```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from bs4 import BeautifulSoup
import time
import pandas as pd

options = Options()
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36")

driver = webdriver.Chrome(options=options)

categories = {
    "laptops": "https://www.flipkart.com/search?q=laptop",
    "smartphones": "https://www.flipkart.com/search?q=smartphone",
    "cameras": "https://www.flipkart.com/search?q=camera"
}

product_names, prices, ratings, discounts, product_links, image_urls, product_categories = [], [], [], [], [], [], []

for category_name, category_url in categories.items():
    page = 1
    while True:
        driver.get(f"{category_url}&page={page}")
        time.sleep(5)

        for _ in range(5):
            driver.execute_script("window.scrollBy(0, document.body.scrollHeight / 3);")
```

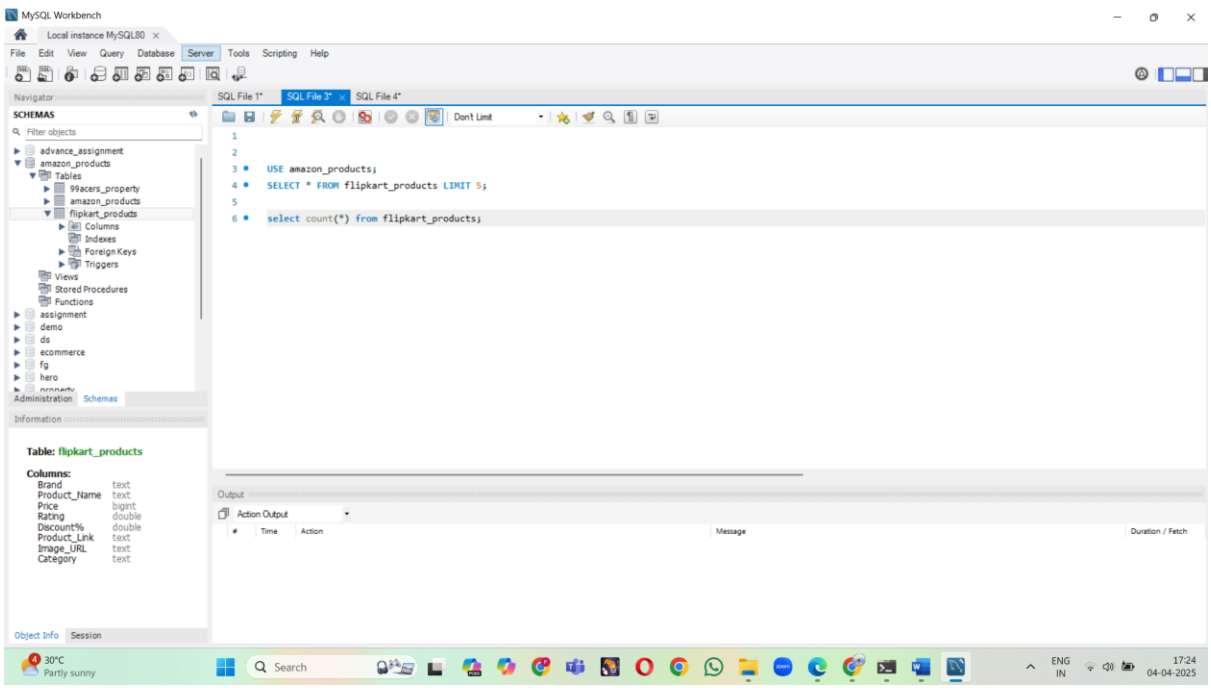
```
            time.sleep(2)

        try:
            WebDriverWait(driver, 15).until(
                EC.presence_of_element_located((By.CLASS_NAME, "KzDlHZ"))
            )
        except:
            print(f" Timeout: Couldn't load page {page} for category: {category_name}")
            break

        soup = BeautifulSoup(driver.page_source, 'html.parser')



        products = soup.find_all('a', class_="CGtC98")
        category_section = category_name

        for product in products:
            name = product.find('div', class_="KzDlHZ")
            product_names.append(name.get_text(strip=True) if name else "N/A")

            product_link = f"https://www.flipkart.com{product['href']}" if product and
product['href'].startswith('/') else product['href']
            product_links.append(product_link)



            price = product.find('div', class_="Nx9bqj _4b5DiR")
            prices.append(price.get_text(strip=True) if price else "N/A")

            rating = product.find('div', class_="XQDdHH")
            ratings.append(rating.get_text(strip=True) if rating else "N/A")
```

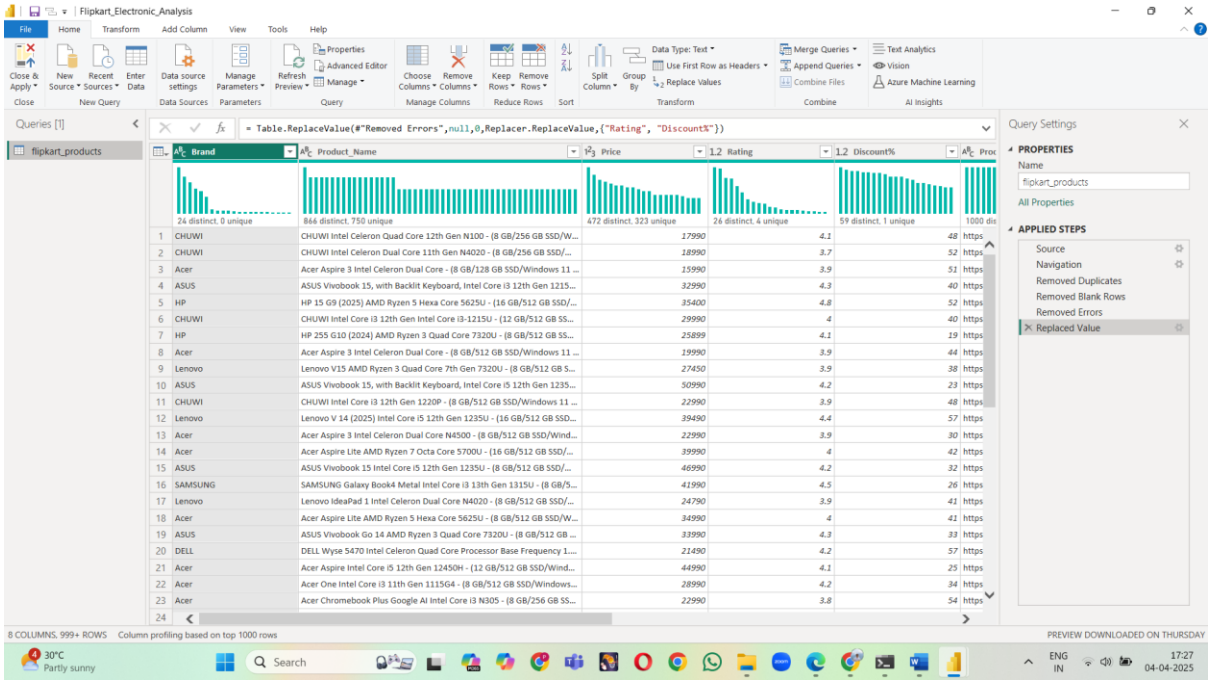
```
            discount = product.find('div', class_="UkUFwK")
            discount_percentage = discount.span.get_text(strip=True).replace("% off", "") if discount else
"N/A"
