***CHAPTER 1***

***INTRODUCTION***

* **INTRODUCTION TO FILE STRUCTURE**

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimizes the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. File Structure allows applications to read, write and modify data. It also supports finding the data that matches some search criteria or reading through the data in some particular order.

The goal of File Structure is to get the information we need with one access to the disk. If it is not possible, then get the information with as few accesses as possible. Group information so that we are likely to get everything we need with only one trip of the disk. It is relatively easy to come up with File Structure designs that meet the general goals when the files never change. When files grow or shrink when information is added and deleted, it is much more difficult.

Goal of this course is with reference to time and space is to first minimize number of trips to the disk in order to get desired information. Ideally get what we need in one disk access or get it with as few disk accesses as possible. Secondly grouping related information so that we are likely to get everything we need with only one trip to the disk for example name, address, phone number, account balance.

Good File Structure design must have:

* Fast access to great capacity.
* Reduce the number of disk accesses.
* By collecting data into buffers, blocks or buckets.
* Manage growth by splitting these collections.
* **HISTORY**

History of File Structure design:

* In the beginning the file access was sequential, and the cost of access grew in direct proportional to the size of the file. So Indexes were added to files.
* Indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly. Simple indexes become difficult to manage for dynamic files in which the set of keys changes, Hence Tree Structures were introduced.
* Trees grow unevenly as records were added and deleted, resulting in long searches requiring multiple disk accesses to find a record. Hence an elegant, self adjusting binary tree structure called an AVL Tree was developed for data in memory.
* Even with a balanced binary tree, dozens of accesses were required to find a record in moderate sized files.
* A method was needed to keep a tree balanced when each node of the tree was not a single record as in a binary tree, but a file block containing hundreds of records. Hence B-trees were introduced.
* AVL trees grow from top down as records are added, B-trees grow from the bottom up. B-trees provided excellent access performance but, a file could not be accessed sequentially with efficiency.
* The above problem was solved using B+ tree which is a combination of a B-tree and a sequential linked list added at the bottom level of the B-tree.
* To further reduce the number of disk accesses, Hashing was introduced for files that do not change size greatly overtime.
* **ABOUT THE FILE**

A File is an object on a computer that stores data, information, settings, or commands used with a computer program. In a graphical user interface such as Microsoft Windows, files display as icons that relate to the program that opens the file. For example, the picture is an icon associated with adobe acrobat PDF files, if the file was on your computer, double clicking the icon in Windows would open that file in adobe acrobat or the PDF reader installed on the computer.

A File is created using a software program on the computer. For example, to create a text file we use text editor, to create an image file we use an image editor, and to create a document we use a word processor.

Files are not made for just reading the contents, we can also perform some operations on the Files.

* **Read operation**: Meant to read the information which is stored into the files.
* **Write operation**: For inserting some new contents into a file.
* Rename or Change the Name of the file.
* Copy the file from one location to the other.
* Sorting or arrange the contents of the file.
* Move or cut the file from one place to another.
* Delete a file.
* Execute and display the output.

We can also link a file with any other File. These are also called Symbolic Links, in the symbolic links all the files are linked by using some text alias.

* **VARIOUS STORAGE KIND OF FIELDS AND RECORDS**

**Field Structures:**

There are many ways of adding structure to files to maintain the identity of fields. Four most common methods follow:

* **Method 1**: Force the fields into a predictable length. Fix the length of fields. The fields of the file vary in length to make the fields fixed length we have to predict lengths.
* **Method 2**: Begin each field with a length indicator, store the field length just ahead of the field.
* **Method 3**: Place a delimiter at the end of each field to separate it from the next field.
* **Method 4**: Use a “keyword = Value” expression to identify each field and its contents.

**Record Structures:**

A record can be defined as a set of fields that belong together when the file is viewed in terms of a higher level organization. Five most common methods follow:

* **Method 1**: Make the records be a predictable number of bytes in length.
* **Method 2**: Make the record be a predictable number of fields in length.
* **Method 3**: Begin each record with a length indicator consisting of a count of the number of bytes that the record contains.
* **Method 4**: Use a second file to keep track of the beginning byte address for each record.
* **Method 5**: Place a delimiter at the end of each record to separate it from the next record.
* **APPLICATION OF FILE STRUCTURE**

**AMAZE FILE MANAGER:**

Amaze File Manager is a newer app comparatively speaking and it’s a pretty good. It’s open source and focuses on as lighter experience for those who just need to do some light file browsing. It features Material Design, SMB file sharing, a built-in app manager to uninstall apps, root explorer, and more. It manages to include the most important stuff without feeling bloated. It’s free to download and use with optional in-app purchases in case you want to help fund development.

**ASUS FILE MANAGER:**

It’s not every day we see on OEM app make an app list, but File Manager by ASUS is actually really good. It’s compatible with most devices, even non-ASUS ones. You’ll also get clean, simple interface with LAN and SMB support, could storage support, support for various types of files, archiving support, and more. It’s entirely free with no in-app purchases and provides a great experience for a simple file browser. About the only negative part is the lack of root access.

**ES FILE EXPLORER PRO:**

ES File Explorer has been around as long as most Android nerds can remember and comes with pretty much every feature you can ask for in a file browser. A while back, it was purchased by another company. Since then, things haven’t gone well. The free version of the app, while very capable, now has a ton of added bloat ware that not only doesn’t add to the experience, but activity subtracts from it. Thankfully, the pro version of the app doesn’t have these features.

**FILE MANAGER:**

File Manager is blandly named, but it’s actually quite good. It’s a newer file manager app that gives you one of the best sets of features without adding too much bloat. You’ll get basic file management features along with cloud storage features, NAS support, and more. You can even browse your installed apps, music, and video with this and the player isn’t half bad. Perhaps the best part is that the app is free with no in-app purchases and advertising.

**MK EXPLORER:**

MK Explorer is another newer file manager option. It’s a simple option that doesn’t have a whole lot of flair. That is extremely preferable if you really just want something simple. It features a Material Design interface, the basic file management features (copy, paste, delete, SD Card support for Lollipop), and root access. There are also support for 20 languages and it has a built-in text editor, gallery, and music player. It doesn’t have anything like cloud storage or network storage support, but that’s not really what it’s for. It’s a good, cheap option.

**X-PLORE FILE MANAGER:**

X-Plore File Manager is one of the more unique options on the list. It’s a forced dual-pane app which means you’ll be managing two windows at once pretty much all the time. This is kind of cool if you’re copy/pasting between folders or need to move files quickly. It also comes with support for various types of files, cloud storage, network storage, a built-in hex editor, root support, and plenty of other features. You can even view APK files as zips if you’re into that kind of thing.

**1.2. Indexing**

What is an Index?

Index: A structure containing a set of entries, each consisting of a key field and a reference field, which is used to locate records in a data file.

Key Field: The part of an index which contains keys.

Reference Field: The part of an index which contains information to locate records.

* An index imposes order on a file without rearranging the file.

A Simple Index for Entry-Sequenced Files

An index in which the entries are a key ordered linear list.

* Simple indexing can be useful when the entire index can be held in memory.
* Changes (additions and deletions) require both the index and the data file to be changed.
* Updates affect the index if the key field is changed, or if the record is moved.
* An update which moves a record can be handled as a deletion followed by an addition.

A file in which the record order is determined by the order in which they are entered

* The physical order of records in the file may not be the same as order of entry, because of record deletions and space reuse.
* The index should be read into memory when the data file is opened.
* Searching of a simple index on disk takes too much time.
* Maintaining a simple index on disk in sorted order takes too much time.

**TOOLS used:**

JAVA eclipse

Ram 2GB

***CHAPTER 2***

***SYSTEM ANALYSIS***

**2.1 Analysis of Application**

Doctor details has become an important part of the MEDICAL field and in portal development initiatives. Doctor Portal System comprises of a doctor details program which is designed with the main intension of generating an automated system for doctor details. It would help in an easy management of reservation and data of the doctor who are using the portal service.

Doctor Portal System is developed as per seeing the increasing requirement to speed up the work and incorporate a new work culture. Thus a new software has been proposed to reduce manual work, improving data efficiency, saving time and to provide greater availability and user- friendliness.

Doctor Portal System is to minimize the work and at the same time increase the speed of the work done and also the information retrieval will become easy. The purpose of the project is to develop a system which is user friendly, easy to use, maintain and satisfies all the requirements of the user of the specified system.

**2.2 Structure used to store the Fields and Records**

* A field is an item of stored data. A field could be a name, an address, a description etc.
* A record is the collection of fields that relate to a single entity.

For example, we could have a record of a doctor that includes fields for the doctor’s id, name, phone number,etc.

* A file is a collection of related records.
* Files are stored on secondary storage devices such as hard disk, CD-ROMs etc.
* Within a file, all records usually have the same structure. That is, every record in the file contains the same fields. Only the data stored in the fields of different record will be different.

**2.3 Operation performed on a file**

The files are used to store the required information for its later uses. There are many file operations that can be perform by the computer system.

Here is the list of some common file operations:

* File Create operation
* File Delete operation
* File Open operation
* File Close operation
* File Read operation
* File Write operation
* File Seek operation

**File Create operation**

* The file is created with no data.
* The file create operation is the first step of the file.
* Without creating any file, there is no any operation can be performed.

