# Big Data Hadoop Training

Session 16 Assignment 2 Solution:

**Write a hive UDF that implements functionality of string concat\_ws(string SEP, array<string>).**

**This UDF will accept two arguments, one string and one array of string.**

**It will return a single string where all the elements of the array are separated by the SEP.**

1. **Steps involved in writing a Hive UDF:**
2. Create Java Class for a User Defined Function which extends org.apache.hadoop.hive.sq.exec.UDF and implements evaluate() method**.**

**Java code:**

import org.apache.hadoop.hive.ql.exec.UDF;

import org.apache.hadoop.io.Text;

public class hive\_concat\_with\_sep extends UDF {

private Text result = new Text();

public Text evaluate(String sep, String[] array\_str) {

try {

result.set(array\_str[0]);

for (int i=1 ; i < array\_str.length; i++) {

result.set(result.toString()+sep+array\_str[i]);

}

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("FAILED: SemanticException : Arguments length mismatch");

System.out.println("This HIVE UDF that implements functionality of string concat\_ws(separator,[string | array(string)]+) needs at least two arguments");

}

return result;

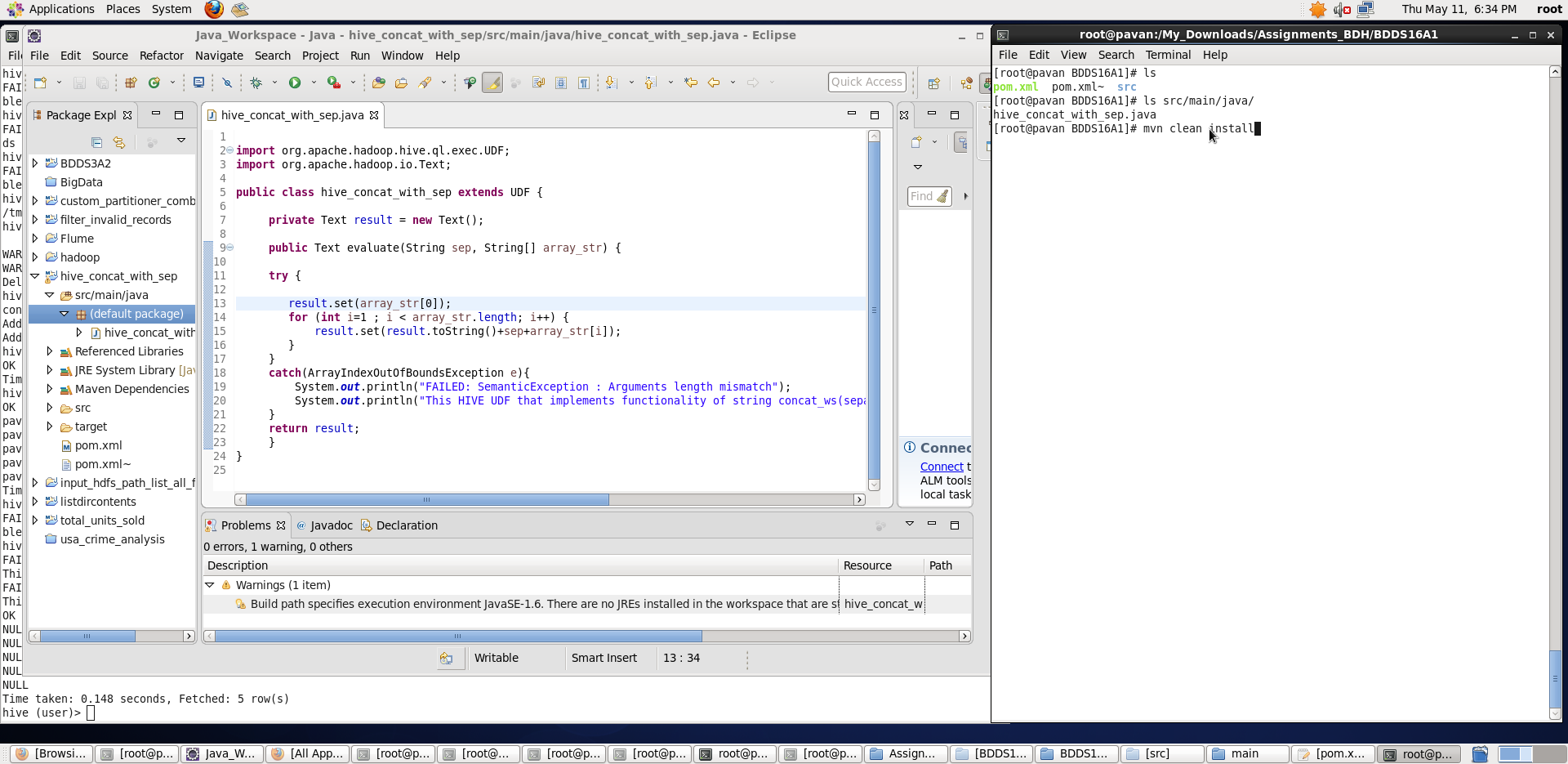
}

}

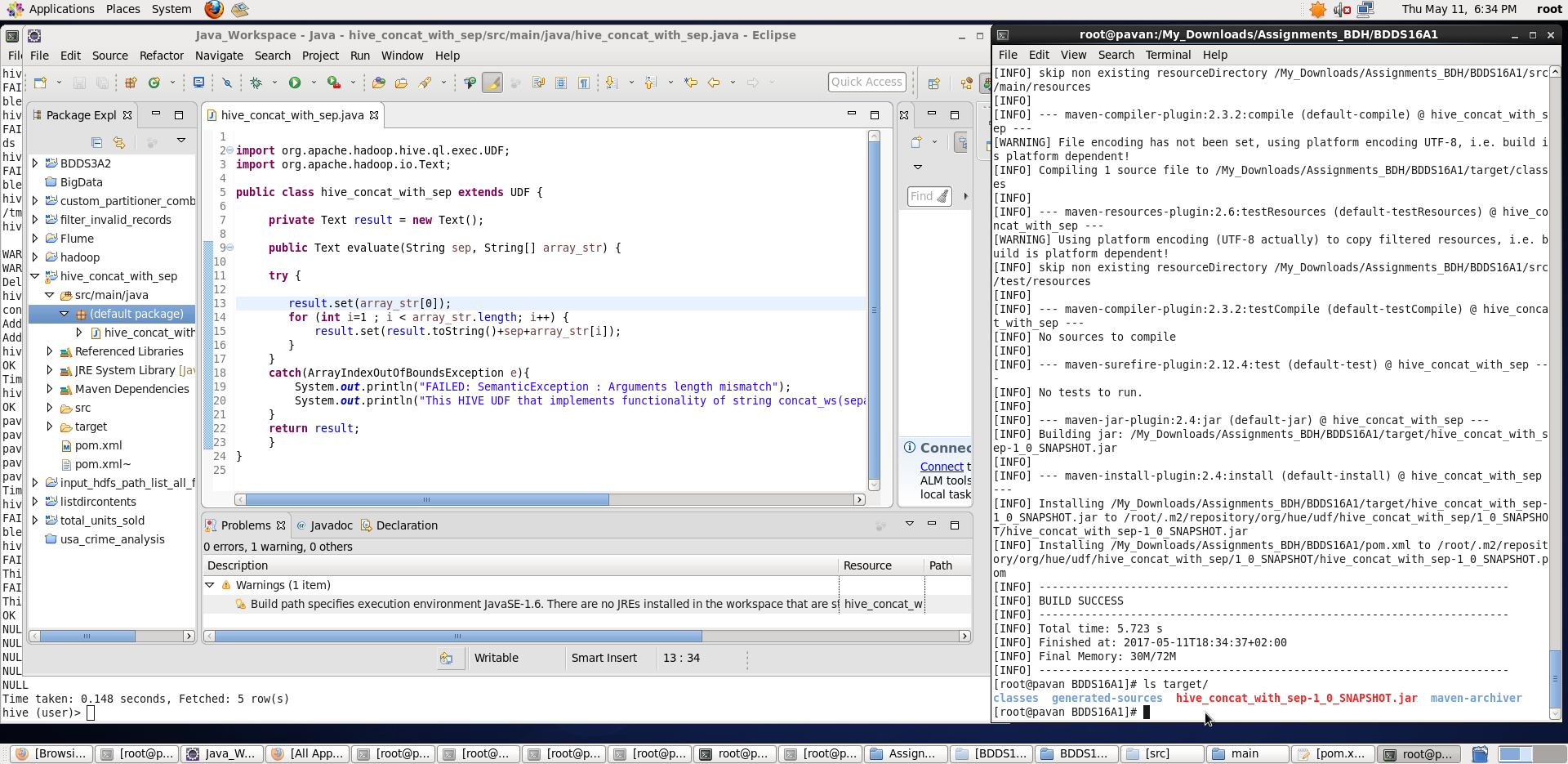
1. Packaging the Java class into a JAR file (I am using Maven)

