Swinburne University of Technology Hawthorn Campus
Dept. of Computer Science and Software Engineering

COS20028 Big Data Architecture and Application
Application Project - *Semester 2, 2022*

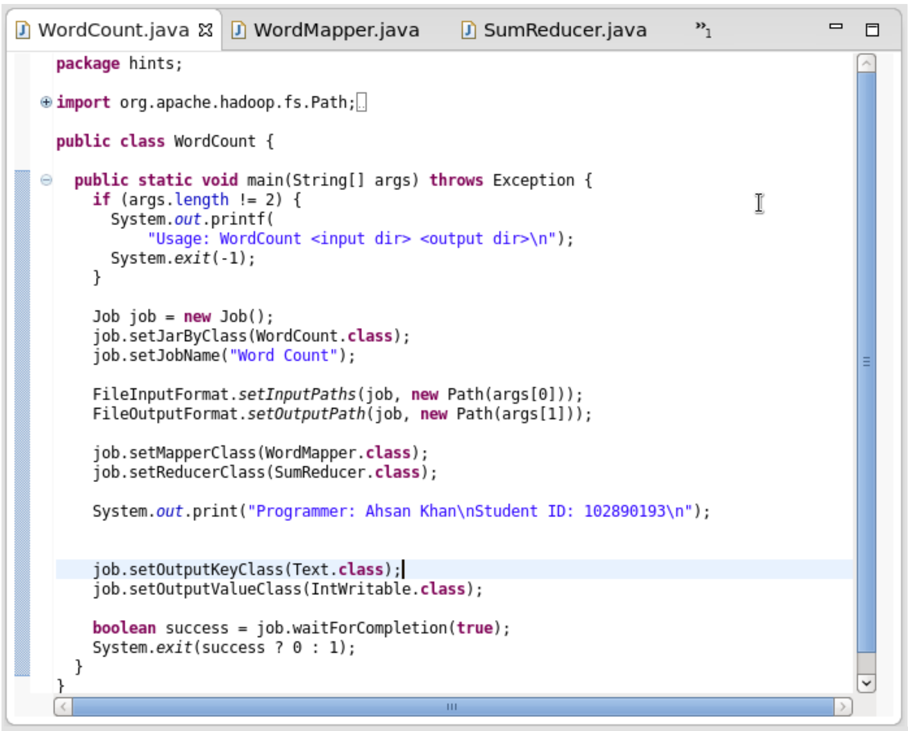
Name: Ahsan Khan
Student ID: 102890193
Submission Date: 09/11/2022

Assignment Task

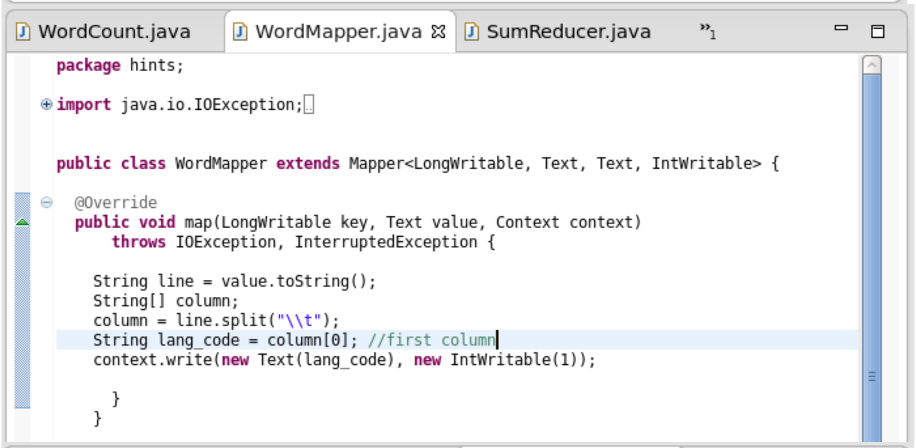
1. Find a way with tools taught in this unit to check whether the language_code attribute in the source data is unique. **Note that you cannot eyeball the result. The result should be found by combining valid tools in a sequence.**
   1.1. List the sequence of tools you used for finding the answer:
   Ans: MapReduce then SQL

   1.2. Give a short explanation of which tool did what.
   Ans: MapReduce to split the data to get the language code attribute and the count for each of them. SQL is used to query the result for any results greater than 1 (indicating that it is not unique).

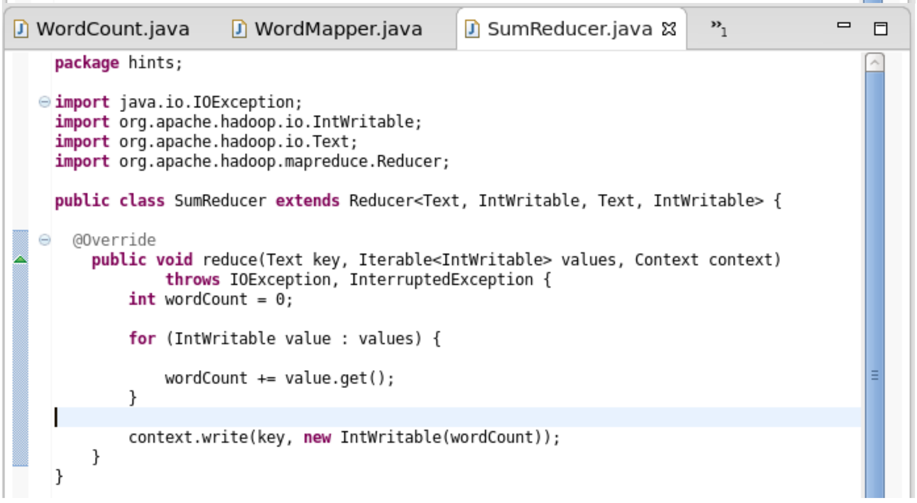
   1.3. List the code/command/statement and the outcome screenshot of the step in the sequence: (Note that only meaningful screenshot is required. For example, the screenshot of the MapReduce execution result is necessary, but the screenshot of the outcome from the "cd .." command is not meaningful)
   [Phase 1 – MapReduce]



```java
package hints;

import org.apache.hadoop.fs.Path;

public class WordCount {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.out.printf(
          "Usage: WordCount <input dir> <output dir>\n");
      System.exit(-1);
    }

    Job job = new Job();
    job.setJarByClass(WordCount.class);
    job.setJobName("Word Count");

    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapperClass(WordMapper.class);
    job.setReducerClass(SumReducer.class);

    System.out.print("Programmer: Ahsan Khan\nStudent ID: 102890193\n");


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

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

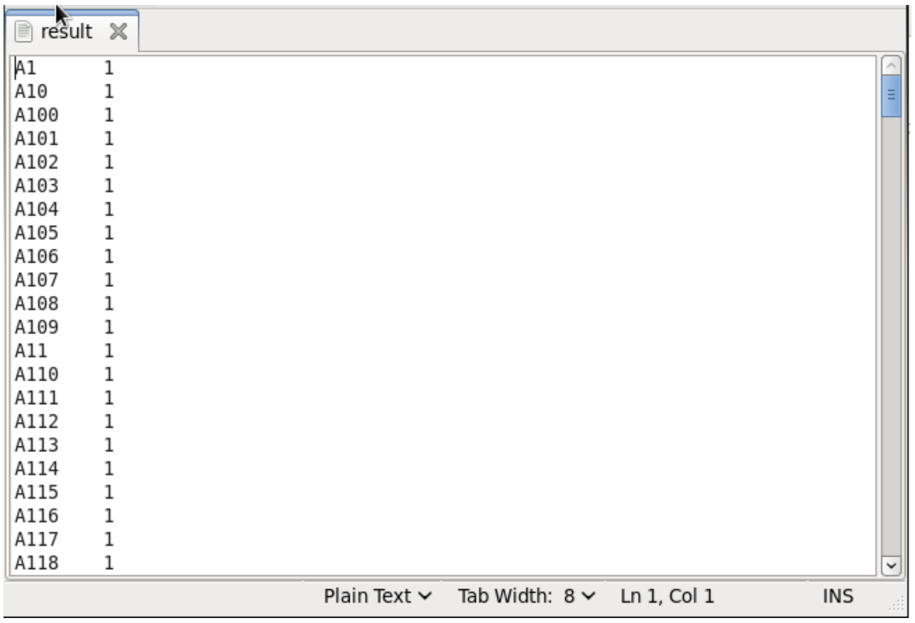


```java
package hints;

import java.io.IOException;


public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

    String line = value.toString();
    String[] column;
    column = line.split("\\t");
    String lang_code = column[0]; //first column
    context.write(new Text(lang_code), new IntWritable(1));

      }
    }
```



```java
package hints;

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

public class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

  @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int wordCount = 0;

        for (IntWritable value : values) {

            wordCount += value.get();
        }

        context.write(key, new IntWritable(wordCount));
    }
}
```

Outcome:



```
A1      1
A10     1
A100    1
A101    1
A102    1
A103    1
A104    1
A105    1
A106    1
A107    1
A108    1
A109    1
A11     1
A110    1
A111    1
A112    1
A113    1
A114    1
A115    1
A116    1
A117    1
A118    1
```

[Phase 2 – SQL]

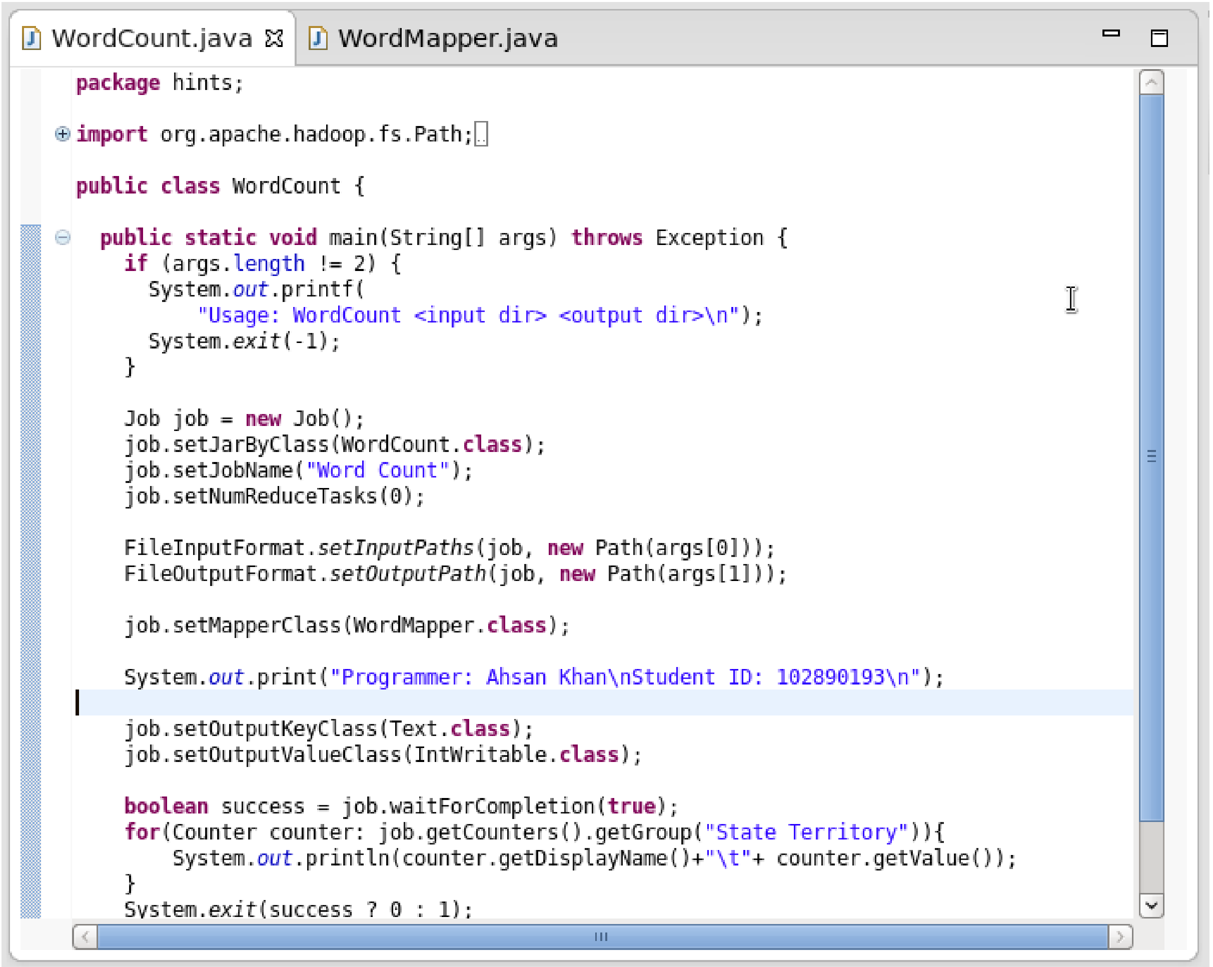
```
mysql> LOAD DATA INFILE '/home/training/training_materials/dataset/result' INTO
TABLE first_qs FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n';
Query OK, 1209 rows affected (0.00 sec)
Records: 1209  Deleted: 0  Skipped: 0  Warnings: 0
```

Outcome:

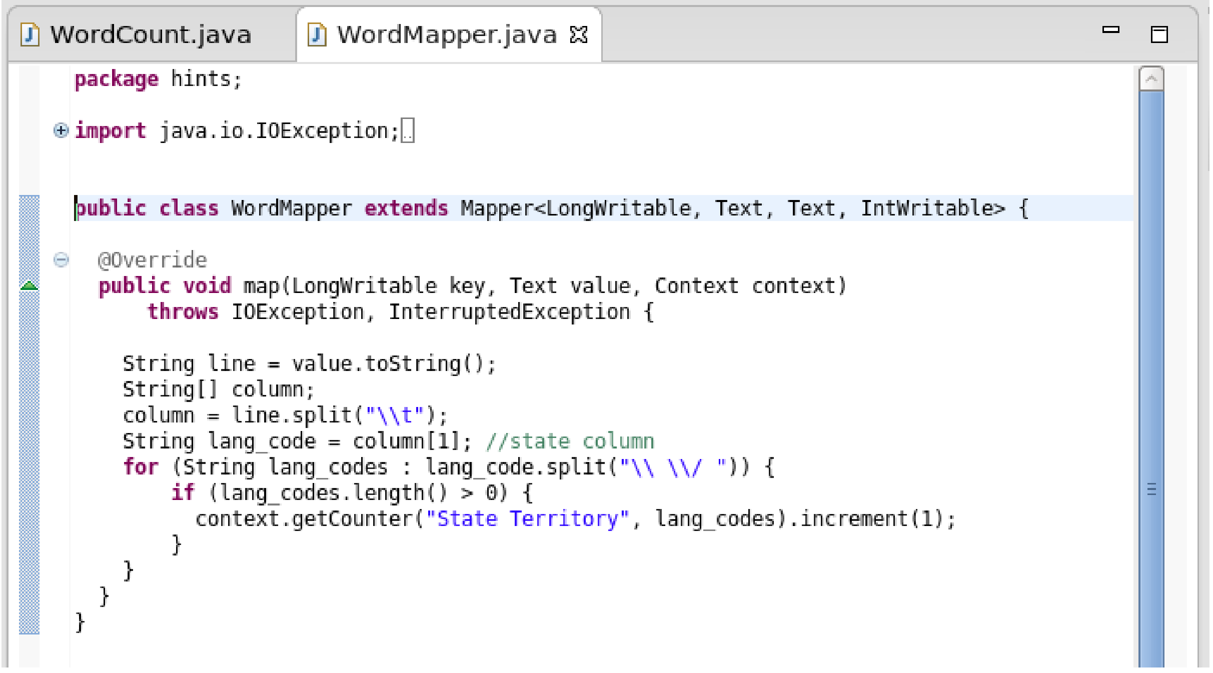
```
mysql> SELECT lang_code, COUNT(lang_code) FROM first_qs GROUP BY lang_code HAVIN
G COUNT(lang_code) > 1;
Empty set (0.00 sec)
```

Since there is no result for a lang_code that has a count greater than 1, this indicates that all of the lang_code values are unique.

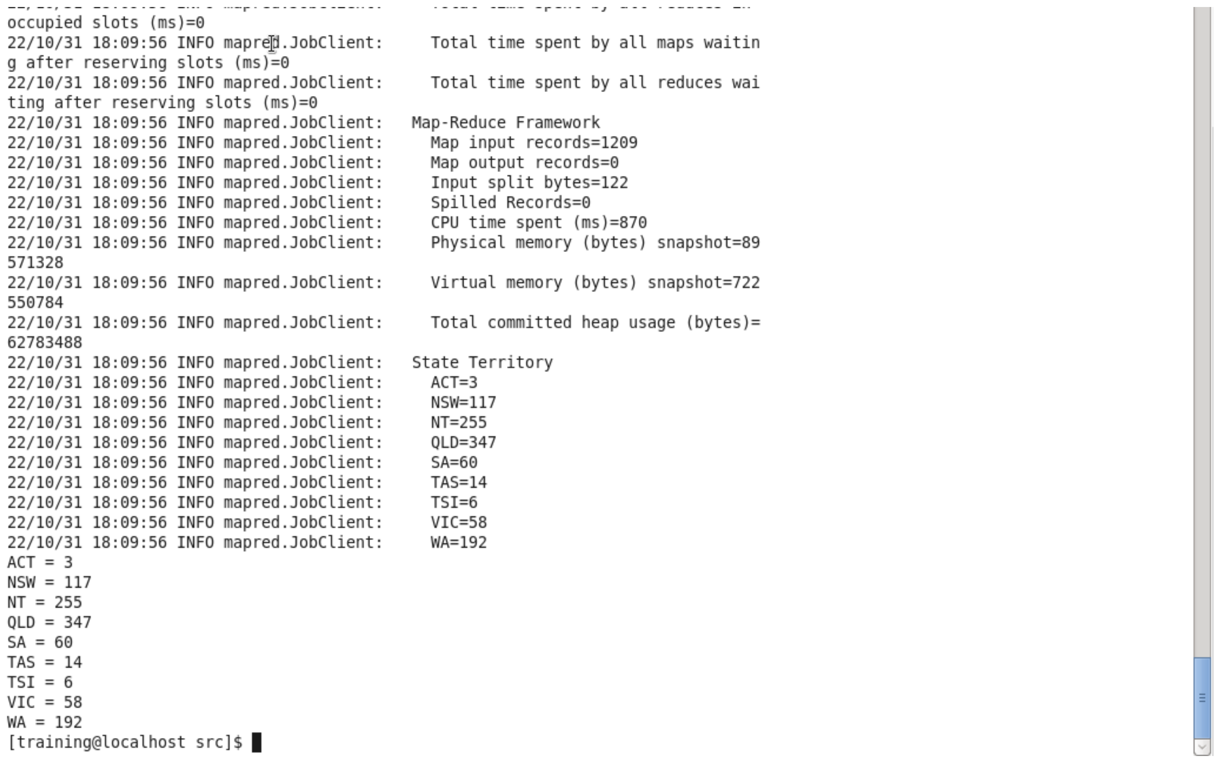
2. Find a way with tools taught in this unit to list the unique values for all entities (tables with the solid boundary) except lng_id.
   2.1. What is the best tool of choice to perform this task?
   Ans: MapReduce
   2.2. Assume you choose MapReduce to be the tool. Should it be the Map-only design or the complete MapReduce with mapper and reducer involved in the process?
   Ans: Map-only design
   2.3. Assume you choose MapReduce to be the tool. Does the count in the output matter for preparing the data for creating the entities?
   Ans: No because we only need to know the unique values for all entities, not their counts.
   2.4. List the code/command/statement and the outcome screenshot of preparing the data for lng_st.
   Ans:



```
package hints;

import org.apache.hadoop.fs.Path;

public class WordCount {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.out.printf(
          "Usage: WordCount <input dir> <output dir>\n");
      System.exit(-1);
    }

    Job job = new Job();
    job.setJarByClass(WordCount.class);
    job.setJobName("Word Count");
    job.setNumReduceTasks(0);

    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapperClass(WordMapper.class);

    System.out.print("Programmer: Ahsan Khan\nStudent ID: 102890193\n");

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

    boolean success = job.waitForCompletion(true);
    for(Counter counter: job.getCounters().getGroup("State Territory")){
        System.out.println(counter.getDisplayName()+"\t"+ counter.getValue());
    }
    System.exit(success ? 0 : 1);
```



```
package hints;

import java.io.IOException;


public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

    String line = value.toString();
    String[] column;
    column = line.split("\\t");
    String lang_code = column[1]; //state column
    for (String lang_codes : lang_code.split("\\ \\/ ")) {
        if (lang_codes.length() > 0) {
          context.getCounter("State Territory", lang_codes).increment(1);
        }
    }
  }
}
```



```
occupied slots (ms)=0
22/10/31 18:09:56 INFO mapred.JobClient:     Total time spent by all maps waitin
g after reserving slots (ms)=0
22/10/31 18:09:56 INFO mapred.JobClient:     Total time spent by all reduces wai
ting after reserving slots (ms)=0
22/10/31 18:09:56 INFO mapred.JobClient:   Map-Reduce Framework
22/10/31 18:09:56 INFO mapred.JobClient:     Map input records=1209
22/10/31 18:09:56 INFO mapred.JobClient:     Map output records=0
22/10/31 18:09:56 INFO mapred.JobClient:     Input split bytes=122
22/10/31 18:09:56 INFO mapred.JobClient:     Spilled Records=0
22/10/31 18:09:56 INFO mapred.JobClient:     CPU time spent (ms)=870
22/10/31 18:09:56 INFO mapred.JobClient:     Physical memory (bytes) snapshot=89
571328
22/10/31 18:09:56 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=722
550784
22/10/31 18:09:56 INFO mapred.JobClient:     Total committed heap usage (bytes)=
62783488
22/10/31 18:09:56 INFO mapred.JobClient:   State Territory
22/10/31 18:09:56 INFO mapred.JobClient:     ACT=3
22/10/31 18:09:56 INFO mapred.JobClient:     NSW=117
22/10/31 18:09:56 INFO mapred.JobClient:     NT=255
22/10/31 18:09:56 INFO mapred.JobClient:     QLD=347
22/10/31 18:09:56 INFO mapred.JobClient:     SA=60
22/10/31 18:09:56 INFO mapred.JobClient:     TAS=14
22/10/31 18:09:56 INFO mapred.JobClient:     TSI=6
22/10/31 18:09:56 INFO mapred.JobClient:     VIC=58
22/10/31 18:09:56 INFO mapred.JobClient:     WA=192
ACT = 3
NSW = 117
NT = 255
QLD = 347
SA = 60
TAS = 14
TSI = 6
VIC = 58
WA = 192
[training@localhost src]$
```

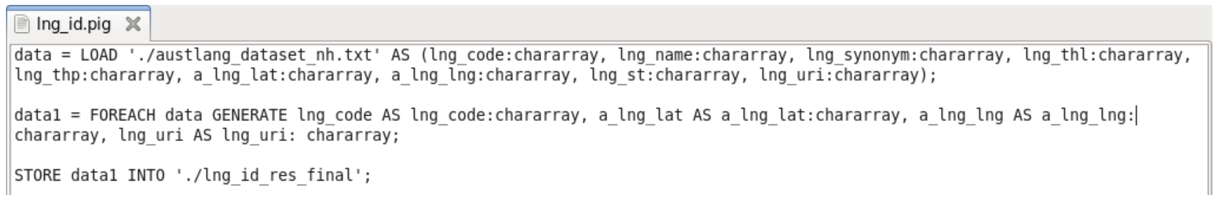
2.5. How many counts does QLD have?
Ans: 347

3. Find a way with tools taught in this unit to prepare the data for all entities (tables with the solid boundary) for MySQL and Hive.

3.1. Assume your tool of choice is MapReduce for this task. Will this be a Map-only job or a complete MapReduce job for preparing the desired data?
Ans: Map-only job

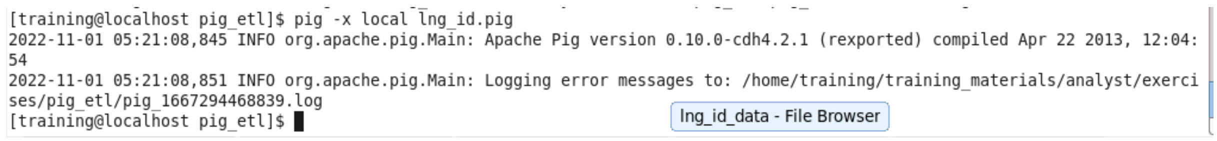
3.2. List the code/command/statement and the outcome screenshot of preparing the data for lng_id.
Ans:

Code -



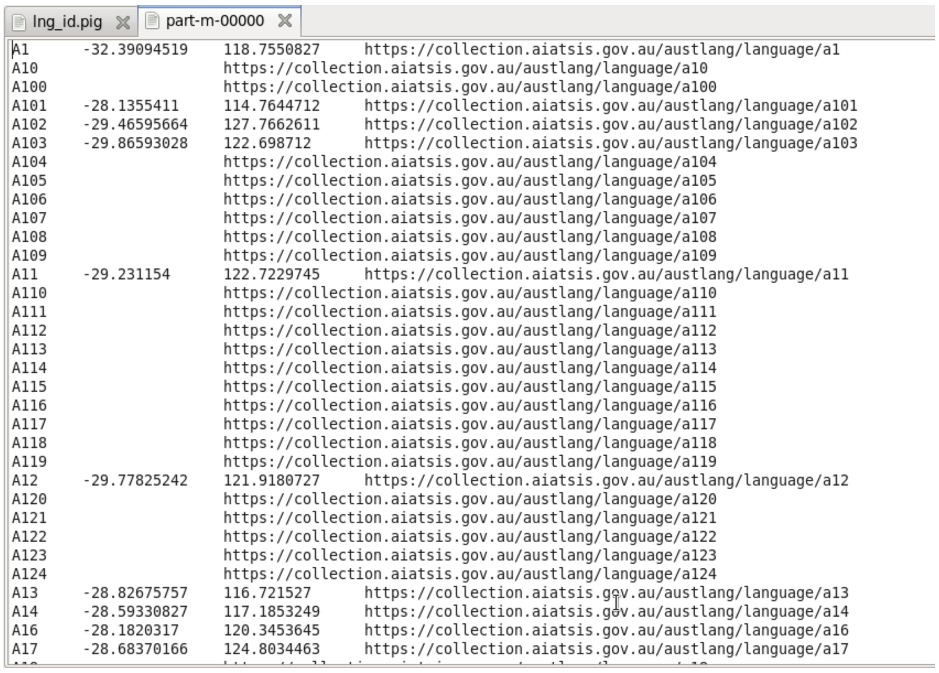
```
data = LOAD './austlang_dataset_nh.txt' AS (lng_code:chararray, lng_name:chararray, lng_synonym:chararray, lng_thl:chararray,
lng_thp:chararray, a_lng_lat:chararray, a_lng_lng:chararray, lng_st:chararray, lng_uri:chararray);

data1 = FOREACH data GENERATE lng_code AS lng_code:chararray, a_lng_lat AS a_lng_lat:chararray, a_lng_lng AS a_lng_lng:
chararray, lng_uri AS lng_uri: chararray;

