# Big Data Hadoop Training

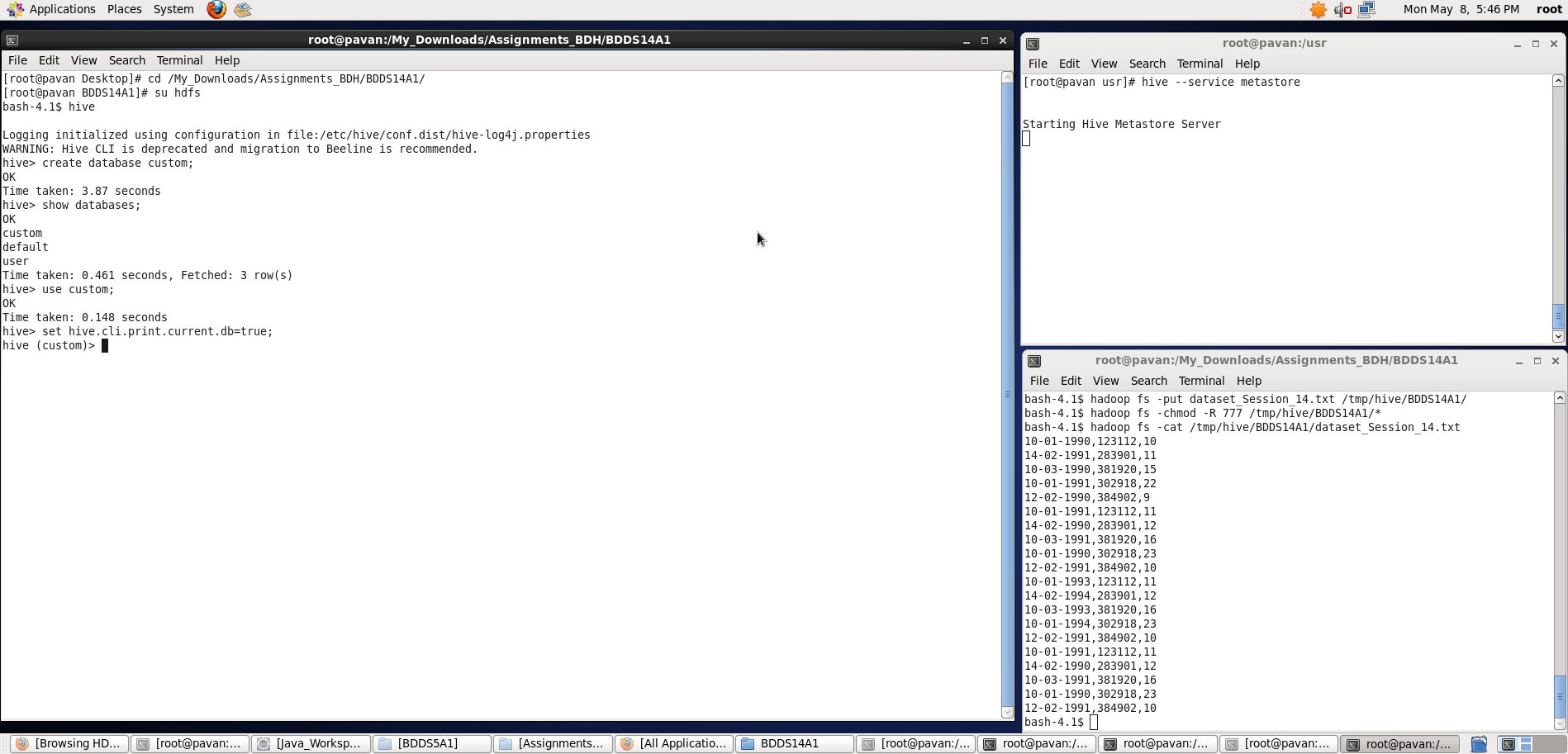
Session 14 Assignment 1 Solution

1. **Create a database named 'custom'.**
2. Start hive as hdfs user as shown below, Also start **hive metastore service.**

**hive> create database custom;**  -- creating ‘custom’ database

For using the created database :

