# Big Data Hadoop Training

Session 8 Assignment 2 Solution:

Task

**Create a sample dataset and implement the below Pig commands on the same dataset.**

**Sample dataset (used) –**

/My\_Downloads/Assignments\_BDH/BDDS8A2/student\_details.txt

/My\_Downloads/Assignments\_BDH/BDDS8A2/duplicate\_student\_details.txt



**Pig Commands:**

1. Concat

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

concat\_output = FOREACH student\_details GENERATE CONCAT (firstName,'\*',lastName);

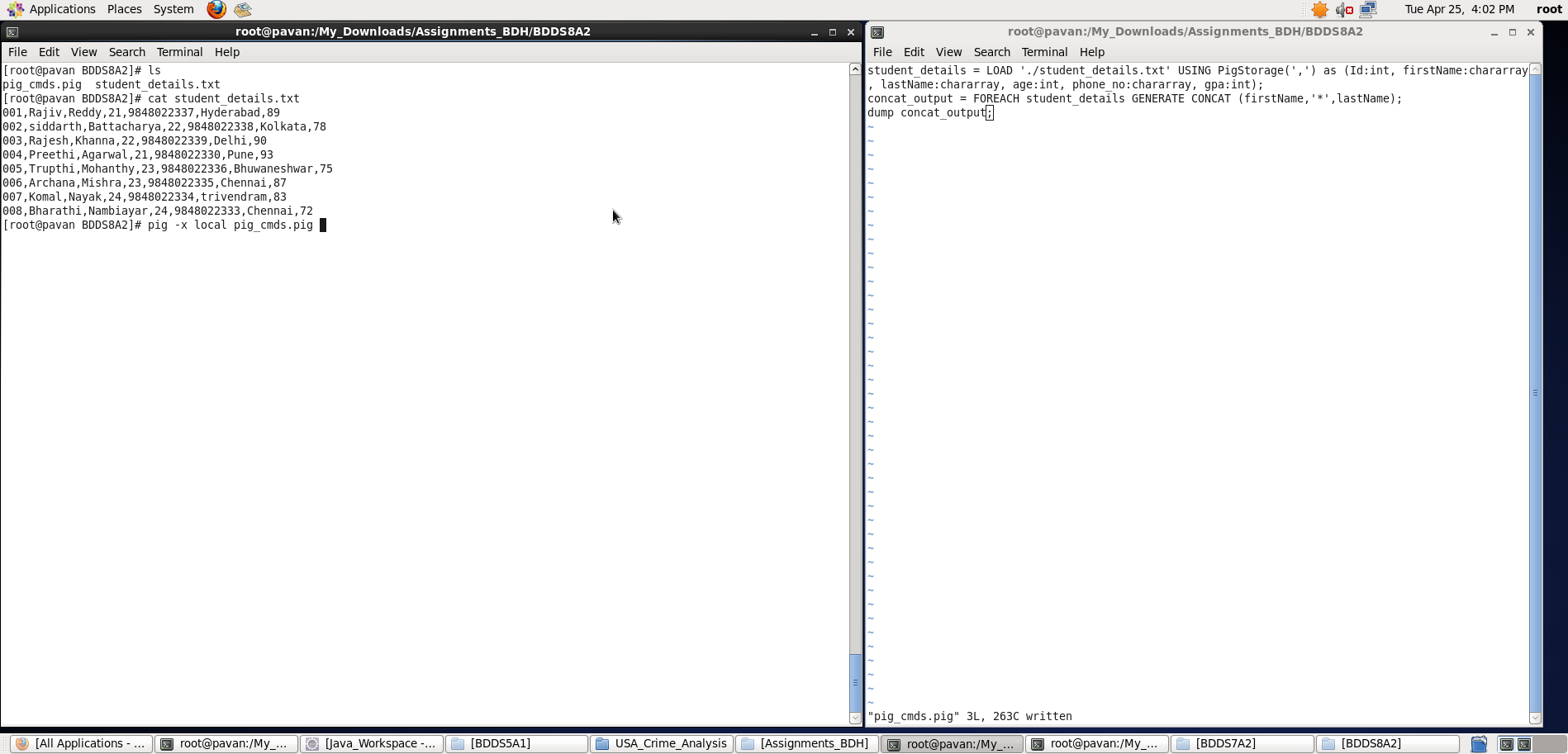
dump concat\_output;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, for all records I have generated a new relation which contains concatenated result of the firstName and lastName using CONCAT function. Here ‘\*’ is kept in between the firstName and lastName and all the three expressions are concatenated.
* Outputting the concat\_output

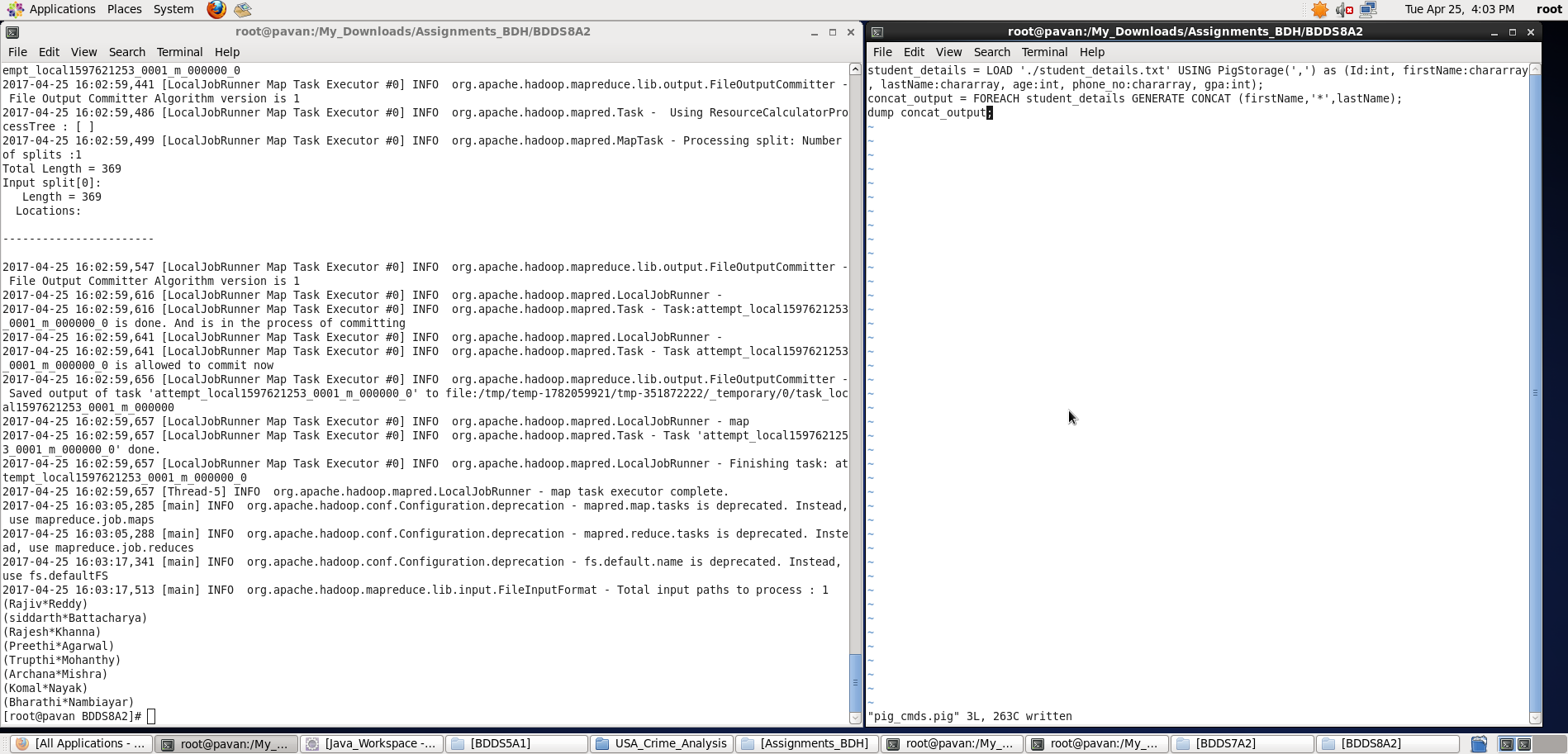
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory

****

Output displayed: **<concat\_output>**

**Note : We can store the output if required in a file using “store” command.**

****

1. Tokenize

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

concat\_output = FOREACH student\_details GENERATE CONCAT (firstName,'\*',lastName);

tokenize\_output = FOREACH concat\_output GENERATE TOKENIZE ($0);

