Hacettepe University
Department Of Computer Engineering
BBM204 Programming Laboratory
Experiment 2

Subject : Binary searching and prefix searching over the data table
Submission Date : 3.3.2016
Due Date : 17.3.2016
Programming Environment: Java 7 SDK, Eclipse
Advisors : Asist Prof. Erkut ERDEM, Asist Prof. Adnan ÖZSOY,
Asist Prof. Gönenç ERCAN
R.A. Ahmet Selman BOZKIR

INTRODUCTION

Binary search, as fundamental algorithm in searching, is employed to rapidly find a value in a sorted sequence. Binary search actually works on a diminishing subsequence of the starting sequence where the target value is searched which is called the search space. The whole sequence is considered the search space at the initial stage. At each stage, the median value in the search space is compared to target value and half of the search space is abolished. As a result, the algorithm leads us to have a search space consisting of a single element, the target value.

As binary searching expedite searching process, it can be used in database related searching events. Here in this homework, you will find out how sorting and binary searching can be efficiently employed for seeking specific records located in a database.

On the other hand, SQL (Structured Query Language) as a well known data manipulation language is employed to communicate with a database. For relational databases such as Oracle, SQL Server and MySQL, it has been assumed as the standard language. SQL statements are used to perform tasks such as insertion or retrieving data from a database. The standard SQL commands such as "Select", "Insert", "Update", "Delete" constitute the CRUD (create, update, delete) operations.

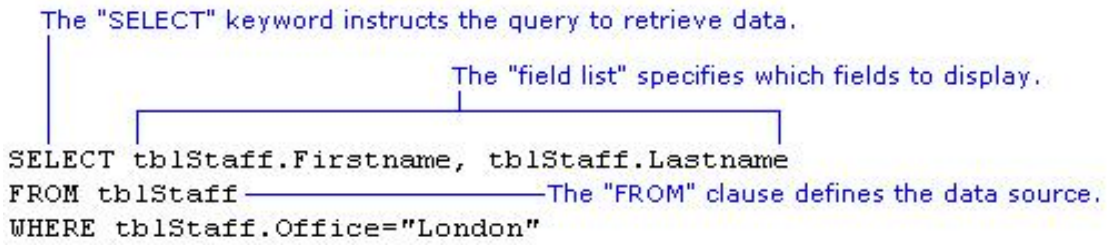


Fig.1 Syntactic form of SQL select statement.

Fundamental SQL syntax has been depicted in Fig. 1 with an example. Every SELECT statement starts with the “SELECT” keyword and takes variable number of column names in order to show. For selecting records according to specific criteria, the “WHERE” clause is employed However in our experiment we will use a reduced and modified version of it. In this experiment integer comparison and string prefix searching will be enabled. The other types of selections were

discarded. As a consequence, generic form of our SQL like select statement is given below:

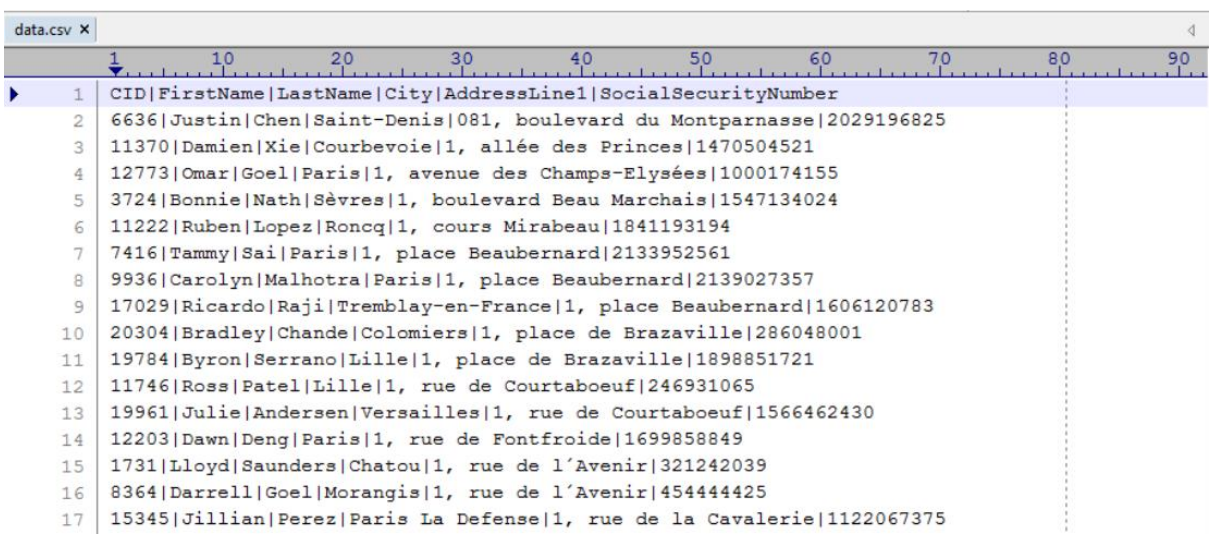
```
SELECT columnname1,columnnname2,… WHERE columnnameX[<,>,=]value AND
columnnameY~xyz
```

