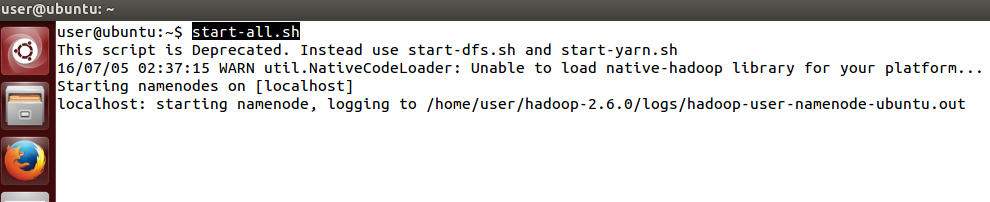
Hadoop Retails Project steps

===========================

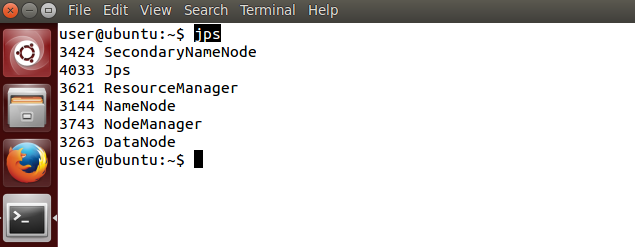
Step 1 : Start your hadoop into terminal

Command : **start-all.sh**



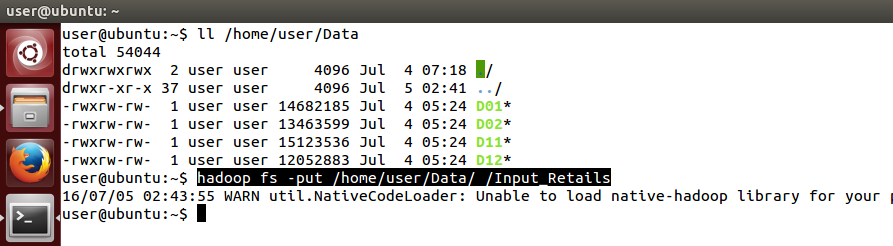
Make sure all nodes are running (**DataNode, NameNode. ResourceManager, SecondaryNamenode and NodeManager** )

Command : jps



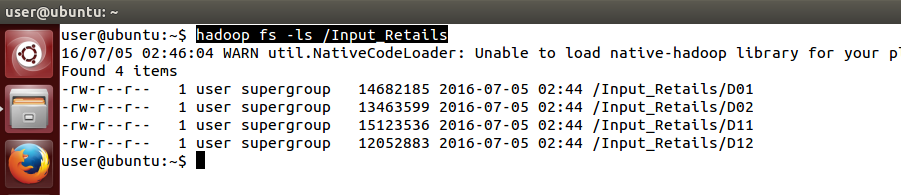
Then copy your input data into your HDFS which present into your **/home/user** Directory.

