**Visvesvaraya Technological University**

**Belagavi-590 018, Karnataka**

A Mini Project Report on

**"Implementation of Indexing on Chicago crimes dataset"**
**Mini Project Report submitted in partial fulfilment of the requirement for**

**the File Structures Lab [17ISL68]**

**Bachelor of Engineering**

**In**

**Information Science and Engineering**

**Submitted by**

**Aishwarya S [1JT17IS003]**

**Under the Guidance of**

**Mr.Vadiraja A**

**Asst. Professor**

**Dept. Of ISE**


**Department of Information Science and Engineering**

**Jyothy Institute of Technology**

**Tataguni, Bengaluru-560082**

**2020-21**

# Jyothy Institute of Technology

## Tataguni, Bengaluru-560082

## Department of Information Science and Engineering


## CERTIFICATE

Certified that the mini project work entitled **"Implementation of Indexing on Chicago crimes dataset"** carried out **by Aishwarya S [1JT17IS003]** bona fide student of Jyothy Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering** in **Information Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Mr .VadirajaA
Guide, Asst.Professor
Dept.Of ISE

Dr. HarshvardhanTiwari
Associate. Professor and HOD
Dept.Of ISE

External Viva Examiner

1.

2.

Signature with Date:

# ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution **"Jyothy Institute of Technology**" for providing us an opportunity to complete our project.

We express our sincere thanks to our Principal **Dr. Gopalakrishna K for** providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Harshvardhan Tiwari, Associate Prof. and Head** of Information Science and Engineering Department for providing for his valuable support.

We would like to thank our guides **Mr.Vadiraja A, Asst. Prof.** for their keen interest and guidance in preparing this work.

Finally, we would thank all our friends who have helped us directly or indirectly in this project.

Aishwarya S [1JT17IS003]

# ABSTRACT

Indexing is the process of associating a key with the location of a corresponding data record. An external sort typically uses the concept of a key sort, in which an index file is created whose records consist of key pairs. Here, each key is associated with a pointer to a complete record in the main database file. The index file could be sorted or organised using a tree structure, thereby imposing a logical order on the records without physically rearranging them. Each record of a database normally has a unique identifier, called the primary key. A particular key value might be duplicated in multiple records, is called a secondary key. The secondary key index will associate a secondary key value with the primary key of each record having that secondary key value .The full database might be searched directly for the record with that primary key, or there might be a primary key index that relates each primary key value with a pointer to the actual record on the disk. In this case, the primary index provides the location of the actual record on disk, while the secondary disk indices refer to the primary index. Indexing is an important technique for organising large databases.

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

# 1. INTRODUCTION

## 1.1 Introduction to File Structure

A disk's relatively slow access time and the enormous, non-volatile capacity is the driving force behind file structure design. FS should give access to all the capacity without making the application spend a lot of time waiting for the disk. FS is a combination of representation for data in files and of operations for accessing the data.

- It allows applications to read, write and modify data
- Also finding the data
- Or reading the data in a particular order

Efficiency of FS design for a particular application is decided on,
1. Details of the representation of the data
2. Implementation of the operations.

A large variety in the types of data and in the needs of application makes FS design important. What is best for one situation may be terrible for other.
A file system is a process that manages how and where data on a storage disk, typically a hard disk drive (HDD), is stored, accessed and managed. It is a logical disk component that manages a disk's internal operations as it relates to a computer and is abstract to a human user. Regardless of type and usage, a disk contains a file system and information about where disk data is stored and how it may be accessed by a user or application. A file system typically manages operations, such as storage management, file naming, directories/folders, metadata, access rules and privilege.

## 1.2 Introduction to Python

- Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.
- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.
- Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

## 1.3 Introduction to Indexing

- Indexing is a data structure technique which allows you to quickly retrieve records from a database file.
- An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of pointers for holding the address of the disk block where that specific key value stored.
- An index takes a search key as input and efficiently returns a collection of matching records.

