Internals Project #4 Documentation

Chen Zou, Jiaqi Liu,
czou@wpi.edu, jliu6@wpi.edu

- Assumption:
  - Based on the project 1 and project 3. We revise the Relation class, integrating two table in one database file and implements several APIs for the task. A new class Logging is also created especially for recording any change in referred tables in primary database (CS542.db) and using the logging just created to update old version of database file (CS542A.db).
  - We assume that we have sufficient information about architectures of two tables and are fully aware of what we are changing.
  - We assume population is increased by 2%. We change the population for all the cities and countries by using auto_increase() function, which will be introduced in Design decision part.
  - To establish a redo/undo logging system, we record both old values and new values of every revised records.
- Design decision:
  - Relation()
    - open()
      - While trying to open the database file, we try to open logging file (also in database file form) mean time.
      - Rule for both open process is 'if not exists, create an empty one'.
    - auto_increase()
      - This function is actually integrated with logging process.
    - recover()
      - Recover function uses information in the head of every logging records to locate tuple and recover them.
    - recover_show()
      - This function is to test the ability of our recovery system.
      - The function randomly selects in CS542A.db file 5 cities and 5 countries to show the updated value and meanwhile, read old value from log file to verify the recover() function worked.
    - We also implement other helper functions like back_up() to keep main.py file uncluttered.
  - Logging()
    - We implement record() function to record every changes in the given database file and its mirror function to recover the given database file according to it.
  - Close()
    - Close two FakeDict instances and leave other variables to python garbage collector to deal with.
- Data Structure

- Data
  - {‘cities’:{country1_code: [[city1_info], [city2_info]], country2_code: [[city1_info], [city2_info]]}, ‘countries’:{country1_code: [[country1_info]], country2_code: [[country2_info]]}}
- Log
  - {‘records’:[[table, key, idx1, idx2, old, new],]}