Command : **hadoop fs -put /home/user/Data/ /Input\_Retails**

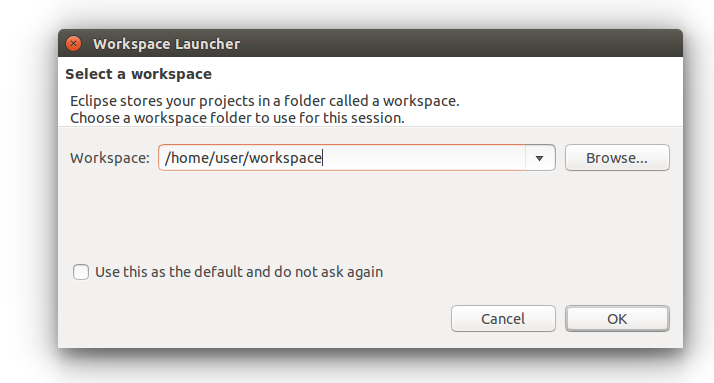


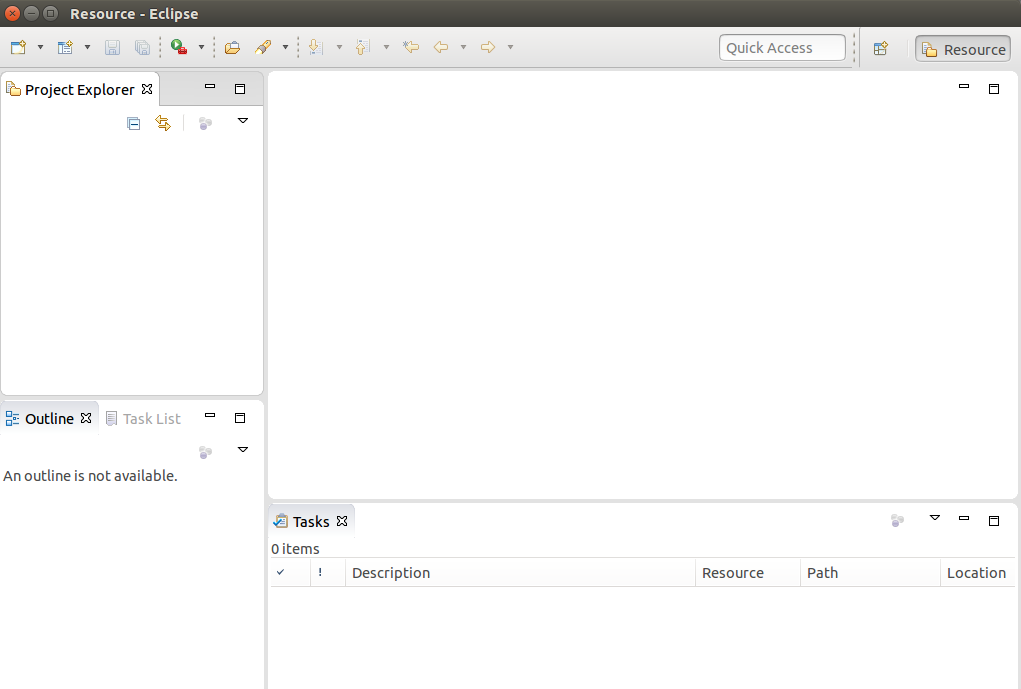
Make sure your data is copied

Command : hadoop fs -ls /Input\_Retails



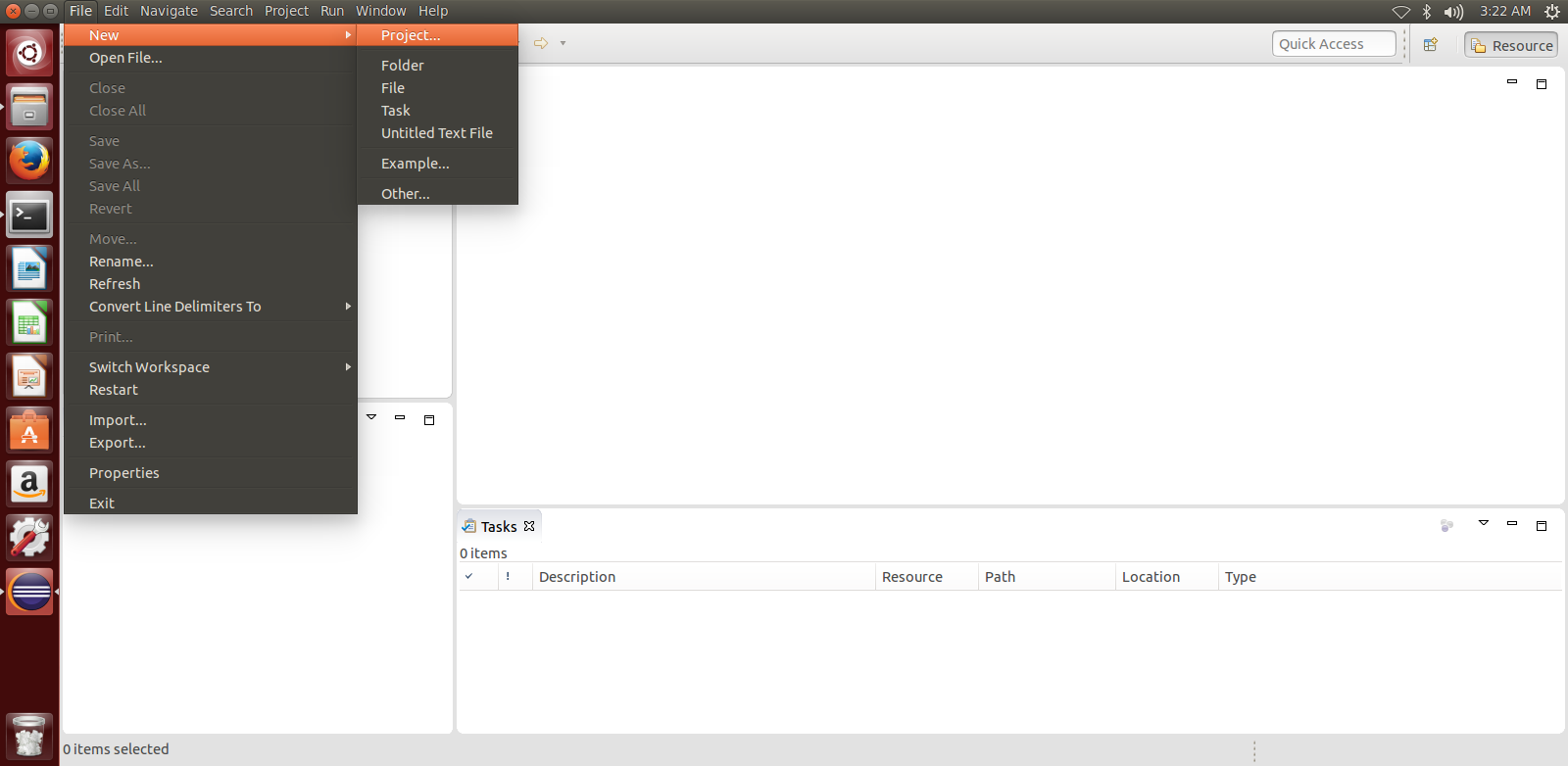
Then Open your Eclipse IDE for Programming and set your workspace for Eclipse.



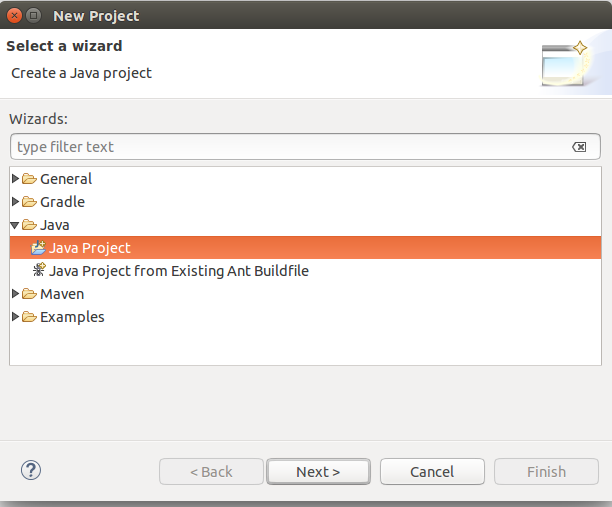


After Eclipse started Create a new java project for the 1st Use Case Solution Give a name Retails\_Project\_UseCase\_1 .