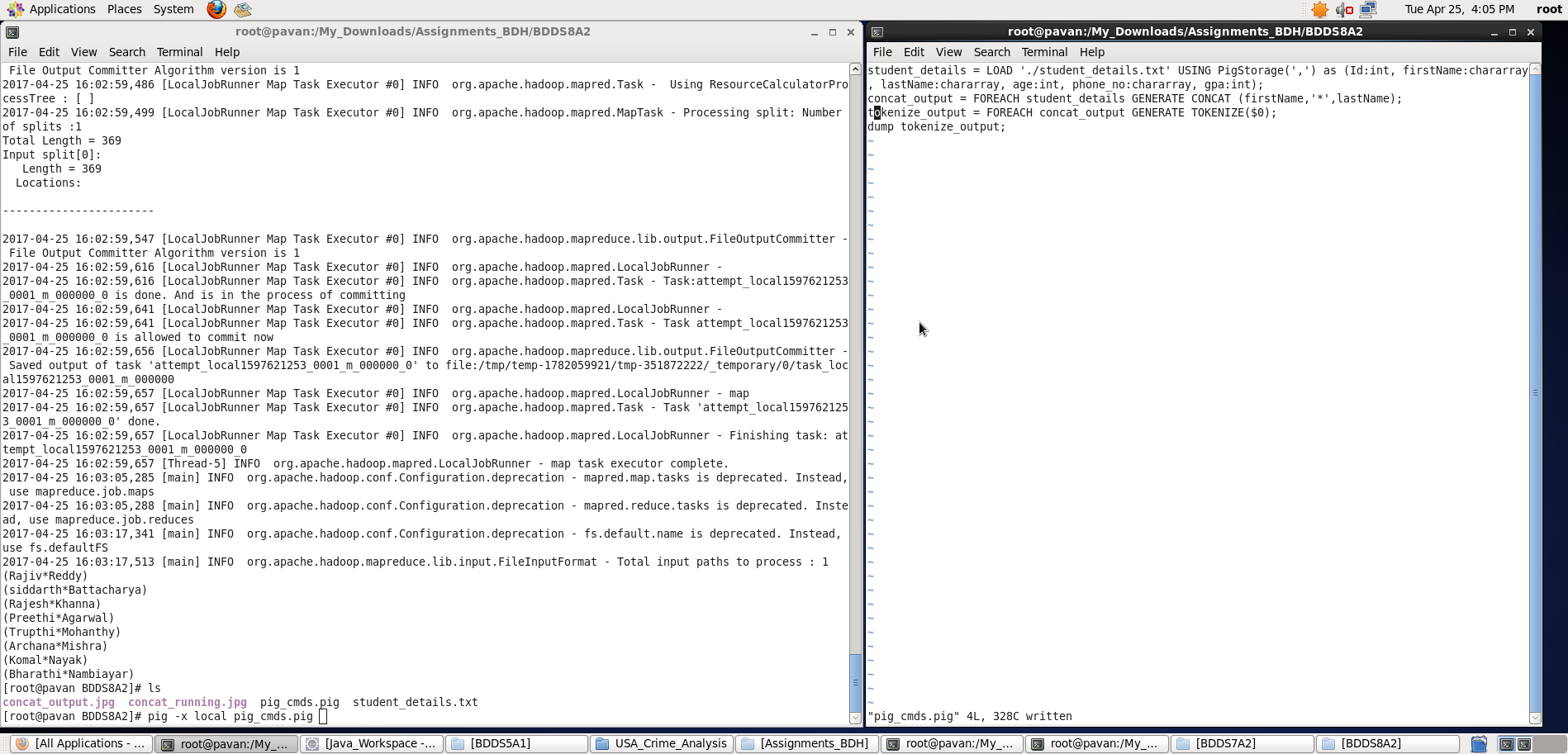
dump tokenize\_output;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, for all records I have generated a new relation which contains concatenated result of the firstName and lastName using CONCAT function. Here ‘\*’ is kept in between the firstName and lastName and all the three expressions are concatenated.
* Now, using TOKENIZE function , split the concat\_output (which contains a group of words) in a single tuple and returns a bag which contains the output of the split operation.
* Outputting the tokenize\_output

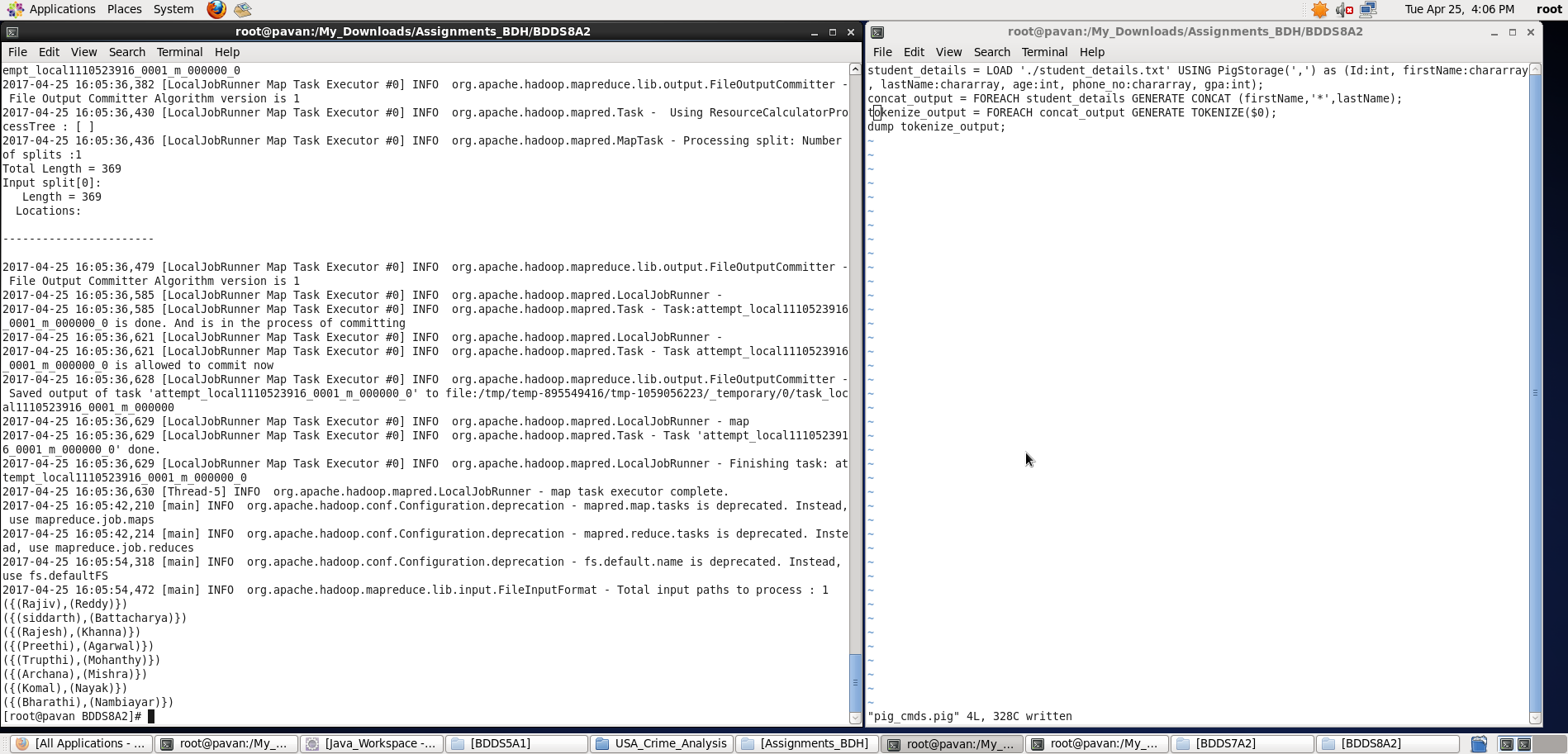
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<tokenize\_output>**

**Note : We can store the output if required in a file using “store” command.**



1. Sum

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

grpallstudents = GROUP student\_details all;

age\_total = FOREACH grpallstudents GENERATE SUM(student\_details.age);

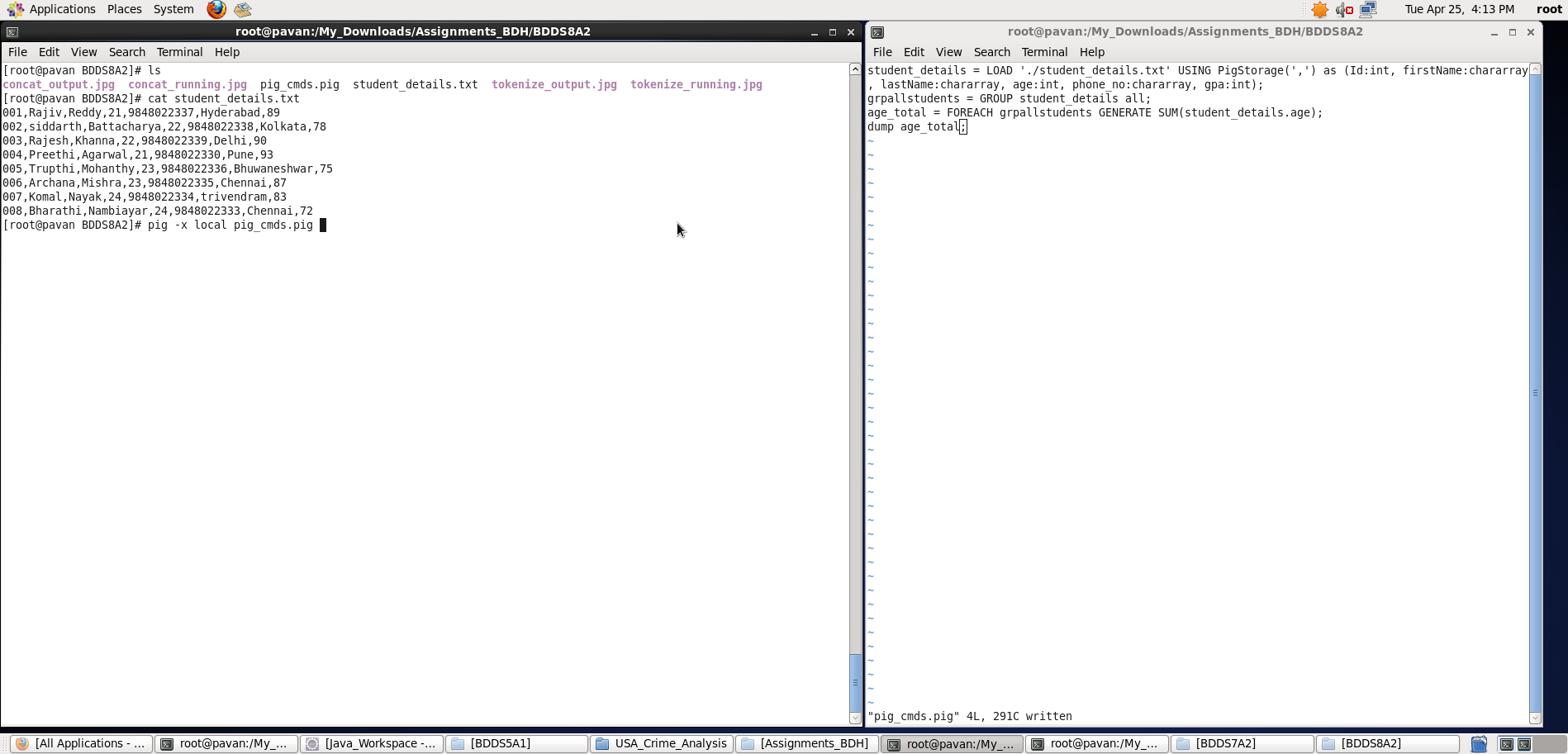
dump age\_total;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, grouping all the records into a single relation.
* Finding the sum of all ages of all students using SUM function.
* Outputting the age\_total relation.

