Nosql-riak-MapReduce

Please do exercises -

Section 2.1 - Population of buckets:

The four objects with the text “pizza”:

```
tdend2:~/riak-1.2.1/dev$
tdend2:~/riak-1.2.1/dev$ curl -XPUT http://localhost:8091/buckets/training/keys/
/bar -H 'Content-Type: text/plain' -d 'pizza pizza pizza pizza'
tdend2:~/riak-1.2.1/dev$ curl -XPUT http://localhost:8091/buckets/training/keys/
/baz -H 'Content-Type: text/plain' -d 'nothing to see here'
tdend2:~/riak-1.2.1/dev$ curl -XPUT http://localhost:8091/buckets/training/keys/
/bam -H 'Content-Type: text/plain' -d 'pizza pizza pizza'
tdend2:~/riak-1.2.1/dev$ curl -XPUT http://localhost:8091/buckets/training/keys/
/foo -H 'Content-Type: text/plain' -d 'pizza data goes here'
```

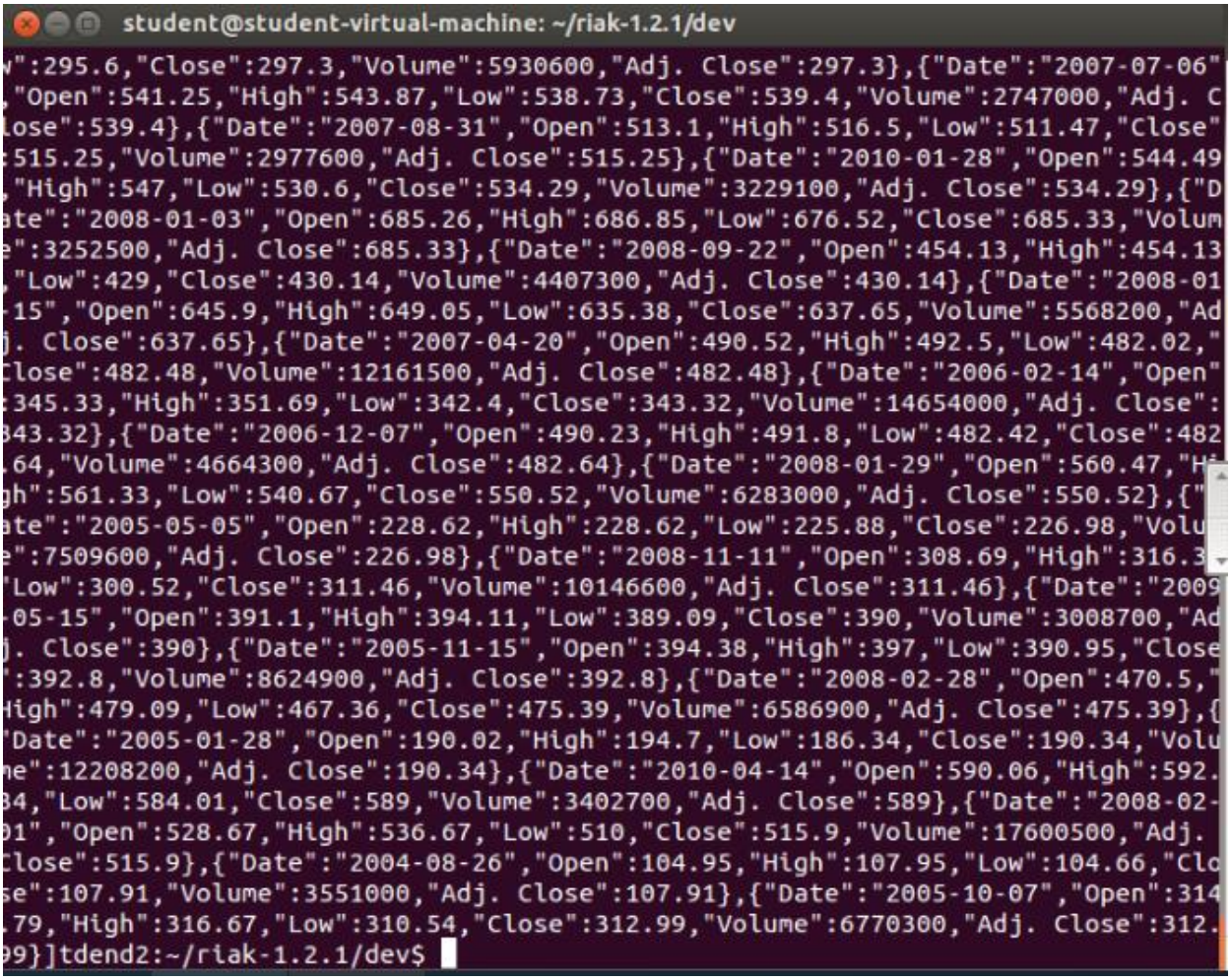
Section 2.2 - the MapReduce script:

```
tdend2:~/riak-1.2.1/dev$ curl -XPOST http://localhost:8091/mapred -H 'Content-Ty
pe: application/json' -d '{"inputs":"training",
"query":[{"map":{"language":"javascript",
"source":"function(riakObject) {
   var m = riakObject.values[0].data.match(/pizza/g);
   return [[riakObject.key, (m ? m.length : 0 )]];
}"}}]}'
[["baz",0],["foo",1],["bar",4],["bam",3]]tdend2:~/riak-1.2.1/dev$
tdend2:~/riak-1.2.1/dev$
```

