

# PROGRAMS

- 1.

### **Binary Search:**

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
int Binary_Search(int a[], int N, int data)
```

```
{
```

```
int first,last,mid;
```

```
int found=0;
```

```
first=0;
```

```
last=N-1;
```

```
while(first<=last && found==0)
```

```
{
```

```
mid=int((first+last)/2);
```

```
if(a[mid]==data)
```

```
found=1;
```

```
if(a[mid]<data)
```

```
first=mid+1;
```

```
if(a[mid]>data)
```

```
last=mid-1;
```

```
}
```

```
return(found);
```

```
}
```

```
int main()
```

```
{  
clrscr();  
int x[10],data, i;  
cout<<"\n Enter 10 numbers in ascending order:";  
for(i=0;i<10;i++)  
{  
cin>>x[i];  
}  
cout<<"\n Enter No. to be searched:";  
cin>>data;  
int res=Binary_Search(x,10,data);  
if(res==1)  
cout<<"\n found";  
else  
cout<<"\n not found";  
getch();  
return 0;  
  
}
```

/\*

Enter 10 numbers in ascending order:1

4

7

9

11

14

16

19

20

23

Enter No. to be searched:11

found

\*/

2.

## Linear Search

```
#include<conio.h>

#include<iostream.h>

int Linear_Search (int x[],int N,int data)

{

int i, found=0;

for(i=0;i<N;i++)

{

if(x[i]==data)

found=1;

}

return(found);

}

int main()

{

clrscr();

int x[10],data, i;

cout<<"\n Enter 10 numbers:";

for(i=0;i<10;i++)

{

cin>>x[i];

}

cout<<"\n Enter No. to be searched:";

cin>>data;
```

```
int res=Linear_Search(x,10,data);  
if(res==1)  
cout<<"\n found";  
else  
cout<<"\n not found";  
getch();  
return 0;  
}
```

/\* Enter 10 numbers:1

2

3

4

5

6

7

8

9

10

Enter No. to be searched:5

found \*/

3.

## Bubble Sort

```
#include<conio.h>

#include<iostream.h>

#include<iomanip.h>


int main()

{

int x[10],i,j,temp;

//input phase

clrscr();

for(i=0;i<10;i++)

{

cout<<"\n Enter "<<i+1<<" value";

cin>>x[i];

}

// Processing

for(i=0;i<9;i++)

{

    for(j=0;j<=(9-i);j++)

    {

        if(x[j]>x[j+1])

        {

            temp=x[j];

            x[j]=x[j+1];

            x[j+1]=temp;
```

```

        }
    }
}

// Output Phase

cout<<"\n Sorted Array";

    for(i=0;i<10;i++)
    {
        cout<<setw(6)<<x[i];

    }

getch();

return 0;

}

/*

```

Enter 1 value 3

Enter 2 value 45

Enter 3 value 67

Enter 4 value 88

Enter 5 value 12

Enter 6 value 90

Enter 7 value 34

Enter 8 value 76

Enter 9 value 29

Enter 10 value 12

Sorted Array   3   12   12   29   34   45   67   76   88   90



```
*/
```

#### **4.Selection Sort**

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int x[10],i,j,temp,low,pos;
```

```
// input
```

```
cout<<"\n Enter 10 numbers to be sort:";
```

```
for(i=0;i<10;i++)
```

```
cin>>x[i];
```

```
// Processing
```

```
for(i=0;i<9;i++)
```

```
{
```

```
low=x[i];
```

```
pos=i;
```

```
for(j=i+1;j<10;j++)
```

```
{
```

```
if(low>x[j])
```

```
{
```

```
low=x[j];
```

```
    pos=j;
}
}
temp=x[i];
x[i]=x[pos];
x[pos]=temp;
}        // end of i for loop
// output phase
cout<<"\n Sorted Array:";
for(i=0;i<10;i++)
    cout<<setw(4)<<x[i];
```

```
getch();
```

```
return ;
```

```
}
```

```
/*
```

```
Enter 10 numbers to be sort:23
```

```
74
```

```
88
```

```
92
```

```
34
```

```
54
```

```
60
```

```
71
```

```
22
```

29

Sorted Array: 22 23 29 34 54 60 71 74 88 92

\*/

## 5.Insertion Sort

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int x[10],i,j,temp;
```

```
// input
```

```
cout<<"Enter 10 numbers to be sort:";
```

```
for(i=0;i<10;i++)
```

```
cin>>x[i];
```

```
// Processing
```

```
for(i=1;i<10;i++)
```

```
{
```

```
    temp=x[i];
```

```
    j=i-1;
```

```
    while(temp<x[j] && j>=0)
```

```
    {
```

```

        x[j+1]=x[j];

        j=j-1;

    }

    x[j+1]=temp;
}

// output

cout<<"\n sorted Array:";

for(i=0;i<10;i++)

    cout<<setw(4)<<x[i];


getch();

return;

}

/*

Enter 10 numbers to be sort:

23

65

87

66

12

16

62

94

74

```

sorted Array: 12 16 23 62 65 66 74 87 90 94

\*/

## 6.Copy Constructor

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<string.h>
```

```
class student
```

```
{
```

```
int roll;
```

```
char name[30];
```

```
public:
```

```
student()
```

```
{
```

```
cout<<"\n Constructor:";
```

```
roll=10;
```

```
strcpy(name,"Rahul");
```

```
}
```

```
student(student &s)
```

```
{
```

```
cout<<"\n Copy constructor:";
```

```
roll=s.roll;
```

```
strcpy(name,s.name);  
}  
void input_void()  
{  
cout<<"\n Enter roll no:";  
cin>>roll;  
cout<<"\n Enter name:";  
cin>>name;  
}  
void show_data()  
{  
cout<<"\n Roll no:";  
cout<<roll;  
cout<<"\n Name:";  
cout<<name;  
}};  
int main()  
{  
student s;  
s.show_data();  
cout<<"\n";  
student A(s);  
A.show_data();  
getch();  
return 0;
```

```
}
```

```
/*
```

Constructor:

Roll no:10

Name:Rahul

Copy constructor:

Roll no:10

Name:Rahul \*/

## 7.Parametrised Constructor

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
class read_constructor
```

```
{
```

```
int x;
```

```
public:
```

```
    read_constructor(int a)
```

```
    {
```

```
        x=a;
```

```
    }
```

```
    void read_data()
```

```
    {
```

```
        cin>>x;
```

```
    }
```

```
    void show_data()
```

```
    {
```

```
        cout<<"Value of x:"<<x;
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
    read_constructor obj(10);
```



```
obj.show_data();
```

```
getch();
```

```
return 0;
```

```
}
```

```
/*
```

```
Value of x:10
```

```
*/
```

## 8.Lower Matrix

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```
int main()
```

```
{
```

```
clrscr();
```

```
int x[4][4]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
```

```
int i,j;
```

```
// output phase
```

```
cout<<"Lower matrix:";
```

```
for(i=0;i<4;i++)
```

```
{
```

```
cout<<endl;
```

```
for(j=0;j<4;j++)
```

```
if(i>=j)
```

```
cout<<setw(6)<<x[i][j];
```

```
else
```

```
cout<<setw(6)<<" ";
```

```
}
```

```
getch();
```

```
return 0;
```

}

/\*

Lower matrix:

1

5 6

9 10 11

13 14 15 16

\*/

## 9.Sum of Matrices

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
int main()
```

```
{
```

```
clrscr();
```

```
int x[2][3]={10,9,8,8,3,4};
```

```
int i,j,sum;
```

```
// Processing
```

```
sum=0;
```

```
for(i=0;i<2;i++)
```

```
    for(j=0;j<3;j++)
```

```
        sum=sum +x[i][j];
```

```
// output
```

```
cout<<"\n Sum of elements of matrix :"<<sum;
```

```
cout<<"\n The matrix is:";
```

```
for(i=0;i<2;i++)
```

```
{
```

```
    cout<<endl;
```

```
        for(j=0;j<3;j++)
```

```
    cout<<" \t"<<x[i][j];  
}  
getch();  
return 0;  
}
```

```
/*
```

Sum of elements of matrix :42

The matrix is:

10	9	8
8	3	4

```
*/
```

## 10.Nesting

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
class area
```

```
{
```

```
int b, h;
```

```
double ar;
```

```
public:
```

```
void input_data()
```

```
{
```

```
cout<<"\n Enter base:";
```

```
cin>>b;
```

```
cout<<"\n Enter height:";
```

```
cin>>h;
```

```
}
```

```
void calculate()
```

```
{
```

```
input_data();
```

```
ar=0.5*b*h;
```

```
}
```

```
void output()
{
    calculate();
    cout<<"\n Area of triangle:"<<ar;
}
};
```

```
int main()
{
    clrscr();
    area A;
    A.output();
    getch();
    return 0;
}
```

```
/*
```

