UNIVERSITI MALAYA

Implementation of Hive, HBase and Pig in Social Media Dataset

WQD7007

Big Data

Management

Group Project

Instructor: Dr Hoo Wai Lam

Table of Contents

1.0 Introduction

1.1 Background

Social media is an internet-based communication platform that allows users to have conversations, share information, and create web content. Social media is used by over 4.7 billion people or 60% of the world's population. People can use social media platforms to get up-to-date information, connect with others, and discover niche communities. Briefly, social media is an online resource that is meant to enhance interaction between persons (Bishop M, 2019) Many people have used it to connect with others online, making the world appear more interconnected and accessible. They use social media applications to network for job opportunities, find people with similar interests all over the world, and share political opinions. Entertainers and politicians also utilize social media to communicate with residents and voters. It has turned into an important business tool. The businesses that utilize platforms to locate and engage consumers, boost sales through advertising and marketing, understand consumer trends, and give customer care or support. The capabilities of social media have led to businesses using it to promote their products and services by allowing the dissemination of targeted, timely, and exclusive deals and coupons to potential consumers. Social media messaging applications and platforms are currently among the most popular websites on the internet. In early 2023, chat and messaging applications and websites were utilized by 94.8% of people, with social platforms following in second at 94.6%. With 81.8% of visitors viewing them, search engine sites came in second. These platforms can be categorized based on the interests and goals of its members. Platforms for video gamers, social gamers, video sharers, professional business networks, virtual worlds, review platforms, and more are available. Most Americans claim they use YouTube and Facebook, while Instagram, Snapchat, and TikTok are especially popular among those under 30. (Auxier & Anderson, 2021).

Big data technologies are software systems that manage several types of datasets and convert them into business insights. There are many examples of big data technology. One of them is Hadoop HDFS. Hadoop HDFS is a distributed file system, data storage platforms, analytics platforms, and a layer that controls parallel processing, rate of flow (workflow), and configuration management comprise Hadoop. The HDFS file system was created to handle high volumes of data. (Jach, Magiera, & Froelich, 2015). It is intended for streaming, which involves reading enormous volumes of data from discs in bulk. It connects the file systems on numerous input and output data nodes to construct one big file system across the nodes of a Hadoop cluster. (Priya P. & Chandrakant P., 2014). It is also a highly fault-tolerant distributed file system that oversees storing data on clusters. It is used when the volume of data is too large for a single machine to handle. Another example of big data technology is Hive. Hive is a Hadoop-based data warehouse infrastructure that provides data summarization, querying, and analysis. It was created by Facebook, and it is currently utilized and developed by other firms like Netflix (Chavan & Phursule, 2014). Hive can analyze massive datasets stored in HDFS and other compatible file systems. It offers a SQL-like language called HiveQL that has schema on read and transparently translates queries to map/reduce and Apache Tez (Alvarez-Dionisi, 2017). It also facilitates the integration of Hadoop with business intelligence and tool visualization. Another example of big data technology is Pig. Pig is a high-level data programming language for analyzing Hadoop data. It was originally developed at Yahoo! to allow Hadoop users to focus more on huge data set analysis and spend less time writing mapper and reducer programs, and it is now part of the Apache software foundation. Pig is designed to handle data flow language and execution environment for investigating large datasets (Chavan & Phursule, 2014). Pig runs on HDFS and MapReduce clusters. Given its data model, it is more elastic than Hive in terms of conceivable data format.

Big data technology found its use in social media websites. For example, Facebook makes use of the Hadoop HDFS architecture. It gets data from two sources. After storing user data in the federated MySQL layer, web servers generate event-based log data. Second, data is collected from web servers and delivered to Scribe servers, located on Hadoop clusters. The analyzed data findings are saved in the Hadoop Hive cluster or, for Facebook users, in the MySQL tier. Ad hoc analytic queries (Hive CLI) are created with either a graphical user interface (HiPal) or a command-line interface (Hive). Facebook uses a Python framework to operate the database and schedule periodic batch activities in the production cluster. Because we must deal with huge volumes of data daily, big data technologies are quite beneficial in organizing social media.

1.2 Problem Statement

The big data age has resulted in the development and deployment of technology and methodologies for successfully utilizing enormous volumes of data to assist decision-making and knowledge-discovery operations. (Storey & Song, 2017). The integration of Big Data technology must deal with the difficulty of the Big Data component of the analyzed data. (Sebei, Hadj Taieb, & Ben Aouicha, 2018). When investigating Big Data sets and extracting value and knowledge from such information mines, researchers face several problems, including issues in data capture, storage, searching, sharing, analysis, management, and visualization. (Ahmed, Fatima-Zahra, Ayoub Ait, & Samir, 2018). Our problem is that we are missing a big data technologies implementation to analyze the usage of preferred social media to understand which platform is universally used. According to Can & Bilal (2017), various disciplines must analyze social network data using big data techniques to better understand the discipline and generate accurate forecasts in each sector.

1.3 Objective

To solve the problem, our objective is to implement big data technologies to analyze the usage of preferred social media to understand which platform is used. Specifically, we need,

1. To implement Hive, Pig, HBase to obtain insights from social media usage dataset.

2. To compare Hive, Pig, HBase based on execution time and code complexity.

1.4 Question

Our question is

1. How to use Hive, Pig and HBase to execute a set of queries that will obtain the insights?

2. Among Hive, Pig and HBase, which tool is efficient to execute queries in terms of execution time and complexity of the code?

2.0 Methodology

2.1 Overview

Figure 1

Data Pipeline used for this study

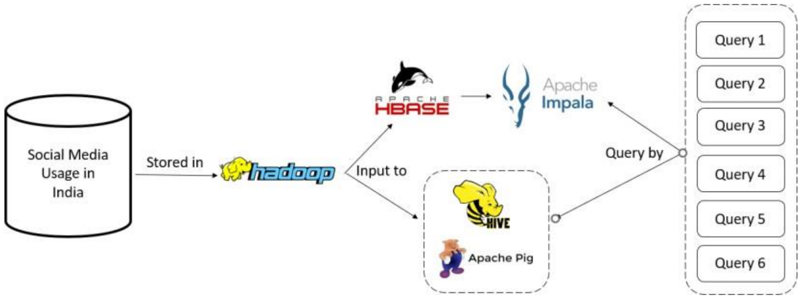


Figure 1 shows the data pipeline that we use for this project. We start with the HDFS (Hadoop Distributed File System) that will be storing our dataset. We set up the HDFS in the hardware and virtual machine shown in Table 1 and Table 2.

Table 1

Hardware configurations used for this study

	Configuration	**Description**
Hive	Processor	12th Gen Intel(R) Core i7-1225U
	RAM	16 GB
	Operating System	Windows
Pig	Processor	Core i5 (10th Gen)
	RAM	20GB
	Operating System	Windows
HBase	Processor	Core i5 (11th Gen)
	RAM	8 GB
	Operating System	Windows

Table 2

Virtual machine configurations used for this study

	Configuration	**Description**
Hive	Operating System	Ubuntu (64-bit)
	RAM	4 GB
	Disk	50 GB
	vCPUs	4
Pig	Operating System	Ubuntu (64-bit)
	RAM	13GB

	Disk	42.98 GB
	vCPUs	5
HBase	Operating System	RHEL (64-bit)
	RAM	4 GB
	Disk	64 GB
	vCPUs	4

After setting up Hadoop HDFS, we upload the dataset into HDFS in Hadoop (Refer Figure HDFS.UploadData.1). The dataset is about the social media usage trends in India. This dataset has 26 columns and 1629 rows (including header). Each row represents users of social media with different attributes. The full description of each column can be viewed in Table Dataset.1.

Then, we proceed to implement 3 Big Data tools, Hive, HBase and Pig to query the dataset. The following shows some queries that will be used as the test case for the big data tools. We chose these queries because these are some of the queries that may be made when analyzing the usage of preferred social media to understand which platform is universally used.

1. How many male and female use respective different operating systems (iOS, Android, others)?

2. What is the average usage in all three social media platforms given different age groups (children: 0-18), (adults: 18-60) and (elderly: > 60)?

3. Which operating system shows the highest usage record for each social media platform?

4. Which gender shows the highest usage record for each social media platform?

5. Which status shows the highest usage record for each social media platform?

6. Which education level shows the highest usage record for each social media platform?

2.2 Hive

In Hive, we start with importing the dataset from HDFS into Hive. To do this, we use the create external table syntax (Refer Figure Hive.Import/Scan.1). In this step, we firstly set up an empty table “Set01” with 26 different columns and then we import our dataset to Hive by “load data inpath”. Figure Hive.Import/Scan.2 shows that the data has been successfully imported. After importing the dataset from HDFS to Hive, we can explore insights from the datasets by executing the Hive queries on the dataset.

We start with Query 1. To do this, we filter the data by the different gender and operating system using “where” syntax and count the row in the filtered data using the “count” function. (Refer Figure Hive.Query1.1 - Hive.Query1.6).

Next is Query 2. To do this, we filter the data by the different age using “where” syntax and average the total usage of each social media platform using the “avg” syntax (Refer Figure Hive.Query2.1 - Hive.Query2.13).

Next is Query 3 – 6. To do these, we group the data by operating system, gender, status and education level respectively using the “group by” syntax and get the maximum record of each social media platform in each group using the “max” syntax. (For query 3, refer Figure Hive.Query3.1 - Figure Hive.Query3.3; For query 4, Refer Figure Hive.Query4.1 - Figure Hive.Query4.3; For query 5, refer Figure Hive.Query5.1 - Figure Hive.Query5.3; For query 6, refer Figure Hive.Query6.1 - Figure Hive.Query6.3).

2.3 Pig

In Pig, we start with importing the dataset from HDFS. To do this, we use the create table syntax (Refer Figure Pig.Import/Scan.1). In this step, we create an empty table called “data” with 26 different columns, and we load our dataset to Pig using “load - using PigStorage” syntax. After loading the dataset from HDFS to Pig, we can explore insights from the datasets by executing the Pig queries on the dataset.

We start with Query 1. To do this, we use “group-by” syntax to group the data by gender and operating system. Then, we use “foreach-generate-count" syntax to count the number of rows in each group (Refer Figure Pig.Query1.1).

Next is Query 2. To do this, we filter the data by different age range using “filter-by” syntax and store the filtered data in different variables. Then, for each filtered data, we get the column that indicates the usage in a social media platform using the “generate” syntax and average the column using the “avg” syntax. This step is repeated for other social media platform (Refer Figure Pig.Query2.1 - Figure Pig.Query2.10)

Next is Query 3 – 6. To do this, we group the data by operating system, gender, status and education level respectively using the “group-by” syntax. Then, we use “foreach-generate-max" syntax to get the maximum of each social media platform in each group (For query 3, refer Figure Pig.Query3.1 - Figure Pig.Query3.3; For query 4, refer Figure Pig.Query4.1 - Figure Pig.Query4.3; For query 5, refer Figure Pig.Query5. - Figure Pig.Query5.3; For query 6, refer Figure Pig.Query6.1 - Figure Pig.Query6.3).

2.4 HBase

In HBase, we start with importing the dataset from HDFS into HBase. To do this, we first create a new table using the “create” syntax with name ‘sm_india’ alongside 4 columns families, ‘general’, ‘facebook’, ‘instagram’ and ‘whatsapp’ (Refer Figure HBase.Import/Scan.1). Then, we will import the dataset to the table using the ‘hbase org.apache.hadoop.hbase.mapreduce.ImportTsv’ syntax (Refer Figure HBase.Import/Scan.2). Because this command cannot skip the first row and needs an identifier column to act as the HBASE_ROW_KEY, we perform the following modification to the data file in HDFS: 1) remove the header row and 2) add a new column called ‘ID’ to act as the identifier column. After importing the dataset from HDFS to HBase, we can explore insights from the datasets by executing the Hbase queries on the dataset.

We start with Query 1. To do this, we filter the data using the filter command and specify the different gender and operating system as the filtering condition using the “SingleColumnValueFilter” syntax. This command returns how many rows are in the filtered data which answer our query (Refer Figure Hbase.Query1.1)

Next is Query 2. Because HBase shell cannot run advanced queries with conditional filters, we use a 3rd party tool called Impala to obtain insights from the HBase table. Apache Impala is an open-source SQL query engine for data stored in a computer cluster running Hadoop. Impala is like Hive where SQL-like commands are executed to perform queries. To perform Query 2, we first create a table using the "create external table" syntax. To link the table to the Hbase table, the "stored as" and "SerDeProperties" parameter has been set accordingly (Refer Figure Figure Hbase.Query2.1). The HBase table is then imported to Hive and made visible in Impala using the "invalidate metadata" syntax (to Figure Hbase.Query2.2) Then, we proceed like Hive which is to filter the data by the different age using "where" syntax and average the total usage of each social media platform using the "avg" syntax (Refer Figure Hbase.Query2.4).

Next is Query 3 – 6. These queries are also being executed in Impala. Since Impala are identical to Hive, the query 3 – 6 used in Impala are identical to the query 3 – 6 used in Hive (For query 3, refer Figure Hbase.Query3.1; For query 4, refer Figure Hbase.Query4.1; For query 5, refer Figure Hbase.Query5.1H For query 6, refer Figure Hbase.Query6.1).

3.0 Result

3.1 Execution Time

Table 3 compares the execution times of preliminary actions performed using three popular big data processing tools: Hive, HBase, and Pig. The table shows the time taken in executing these actions. We can see that Hive and Pig have similar performance and HBase took longer execution times.

Table 3

Preliminary actions for Hive, HBase and Pig

Action	Hive	HBase	Pig
Create table	0.195	2.7380	19
Import file from HDFS	0.448	114	-
Scan/select all rows in dataset	0.072	14.05	-
Total	**0.715**	**130.79**	**19**

Table 4 shows the elapsed time taken by Hive, HBase, and Pig to execute each subquery. We visualize Table 4 on a bar chart as shown in Figure 2. We see that for executing queries to gather insight, in terms of total time taken for each query aggregated across the subqueries, HBase is the slowest followed by Pig and Hive.

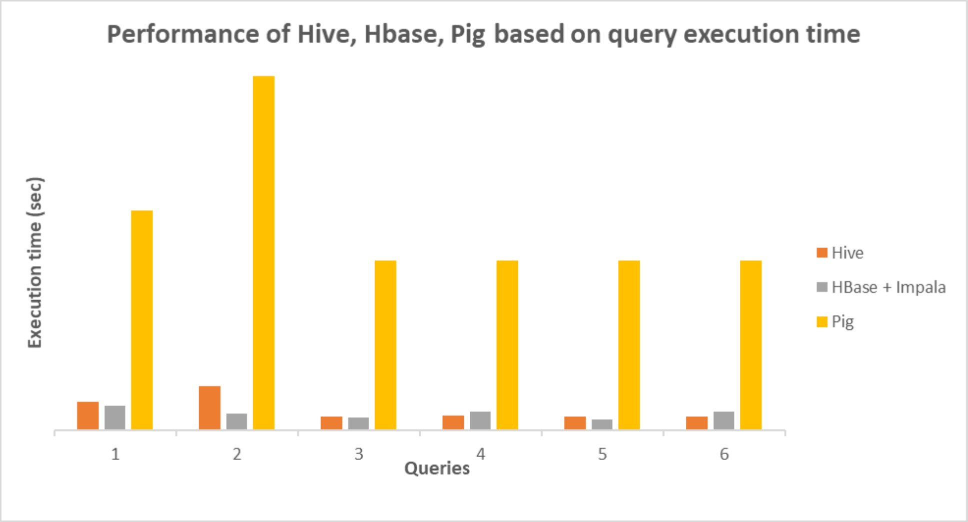
Table 4

Queries and execution time for Hive, HBase and Pig

Queries	Sub-queries	Hive	HBase + Impala	Pig (sec)
1	Male using iOS	1.335	1.5710	62
	Female using iOS	1.366	0.9950	
	Male using Android	1.429	2.1890	
	Female using Android	1.31	1.9590	
	Male using Others	1.329	0.0740	
	Female using Others	1.386	0.0700	
	Total	**8.155**	**6.858**	**62**
2	Age 0-18 (Average Instagram Usage)	1.479	0.22	46
	Age 0-18 (Average Facebook Usage)	1.241	0.87	39
	Age 0-18 (Average WhatsApp Usage)	1.22	0.58	32
	Age 18-60 (Average Instagram Usage)	1.362	0.19	33
	Age 18-60 (Average Facebook Usage)	1.29	0.97	32
	Age 18-60 (Average WhatsApp Usage)	1.566	0.53	32
	Age >60 (Average Instagram Usage)	1.302	0.20	34
	Age >60 (Average Facebook Usage)	1.585	0.52	33
	Age >60 (Average WhatsApp Usage)	1.549	0.71	38
	Total	**12.594**	**4.79**	**319**
3	OS (Highest Instagram Usage record)	1.336	1.82	16
	OS (Highest Facebook Usage record)	1.343	1.09	16
	OS (Highest WhatsApp Usage record)	1.381	0.89	16
	Total	**4.06**	**3.8**	**48**
4	Gender (Highest Instagram Usage record)	1.405	2.00	16
	Gender (Highest Facebook Usage record)	1.485	1.47	16
	Gender (Highest WhatsApp Usage record)	1.337	1.84	16
	Total	**4.227**	**5.31**	**48**
5	Status (Highest Instagram Usage record)	1.265	0.80	16
	Status (Highest Facebook Usage record)	1.306	1.06	16
	Status (Highest WhatsApp Usage record)	1.327	1.26	16
	Total	**3.898**	**3.12**	**48**
6	Education (Highest Instagram Usage record)	1.325	2.30	16
	Education (Highest Facebook Usage record)	1.313	1.40	16
	Education (Highest WhatsApp Usage record)	1.299	1.50	16
	Total	**3.937**	**5.2**	**48**

Figure 2

Performance comparison of the Hive, HBase and Pig



3.2 Code Complexity

Table 4

Code complexity comparison for Hive, Pig and HBase

	Hive	Pig	HBase
Lines of code	Short (34 Lines)	Long (97 Lines)	Fare (38 Lines)
Development time	Rapid development	More development effort	More development effort

Table 4 shows the code complexity comparison for each tool. We observe that Hive requires the shortest lines to execute the query. Pig has longer lines of code. When developing each tool, we also observe that HBase and Pig needs longer development time compared to Hive because 1) Hive can return the result quickly which mean the debugging process can be done rapid 2) Pig's dump command is slow to execute which slow down the debugging process 3) HBase uses basic Create, Read, Update and Delete (CRUD) operations which results in longer development time required.

4.0 Discussion

4.1 Main Findings

Our research problem is that we lack a big data technologies implementation to analyze the usage of preferred social media to understand which platform is universally used. To solve this problem, we set up the HDFS, upload a test dataset on social media trends on HDFS, and perform some queries to analyze the usage of preferred social media in big data technologies implementation which are Hive, Pig and HBase. Then, we compare the tools execution time and code complexity. Our findings are 1) For preliminary action before executing query, Hive performs the best followed by HBase + Impala and Pig 2) For executing queries to gather insight, Impala is the fastest followed by Hive and Pig 3) We observed that Hive can be developed rapidly whereas Pig and HBase requires more effort. We will now proceed to discuss the findings.

Hive outperforms HBase and Pig across for preliminary action that involves table creation and dataset import. This is because Hive supports various file formats and is optimized for columnar storage and effective query processing, hence achieved total time of 0.715 seconds. This could be because Hive is a query language designed for Online Analytical Processing (OLAP) that is used to process and store data in tables (Liu et al., 2022), HBase is a column-oriented database that stores data as a key-value pair (Bhupathiraju & Ravuri, 2014). Data not evenly distributed or if regions are not balanced properly, could lead to uneven data loading and slower import performance. Pig focuses on data manipulation and transformation using Pig Latin scripts which may introduce additional overhead and latency.

For executing queries to gather insight, HBase + Impala is the fastest followed by Hive and Pig. HBase uses Bloom filters to improve reading performance by reducing disk reads (Aiyer, 2012). Although HBase performed well in executing basic count function for Query 2, for advanced conditional queries it had limitations. HBase does not have query optimization mechanisms as it is a NoSQL database providing low-latency random read and write access (Casado & Younas, 2015). In the case of Query 1, the multiple count queries had to be run manually (for individual attribute) for HBase to measure the relationship between gender and operating systems used while single query can be executed in Hive and Pig to obtain the same insights. Pig requires multiple map-reduce jobs when executing every query, resulting in increased latency. When comparing Hive and Impala, both share the same metastore database. Both had comparable executing times when executing queries for analysis. While Hive translates queries to be executed into a series of MapReduce jobs, Impala responds relatively faster. This could be due to Impala having massively parallel processing (MPP) architecture that allows query execution to be in a distributed and parallelized manner. It engages directly with the source (HBase table) and avoids overhead of translating queries into MapReduce jobs such as in Hive.

We observed that Hive can be developed rapidly whereas Pig and HBase require more effort. This is supported by Hive requiring the shortest lines to execute the query with its query optimization techniques. Hive needs the shortest lines of code because Hive is a declarative language where we write the output that we want. On the other hand, Pig is a procedural language where we write the step to get what we want (ProjectPro, 2023; StackOverflow, 2018)

Since Hive can be developed rapidly and its execution time is only slightly slower than Impala and a lot faster than Pig, Hive is the most suitable among the tools to analyze the usage of preferred social media to understand which platform is universally used. This achieves the objective and solves the problem in our analysis. Our limitation is that 1) we did not execute the preliminary action and query in different optimization of the tools. This is important because execution time can vary depending on the optimizations of each tool.

4.2 Additional Findings - Insights on Social Media Trend

We conducted a study and found that Android is the most widely used operating system among the various social media platforms. Interestingly, men tend to prefer Android, while women tend to prefer iOS. Among social media apps, Instagram is the most popular in the 0 to 18 age group, followed by WhatsApp and Facebook, respectively. However, among adults (ages 19 to 60) and the elderly (ages 60 to 100), WhatsApp tops the list of most-used apps. Instagram remains the second most popular app among adults, while Facebook takes this position among older people, followed by Instagram.

In terms of operating systems, Instagram is used on iOS devices, followed by Android. Facebook and WhatsApp, on the other hand, are used more by Android users and then by iOS users. Looking at social media usage by gender, Instagram is more popular among men than women, while Facebook is more popular among women than men. Interestingly, men and women use WhatsApp equally.

Further analysis based on status shows that students use Instagram more than any other app, while professionals use Facebook and WhatsApp. Looking at the use of social media in the different education groups, we see that high school students use Instagram more, postgraduates use Facebook more, and graduates use WhatsApp more than other education categories.

The results obtained from the five queries provide valuable insights into the usage patterns of three social media platforms: Instagram, Facebook, and WhatsApp. Each query presents a breakdown of the average usage across different demographic categories, shedding light on the preferences and trends within each group.

