



**GEMS**  
United Indian School  
ABU DHABI

# PRACTICAL RECORD

## 2022-2023

Roll No. \_\_\_\_\_

Name: \_\_\_\_\_

Class: \_\_\_\_\_ Sec: \_\_\_\_\_

Subject: \_\_\_\_\_



## *Certificate*

**Roll No.** \_\_\_\_\_

This is to certify that  
Miss/Master \_\_\_\_\_ of grade  
\_\_\_\_\_ section \_\_\_\_\_

has carried out practical work in Assignment  
prescribed by the Central Board of Secondary  
Education, New Delhi during the academic year  
2022-2023.

Teacher-in charge: Dr.Harini Priyadharsini

Date: \_\_\_\_\_

\_\_\_\_\_  
External Examiner

\_\_\_\_\_  
Internal Examiner

# INDEX

- **PYTHON PROGRAMS**

1. Factorial of a number
2. Checking prime number
3. Search any word in given string
4. Program to Bubble Sort using a user-defined function.
5. Display file content line by line
6. Display the total number of consonants, uppercase, vowels and lower-case characters in a text file
7. Create a binary file and display contents
8. Create a binary file and update contents
9. Copy a file except that lines start with 'a'
10. Create and display a CSV file
11. Simulate a dice
12. Implement Stack in Python using list
13. Program to implement Binary Search/Linear Search in Python
14. Count common words in phishing email
15. Store and display data using MySQL connectivity
16. Search data using MySQL connectivity
17. Update data using MySQL connectivity
18. Delete data using MySQL connectivity
19. Create a CSV file with given details
20. Copy all contents of a CSV file

- **mySQL**

# PYTHON PROGRAMS

---

## *Experiment No: 1*

---

**Aim: Input any number from user and calculate factorial of a number**

**CODE:**

```
num = int(input("Enter any number :"))
f = 1
n = num
while num>1:
    f = f * num
    num-=1
print("Factorial of ", n , " is :",f)
```

**OUTPUT:**

```
Enter any number :5
Factorial of 5 is : 120
```

---

## Experiment No: 2

---

### AIM: Program to input any number from user

# Check it is Prime number or not

#### **CODE**

```
num = int(input("Enter the number: "))
if num > 1:
    for i in range(2, int(num/2)+1):
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

#### **OUTPUT**

Enter the number: 17  
17 is a prime number

Enter the number: 38  
38 is not a prime number

---

## Experiment No: 3

---

**Aim : Program to search any word in given string/sentence**

**CODE:**

```
def countWord(sent,word):
    s = sent.split()
    count=0
    for w in s:
        if w==word:
            count+=1
    return count
sent = input("Enter any sentence :")
word = input("Enter word to search in sentence :")
count = countWord(sent,word)
if count==0:
    print(word," not found")
else:
    print(word," occurs ",count," times")
```

**OUTPUT**

Enter any sentence :my computer your computer our computer everyones computer  
Enter word to search in sentence :computer  
computer occurs 4 times

Enter any sentence :Learning python is so much fun  
Enter word to search in sentence :c++  
c++ not found

---

## *Experiment No: 4*

---

**Aim : Program to Bubble Sort using a user-defined function.**

**CODE:**

```
def BubbleSort():
    L = [eval(i) for i in input("Enter the list items : ").split()]
    for i in range(len(L)):
        for j in range(len(L)-1):
            if L[j] > L[j+1]:
                L[j], L[j+1] = L[j+1], L[j]
    return L

print(BubbleSort())
```

**OUTPUT:**

Enter the list items : 9 3 10 232 910 1080 69 420  
[3, 9, 10, 69, 232, 420, 910, 1080]

Enter the list items : 'Albin' 'Kavan' 'Raed' 'Aryan' 'Sebastian'  
['Albin', 'Aryan', 'Kavan', 'Raed', 'Sebastian']



---

## *Experiment No: 5*

---

**Aim : Program to read and display file content line by line with each word separated by “#”**

**CODE:**

```
f=open('txt1.txt')
item=[]
for line in f:
    words=line.split()
    for i in words:
        print(i+'#', end='')
    print()
```

**FILE CONTENT:**

India is my country  
I love India  
I'm learning python  
Learning python is fun

**OUTPUT:**

India#is#my#country#  
I#love#India#  
I'm#learning#python#  
Learning#python#is#fun#

---

## Experiment No: 6

---

**Aim : Program to read the content of file and display the total number of consonants, uppercase, vowels and lower case characters.**

**CODE:**

```
f = open("txt2.txt")
v=0
c=0
u=0
l=0
o=0
data = f.read()
vowels=['a','e','i','o','u']

for ch in data:
    if ch.isalpha():
        if ch.lower() in vowels:
            v+=1
        else:
            c+=1
    if ch.isupper():
        u+=1
    elif ch.islower():
        l+=1
    elif ch!=' ' and ch!='\n':
        o+=1
print("Total Vowels in file      :",v)
print("Total Consonants in file n    :",c)
print("Total Capital letters in file  :",u)
print("Total Small letters in file    :",l)
print("Total Other than letters      :",o)
f.close()
```

### **CONTENT IN FILE**

India is my country  
I love India  
I'm learning python  
Learning python is fun

### **OUTPUT:**

Total Vowels in file : 23  
Total Consonants in file n : 38  
Total Capital letters in file : 5  
Total Small letters in file : 56  
Total Other than letters : 1

---

## *Experiment No: 7*

---

**Aim : Program to create binary file to store Rollno and Name, Search any Rollno and display name if Rollno found otherwise “Rollno not found”**

**CODE:**

```
import pickle
student=[]
f=open('FILE1.dat','wb')
print("-----ENTER DETAILS-----")
while True:
    roll = int(input("Enter Roll Number :"))
    name = input("Enter Name :")
    student.append([roll,name])
    ans=input("Add More ?(Y)")
    if ans.lower()!='y':
        break
pickle.dump(student,f)
f.close()

print("\n-----SEARCH DETAILS-----")
f=open('FILE1.dat','rb')
student=[]
while True:
    try:
        student = pickle.load(f)
    except EOFError:
        break

while True:
    found=False
    r = int(input("Enter Roll number to search :"))
    for s in student:
        if s[0]==r:
            print("Name is :",s[1])
            found=True
            break
```

```
if not found:
    print("Roll number not found")
ans=input("Wish to continue? (Y): ")
if ans.lower()!='y':
    break
f.close()
print("\nThank you!!!")
```

### **OUTPUT:**

-----ENTER DETAILS-----

Enter Roll Number :101

Enter Name :Albin

Add More ?(Y)y

Enter Roll Number :102

Enter Name :Kavan

Add More ?(Y)y

Enter Roll Number :103

Enter Name :Aryan

Add More ?(Y)y

Enter Roll Number :104

Enter Name :Raed

Add More ?(Y)y

Enter Roll Number :105

Enter Name :Sebastian

Add More ?(Y)n

-----SEARCH DETAILS-----

Enter Roll number to search :104

Name is : Raed

Wish to continue? (Y): y

Enter Roll number to search :107

Roll number not found

Wish to continue? (Y): n

Thank You!!!

---

## Experiment No: 8

---

**Aim : Program to create binary file to store Rollno,Name and Marks and update marks of entered Rollno**

**CODE:**