Type of Indexes:

- Primary Indexing - Primary Index is an ordered file which is fixed length size with two fields. The first field is the same a primary key and second, filed is pointed to that specific data block. In the primary Index, there is always one to one relationship between the entries in the index table.
- Secondary Indexing -The secondary Index can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index.

## 1.4 Scope and importance of work

- Data analysis is a process used to inspect, clean, transform and remodel data with a view to reach to a certain conclusion for a given situation.
- Data analysis helps in structuring the findings from different sources of data.
- Data analysis is very helpful in breaking a macro problem into micro parts.
- Data analysis acts like a filter when it comes to acquiring meaningful insights out of huge data set.
- Data analysis helps in keeping human bias away from the research conclusion with the help of proper statistical treatment.

When discussing data analysis it is important to mention that a methodology to analyse data needs to be picked. If a specific methodology is not selected data can neither be collected nor analysed. The methodology should be present in the dissertation as it enables the reader to understand which methods have been used during the research and what type of data has been collected and analysed throughout the process. The dissertation also presents a critical analysis of various methods and techniques that were considered but ultimately not used for the data analysis. An effective research methodology leads to better data collection and analysis and leads the researcher to arrive at valid and logical conclusions in the research. Without a specific methodology, observations and findings in a research cannot be made which means methodology is an essential part of a research or dissertation.
User will be able analysis time taken to build the index. This in turn will help them in design the index based on the time taken for the index to build. Data analysis plays a very major in any of the project we do.

# CHAPTER 2

# IMPLEMENTATION

## 2.1 Basic operations on Indexing

In this section, we present the details of the operations of indexing:

- Entering the details.
- Searching.
- Deleting.
- Build Index
- Modify

## 2.2 Algorithm

Step 1: Accessing a particular dataset.
Step 2: Create 2 new index files, one for primary indexing and the other for secondary indexing.
Step 3: The first column is the search key that contains a copy of primary key or candidate key of the table.
Step 4: The second column is the pointer which contains a set of pointers holding the address of the disk block where that particular key value is found.
Step 5: Record insertion - This consists of appending the data file and inserting a new record. The rearrangement of the index consists of sliding down the records with keys larger than the inserted key and then placing new record in the opened space.
Step 6: Record deletion-This should use the techniques for reclaiming space in files when deleting from the data file. We must delete the corresponding entry from the index. Shift all records with keys larger than key of the deleted record to the previous position in memory or make the index entry as deleted.
Step 7: In our record file we built an index for 'ID' which is primary key and there is 'S_ID' as secondary key.
Step 8: Record deletion in secondary key: Deleting a record implies removing all the references to the record in primary index and in all secondary indexes. When accessing the file through secondary key, the primary indexed file will be checked and a deleted record can be identified.
Step 9: It allows binary search to obtain a keyed access to a record in variable length record file.
Step 10: Time taken for dataset has been calculated for each functionality.

# CHAPTER 3

# SEARCH ALGORITHM AND TIME COMPLEXITY

## 3. Time Complexity is often measured in terms of:

### 3.3.1 Big Oh (O): worst case running time

In analysis algorithm, Big Oh is often used to describe the worst-case of an algorithm by taking the highest order of a polynomial function and ignoring all the constants value since they aren't too influential for sufficiently large input. So, if an algorithm has running time like f(n) = 3n + 100, we can simply state that algorithm has the complexity O(n) which means it always execute at most n procedures(ignoring the constant '100' in between and also the constant '3' being multiplied by n). Thus, we can guarantee that algorithm would not be bad than the worst-case.

Example: prove that f (n) = $5n^2 + 3n + 2$ has Big Oh O ($n^2$)

### 3.3.2. Big Omega (Ω): best case running time

Big Omega is often used to describe the best-case running time of an algorithm by choosing the lowest order of the polynomial function and ignoring all the constants.

