* 姓名：劉心慈
* 學號：F74022078
* 系級：資訊系大三
* 作業說明：

環境：Linux Ubuntu (32-bit)

使用語言：C++

系統架構：

1. Hash tables

hash\_table下分為8個子tables，每個子table下面再分為十個bucket，每個bucket下再分為不固定數量的slots

1. Tables  
     
   tables下分為books和sellRecord兩大table，books下分為五個attribute，而sellRecord下分為三個attribute，每個attribute下都記錄了該attribute的各個value
2. table\_row\_num  
     
   table\_row\_num紀錄了books和sellRecord個別的row數(即該table共有幾筆資料)
3. sets

sets下也是分為books和sellRecord兩個subset，各自儲存books和sellRecord有的attributes

使用說明：

SELECT、DISTINCT、FROM、WHERE、AND等keyword要使用大寫，其餘attribute及table name則使用小寫輸入，DISTINCT只可出現在SELECT之後，如SELECT DISTINCT …，單字與單字之間須有空格，若是要將兩個table的attributes作比較，使用語法：attr1 = attr2，如isbn = isbn\_no (等號前後需有空格)，若是要將單一attribute和常數作比較，則常數需用單引號包圍，如author = ‘J. K. Rowling’，每個指令需以分號作結尾，且分號前不可有空格，如SELECT \* FROM books;

輸入0結束程式。

實作方法說明：  
一開始先將books和sellRecord的資料分別讀入並建立hash\_table，hash\_table是用C++的map來實作，以各attribute的名稱來作為index，每個attribute下建立十個bucket(用malloc的方式產生)，將讀入的資料利用hash33 function運算後，得出該筆data應在的bucket位置，用malloc動態產生一個新的slot，放入該bucket中。

hash\_table建立完成後，將各hash table一一輸出成.txt檔，接著進入while loop，也就是不斷讀取SQL指令並執行的地方，一開始會先讀入SQL指令，若SELECT後接有DISTINCT，則distinct flag會被設起，SQL指令中所包含的tables會被存入q\_tables中，所包含的attributes則會被存入q\_attrs中，若指令包含關鍵字WHERE，則分為有join condition和沒有join condition的兩個情況處理，若沒有join condition，則先將要比較的常數丟入hash33中，算出bucket位置，接著只在該bucket中尋找符合條件的資料；若是有join condition的情況，與常數比較的方法同沒有join condition的處理方式，若是兩個table的attribute相比，則會用for loop將其中一個table全部跑過一次，但每筆資料都會丟進hash33中算出對應的bucket，再用算出來的bucket位置去另一個attribute的hash table中比對，只有屬於該hash table中且符合條件的資料會留下，最後將結果一一印出，清空q\_tables和q\_attrs，while loop繼續直到輸入0結束。