Coursework for CS5607 High Performance Computational Infrastructure

Student ID: 1639420

2016 – 2017, Semester 1

This file contains the report for the project and all lab work for the module CS5607.

Table of Contents

Analysing the distribution of different amino acids in proteins from different locations

Table of Contents

Introduction

Amino acids are biologically important organic compounds containing amine and carboxyl functional groups, along with a side-chain (R group) specific to each amino acid. About 500 amino acids are known, but only 20 appear in the genetic code and can be categorised in many ways. The classification of amino acids is based on the different characteristics such as polarity, pH level, and side chain group type (aliphatic, acyclic, aromatic, containing hydroxyl or sulphur, etc.). In the form of proteins, amino acids compromise the second-largest component of human muscles, cells and other tissues (Wikipedia: Amino Acid).

Proteins are large biomolecules, or macromolecules, consisting of one or more long chains of amino acid residues. Proteins perform a vast array of functions within organisms from one location to another. Proteins differ from one another primarily in their sequence of amino acids. One protein chain contains at least 20-30 amino acid residues. It has been estimated that average-sized bacteria contain about 2 million proteins per cell and human cells on the order of 1 to 3 billion (Wikipedia: Protein).

Considering the scale and complexity of the data, it is simply impossible to study all proteins experimentally. Currently, a lot of computational methods have been developed to analyse the structure, function, and evolution of proteins (Xin et al. 2012, Semih et al. 2014, Rhonda et al. 2016). However, although most of these methods are good for analysing protein sequences individually, they are still not very appropriate for the analysis of large datasets.

In 2015, MapR Technologies, Inc. published a white paper called 'Next Generation Genome Sequencing Using MapR', illustrating the compatibility and efficiency of using high performance computational infrastructure (HPCI) in processing genome sequencing[1].

In fact, HPCI is not only good with the processing of genome sequencing, but also very suitable for analysing other biological data thanks to the parallel processing platform (Aisling et al. 2013, Ibrahim et al. 2015, Rashmi et al. 2016). Herein, the project presents a software prototype for the analysis of the distribution of different amino acids in proteins from different locations based on HPCI, aiming at better understanding the relationship between amino acids and protein functions.

The objectives of the project are:

1. Explore the distribution of different amino acids in proteins from different locations
2. Provide a prototype solution using modern large-scale data storage and processing infrastructures – Hadoop – to enhance the calculation efficiency
3. Discuss the advantages and drawbacks of the solution provided

Methods and Results

Datasets (all converted to UTF-8 coded):
Protein sequences (a part of the data):

[1] MapR Technologies, Inc. *Next Generation Genome Sequencing Using R.* White Paper, 2015.

```
NP_001248143 ,        1 madfddrvsd eekvriaakf ithappgefn evfndvrlll nndnllrega ahafaqynmd
NP_001248143 ,       61 qftpvkiegy edqvlitehg dlgnsrfldp rnkisfkfdh lrkeasdpqp eevdgglksw
NP_001248143 ,      121 rescdsalra yvkdhysngf ctvyaktidg qqtiiacies hqfqpknfwn grwrsewkft
NP_001248143 ,      181 itpptaqvvg vlkiqvhyye dgnvqlvshk dvqdsltvsn eaqtakefik iienaeneyq
NP_001248143 ,      241 taisenyqtm sdttfkalrr qlpvtrtkid wnkilsykig kemqna
NP_004093 ,1 mvdaflgtwk lvdsknfddy mkslgvgfat rqvasmtkpt tiiekngdil tlkthstfkn
NP_004093 ,       61 teisfklgve fdettaddrk vksivtldgg klvhlqkwdg qettlvreli dgkliltlth
NP_004093 ,      121 gtavctrtye kea
NP_006126, 1 madfddrvsd eekvriaakf ithappgefn evfndvrlll nndnllrega ahafaqynmd
NP_006126,        61 qftpvkiegy edqvlitehg dlgnsrfldp rnkisfkfdh lrkeasdpqp eeadgglksw
NP_006126,       121 rescdsalra yvkdhysngf ctvyaktidg qqtiiacies hqfqpknfwn grwrsewkft
NP_006126,       181 itpptaqvvg vlkiqvhyye dgnvqlvshk dvqdsltvsn eaqtakefik iienaeneyq
```

Protein names and locations (a part of the data):

```
NP_001248143 ,Muscle,chromosome 2A,F-actin-capping protein subunit alpha-1 [Macaca mulatta]
NP_004093 ,Muscle,chromosome 3,"fatty acid-binding protein, heart isoform 2 [Homo sapiens]"
NP_006126,Muscle,chromosome 2A, F-actin-capping protein subunit alpha-1 [Homo sapiens]
NP_001223,Muscle,chromosome 1,calsequestrin-2 precursor [Homo sapiens]
NP_004921,Muscle,chromosome 2A,F-actin-capping protein subunit beta isoform 1 [Homo sapiens]
NP_001222,Muscle,chromosome 2B,calsequestrin-1 precursor [Homo sapiens]
NP_001127090,Muscle,chromosome 2A,F-actin-capping protein subunit alpha-1 [Pongo abelii]
NP_001125686,Muscle,chromosome 2B,calsequestrin-2 precursor [Pongo abelii]
NP_001127638,Muscle,chromosome 2A,F-actin-capping protein subunit beta [Pongo abelii]
NP_001193469 ,Muscle,chromosome 2A,F-actin-capping protein subunit beta isoform 2 [Homo sapiens]
NP_005252,Muscle,chromosome 1,GTP-binding protein GEM [Homo sapiens]
NP_859053,Muscle,chromosome 1,GTP-binding protein GEM [Homo sapiens]
NP_001126848,Muscle,chromosome 1,GTP-binding protein GEM [Pongo abelii]
NP_201585,Muscle,chromosome 1,F-actin-capping protein subunit alpha-3 [Homo sapiens]
NP_001233504,Muscle,chromosome 1,F-actin-capping protein subunit alpha-3 [Pan troglodytes]
NP_001001937,Muscle,chromosome 1,"ATP synthase subunit alpha, mitochondrial isoform a precursor [Homo
sapiens]"
NP_001126846 ,Muscle,chromosome 1,"ATP synthase subunit alpha, mitochondrial precursor [Pongo abelii]"
```

Amino acid categories:

```
Alanine,Ala,A,89.079,Aliphatic,Neutral,Non-polar,Hydrophobic,Unecessary
Arginine,Arg,R,174.188,Basic,Basic,Polar,Hydrophilic,Unecessary
Asparagine,Asn,N,132.104,Neutral,Neutral,Polar,Hydrophilic,Unecessary
Asparticacid,Asp,D,133.089,Acidic,Acidic,Polar,Hydrophilic,Unecessary
Cystine,Cys,C,121.145,Surfuric,Neutral,Polar,Hydrophilic,Unecessary
Glutamicacid,Glu,E,146.131,Neutral,Neutral,Polar,Hydrophilic,Unecessary
Glutarnine,Gln,Q,147.116,Acidic,Acidic,Polar,Hydrophilic,Unecessary
Glycine,Gly,G,75.052,Aliphatic,Neutral,Non-polar,Hydrophobic,Unecessary
Histidine,His,H,155.141,Basic,Basic,Polar,Hydrophilic,Unecessary
Isoleucine,Ile,I,131.16,Aliphatic,Neutral,Non-polar,Hydrophobic,necessary
Leucine,Leu,L,131.16,Aliphatic,Neutral,Non-polar,Hydrophobic,necessary
Lysine,Lys,K,146.17,Basic,Basic,Polar,Hydrophilic,necessary
Methionine,Met,M,149.199,Surfuric,Neutral,Polar,Hydrophilic,necessary
Phenylalanine,Phe,F,165.177,Aromatic,Neutral,Non-polar,Hydrophobic,necessary
Proline,Pro,P,115.117,Unique,Neutral,Non-polar,Hydrophobic,Unecessary
Serine,Ser,S,105.078,Hydroxic,Neutral,Polar,Hydrophilic,Unecessary
Threonine,Thr,T,119.105,Hydroxic,Neutral,Polar,Hydrophilic,necessary
Tryptophan,Trp,W,204.213,Aromatic,Neutral,Polar,Hydrophobic,necessary
Tyrosine,Tyr,Y,181.176,Aromatic,Neutral,Polar,Hydrophobic,Unecessary
Valine,Val,V,117.133,Aliphatic,Neutral,Non-polar,Hydrophobic,necessary
```

1. Use map-reduce to count the amino acid in all the proteins in 'Protein sequences'.

Driver:

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class ProteinSeqMod
{
    public static void main(String[] args) throws Exception
    {
      Configuration conf = new Configuration();
      if (args.length != 3)
      {
          System.err.println("Usage: ProteinAACount<input path><output path>");
          System.err.println(args.length);
          for(String s:args) System.out.println(s);
          System.exit(-1);
      }
      Job job;
      job=Job.getInstance(conf, "Protein AA count");
      job.setJarByClass(ProteinSeqMod.class);

      FileInputFormat.addInputPath(job, new Path(args[1]));
      FileOutputFormat.setOutputPath(job, new Path(args[2]));
```

```
        job.setMapperClass(ProteinSeqModMapper.class);
        job.setReducerClass(ProteinSeqModReducer.class);
        job.setCombinerClass(ProteinSeqModReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Mapper:

```
package org.myorg;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class ProteinSeqModMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{
    private final static IntWritable one = new IntWritable(1);
    private Text proteinAA = new Text();

    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException,
    interruptedException
    {
        String[] line = value.toString().split(",");
        String proteinSeq = line[1].toUpperCase().replaceAll(" ", "");
        String proteinSeq1 = proteinSeq.replaceAll("[0-9]", "");
        for(char AminoAcid : proteinSeq1.toCharArray())
        {
            proteinAA.set(line[0].trim()+':'+ AminoAcid);
            context.write(proteinAA, one);
        }
    }
}
```

Reducer:

```
package org.myorg;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class ProteinSeqModReducer extends Reducer<Text, IntWritable, Text, IntWritable>

{
    @Override

    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
    InterruptedException
    {
        int count = 0;
        for (IntWritable value : values)
        {
            count += value.get();
        }
        context.write(key, new IntWritable(count));
    }
}
```

hadoop command line:

```
hadoop@hadoop:~$ hdfs dfs -mkdir /input1
hadoop@hadoop:~$ hdfs dfs -copyFromLocal Downloads/coursework/proteinSeq.txt /input1
```

```
hadoop@hadoop:~$ hadoop jar Downloads/ProteinAA.jar ProteinSeqMod /input1 output2
hadoop@hadoop:~$ hdfs dfs -get /output2
```

(Some of the output)

```
NP_919433:W 4
NP_919433:Y 10
NP_997199:A 9
NP_997199:C 7
NP_997199:D 7
NP_997199:E 6
NP_997199:F 4
NP_997199:G 8
NP_997199:H 3
NP_997199:I 9
NP_997199:K 1
NP_997199:L 21
NP_997199:M 3
NP_997199:N 3
NP_997199:P 13
NP_997199:Q 5
NP_997199:R 7
NP_997199:S 10
NP_997199:T 9
NP_997199:V 13
NP_997199:W 3
NP_997199:Y 5
```

2. Use hadoop streaming to re-format the output (only mapper, no reducer)

Mapper (proteinAAformat.py)**:**

```
#! /usr/bin/env python
import sys
for line in sys.stdin:
print line.replace("\t",":")
```

```
hadoop@hadoop:~$ chmod a+x proteinAAformat.py //allow the program to be executable
hadoop@hadoop:~$ hdfs dfs -mkdir /user/hadoop/input2
hadoop@hadoop:~$ hdfs dfs -copyFromLocal Downloads/courseworkOutput/ProteinAA.txt
/user/hadoop/input2
hadoop@hadoop:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.5.1.jar -
input input2/ProteinAA.txt -output output4 -mapper 'python proteinAAformat.py' -file proteinAAformat.py -
reducer NONE  //use hadoop streaming to run the program, and specify that no reducer is needed for
the current job
hadoop@hadoop:~$ hdfs dfs -get output4
```

(Some of the output)

```
NP_001247455:R:1
NP_001247455:S:5
NP_001247455:T:3
NP_001247455:V:7
NP_001247455:Y:1
NP_001247615:A:5
NP_001247615:D:9
NP_001247615:E:10
NP_001247615:F:8
NP_001247615:G:4
NP_001247615:H:3
NP_001247615:I:5
NP_001247615:K:13
NP_001247615:L:8
NP_001247615:M:2
NP_001247615:N:3
NP_001247615:P:9
NP_001247615:Q:3
NP_001247615:R:8
```

NP_001247615:S:4

3. Use Pig to generate information from the dataset

In the command line:
hadoop@hadoop:~$ hdfs dfs -put Downloads/coursework/ProteinAAMod /
hadoop@hadoop:~$ hdfs dfs -put Downloads/coursework/proteinName.txt /

In the pig script:
```
PAA = LOAD '/ProteinAAMod' USING PigStorage(':') AS (ProteinID:chararray, AAabv:chararray,
Count:int);
pNAME = LOAD '/proteinName.txt' USING PigStorage(',') AS (ProteinID:chararray,
ProPosition:chararray, Chromosome:chararray, ProName:chararray, Species:chararray);
pPos = FOREACH pNAME GENERATE $0, $1;
PosAAjn = JOIN PAA BY (ProteinID), pPos BY (ProteinID);
STORE PosAAjn INTO 'output1';
```

In command line:
hadoop@hadoop:~$ pig -x mapreduce AAinProtein.pig
hadoop@hadoop:~$ hdfs dfs -get output1

(Some of the output)

```
NP_001244700 M 3 NP_001244700 tumor
NP_001244700 P 31 NP_001244700 tumor
NP_001244700 N 3 NP_001244700 tumor
NP_001244700 C 3 NP_001244700 tumor
NP_001244700 A 28 NP_001244700 tumor
NP_001244704 A 24 NP_001244704 unDef
NP_001244704 Y 15 NP_001244704 unDef
NP_001244704 W 4 NP_001244704 unDef
NP_001244704 V 25 NP_001244704 unDef
NP_001244704 T 15 NP_001244704 unDef
NP_001244704 S 28 NP_001244704 unDef
NP_001244704 R 25 NP_001244704 unDef
NP_001244704 Q 10 NP_001244704 unDef
NP_001244704 P 19 NP_001244704 unDef
NP_001244704 N 9 NP_001244704 unDef
NP_001244704 M 9 NP_001244704 unDef
NP_001244704 L 31 NP_001244704 unDef
NP_001244704 K 23 NP_001244704 unDef
NP_001244704 I 28 NP_001244704 unDef
NP_001244704 H 9 NP_001244704 unDef
NP_001244704 G 28 NP_001244704 unDef
NP_001244704 F 20 NP_001244704 unDef
NP_001244704 E 31 NP_001244704 unDef
NP_001244704 D 21 NP_001244704 unDef
NP_001244704 C 2 NP_001244704 unDef
NP_001244710 K 11 NP_001244710 cytoplasm
NP_001244710 V 28 NP_001244710 cytoplasm
NP_001244710 L 51 NP_001244710 cytoplasm
NP_001244710 T 23 NP_001244710 cytoplasm
NP_001244710 M 8 NP_001244710 cytoplasm
NP_001244710 S 30 NP_001244710 cytoplasm
NP_001244710 N 11 NP_001244710 cytoplasm
NP_001244710 P 21 NP_001244710 cytoplasm
NP_001244710 Q 21 NP_001244710 cytoplasm
NP_001244710 R 35 NP_001244710 cytoplasm
NP_001244710 E 23 NP_001244710 cytoplasm
```

(Find out the counts of amino acids in different locations)
```
PosAAjn = LOAD '/PosAAjn' USING PigStorage() AS (ProteinID:chararray, AAabv:chararray, Count:int,
ProteinID2:chararray, ProPosition:chararray);
PosAAgrp = GROUP PosAAjn BY (AAabv, ProPosition);
PosAA = FOREACH PosAAgrp GENERATE group, SUM(PosAAjn.Count);
STORE PosAA INTO 'output2';
```

In command line:

```
hadoop@hadoop:~$ pig -x mapreduce AAinProtein.pig
hadoop@hadoop:~$ hdfs dfs -get output2
```

(Some of the output)

```
(L,Neuron) 38
(L,membrane) 173
(L,ribosome) 72
(L,cytoplasm) 687
(L,cell adhesion) 117
(L,cell division) 19
(L,associated with neuron) 46
(L,endothelial cell growth) 71
(M,Blood) 30
(M,brain) 4
(M,tumor) 11
(M,unDef) 280
(M,Muscle) 58
(M,Neuron) 7
(M,membrane) 40
(M,ribosome) 44
(M,cytoplasm) 130
(M,cell adhesion) 16
(M,cell division) 2
(M,associated with neuron) 7
(M,endothelial cell growth) 11
(N,Blood) 37
(N,brain) 5
(N,tumor) 6
(N,unDef) 446
(N,Muscle) 110
(N,Neuron) 7
(N,membrane) 38
(N,ribosome) 40
(N,cytoplasm) 187
(N,cell adhesion) 68
(N,cell division) 5
(N,associated with neuron) 11
(N,endothelial cell growth) 3
(P,Blood) 97
(P,brain) 1
(P,tumor) 43
(P,unDef) 806
(P,Muscle) 105
(P,Neuron) 29
(P,membrane) 57
```

(Find out the distinct values of the 'location variable')

```
pNAME = LOAD '/proteinName.txt' USING PigStorage(',') AS (ProteinID:chararray,
ProPosition:chararray, Chromosome:chararray, ProName:chararray, Species:chararray);
Pos = FOREACH pNAME GENERATE ProPosition;
DisPos = DISTINCT Pos;
STORE DisPos INTO 'output3';
```

In command line:

```
hadoop@hadoop:~$ pig -x mapreduce AAinProtein.pig
hadoop@hadoop:~$ hdfs dfs -get output3
```

(Results of distinct locations)

```
bone
Blood
brain
tumor
unDef
Muscle
Neuron
nuclear
membrane
```

ribosome
cytoplasm
signaling
fibroblast
DNA upstream
cell adhesion
cell division
immune system
cell carcinoma
fibroblast growth
transcript factor
cytoplasmic network
Cancer transmembrane
associated with neuron
endothelial cell growth

(Find out the amino acid counts in different locations with Filter Function)

```
PosAAjn = LOAD '/PosAAjn' USING PigStorage() AS (ProteinID:chararray, AAabv:chararray, Count:int,
ProteinID2:chararray, ProPosition:chararray);
Filtered = FILTER PosAAjn BY ProPosition == 'Blood';
Filtgrp = GROUP Filtered BY AAabv;
FiltAA = FOREACH Filtgrp GENERATE group, SUM(Filtered.Count), AVG(Filtered.Count),
Max(Filtered.Count), Min(Filtered.Count);
STORE FiltAA INTO 'output4';
```

In command line:

```
hadoop@hadoop:~$ pig -x mapreduce AAinProtein.pig
hadoop@hadoop:~$ hdfs dfs -get output4
```

(Amino acid counts in 'Blood')
//Abv. TotalCounts AvgCounts MaxCounts MinCounts
A 118 29.5 60 6
C 25 6.25 11 4
D 75 18.75 34 1
E 98 24.5 50 2
F 86 21.5 57 3
G 124 31.0 63 5
H 39 13.0 16 7
I 78 19.5 47 2
K 79 19.75 30 4
L 229 57.25 132 10
M 30 7.5 19 2
N 37 9.25 13 2
P 97 32.333333333333336 54 11
Q 59 19.666666666666668 35 9
R 84 28.0 45 11
S 125 31.25 56 3
T 77 19.25 37 7
V 136 34.0 71 3
W 26 6.5 11 2
Y 34 8.5 17 4

(A similar program was run for 'Muscle' and 'tumor')

(Amino acid counts in 'Muscle')
//Abv. TotalCounts AvgCounts MaxCounts MinCounts
A 329 23.5 58 13
C 67 5.153846153846154 10 2
D 391 27.928571428571427 54 15
E 396 28.285714285714285 56 15
F 191 13.642857142857142 27 5
G 282 20.142857142857142 51 11
H 111 7.928571428571429 13 4
I 292 20.857142857142858 41 12
K 327 23.357142857142858 32 18
L 466 33.285714285714285 54 17
M 106 7.571428571428571 12 4
N 198 14.142857142857142 18 11

P 161 11.5 20 6
Q 212 15.142857142857142 25 8
R 278 19.857142857142858 38 9
S 324 23.142857142857142 40 16
T 200 14.285714285714286 27 7
V 361 25.785714285714285 46 17
W 58 4.833333333333333 5 4
Y 153 10.928571428571429 16 7

(Amino acid counts in 'tumor')
//Abv. TotalCounts AvgCounts MaxCounts MinCounts
A 177 35.4 79 15
C 16 3.2 7 1
D 74 14.8 25 7
E 146 29.2 66 7
F 55 11.0 20 8
G 101 20.2 37 12
H 34 6.8 14 2
I 65 13.0 26 5
K 93 18.6 33 3
L 187 37.4 89 20
M 39 7.8 18 3
N 37 7.4 17 1
P 104 20.8 37 5
Q 79 15.8 37 6
R 112 22.4 46 10
S 102 20.4 37 12
T 56 11.2 24 6
V 91 18.2 35 9
W 15 3.75 7 2
Y 33 6.6 12 3

(Find out other statistic information (e.g. quantiles) about the amino acids in certain location, because the amount of data for the known locations in this dataset is too small, herein, the 'unDef' locations data is used as a demo.)