Example: prove that f (n) = $5n^2 + 3n$ has Big Ω (n)

We know that the lowest order of the polynomial function f (n) (i.e. 1) is less than $n^2$, thus we can conclude that f (n) has a big Ω (n)

### 3.3.3. Big Theta (Θ): both best and worst case running time

Big Theta describes the running time of an algorithm which is both best case and worst case.

This idea might be a big tricky to grasp at first but don't worry, it's not that different from Big Oh or Big Omega, though if you don't completely get the concept of Big Oh and Big Omega, you might want to go through that again before moving onto this. An algorithm has both best and worst case running times. What does that mean? It means that the algorithm is both Big Oh and Big Omega at the same time. In simpler words, if we consider a polynomial function f (n) = $3n^3 + 4n$, the Big Theta of the function is going to neither be greater nor smaller than the highest order of the function. It's going to be exactly equal to it, which in this case is going to be $n^3$.

## 3.2 Binary search:

Binary search is the search technique which works efficiently on the sorted lists. Hence, in order to search an element into some list by using binary search technique, we must ensure that the list is sorted.

Binary search follows divide and conquer approach in which, the list is divided into two halves and the item is compared with the middle element of the list. If the match is found then, the location of middle element is returned otherwise, we search into either of the halves depending upon the result produced through the match.

```
Procedure binary_search
   A ← sorted array
   n ← size of array
   x ← value to be searched

   Set lowerBound = 1
   Set upperBound = n

   while x not found
      if upperBound < lowerBound
         EXIT: x does not exists.

      set midPoint = lowerBound + ( upperBound - lowerBound ) / 2

      if A[midPoint] < x
         set lowerBound = midPoint + 1

      if A[midPoint] > x
         set upperBound = midPoint - 1

      if A[midPoint] = x
         EXIT: x found at location midPoint
   end while

end procedure
```

**Fig 3.2.1:** Binary search algorithm

# CHAPTER 4
# RESULTS AND SCREENSHOTS

## 4.1 Primary Indexing

| | A | B | C |
|---|---|---|---|
| 1 | PrimaryKe | Index Value | |
| 2 | 634 | 93929 | |
| 3 | 635 | 94383 | |
| 4 | 636 | 94710 | |
| 5 | 637 | 95183 | |
| 6 | 638 | 95929 | |
| 7 | 639 | 96447 | |
| 8 | 640 | 97033 | |
| 9 | 641 | 97995 | |
| 10 | 642 | 98540 | |
| 11 | 643 | 99148 | |
| 12 | 644 | 99784 | |
| 13 | 645 | 100517 | |
| 14 | 646 | 100964 | |
| 15 | 647 | 101435 | |
| 16 | 648 | 101890 | |
| 17 | 649 | 102331 | |
| 18 | 650 | 102634 | |
| 19 | 651 | 103262 | |
| 20 | 652 | 103836 | |
| 21 | 653 | 104029 | |
| 22 | 654 | 104326 | |
| 23 | 655 | 104969 | |
| 24 | 656 | 105163 | |
| 25 | 657 | 105684 | |

**Fig 4.1.1** Primary Indexing file

## 4.2 Secondary Index

| | A | B |
|---|---|---|
| 1 | secondary | Index Value |
| 2 | 1 | 2723926 |
| 3 | 2 | 1823851 |
| 4 | 3 | 1823921 |
| 5 | 4 | 1823968 |
| 6 | 5 | 1824014 |
| 7 | 6 | 1824074 |
| 8 | 7 | 1824142 |
| 9 | 8 | 1824197 |
| 10 | 9 | 1824247 |
| 11 | 10 | 1824301 |
| 12 | 11 | 1823798 |
| 13 | 12 | 1824350 |
| 14 | 13 | 1824457 |
| 15 | 14 | 1824528 |
| 16 | 15 | 1824575 |
| 17 | 16 | 1824635 |
| 18 | 17 | 1824682 |
| 19 | 18 | 1824733 |
| 20 | 19 | 1824790 |
| 21 | 20 | 1824849 |
| 22 | 21 | 1824899 |
| 23 | 22 | 1824401 |
| 24 | 23 | 1824948 |
| 25 | 24 | 1823735 |