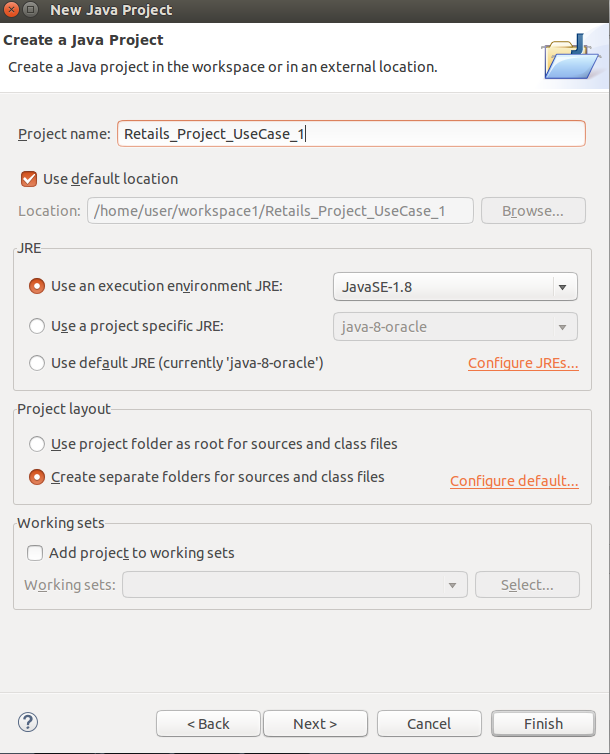
Click into **file -> Project....**



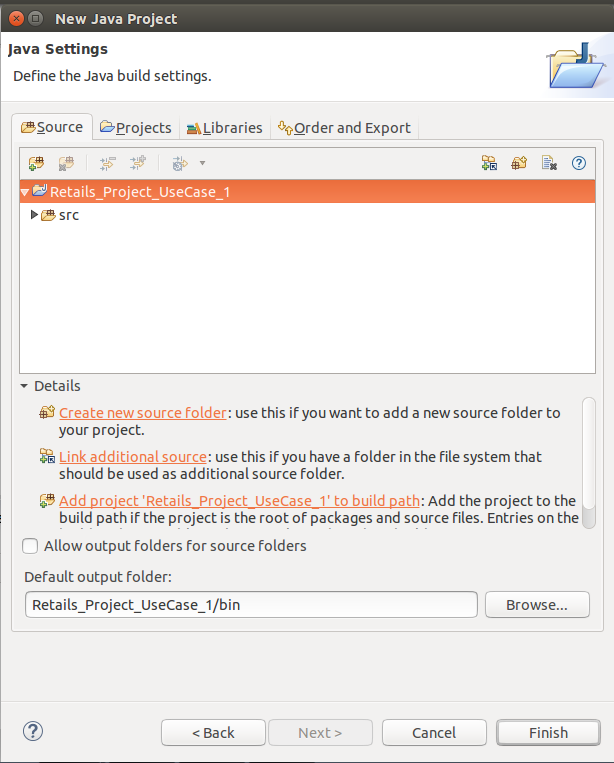
Select Java Project then click into **next**



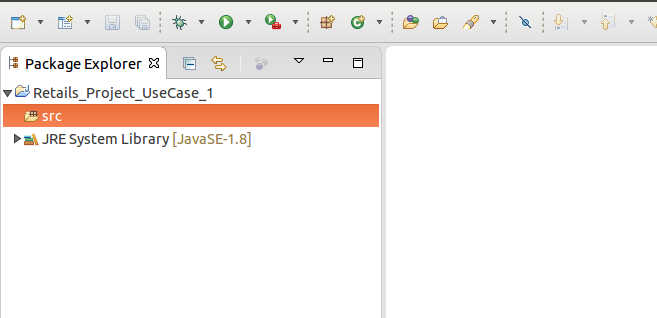
Provide the project name "Retails\_Project\_UseCase\_1" and then click on **Next**



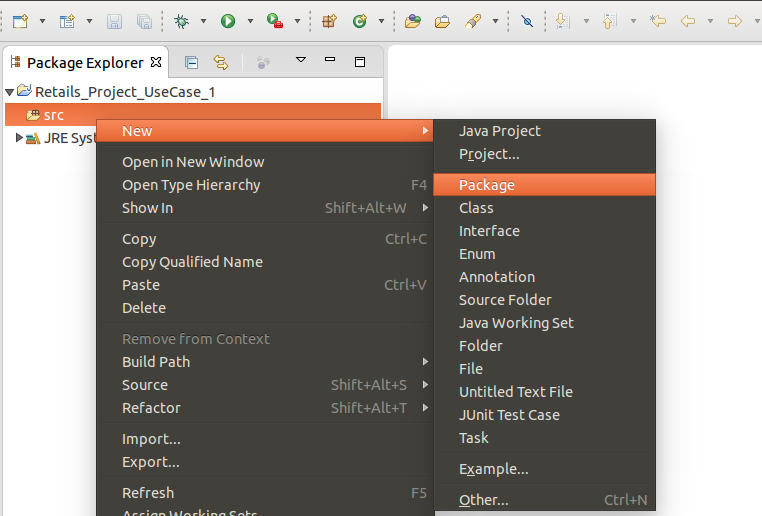
Click **Finish** Button to create a project



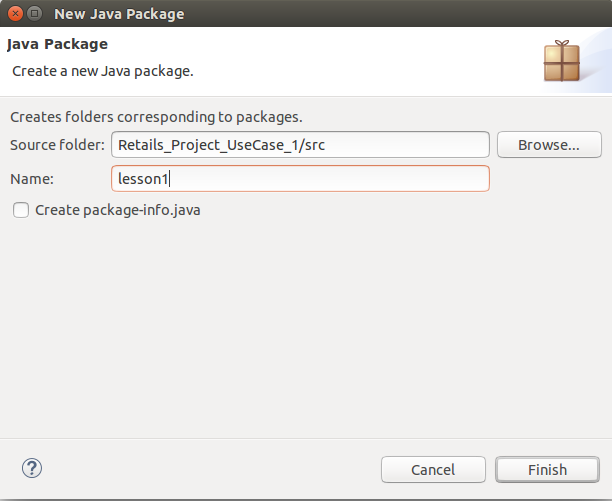
Below Project is create



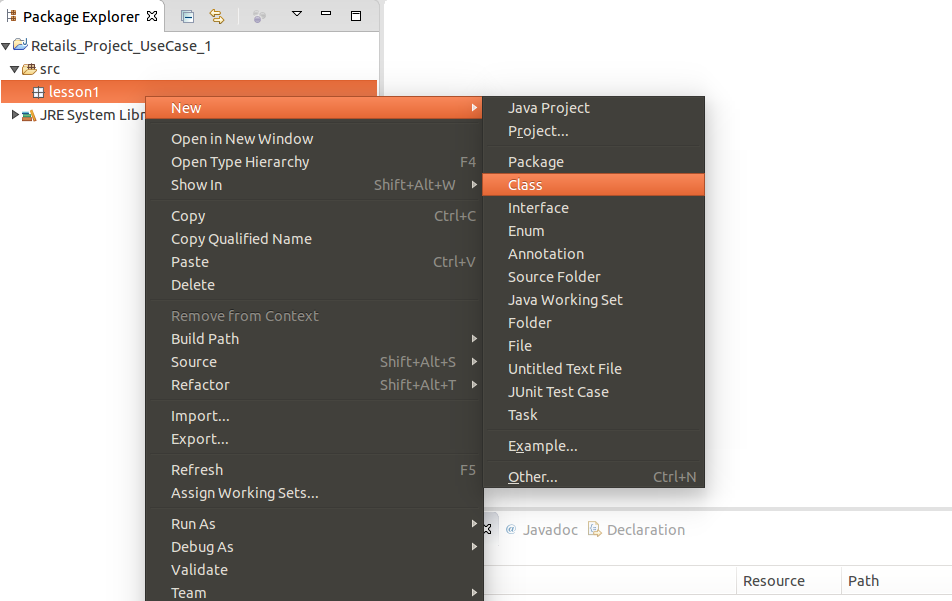
Create a New **Package** into src.



