**BIG DATA**

CSDAIA24GP003

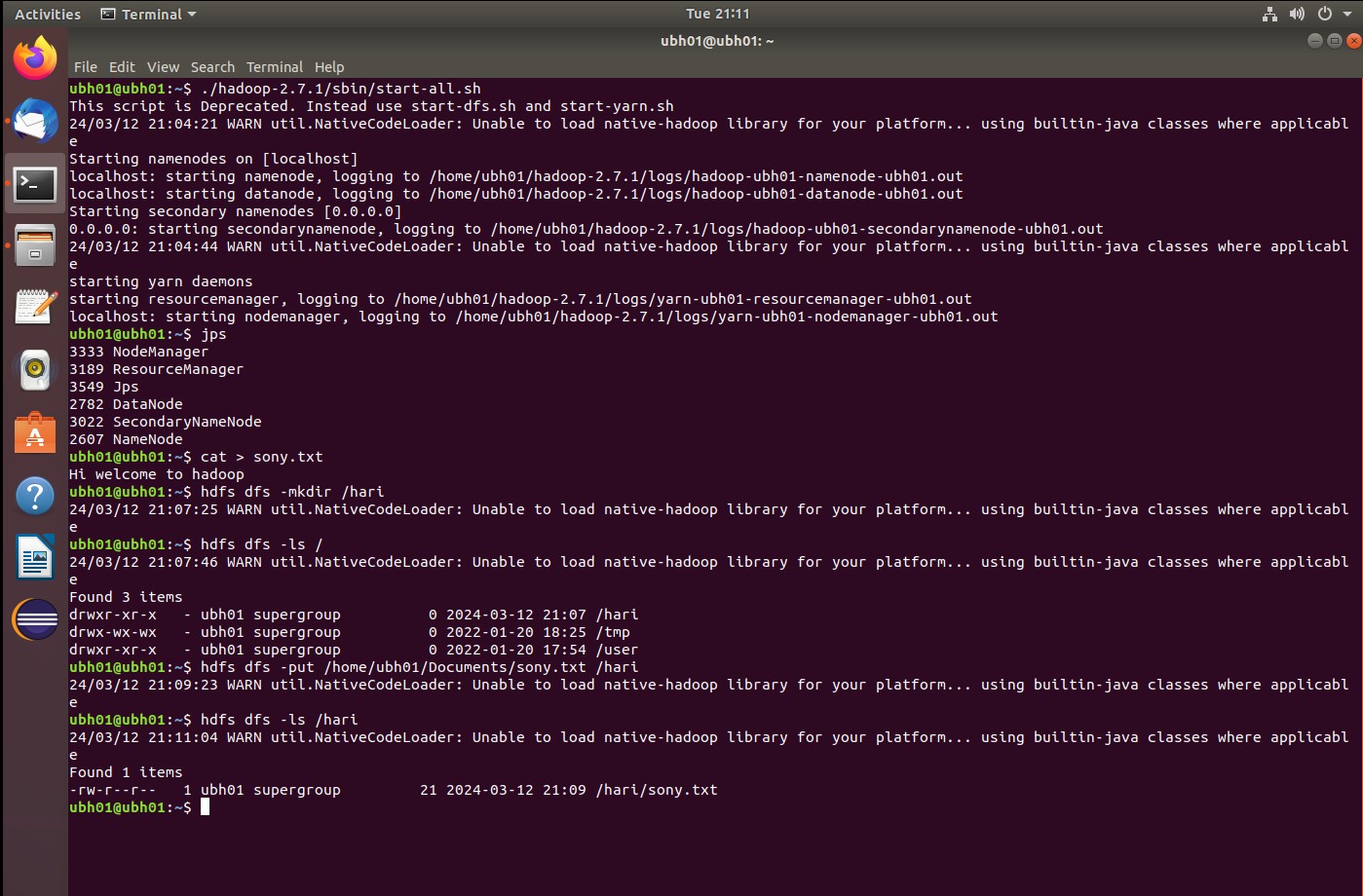
Chakka Sudheer Kumar

Emp-id:2320418

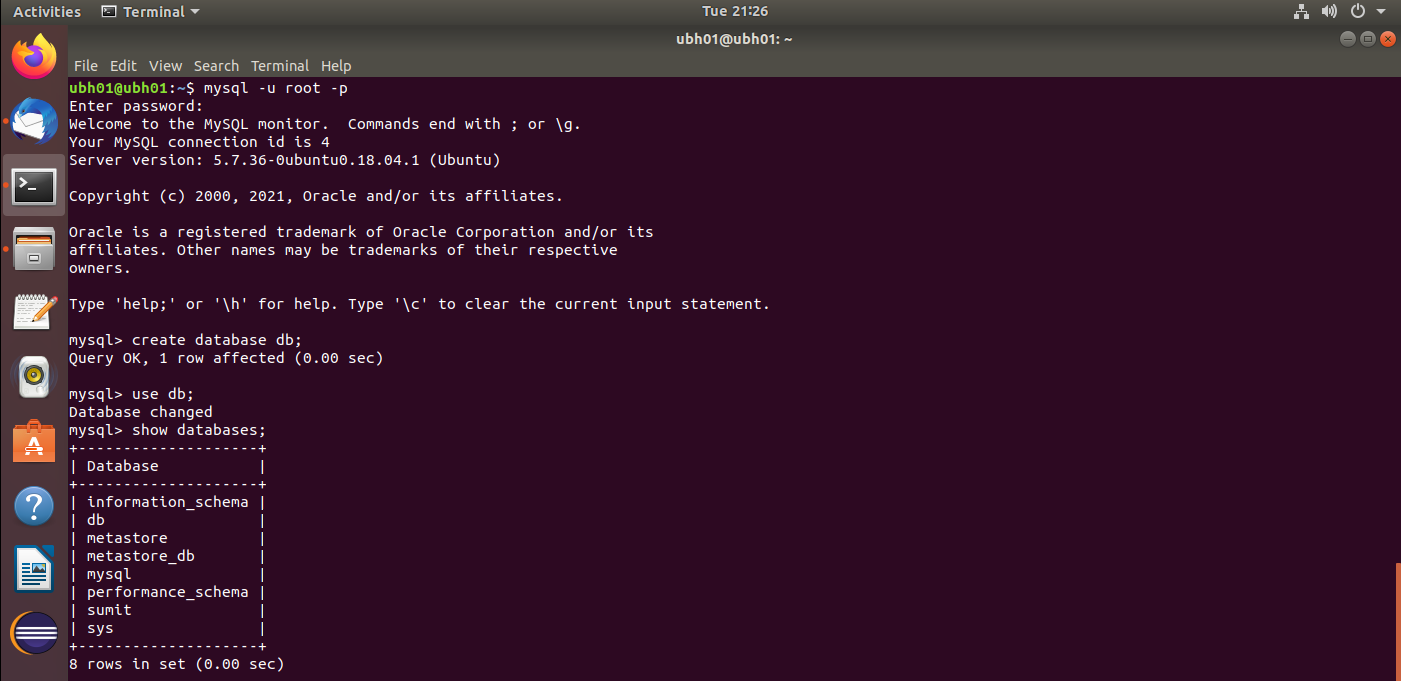
**Hadoop**

* Hadoop is an open-source framework used for storing and processing big data.
* It provides a distributed file system called Hadoop Distributed File System (HDFS).

1. To start the Hadoop we use --**./hadoop-2.7.1/sbin/start-all.sh**
2. To check whether Hadoop daemons are running or not --- **jps**
3. Create a file in local -- **cat > filename.txt**
4. Create a directory in local --- **mkdir name**
5. Just check whether file is copied or not using --- **ls**



\*\*\* Start MYSQL and create a database with named db



* Created table with table name student

Syntax:

Create table student(name varchar(20),roll int primary key, age int);

A screenshot of a computer

Description automatically generated

* Inserted the values manually or load values from csv file as shown below:

Syntax:

1. Load data local infile ‘/home/ubh01/Documents/students.csv’ into table student;

Or

1. Insert into student values ( name, roll, age);

**SQOOP:**

It is a command line tool that helps to import and export data from database.