In Pig script:

```
PosAAjn = LOAD '/PosAAjn' USING PigStorage() AS (ProteinID:chararray, AAabv:chararray, Count:int,
ProteinID2:chararray, ProPosition:chararray);
Filtered = FILTER PosAAjn BY ProPosition == 'unDef';
FilteredAA = FOREACH Filtered GENERATE AAabv, Count;
STORE FiteredAA INTO ‘output4’;

register /home/hadoop/Downloads/datafu-1.2.0.jar;
define Quantile datafu.pig.stats.StreamingQuantile('0.0','0.25','0.5','0.75','1.0');
FilteredAA = LOAD ‘/FilteredAA’ USING PigStorage() AS (AAabv:chararray, Count:int);
sorted = ORDER FilteredAA BY AAabv, Count ASC;
Filtgrp = GROUP sorted BY AAabv;
FiltAA = FOREACH Filtgrp GENERATE group, Quantile(sorted.Count);
STORE FiltAA INTO ‘output5’
```

In command line:

```
hadoop@hadoop:~$ pig -x mapreduce AAinProtein.pig
hadoop@hadoop:~$ hdfs dfs -get output5
```

(Amino acid counts for the ‘unDef’ (undefined locations) proteins)
//AAabv 0.0th 0.25th 0.5th 0.75th 1.00th
A 1 11 18 32 94
C 1 3 7 12 34
D 3 8.75 16 23 78
E 2 14 19 36 180
F 2 7 12 19 54
G 4 9 17 31 82
H 1 3.5 8 13 30
I 1 8.75 12.5 21 63
K 1 10 16 26.5 116
L 5 15.75 25 43170
M 1 4 7 11 28

N 1 6 10 16.5 55
P 1 9 14 29 112
Q 1 7.75 12 19 143
R 1 9 15 30 120
S 1 13.5 19 35 116
T 3 10 14 23 65
V 2 10.5 17 28.5 80
W 1 2 3 5 12
Y 1 5.5 9 15 31

To draw a 'counts vs amino acid' box-plot, the inter-quartile range and the exteriors should be found out.

Pig script:

```
FilteredAA = LOAD '/FilteredAA' USING PigStorage() AS (AAabv:chararray, Count:int);
unDefAA = LOAD '/FiltAA' USING PigStorage() AS (AAabv:chararray, 0th:double, 25th:double,
50th:double, 75th:double, 100th:double);
IQR = unDefAA.75th - unDefAA.25th;
Filter1 = unDefAA.25th – 1.5*IQR;
Filter2 = unDefAA.75th + 1.5*IQR;
Low = FILTER FilteredAA BY Count < Filter1;
Lowgrp = GROUP Low by AAabv;
LowC = FOREACH Lowgrp GENERATE group, Low.Count;
STORE LowC INTO 'output1';

High = FILTER FilteredAA BY Count > Filter2;
Highgrp = GROUP High by AAabv;
HighC = FOREACH Highgrp GENERATE group, High.Count;
STORE HighC INTO 'output2';

Mid = FILTER FilteredAA BY (Count >= Filter1 and Count <= Filter2);
Midgrp = GROUP Mid BY AAabv;
Boundary = FOREACH Midgrp GENERATE group, MAX(Mid.Count), MIN(Mid.Count);
STORE Boundary INTO 'output3';
```

(Low)
(NO RECORD)

(High)
A 66 67 77 94
C 27 27 32 34
D 48 51 57 75 78
E 78 87 95 101 117 180
F 48 54
G 72 72 82
H 28 30
I 44 45 52 63
K 59 63 64 80 82 84 116
L 86 88 94 96 103 109 170
M 22 27 28
N 34 36 39 44 44 45 55
P 70 112
Q 40 45 50 54 56 65 76 98 143
R 120
S 71 76 77 79 83 91 95 110 116
T 52 56 56 56 59 65
V 53 56 56 80
W 10 11 12
Y 31

(Boundary)
A 1 63
C 1 23
D 3 44
E 2 68
F 2 31
G 4 63
H 1 26

I 1 38
K 1 46
L 5 81
M 1 21
N 1 30
P 1 59
Q 1 34
R 1 59
S 1 60
T 3 42
V 2 52
W 1 8
Y 1 27

The above information is sufficient enough to for making a box-plot. The method of making a boxplot with R will be shown in the following section.

Now we have got the table containing all the amino acid information in different locations, we also have the table of amino acid categories. In order to get the information of the number of different categorized amino acids in different locations, we can use "JOIN" "FILTER" "GROUP" "FILTER" Pig operations with PigStats.

Take the "different amino acid charge in blood" by example, the Pig script is written as follows:

```
PosAAjn = LOAD '/PosAAjn' USING PigStorage() AS (ProteinID:chararray, AAabv:chararray, Count:int,
ProteinID2:chararray, ProPosition:chararray);
Filtered = FILTER PosAAjn BY ProPosition == 'Blood';
FilteredAA = FOREACH Filtered GENERATE AAabv, Count;
STORE FilteredAA INTO 'output1';

AA = LOAD '/AminoAcid.txt' USING PigStorage(',') AS (AAname:chararray, AAshort:chararray,
AAabv:chararray, AAmw:float, AAcharacter:chararray, AAcharge:chararray, AApolar:chararray,
AAhydro:chararray, AAnecessary:chararray);
bld = LOAD '/bld' USING PigStorage() AS (AAabv:chararray, count: int);
charge = FOREACH AA GENERATE AAabv, AAcharge;
AAbldjn = JOIN charge BY (AAabv) FULL OUTER JOIN, bld BY (AAabv);
STORE AAbldjn INTO 'output2';

AAbldjn = LOAD '/AAbldjn' USING PigStorage() AS (AAabv1:chararray, AAcharge:chararray,
AAabv2:chararray, count:int);
avg = FOREACH (GROUP AAbldjn BY AAcharge) GENERATE group, AVG(AAbldjn.count),
stderr(AAbldjn.count), COUNT(AAbldjn.count);
STORE avg INTO 'output3';
```

Similar code was used for the counts of hydrophilia and necessity of the amino acid in all data (no filtering), blood, and muscle. The results have been summarized below:

Average counts of amino acids with different charges									
Charge	All data			Blood			Muscle		
	AVG	Number of data	STD	AVG	Number of data	STD	AVG	Number of data	STD
Acidic	22.4	314	24.48	19.14	7	12.52	21.54	28	12
Basic	21.58	463	24.51	20.2	10	12.58	17.05	42	9.35
Neutral	22.67	2325	29.76	22.37	59	23.92	17.31	207	11.85

Average counts of amino acids with different hydrophilia									
Hydrophilia	All data			Blood			Muscle		
	AVG	Number	STD	AVG	Number	STD	AVG	Number	STD
Hydrophilic	21.59	1709	27.87	17.76	41	14.23	17.06	153	10.76
Hydrophobic	23.57	1393	29.57	26.51	35	27.7	18.49	124	12.46

Average counts of amino acids with different necessity in different locations									
necessity	All data			Blood			Muscle		
	AVG	Number	STD	AVG	Number	STD	AVG	Number	STD
Necessary	22.1	1237	28.8	23.16	32	27.13	18.19	110	11.61
Unnecessary	22.74	1865	28.35	20.8	44	17.2	17.38	167	11.55

The above generated information can be used for t-test in R (take 'All data' 'Hydrophilia' for example):

```
All_Hphil_n <- 1709
All_Hphil_mean <- 21.59
All_Hphil_sd <- 27.87

All_Hpho_n <- 1393
All_Hpho_mean <- 23.57
All_Hpho_sd <- 29.57

s <- sqrt( ((All_Hphil_n - 1)*(All_Hphil_sd^2) + (All_Hpho_n - 1)*(All_Hpho_sd^2)) / (All_Hphil_n +
All_Hpho_n - 2))
t <- (All_Hphil_mean – All_Hpho_mean) / (s*sqrt((1/All_Hphil_n) + (1/All_Hpho_n)))
p.value <- 1 – pt(t, df = All_Hphil_n + All_Hpho_n - 2)
print(p.value)
```

Result: 0.9721 #indicating that there are no statistically significant difference between hydrophilic and hydrophobic amino acids in all the protein data based on the current dataset

Box-plot related to the number of different charged amino acids in different locations can be plotted based on the quantiles information generated in the following Pig script (still take the 'Blood', 'charge' for example).

```
register /home/hadoop/Downloads/datafu-1.2.0.jar;
define Quantile datafu.pig.stats.StreamingQuantile('0.0','0.25','0.5','0.75','1.0');
AAbldjn = LOAD '/AAbldjn' USING PigStorage() AS (AAabv1:chararray, AAcharge:chararray,
AAabv2:chararray, count:int);
bldcharge = FOREACH AAbldjn GENERATE AAcharge,count;
sorted = ORDER bldcharge BY AAcharge,count ASC;
quantiles = FOREACH (GROUP sorted BY AAcharge) GENERATE group, Quantile(sorted.count);
STORE quantiles INTO 'output1';
```

Results:
Acidic 1 12 17 28.5 35
Basic 4 12.25 16 28.75 45
Neutral 2 5 13 31.5 132

```
AAbldjn = LOAD '/AAbldjn' USING PigStorage() AS (AAabv1:chararray, AAcharge:chararray,
AAabv2:chararray, Count:int);
quantiles = LOAD '/BldQtil' USING PigStorage() AS (AAabv:chararray, 0th:double, 25th:double,
50th:double, 75th:double, 100th:double);
IQR = quantiles.75th – quantiles.25th;
Filter1 = quantiles.25th – 1.5*IQR;
Filter2 = quantiles.75th + 1.5*IQR;

Low = FILTER AAbldjn BY Count < Filter1;
Lowgrp = GROUP Low by AAcharge;
LowC = FOREACH Lowgrp GENERATE group, Low.Count;
STORE LowC INTO 'output1';

High = FILTER AAbldjn BY Count > Filter2;
Highgrp = GROUP High by AAcharge;
HighC = FOREACH Highgrp GENERATE group, High.Count;
STORE HighC INTO 'output2';

Mid = FILTER AAbldjn BY (Count >= Filter1 and Count <= Filter2);
Midgrp = GROUP Mid BY AAcharge;
Boundary = FOREACH Midgrp GENERATE group, MAX(Mid.Count), MIN(Mid.Count);
STORE Boundary INTO 'output3';
```

Results:
(Low)
NA

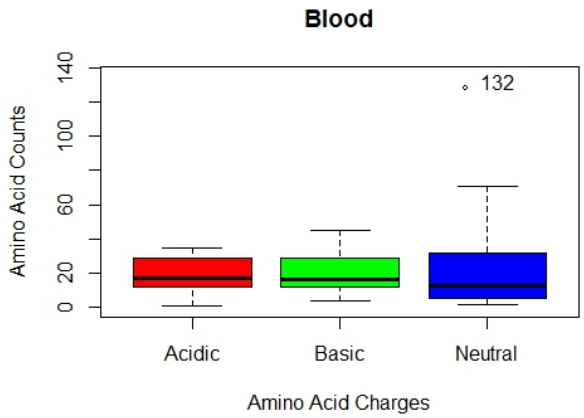
(High)
Neutral 132

(Boundary)
Acidic 1 35
Basic 4 45
Neutral 2 71

We then use the above information to draw the box-plot in R.

```
a <- c(1,12,17,28.5,35) # Quantiles with the 0th and 100th replaced by the boundary values
b <- c(4,12.25,16,28.75,45)
c <- c(2,5,13,31.5,71) # The maximum value here is 71, because 132 has been identified as the outlier in the data, and 71 is the boundary value calculated in the previous step
x <- cbind(a,b,c)
boxplot(x, names=c("Acidic","Basic","Neutral"),main="Blood",xlab="Amino Acid Charges", ylab="Amino Acid Counts",ylim=range(0:135),col=rainbow(3))
text(3,132,labels="。132") # add the outlier point into the plot
```

Results:



Discussion

The presented software prototype first utilized the map-reduce framework built in Hadoop to tide-up the data sheet. 'ProteinID, Amino Acid' pair was returned as the key, and the total counts of the relevant amino acid was returned as the value.

The second step was to use Pig for the operations between different datasets. Information of different amino acid counts in different locations was generated, and PigStats was used for generating the statistic information of the data.

Based on different filtering conditions, the average, standard error, count, quantiles values were generated. The information was intepreted in a statistic way using R. In the current prototype, the methods for doing t-test and making box-plot were shown.

The current data sheet contains only less than 200 proteins, and all the proteins are coded by the 20 well-known amino acids. But in the nature, there are actually a small portion of proteins coded by other types of amino acids. For future reference, to identify whether there are any other amino acids in the dataset is not hard - the 'DISTINCT' function in Pig could easily solve it. Meanwhile, the classification information of those amino acids should be added into the dataset.

Noticing when using Pig, especially PigStats, in the current work, it sometimes took a long time (several minutes) for one operation to be finished. It is because in the current work, only one machine was used (single datanode). Therefore, the advantage of using Hadoop is not that obvious. If in the future, datasets containing the information of all the known proteins were generated, and more datanodes were utilized, this software could work much better and the results would be much more valuable.

Conclusion

This work presented a software prototype in using HPCI for analysing the distribution of different amino acids in proteins from different locations. The dataset has three different data sheets. Methods including map-reduce using Java, Hadoop streaming, and Pig were used to generate useful information contained in the dataset. R was used to do t-test and make box-plot for the generated information in order to statistically understand the data better.

The author hopes that the presented work could be useful for the development and research in the biological field.

Reference

Aisling O' Driscoll, Jurate Daugelaite, Roy D. Sleator. 'Big data', *Hadoop and cloud computing in genomics.* Journal of Biomedical Informatics, 2013, **46**, 774-781.

Chuck Lam. *Hadoop in Action.* Manning Publications Co., 2011.

Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan. *The rise of "big data" on cloud computing: Review and open research issues.* Information Systems, 2015, **47**, 98-115.

Rashmi Tripathi, Pawan Sharma, Pavan Chakraborty & Pritish Kumar Varadwaj. *Next-generation sequencing revolution through big data analytics.* Frontiers in Life Science, 2016, 2155-3777.

Rhonda Bacher and Christina Kendziorski. *Design and computational analysis of single-cell RNA-sequencing experiments.* Genome Biology, 2016, **17**, 63

Semih Ekimler and Kaniye Sahin. *Computational Methods for MicroRNA Target Prediction.* Genes, 2014, **5**, 671-683

Tom White. *Hadoop, the Definitive Guide (4thEdition).* O'Reilly Media Inc., 2015.

Xin Victoria Wang, Natalie Blades, Jie Ding, Razvan Sultana and Giovanni Parmigiani. *Estimation of sequencing error rates in short reads.* BMC Bioinformatics, 2012, **13**, 185.

