E-Commerce Data Engineering Project

Team members:

1. Alistair Pereira
2. Blessy Aaron
3. Gourav Jadhav
4. Samyukta Tarun
5. Suyash Mali

Introduction to Data Engineering

Data engineering focuses on the practical application of data collection, storage, and retrieval.

Key tasks include Scraping data, storing data into HDFS, injecting data into SQL server ensuring data quality, and integrating data from multiple sources.

Domain and its Relevance

Why E-commerce?

- High data availability and variety.
- Significant impact on business strategy and customer satisfaction.

Focus on Mobile Products:

- High consumer demand and frequent updates.
- Valuable insights into market trends and consumer preferences.

Data Sources and Database used

Selected Platforms:

- Flipkart
- Newegg
- eBay

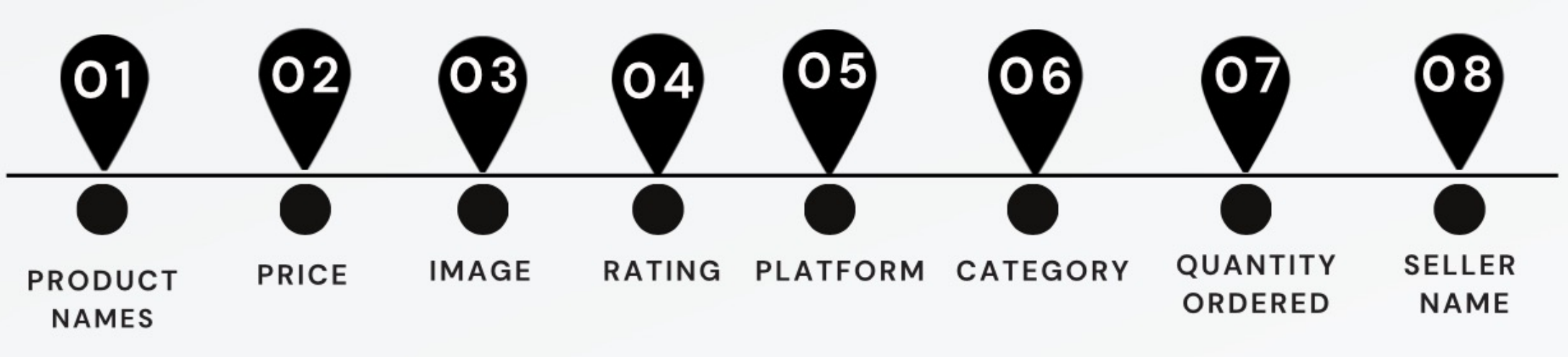
Reason for selection:

- Diverse product range
- Large user base
- Rich dataset for comprehensive analysis

DATABASE USED:

MySQL

Data Points Collected



Data Collection Methods

Tools and Techniques:

Web scraping using BeautifulSoup and Selenium.

BeautifulSoup : install BeautifulSoup and send requests to libraries

send HTTP request to fetch the webpage, parse and extract

using BeautifulSoup's parsing methods

Selenium : install selenium and a web driver, then setup the web browser

interact with elements and extract the data

Data Processing

STEPS INVOLVED:

Data Cleaning: Handling missing values, removing duplicates.

Data Transformation: Standardizing formats, normalizing data.

