# TITLEA

**Project Report**

*submitted in partial fulfillment of the requirements for the award of the degree of*

# BACHELOR OF TECHNOLOGY

**in COMPUTER SCIENCE**

**Specialization in**

**Virtualization**

By :

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll No** | **Branch** |
| Siddhant Gupta | R142218059 | GG |
| Akash Srivastava | R142218079 | GG |
| Amishi Tyagi | R142218010 | GG |
| Jyoti Mishra | R110218075 | CCVT |

***Under the guidance of***

# Mr. Lalit Kane

# Department of Virtualization



**Department of Informatics School of Computer Science**

### UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

**Bidholi, Via Prem Nagar, Dehradun, Uttarakhand 2020-21**



**CANDIDATES DECLARATION**

I/We hereby certify that the project work entitled **Cloning of DBMS** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Cloud Computing and Virtualization and submitted to the Department of Virtualization at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from Augustto December 2020under the supervision of **Lalit Kane, Department of Virtualization**.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

### Siddhant Gupta(059)

### Akash Srivastava(079)

### Amishi Tyagi(010)

### Jyoti Mishra(075)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 27 December 2020) Mr. **Lalit Kane**

Project Guide

### Dr. Deepshikha Bhargava

Head

Department of Virtualization School of Computer Science

University of Petroleum And Energy Studies Dehradun - 248 001 (Uttarakhand)

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Lalit Kane**, for all advice, encouragement and constant support he has given us through out our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of the Department, **Dr. Deepshikha Bhargava**, for his great support in doing our **project Cloning of DBMS** at **SoCS**.

We are also grateful to **Dr. Manish Prateek Professor and Director SoCS** and **Dr. Kamal Bansal Dean CoES**, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Siddhant Gupta** | **Akash Srivastava** | **Amishi Tyagi** | **Jyoti Mishra** |
| **Roll No.** | **R142218059** | **R142218079** | **R142218010** | **R110218075** |

# ABSTRACT

The database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. A database management system stores data in such a way that it becomes easier to retrieve, manipulate, and produce information. It consists of a group of programs that manipulate the database and provide an interface between the databases. It includes the use of the database and other application programs. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data. DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.

The current trend of research on Accessing File System through SQL with several variants of these algorithms has been developed. The current research is to study and develop algorithms that can access the file system through SQL commands using C programming language.

# TABLE OF CONTENTS

**Contents**

|  |  |  |
| --- | --- | --- |
| **1** | **Introduction** | **7** |
| **2** | **Literature Review** | **7** |
| **3** | **Problem Statement** | **7** |
| **4** | **Objective** | **7** |
| **5** | **Design Methodology** | **7** |
| **6** | **Implementation** | **7** |
|  | 6.1 Pseudocode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 6.2 Output Screen . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 6.3 Result Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| **7** | **Conclusion and Future Scope** | **9** |
| **A** | **APPENDIX I PROJECT CODE** | **11** |

**LIST OF TABLES**

**List of Tables**

1 First table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

# LIST OF FIGURES

**List of Figures**

1 *Pert Chart* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

# Introduction

The database is a gathering of related information which aides in effective retrieval, insertion, and deletion of information from the database and arranges the information as tables, sees, outlines, reports and so on. The software used to manage the database is called Database Management System. A Database Management System generally manipulates the data itself, the data format, field names, record structure, and file structure. It also defines rules to validate and manipulate this data. A Database Management System relieves users of framing programs for data maintenance and always provides data independence. Any change in storage mechanism and formats are performed without modifying the entire application. Database systems are meant to handle a large collection of data which is important to manage and gain important information and insights. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information.

# Literature Review

SQLite is a C-language library that implements a [small](https://www.sqlite.org/footprint.html), [fast](https://www.sqlite.org/fasterthanfs.html), [self-contained](https://www.sqlite.org/selfcontained.html), [high- reliability](https://www.sqlite.org/hirely.html), [full-featured](https://www.sqlite.org/fullsql.html), SQL database engine. SQLite is the [most used](https://www.sqlite.org/mostdeployed.html) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

The SQLite [file format](https://www.sqlite.org/fileformat2.html) is stable, cross-platform, and backward compatible and the developers pledge to keep it that way through at least the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems and as a long-term archival format for data. There are over 1 trillion (1e12) SQLite databases in active use.

# Problem Statement

No doubt DBMS is the one of the majorly used application in every field that requires storing and managing of data. So there are a number of open source (like MySQL) as well as licensed database management system softwares which are already available for everyone to use. We just need to enter SQL commands in a proper syntax as an input and accordingly the application processes our queries and generates the result as the output. So we as a team decided to implement the same in C as it would give us a good knowledge of storing of data in the backend as well as we also get to revise and learn a lot of new commands in C language.