index3

**Fig 4.2.1** Secondary Indexing file

## 4.3 Inserting records

```
------WELCOME TO CHICAGO CRIMES DATASET------
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 1

Enter ID:
232020

Enter Casenumber:
GHN343

Enter Description:
MURDER

Enter District:
78

Enter Ward:
9

Enter FBI code:
7

Enter Year:
2020
Time taken to insert a record in ms is :  42
```

**Fig 4.4.1** Entering the details

In the above image the user has selected primary indexing and further has selected choice 1 in which you can enter details of Chicago-crimes dataset, thime taken to insert a record is 42ms.

## 4.4 Searching based on primary key

```
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 3

enter key 4786321
found
4786321,HM399414,FINANCIAL ID THEFT: OVER $300,4,7.0,6,2004,1.0

ID: 4786321
Case number: HM399414
Description: FINANCIAL ID THEFT: OVER $300
District: 4
Ward: 7.0
FBI code: 6
Year: 2004
Time taken to search a record in ms is :  349
```

**Fig 4.4.1** Searching based on  primary key

Here the user has selected choice 2 in which you can search based on the primary key. Time taken to search a record is 349ms.

## 4.5 Searching based on Secondary key

```
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 4

enter key 5
found
1.0,6,2004,4.0

ID: 1.0
Case number: 6
Description: 2004
District: 4.0
```

**Fig 4.5.1** Searching based on secondary key

## 4.6 Deletion based on primary key

```
------WELCOME TO CHICAGO CRIMES DATASET------
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 2

Enter the ID to delete:
4786321
DELETED SUCCESSFULLY
Time taken to delete a record in ms is :  33
```

**Fig 4.6.1** Deletion based on primary key

In this image user has selected choice 2, in which you can perform deletion using primary key(ID) . Time taken to delete a record is 33ms.

## 4.7 Deletion based on secondary key

```
------WELCOME TO CHICAGO CRIMES DATASET------
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 6

Enter the S_ID to delete:
5000
DELETED SUCCESSFULLY
Time taken to delete a record in ms is :  20
```

**Fig 4.7.1** Deletion based on secondary key

In this image user has selected choice 6, in which they perform deletion using secondary key(S_ID). Time taken to delete a record is 20ms.

## 4.8 Modify records

```
------WELCOME TO CHICAGO CRIMES DATASET------
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY

PLEASE ENTER YOUR OPTION: 5

Enter key: 1372507
FOUND
ID: 1372507
Case Number: G066638
Description: SIMPLE
District: 6
Ward:
FBI Code: 08B
Year: 2001
Enter 1 to modify Description
Enter 2 to modify District
Enter 3 to modify Ward number
Enter 4 to modify FBI code
Enter 5 to go back to main menu

PLEASE ENTER YOUR OPTION: 4

Enter the new FBI code: 20
1372507,G066638,SIMPLE,20
```

**Fig 4.8.1** Modify record

In image 4.4.7 user has selected choice 5, in which they perform record modification. Here, they choose to modify the field ‘FBI code’.