Data Aggregation: Summarizing data for analysis.

```python
    # Regular expression to find all numbers in the string
    prices = re.findall(r'\d+\.\d+', s)

    # Convert found price strings to floats
    prices = [float(price.replace(",", "")) for price in prices]

    if len(prices) == 2:
        # If there are two prices, calculate the average
        avg = (prices[0] + prices[1]) / 2
        return avg
    elif len(prices) == 1:
        # If there is one price, return it
        return prices[0]
    else:
        # Handle case where no price is found
        return None

s1 = "EUR 260.75 to EUR 308.16"
s2 = "EUR 260.75"
s3 = "$255.31 to $383.60"
s4 = "$255.31"

cleaned_price1 = clean_price(s1)
cleaned_price2 = clean_price(s2)
cleaned_price3 = clean_price(s3)
cleaned_price4 = clean_price(s4)

print(cleaned_price1)
print(cleaned_price2)
print(cleaned_price3)
print(cleaned_price4)
```

```
284.45500000000004
260.75
319.45500000000004
255.31
```

```python
def clean_price(s):
    if "$" in s:
        s = float("".join(s.split("$")[1].split(".")[0].split(",")))
        return s
    else:
        return None
```

```python
res =clean_price("$545.00\xa0(4 Offers)-")
res
```

```
545.0
```

```python
res =clean_price("$329.97\xa0-")
res
```

```
329.0
```

```python
res =clean_price("$1,139.99\xa0(2 Offers)-")
res
```

```
1139.0
```

TAILSCALE

srh

Introduction to Tailscsale:

Tailscale is a secure and easy-to-use mesh VPN service that connects devices creating a secure network across multiple environments.

Why Tailscale?

Simple setup, high security, and seamless integration with existing network infrastructure make Tailscale ideal for managing remote access and network security.

Tailscale Architecture:

•**Nodes**: Individual devices connected to the Tailscale network.

•**Control Plane**: Manages authentication, device registration, and coordination.

group05de@gmail.com

Download Support Docs G

Machines Apps Services Users Access controls Logs DNS Settings Get started

Machines

Manage the devices connected to your tailnet. Learn more ↗

Add device

Search by name, owner, tag, version...

Filters

10 machines

MACHINE	ADDRESSES	VERSION	LAST SEEN
blessys-air group05de@gmail.com	100.74.126.103	1.68.1 macOS 13.5.0	Connected
gj group05de@gmail.com	100.73.164.47	1.68.2 Windows 11 23H2	5:26 PM GMT+2
laptop-hggr060l group05de@gmail.com	100.69.0.12	1.68.2 Windows 11 23H2	Connected
master-node group05de@gmail.com	100.118.157.91	1.68.2 Linux 5.15.0-113-generic	Connected
samyuktas-macbook-pro group05de@gmail.com	100.94.0.71	1.68.1 macOS 13.6.6	Connected
suyashs-macbook-air group05de@gmail.com	100.72.230.93	1.68.1 macOS 14.5.0	Connected
ubuntu-blessy group05de@gmail.com	100.82.140.82	1.68.2 Linux 5.15.0-113-generic	Connected

SWARM

- A tool that helps manage and organize a group of computers running Docker.
- It allows you to combine multiple machines into a single, large computer that can run and manage applications smoothly.
- Swarm handles tasks like distributing work evenly, scaling up services, and ensuring everything runs reliably.

root@ubuntu: /home/alistair

```
root@ubuntu:/home/alistair# sudo docker node ls
ID                            HOSTNAME            STATUS    AVAILABILITY   MANAG
ER STATUS   ENGINE VERSION
q7d8qqemj9rz2f6l91wrxkiao     gourav-VirtualBox   Ready     Active
            26.1.4
eu60gwffaufd5hg47li8r24ef *   ubuntu              Ready     Active         Leade
r           26.1.4
oc4hsfn3yw89rjn0zxpbgddnx     ubuntu              Down      Active
            27.0.3
nvl0eg25rkq8dwddwjlpmlkwr     ubuntue             Ready     Active
            24.0.7
zu3vfus3j8y4np5m0vodowlri     ubuntue             Ready     Active
            27.0.3
root@ubuntu:/home/alistair# sudo docker swarm leave --force
Node left the swarm.
root@ubuntu:/home/alistair# sudo docker node ls
Error response from daemon: This node is not a swarm manager. Use "docker swarm
init" or "docker swarm join" to connect this node to swarm and try again.
root@ubuntu:/home/alistair# sudo docker swarm init --advertise-addr 100.118.157.
91
Swarm initialized: current node (rlf3w3b9r15160o1k4802oa8h) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-40nshz7lw3fxcm124k3f5qslccfm9gpsxi5q56xef
mz7vnzncq-9ctzqaz92zw9gl1v77swe3svy 100.118.157.91:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
 the instructions.

root@ubuntu:/home/alistair# ^C
root@ubuntu:/home/alistair#
```

HDFS

Introduction to HDFS:

The Hadoop Distributed File System (HDFS) is designed for reliable, scalable, and fault-tolerant storage of large datasets.