**hive> use custom;**

****

**2. Create a table named temperature\_data inside custom having below fields:**

**1. date (mm-dd-yyyy) format**

**2. zip code**

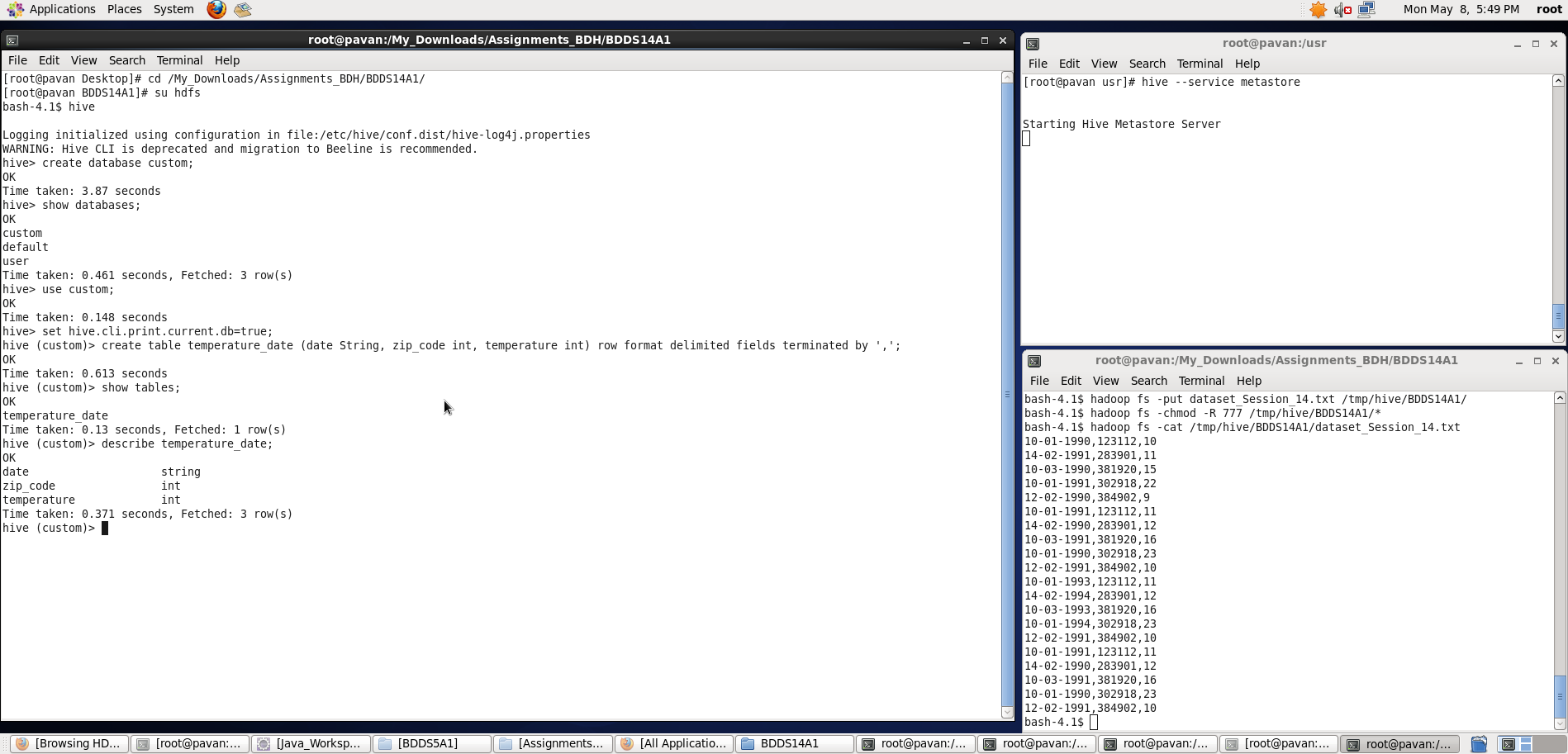
**3. temperature**

**The table will be loaded from comma-delimited file.**

1. For creating a table with given fields

**hive> create table temperature\_date (date String, zip\_code int, temperature int) row format delimited fields terminated by ‘,’;**

We can see that a table being created. Use **show and describe** commands to see whether table with specified fields got created or not.

