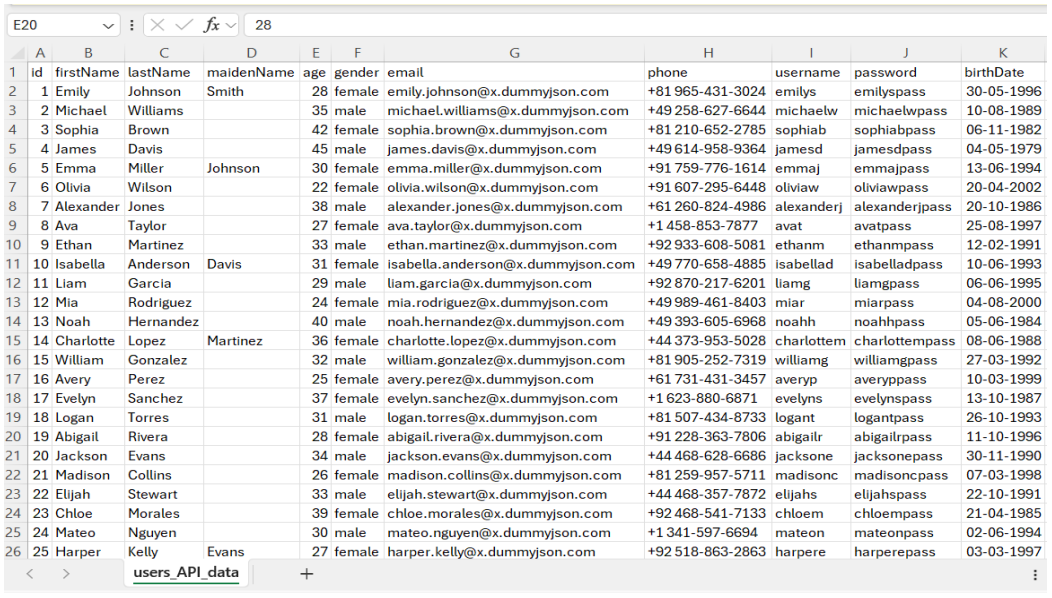


```
SELECT * FROM [dataExtraction].[dbo].[Users]
```