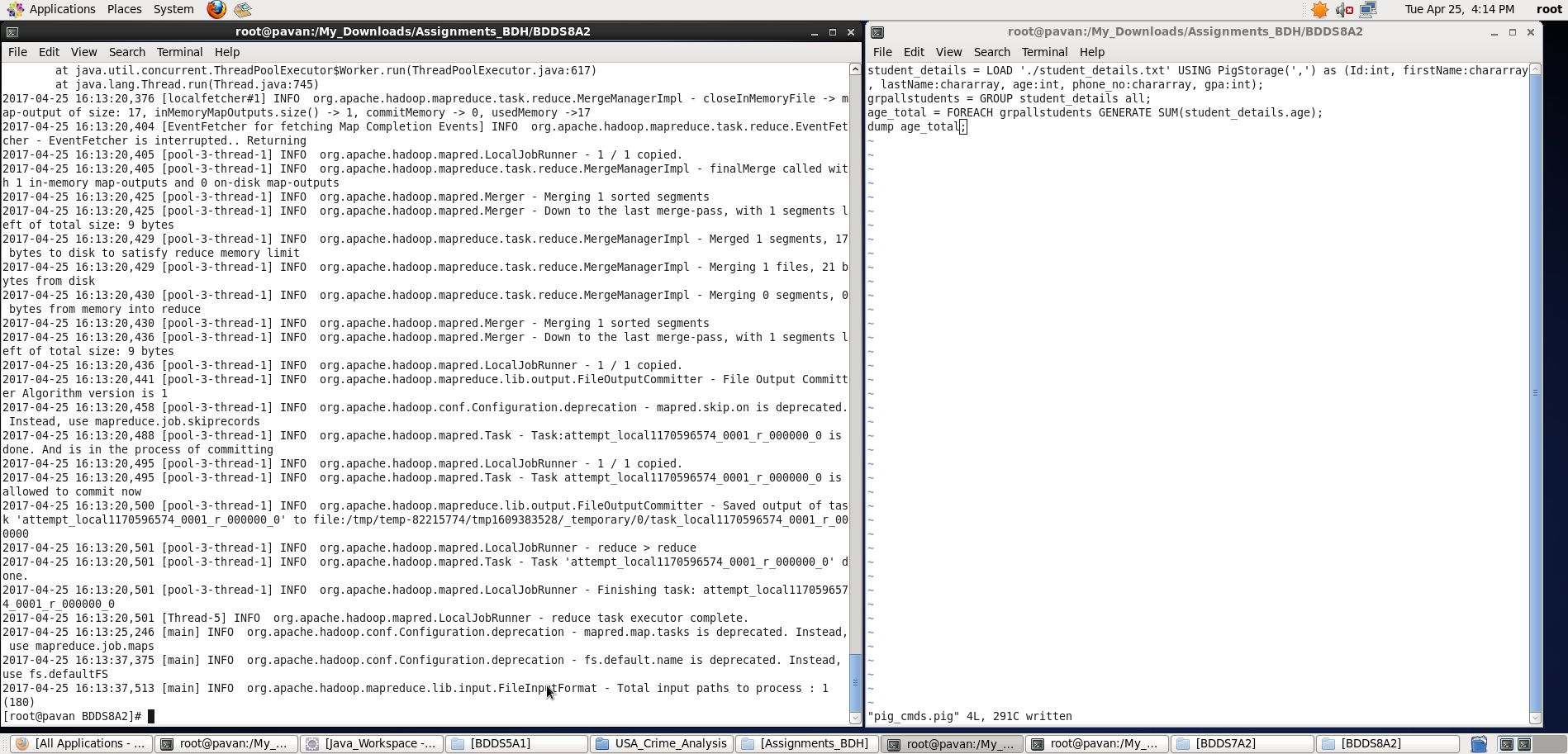
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<age\_total>**

**Note : We can store the output if required in a file using “store” command**



1. Min

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

grpallstudents = GROUP student\_details all;

age\_min = FOREACH grpallstudents GENERATE MIN(student\_details.age);

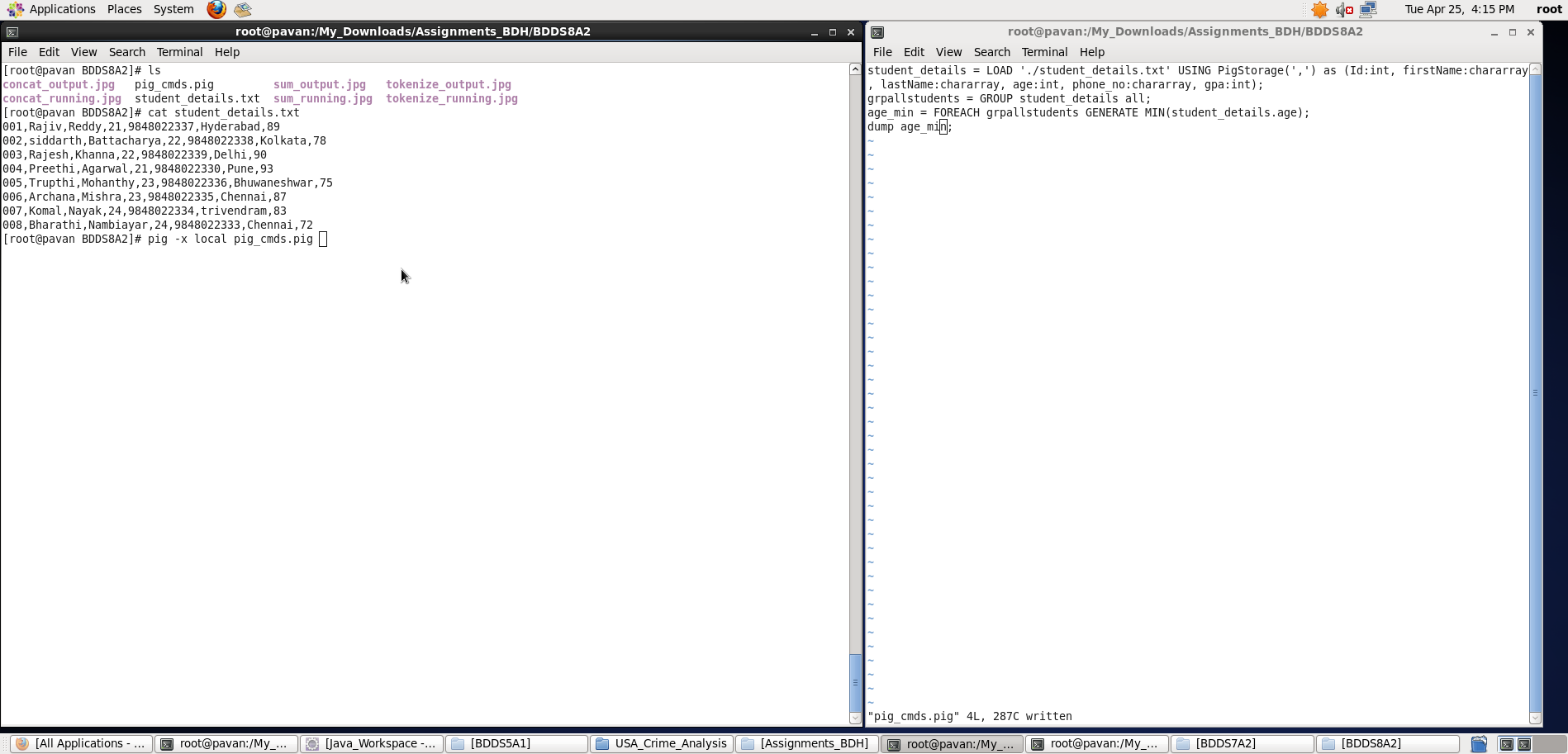
dump age\_min;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, grouping all the records into a single relation.
* Finding the minimum of all ages of all students using MIN function.
* Outputting the age\_min relation

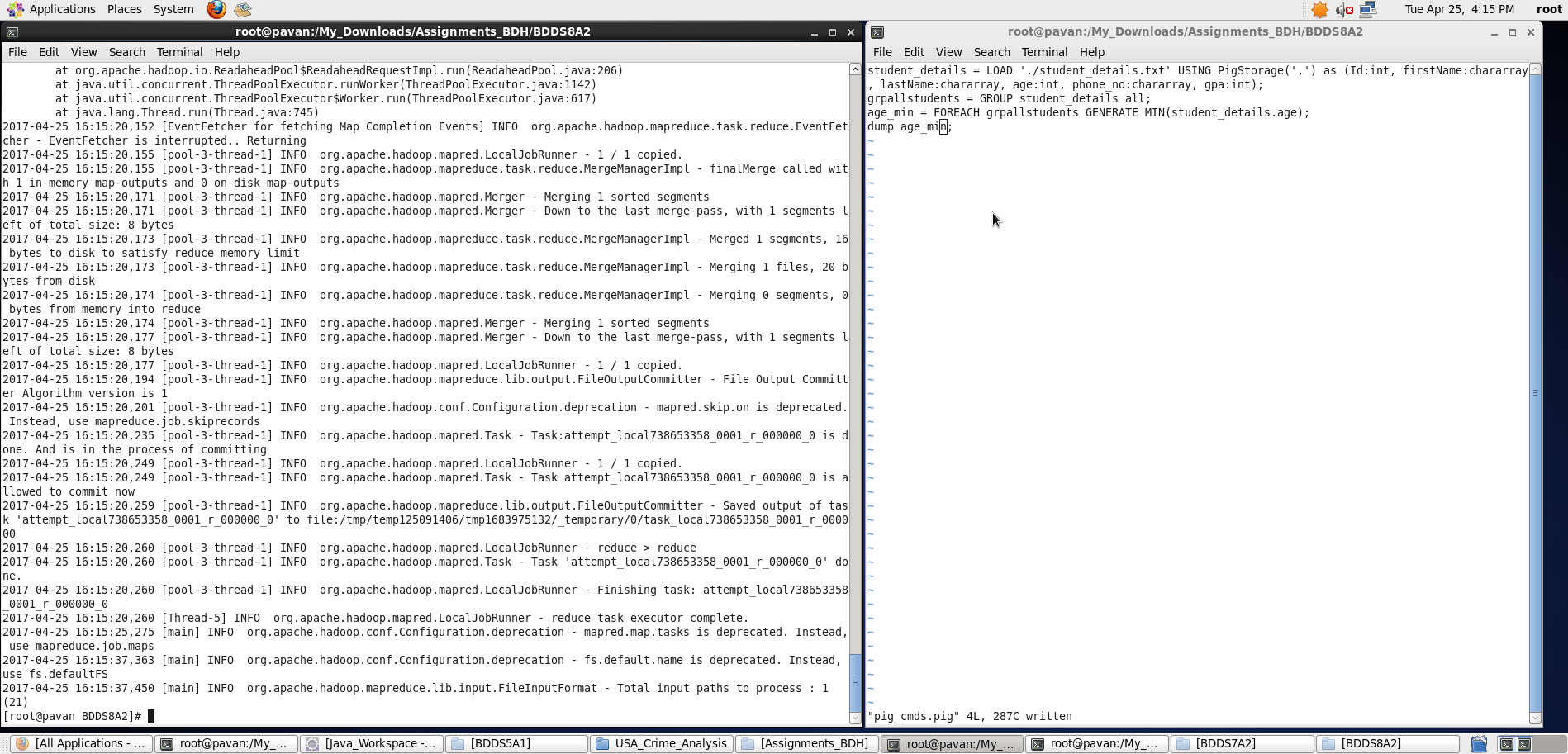
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<age\_min>**

