Design of Query Execution Engine

Author: Kuang Xiong, Tengyang Jia, Mei Yang

Assumption

The select condition are set by hard code.
The left relation of the join is the smaller table(in this project is the city.csv), the right relation of the join is the relatively bigger table(the country.csv in this project).

Design

Here is the brief introduction of Query Execution Engine Design . We have these java files in this project: Tables.java, Main.java and Join.java.

Table.java

The “tables” class represent a relation by holding a pointer to the relation csv file and implementing open(), getNext() and close() methods.
open(): open the csv file and loaded it to a String array list
getnext(): get the next row of the relation
close():close and clear all the resources we allocated for the process

Join.java

Join two relations under certain condition. Using iterator methods for tuple-based nested join.

```
open {
   left.open()
   right.open()
}

getNext {
   if (right == null) {
      left.getNext()
      if (left.getNext() is null);
         close()
      right.close()
      right.open()
   }
   if (t in Right match t in Left) {
```

```
        build the union and output
    }
}

close {
    left.close()
    right.close()
}
```

Main.java

Main method to execute query with tables stored in csv files and get right result with iterator method.

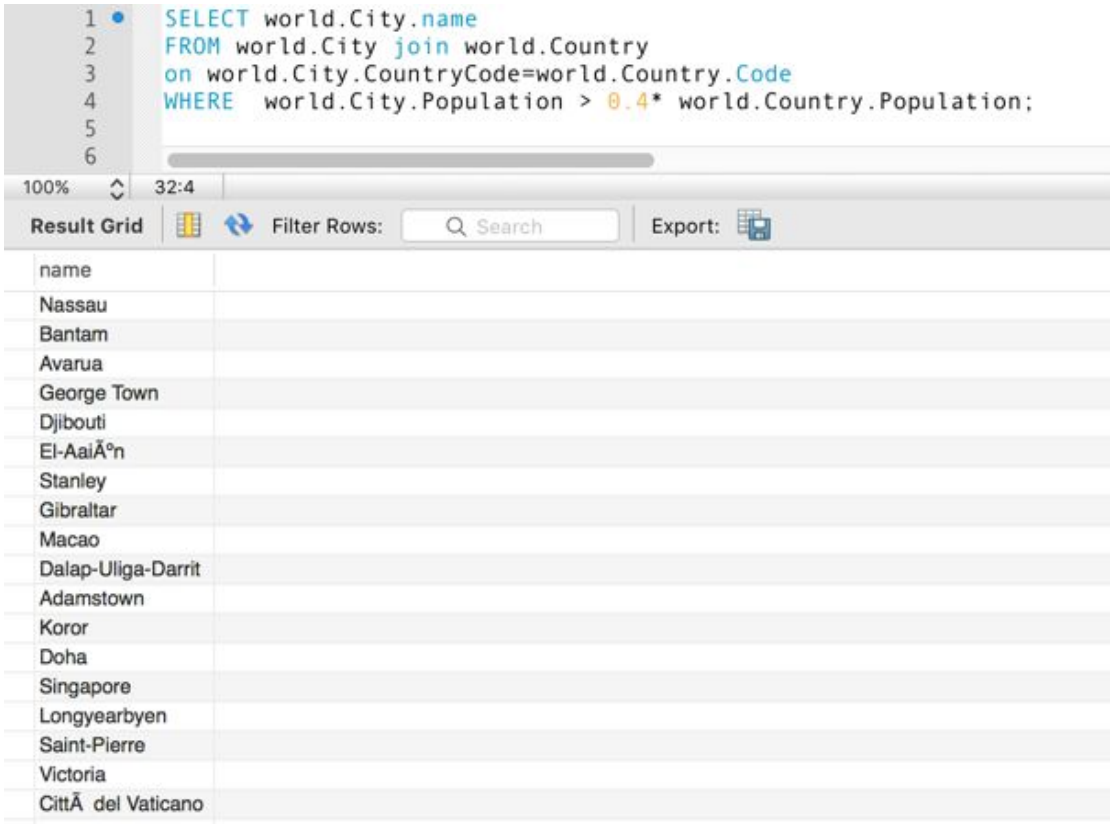
The pipeline:
Every getNext() of the Join will give a joined tuple, we will verify this tuple immediately to see if it meets the selected condition. You will see the result is printed one by one in the console.

Validation:

Select all the cities in the City.csv file and Country.csv file whose population is more than 40% of the population of their entire country.
After run the main.java of this project, we get the result as follows:

```
Here is all the cities whose population is more than 40% of the population of their entire country:
'Nassau'
'George Town'
'Avarua'
'Djibouti'
'Stanley'
'Gibraltar'
'Longyearbyen'
'Bantam'
'El-Aaiún'
'Macao'
'Dalap-Uliga-Darrit'
'Koror'
'Adamstown'
'Doha'
'Saint-Pierre'
'Victoria'
'Singapore'
'Città del Vaticano'
The End
```

Here is the result from the MySQLWorkbench, the results are the same except they are in different sequence.



```
1  SELECT world.City.name
2  FROM world.City join world.Country
3  on world.City.CountryCode=world.Country.Code
4  WHERE  world.City.Population > 0.4* world.Country.Population;
5
6
```

100% 32:4

Result Grid | Filter Rows: Search | Export:

name
Nassau
Bantam
Avarua
George Town
Djibouti
El-AaiÃºn
Stanley
Gibraltar
Macao
Dalap-Uliga-Darrit
Adamstown
Koror
Doha
Singapore
Longyearbyen
Saint-Pierre
Victoria
CittÃ del Vaticano