**File Delete operation**

* File must has to be deleted when it is no longer needed just to free up the disk space.
* The file delete operation is the last step of the file.
* After deleting the file, if doesn’t exist.

**File Open operation**

* The process must open the file before using it.

**File Close operation**

* The file must be closed to free up the internal table space, when all the accesses are finished and the attributes and the disk addresses are no longer needed.

**File Read operation**

* The file read operation is performed just to read the data that are stored in the required file.

**File Write operation**

* The file write operation is used to write the data to the file,again,generally at the current position.

**File Seek operation**

* For random access files, a method is needed just to justify to specify from where to take the data. Therefore, the file seek operation performs this task.

**2.4 Indexing Used**

An Index is a tool for finding records in a file. It consists of

* Key field on which the index is searched.
* Reference field that tells where to find the data file record associated with particular key.

Here we have used a simple indexing.

* We chose to organise the file as variable length record with a size field preceding each record. The fields within each record are also of variable – length but are separated by delimiters.
* We form a **Primary key** by the record’s ID number that is bus number for doctor id for booking. This should form a **unique key**

***CHAPTER* 3**

***SYSTEM* *DESIGN***

**3.1 Design of the Fields and records**

* **DOCTOR DETAILS -** Id, Name, Gender, Specialization, NoOfServices
* **EXIT**

**1. DOCTOR DETAILS:** Doctor Details contains elements such as,

**Add**->Add new record - Adds a new doctor record

**Delete**->Delete existing record - Deletes an existing doctor record

**Search**->Search the record - Searches an existing doctor records

**Main menu**->Exit - Exit and comes back to the main menu

Doctor Portal System is a menu-driven application that helps to manage the records of the doctors.

**2. EXIT:** Exits the current screen/application.

**3.2 User Interface**

* The junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program.
* A menu driven interface is one in which you select command choices from various menus displayed on the screen.

**3.2.1 Insertion of a record**

Here in the chosen doctor portal dataset, user can also insert the new doctor details if needed, they may not restrict to few records. By entering the id, name,age,specialization and the number of sevices a record of a doctor can be added. This operation helps when a doctor is newly added.

**3.2.2 Display of a Record**

The record is displayed in such a way, where it has 5 columns like- identifier, name, age, specialization, no of services of a dataset holding 8+data to 1lakh+ data which is of csv format.

**3.2.3 Deletion of a Record**

Here the specified field can be deleted based on the given primary and secondary keys. Here, the primary key id identifier, secondary key is no of services and an other secondary key is specialization. so if any of these key wants to be deleted it can be.

**3.2.4 Search of a Record**

Here in this operation for every case in the main program this operation has to be done. Like if a details of a doctor has to be obtained uding a primary key , then id has to be searched and to make the search more simpler we can find using the secondary keys, where specialization and no of sevices of a doctor has termed as the secondary keys in this doctor portal.

***CHAPTER* 4**

***IMPLEMENTATION***

**About JAVA:**

Java technology is both a programming language and a platform. The Java programming language is a high-level language.

In the Java programming language, all source code is first written in plain text files ending with the .java extension.

Those source files are then compiled into .class files by the javac compiler.

class file does not contain code that is native to your processor; it instead contains bytecodes the machine language of the Java Virtual Machine (Java VM).The java launcher tool then runs your application with an instance of the Java Virtual Machine.

**There are two types of indexes:**

Primary index and secondary index. Primary index is automatically created using the primary keys defined.

Secondary index could be created as per the user requirement. This article discusses about creating a secondary index.

**Steps for Indexing:**

The first process is the Search key that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly.

The second process is the Data Reference or Pointer which contains a set of pointers holding the address of the disk block where that particular key value can be found.

**The indexing has various attributes**:

->Access Types: This refers to the type of access such as value based search, range access, etc.

->Access Time: It refers to the time needed to find particular data element or set of elements.

->Insertion Time: It refers to the time taken to find the appropriate space and insert a new data.

->Deletion Time: Time taken to find an item and delete it as well as update the index structure.

->Space Overhead: It refers to the additional space required by the index.

**Pseudo Code for Indexing:**

Step 1: Start the program.

Step 2: Open the file.

Step 3: While file is not equal to end of file

Get the position from file. Erase the contents from buffer.

Copy the contents from file to buffer. If the contents of buffer is not delete

then, If buffer is empty then **break.**

Extract the doctor details.

Step 4: close the file.

Step5 : Call pi\_index() and si\_index().

Step6 : Erase the contents of buffer.

**IMPLEMENTATION Process**:

So, to begin with i had started with the coding process for primary and secondary indexing. later searched for the dataset which contains 1L data which holds doctor details.

secondary indexing was a bit challenging for me as we had to choose two referencing keys for the process of searching and deleting. Here in my project i have taken Identifier as the primary key, NoOfServices as the first referencing key and specialization as the second referencing key.

coming to the building inexes of primary and secondary i had two text files namely index.txt and secondary.txt to store them. Next coming to the dataset, it has five attributes- id, name, age, specialization, noofservices. For the second secondary key i had implemented the secondary indexing code itself, as it resembles the same process.

My project has the following operations imolemented:

add(), sort(), search() and delete()

where all these operations was for all the keys followed by pri, sec, sec1.

And I also got to know that if in case of a large data, if it holds any blank space, we do endup having the exceptions instead of getting the output.

***CHAPTER 5***

***RESULTS AND SNAPSHOTS***

**5.1**

**TIME COMPLEXITY:**

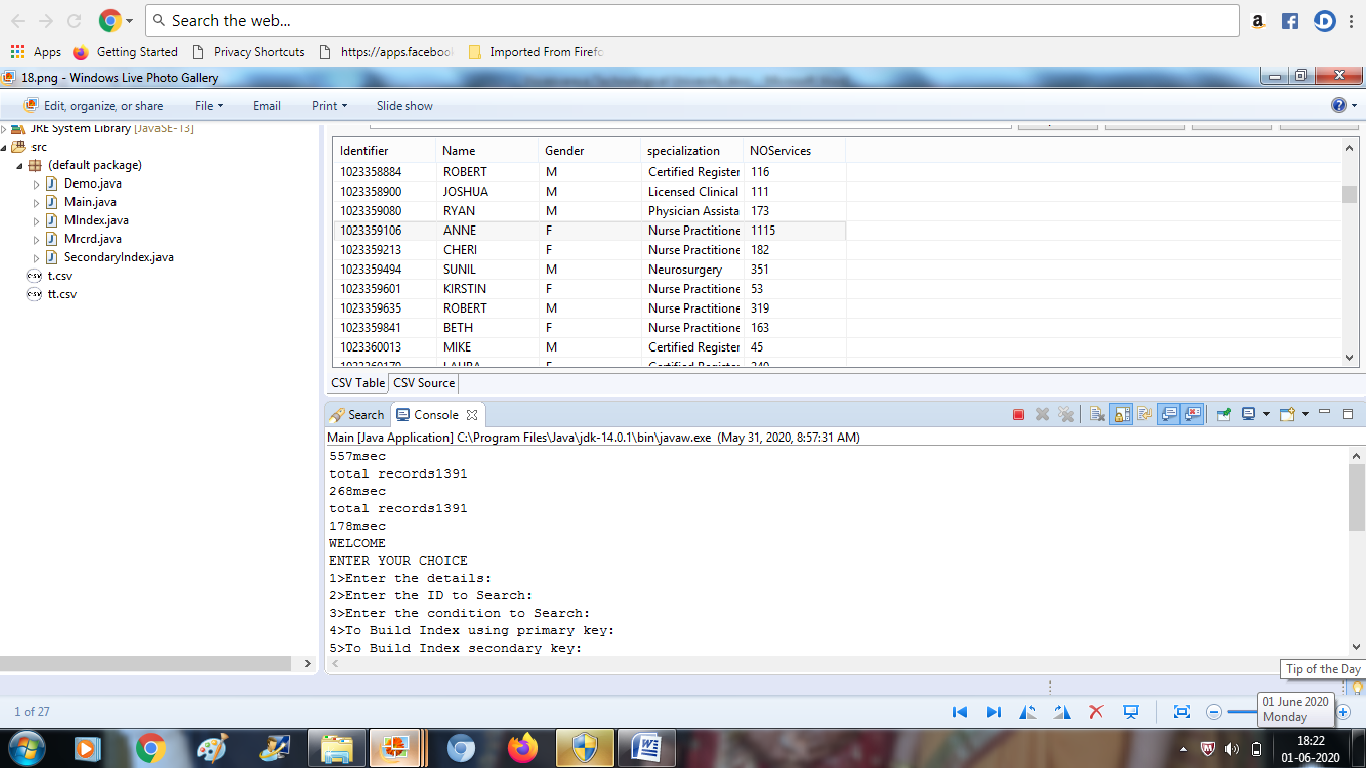
Sometimes, there are more than one way to solve a problem. We need to learn how to compare the performance different algorithms and choose the best one to solve a particular problem. While analyzing an algorithm, we mostly consider time complexity and space complexity. Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input. Similarly, Space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.

Time and space complexity depends on lots of things like hardware, operating system, processors, etc. However, we don't consider any of these factors while analyzing the algorithm. We will only consider the execution time of an algorithm.

