Undo/Redo Logging

Boya Zhou,Yuting Liang

Introduction

Undo/Redo Logging. Over time, the populations of cities and countries changes. We will change the population of each by 2% to represent the passage of a year. The purpose of this programming assignment is to programmatically make this change in all records and to generate undo/redo logs as we are doing it. Then we will move those logs to another machine that has a copy of the same data and apply the logs and observe that the data has changed.

In the style of project 3, create an operator to update the population values in a table. You may choose to write one operator that applies to both the city and the country tables, or two separate ones. As a side effect, the operator should produce a log file.

To test,
Make a backup copy of cs542.db, say cs542A.db,
Run the update that increases the populations by 2%, <
Apply the log file to cs542A.db.
Run spot-checks to verify that the cs542.db and cs542A.db contain identical data and that both show populations that are 2% higher than the original.

Assumption

1.We give city.db and country.db as origin one and also do update on them. And cityA.db and countryA.db as backup.

2.We consider about block conception in this update db file.(for simplicity, we do not consider concurrency control.)

3.We do not consider about block conception in recovery task

4.We do not consider about checkpoint in this task

5.part of the code describe tuple and database were taken from projection 3, changed from project 1

6.in the process of undo/redo logging, we only give the process of redo(for simplicity, redo can finish the task)

7.in<T1,A,old,new> format, we use the tupleId to represent the variable A.

Design Decision

1.In update() function: as same as project 3, we need open(), getNext() and close() process to update each tuple in db file, and the process of update each tuple are almost as same as what out book give.

2.In undoRedo() function: we write a simple function to use log file to recover backup file. We define logs between START and COMMIT as valid logging.

Unfortunately, we didn't get result before deadline, we will improve that in the future.