Enter base:10

Enter height:5

Area of triangle:25

```
*/
```

## 11.Pointers

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
int* read()
```

```
{
```

```
    int x;
```

```
    x=45;
```

```
    return(&x);
```

```
}
```

```
int main()
```

```
{
```

```
    clrscr();
```

```
    int *res;
```

```
    res=read();
```

```
    cout<<"\n Value at res:"<<res;
```

```
    getch();
```

```
    return 0;
```

```
}
```



```
/*  
Value at res:0x8fc7ffee  
*/
```

## 12.Linear Queue

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```
//function prototype
```

```
void add(int &front, int &rear, int x[], int N, int value);
```

```
void deletion(int &front, int &rear);
```

```
void display(int front, int rear, int x[]);
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int x[100],front,rear,choice,value;
```

```
front=-1;
```

```
rear=-1;
```

```
do
```

```
{
```

```
cout<<"\n Queue Menu";
```

```
cout<<"\n 1. Addition";
```

```
cout<<"\n 2. Deletion";
```

```
cout<<"\n 3. Display";

cout<<"\n 4. Exit";

cout<<"\n Enter your choice";

cin>>choice;

switch(choice)
{
    case 1:

        cout<<"\n Enter value";

        cin>>value;

        add(front,rear,x,100,value);

        break;

    case 2:

        deletion(front,rear);

        break;

    case 3:

        display(front,rear,x);

        getch();

        break;

    case 4:

        break;

    default:

        cout<<" wrong choice";

        getch();

}

}while(choice!=4);
```

```
getch();
```

```
return ;
```

```
}
```

```
void add(int &front, int &rear, int x[], int N, int value)
```

```
{
```

```
    if(rear==-1)
```

```
    {
```

```
        front=0;
```

```
        rear=0;
```

```
        x[rear]=value;
```

```
    }
```

```
    else
```

```
    {
```

```
        if(rear>=(N-1))
```

```
            cout<<"full";
```

```
        else
```

```
        {
```

```
            rear=rear+1;
```

```
            x[rear]=value;
```

```
        }
```

```
    }
```

```
}
```

```
void deletion (int &front, int &rear)
```

```
{  
if(front==-1)  
    cout<<"Empty";  
else  
{  
    if(front==rear)  
    {  
        front=-1;  
        rear=-1;  
    }  
    else  
        front=front+1;  
}  
return;  
}
```

```
void display (int front, int rear, int x[])  
{  
    int i;  
    if(front==-1)  
        cout<<"Empty";  
    else  
    {  
        for(i=front;i<=rear;i++)  
            cout<<setw(4)<<x[i];  
    }
```

```
}
```

```
return;
```

```
}
```

```
/*
```

3. Display

4. Exit

Enter your choice 1

Enter value 8

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 3

2 5 8

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 2

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 3

5 8

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 1

Enter value 9

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 3

5 8 9

Queue Menu

1. Addition

2. Deletion

3. Display

4. Exit

Enter your choice 4

\*/

### **13.Circular Queue**

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```
//Function prototype
```

```
void push(int x[], int N, int &M, int data);
```

```
void pop(int &m);
```

```
void display(int x[], int &m);
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int x[100],m,data,choice;
```

```
m=0;
```

```
do
```

```
{
```

```
cout<<"\n STACK MENU";
```

```
cout<<"\n 1. Push";
```

```
cout<<"\n 2. Pop";
```

```
cout<<"\n 3. Display";
```

```
cout<<"\n 4. Exit";

cout<<"\n Enter choice:";

cin>>choice;

    switch(choice)
    {
        case 1:
            cout<<" Enter value:";

            cin>>data;

            push(x,100,m,data);

            break;

        case 2:

            pop(m);

            break;

        case 3:

            display(x,m);

            getch();

            break;

        case 4:

            break;

        default:

            cout<<"\n Wrong choice";

            getch();

    }

}while(choice!=4);

}
```



```
void push(int x[], int N, int &M, int data)
```

```
{
```

```
if(M>=N)
```

```
cout<<"STACK full";
```

```
else
```

```
{
```

```
x[M]=data;
```

```
    M=M+1;
```

```
}
```

```
return;
```

```
}
```

```
void pop(int &m)
```

```
{
```

```
if(m<1)
```

```
cout<<"\n STACK Empty";
```

```
else
```

```
    m=m-1;
```

```
return;
```

```
}
```

```
void display(int x[],int &m)
```

```
{
```

```
int i;
```

```
if(m<1)
cout<<"\n STACK Empty";
else
for(i=0;i<m;i++)
cout<<setw(6)<<x[i];
return;
}
```

/\*

2. Pop

3. Display

4. Exit

Enter choice:1

Enter value:4

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:1

Enter value:6

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:1

Enter value:8

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:3

2 4 6 8

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:2

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:3

2 4 6

STACK MENU

1. Push

2. Pop

3. Display

4. Exit

Enter choice:4

\*/

## 14.Read Binary File

```
#include<fstream.h>
```

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
#include<string.h>
```

```
struct student{
```

```
int roll;
```

```
char name[30];
```

```
char address[80];
```

```
};
```

```
int main()
```

```
{
```

```
clrscr();
```

```
ifstream obj("student.dat");
```

```
student s;
```

```
cout<<"\n Reading student.dat file";
```

```
while(obj.read((char*)&s, sizeof(student)))
```

```
{
```

```
cout<<"\nRoll no:"<<s.roll;
```

```
cout<<"\n Name:"<<s.name;  
cout<<"\n Address:"<<s.address;  
}  
obj.close();  
getch();  
return 0;  
}
```

```
/*
```

Reading student.dat file

Roll no:1

Name:ashish

Address:delhi

```
*/
```

### **15. Write in Binary File**

```
#include<fstream.h>
```

```
#include<conio.h>
```

```
#include<iostream.h>
```

```
struct student
```

```
{
```

```
int roll;
```

```
char name[30];
```

```
char address[80];
```

```
};
```

```
int main()
```

```
{
```

```
clrscr();
```

```
ifstream fin("student.dat");
```

```
ofstream fout("temp.dat");
```

```
int troll;
```

```
student s;
```

```
cout<<"\n Enter roll no to modify";
```

```
cin>>troll;
```

```

while(fin.read((char*)&s, sizeof(student)))
{
    if(troll==s.roll)
    {
        cout<<"\n Enter New name:";
        cin>>s.name;
        cout<<"\n Enter Address:";
        cin>>s.address;
    }
    fout.write((char*)&s, sizeof(student));
}
fin.close();
fout.close();
//remove("student.dat");
//rename("temp.dat","student.dat");
getch();
return 0;
}

/*