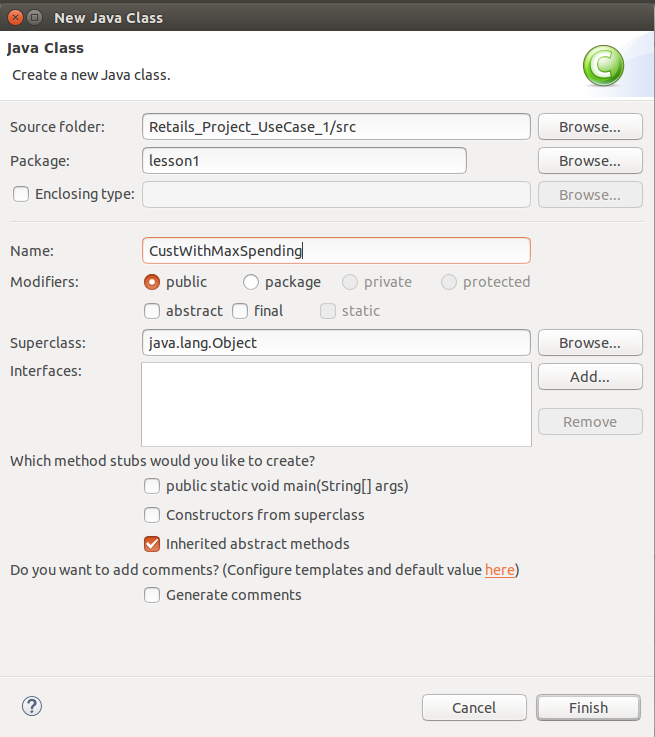
Give a Package name "**lession1**" and click to Finish **button**



Create a Driver Class into the lession1 Package.

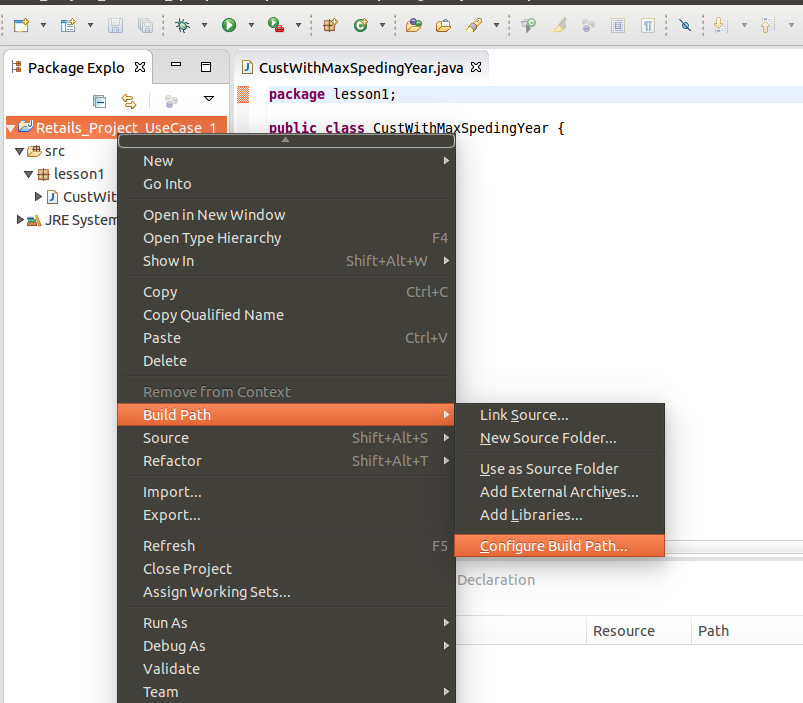


The Name of the Java Class is **CustWithMaxSpending** and Click to **Finish** button.

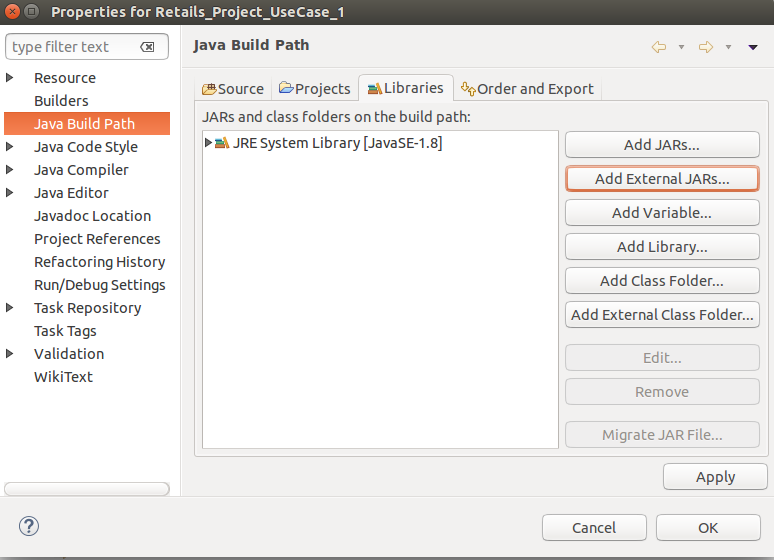


Then you have new CustWithmaxSpending.java file added into your project. So we need to add some jar file for the Framework function

Write click on project and go to **Build Path** add all necessary hadoop jar files into projects.



Click on Libraries and Add External JARs.. into Hadoop Directory.