	id	firstName	lastName	maidenName	age	gender	email	phone	username	password	birthDate	image	bloodGroup	height	weight	eyeColor	hairColor
1	1	Emily	Johnson	Smith	28	female	emily.johnson@x.dummyjson.com	+81 965-431-3024	emilys	emilyspass	1996-05-30	https://dummyjson.com/icon/emilys/128	O-	193.24	63.16	Green	Brown
2	2	Michael	Williams		35	male	michael.williams@x.dummyjson.com	+49 258-627-6644	michaelw	michaelwpass	1989-08-10	https://dummyjson.com/icon/michaelw/128	B+	186.22	76.32	Red	Green
3	3	Sophia	Brown		42	female	sophia.brown@x.dummyjson.com	+81 210-652-2785	sophiab	sophiabpass	1982-11-06	https://dummyjson.com/icon/sophiab/128	O-	177.72	52.6	Hazel	White
4	4	James	Davis		45	male	james.davis@x.dummyjson.com	+49 614-958-9364	jamesd	jamesdpass	1979-05-04	https://dummyjson.com/icon/jamesd/128	AB+	193.31	62.1	Amber	Blonde
5	5	Emma	Miller	Johnson	30	female	emma.miller@x.dummyjson.com	+91 759-776-1614	emmaj	emmajpass	1994-06-13	https://dummyjson.com/icon/emmaj/128	AB-	192.8	63.62	Green	White
6	6	Olivia	Wilson		22	female	olivia.wilson@x.dummyjson.com	+91 607-295-6448	oliviaw	oliviawpass	2002-04-20	https://dummyjson.com/icon/oliviaw/128	B+	182.61	58	Hazel	Gray
7	7	Alexander	Jones		38	male	alexander.jones@x.dummyjson.com	+61 260-824-4986	alexanderj	alexanderjpass	1986-10-20	https://dummyjson.com/icon/alexanderj/128	AB-	153.89	77.42	Blue	White
8	8	Ava	Taylor		27	female	ava.taylor@x.dummyjson.com	+1 458-853-7877	avat	avatpass	1997-08-25	https://dummyjson.com/icon/avat/128	AB-	168.47	57.08	Hazel	Red
9	9	Ethan	Martinez		33	male	ethan.martinez@x.dummyjson.com	+92 933-608-5081	ethanm	ethanmpass	1991-02-12	https://dummyjson.com/icon/ethanm/128	AB+	159.19	68.81	Hazel	Purple
10	10	Isabella	Anderson	Davis	31	female	isabella.anderson@x.dummyjson.com	+49 770-658-4885	isabellad	isabelladpass	1993-06-10	https://dummyjson.com/icon/isabellad/128	A-	150.56	50.1	Brown	Blonde
11	11	Liam	Garcia		29	male	liam.garcia@x.dummyjson.com	+92 870-217-6201	liamg	liamgpass	1995-06-06	https://dummyjson.com/icon/liamg/128	AB-	162.32	93.16	Violet	Red
12	12	Mia	Rodriguez		24	female	mia.rodriguez@x.dummyjson.com	+49 989-461-8403	miar	miarpass	2000-08-04	https://dummyjson.com/icon/miar/128	O-	188.08	56.03	Blue	Purple
13	13	Noah	Hernandez		40	male	noah.hernandez@x.dummyjson.com	+49 393-605-6968	noahh	noahhpass	1984-06-05	https://dummyjson.com/icon/noahh/128	AB+	188.62	69.49	Brown	Red
14	14	Charlotte	Lopez	Martinez	36	female	charlotte.lopez@x.dummyjson.com	+44 373-953-5028	charlottem	charlottempass	1988-06-08	https://dummyjson.com/icon/charlottem/128	AB-	178.92	82.46	Brown	Gray
15	15	William	Gonzalez		32	male	william.gonzalez@x.dummyjson.com	+81 905-252-7319	williamg	williamgpass	1992-03-27	https://dummyjson.com/icon/williamg/128	B-	173.21	82.41	Hazel	Gray
16	16	Avery	Perez		25	female	avery.perez@x.dummyjson.com	+61 731-431-3457	averyp	averyppass	1999-03-10	https://dummyjson.com/icon/averyp/128	O-	172.68	93.9	Brown	Green
17	17	Evelyn	Sanchez		37	female	evelyn.sanchez@x.dummyjson.com	+1 623-880-6871	evelyns	evelynspass	1987-10-13	https://dummyjson.com/icon/evelyns/128	B+	184.08	83.15	Violet	Blue
18	18	Logan	Torres		31	male	logan.torres@x.dummyjson.com	+81 507-434-8733	logant	logantpass	1993-10-26	https://dummyjson.com/icon/logant/128	A+	190.04	72.43	Green	Green
19	19	Abigail	Rivera		28	female	abigail.rivera@x.dummyjson.com	+91 228-363-7806	abigailr	abigailrpass	1996-10-11	https://dummyjson.com/icon/abigailr/128	B+	186.39	74.61	Violet	Blue
20	20	Jackson	Evans		34	male	jackson.evans@x.dummyjson.com	+44 468-628-6686	jacksone	jacksonepass	1990-11-30	https://dummyjson.com/icon/jacksone/128	O-	162.57	74.37	Green	Red
21	21	Madison	Collins		26	female	madison.collins@x.dummyjson.com	+81 259-957-5711	madisonc	madisoncpass	1998-03-07	https://dummyjson.com/icon/madisonc/128	B-	189.28	56.96	Red	Gray
22	22	Elijah	Stewart		33	male	elijah.stewart@x.dummyjson.com	+44 468-357-7872	elijahs	elijahspass	1991-10-22	https://dummyjson.com/icon/elijahs/128	A-	195.33	81.64	Blue	Purple
23	23	Chloe	Morales		39	female	chloe.morales@x.dummyjson.com	+92 468-541-7133	chloem	chloempass	1985-04-21	https://dummyjson.com/icon/chloem/128	O+	185.07	63.97	Brown	Red
24	24	Mateo	Nguyen		30	male	mateo.nguyen@x.dummyjson.com	+1 341-597-6694	mateon	mateonpass	1994-06-02	https://dummyjson.com/icon/mateon/128	O+	174.29	59.98	Red	Purple
25	25	Harper	Kelly	Evans	27	female	harper.kelly@x.dummyjson.com	+92 518-863-2863	harpere	harperepass	1997-03-03	https://dummyjson.com/icon/harpere/128	AB-	184.32	81.69	Amber	Red
26	26	Evelyn	Gonzalez		35	female	evelyn.gonzalez@x.dummyjson.com	+61 708-508-4638	evelyng	evelyngpass	1989-02-05	https://dummyjson.com/icon/evelyng/128	O+	168.94	58.47	Red	Black
27	27	Daniel	Cook		41	male	daniel.cook@x.dummyjson.com	+44 254-761-6843	danielc	danielcpass	1983-12-25	https://dummyjson.com/icon/danielc/128	AB+	186.21	83.72	Brown	Blonde
28	28	Lily	Lee	Brown	29	female	lily.lee@x.dummyjson.com	+1 808-757-9867	lilyb	lilybpass	1995-12-03	https://dummyjson.com/icon/lilyb/128	AB-	181.42	51.49	Gray	Purple
29	29	Henry	Hill		38	male	henry.hill@x.dummyjson.com	+1 240-833-4680	henryh	henryhpass	1986-08-19	https://dummyjson.com/icon/henryh/128	O-	180.25	95.84	Gray	Black
30	30	Addison	Wright		32	female	addison.wright@x.dummyjson.com	+1 514-384-3300	addisonw	addisonwpass	1992-01-03	https://dummyjson.com/icon/addisonw/128	B+	179.32	76.93	Brown	Blonde