```

Enter roll no to modify 1

Enter New name:AG



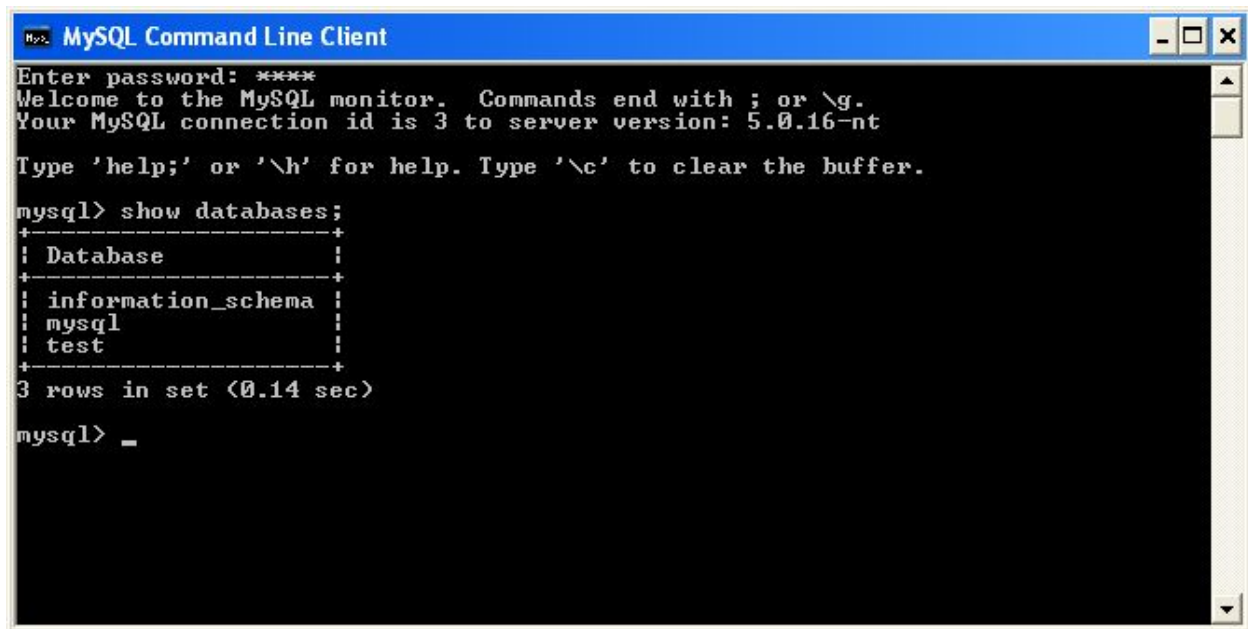
Enter Address:anand vihar

\*/

# SQL COMMANDS

Command 1:

Show databases;



```
MySQL Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 5.0.16-nt

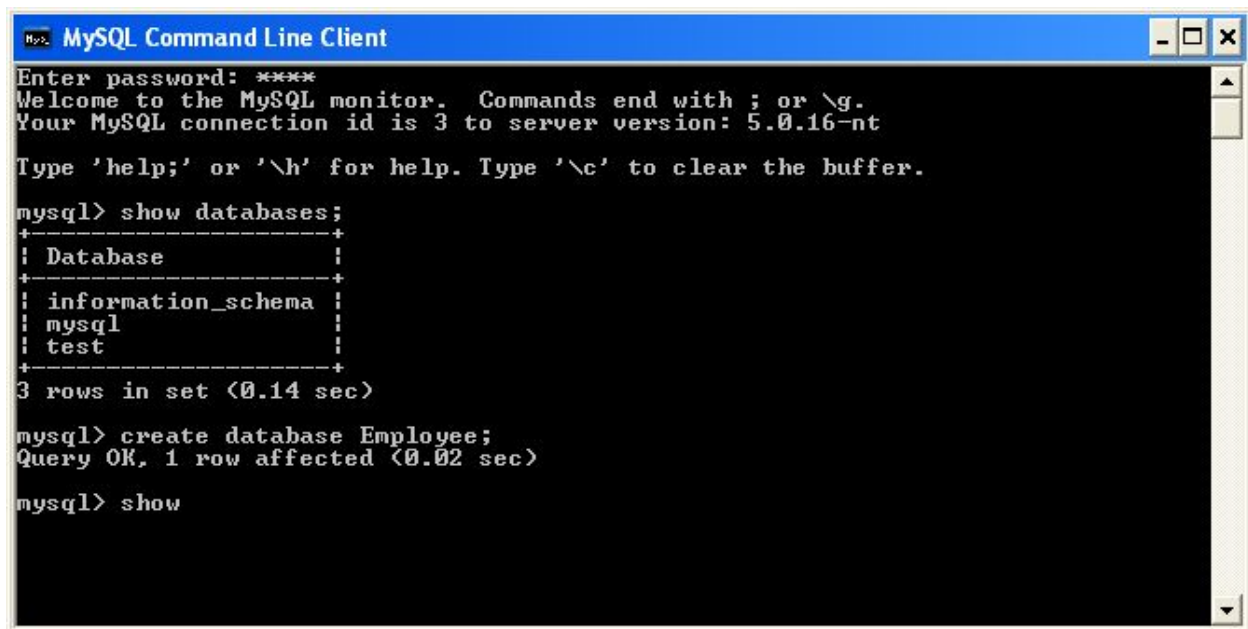
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.14 sec)

mysql> _
```