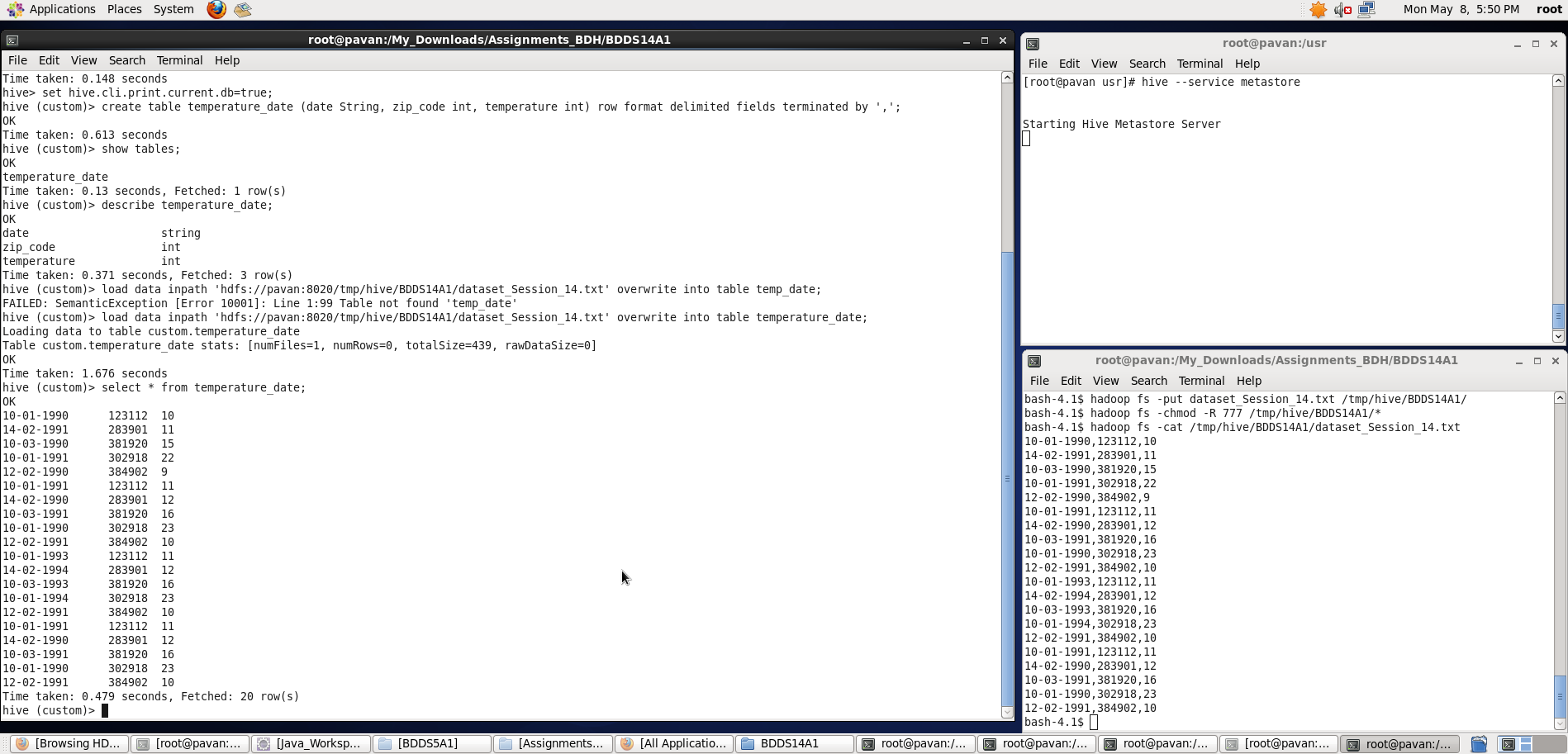
****

**3. Load the dataset.txt (which is ',' delimited) in the table.**

1. For loading the dataset :

**hive> load data inpath ‘hdfs://pavan:8020/tmp/hive/BDDS14A1/dataset\_Session\_14.txt’ overwrite into table temperature\_date;**

We can see that the dataset in specified hdfs location got loaded into the table as shown in the screenshot.