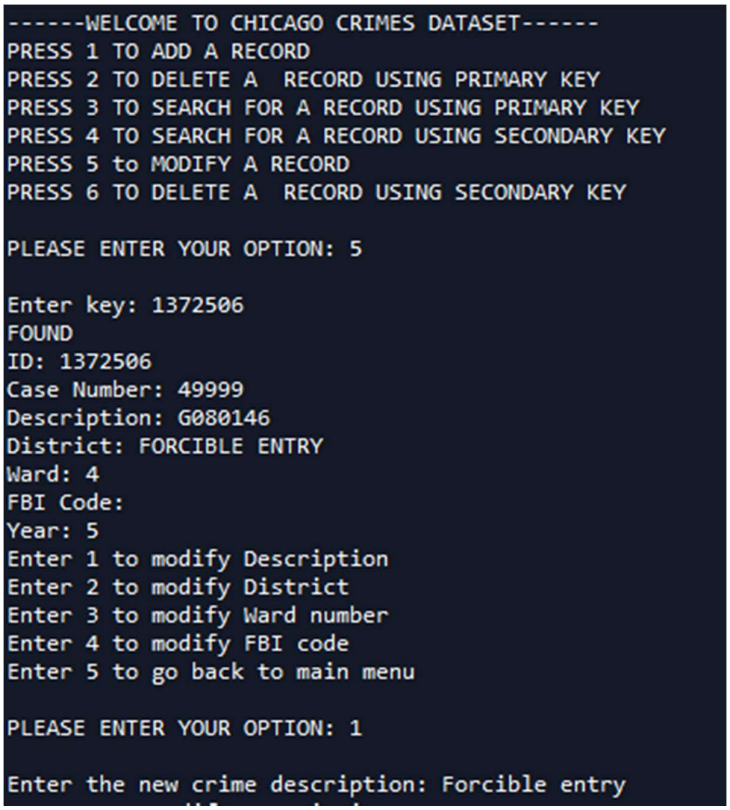


**Fig 4.8.2** Modify record

In image 4.4.8 user has selected choice 5, in which they perform record modification. Here, they choose to modify the field ‘Description’.

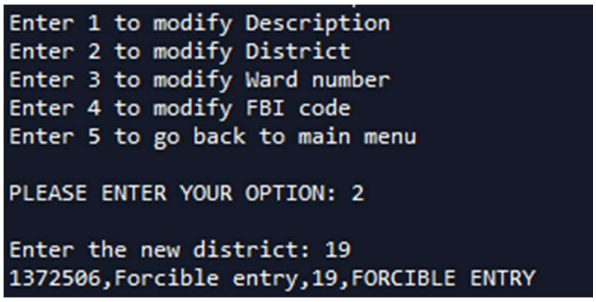


**Fig 4.8.3** Modify record

In image 4.4.9 user has selected choice 5, in which they perform record modification. Here, they choose to modify the field ‘District’.

```
Enter 1 to modify Description
Enter 2 to modify District
Enter 3 to modify Ward number
Enter 4 to modify FBI code
Enter 5 to go back to main menu

PLEASE ENTER YOUR OPTION: 3

Enter the new ward number: 6
1372506,Forcible entry,19,6
```

**Fig 4.8.4** Modify record

In image 4.4.10 user has selected choice 5, in which they perform record modification. Here, they choose to modify the field 'Ward number'.

```
Enter 1 to modify Description
Enter 2 to modify District
Enter 3 to modify Ward number
Enter 4 to modify FBI code
Enter 5 to go back to main menu

PLEASE ENTER YOUR OPTION: 4

Enter the new FBI code: 10
1372506,Forcible entry,19,10
Enter 1 to modify Description
Enter 2 to modify District
Enter 3 to modify Ward number
Enter 4 to modify FBI code
Enter 5 to go back to main menu

PLEASE ENTER YOUR OPTION: 5
------WELCOME TO CHICAGO CRIMES DATASET------
PRESS 1 TO ADD A RECORD
PRESS 2 TO DELETE A  RECORD USING PRIMARY KEY
PRESS 3 TO SEARCH FOR A RECORD USING PRIMARY KEY
PRESS 4 TO SEARCH FOR A RECORD USING SECONDARY KEY
PRESS 5 to MODIFY A RECORD
PRESS 6 TO DELETE A  RECORD USING SECONDARY KEY
```

**Fig 4.8.5** Modify record

In image 4.4.11 user has selected choice 5, in which they perform record modification. Here, they choose to modify the field 'FBI code' and then exits out of the Modify option.