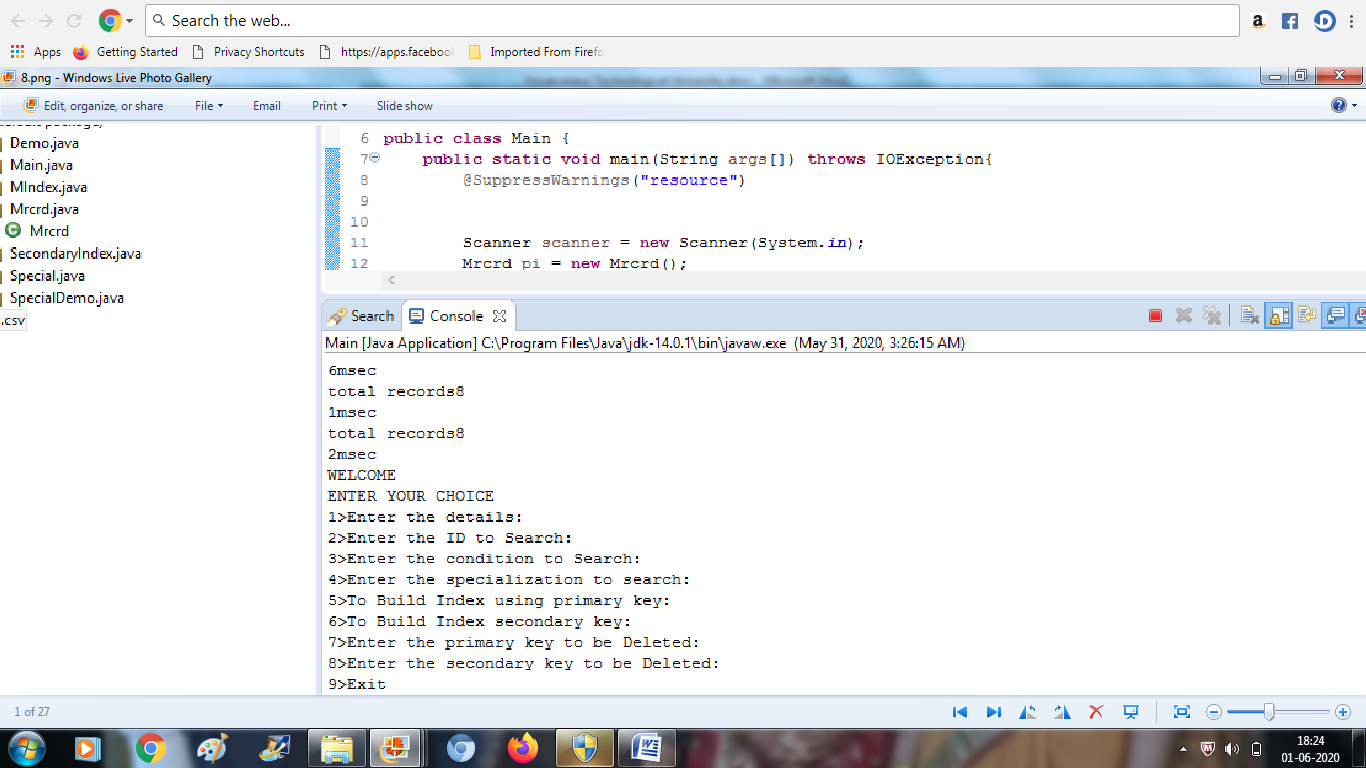
So here in my project if we draw a graph to show the time complexity, then x-axis holds number of records and y-axis holds time in ms.

I have taken some three comparisons to show the time complexity, where the first csv dataset holds 1309 records, second csv dataset holds 8 records and third csv dataset holds 110000 records.

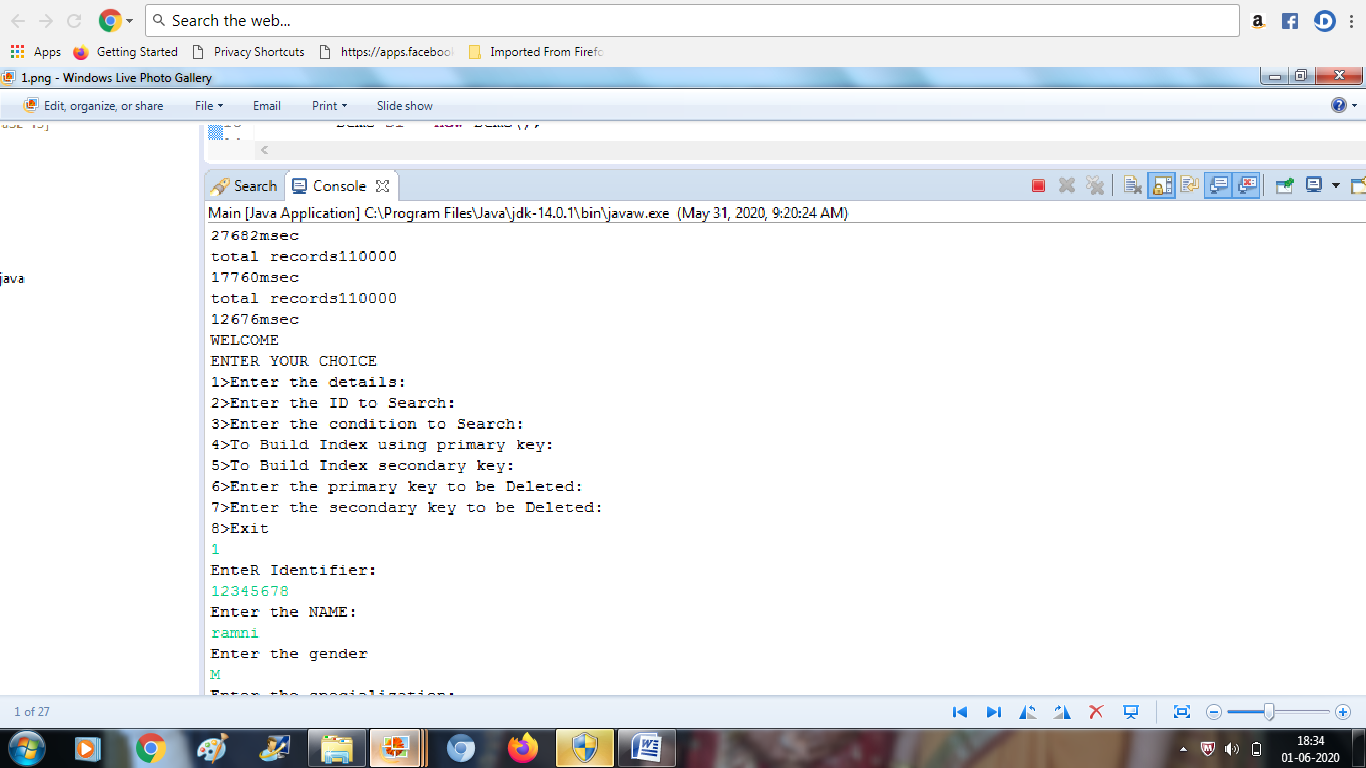
1.