****

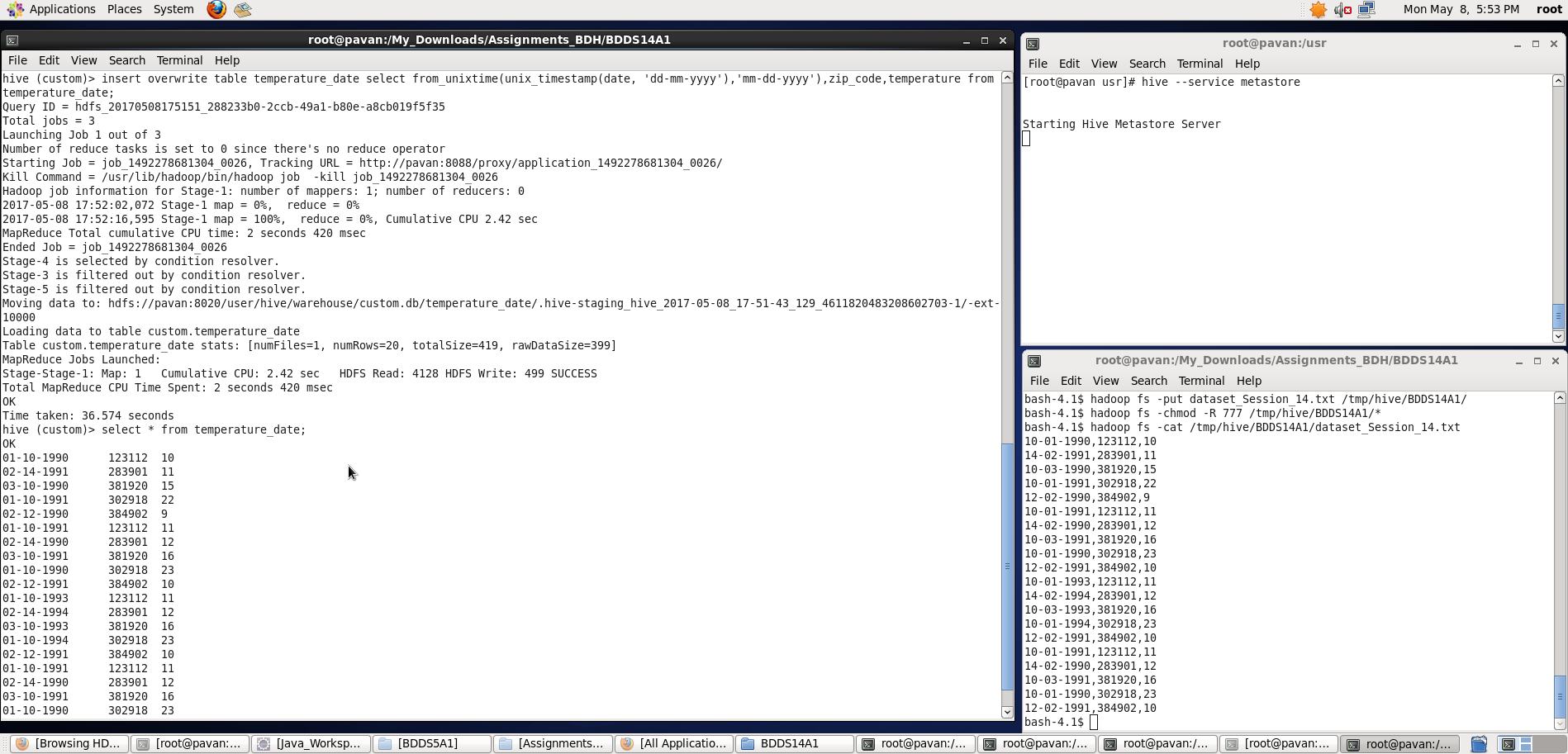
But we want the date field to be loaded in (mm-dd-yyyy) format, so we need to overwrite the data by doing some transformations as below:

**hive> insert overwrite table temperature\_date select from\_unixtime(unix\_timestamp(date, ‘dd-mm-yyyy’),’mm-dd-yyyy’),zip\_code,temperature from temperature\_date;**

**Here, unix\_timestamp :** converts time string with given pattern to Unix time stamp (in seconds)

**from\_unixtime** : converts the number of seconds from unix epoch (1970-01-01 00:00:00 UTC) to a specified string pattern.