HDFS Architecture:

- **NameNode**: Manages metadata and directory structure.
- **DataNode**: Stores actual data blocks, handles read/write requests.
- **Replication**: Default factor is 3, ensuring data availability.
- **Block Size**: Typically 128 MB for efficient large file handling.

In operation

DataNode State All Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used		Capacity	Blocks	Block pool used	Version
/default-rack/ubuntu-samyukta.tail67005d.ts.net:9866 (100.90.32.46:9866)	http://ubuntu-samyukta.tail67005d.ts.net:9864	1s	2m	63.73 KB	12.88 GB	47.41 GB		0	63.73 KB (0%)	3.3.2
/default-rack/master-node.tail67005d.ts.net:9866 (100.118.157.91:9866)	http://master-node.tail67005d.ts.net:9864	2s	2m	51.93 KB	30.97 GB	47.93 GB		0	51.93 KB (0%)	3.3.2
/default-rack/ubuntu-gaurav.tail67005d.ts.net:9866 (100.119.234.100:9866)	http://ubuntu-gaurav.tail67005d.ts.net:9864	1s	2m	240 KB	19.46 GB	97.87 GB		0	240 KB (0%)	3.3.2
/default-rack/ubuntu-blessy.tail67005d.ts.net:9866 (100.82.140.82:9866)	http://ubuntu-blessy.tail67005d.ts.net:9864	2s	1m	320 KB	16.41 GB	96.84 GB		0	320 KB (0%)	3.3.2
/default-rack/ubuntu-suyash.tail67005d.ts.net:9866 (100.73.150.109:9866)	http://ubuntu-suyash.tail67005d.ts.net:9864	2s	1m	52 KB	14.32 GB	96.84 GB		0	52 KB (0%)	3.3.2

Showing 1 to 5 of 5 entries Previous 1 Next

All 5 worker nodes are connected

In operation

DataNode State All Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used		Capacity	Blocks	Block pool used	Version
/default-rack/ubuntu-samyukta.tail67005d.ts.net:9866 (100.90.32.46:9866)	http://ubuntu-samyukta.tail67005d.ts.net:9864	0s	16m	96 KB	12.88 GB	47.41 GB		3	96 KB (0%)	3.3.2
/default-rack/master-node.tail67005d.ts.net:9866 (100.118.157.91:9866)	http://master-node.tail67005d.ts.net:9864	1s	16m	392 KB	30.97 GB	47.93 GB		1	392 KB (0%)	3.3.2
/default-rack/ubuntu-gaurav.tail67005d.ts.net:9866 (100.119.234.100:9866)	http://ubuntu-gaurav.tail67005d.ts.net:9864	1s	16m	240 KB	19.46 GB	97.87 GB		0	240 KB (0%)	3.3.2
/default-rack/ubuntu-suyash.tail67005d.ts.net:9866 (100.73.150.109:9866)	http://ubuntu-suyash.tail67005d.ts.net:9864	623s	15m	52.06 KB	14.32 GB	96.84 GB		1	52.06 KB (0%)	3.3.2
/default-rack/ubuntu-blessy.tail67005d.ts.net:9866 (100.82.140.82:9866)		Tue Jul 16 05:34:44 +0530 2024								