Run above Queries 1 - 5 in Section 5.

```
tdend2:~/riak-1.2.1/dev$ curl -X POST http://127.0.0.1:8091/mapred -H "Content-Ty
pe: application/json" -d @-
{"inputs":"goog",
 "query":[{"map":{"language":"javascript",
                  "name":"Riak.mapValuesJson",
                  "keep":true}}
        ]
}
```

Result:



5.2.Query 2: simple subset A simple map-only job returns the values for the first week of January 2010.

```
tdend2:~/riak-1.2.1/dev$ curl -X POST http://127.0.0.1:8091/mapred -H "Content-Ty
pe: application/json" -d @-
{"inputs":[["goog","2010-01-04" ],
          ["goog","2010-01-05" ],
          ["goog","2010-01-06"],
          ["goog","2010-01-07" ],
          ["goog","2010-01-08" ]],
 "query":[{"map":{"language":"javascript",
                  "name":"Riak.mapValuesJson",
                  "keep":true}}
        ]
}

[{"Date":"2010-01-04","Open":626.95,"High":629.51,"Low":624.24,"Close":626.75,"Vo
lume":1956200,"Adj. Close":626.75},{"Date":"2010-01-05","Open":627.18,"High":627.
84,"Low":621.54,"Close":623.99,"Volume":3004700,"Adj. Close":623.99},{"Date":"201
0-01-06","Open":625.86,"High":625.86,"Low":606.36,"Close":608.26,"Volume":3978700
,"Adj. Close":608.26},{"Date":"2010-01-07","Open":609.4,"High":610,"Low":592.65,"
Close":594.1,"Volume":6414300,"Adj. Close":594.1},{"Date":"2010-01-08","Open":592
,"High":603.25,"Low":589.11,"Close":602.02,"Volume":4724300,"Adj. Close":602.02}]
tdend2:~/riak-1.2.1/dev$
```

5.3.Query 3: map subset A map-only job that returns the “High” sell values for the first week of January 2010.

```
tdend2:~/riak-1.2.1/dev$ curl -X POST http://127.0.0.1:8091/mapred -H "Content-Ty
pe: application/json" -d @-
{"inputs":[["goog","2010-01-04" ],
          ["goog","2010-01-05" ],
          ["goog","2010-01-06"],
          ["goog","2010-01-07" ],
          ["goog","2010-01-08" ]],
 "query":[{"map":{
"language":"javascript",
 "source":"function(value,keyData,arg){
              var data = Riak.mapValuesJson(value)[0];
              return [data.High];
}"
,"keep":true}}]

}

[629.51,625.86,627.84,610,603.25]tdend2:~/riak-1.2.1/dev$
```

5.4.Query 4: A MapReduce query: the highest “High” value by adding reduce to create a job that returns the maximum high sell value in the first week of January.