# Objective

1. To design an interface for user interaction with the system.
2. Implementation of selected DDL, DML and DQL of the SQL queries to implement and design a SQL complier.

* CREATE
* INSERT
* SELECT
* DROP

1. Design and implement data structure for file data, update and retrieval.

# Design Methodology

Step 1: Identify how data will be stored on the disk.

Step 2: Design an interface for user interaction with the system, REPL.

Step 3: Select SQL queries to implement and design a SQL compiler.

Step 4: Implement a data structure for storage, retrieval, and deletion of data.

# Pert Chart

# 

# Implementation

Database systems are meant to handle a large collection of data which is important to manage and gain important information and insights. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information.

## Pseudocode

//Cloning of dbms

//This project is created under the guidance of Mr. Lalit Kane

//Team members: Akash Srivastava Siddhant Gupta Amishi Tyagi Jyoti Mishra

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <regex.h>

#include <dirent.h>

#include <errno.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/types.h>

const char \*r1 = "use [a-zA-Z\_][a-zA-Z0-9\_]+;$";

const char \*r2 = "show databases;$";

const char \*r3 = "create database [a-zA-Z\_][a-zA-Z0-9\_]+;$";

const char \*r4 = "create table [a-zA-Z\_][a-zA-Z0-9\_]+ ?\\(([a-zA-Z\_][a-zA-Z0-9\_]+ (int|string|double)\\,? ?)+\\);$";

const char \*r5 = "insert into [a-zA-Z\_][a-zA-Z0-9\_]+ values ?\\(([a-zA-Z0-9\_\\.]+\\,?)+\\);$";

const char \*r6 = "show tables;$";

const char \*r7 = "select \\\* ?from [a-zA-Z\_][a-zA-Z0-9\_]+;$";

const char \*r8 = "drop table [a-zA-Z\_][a-zA-Z0-9\_]+;$";

struct key

{

char \*input;

struct key \*next;

}\*head=NULL;

char database[10] = "default";

void createDatabase(char \*d)

{

chdir("database");

int check = mkdir(d,0700);//0 success -1 fail

chdir("..");

if(!check)

{

printf("%s Created\n",d);

}

else

{

printf("Database already exists\n");

return;

}

}

void openDatabase(char \*d)

{

chdir("database");

DIR \*check = opendir(d);//traversing

chdir("..");

if(check)

{

closedir(check);

strcpy(database,d);

printf("Switched to %s\n",database);

}

else

{

printf("%s does not exist\n",d);

}

}

void showDatabases()

{

chdir("database");

struct dirent \*dir;

DIR \*d = opendir(".");

chdir("..");

if(d)

{

while((dir=readdir(d)) != NULL)

{

printf("%s\n",dir->d\_name);

}

//close(d);

}

else

{

printf("No database created\n");

}

}

void createTable(struct key \*K)

{

chdir("database");

chdir(database);

FILE \*fp;

if(access(K->input,F\_OK) == -1)

{

fp = fopen(K->input,"w");

K = K->next;

while(K != NULL)

{

fputs(K->input,fp);

K = K->next;

if(K != NULL)

fputc(',',fp);

}

fclose(fp);

printf("Table created\n");

}

else

{

printf("Table already exists\n");

}

chdir("..");

chdir("..");

}

void showTables()

{

chdir("database");

chdir(database);

struct dirent \*dir;

DIR \*d = opendir(".");

if(d != NULL)

{

while((dir=readdir(d)) != NULL)

{

printf("%s\n",dir->d\_name);

}

//close(d);

}

else

{

printf("No tables created\n");

}

chdir("..");

chdir("..");

}

void deleteTable(char \*t)

{

chdir("database");

chdir(database);

if(remove(t) == 0)

{

printf("Table deleted\n");

}

else

{

printf("Table does not exist\n");

}

chdir("..");

chdir("..");

}

int validateData(struct key \*K)

{

FILE \*fp;

fp = fopen(K->input,"r");

char s[100];

fgets(s,100,fp);

fclose(fp);

K = K->next->next;

char \*t = strtok(s,",");

int i=0;

while(t != NULL)

{

i++;

t= strtok(NULL,",");

if(strcmp("int",t)==0)

{

char p[20];

snprintf(p,20,"%d",atoi(K->input)); //atoi- typcasting to convert string to integer snprintf- write in file

if(strcmp(p,K->input) != 0)

{

printf("Error: '%s' is not of type 'int'\n",K->input);

return 0;

}

}

else if(strcmp("double",t)==0)

{

char p[20];

snprintf(p,20,"%g",atof(K->input));

if(strcmp(p,K->input) != 0)

{

printf("Error: '%s' is not of type 'double'\n",K->input);

return 0;

}

}

K = K->next;

t= strtok(NULL,",");

if(K == NULL && t != NULL)