STORE data1 INTO './lng_id_res_final';
```

Command -



```
[training@localhost pig_etl]$ pig -x local lng_id.pig
2022-11-01 05:21:08,845 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4.2.1 (rexported) compiled Apr 22 2013, 12:04:
54
2022-11-01 05:21:08,851 INFO org.apache.pig.Main: Logging error messages to: /home/training/training_materials/analyst/exerci
ses/pig_etl/pig_1667294468839.log
[training@localhost pig_etl]$
```

Result –

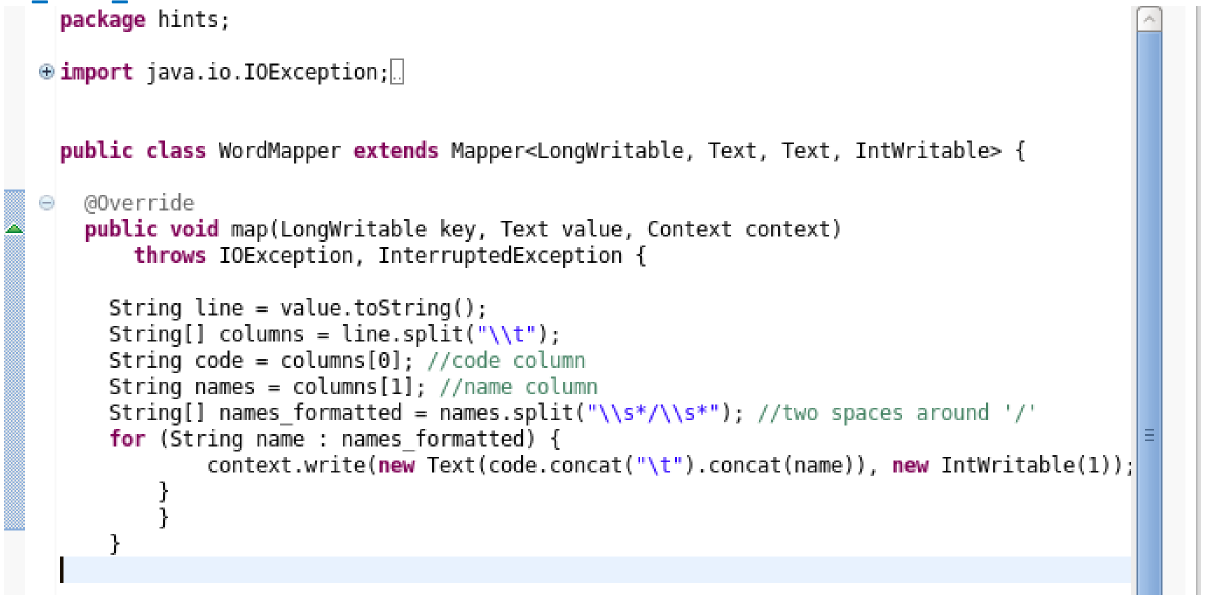


```
A1      -32.39094519    118.7550827     https://collection.aiatsis.gov.au/austlang/language/a1
A10                     https://collection.aiatsis.gov.au/austlang/language/a10
A100                    https://collection.aiatsis.gov.au/austlang/language/a100
A101    -28.1355411     114.7644712     https://collection.aiatsis.gov.au/austlang/language/a101
A102    -29.46595664    127.7662611     https://collection.aiatsis.gov.au/austlang/language/a102
A103    -29.86593028    122.698712      https://collection.aiatsis.gov.au/austlang/language/a103
A104                    https://collection.aiatsis.gov.au/austlang/language/a104
A105                    https://collection.aiatsis.gov.au/austlang/language/a105
A106                    https://collection.aiatsis.gov.au/austlang/language/a106
A107                    https://collection.aiatsis.gov.au/austlang/language/a107
A108                    https://collection.aiatsis.gov.au/austlang/language/a108
A109                    https://collection.aiatsis.gov.au/austlang/language/a109
A11     -29.231154      122.7229745     https://collection.aiatsis.gov.au/austlang/language/a11
A110                    https://collection.aiatsis.gov.au/austlang/language/a110
A111                    https://collection.aiatsis.gov.au/austlang/language/a111
A112                    https://collection.aiatsis.gov.au/austlang/language/a112
A113                    https://collection.aiatsis.gov.au/austlang/language/a113
A114                    https://collection.aiatsis.gov.au/austlang/language/a114
A115                    https://collection.aiatsis.gov.au/austlang/language/a115
A116                    https://collection.aiatsis.gov.au/austlang/language/a116
A117                    https://collection.aiatsis.gov.au/austlang/language/a117
A118                    https://collection.aiatsis.gov.au/austlang/language/a118
A119                    https://collection.aiatsis.gov.au/austlang/language/a119
A12     -29.77825242    121.9180727     https://collection.aiatsis.gov.au/austlang/language/a12
A120                    https://collection.aiatsis.gov.au/austlang/language/a120
A121                    https://collection.aiatsis.gov.au/austlang/language/a121
A122                    https://collection.aiatsis.gov.au/austlang/language/a122
A123                    https://collection.aiatsis.gov.au/austlang/language/a123
A124                    https://collection.aiatsis.gov.au/austlang/language/a124
A13     -28.82675757    116.721527      https://collection.aiatsis.gov.au/austlang/language/a13
A14     -28.59330827    117.1853249     https://collection.aiatsis.gov.au/austlang/language/a14
A16     -28.1820317     120.3453645     https://collection.aiatsis.gov.au/austlang/language/a16
A17     -28.68370166    124.8034463     https://collection.aiatsis.gov.au/austlang/language/a17
```

4. Find a way with tools taught in this unit to prepare the data for all weak entities (tables with the dashed boundary) for MySQL and Hive.
   4.1. Which tool would be the best choice for handling this task?
   Ans: Map-only

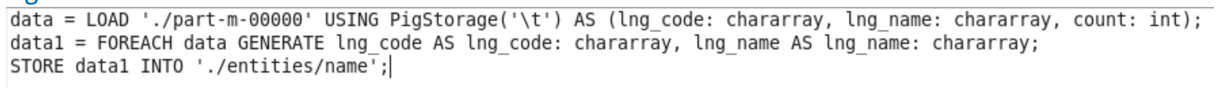
   4.2. List the code/statement of preparing the data for all weak entities.
   Ans:

Rel_code_name code:



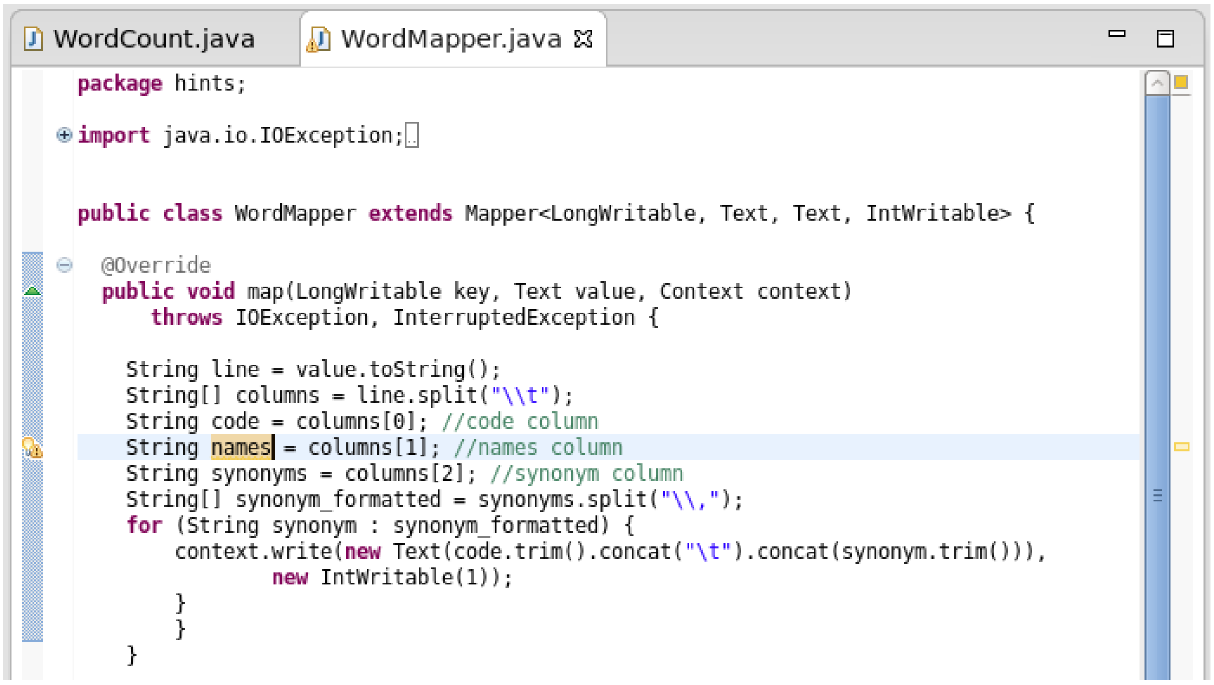
```
package hints;