Command 2:

Create database;



```
MySQL Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 5.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.14 sec)

mysql> create database Employee;
Query OK, 1 row affected (0.02 sec)

mysql> show
```

```
MySQL Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| employee |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)

mysql> _
```

Command 3 & 4:

Create and desc table

```
MySQL Command Line Client

mysql> use employee;
Database changed
mysql> create table emplyee_data
-> ( emp_id int(5),
-> emp_name char(25),
-> emp_dept char(10),
-> age int(2),
-> salary int (10),
-> address char(20)
-> );
Query OK, 0 rows affected (0.22 sec)

mysql> desc emplyee_data;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(5) | YES | | NULL | |
| emp_name | char(25) | YES | | NULL | |
| emp_dept | char(10) | YES | | NULL | |
| age | int(2) | YES | | NULL | |
| salary | int(10) | YES | | NULL | |
| address | char(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.05 sec)
```

## Command 5 & 6:

Insert & select

```
MySQL Command Line Client

mysql> insert into employee_data values('10001','ASHISH','HR','21','25000','DELHI');
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee_data values('10002','AKANSHA','FINANCE','21','20000','DELHI');
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee_data values('10003','VISHAL','FINANCE','23','21000','GHAZIABAD');
Query OK, 1 row affected (0.05 sec)

mysql> insert into employee_data values('10004','MOHIT','RESEARCH','30','40000','NOIDA');
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee_data values('10005','RASHI','DEVELOPER','28','35000','NOIDA');
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee_data;
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10001 | ASHISH | HR | 21 | 25000 | DELHI |
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
| 10003 | VISHAL | FINANCE | 23 | 21000 | GHAZIABAD |
| 10004 | MOHIT | RESEARCH | 30 | 40000 | NOIDA |
| 10005 | RASHI | DEVELOPER | 28 | 35000 | NOIDA |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)

mysql>
```

## Command 7:

Alter table- add

```
MySQL Command Line Client

mysql> alter table employee_data add(bloodGroup char(4));
Query OK, 5 rows affected (0.22 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> desc employee_data;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id     | int(5)    | YES  |     | NULL    |       |
| emp_name   | char(25)  | YES  |     | NULL    |       |
| emp_dept   | char(10)  | YES  |     | NULL    |       |
| age        | int(2)    | YES  |     | NULL    |       |
| salary     | int(10)   | YES  |     | NULL    |       |
| address    | char(20)  | YES  |     | NULL    |       |
| bloodGroup | char(4)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.02 sec)

mysql>
```

### Command 8:

Alter table drop

```
MySQL Command Line Client

mysql> alter table employee_data drop bloodGroup;
Query OK, 5 rows affected (0.17 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> desc employee_data;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id     | int(5)    | YES  |     | NULL    |       |
| emp_name   | char(25)  | YES  |     | NULL    |       |
| emp_dept   | char(10)  | YES  |     | NULL    |       |
| age        | int(2)    | YES  |     | NULL    |       |
| salary     | int(10)   | YES  |     | NULL    |       |
| address    | char(20)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

### Command 9:

Select-Like

```
MySQL Command Line Client
mysql> select * from employee_data;
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10001 | ASHISH | HR | 21 | 25000 | DELHI |
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
| 10003 | VISHAL | FINANCE | 23 | 21000 | GHAZIABAD |
| 10004 | MOHIT | RESEARCH | 30 | 40000 | NOIDA |
| 10005 | RASHI | DEVELOPER | 28 | 35000 | NOIDA |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from employee_data
-> where emp_name like 'A%';
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10001 | ASHISH | HR | 21 | 25000 | DELHI |
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from employee_data
-> where emp_dept like 'F_____';
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
| 10003 | VISHAL | FINANCE | 23 | 21000 | GHAZIABAD |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Command 10:**

Select- order by

```
MySQL Command Line Client

mysql> select emp_name from employee_data
-> order by emp_name;
+-----+
| emp_name |
+-----+
| AKANSHA  |
| ASHISH   |
| MOHIT    |
| RASHI    |
| VISHAL   |
+-----+
5 rows in set (0.03 sec)

mysql> select * from employee_data
-> order by emp_name desc;
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10003 | VISHAL   | FINANCE  | 23  | 21000  | GHAZIABAD |
| 10005 | RASHI    | DEVELOPER | 28  | 35000  | NOIDA     |
| 10004 | MOHIT    | RESEARCH | 30  | 40000  | NOIDA     |
| 10001 | ASHISH   | HR       | 21  | 25000  | DELHI     |
| 10002 | AKANSHA  | FINANCE  | 21  | 20000  | DELHI     |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Command 11:

Select-distinct