## 4.9 TIME ANALYSIS

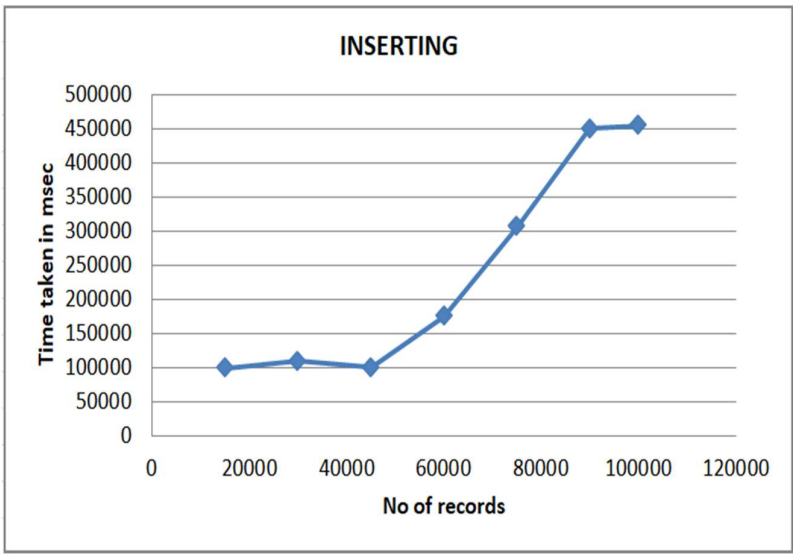


**Fig 4.9.1** : The time taken to insert record into the file, as the number of record increases the time also increases.

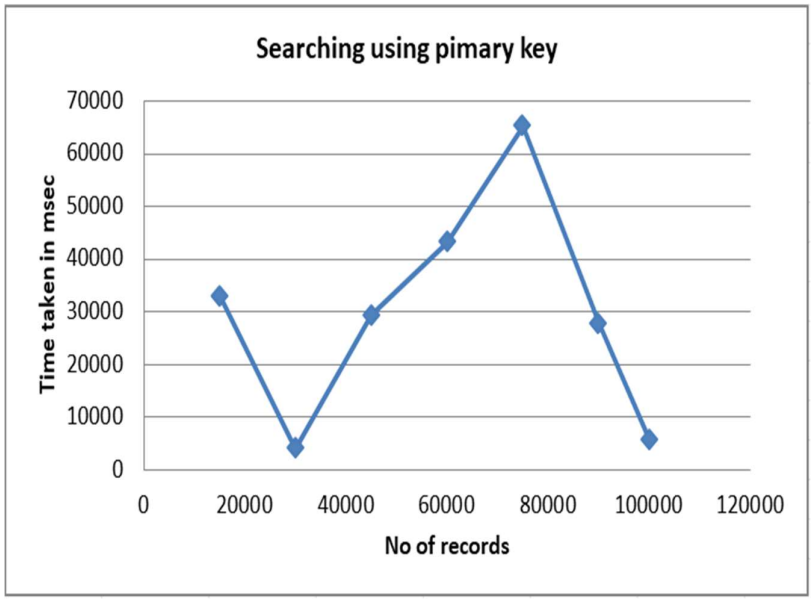


**Fig 4.9.2** : Time analysis for searching the record in file by using primary key, as the number of record increases the time will be delayed to search the record.