* **Sqoop import:**

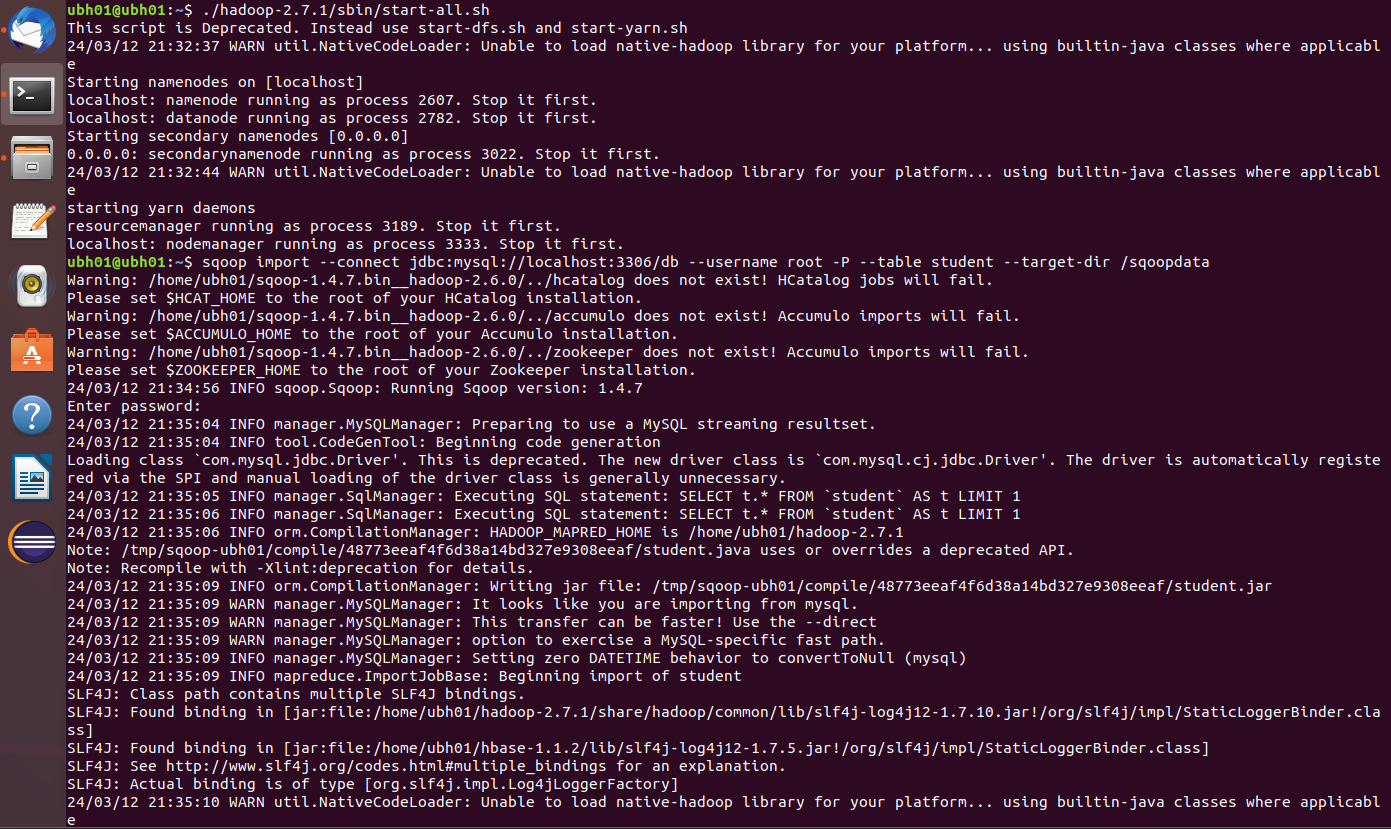
Sqoop import is used to import the data from RDBMS to HDFS.

**\*\*\*To start the Hadoop we use ./hadoop-2.7.1/sbin/start-all.sh**

* Using the below command the table nycbike from database gcp is imported to hdfs through sqoop import

Syntax:

Sqoop import –connectjdbc:mysql://localhost:3306/gcp –username root -p –query –table student –target-dir /sqoopdata

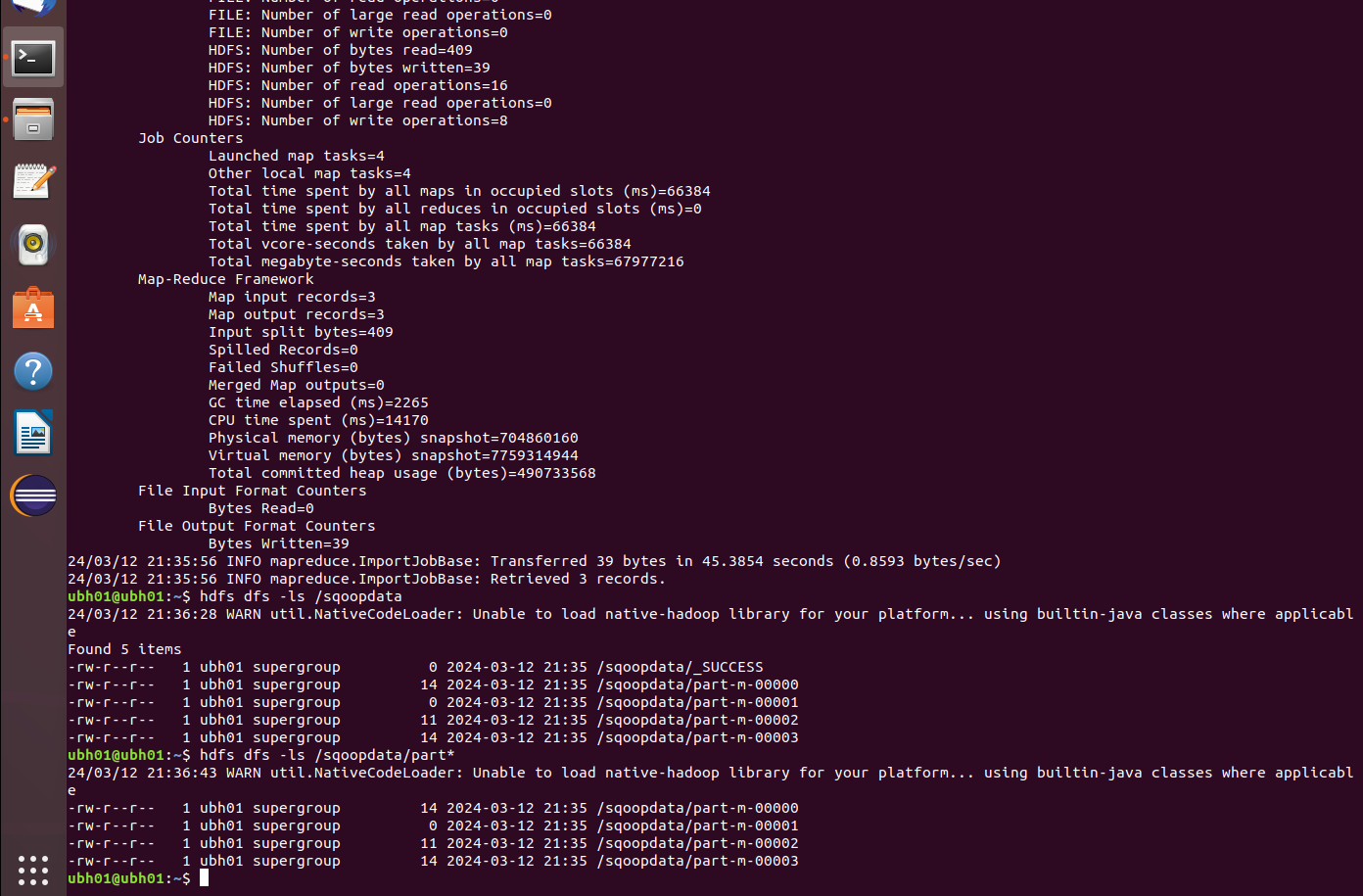


* Listing the output files of sqoopdata file which includes \_SUCCESS and PART FILES using ls command.

A screenshot of a computer

Description automatically generated

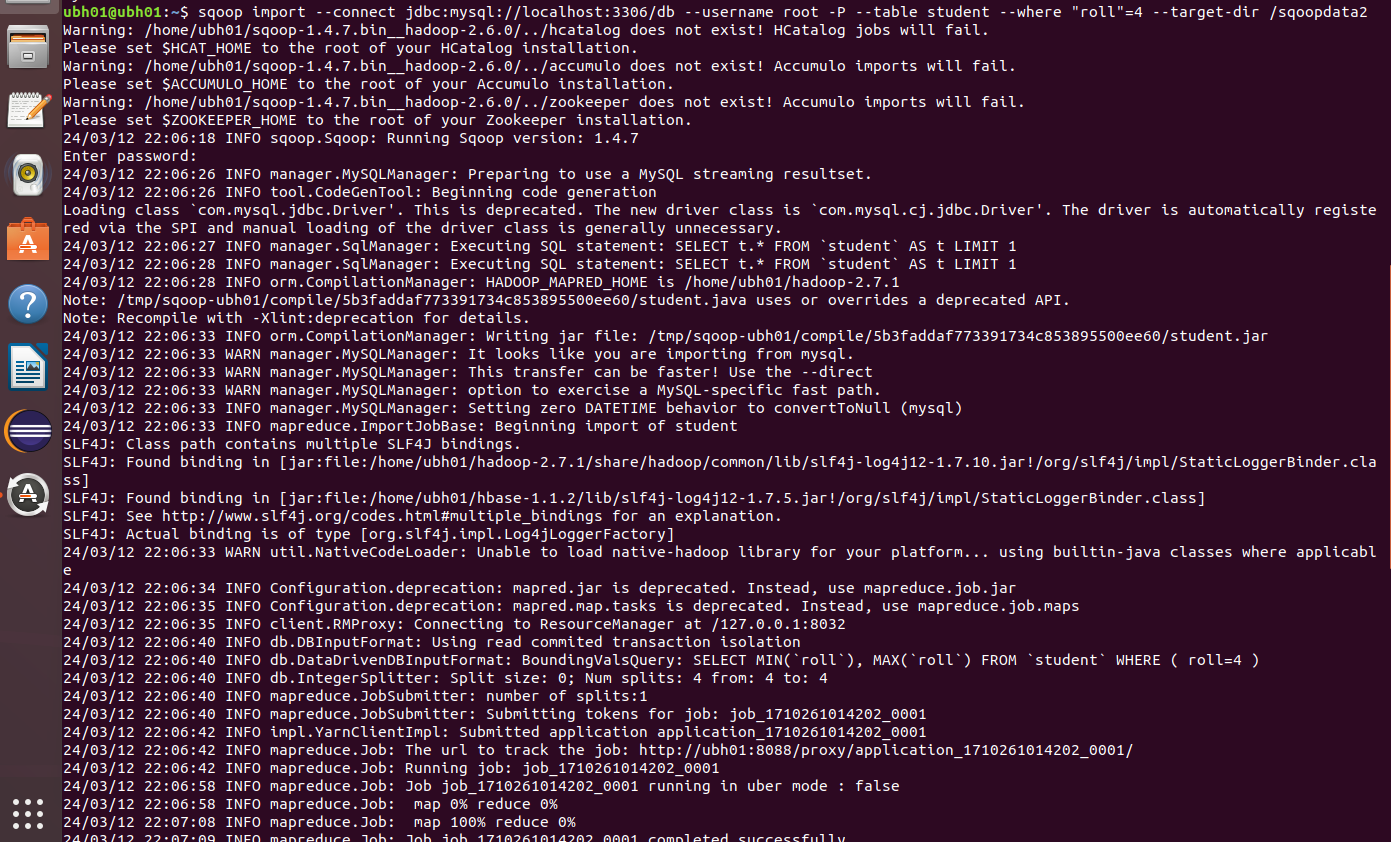
* Display the content inside PART FILE using cat command.