The discussions and results demonstrate that Instagram holds a prominent position among the three social media platforms, consistently capturing the highest average usage across various demographic categories. WhatsApp secures the second spot, while Facebook consistently occupies the least amount of time spent. These findings provide valuable insights into user preferences and usage trends, which can be utilized for marketing strategies and platform development in the realm of social media.

5.0 Conclusion

Our project aims to implement Hive, Pig, HBase to obtain insights from social media usage dataset and compare and select among Hive, Pig HBase based on execution time and code complexity. To achieve this objective, we set up the Hadoop HDFS, upload a test dataset on social media trends on Hadoop HDFS, and perform some queries to analyze the usage of preferred social media in big data technologies implementation which are Hive, Pig and HBase. Our findings are 1) Hive outperforms HBase and Pig across all data for preliminary actions (table creation and data import tasks). 2) For executing queries to gather insights, HBase + Impala is the fastest followed by Hive and Pig. 3) We observed that Hive can be developed rapidly whereas Pig and HBase required more effort. Since Hive can be developed rapidly and its execution time is only slightly slower than Impala and a lot faster than Pig, Hive is the most suitable among the tools to analyze the usage of preferred social media to understand which platform is universally used. This achieves the objective and solves the problem. Our project has significant implications because it is important for data analysts who are analyzing social media data because it tells the data analyst which tools, they should use for this task. It is also important for data engineers who want to encourage their data analyst to analyze social media data because it tells the data engineer how to store the social media data so that the data analyst can analyze the data easily. We recommend future work should test the query using different optimization of the tools to ensure that our findings are generalizable across different optimization settings.

6.0 Reference

Alvarez-Dionisi, L. E. (2017). Envisioning Skills for Adopting, Managing, and Implementing Big Data Technology in the 21st Century. *Information Technology and Computer Science*, 18-25.

Ahmed, O., Fatima-Zahra, B., Ayoub Ait, L., & Samir, B. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 431-448.

Auxier, B., & Anderson, M. (2021, July 7). Social media use in 2021. *Pew Research Center*, 1-14. Retrieved from Social Media Use in 2021: http://www.pewresearch.org

Bhupathiraju, V., & Ravuri, R. P. (2014). The dawn of big data-HBase. In 2014 *Conference on IT in Business, Industry and Government (CSIBIG)* (pp. 1-4). IEEE.

Bishop, M. (2019). Healthcare Social Media for Consumer Informatics. *Consumer Informatics and Digital Health*, 61-86.

Can, U., & Bilal, A. (2017). Big Social Network Data and Sustainable. *Sustainability*, 9(11).

Casado, R., & Younas, M. (2015). Emerging trends and technologies in big data processing. *Concurrency and Computation: Practice and Experience*, *27*(8), 2078-2091.

Chavan, M. V., & Phursule, P. R. (2014). Survey Paper On Big Data. *International Journal of Computer Science and Information Technologies*, 7932-7939.

Jach, T., Magiera, E., & Froelich, W. (2015). Application of Hadoop to store and process big data gathered from an urban water distribution system. *Procedia Engineering*, 1375–1380.

Liu, H., Tang, B., Zhang, J., Deng, Y., Zheng, X., Shen, Q., ... & Luo, Z. (2022). GHive: A Demonstration of GPU-Accelerated Query Processing in Apache Hive. In *Proceedings of the 2022 International Conference on Management of Data* (pp. 2417-2420).

Priya P., S., & Chandrakant P., N. (2014). Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution. *International Journal of Computer Science and Information Technologies*, 2126-2131.

ProjectPro. (2023). Difference between Pig and Hive-The Two Key Components of Hadoop Ecosystem. https://www.projectpro.io/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79

Sebei, H., Hadj Taieb, M. A., & Ben Aouicha, M. (2018). Review of social media analytics process and Big Data pipeline. *Social Network Analysis and Mining*, 8:30.

StackOverflow. (2018). What is the difference between declarative and procedural programming paradigms? https://stackoverflow.com/questions/1619834/what-is-the-difference-between-declarative-and-procedural-programming-paradigms

Storey, V. C., & Song, I.-Y. (2017). Big data technologies and Management: What conceptual modeling can do. *Data & Knowledge Engineering*, 50-67.

Appendix A

Dataset Link

https://www.kaggle.com/datasets/prachirikhari/social-media-usage-trends-india

Table Dataset.1

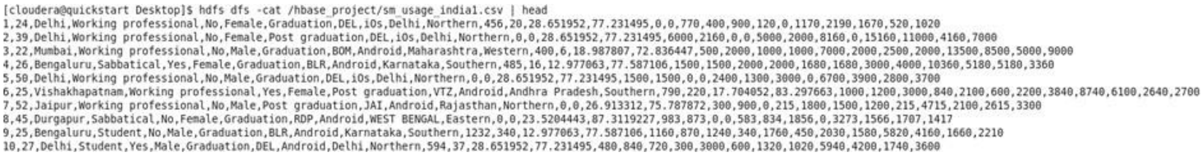
Dataset description

Column	Data Type	Description
Age	Numeric	Integer value to indicate age
City	String	Different cities in India e.g., Agra, Ahmedabad, Allahabad, etc.
Current_Status	String	Categorical value to indicate current working status. i.e., Sabbatical, Self Employed, Student, Working professional
Do_you_own_multiple_profiles_on_Instagram	String	Binary value to indicate one has multiple profile on Instagram i.e. Yes/No
Gender	String	Categorical value to indicate gender. i.e., Male/Female/Non-Binary
Highest_Education	String	Categorical value to indicate education level. i.e., Graduation, High School, Post graduation
Location_City_Airport_Code	String	Different codes to represent different airport in India e.g., AGR, AMD, ATQ, etc.
Phone_OS	String	Categorical value to indicate the phone operating system used by the users. i.e., Android, iOS, Others
State	String	Different states in India e.g., Andhra Pradesh, Assam, Bihar, etc.

Zone	String	Different zones in India e.g., Central, Eastern, North-Eastern, etc.
How_many_followers_do_you_have_on_Instagram	Numeric	Integer value to indicate number of followers on Instagram
How_many_posts_do_you_have_on_Instagram	Numeric	Integer value to indicate number of posts on Instagram
Latitude	Numeric	Floating point value to indicate the coordinate
Longitude	Numeric	Floating point value to indicate the coordinate
Time_Spent_on_Facebook_in_last_week	Numeric	Integer value to indicate time spent on Facebook last week
Time_Spent_on_Facebook_in_last_weekend	Numeric	Integer value to indicate time spent on Facebook last weekend
Time_Spent_on_Instagram_in_last_week	Numeric	Integer value to indicate time spent on Instagram last week
Time_Spent_on_Instagram_in_last_weekend	Numeric	Integer value to indicate time spent on Instagram last weekend
Time_Spent_on_WhatsApp_in_last_week	Numeric	Integer value to indicate time spent on WhatsApp last week
Time_Spent_on_WhatsApp_in_last_weekend	Numeric	Integer value to indicate time spent on WhatsApp last weekend
Total_Facebook_Usage	Numeric	Integer value to indicate total Facebook usage
Total_Instagram_Usage	Numeric	Integer value to indicate total Instagram usage
Total_Social_Media_Usage	Numeric	Integer value to indicate total social media usage
Total_Week_Usage	Numeric	Integer value to indicate total usage in a week
Total_Weekend_Usage	Numeric	Integer value to indicate total usage in a weekend
Total_WhatsApp_Usage	Numeric	Integer value to indicate total WhatsApp usage

HDFS

Figure HDFS.UploadData.1



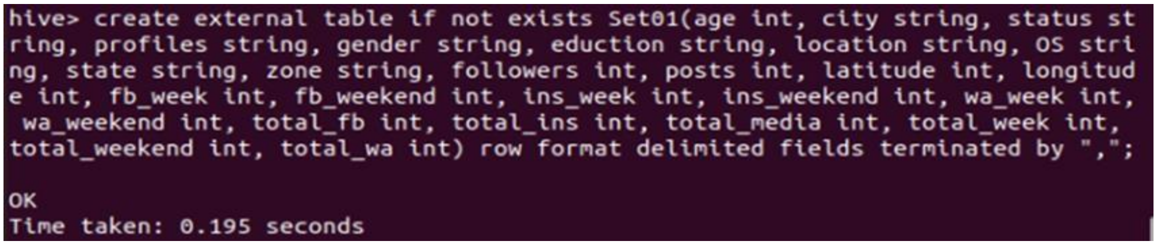
```
[cloudera@quickstart Desktop]$ hdfs dfs -cat /hbase_project/sm_usage_india1.csv | head
1,24,Delhi,Working professional,No,Female,Graduation,DEL,iOs,Delhi,Northern,456,20,28.651952,77.231495,0,0,770,400,900,120,0,1170,2190,1670,520,1020
2,39,Delhi,Working professional,No,Female,Post graduation,DEL,iOs,Delhi,Northern,0,0,28.651952,77.231495,6000,2160,0,0,5000,2000,8160,0,15160,11000,4160,7000
3,22,Mumbai,Working professional,No,Male,Graduation,BOM,Android,Maharashtra,Western,400,6,18.987807,72.836447,500,2000,1000,1000,7000,2000,2500,2000,13500,8500,5000,9000
4,26,Bengaluru,Sabbatical,Yes,Female,Graduation,BLR,Android,Karnataka,Southern,485,16,12.977063,77.587106,1500,1500,2000,2000,1680,1680,3000,4000,10360,5180,5180,3360
5,50,Delhi,Working professional,No,Male,Graduation,DEL,iOs,Delhi,Northern,0,0,28.651952,77.231495,1500,1500,0,0,2400,1300,3000,0,6700,3900,2800,3700
6,25,Vishakhapatnam,Working professional,Yes,Female,Post graduation,VTZ,Android,Andhra Pradesh,Southern,790,220,17.704052,83.297663,1000,1200,3000,840,2100,600,2200,3840,8740,6100,2640,2700
7,52,Jaipur,Working professional,No,Male,Post graduation,JAI,Android,Rajasthan,Northern,0,0,26.913312,75.787872,300,900,0,215,1800,1500,1200,215,4715,2100,2615,3300
8,45,Durgapur,Sabbatical,No,Female,Graduation,RDP,Android,WEST BENGAL,Eastern,0,0,23.5204443,87.3119227,983,873,0,0,583,834,1856,0,3273,1566,1707,1417
9,25,Bengaluru,Student,No,Male,Graduation,BLR,Android,Karnataka,Southern,1232,340,12.977063,77.587106,1160,870,1240,340,1760,450,2030,1580,5820,4160,1660,2210
10,27,Delhi,Student,Yes,Male,Graduation,DEL,Android,Delhi,Northern,594,37,28.651952,77.231495,480,840,720,300,3000,600,1320,1020,5940,4200,1740,3600
```

Appendix B

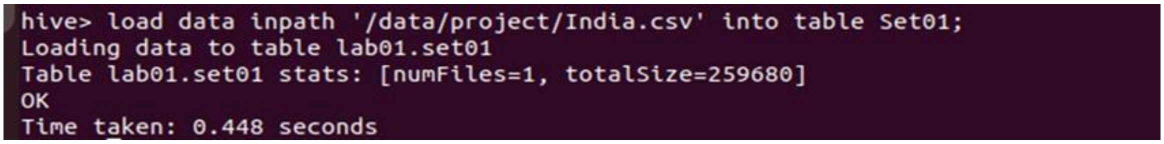
Hive

Figure Hive.Import/Scan.1

Hive import and scan for importing the dataset to Hive



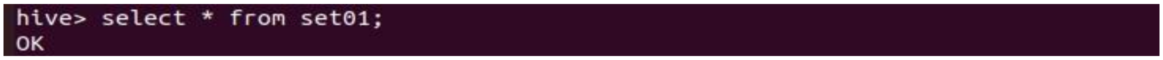
```
hive> create external table if not exists Set01(age int, city string, status st
ring, profiles string, gender string, eduction string, location string, OS stri
ng, state string, zone string, followers int, posts int, latitude int, longitud
e int, fb_week int, fb_weekend int, ins_week int, ins_weekend int, wa_week int,
 wa_weekend int, total_fb int, total_ins int, total_media int, total_week int,
total_weekend int, total_wa int) row format delimited fields terminated by ",";

OK
Time taken: 0.195 seconds
```



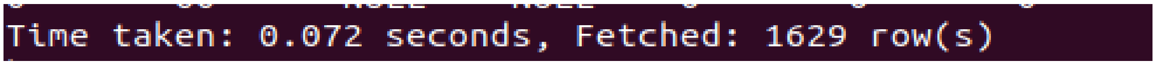
```
hive> load data inpath '/data/project/India.csv' into table Set01;
Loading data to table lab01.set01
Table lab01.set01 stats: [numFiles=1, totalSize=259680]
OK
Time taken: 0.448 seconds
```

Figure Hive.Import/Scan.2

Hive import and scan for code of selecting all rows in the dataset



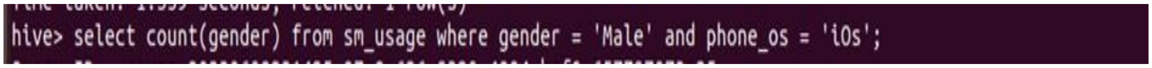
```
hive> select * from set01;
OK
```



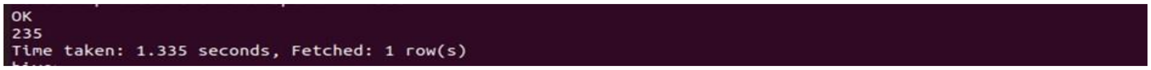
```
Time taken: 0.072 seconds, Fetched: 1629 row(s)
```

Figure Hive.Query1.1

Query 1 for Hive male using iOS



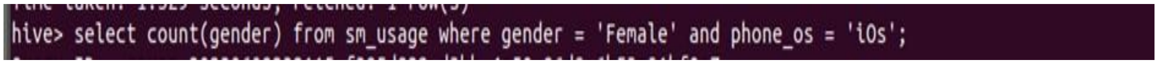
```
hive> select count(gender) from sm_usage where gender = 'Male' and phone_os = 'iOs';
```



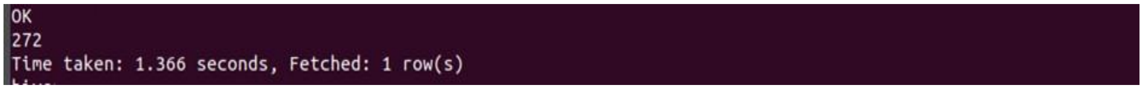
```
OK
235
Time taken: 1.335 seconds, Fetched: 1 row(s)
```

Figure Hive.Query1.2

Query 1 for Hive female using iOS



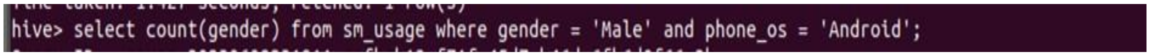
```
hive> select count(gender) from sm_usage where gender = 'Female' and phone_os = 'iOs';
```



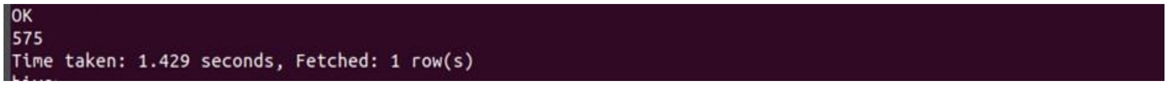
```
OK
272
Time taken: 1.366 seconds, Fetched: 1 row(s)
```

Figure Hive.Query1.3

Query 1 for Hive male using android



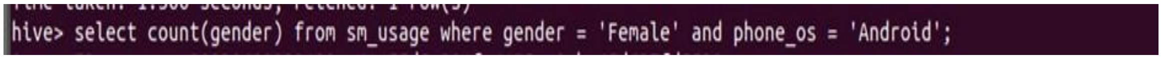
```
hive> select count(gender) from sm_usage where gender = 'Male' and phone_os = 'Android';
```



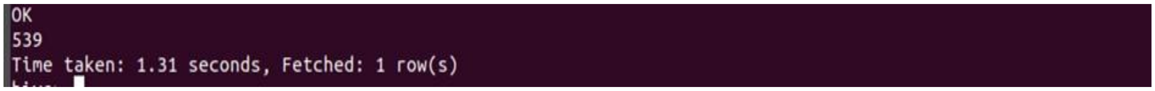
```
OK
575
Time taken: 1.429 seconds, Fetched: 1 row(s)
```

Figure Hive.Query1.4

Query 1 for Hive female using android



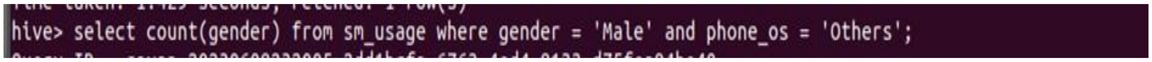
```
hive> select count(gender) from sm_usage where gender = 'Female' and phone_os = 'Android';
```



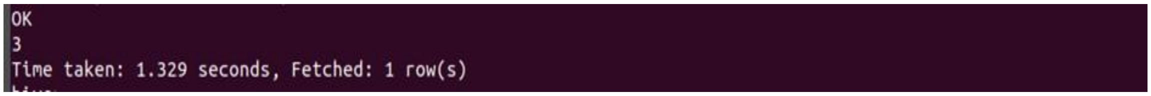
```
OK
539
Time taken: 1.31 seconds, Fetched: 1 row(s)
```

Figure Hive.Query1.5

Query 1 for Hive male using other operating systems



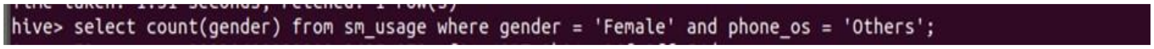
```
hive> select count(gender) from sm_usage where gender = 'Male' and phone_os = 'Others';
```



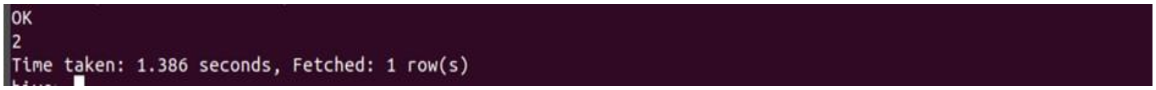
```
OK
3
Time taken: 1.329 seconds, Fetched: 1 row(s)
```

Figure Hive.Query1.6

Query 1 for Hive female using other operating systems



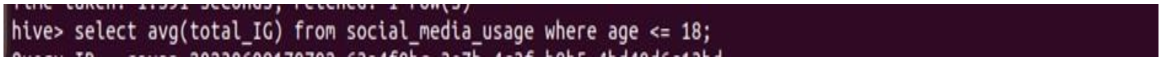
```
hive> select count(gender) from sm_usage where gender = 'Female' and phone_os = 'Others';
```



```
OK
2
Time taken: 1.386 seconds, Fetched: 1 row(s)
```

***Figure* Hive.Query2.1**

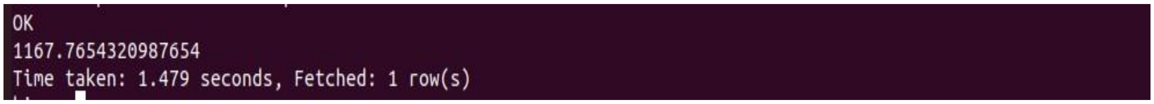
Query 2 for Hive code of selecting average Instagram usage for age 0-18



```
hive> select avg(total_IG) from social_media_usage where age <= 18;
```

Figure Hive.Query2.2

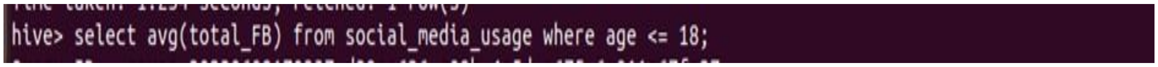
Query 2 for Hive results of selecting average Instagram usage for age 0-18



```
OK
1167.7654320987654
Time taken: 1.479 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.3

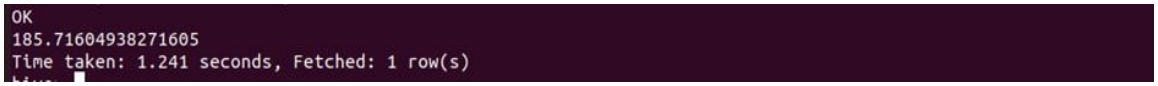
Query 2 for Hive code of selecting average Facebook usage for age 0-18



```
hive> select avg(total_FB) from social_media_usage where age <= 18;
```

Figure Hive.Query2.4

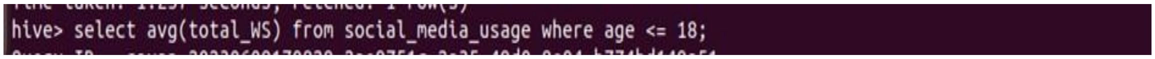
Query 2 for Hive results of selecting average Facebook usage for age 0-18



```
OK
185.71604938271605
Time taken: 1.241 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.5

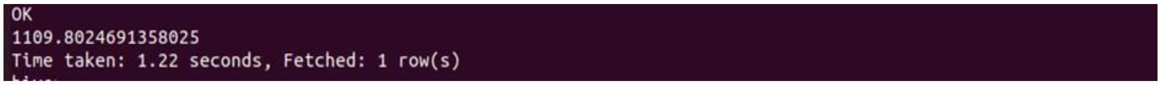
Query 2 for Hive code of selecting average WhatsApp usage for age 0-18



```
hive> select avg(total_WS) from social_media_usage where age <= 18;
```

Figure Hive.Query2.6

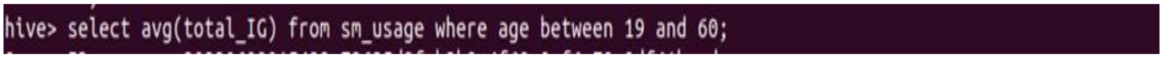
Query 2 for Hive results of selecting average WhatsApp usage for age 0-18



```
OK
1109.8024691358025
Time taken: 1.22 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.7

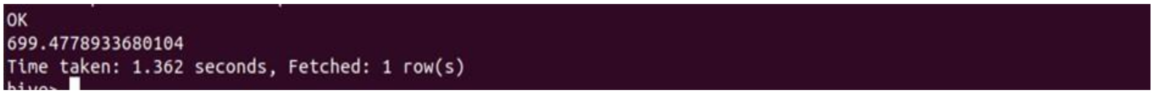
Query 2 for Hive code of selecting average Instagram usage for age19-60