Query executed successfully. JYOTHI\SQLEXPRESS (16.0 RTM) JYOTHI\jyoth (60) dataExtraction 00:00:00 30 rows

Result Set saved in CSV File:



E20 | 28

	A	B	C	D	E	F	G	H	I	J	K
1	id	firstName	lastName	maidenName	age	gender	email	phone	username	password	birthDate
2	1	Emily	Johnson	Smith	28	female	emily.johnson@x.dummyjson.com	+81 965-431-3024	emilys	emilyspass	30-05-1996
3	2	Michael	Williams		35	male	michael.williams@x.dummyjson.com	+49 258-627-6644	michaelw	michaelwpass	10-08-1989
4	3	Sophia	Brown		42	female	sophia.brown@x.dummyjson.com	+81 210-652-2785	sophiab	sophiabpass	06-11-1982
5	4	James	Davis		45	male	james.davis@x.dummyjson.com	+49 614-958-9364	jamesd	jamesdpass	04-05-1979
6	5	Emma	Miller	Johnson	30	female	emma.miller@x.dummyjson.com	+91 759-776-1614	emmaj	emmajpass	13-06-1994
7	6	Olivia	Wilson		22	female	olivia.wilson@x.dummyjson.com	+91 607-295-6448	oliviaw	oliviawpass	20-04-2002
8	7	Alexander	Jones		38	male	alexander.jones@x.dummyjson.com	+61 260-824-4986	alexanderj	alexanderjpass	20-10-1986
9	8	Ava	Taylor		27	female	ava.taylor@x.dummyjson.com	+1 458-853-7877	avat	avatpass	25-08-1997
10	9	Ethan	Martinez		33	male	ethan.martinez@x.dummyjson.com	+92 933-608-5081	ethanm	ethanmpass	12-02-1991
11	10	Isabella	Anderson	Davis	31	female	isabella.anderson@x.dummyjson.com	+49 770-658-4885	isabellad	isabelladpass	10-06-1993
12	11	Liam	Garcia		29	male	liam.garcia@x.dummyjson.com	+92 870-217-6201	liamg	liamgpass	06-06-1995
13	12	Mia	Rodriguez		24	female	mia.rodriguez@x.dummyjson.com	+49 989-461-8403	miar	miarpass	04-08-2000
14	13	Noah	Hernandez		40	male	noah.hernandez@x.dummyjson.com	+49 393-605-6968	noahh	noahhpass	05-06-1984
15	14	Charlotte	Lopez	Martinez	36	female	charlotte.lopez@x.dummyjson.com	+44 373-953-5028	charlottem	charlottempass	08-06-1988
16	15	William	Gonzalez		32	male	william.gonzalez@x.dummyjson.com	+81 905-252-7319	williamg	williamgpass	27-03-1992
17	16	Avery	Perez		25	female	avery.perez@x.dummyjson.com	+61 731-431-3457	averyp	averyppass	10-03-1999
18	17	Evelyn	Sanchez		37	female	evelyn.sanchez@x.dummyjson.com	+1 623-880-6871	evelyns	evelynspass	13-10-1987
19	18	Logan	Torres		31	male	logan.torres@x.dummyjson.com	+81 507-434-8733	logant	logantpass	26-10-1993
20	19	Abigail	Rivera		28	female	abigail.rivera@x.dummyjson.com	+91 228-363-7806	abigailr	abigailrpass	11-10-1996
21	20	Jackson	Evans		34	male	jackson.evans@x.dummyjson.com	+44 468-628-6686	jacksone	jacksonepass	30-11-1990
22	21	Madison	Collins		26	female	madison.collins@x.dummyjson.com	+81 259-957-5711	madisonc	madisoncpass	07-03-1998
23	22	Elijah	Stewart		33	male	elijah.stewart@x.dummyjson.com	+44 468-357-7872	elijahs	elijahspass	22-10-1991
24	23	Chloe	Morales		39	female	chloe.morales@x.dummyjson.com	+92 468-541-7133	chloem	chloempass	21-04-1985
25	24	Mateo	Nguyen		30	male	mateo.nguyen@x.dummyjson.com	+1 341-597-6694	mateon	mateonpass	02-06-1994
26	25	Harper	Kelly	Evans	27	female	harper.kelly@x.dummyjson.com	+92 518-863-2863	harpere	harperepass	03-03-1997