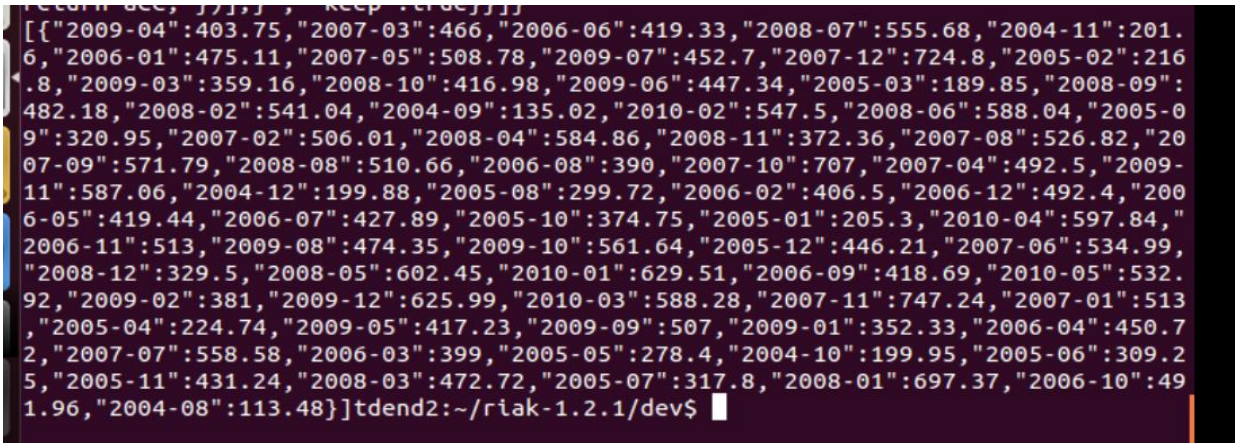
```
tdend2:~/riak-1.2.1/dev$ curl -X POST http://127.0.0.1:8091/mapred -H "Content-Ty
pe: application/json" -d @-
{"inputs":[["goog","2010-01-04" ],
          ["goog","2010-01-05" ],
          ["goog","2010-01-06"],
          ["goog","2010-01-07" ],
          ["goog","2010-01-08" ]],
 "query":[{"map":{
"language":"javascript",
 "source":"function(value,keyData,arg){
              var data = Riak.mapValuesJson(value)[0];
              return [data.High];
}"
}},
{"reduce":{"language":"javascript","name":"Riak.reduceMax",
"keep":true}}]

}

[629.51]tdend2:~/riak-1.2.1/dev$
```

Hence, 629.51 is the highest value.

5.5- query to collect the maximum highs by month.