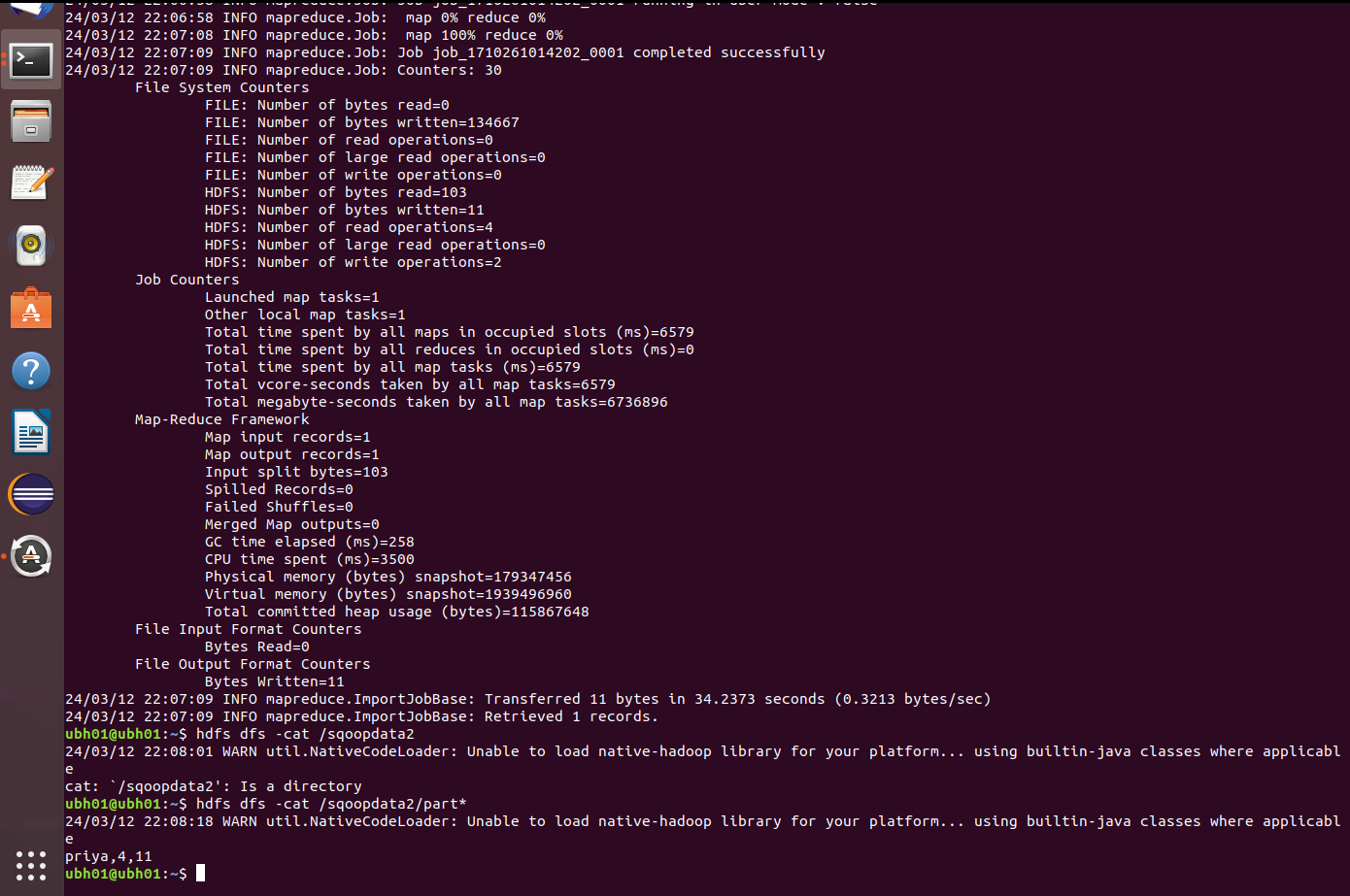


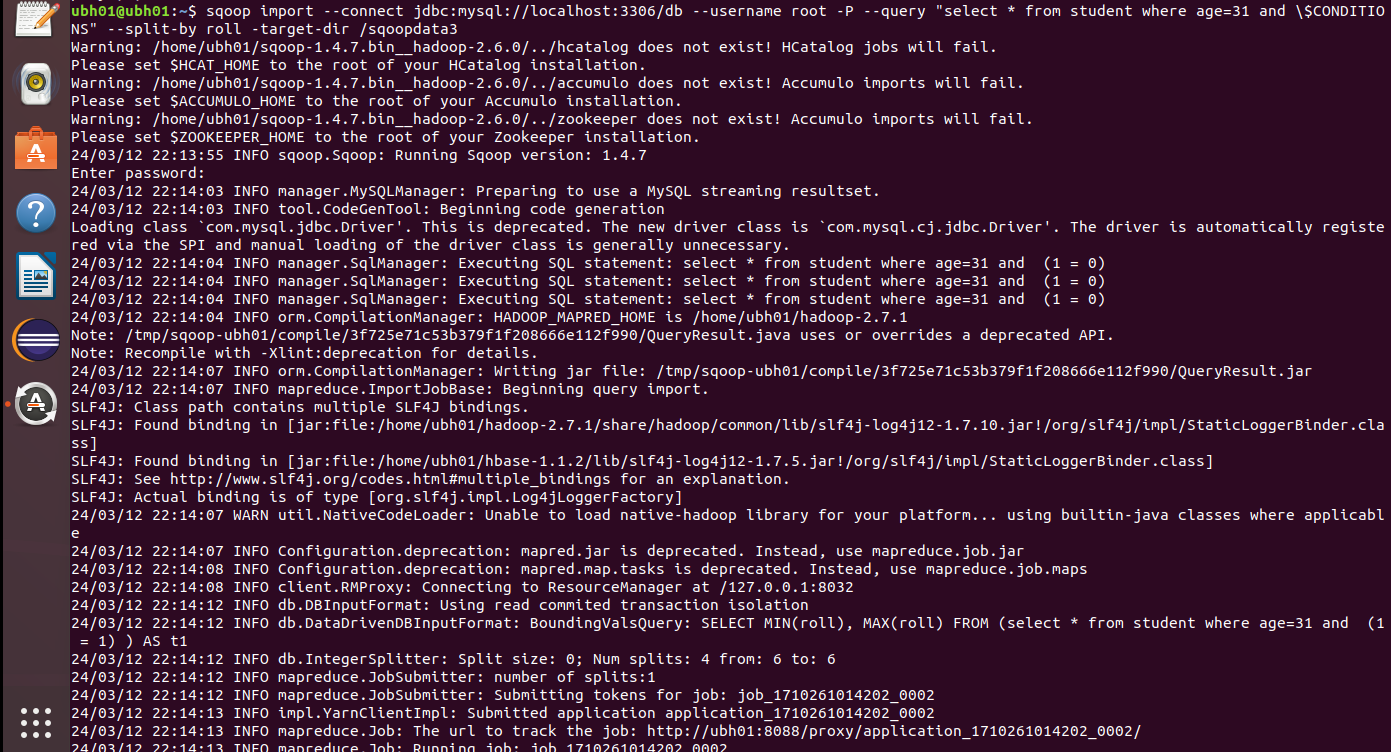
* Sqoop import command with **where** clause.

Syntax:

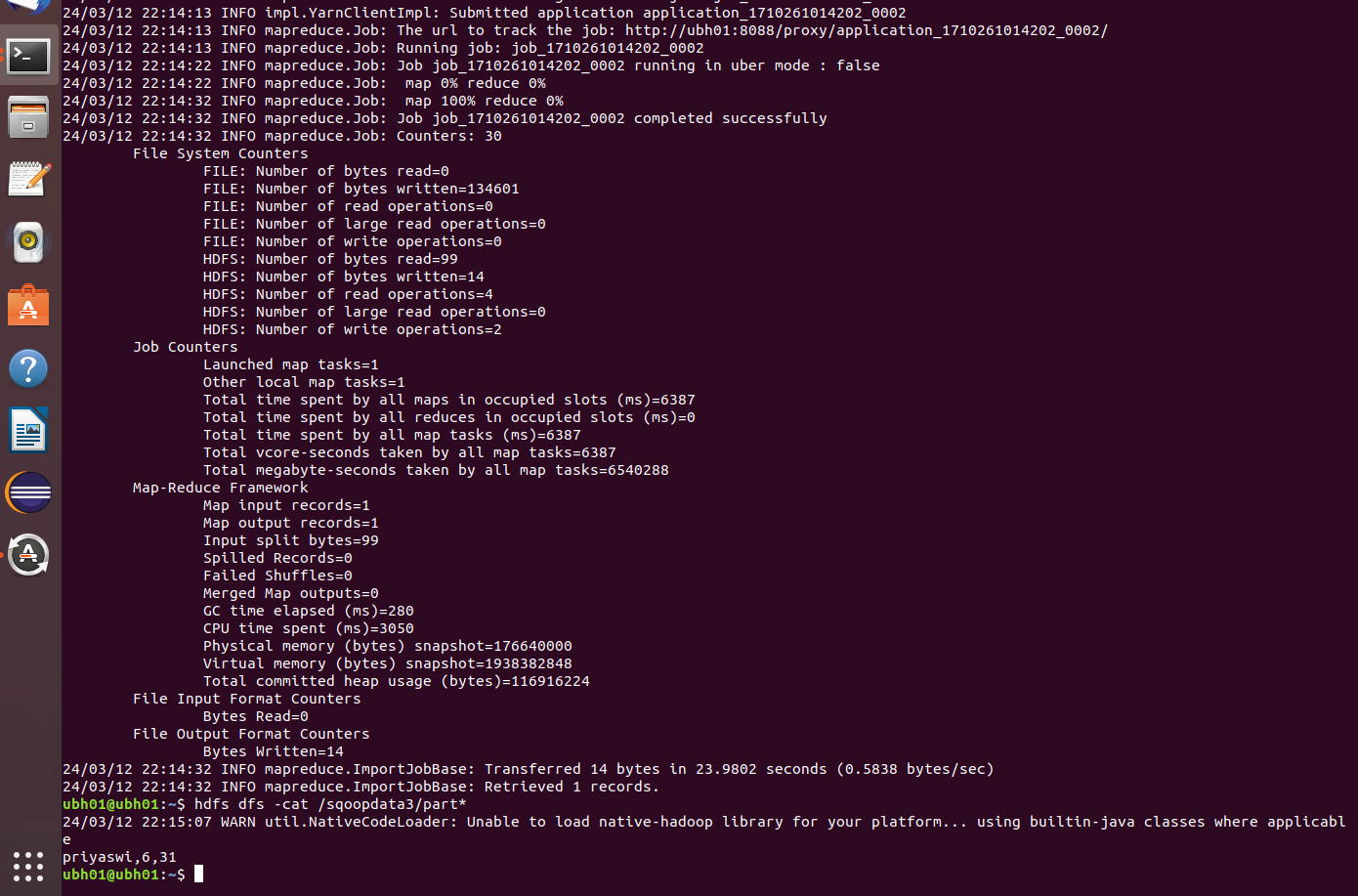
Sqoop import –connect jdbc:mysql://localhost:3306/db –username root -p – query “select \* from student where ‘roll’=4 –target-dir /sqoopdata2



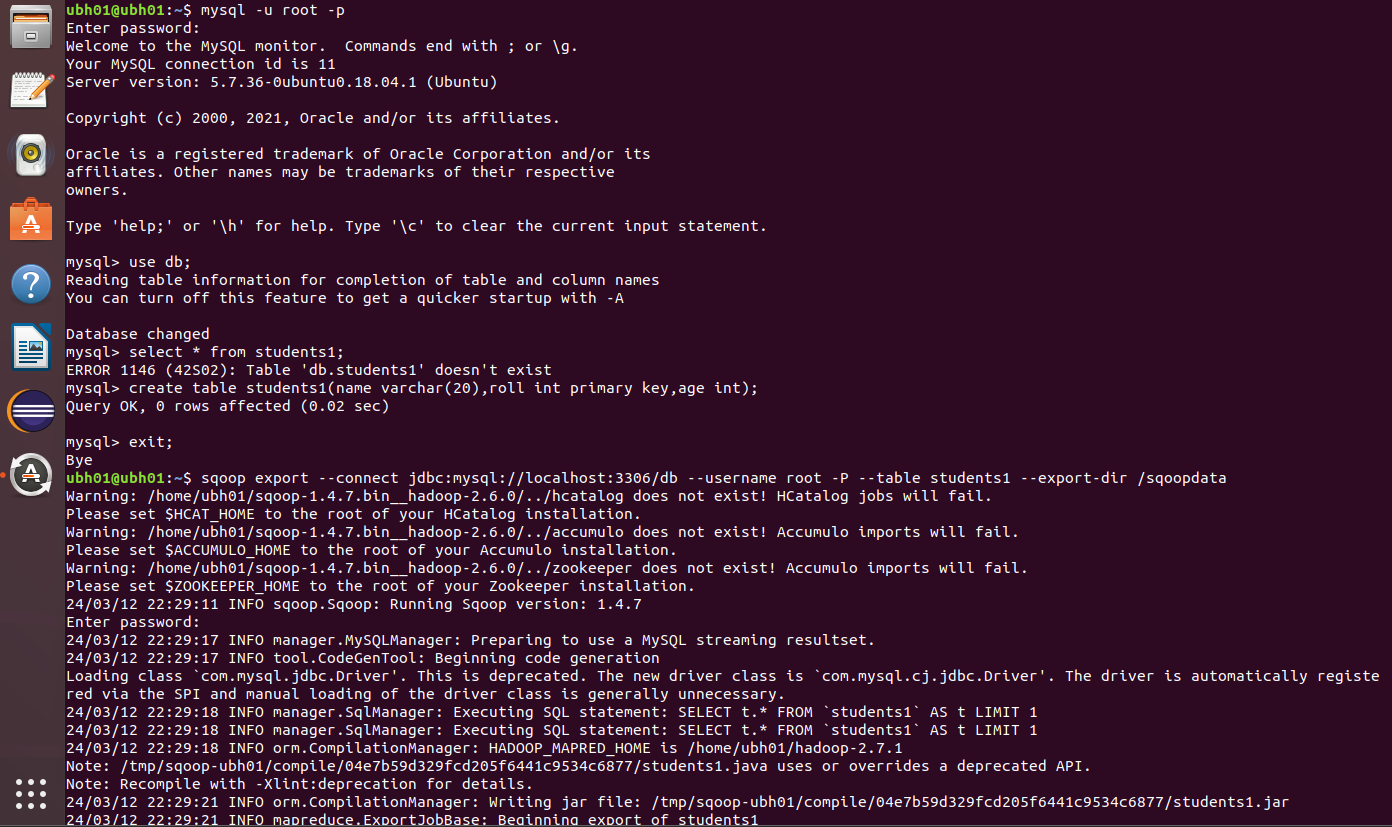
* If we want to verify whether values that satisfies where condition imported to hdfs or not, run the below hdfs command **--hdfs dfs -cat /sqoopdata2/part\***
* 
* Sqoop import with –query command where we will specify mysql query and \$CONDITIONS that is a placeholder in query which divide the data with more where clauses internally and –split-by is used to split the data by particular column that is primary key.



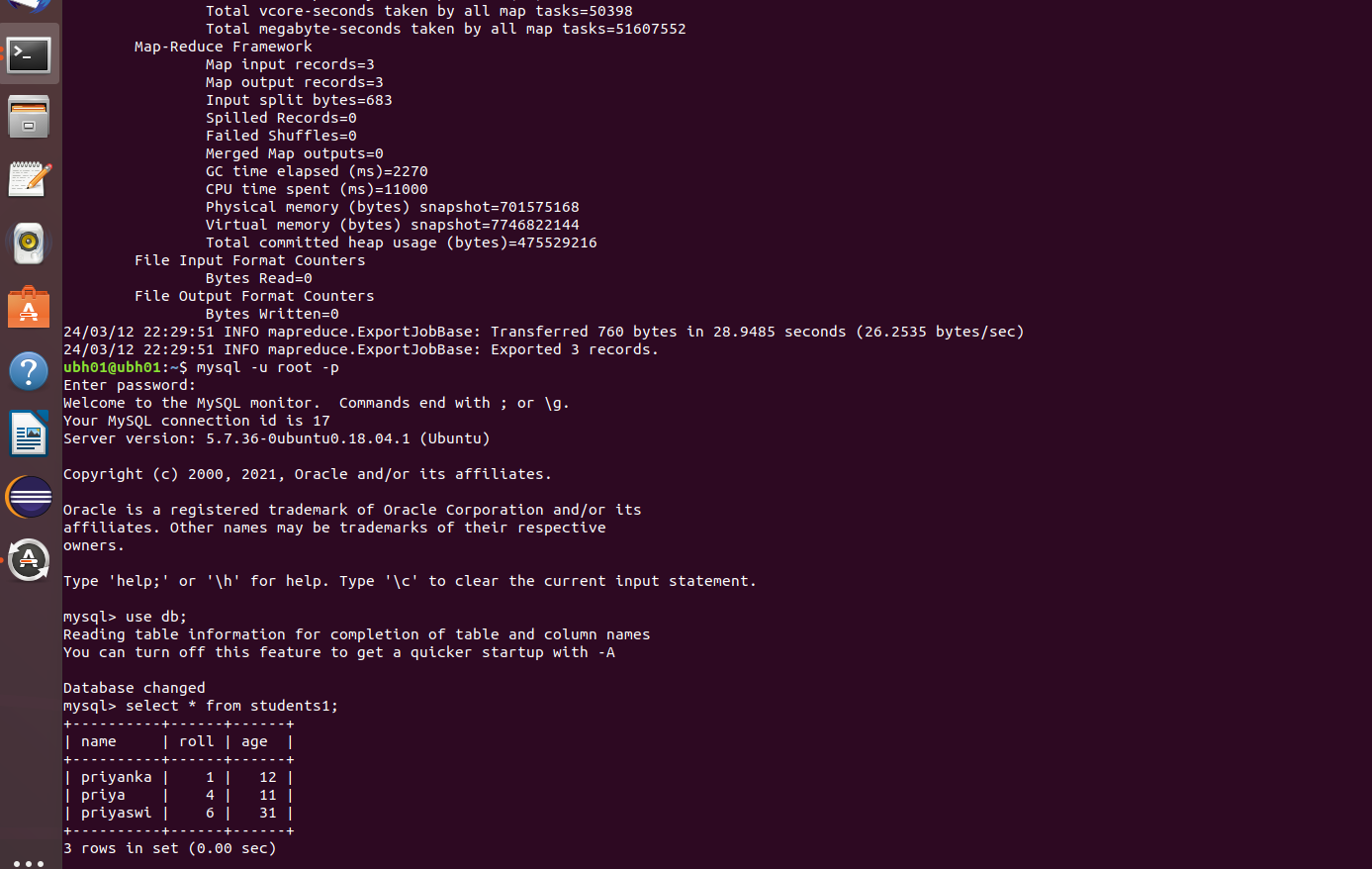
* If we want to verify whether values that satisfies QUERY imported to hdfs or not, run the below hdfs command **--hdfs dfs -cat /sqoopdata3/part\***



* **Sqoop export**:
* Sqoop export is used to import the data from HDFS to RDBMS.
* Here we have to mention export keyword and –export-directory and before this we have to create a empty schema or table in mysql.