```
hive> select avg(total_IG) from sm_usage where age between 19 and 60;
```

Figure Hive.Query2.8

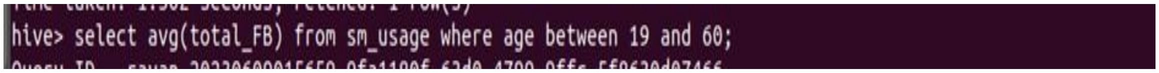
Query 2 for Hive results of selecting average Instagram usage for age 19-60



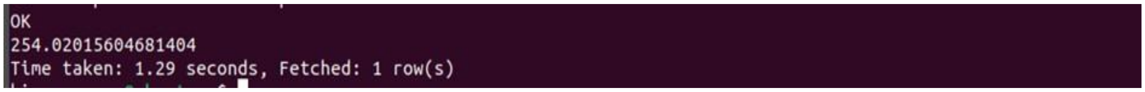
```
OK
699.4778933680104
Time taken: 1.362 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.9

Query 2 for Hive code and results of selecting average Facebook usage for age 19-60



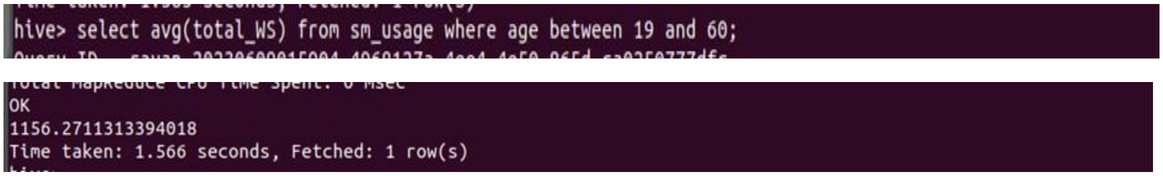
```
hive> select avg(total_FB) from sm_usage where age between 19 and 60;
```



```
OK
254.02015604681404
Time taken: 1.29 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.10

Query 2 for Hive code and results of selecting average WhatsApp usage for age 19-60

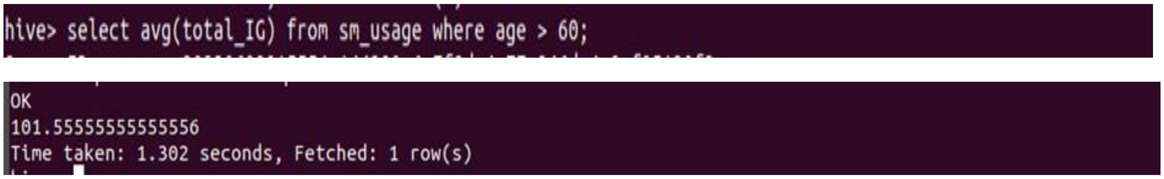


```
hive> select avg(total_WS) from sm_usage where age between 19 and 60;
```

```
Total MapReduce CPU Time Spent: 0 msec
OK
1156.2711313394018
Time taken: 1.566 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.11

Query 2 for Hive code and results of selecting average Instagram usage for age 60-100

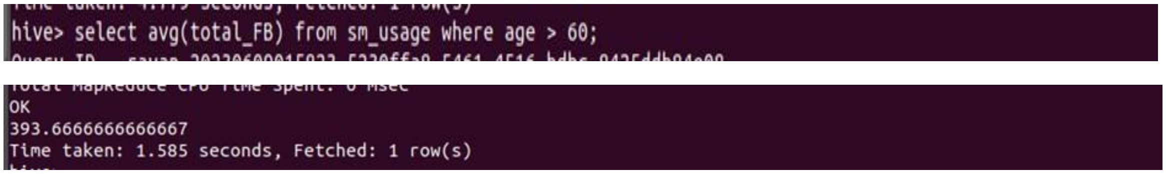


```
hive> select avg(total_IG) from sm_usage where age > 60;
```

```
OK
101.55555555555556
Time taken: 1.302 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.12

Query 2 for Hive code and results of selecting average Facebook usage for age 60-100

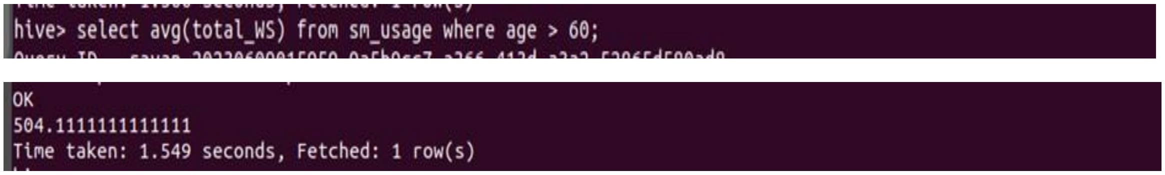


```
hive> select avg(total_FB) from sm_usage where age > 60;
```

```
Total MapReduce CPU Time Spent: 0 msec
OK
393.6666666666667
Time taken: 1.585 seconds, Fetched: 1 row(s)
```

Figure Hive.Query2.13

Query 2 for Hive code and results of selecting average WhatsApp usage for age 60-100

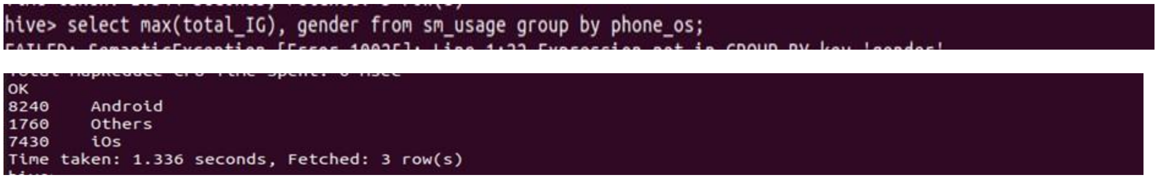


```
hive> select avg(total_WS) from sm_usage where age > 60;
```

```
OK
504.1111111111111
Time taken: 1.549 seconds, Fetched: 1 row(s)
```

Figure Hive.Query3.1

Query 3 for Hive code and results of selecting highest Instagram usage for operating systems

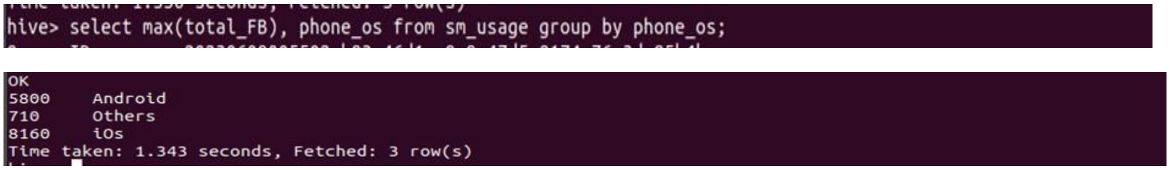


```
hive> select max(total_IG), gender from sm_usage group by phone_os;
FAILED: SemanticException [Error 10025]: Line 1:22 Expression not in GROUP BY key 'gender'
```

```
Total MapReduce CPU Time Spent: 0 msec
OK
8240    Android
1760    Others
7430    iOs
Time taken: 1.336 seconds, Fetched: 3 row(s)
```

Figure Hive.Query3.2

Query 3 for Hive code and results of selecting highest Facebook usage for operating systems

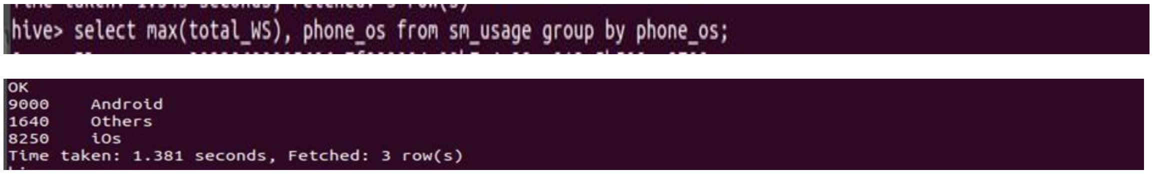


```
hive> select max(total_FB), phone_os from sm_usage group by phone_os;
```

```
OK
5800    Android
710     Others
8160    iOs
Time taken: 1.343 seconds, Fetched: 3 row(s)
```

Figure Hive.Query3.3

Query 3 for Hive code and results of selecting highest WhatsApp usage for operating systems

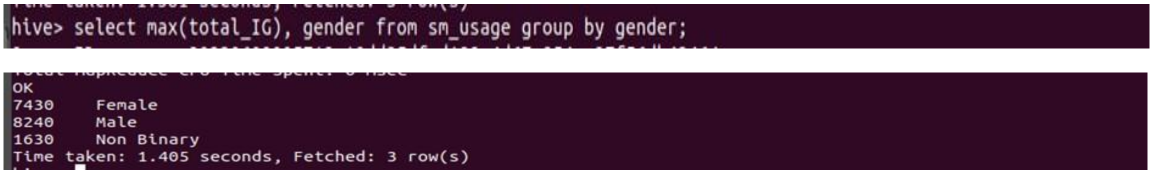


```
hive> select max(total_WS), phone_os from sm_usage group by phone_os;
```

```
OK
9000    Android
1640    Others
8250    iOs
Time taken: 1.381 seconds, Fetched: 3 row(s)
```

Figure Hive.Query4.1

Query 4 for Hive code and results of selecting highest Instagram usage for gender

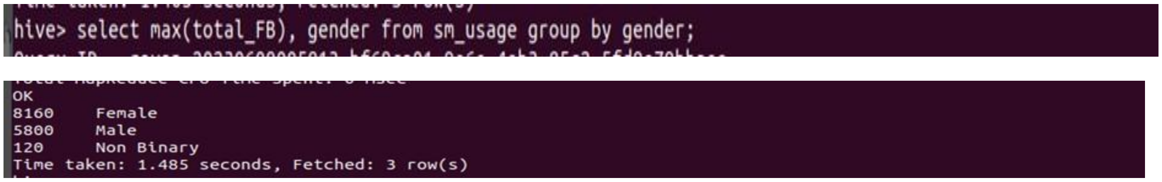


```
hive> select max(total_IG), gender from sm_usage group by gender;
```

```
OK
7430    Female
8240    Male
1630    Non Binary
Time taken: 1.405 seconds, Fetched: 3 row(s)
```

Figure Hive.Query4.2

Query 4 for Hive code and results of selecting highest Facebook usage for gender

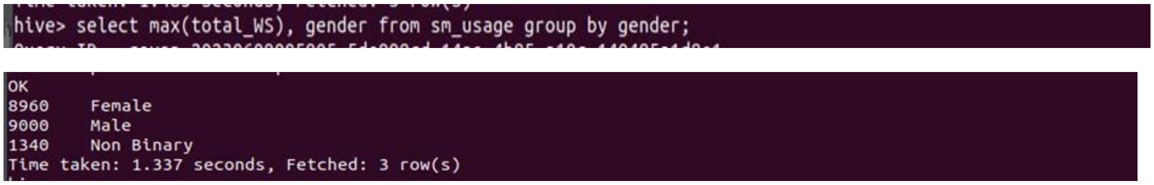


```
hive> select max(total_FB), gender from sm_usage group by gender;
```

```
OK
8160    Female
5800    Male
120     Non Binary
Time taken: 1.485 seconds, Fetched: 3 row(s)
```

Figure Hive.Query4.3

Query 4 for Hive code and results of selecting highest WhatsApp usage for gender

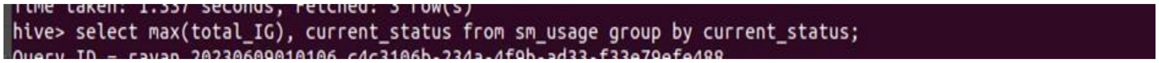


```
hive> select max(total_WS), gender from sm_usage group by gender;
```

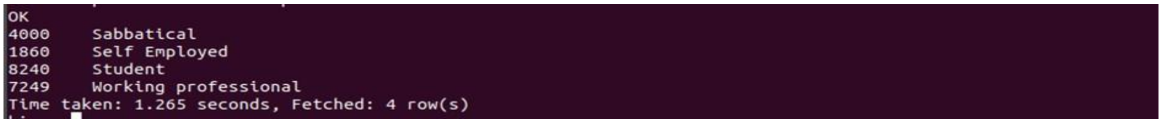
```
OK
8960    Female
9000    Male
1340    Non Binary
Time taken: 1.337 seconds, Fetched: 3 row(s)
```

Figure Hive.Query5.1

Query 5 for Hive code and results of selecting highest Instagram usage for status



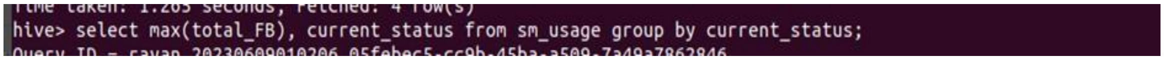
```
Time taken: 1.337 seconds, Fetched: 3 row(s)
hive> select max(total_IG), current_status from sm_usage group by current_status;
Query ID = ravan_20230609010106_c4c3106b-234a-4f9b-ad33-f33e79efe488
```



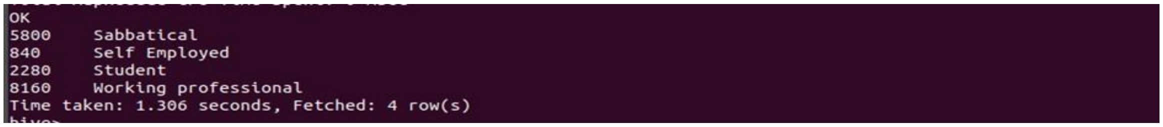
```
OK
4000    Sabbatical
1860    Self Employed
8240    Student
7249    Working professional
Time taken: 1.265 seconds, Fetched: 4 row(s)
```

Figure Hive.Query5.2

Query 5 for Hive code and results of selecting highest Facebook usage for status



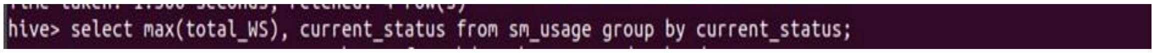
```
Time taken: 1.265 seconds, Fetched: 4 row(s)
hive> select max(total_FB), current_status from sm_usage group by current_status;
Query ID = ravan_20230609010206_05febec5-cc9b-45ba-a509-7a49a7862846
```



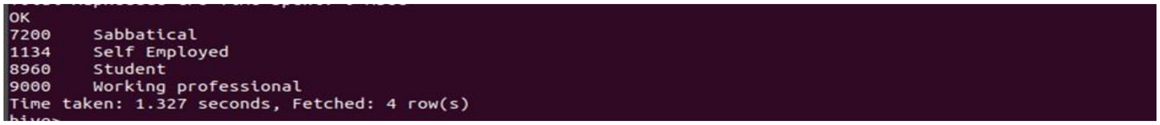
```
OK
5800    Sabbatical
840     Self Employed
2280    Student
8160    Working professional
Time taken: 1.306 seconds, Fetched: 4 row(s)
hive>
```

Figure Hive.Query5.3

Query 5 for Hive code and results of selecting highest WhatsApp usage for status



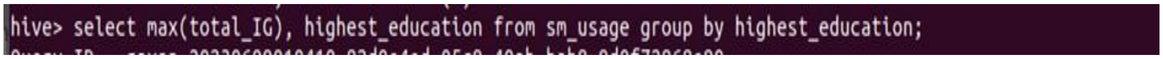
```
hive> select max(total_WS), current_status from sm_usage group by current_status;
```



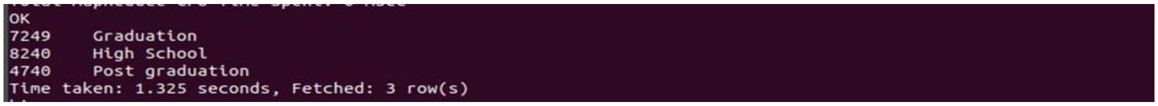
```
OK
7200    Sabbatical
1134    Self Employed
8960    Student
9000    Working professional
Time taken: 1.327 seconds, Fetched: 4 row(s)
hive>
```

Figure Hive.Query6.1

Query 6 for Hive code and results of selecting highest Instagram usage for education



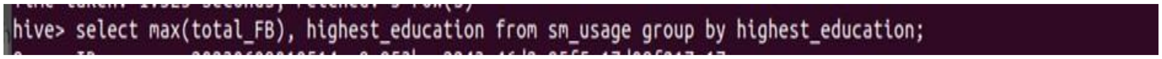
```
hive> select max(total_IG), highest_education from sm_usage group by highest_education;
```



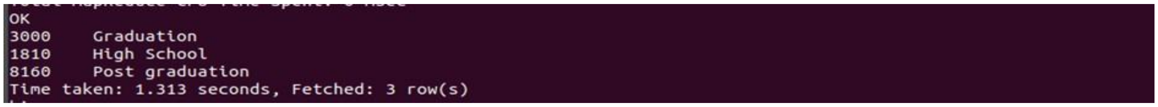
```
OK
7249    Graduation
8240    High School
4740    Post graduation
Time taken: 1.325 seconds, Fetched: 3 row(s)
```

Figure Hive.Query6.2

Query 6 for Hive code and results of selecting highest Facebook usage for education



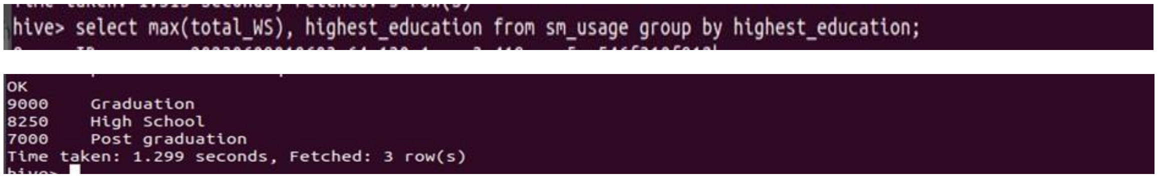
```
hive> select max(total_FB), highest_education from sm_usage group by highest_education;
```



```
OK
3000    Graduation
1810    High School
8160    Post graduation
Time taken: 1.313 seconds, Fetched: 3 row(s)
```

Figure Hive.Query6.3

Query 6 for Hive code and results of selecting highest WhatsApp usage for education



```
hive> select max(total_WS), highest_education from sm_usage group by highest_education;
```

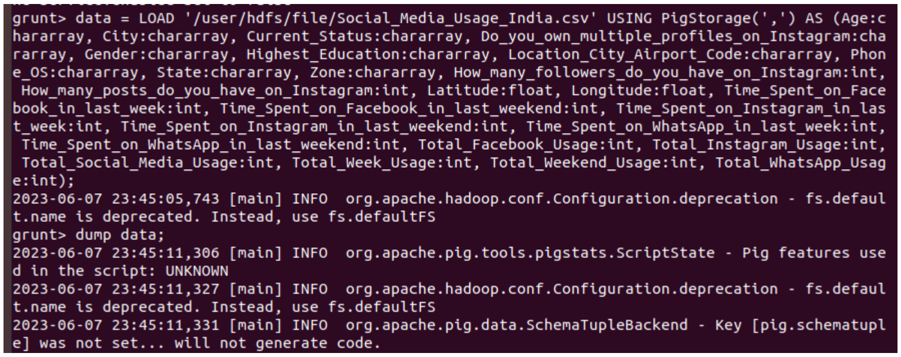
```
OK
9000    Graduation
8250    High School
7000    Post graduation
Time taken: 1.299 seconds, Fetched: 3 row(s)
hive>
```

Appendix C

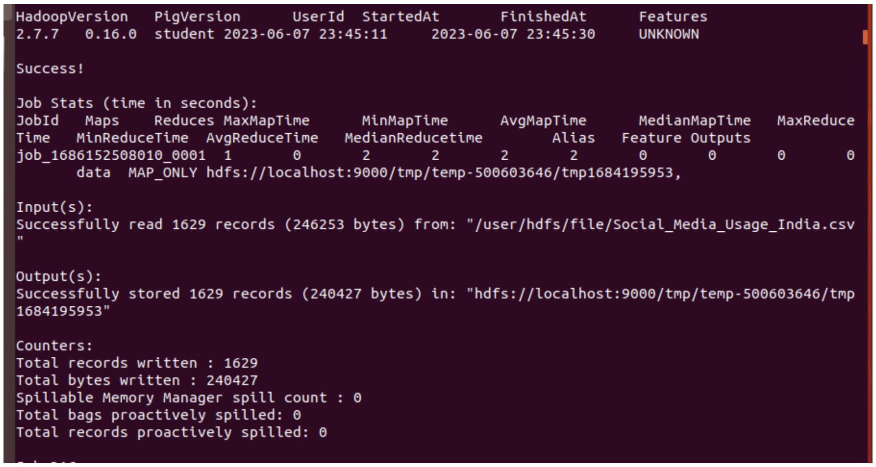
Pig

Figure Pig.Import/Scan.1

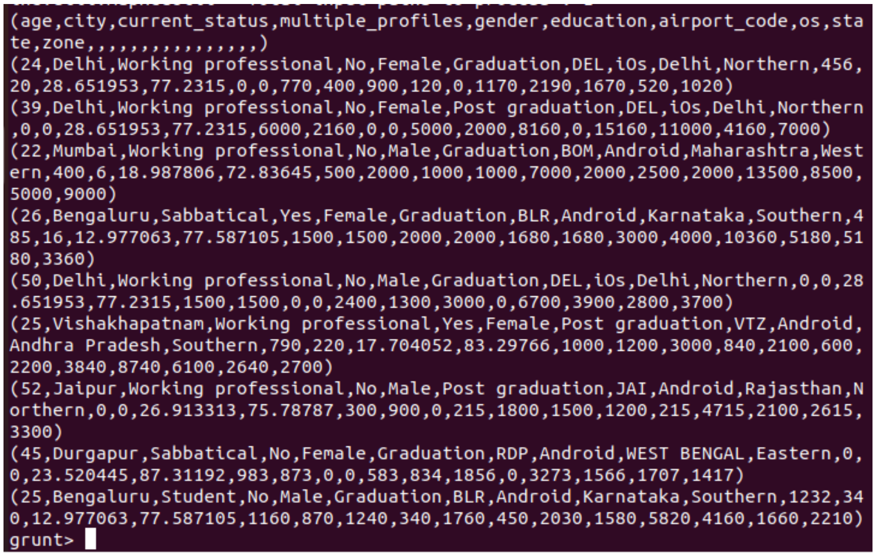
Import and scan for Pig create table, load data from HDFS and shows all data



