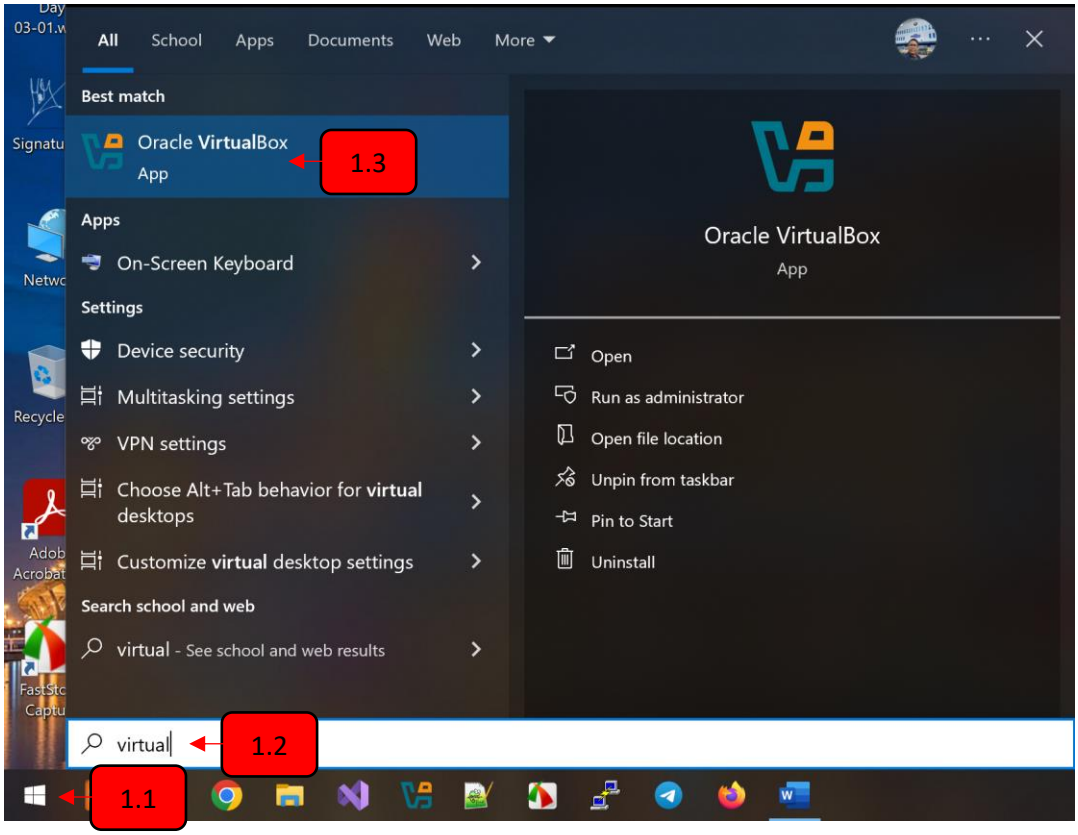
Instruction 1: Big Data Analytics Via Terminal in Ubuntu Linux

Part 1/5: Start VirtualBox and Virtual Machine

Step 1.1: Click the **Windows** Icon.