Showing 1 to 5 of 5 entries Previous 1 Next

1 worker node is down

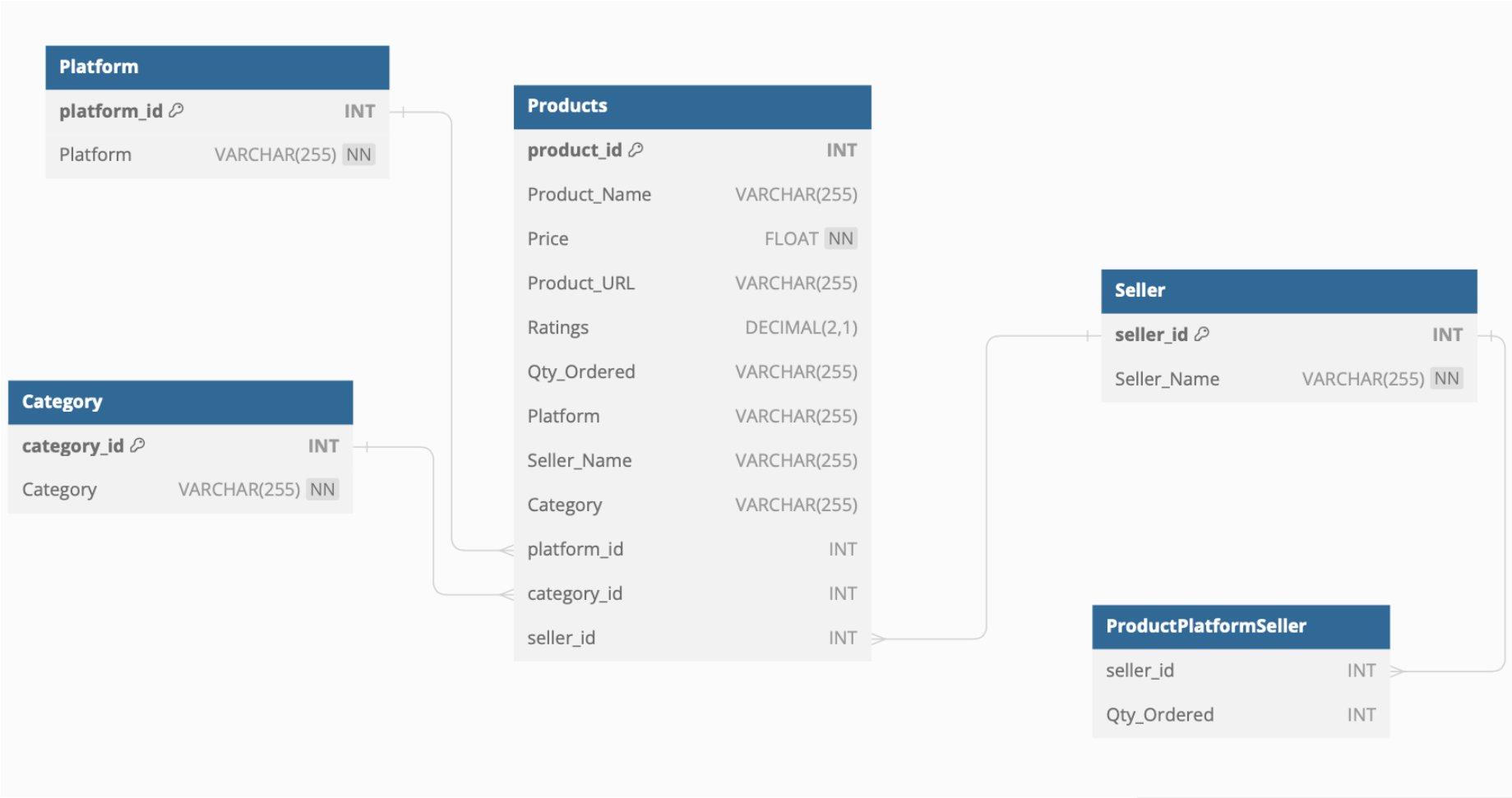
Integration with our project:

- **HDFS**

Data ingestion from Flipkart, Newegg, and eBay into HDFS.

Storage and processing of raw, cleaned, and aggregated data.