            discounts.append(discount_percentage)

            image = product.find('img', class_="DByuf4")
            image_url = image['src'] if image else "N/A"
            image_urls.append(image_url)

            product_categories.append(category_section)

        try:
            next_page_button = WebDriverWait(driver, 10).until(
                EC.element_to_be_clickable((By.CLASS_NAME, "_9QVEpD"))
            )
            driver.execute_script("arguments[0].click();", next_page_button)
            page += 1
            time.sleep(5)
        except:
            print(f" No next page button or it was not clickable on page {page}")
            break

driver.quit()

df = pd.DataFrame({
    'Product_Name': product_names,
    'Price': prices,
    'Rating': ratings,
    'Discount%': discounts,
    'Product_Link': product_links,
    'Image_URL': image_urls,
```

```
    'Category': product_categories
})

print(f" Total products scraped: {len(df)}")

df.to_csv('flipkart_products.csv', index=False)

import pandas as pd
from sqlalchemy import create_engine

# Load the data
df = pd.read_csv("flipkart_products.csv")

# 1. Standardize Price Format: Remove ₹ and commas, then convert to numeric
df["Price"] = (
    df["Price"]
    .astype(str)  # Ensure all values are strings
    .str.replace("₹", "", regex=True)
    .str.replace(",", "", regex=True)
)

# Convert to numeric, forcing errors to NaN
df["Price"] = pd.to_numeric(df["Price"], errors="coerce")

# 2. Handle Missing Prices: Drop rows where Price is NaN
df.dropna(subset=["Price"], inplace=True)

# Convert Price to integer after dropping NaN values
df["Price"] = df["Price"].astype(int)

# 3. Handle Unavailable Ratings: Replace "N/A" with None and convert to float
```

```
df["Rating"] = df["Rating"].replace("N/A", None).astype(float)

# 4. Extract Brand from Product_Name (First word)
df["Brand"] = df["Product_Name"].str.split().str[0]

# Reorder columns for better readability
df = df[["Brand", "Product_Name", "Price", "Rating", "Discount%", "Product_Link", "Image_URL", "Category"]]



# Save cleaned data to CSV as well
df.to_csv("flipkart_cleaned_data.csv", index=False)

print("Data cleaning and storage completed successfully!")