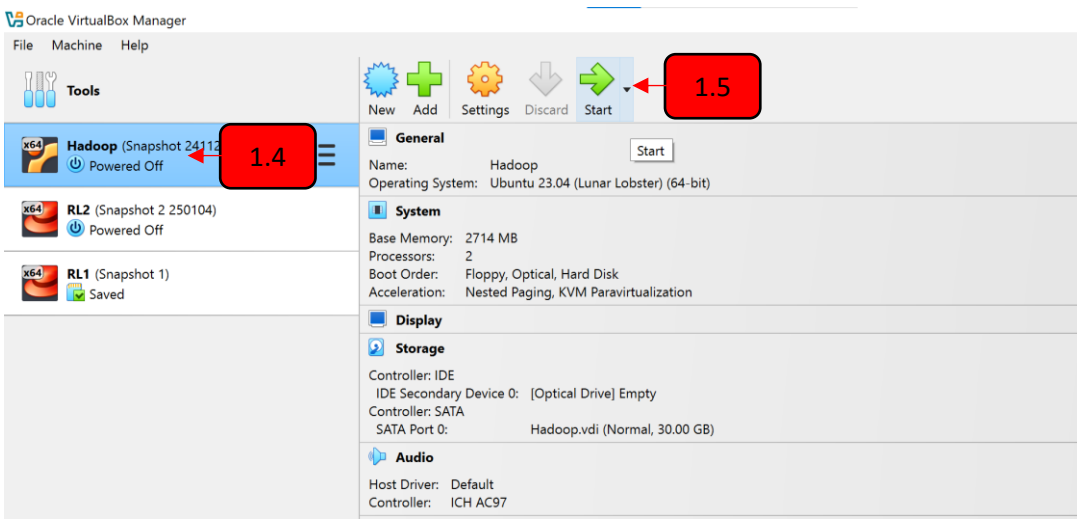
Step 1.2: Type **VirtualBox**.

Step 1.3: Click on the **Oracle VirtualBox** app.



Step 1.4: Select the **Hadoop** Virtual Machine.

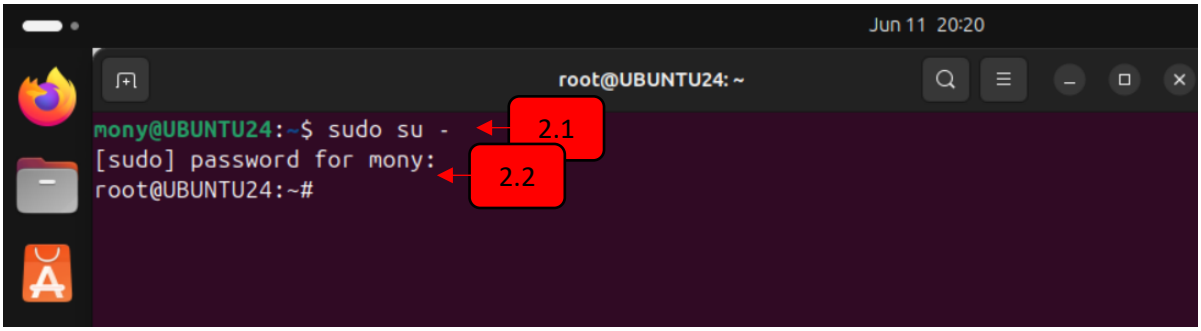
Step 1.5: Click on the **Start** icon.



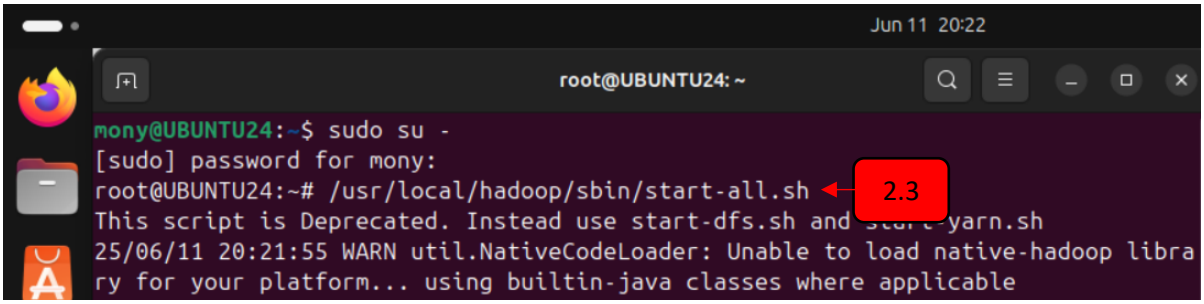
Part 2/5: Start Hadoop Services

Step 2.1: Click on the **Terminal** Icon on the Dash, type **sudo su -** and press the **Enter** key.

Step 2.2: Type the password **Admin1111** and press the **Enter** key.



