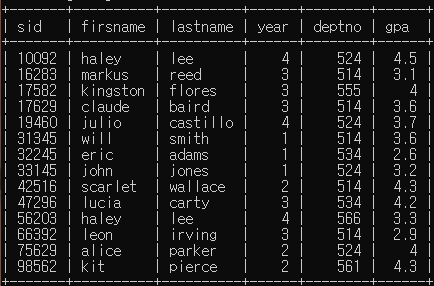
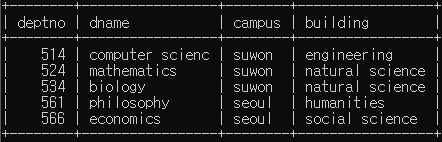
**(1) [50 pts]** Implement the following SQL statement using the MapReduce framework. Assume that there are two tables: student(sid, firstname, lastname, year, deptno, gpa) and dept(deptno, dname, campus, building) as follows. **Implement and explain your map and reduce function with execution snapshots.**



**TABLE “student”**

****

**TABLE “dept”**

|  |
| --- |
| **SELECT d.dname, max(s.gpa), d.campus**  **FROM student s**  **JOIN dept d ON s.deptno = d.deptno**  **GROUP BY d.dname**  **HAVING avg(s.gpa) > 3.5;** |

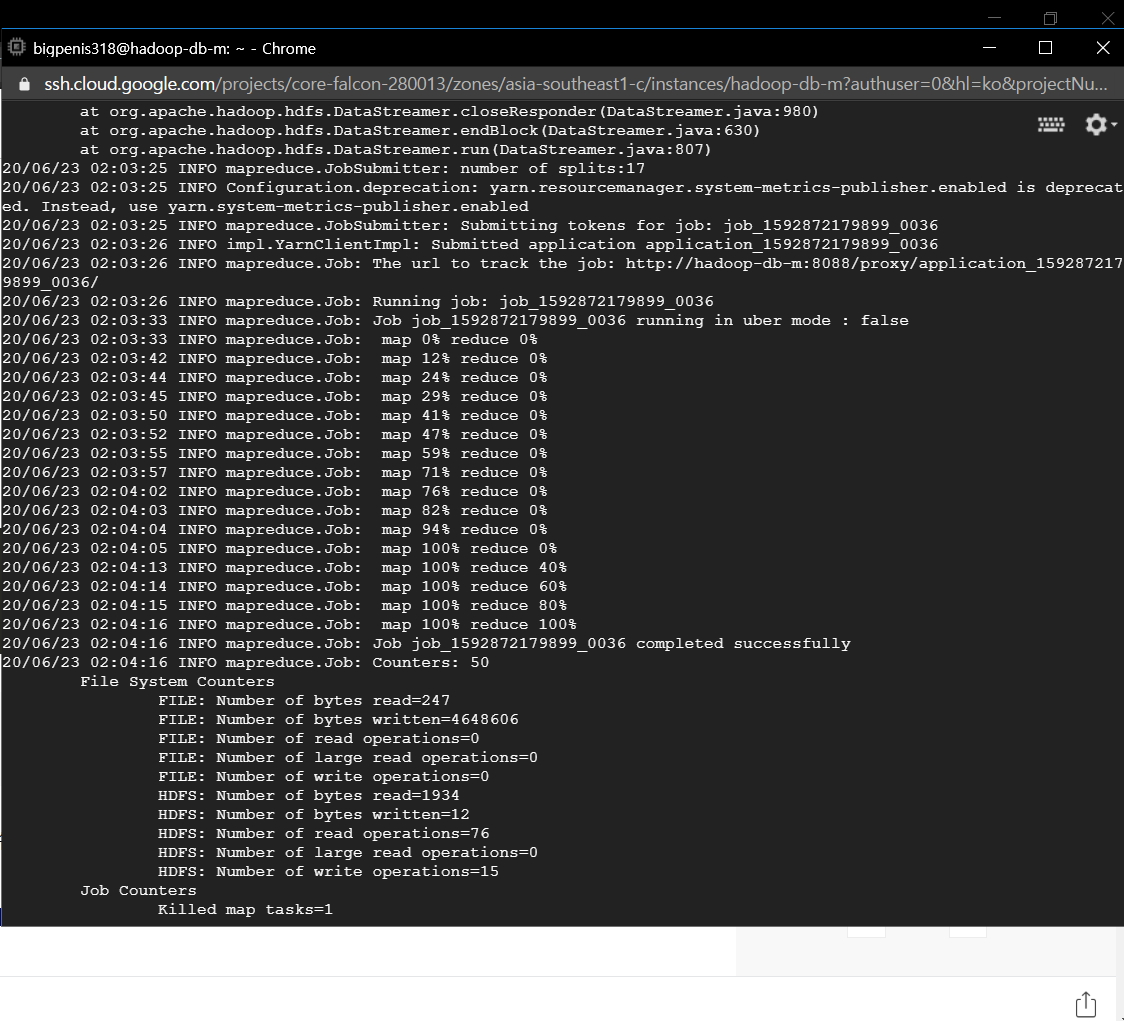
**NOTE 1**: You should write your codes using Hadoop streaming with Python.

