# Tutorial of MicroDB

## Tutorial



# Introduction

NoSQL databases have proven popular because they allow the data to be stored in ways that are easier to understand or closer to the way the data is used by applications. Fewer transformations are required when the data is stored or retrieved for use. Many different types of data, whether structured, unstructured, or semi-structured, can be stored and retrieved more easily. And they are used more and more commonly. MicroDB is designed to fulfill the needs of more flexible usage in the real world.

This is a tutorial for users on how to use and operate the MicroDB. It contains the usage and examples of every step and function. After reading this tutorial, users can know exactly how to use that database system and those who never used NoSQL databases before will have a basic understanding of them. Compare to other popular NoSQL databases, we want to make our product more friendly for new users. Suppose lots of users are new to this product, they can quickly get familiar with this system by the inside instructions.

In this database, we added a lot of common functions such as inserting, searching, and deleting. To make our product more rigorous, we wrote another program to test the performance and correctness of the database.

Now let’s start!

# Example Analysis

Since we need to do some operations about databases and key-value pairs inside them. Suppose we will create a database called “123”. We want to insert a key called key1 and value of a set{value1, value2, value3}, then we want to search the value of that key, and delete it. Then we want to insert a new key still named key1 with value v1, v2, v3 in it. Suppose we also want to find a key whose 3rd key is v3, we can use to query function to find key1. In the end, we will delete the database ”123” because it’s no longer useful to us.

# Instructions

The first step of this program is creating, finding, or deleting a database based on the input value. The initialization interface looks like that:

```
Welcome to MicroDB!
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
```

## 1. Creating or finding a database:

If we type in the name of a database that we never created before, the database with that name will be auto-created:

```
Welcome to MicroDB!
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
123
An empty DB will be built!
Db has been open.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

If the database already exists, the database can be found and we can do some further operations in that database.

```
Welcome to MicroDB!
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
123
-1
Db has been found.
Db has been open.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

## 2. Deleting a database:

If we want to delete the database, we can just use “delete-” + name as the command:

```
Welcome to MicroDB!
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
delete-123
Db has been found.
database-123has been successfully cleared!
Enter 1 to continue the program. Enter -1 to exit the program
```

If there's no database named "123", we will get the notice that database not found:

```
Welcome to MicroDB!
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
delete-123
database not found
Enter 1 to continue the program. Enter -1 to exit the program
```

Press 1 to continue or -1 to exit the program.

## 3. Inserting keys:

Then we go through the specific operations of a database:

```
Enter the name of db. if you want to delete a db, please enter 'delete-' + name of db:
123
An empty DB will be built!
Db has been open.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

If we want to insert the key and value, we just press 1 and then insert the key and value.
(Notice: We can add multiple values and they are split and recognized by ",";
Warning: At most 10 values are allowed in a key-value pair.):

```
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
1
------------------------insert operation----------------------
insert key :
key1
insert value (use comma to split value):
value1,value2,value3
Insert operation complete.
```

## 4. Searching for keys:

If we want to search for the value of a key, we just press 2 then input the key, and all the values can be shown:

```
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
2
------------------------find operation----------------------
enter key :
key1
the 0th  value is: value1
the 1th  value is: value2
the 2th  value is: value3
Find operation complete.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

## 5. Deleting keys:

Press 3 then input the key to delete this key and its value:

```
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
3
------------------------delete operation----------------------
enter key :
key1
delete operation complete.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

Then we can insert a new key with the same name of the key we deleted just now:

```
------------------------insert operation----------------------
insert key :
key1
insert value (use comma to split value):
v1,v2,v3
Insert operation complete.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
2
------------------------find operation----------------------
enter key :
key1
the 0th  value is: v1
the 1th  value is: v2
the 2th  value is: v3
Find operation complete.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

## 6. Query function:

Press 4 then input a single string of a value to use the query function, it will return the key whose value at the specified location is the input value:

```
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
4
------------------------query operation----------------------
enter the value :
v3
enter the position of value :
2
The following keys are what you want :
1. key1
The query operation is complete.
Enter 1 to insert, enter 2 to search, enter 3 to delete, enter 4 to query, enter -1 to return to the previous menu
```

(******In this example, the key is key1 and its values are v1, v2, v3, we input v3 and index 2 to find key1, which means that the third value of key1 is v3. ******)