```
MySQL Command Line Client

mysql> select distinct(salary) from employee_data;
+-----+
| salary |
+-----+
| 25000  |
| 20000  |
| 21000  |
| 40000  |
| 35000  |
+-----+
5 rows in set (0.00 sec)

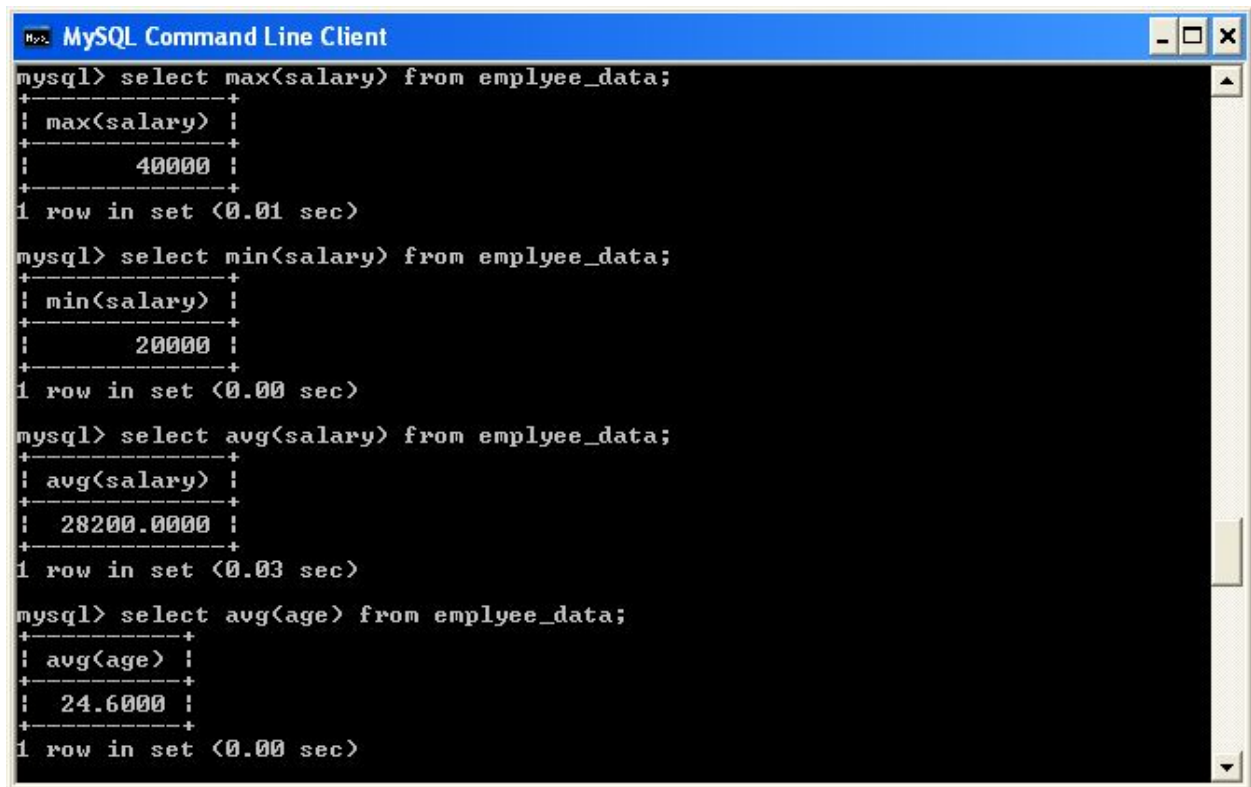
mysql> select distinct(address) from employee_data;
+-----+
| address |
+-----+
| DELHI   |
| GHAZIABAD |
| NOIDA   |
+-----+
3 rows in set (0.00 sec)

mysql>
```



### Command 12:

Grouping functions – max, min, avg



```
mysql> select max(salary) from employee_data;
+-----+
| max(salary) |
+-----+
|      40000 |
+-----+
1 row in set (0.01 sec)

mysql> select min(salary) from employee_data;
+-----+
| min(salary) |
+-----+
|      20000 |
+-----+
1 row in set (0.00 sec)

mysql> select avg(salary) from employee_data;
+-----+
| avg(salary) |
+-----+
|  28200.0000 |
+-----+
1 row in set (0.03 sec)

mysql> select avg(age) from employee_data;
+-----+
| avg(age) |
+-----+
|  24.6000 |
+-----+
1 row in set (0.00 sec)
```

### Command 13:

Select – grouping functions- sum, count



```
MySQL Command Line Client

mysql> select sum(salary) from employee_data;
+-----+
| sum(salary) |
+-----+
|      141000 |
+-----+
1 row in set (0.00 sec)

mysql> select count(emp_id) from employee_data;
+-----+
| count(emp_id) |
+-----+
|              5 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Command 14:

Select – group by

```
MySQL Command Line Client

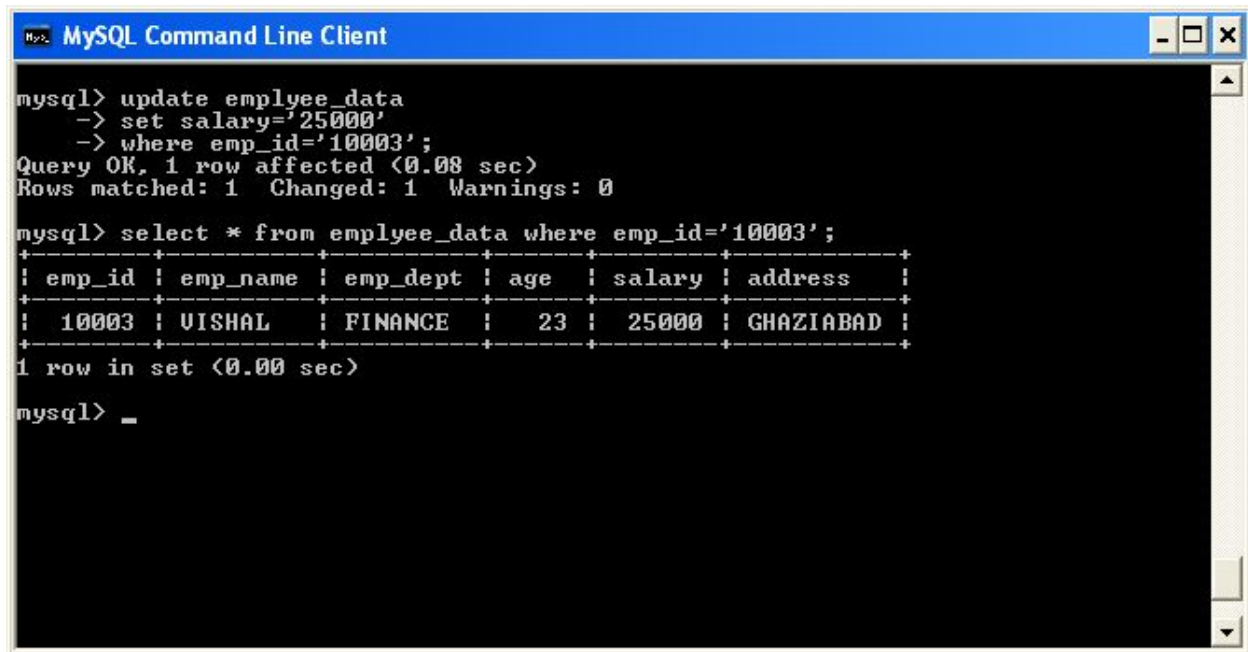
mysql> select * from employee_data
-> group by emp_dept;
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10005 | RASHI | DEVELOPER | 28 | 35000 | NOIDA |
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
| 10001 | ASHISH | HR | 21 | 25000 | DELHI |
| 10004 | MOHIT | RESEARCH | 30 | 40000 | NOIDA |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from employee_data
-> group by emp_name desc;
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10003 | VISHAL | FINANCE | 23 | 21000 | GHAZIABAD |
| 10005 | RASHI | DEVELOPER | 28 | 35000 | NOIDA |
| 10004 | MOHIT | RESEARCH | 30 | 40000 | NOIDA |
| 10001 | ASHISH | HR | 21 | 25000 | DELHI |
| 10002 | AKANSHA | FINANCE | 21 | 20000 | DELHI |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Command 15:

## Update



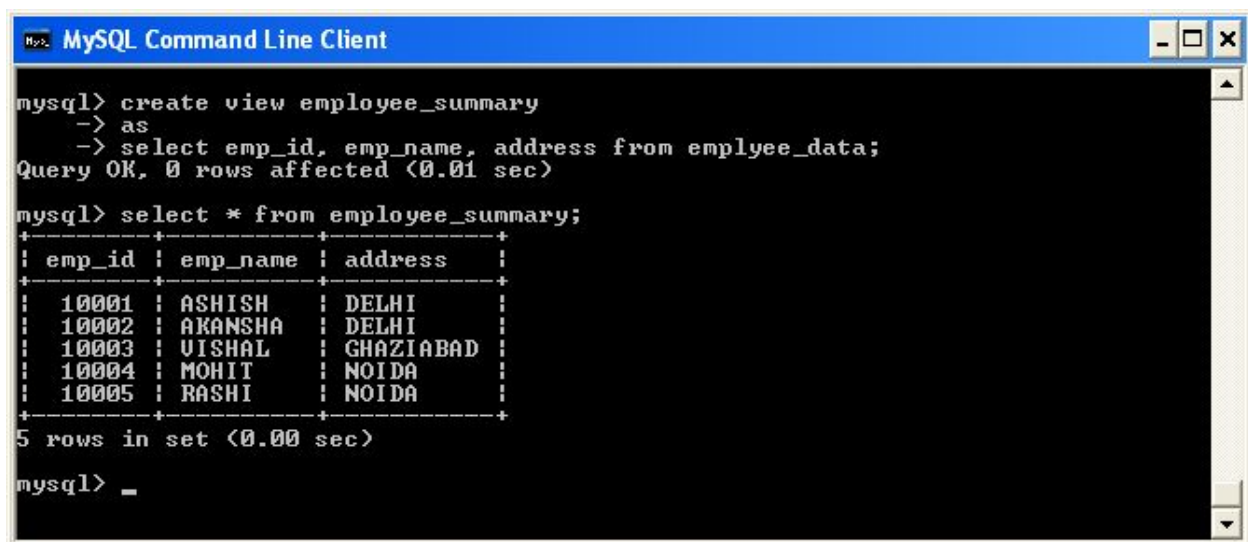
```
mysql> update employee_data
-> set salary='25000'
-> where emp_id='10003';
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee_data where emp_id='10003';
+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | age | salary | address |
+-----+-----+-----+-----+-----+-----+
| 10003 | VISHAL | FINANCE | 23 | 25000 | GHAZIABAD |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