The rules related to our SQL like command were given below:

- For integer comparisons, smaller '<', larger '>' and equals to '=' operators have to be implemented. (e.g. AGE<18)
- For string prefix searching the form of "columnname~prefix" (NAME~ca). In this case, we are looking for the records where the NAME column starts with "ca" prefix. (e.g. NAME~ca may return the records such as "Carol" and "Catherina" but not "Colin"
- Maximum number of AND operators is 4. However some statements may or may not contain any AND operator
- SELECT statement takes variable number of column names. Note that, each column name can be used only once.

AIM & DESIGN

In this experiment you are supposed to develop a console based hypothetic database query application. In order to achieve this, you will be supplied with a data CSV (comma separated value) file that is consists of 18750 customer records. Although the name of your data file is "data.csv" your application must be implemented as an argument based start-up application. So, your program will take 2 file name as argument. While the first file (e.g. "commands.txt") will tell you what to do, the second parameter will supply the name of the data file. Please be ensure that extension of the data file will be ".csv".



```
CID|FirstName|LastName|City|AddressLine1|SocialSecurityNumber
6636|Justin|Chen|Saint-Denis|081, boulevard du Montparnasse|2029196825
11370|Damien|Xie|Courbevoie|1, allée des Princes|1470504521
12773|Omar|Goel|Paris|1, avenue des Champs-Elysées|1000174155
3724|Bonnie|Nath|Sèvres|1, boulevard Beau Marchais|1547134024
11222|Ruben|Lopez|Roncq|1, cours Mirabeau|1841193194
7416|Tammy|Sai|Paris|1, place Beaubernard|2133952561
9936|Carolyn|Malhotra|Paris|1, place Beaubernard|2139027357
17029|Ricardo|Raji|Tremblay-en-France|1, place Beaubernard|1606120783
20304|Bradley|Chande|Colomiers|1, place de Brazaville|286048001
19784|Byron|Serrano|Lille|1, place de Brazaville|1898851721
11746|Ross|Patel|Lille|1, rue de Courtaboeuf|246931065
19961|Julie|Andersen|Versailles|1, rue de Courtaboeuf|1566462430
12203|Dawn|Deng|Paris|1, rue de Fontfroide|1699858849
1731|Lloyd|Saunders|Chatou|1, rue de l´Avenir|321242039
8364|Darrell|Goel|Morangis|1, rue de l´Avenir|454444425
15345|Jillian|Perez|Paris La Defense|1, rue de la Cavalerie|1122067375
```

Fig.2 Data file content

As can be seen in Fig. 2, the column names of your data file listed as {CID, FirstName, LastName, City, AddressLine1,SocialSecurityNumber}. CID, here, corresponds to unique id of the respective row. In other words, CID column constitutes the identity information of whole row. On the other

hand, again, it can be seen that '|' character is used as the delimiter character between the values of each column.