{

printf("Error: Less number of attributes are given\n");

return 0;

}

}

if(K != NULL)

{

printf("Error: Table has %d attributes\n",i);

return 0;

}

return 1;

}

void insertTable(struct key \*K)

{

chdir("database");

chdir(database);

if(access(K->input,F\_OK) != -1)

{

if(validateData(K)==1)

{

FILE \*fp;

fp = fopen(K->input,"a");

K = K->next->next;

fputc('\n',fp);

while(K != NULL)

{

fputs(K->input,fp);

K = K->next;

if(K != NULL)

fputc(',',fp);

}

fclose(fp);

printf("1 row inserted\n");

}

}

else

{

printf("Table does not exist\n");

}

chdir("..");

chdir("..");

}

void selectAllCol(char \*f)

{

chdir("database");

chdir(database);

if(access(f,F\_OK) != -1)

{

FILE \*fp;

fp = fopen(f,"r");

char s[100];

fgets(s,100,fp);

char \*t = strtok(s,",");

while(t != NULL)

{

printf("\t%s",t);

t= strtok(NULL,",");

t= strtok(NULL,",");

}

printf("\n");

int i=0;

while(fgets(s,100,fp) != NULL)

{

t = strtok(s,",");

while(t != NULL)

{

printf("\t%s",t);

t= strtok(NULL,",");

}

i++;

}

printf("\n%d row(s) selected\n",i);

fclose(fp);

}

else

{

printf("Table does not exist\n");

}

chdir(".."); //previous directory

chdir("..");

}

void selectTable(struct key \*K)

{

if (strcmp("from",K->input)==0)

{

K = K->next;

selectAllCol(K->input);

}

}

char query[100];

int match(const char \*string, const char \*pattern)

{

regex\_t re;

if (regcomp(&re, pattern, REG\_EXTENDED|REG\_NOSUB) != 0) return 0; //Use Extended Regular Expressions. //Report only success/fail in regexec()

int status = regexec(&re, string, 0, NULL, 0);//regular expression matching

regfree(&re);

if (status != 0) return 0;

return 1;

}

int checkSyntax()

{

char \*p = &query[0];

if(match(p,r1)==1)

return 1;

else if(match(p,r2)==1)

return 1;

else if(match(p,r3)==1)

return 1;

else if(match(p,r4)==1)

return 1;

else if(match(p,r5)==1)

return 1;

else if(match(p,r6)==1)

return 1;

else if(match(p,r7)==1)

return 1;

else if(match(p,r8)==1)

return 1;

else

return 0;

}

void storeKeywords(char \*p)

{

struct key \*K = (struct key\*)malloc(sizeof(struct key));

K->input=p;

K->next = NULL;

if(head == NULL)

head = K;

else

{

struct key \*t = head;

while(t->next != NULL)

t = t->next;

t->next = K;

}

}

void findKeywords()

{

char \*t = strtok(query," (),;\*");

while(t != NULL)

{

storeKeywords(t);

t= strtok(NULL," (),;\*");

}

}

void processQuery()

{

struct key \*K = head;

if(strcmp("use",K->input)==0)

{

K = K->next;

openDatabase(K->input);

}

else if(strcmp("show",K->input)==0)

{

K = K->next;

if (strcmp("databases",K->input)==0)

showDatabases();

else if (strcmp("tables",K->input)==0);

showTables();

}

else if(strcmp("drop",K->input)==0)

{

K = K->next;

deleteTable(K->next->input);

}

else if(strcmp("create",K->input)==0)

{

K = K->next;

if(strcmp("database",K->input)==0)

{

K = K->next;

createDatabase(K->input);

}

else if(strcmp("table",K->input)==0)

{

K = K->next;

createTable(K);

}

}

else if(strcmp("insert",K->input)==0)

{

K = K->next->next;

insertTable(K);

}

else if(strcmp("select",K->input)==0)

{

K = K->next;

selectTable(K);

}

}

void deleteKeywords()

{

struct key \*current = head;

struct key \*next;

while(current != NULL)

{

next = current->next;

free(current);

current = next;

}

head = NULL;

}

int main()

{

int loop = 1;

printf("\n\t MINOR 1 PROJECT\t\n\n\t CLONING OF DBMS\t\n\n");

while (loop)

{

printf("DBMS> ");

// step 1

gets(query);

if(strcmp(query,"exit") == 0)

loop=0;

else if(strcmp(query,"clear") == 0)

system("clear");

else

{

// Step 2

if(checkSyntax()==1)

{

//printf("No Error\n");

findKeywords();

// Step 4

processQuery();

deleteKeywords();

}

else

printf("Syntax Error\n");