Lab 1. SQL

Task 1.3

CREATE DATABASE UIS

-- Database: `uis`

```
CREATE DATABASE UIS
```

[Edit inline] [Edit] [Create PHP code

Task 1.4

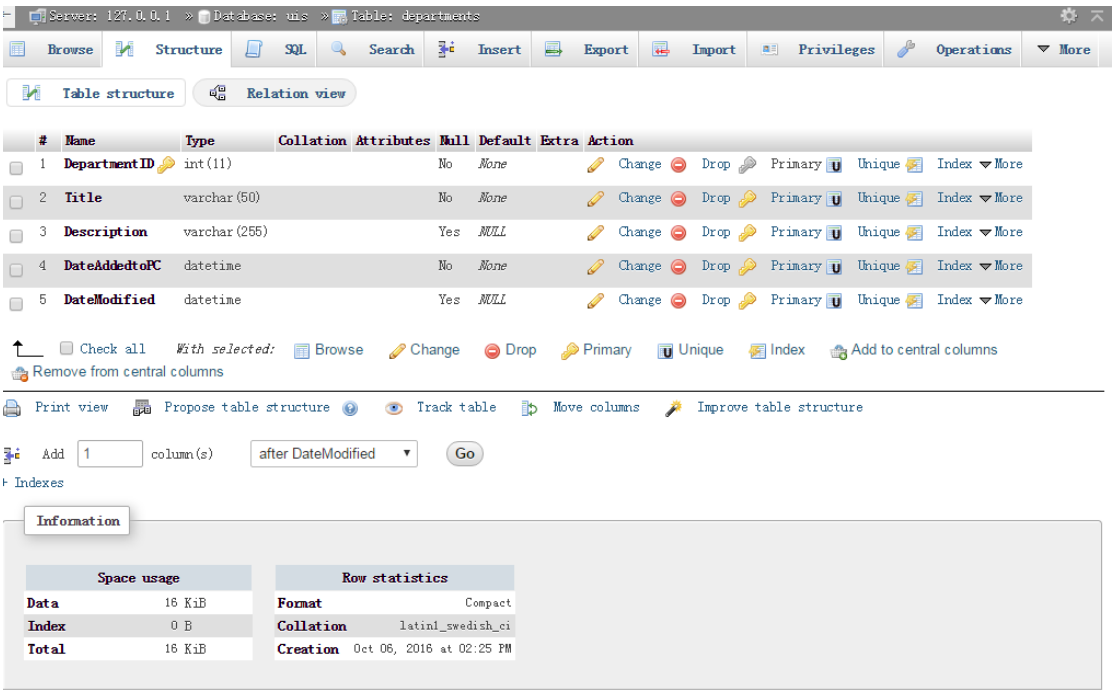
USE UIS

```
USE UIS
```

[Edit inline] [Edit] [Create PHP code]

Task 1.5

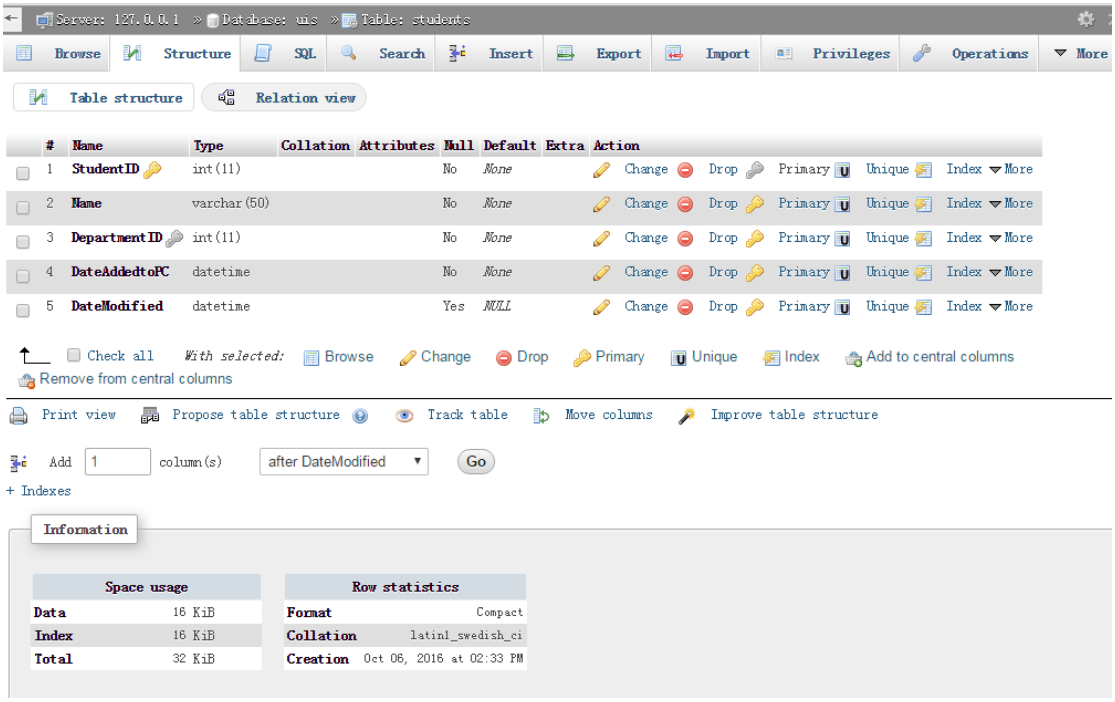
```
CREATE TABLE Departments
(
   DepartmentID INT NOT NULL PRIMARY KEY,
   Title VARCHAR(50) NOT NULL,
   Description VARCHAR(255),
   DateAddedtoPC DATETIME NOT NULL,
   DateModified DATETIME)
```



Task 1.6

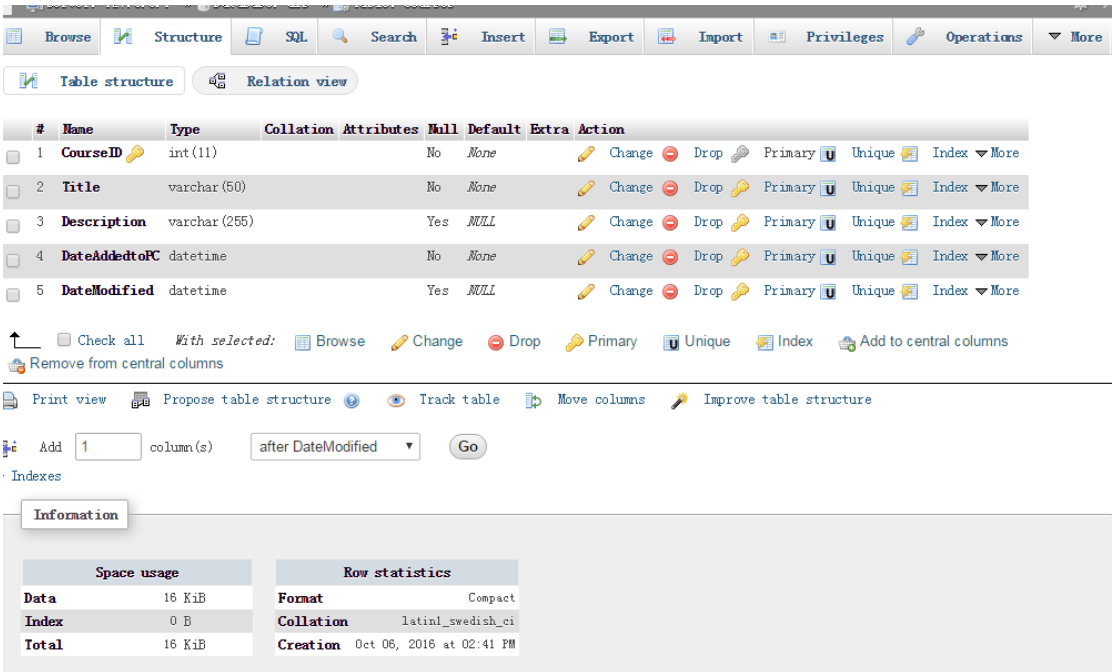
```
USE UIS;

CREATE TABLE Students(
   StudentID INT NOT NULL,
   Name VARCHAR(50) NOT NULL,
   DepartmentID INT NOT NULL,
   DateAddedtoPC DATETIME NOT NULL,
   DateModified DATETIME,
   PRIMARY KEY(StudentID),
   FOREIGN KEY(DepartmentID) REFERENCES Departments(DepartmentID)
```

)



USE UIS;

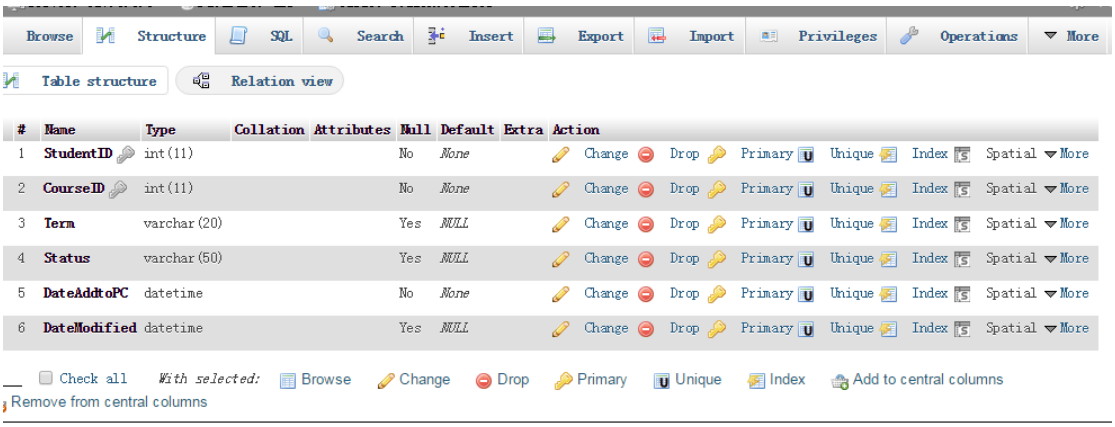
CREATE TABLE Courses(
CourseID INT NOT NULL PRIMARY KEY,
Title VARCHAR(50) NOT NULL,
Description VARCHAR(255),
DateAddedtoPC DATETIME NOT NULL,
DateModified DATETIME)



USE UIS;

CREATE TABLE Studentcourses(
StudentID INT NOT NULL,
CourseID INT NOT NULL,
Term VARCHAR(20),

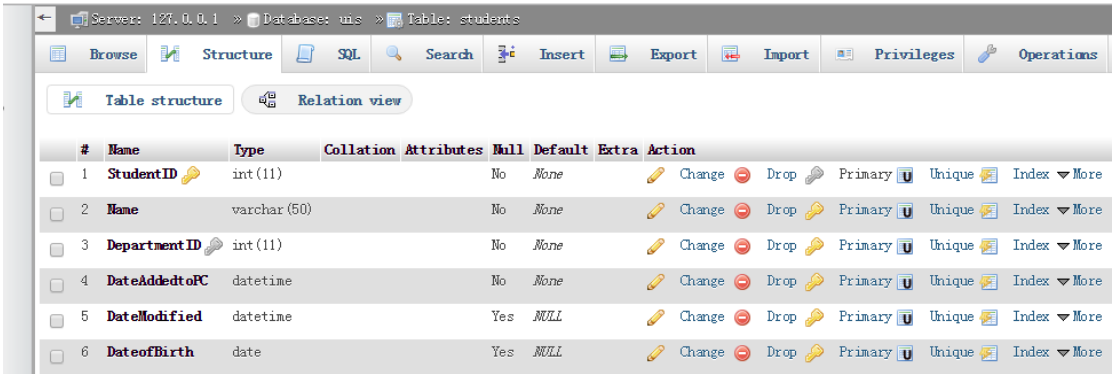
```
Status VARCHAR (50),
DateAddtoPC DATETIME NOT NULL,
DateModified DATETIME,
FOREIGN KEY(StudentID) REFERENCES Students(StudentID),
FOREIGN KEY(CourseID) REFERENCES Courses(CourseID)
)
```



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	StudentID	int(11)			No	None		Change Drop Primary Unique Index Spatial More
2	CourseID	int(11)			No	None		Change Drop Primary Unique Index Spatial More
3	Term	varchar(20)			Yes	NULL		Change Drop Primary Unique Index Spatial More
4	Status	varchar(50)			Yes	NULL		Change Drop Primary Unique Index Spatial More
5	DateAddtoPC	datetime			No	None		Change Drop Primary Unique Index Spatial More
6	DateModified	datetime			Yes	NULL		Change Drop Primary Unique Index Spatial More

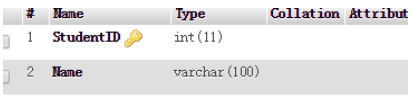
Task 1.7

```
USE UIS;
ALTER TABLE Students
ADD DateofBirth DATE
```



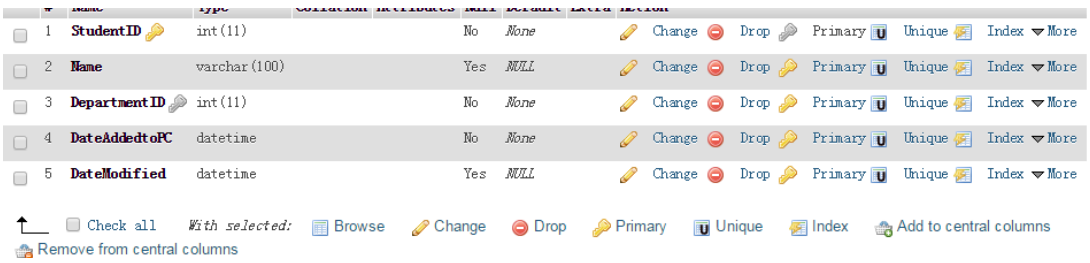
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	StudentID	int(11)			No	None		Change Drop Primary Unique Index More
2	Name	varchar(50)			No	None		Change Drop Primary Unique Index More
3	DepartmentID	int(11)			No	None		Change Drop Primary Unique Index More
4	DateAddedtoPC	datetime			No	None		Change Drop Primary Unique Index More
5	DateModified	datetime			Yes	NULL		Change Drop Primary Unique Index More
6	DateofBirth	date			Yes	NULL		Change Drop Primary Unique Index More

```
USE UIS;
ALTER TABLE Students
MODIFY COLUMN Name VARCHAR(100)
```



#	Name	Type	Collation	Attribut
1	StudentID	int(11)		
2	Name	varchar(100)		

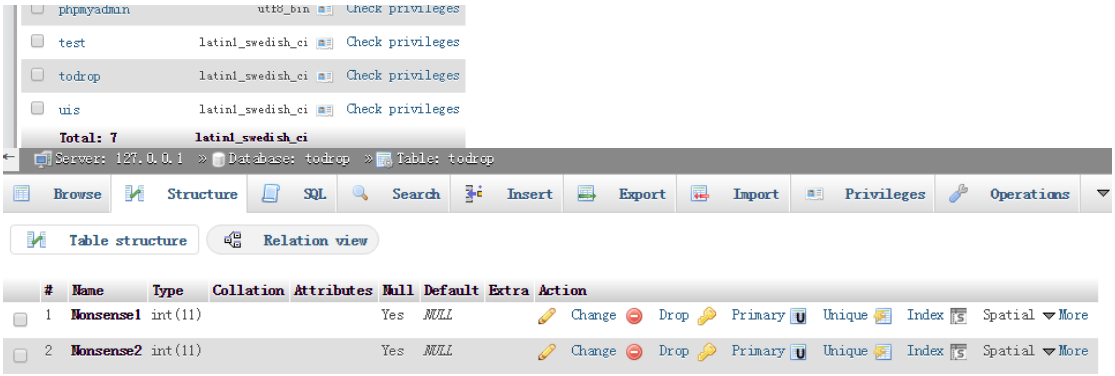
```
USE UIS;
ALTER TABLE Students
DROP COLUMN DateofBirth
```



#	Name	Type	Null	Default	Action
1	StudentID	int(11)	No	None	Change Drop Primary Unique Index More
2	Name	varchar(100)	Yes	NULL	Change Drop Primary Unique Index More
3	DepartmentID	int(11)	No	None	Change Drop Primary Unique Index More
4	DateAddedtoPC	datetime	No	None	Change Drop Primary Unique Index More
5	DateModified	datetime	Yes	NULL	Change Drop Primary Unique Index More

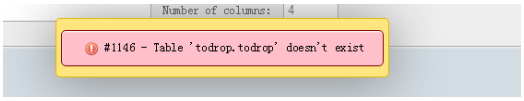
Task 1.8

CREATE DATABASE Todrop;
USE Todrop;
CREATE TABLE Todrop
(Nonsense1 INT,
Nonsense2 INT)



USE Todrop;

DROP TABLE Todrop



DROP DATABASE Todrop

Task 1.9

USE UIS;
INSERT INTO Departments
VALUES(1,"Computer Science","Offers degrees in computer science","2014-10-01",NULL)
|1|Computer Science|Offers degrees in computer science|2014-10-01 00:00:00|NULL

USE UIS;
INSERT INTO Departments(DepartmentID, Title, Description, DateAddedtoPC, DateModified)
VALUES (2, "Mathematics", 'Offers degrees in Mathematics', '2014-10-1', NULL)
|2|Mathematics|Offers degrees in Mathematics|2014-10-01 00:00:00|NULL

USE UIS;
INSERT INTO Departments(DepartmentID,Title,DateAddedtoPC)
VALUES(3, "Information Systems","2014-10-01")
|3|Information Systems|NULL|2014-10-01 00:00:00|NULL

Task 1.10, Task 1.11

USE UIS;

UPDATE Students

SET DepartmentID=2;

DELETE FROM Students

Task 1.12

(Some of the output)

Courses (84 in total, 15 are shown here):

|1|Introductory Programming|NULL|2011-01-17 00:00:00|2013-03-18 00:00:00
|2|Data and Information|NULL|2011-01-17 00:00:00|2014-05-13 00:00:00
|3|Information Systems and Organisations|NULL|2011-01-17 00:00:00|2014-05-13 00:00:00
|4|Logic and Computation|NULL|2011-01-17 00:00:00|2012-07-03 00:00:00
|5|Software Development and Management|NULL|2011-01-17 00:00:00|2012-07-03 00:00:00
|6|Usability Engineering|NULL|2011-01-17 00:00:00|2013-03-18 00:00:00
|7|Algorithms and their Applications|NULL|2011-01-17 00:00:00|NULL
|8|Networks and Operating Systems|NULL|2011-01-17 00:00:00|NULL
|9|Final Year Project in Artificial Intelligence|NULL|2011-01-17 00:00:00|2011-08-15 00:00:00
|10|Software Project Management|NULL|2011-01-17 00:00:00|2011-08-16 00:00:00
|11|Advanced Topics in Computer Science|NULL|2011-01-17 00:00:00|NULL
|12|Artificial Intelligence|NULL|2011-01-17 00:00:00|2012-07-03 00:00:00
|13|Software Engineering|NULL|2011-01-17 00:00:00|2012-07-03 00:00:00
|14|Network Computing|NULL|2011-01-17 00:00:00|NULL
|15|Digital Media and Games|NULL|2011-01-17 00:00:00|NULL

Departments (13 in total, 6 are shown here):

|1|Computer Science|The Department of Computer Science is home to a vibrant and talented community of academics, researchers and students.?Recognised for high quality teaching and research, we attract staff and students from all over the world.|2011-01-17 00:00:00|2013-03-18 00:00:00

|2|Design|We believe that good design is the combination of commercial awareness and creative and inspirational thought validated by sound technological reasoning, defined through the design process. We produce communicators who are at ease working with members of |2011-01-17 00:00:00|2014-05-13 00:00:00

|3|Electronic and Computer Engineering|Electronic and Computer Engineering (ECE) at Brunel is one of the largest disciplines in the University with almost 50 full-time academic staff and extensive teaching and research portfolios.|2011-01-17 00:00:00|2014-05-13 00:00:00

|4|Mathematics|The Department of Mathematical Sciences is committed to excellence in research and teaching. We are a vibrant and friendly department for undergraduate, postgraduate and research students with a well established reputation for student achievement and suc|2011-01-17 00:00:00|2012-07-03 00:00:00

|5|Mechanical, Aerospace and Civil Engineering|The Department of Mechanical, Aerospace and Civil Engineering brings together the disciplines of Advanced Manufacturing & Enterprise Engineering (AMEE), one of the first integrated innovative engineering disciplines in the United Kingdom, Mechanical Engin|2011-01-17 00:00:00|2012-07-03 00:00:00

|6|Brunel Business School|Vibrant, innovative, forward-looking and with ambitious plans for the future, Brunel Business School is one of the largest schools at Brunel University, London.|2011-01-17 00:00:00|2013-03-18 00:00:00

Studentcourses (part of the results):

|119339|1|1|R|2011-01-17 00:00:00|2013-03-18 00:00:00
|303346|2|1|R|2011-01-17 00:00:00|2014-05-13 00:00:00
|527084|3|1|R|2011-01-17 00:00:00|2014-05-13 00:00:00
|636079|4|1|R|2011-01-17 00:00:00|2012-07-03 00:00:00
|419573|5|1|R|2011-01-17 00:00:00|2012-07-03 00:00:00
|626516|6|1|R|2011-01-17 00:00:00|2013-03-18 00:00:00
|34180|7|1|R|2011-01-17 00:00:00|NULL
|338919|8|1|R|2011-01-17 00:00:00|NULL
|502925|9|1|R|2011-01-17 00:00:00|2011-08-15 00:00:00
|407115|10|1|R|2011-01-17 00:00:00|2011-08-16 00:00:00
|538479|11|1|R|2011-01-17 00:00:00|NULL
|538620|12|1|R|2011-01-17 00:00:00|2012-07-03 00:00:00
|510605|13|1|R|2011-01-17 00:00:00|2012-07-03 00:00:00

|518939|14|1|R|2011-01-17 00:00:00|NULL
|506279|15|1|R|2011-01-17 00:00:00|NULL
|535174|16|1|R|2011-01-17 00:00:00|NULL
|509083|17|1|R|2011-01-17 00:00:00|NULL
|512424|18|1|R|2011-01-17 00:00:00|2011-08-15 00:00:00
|509308|19|1|R|2011-01-17 00:00:00|NULL
|335483|20|1|R|2011-01-17 00:00:00|NULL
|418076|21|1|R|2011-01-17 00:00:00|NULL
|510170|22|1|R|2011-01-17 00:00:00|NULL
|517026|23|1|R|2011-01-17 00:00:00|NULL
|508053|24|1|R|2011-01-17 00:00:00|2013-11-18 00:00:00
|407761|25|1|R|2011-01-17 00:00:00|2012-07-03 00:00:00