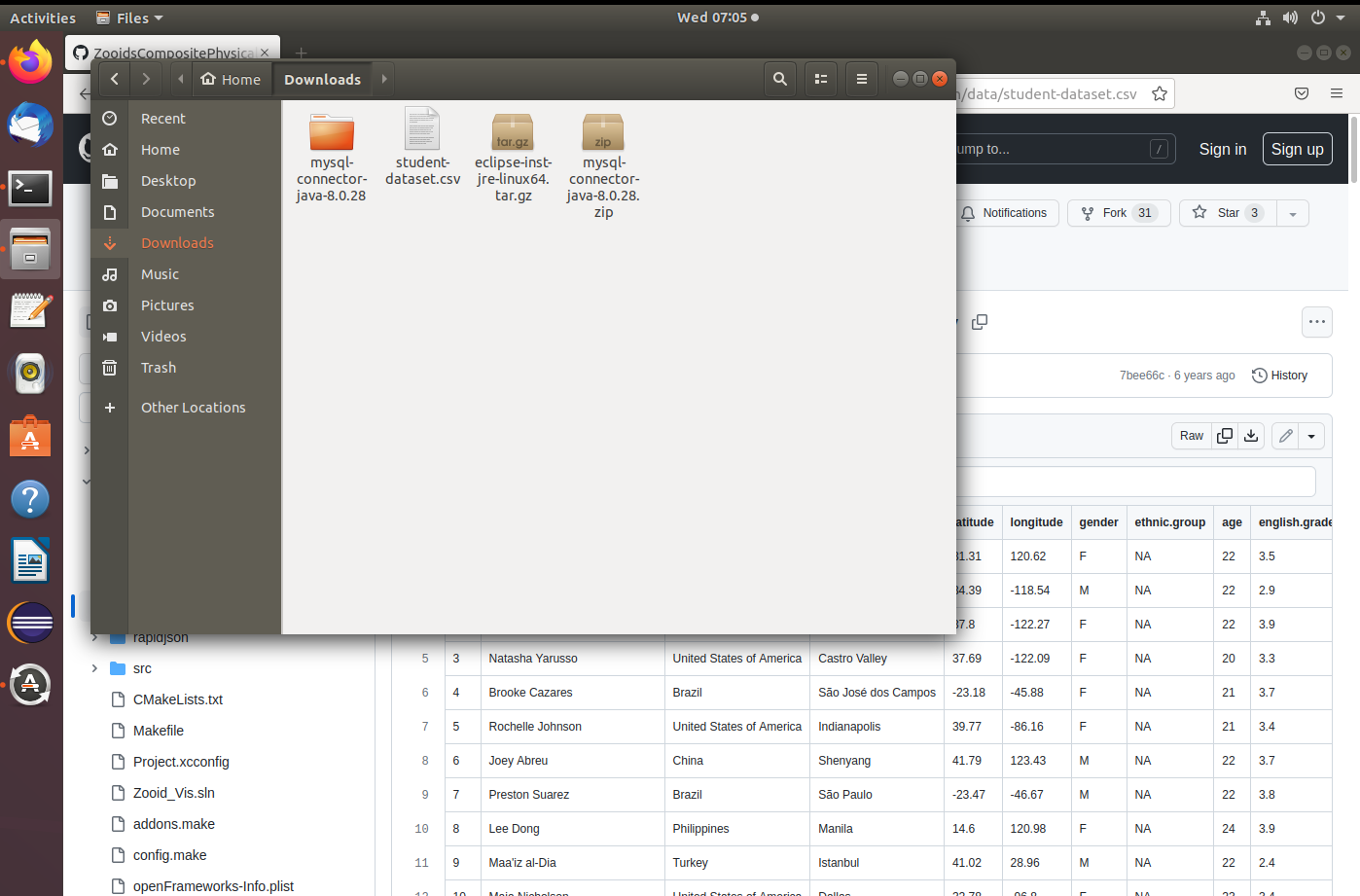
* If we want to verify whether values are exported from hdfs to mysql or not by running the below command in mysql—select \* from students1;



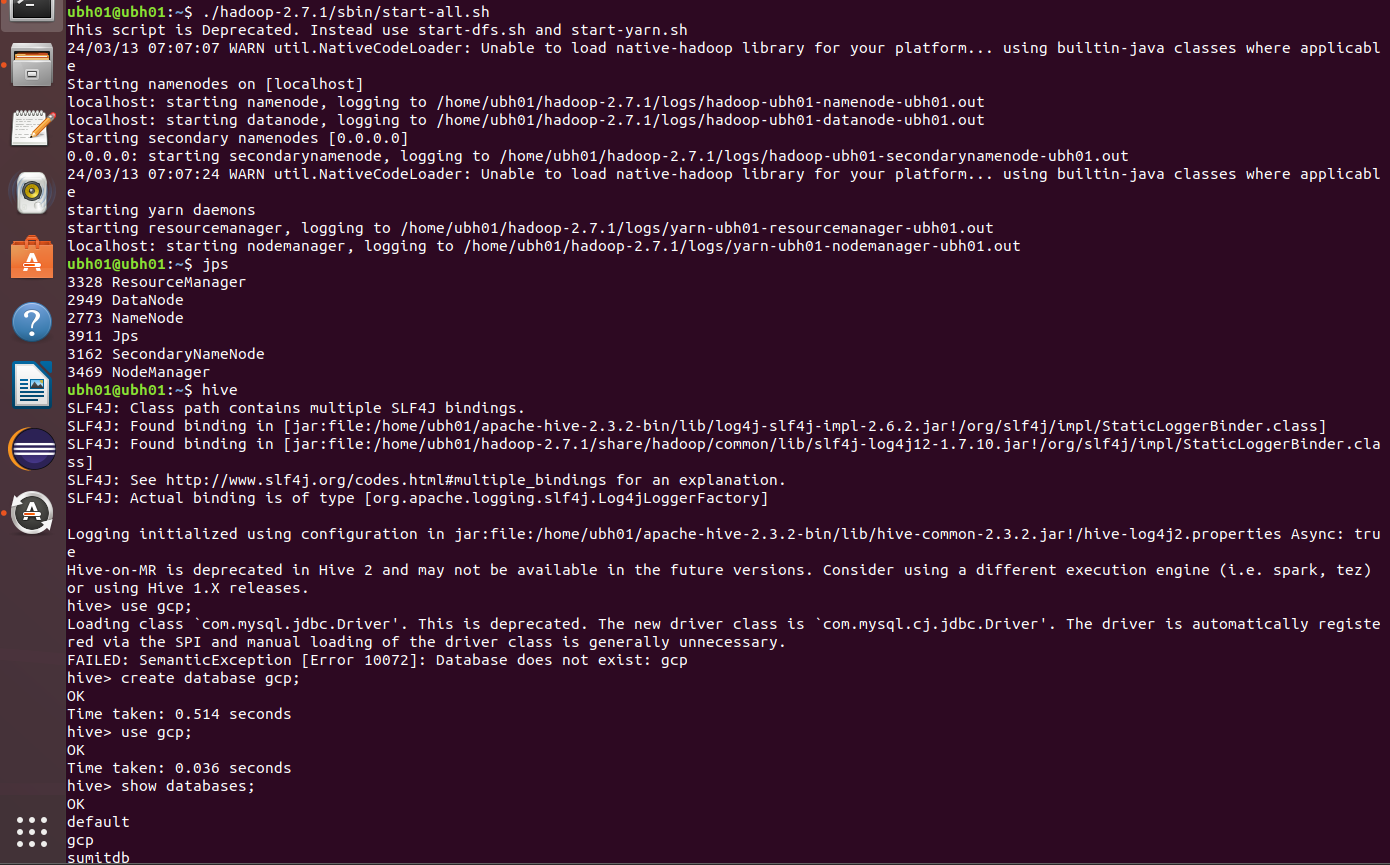
**HIVE**

Hive is a data warehouse system which is used for querying and analysing large datasets stored in HDFS.