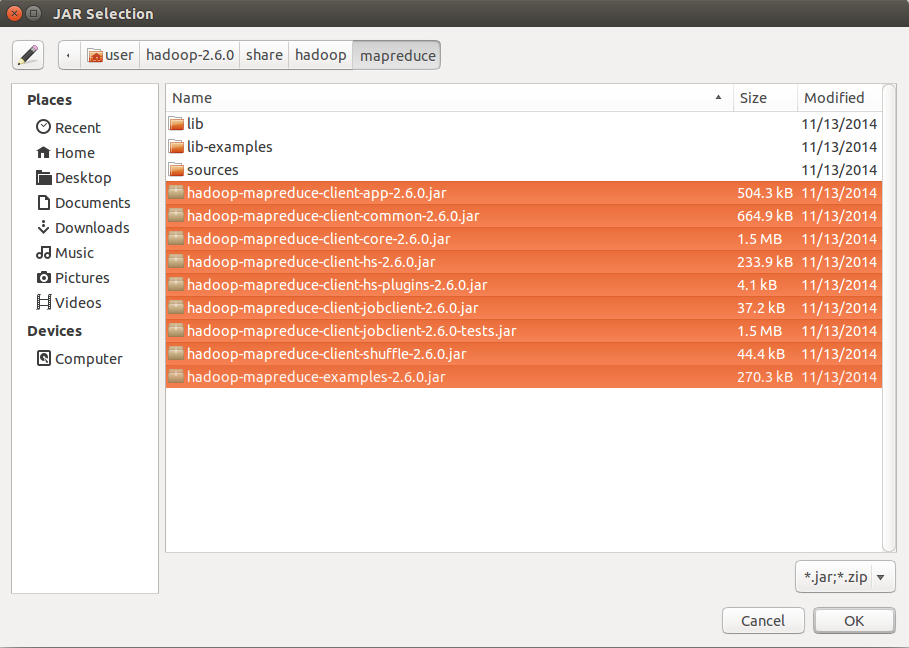
Add Jars from

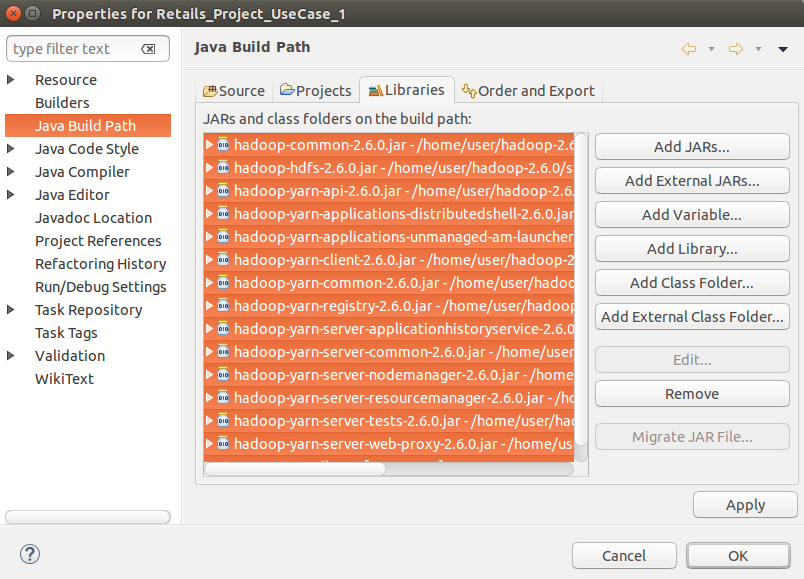
/homeuserhadoop-2.\*/share/hadoop/common/hadoop-common-2.6.0.jar

/homeuserhadoop-2.\*/share/hadoop/hdfs/hadoop-hdfs.2.6.0.jar

/homeuserhadoop-2.\*/share/hadoop/mapreduce/\*.jar

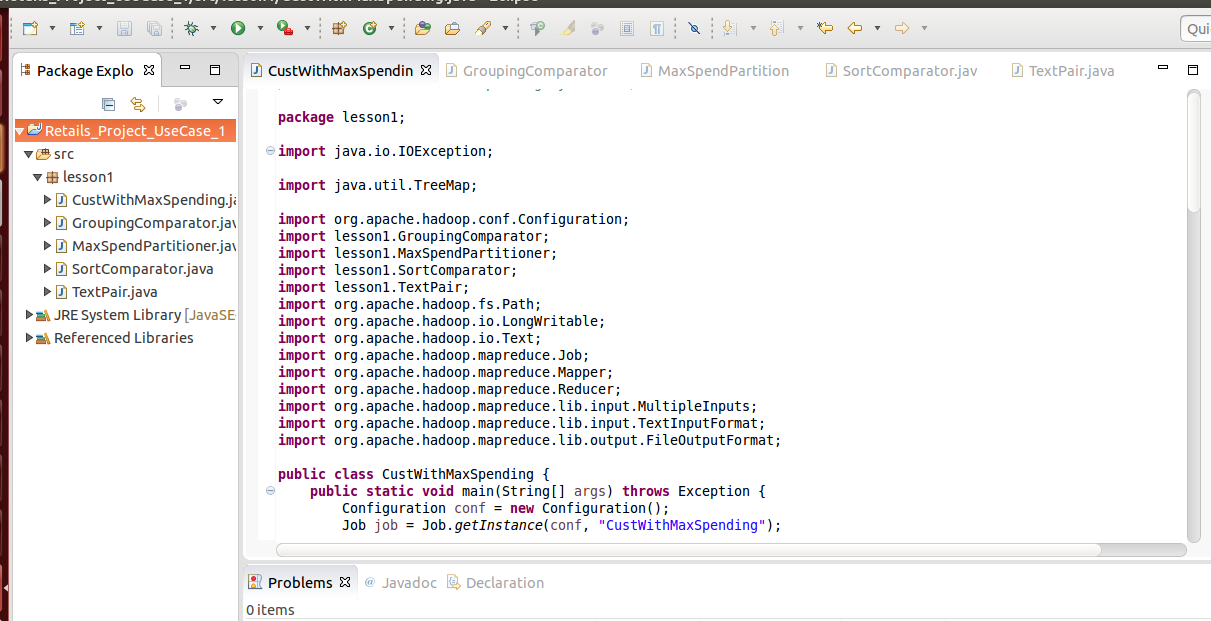
/homeuserhadoop-2.\*/share/hadoop/yarn/\*.jar