```
grunt> data = LOAD '/user/hdfs/file/Social_Media_Usage_India.csv' USING PigStorage(',') AS (Age:c
hararray, City:chararray, Current_Status:chararray, Do_you_own_multiple_profiles_on_Instagram:cha
rarray, Gender:chararray, Highest_Education:chararray, Location_City_Airport_Code:chararray, Phon
e_OS:chararray, State:chararray, Zone:chararray, How_many_followers_do_you_have_on_Instagram:int,
 How_many_posts_do_you_have_on_Instagram:int, Latitude:float, Longitude:float, Time_Spent_on_Face
book_in_last_week:int, Time_Spent_on_Facebook_in_last_weekend:int, Time_Spent_on_Instagram_in_las
t_week:int, Time_Spent_on_Instagram_in_last_weekend:int, Time_Spent_on_WhatsApp_in_last_week:int,
 Time_Spent_on_WhatsApp_in_last_weekend:int, Total_Facebook_Usage:int, Total_Instagram_Usage:int,
 Total_Social_Media_Usage:int, Total_Week_Usage:int, Total_Weekend_Usage:int, Total_WhatsApp_Usag
e:int);
2023-06-07 23:45:05,743 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.defaul
t.name is deprecated. Instead, use fs.defaultFS
grunt> dump data;
2023-06-07 23:45:11,306 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: UNKNOWN
2023-06-07 23:45:11,327 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.defaul
t.name is deprecated. Instead, use fs.defaultFS
2023-06-07 23:45:11,331 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematupl
e] was not set... will not generate code.
```



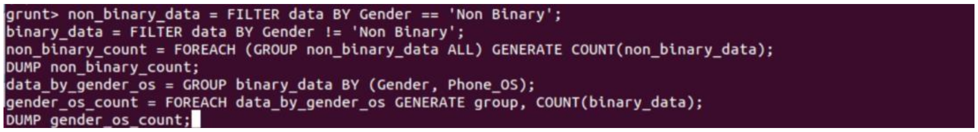
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-07 23:45:11     2023-06-07 23:45:30     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0001  1       0       2       2       2       2       0       0       0       0
        data    MAP_ONLY hdfs://localhost:9000/tmp/temp-500603646/tmp1684195953,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 1629 records (240427 bytes) in: "hdfs://localhost:9000/tmp/temp-500603646/tmp
1684195953"

Counters:
Total records written : 1629
Total bytes written : 240427
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



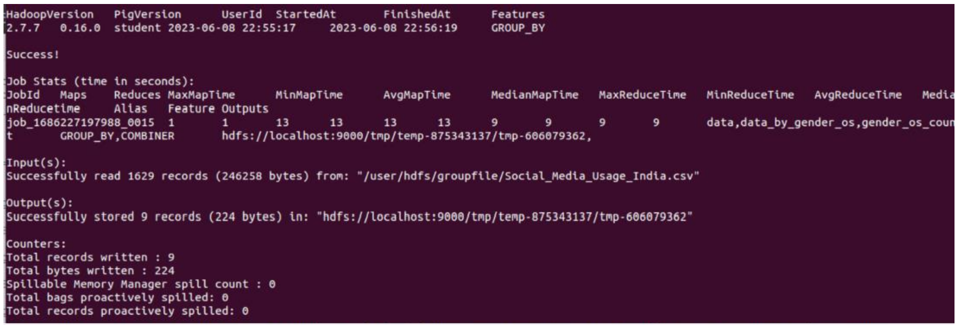
```
(age,city,current_status,multiple_profiles,gender,education,airport_code,os,sta
te,zone,,,,,,,,,,,,,,,)
(24,Delhi,Working professional,No,Female,Graduation,DEL,iOs,Delhi,Northern,456,
20,28.651953,77.2315,0,0,770,400,900,120,0,1170,2190,1670,520,1020)
(39,Delhi,Working professional,No,Female,Post graduation,DEL,iOs,Delhi,Northern
,0,0,28.651953,77.2315,6000,2160,0,0,5000,2000,8160,0,15160,11000,4160,7000)
(22,Mumbai,Working professional,No,Male,Graduation,BOM,Android,Maharashtra,West
ern,400,6,18.987806,72.83645,500,2000,1000,1000,7000,2000,2500,2000,13500,8500,
5000,9000)
(26,Bengaluru,Sabbatical,Yes,Female,Graduation,BLR,Android,Karnataka,Southern,4
85,16,12.977063,77.587105,1500,1500,2000,2000,1680,1680,3000,4000,10360,5180,51
80,3360)
(50,Delhi,Working professional,No,Male,Graduation,DEL,iOs,Delhi,Northern,0,0,28
.651953,77.2315,1500,1500,0,0,2400,1300,3000,0,6700,3900,2800,3700)
(25,Vishakhapatnam,Working professional,Yes,Female,Post graduation,VTZ,Android,
Andhra Pradesh,Southern,790,220,17.704052,83.29766,1000,1200,3000,840,2100,600,
2200,3840,8740,6100,2640,2700)
(52,Jaipur,Working professional,No,Male,Post graduation,JAI,Android,Rajasthan,N
orthern,0,0,26.913313,75.78787,300,900,0,215,1800,1500,1200,215,4715,2100,2615,
3300)
(45,Durgapur,Sabbatical,No,Female,Graduation,RDP,Android,WEST BENGAL,Eastern,0,
0,23.520445,87.31192,983,873,0,0,583,834,1856,0,3273,1566,1707,1417)
(25,Bengaluru,Student,No,Male,Graduation,BLR,Android,Karnataka,Southern,1232,34
0,12.977063,77.587105,1160,870,1240,340,1760,450,2030,1580,5820,4160,1660,2210)
grunt> 
```

Figure Pig.Query1.1

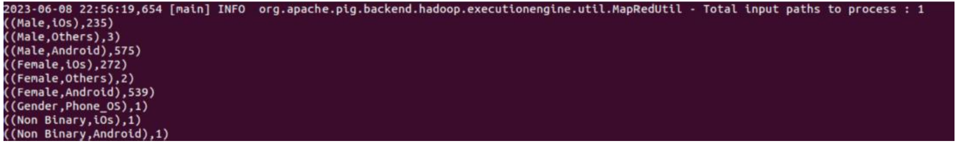
Query 1 for Pig count number of male and female using android, iOS and other operating systems



```
grunt> non_binary_data = FILTER data BY Gender == 'Non Binary';
binary_data = FILTER data BY Gender != 'Non Binary';
non_binary_count = FOREACH (GROUP non_binary_data ALL) GENERATE COUNT(non_binary_data);
DUMP non_binary_count;
data_by_gender_os = GROUP binary_data BY (Gender, Phone_OS);
gender_os_count = FOREACH data_by_gender_os GENERATE group, COUNT(binary_data);
DUMP gender_os_count;
```



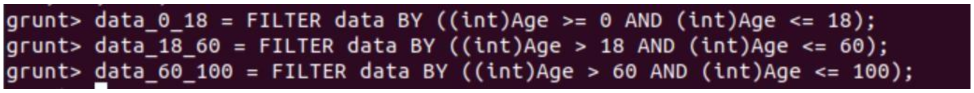
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 22:55:17     2023-06-08 22:56:19     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   Media
nReducetime     Alias   Feature Outputs
job_1686227197988_0015  1       1       13      13      13      13      9       9       9       9       data,data_by_gender_os,gender_os_coun
t       GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp-875343137/tmp-606079362,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 9 records (224 bytes) in: "hdfs://localhost:9000/tmp/temp-875343137/tmp-606079362"

Counters:
Total records written : 9
Total bytes written : 224
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



```
2023-06-08 22:56:19,654 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
((Male,iOs),235)
((Male,Others),3)
((Male,Android),575)
((Female,iOs),272)
((Female,Others),2)
((Female,Android),539)
((Gender,Phone_OS),1)
((Non Binary,iOs),1)
((Non Binary,Android),1)
```

Figure Pig.Query2.1

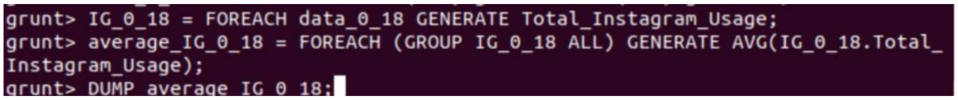
Query 2 for Pig defining ranges for age groups



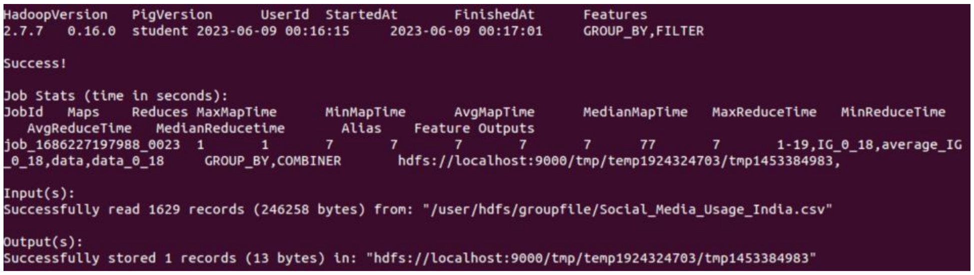
```
grunt> data_0_18 = FILTER data BY ((int)Age >= 0 AND (int)Age <= 18);
grunt> data_18_60 = FILTER data BY ((int)Age > 18 AND (int)Age <= 60);
grunt> data_60_100 = FILTER data BY ((int)Age > 60 AND (int)Age <= 100);
```

Figure Pig.Query2.2

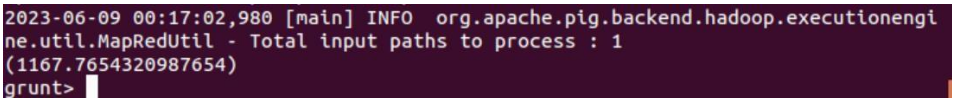
Query 2 for Pig average Instagram usage for age 0-18



```
grunt> IG_0_18 = FOREACH data_0_18 GENERATE Total_Instagram_Usage;
grunt> average_IG_0_18 = FOREACH (GROUP IG_0_18 ALL) GENERATE AVG(IG_0_18.Total_
Instagram_Usage);
grunt> DUMP average_IG_0_18;
```



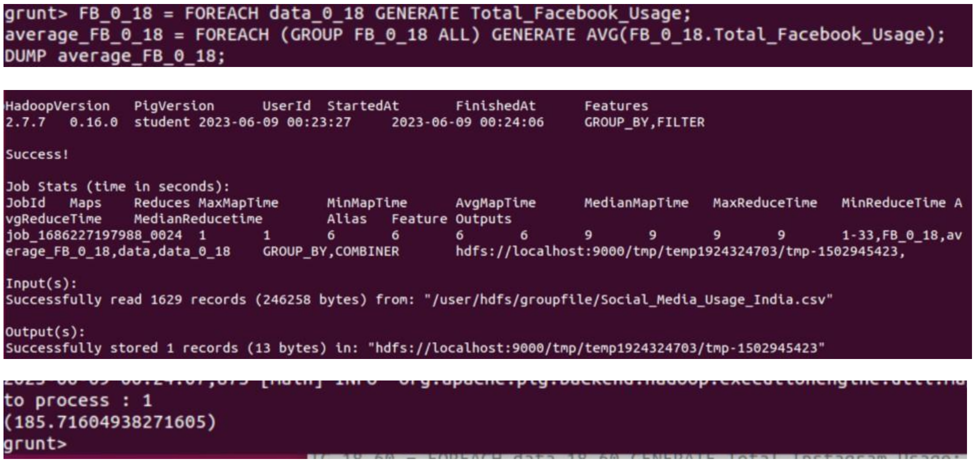
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:16:15     2023-06-09 00:17:01     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime
   AvgReduceTime   MedianReducetime       Alias    Feature Outputs
job_1686227197988_0023  1       1       7       7       7       7       7       77      7       1-19,IG_0_18,average_IG
_0_18,data,data_0_18     GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp1453384983,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp1453384983"
```



```
2023-06-09 00:17:02,980 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(1167.7654320987654)
grunt> 
```

Figure Pig.Query2.3

Query 2 for Pig average Facebook usage for age 0-18



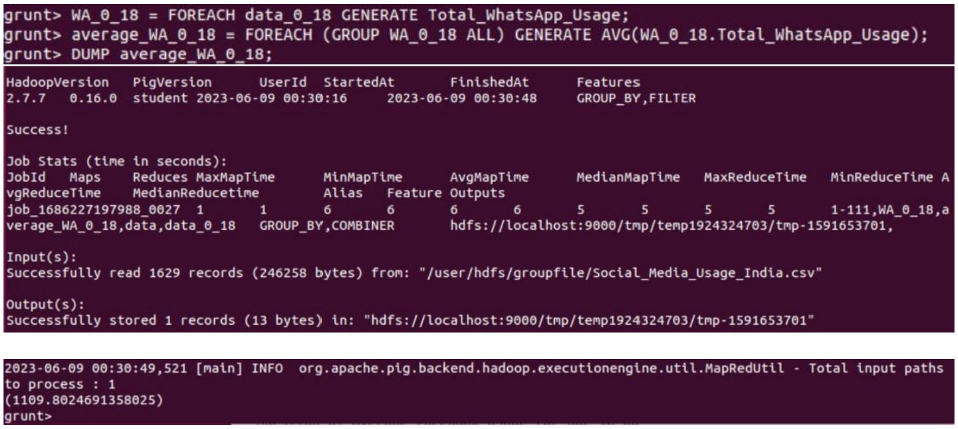
```
grunt> FB_0_18 = FOREACH data_0_18 GENERATE Total_Facebook_Usage;
average_FB_0_18 = FOREACH (GROUP FB_0_18 ALL) GENERATE AVG(FB_0_18.Total_Facebook_Usage);
DUMP average_FB_0_18;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:23:27     2023-06-09 00:24:06     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime A
vgReduceTime    MedianReducetime        Alias   Feature Outputs
job_1686227197988_0024  1       1       6       6       6       6       9       9       9       9       1-33,FB_0_18,av
erage_FB_0_18,data,data_0_18    GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp-1502945423,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp-1502945423"
```

```
2023-06-09 00:24:07,875 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.Ma
to process : 1
(185.71604938271605)
grunt>
```

Figure Pig.Query2.4

Query 2 for Pig average WhatsApp usage for age 0-18



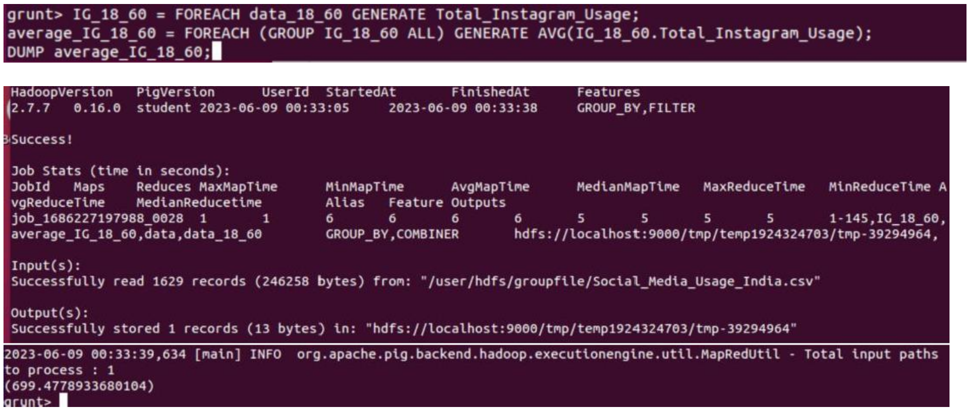
```
grunt> WA_0_18 = FOREACH data_0_18 GENERATE Total_WhatsApp_Usage;
grunt> average_WA_0_18 = FOREACH (GROUP WA_0_18 ALL) GENERATE AVG(WA_0_18.Total_WhatsApp_Usage);
grunt> DUMP average_WA_0_18;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:30:16     2023-06-09 00:30:48     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime A
vgReduceTime    MedianReducetime        Alias   Feature Outputs
job_1686227197988_0027  1       1       6       6       6       6       5       5       5       5       1-111,WA_0_18,a
verage_WA_0_18,data,data_0_18   GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp-1591653701,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp-1591653701"
```

```
2023-06-09 00:30:49,521 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths
to process : 1
(1109.8024691358025)
grunt>
```

Figure Pig.Query2.5

Query 2 for Pig average Instagram usage for age 18-60

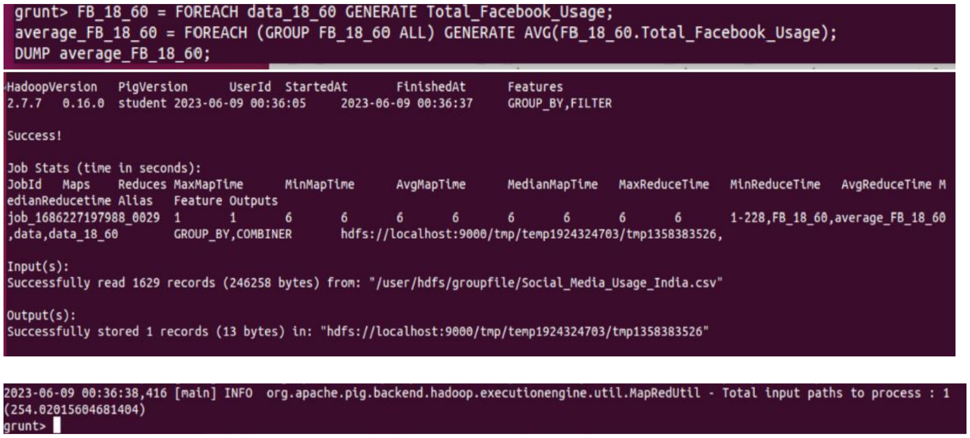


```
grunt> IG_18_60 = FOREACH data_18_60 GENERATE Total_Instagram_Usage;
average_IG_18_60 = FOREACH (GROUP IG_18_60 ALL) GENERATE AVG(IG_18_60.Total_Instagram_Usage);
DUMP average_IG_18_60;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:33:05     2023-06-09 00:33:38     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime A
vgReduceTime    MedianReducetime        Alias   Feature Outputs
job_1686227197988_0028  1       1       6       6       6       6       5       5       5       5       1-145,IG_18_60,
average_IG_18_60,data,data_18_60        GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp-39294964,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp-39294964"
2023-06-09 00:33:39,634 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths
to process : 1
(699.4778933680104)
grunt>
```

Figure Pig.Query2.6

Query 2 for Pig average Facebook usage for age 18-60



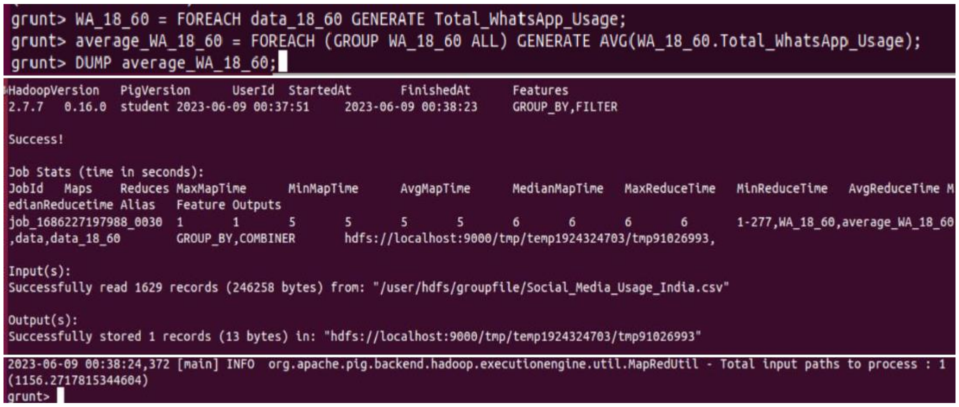
```
grunt> FB_18_60 = FOREACH data_18_60 GENERATE Total_Facebook_Usage;
average_FB_18_60 = FOREACH (GROUP FB_18_60 ALL) GENERATE AVG(FB_18_60.Total_Facebook_Usage);
DUMP average_FB_18_60;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:36:05     2023-06-09 00:36:37     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime M
edianReducetime Alias   Feature Outputs
job_1686227197988_0029  1       1       6       6       6       6       6       6       6       6       1-228,FB_18_60,average_FB_18_60
,data,data_18_60        GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp1358383526,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp1358383526"
```

```
2023-06-09 00:36:38,416 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(254.02015604681404)
grunt>
```

Figure Pig.Query2.7

Query 2 for Pig average WhatsApp usage for age 18-60

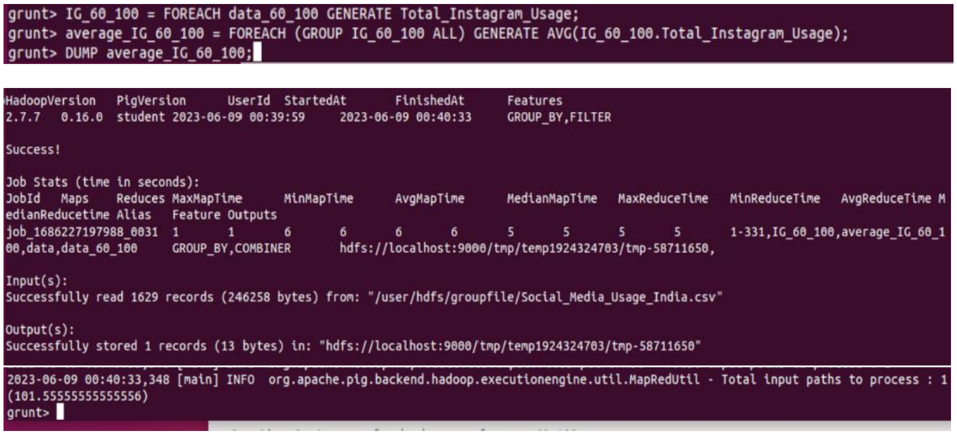


```
grunt> WA_18_60 = FOREACH data_18_60 GENERATE Total_WhatsApp_Usage;
grunt> average_WA_18_60 = FOREACH (GROUP WA_18_60 ALL) GENERATE AVG(WA_18_60.Total_WhatsApp_Usage);
grunt> DUMP average_WA_18_60;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:37:51     2023-06-09 00:38:23     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime M
edianReducetime Alias   Feature Outputs
job_1686227197988_0030  1       1       5       5       5       5       6       6       6       6       1-277,WA_18_60,average_WA_18_60
,data,data_18_60        GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp91026993,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp91026993"
2023-06-09 00:38:24,372 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(1156.2717815344604)
grunt>
```