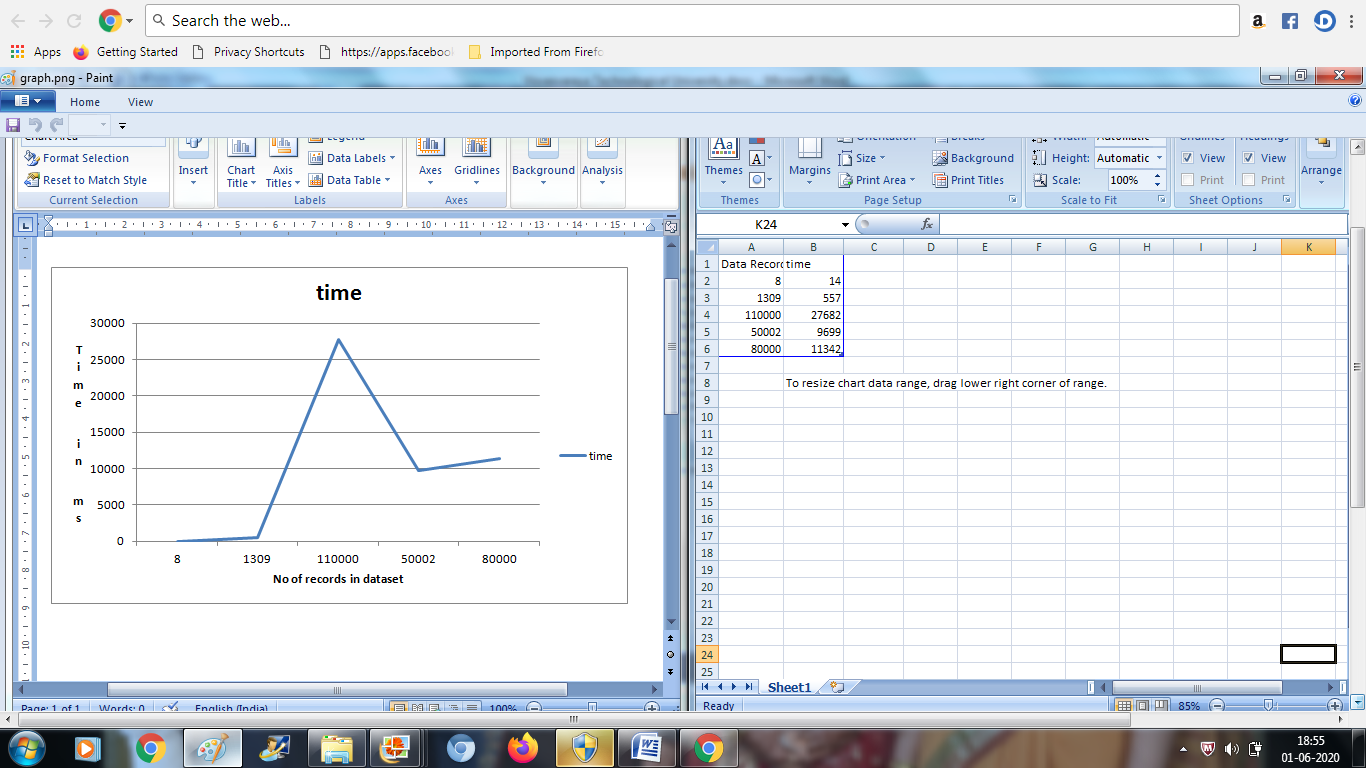
2

******

3

******

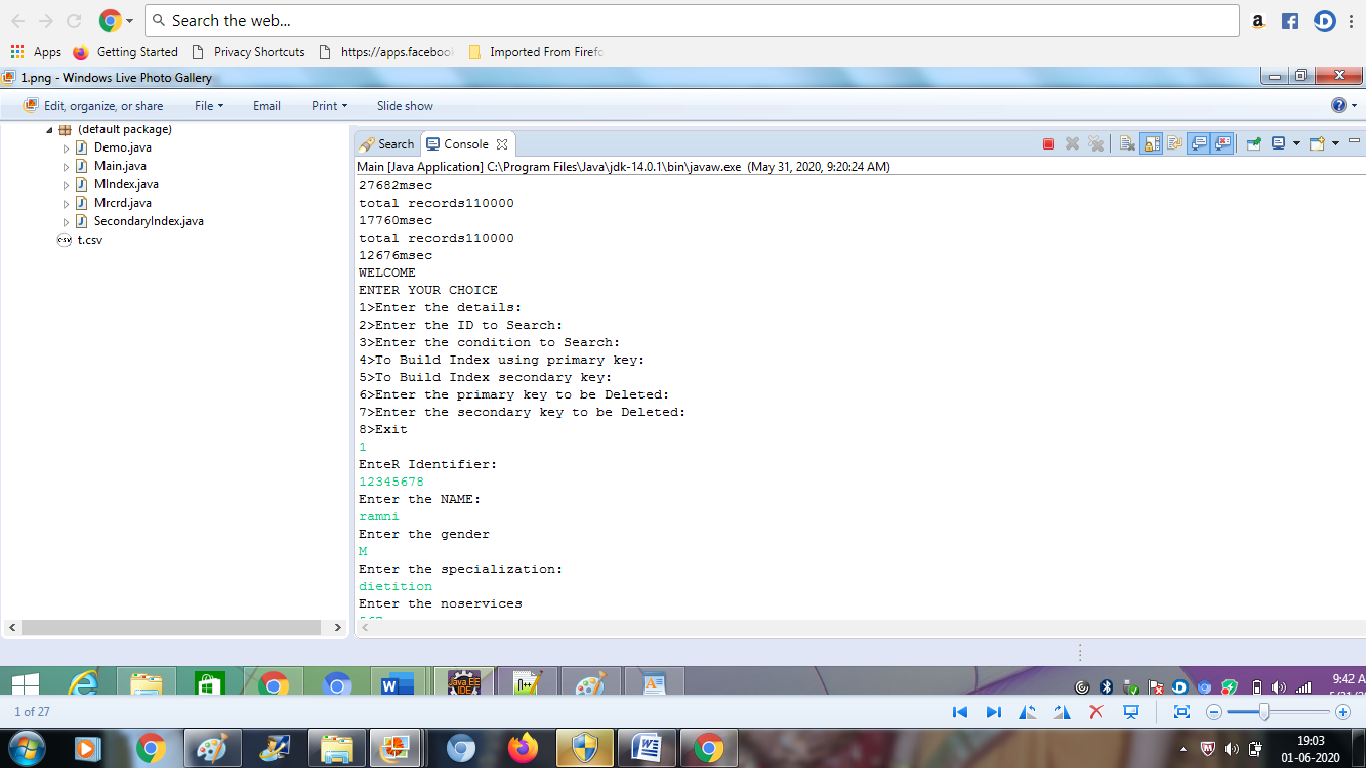
So here’s the graph of the three datasets after comparison:



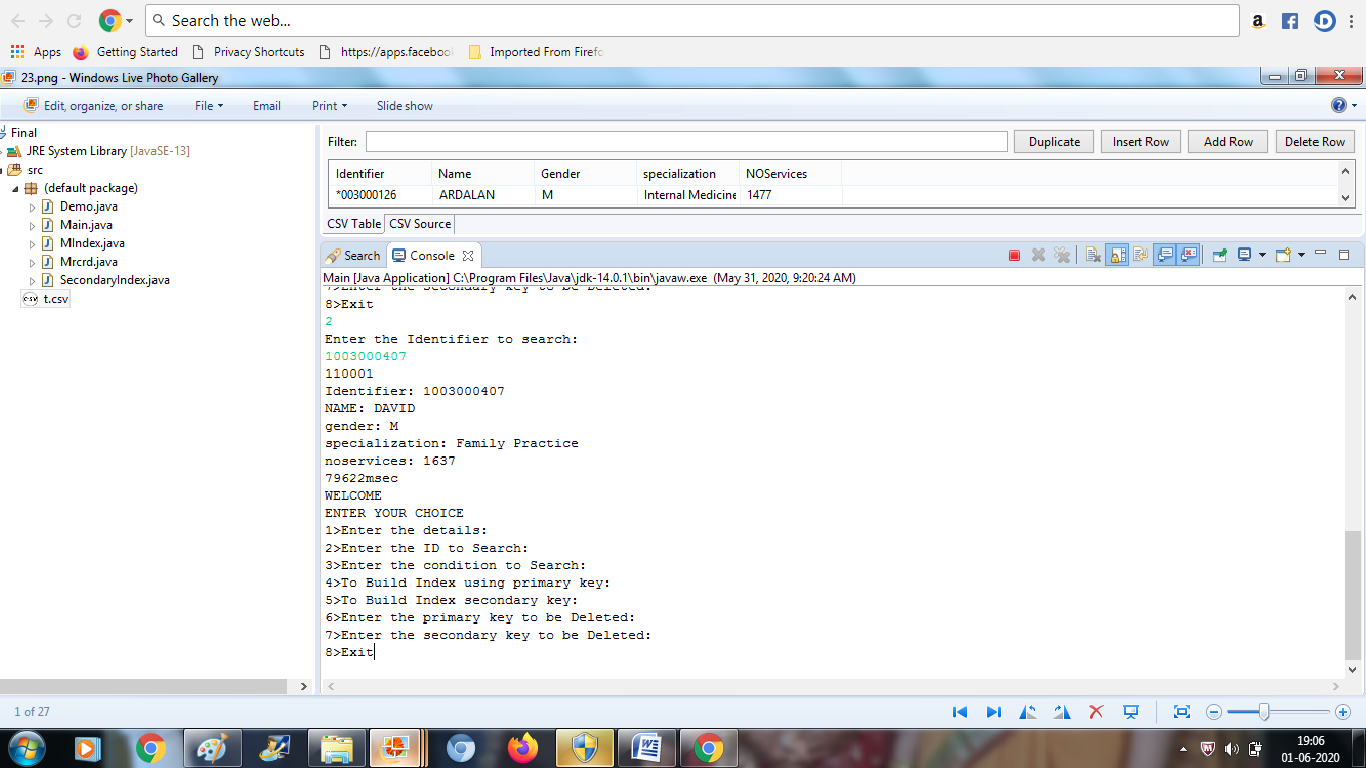
5.2

* Here we have totally 9 cases where it holds the operations such as insert(), search(), delete() and others.