Database Schema



Data Analysis

Business Queries

1. Seller name and total sales quantities
2. Find price as per e-Bay platform on basis of products
3. Most expensive product sold on each e-commerce website
4. Product with highest ratings
5. Product with lowest ratings
6. As per seller name, total quantities sold
7. Max sales by which seller for which product

```sql
#1 seller name and total sales quantites
SELECT
    s.seller_id,
    s.Seller_Name,
    SUM(pps.Qty_Ordered) AS Total_Sales_Quantity
FROM
    Seller s
LEFT JOIN
    ProductPlatformSeller pps ON s.seller_id = pps.seller_id
GROUP BY
    s.seller_id, s.Seller_Name;
```

seller_id	Seller_Name	Total_Sales_Quantity
1	shop4u2011	6637
2	reunikat	3585
3	PowerSimShop	11333
4	Happykidsroom	3924
5	examinneer	9730
6	Bieco-Spielwaren	10910
7	trade-m	12190
8	lanhk_tddfc3uu	4585
9	enstore10	6291
10	loooping-lu Online-Shop	3258

```sql
#2 Find Price as per the Ebay platform on basis of Products
SELECT Product_Name, Price, Platform.Platform
FROM de.Product
INNER JOIN Platform ON Product.platform_id = Platform.platform_id
WHERE Platform.Platform = 'ebay'
LIMIT 0, 1000;
```

product_id	Product_Name	Price	Product_URL	Ratings	Qty_Ordered	Platform	Seller_Name	Category	platform_id	category_id	seller_id
337	Samsung Galaxy S21 5G 128GB G991U Unlocke...	156.99	https://i.ebayimg.com/thumbs/images/g/C0EAA...	4.5	6558	ebay	taylor-wheels	Samsung	1	2	269
701	Samsung Galaxy S21 5G 128GB G991U Unlocke...	156.99	https://i.ebayimg.com/thumbs/images/g/C0EAA...	4.5	6558	ebay	bit-electronix	Samsung	1	2	74

```sql
#3 Max sales by which Seller for which Product
select max(Qty_Ordered) from de.ProductPlatformSeller;
select * from de.Product where Qty_Ordered = 6558;
```

	max(Qty_Ordered)
▶	6558

```
#4 Most Expensive Product sold on each Ecommerce website;

select max(Price) from de.Product where platform = "flipkart" ;
select Product_Name,Price from de.Product where Price = 1000  ;

select max(Price) from de.Product where platform = "ebay";
select Product_Name,Price from de.Product where Price = 989.99 limit 1;

select max(Price) from de.Product where platform = "newegg";
select Product_Name,Price from de.Product where Price = 2419;
```

Product_Name	Price
Apple iPhone 15 Plus (Blue, 128 GB)	1000
Apple iPhone 15 Plus (Green, 128 GB)	1000

```
#5 Product with highest ratings.
select max(Ratings) from de.Product;
select * from de.Product where Ratings = 5;
select count(product_id) from de.Product where Ratings = 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	count(product_id)
▶	547

```sql
#6 Product with lowest ratings.
select min(Ratings) from de.Product;
select * from de.Product where Ratings = 1;
select count(product_id) from de.Product where Ratings = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	count(product_id)
▶	85

```sql
#7 As per Seller name total quantity sold
SELECT  s.Seller_Name,
    SUM(ps.Qty_Ordered) AS Total_Quantity_Sold
FROM
    de.Product p
JOIN
    ProductPlatformSeller ps ON p.seller_id = ps.seller_id
JOIN
    Seller s ON p.seller_id = s.seller_id
GROUP BY
    s.Seller_Name
ORDER BY
    Total_Quantity_Sold DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Seller_Name	Total_Quantity_Sold
bau-tech Solarenergie GmbH	338525
Die ELEKTROHALLE	254820
Elektriese_eu	239148
llilysarroy02	207285
EcoFlow_Germany	170672

Challenges

- Even after worker node is Active it was not displayed on the HDFS UI.
- In WebScraping, while fetching the seller data from Flipkart website we were not able to fetch it using beautiful soup, but selenium helped us to fetch the seller.
- Data Cleaning was a big challenge as we had to handle various kinds of data.
- During data ingestion from HDFS to SQL, maintaining the relationship between primary and foreign key.

Thank you!