The objectives and requirements of your task are listed below:

- Your application should apply binary searching over long integers and strings. Therefore, you must first sort them upon first execution.
- Variable number of column names in SELECT clause must be handled by considering the delimiter character of ','
- As it was stated before, 'AND' operator actually intersects the retrieved rows. So, your program must capable of handling at most 4 'AND' operators. As it can be deduced, you must implement a logic which does binary searching (and sequential searching if necessary) and intersects the retrieved rows by considering their CID values.
- Each processed command and its output as well as its total process runtime (in milliseconds) must be written in a file ("output.txt"). (You can append each command's result to the end of output.txt)
- For sorting, use quick sort method.
- You can use core Java library but it is strictly prohibited to use predefined/preimplemented Java binary search routines.
- Using Java Comparator interfaces is mandatory.
- Pay attention at output formatting (spaces between columns)
- Use of Hashtables or Hashmaps are prohibited.
- Design your classes according to object oriented programming paradigm.

The following lists show two commands and their respective outputs written in output.txt

SAMPLE I/O

Input file

```
SELECT FirstName,LastName,City,AddressLine1 WHERE SocialSecurityNumber<2193000
SELECT FirstName,LastName,City,SocialSecurityNumber WHERE
SocialSecurityNumber>2144193194 AND LastName~Ba
```

Output file

```
CommandText: "SELECT FirstName,LastName,City,AddressLine1 WHERE
SocialSecurityNumber<2193000"
Result:

FirstName   LastName     City                      AddressLine1
Lucas Thomas Imperial    Beach                     791 Monte Cresta
Arthur      Ruiz         Paris                     22, rue des Rosiers
Roy         Ramos        Melton                    1868 Alexander Pl
Christian   Simmons      Olympia                   1930 Many Lane
Chad        Shan         Cliffside                 6643 Mt. Whitney
Thomas      Simmons      Imperial Beach            1207 Erie Dr
Jeremy      Peterson     San Diego                 1035 Arguello Blvd.
Denise      Madan        South Melbourne           9697 Mcelroy Court
Carlos      Hill         Langford                  4200 Greenbrook Dr.
Franklin    Yuan         Gold Coast                5691 Coldwater Drive
Paula       Romero       Perth                     4345 Azoras Circle
Spencer     Hayes        Milwaukie                 8336 Newport Dr.
Alexandra   Rogers       Saint Germain en Laye     10570, rue Lamarck
Seth        Hernandez    Corvallis                 4107 St. Raphael Drive
Carolyn     Suarez St.   Leonards                  6111 Lancaster
```

```
-------------------
ProcessTime: xxx milliseconds

CommandText: "SELECT FirstName,LastName,City,SocialSecurityNumber WHERE
SocialSecurityNumber>2144193194 AND LastName~Ba"
Result:
FirstName   LastName     City                      SocialSecurityNumber
Lucas       Baker        National City             2129073404
Eduardo     Baker        N. Vancouver              2140716512
Marcus      Barnes       Redmond                   2141107626
Patrick     Bailey       Spokane                   2143088506
Sarah       Barnes       Bellingham                2138651868
--------------------
ProcessTime: yyy milliseconds
```

IMPORTANT

- If your application does not read and write in the appropriate folder, then your mark will be degraded by -40 points.
- SAVE all your work until the experiment is graded.
- The assignment **must be original**, INDIVIDUAL work. Downloaded or modified source codes will be considered as cheating. Also the students who share their works will be punished in the same way.
- You can ask your question via course's piazza group.
- Pay attention for the following items while coding: have a short main function, write English comments for your source codes, design your code according to OOP concept.

SUBMISSONS

- The experiment code will be tested in Eclipse development environment.
- Your submission will be in the format below
  <BBM204_1516_2_StudentID>
  |-- source
  |-- All your solution folder
- You have to use "Online Experiment Submission System". **http://submit.cs.hacettepe.edu.tr** Other type of submissions especially by e-mail WILL NOT BE ACCEPTED.
- Submission deadline is 17.3.2016 23.59 pm. **No further extension will be given!!!**