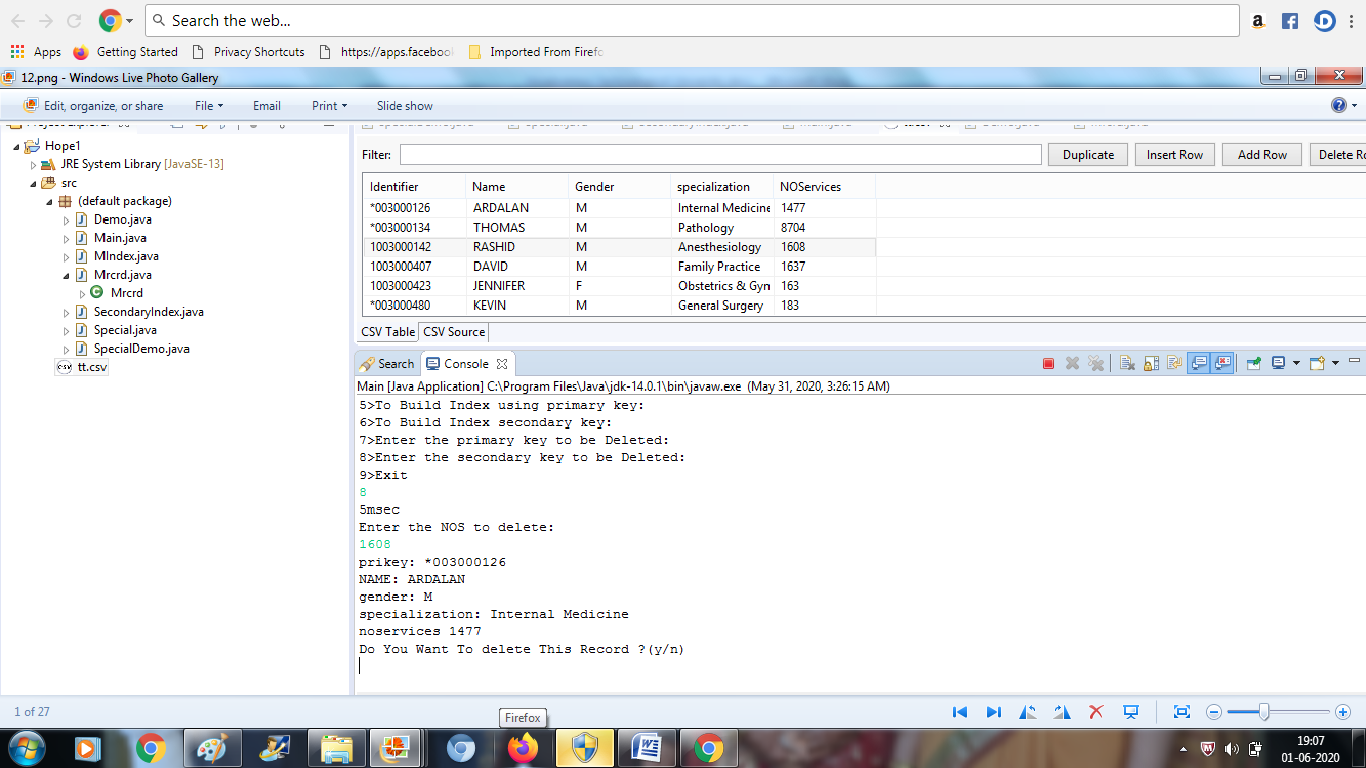
1.Inserting the details of a doctor:



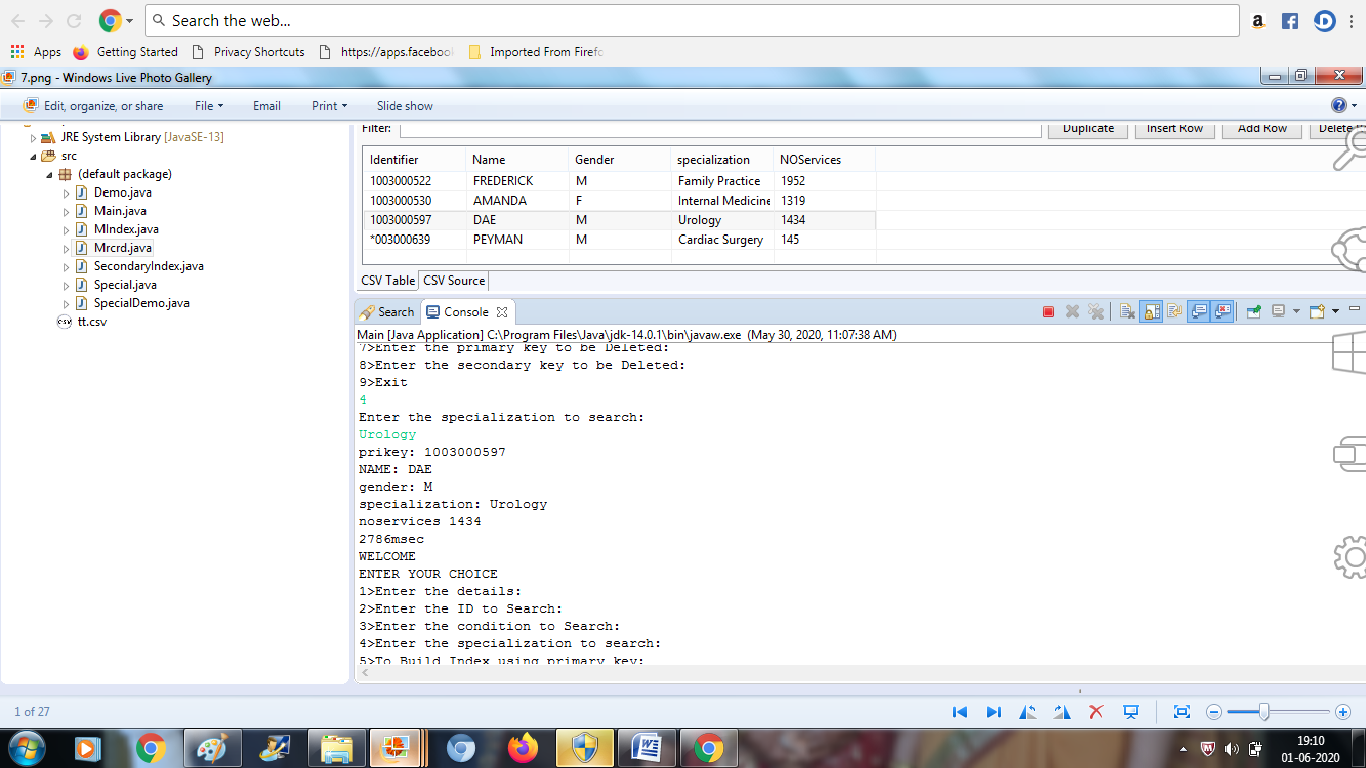
2. Searching :



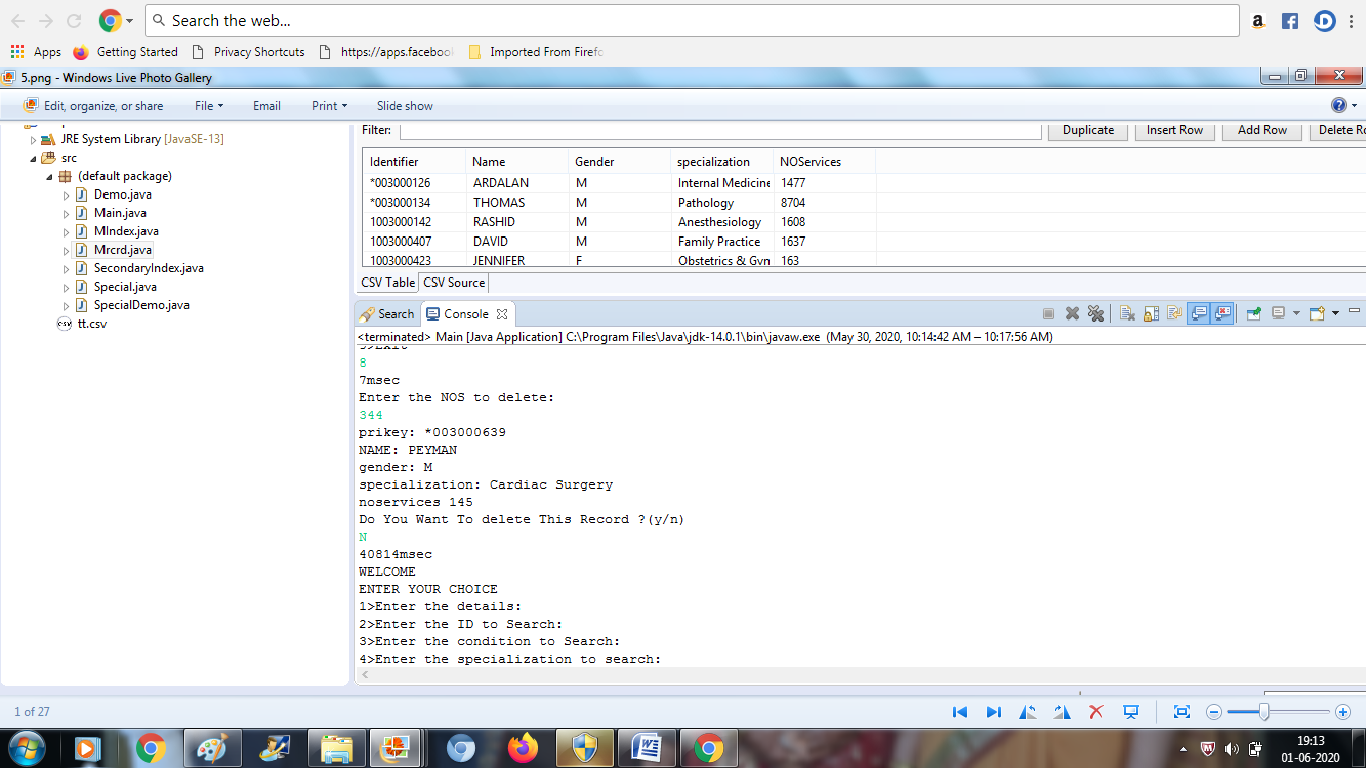
3. Searching using the secondary key:

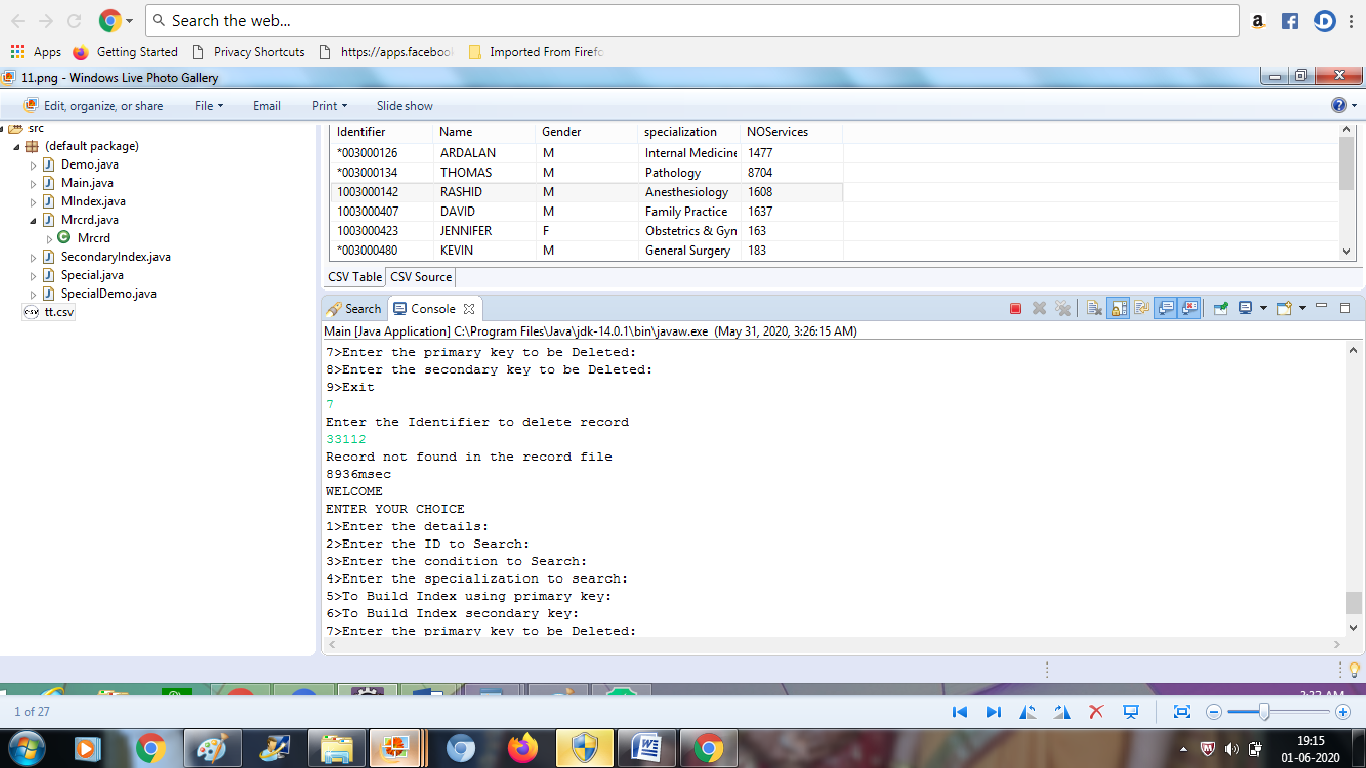


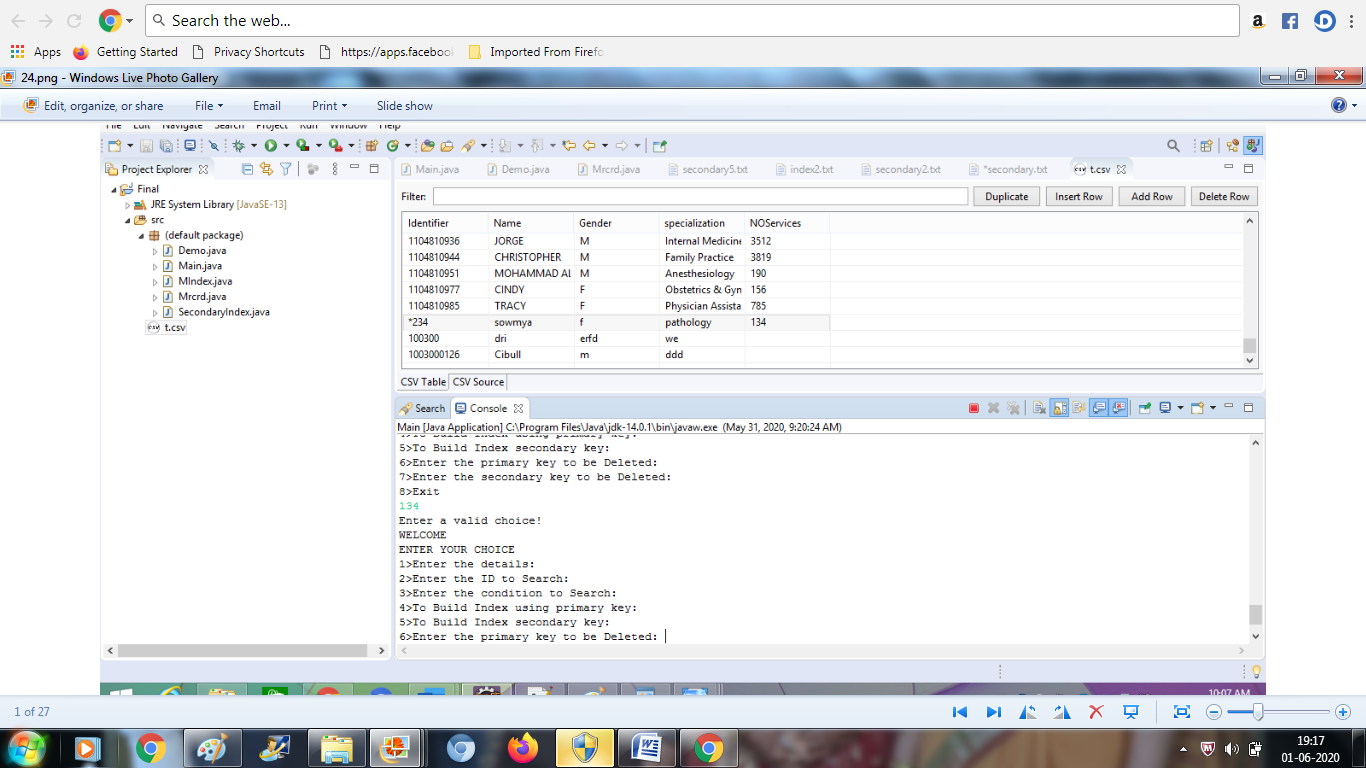
4. Searching using the second secondary key:



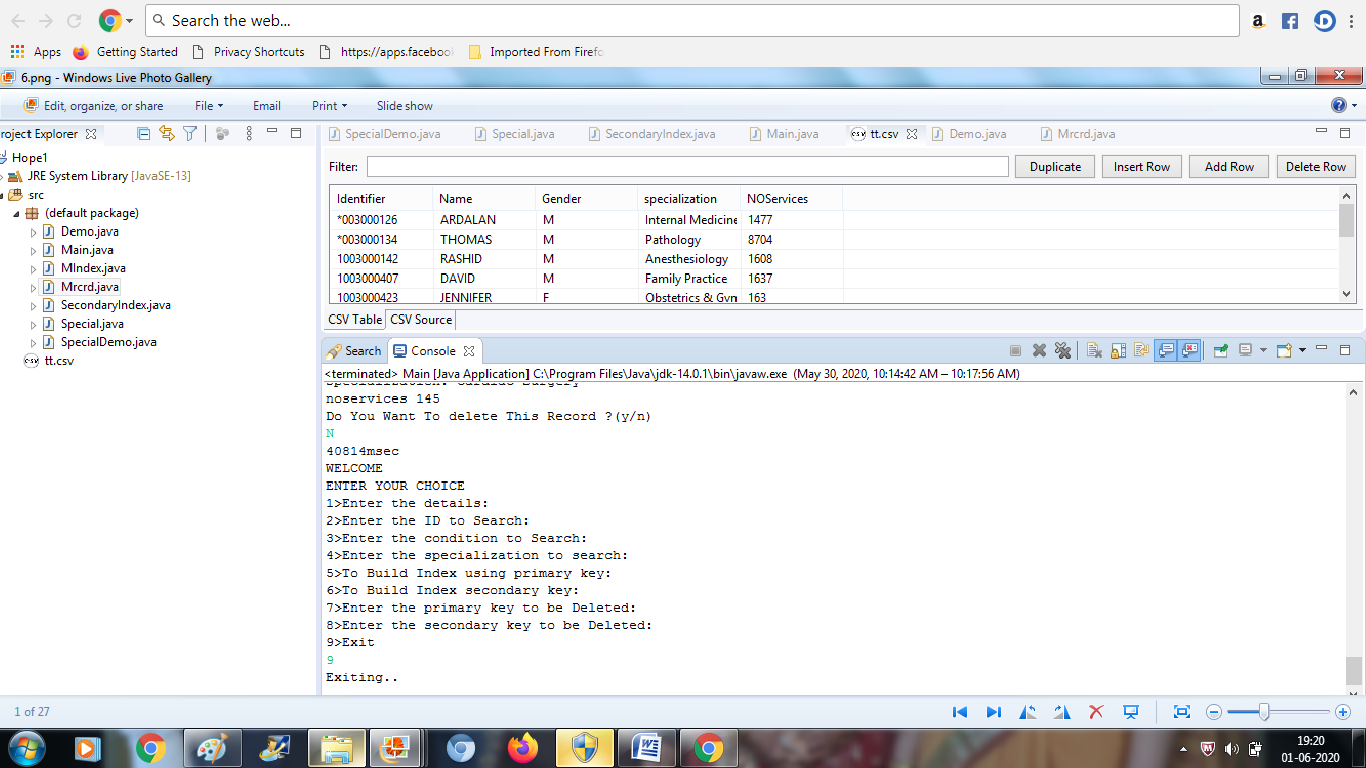
5. Deletion:





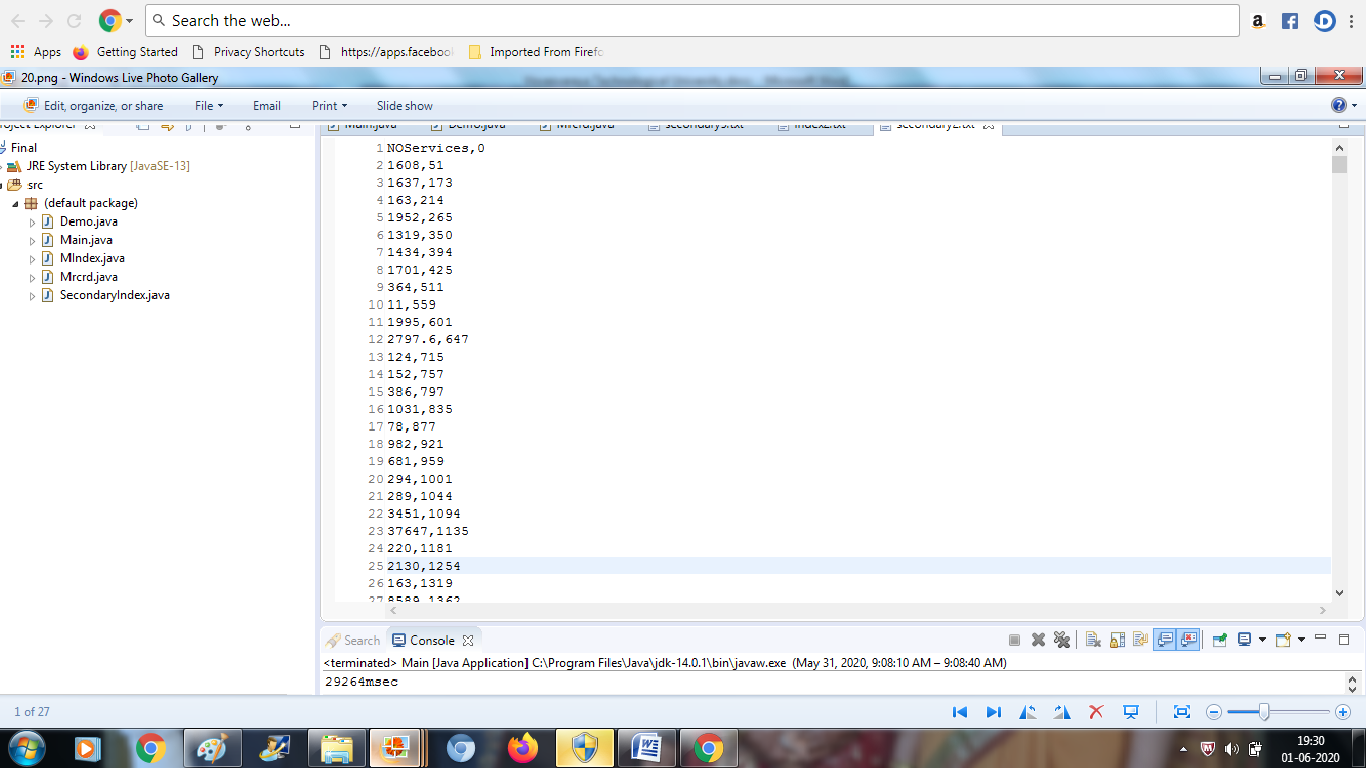


Here we can see that it has a contition to whether to delete or not:

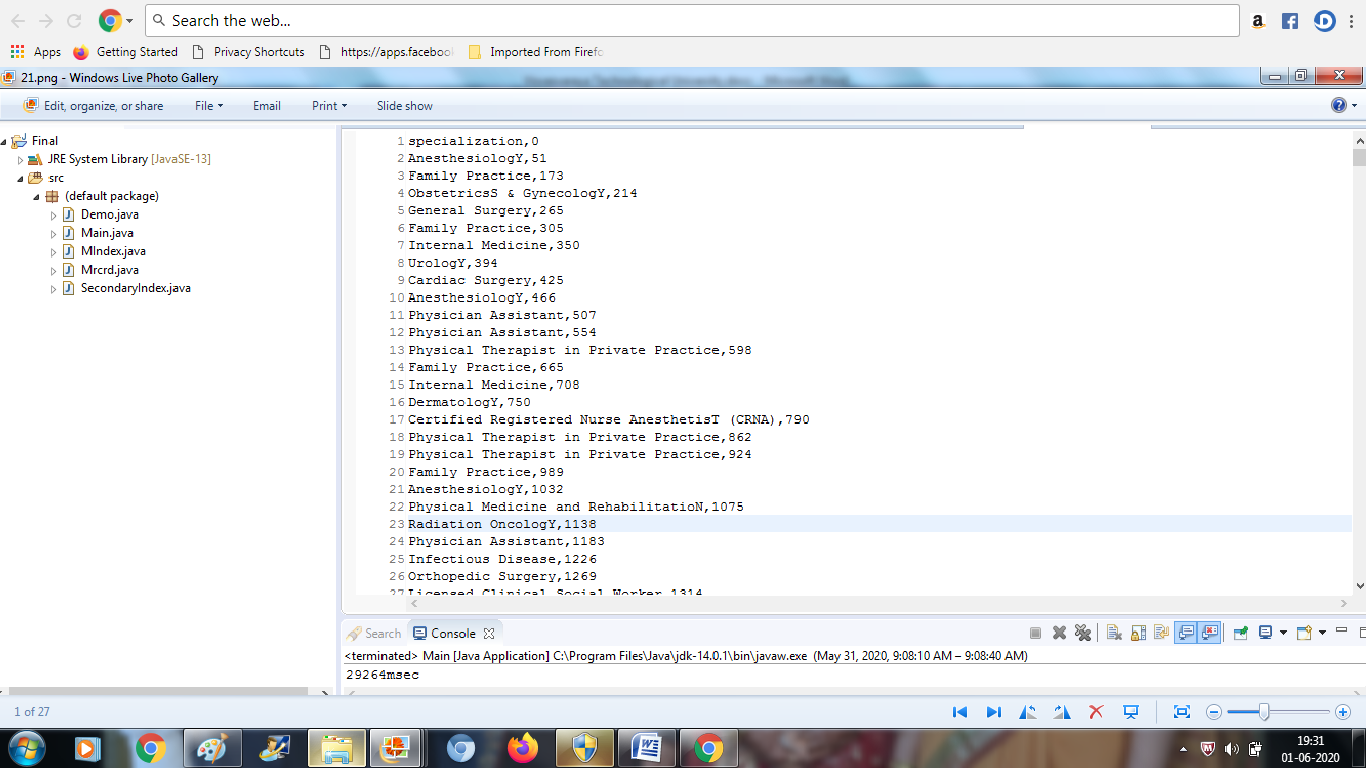


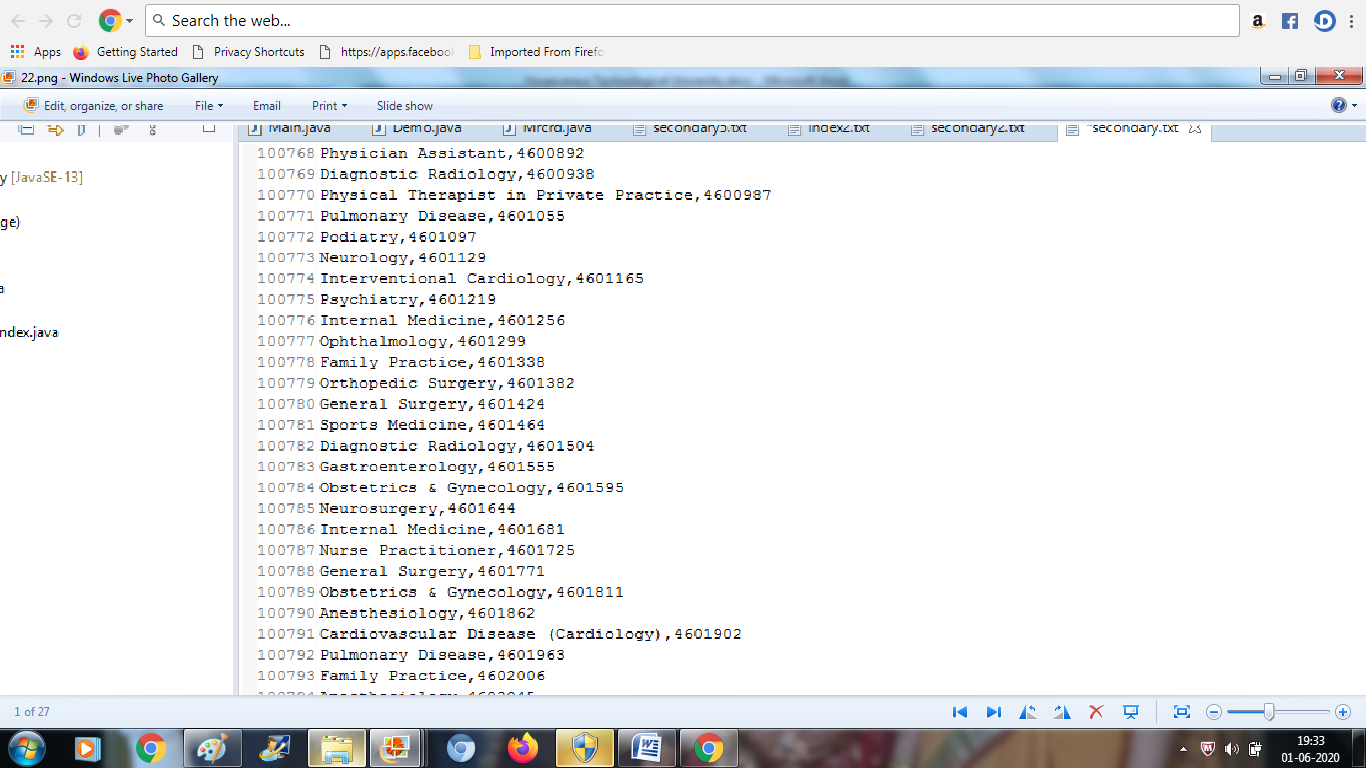
Formation of the PRIMARY and SECONDARY indexes:

Secondary index formation:

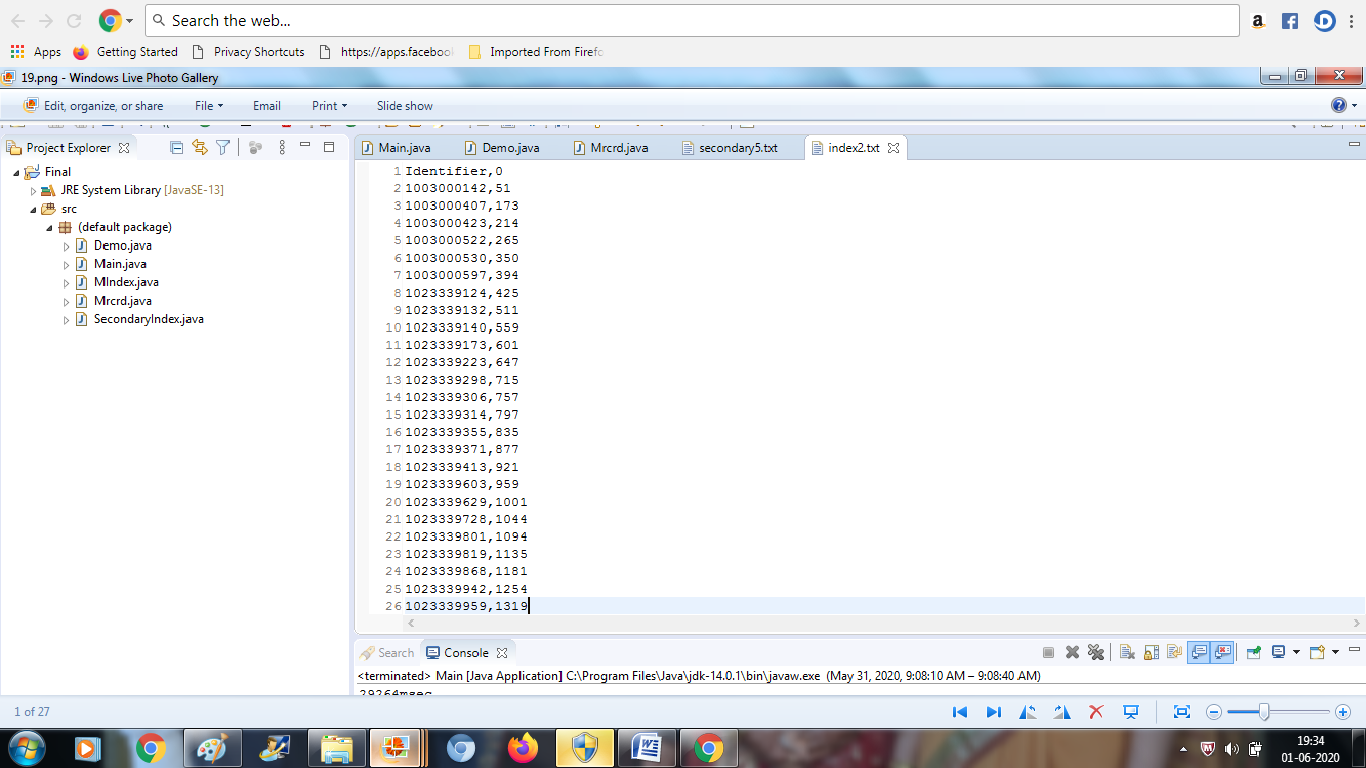


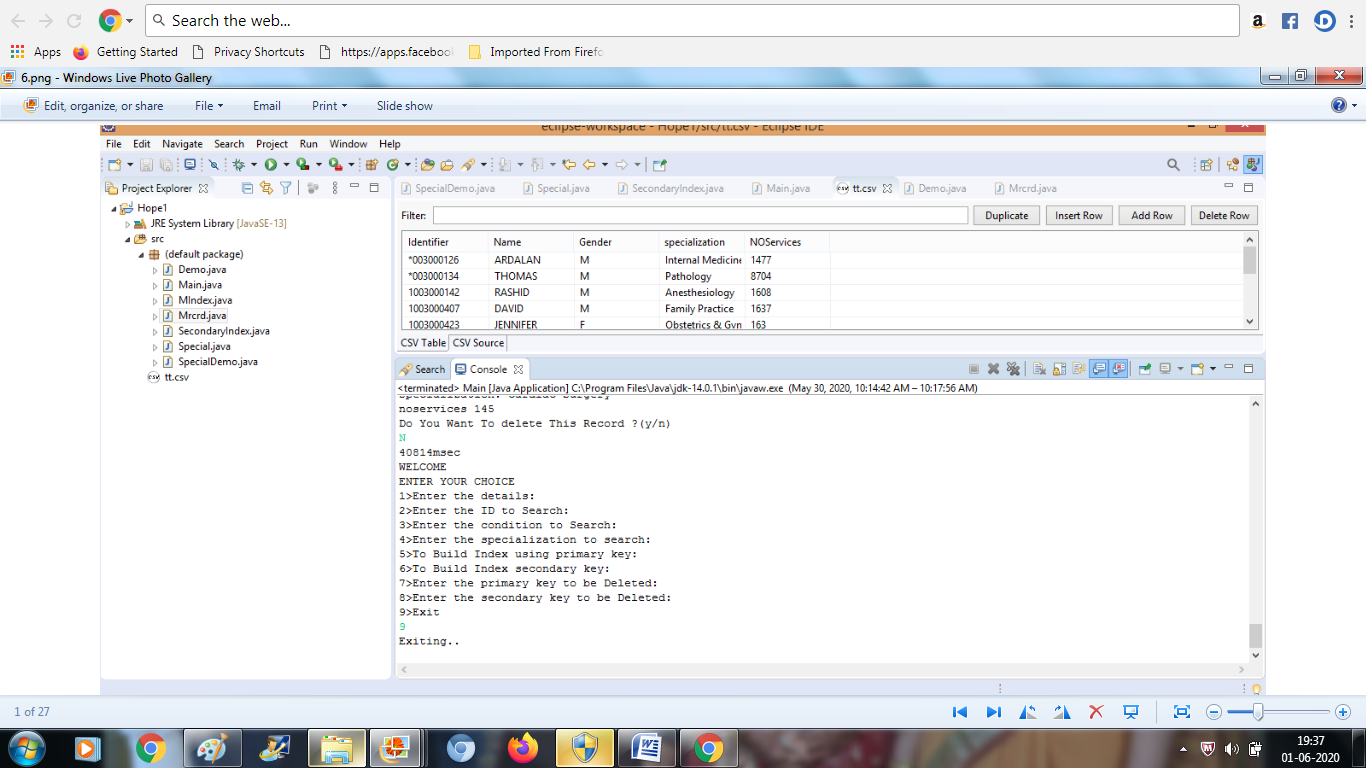
Second secondary index formation:





Primary index formation:





So the above snapshot represents the end of the operations by exiting.

***CONCLUSION:***

We can maintain the records of the doctors with their details. There is a scope for introducing a method to patients to book appointments, which inturn can be developed as Doctor – Patient system.

The purpose of the project is to build an application program to reduce manual work for managing doctor details. This application is just used to store the doctors details where in it holds some operations such as insertion, searching, deletion and so on. This application takes minimum access, any future enhancements can be done.

***REFERENCES:***

1.JAVA TUTORIAL - TUTORIALSPOINT: this site helped me in learning some of the coding techniques in java. I followed some of the basics for the operations to be performed for my applications, such as insertion, deletion and searching.

2.JAVATPOINT- this site was helpful in learning how to import a csv in eclipse.

3.GEEKSFORGEEKS- this site helped me in learning some of the theoretical concepts on indexing and other operations performed.