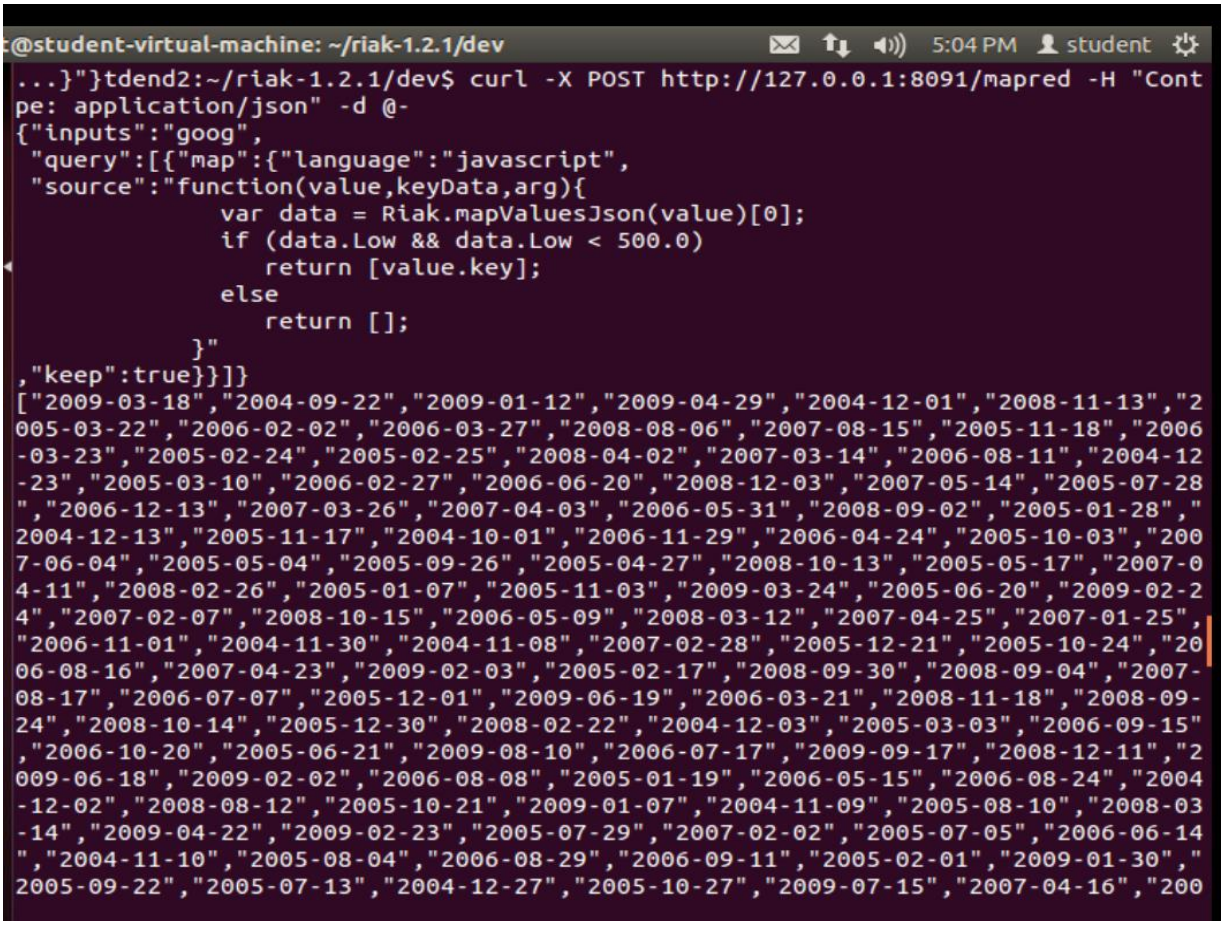
```
v$
tdend2:~/riak-1.2.1/dev$ curl -X POST http://127.0.0.1:8091/mapred -H "Content-Ty
pe: application/json" -d @-
{ "inputs":"goog",
"query":[
{"map":{"language":"javascript",
"source":"function(value, keyData, arg){
var data = Riak.mapValuesJson(value)[0];
var month = value.key.split('-').slice(0,2).join('-');
var obj = {};
obj[month] = data.High; return [obj];}"}},
{"reduce":{"language":"javascript",
"source":"function(values, arg){
return [ values.reduce(function(acc, item){
for(var month in item){
if(acc[month]) {
acc[month] = (acc[month] < item[month]) ? item[month] : acc[month]; }
else { acc[month] = item[month]; }
}
return acc; })];}", "keep":true}}]}
[{"2009-04":403.75,"2007-03":466,"2006-06":419.33,"2008-07":555.68,"2004-11":201.
6,"2006-01":475.11,"2007-05":508.78,"2009-07":452.7,"2007-12":724.8,"2005-02":216
.8,"2009-03":359.16,"2008-10":416.98,"2009-06":447.34,"2005-03":189.85,"2008-09":
482.18,"2008-02":541.04,"2004-09":135.02,"2010-02":547.5,"2008-06":588.04,"2005-0
9":320.95,"2007-02":506.01,"2008-04":584.86,"2008-11":372.36,"2007-08":526.82,"20
07-09":571.79,"2008-08":510.66,"2006-08":390,"2007-10":707,"2007-04":492.5,"2009-
11":587.06,"2004-12":199.88,"2005-08":299.72,"2006-02":406.5,"2006-12":492.4,"200
6-05":419.44,"2006-07":427.89,"2005-10":374.75,"2005-01":205.3,"2010-04":597.84,"
2006-11":513,"2009-08":474.35,"2009-10":561.64,"2005-12":446.21,"2007-06":534.99,
"2008-12":329.5,"2008-05":602.45,"2010-01":629.51,"2006-09":418.69,"2010-05":532.
92,"2009-02":381,"2009-12":625.99,"2010-03":588.28,"2007-11":747.24,"2007-01":513
```

Result:

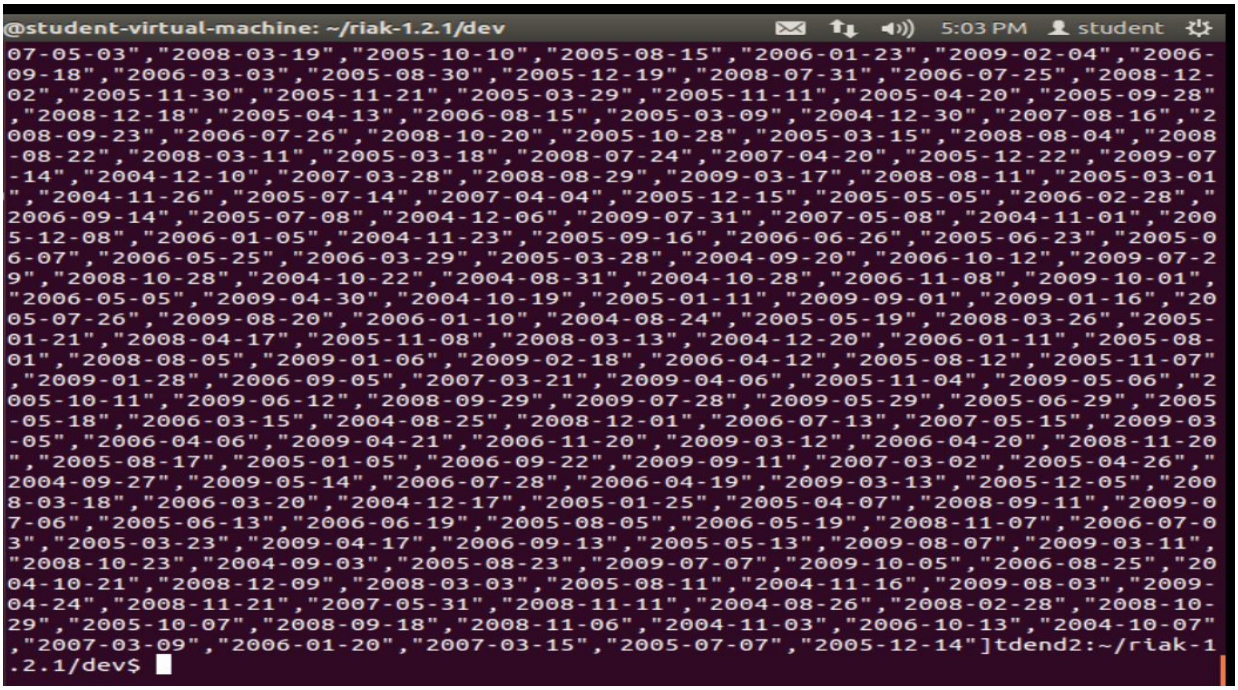


Write and run a map-only query for our Google data that finds all the days where the low was less than $500.00. The map phase function you are going to use is shown below:

Query with first half of the result is provided below:



Last portion of the result of the above query:



Write and run a map-only query for our Google data that finds all of the days where the open was lower than the close. You will need to write your map phase function. You may modify and reuse the one given above.