Figure Pig.Query2.8

Query 2 for Pig average Instagram usage for age 60-100



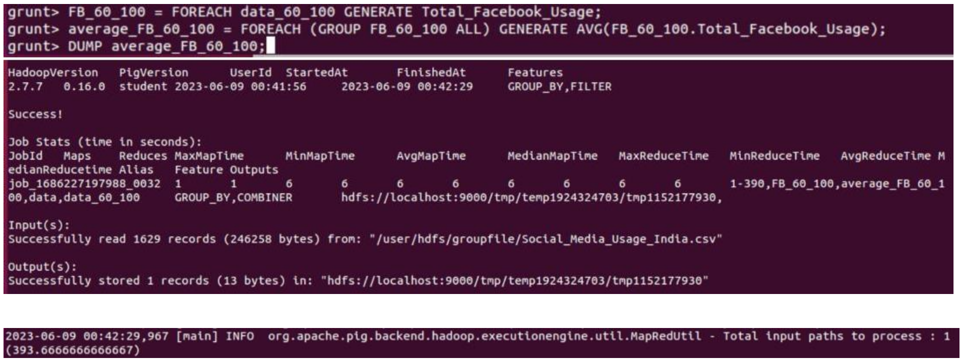
```
grunt> IG_60_100 = FOREACH data_60_100 GENERATE Total_Instagram_Usage;
grunt> average_IG_60_100 = FOREACH (GROUP IG_60_100 ALL) GENERATE AVG(IG_60_100.Total_Instagram_Usage);
grunt> DUMP average_IG_60_100;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:39:59     2023-06-09 00:40:33     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime M
edianReducetime Alias   Feature Outputs
job_1686227197988_0031  1       1       6       6       6       6       5       5       5       5       1-331,IG_60_100,average_IG_60_1
00,data,data_60_100     GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp-58711650,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp-58711650"
```

```
2023-06-09 00:40:33,348 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101.55555555555556)
grunt>
```

Figure Pig.Query2.9

Query 2 for Pig average Facebook usage for age 60-100



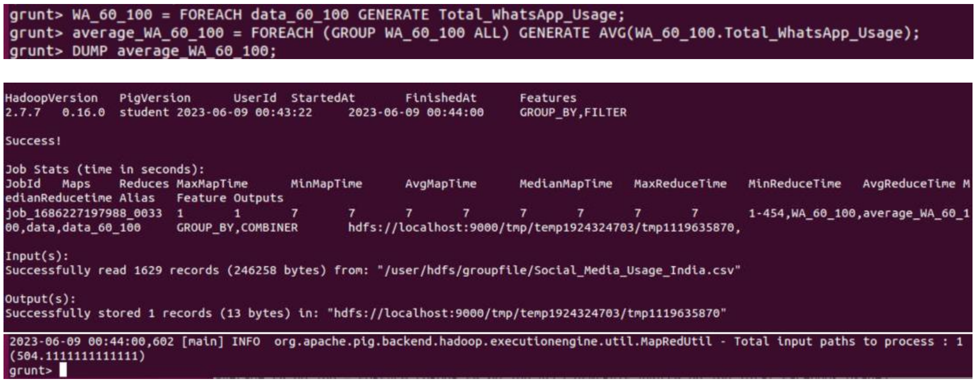
```
grunt> FB_60_100 = FOREACH data_60_100 GENERATE Total_Facebook_Usage;
grunt> average_FB_60_100 = FOREACH (GROUP FB_60_100 ALL) GENERATE AVG(FB_60_100.Total_Facebook_Usage);
grunt> DUMP average_FB_60_100;
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:41:56     2023-06-09 00:42:29     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime M
edianReducetime Alias   Feature Outputs
job_1686227197988_0032  1       1       6       6       6       6       6       6       6       6       1-390,FB_60_100,average_FB_60_1
00,data,data_60_100     GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp1152177930,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp1152177930"
```

```
2023-06-09 00:42:29,967 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(393.6666666666667)
```

Figure Pig.Query2.10

Query 2 for Pig average WhatsApp usage for age 60-100

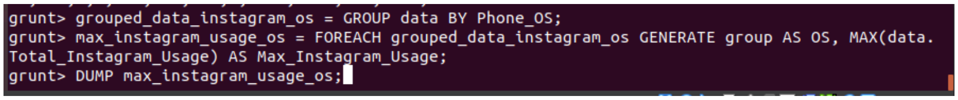


```
grunt> WA_60_100 = FOREACH data_60_100 GENERATE Total_WhatsApp_Usage;
grunt> average_WA_60_100 = FOREACH (GROUP WA_60_100 ALL) GENERATE AVG(WA_60_100.Total_WhatsApp_Usage);
grunt> DUMP average_WA_60_100;
```

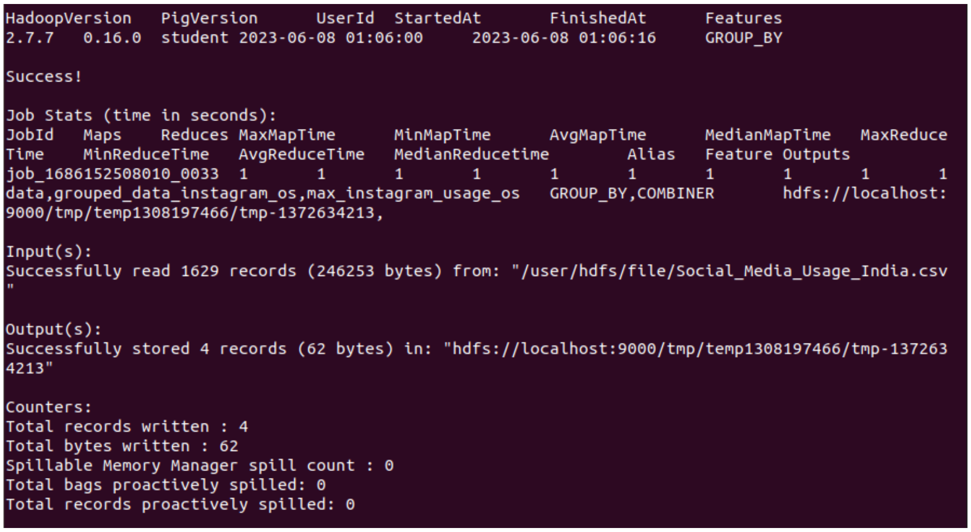
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-09 00:43:22     2023-06-09 00:44:00     GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime M
edianReducetime Alias   Feature Outputs
job_1686227197988_0033  1       1       7       7       7       7       7       7       7       7       1-454,WA_60_100,average_WA_60_1
00,data,data_60_100     GROUP_BY,COMBINER       hdfs://localhost:9000/tmp/temp1924324703/tmp1119635870,

Input(s):
Successfully read 1629 records (246258 bytes) from: "/user/hdfs/groupfile/Social_Media_Usage_India.csv"

Output(s):
Successfully stored 1 records (13 bytes) in: "hdfs://localhost:9000/tmp/temp1924324703/tmp1119635870"
2023-06-09 00:44:00,602 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(504.1111111111111)
grunt> 
```

Figure Pig.Query3.1

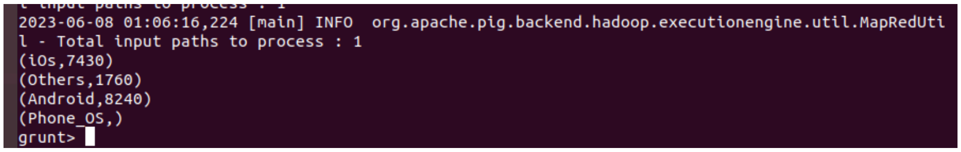
Query 3 for Pig highest Instagram usage for operating systems



```
grunt> grouped_data_instagram_os = GROUP data BY Phone_OS;
grunt> max_instagram_usage_os = FOREACH grouped_data_instagram_os GENERATE group AS OS, MAX(data.
Total_Instagram_Usage) AS Max_Instagram_Usage;
grunt> DUMP max_instagram_usage_os;
```



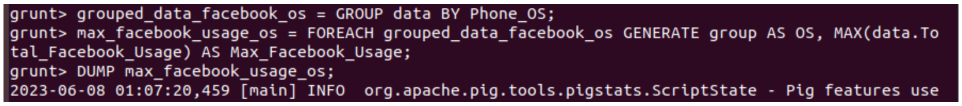
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:06:00     2023-06-08 01:06:16     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0033  1       1       1       1       1       1       1       1       1       1
data,grouped_data_instagram_os,max_instagram_usage_os   GROUP_BY,COMBINER       hdfs://localhost:
9000/tmp/temp1308197466/tmp-1372634213,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (62 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-137263
4213"

Counters:
Total records written : 4
Total bytes written : 62
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



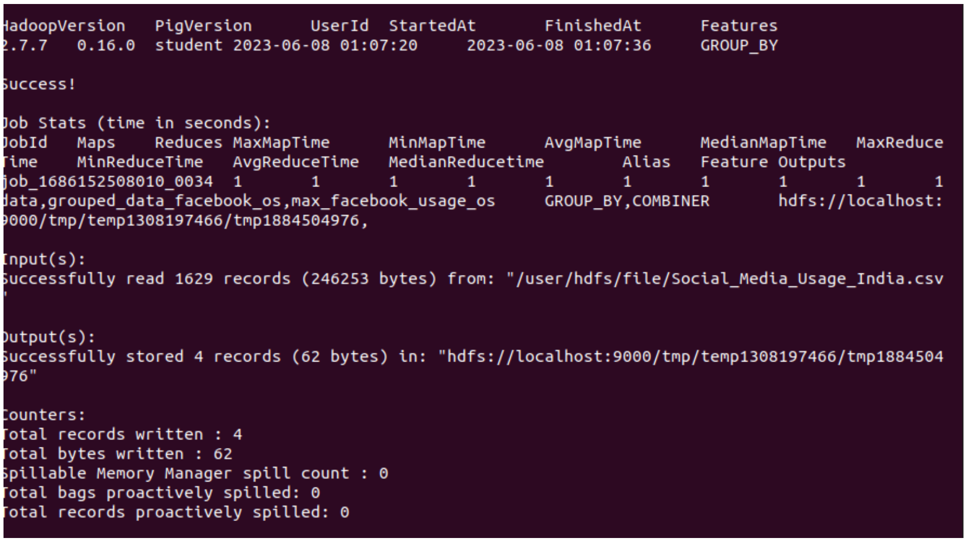
```
2023-06-08 01:06:16,224 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(iOs,7430)
(Others,1760)
(Android,8240)
(Phone_OS,)
grunt> 
```

Figure Pig.Query3.2

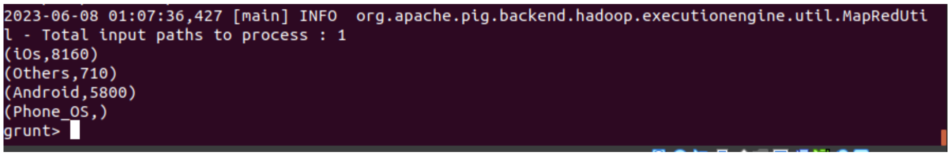
Query 3 for Pig highest Facebook usage for operating systems



```
grunt> grouped_data_facebook_os = GROUP data BY Phone_OS;
grunt> max_facebook_usage_os = FOREACH grouped_data_facebook_os GENERATE group AS OS, MAX(data.To
tal_Facebook_Usage) AS Max_Facebook_Usage;
grunt> DUMP max_facebook_usage_os;
2023-06-08 01:07:20,459 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
```



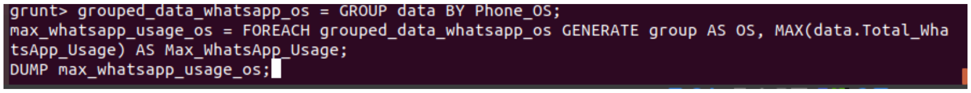
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:07:20     2023-06-08 01:07:36     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0034  1       1       1       1       1       1       1       1       1       1
data,grouped_data_facebook_os,max_facebook_usage_os     GROUP_BY,COMBINER       hdfs://localhost:
9000/tmp/temp1308197466/tmp1884504976,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (62 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp1884504
976"

Counters:
Total records written : 4
Total bytes written : 62
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



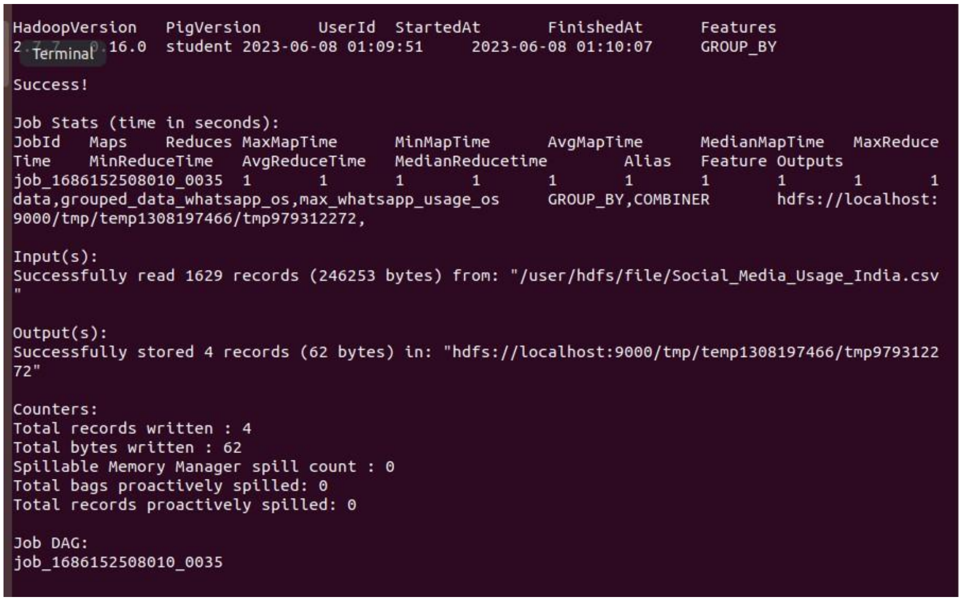
```
2023-06-08 01:07:36,427 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(iOs,8160)
(Others,710)
(Android,5800)
(Phone_OS,)
grunt> 
```

Figure Pig.Query3.3

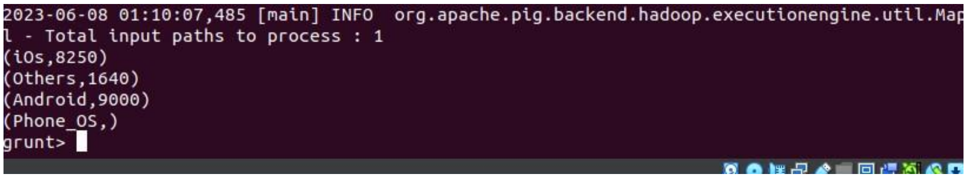
Query 3 for Pig highest WhatsApp usage for operating systems



```
grunt> grouped_data_whatsapp_os = GROUP data BY Phone_OS;
max_whatsapp_usage_os = FOREACH grouped_data_whatsapp_os GENERATE group AS OS, MAX(data.Total_Wha
tsApp_Usage) AS Max_WhatsApp_Usage;
DUMP max_whatsapp_usage_os;
```



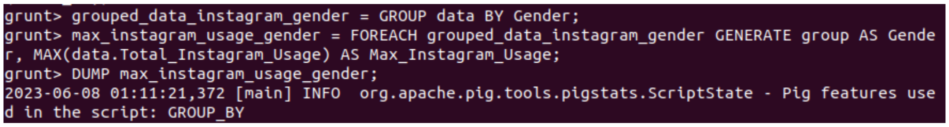
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:09:51     2023-06-08 01:10:07     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0035  1       1       1       1       1       1       1       1       1
data,grouped_data_whatsapp_os,max_whatsapp_usage_os     GROUP_BY,COMBINER       hdfs://localhost:
9000/tmp/temp1308197466/tmp979312272,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (62 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp9793122
72"

Counters:
Total records written : 4
Total bytes written : 62
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1686152508010_0035
```



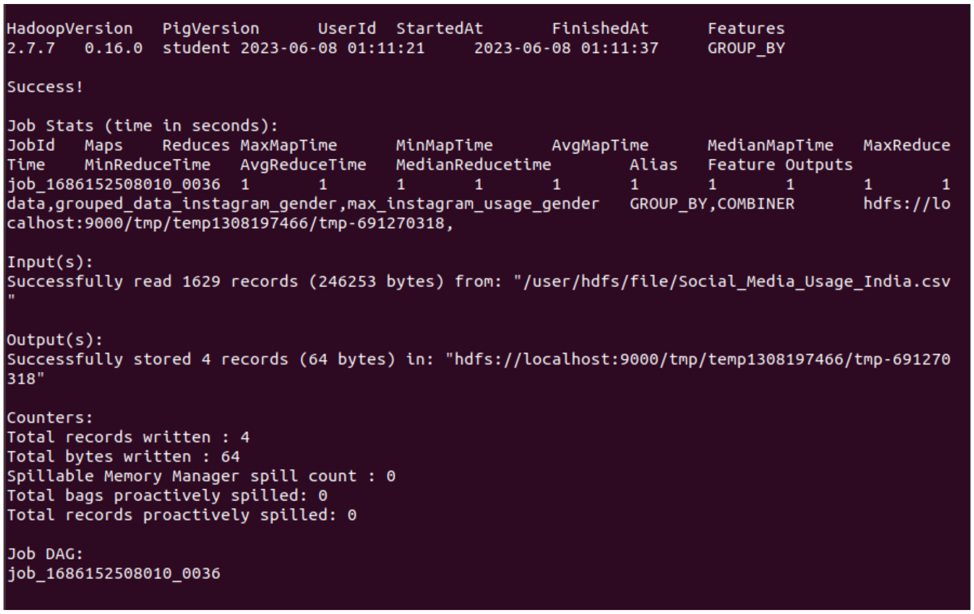
```
2023-06-08 01:10:07,485 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.Map
l - Total input paths to process : 1
(iOs,8250)
(Others,1640)
(Android,9000)
(Phone_OS,)
grunt>
```

Figure Pig.Query4.1

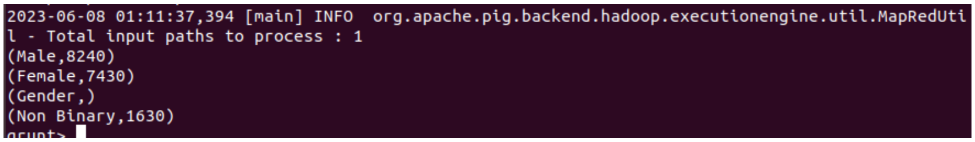
Query 4 for Pig highest Instagram usage for gender



```
grunt> grouped_data_instagram_gender = GROUP data BY Gender;
grunt> max_instagram_usage_gender = FOREACH grouped_data_instagram_gender GENERATE group AS Gende
r, MAX(data.Total_Instagram_Usage) AS Max_Instagram_Usage;
grunt> DUMP max_instagram_usage_gender;
2023-06-08 01:11:21,372 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
```



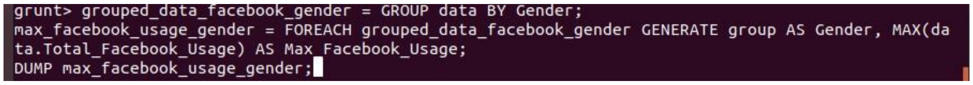
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:11:21     2023-06-08 01:11:37     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0036  1       1       1       1       1       1       1       1       1       1
data,grouped_data_instagram_gender,max_instagram_usage_gender   GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp-691270318,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (64 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-691270
318"

Counters:
Total records written : 4
Total bytes written : 64
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1686152508010_0036
```



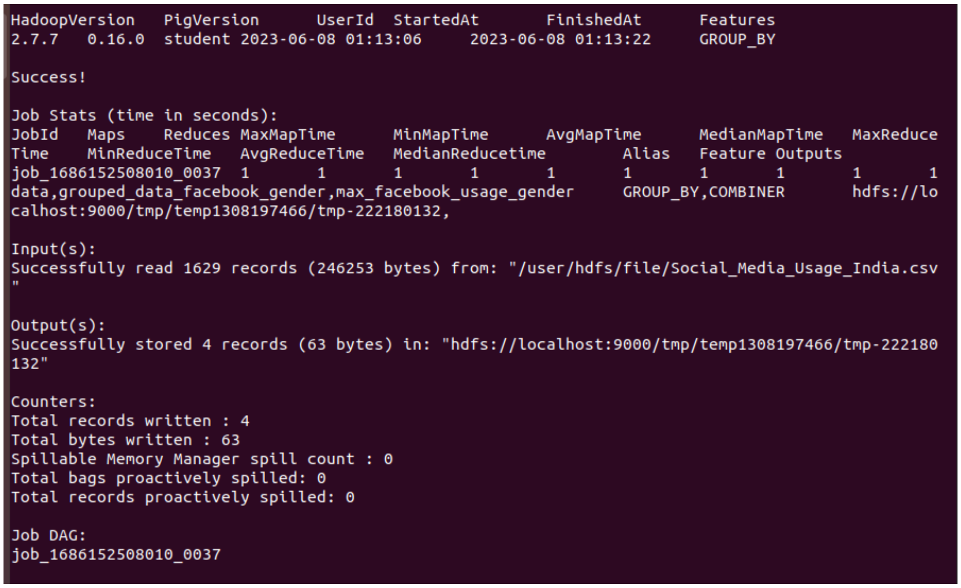
```
2023-06-08 01:11:37,394 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Male,8240)
(Female,7430)
(Gender,)
(Non Binary,1630)
grunt> 
```

Figure Pig.Query4.2

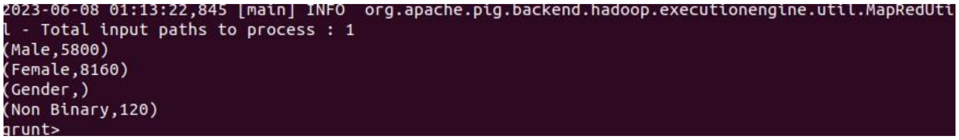
Query 4 for Pig highest Facebook usage for gender



```
grunt> grouped_data_facebook_gender = GROUP data BY Gender;
max_facebook_usage_gender = FOREACH grouped_data_facebook_gender GENERATE group AS Gender, MAX(da
ta.Total_Facebook_Usage) AS Max_Facebook_Usage;
DUMP max_facebook_usage_gender;
```



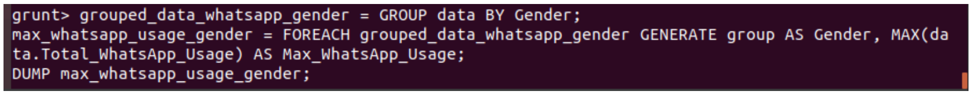
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:13:06     2023-06-08 01:13:22     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0037  1       1       1       1       1       1       1       1       1       1
data,grouped_data_facebook_gender,max_facebook_usage_gender     GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp-222180132,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (63 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-222180
132"

Counters:
Total records written : 4
Total bytes written : 63
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1686152508010_0037
```