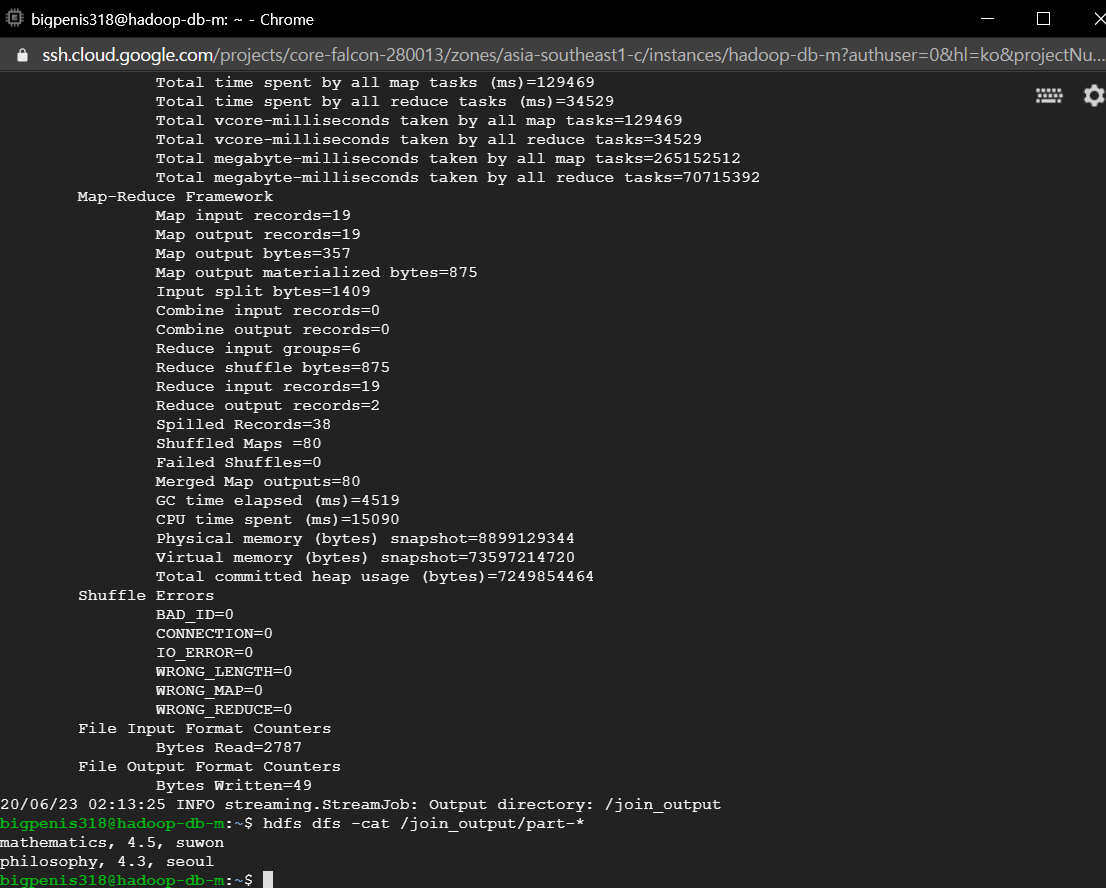
**NOTE 2**: Your code should consist of ‘map.py’ and ‘reduce.py.’