Make the folder structure BDDS16A2/src/main/java/hive\_concat\_with\_sep.java. Place the pom.xml file in BDDS16A2/ and run the maven build command

**# mvn clean install**

****

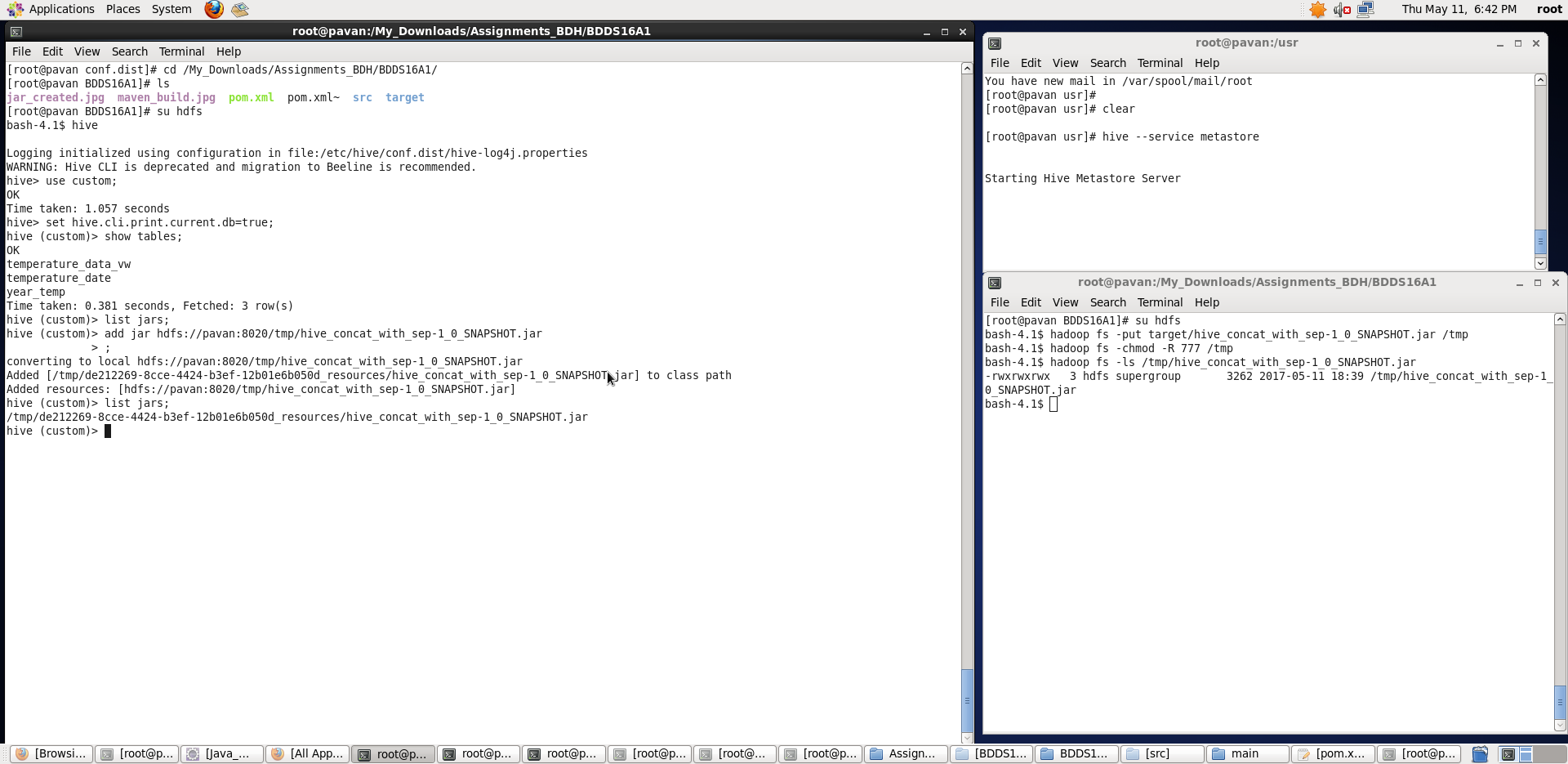
Once, the build is successful, we can see the **hive\_concat\_with\_sep-1.0\_SNAPSHOT.jar** being created inside the target directory. Place it in HDFS location : hdfs://pavan:8020/tmp