```
import pickle
student=[]
f=open('FILE2.dat','wb')
print("----- Enter Details -----")
while True:
    roll = int(input("Enter Roll Number :"))
    name = input("Enter Name :")
    marks = int(input("Enter Marks :"))
    student.append([roll,name,marks])
    ans=input("Add More ? (Y)")
    if ans.lower()!='y':
        break
pickle.dump(student,f)
f.close()
f=open('FILE2.dat','rb+')
student=[]
while True:
    try:
        student = pickle.load(f)
    except EOFError:
        break
print("\n----- Update Details -----")
while True:
    found=False
    r = int(input("Enter Roll number to update :"))
    for s in student:
        if s[0]==r:
            print("Name: ",s[1])
            print("Current Marks: ",s[2])
            m = int(input("Enter new marks: "))
            s[2]=m
            print("Record Updated")
            found=True
            break
```

```
if not found:
    print("Roll number not found")
ans=input("Wish to continue? (Y): ")
if ans.lower()!='y':
    break
f.close()
```

### **OUTPUT:**

----- Enter Details -----

Enter Roll Number :101

Enter Name :albin

Enter Marks :90

Add More ? (Y)y

Enter Roll Number :102

Enter Name :kavan

Enter Marks :85

Add More ? (Y)y

Enter Roll Number :103

Enter Name :raed

Enter Marks :80

Add More ? (Y)y

Enter Roll Number :104

Enter Name :aryan

Enter Marks :70

Add More ? (Y)y

Enter Roll Number :105

Enter Name :sebastian

Enter Marks :95

Add More ? (Y)n

----- Update Details -----

Enter Roll number to update :104

Name: aryan

Current Marks: 70

Enter new marks: 79

Record Updated

Wish to continue? (Y): y

Enter Roll number to update :106

Roll number not found

Wish to continue? (Y): n

---

## *Experiment No: 9*

---

**Aim : Program to read the content of file line by line and write it to another file except for the lines contains “a” letter in it.**

### **CODE:**

```
f1 = open("txt3.txt")
f2 = open("txt3copy.txt","w")

for line in f1:
    if 'a' not in line:
        f2.write(line)
print("File Copied Successfully!")
f1.close()
f2.close()
```

### **FILE CONTENT**

a quick brown fox  
one two three four  
five six seven  
India is my country eight nine ten  
bye!

### **OUTPUT:**

File Copied Successfully!

### **FILE CONTENT**

one two three four  
five six seven  
bye!



---

## Experiment No: 10

---

**Aim : Program to create CSV file and store empno, name, salary and search any empno and display name, salary and if not found appropriate message.**

**CODE:**

```
import csv
with open('csv1.csv',mode='a') as csvfile:
    mywriter = csv.writer(csvfile,delimiter=',')
    print("-----Enter Details-----")
    while True:
        eno=int(input("Enter Employee Number "))
        name=input("Enter Employee Name ")
        salary=int(input("Enter Employee Salary :"))
        mywriter.writerow([eno,name,salary])
        print(" Data Saved...")
        ans=input("\nAdd More: ")
        if ans.lower()!='y':
            break

with open('csv1.csv',mode='r') as csvfile:
    myreader = csv.reader(csvfile,delimiter=',')
    print("----- Search Details -----")
    while True:
        found=False
        e = int(input("\nEnter Employee Number to search :"))
        for row in myreader:
            if len(row)!=0:
                if int(row[0])==e:
                    print("=====")
                    print("NAME      :",row[1])
                    print("SALARY :",row[2])
                    found=True
                    break

        if not found:
            print("=====\nEMPNO NOT
FOUND\n=====")
            ans = input("Search More ? (Y)")
            if ans.lower()!='y':
                break
```

## **OUTPUT:**

-----Enter Details-----

Enter Employee Number 101

Enter Employee Name aaa

Enter Employee Salary :1000

Data Saved...

Add More: y

Enter Employee Number 102

Enter Employee Name bbb

Enter Employee Salary :2000

Data Saved...

Add More: y

Enter Employee Number 103

Enter Employee Name ccc

Enter Employee Salary :2000

Data Saved...

Add More: n

----- Search Details -----

Enter Employee Number to search :101

=====

NAME : aaa

SALARY : 1000

Enter Employee Number to search :103

=====

NAME : ccc

SALARY : 2000

Enter Employee Number to search :104

=====

EMPNO NOT FOUND

=====

Search More ? (Y)n

---

## *Experiment No: 11*

---

**Aim : Program to generate random number 1-6, simulating a dice**

**CODE:**

```
import random
import time
print("Press CTRL+C to stop the dice ")
while True:
    try:
        while True:
            for i in range(10):
                print()
                n = random.randint(1,6)
                print(n,end="")
                time.sleep(.00001)
            except KeyboardInterrupt:
                print("Your Number is :",n)
                ans=input("Play More? (Y) :")
                if ans.lower()!='y':
                    break
```

**OUTPUT:**

```
4
Your Number is : 4
Play More? (Y) :y

1Your Number is : 1
Play More? (Y) :y

6Your Number is : 6
Play More? (Y) :n
```

---

## Experiment No: 12

---

**Aim : Program to implement Stack in Python using List**

**CODE:**

```
def isEmpty(S):
    if len(S)==0:
        return True
    else:
        return False

def Push(S,item):
    S.append(item)
    top=len(S)-1

def Pop(S):
    if isEmpty(S):
        return "Underflow"
    else:
        val = S.pop()
        if len(S)==0:
            top=None
        else:
            top=len(S)-1
        return val

def Peek(S):
    if isEmpty(S):
        return "Underflow"
    else:
        top=len(S)-1
        return S[top]

def Show(S):
    if isEmpty(S):
        print("Sorry No items in Stack ")
    else:
        t = len(S)-1
        print("Stack (Top)")
        while(t>=0):
            print(S[t])
```

```
t=1
print()
```

```
S=[]
top=None
print("""***** STACK DEMONSTRATION *****
1. PUSH
2. POP
3. PEEK
4. SHOW STACK
0. EXIT""")
```

```
while True:
    ch = int(input("\nEnter your choice :"))
    if ch==1:
        val = int(input("Enter Item to Push :"))
        Push(S,val)
    elif ch==2:
        val = Pop(S)
        if val=="Underflow":
            print("Stack is Empty")
        else:
            print("\nDeleted Item was :",val)
    elif ch==3:
        val = Peek(S)
        if val=="Underflow":
            print("Stack Empty")
        else:
            print("Top item:",val)
    elif ch==4:
        Show(S)
    elif ch==0:
        print("Bye")
        break
```

### **OUTPUT:**

```
**** STACK DEMONSTRATION ****
1. PUSH
2. POP
3. PEEK
4. SHOW STACK
0. EXIT
```

Enter your choice :1  
Enter Item to Push :23

Enter your choice :1  
Enter Item to Push :434

Enter your choice :1  
Enter Item to Push :43

Enter your choice :1  
Enter Item to Push :76

Enter your choice :1  
Enter Item to Push :989

Enter your choice :1  
Enter Item to Push :20

Enter your choice :4  
Stack (Top)  
20  
989  
76  
43  
434  
23

Enter your choice :2

Deleted Item was : 20

Enter your choice :4  
Stack (Top)  
989  
76  
43  
434  
23

Enter your choice :3  
Top item: 989

Enter your choice :0  
Bye

---

## Experiment No: 13

---

### Aim : Program to implement Binary Search/Linear Search

#### **CODE:**

