

# Internals Project #3 Documentation

Chen Zou, Jiaqi Liu,  
[czou@wpi.edu](mailto:czou@wpi.edu), [jliu6@wpi.edu](mailto:jliu6@wpi.edu)

- **Assumption:**
  - Based on the case 1, and using the data, we implement the additional function which work for dealing with multiple table, which is city and country table, and find all cities whose population is more than 40% of the national population.
  - We assume that we have sufficient information about two tables and only need to do work related to specific request instead of general cases.
  - We assume all data as a whole will fit in main memory
  - Because if we use naive algorithm, which is to scan all city table if we are holding one tuple from country table, the workload is too much and the system will be definitely too slow.
  - So we use thought from hash map to implement the system, and the performance algorithm will be  $O(1)$  to find cities which belong to individual country.
- **Design decision:**
  - **Open()**
    - Set up two FakeDict instances according to database files in disks (if exists). If not, the system will create them. Both instances use distinct country codes as their keys. More details about their structures in demonstrated in Data Structure part.
    - Pipeline consists of two FakeDict instances and working as below.
  - **getNext()**
    - The getNext() operator is to read the city and country data, which is gotten from open() operator.
    - Then, find the city cities whose population have more than 40% national population.
    - Because the data in the world database is not as complete as we thought. After we finish the program, we found some data of city do not have district value, so we change the “\N” record to “no record” to clearly display.
    - Finally, in order to display the result we show the result by employing the PrettyTable.
  - **Close()**
    - Close two FakeDict instances and leave other variables to python garbage collector to deal with.
- **Data Structure**
  - **City**
    - {country1\_code: [[city1\_info], [city2\_info]],  
country2\_code: [[city1\_info], [city2\_info]]}
  - **Country**
    - {country1\_code: [country1\_info],  
country2\_code: [country2\_info]}