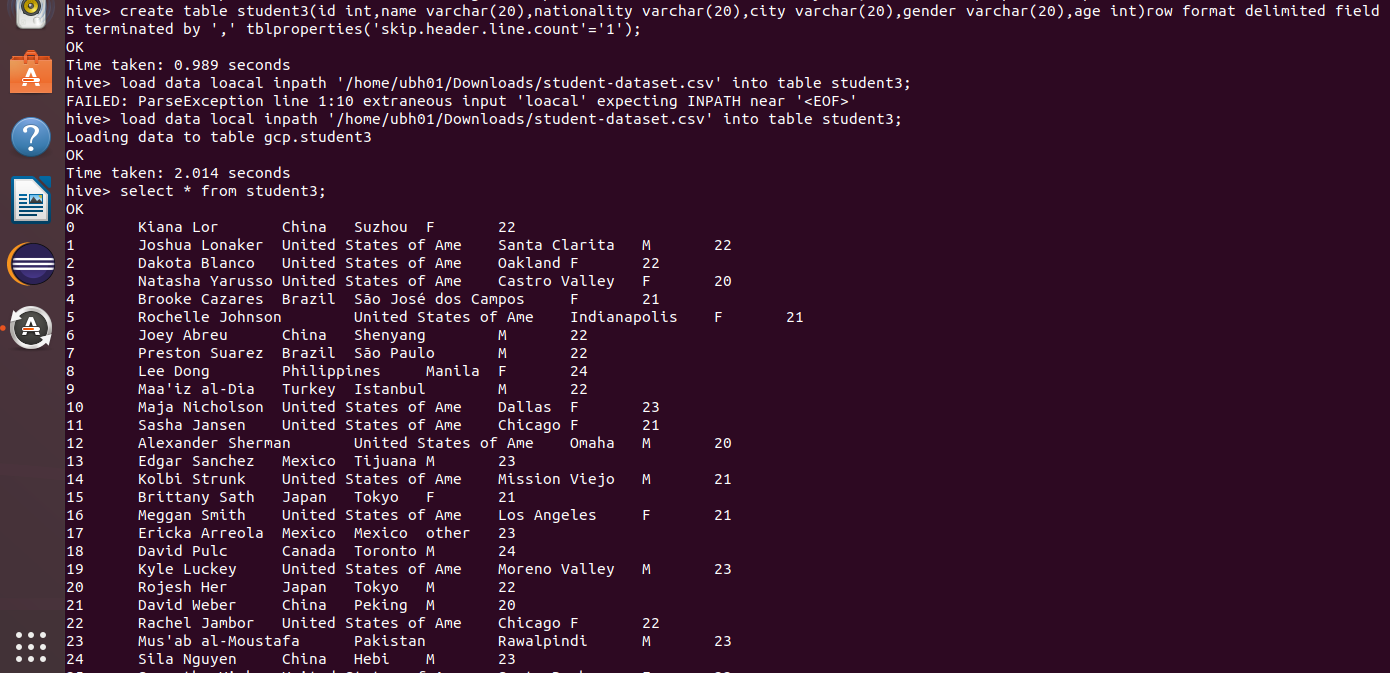
In hive here we are downloading the dataset(student-dataset.csv) from github.



* To entering into the hive, use **hive** command.
* Create a database and use database.



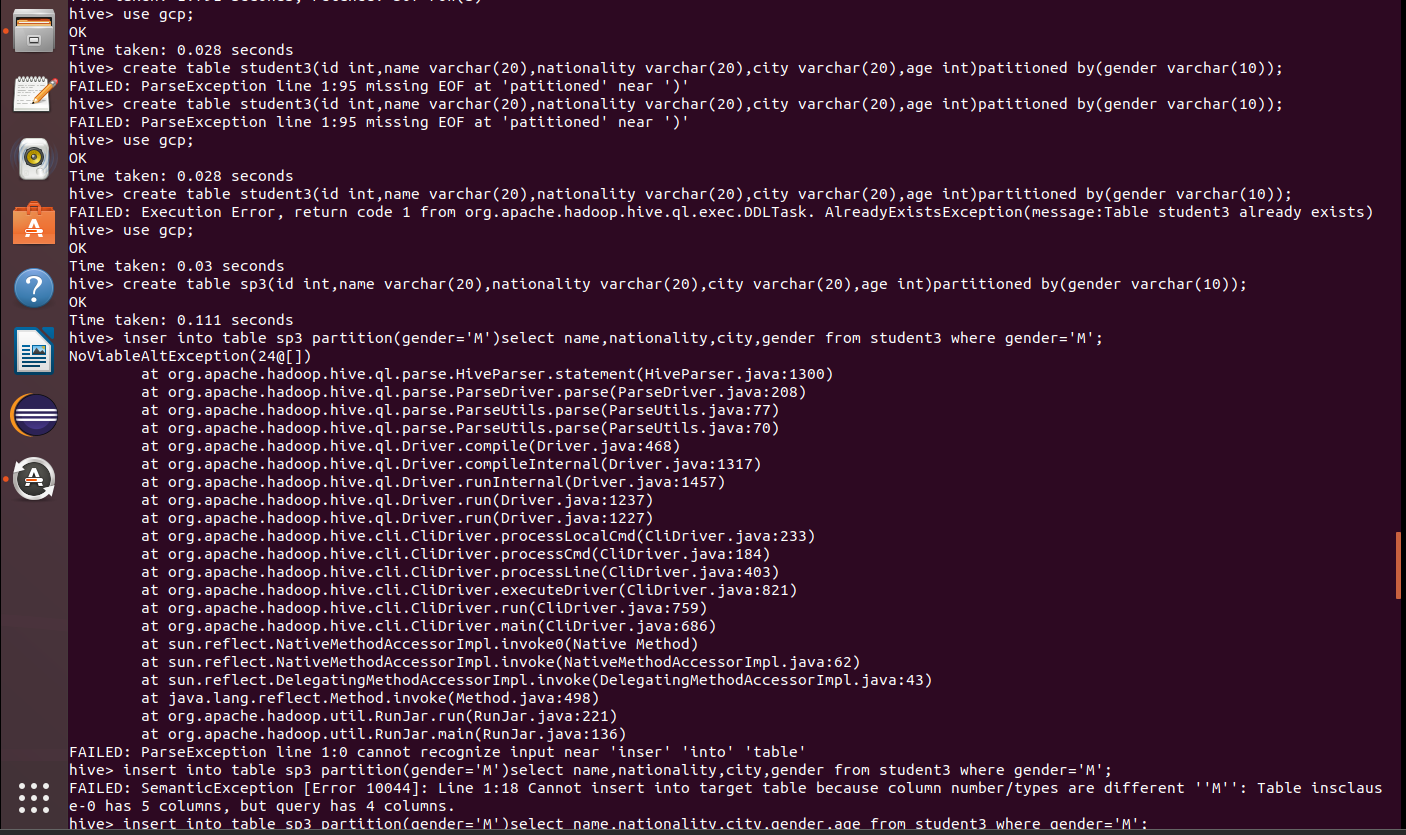
* Here we are creating a hive table and inserted values into table using load command.
* To verify whether the data is loaded are not, use command**--select \* from student3;**



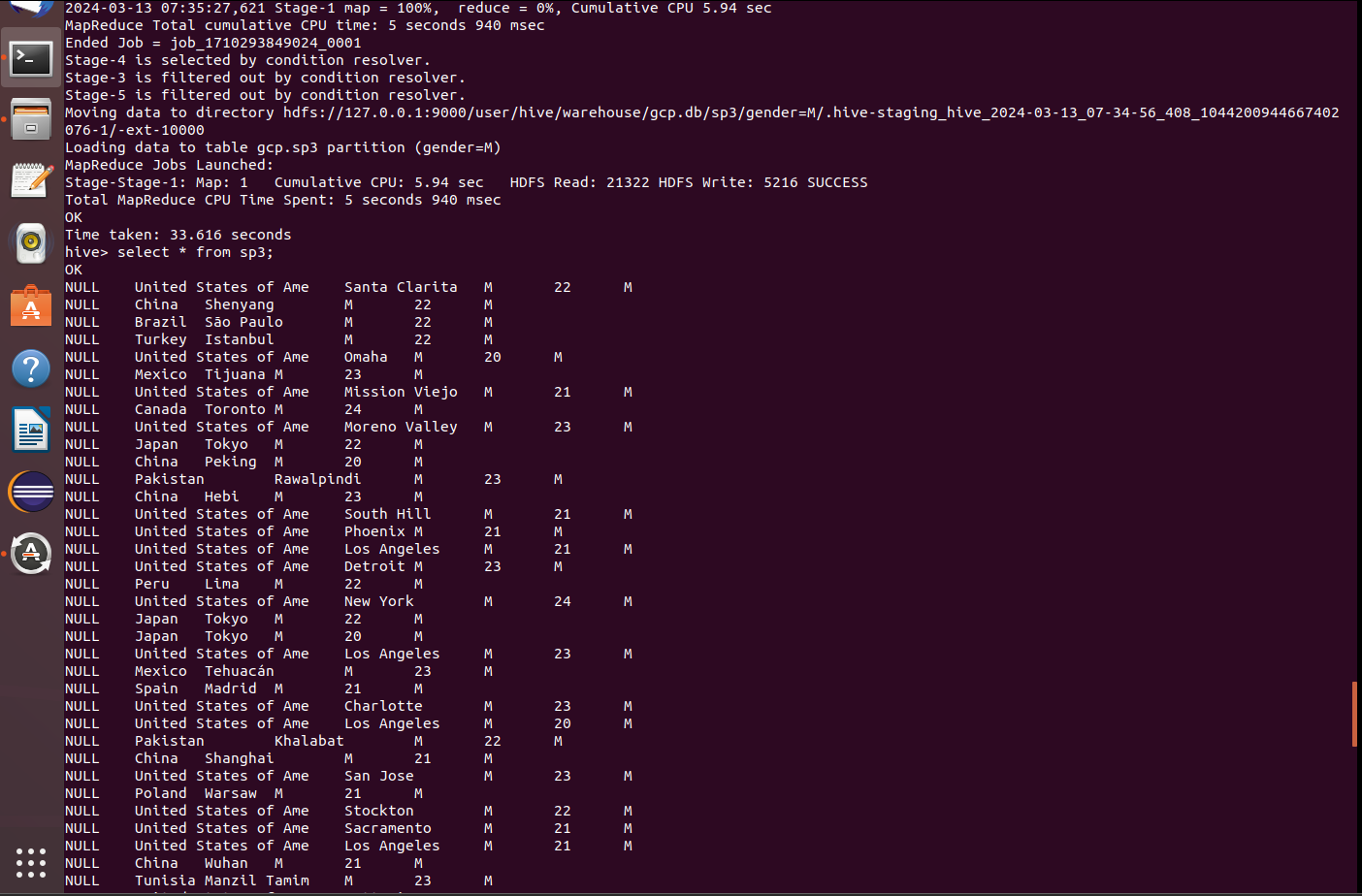
**Hive Partitioning**:

Create a partition table and follow either static or dynamic partitioning to insert values from main table following conditions.