Students (part of the results):

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|320117|CHARLES WATSON|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|407761|JONATHAN SILVER|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|413008|OLUKOTUN ASERU|6|2011-08-15 00:00:00|2012-11-27 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|415318|RAHUL BIST|7|2012-09-12 00:00:00|2012-10-24 00:00:00
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL
|419392|YASSAR CHOUDHRY|7|2012-09-18 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|431701|MAXAMED ABDULKADIR|6|2013-09-18 00:00:00|NULL
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|503045|VIVEK SRILAL|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|504653|AZIM AHMAD|6|2011-01-17 00:00:00|NULL

Lab 2. SQL

Task 2.1

SELECT Title, Description FROM Departments

|Computer Science|The Department of Computer Science is home to a vibrant and talented community of academics, researchers and students.?Recognised for high quality teaching and research, we attract staff and students from all over the world.

|Design|We believe that good design is the combination of commercial awareness and creative and inspirational thought validated by sound technological reasoning, defined through the design process. We produce communicators who are at ease working with members of

|Electronic and Computer Engineering|Electronic and Computer Engineering (ECE) at Brunel is one of the largest disciplines in the University with almost 50 full-time academic staff and extensive teaching and research portfolios.

|Mathematics|The Department of Mathematical Sciences is committed to excellence in research and teaching. We are a vibrant and friendly department for undergraduate, postgraduate and research students with a well established reputation for student achievement and suc

|Mechanical, Aerospace and Civil Engineering|The Department of Mechanical, Aerospace and Civil Engineering brings together the disciplines of Advanced Manufacturing & Enterprise Engineering (AMEE), one of the first integrated innovative engineering disciplines in the United Kingdom, Mechanical Engin

|Brunel Business School|Vibrant, innovative, forward-looking and with ambitious plans for the future, Brunel Business School is one of the largest schools at Brunel University, London.

|Arts and Humanities|The Department of Arts and Humanities covers Theatre, Music, English, and Creative Writing, running undergraduate, postgraduate and research programmes that are designed to sharpen creative and analytical skills, develop confidence in working in teams and

|Economics and Finance|We are one of the ten largest economics departments in the UK with a distinctive focus on integrating Economics, Finance and Accounting. This is reflected in our undergraduate and postgraduate taught programmes. Our research informs industry, government a

|Education|We strive to be the most innovative Education department in London. Based on the oldest teacher training colleges in the British Commonwealth ?each with a radical history - we offer research-led undergraduate and postgraduate programmes for teachers, yout

|Politics, History and the Brunel Law School|The Department of Politics, History and the Brunel Law School is a highly-ranked department, regularly scoring extremely well in league tables for both teaching and research across all its disciplines.

|Social Sciences, Media and Communications|The department is comprised of three divisions: Anthropology, Media (Film and TV, Games Design, Journalism), and Sociology and Communications, offering a range of highly-rated undergraduate and postgraduate programmes delivered by leading researchers in t

|Clinical Sciences|The Department of Clinical Sciences conducts research and teaching in five major subject areas. Occupational Therapy, Physiotherapy, Social Work (incorporating Human Geography and Youth and Community Work), Public Health and Health Promotion, Specialist C

|Life Sciences|The Department of Life Sciences runs courses in Sports Sciences, Biosciences and Psychology. With a strong emphasis on interdisciplinary research, and a commitment to excellence, we aim to push the boundaries of human health and performance.

SELECT * FROM departments

|1|Computer Science|The Department of Computer Science is home to a vibrant and talented community of academics, researchers and students.?Recognised for high quality teaching and research, we attract staff and students from all over the world.|2011-01-17 00:00:00|2013-03-18 00:00:00

|2|Design|We believe that good design is the combination of commercial awareness and creative and inspirational thought validated by sound technological reasoning, defined through the design process. We produce communicators who are at ease working with members of |2011-01-17 00:00:00|2014-05-13 00:00:00

|3|Electronic and Computer Engineering|Electronic and Computer Engineering (ECE) at Brunel is one of the largest disciplines in the University with almost 50 full-time academic staff and extensive teaching and research portfolios.|2011-01-17 00:00:00|2014-05-13 00:00:00

|4|Mathematics|The Department of Mathematical Sciences is committed to excellence in research and teaching. We are a vibrant and friendly department for undergraduate, postgraduate and research students with a well established reputation for student achievement and suc|2011-01-17 00:00:00|2012-07-03 00:00:00

|5|Mechanical, Aerospace and Civil Engineering|The Department of Mechanical, Aerospace and Civil Engineering brings together the disciplines of Advanced Manufacturing & Enterprise Engineering (AMEE), one of the first integrated innovative engineering disciplines in the United Kingdom, Mechanical Engin|2011-01-17 00:00:00|2012-07-03 00:00:00

|6|Brunel Business School|Vibrant, innovative, forward-looking and with ambitious plans for the future, Brunel Business School is one of the largest schools at Brunel University, London.|2011-01-17 00:00:00|2013-03-18 00:00:00

|7|Arts and Humanities|The Department of Arts and Humanities covers Theatre, Music, English, and Creative Writing, running undergraduate, postgraduate and research programmes that are designed to sharpen creative and analytical skills, develop confidence in working in teams and|2011-01-17 00:00:00|NULL

|8|Economics and Finance|We are one of the ten largest economics departments in the UK with a distinctive focus on integrating Economics, Finance and Accounting. This is reflected in our undergraduate and postgraduate taught programmes. Our research informs industry, government a|2011-01-17 00:00:00|NULL

|9|Education|We strive to be the most innovative Education department in London. Based on the oldest teacher training colleges in the British Commonwealth ?each with a radical history - we offer research-led undergraduate and postgraduate programmes for teachers, yout|2011-01-17 00:00:00|2011-08-15 00:00:00

|10|Politics, History and the Brunel Law School|The Department of Politics, History and the Brunel Law School is a highly-ranked department, regularly scoring extremely well in league tables for both teaching and research across all its disciplines.|2011-01-17 00:00:00|2011-08-16 00:00:00

|11|Social Sciences, Media and Communications|The department is comprised of three divisions: Anthropology, Media (Film and TV, Games Design, Journalism), and Sociology and Communications, offering a range of highly-rated undergraduate and postgraduate programmes delivered by leading researchers in t|2011-01-17 00:00:00|NULL

|12|Clinical Sciences|The Department of Clinical Sciences conducts research and teaching in five major subject areas. Occupational Therapy, Physiotherapy, Social Work (incorporating Human Geography and Youth and Community Work), Public Health and Health Promotion, Specialist C|2011-01-17 00:00:00|2012-07-03 00:00:00

|13|Life Sciences|The Department of Life Sciences runs courses in Sports Sciences, Biosciences and Psychology. With a strong emphasis on interdisciplinary research, and a commitment to excellence, we aim to push the boundaries of human health and performance.|2011-01-17 00:00:00|2012-07-03 00:00:00

Task 2.2

SELECT DISTINCT DepartmentID FROM students

|1
|2
|3
|4
|5
|6
|7
|8

Task 2.3

SELECT * FROM students WHERE DepartmentID < 2

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00

|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|527084|UGOCHUKWU IWU|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|626516|AMAN PATEL|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|636079|YASODA JAYAWEERA|1|2011-01-17 00:00:00|2012-07-03 00:00:00

Task 2.4
SELECT * FROM students
WHERE DepartmentID = 2
AND DateAddedtoPC = "2011-01-17"
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00

SELECT * FROM students
WHERE DepartmentID = 2
OR DepartmentID = 7
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|415318|RAHUL BIST|7|2012-09-12 00:00:00|2012-10-24 00:00:00
|419392|YASSAR CHOUDHRY|7|2012-09-18 00:00:00|NULL
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|504966|HELEN BUI|7|2012-09-12 00:00:00|NULL
|509371|DANIEL CHUNG|7|2012-10-24 00:00:00|NULL
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|513524|KIRENDEEP DHINSA|7|2011-01-17 00:00:00|2013-03-18 00:00:00
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|633977|DALAI DOS SANTOS RIBEIRO|7|2011-01-17 00:00:00|2014-05-13 00:00:00

SELECT * FROM students WHERE DateAddedtoPC = "2011-01-17" AND (DepartmentID = 2 OR DepartmentID = 7)

|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|513524|KIRENDEEP DHINSA|7|2011-01-17 00:00:00|2013-03-18 00:00:00
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|633977|DALAI DOS SANTOS RIBEIRO|7|2011-01-17 00:00:00|2014-05-13 00:00:00

Task 2.5
SELECT * FROM students
ORDER BY DepartmentID

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|626516|AMAN PATEL|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|527084|UGOCHUKWU IWU|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|636079|YASODA JAYAWEERA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|518939|AKTHER HUSSAIN|3|2011-01-17 00:00:00|NULL
|509083|RAMEESA KHAN|3|2011-01-17 00:00:00|NULL
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL
|535174|SEYED KASHAN|3|2011-01-17 00:00:00|NULL
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00

|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|517026|ASHIKA RAMJEE|4|2011-01-17 00:00:00|NULL
|512424|HARIS KHAN|4|2011-01-17 00:00:00|2011-08-15 00:00:00
|510170|NAMRATA PUN|4|2011-01-17 00:00:00|NULL
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|510586|KOULMIT SONI|5|2011-01-17 00:00:00|2013-02-15 00:00:00

SELECT * FROM students
ORDER BY DepartmentID DESC

|510400|NEAMAN DOUSHOUKI|8|2011-01-17 00:00:00|2012-07-03 00:00:00
|633977|DALAI DOS SANTOS RIBEIRO|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|513524|KIRENDEEP DHINSA|7|2011-01-17 00:00:00|2013-03-18 00:00:00
|509371|DANIEL CHUNG|7|2012-10-24 00:00:00|NULL
|504966|HELEN BUI|7|2012-09-12 00:00:00|NULL
|419392|YASSAR CHOUDHRY|7|2012-09-18 00:00:00|NULL
|415318|RAHUL BIST|7|2012-09-12 00:00:00|2012-10-24 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|413008|OLUKOTUN ASERU|6|2011-08-15 00:00:00|2012-11-27 00:00:00
|431701|MAXAMED ABDULKADIR|6|2013-09-18 00:00:00|NULL
|633976|DASHA BARG PINTO|6|2011-08-15 00:00:00|2012-07-03 00:00:00
|504653|AZIM AHMAD|6|2011-01-17 00:00:00|NULL
|540742|MUHAMMAD AMMAR ABDULKARIM|6|2013-09-18 00:00:00|NULL
|508734|MUHAMMAD BHATTI|6|2012-09-12 00:00:00|NULL
|513958|BACHIR SOGUI|5|2011-01-17 00:00:00|NULL
|320117|CHARLES WATSON|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|510586|KOULMIT SONI|5|2011-01-17 00:00:00|2013-02-15 00:00:00
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|407761|JONATHAN SILVER|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|503045|VIVEK SRILAL|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|512424|HARIS KHAN|4|2011-01-17 00:00:00|2011-08-15 00:00:00
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL

SELECT * FROM students
ORDER BY DepartmentID,Name

|626516|AMAN PATEL|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|527084|UGOCHUKWU IWU|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|636079|YASODA JAYAWEERA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|518939|AKTHER HUSSAIN|3|2011-01-17 00:00:00|NULL
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL
|509083|RAMEESA KHAN|3|2011-01-17 00:00:00|NULL
|535174|SEYED KASHAN|3|2011-01-17 00:00:00|NULL
|517026|ASHIKA RAMJEE|4|2011-01-17 00:00:00|NULL
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|512424|HARIS KHAN|4|2011-01-17 00:00:00|2011-08-15 00:00:00
|510170|NAMRATA PUN|4|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00

Task 2.6
SELECT * FROM students WHERE Name LIKE "jo%"

|407761|JONATHAN SILVER|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL

SELECT * FROM students WHERE Name LIKE "%s"

|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL

SELECT * FROM students WHERE Name LIKE "%and%"

|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00

SELECT * FROM students WHERE Name NOT LIKE "%and%"

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|320117|CHARLES WATSON|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|407761|JONATHAN SILVER|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|413008|OLUKOTUN ASERU|6|2011-08-15 00:00:00|2012-11-27 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|415318|RAHUL BIST|7|2012-09-12 00:00:00|2012-10-24 00:00:00
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL
|419392|YASSAR CHOUDHRY|7|2012-09-18 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|431701|MAXAMED ABDULKADIR|6|2013-09-18 00:00:00|NULL
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|503045|VIVEK SRILAL|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|504653|AZIM AHMAD|6|2011-01-17 00:00:00|NULL
|504966|HELEN BUI|7|2012-09-12 00:00:00|NULL
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00
|508734|MUHAMMAD BHATTI|6|2012-09-12 00:00:00|NULL
|509083|RAMEESA KHAN|3|2011-01-17 00:00:00|NULL
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|509371|DANIEL CHUNG|7|2012-10-24 00:00:00|NULL
|510170|NAMRATA PUN|4|2011-01-17 00:00:00|NULL

Task 2.7
SELECT * FROM students WHERE DepartmentID IN (1,3)

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL
|509083|RAMEESA KHAN|3|2011-01-17 00:00:00|NULL
|518939|AKTHER HUSSAIN|3|2011-01-17 00:00:00|NULL
|527084|UGOCHUKWU IWU|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|535174|SEYED KASHAN|3|2011-01-17 00:00:00|NULL
|626516|AMAN PATEL|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|636079|YASODA JAYAWEERA|1|2011-01-17 00:00:00|2012-07-03 00:00:00

SELECT * FROM students WHERE DepartmentID NOT IN (1,3)

|320117|CHARLES WATSON|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|407483|ANDREAS VICTOROS|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|407761|JONATHAN SILVER|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|413008|OLUKOTUN ASERU|6|2011-08-15 00:00:00|2012-11-27 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL

|431701|MAXAMED ABDULKADIR|6|2013-09-18 00:00:00|NULL
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|503045|VIVEK SRILAL|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|504653|AZIM AHMAD|6|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00
|508734|MUHAMMAD BHATTI|6|2012-09-12 00:00:00|NULL
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|510170|NAMRATA PUN|4|2011-01-17 00:00:00|NULL
|510586|KOULMIT SONI|5|2011-01-17 00:00:00|2013-02-15 00:00:00
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|512424|HARIS KHAN|4|2011-01-17 00:00:00|2011-08-15 00:00:00
|513958|BACHIR SOGUI|5|2011-01-17 00:00:00|NULL
|517026|ASHIKA RAMJEE|4|2011-01-17 00:00:00|NULL
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|540742|MUHAMMAD AMMAR ABDULKARIM|6|2013-09-18 00:00:00|NULL
|633976|DASHA BARG PINTO|6|2011-08-15 00:00:00|2012-07-03 00:00:00

Task 2.8
SELECT * FROM students WHERE DepartmentID BETWEEN 1 AND 5

|34180|NUWAN SUDASINGHAGE DON|1|2011-01-17 00:00:00|NULL
|119339|TASDEED AZIZ|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|303346|NISHANTH CHANDRADAS|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|320117|CHARLES WATSON|5|2011-01-17 00:00:00|2012-07-03 00:00:00
|335483|DANIEL ONUIGWE|4|2011-01-17 00:00:00|NULL
|338919|MUSA YERO|1|2011-01-17 00:00:00|NULL
|407115|JAMES GLOVER|2|2011-01-17 00:00:00|2011-08-16 00:00:00
|418076|VISHAL PATEL|4|2011-01-17 00:00:00|NULL
|419573|SAMANTHA O'HARA|1|2011-01-17 00:00:00|2012-07-03 00:00:00
|502925|SHAH ELAHI|2|2011-01-17 00:00:00|2011-08-15 00:00:00
|506279|JOANNA JOSS|3|2011-01-17 00:00:00|NULL
|508053|SANA SALEEM|4|2011-01-17 00:00:00|2013-11-18 00:00:00
|509083|RAMEESA KHAN|3|2011-01-17 00:00:00|NULL
|509308|SIDRA KHAN|4|2011-01-17 00:00:00|NULL
|510170|NAMRATA PUN|4|2011-01-17 00:00:00|NULL
|510605|PATRICK HARGAN|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|512424|HARIS KHAN|4|2011-01-17 00:00:00|2011-08-15 00:00:00
|517026|ASHIKA RAMJEE|4|2011-01-17 00:00:00|NULL
|518939|AKTHER HUSSAIN|3|2011-01-17 00:00:00|NULL
|527084|UGOCHUKWU IWU|1|2011-01-17 00:00:00|2014-05-13 00:00:00
|535174|SEYED KASHAN|3|2011-01-17 00:00:00|NULL
|538479|HAJI HAJI ISHAK|2|2011-01-17 00:00:00|NULL
|538620|HAMZA HAMEED|2|2011-01-17 00:00:00|2012-07-03 00:00:00
|626516|AMAN PATEL|1|2011-01-17 00:00:00|2013-03-18 00:00:00
|636079|YASODA JAYAWEERA|1|2011-01-17 00:00:00|2012-07-03 00:00:00

SELECT * FROM students WHERE DepartmentID NOT BETWEEN 1 AND 5

|413008|OLUKOTUN ASERU|6|2011-08-15 00:00:00|2012-11-27 00:00:00
|413554|THEOPHILUS DOLOR|7|2011-01-17 00:00:00|2014-05-13 00:00:00
|415318|RAHUL BIST|7|2012-09-12 00:00:00|2012-10-24 00:00:00
|419392|YASSAR CHOUDHRY|7|2012-09-18 00:00:00|NULL
|431701|MAXAMED ABDULKADIR|6|2013-09-18 00:00:00|NULL
|504653|AZIM AHMAD|6|2011-01-17 00:00:00|NULL
|504966|HELEN BUI|7|2012-09-12 00:00:00|NULL
|508734|MUHAMMAD BHATTI|6|2012-09-12 00:00:00|NULL
|509371|DANIEL CHUNG|7|2012-10-24 00:00:00|NULL
|510400|NEAMAN DOUSHOUKI|8|2011-01-17 00:00:00|2012-07-03 00:00:00
|513524|KIRENDEEP DHINSA|7|2011-01-17 00:00:00|2013-03-18 00:00:00
|540742|MUHAMMAD AMMAR ABDULKARIM|6|2013-09-18 00:00:00|NULL
|633976|DASHA BARG PINTO|6|2011-08-15 00:00:00|2012-07-03 00:00:00
|633977|DALAI DOS SANTOS RIBEIRO|7|2011-01-17 00:00:00|2014-05-13 00:00:00

Task 2.9
SELECT a.StudentID,a.Name,b.Title
FROM Students AS a
INNER JOIN departments AS b