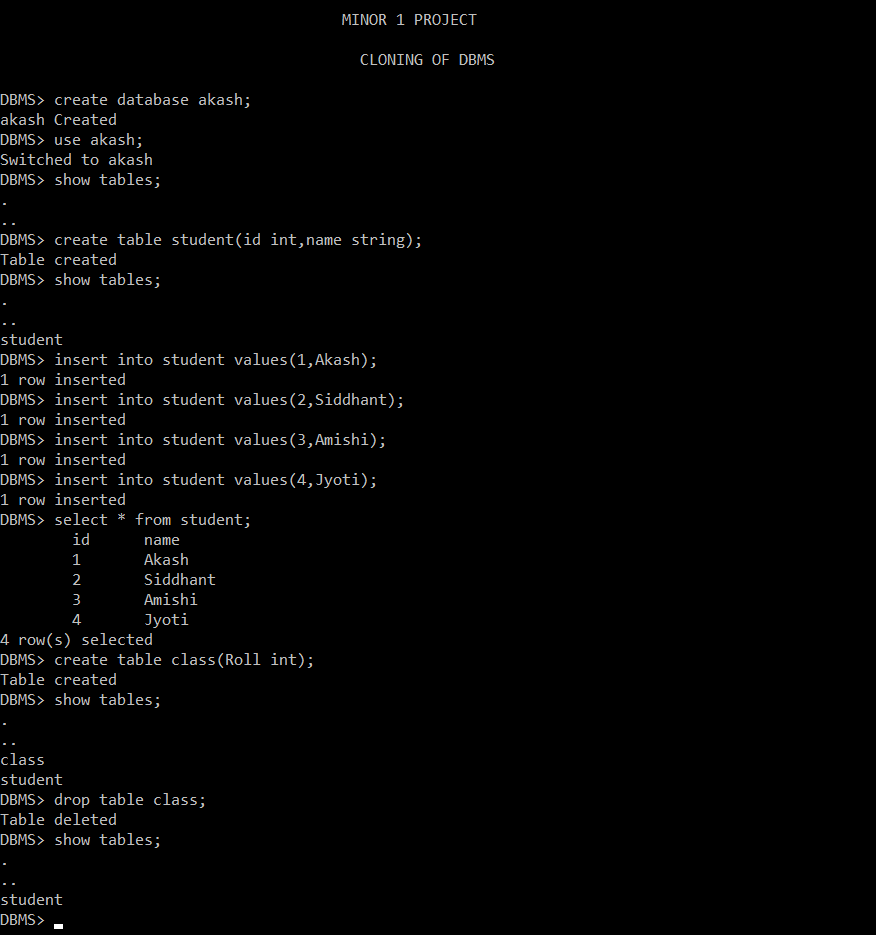
}

}

return 0;

}

## Output Screen



## Result Analysis

We just need to enter SQL commands in a proper syntax as an input and accordingly the application processes our queries and generates the result as the output. So we as a team decided to implement the same in C as it would give us a good knowledge of storing of data in the backend as well as we also get to revise and learn a lot of new commands in C language.

# Conclusion and Future Scope

The conclusion comes out to be a database in which tables can be created, deleted and data can be retrieved. Improved data access to users through the use of host and query languages. As you can see, from learning about the database management system much of what we have discussed can be very useful in applying it to our everyday lives. The database management system can be a very useful asset in the lives of people every day. After learning about the database management, it would be of great assistance when people become more aware of what they are using in the business or in their everyday life. This project gave us the opportunity to try our new skills in practice. While doing this project we also gained deeper understanding on database design and how it can be implemented in real life situations.

# References

1. Y. Ramadevi, T. Sridevi, B. Poornima, and B. Kalyani, “S Egmentation and O Bject R Ecognition Using Edge Detection Techniques,” *International Journal of Computer Science & Information Technology*, vol. 2, no. 6, pp. 153–161, 2010.
2. E. Tekin and J. Coughlan, “A bayesian algorithm for reading 1D barcodes,” *Proceedings of the 2009 Canadian Conference on Computer and Robot Vision, CRV 2009*, pp. 61–67, 2009.
3. V. Ong and D. Suhartono, “Using k-Nearest Neighbours in Optical Character Recognition,”

*ComTech*, vol. 7, no. 1, pp. 53–65, 2016.

1. L. Juan and O. Gwun, “A comparison of sift, pca-sift and surf,” *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.
2. Y. Liu and M. Liu, “Automatic Recognition Algorithm of Quick Response Code Based on Embedded System,” *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 783–788, 2006.
3. T. Gro´sz, P. Bodna´r, L. To´th, and L. G. Nyu´l, “Qr Code Localization Using Deep Neural Networks,” pp. 190–194, 2014.
4. D. Chai and F. Hock, “Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras,” *2005 5th International Conference on Information Communications & Signal Processing*, no. 01010, pp. 1595–1599, 2