## Command 16:

### Create view



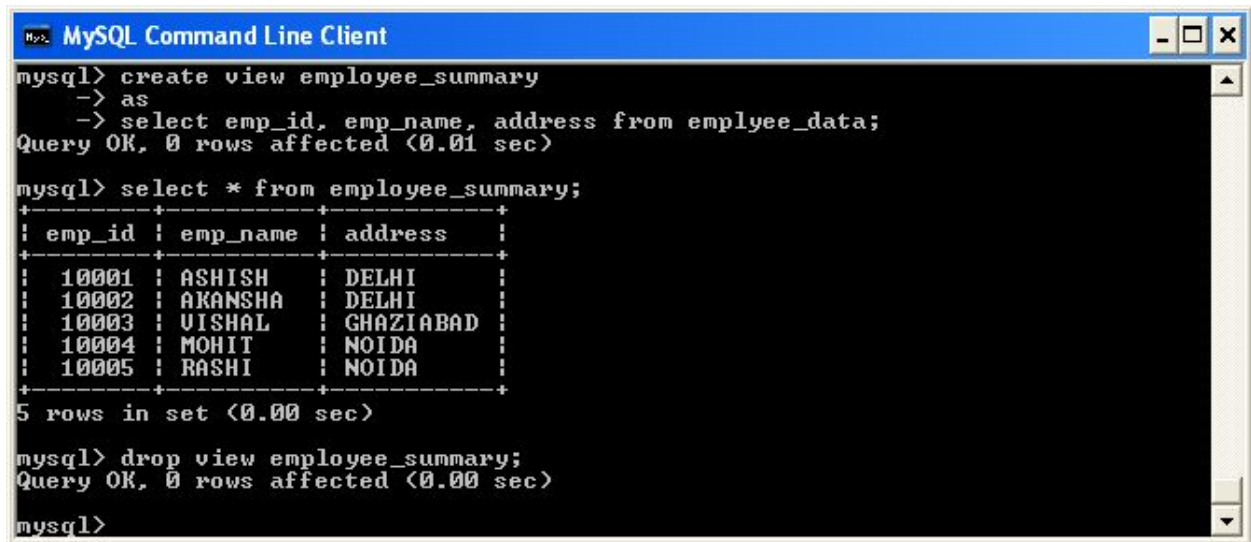
```
mysql> create view employee_summary
-> as
-> select emp_id, emp_name, address from employee_data;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from employee_summary;
+-----+-----+-----+
| emp_id | emp_name | address |
+-----+-----+-----+
| 10001 | ASHISH | DELHI |
| 10002 | AKANSHA | DELHI |
| 10003 | VISHAL | GHAZIABAD |
| 10004 | MOHIT | NOIDA |
| 10005 | RASHI | NOIDA |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

## Command 17:

Drop view



```
mysql> create view employee_summary
-> as
-> select emp_id, emp_name, address from employee_data;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from employee_summary;
+-----+-----+-----+
| emp_id | emp_name | address |
+-----+-----+-----+
| 10001 | ASHISH | DELHI |
| 10002 | AKANSHA | DELHI |
| 10003 | VISHAL | GHAZIABAD |
| 10004 | MOHIT | NOIDA |
| 10005 | RASHI | NOIDA |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> drop view employee_summary;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

# INDEX

- **PROGRAMS:**

- Binary Search
- Linear Search
- Bubble Sorting
- Selection Sorting
- Insertion Sorting
- Copy Constructor
- Paramterised Constructor
- Lower Triangular Matrix
- Sum of two Matrices
- To calculate area of a triangle using class
- Pointer
- Linear Queue
- Circular Queue
- To read content from a Binary File
- To write content in the Binary File

- **SQL COMMANDS:**

- Show database
- Create database
- To insert data and use of SELECT command
- Creating and printing the table in descending order
- Alter the table with ADD command
- Alter the table with DROP command
- LIKE command
- SELECT-ORDER BY command
- SELECT-DISTINCT command
- Grouping function – MAX,MIN,AVG
- SELECT-Grouping function- SUM,COUNT
- SELECT-GROUP BY command
- UPDATE command
- CREATE VIEW command
- DROP VIEW command