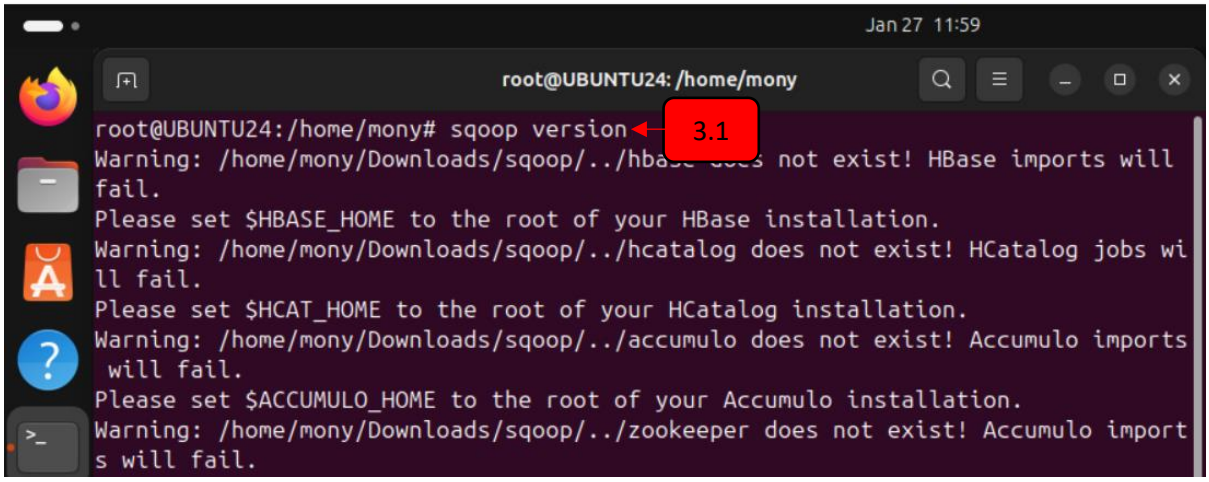
Step 2.3: Type **/usr/local/hadoop/sbin/start-all.sh** and press **Enter**.



Step 2.4: Type **jps** to check if Hadoop services are running.

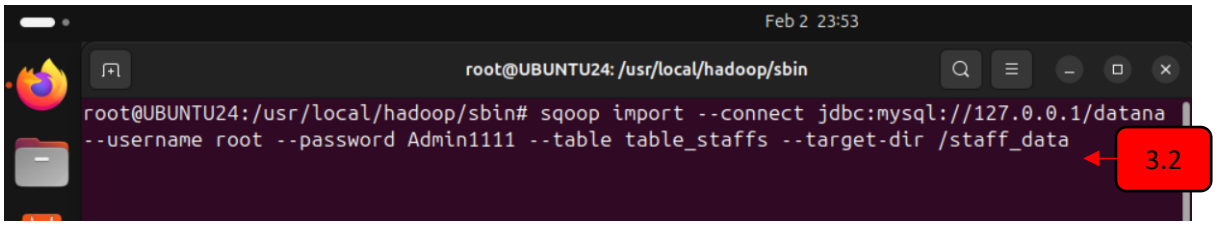
Part 3/5: Check Sqoop Version and Import Data from Database