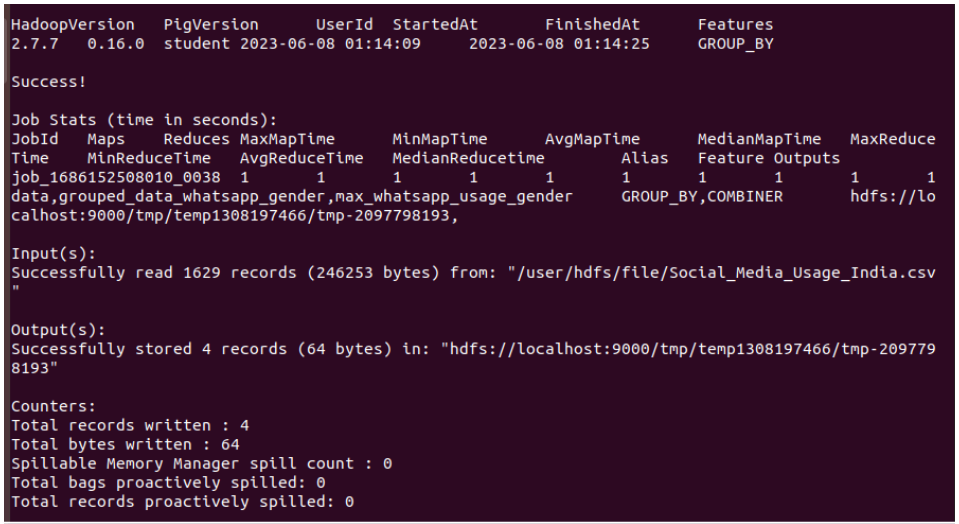
```
2023-06-08 01:13:22,845 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Male,5800)
(Female,8160)
(Gender,)
(Non Binary,120)
grunt>
```

Figure Pig.Query4.3

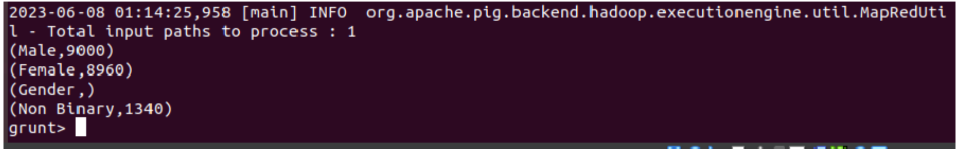
Query 4 for Pig highest WhatsApp usage for gender



```
grunt> grouped_data_whatsapp_gender = GROUP data BY Gender;
max_whatsapp_usage_gender = FOREACH grouped_data_whatsapp_gender GENERATE group AS Gender, MAX(da
ta.Total_WhatsApp_Usage) AS Max_WhatsApp_Usage;
DUMP max_whatsapp_usage_gender;
```



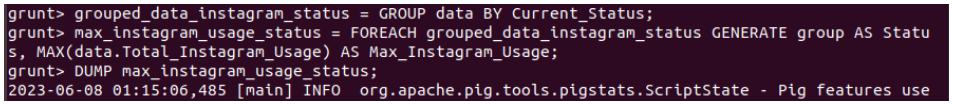
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:14:09     2023-06-08 01:14:25     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0038  1       1       1       1       1       1       1       1       1       1
data,grouped_data_whatsapp_gender,max_whatsapp_usage_gender     GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp-2097798193,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (64 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-209779
8193"

Counters:
Total records written : 4
Total bytes written : 64
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



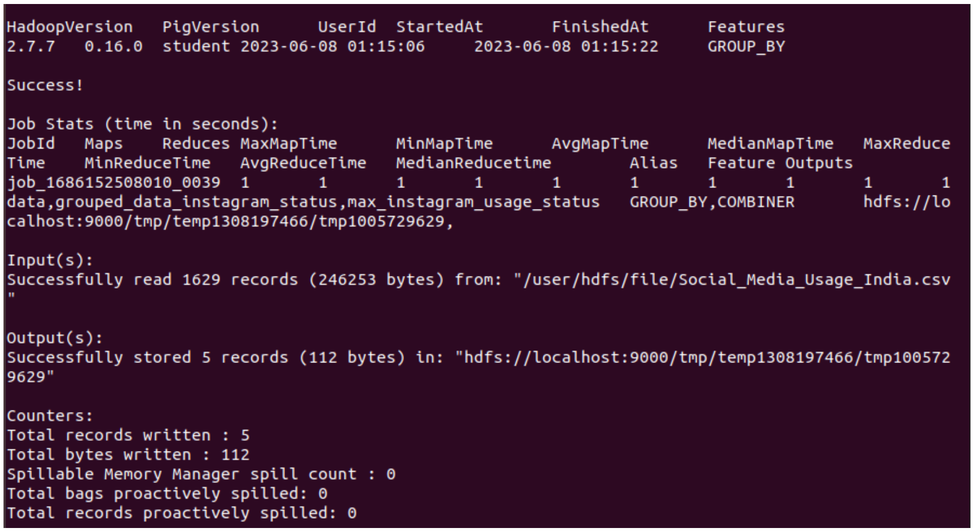
```
2023-06-08 01:14:25,958 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Male,9000)
(Female,8960)
(Gender,)
(Non Binary,1340)
grunt>
```

Figure Pig.Query5.1

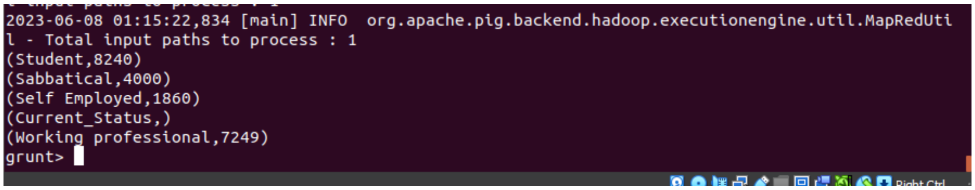
Query 5 for Pig highest Instagram usage for status



```
grunt> grouped_data_instagram_status = GROUP data BY Current_Status;
grunt> max_instagram_usage_status = FOREACH grouped_data_instagram_status GENERATE group AS Statu
s, MAX(data.Total_Instagram_Usage) AS Max_Instagram_Usage;
grunt> DUMP max_instagram_usage_status;
2023-06-08 01:15:06,485 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
```



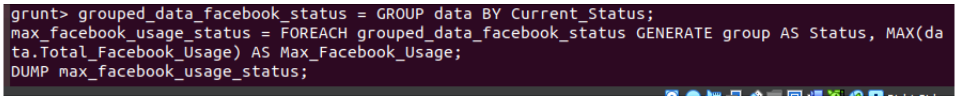
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:15:06     2023-06-08 01:15:22     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0039  1       1       1       1       1       1       1       1       1       1
data,grouped_data_instagram_status,max_instagram_usage_status   GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp1005729629,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 5 records (112 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp100572
9629"

Counters:
Total records written : 5
Total bytes written : 112
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



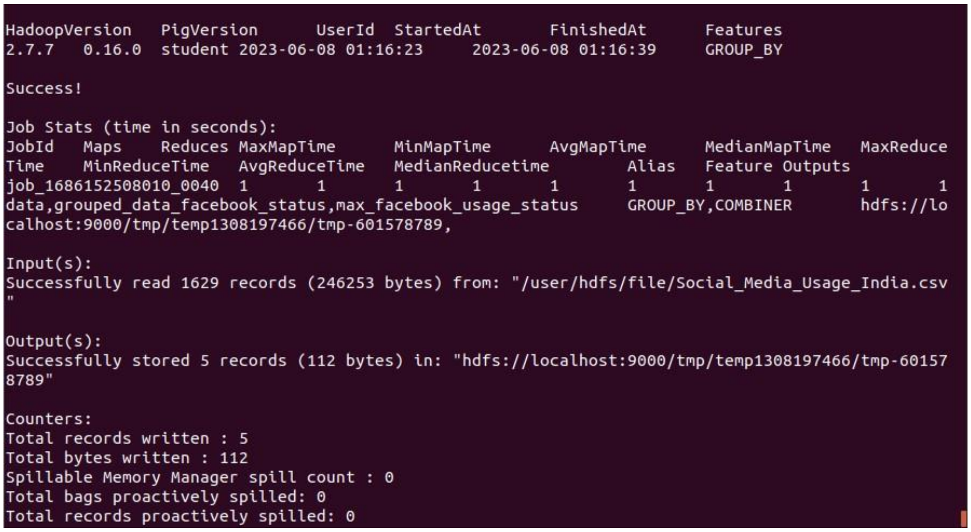
```
2023-06-08 01:15:22,834 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Student,8240)
(Sabbatical,4000)
(Self Employed,1860)
(Current_Status,)
(Working professional,7249)
grunt>
```

Figure Pig.Query5.2

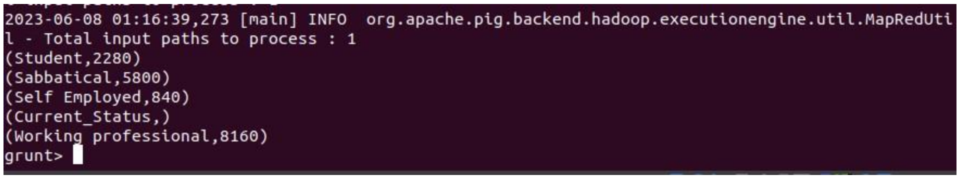
Query 5 for Pig highest Facebook usage for status



```
grunt> grouped_data_facebook_status = GROUP data BY Current_Status;
max_facebook_usage_status = FOREACH grouped_data_facebook_status GENERATE group AS Status, MAX(da
ta.Total_Facebook_Usage) AS Max_Facebook_Usage;
DUMP max_facebook_usage_status;
```



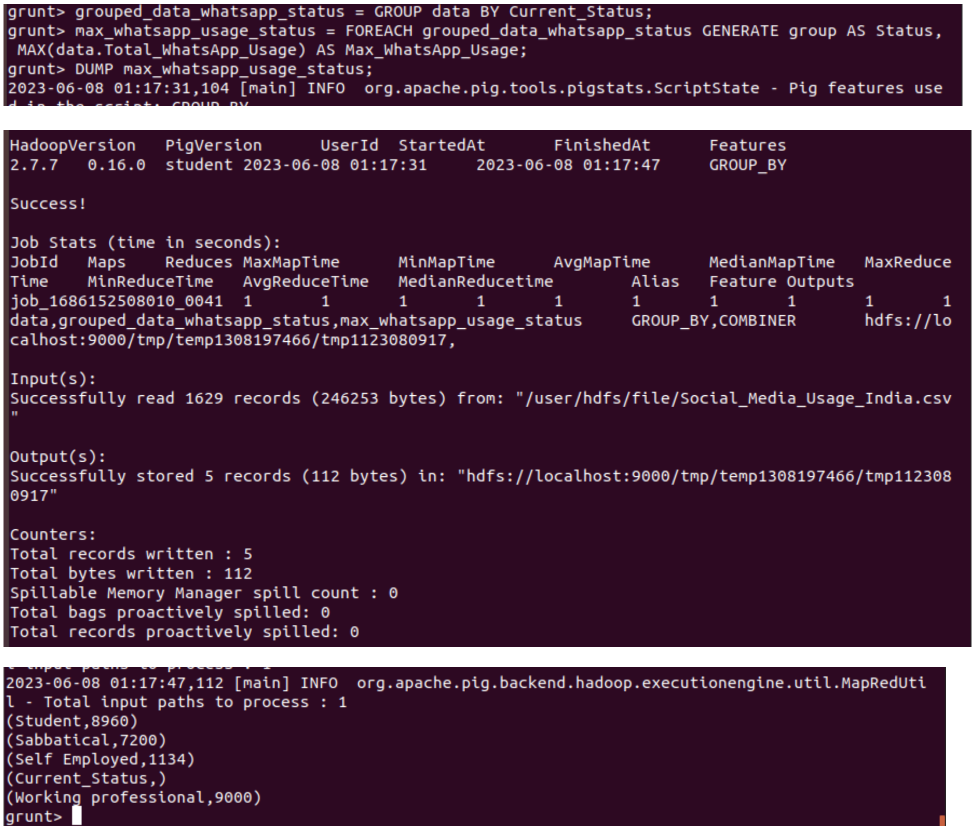
```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:16:23     2023-06-08 01:16:39     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0040  1       1       1       1       1       1       1       1       1       1
data,grouped_data_facebook_status,max_facebook_usage_status     GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp-601578789,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 5 records (112 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-60157
8789"

Counters:
Total records written : 5
Total bytes written : 112
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



```
2023-06-08 01:16:39,273 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Student,2280)
(Sabbatical,5800)
(Self Employed,840)
(Current_Status,)
(Working professional,8160)
grunt>
```

Figure Pig.Query5.3

Query 5 for Pig highest WhatsApp usage for status



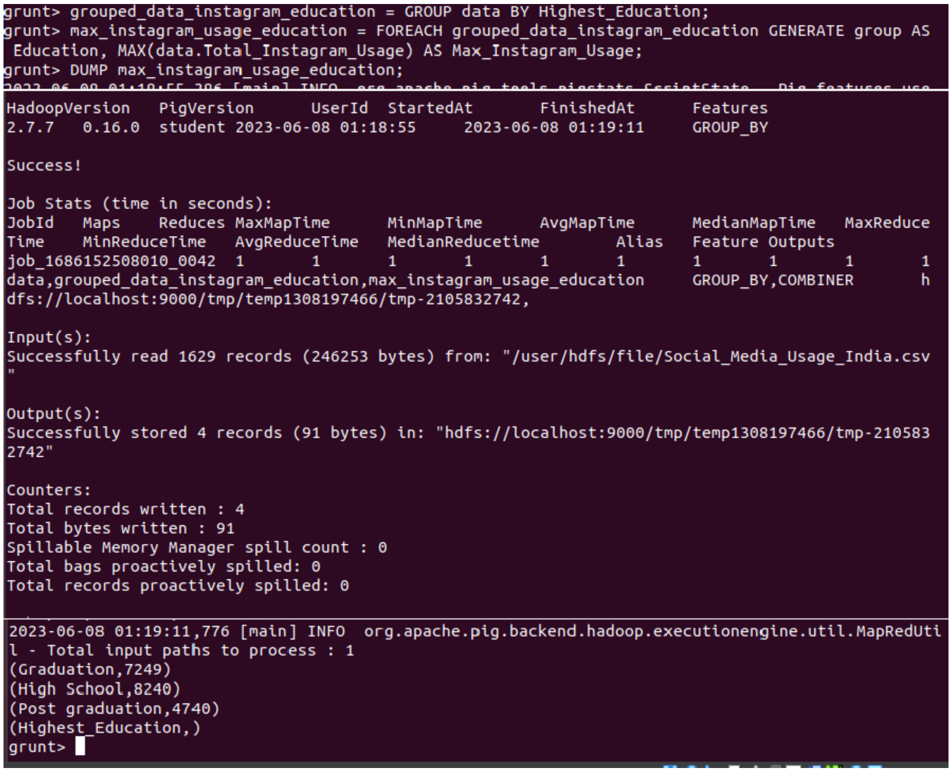
```
grunt> grouped_data_whatsapp_status = GROUP data BY Current_Status;
grunt> max_whatsapp_usage_status = FOREACH grouped_data_whatsapp_status GENERATE group AS Status,
 MAX(data.Total_WhatsApp_Usage) AS Max_WhatsApp_Usage;
grunt> DUMP max_whatsapp_usage_status;
2023-06-08 01:17:31,104 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:17:31     2023-06-08 01:17:47     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0041  1       1       1       1       1       1       1       1       1       1
data,grouped_data_whatsapp_status,max_whatsapp_usage_status     GROUP_BY,COMBINER       hdfs://lo
calhost:9000/tmp/temp1308197466/tmp1123080917,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 5 records (112 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp112308
0917"

Counters:
Total records written : 5
Total bytes written : 112
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

```
l - Total input paths to process : 1
2023-06-08 01:17:47,112 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Student,8960)
(Sabbatical,7200)
(Self Employed,1134)
(Current_Status,)
(Working professional,9000)
grunt> 
```

Figure Pig.Query6.1

Query 6 for Pig highest Instagram usage for education



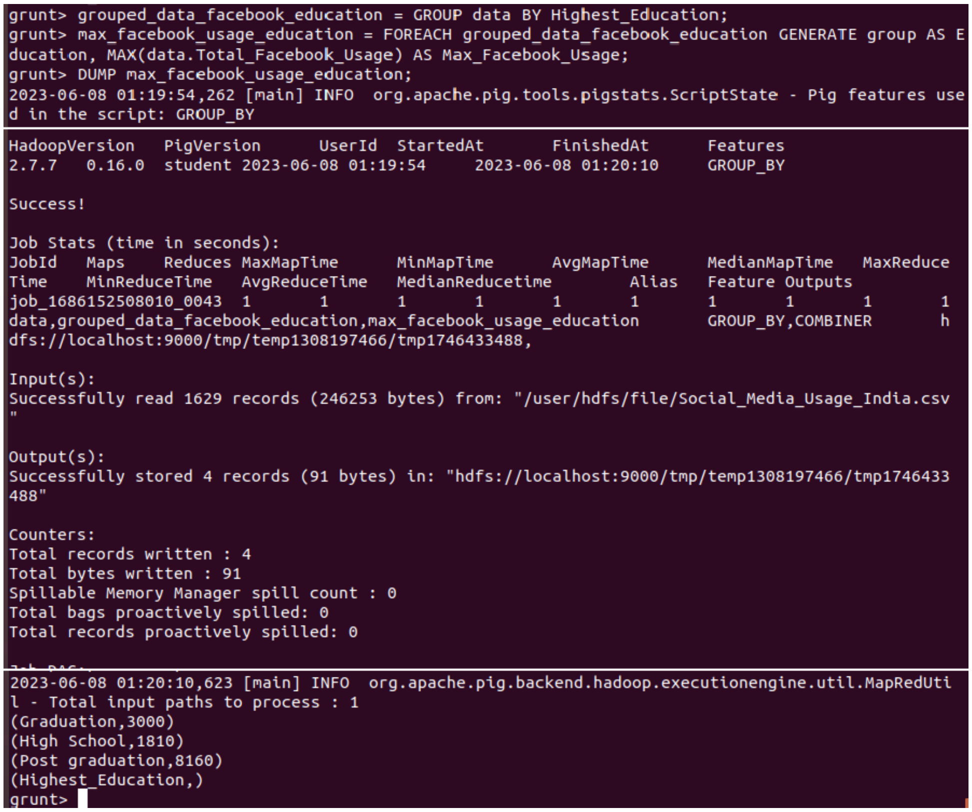
```
grunt> grouped_data_instagram_education = GROUP data BY Highest_Education;
grunt> max_instagram_usage_education = FOREACH grouped_data_instagram_education GENERATE group AS
 Education, MAX(data.Total_Instagram_Usage) AS Max_Instagram_Usage;
grunt> DUMP max_instagram_usage_education;
2023-06-08 01:18:55,386 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:18:55     2023-06-08 01:19:11     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0042  1       1       1       1       1       1       1       1       1       1
data,grouped_data_instagram_education,max_instagram_usage_education     GROUP_BY,COMBINER       h
dfs://localhost:9000/tmp/temp1308197466/tmp-2105832742,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (91 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp-210583
2742"

Counters:
Total records written : 4
Total bytes written : 91
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

```
2023-06-08 01:19:11,776 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Graduation,7249)
(High School,8240)
(Post graduation,4740)
(Highest_Education,)
grunt>
```

Figure Pig.Query6.2

Query 6 for Pig highest Facebook usage for education



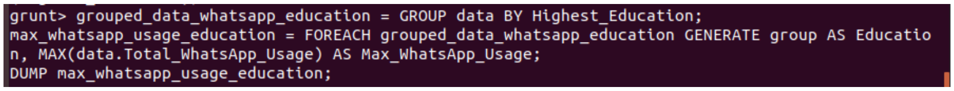
```
grunt> grouped_data_facebook_education = GROUP data BY Highest_Education;
grunt> max_facebook_usage_education = FOREACH grouped_data_facebook_education GENERATE group AS E
ducation, MAX(data.Total_Facebook_Usage) AS Max_Facebook_Usage;
grunt> DUMP max_facebook_usage_education;
2023-06-08 01:19:54,262 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
```

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:19:54     2023-06-08 01:20:10     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0043  1       1       1       1       1       1       1       1       1       1
data,grouped_data_facebook_education,max_facebook_usage_education       GROUP_BY,COMBINER       h
dfs://localhost:9000/tmp/temp1308197466/tmp1746433488,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (91 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp1746433
488"

Counters:
Total records written : 4
Total bytes written : 91
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

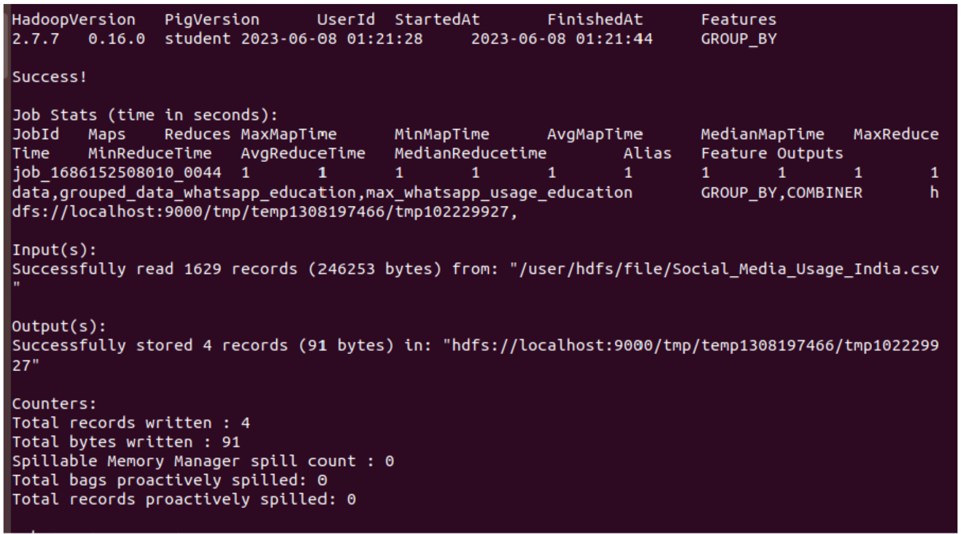
```
2023-06-08 01:20:10,623 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Graduation,3000)
(High School,1810)
(Post graduation,8160)
(Highest_Education,)
grunt> 
```

Figure Pig.Query6.3

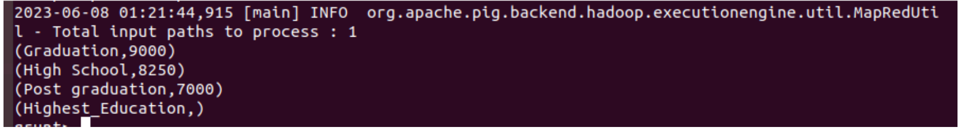
Query 6 for Pig highest WhatsApp usage for education



