Database Homework 4

0316055 許庭嫣

1. How do you design you database in detail?

一開始initialize時把db這個class裡面的變數都先歸零初始化。設定temp資料夾時把該字串存數db的變數之中，同時儲存好import時要儲存的table的檔案的路徑，並確保該檔案是空白的。

import檔案資料時，第一行attribute的名稱略去，其他一行一行做讀取，並將需要的資料(ArrDelay, Origin, Dest)依逗號做分割拿出來（若ArrDelay讀取到的資料為NA就把該筆資料整個丟掉，並繼續進行迴圈），Stage 3和Stage 4分別把Origin和Dest用c-string和整數的方法儲存，把這些資料用binary的方式寫進temp資料夾下的memory.bin檔案中。其中因Origin和Dest皆由三個大寫字母構成，因此可用類似26進位的方式儲存成整數，為了讓之後比對速度可以更快。

做query時，若沒有建index，讀檔後就直接比對每筆資料的Origin和Dest是否和query要求的兩個地點字串相同，把相同資料的ArrDelay加起來算平均值；若有index，就可以從Origin和Dest合成的key找出所需要的資料的位置(offset)，可以直接讀取該位置的ArrDelay並做相加球平均值，不需要跑完整個檔案的record。

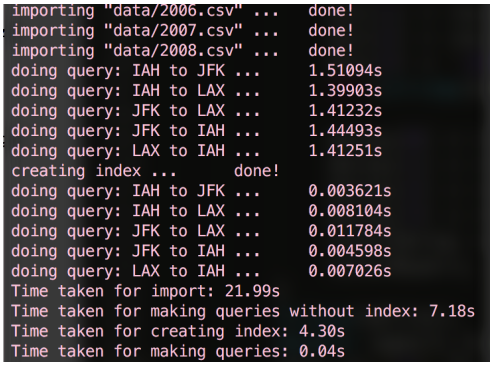
最後clean up時，把剛剛儲存table的檔案開檔不寫入任何東西再關檔，讓原先儲存的資料都消失，之後再將class db的各個變數歸零。

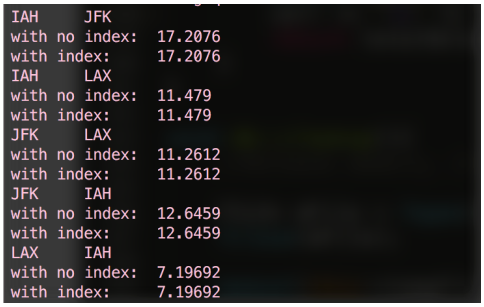
2. How do you create the index in detail?

Stage 3的Origin和Dest是用c-string寫，用map實作。因map的key沒辦法是char[]，必須先用一個struct將index的key包起來，並另外寫比對的function。用vector<int>儲存各資料的offset，求出index的key（將Origin和Dest字串相接）之後，就可以直接跑vector的迴圈知道各個所需的offset。

Stage 4的Origin和Dest是用int儲存，用unordered_map實作。unordered_map的key直接是int，如此也不需要另外寫比對的function。也是用vector<int>儲存各資料的offset，index的key則是將所讀出的Origin乘26的3次方和Dest相加（因兩字串個是三個字母組成）。

3. The screenshot of one of your program's result. The table described below.





Origin	Dest	Average ArrDelay Time (min)	Query time without indexing (sec)	Index time (sec)	Query time with indexing (sec)
IAH	JFK	17.2076	1.51094	4.30	0.003621
IAH	LAX	11.479	1.39903	4.30	0.008104
JFK	LAX	11.2612	1.41232	4.30	0.011784
JFK	IAH	12.6459	1.44493	4.30	0.004598
LAX	IAH	7.19692	1.41251	4.30	0.007026

4. Any other discussion.

這份作業的用意感覺是讓我們能夠更清楚sql和database的實作方式，寫這份code前真的要先懂每個function相對實作時的用意，然而實際上在寫的時候就是應用C/C++的各種STL function等，像是資料結構的應用。若要比速度的話，感覺就會開始鑽牛角尖，甚至有同學連complier或system的指令都下了，感覺到這個地步分數差異太大就會有點超出這門課了。覺得應該測資跑成功，時間不會花太久ranking的分數都可以全給吧...？