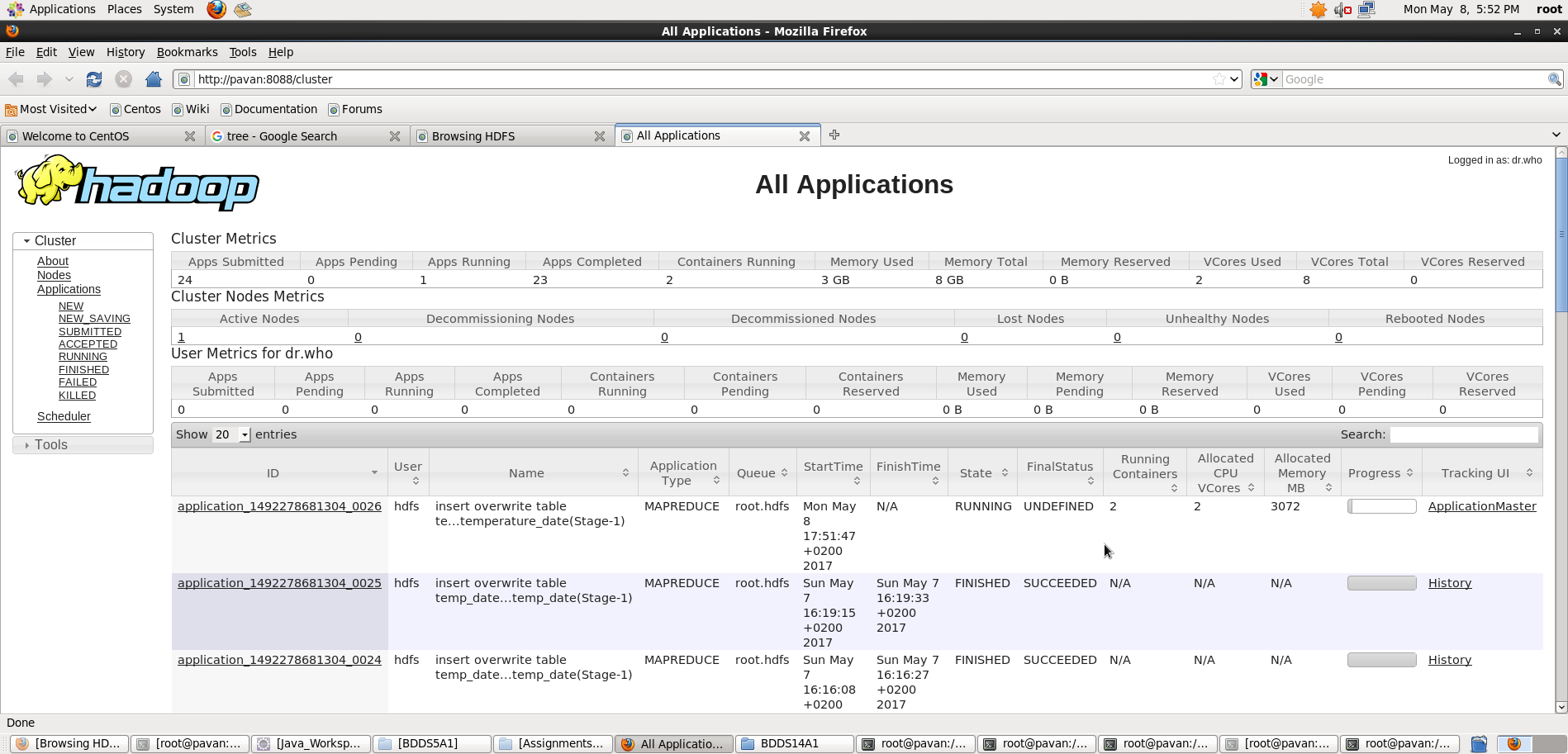
**Note:** For directly storing the date field as datatype:Date – input should be **yyyy-mm-dd** – Otherwise NULL will be stored. For this reason, initially, I have stored the date field as a string and converted it into required format later**.**