Step 3.1: Type **sqoop version** and press **Enter**

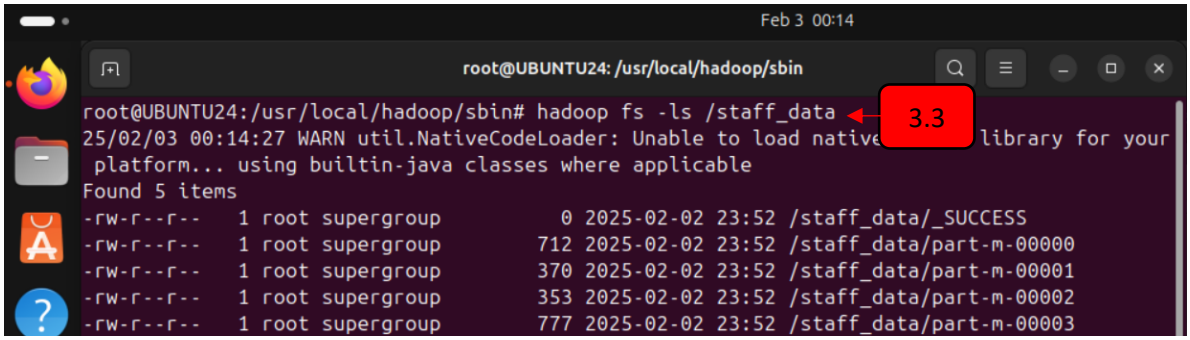


Step 3.2: type the command below and press **Enter**.

sqoop import --connect jdbc:mysql://127.0.0.1/dbtest --username usertest --password Admin1111 --table staff_data --target-dir /staff_data --delete-target-dir



Step 3.3: Type **hadoop fs -ls /staff_data** and press **Enter** key to view imported data.



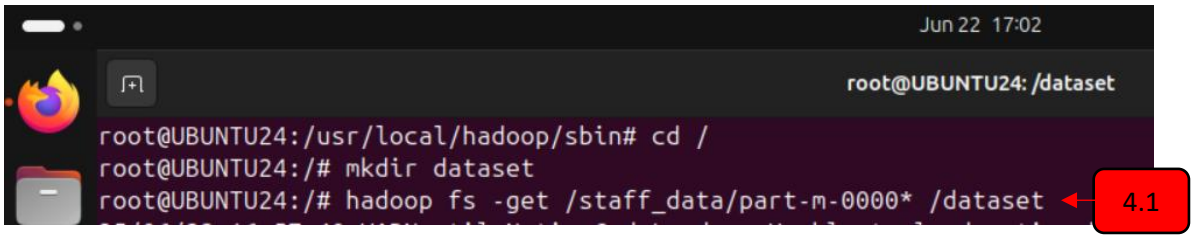
Part 4/5: Use Python to Clean Data in HDFS

Step 4.1: Type the following to import data from Hadoop to a local directory:

cd /

mkdir dataset

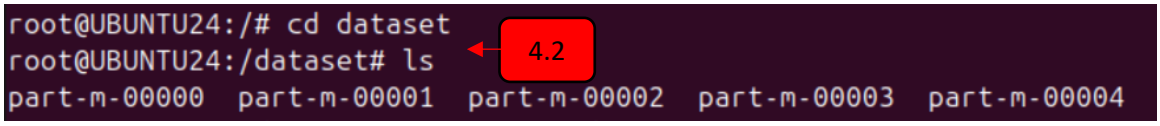
hadoop fs -get /staff_data/part-m-0000* /dataset



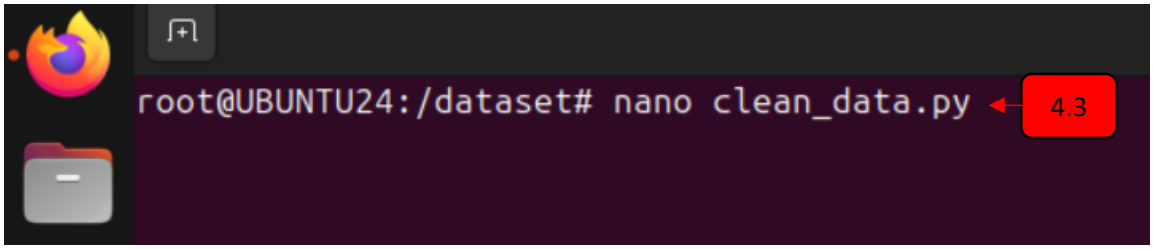
Step 4.2: Verify the imported data:

cd dataset

ls



Step 4.3: Type **nano clean_data.py** to open a new Python script.