import java.io.IOException;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

      String line = value.toString();
      String[] columns = line.split("\\t");
      String code = columns[0]; //code column
      String names = columns[1]; //name column
      String[] names_formatted = names.split("\\s*/\\s*"); //two spaces around '/'
      for (String name : names_formatted) {
              context.write(new Text(code.concat("\t").concat(name)), new IntWritable(1));
          }
          }
      }
```

Pig code to remove the count:



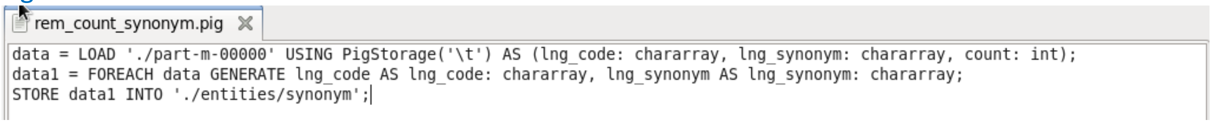
```
data = LOAD './part-m-00000' USING PigStorage('\t') AS (lng_code: chararray, lng_name: chararray, count: int);
data1 = FOREACH data GENERATE lng_code AS lng_code: chararray, lng_name AS lng_name: chararray;
STORE data1 INTO './entities/name';
```

Rel_code_synonym code:



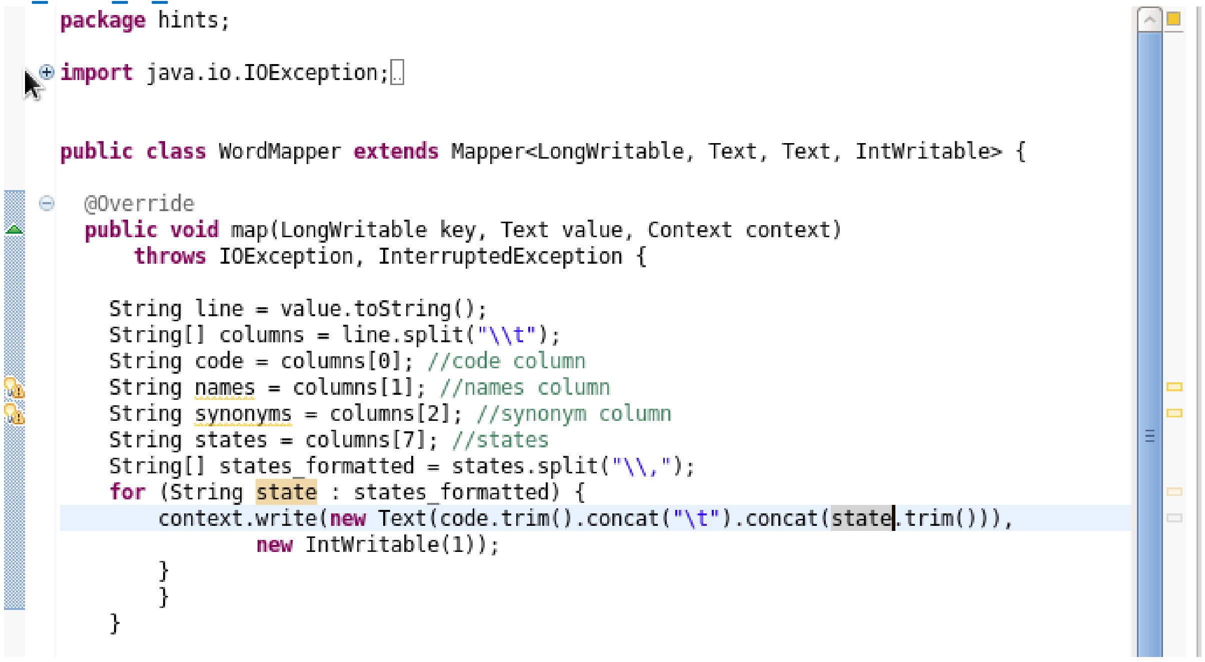
```java
package hints;

import java.io.IOException;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

    String line = value.toString();
    String[] columns = line.split("\\t");
    String code = columns[0]; //code column
    String names = columns[1]; //names column
    String synonyms = columns[2]; //synonym column
    String[] synonym_formatted = synonyms.split("\\,");
    for (String synonym : synonym_formatted) {
        context.write(new Text(code.trim().concat("\t").concat(synonym.trim())),
                new IntWritable(1));
        }
        }
    }
```

Pig code to remove the count:



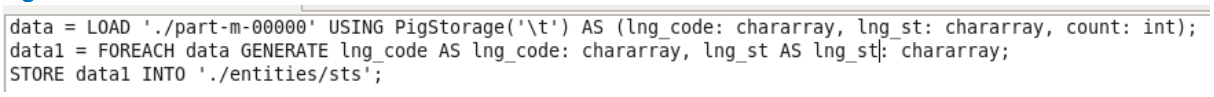
```
data = LOAD './part-m-00000' USING PigStorage('\t') AS (lng_code: chararray, lng_synonym: chararray, count: int);
data1 = FOREACH data GENERATE lng_code AS lng_code: chararray, lng_synonym AS lng_synonym: chararray;
STORE data1 INTO './entities/synonym';
```

Rel_code_st_code:



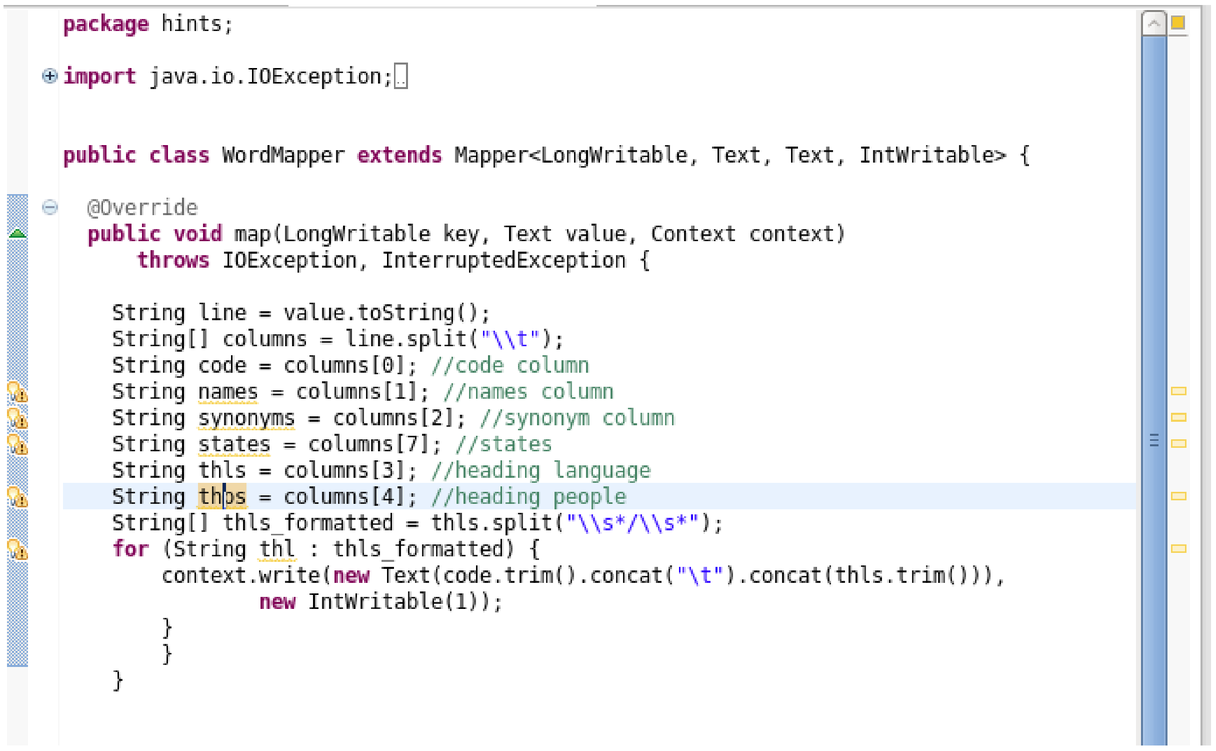
```java
package hints;

import java.io.IOException;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

    String line = value.toString();
    String[] columns = line.split("\\t");
    String code = columns[0]; //code column
    String names = columns[1]; //names column
    String synonyms = columns[2]; //synonym column
    String states = columns[7]; //states
    String[] states_formatted = states.split("\\,");
    for (String state : states_formatted) {
        context.write(new Text(code.trim().concat("\t").concat(state.trim())),
                new IntWritable(1));
        }
        }
    }
```

Pig code to remove the count:



```
data = LOAD './part-m-00000' USING PigStorage('\t') AS (lng_code: chararray, lng_st: chararray, count: int);
data1 = FOREACH data GENERATE lng_code AS lng_code: chararray, lng_st AS lng_st: chararray;
STORE data1 INTO './entities/sts';
```

Rel_code_thl code:

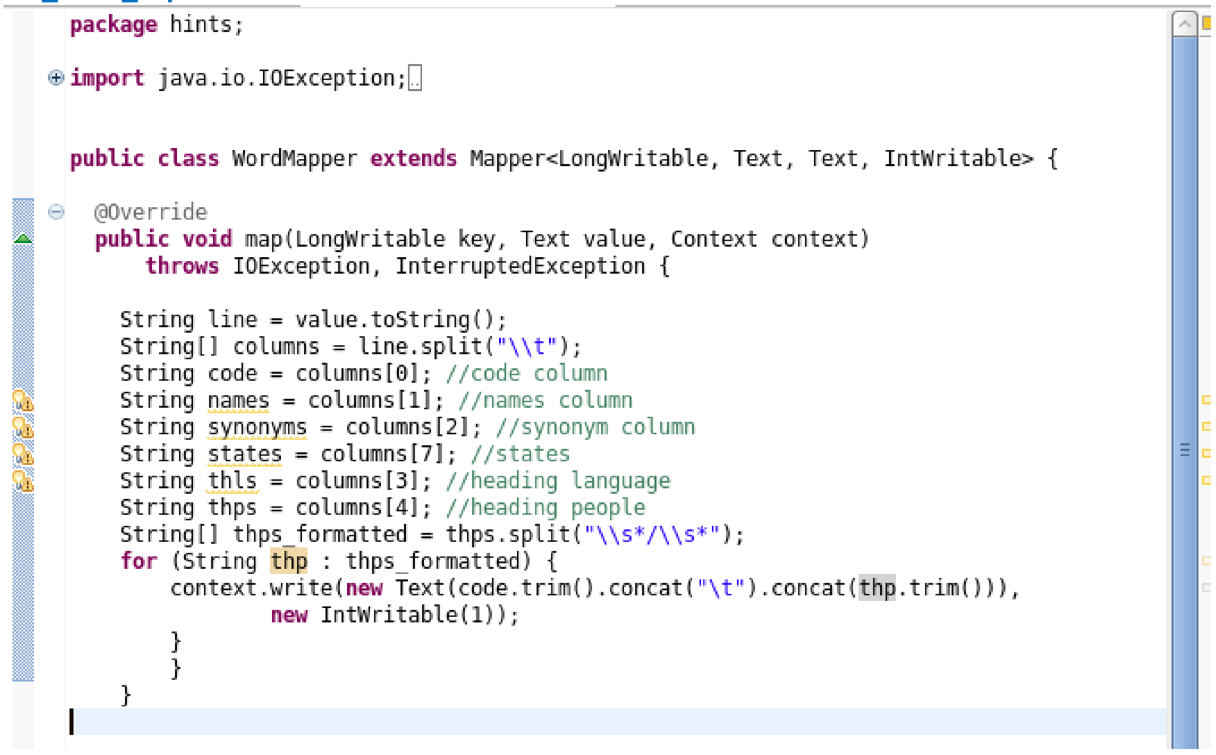