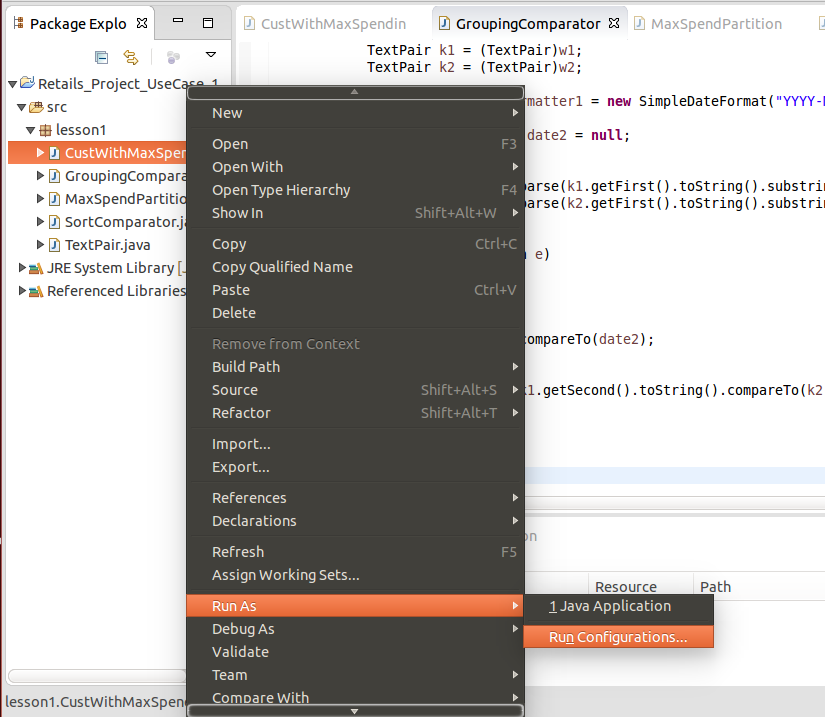
Write your Java code OR you can copy our java code into your package.

Add all java file into the package.



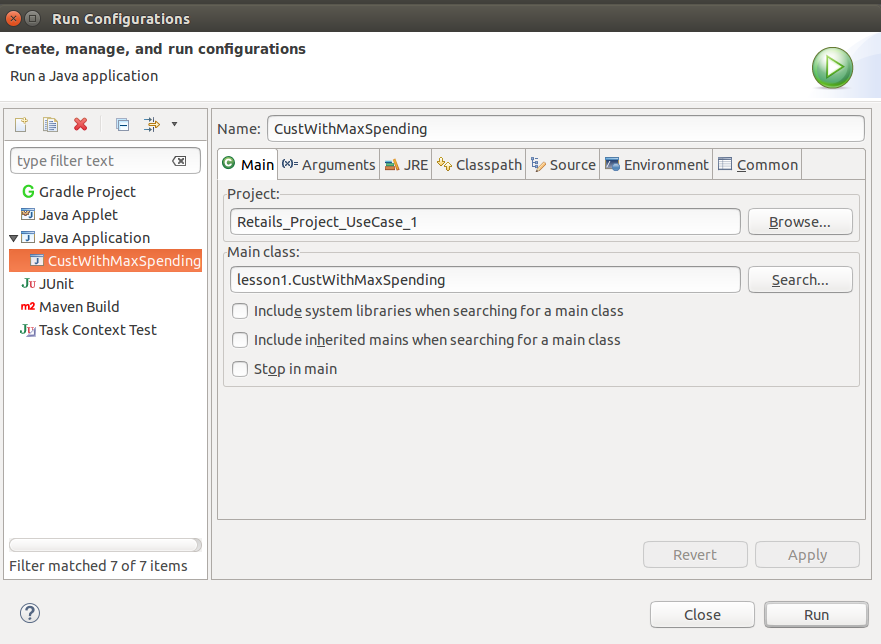
Then Run your Driver class for runable jar file.

Right Click to Driver Class -> Run As -> Run Configuration



Then 1st click to **Java Application** then Click to image show into the screen-shot to create CustwithMaxSpending.class file.

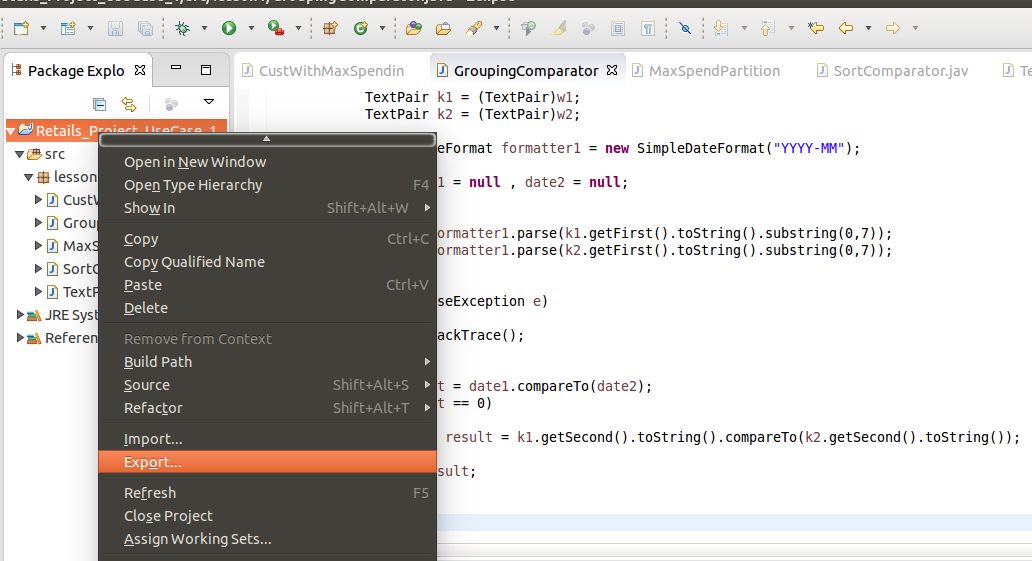
Then Run the Class file.