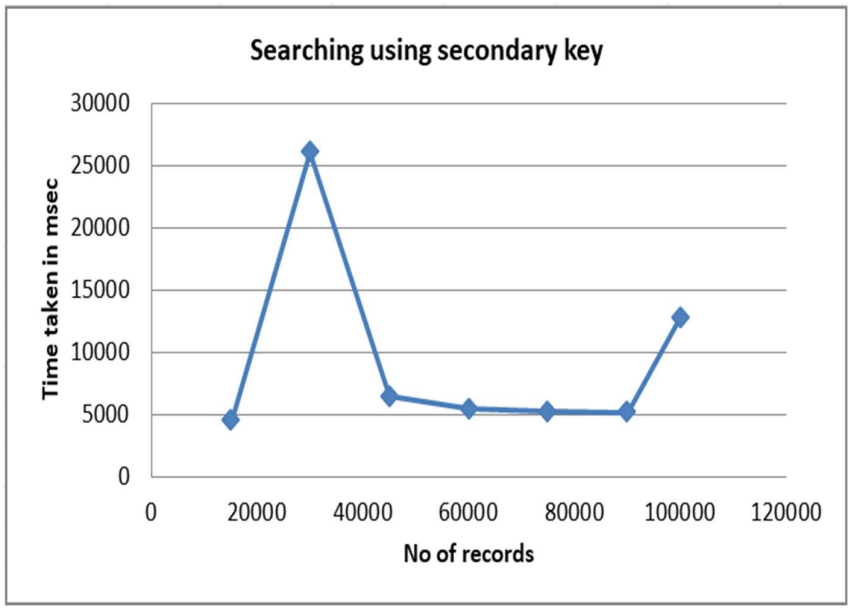


**Fig 4.9.3** : Time analysis for searching the record in file by using secondary key, as the number of record increases the time will be delayed to search the record.

