Database Modelling using C and Hash Tables

Authored by:
Vishwas Rajashekar
Department of Computer Science and Engineering
PES University, Bangalore

Project Inspired By:
Prof. V.R. Badriprasad
Department of Computer Science and Engineering
PES University, Bangalore

13/01/2019

0.1 About this Project

This project demonstrates the use of Hash Tables and Dynamic Memory Allocation in C to model the basic functionalities of a crude Database Management System (DBMS).

0.2 Features

- It offers the ability to store multiple tables
- It offers the ability to store multiple fields
- Supports 3 datatypes (int, double, and string) for each field
- Allows performing of simple queries
- Allows simple printing of database records
- Allows editting of records

0.3 How it works

0.3.1 Database management

The Database is represented by a struct type that contains:
- The name of the database
- The number of records
- The number of fields
- The total capacity
- An array of pointers to the hash tables
- An array of strings that contain the headers for the database
- An array of integers that represents the discriminator variable for each field.

0.3.2 Database Records

The records stored in the database are actually an array of the Union type that contains the datatypes. There are as many of these Unions in the array as there are fields in the database.

0.3.3 Support for multiple variables

The program allows multiple data types as the storage of data depends upon a Union. This union can contain an integer, char* or a double. To differentiate betweeen the types, a variable known as the discriminator is used. This variable stores a value that informs the system how to access the data stored in the corresponding union.

0.3.4 Storing Records

All records are stored in n different Hash Tables based on the values in their fields, where n is the number of fields for each record. This improves the searching and query performance by reducing the number of sequential search comparisons that are required to locate a record. However, it also causes more memory use as the same record is stored in multiple places. The appropriate hash function to be used is also decided using the discriminator.

0.3.5 Hash Functions

The hash functions used in this project are all simple and make use of the Linear Probing Technique.
- For strings, the length is used as the key.
- For doubles, the key value is type cast into an integer and this result is used as the key for the hash function.
- For integers, the value itself is the key.

The function used is defined as:

$$\text{Hash Value } = \text{Key } \textit{mod} \text{ HASH CONSTANT } + i$$

where the HASH CONSTANT value is the size of the hash table and i is a counter that tracks the number of collisions that occurred. It initially start at 0 and increments. This is the principle of Linear Probing.

0.3.6 Querying the Database

The program currently only supports simple search queries. The input considered is the index value of the column as displayed and the value that the user would like to find inside the database.

0.4 Sample Inputs and Outputs

```
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
1
Enter the name of the dbase and press Enter
STUDENTS
Enter number of fields and press Enter: 3
Enter FIELDNAME<space>DATATYPE where DATATYPE is (int,dbl,str)
NAME str
AGE int
ID str
Database:  successfully created
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
3
Enter the index number of the database you want
0 |--> STUDENTS
0
NAME    AGE     ID
vishwas 19 P1234

The record was successfully created
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
```

Figure 1: Creation of Database and Record

```
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
3
Enter the index number of the database you want
0 |--> STUDENTS
0
NAME    AGE     ID
personx 23 U876

The record was successfully created
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
6
Enter the index number of the database you want
0 |--> STUDENTS
0
NAME    AGE     ID
vishwas 19      P1234
personx 23      U876

personx 23      U876
vishwas 19      P1234

personx 23      U876
vishwas 19      P1234
```

Figure 2: Displaying Records

```
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
4
Enter the index number of the database you want
0 |--> STUDENTS
0

Columns:
0|-->NAME
1|-->AGE
2|-->ID

Enter (column_index search_value) of choice:
0 personx
Record Found
personx 23      U84
Record Deleted
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
6
Enter the index number of the database you want
0 |--> STUDENTS
0
NAME    AGE     ID
vishwas 19      P442
```

Figure 3: Deleting Record

```
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
5
0 |--> test
Enter the index number of the database you want and press Enter
0

Columns:
0|-->f1
1|-->f2

Enter (column_index search_value) of choice:
0 hello
Record Found
Re-enter the record
f1      f2
hello   world
bye world
Record successfully editted
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
6
Enter the index number of the database you want
0 |--> test
0
f1      f2
bye     world

bye     world
```

Figure 4: Edit a Record

```
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
2
Enter the index number of the database you want and press Enter
0

Columns:
0|-->NAME
1|-->AGE

Enter (column_index search_value) of choice:
0 vishwas
Record Found
vishwas 19
What would you like to do?
 1) Create a new database
 2) Query an existing database
 3) Create a new record
 4) Delete a record
 5) Edit a record
 6) Print all records
 7) Quit
6
Enter the index number of the database you want
0 |--> STUDENTS
0
NAME    AGE
vishwas 19
personx 89
persony 23

personx 89
persony 23
vishwas 19
```

Figure 5: Query the database