**NOTE 3**: You should use the following files (i.e., dept, student) as input.

**Answer: (Submit your code to i-campus. You don’t have to write your code in the documentation.)**

**If tuple is derived from student file, map function transforms tuple to (deptno, ‘student,’+gpa) which is (key, value) pair. If tuple is derived from dept file, map function transforms tuple to (deptno, ‘dept,’+dname+‘:’+campus) which is (key, value) pair. Then, reduce function group these (key, value) pairs. The pairs are sorted by deptno. If a (key, value) pair came from student file, we need only gpa from them. So append value of gpa in a list. If a (key, value) pair came from dept file, we need dname, location from them. So split the string by character “:” to separate dname and location. If deptno were changed, the tuples which came until just before are in same group. So get sum of gpa values and get average gpa. If average is bigger than 3.5, print dname, max(gpa), and location. Then one group is totally processed, so empty the content of dname, location, gpa. Repeat these algorithm until we finish checking all tuples. Then after all, we can get dname, gpa, location only if an average of gpa in department is bigger than 3.5.**

****

****

**(2) [100 pts]** Solve the problem of finding a dominant set using the MapReduce framework. Assume that data is divided into multiple files. **Implement and explain your map and reduce functions with execution snapshots.**

**[Definition 1: Dominant relationship]**

For tuple , satisfies the following two conditions:

**[Definition 2: Dominant tuple]**

Given a set of tuples , a set of dominant tuples is defined as:

**[Example]**

|  |  |  |
| --- | --- | --- |
| **Name**  110  120  **Price (\)**  **Distance (m)**  a  b  c  d  e  f  g  i  h  j  k  m  l  1,500  3,000  4,500  6,000  7,500  9,000  10,500  13,000  14,500  16,000  10  20  30  40  50  60  70  80  90  100 | **Price (\)** | **Distance (m)** |
| **a** | **12,500** | **20m** |
| **h** | **4,800** | **30m** |
| **i** | **7,000** | **43m** |
| **b** | **14,700** | **45m** |
| **c** | **12,500** | **58m** |
| **k** | **3,200** | **61m** |
| **f** | **8,500** | **67m** |
| **j** | **4,500** | **78m** |
| **d** | **10,700** | **88m** |
| **l** | **6,300** | **105m** |
| **m** | **1,600** | **107m** |
| **e** | **14,700** | **115m** |
| **g** | **8,800** | **115m** |

**TABLE “Restaurant”**

The dominant relationships for the relation “Restaurant” are:

*, , , , , , , , , ……*

As a result, a set of dominant tuples is = {a, h, k, m}

**[Problem]**

When recommending the restaurants according to two attributes, the distance and price, we want to find a set of dominant tuples. Implement an algorithm of finding a set of dominant tuples using the MapReduce framework.