```
package hints;

import java.io.IOException;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

        String line = value.toString();
        String[] columns = line.split("\\t");
        String code = columns[0]; //code column
        String names = columns[1]; //names column
        String synonyms = columns[2]; //synonym column
        String states = columns[7]; //states
        String thls = columns[3]; //heading language
        String thps = columns[4]; //heading people
        String[] thls_formatted = thls.split("\\s*/\\s*");
        for (String thl : thls_formatted) {
            context.write(new Text(code.trim().concat("\t").concat(thls.trim())),
                    new IntWritable(1));
            }
            }
        }
```

Pig code to remove the count:

```
data = LOAD './part-m-00000' USING PigStorage('\t') AS (lng_code: chararray, lng_thl: chararray, count: int);
data1 = FOREACH data GENERATE lng_code AS lng_code: chararray, lng_thl AS lng_thl: chararray;
STORE data1 INTO './entities/thl';
```

Rel_code_thp code:



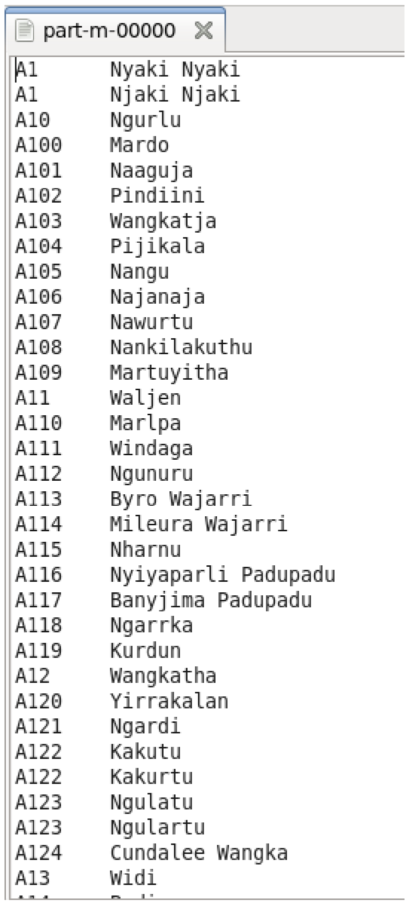
```
package hints;

import java.io.IOException;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

        String line = value.toString();
        String[] columns = line.split("\\t");
        String code = columns[0]; //code column
        String names = columns[1]; //names column
        String synonyms = columns[2]; //synonym column
        String states = columns[7]; //states
        String thls = columns[3]; //heading language
        String thps = columns[4]; //heading people
        String[] thps_formatted = thps.split("\\s*/\\s*");
        for (String thp : thps_formatted) {
            context.write(new Text(code.trim().concat("\t").concat(thp.trim())),
                    new IntWritable(1));
            }
            }
        }
```

Pig code to remove the count:

```
data = LOAD './part-m-00000' USING PigStorage('\t') AS (lng_code: chararray, lng_thp: chararray, count: int);
data1 = FOREACH data GENERATE lng_code AS lng_code: chararray, lng_thp AS lng_thp: chararray;
STORE data1 INTO './entities/thp';
```

4.3. Post the first-page screenshot of the rel_code_name processed result.

Ans:



part-m-00000

```
A1      Nyaki Nyaki
A1      Njaki Njaki
A10     Ngurlu
A100    Mardo
A101    Naaguja
A102    Pindiini
A103    Wangkatja
A104    Pijikala
A105    Nangu
A106    Najanaja
A107    Nawurtu
A108    Nankilakuthu
A109    Martuyitha
A11     Waljen
A110    Marlpa
A111    Windaga
A112    Ngunuru
A113    Byro Wajarri
A114    Mileura Wajarri
A115    Nharnu
A116    Nyiyaparli Padupadu
A117    Banyjima Padupadu
A118    Ngarrka
A119    Kurdun
A12     Wangkatha
A120    Yirrakalan
A121    Ngardi
A122    Kakutu
A122    Kakurtu
A123    Ngulatu
A123    Ngulartu
A124    Cundalee Wangka
A13     Widi
```

5. Follow the given ERDs, create tables and import data into the created tables. All tables should be put in the database entitled "indigenous".

5.1. List all statements of this task for MySQL:

Ans:

Database:

Use indigenous;

1. Lng_id:

Create Table:

CREATE TABLE lng_id (lng_code VARCHAR(20), a_lng_lat DECIMAL(18,7), a_lng_lng DOUBLE(18,7), lng_uri VARCHAR(300), PRIMARY KEY (lng_code));

Import Data:

LOAD DATA LOCAL INFILE '/home/training/app-proj/id/part-m-00000' INTO TABLE lng_id FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

2. Lng_name

Create Table:

CREATE TABLE lng_name (lng_name VARCHAR(20), PRIMARY KEY (lng_name));

Import Data:

LOAD DATA LOCAL INFILE '/home/training/app-proj/name/part-m-00000' INTO TABLE lng_name FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

3. Lng_synonym

Create Table:

CREATE TABLE lng_synonym (lng_synonym VARCHAR(200), PRIMARY KEY (lng_synonym));

Import Data:

LOAD DATA LOCAL INFILE '/home/training/app-proj/synonym/part-m-00000' INTO TABLE lng_synonym FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

4. Lng_thl
   Create Table:
   CREATE TABLE lng_thl (lng_thl VARCHAR(200), PRIMARY KEY (lng_thl));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/thl/part-m-00000' INTO TABLE lng_thl FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

5. Lng_thp
   Create Table:
   CREATE TABLE lng_thp (lng_thp VARCHAR(200), PRIMARY KEY (lng_thp));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/thp/part-m-00000' INTO TABLE lng_thp FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

6. Lng_st
   Create Table:
   CREATE TABLE lng_st (lng_st VARCHAR(200), PRIMARY KEY (lng_st));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/st/part-m-00000' INTO TABLE lng_st FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

7. Rel_code_name
   Create Table:
   CREATE TABLE rel_code_name (lng_code VARCHAR(20), lng_name VARCHAR(20), IDKey int NOT NULL AUTO_INCREMENT, PRIMARY KEY (IDKey), FOREIGN KEY (lng_name) REFERENCES lng_name(lng_name), FOREIGN KEY (lng_code) REFERENCES lng_id(lng_code));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/code_name/part-m-00000' INTO TABLE rel_code_name FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

8. Rel_code_synonym
   Create Table:
   CREATE TABLE rel_code_synonym (lng_synonym VARCHAR(200), lng_code VARCHAR(20), IDKey int NOT NULL AUTO_INCREMENT, PRIMARY KEY (IDKey), FOREIGN KEY (lng_synonym) REFERENCES lng_synonym(lng_synonym), FOREIGN KEY (lng_code) REFERENCES lng_id(lng_code));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/code_synonym/part-m-00000' INTO TABLE rel_code_synonym FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

9. Rel_code_thl
   Create Table:
   CREATE TABLE rel_code_thl (lng_code VARCHAR(20), lng_thl VARCHAR(200), IDKey int NOT NULL AUTO_INCREMENT, PRIMARY KEY (IDKey), FOREIGN KEY (lng_thl) REFERENCES lng_thl(lng_thl), FOREIGN KEY (lng_code) REFERENCES lng_id(lng_code));
   Import Data:
   LOAD DATA LOCAL INFILE '/home/training/app-proj/code_thl/part-m-00000' INTO TABLE rel_code_thl FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

10. Rel_code_thp
    Create Table:
    CREATE TABLE rel_code_thp (lng_code VARCHAR(20), lng_thp VARCHAR(200), IDKey int NOT NULL AUTO_INCREMENT, PRIMARY KEY (IDKey), FOREIGN KEY (lng_thp) REFERENCES lng_thp(lng_thp), FOREIGN KEY (lng_code) REFERENCES lng_id(lng_code));
    Import Data:
    LOAD DATA LOCAL INFILE '/home/training/app-proj/code_thp/part-m-00000' INTO TABLE rel_code_thp FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

11. Rel_code_st
    Create Table:
    CREATE TABLE rel_code_st (lng_code VARCHAR(20), lng_st VARCHAR(200), IDKey int NOT NULL AUTO_INCREMENT, PRIMARY KEY (IDKey), FOREIGN KEY (lng_st) REFERENCES lng_st(lng_st), FOREIGN KEY (lng_code) REFERENCES lng_id(lng_code));

    Import Data:
    LOAD DATA LOCAL INFILE '/home/training/app-proj/code_st/part-m-00000' INTO TABLE rel_code_st FIELDS TERMINATED BY '\t' ENCLOSED BY "" LINES TERMINATED BY '\n';

5.2. List all statements of this task for Hive:
    Ans:
Database:
    Use indigenous;

1. Lng_name
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.lng_name (lng_name string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    LINES TERMINATED BY '\n' STORED AS TEXTFILE;
    Import Data:
    LOAD DATA INPATH 'lng_name' INTO TABLE indigenous.lng_name;

2. Lng_synonym
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.lng_synonym (lng_synonym string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    LINES TERMINATED BY '\n' STORED AS TEXTFILE;
    Import Data:
    LOAD DATA INPATH 'lng_synonym' INTO TABLE indigenous.lng_synonym;

3. Lng_thl
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.lng_thl (lng_thl string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    LINES TERMINATED BY '\n' STORED AS TEXTFILE;
    Import Data:
    LOAD DATA INPATH 'thl' INTO TABLE indigenous.lng_thl;

4. Lng_thp
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.lng_thp (lng_thp string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    LINES TERMINATED BY '\n' STORED AS TEXTFILE;

Import Data:
LOAD DATA INPATH 'lng_thp' INTO TABLE indigenous.lng_thp;

5. Lng_st
   Create Table:
   CREATE TABLE IF NOT EXISTS indigenous.lng_st (lng_st string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
   LINES TERMINATED BY '\n' STORED AS TEXTFILE;
   Import Data:
   LOAD DATA INPATH 'lng_st' INTO TABLE indigenous.lng_st;

6. Lng_id
   Create Table:
   CREATE TABLE IF NOT EXISTS indigenous.lng_id (lng_code string, a_lng_lat double, a_lng_lng double, lng_uri string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
   Import Data:
   LOAD DATA INPATH 'lng_id' INTO TABLE indigenous.lng_id;

7. Rel_code_name
   Create Table:
   CREATE TABLE IF NOT EXISTS indigenous.rel_code_name (lng_code string, lng_name string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
   Import Data:
   LOAD DATA INPATH 'rel_code_name' INTO TABLE indigenous.rel_code_name;

8. Rel_code_synonym
   Create Table:
   CREATE TABLE IF NOT EXISTS indigenous.rel_code_synonym (lng_code string, lng_synonym string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
   Import Data:
   LOAD DATA INPATH 'rel_code_synonym' INTO TABLE indigenous.rel_code_synonym;

9. Rel_code_thl
   Create Table:
   CREATE TABLE IF NOT EXISTS indigenous.rel_code_thl (lng_code string, lng_thl string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
   Import Data:
   LOAD DATA INPATH 'rel_code_thl' INTO TABLE indigenous.rel_code_thl;

10. Rel_code_thp
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.rel_code_thp (lng_code string, lng_thp string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
    Import Data:
    LOAD DATA INPATH 'rel_code_thp' INTO TABLE indigenous.rel_code_thp;

11. Rel_code_st
    Create Table:
    CREATE TABLE IF NOT EXISTS indigenous.rel_code_st (lng_code string, lng_st string) ROW

FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

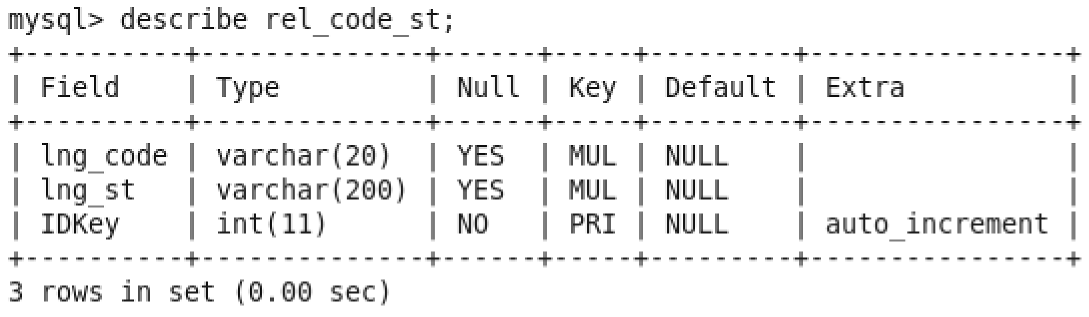
Import Data:

LOAD DATA INPATH 'rel_code_st' INTO TABLE indigenous.rel_code_st;

6. Show the detailed information of table rel_code_st in MySQL.
   6.1. Statement:

   Ans: describe rel_code_st

   6.2. Result screenshot:



```
mysql> describe rel_code_st;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| lng_code | varchar(20)  | YES  | MUL | NULL    |                |
| lng_st   | varchar(200) | YES  | MUL | NULL    |                |
| IDKey    | int(11)      | NO   | PRI | NULL    | auto_increment |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

7. Collect the results containing "lng_code", "lng_name", and "lng_st" of tuples whose lng_name starts with the upper case "D". Show bot statements and the last page screenshot of your query result in MySQL and Hive.
   7.1. MySQL query statement:

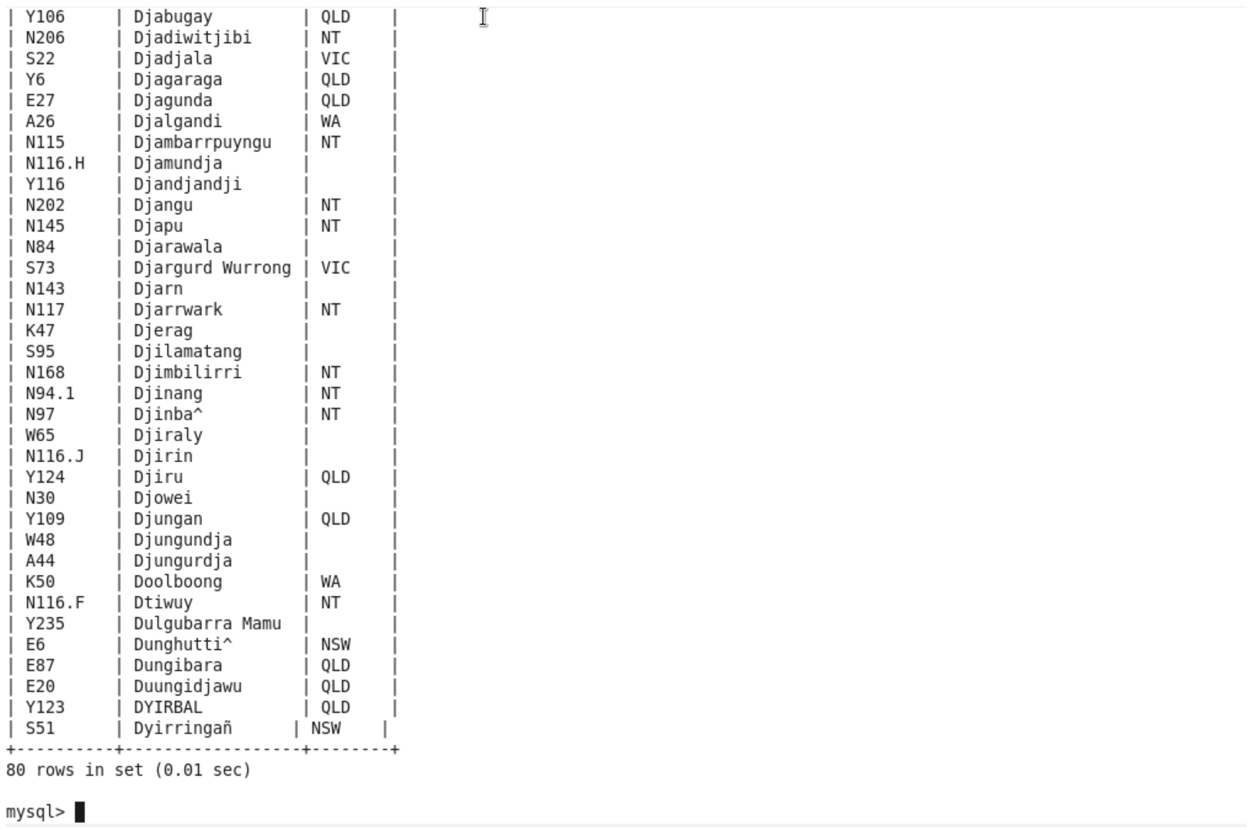
   Ans:

```
SELECT lng_id.lng_code, rel_code_name.lng_name, rel_code_st.lng_st
FROM lng_id
INNER JOIN rel_code_name
ON lng_id.lng_code = rel_code_name.lng_code
INNER JOIN rel_code_st
ON rel_code_st.lng_code = lng_id.lng_code
WHERE rel_code_name.lng_name LIKE "D%";
```

//% at the end to include all the characters after 'D'.