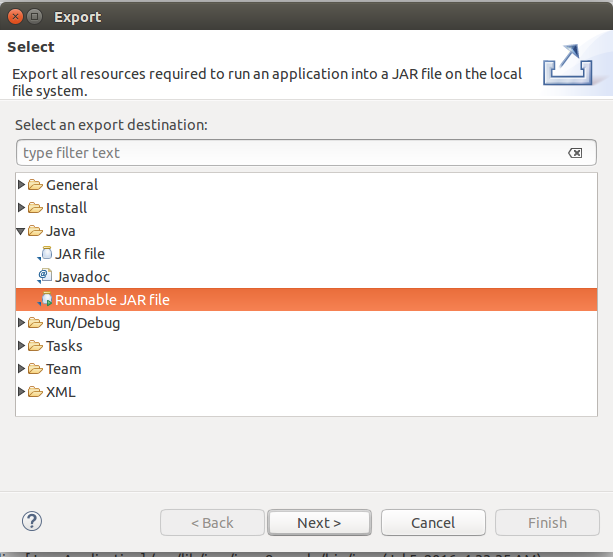


It will throw Exception Because of the Input and Output file. Please ignore this Exception.

Create a Runable jar file into your local directory for the execution of Project

Right Click On Project -> Click to Export

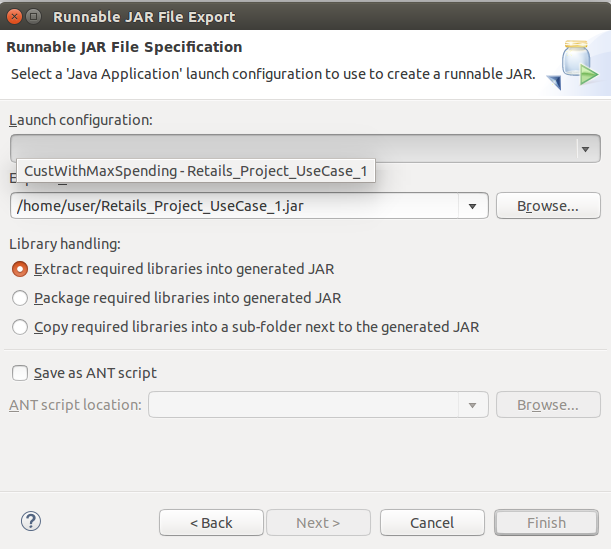




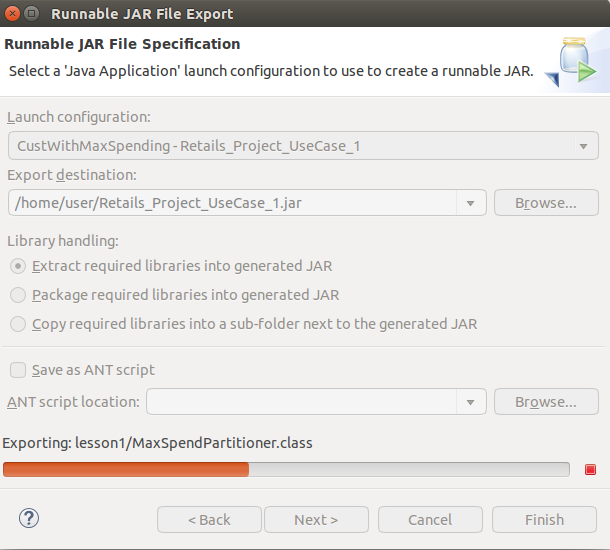
Browse your Files where you want to store this jar file and give a name Retails\_Project\_useCase\_1.jar

I am selected into /home/user/Retails\_Project\_useCase\_1.jar

And Select your class name into "**Lunch Configuration**"

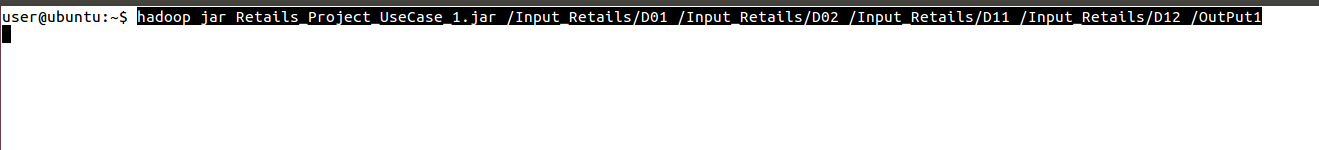


The Runable Jar file is generating.