Step 4.4: Type the following Python code, then press **Ctrl + X**, then **Y**, then **Enter**:

```
import glob
# Match all part files downloaded from Hadoop
file_list = glob.glob("part-m-0000*")
cleaned_data = []
for file_name in file_list:
    with open(file_name, "r", encoding="utf-8") as file:
        for line in file:
            parts = line.strip().split(",")
            if len(parts) >= 13:
                # Extract selected fields: staff_id, staff_card_number, staff_full_name, sex, salary
                selected = parts[0:3] + [parts[5], parts[10]]
                cleaned_data.append(selected)
# Write cleaned data to a single output file
with open("staff_filtered.csv", "w", encoding="utf-8") as out:
    for row in cleaned_data:
        out.write(",".join(row) + "\n")
```

4.4

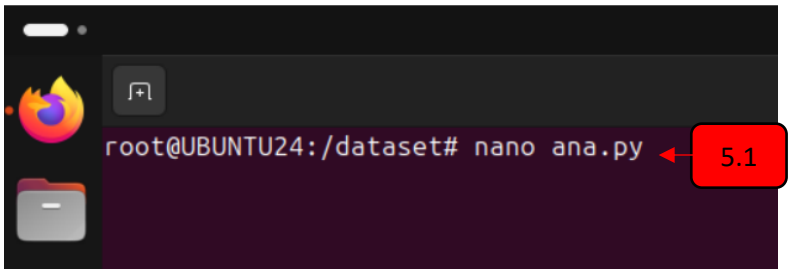
Step 4.5: Run the script to generate **staff_filtered.csv** with the selected cleaned fields.

```
python3 clean_data.py
```

4.5

Part 5/5: Analyze Data Using Python

Step 5.1: Type **nano ana.py** to create a new analysis script.

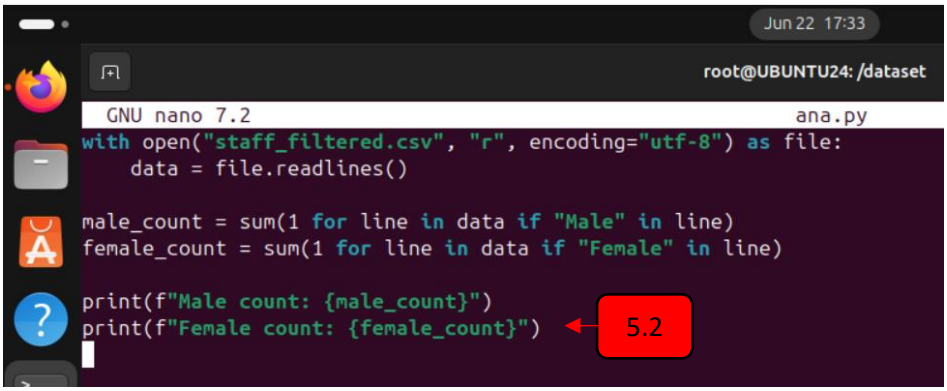


Step 5.2: Type the following code:

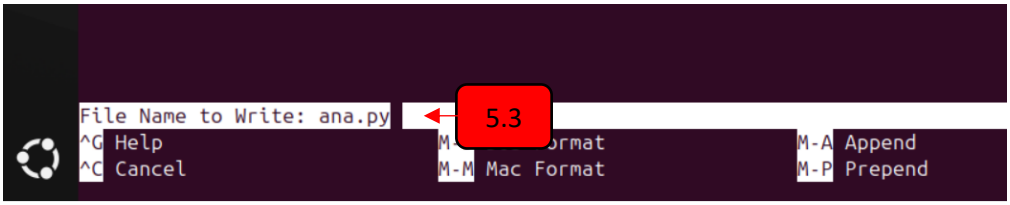
```
with open("staff_filtered.csv", "r", encoding="utf-8") as file:
        data = file.readlines()


male_count = sum(1 for line in data if "Male" in line)
female_count = sum(1 for line in data if "Female" in line)


print(f"Male count: {male_count}")
print(f"Female count: {female_count}")
```



Step 5.3: Save the file by pressing **Ctrl + X**, then **Y**, then **Enter**.



Step 5.4: Type **python3 ana.py** and press **Enter** to run the script.