****

1. **Go to the Hive CLI and Add the UDF JAR :**

**hive> add jar hdfs://pavan:8020/tmp/hive\_concat\_with\_sep-1.0\_SNAPSHOT.jar**

We can see the jar being added. **hive> list jars**

****

Now, we will test the UDF on a table – temperature\_date loaded with some data as shown below:

1. **Create Temporary Function**

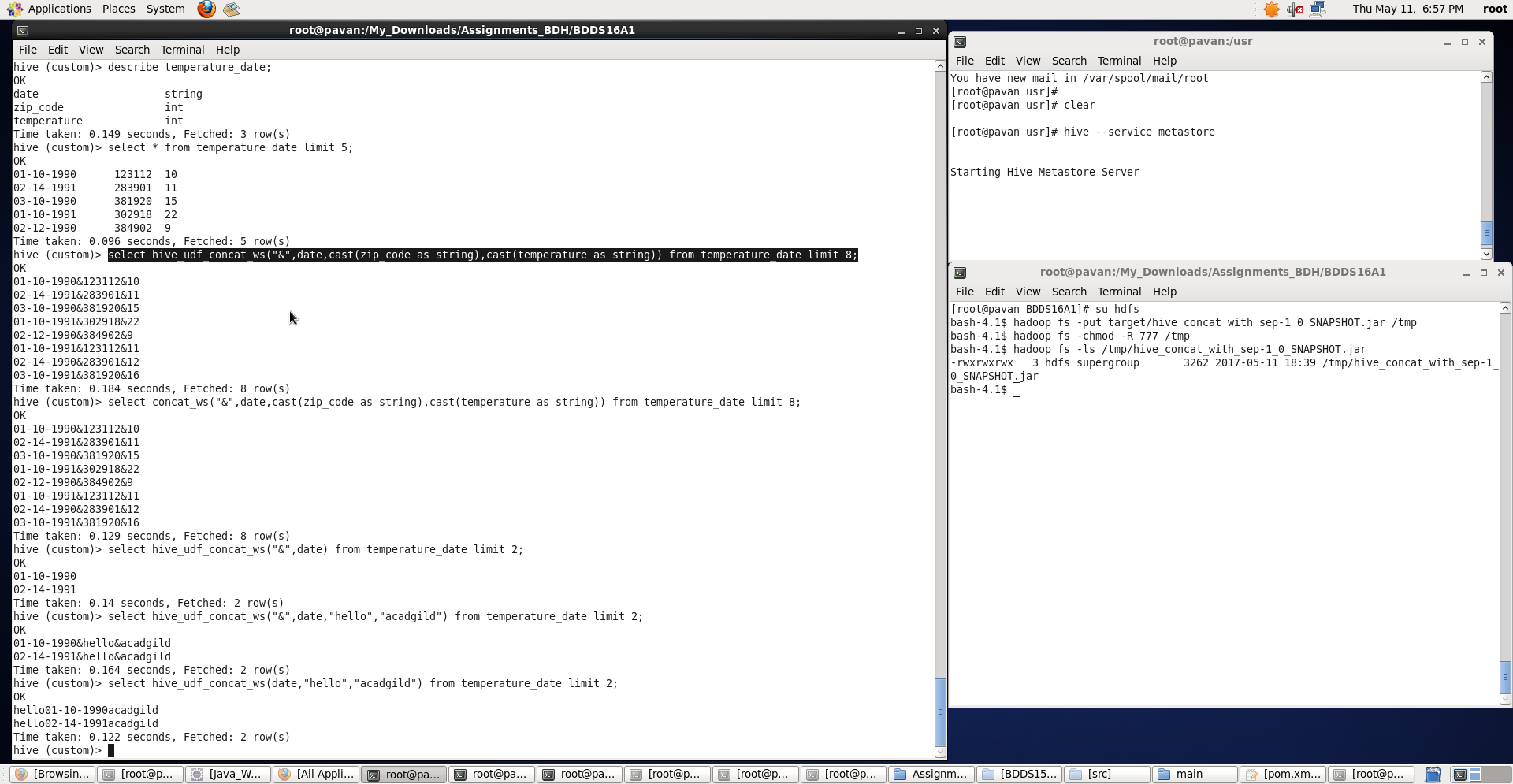
It does not have to be a temporary function. It can create be our own function as follows:

**hive> CREATE TEMPORARY FUNCTION hive\_udf\_concat\_ws AS 'hive\_concat\_with\_sep';**

Here **class\_name :** hive\_concat\_with\_sep , **hive\_udf\_func** : hive\_udf\_concat\_ws

