

# Homework D – Database Query Processor

TA: T.H. Liu(Qtherliu@gmail.com)

Deadline: Jan. 7, 11:59pm

## 1 About this homework

Recent years, databases are wildly used to store data. We can easily query the data we want by using the query interface database provided. In this homework, you are going to implement a database query processor. A CSV file of contacts is provided as a database table. First, you need to provide a user an interface to query data from your database by inputting the query command. Second, you need to read the raw data from a database (in this homework, read from a file) and to process the query command, and in the end, output them to user. The detail SPECs are listed below.

### 1.1 Database file format (CSV file)

Each line in the file is considered as a raw data. The columns of each raw data are defined as below:

Column Name	DataType
Id	int(11)
FirstName	varchar(10)
LastName	varchar(10)
Gender	varchar(1)
Age	int(3)
PhoneNum	varchar(15)

Notice that the column names are case-sensitive.

### 1.2 Query command format

A user should follow the format when query data. Otherwise, your program should print out the error message. Query format are defined as below:

```
SELECT columns
FROM [FileName].csv
ORDER BY col1 [ASC/DESC] -sort1 [, col2 [ASC/DESC] -sort2]
```

1. 'SELECT', 'FROM', 'ORDER BY' are case-insensitive, and others are case-sensitive.
2. All the query commands are inputted as a single line.
3. In the query command, SELECT and FROM clauses are required. In case of any incomplete commands, print out an error message.

## 2 Implement the query processor

A test file (Contacts.csv) is provided.

### 2.1 User Interface

Your program needs to allow the user to query as many times as they want until they input 'quit'.

```
lthsu104m@csie0[7:35pm]~/DS-TA/HWD>./a.out
this is a test
You have an error in your SQL syntax.
quit
lthsu104m@csie0[7:35pm]~/DS-TA/HWD>
```

### 2.2 SELECT Columns

SELECT columns clause indicates what columns need to be shown. User can either input the specific column names, or simply input \* to show all the columns.

Examples:

```
SELECT FirstName, LastName FROM Contacts.csv
```

```
SELECT Age, PhoneNum, Id FROM Contacts.csv
```

```
SELECT * FROM Contacts.csv
```

### 2.3 FROM FileName.csv

FROM FileName.csv indicates the data source. You should read the raw data from the FileName.csv.

```
SELECT * FROM Contacts.csv
id      FirstName  LastName  Gender  Age  PhoneNum
1       Yuling    Hsueh    F       18   0979461352
2       T.H.      Liu      F       25   0945368720
3       Wen       Song     M       25   0942452888
4       Kurt      Chen     M       23   0988252522
5       Walter    Lin      M       26   0986432587
6       HoChien   Chen     M       23   0912345876
7       Wayne     Lin      M       22   0987654248
8       Kevin     Lin      M       23   0975648213
9       Mike      Lin      M       22   0913548762
10      Jim       Lin      M       24   0912457762
```

## 2.4 ORDER BY

ORDER BY clause requires three parameters: the column, sorting type, and sorting algorithm that you need to process the data before printing them. In this homework, **you need to implement two sorting algorithms**. You can choose any two algorithms that are listed in the lecture slides.

### 2.4.1 Implement two algorithms

The ORDER BY command format consists of the following three parameters in order:

1. **Sorted column**: the specific column the user wants to sort
2. **Sorting Type**: ASC means an ascending order, while DESC means a descending order. Notice that sorting type can be ignored, and if it does, the default value is ASC.
3. **Sorting algorithm**: you need to implement two algorithms which can be chosen by the user.

The command format is given here:

```
SELECT * FROM Contacts.csv ORDER BY columns [ASC/DESC] [-1/-2]
```

Examples:

```
SELECT * FROM Contacts.csv ORDER BY LastName -1
```

id	FirstName	LastName	Gender	Age	PhoneNum
4	Kurt	Chen	M	23	0988252522
6	HoChien	Chen	M	23	0912345876
1	Yuling	Hsueh	F	18	0979461352
5	Walter	Lin	M	26	0986432587
7	Wayne	Lin	M	22	0987654248
8	Kevin	Lin	M	23	0975648213
9	Mike	Lin	M	22	0913548762
10	Jim	Lin	M	24	0912457762
2	T.H.	Liu	F	25	0945368720
3	Wen	Song	M	25	0942452888

### 2.4.2 Double sort

The second sorting command is optional for the user. When the first sort is finished, the second sort looks up the columns based upon the first sort. If there existing multiple data records have the same value based upon the first sort, the second sort sorts those records by the second specified column.

```
SELECT FirstName, LastName, Age FROM Contacts.csv ORDER BY LastName -1, Age DESC -2
```

FirstName	LastName	Age
HoChien	Chen	23
Kurt	Chen	23
Yuling	Hsueh	18
Walter	Lin	26
Jim	Lin	24
Kevin	Lin	23
Wayne	Lin	22
Mike	Lin	22
T.H.	Liu	25
Wen	Song	25

## 2.5 Mistake-proofing

You should do some basic mistake-proofing.

1. Typo of 'SELECT', 'FROM', 'ORDER BY'
2. A valid Query command requires both 'SELECT' and 'FROM' clauses to work.
3. Check whether the input file exists.

```
select * from
You have an error in your SQL syntax.
select id
You have an error in your SQL syntax.
select id from haha
This file does not exist!
select id from Contacts.csv order id
You have an error in your SQL syntax.
```

## 2.6 Print out format

### 1. Data record:

You should print the header for the first line, and then print out the data record each for a line. You should print '\t' between columns to make it look better.

### 2. Error message:

There are some messages for the mistake-proofing function. When there's a mistake in user's input, print out "You have an error in your SQL syntax". On the other hand, if the data source file does not exist, you should print out "This file does not exist!".

## 2.7 Readme, comments and style

An indicator for good source code is readability. To keep source code maintainable and readable, you should add comments to your source code where reasonable. A consistent coding style also helps a lot in reading source code. For this assignment, please also compose a small readme in \*.txt format and name it "README.TXT". This should contain a brief explanation of **how to compile your program** and **what sorting algorithms do you implement**. Please remember to have your source code comments and readme **in English**.

### 3 Grading

The TA(s) will mark and give points according to the following grading policies:

**90% Run your program with 5 testing data on PC^2.**

- 15% D1- User interface, Mistake proofing
- 30% D2- 'SELECT', 'FROM' clauses
- 15% D3- Single sort
- 15% D4- Single sort - Advanced
- 15% D5- Double sort

**10% Readme, comments and style.**

Provide a README.TXT that contains information about the program and what sorting algorithms do you implement all in English. Source code is readable and has comments where reasonable.

### 4 How to submit

The files you MUST submit:

1. Source Code
2. README.TXT

To submit your file electronically, enter the following command in csie workstation:  
**turnin ds.HWD [your\_files...]**

To check the files you had turnin, enter the following command in csie workstation:  
**turnin -ls ds.HWD**

You can see other description about turnin from following link:

<https://www.cs.ccu.edu.tw/lab401/doku.php?id=turninhowto>