ON a.DepartmentID=b.DepartmentID

34180	NUWAN SUDASINGHAGE DON	Computer Science
119339	TASDEED AZIZ	Computer Science
303346	NISHANTH CHANDRADAS	Computer Science
338919	MUSA YERO	Computer Science
419573	SAMANTHA O'HARA	Computer Science
527084	UGOCHUKWU IWU	Computer Science
626516	AMAN PATEL	Computer Science
636079	YASODA JAYAWEERA	Computer Science
407115	JAMES GLOVER	Design
502925	SHAH ELAHI	Design
510605	PATRICK HARGAN	Design
538479	HAJI HAJI ISHAK	Design
538620	HAMZA HAMEED	Design
506279	JOANNA JOSS	Electronic and Computer Engineering
509083	RAMEESA KHAN	Electronic and Computer Engineering
518939	AKTHER HUSSAIN	Electronic and Computer Engineering
535174	SEYED KASHAN	Electronic and Computer Engineering
335483	DANIEL ONUIGWE	Mathematics
418076	VISHAL PATEL	Mathematics
508053	SANA SALEEM	Mathematics
509308	SIDRA KHAN	Mathematics
510170	NAMRATA PUN	Mathematics
512424	HARIS KHAN	Mathematics
517026	ASHIKA RAMJEE	Mathematics
320117	CHARLES WATSON	Mechanical, Aerospace and Civil Engineering
407483	ANDREAS VICTOROS	Mechanical, Aerospace and Civil Engineering
407761	JONATHAN SILVER	Mechanical, Aerospace and Civil Engineering
503045	VIVEK SRILAL	Mechanical, Aerospace and Civil Engineering
510586	KOULMIT SONI	Mechanical, Aerospace and Civil Engineering
513958	BACHIR SOGUI	Mechanical, Aerospace and Civil Engineering
413008	OLUKOTUN ASERU	Brunel Business School
431701	MAXAMED ABDULKADIR	Brunel Business School
504653	AZIM AHMAD	Brunel Business School
508734	MUHAMMAD BHATTI	Brunel Business School
540742	MUHAMMAD AMMAR ABDULKARIM	Brunel Business School
633976	DASHA BARG PINTO	Brunel Business School
413554	THEOPHILUS DOLOR	Arts and Humanities
415318	RAHUL BIST	Arts and Humanities
419392	YASSAR CHOUDHRY	Arts and Humanities
504966	HELEN BUI	Arts and Humanities
509371	DANIEL CHUNG	Arts and Humanities
513524	KIRENDEEP DHINSA	Arts and Humanities
633977	DALAI DOS SANTOS RIBEIRO	Arts and Humanities
510400	NEAMAN DOUSHOUKI	Economics and Finance

SELECT a.StudentID, a.Name, b.CourseID
FROM students AS a
LEFT OUTER JOIN studentcourses AS b
ON a.StudentID = b.StudentID
ORDER BY a.StudentID

34180	NUWAN SUDASINGHAGE DON	7
34180	NUWAN SUDASINGHAGE DON	51
34180	NUWAN SUDASINGHAGE DON	15
119339	TASDEED AZIZ	1
119339	TASDEED AZIZ	45
119339	TASDEED AZIZ	9
303346	NISHANTH CHANDRADAS	2
303346	NISHANTH CHANDRADAS	46
303346	NISHANTH CHANDRADAS	10
320117	CHARLES WATSON	30
320117	CHARLES WATSON	17
320117	CHARLES WATSON	38
335483	DANIEL ONUIGWE	20
335483	DANIEL ONUIGWE	7
335483	DANIEL ONUIGWE	28

338919	MUSA YERO	8
338919	MUSA YERO	52
338919	MUSA YERO	16
407115	JAMES GLOVER	10
407115	JAMES GLOVER	54
407115	JAMES GLOVER	18
407483	ANDREAS VICTOROS	29
407483	ANDREAS VICTOROS	16
407483	ANDREAS VICTOROS	37
407761	JONATHAN SILVER	25
407761	JONATHAN SILVER	12
407761	JONATHAN SILVER	33
413008	OLUKOTUN ASERU	34
413008	OLUKOTUN ASERU	21
413008	OLUKOTUN ASERU	42
413554	THEOPHILUS DOLOR	42
413554	THEOPHILUS DOLOR	29
413554	THEOPHILUS DOLOR	50
415318	RAHUL BIST	37
415318	RAHUL BIST	24
415318	RAHUL BIST	45
418076	VISHAL PATEL	21
418076	VISHAL PATEL	8
418076	VISHAL PATEL	29
419392	YASSAR CHOUDHRY	39
419392	YASSAR CHOUDHRY	26
419392	YASSAR CHOUDHRY	47
419573	SAMANTHA O'HARA	5
419573	SAMANTHA O'HARA	49
419573	SAMANTHA O'HARA	13
431701	MAXAMED ABDULKADIR	31
431701	MAXAMED ABDULKADIR	18
431701	MAXAMED ABDULKADIR	39
502925	SHAH ELAHI	9
502925	SHAH ELAHI	53
502925	SHAH ELAHI	17
503045	VIVEK SRILAL	28
503045	VIVEK SRILAL	15
503045	VIVEK SRILAL	36
504653	AZIM AHMAD	33
504653	AZIM AHMAD	20
504653	AZIM AHMAD	41
504966	HELEN BUI	38
504966	HELEN BUI	25
504966	HELEN BUI	46
506279	JOANNA JOSS	15
506279	JOANNA JOSS	2
506279	JOANNA JOSS	23
508053	SANA SALEEM	24
508053	SANA SALEEM	11
508053	SANA SALEEM	32
508734	MUHAMMAD BHATTI	36
508734	MUHAMMAD BHATTI	23
508734	MUHAMMAD BHATTI	44
509083	RAMEESA KHAN	17
509083	RAMEESA KHAN	4
509083	RAMEESA KHAN	25
509308	SIDRA KHAN	19
509308	SIDRA KHAN	6
509308	SIDRA KHAN	27
509371	DANIEL CHUNG	40
509371	DANIEL CHUNG	27
509371	DANIEL CHUNG	48
510170	NAMRATA PUN	22
510170	NAMRATA PUN	9
510170	NAMRATA PUN	30
510400	NEAMAN DOUSHOUKI	44
510400	NEAMAN DOUSHOUKI	31

510400 NEAMAN DOUSHOUKI 52
510586 KOULMIT SONI 27
510586 KOULMIT SONI 14
510586 KOULMIT SONI 35
510605 PATRICK HARGAN 13
510605 PATRICK HARGAN 57
510605 PATRICK HARGAN 21
512424 HARIS KHAN 18
512424 HARIS KHAN 5
512424 HARIS KHAN 26
513524 KIRENDEEP DHINSA 41
513524 KIRENDEEP DHINSA 28
513524 KIRENDEEP DHINSA 49
513958 BACHIR SOGUI 26
513958 BACHIR SOGUI 13
513958 BACHIR SOGUI 34
517026 ASHIKA RAMJEE 23
517026 ASHIKA RAMJEE 10
517026 ASHIKA RAMJEE 31
518939 AKTHER HUSSAIN 14
518939 AKTHER HUSSAIN 1
518939 AKTHER HUSSAIN 22
527084 UGOCHUKWU IWU 3
527084 UGOCHUKWU IWU 47
527084 UGOCHUKWU IWU 11
535174 SEYED KASHAN 16
535174 SEYED KASHAN 3
535174 SEYED KASHAN 24
538479 HAJI HAJI ISHAK 11
538479 HAJI HAJI ISHAK 55
538479 HAJI HAJI ISHAK 19
538620 HAMZA HAMEED 12
538620 HAMZA HAMEED 56
538620 HAMZA HAMEED 20
540742 MUHAMMAD AMMAR ABDULKARIM 32
540742 MUHAMMAD AMMAR ABDULKARIM 19
540742 MUHAMMAD AMMAR ABDULKARIM 40
626516 AMAN PATEL 6
626516 AMAN PATEL 50
626516 AMAN PATEL 14
633976 DASHA BARG PINTO 35
633976 DASHA BARG PINTO 22
633976 DASHA BARG PINTO 43
633977 DALAI DOS SANTOS RIBEIRO 43
633977 DALAI DOS SANTOS RIBEIRO 30
633977 DALAI DOS SANTOS RIBEIRO 51
636079 YASODA JAYAWEERA 4
636079 YASODA JAYAWEERA 48
636079 YASODA JAYAWEERA 12
517026 ASHIKA RAMJEE 10
517026 ASHIKA RAMJEE 31
518939 AKTHER HUSSAIN 14
518939 AKTHER HUSSAIN 1
518939 AKTHER HUSSAIN 22
527084 UGOCHUKWU IWU 3
527084 UGOCHUKWU IWU 47
527084 UGOCHUKWU IWU 11
535174 SEYED KASHAN 16
535174 SEYED KASHAN 3
535174 SEYED KASHAN 24
538479 HAJI HAJI ISHAK 11
538479 HAJI HAJI ISHAK 55
538479 HAJI HAJI ISHAK 19
538620 HAMZA HAMEED 12
538620 HAMZA HAMEED 56
538620 HAMZA HAMEED 20
540742 MUHAMMAD AMMAR ABDULKARIM 32
540742 MUHAMMAD AMMAR ABDULKARIM 19

540742	MUHAMMAD AMMAR ABDULKARIM		40
626516	AMAN PATEL	6	
626516	AMAN PATEL	50	
626516	AMAN PATEL	14	
633976	DASHA BARG PINTO	35	
633976	DASHA BARG PINTO	22	
633976	DASHA BARG PINTO	43	
633977	DALAI DOS SANTOS RIBEIRO	43	
633977	DALAI DOS SANTOS RIBEIRO	30	
633977	DALAI DOS SANTOS RIBEIRO	51	
636079	YASODA JAYAWEERA	4	
636079	YASODA JAYAWEERA	48	
636079	YASODA JAYAWEERA	12	

```
SELECT a.CourseID, b.Title, a.StudentID
FROM studentcourses AS a
RIGHT OUTER JOIN courses AS b
ON a.CourseID = b.CourseID
ORDER BY a.CourseID
```

NULL	Computing, Analytical Methods, Control and Instrum		NULL
NULL	Professional Engineering Practice	NULL	
NULL	Financial Markets	NULL	
NULL	Statistics	NULL	
NULL	Corporate Investment	NULL	
NULL	Differential and Integral Equations	NULL	
NULL	Numerical and Variational Methods for PDEs	NULL	
NULL	Major Individual Project	NULL	
NULL	Linear Algebra	NULL	
NULL	Introduction to Financial Accounting	NULL	
NULL	Algebra and Discrete Mathematics	NULL	
NULL	Major Project (see below for more)	NULL	
NULL	Financial Engineering	NULL	
NULL	Principles of Aircraft Design	NULL	
NULL	Propulsion Systems, Aircraft Structures and Materi		NULL
NULL	Calculus and Numerical Methods	NULL	
NULL	Linear and Numerical Methods	NULL	
NULL	Analysis	NULL	
NULL	Stochastic Models and Mathematical Finance		NULL
NULL	Statistics	NULL	
NULL	Professional Engineering Applications and Practice		NULL
NULL	FEA, CFD and Design of Engineering Systems		NULL
NULL	Discrete Mathematics, Probability and Statistics		NULL
NULL	Communication Skills and Operational Research		NULL
NULL	Corporate Finance	NULL	
NULL	Risk and Optimisation in Finance	NULL	
NULL	Differential and Integral Equations	NULL	
1	Introductory Programming	119339	
1	Introductory Programming	518939	
2	Data and Information	303346	
2	Data and Information	506279	
3	Information Systems and Organisations		527084
3	Information Systems and Organisations		535174
4	Logic and Computation	636079	
4	Logic and Computation	509083	
5	Software Development and Management		419573
5	Software Development and Management		512424
6	Usability Engineering	626516	
6	Usability Engineering	509308	
7	Algorithms and their Applications	34180	
7	Algorithms and their Applications	335483	
8	Networks and Operating Systems	338919	
8	Networks and Operating Systems	418076	
9	Final Year Project in Artificial Intelligence		119339
9	Final Year Project in Artificial Intelligence		502925
9	Final Year Project in Artificial Intelligence		510170
10	Software Project Management	407115	
10	Software Project Management	517026	
10	Software Project Management	303346	

11	Advanced Topics in Computer Science			538479
11	Advanced Topics in Computer Science			508053
11	Advanced Topics in Computer Science			527084
12	Artificial Intelligence	407761		
12	Artificial Intelligence	636079		
12	Artificial Intelligence	538620		
13	Software Engineering	419573		
13	Software Engineering	510605		
13	Software Engineering	513958		
14	Network Computing	518939		
14	Network Computing	510586		
14	Network Computing	626516		
15	Digital Media and Games	506279		
15	Digital Media and Games	503045		
15	Digital Media and Games	34180		
16	Creative Engineering Practice		407483	
16	Creative Engineering Practice		338919	
16	Creative Engineering Practice		535174	
17	Design Process 1	502925		
17	Design Process 1	509083		
17	Design Process 1	320117		
18	Graphic Communication	512424		
18	Graphic Communication	431701		
18	Graphic Communication	407115		
19	Product Analysis	509308		
19	Product Analysis	540742		
19	Product Analysis	538479		
20	Workshops and Materials	504653		
20	Workshops and Materials	538620		
20	Workshops and Materials	335483		
21	Design Process 2	510605		
21	Design Process 2	418076		
21	Design Process 2	413008		
22	Design for Manufacture and Communication			510170
22	Design for Manufacture and Communication			633976
22	Design for Manufacture and Communication			518939
23	Systems Design	517026		
23	Systems Design	508734		
23	Systems Design	506279		
24	Design Applications	415318		
24	Design Applications	535174		
24	Design Applications	508053		
25	Professional Practice	509083		
25	Professional Practice	407761		
25	Professional Practice	504966		
26	Major Project (core)	513958		
26	Major Project (core)	419392		
26	Major Project (core)	512424		
27	Innovation Management (core)		510586	
27	Innovation Management (core)		509371	
27	Innovation Management (core)		509308	
28	Computer-based Design Methods (core)			513524
28	Computer-based Design Methods (core)			335483
28	Computer-based Design Methods (core)			503045
29	Environmentally Sensitive Design		418076	
29	Environmentally Sensitive Design		407483	
29	Environmentally Sensitive Design		413554	
30	Graphics	320117		
30	Graphics	633977		
30	Graphics	510170		
31	Contextual Design	431701		
31	Contextual Design	510400		
31	Contextual Design	517026		
32	Embedded Systems for Design		508053	
32	Embedded Systems for Design		540742	
33	Human Factors	504653		
33	Human Factors	407761		
34	Digital Systems and Microprocessors			513958

34 Digital Systems and Microprocessors 413008
35 Web Design and Development 633976
35 Web Design and Development 510586
36 Problem Solving and Programming 503045
36 Problem Solving and Programming 508734
37 Computer Systems Mathematics 415318
37 Computer Systems Mathematics 407483
38 Internet and Network Technologies 320117
38 Internet and Network Technologies 504966
39 Computer Systems Workshop 419392
39 Computer Systems Workshop 431701
40 Data Networks, Services and Security 540742
40 Data Networks, Services and Security 509371
41 Computer Architecture and Interfacing 513524
41 Computer Architecture and Interfacing 504653
42 Digital System Design and Reliability Engineering 413008
42 Digital System Design and Reliability Engineering 413554
43 Multimedia Content Analysis and Delivery 633977
43 Multimedia Content Analysis and Delivery 633976
44 Object Oriented Systems Programming 508734
44 Object Oriented Systems Programming 510400
45 Engineering Group Design Project 119339
45 Engineering Group Design Project 415318
46 Management 504966
46 Management 303346
47 Individual Project 527084
47 Individual Project 419392
48 Distributed Systems and Computing509371
48 Distributed Systems and Computing636079
49 Network Design and Advanced Data Security 419573
49 Network Design and Advanced Data Security 513524
50 Fundamentals of Solid Body Mechanics 413554
50 Fundamentals of Solid Body Mechanics 626516
51 Fundamentals of Thermofluids 34180
51 Fundamentals of Thermofluids 633977
52 Analytical Methods and Skills 510400
52 Analytical Methods and Skills 338919
53 Engineering Materials, Manufacturing and Electrica 502925
54 Introduction to Engineering Design 407115
55 Aerospace Laboratories, Technical Drawing and Work538479
56 Solid Body Mechanics 538620
57 Thermofluids 510605

Task 2.10

SELECT COUNT(DepartmentID) AS Departmentnumber FROM departments
|13
SELECT COUNT(DISTINCT courseID) FROM studentcourses
|57
SELECT AVG(courseID) FROM studentcourses
|26.4470
SELECT MAX(courseID) FROM studentcourses
|57
SELECT MIN(courseID) FROM studentcourses
|1
SELECT SUM(courseID) FROM studentcourses
|3491

Task 2.11

SELECT DepartmentID, COUNT(DepartmentID) AS NumberofStudents FROM students
GROUP BY DepartmentID
|1|8
|2|5
|3|4
|4|7
|5|6
|6|6
|7|7
|8|1

DepartmentID	NumberofStudents
1	8
2	5
3	4
4	7
5	6
6	6
7	7
8	1

```
SELECT b.Title, COUNT(a.StudentID) AS NumberofStudents FROM students AS a
LEFT OUTER JOIN departments AS b
ON a.DepartmentID = b.DepartmentID
GROUP BY b.Title
ORDER BY b.Title
```

Title ▲ 1	NumberofStudents
Arts and Humanities	7
Brunel Business School	6
Computer Science	8
Design	5
Economics and Finance	1
Electronic and Computer Engineering	4
Mathematics	7
Mechanical, Aerospace and Civil Engineering	6

```
SELECT a.DepartmentID, b.Title, COUNT(a.StudentID) AS NumberofStudents FROM students AS a
LEFT OUTER JOIN departments AS b
ON a.DepartmentID = b.DepartmentID
GROUP BY a.DepartmentID, b.Title
ORDER BY b.Title
```

DepartmentID	Title ▲ 1	NumberofStudents
7	Arts and Humanities	7
6	Brunel Business School	6
1	Computer Science	8
2	Design	5
8	Economics and Finance	1
3	Electronic and Computer Engineering	4
4	Mathematics	7
5	Mechanical, Aerospace and Civil Engineering	6

Task 2.12

```
SELECT a.DepartmentID, b.Title, COUNT(a.StudentID) AS NumberofStudents FROM students AS a
LEFT OUTER JOIN departments AS b
ON a.DepartmentID = b.DepartmentID
GROUP BY b.Title
HAVING COUNT(a.StudentID) < 10
ORDER BY b.Title
```

DepartmentID	Title ▲ 1	NumberofStudents
7	Arts and Humanities	7
6	Brunel Business School	6
1	Computer Science	8
2	Design	5
8	Economics and Finance	1
3	Electronic and Computer Engineering	4
4	Mathematics	7
5	Mechanical, Aerospace and Civil Engineering	6

Lab 3. Hadoop 1

Task 3.1
hadoop@hadoop:~$ start-dfs.sh

Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-hadoop.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-hadoop.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-hadoop.out

