CS542 Project III report

This project aimed to build a query execution engine by implementing requirements in course website.

First, we need a relation class, representing table in database. As project I, we use hashMap to store records. In this case, we use cityCode as key in city table and countryCode as key in country table. When user wants to create a table, he needs to pass in a file name so the program knows where to find data. After that, call open() to read desired data into hashmap. If user wants to iterate all records, call getNext() method. If this table is no longer used, call close() to clear main memory where is occupied by this table.
Open(): After initializing relation class, user need call open method to read the desired data file into hashmap.
GetNext(): iterating relation and return next row if found, otherwise return null.
Close(): End this process.

Second, operators are also important to a database. For test in our course website, SQL is like following:

```
Select city.*
From city join country
On city.countrycode=country.code
And city.population>country.population*0.4;
```

From this syntax, we can see only two operators are needed: Join and project. However, in this case, implementing a project class is not necessary because only one method, printing out qualified city information, is needed in project class. So, we only implement join class in terms of operators.

When initializing join object, user needs to pass in two relations that are being joined. For theta Join, there are some conditions. In order to simplify our model, we add these desired conditions directly in join class rather than asking users to offer constraints. In this case, conditions are: city.countrycode=country.code and city.population>country.population*0.4. The conditions in this case are designed and implemented as following: For each record in city, find its countryCode that is stored in city table. Find associated country record for given countryCode and get its population. Then compare population of this city and population of the country. If city.population is greater than country.population times 0.4, print out this city record. Then call City.getNext() to read next record.

The last class is test class, it contain the main function and the file path of our two relations.

```
public static void main(String[] args) throws IOException{
    String path="/Users/HuanYe/Documents/workspace/DBTEST/src/DB3/";
    // read data from db
    Relation citys=new Relation(path+"city.txt",0);
    Relation countrys=new Relation(path+"country.txt",0);

    citys.open();
    countrys.open();
    /* when calling join.open, join.getNext() will be triggered automatically
     * and all qualified record will be printed out.
     */
    Join joinResult=new Join(citys, countrys);
    joinResult.open();

    // after finishing, close all stuff.
    joinResult.close();
    countrys.close();
    citys.close();
    }
```

Testing result:

```
2316  Bantam  CCK  Home Island  503
2454  Macao  MAC  Macau  437500
2453  El-Aaiún  ESH  El-Aaiún  169000
2507  Dalap-Uliga-Darrit  MHL  Majuro  28000
3067  Saint-Pierre  SPM  Saint-Pierre  5808
553  George Town  CYM  Grand Cayman  19600
583  Avarua  COK  Rarotonga  11900
585  Djibouti  DJI  Djibouti  383000
3208  Singapore  SGP  4017733
3206  Victoria  SYC  Mahé  41000
2881  Koror  PLW  Koror  12000
763  Stanley  FLK  East Falkland  1636
2912  Adamstown  PCN  42
2973  Doha  QAT  Doha  355000
915  Gibraltar  GIB  27025
3538  Città del Vaticano  VAT  455
938  Longyearbyen  SJM  Länsimaa  1438
148  Nassau  BHS  New Providence  172000
```