```
grunt> grouped_data_whatsapp_education = GROUP data BY Highest_Education;
max_whatsapp_usage_education = FOREACH grouped_data_whatsapp_education GENERATE group AS Educatio
n, MAX(data.Total_WhatsApp_Usage) AS Max_WhatsApp_Usage;
DUMP max_whatsapp_usage_education;
```



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.7   0.16.0  student 2023-06-08 01:21:28     2023-06-08 01:21:44     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduce
Time    MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1686152508010_0044  1       1       1       1       1       1       1       1       1       1
data,grouped_data_whatsapp_education,max_whatsapp_usage_education       GROUP_BY,COMBINER       h
dfs://localhost:9000/tmp/temp1308197466/tmp102229927,

Input(s):
Successfully read 1629 records (246253 bytes) from: "/user/hdfs/file/Social_Media_Usage_India.csv
"

Output(s):
Successfully stored 4 records (91 bytes) in: "hdfs://localhost:9000/tmp/temp1308197466/tmp1022299
27"

Counters:
Total records written : 4
Total bytes written : 91
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```



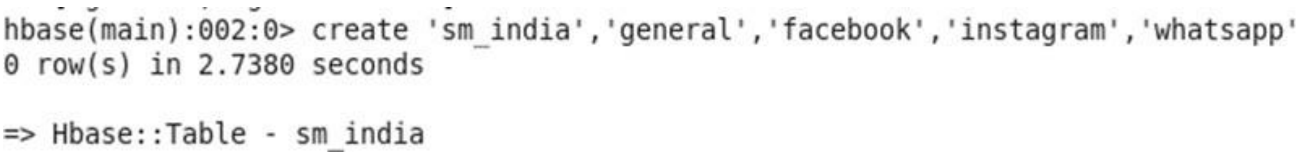
```
2023-06-08 01:21:44,915 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
l - Total input paths to process : 1
(Graduation,9000)
(High School,8250)
(Post graduation,7000)
(Highest_Education,)
grunt>
```

Appendix D

HBase+Impala

Figure HBase.Import/Scan.1

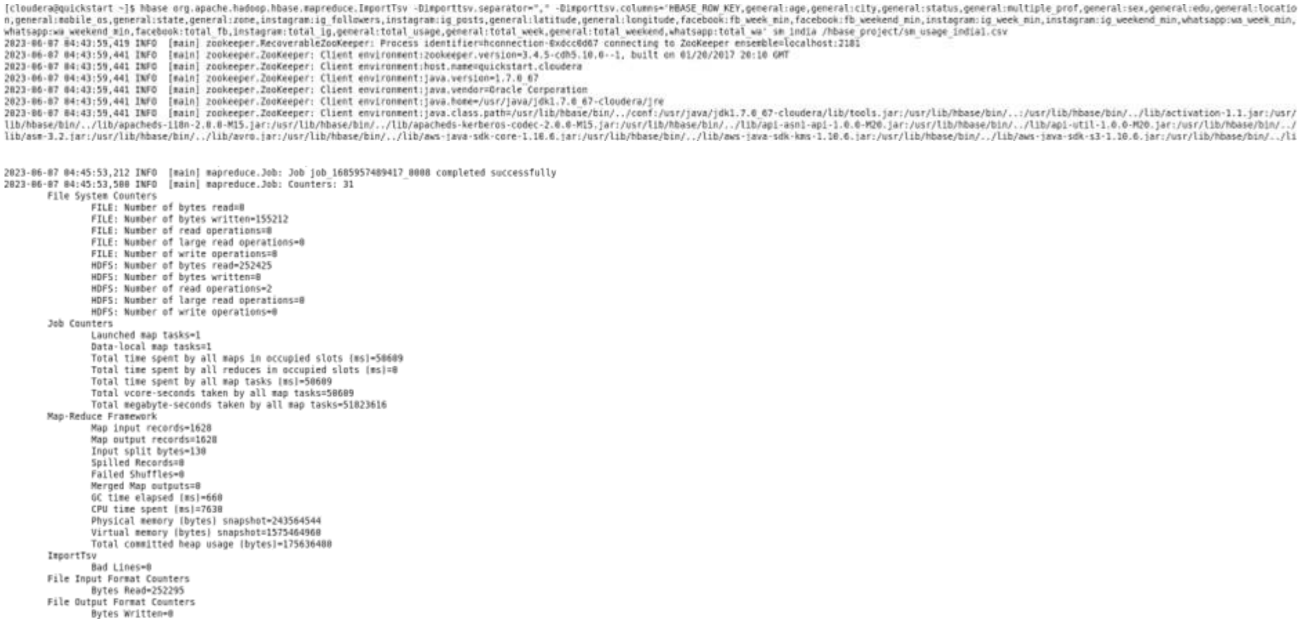
HBase import and scan to create table in HBase



```
hbase(main):002:0> create 'sm_india','general','facebook','instagram','whatsapp'
0 row(s) in 2.7380 seconds

=> Hbase::Table - sm_india
```

Figure HBase.Import/Scan.2

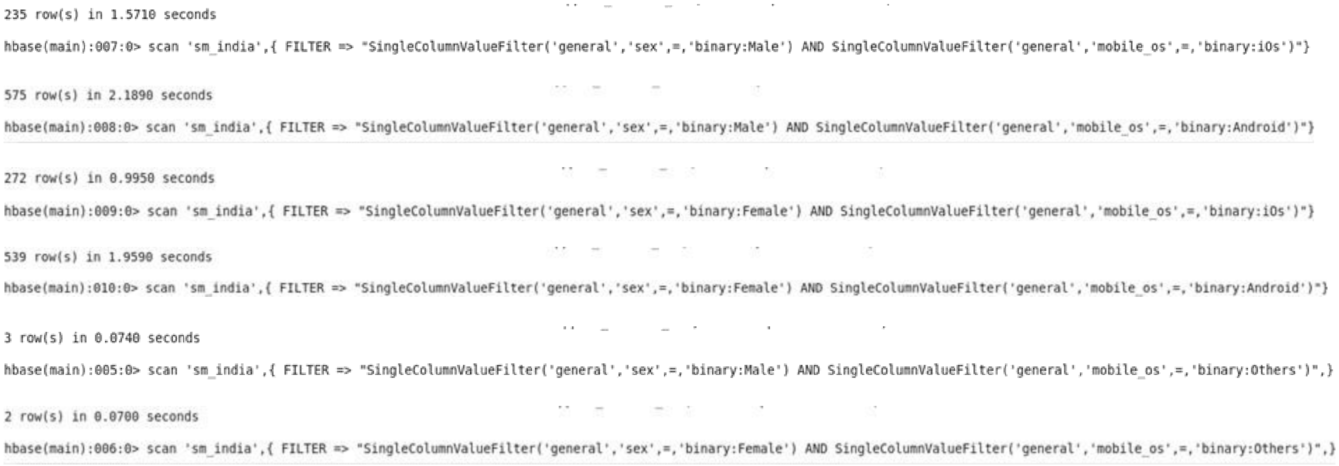
Hbase Import data from HDFS into HBase



```
[cloudera@quickstart ~]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator="," -Dimporttsv.columns='HBASE_ROW_KEY,general:age,general:city,general:status,general:multiple_prof,general:sex,general:edu,general:locatio
n,general:mobile_os,general:state,general:zone,instagram:ig_followers,instagram:ig_posts,general:latitude,general:longitude,facebook:fb_week_min,facebook:fb_weekend_min,instagram:ig_week_min,instagram:ig_weekend_min,whatsapp:wa_week_min,
whatsapp:wa_weekend_min,facebook:total_fb,instagram:total_ig,general:total_usage,general:total_week,general:total_weekend,whatsapp:total_wa' sm_india /hbase_project/sm_usage_india1.csv
2023-06-07 04:43:59,419 INFO  [main] zookeeper.RecoverableZooKeeper: Process identifier=hconnection-0xdcc0d67 connecting to ZooKeeper ensemble=localhost:2181
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:zookeeper.version=3.4.5-cdh5.10.0--1, built on 01/20/2017 20:10 GMT
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:host.name=quickstart.cloudera
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:java.version=1.7.0_67
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:java.vendor=Oracle Corporation
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:java.home=/usr/java/jdk1.7.0_67-cloudera/jre
2023-06-07 04:43:59,441 INFO  [main] zookeeper.ZooKeeper: Client environment:java.class.path=/usr/lib/hbase/bin/../conf:/usr/java/jdk1.7.0_67-cloudera/lib/tools.jar:/usr/lib/hbase/bin/..:/usr/lib/hbase/bin/../lib/activation-1.1.jar:/usr/
lib/hbase/bin/../lib/apacheds-i18n-2.0.0-M15.jar:/usr/lib/hbase/bin/../lib/apacheds-kerberos-codec-2.0.0-M15.jar:/usr/lib/hbase/bin/../lib/api-asn1-api-1.0.0-M20.jar:/usr/lib/hbase/bin/../lib/api-util-1.0.0-M20.jar:/usr/lib/hbase/bin/../
lib/asm-3.2.jar:/usr/lib/hbase/bin/../lib/avro.jar:/usr/lib/hbase/bin/../lib/aws-java-sdk-core-1.10.6.jar:/usr/lib/hbase/bin/../lib/aws-java-sdk-kms-1.10.6.jar:/usr/lib/hbase/bin/../lib/aws-java-sdk-s3-1.10.6.jar:/usr/lib/hbase/bin/../li


2023-06-07 04:45:53,212 INFO  [main] mapreduce.Job: Job job_1685957489417_0008 completed successfully
2023-06-07 04:45:53,500 INFO  [main] mapreduce.Job: Counters: 31
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=155212
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=252425
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=2
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=50609
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=50609
                Total vcore-seconds taken by all map tasks=50609
                Total megabyte-seconds taken by all map tasks=51823616
        Map-Reduce Framework
                Map input records=1628
                Map output records=1628
                Input split bytes=130
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=660
                CPU time spent (ms)=7630
                Physical memory (bytes) snapshot=243564544
                Virtual memory (bytes) snapshot=1575464960
                Total committed heap usage (bytes)=175636480
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=252295
        File Output Format Counters
                Bytes Written=0
```

Figure Hbase.Query1.1

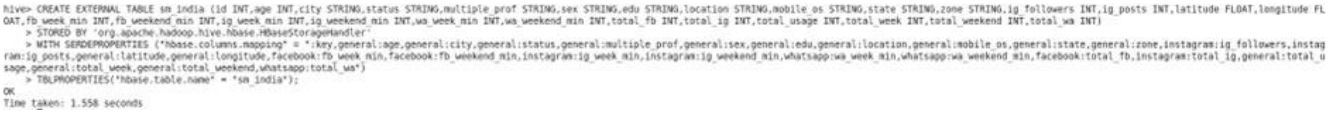
Query 1 for HBase different types of operating systems used based on gender



```
235 row(s) in 1.5710 seconds

hbase(main):007:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Male') AND SingleColumnValueFilter('general','mobile_os',=,'binary:iOs')"}

575 row(s) in 2.1890 seconds

hbase(main):008:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Male') AND SingleColumnValueFilter('general','mobile_os',=,'binary:Android')"}

272 row(s) in 0.9950 seconds

hbase(main):009:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Female') AND SingleColumnValueFilter('general','mobile_os',=,'binary:iOs')"}

539 row(s) in 1.9590 seconds

hbase(main):010:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Female') AND SingleColumnValueFilter('general','mobile_os',=,'binary:Android')"}

3 row(s) in 0.0740 seconds

hbase(main):005:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Male') AND SingleColumnValueFilter('general','mobile_os',=,'binary:Others')",}

2 row(s) in 0.0700 seconds

hbase(main):006:0> scan 'sm_india',{ FILTER => "SingleColumnValueFilter('general','sex',=,'binary:Female') AND SingleColumnValueFilter('general','mobile_os',=,'binary:Others')",}
```

Figure Hbase.Query2.1

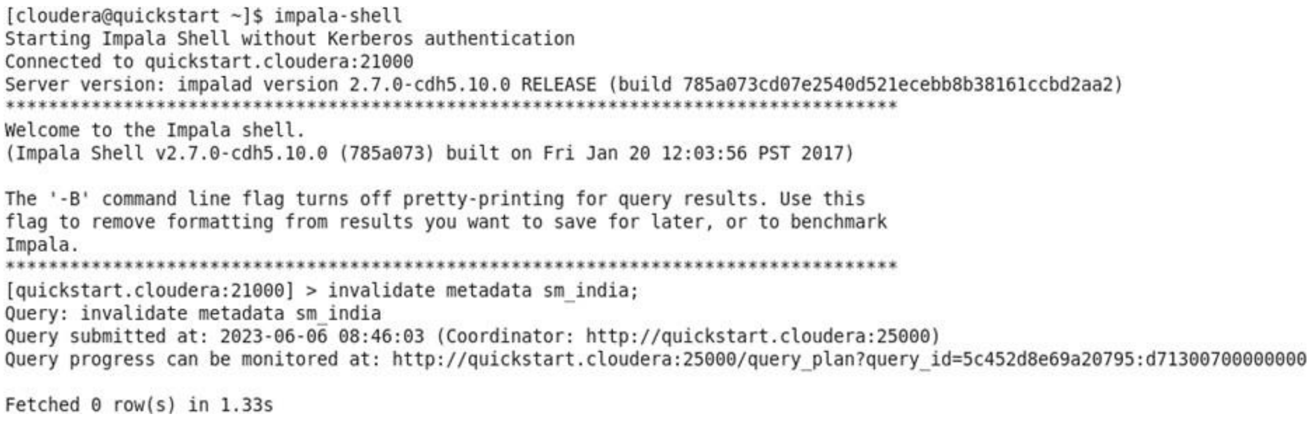
Query 2 for HBase create external table in Hive to link HBase table into Impala



```
hive> CREATE EXTERNAL TABLE sm_india (id INT,age INT,city STRING,status STRING,multiple_prof STRING,sex STRING,edu STRING,location STRING,mobile_os STRING,state STRING,zone STRING,ig_followers INT,ig_posts INT,latitude FLOAT,longitude FL
OAT,fb_week_min INT,fb_weekend_min INT,ig_week_min INT,ig_weekend_min INT,wa_week_min INT,wa_weekend_min INT,total_fb INT,total_ig INT,total_usage INT,total_week INT,total_weekend INT,total_wa INT)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,general:age,general:city,general:status,general:multiple_prof,general:sex,general:edu,general:location,general:mobile_os,general:state,general:zone,instagram:ig_followers,instag
ram:ig_posts,general:latitude,general:longitude,facebook:fb_week_min,facebook:fb_weekend_min,instagram:ig_week_min,instagram:ig_weekend_min,whatsapp:wa_week_min,whatsapp:wa_weekend_min,facebook:total_fb,instagram:total_ig,general:total_u
sage,general:total_week,general:total_weekend,whatsapp:total_wa")
    > TBLPROPERTIES("hbase.table.name" = "sm_india");
OK
Time taken: 1.558 seconds
```

Figure Hbase.Query2.2

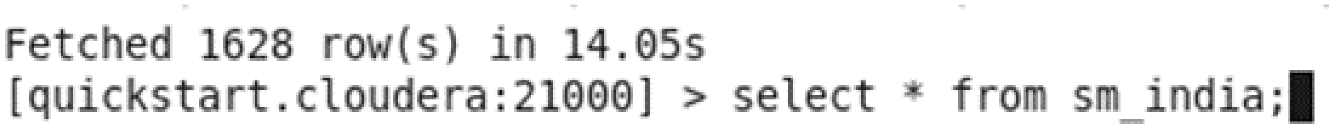
Query 2 for HBase run command to make table in Hive visible in Impala



```
[cloudera@quickstart ~]$ impala-shell
Starting Impala Shell without Kerberos authentication
Connected to quickstart.cloudera:21000
Server version: impalad version 2.7.0-cdh5.10.0 RELEASE (build 785a073cd07e2540d521ecebb8b38161ccbd2aa2)
***********************************************************************************
Welcome to the Impala shell.
(Impala Shell v2.7.0-cdh5.10.0 (785a073) built on Fri Jan 20 12:03:56 PST 2017)

The '-B' command line flag turns off pretty-printing for query results. Use this
flag to remove formatting from results you want to save for later, or to benchmark
Impala.
***********************************************************************************
[quickstart.cloudera:21000] > invalidate metadata sm_india;
Query: invalidate metadata sm_india
Query submitted at: 2023-06-06 08:46:03 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=5c452d8e69a20795:d71300700000000

Fetched 0 row(s) in 1.33s
```

Figure Hbase.Query2.3

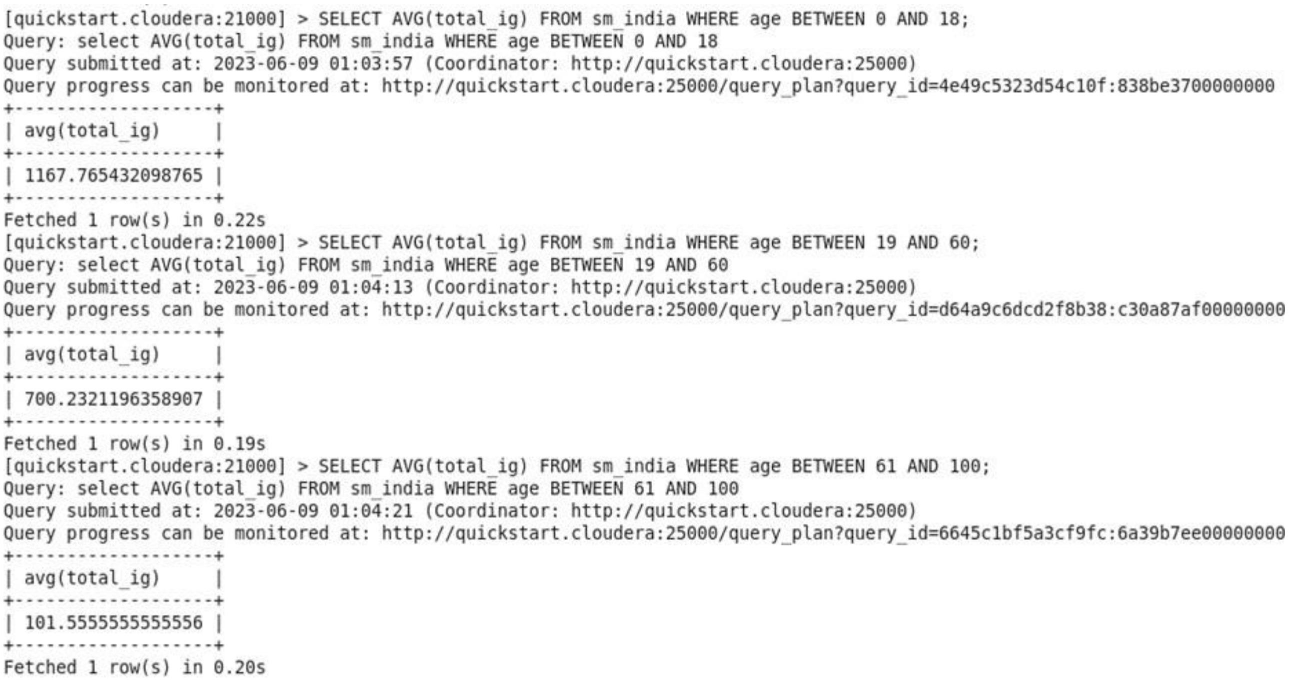
Query 2 for HBase scan all data in Impala



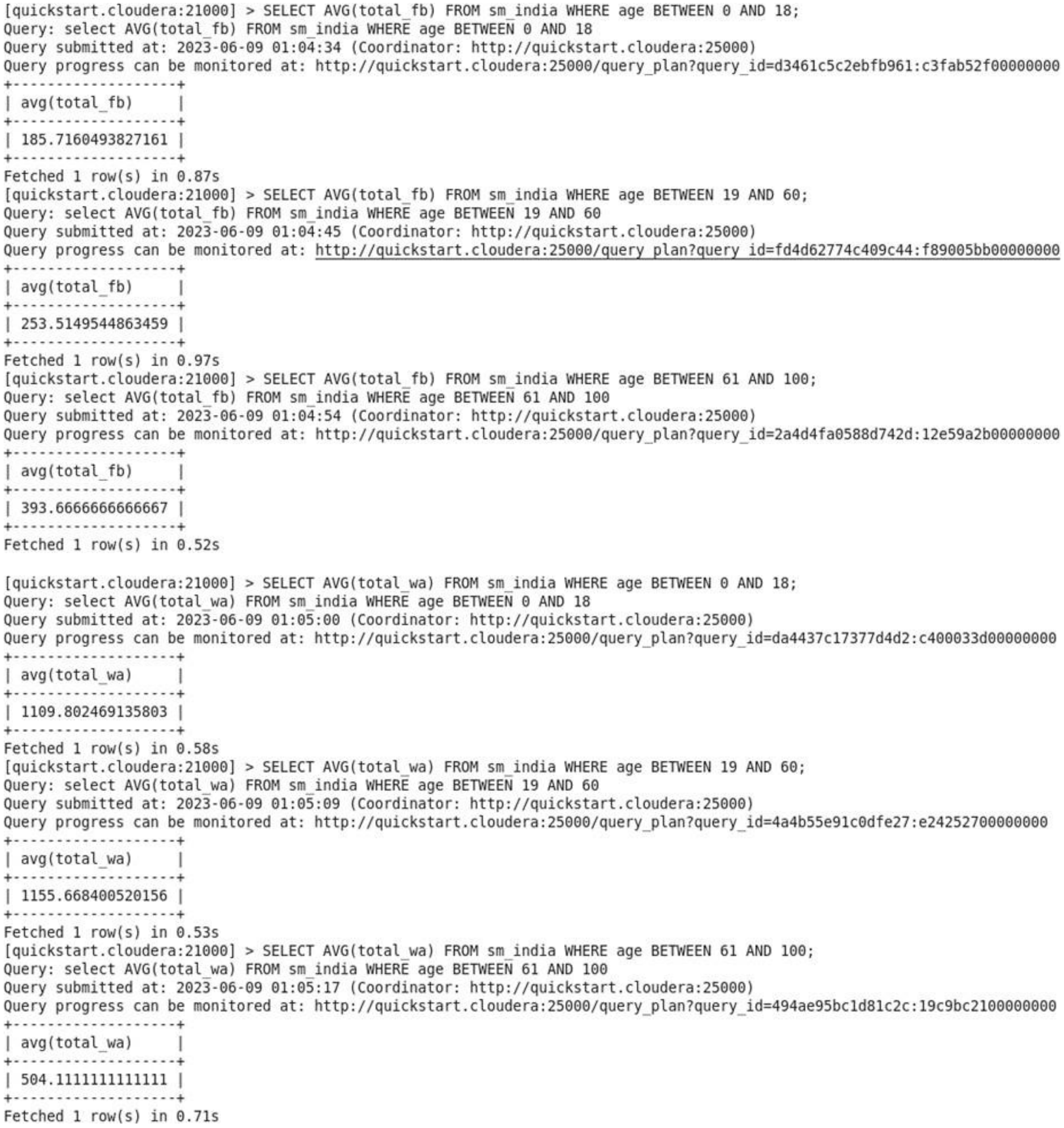
```
Fetched 1628 row(s) in 14.05s
[quickstart.cloudera:21000] > select * from sm_india;
```

Figure Hbase.Query2.4

Query 2 for HBase total Facebook, Instagram and WhatsApp average usage based on age category



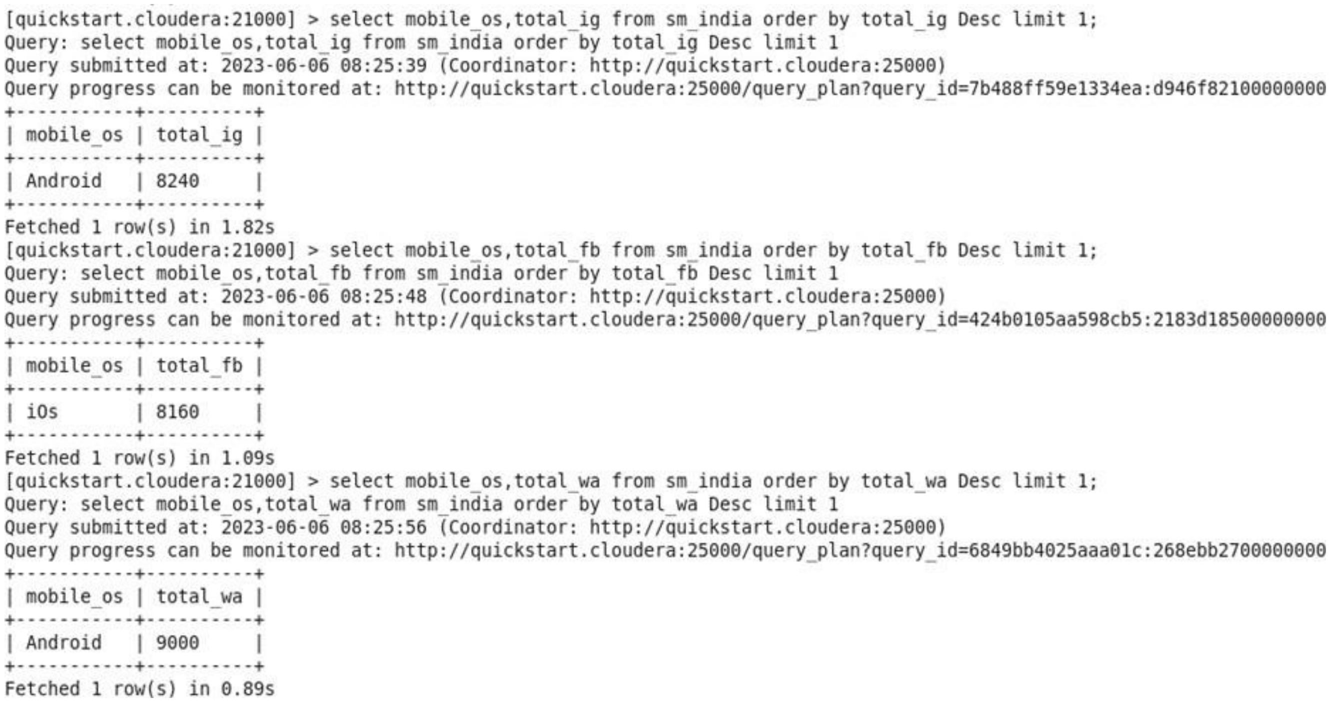
```
[quickstart.cloudera:21000] > SELECT AVG(total_ig) FROM sm_india WHERE age BETWEEN 0 AND 18;
Query: select AVG(total_ig) FROM sm_india WHERE age BETWEEN 0 AND 18
Query submitted at: 2023-06-09 01:03:57 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=4e49c5323d54c10f:838be3700000000
+-------------------+
| avg(total_ig)     |
+-------------------+
| 1167.765432098765 |
+-------------------+
Fetched 1 row(s) in 0.22s
[quickstart.cloudera:21000] > SELECT AVG(total_ig) FROM sm_india WHERE age BETWEEN 19 AND 60;
Query: select AVG(total_ig) FROM sm_india WHERE age BETWEEN 19 AND 60
Query submitted at: 2023-06-09 01:04:13 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=d64a9c6dcd2f8b38:c30a87af00000000
+-------------------+
| avg(total_ig)     |
+-------------------+
| 700.2321196358907 |
+-------------------+
Fetched 1 row(s) in 0.19s
[quickstart.cloudera:21000] > SELECT AVG(total_ig) FROM sm_india WHERE age BETWEEN 61 AND 100;
Query: select AVG(total_ig) FROM sm_india WHERE age BETWEEN 61 AND 100
Query submitted at: 2023-06-09 01:04:21 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=6645c1bf5a3cf9fc:6a39b7ee00000000
+-------------------+
| avg(total_ig)     |
+-------------------+
| 101.5555555555556 |
+-------------------+
Fetched 1 row(s) in 0.20s
```