**Note : We can store the output if required in a file using “store” command**



1. Max

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

grpallstudents = GROUP student\_details all;

age\_max = FOREACH grpallstudents GENERATE MAX(student\_details.age);

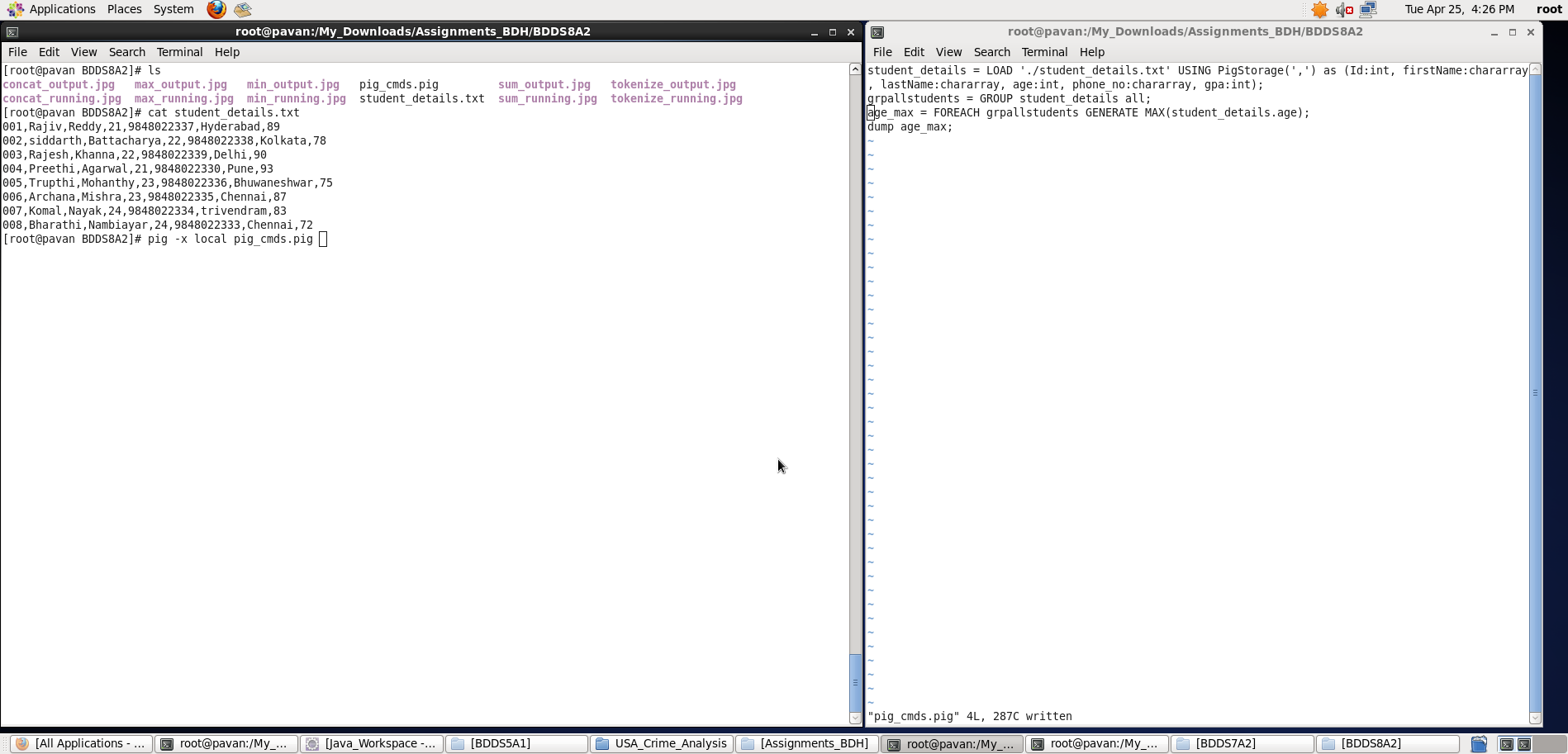
dump age\_max;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, grouping all the records into a single relation.
* Finding the maximum of all ages of all students using MAX function.
* Outputting the age\_max relation

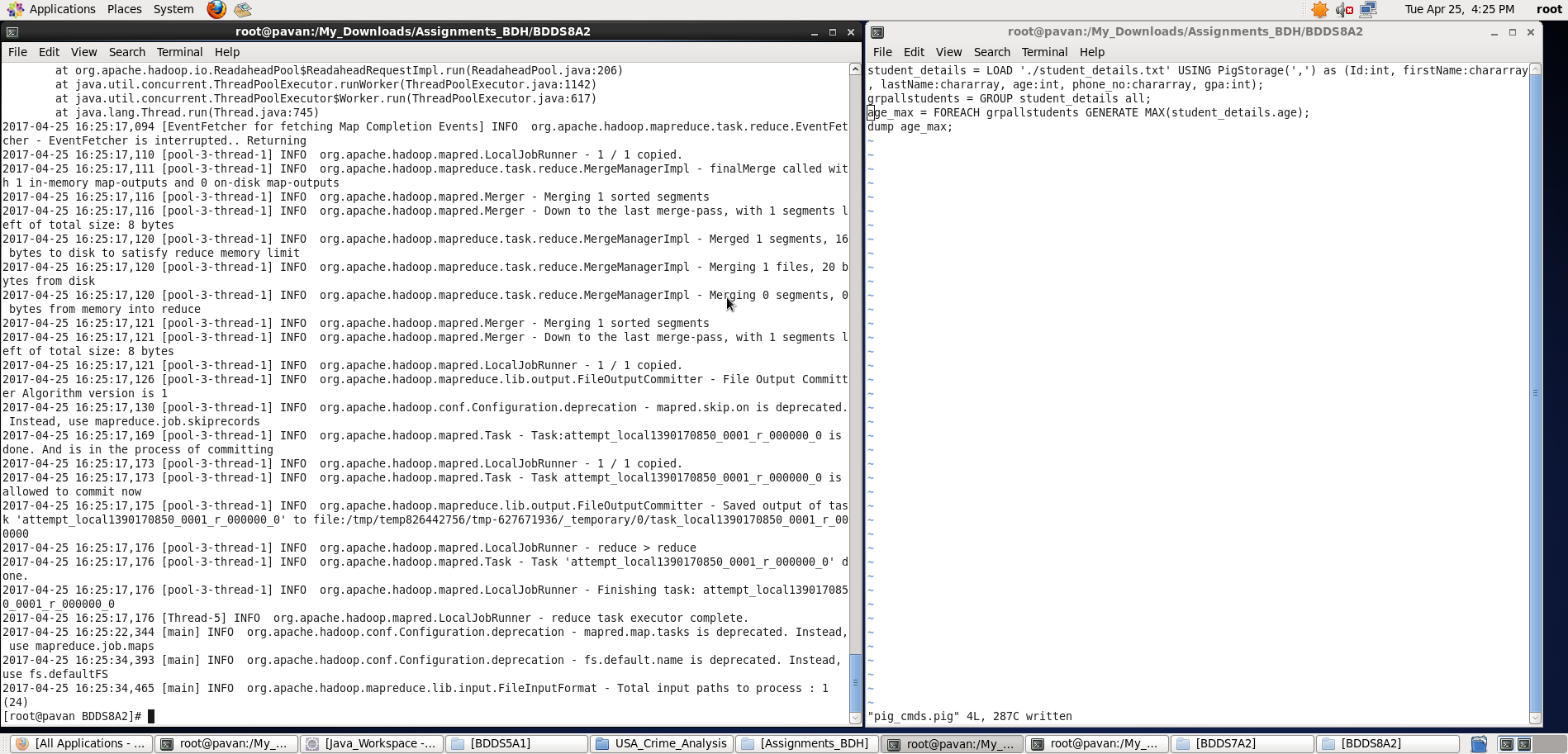
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<age\_max>**

**Note : We can store the output if required in a file using “store” command**



1. Limit

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

list\_first\_3\_std\_details = LIMIT student\_details 3;

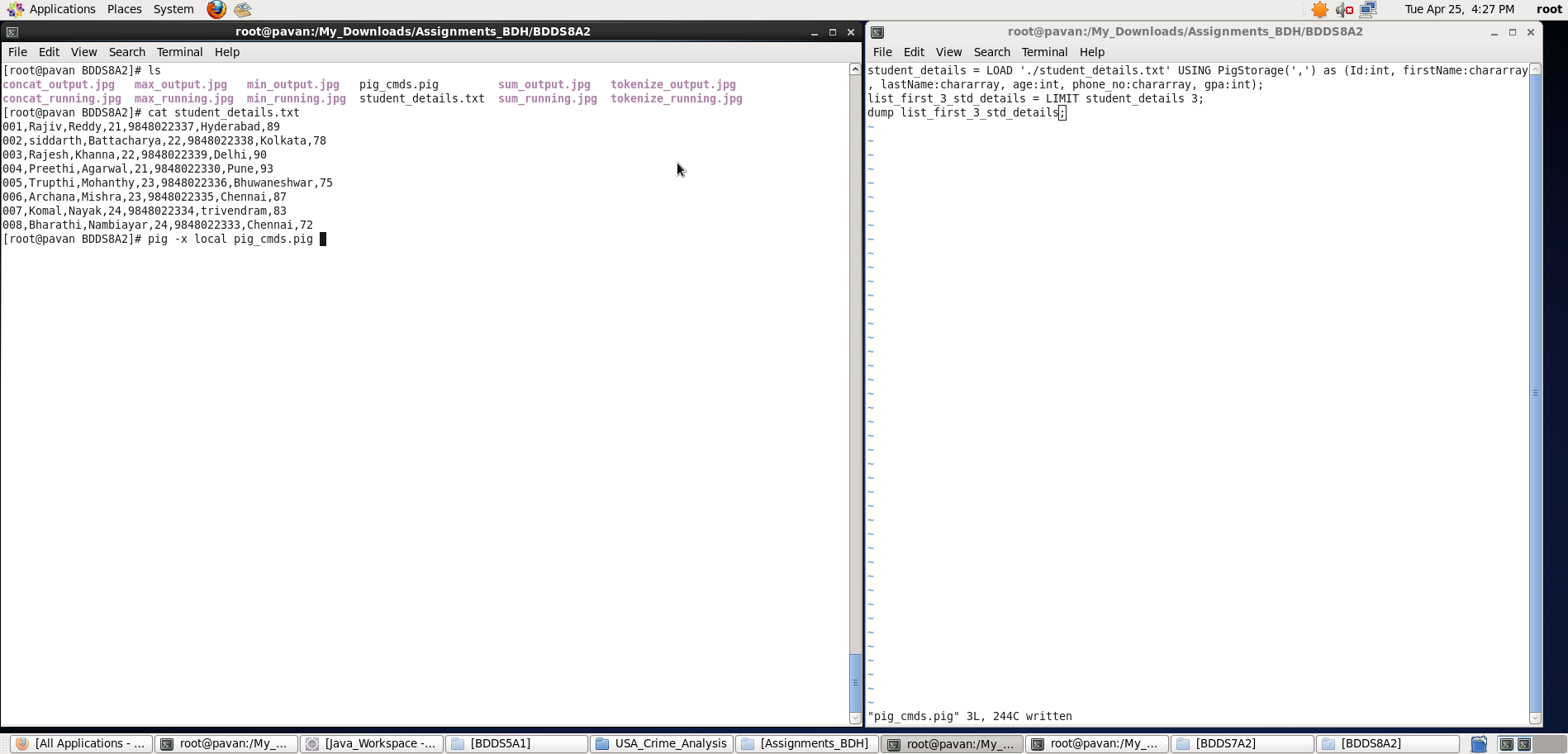
dump list\_first\_3\_std\_details;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, storing the first n records say, n=3 here, from the student\_details relation into list\_first\_3\_std\_details using LIMIT function.
* Outputting the list\_first\_3\_std\_details.