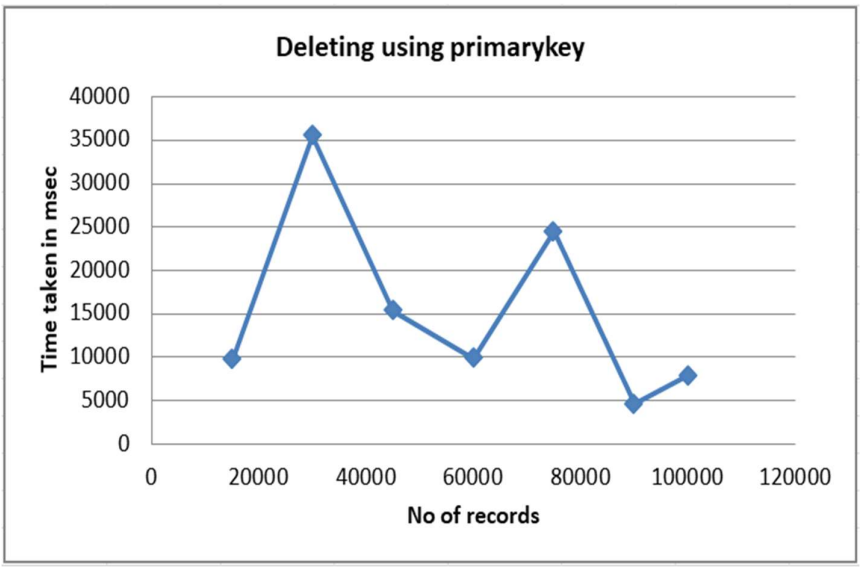


**Fig 4.9.4**: Time analysis for deleting the record from the file, decreases as the record increases.

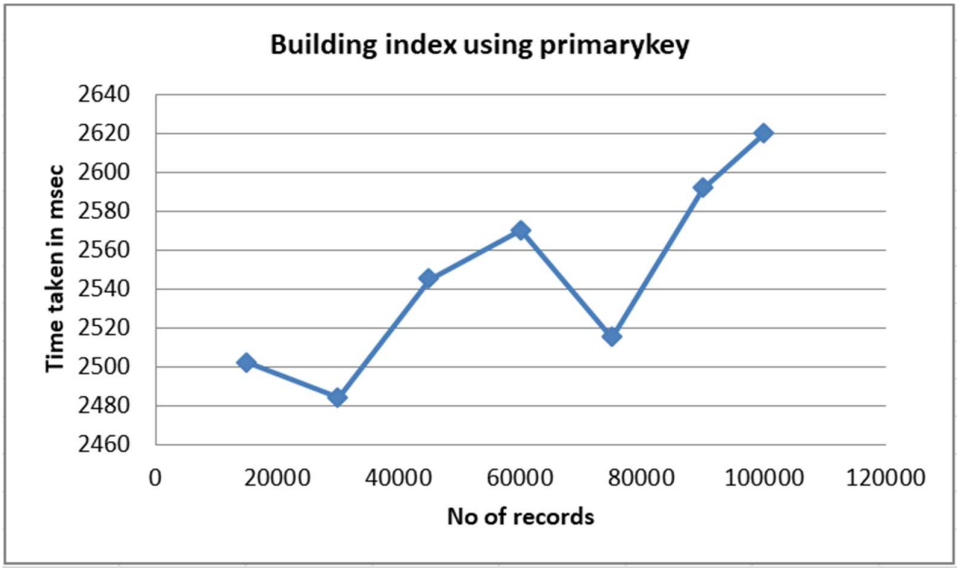


**Fig 4.9.5:** Time analysis for building primary index.

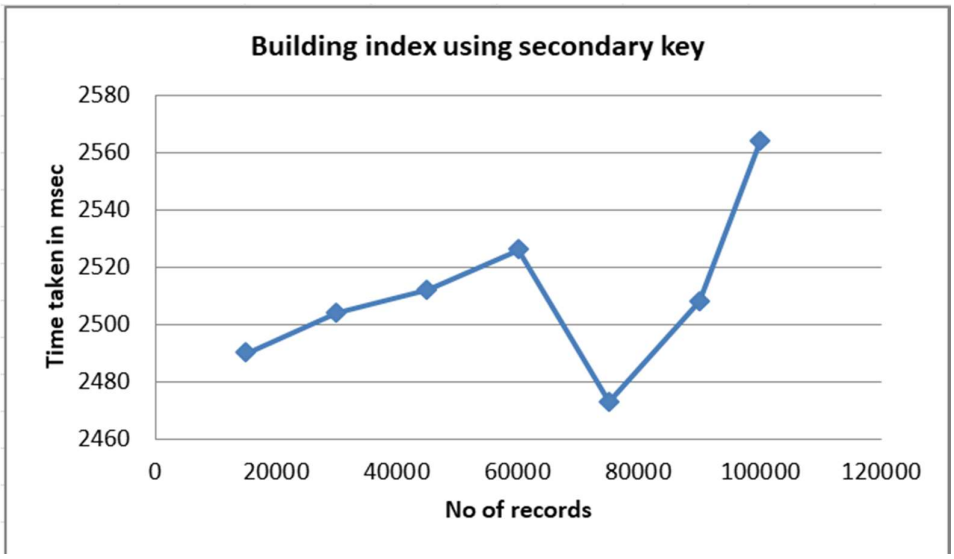


**Fig 4.9.6**: Time analysis for building secondary index.

# CONCLUSION

We have successfully implemented indexing which helps us in administrating the data used for managing the tasks performed.

View tables are used to display all the components at once so that user can see all the components of a particular type at once. One can just select the component and modify and remove the component.

We have successfully used various functionalities of Python and created the File structures.

Features:

1. Clean separation of various components to facilitate easy modification and revision.
2. All the data is maintained in a separate file to facilitate easy modification
3. All the data required for different operations is kept in a separate file.
4. Quick and easy saving and loading of database file.

## REFERENCES

The information about indexing was gathered by referring to the following sites**:**


- Wikipedia(www.wikipedia.org)
- Stackoverflow(stackoverflow.com)
- GeeksforGeeks(GeeksforGeeks.com)
- Michael J.Folk, Bill Zoeclick, Greg Riccardi:File Structures-An Object Oriented Approach with C++,3rd Edition,Pearson Education ,1998.
- Scott Robert Ladd:C++ Components and Algorithms,BPB Publications,1993.