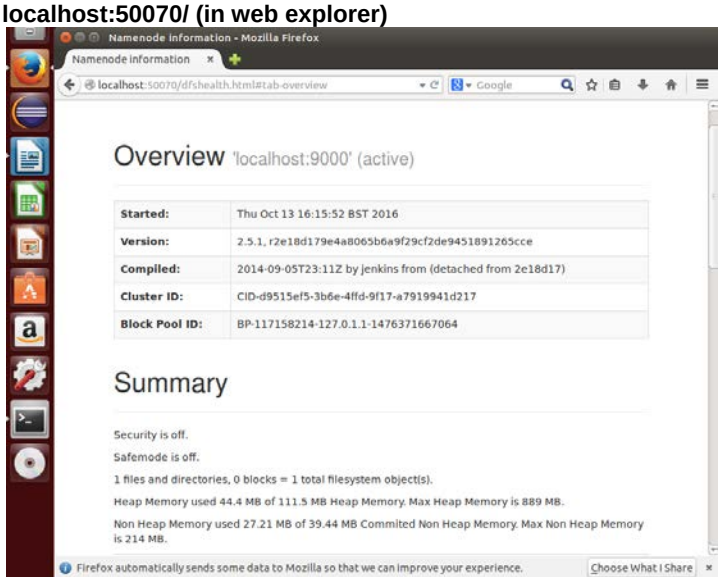
hadoop@hadoop:~$ start-yarn.sh

starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-resourcemanager-hadoop.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-nodemanager-hadoop.out

hadoop@hadoop:~$ jps

2778 NameNode
3119 SecondaryNameNode
3266 ResourceManager
2937 DataNode
3395 NodeManager
3685 Jps

localhost:50070/ (in web explorer)



hadoop@hadoop:~$ stop-dfs.sh

Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode

Task 3.2
hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/pg4300.txt /input
hadoop@hadoop:~$ hdfs dfs -ls /input

Found 1 items
-rw-r--r-- 1 hadoop supergroup 1573150 2016-10-13 17:00 /input/pg4300.txt

hadoop@hadoop:~$ hdfs dfs -get output output
hadoop@hadoop:~$ cat output/*
465 Stephen

```
16      Stephen.
6       Stephens
2       Stephens.
1       Stephanoumenos.
1       Stephanoumenos
1       Stephanos
1       Stephano
1       Stephaneforos.
```

hadoop@hadoop:~$ hdfs dfs -cat output/*

```
465     Stephen
16      Stephen.
6       Stephens
2       Stephens.
1       Stephanoumenos.
1       Stephanoumenos
1       Stephanos
1       Stephano
1       Stephaneforos.
```

Task 3.3
hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/pg4300.txt /input
hadoop@hadoop:~$ hdfs dfs -ls /input

```
Found 1 items
-rw-r--r--   1 hadoop supergroup    1573150 2016-10-14 14:36

/input/pg4300.txt
```

hadoop@hadoop:~$ hadoop jar Downloads/wordcount.jar WordCount /input output

```
16/10/14 14:38:14 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/10/14 14:38:15 WARN mapreduce.JobSubmitter: Hadoop command-line option
parsing not performed. Implement the Tool interface and execute your
application with ToolRunner to remedy this.
**************  a lot of other sentences, omitted   ****************
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1573150
        File Output Format Counters
                Bytes Written=527726
```

hadoop@hadoop:~$ hadoop fs -ls output

```
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2016-10-14 14:39
output/_SUCCESS
-rw-r--r--   1 hadoop supergroup     527726 2016-10-14 14:38 output/part-
r-00000
```

hadoop@hadoop:~$ hadoop fs -cat output/*

```
(last page)
zephyrs, 1
zero     1
zest.    1
zigzag   2
zigzagging        1
zigzags, 1
zivio,   1
zmellz   1
zodiac   1
zodiac.  1
zodiacal 2
zoe)_    1
```

zones: 1
zoo. 1
zoological 1
zouave's 1
zrads, 2
zrads. 1
É 1
Élus,_ 1
à 3
è 3
état_. 1

Lab 4. Hadoop 2

Task 4.1

4.1.1. Eclipse, establish project and create new jars for the work

MaxTemp.java

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemp
{
public static void main (String[] args) throws Exception
{
Configuration conf = new Configuration();
if (args.length != 3)
{
System.err.println("Usage: MaxTemperature <input path> <output path>");
System.err.println(args.length);
for(String s:args) System.out.println(s);
System.exit(-1);
}
Job job;
job = Job.getInstance(conf, "Max Temperature");
job.setJarByClass(MaxTemp.class);
FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));
job.setMapperClass (MaxTempMapper.class);
job.setReducerClass (MaxTempReducer.class);
job.setCombinerClass (MaxTempReducer.class);
job.setOutputKeyClass (Text.class);
job.setOutputValueClass (DoubleWritable.class);
System.exit (job.waitForCompletion(true)? 0 : 1);
}
}
```

MaxTempMapper.java

```
package org.myorg;

import java.io.IOException;

import java.util.regex.Pattern;
import java.util.regex.Matcher;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends Mapper<LongWritable,Text,Text,DoubleWritable>
{
```

```
private final static DoubleWritable tempWritable = new DoubleWritable(0);
private Text StationID = new Text();
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String[] line = value.toString().split(",");
StationID.set(line[0]);
double temp = Double.parseDouble(line[3].trim());
tempWritable.set(temp);
context.write(StationID, tempWritable);
}
}
```

MaxTempReducer.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTempReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable>
{
@Override
public void reduce (Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException
{
double maxValue=0;
for (DoubleWritable value : values)
{
maxValue = Math.max(maxValue, value.get());
}
context.write(key, new DoubleWritable(maxValue));
}
}
```

4.1.2. Hadoop Command:

hadoop@hadoop:~$ hdfs namenode -format
hadoop@hadoop:~$ start-dfs.sh
hadoop@hadoop:~$ start-yarn.sh
hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put /home/hadoop/Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hdfs dfs -ls /input

Found 1 items
-rw-r--r-- 1 hadoop supergroup 20339340 2016-10-20 12:52 /input/UK_Temperature.txt

hadoop@hadoop:~$ hadoop jar /home/hadoop/Downloads/MaxTemp.jar MaxTemp /input output
hadoop@hadoop:~$ hadoop fs -ls output

Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2016-10-20 12:55 output/_SUCCESS

-rw-r--r-- 1 hadoop supergroup 3108 2016-10-20 12:55 output/part-r-00000

hadoop@hadoop:~$ hadoop fs -cat output/*

(data from the last page)
995760 68.5
995780 60.8
995850 69.5
995920 66.0
995940 67.1
995950 60.6
996050 65.6
996070 67.3
996090 65.5
996120 63.3
996440 65.2
996480 70.7
996570 66.2
996580 63.4
996630 63.2
996770 61.9
996840 58.0
996850 67.7
996860 64.9
996920 68.0
997233 68.3
997252 62.4

Task 4.2 (2 methods)

Method 1 (set the filtering information in Java)
4.2-1.1. Java code:
MaxTemp.java

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemp
{
public static void main (String[] args) throws Exception
{
Configuration conf = new Configuration();
if (args.length != 3)
{
System.err.println("Usage: MaxTemperature <input path> <output path>");
System.err.println(args.length);
for(String s:args) System.out.println(s);
System.exit(-1);
}
Job job;
```

```
job = Job.getInstance(conf, "Max Temperature");
job.setJarByClass(MaxTemp.class);
FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));
job.setMapperClass (MaxTempMapper.class);
job.setReducerClass (MaxTempReducer.class);
job.setCombinerClass (MaxTempReducer.class);
job.setOutputKeyClass (Text.class);
job.setOutputValueClass (DoubleWritable.class);
System.exit (job.waitForCompletion(true)? 0 : 1);
}
}
```

MaxTempMapper.java

```
package org.myorg;

import java.io.IOException;

import java.util.regex.Pattern;
import java.util.regex.Matcher;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends Mapper<LongWritable,Text,Text,DoubleWritable>
{
private final static DoubleWritable tempWritable = new DoubleWritable(0);
private Text StationID = new Text();
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String[] line = value.toString().split(",");
StationID.set(line[0]);
double temp = Double.parseDouble(line[3].trim());
tempWritable.set(temp);
Pattern p = Pattern.compile("0300(.*)"); //set the filter value here
Matcher m = p.matcher(line[0]);
if (m.find())
{
context.write(StationID, tempWritable);
}
}
}
```

MaxTempReducer.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
```

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTempReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable>
{
@Override
public void reduce (Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException
{
double maxValue=0;
for (DoubleWritable value : values)
{
maxValue = Math.max(maxValue, value.get());
}
context.write(key, new DoubleWritable(maxValue));
}
}
```

4.2-1.2. Hadoop command:

hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hdfs dfs jar Downloads/MaxTemp_2.jar MaxTemp /input output
hadoop@hadoop:~$ hdfs dfs -cat output/*

```
030020 60.2
030030 60.7
030050 59.8
030064 62.9
030080 58.6
```

Method 2 (use get "FilterValue" method, allow users to input the values they want)
4.2-2.1. Java code
MaxTemp.java

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemp
{
public static void main (String[] args) throws Exception
{
Configuration conf = new Configuration();
conf.set("FilterValue", args[3]);
if (args.length != 4)
{
System.err.println("Usage: MaxTemperature <input path> <output path>");
System.err.println(args.length);
for(String s:args) System.out.println(s);
System.exit(-1);
```

```
}
Job job;
job = Job.getInstance(conf, "Max Temperature");
job.setJarByClass(MaxTemp.class);
FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));
job.setMapperClass (MaxTempMapper.class);
job.setReducerClass (MaxTempReducer.class);
job.setCombinerClass (MaxTempReducer.class);
job.setOutputKeyClass (Text.class);
job.setOutputValueClass (DoubleWritable.class);
System.exit (job.waitForCompletion(true)? 0 : 1);
}
}
```

MaxTempMapper.java

```
package org.myorg;

import java.io.IOException;

import java.util.regex.Pattern;
import java.util.regex.Matcher;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends Mapper<LongWritable,Text,Text,DoubleWritable>
{
private final static DoubleWritable tempWritable = new DoubleWritable(0);
private Text StationID = new Text();
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String[] line = value.toString().split(",");
StationID.set(line[0]);
double temp = Double.parseDouble(line[3].trim());
tempWritable.set(temp);
Configuration conf = context.getConfiguration();
Pattern p = Pattern.compile(conf.get("FilterValue"));
Matcher m = p.matcher(line[0]);
if (m.find())
{
context.write(StationID, tempWritable);
}
}
}
```

MaxTempReducer.java

```
package org.myorg;

import java.io.IOException;
```

```
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTempReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable>
{
@Override
public void reduce (Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException
{
double maxValue=0;
for (DoubleWritable value : values)
{
maxValue = Math.max(maxValue, value.get());
}
context.write(key, new DoubleWritable(maxValue));
}
}
```

4.2-2.2. Hadoop command

```
hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hadoop jar Downloads/MaxTemp_2_1.jar MaxTemp /input output "030020"
hadoop@hadoop:~$ hdfs dfs -cat output/*
```

030020 60.2

Task 4.3 (3 methods)

Method 1

Use only mapper and reducer, no combiner class, so the values obtained from the mapper would go directly to the reducer. Calculate the average temp in the reducer. But this method might be very time consuming for large quantities of data.

MeanTemp.java

```
package org.myorg;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MeanTemp
{
public static void main (String[] args) throws Exception
{
Configuration conf = new Configuration();

if (args.length != 3)
{
System.err.println("Usage: MaxTemperature <input path> <output path>");
System.err.println(args.length);
for(String s:args) System.out.println(s);
System.exit(-1);
}

Job job;
job = Job.getInstance(conf, "Max Temperature");
```

```
job.setJarByClass(MeanTemp.class);

FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));

job.setMapperClass (MeanTempMapper.class);
job.setReducerClass (MeanTempReducer.class);

job.setOutputKeyClass (Text.class);
job.setOutputValueClass (DoubleWritable.class);

System.exit (job.waitForCompletion(true)? 0 : 1);
}
}
```

MeanTempMapper.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MeanTempMapper extends Mapper<LongWritable,Text,Text,DoubleWritable>
{
private final static DoubleWritable tempWritable = new DoubleWritable(0);
private Text StationID = new Text();

@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String[] line = value.toString().split(",");
StationID.set(line[0]);
double temp = Double.parseDouble(line[3].trim());
tempWritable.set(temp);
context.write(StationID, tempWritable);
}
}
```

MeanTempReducer.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MeanTempReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable>
{
@Override
public void reduce (Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException
{
double sum = 0;
double i = 0;
for (DoubleWritable value : values)
{
sum = sum + value.get();
i++;
}
double meanValueRaw = sum / i;
String meanValueString = String.format("%.2f",meanValueRaw);
double meanValue = Double.parseDouble(meanValueString);
```

```
context.write(key, new DoubleWritable(meanValue));
}
}
```

hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hadoop jar Downloads/MeanTemp.jar MeanTemp /input output
hadoop@hadoop:~$ hdfs dfs -cat output/*

(last few results)

```
039110 47.7
039150 49.4
039160 48.15
039170 49.7
039230 48.64
039240 51.96
888780 42.38
888830 43.64
888970 42.93
889860 66.54
992700 54.47
992880 59.01
992900 53.04
994990 52.48
995120 47.75
995140 53.08
995150 49.41
995190 48.39
995220 50.93
995250 46.53
995270 49.25
995280 47.1
995290 49.36
995320 47.85
995380 56.94
995420 46.44
995430 47.24
995440 49.36
995700 59.23
995720 47.57
995730 47.72
995750 47.15
995760 55.02
995780 46.91
995850 50.93
995920 54.45
995940 52.03
995950 58.65
996050 57.54
996070 49.25
996090 47.41
996120 52.51
996440 49.1
996480 53.42
996570 54.27
996580 46.87
996630 47.59
996770 50.0
996840 49.54
996850 53.43
996860 53.4
996920 50.2
997233 52.72
997252 50.96
```

Method 2
Use a combiner, set the output temperature values in the mapper and combiner to be pairs containing "temperature" and "count", calculate the sum for each station temperatures in the combiner, and calculate the average values in the reducer.
An extra class called "MeanTempPair.java" was written as the output pair class.

MeanTemp.java
```
package org.myorg1;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MeanTemp
{
public static void main (String[] args) throws Exception
{
Configuration conf = new Configuration();
if (args.length != 3)
{
System.err.println("Usage: MeanTemperature <input path> <output path>");
System.err.println(args.length);
for(String s:args) System.out.println(s);
System.exit(-1);
}
Job job;
job = Job.getInstance(conf, "Mean Temperature");
job.setJarByClass(MeanTemp.class);
FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));
job.setMapOutputValueClass(MeanTempPair.class);
job.setMapperClass(MeanTempMapper.class);
job.setReducerClass(MeanTempReducer.class);
job.setCombinerClass(MeanTempCombiner.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(DoubleWritable.class);
System.exit (job.waitForCompletion(true)? 0 : 1);
}
}
```

MeanTempMapper.java
```
package org.myorg1;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;


public class MeanTempMapper extends Mapper<LongWritable,Text,Text,MeanTempPair>
{
private MeanTempPair pair = new MeanTempPair();
private Text StationID = new Text();
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
{
String[] line = value.toString().split(",");
StationID.set(line[0]);
double temp = Double.parseDouble(line[3].trim());
pair.set(temp,1);
context.write(StationID, pair);
```

```
}
}
```

MeanTempCombiner.java

```
package org.myorg1;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class MeanTempCombiner extends Reducer<Text, MeanTempPair, Text, MeanTempPair>
{
private MeanTempPair pair = new MeanTempPair();
@Override

public void reduce(Text key, Iterable<MeanTempPair> values, Context context)
throws IOException, InterruptedException
{
double sum = 0;
int i = 0;
for (MeanTempPair value : values)
{
sum += value.getTemp().get();
i += value.getCount().get();
}
pair.set(sum,i);
context.write(key, pair);
}
}
```

MeanTempReducer.java

```
package org.myorg1;

import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MeanTempReducer extends Reducer<Text, MeanTempPair, Text, DoubleWritable>
{
private DoubleWritable meanTemp = new DoubleWritable();
@Override

public void reduce(Text key, Iterable<MeanTempPair> values, Context context)
throws IOException, InterruptedException
{
double sum = 0;
int n = 0;

for(MeanTempPair value : values)
{
sum += value.getTemp().get();
n += value.getCount().get();
}
double aveRaw = sum/n;
String aveFloat = String.format("%.2f", aveRaw);
double AveTemp = Double.parseDouble(aveFloat);
meanTemp.set(AveTemp);

context.write(key, meanTemp);
}
}
```

MeanTempPair.java

```
package org.myorg1;
```

```
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Writable;
import org.apache.hadoop.io.WritableComparable;

import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;

public class MeanTempPair implements Writable, WritableComparable<MeanTempPair>
{
private DoubleWritable temp;
private IntWritable count;
public MeanTempPair()
{
set(new DoubleWritable(0), new IntWritable(0));
}
public void set(double temp, int count)
{
this.temp.set(temp);
this.count.set(count);
}
public void set(DoubleWritable temp, IntWritable count)
{
this.temp = temp;
this.count = count;
}
@Override
public void write(DataOutput out) throws IOException
{
temp.write(out);
count.write(out);
}
@Override
public void readFields(DataInput in) throws IOException
{
temp.readFields(in);
count.readFields(in);
}
@Override
public int compareTo(MeanTempPair other)
{
int compareVal = this.temp.compareTo(other.getTemp());
if (compareVal != 0)
{
return compareVal;
}
return this.count.compareTo(other.getCount());
}
public static MeanTempPair read (DataInput in) throws IOException
{
MeanTempPair meanPair = new MeanTempPair();
meanPair.readFields(in);
return meanPair;
}

@Override
public boolean equals(Object o)
{
if (this == o) return true;
if (o == null || getClass() != o.getClass()) return false;
MeanTempPair that = (MeanTempPair) o;
if (!count.equals(that.count)) return false;
if(!temp.equals(that.temp)) return false;
return true;
}
@Override
public int hashCode()
```

```
{
int result = temp.hashCode();
result = 163*result + count.hashCode();
return result;
}
@Override
public String toString()
{
return "MeanTemperaturePair{"+"temp="+temp+", count="+count+'}';
}
public DoubleWritable getTemp()
{
return temp;
}
public IntWritable getCount()
{
return count;
}

}
```

hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hadoop jar Downloads/MeanTemp.jar MeanTemp /input output1
hadoop@hadoop:~$ hdfs dfs -cat output1/*

(Last few results)
039150 49.4
039160 48.15
039170 49.7
039230 48.64
039240 51.96
888780 42.38
888830 43.64
888970 42.93
889860 66.54
992700 54.47
992880 59.01
992900 53.04
994990 52.48
995120 47.75
995140 53.08
995150 49.41
995190 48.39
995220 50.93
995250 46.53
995270 49.25
995280 47.1
995290 49.36
995320 47.85
995380 56.94
995420 46.44
995430 47.24
995440 49.36
995700 59.23
995720 47.57
995730 47.72
995750 47.15
995760 55.02
995780 46.91
995850 50.93
995920 54.45
995940 52.03
995950 58.65
996050 57.54
996070 49.25
996090 47.41
996120 52.51

996440 49.1
996480 53.42
996570 54.27
996580 46.87
996630 47.59
996770 50.0
996840 49.54
996850 53.43
996860 53.4
996920 50.2
997233 52.72
997252 50.96

Method 3
Use Text as output form of the temperature values in mapper and combiner.