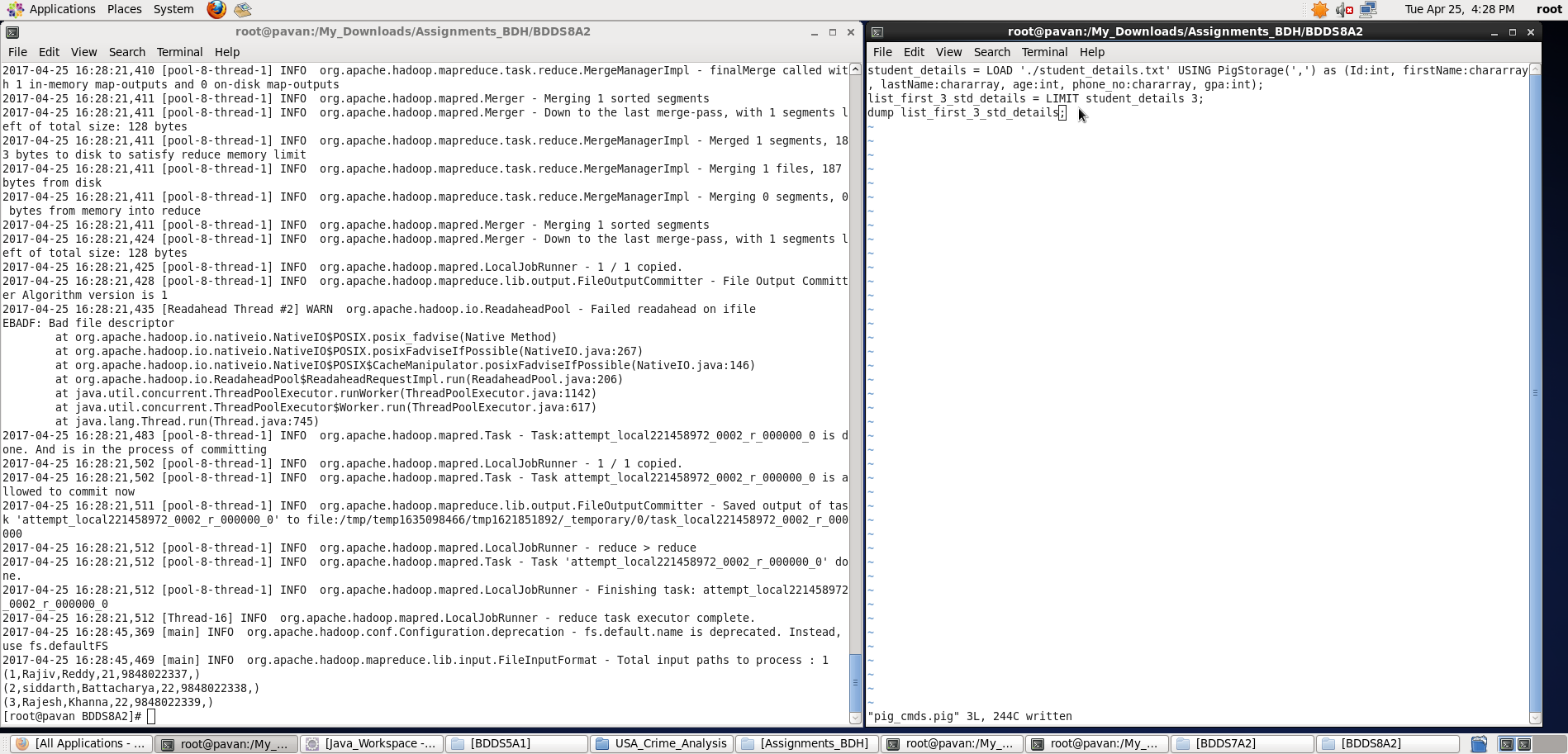
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<** **list\_first\_3\_std\_details>**

**Note : We can store the output if required in a file using “store” command**



1. Store

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

concat\_name = FOREACH student\_details GENERATE CONCAT (firstName,'\_',lastName);

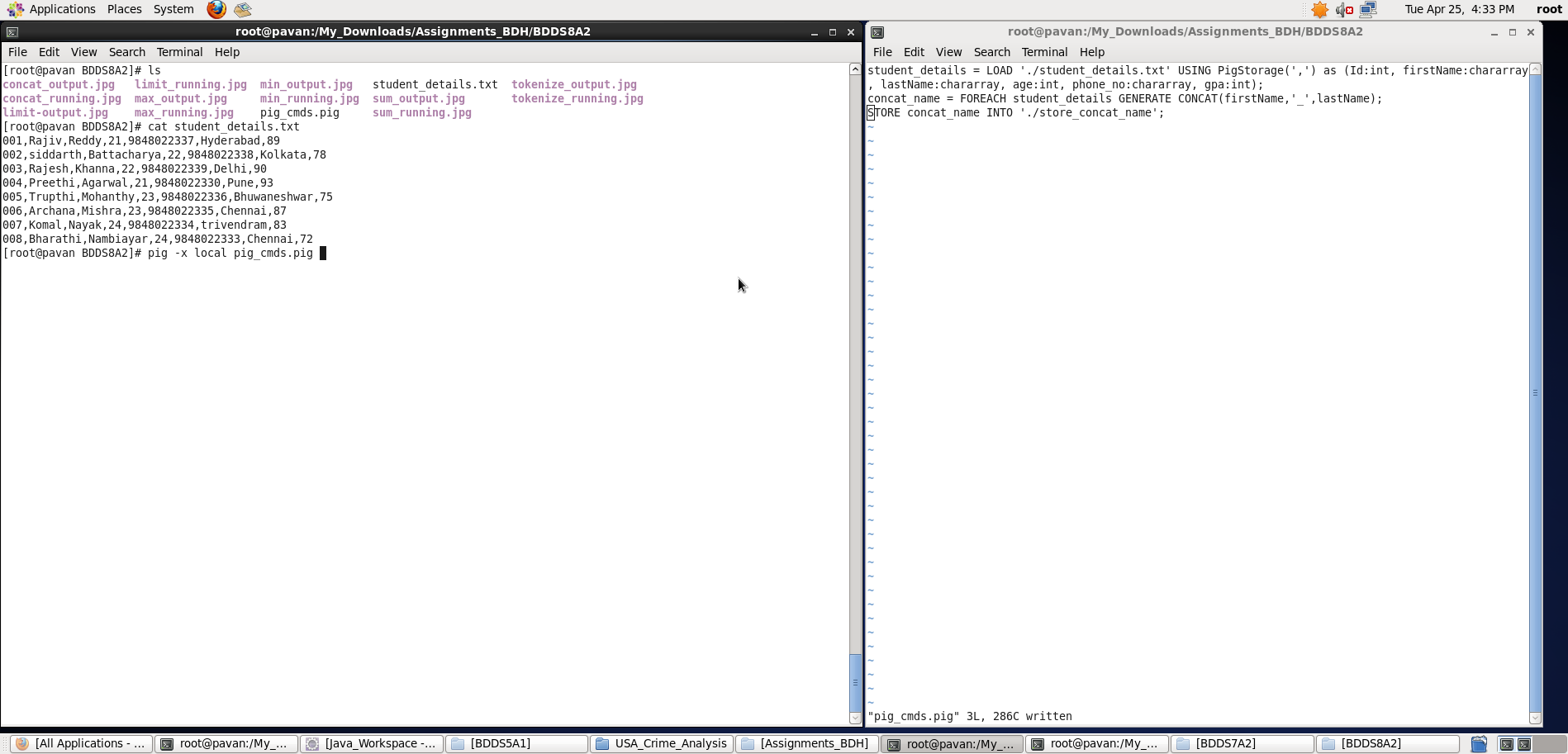
STORE concat\_name INTO ‘./store\_concat\_name’;