```
def BinarySearch():
    L = [eval(i) for i in input("Enter the list items : ").split()]
    c = eval(input("Enter element to search : "))
    L.sort()
    low = found = 0
    high = len(L) - 1
    while low <= high:
        mid = (low + high) // 2
        if L[mid] == c:
            found = 1
            break
        elif L[mid] > c:
            high = mid - 1
        else:
            low = mid + 1
    if found == 1:
        print("Element found")
    else:
        print("Element not found")
```

LinearSearch()

BinarySearch()

#### **OUTPUT:**

Enter the list items : 1 3 4 2 5 10 20 32 19

Enter element to search : 32

Element found at index 7

Enter the list items : 1 3 4 2 5 34 29 30 99 72

Enter element to search : 98

Element not found

---

## Experiment No: 14

---

**Aim : Program to take 10 sample phishing email, and find the most common word occurring**

**CODE:**

```
fakemails=[
    "jackpotwin@lottery.com",
    "claimtheprize@mylife.com",
    "youarethewinner@lottery.com",
    "luckywinner@mylife.com",
    "spinthewheel@flipkart.com",
    "dealwinner@snapdeal.com"
    "luckywinner@snapdeal.com"
    "luckyjackpot@americanlottery.com"
    "claimtheprize@looto lottery.com"
    "youarelucky@mylife.com"
]
myd={}
for e in fakemails:
    x=e.split('@')
    for w in x:
        if w not in myd:
            myd[w]=1
        else:
            myd[w]+=1
key_max = max(myd,key=myd.get)
print("Most Common Occuring word :",key_max)
```

**OUTPUT:**

Most Common Occuring word : mylife.com



---

## Experiment No: 15

---

**Aim : Program to connect with database and store record of employee and display records.**

**CODE:**

```
import mysql.connector as mycon
con = mycon.connect(host='localhost',user='root',password="root")
cur = con.cursor()
cur.execute("create database if not exists company")
cur.execute("use company")
cur.execute("create table if not exists employee(empno int, name varchar(20), dept
varchar(20),
salary
int)")
con.commit()

while True:
    print(""" \n~~~ MENU ~~~
1. ADD RECORD
2. DISPLAY RECORD
0. EXIT""")
    choice = int(input("Enter Choice :"))

    if choice == 1:
        e = int(input("Enter Employee Number :"))
        n = input("Enter Name :")
        d = input("Enter Department :")
        s = int(input("Enter Salary :"))
        query="insert into employee values({},'{}','{}',{})".format(e,n,d,s)
        cur.execute(query)
        con.commit()
        print("Details Saved")

    elif choice == 2:
        query="select * from employee"
        cur.execute(query)
        result = cur.fetchall()
```

```

        print("%10s"%EMPNO,"%20s"%NAME,"%15s"%DEPARTMENT",
"%10s"%SALARY")
    for row in result:
        print("%10s"%row[0],"%20s"%row[1],"%15s"%row[2],"%10s"%row[3])

elif choice==0:
    con.close()
    print("Thank you")
    break
else:
    print("INCORRECT OPTION!!")

```

### **OUTPUT:**

~~~ MENU ~~~

1. ADD RECORD
2. DISPLAY RECORD
0. EXIT

Enter Choice :1

Enter Employee Number :101

Enter Name :Albin

Enter Department :IT

Enter Salary :3000

Details Saved

~~~ MENU ~~~

1. ADD RECORD
2. DISPLAY RECORD
0. EXIT

Enter Choice :1

Enter Employee Number :102

Enter Name :Raed

Enter Department :Sales

Enter Salary :2900

Details Saved

~~~ MENU ~~~

- 1. ADD RECORD
- 2. DISPLAY RECORD
- 0. EXIT

Enter Choice :1

Enter Employee Number :103

Enter Name :Kavan

Enter Department :Advertisment

Enter Salary :2950

Details Saved

~~~ MENU ~~~

- 1. ADD RECORD
- 2. DISPLAY RECORD
- 0. EXIT

Enter Choice :2

| EMPNO | NAME  | DEPARTMENT   | SALARY |
|-------|-------|--------------|--------|
| 101   | Albin | IT           | 3000   |
| 102   | Raed  | Sales        | 2900   |
| 103   | Kavan | Advertisment | 2950   |

~~~ MENU ~~~

- 1. ADD RECORD
- 2. DISPLAY RECORD
- 0. EXIT

Enter Choice :19

INCORRECT OPTION!!

~~~ MENU ~~~

- 1. ADD RECORD
- 2. DISPLAY RECORD
- 0. EXIT

Enter Choice :0

Thank you

---

## Experiment No: 16

---

**Aim:** Program to connect with database and search employee number in table employee and display record, if empno not found display appropriate message.

### **CODE:**

```
import mysql.connector as mycon
con = mycon.connect(host='localhost',user='root',password="root", database="company")
cur = con.cursor()
print("-----")
print("EMPLOYEE SEARCHING FORM")
print("-----")

while True:
    eno = int(input("\nENTER EMPNO TO SEARCH :"))
    query="select * from employee where empno={}".format(eno)
    cur.execute(query)
    result = cur.fetchall()
    if cur.rowcount==0:
        print("Sorry! Empno not found ")
    else:
        print("%10s"% "EMPNO", "%20s"% "NAME", "%15s"% "DEPARTMENT",
"%10s"% "SALARY")
        for row in result:
            print("%10s"%row[0], "%20s"%row[1], "%15s"%row[2], "%10s"%row[3])
        ans=input("SEARCH MORE (Y) :")
        if ans.lower()!='y':
            break
    print("Thank you!!")
```

### **TABLE CONTENTS:** (SAME FOR EXPERIMENT 17 &18)

| EMPNO | NAME | DEPARTMENT | SALARY |
|-------|------|------------|--------|
|-------|------|------------|--------|

|     |       |               |      |
|-----|-------|---------------|------|
| 101 | Albin | IT            | 3000 |
| 102 | Raed  | Sales         | 2900 |
| 103 | Kavan | Advertisement | 2950 |

### OUTPUT:

#### EMPLOYEE SEARCHING FORM

ENTER EMPNO TO SEARCH :101

| EMPNO | NAME  | DEPARTMENT | SALARY |
|-------|-------|------------|--------|
| 101   | Albin | IT         | 3000   |

SEARCH MORE (Y) :y

ENTER EMPNO TO SEARCH :105

Sorry! Empno not found

ENTER EMPNO TO SEARCH :104

Sorry! Empno not found

ENTER EMPNO TO SEARCH :102

| EMPNO | NAME | DEPARTMENT | SALARY |
|-------|------|------------|--------|
| 102   | Raed | Sales      | 2900   |

SEARCH MORE (Y) :y

ENTER EMPNO TO SEARCH :103

| EMPNO | NAME  | DEPARTMENT    | SALARY |
|-------|-------|---------------|--------|
| 103   | Kavan | Advertisement | 2950   |

SEARCH MORE (Y) :n

Thank you!!

---

## Experiment No: 17

---

**Aim: Program to connect with database and update the employee record of entered empno**

**CODE:**