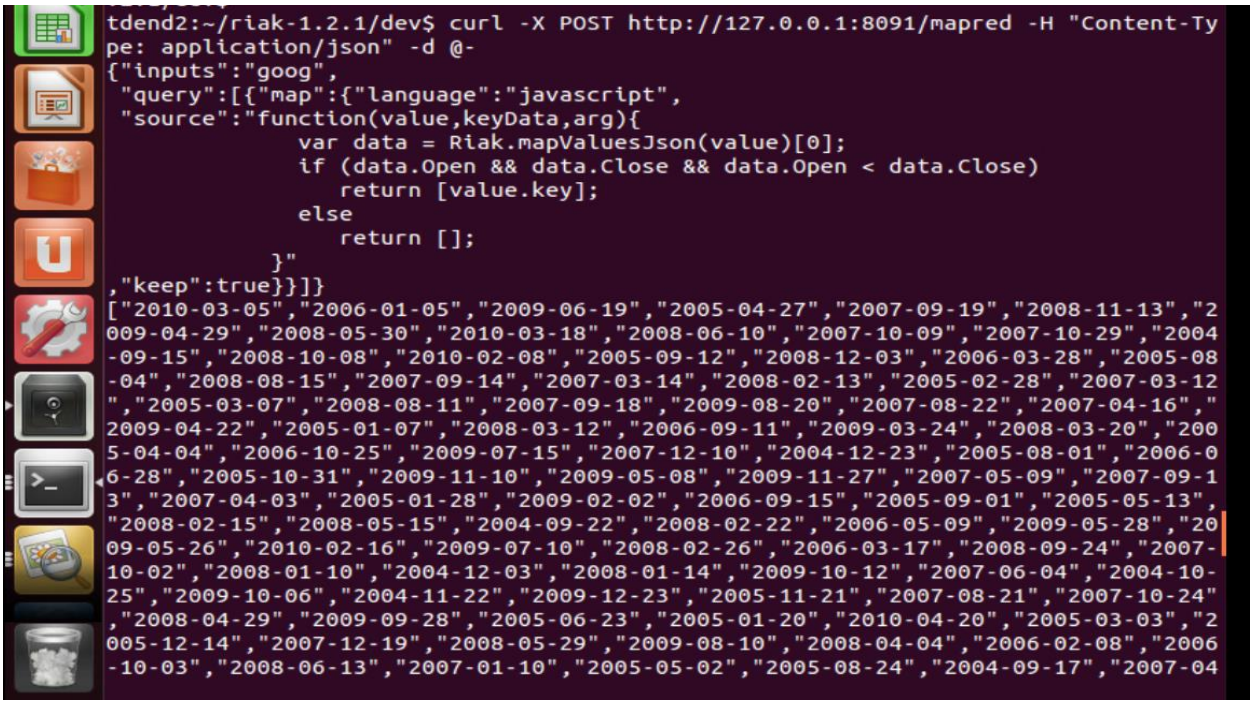
The below function used in the map phase:

```
function(value, keyData, arg) {
        var data = Riak.mapValuesJson(value)[0];
        if (data.Open && data.Close && data.Open < data.Close) {
                return [value.key];
        } else {
                return [];
        }
}
```

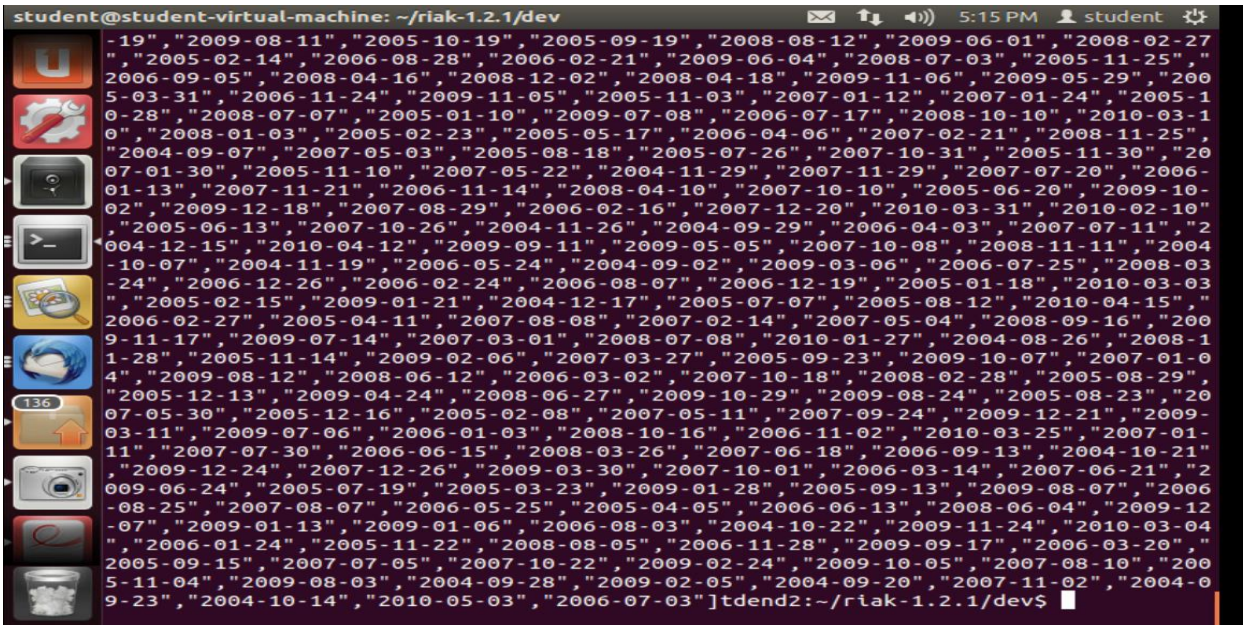
Query and results are contd. In next page

Query and the results:

The query with first half of the result:



Last half of the result:



==========THE END==========