**Explanation:**

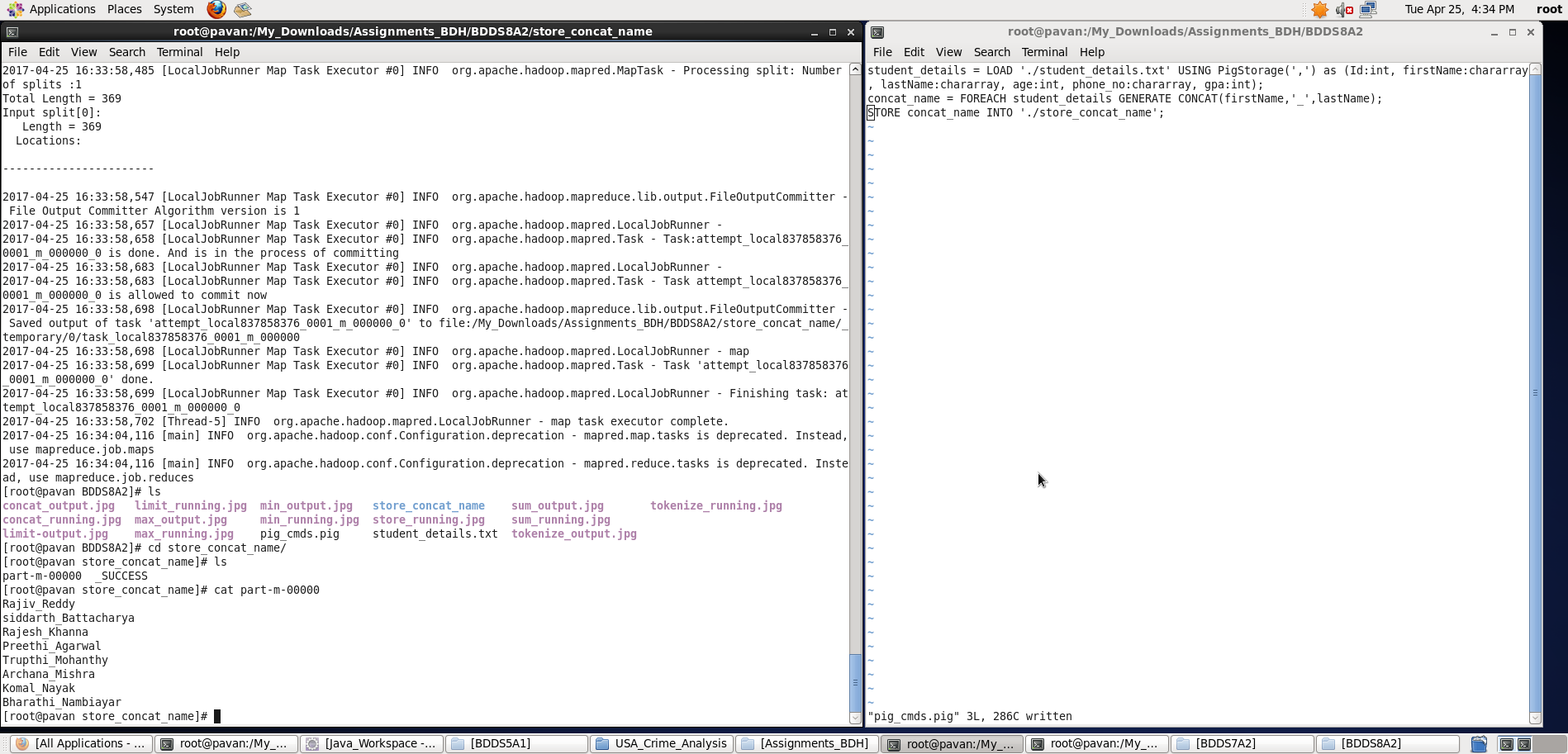
* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, for all records I have generated a new relation which contains concatenated result of the firstName and lastName using CONCAT function. Here ‘\_’ is kept in between the firstName and lastName and all the three expressions are concatenated.
* Instead of displaying concat\_name relation using DUMP command, we can store the result into a specified directory using STORE command.

**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Once, the result got stored, we can view the result as follows using ‘cat’ command.

****

**Output stored <concat\_name> inside ./store\_concat\_name/part-m-00000 file.**

1. Distinct

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

distinct\_student\_details = DISTINCT student\_details;

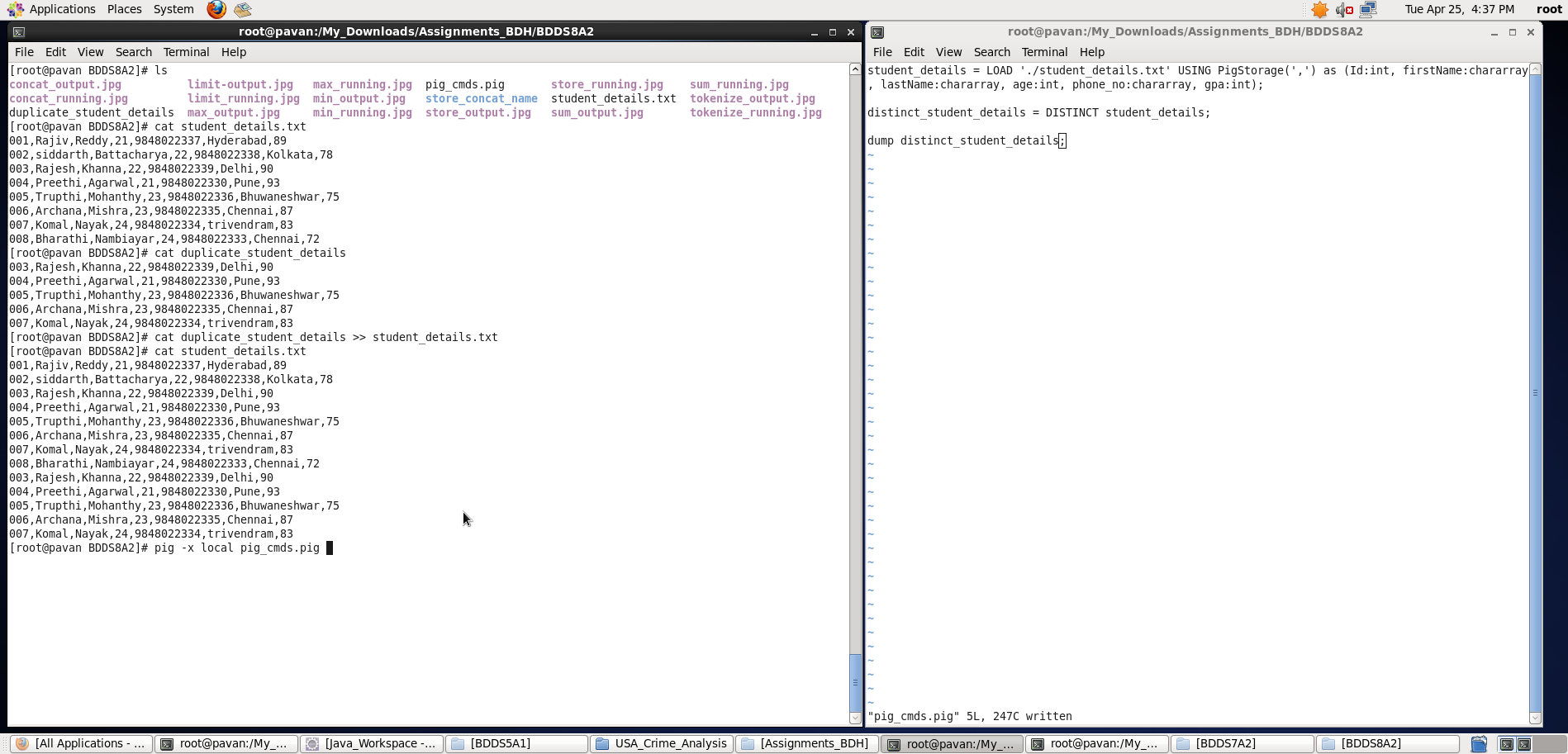
dump distinct\_student\_details;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* **Note: I have copied some duplicate records into the student\_details.txt file (as shown in below screenshot) using cat command.**
* Now, with the help of DISTINCT function, I have eliminated the duplicate records from the student\_details relation.
* Outputting distinct\_student\_details.

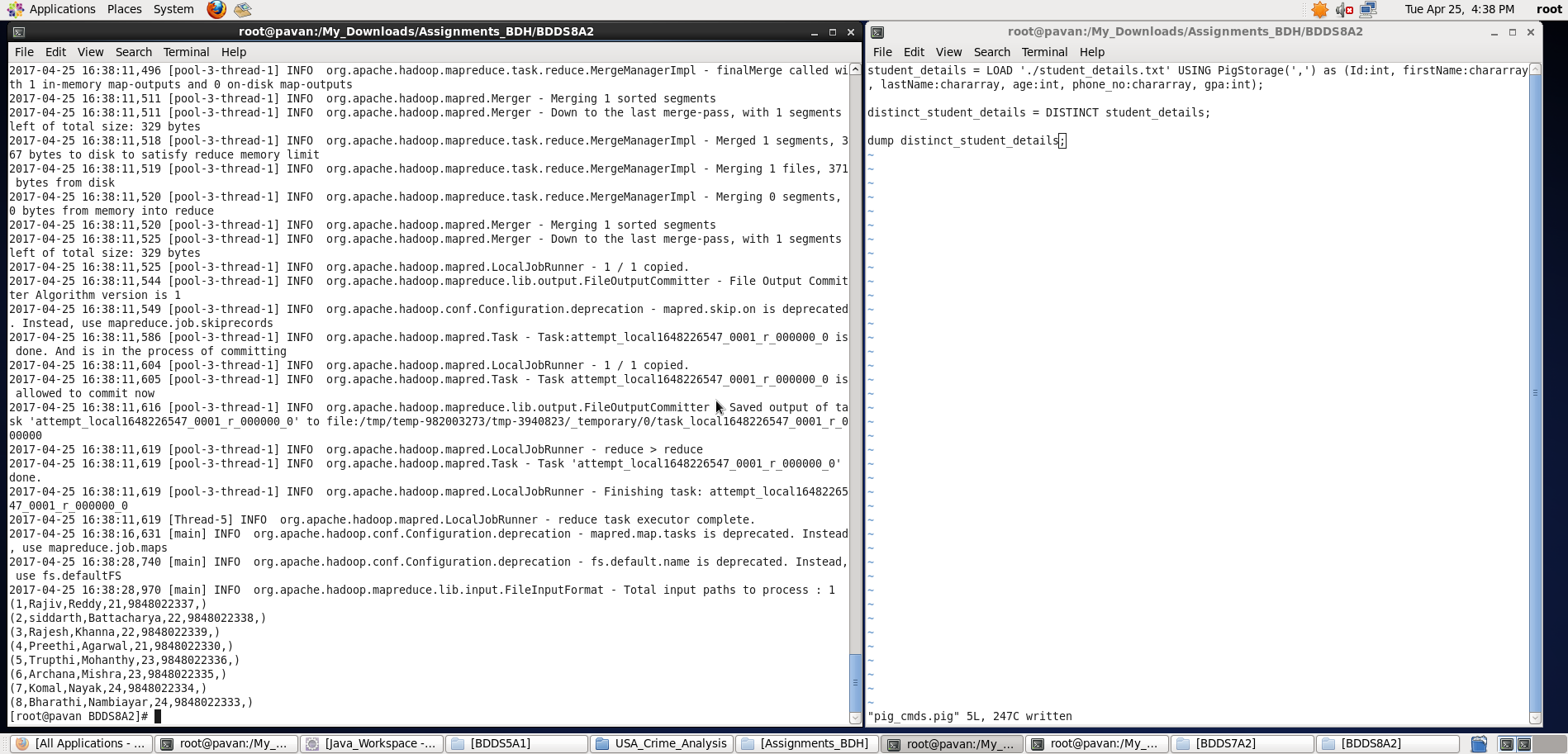
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<distinct\_student\_details>**