users_API_data

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 requests
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'})

        # Inner NavigatableString Object
        title_value = title.text

        # Title as a string value
        title_string = title_value.strip()

    except AttributeError:
        title_string = ""

    return title_string


# Function to extract Product Rating
def get_rating(soup):

    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):
    try:
        review_count = soup.find("span", attrs={'id':'acrCustomerReviewText'}).string.strip()

    except AttributeError:
        review_count = ""

    return review_count

# Function to extract Availability Status
def get_availability(soup):
    try:
        available = soup.find("div", attrs={'id':'availability'})
        available = available.find("span").string.strip()

    except AttributeError:
        available = "Not Available"

    return available
```

```
if __name__ == '__main__':

    # add 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=18Y0R7TERHJ0J&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'})

    # Store the links
    links_list = []

    # Loop for extracting links from Tag Objects
    for link in links:
            links_list.append(link.get('href'))

    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_count(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:

A2 | Vasco V4 Language Translator Device | 108 Languages | Free Lifetime Internet in Almost 200 Cou

	A	B	C	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
10	Amazon Echo Pop (newest model), Our smalle	4.7 out of 5 stars	77,241 ratings	Not Available
11	70M Laser Measure Device, Mileseey Digital La	4.4 out of 5 stars	1,708 ratings	Not Available
12	Solid-State Electronic Devices: An Introductior	3.7 out of 5 stars	4 ratings	In Stock
13	Electronic Devices (Conventional Current Vers	4.3 out of 5 stars	90 ratings	Not Available
14	Offline Language Voice Translator Device W09	3.9 out of 5 stars	155 ratings	In Stock
15	M9 AI Language Translator Earbuds Open-Ear I	4.0 out of 5 stars	41 ratings	In Stock
16	Translator Earbuds, 3-in-1 Language Translato	2.0 out of 5 stars	1 rating	In Stock
17	Language Translator Device No WiFi Needed Ir	3.8 out of 5 stars	77 ratings	In Stock
18	SooPii 60W 6-Port Charging Station for Multiple	4.7 out of 5 stars	4,414 ratings	In Stock
19	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
20	Unitek Fast Charging Station for Multiple Devic	4.3 out of 5 stars	235 ratings	In Stock
21	Skylight Calendar: 15 inch Digital Calendar & C	4.4 out of 5 stars	1,769 ratings	In Stock
22	Amazon Echo (newest model), Alexa speaker v	4.7 out of 5 stars	149,562 ratings	Not Available
23	Power-Bank-Portable-Charger - 40000mAh Pc	4.3 out of 5 stars	165 ratings	Not Available
24	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
25	Electronic Devices and Circuit Theory	4.4 out of 5 stars	712 ratings	Not Available
26	320W Charging Station for Multiple Devices, Zo	4.6 out of 5 stars	430 ratings	In Stock

amazon_data