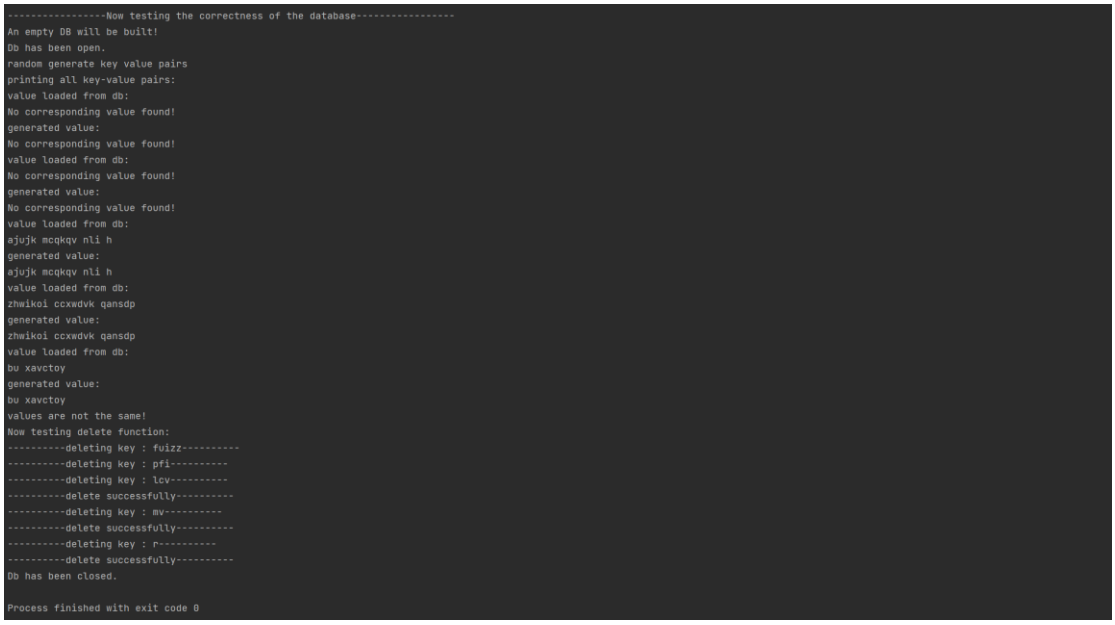
In the end, press -1 to end the program:

```
Enter 1 to continue the program. Enter -1 to exit the program
-1
Goodbye!
```

********Warning: We deal with some exceptions of some potential errors, including but not limited to inputting a character or a word when the system wants us to input a number or inputting a number that the system didn't want us to input. Like we should input the number from 1 to 4, but actually, we input 6. Although we took these situations into consideration, but users still shoulf notice that. ********

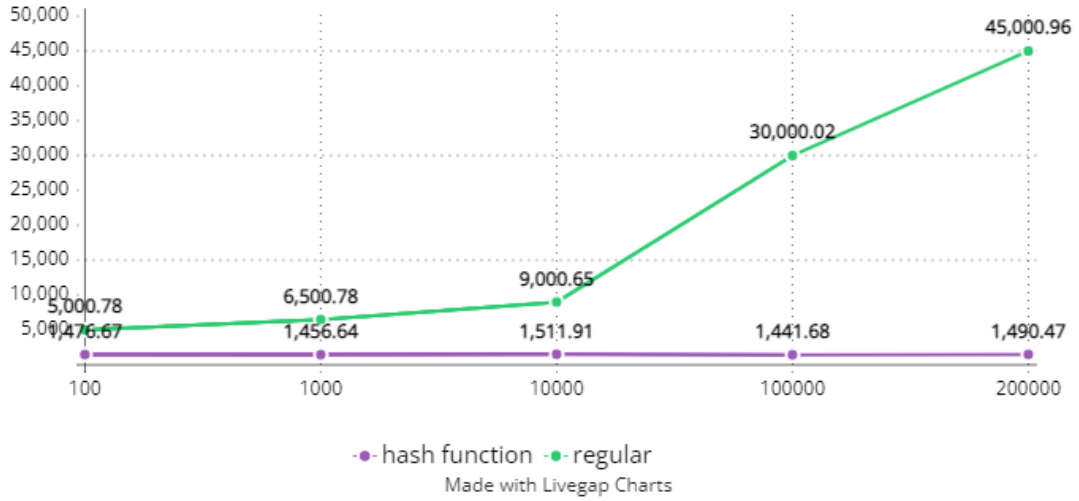
# Test function

We also provide a test function. We can just run the test function program and it will automatically test the performance and correctness of the database. When we are testing the performance, we used some CSV files (We grabbed the data from IMDB and the attributes of the movies as the values of the keys) as our data source. Here is the interface:

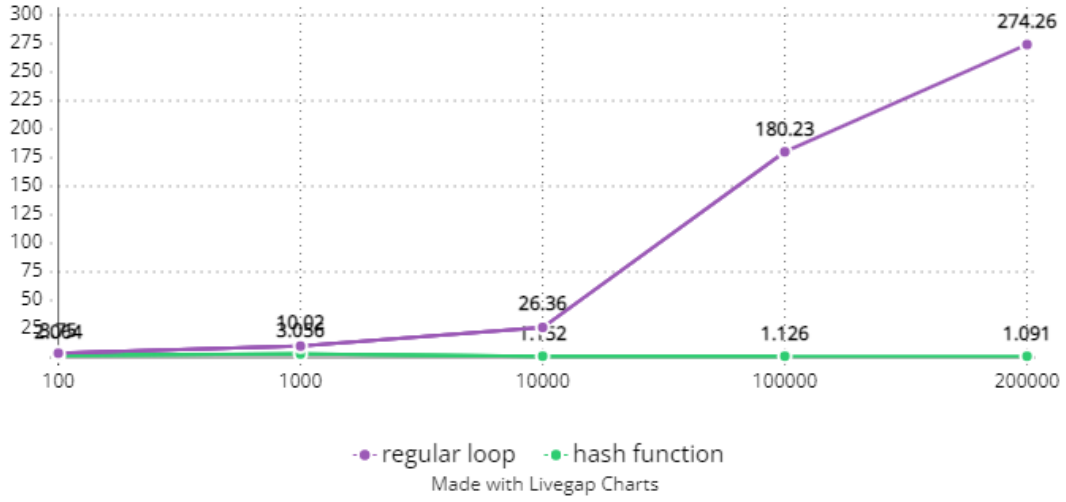




We compared the time of searching values in database between searching with hash functions and without hash function. The results are as following:

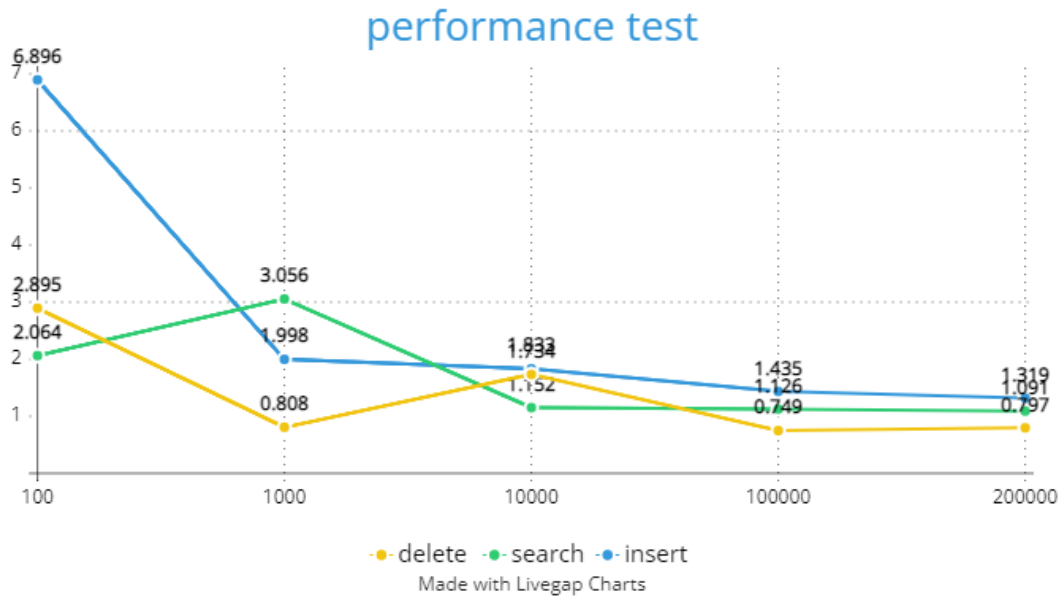
## query performance between hash and regular




## search performance between hash and regular




This table records the time of running functions including delete, search and insert in different volumes of datasets.




If you have further questions, feel free to contact us at cz2664@columbia.edu, lz2761@columbia.edu, and sp3895@columbia.edu.