import pandas as pd
from sqlalchemy import create_engine
import datetime

# Start time
start_time = datetime.datetime.now()
print('Begin:', start_time)

# Correct MySQL connection string
engine = create_engine('mysql+pymysql://root:Sharma%40123@localhost:3306/amazon_products')

df = pd.read_csv("flipkart_cleaned_data.csv")

# Try inserting data into MySQL
try:
```

```
    df.to_sql(name='flipkart_products', con=engine, index=False, if_exists='replace')
    print("Successfully imported")
except Exception as e:
    print(f"Failed to import. Error: {e}")

# End time
end_time = datetime.datetime.now()
print('End:', end_time)

# Total execution time
total_time = end_time - start_time
print('Total time:', total_time)
```




Flipkart_Electronic_Analysis
File
Home
Transform
Add Column
View
Tools
Help
Close & Apply
New Source
Recent Sources
Enter Data
Data source settings
Manage Parameters
Refresh Preview
Properties
Advanced Editor
Manage
Choose Columns
Remove Columns
Keep Rows
Remove Rows
Split Column
Group By
Data Type: Text
Use First Row as Headers
Replace Values
Merge Queries
Append Queries
Combine Files
Text Analytics
Vision
Azure Machine Learning
Close
New Query
Data Sources
Parameters
Query
Manage Columns
Reduce Rows
Sort
Transform
Combine
AI Insights
Queries [1]
flipkart_products
= Table.ReplaceValue(#"Removed Errors",null,0,Replacer.ReplaceValue,{"Rating", "Discount%"})
Brand
Product_Name
Price
Rating
Discount%
24 distinct, 0 unique
866 distinct, 750 unique
472 distinct, 323 unique
26 distinct, 4 unique
59 distinct, 1 unique
1 CHUWI CHUWI Intel Celeron Quad Core 12th Gen N100 - (8 GB/256 GB SSD/W... 17990 4.1 48 https
2 CHUWI CHUWI Intel Celeron Dual Core 11th Gen N4020 - (8 GB/256 GB SSD/... 18990 3.7 52 https
3 Acer Acer Aspire 3 Intel Celeron Dual Core - (8 GB/128 GB SSD/Windows 11 ... 15990 3.9 51 https
4 ASUS ASUS Vivobook 15, with Backlit Keyboard, Intel Core i3 12th Gen 1215... 32990 4.3 40 https
5 HP HP 15 G9 (2025) AMD Ryzen 5 Hexa Core 5625U - (16 GB/512 GB SSD/... 35400 4.8 52 https
6 CHUWI CHUWI Intel Core i3 12th Gen Intel Core i3-1215U - (12 GB/512 GB SS... 29990 4 40 https
7 HP HP 255 G10 (2024) AMD Ryzen 3 Quad Core 7320U - (8 GB/512 GB SS... 25899 4.1 19 https
8 Acer Acer Aspire 3 Intel Celeron Dual Core - (8 GB/512 GB SSD/Windows 11 ... 19990 3.9 44 https
9 Lenovo Lenovo V15 AMD Ryzen 3 Quad Core 7th Gen 7320U - (8 GB/512 GB S... 27450 3.9 38 https
10 ASUS ASUS Vivobook 15, with Backlit Keyboard, Intel Core i5 12th Gen 1235... 50990 4.2 23 https
11 CHUWI CHUWI Intel Core i3 12th Gen 1220P - (8 GB/512 GB SSD/Windows 11 ... 22990 3.9 48 https
12 Lenovo Lenovo V 14 (2025) Intel Core i5 12th Gen 1235U - (16 GB/512 GB SSD... 39490 4.4 57 https
13 Acer Acer Aspire 3 Intel Celeron Dual Core N4500 - (8 GB/512 GB SSD/Wind... 22990 3.9 30 https
14 Acer Acer Aspire Lite AMD Ryzen 7 Octa Core 5700U - (16 GB/512 GB SSD/... 39990 4 42 https
15 ASUS ASUS Vivobook 15 Intel Core i5 12th Gen 1235U - (8 GB/512 GB SSD/... 46990 4.2 32 https
16 SAMSUNG SAMSUNG Galaxy Book4 Metal Intel Core i3 13th Gen 1315U - (8 GB/5... 41990 4.5 26 https
17 Lenovo Lenovo IdeaPad 1 Intel Celeron Dual Core N4020 - (8 GB/512 GB SSD/... 24790 3.9 41 https
18 Acer Acer Aspire Lite AMD Ryzen 5 Hexa Core 5625U - (8 GB/512 GB SSD/W... 34990 4 41 https
19 ASUS ASUS Vivobook Go 14 AMD Ryzen 3 Quad Core 7320U - (8 GB/512 GB ... 33990 4.3 33 https
20 DELL DELL Wyse 5470 Intel Celeron Quad Core Processor Base Frequency 1.... 21490 4.2 57 https
21 Acer Acer Aspire Intel Core i5 12th Gen 12450H - (12 GB/512 GB SSD/Wind... 44990 4.1 25 https
22 Acer Acer One Intel Core i3 11th Gen 1115G4 - (8 GB/512 GB SSD/Windows... 28990 4.2 34 https
23 Acer Acer Chromebook Plus Google AI Intel Core i3 N305 - (8 GB/256 GB SS... 22990 3.8 54 https
24
Query Settings
PROPERTIES
Name
flipkart_products
All Properties
APPLIED STEPS
Source
Navigation
Removed Duplicates
Removed Blank Rows
Removed Errors
Replaced Value
8 COLUMNS, 999+ ROWS
Column profiling based on top 1000 rows
PREVIEW DOWNLOADED ON THURSDAY
30°C
Partly sunny
Search
ENG
IN
17:27
04-04-2025