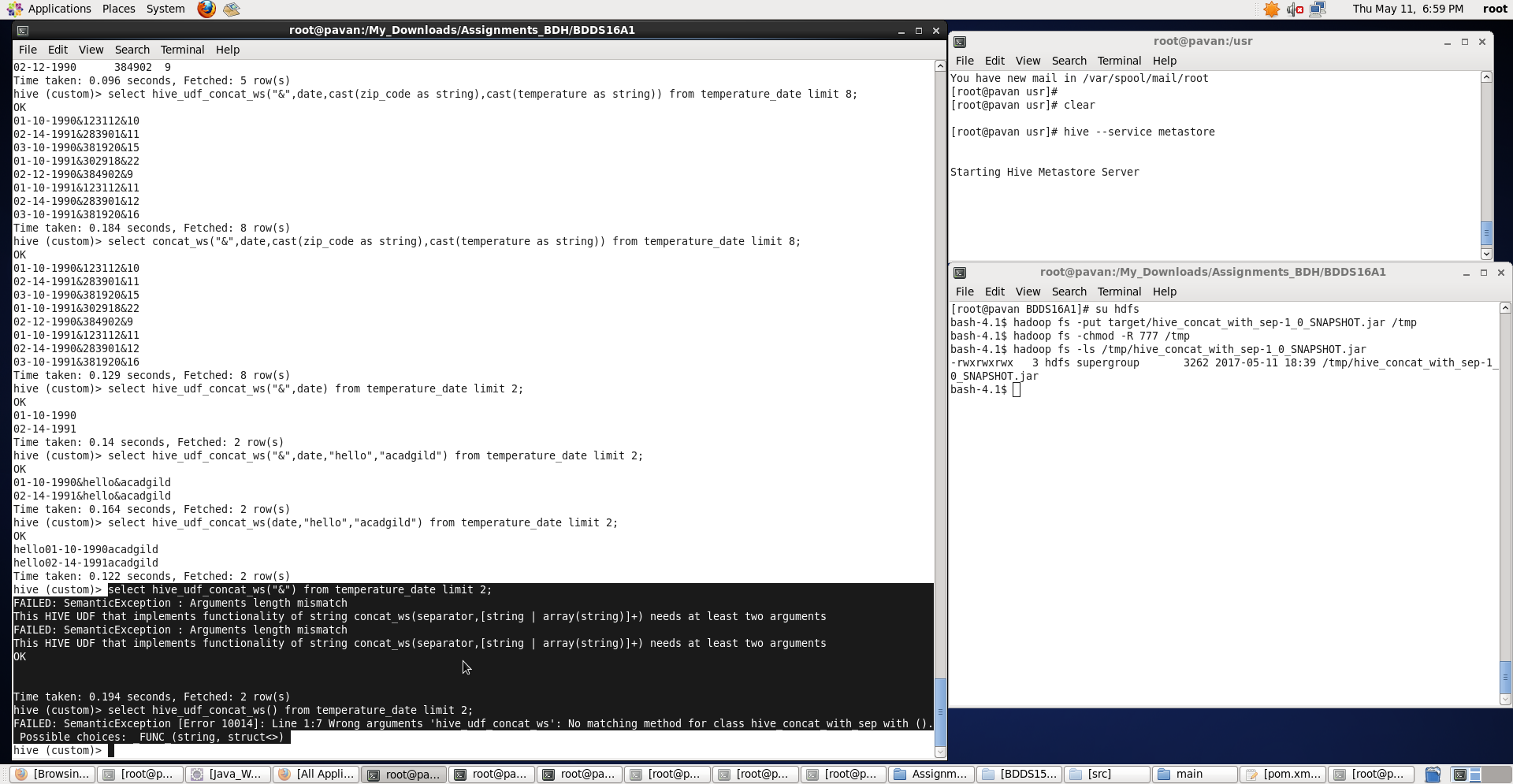
1. **Use UDF in Hive as: select <Hive\_UDF\_Func(<Arguments>)> from temperature\_date;**

**hive> select hive\_udf\_concat\_ws(“&”,date, cast(zip\_code as string), cast(temperature as string)) from temperature\_date limit 4;**

****

We can see the fields inside table being concatenated with “&” separator. Also, observe that the Hive UDF’s output is same as Hive in-built **concat\_ws** function.

**Note:** I have also handled some error scenarios wherein all the necessary arguments are not provided.

****

Thus, a hive UDF that implements functionality of string concat\_ws(string SEP, array<string>) has been written and example shown wherein the UDF accepted two arguments, one string and one array of string and returned a single string where all the elements of the array are separated by the SEP.