* Static partitioning:
* It is setup when the table is created and remains static, meaning the partitions don’t change unless explicitly altered by the user.
* Here we are inserting values of a main table into partition table using static partitioning.



* Here is the partition table for partition gender = ’M’ shown below



* **Dynamic partitioning**:
  + It involves automatically creating partitions for data based on the values of specified columns during the insertion process.
  + Here we are inserting values of a main table into partition table using dynamic partitioning.

A computer screen shot of text

Description automatically generated

* To verify whether the data is loaded are not, use command**--select \* from dp;**

A computer screen with white dots

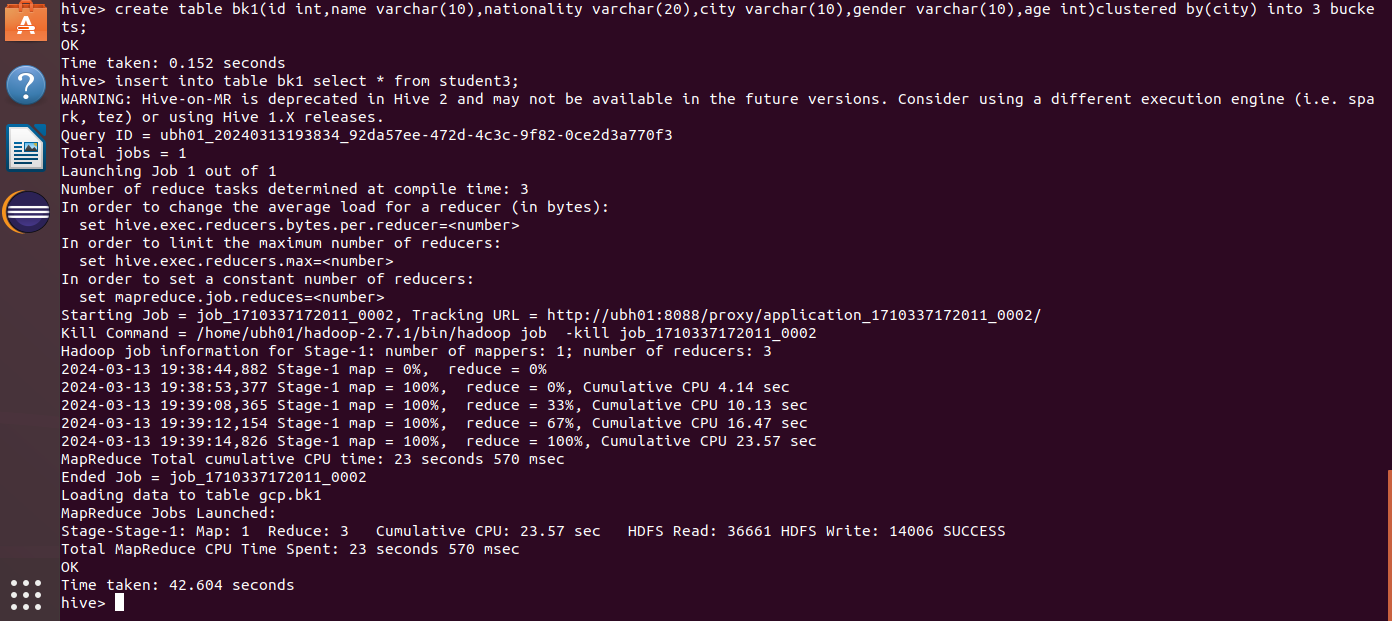
Description automatically generated

**Bucketing:**

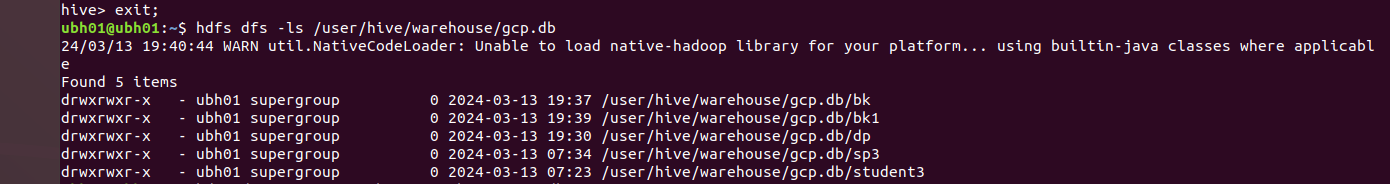
It is a method of dividing data within partitions into fixed-size buckets based on the hash value of a specified column. It helps distribute data evenly, Improving the query performance by reducing the data.

* Create a bucket table and mention how many buckets you need.
* Now insert main table values into bucket table using--

**insert into table buckettablename select \* from maintablename**



Check for bucket creation in hive directory of hdfs:

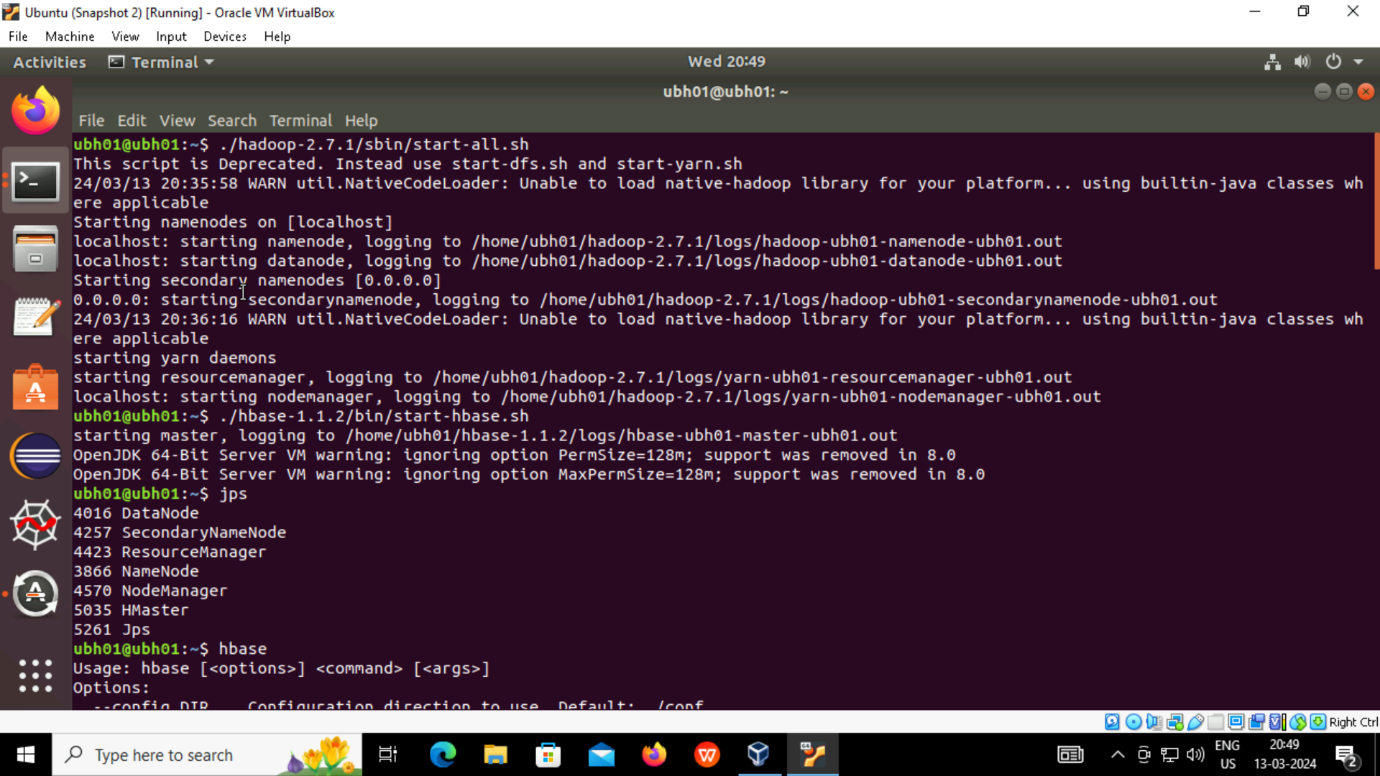


**HBase:**

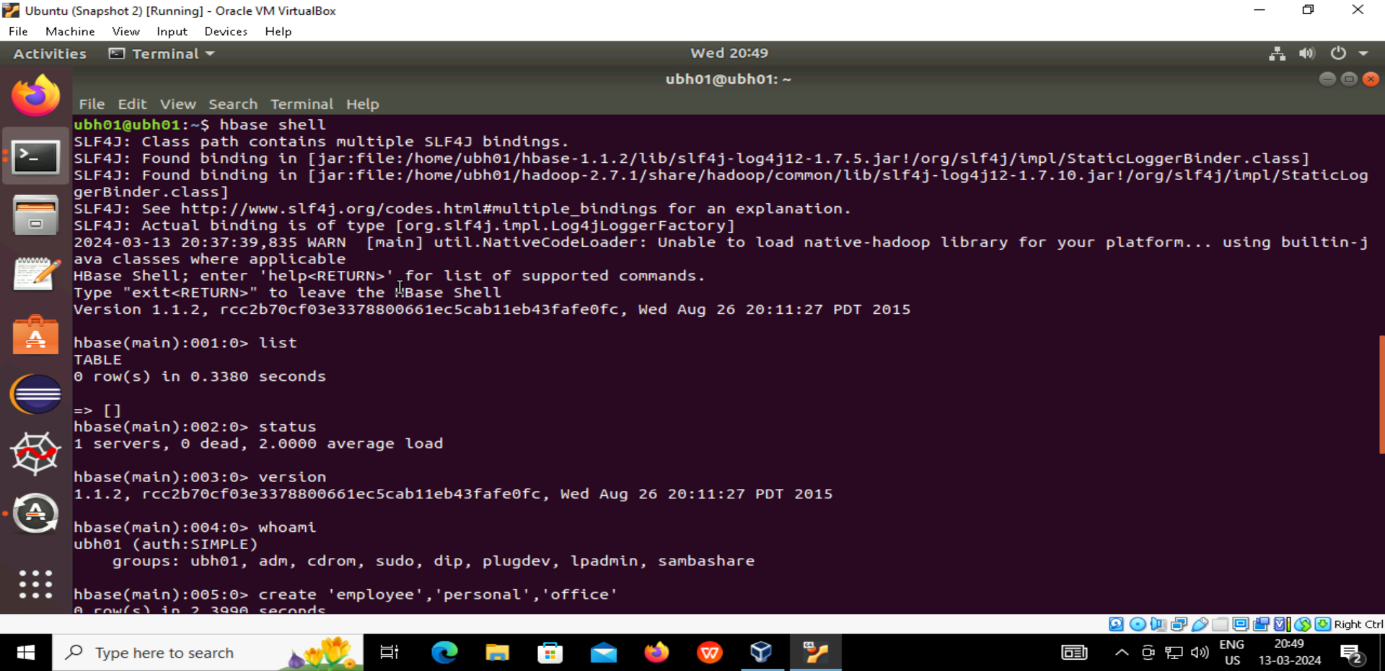
HBase is an open-source, distributed, column-oriented database built on top of the HDFS.

It provides real-time read/write access to large datasets, with horizontal scalability and fault tolerance.

* Start Hadoop service by running Hadoop start command
* Then start HBase service by running command –**hbase**



* Now start hbase script by running command – **hbase shell**
* To verify list of tables in hbase, the command is – **list**
* To check status and version of hbase, the commands are – **status** and – **version**
* To verify current user, the command is –whoami



* To create columnar table in HBase,

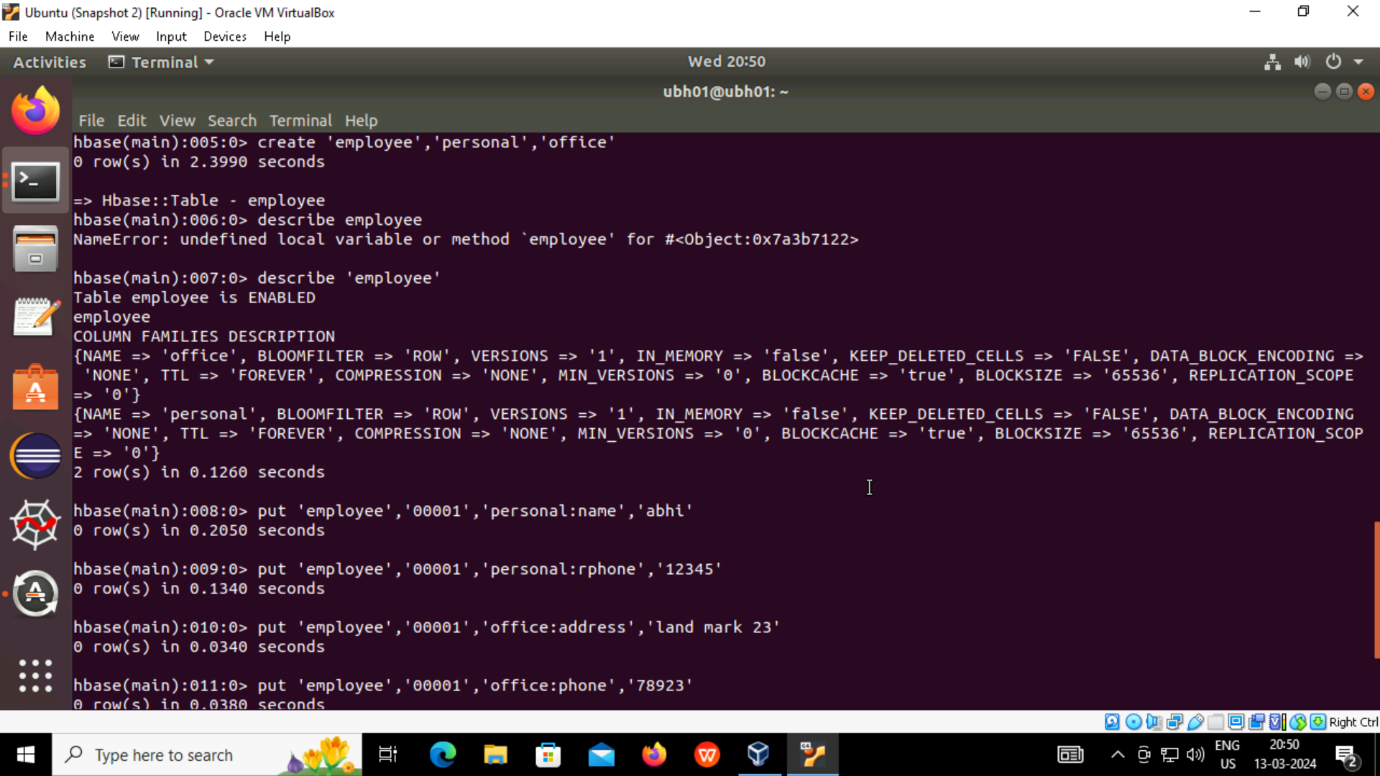
Syntax:

create ‘tablename’,’columnarfamily\_name1’,’columnarfamily\_name2’

* To get details of a table, the command is –**describe ‘tablename’**
* To insert a value into table,

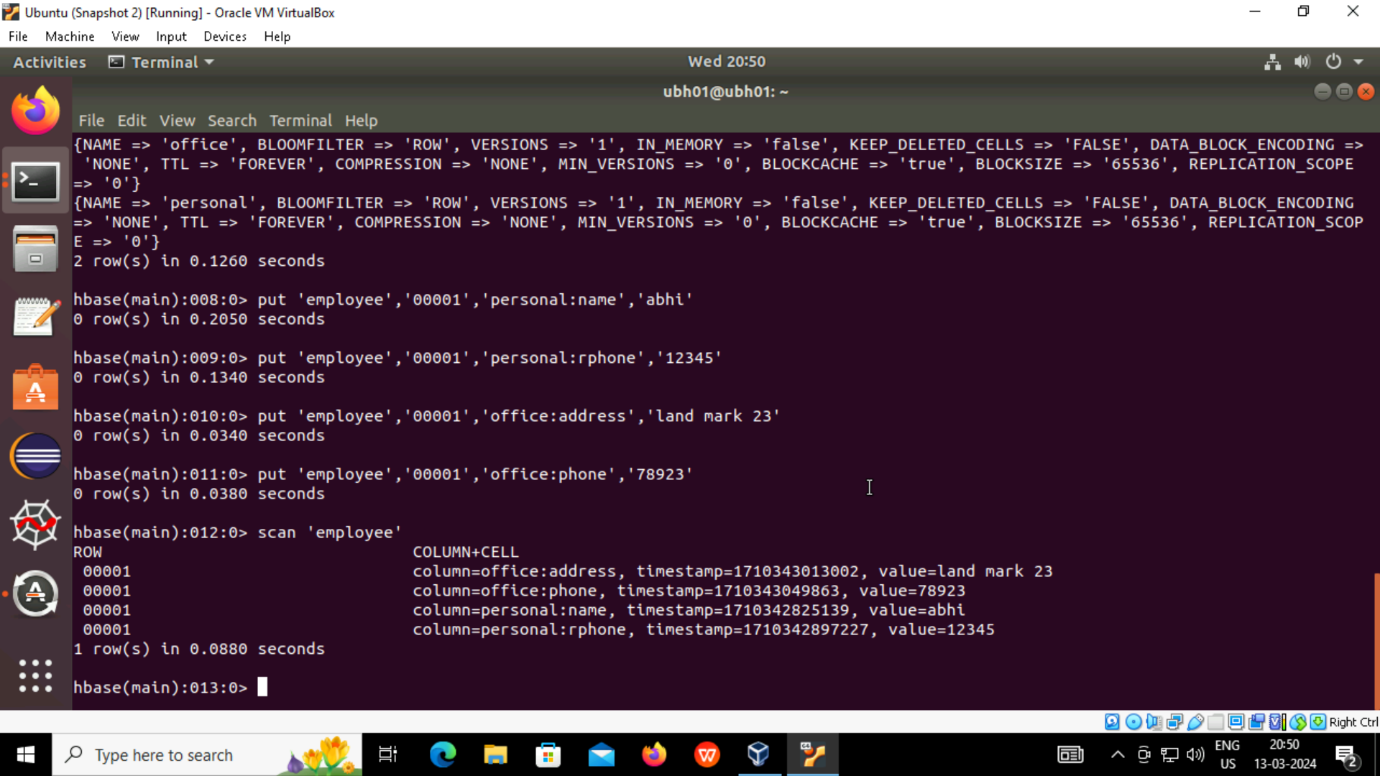
Syntax:

**put ‘tablename’,’rowkey’,’columnarfamily\_name:column\_name’,’value’**



* After inserting all values into table, to show all values in the table,

The command is – **scan ‘tablename’**



* To retrieve a particular row of a columnar table,

The command is –**get ‘tablename’,’rowkey’**