MeanTemperature.java

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class MeanTemperature {


public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();

if (args.length != 3) {
System.err.println("Usage: MeanTemperature <input path> <output path>");
System.err.println(args.length);
for(String s : args) System.out.println(s);
System.exit(-1); }

Job job;
job=Job.getInstance(conf, "MeanTempature");
job.setJarByClass(MeanTemperature.class);

FileInputFormat.addInputPath(job, new Path(args[1]));
FileOutputFormat.setOutputPath(job, new Path(args[2]));

job.setMapperClass(AveTempMapper.class);
job.setReducerClass(AveTempReducer.class);
job.setCombinerClass(AveTempCombiner.class);

job.setMapOutputValueClass(Text.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(DoubleWritable.class);

System.exit(job.waitForCompletion(true) ? 0 : 1); } }
```

AveTempMapper.java

```
package org.myorg;

import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class AveTempMapper extends Mapper<LongWritable, Text, Text, Text>
```

```
{

private Text temp = new Text();
private Text StationID = new Text();

@Override

public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException

{

String[] line = value.toString().split(",");
StationID.set(line[0]);
temp.set(line[3].trim());
context.write(StationID, temp);
}
}
```

AveTempCombiner.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AveTempCombiner extends Reducer<Text, Text, Text, Text>

{
private Text AveCal = new Text();

@Override

public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException
{
double sum = 0;
double i = 0;

for (Text value : values)
{
String temp = value.toString().trim();
sum += Double.parseDouble(temp);
i++;
}

String sumvalue = Double.toString(sum);
String ivalue = Double.toString(i);
String aveCal = sumvalue + "," + ivalue;
AveCal.set(aveCal);

context.write(key, AveCal);
}
}
```

AveTempReducer.java

```
package org.myorg;

import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AveTempReducer extends Reducer<Text, Text, Text, DoubleWritable>
{
private DoubleWritable MeanTemp = new DoubleWritable();
```

```
@Override

public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException

{
double sum = 0;
double n = 0;

for (Text value : values)
{
String[] AveCal = value.toString().split(",");
sum += Double.parseDouble(AveCal[0].trim());
n += Double.parseDouble(AveCal[1].trim());
}

double aveRaw = sum/n;
String aveFloat = String.format("%.2f", aveRaw);
double AveTemp = Double.parseDouble(aveFloat);
MeanTemp.set(AveTemp);

context.write(key, MeanTemp);
}
}
```

hadoop@hadoop:~$ hdfs dfs -mkdir /input
hadoop@hadoop:~$ hdfs dfs -put Downloads/UK_Temperature.txt /input
hadoop@hadoop:~$ hadoop jar Downloads/MeanTempString.jar MeanTemperature /input output1
hadoop@hadoop:~$ hdfs dfs -cat output1/*

(last few results)
039240 51.96
888780 42.38
888830 43.64
888970 42.93
889860 66.54
992700 54.47
992880 59.01
992900 53.04
994990 52.48
995120 47.75
995140 53.08
995150 49.41
995190 48.39
995220 50.93
995250 46.53
995270 49.25
995280 47.1
995290 49.36
995320 47.85
995380 56.94
995420 46.44
995430 47.24
995440 49.36
995700 59.23
995720 47.57
995730 47.72
995750 47.15
995760 55.02
995780 46.91
995850 50.93
995920 54.45
995940 52.03
995950 58.65
996050 57.54
996070 49.25
996090 47.41

996120 52.51
996440 49.1
996480 53.42
996570 54.27
996580 46.87
996630 47.59
996770 50.0
996840 49.54
996850 53.43
996860 53.4
996920 50.2
997233 52.72
997252 50.96

Lab 5. Hadoop 3

Task 5.1

In the mapper, get the line numbers first. Make the words in the line into an array, filter out the non-alphabetic symbols, and then output the word and line numbers into the reducer. Collect the line numbers by StringBuffer and output the word and line numbers. (In the reducer, I have also put all the line numbers in order.)

Driver

InverseIndex.java

```
package org.myorg;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class InverseIndex
{
  public static void main(String[] args) throws Exception
  {
     Configuration conf = new Configuration();
     if (args.length != 3)
         {
             System.err.println("Usage: InverseIndex<input path><output path>");
             System.err.println(args.length);
             for(String s:args) System.out.println(s);
             System.exit(-1);
         }

    Job job;
    job=Job.getInstance(conf, "Inverse Index");
    job.setJarByClass(InverseIndex.class);

    FileInputFormat.addInputPath(job, new Path(args[1]));
    FileOutputFormat.setOutputPath(job, new Path(args[2]));

    job.setMapperClass(InverseIndexMapper.class);
    job.setReducerClass(InverseIndexReducer.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);

    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Mapper

InverseIndexMapper.java

```
package org.myorg;

import java.io.IOException;
import java.util.regex.Pattern;
import java.util.regex.Matcher;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class InverseIndexMapper extends Mapper<LongWritable, Text, Text, Text>
{
    private final static Text lineNo = new Text();
    private Text word = new Text();
    @Override
    public void map(LongWritable key, Text value, Context context)
    throws IOException, InterruptedException
        {
```

```
            String[] line = value.toString().trim().split(": ",2);
            if(line.length == 2)
                {
                  lineNo.set(line[0]);
                  String a = line[1].replaceAll("[^a-zA-Z]", " ");
                  String[] wordVector = a.toString().split(" ");
                  for ( int i = 0; i < wordVector.length; i++)
                      {
                        Pattern p = Pattern.compile("[a-zA-Z]");
                        Matcher m = p.matcher(wordVector[i]);
                        if (m.find())
                            {
                                word.set(wordVector[i]);
                                context.write(word,lineNo);
                            }
                      }
                }
        }
}
```

Reducer
InverseIndexReducer.java

```
package org.myorg;

import java.io.IOException;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.util.StringUtils;

public class InverseIndexReducer extends Reducer<Text, Text, Text, Text>
{
   private Text lineNo = new Text();

   @Override
   public void reduce(Text key, Iterable<Text> values, Context context)
         throws IOException, InterruptedException
   {
      StringBuffer buffer = new StringBuffer();

      for (Text value : values)
      {
         if(buffer.length() != 0)
         {
            buffer.append(",");
         }
         buffer.append(value.toString());
      }

      String[] Buffer = buffer.toString().split(",");
      int[] BufferDouble = new int[Buffer.length];
      for (int k=0; k<Buffer.length; k++)
      {
         BufferDouble[k] = Integer.parseInt(Buffer[k]);
      }

      for (int j=0; j<BufferDouble.length;j++)
      {
         for (int i=j+1; i<BufferDouble.length; i++)
         {
            if(BufferDouble[i] < (BufferDouble[j]))
            {
               int lineNumber = BufferDouble[j];
               BufferDouble[j] = BufferDouble[i];
               BufferDouble[i] = lineNumber;
            }
         }
```

```
        }
        String[] sortedBuffer = new String[BufferDouble.length];
        for (int m=0; m < BufferDouble.length; m++)
        {
            sortedBuffer[m] = Integer.toString(BufferDouble[m]);
        }
        String lineBuffer = StringUtils.arrayToString(sortedBuffer);
        lineNo.set(lineBuffer);
        context.write(key, lineNo);
    }
}
```

```
hadoop@hadoop:~$ start-dfs.sh
hadoop@hadoop:~$ start-yarn.sh
hadoop@hadoop:~$ hdfs dfs -mkdir /input2
hadoop@hadoop:~$ hdfs dfs -put Downloads/pg4400.txt /input2
hadoop@hadoop:~$ hadoop jar Downloads/InverseIndex2.jar InverseIndex /input2 output2
hadoop@hadoop:~$ hdfs dfs -cat output2/*
```

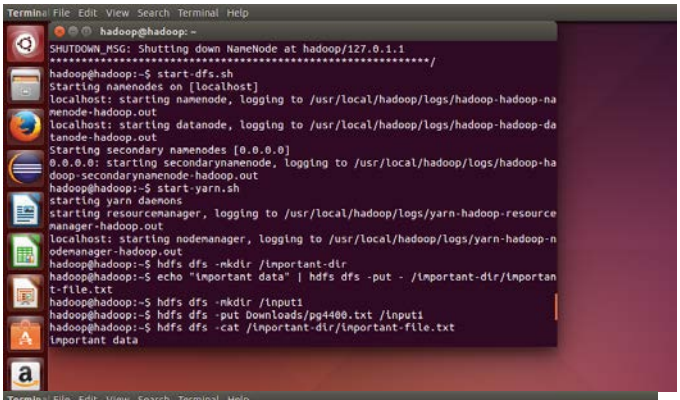
(last few lines)

```
youd    31020,31737,31843,32011,32035,32040
young    912,1052,1059,1074,1109,1901,2010,2044,2139,2143,2153,2579,2672,2838,2936,2943,3002,
3017,3988,4344,4496,4721,5028,5029,6065,6077,6908,6974,7055,7600,7608,7619,7631,7632,7758,7
806,7897,7989,8184,8525,8710,8752,8886,8892,9009,9200,9875,10095,10571,10598,10613,10614,10
614,10615,10787,10821,10987,10990,11270,11514,12395,12483,12484,13554,13559,13812,13819,13
844,13903,14525,14981,14992,14994,14998,15170,15884,16712,16840,16851,16918,17074,17155,17
367,17530,17542,17783,17953,18015,18102,18130,18141,18276,18284,18314,18388,18390,18392,18
395,18401,18408,18409,18417,18421,18432,18437,18440,18460,18477,18480,18541,18563,18641,18
693,18819,18878,18946,18971,19036,19047,19053,19085,19125,19134,19298,19299,19315,19361,19
385,19405,19481,19653,19931,20902,20967,21305,22637,22928,22967,23304,23339,24134,24312,24
674,25023,25394,25482,26013,26161,26176,26774,27072,27112,27194,27249,28472,28726,29938,31
037,31079,31084,31337,31806,31862,31969,32187,32367,32378,32390,32396,32405,32410,32411,32
417,32425,32467,32477,32532
younger    325,3862,9019,9066,21657,27191,28309,28315,29228,31587
youngling    10238
youngly    9200
youngster    4715,11769
youngsters    4756
youngun    19778
your    68,139,166,171,173,269,271,308,318,332,333,336,343,456,531,568,684,719,724,728,735,747,7
73,779,779,800,810,853,989,1261,1429,1568,1593,1597,1644,1659,1780,1839,1840,1847,1860,1876,
1906,1948,1959,2003,2015,2061,2072,2082,2089,2124,2231,2346,2365,2942,3371,3520,3524,3534,3
538,3545,3573,3649,3649,3736,3769,3796,3814,3829,3832,3957,4304,4926,4973,5071,5137,5144,51
76,5317,5724,5742,5774,6346,6349,6349,6523,6681,7097,7309,7336,7356,7495,7572,7642,7657,765
7,7833,7834,7965,7970,8175,8186,8248,8396,8408,8544,8593,8676,8767,8900,8917,8917,9098,9118,
9123,9407,9431,9446,9458,9522,9687,9795,9800,9948,9965,10011,10126,10168,10312,10836,10934,
10970,11021,11096,11204,11318,11358,11371,11424,11427,11462,11601,11698,11748,11965,12306,
12372,12385,12448,12469,12484,12485,12625,12706,12790,13086,13255,13260,13302,13362,13467,
13530,13536,13687,13830,13848,13848,13858,13859,13980,14085,14086,14115,14314,14330,14644,
14644,14664,14681,14739,14822,15102,15153,15155,15158,15158,15174,15215,15220,15278,15470,
15488,15496,15537,15585,15668,15675,15699,15758,15845,15900,15993,16022,16032,16102,16117,
16126,16255,16258,16352,16352,16475,16482,16527,16530,16659,16730,16766,16768,16768,16773,
16825,16826,17007,17013,17291,17388,17504,17754,17786,17859,17880,17929,17930,17941,17943,
18138,18140,19380,19416,19614,19618,19764,19774,19779,19779,19779,19789,19792,19857,19864,
19992,20023,20034,20155,20179,20187,20193,20213,20244,20290,20309,20372,20382,20414,20464,
20494,20554,20680,20716,20804,20859,20860,21103,21112,21184,21184,21199,21366,21510,21520,
21520,21538,21567,21582,21696,22050,22050,22165,22181,22214,22269,22289,22289,22291,22307,
22453,22458,22460,22461,22476,22494,22495,22508,22569,22612,22843,22927,22977,22987,23019,
23024,23044,23052,23053,23127,23127,23136,23138,23144,23152,23153,23161,23177,23177,23186,
23187,23204,23222,23231,23241,23241,23242,23243,23287,23295,23298,23302,23316,23322,23334,
23347,23357,23357,23358,23369,23442,23450,23670,23692,23693,23694,23700,23700,23701,23711,
23751,23879,23887,23895,23985,24000,24024,24124,24216,24246,24331,24421,24477,24493,24574,
24591,24599,24695,24705,24728,24770,24773,24887,24973,24973,25070,25128,25167,25191,25385,
25417,25427,25466,25466,25506,25617,25621,25654,25659,25675,25733,25766,25864,25974,26027,
26041,26156,26214,26299,26410,26416,26466,26565,26742,26743,26744,26744,26957,27485,28644,
28782,30248,30330,31093,31099,31106,31108,31143,31163,31167,31451,31527,31538,31615,31650,
31732,31732,31761,31770,31922,31951,32059,32082,32087,32122,32209,32233,32495,32577,32638,
```

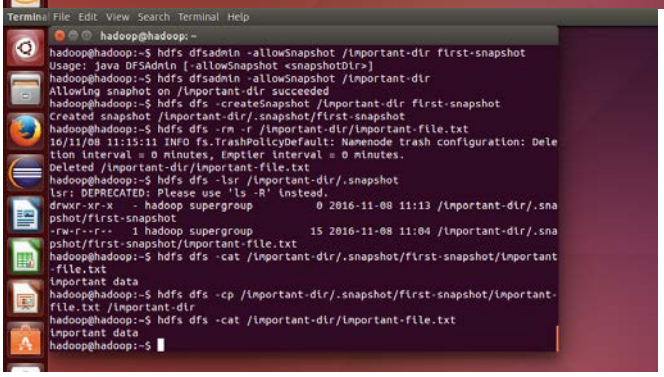
32748,32783,32816,32854,32859,32899,32920,32968,32982
youre 31121,31217,31349,31779,31880,32178,32201,32346
yourn 19848
yours 332,3991,5806,7635,8363,11158,13092,14337,15196,19320,21720,23714,23810,25653,28164,31768
yourself 231,989,1375,1404,2011,2068,3015,4244,5603,5816,6467,6514,6682,10136,10423,10933,11402,12564,13728,16528,17624,17890,17947,18993,20336,20364,20748,20812,22144,23702,23998,24025,25211,25613,26743,31104,32005,32407
yourselves 5226,23679
yous 31763
youth 1518,6346,6673,8260,8703,9257,9288,9407,9840,10180,12748,13753,14106,14461,16405,17939,18645,19109,19677,20181,22638,23427,23883,28171,28297,28327,28811,29697,30006,30764
youthful 6677,6720,12642,19330,20901,27523
youths 7626
youve 31491,31765
yrs 26887,31650
yu 19831
yum 17936,17936
yumyum 18021,24616
yung 19832
ywimpled 18264
z 26888
zamatejch 28647
zeal 1384,1702,9489
zealous 8666,10408
zebra 2153
zenith 22275,29362,29975
zephyrs 5749
zero 29422
zest 8419
zigzag 22824,24629
zigzagging 6074
zigzags 24627
zip 32699
zique 14828
zivio 14930
zmellz 2200
zodiac 22379,28410
zodiacal 19349,29363
zoe 22771
zones 27879
zoo 18030
zoological 28412
zouave 21920
zrads 17751,17751,17751

Task 5.3

1. Snapshot for backup data

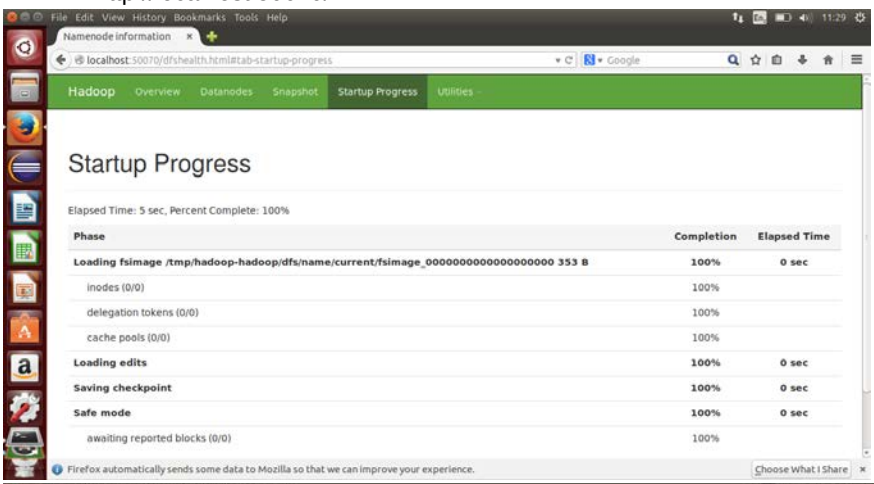


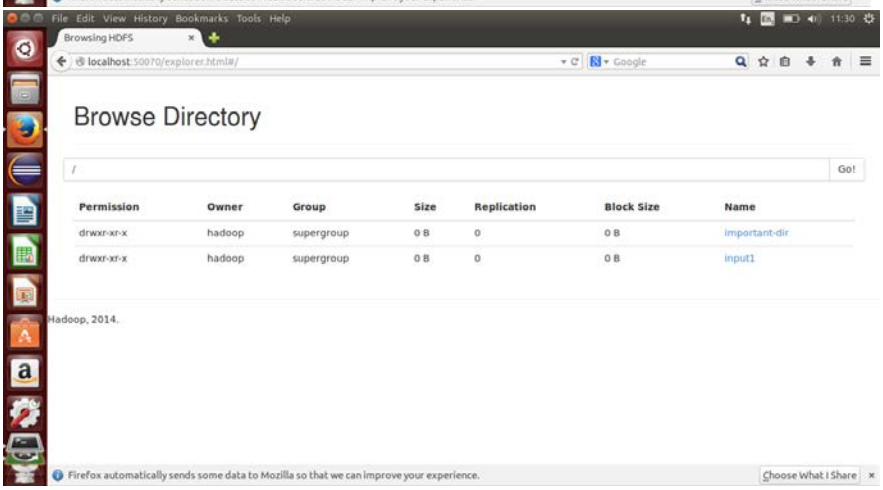
```
hadoop@hadoop: ~
SHUTDOWN_MSG: Shutting down NameNode at hadoop/127.0.1.1
************************************************************/
hadoop@hadoop:~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-na
menode-hadoop.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-da
tanode-hadoop.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-ha
doop-secondarynamenode-hadoop.out
hadoop@hadoop:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-resource
manager-hadoop.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-n
odemanager-hadoop.out
hadoop@hadoop:~$ hdfs dfs -mkdir /important-dir
hadoop@hadoop:~$ echo "important data" | hdfs dfs -put - /important-dir/importan
t-file.txt
hadoop@hadoop:~$ hdfs dfs -mkdir /input1
hadoop@hadoop:~$ hdfs dfs -put Downloads/pg4400.txt /input1
hadoop@hadoop:~$ hdfs dfs -cat /important-dir/important-file.txt
important data
```

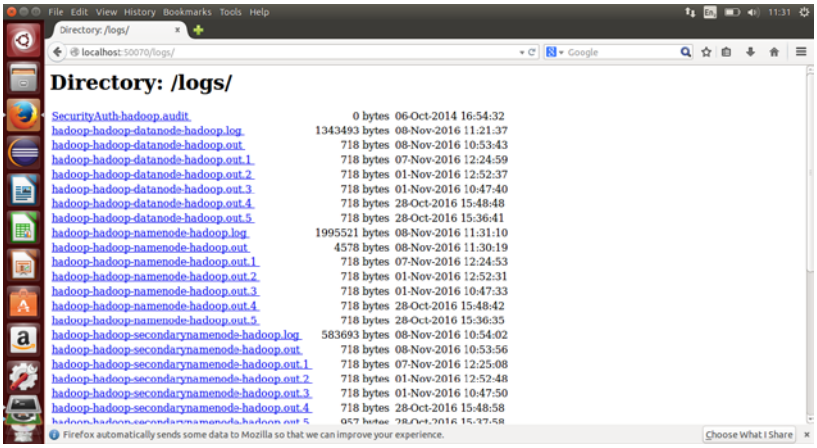