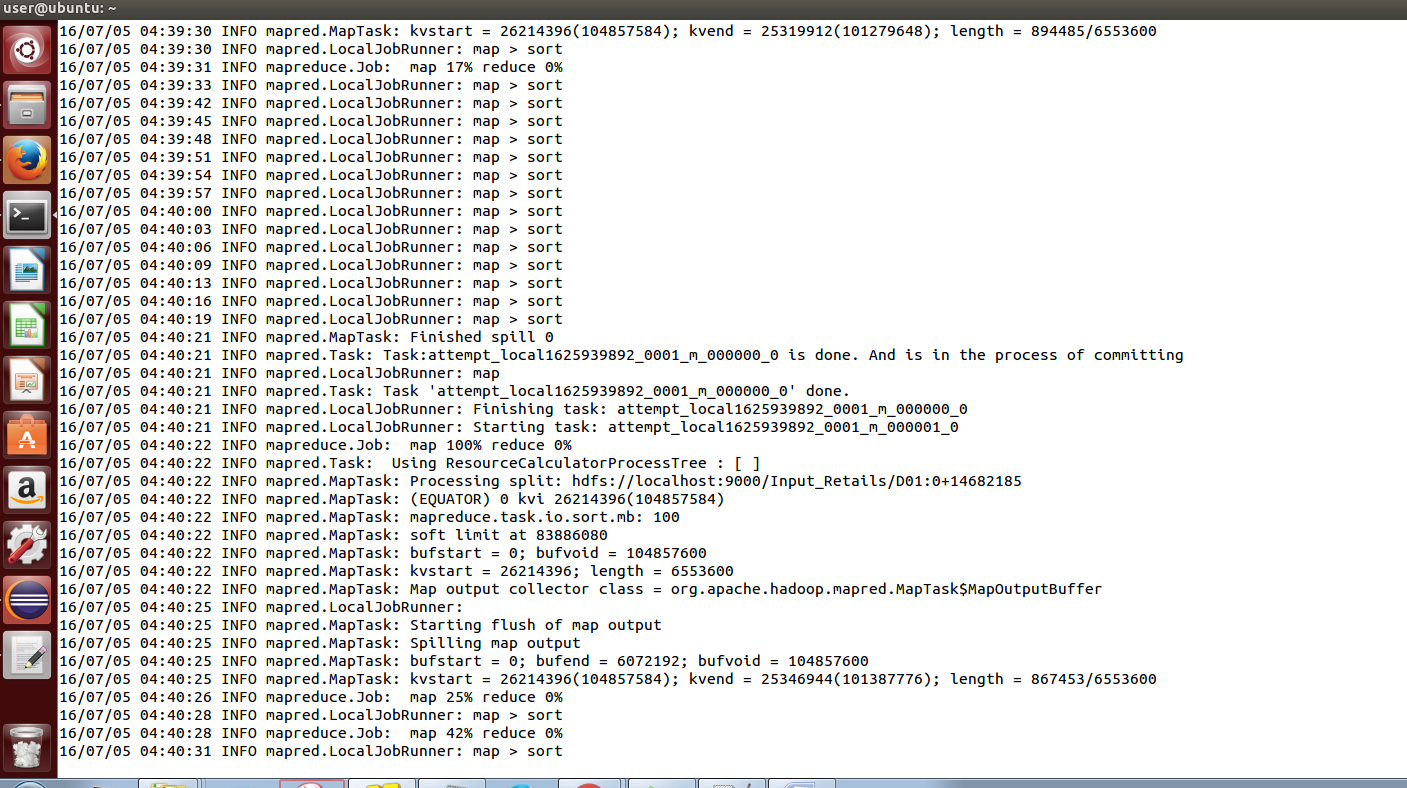


Then Run your Runable Jar file with input file and store your Output into HDFS file system "Output1"

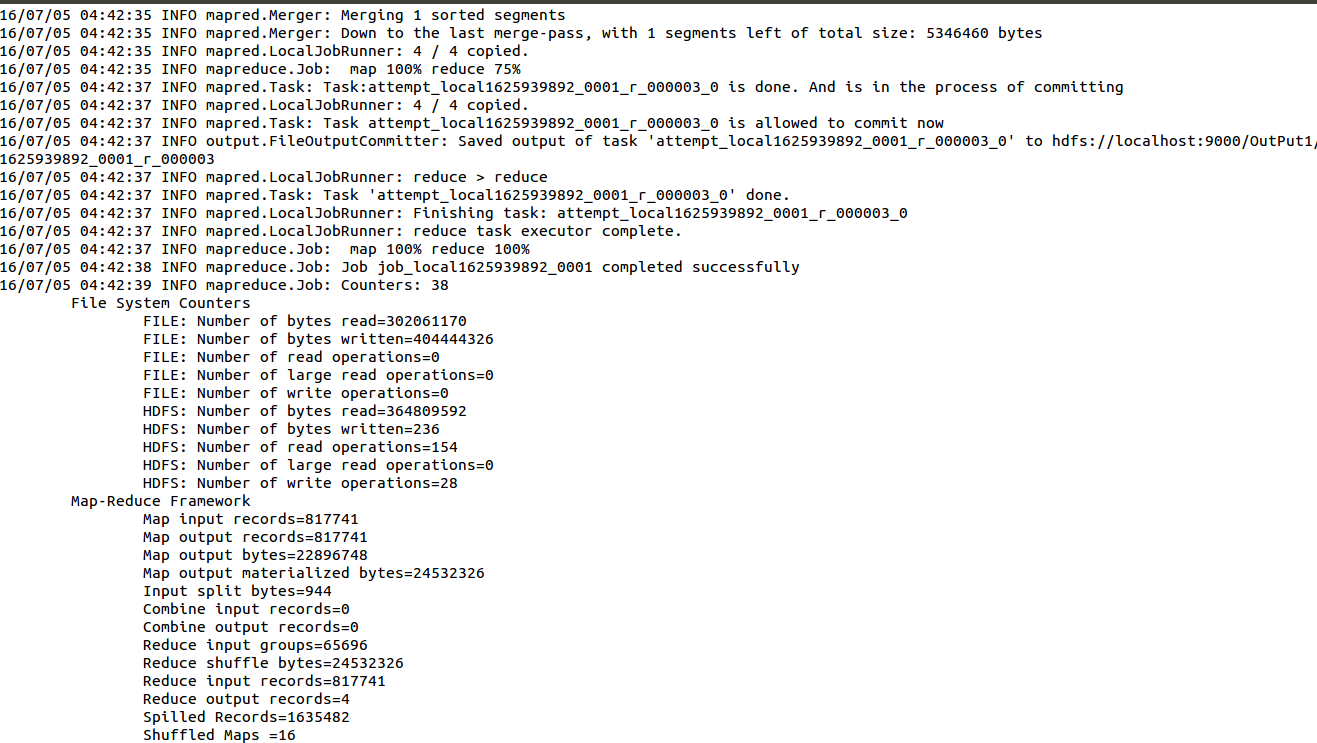
Command : **hadoop jar Retails\_Project\_UseCase\_1.jar /Input\_Retails/D01 /Input\_Retails/D02 /Input\_Retails/D11 /Input\_Retails/D12 /OutPut1**



Here The Job is created and the maper part is running

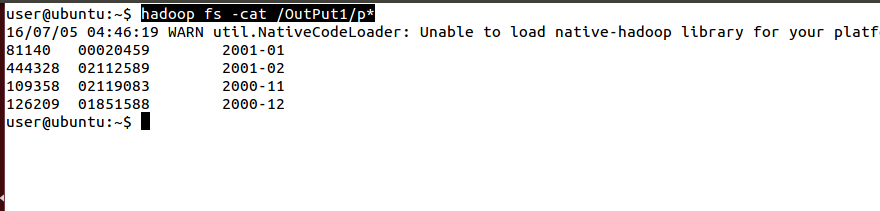


Here the Mapper Part and Reducer Part 100% Completed.



After completed the execution the output is created into HDFS "output1" Directory.

Command : **hadoop fs -cat /OutPut1/p\***



The Output is

**81140 00020459 2001-01**

**444328 02112589 2001-02**

**109358 02119083 2000-11**

**126209 01851588 2000-12**

Same steps you can follow for all UseCases. The Solution is in your Map-Reduce Directory.