```
import mysql.connector as mycon
con = mycon.connect(host='localhost',user='root',password="root",database="company")
cur = con.cursor()
print("-----\nEMPLOYEE UPDATION FORM\n-----")

while True:
    eno = int(input("\nENTER EMPNO TO UPDATE :"))
    query="select * from employee where empno={}".format(eno)
    cur.execute(query)
    result = cur.fetchall()
    if cur.rowcount==0:
        print("Sorry! Empno not found ")
    else:
        print("%10s"% "EMPNO", "%20s"% "NAME", "%15s"% "DEPARTMENT",
"%10s"% "SALARY")
        for row in result:
            print("%10s"%row[0], "%20s"%row[1], "%15s"%row[2], "%10s"%row[3])
            choice=input("\n--- ARE YOUR SURE TO UPDATE ? (Y) :")
            if choice.lower()=='y':
                print("== YOU CAN UPDATE ONLY DEPT AND SALARY ==")
                print("== FOR EMPNO AND NAME CONTACT ADMIN ==")
                d = input("ENTER NEW DEPARTMENT,(LEAVE BLANK IF NOT WANT TO CHANGE)")
                if d=="":
                    d=row[2]
                try:
                    s = int(input("ENTER NEW SALARY,(LEAVE BLANK IF NOT WANT TO CHANGE)")
                except:
                    s=row[3]
            else:
                continue
        query="update employee set dept='{',salary={} where empno={}".format(d,s,eno)
        cur.execute(query)
        con.commit()
        print("***RECORD UPDATED****")
        ans=input("UPDATE MORE (Y) : ")
```

```
if ans.lower()!='y':
    break
print('Thank you!!!')
```

### **OUTPUT:**

#### ----- EMPLOYEE UPDATION FORM -----

ENTER EMPNO TO UPDATE :101

| EMPNO | NAME  | DEPARTMENT | SALARY |
|-------|-------|------------|--------|
| 101   | Albin | IT         | 3000   |

---- ARE YOUR SURE TO UPDATE ? (Y) :y

== YOU CAN UPDATE ONLY DEPT AND SALARY ==

== FOR EMPNO AND NAME CONTACT ADMIN ==

ENTER NEW DEPARTMENT,(LEAVE BLANK IF NOT WANT TO CHANGE )Data Analyst

ENTER NEW SALARY,(LEAVE BLANK IF NOT WANT TO CHANGE)3100

\*\*\*RECORD UPDATED\*\*\*

UPDATE MORE (Y) : y

ENTER EMPNO TO UPDATE :102

| EMPNO | NAME | DEPARTMENT | SALARY |
|-------|------|------------|--------|
| 102   | Raed | Sales      | 2900   |

---- ARE YOUR SURE TO UPDATE ? (Y) :y

== YOU CAN UPDATE ONLY DEPT AND SALARY ==

== FOR EMPNO AND NAME CONTACT ADMIN ==

ENTER NEW DEPARTMENT,(LEAVE BLANK IF NOT WANT TO CHANGE )Marketing

ENTER NEW SALARY,(LEAVE BLANK IF NOT WANT TO CHANGE

\*\*\*RECORD UPDATED\*\*\*

UPDATE MORE (Y) : y

ENTER EMPNO TO UPDATE :103

| EMPNO | NAME  | DEPARTMENT | SALARY |
|-------|-------|------------|--------|
| 103   | Kavan | Sales      | 2900   |

---- ARE YOUR SURE TO UPDATE ? (Y) :n

ENTER EMPNO TO UPDATE :104

Sorry! Empno not found

ENTER EMPNO TO UPDATE :103

| EMPNO | NAME  | DEPARTMENT | SALARY |
|-------|-------|------------|--------|
| 103   | Kavan | Sales      | 2900   |

---- ARE YOUR SURE TO UPDATE ? (Y) :y

== YOU CAN UPDATE ONLY DEPT AND SALARY ==

== FOR EMPNO AND NAME CONTACT ADMIN ==

ENTER NEW DEPARTMENT,(LEAVE BLANK IF NOT WANT TO CHANGE )Advertisment

ENTER NEW SALARY,(LEAVE BLANK IF NOT WANT TO CHANGE

\*\*\*RECORD UPDATED\*\*\*

UPDATE MORE (Y) : n

Thank you!!!



---

## Experiment No: 18

---

**Aim:** Program to connect with database and delete the record of entered employee number.

**CODE:**

```
import mysql.connector as mycon
con = mycon.connect(host='localhost',user='root',password="root",database="company")
cur = con.cursor()
print("-----\nEMPLOYEE UPDATION FORM\n-----")

while True:
    eno = int(input("ENTER EMPNO TO DELETE :"))
    query="select * from employee where empno={}".format(eno)
    cur.execute(query)
    result = cur.fetchall()
    if cur.rowcount==0:
        print("Sorry! Empno not found ")
    else:
        print("%10s"% "EMPNO", "%20s"% "NAME", "%15s"% "DEPARTMENT",
"%10s"% "SALARY")
        for row in result:
            print("%10s"%row[0], "%20s"%row[1], "%15s"%row[2], "%10s"%row[3])
            choice=input("\n## ARE YOUR SURE TO DELETE ? (Y) :")
            if choice.lower()=='y':
                query="delete from employee where empno={}".format(eno)
                cur.execute(query)
                con.commit()
            print("=== RECORD DELETED SUCCESSFULLY! ===")
            ans=input("DELETE MORE ? (Y) :")
            if ans.lower()!='y':
                break
```

## **OUTPUT:**

### ----- EMPLOYEE UPDATION FORM -----

ENTER EMPNO TO DELETE :101

| EMPNO | NAME  | DEPARTMENT   | SALARY |
|-------|-------|--------------|--------|
| 101   | Albin | Data Analyst | 3100   |

## ARE YOUR SURE TO DELETE ? (Y) :n

=== RECORD DELETED SUCCESSFULLY! ===

DELETE MORE ? (Y) :y

ENTER EMPNO TO DELETE :105

Sorry! Empno not found

ENTER EMPNO TO DELETE :103

| EMPNO | NAME  | DEPARTMENT    | SALARY |
|-------|-------|---------------|--------|
| 103   | Kavan | Advertisement | 2900   |

## ARE YOUR SURE TO DELETE ? (Y) :y

=== RECORD DELETED SUCCESSFULLY! ===

DELETE MORE ? (Y) :n

---

## Experiment No: 19

---

**Aim:** Write a method CREATE() to create an EMP.csv file with the following details:

Emp\_no to store employee number of integer type,  
Emp\_name to store employee name of string type,  
Emp\_dep to store their respective department of string type,  
Emp\_basic to store basic salary of respective employee,  
Emp\_hra to be calculated from his/her basic salary which is 10% of basic  
Emp\_sal to be calculated as salary = basic\_salary + hra

**CODE:**

```
import csv
def CREATE():
    f=open("Emp.csv","a",newline="")
    obj=csv.writer(f)
    L=[]
    print("---Enter Details---")
    while True:
        print()
        emp_no = input("Enter employee number:")
        emp_name = input("Enter employee name:")
        emp_dep = input("Enter employee department:")
        emp_basic = int(input("Enter employee basic:"))
        emp_hra = emp_basic*10/100
        emp_sal = emp_basic + emp_hra
        L.append([emp_no,emp_name,emp_dep,emp_basic,emp_hra,emp_sal])
        print("%10s"%emp_no,"%10s"%emp_name,"%15s"%emp_dept,
"%15s"%emp_basic,"%10s"%emp_hra,"%15s"%emp_salary)
        print("%10s"%emp_no,"%10s"%emp_name,"%15s"%emp_dep,
"%15s"%emp_basic,"%10s"%emp_hra,"%15s"%emp_sal)
        ch=input("do you want to continue?")
        if ch.lower()!='y':
            break
    obj.writerows(L)
    f.close()