**NOTE 1: You should get the result in one MapReduce process.**

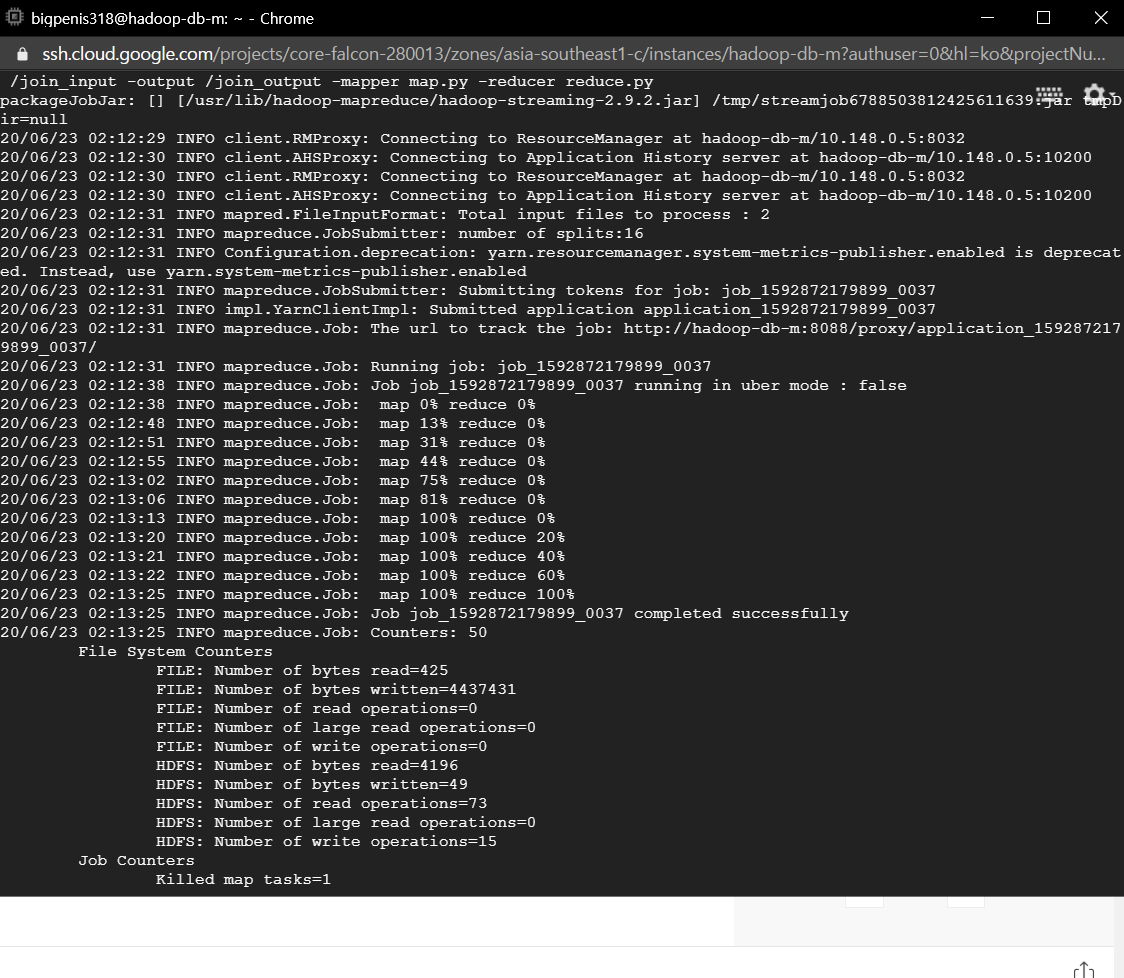
**NOTE 2**: You should use the following files as input.