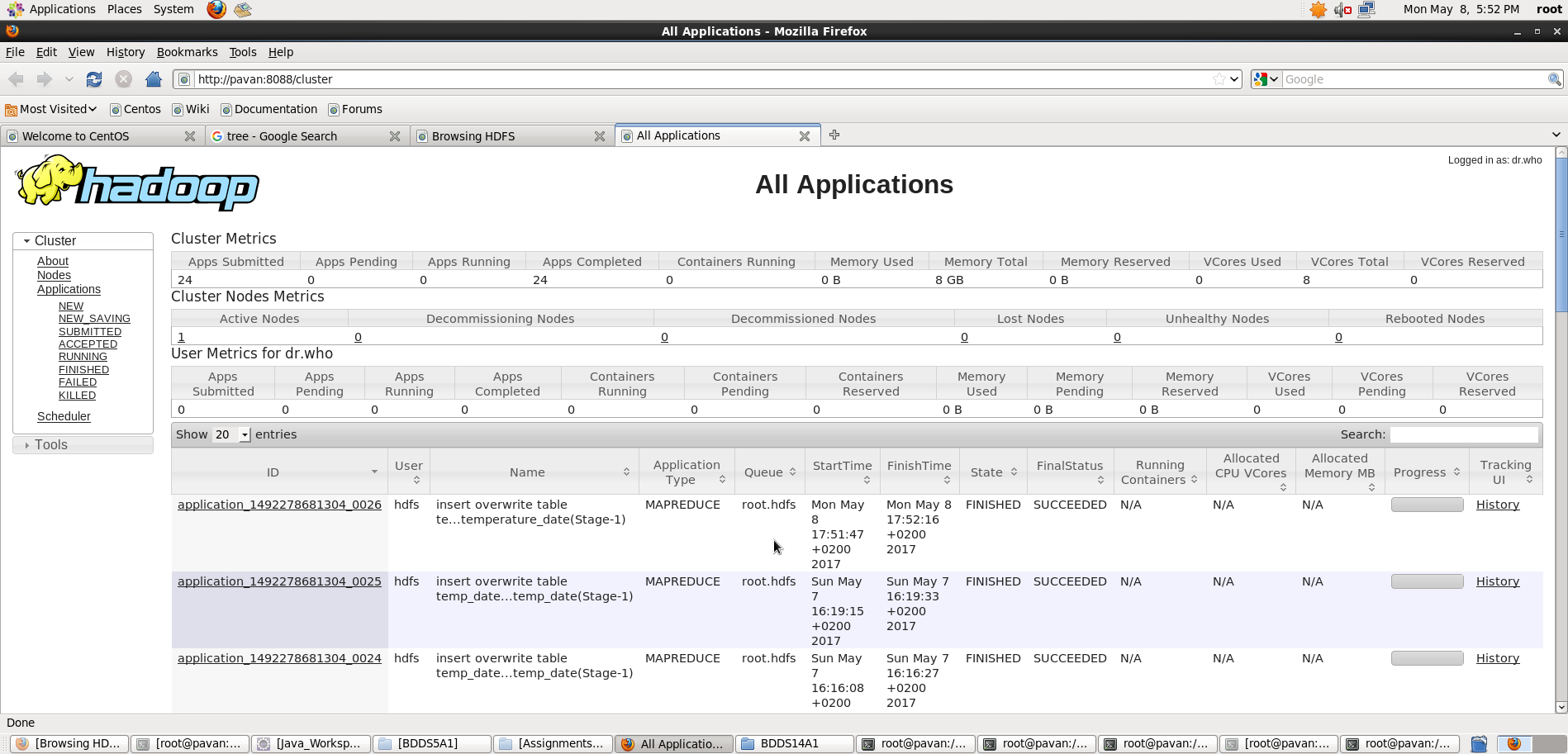
Now, we can see the data being rightly stored inside the table.

**We can observe the MapReduce application status from Browser while we overwrite the table contents using insert command:**

**Running status (top one):**

****

**Finished status (top one):**

****

Thus, we have created a table inside ‘custom’ database and loaded the dataset in specified field format.