CREATE()
```

## OUTPUT:

---Enter Details---

Enter employee number:101

Enter employee name:Albin

Enter employee department:Data Analytics

Enter employee basic:3000

| emp_no | emp_name | emp_dept       | emp_basic | emp_hra | emp_salary |
|--------|----------|----------------|-----------|---------|------------|
| 101    | Albin    | Data Analytics | 3000      | 300.0   | 3300.0     |

do you want to continue?y

Enter employee number:102

Enter employee name:Kavan

Enter employee department:Advertisement

Enter employee basic:2900

| emp_no | emp_name | emp_dept      | emp_basic | emp_hra | emp_salary |
|--------|----------|---------------|-----------|---------|------------|
| 102    | Kavan    | Advertisement | 2900      | 290.0   | 3190.0     |

do you want to continue?y

Enter employee number:103

Enter employee name:Raed

Enter employee department:Sales

Enter employee basic:2800

| emp_no | emp_name | emp_dept | emp_basic | emp_hra | emp_salary |
|--------|----------|----------|-----------|---------|------------|
| 103    | Raed     | Sales    | 2800      | 280.0   | 3080.0     |

do you want to continue?n

---

## Experiment No: 20

---

**Aim : Write a Python program to copy file1.csv into file2.csv.**

**CODE:**

```
def COPY():
    import csv
    with open('csv1_copy.csv', 'w') as outfile:
        with open('csv1.csv', 'r') as infile:
            reader = csv.reader(infile)
            writer = csv.writer(outfile)

            for row in reader:
                writer.writerow(row)
COPY()
```

**OUTPUT:**

(Pic from csv1\_copy.csv file)

|    | A   | B | C   | D | E    | F | G |
|----|-----|---|-----|---|------|---|---|
| 1  | 101 |   | aaa |   | 1000 |   |   |
| 2  |     |   |     |   |      |   |   |
| 3  | 102 |   | bbb |   | 2000 |   |   |
| 4  |     |   |     |   |      |   |   |
| 5  | 103 |   | ccc |   | 3000 |   |   |
| 6  |     |   |     |   |      |   |   |
| 7  |     |   |     |   |      |   |   |
| 8  |     |   |     |   |      |   |   |
| 9  |     |   |     |   |      |   |   |
| 10 |     |   |     |   |      |   |   |
| 11 |     |   |     |   |      |   |   |
| 12 |     |   |     |   |      |   |   |

# mysql QUERIES

## MySQL Query

### 1. Display all the existing databases in your system

```
mysql> show databases;
```

| Database           |
|--------------------|
| information_schema |
| mysql              |
| performance_schema |
| sys                |

4 rows in set (0.00 sec)

### 2. Create a database AY2022

```
mysql> create database AY2022;
```

Query OK, 1 row affected (0.00 sec)

### 3. Check whether AY2022 database created or not

```
mysql> show databases;
```

| Database           |
|--------------------|
| information_schema |
| ay2022             |
| mysql              |
| performance_schema |
| sys                |

5 rows in set (0.00 sec)

### 4. Open database AY2022

```
mysql> USE Ay2022;
```

Database changed

### 5. Create a table STUDENT with the following specifications:

```
mysql> create table student(ROLL_NO int(5), STUD_NAME varchar(35), STREAM  
char(20), MARK1 decimal(5), MARK2 int(5), DOB date );
```

Query OK, 0 rows affected (0.06 sec)

### 6. Assyn work table

```
mysql> create table TEACHER(TR_ID INT(10),TR_NAME VARCHAR(35),TR_SAL  
DECIMAL(15,2),ROLL_NO INT(5));
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> desc student;
```

| Field     | Type        | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| ROLL_NO   | int(5)      | YES  |     | NULL    |       |
| STUD_NAME | varchar(35) | YES  |     | NULL    |       |
| STREAM    | char(20)    | YES  |     | NULL    |       |

|       |              |     |  |      |  |
|-------|--------------|-----|--|------|--|
| MARK1 | decimal(5,0) | YES |  | NULL |  |
| MARK2 | int(5)       | YES |  | NULL |  |
| DOB   | date         | YES |  | NULL |  |

6 rows in set (0.04 sec)

#### 7. Insert values into table student

```
mysql> insert into student values (101,'Surya Takur', 'Science', 90,87, '1990/10,10');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into student values (102,'Chris Tom','Humanities',88,91, '1994/1/5');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into student values (103,'Abel George','Commerce',93,95, '1993/10/7');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into student values (104,'Nathel Pillai','Science',56,58, '1994/8/8');
```

Query OK, 1 row affected (0.00 sec)

#### 8. To show all tables

```
mysql> show tables;
```

|                  |
|------------------|
| Tables_in_ay2022 |
| student          |
| teacher          |

2 rows in set (0.00 sec)

#### 9. Delete the table TEACHER and check

```
mysql> drop database school;
```

Query OK, 0 rows affected (0.01 sec)

#### 10. Create a database SCHOOL, display all databases, delete it and check

```
mysql> show databases;
```

|                    |
|--------------------|
| Database           |
| information_schema |
| ay2021             |
| ay2022             |
| mysql              |
| performance_schema |
| sys                |

6 rows in set (0.00 sec)

#### 11. Display all the contents from STUDENT table

```
mysql> select * from student;
```



```

+-----+-----+-----+-----+-----+-----+
| ROLL_NO | STUD_NAME | STREAM | MARK1 | MARK2 | DOB |
+-----+-----+-----+-----+-----+-----+
| 101 | Surya Takur | Science | 90.00 | 87.00 | 1990-10-10 |
| 102 | Chris Tom | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103 | Abel George | Commerce | 93.00 | 95.00 | 1993-10-07 |
| 103 | Nathel Pillai | Science | 56.00 | 58.00 | 1994-08-08 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

## 12. Display only the student name along with his/her stream.

### WHERE clause

```
mysql> SELECT stud_name, stream from STUDENT;
```

```

+-----+-----+
| stud_name | stream |
+-----+-----+
| Surya Takur | Science |
| Chris Tom | Humanities |
| Abel George | Commerce |
| Nathel Pillai | Science |
+-----+-----+
4 rows in set (0.00 sec)

```

## 13. Display all the student in science stream.

```
mysql> SELECT * FROM student WHERE stream='Science';
```

```

+-----+-----+-----+-----+-----+-----+
| ROLL_NO | STUD_NAME | STREAM | MARK1 | MARK2 | DOB |
+-----+-----+-----+-----+-----+-----+
| 101 | Surya Takur | Science | 90.00 | 87.00 | 1990-10-10 |
| 103 | Nathel Pillai | Science | 56.00 | 58.00 | 1994-08-08 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

## 14. Display all student name and both marks whose mark1 is greater than 89

```
mysql> select stud_name, mark1, mark2 from student where mark1>89;
```

```

+-----+-----+-----+
| stud_name | mark1 | mark2 |
+-----+-----+-----+
| Surya Takur | 90.00 | 87.00 |
| Abel George | 93.00 | 95.00 |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