   7.2. MySQL result screenshot:

   Ans:



```
| Y106      | Djabugay          | QLD    |
| N206      | Djadiwitjibi      | NT     |
| S22       | Djadjala          | VIC    |
| Y6        | Djagaraga         | QLD    |
| E27       | Djagunda          | QLD    |
| A26       | Djalgandi         | WA     |
| N115      | Djambarrpuyngu    | NT     |
| N116.H    | Djamundja         |        |
| Y116      | Djandjandji       |        |
| N202      | Djangu            | NT     |
| N145      | Djapu             | NT     |
| N84       | Djarawala         |        |
| S73       | Djargurd Wurrong  | VIC    |
| N143      | Djarn             |        |
| N117      | Djarrwark         | NT     |
| K47       | Djerag            |        |
| S95       | Djilamatang       |        |
| N168      | Djimbilirri       | NT     |
| N94.1     | Djinang           | NT     |
| N97       | Djinba^           | NT     |
| W65       | Djiraly           |        |
| N116.J    | Djirin            |        |
| Y124      | Djiru             | QLD    |
| N30       | Djowei            |        |
| Y109      | Djungan           | QLD    |
| W48       | Djungundja        |        |
| A44       | Djungurdja        |        |
| K50       | Doolboong         | WA     |
| N116.F    | Dtiwuy            | NT     |
| Y235      | Dulgubarra Mamu   |        |
| E6        | Dunghutti^        | NSW    |
| E87       | Dungibara         | QLD    |
| E20       | Duungidjawu       | QLD    |
| Y123      | DYIRBAL           | QLD    |
| S51       | Dyirringañ       | NSW    |
+-----------+-------------------+--------+
80 rows in set (0.01 sec)

mysql> 
```

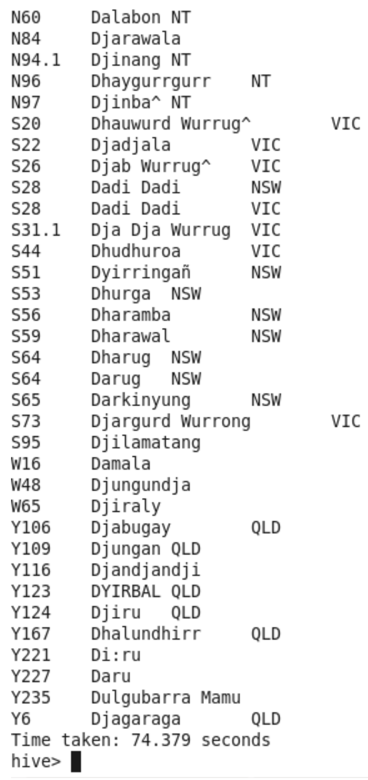
7.3. Hive query statement:

Ans:

SELECT lng_id.lng_code, rel_code_name.lng_name, rel_code_st.lng_st
FROM indigenous.lng_id
INNER JOIN indigenous.rel_code_name
ON lng_id.lng_code = rel_code_name.lng_code
INNER JOIN indigenous.rel_code_st
ON rel_code_st.lng_code = lng_id.lng_code
WHERE rel_code_name.lng_name LIKE "D%";

Hive result screenshot:

Ans:

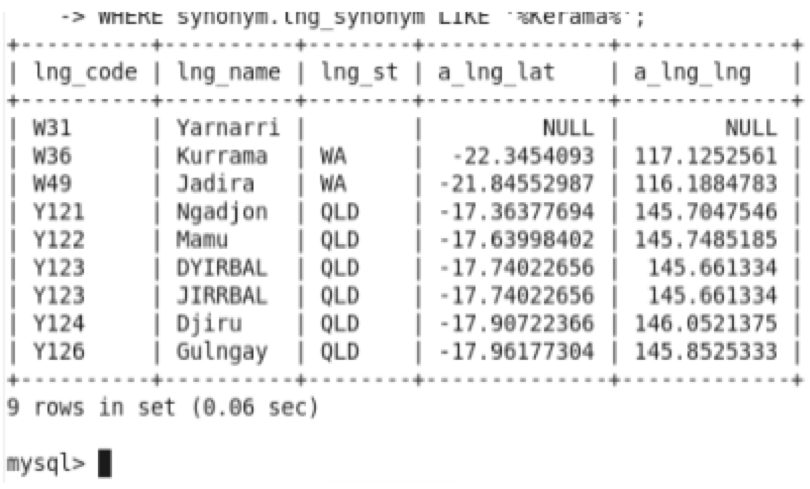


```
N60     Dalabon NT
N84     Djarawala
N94.1   Djinang NT
N96     Dhaygurrgurr    NT
N97     Djinba^ NT
S20     Dhauwurd Wurrug^        VIC
S22     Djadjala        VIC
S26     Djab Wurrug^    VIC
S28     Dadi Dadi       NSW
S28     Dadi Dadi       VIC
S31.1   Dja Dja Wurrug  VIC
S44     Dhudhuroa       VIC
S51     Dyirringañ      NSW
S53     Dhurga  NSW
S56     Dharamba        NSW
S59     Dharawal        NSW
S64     Dharug  NSW
S64     Darug   NSW
S65     Darkinyung      NSW
S73     Djargurd Wurrong        VIC
S95     Djilamatang
W16     Damala
W48     Djungundja
W65     Djiraly
Y106    Djabugay        QLD
Y109    Djungan QLD
Y116    Djandjandji
Y123    DYIRBAL QLD
Y124    Djiru   QLD
Y167    Dhalundhirr     QLD
Y221    Di:ru
Y227    Daru
Y235    Dulgubarra Mamu
Y6      Djagaraga       QLD
Time taken: 74.379 seconds
hive> █
```

8. Collect the results containing “lng_code”, “lng_name”, “lng_st”, “a_lng_lat”, and “a_lng_lng” of tuples whose lng_synonym contains “Kerama”. Show both statements and the last page screenshot of your query result in MySQL and Hive.
   8.1. How many tuples are retrieved at the end?
   Ans: 9

   8.2. MySQL query statement:
   Ans:
   SELECT lng_id.lng_code, rel_code_name.lng_name, rel_code_st.lng_st, lng_id.a_lng_lat, lng_id.a_lng_lng
   FROM lng_id
   INNER JOIN rel_code_name
   ON lng_id.lng_code = rel_code_name.lng_code
   INNER JOIN rel_code_st
   ON lng_id.lng_code = rel_code_st.lng_code
   INNER JOIN rel_code_synonym
   ON lng_id.lng_code = rel_code_synonym.lng_code
   WHERE rel_code_synonym.lng_synonym LIKE "%Kerama%";

   8.3. MySQL result screenshot:
   Ans:



```
    -> WHERE synonym.lng_synonym LIKE '%Kerama%';
+----------+----------+--------+--------------+-------------+
| lng_code | lng_name | lng_st | a_lng_lat    | a_lng_lng   |
+----------+----------+--------+--------------+-------------+
| W31      | Yarnarri |        |         NULL |        NULL |
| W36      | Kurrama  | WA     |  -22.3454093 | 117.1252561 |
| W49      | Jadira   | WA     | -21.84552987 | 116.1884783 |
| Y121     | Ngadjon  | QLD    | -17.36377694 | 145.7047546 |
| Y122     | Mamu     | QLD    | -17.63998402 | 145.7485185 |
| Y123     | DYIRBAL  | QLD    | -17.74022656 |  145.661334 |
| Y123     | JIRRBAL  | QLD    | -17.74022656 |  145.661334 |
| Y124     | Djiru    | QLD    | -17.90722366 | 146.0521375 |
| Y126     | Gulngay  | QLD    | -17.96177304 | 145.8525333 |
+----------+----------+--------+--------------+-------------+
9 rows in set (0.06 sec)

mysql> 
```

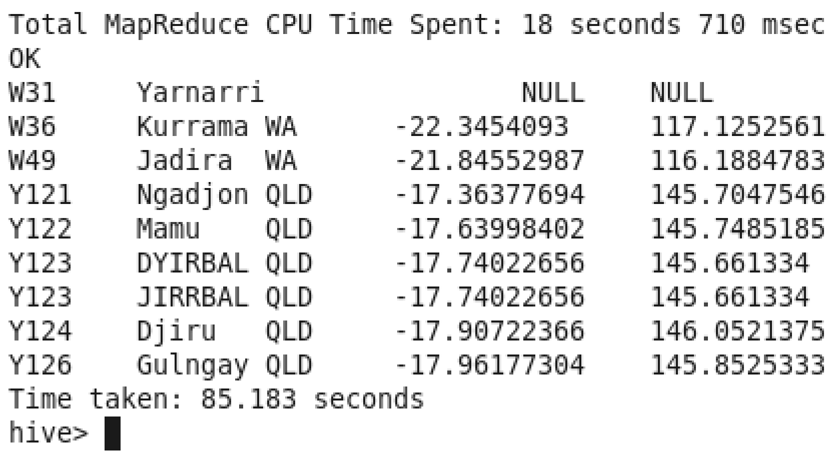
8.4. Hive query statement:

Ans:

SELECT lng_id.lng_code, rel_code_name.lng_name, rel_code_st.lng_st, lng_id.a_lng_lat, lng_id.a_lng_lng
FROM indigenous.lng_id
INNER JOIN indigenous.rel_code_name
ON lng_id.lng_code = rel_code_name.lng_code
INNER JOIN indigenous.rel_code_st
ON lng_id.lng_code = rel_code_st.lng_code
INNER JOIN indigenous.rel_code_synonym
ON lng_id.lng_code = rel_code_synonym.lng_code
WHERE rel_code_synonym.lng_synonym LIKE "%Kerama%";

8.5. Hive result screenshot:

Ans:



```
Total MapReduce CPU Time Spent: 18 seconds 710 msec
OK
W31     Yarnarri                NULL    NULL
W36     Kurrama WA      -22.3454093     117.1252561
W49     Jadira  WA      -21.84552987    116.1884783
Y121    Ngadjon QLD     -17.36377694    145.7047546
Y122    Mamu    QLD     -17.63998402    145.7485185
Y123    DYIRBAL QLD     -17.74022656    145.661334
Y123    JIRRBAL QLD     -17.74022656    145.661334
Y124    Djiru   QLD     -17.90722366    146.0521375
Y126    Gulngay QLD     -17.96177304    145.8525333
Time taken: 85.183 seconds
hive> 
```