**Note : We can store the output if required in a file using “store” command**



1. Flatten

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

grpbyage = GROUP student\_details by age;

flatten\_output = foreach grpbyage generate group, flatten(student\_details);

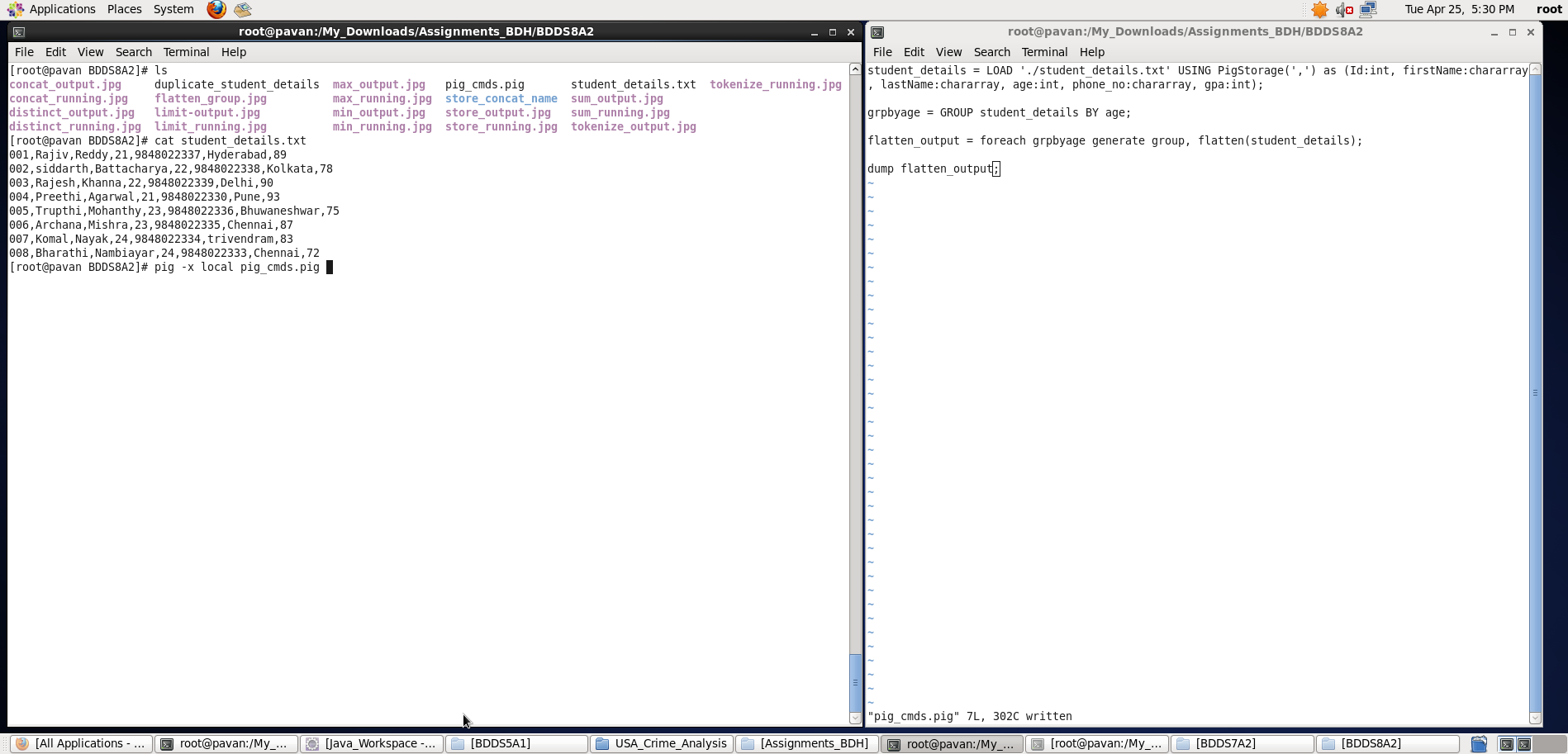
dump flatten\_output;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly
* Then, grouping the records based on age (key).
* We can see the grpbyage relation (output it using dump as shown below) as a tuple
* **The FLATTEN operator looks like a UDF syntactically, but it is actually an operator that changes the structure of tuples and bags in a way that a UDF cannot. Flatten un-nests tuples as well as bags.**
* Now, using flatten operator, I have un-nested the grpbyage tuple into individual records as shown below.

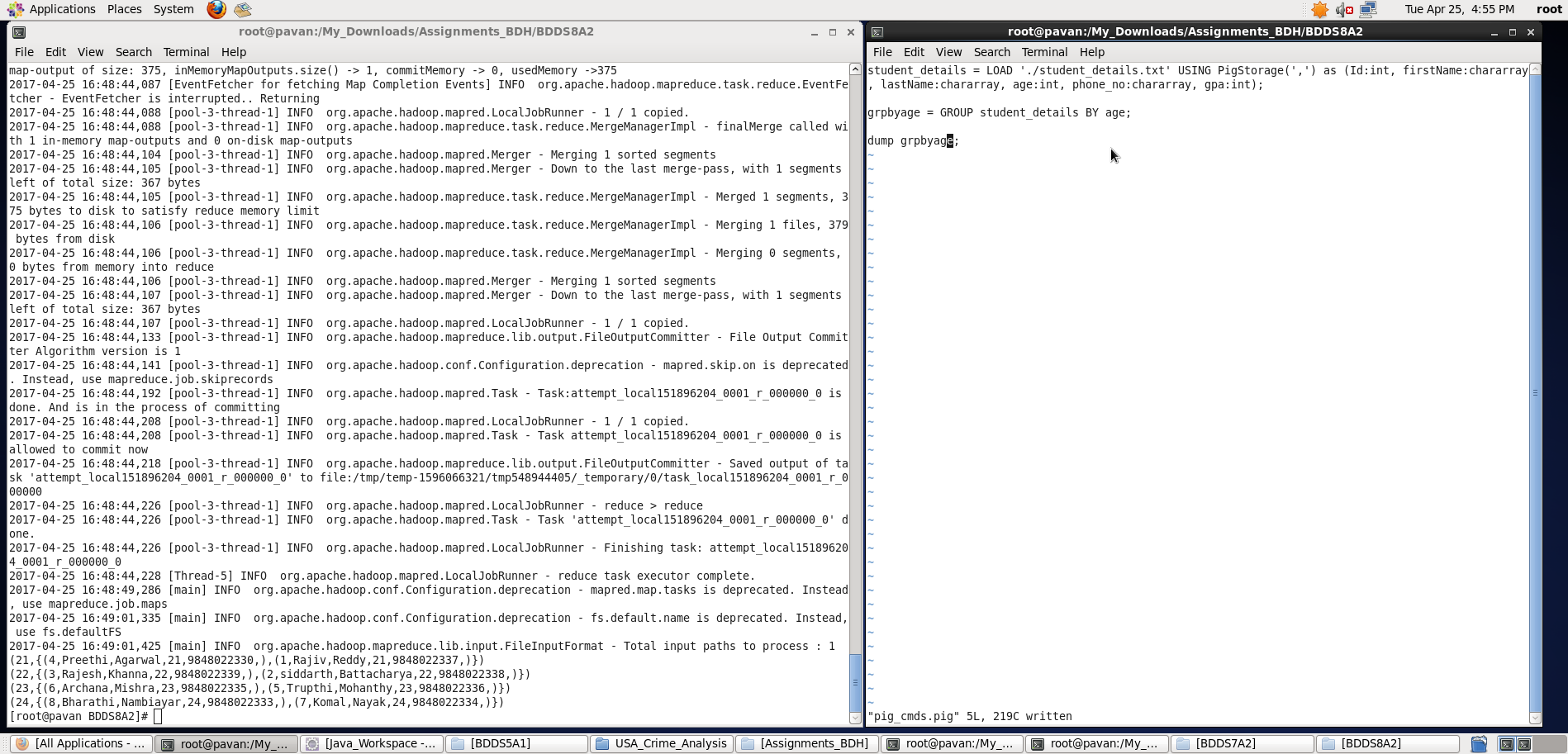
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