## 15. Display all the student details whose marks2 is between 91 and 95.

```
mysql> select * from student where mark2>=91 and mark2<=95;
```

```

+-----+-----+-----+-----+-----+-----+
| ROLL_NO | STUD_NAME | STREAM | MARK1 | MARK2 | DOB |
+-----+-----+-----+-----+-----+-----+
| 102 | Chris Tom | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103 | Abel George | Commerce | 93.00 | 95.00 | 1993-10-07 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

**16. Display all student details whose date of birth is after '1<sup>st</sup> Jan 1994'.**

```
mysql> select * from student where dob>'1994/01/01';
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1 | MARK2 | DOB        |
|---------|---------------|------------|-------|-------|------------|
| 102     | Chris Tom     | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103     | Nathel Pillai | Science    | 56.00 | 58.00 | 1994-08-08 |
| 110     | John          | NULL       | NULL  | NULL  | 1994-11-19 |

3 rows in set (0.00 sec)

**17. Display all student details whose date of birth is 1<sup>st</sup> Jan 1994.**

```
mysql> select * from student where dob>1994/01/01;
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1  | MARK2 | DOB        |
|---------|---------------|------------|--------|-------|------------|
| 101     | Surya Takur   | Science    | 90.00  | 87.00 | 1990-10-10 |
| 102     | Chris Tom     | Humanities | 88.00  | 91.00 | 1994-01-05 |
| 103     | Abel George   | Commerce   | 93.00  | 95.00 | 1993-10-07 |
| 103     | Nathel Pillai | Science    | 56.00  | 58.00 | 1994-08-08 |
| 110     | Karthik Kiran | NULL       | 100.00 | NULL  | 1993-10-10 |
| 110     | John          | NULL       | NULL   | NULL  | 1994-11-19 |

6 rows in set, 1 warning (0.00 sec)

**18. Insert using field names and using null**

a) Roll\_no = 110, stud\_name = 'Karthik Kiran', mark1=100, dob = '1993/10/10' [ using null ]

b) Roll\_no = 111, stud\_name = 'Jerry John' , dob = '19/11/1994' [ using field names ]

```
mysql> insert into student values (110, 'Karthik Kiran' ,null, 100, null,'1993/10/10');
```

Query OK, 1 row affected (0.03 sec)

```
mysql> insert into student (roll_no,stud_name,dob) values (110, 'John', '1994/11/19');
```

Query OK, 1 row affected (0.02 sec)

```
mysql> select * from student;
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1  | MARK2 | DOB        |
|---------|---------------|------------|--------|-------|------------|
| 101     | Surya Takur   | Science    | 90.00  | 87.00 | 1990-10-10 |
| 102     | Chris Tom     | Humanities | 88.00  | 91.00 | 1994-01-05 |
| 103     | Abel George   | Commerce   | 93.00  | 95.00 | 1993-10-07 |
| 103     | Nathel Pillai | Science    | 56.00  | 58.00 | 1994-08-08 |
| 110     | Karthik Kiran | NULL       | 100.00 | NULL  | 1993-10-10 |
| 110     | John          | NULL       | NULL   | NULL  | 1994-11-19 |

6 rows in set (0.00 sec)

### 19. Display all the records with either mark1 or amrk2 is null

```
mysql> select * from student where mark1 IS NULL OR mark2 IS NULL;
```

| ROLL_NO | STUD_NAME     | STREAM | MARK1  | MARK2 | DOB        |
|---------|---------------|--------|--------|-------|------------|
| 110     | Karthik Kiran | NULL   | 100.00 | NULL  | 1993-10-10 |
| 110     | John          | NULL   | NULL   | NULL  | 1994-11-19 |

2 rows in set (0.02 sec)

```
mysql> select * from student where mark1 IS NULL and mark2 IS NULL;
```

| ROLL_NO | STUD_NAME | STREAM | MARK1 | MARK2 | DOB        |
|---------|-----------|--------|-------|-------|------------|
| 110     | John      | NULL   | NULL  | NULL  | 1994-11-19 |

1 row in set (0.00 sec)

### 20. Display all the records with either mark1 or mark2 is not full

```
mysql> select * from student where mark1 IS NOT NULL and mark2 IS NOT NULL;
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1 | MARK2 | DOB        |
|---------|---------------|------------|-------|-------|------------|
| 101     | Surya Takur   | Science    | 90.00 | 87.00 | 1990-10-10 |
| 102     | Chris Tom     | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103     | Abel George   | Commerce   | 93.00 | 95.00 | 1993-10-07 |
| 103     | Nathel Pillai | Science    | 56.00 | 58.00 | 1994-08-08 |

4 rows in set (0.00 sec)

```
mysql> select * from student where mark1 IS NOT NULL or mark2 IS NOT NULL;
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1  | MARK2 | DOB        |
|---------|---------------|------------|--------|-------|------------|
| 101     | Surya Takur   | Science    | 90.00  | 87.00 | 1990-10-10 |
| 102     | Chris Tom     | Humanities | 88.00  | 91.00 | 1994-01-05 |
| 103     | Abel George   | Commerce   | 93.00  | 95.00 | 1993-10-07 |
| 103     | Nathel Pillai | Science    | 56.00  | 58.00 | 1994-08-08 |
| 110     | Karthik Kiran | NULL       | 100.00 | NULL  | 1993-10-10 |

5 rows in set (0.00 sec)

```
mysql> select distinct stream from student;
```

| stream     |
|------------|
| Science    |
| Humanities |
| Commerce   |
| NULL       |

4 rows in set (0.00 sec)

**21. Display all the streams available in STUDENT table without duplicating.**

```
mysql> select DISTINCT stream from student;
```

| stream     |
|------------|
| Science    |
| Humanities |
| Commerce   |
| NULL       |

4 rows in set (0.00 sec)

**22. Display the student names along with roll number with appropriate field names as STUDENT NAME and ROLL NUMBER (alias names)**

```
mysql> select roll_no ROLL_NUMBER, stud_name STUDENT_NAME from student;
```

| ROLL_NUMBER | STUDENT_NAME  |
|-------------|---------------|
| 101         | Surya Takur   |
| 102         | Chris Tom     |
| 103         | Abel George   |
| 103         | Nathel Pillai |
| 110         | Karthik Kiran |
| 110         | John          |

6 rows in set (0.00 sec)

```
mysql> desc student;
```

| Field     | Type         | Null | Key | Default | Extra |
|-----------|--------------|------|-----|---------|-------|
| ROLL_NO   | int(5)       | YES  |     | NULL    |       |
| STUD_NAME | varchar(35)  | YES  |     | NULL    |       |
| STREAM    | char(20)     | YES  |     | NULL    |       |
| MARK1     | decimal(6,2) | YES  |     | NULL    |       |
| MARK2     | decimal(6,2) | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |

6 rows in set (0.02 sec)\

**23. Display student names along with date of birth whose mark2 is 91 and 95 (using IN operator)**

```
mysql> select stud_name, dob from student where mark2 in (91,95);
```

| stud_name   | dob        |
|-------------|------------|
| Chris Tom   | 1994-01-05 |
| Abel George | 1993-10-07 |

2 rows in set (0.00 sec)

```
mysql> select stud_name,dob from student where mark2=91 or mark2=92;
```

```
+-----+-----+
| stud_name | dob       |
+-----+-----+
| Chris Tom | 1994-01-05 |
+-----+-----+
1 row in set (0.00 sec)
```