```
hadoop@hadoop: ~
hadoop@hadoop:~$ hdfs dfsadmin -allowSnapshot /important-dir first-snapshot
Usage: java DFSAdmin [-allowSnapshot <snapshotDir>]
hadoop@hadoop:~$ hdfs dfsadmin -allowSnapshot /important-dir
Allowing snaphot on /important-dir succeeded
hadoop@hadoop:~$ hdfs dfs -createSnapshot /important-dir first-snapshot
Created snapshot /important-dir/.snapshot/first-snapshot
hadoop@hadoop:~$ hdfs dfs -rm -r /important-dir/important-file.txt
16/11/08 11:15:11 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /important-dir/important-file.txt
hadoop@hadoop:~$ hdfs dfs -lsr /important-dir/.snapshot
lsr: DEPRECATED: Please use 'ls -R' instead.
drwxr-xr-x   - hadoop supergroup          0 2016-11-08 11:13 /important-dir/.sna
pshot/first-snapshot
-rw-r--r--   1 hadoop supergroup         15 2016-11-08 11:04 /important-dir/.sna
pshot/first-snapshot/important-file.txt
hadoop@hadoop:~$ hdfs dfs -cat /important-dir/.snapshot/first-snapshot/important
-file.txt
important data
hadoop@hadoop:~$ hdfs dfs -cp /important-dir/.snapshot/first-snapshot/important-
file.txt /important-dir
hadoop@hadoop:~$ hdfs dfs -cat /important-dir/important-file.txt
important data
hadoop@hadoop:~$
```

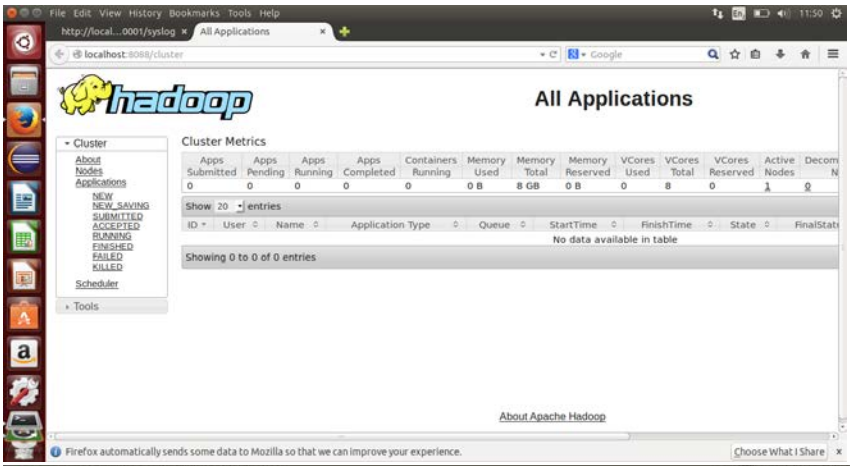
2. http://localhost:50070/

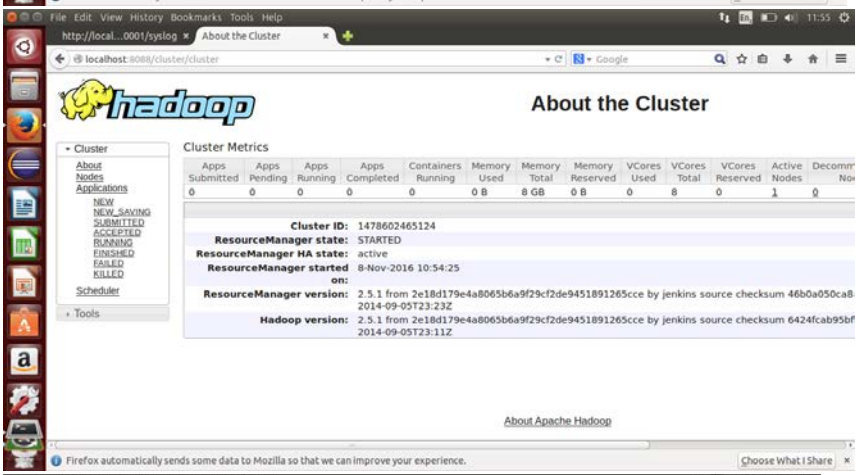


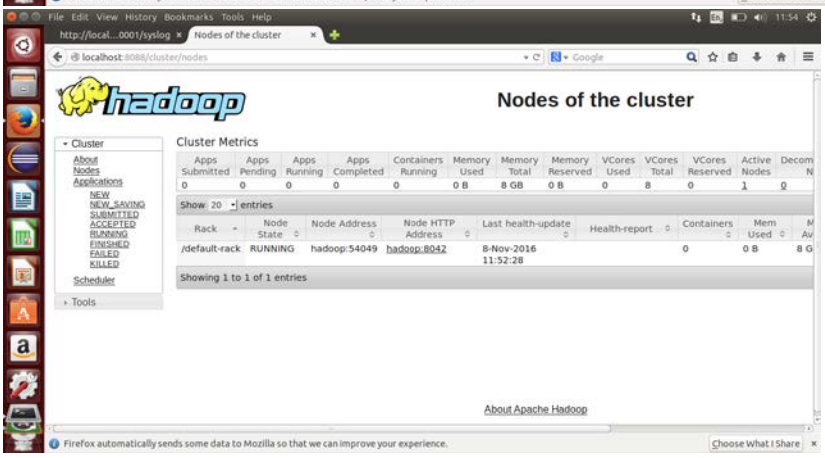


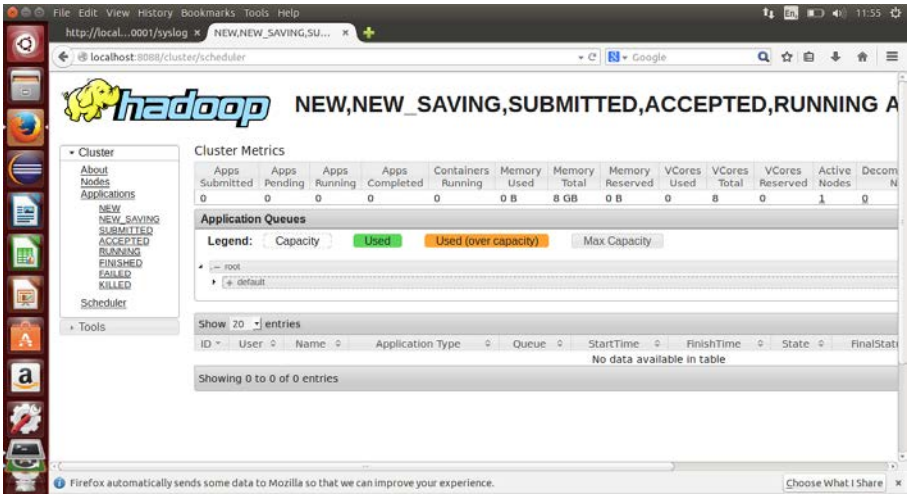


3. http://localhost:8088








File Edit View History Bookmarks Tools Help
http://local...0001/syslog
NEW,NEW_SAVING,SU...
localhost:8088/cluster/scheduler
hadoop
NEW,NEW_SAVING,SUBMITTED,ACCEPTED,RUNNING A
Cluster
About
Nodes
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted
Apps Pending
Apps Running
Apps Completed
Containers Running
Memory Used
Memory Total
Memory Reserved
VCores Used
VCores Total
VCores Reserved
Active Nodes
0 0 0 0 0 0 B 8 GB 0 B 0 8 0 1 0
Application Queues
Legend: Capacity Used Used (over capacity) Max Capacity
root
default
Show 20 entries
ID User Name Application Type Queue StartTime FinishTime State
No data available in table
Showing 0 to 0 of 0 entries
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

Lab 7. PIG and HIVE (No task in lab 6)

1. PIG

```
hadoop@hadoop:~$ hdfs namenode -format
hadoop@hadoop:~$ start-dfs.sh
hadoop@hadoop:~$ start-yarn.sh
hadoop@hadoop:~$ hdfs dfs -put Downloads/data/UK_Temperature.txt /
hadoop@hadoop:~$ pig -x mapreduce
grunt> UKT = LOAD '/UK_Temperature.txt' USING PigStorage(',') AS (STN:chararray, WBAN:int,
YEARMODA:chararray, TEMP:double, tCount:int, DEWP:double, dCount:int, SLP:double, slpCount:int,
STP:double, stpCount:int, VISIB:double, vCount:int, WDSP:double, wCount:int, MXSPD:double,
GUST:double, MAX:chararray, MIN:chararray, PRCP:chararray, SNDP:double, FRSHTT:chararray);
grunt> grpStn = GROUP UKT1 BY STN;
grunt> MaxTemp = FOREACH grpStn GENERATE group, MAX(UKT1.TEMP);
grunt> DUMP MaxTemp;
```

```
(last few lines of output)
(995420,64.0)
(995430,62.5)
(995440,66.4)
(995700,71.9)
(995720,62.5)
(995730,60.8)
(995750,63.2)
(995760,68.5)
(995780,60.8)
(995850,69.5)
(995920,66.0)
(995940,67.1)
(995950,60.6)
(996050,65.6)
(996070,67.3)
(996090,65.5)
(996120,63.3)
(996440,65.2)
(996480,70.7)
(996570,66.2)
(996580,63.4)
(996630,63.2)
(996770,61.9)
(996840,58.0)
(996850,67.7)
(996860,64.9)
(996920,68.0)
(997233,68.3)
(997252,62.4)
```

```
grunt> MeanTemp = FOREACH grpStn GENERATE group, AVG(UKT1.TEMP);
grunt> DUMP MeanTemp;
```

```
(995190,48.38670886075949)
(995220,50.92581196581195)
(995250,46.525949367088614)
(995270,49.24991334488731)
(995280,47.09683544303797)
(995290,49.360481099656404)
(995320,47.84873417721518)
(995380,56.941739130434804)
(995420,46.44240506329114)
(995430,47.24050632911391)
(995440,49.35823223570188)
(995700,59.232586206896606)
(995720,47.570253164556966)
(995730,47.722784810126576)
(995750,47.149999999999984)
(995760,55.02124999999998)
(995780,46.90506329113925)
(995850,50.92507204610951)
```

```
(995920,54.45378006872851)
(995940,52.034529914529884)
(995950,58.650000000000006)
(996050,57.54157608695656)
(996070,49.248526863084905)
(996090,47.40509554140129)
(996120,52.50689655172411)
(996440,49.095454545454544)
(996480,53.41775862068958)
(996570,54.27310344827587)
(996580,46.86962025316454)
(996630,47.58544303797469)
(996770,49.9996551724138)
(996840,49.54074074074068)
(996850,53.43103448275863)
(996860,53.40317848410758)
(996920,50.19723865877711)
(997233,52.716803278688516)
(997252,50.963436123348)

grunt> STORE MeanTemp INTO 'OutputMeanTemp';
```

(IN BATCH MODE)

```
/* UK_Temperature */

UKT = LOAD '/UK_Temperature.txt' USING PigStorage(',') AS (STN:int, WBAN:int,
YEARMODA:chararray, TEMP:double, tCount:int, DEWP:double, dCount:int, SLP:double, slpCount:int,
STP:double, stpCount:int, VISIB:double, vCount:int, WDSP:double, wCount:int, MXSPD:double,
GUST:double, MAX:chararray, MIN:chararray, PRCP:chararray, SNDP:double, FRSHTT:chararray);

Filtered = FILTER UKT BY STN == $fStation;
FiltTemp = FOREACH Filtered GENERATE STN, TEMP;
STORE FiltTemp INTO 'UKTOutput';
```

(IN HDFS)

```
hadoop@hadoop:~$ pig -param fStation="995440" /home/hadoop/UKT.pig
hadoop@hadoop:~$ hdfs dfs -get UKTOutput/*

995440  42.2
995440  44.2
995440  48.0
995440  48.9
995440  47.2
995440  47.4
995440  46.6
995440  45.3
995440  44.2
995440  41.1
995440  40.4
995440  38.6
995440  39.6
995440  39.4
995440  37.1
995440  40.1
995440  39.1
995440  37.4
```

2. HIVE

```
hive> create table UKT (STN INT, WBAN INT, YEARMODA STRING, TEMP DOUBLE, tCount INT,
DEWP DOUBLE, dCount INT, SLP DOUBLE, slpCount INT, STP DOUBLE, stpCount INT, VISIB
DOUBLE, vCount INT, WDSP DOUBLE, wCount INT, MXSPD DOUBLE, GUST DOUBLE, MAX
STRING, MIN STRING, PRCP STRING, SNDP DOUBLE, FRSHTT STRING)
  > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
  > STORED AS TEXTFILE;
```

```
hive> LOAD DATA LOCAL INPATH 'Downloads/data/UK_Temperature.txt' OVERWRITE INTO TABLE
UKT;

hive> SELECT STN, MAX(TEMP) AS MaxTemp FROM UKT GROUP BY STN;

888780  59.3
888830  62.4
888970  53.5
889860  76.3
992700  68.1
992880  66.0
992900  66.6
994990  70.0
995120  61.7
995140  68.1
995150  67.6
995190  62.9
995220  68.0
995250  61.5
995270  67.6
995280  60.1
995290  67.3
995320  60.6
995380  67.8
995420  64.0
995430  62.5
995440  66.4
995700  71.9
995720  62.5
995730  60.8
995750  63.2
995760  68.5
995780  60.8
995850  69.5
995920  66.0
995940  67.1
995950  60.6
996050  65.6
996070  67.3
996090  65.5
996120  63.3
996440  65.2
996480  70.7
996570  66.2
996580  63.4
996630  63.2
996770  61.9
996840  58.0
996850  67.7
996860  64.9
996920  68.0
997233  68.3
997252  62.4

hive> SELECT STN, round(AVG(TEMP),2) AS MeanTemp FROM UKT GROUP BY STN;

995280  47.1
995290  49.36
995320  47.85
995380  56.94
995420  46.44
995430  47.24
995440  49.36
995700  59.23
995720  47.57
995730  47.72
995750  47.15
995760  55.02
```

995780 46.91
995850 50.93
995920 54.45
995940 52.03
995950 58.65
996050 57.54
996070 49.25
996090 47.41
996120 52.51
996440 49.1
996480 53.42
996570 54.27
996580 46.87
996630 47.59
996770 50.0
996840 49.54
996850 53.43
996860 53.4
996920 50.2
997233 52.72
997252 50.96

hive> SELECT * FROM UKT WHERE STN=995760;

995760 99999 20140913 62.0 24 55.5 24 1028.1 24 9999.9
NULL 7.0 24 999.9 NULL 999.9 999.9 65.1* 60.3* 0.00I
999.9 000000
995760 99999 20140914 63.2 24 58.2 24 1024.2 24 9999.9
NULL 6.3 24 999.9 NULL 999.9 999.9 66.2* 60.4* 0.00I
999.9 000000
995760 99999 20140915 65.3 24 60.1 24 1018.3 24 9999.9
NULL 6.8 24 999.9 NULL 999.9 999.9 71.8* 61.3* 0.00I
999.9 000000
995760 99999 20140916 64.6 24 62.2 24 1016.6 24 9999.9
NULL 1.3 24 999.9 NULL 999.9 999.9 68.9* 62.6* 0.00I
999.9 100000
995760 99999 20140917 65.9 24 63.0 24 1013.5 24 9999.9
NULL 1.4 24 999.9 NULL 999.9 999.9 72.7* 60.8* 0.00I
999.9 100000
995760 99999 20140918 65.9 24 62.8 24 1011.6 24 9999.9
NULL 2.4 24 999.9 NULL 999.9 999.9 71.2* 63.9* 0.00I
999.9 000000
995760 99999 20140919 66.8 23 63.7 23 1012.4 23 9999.9
NULL 1.8 23 999.9 NULL 999.9 999.9 74.7* 63.9* 0.00I
999.9 000000
995760 99999 20140920 61.4 24 60.7 24 1015.7 24 9999.9
NULL 2.7 24 999.9 NULL 999.9 999.9 64.0* 59.9* 99.99
999.9 110000
995760 99999 20140921 59.6 24 51.7 24 1022.1 24 9999.9
NULL 7.6 24 999.9 NULL 999.9 999.9 62.1* 57.7* 99.99
999.9 010000
995760 99999 20140922 58.4 24 48.4 24 1024.1 24 9999.9
NULL 3.3 24 999.9 NULL 999.9 999.9 63.1* 56.3* 0.00I
999.9 000000
995760 99999 20140923 59.6 24 51.1 24 1019.3 24 9999.9
NULL 9.7 24 999.9 NULL 999.9 999.9 61.3* 57.2* 0.00I
999.9 000000
995760 99999 20140924 57.3 24 52.7 24 1011.8 24 9999.9
NULL 7.0 24 999.9 NULL 999.9 999.9 59.7* 55.6* 99.99
999.9 010000
995760 99999 20140925 59.0 24 52.9 24 1017.6 24 9999.9
NULL 11.9 24 999.9 NULL 999.9 999.9 62.6* 55.9* 0.00I
999.9 000000
995760 99999 20140926 61.5 24 57.8 24 1021.9 24 9999.9
NULL 11.1 24 999.9 NULL 999.9 999.9 62.4* 60.1* 0.00I
999.9 000000

995760 99999 20140927 60.3 24 54.0 24 1027.7 24 9999.9
NULL 12.2 24 999.9 NULL 999.9 999.9 63.1* 58.6* 0.00I
999.9 000000
995760 99999 20140928 63.7 24 60.5 24 1026.3 NULL 9999.9
NULL 4.1 24 999.9 NULL 999.9 999.9 70.3* 60.8* 0.00I
999.9 000000
995760 99999 20140929 63.2 24 61.4 24 9999.9 NULL 9999.9
NULL 1.6 24 999.9 NULL 999.9 999.9 68.9* 60.3* 99.99
999.9 010000
995760 99999 20140930 60.6 24 58.9 24 9999.9 NULL 9999.9
NULL 3.7 24 999.9 NULL 999.9 999.9 63.1* 58.5* 0.00I
999.9 000000
995760 99999 20141001 61.2 24 58.2 24 9999.9 NULL 9999.9
NULL 6.6 24 999.9 NULL 999.9 999.9 62.8* 59.7* 0.00I
999.9 000000