```
[quickstart.cloudera:21000] > SELECT AVG(total_fb) FROM sm_india WHERE age BETWEEN 0 AND 18;
Query: select AVG(total_fb) FROM sm_india WHERE age BETWEEN 0 AND 18
Query submitted at: 2023-06-09 01:04:34 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=d3461c5c2ebfb961:c3fab52f00000000
+-------------------+
| avg(total_fb)     |
+-------------------+
| 185.7160493827161 |
+-------------------+
Fetched 1 row(s) in 0.87s
[quickstart.cloudera:21000] > SELECT AVG(total_fb) FROM sm_india WHERE age BETWEEN 19 AND 60;
Query: select AVG(total_fb) FROM sm_india WHERE age BETWEEN 19 AND 60
Query submitted at: 2023-06-09 01:04:45 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=fd4d62774c409c44:f89005bb00000000
+-------------------+
| avg(total_fb)     |
+-------------------+
| 253.5149544863459 |
+-------------------+
Fetched 1 row(s) in 0.97s
[quickstart.cloudera:21000] > SELECT AVG(total_fb) FROM sm_india WHERE age BETWEEN 61 AND 100;
Query: select AVG(total_fb) FROM sm_india WHERE age BETWEEN 61 AND 100
Query submitted at: 2023-06-09 01:04:54 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=2a4d4fa0588d742d:12e59a2b00000000
+-------------------+
| avg(total_fb)     |
+-------------------+
| 393.6666666666667 |
+-------------------+
Fetched 1 row(s) in 0.52s

[quickstart.cloudera:21000] > SELECT AVG(total_wa) FROM sm_india WHERE age BETWEEN 0 AND 18;
Query: select AVG(total_wa) FROM sm_india WHERE age BETWEEN 0 AND 18
Query submitted at: 2023-06-09 01:05:00 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=da4437c17377d4d2:c400033d00000000
+-------------------+
| avg(total_wa)     |
+-------------------+
| 1109.802469135803 |
+-------------------+
Fetched 1 row(s) in 0.58s
[quickstart.cloudera:21000] > SELECT AVG(total_wa) FROM sm_india WHERE age BETWEEN 19 AND 60;
Query: select AVG(total_wa) FROM sm_india WHERE age BETWEEN 19 AND 60
Query submitted at: 2023-06-09 01:05:09 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=4a4b55e91c0dfe27:e242527000000000
+-------------------+
| avg(total_wa)     |
+-------------------+
| 1155.668400520156 |
+-------------------+
Fetched 1 row(s) in 0.53s
[quickstart.cloudera:21000] > SELECT AVG(total_wa) FROM sm_india WHERE age BETWEEN 61 AND 100;
Query: select AVG(total_wa) FROM sm_india WHERE age BETWEEN 61 AND 100
Query submitted at: 2023-06-09 01:05:17 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=494ae95bc1d81c2c:19c9bc2100000000
+-------------------+
| avg(total_wa)     |
+-------------------+
| 504.1111111111111 |
+-------------------+
Fetched 1 row(s) in 0.71s
```

Figure Hbase.Query3.1

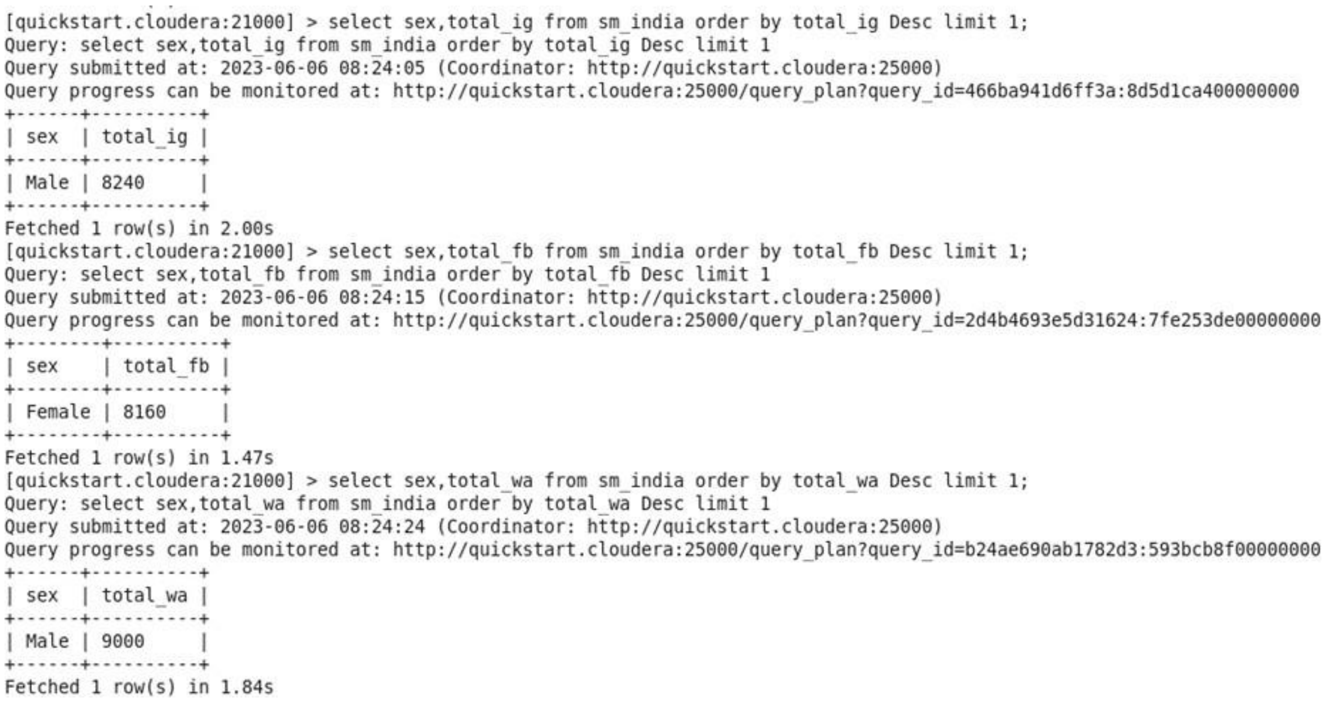
Query 3 for HBase total Facebook, Instagram and WhatsApp highest usage record based on operating systems



```
[quickstart.cloudera:21000] > select mobile_os,total_ig from sm_india order by total_ig Desc limit 1;
Query: select mobile_os,total_ig from sm_india order by total_ig Desc limit 1
Query submitted at: 2023-06-06 08:25:39 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=7b488ff59e1334ea:d946f82100000000
+-----------+----------+
| mobile_os | total_ig |
+-----------+----------+
| Android   | 8240     |
+-----------+----------+
Fetched 1 row(s) in 1.82s
[quickstart.cloudera:21000] > select mobile_os,total_fb from sm_india order by total_fb Desc limit 1;
Query: select mobile_os,total_fb from sm_india order by total_fb Desc limit 1
Query submitted at: 2023-06-06 08:25:48 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=424b0105aa598cb5:2183d18500000000
+-----------+----------+
| mobile_os | total_fb |
+-----------+----------+
| iOs       | 8160     |
+-----------+----------+
Fetched 1 row(s) in 1.09s
[quickstart.cloudera:21000] > select mobile_os,total_wa from sm_india order by total_wa Desc limit 1;
Query: select mobile_os,total_wa from sm_india order by total_wa Desc limit 1
Query submitted at: 2023-06-06 08:25:56 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=6849bb4025aaa01c:268ebb2700000000
+-----------+----------+
| mobile_os | total_wa |
+-----------+----------+
| Android   | 9000     |
+-----------+----------+
Fetched 1 row(s) in 0.89s
```

Figure Hbase.Query4.1

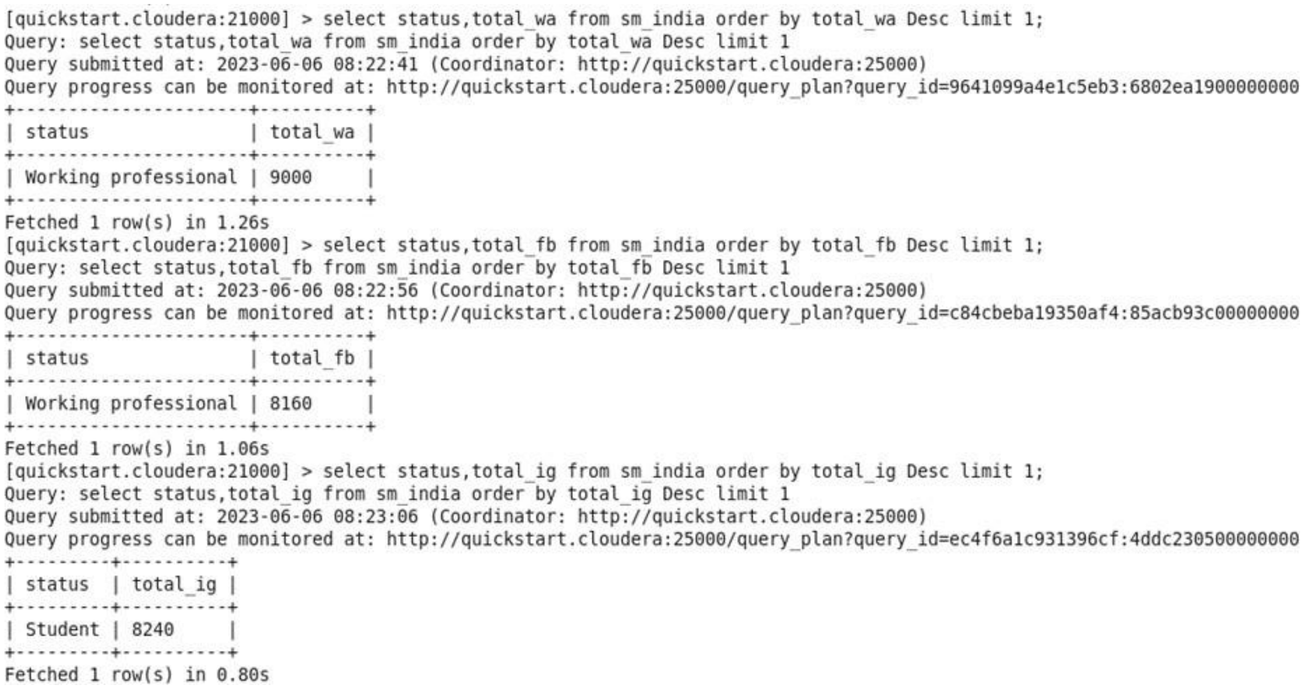
Query 4 for HBase total Facebook, Instagram and WhatsApp highest usage record based on gender



```
[quickstart.cloudera:21000] > select sex,total_ig from sm_india order by total_ig Desc limit 1;
Query: select sex,total_ig from sm_india order by total_ig Desc limit 1
Query submitted at: 2023-06-06 08:24:05 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=466ba941d6ff3a:8d5d1ca400000000
+------+----------+
| sex  | total_ig |
+------+----------+
| Male | 8240     |
+------+----------+
Fetched 1 row(s) in 2.00s
[quickstart.cloudera:21000] > select sex,total_fb from sm_india order by total_fb Desc limit 1;
Query: select sex,total_fb from sm_india order by total_fb Desc limit 1
Query submitted at: 2023-06-06 08:24:15 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=2d4b4693e5d31624:7fe253de00000000
+--------+----------+
| sex    | total_fb |
+--------+----------+
| Female | 8160     |
+--------+----------+
Fetched 1 row(s) in 1.47s
[quickstart.cloudera:21000] > select sex,total_wa from sm_india order by total_wa Desc limit 1;
Query: select sex,total_wa from sm_india order by total_wa Desc limit 1
Query submitted at: 2023-06-06 08:24:24 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=b24ae690ab1782d3:593bcb8f00000000
+------+----------+
| sex  | total_wa |
+------+----------+
| Male | 9000     |
+------+----------+
Fetched 1 row(s) in 1.84s
```

Figure Hbase.Query5.1

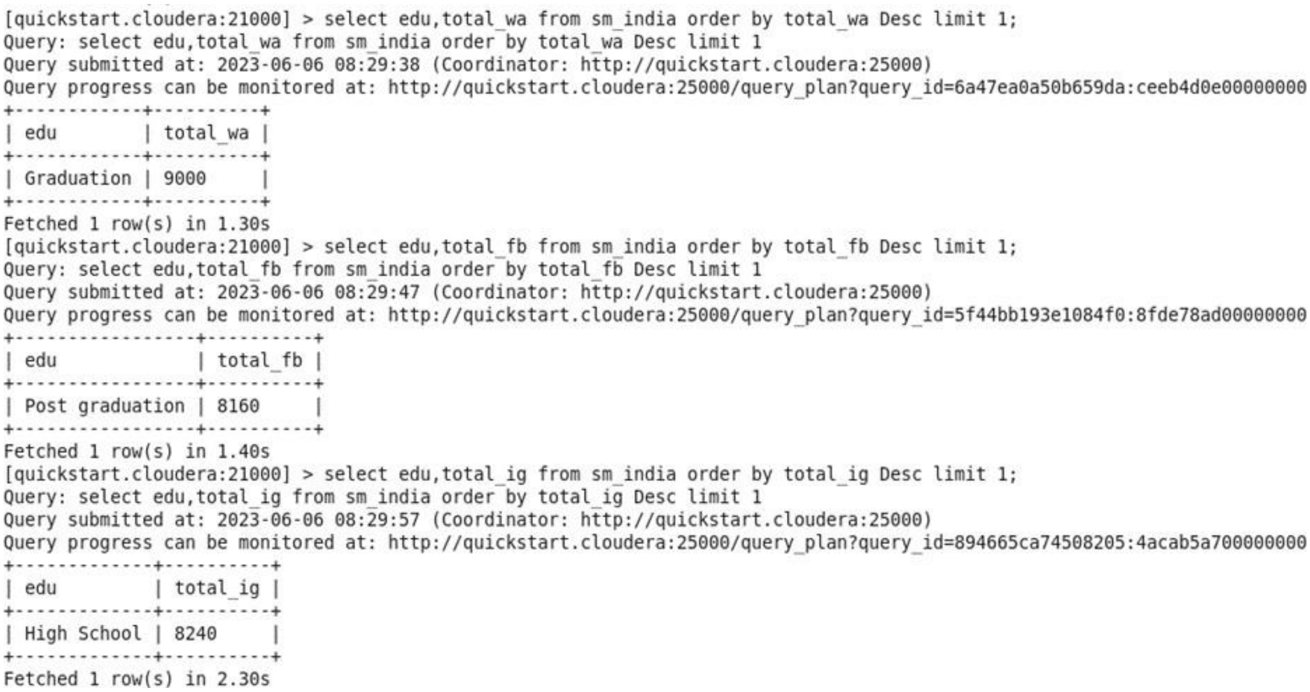
Query 5 for HBase total Facebook, Instagram and WhatsApp highest usage record based on status variable



```
[quickstart.cloudera:21000] > select status,total_wa from sm_india order by total_wa Desc limit 1;
Query: select status,total_wa from sm_india order by total_wa Desc limit 1
Query submitted at: 2023-06-06 08:22:41 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=9641099a4e1c5eb3:6802ea1900000000
+----------------------+----------+
| status               | total_wa |
+----------------------+----------+
| Working professional | 9000     |
+----------------------+----------+
Fetched 1 row(s) in 1.26s
[quickstart.cloudera:21000] > select status,total_fb from sm_india order by total_fb Desc limit 1;
Query: select status,total_fb from sm_india order by total_fb Desc limit 1
Query submitted at: 2023-06-06 08:22:56 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=c84cbeba19350af4:85acb93c00000000
+----------------------+----------+
| status               | total_fb |
+----------------------+----------+
| Working professional | 8160     |
+----------------------+----------+
Fetched 1 row(s) in 1.06s
[quickstart.cloudera:21000] > select status,total_ig from sm_india order by total_ig Desc limit 1;
Query: select status,total_ig from sm_india order by total_ig Desc limit 1
Query submitted at: 2023-06-06 08:23:06 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=ec4f6a1c931396cf:4ddc230500000000
+---------+----------+
| status  | total_ig |
+---------+----------+
| Student | 8240     |
+---------+----------+
Fetched 1 row(s) in 0.80s
```

Figure Hbase.Query6.1

Query 6 for HBase total Facebook, Instagram and WhatsApp highest usage record based on education



```
[quickstart.cloudera:21000] > select edu,total_wa from sm_india order by total_wa Desc limit 1;
Query: select edu,total_wa from sm_india order by total_wa Desc limit 1
Query submitted at: 2023-06-06 08:29:38 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=6a47ea0a50b659da:ceeb4d0e00000000
+------------+----------+
| edu        | total_wa |
+------------+----------+
| Graduation | 9000     |
+------------+----------+
Fetched 1 row(s) in 1.30s
[quickstart.cloudera:21000] > select edu,total_fb from sm_india order by total_fb Desc limit 1;
Query: select edu,total_fb from sm_india order by total_fb Desc limit 1
Query submitted at: 2023-06-06 08:29:47 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=5f44bb193e1084f0:8fde78ad00000000
+-----------------+----------+
| edu             | total_fb |
+-----------------+----------+
| Post graduation | 8160     |
+-----------------+----------+
Fetched 1 row(s) in 1.40s
[quickstart.cloudera:21000] > select edu,total_ig from sm_india order by total_ig Desc limit 1;
Query: select edu,total_ig from sm_india order by total_ig Desc limit 1
Query submitted at: 2023-06-06 08:29:57 (Coordinator: http://quickstart.cloudera:25000)
Query progress can be monitored at: http://quickstart.cloudera:25000/query_plan?query_id=894665ca74508205:4acab5a700000000
+-------------+----------+
| edu         | total_ig |
+-------------+----------+
| High School | 8240     |
+-------------+----------+
Fetched 1 row(s) in 2.30s
```