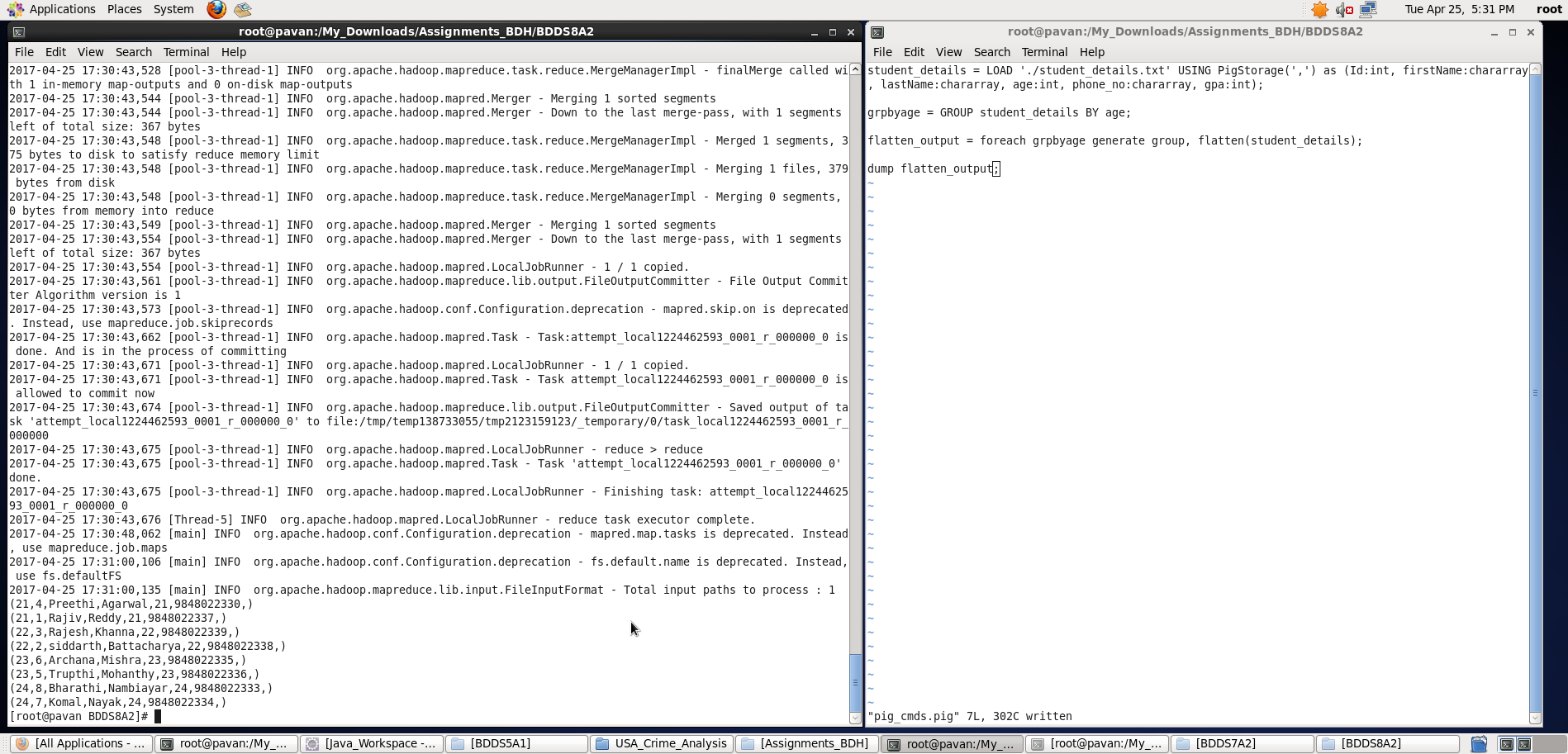
Output displayed: **<grpbyage>**

**Note : We can store the output if required in a file using “store” command**



Output displayed: **<flatten\_output>**

**Note : We can store the output if required in a file using “store” command**

****

1. IsEmpty

**Pig Latin Script**:

student\_details = LOAD './student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

duplicate\_details = LOAD './duplicate\_student\_details.txt' USING PigStorage(',') as (Id:int, firstName:chararray, lastName:chararray, age:int, phone\_no:chararray, gpa:int);

cogroup\_data = COGROUP student\_details by Id, duplicate\_details by Id;

isempty\_data = filter cogroup\_data by IsEmpty(duplicate\_details);

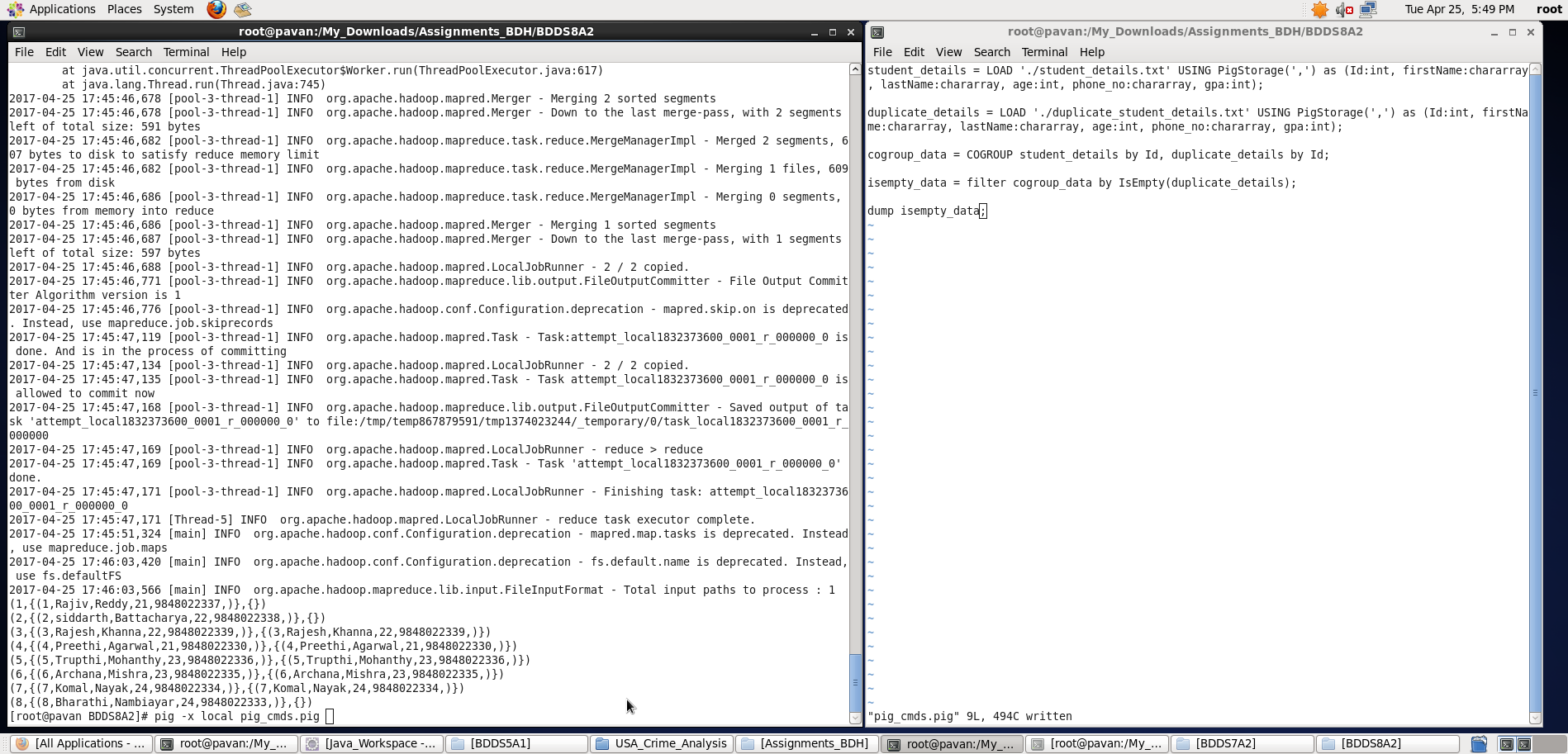
dump isempty\_data;

**Explanation:**

* Loaded data from student\_details.txt into “student\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly.
* Loaded data from duplicate\_student\_details.txt into “duplicate\_details” relation. Here ‘,’ is the delimiter and schema is provided accordingly.
* Now, co-grouping both the relations based on common key – Id (here) into cogroup\_data relation.
* **Note**: **There will be some empty fields in this cogroup\_data (as shown in below screenshot) because all the Id’s in student\_details wont have corresponding records in duplicate\_details.**
* Therefore, filtering based on empty duplicate\_details field in cogroup\_data relation using IsEmpty function.
* **IsEmpty function returns True if that corresponding field is empty.**
* Outputting the isempty\_data

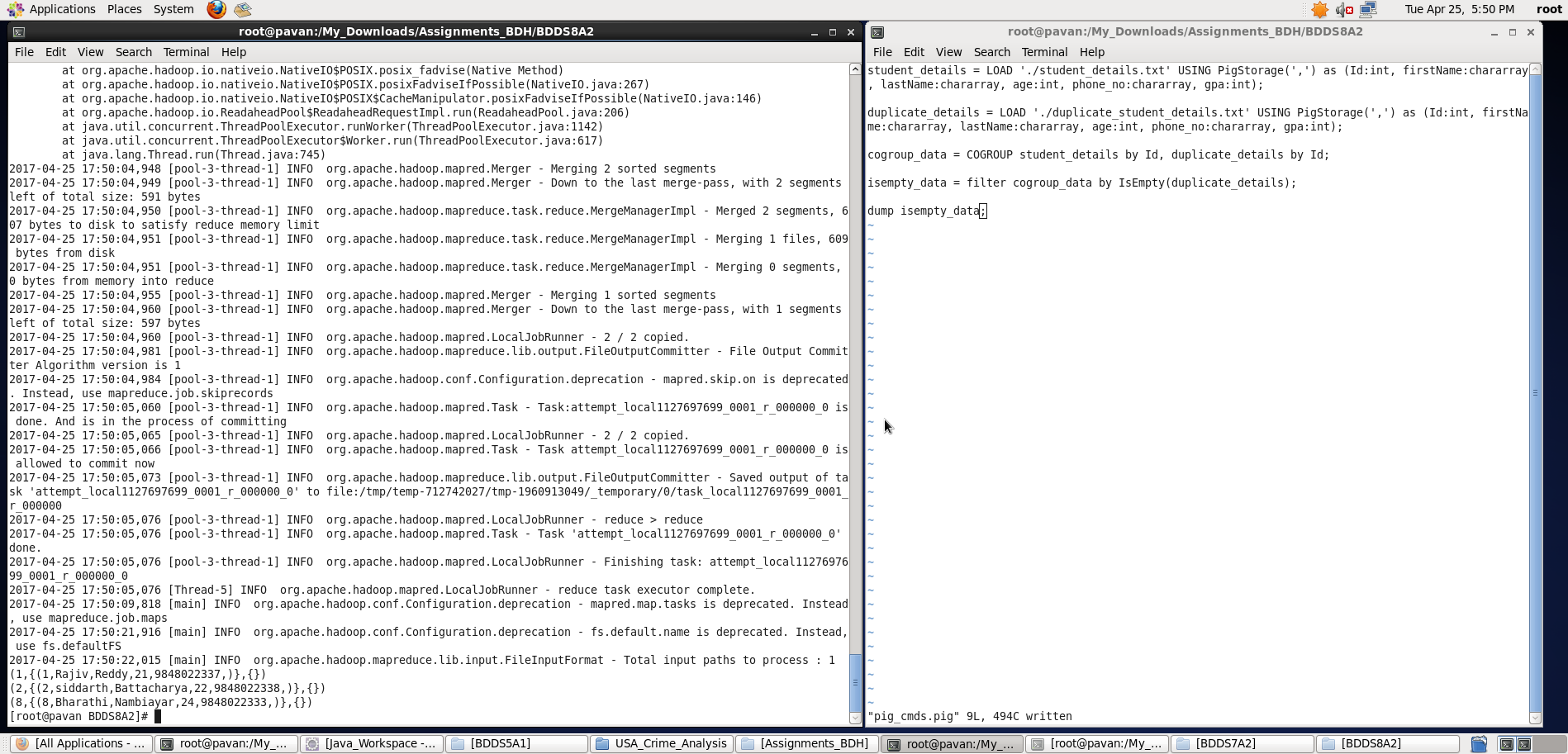
**Screenshots**

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS8A2/ directory



Output displayed: **<isempty\_data>**

**Note : We can store the output if required in a file using “store” command**



**Thus, I have created a sample dataset and implemented the stated Pig commands on the dataset.**