**24. Display all the student details whose date of birth is in the year 1994 (using year() function)**

```
mysql> select * from student where year(dob)=1994;
```

```
+-----+-----+-----+-----+-----+-----+
| ROLL_NO | STUD_NAME | STREAM | MARK1 | MARK2 | DOB       |
+-----+-----+-----+-----+-----+-----+
| 102     | Chris Tom | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103     | Nathel Pillai | Science | 56.00 | 58.00 | 1994-08-08 |
| 110     | John      | NULL    | NULL  | NULL  | 1994-11-19 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

**25. Display the student roll number along with student name whose date of birth is in October month (using month() function)**

```
mysql> select stud_name,roll_no from student where month(dob)=10;
```

```
+-----+-----+
| stud_name | roll_no |
+-----+-----+
| Surya Takur | 101 |
| Abel George | 103 |
| Karthik Kiran | 110 |
+-----+-----+
3 rows in set (0.00 sec)
```

**26. Display roll no and student name of students who are born in October**

```
mysql> select roll_no, stud_name from student where month(dob)=10;
```

```
+-----+-----+
| roll_no | stud_name |
+-----+-----+
| 101     | Surya Takur |
| 103     | Abel George |
| 110     | Karthik Kiran |
+-----+-----+
3 rows in set (0.00 sec)
```

**27. Display the student roll number along with student name whose date of birth is in date after 9<sup>th</sup> of the month (using day() function)**

```
mysql> select roll_no, stud_name from student where day(dob)>9;
```

```
+-----+-----+
| roll_no | stud_name |
+-----+-----+
| 101     | Surya Takur |
| 110     | Karthik Kiran |
| 110     | John      |
+-----+-----+
3 rows in set (0.00 sec)
```

**28. Display the details from student where the names are in descending order**

```
mysql> select * from student order by stud_name desc;
```

| ROLL_NO | STUD_NAME     | STREAM     | MARK1 | MARK2 | DOB        |
|---------|---------------|------------|-------|-------|------------|
| 101     | Surya Takur   | Science    | 90    | 87    | 1990-10-10 |
| 104     | Nathel Pillai | Science    | 56    | 58    | 1994-08-08 |
| 110     | Karthik Kiran | NULL       | 100   | NULL  | 1993-10-10 |
| 110     | John          | NULL       | NULL  | NULL  | 1994-11-19 |
| 102     | Chris Tom     | Humanities | 88    | 91    | 1994-01-05 |
| 103     | Abel George   | Commerce   | 93    | 95    | 1993-10-07 |

6 rows in set (0.01 sec)

**29. Display the name and DOB from student where stream is science and their dob are in ascending order.**

```
mysql> select stud_name, DOB from student where stream='science' order by dob;
```

| stud_name     | DOB        |
|---------------|------------|
| Surya Takur   | 1990-10-10 |
| Nathel Pillai | 1994-08-08 |

2 rows in set (0.00 sec)

**30. Display the details from student where the stream has more than 7 characters.**

```
mysql> select * from student where length(stream)>7;
```

| ROLL_NO | STUD_NAME   | STREAM     | MARK1 | MARK2 | DOB        |
|---------|-------------|------------|-------|-------|------------|
| 102     | Chris Tom   | Humanities | 88    | 91    | 1994-01-05 |
| 103     | Abel George | Commerce   | 93    | 95    | 1993-10-07 |

2 rows in set (0.01 sec)

**31. Display values from student table in descending order of names**

```
mysql> select * from student order by stud_name desc;
```

| roll_no | stud_name     | stream     | mark1 | mark2 | dob        |
|---------|---------------|------------|-------|-------|------------|
| 101     | Surya Takur   | Science    | 90.00 | 87.00 | 1990-10-10 |
| 104     | Nathel Pillai | Science    | 56.00 | 58.00 | 1994-08-08 |
| 102     | Chris Tom     | Humanities | 88.00 | 91.00 | 1994-01-05 |
| 103     | Able George   | Commerce   | 93.00 | 95.00 | 1993-10-07 |

4 rows in set (1.83 sec)

### **32. Display stud name and dob from science stream order by dob**

```
mysql> select stud_name, dob from student where stream='Science' order by dob asc;
```

```
+-----+-----+
| stud_name | dob      |
+-----+-----+
| Surya Takur | 1990-10-10 |
| Nathel Pillai | 1994-08-08 |
+-----+-----+
2 rows in set (0.03 sec)
```

### **32. Display stream names whose length is more than 7 characters**

```
mysql> select stream from student where length(stream)>7;
```

```
+-----+
| stream |
+-----+
| Humanities |
| Commerce |
+-----+
2 rows in set (0.09 sec)
```

### **33. Change the date of birth of roll number 102 as 1/5/1994**

```
mysql> update student set dob = '1994/5/1' where roll_no=102;
```

Query OK, 1 row affected (0.91 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
+-----+-----+-----+-----+-----+-----+
| roll_no | stud_name | stream | mark1 | mark2 | dob      |
+-----+-----+-----+-----+-----+-----+
| 101 | Surya Takur | Science | 92.00 | 87.00 | 1990-10-10 |
| 102 | Chris Tom | Humanities | 90.00 | 91.00 | 1994-05-01 |
| 103 | Able George | Commerce | 95.00 | 95.00 | 1993-10-07 |
| 104 | Nathel Pillai | Science | 58.00 | 58.00 | 1994-08-08 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

### **34. Increase the mark1 for all student by 2**

```
mysql> update student set mark1=mark1+2;
```

Query OK, 1 row affected (0.91 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
+-----+-----+-----+-----+-----+-----+
| roll_no | stud_name | stream | mark1 | mark2 | dob      |
+-----+-----+-----+-----+-----+-----+
| 101 | Surya Takur | Science | 92.00 | 87.00 | 1990-10-10 |
| 102 | Chris Tom | Humanities | 90.00 | 91.00 | 1994-05-01 |
| 103 | Able George | Commerce | 95.00 | 95.00 | 1993-10-07 |
| 104 | Nathel Pillai | Science | 58.00 | 58.00 | 1994-08-08 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**35. Change the name of 'Nathel Pillai' to 'Nithin Pillai' whose roll number is 104**

```
mysql> update student set stud_name="Nithin Pillia" where roll_no=104;
Query OK, 1 row affected (0.91 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

| roll_no | stud_name     | stream     | mark1 | mark2 | dob        |
|---------|---------------|------------|-------|-------|------------|
| 101     | Surya Takur   | Science    | 92.00 | 87.00 | 1990-10-10 |
| 102     | Chris Tom     | Humanities | 90.00 | 91.00 | 1994-05-01 |
| 103     | Able George   | Commerce   | 95.00 | 95.00 | 1993-10-07 |
| 104     | Nithin Pillia | Science    | 58.00 | 58.00 | 1994-08-08 |

4 rows in set (0.00 sec)

**36. Delete student details whose mark1 is less than 88 and mark2 less than 88**

```
mysql> delete from student where mark1<88 and mark2<88;
Query OK, 1 row affected (0.16 sec)
```

| roll_no | stud_name   | stream     | mark1 | mark2 | dob        |
|---------|-------------|------------|-------|-------|------------|
| 101     | Surya Takur | Science    | 92.00 | 87.00 | 1990-10-10 |
| 102     | Chris Tom   | Humanities | 90.00 | 91.00 | 1994-05-01 |
| 103     | Able George | Commerce   | 95.00 | 95.00 | 1993-10-07 |