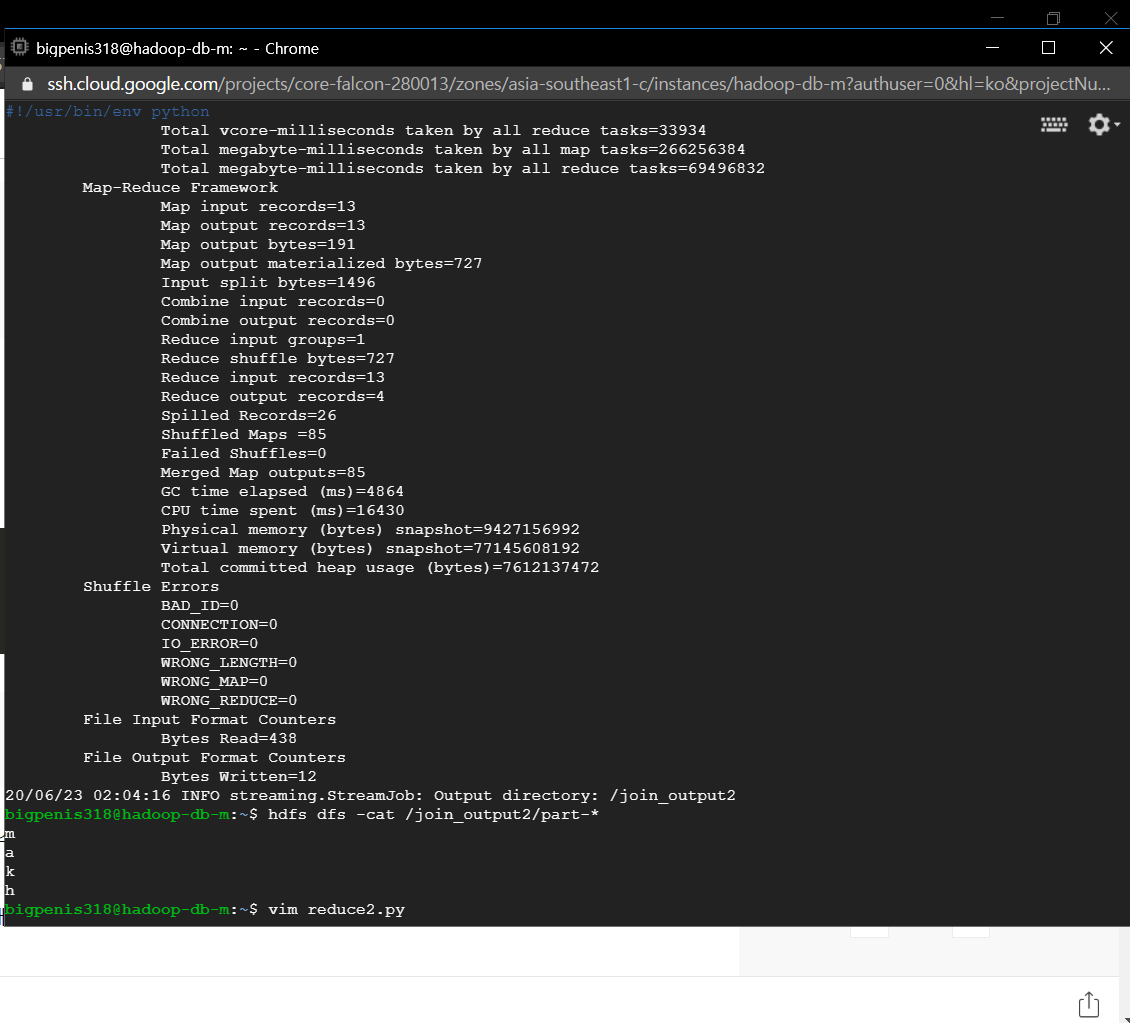
**NOTE 3**: You should write your codes using Hadoop streaming with Python.

**NOTE 4**: Your code should consist of ‘map.py’ and ‘reduce.py.’

**Answer: (Submit your code to i-campus. Don’t write your code here.)**

**In map function, get all tuples and toss them to reduce function. Key of tossed information is unified by named “key”. In reduce function, excluded key, get all information and separate to name, price and distance. Then append to some different lists. If appending is finished, check tuples one by one that whether this tuple is at bigger side of some dominant relationship. If a tuple is not in bigger side of any dominant relationship, it is element of dominant set. So print it.**

****

****