3 rows in set (0.00 sec)

**37. Delete the student table**

```
mysql> delete from student;
```

Query OK, 3 rows affected (0.12 sec)

Empty set (0.00 sec)

( Table structure exists but its empty)

```
mysql> CREATE VIEW name AS SELECT roll_no, stud_name FROM student WHERE
mark1>90 AND mark2>90;
```

Query OK, 0 rows affected (0.03 sec)

**38. Drop view table**

```
mysql> DROP VIEW name;
```

Query OK, 0 rows affected (0.01 sec)

**39. To add column in a table**

```
mysql> ALTER TABLE student ADD COLUMN location varchar(50);
```

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0



| Field     | Type         | Null | Key | Default | Extra |
|-----------|--------------|------|-----|---------|-------|
| ROLL_NO   | int          | YES  |     | NULL    |       |
| STUD_NAME | varchar(35)  | YES  |     | NULL    |       |
| STREAM    | char(20)     | YES  |     | NULL    |       |
| MARK1     | decimal(6,2) | YES  |     | NULL    |       |
| MARK2     | decimal(6,2) | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |
| location  | varchar(50)  | YES  |     | NULL    |       |

7 rows in set (0.00 sec)

#### 40. Drop add column in a table

```
mysql> ALTER TABLE student DROP COLUMN location;
```

Query OK, 0 rows affected (0.09 sec)

Records: 0 Duplicates: 0 Warnings: 0

| Field     | Type         | Null | Key | Default | Extra |
|-----------|--------------|------|-----|---------|-------|
| ROLL_NO   | int          | YES  |     | NULL    |       |
| STUD_NAME | varchar(35)  | YES  |     | NULL    |       |
| STREAM    | char(20)     | YES  |     | NULL    |       |
| MARK1     | decimal(6,2) | YES  |     | NULL    |       |
| MARK2     | decimal(6,2) | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |

6 rows in set (0.00 sec)

#### 41. To modify the column location with new detail as varchar of size 40

```
mysql> alter table student modify location varchar(40);
```

Query OK, 0 rows affected (0.56 sec)

Records: 0 Duplicates: 0 Warnings: 0

#### 42. Rename the location column as street with size as varchar 90

```
mysql> alter table student change location Street varchar(90);
```

Query OK, 0 rows affected (0.48 sec)

Records: 0 Duplicates: 0 Warnings: 0

| Field     | Type         | Null | Key | Default | Extra |
|-----------|--------------|------|-----|---------|-------|
| roll_no   | int(5)       | NO   | PRI | NULL    |       |
| stud_name | varchar(35)  | YES  |     | NULL    |       |
| stream    | char(20)     | YES  |     | NULL    |       |
| mark1     | decimal(6,2) | YES  |     | NULL    |       |
| mark2     | decimal(6,2) | YES  |     | NULL    |       |
| dob       | date         | YES  |     | NULL    |       |
| Street    | varchar(90)  | YES  |     | NULL    |       |

7 rows in set (0.00 sec)

#### 42. Display the total mark1 from student table

```
mysql> select sum(mark1) from student;
```

```
+-----+
| sum(mark1) |
+-----+
|      327.00 |
+-----+
1 row in set (0.05 sec)
```

#### 43. Display the senior most and junior most student from student table

```
mysql> select stud_name, max(dob) from student;
```

```
+-----+-----+
| stud_name | max(dob) |
+-----+-----+
| Surya Takur | 1994-08-08 |
+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> select stud_name, min(dob) from student;
```

```
+-----+-----+
| stud_name | min(dob) |
+-----+-----+
| Surya Takur | 1990-10-10 |
+-----+-----+
1 row in set (0.00 sec)
```

#### 44. Display the average mark2 from the student table whose second name starts with 't'

```
mysql> select avg(mark2) from student where stud_name like '% t%';
```

```
+-----+
| avg(mark2) |
+-----+
|  89.000000 |
+-----+
1 row in set (0.05 sec)
```

```
mysql> select round(avg(mark2),2) from student where stud_name like '% t%';
```

```
+-----+
| round(avg(mark2),2) |
+-----+
|           89.00 |
+-----+
1 row in set (0.08 sec)
```

#### 45. Display the number of students whose marks is more than 90

```
mysql> select count(*) from student where mark1>90;
```

```

+-----+
| count(*) |
+-----+
|         1 |
+-----+
1 row in set (0.05 sec)

```

*mysql> select \* from student group by dob having stud\_name like '%a%';*

**OR**

*mysql> select \* from student where stud\_name like '%a%' group by dob;*

```

+-----+-----+-----+-----+-----+-----+-----+
| roll_no | stud_name | stream | mark1 | mark2 | dob       | Street |
+-----+-----+-----+-----+-----+-----+-----+
|      101 | Surya Takur | Science | 90.00 | 87.00 | 1990-10-10 | Dubai |
|      104 | Abel George | Commerce | 93.00 | 95.00 | 1993-10-07 | Dubai |
|      102 | Nathel Pillai | Science | 56.00 | 58.00 | 1994-08-08 | Dubai |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.16 sec)

```

#### 46. Display number of student in each stream

*mysql> select count(\*) from student group by stream;*

```

+-----+
| count(*) |
+-----+
|         1 |
|         1 |
|         2 |
+-----+
3 rows in set (0.04 sec)

```

#### 47. Display the number of students in each stream whose DOB is after 1990

*mysql> select stream,count(\*) totalstudents from student group by stream;*

```

+-----+-----+
| stream | totalstudents |
+-----+-----+
| Commerce |          1 |
| Humanities |          1 |
| Science |          2 |
+-----+-----+
3 rows in set (0.01 sec)

```

#### 48. Display all details from teacher table and student table whose mark1 is greater than 90

*mysql> select \* from student,teacher where student.roll\_no=teacher.roll\_no and student.mark2>90;*

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| roll_no | stud_name | stream | mark1 | mark2 | dob       | Street | TR_ID | TR_NAME | TR_SAL | ROLL_NO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      103 | Chris Tom | Humanities | 88.00 | 91.00 | 1994-08-08 | Abu Dhabi | 999 | Elias John | 9900.00 |      103 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

49. Display the details from client and product together using the join query.

```
mysql> select * from product,client;
```

| P_ID | Product_Name    | Manufacture | Price | C_ID | Client_Name        | City           | P_ID |
|------|-----------------|-------------|-------|------|--------------------|----------------|------|
| 1001 | Plastic Bottles | XYZ         | 30    | 6901 | DreamZ Disney      | New Delhi      | 1001 |
| 1001 | Plastic Bottles | XYZ         | 30    | 6902 | Life Line Hospital | Mumbai         | 1004 |
| 1001 | Plastic Bottles | XYZ         | 30    | 6903 | 98.7 FM            | New Delhi      | 1003 |
| 1001 | Plastic Bottles | XYZ         | 30    | 6904 | Appolo             | Madhya Pradesh | 1005 |
| 1002 | Bath Soap       | ABC         | 50    | 6901 | DreamZ Disney      | New Delhi      | 1001 |
| 1002 | Bath Soap       | ABC         | 50    | 6902 | Life Line Hospital | Mumbai         | 1004 |
| 1002 | Bath Soap       | ABC         | 50    | 6903 | 98.7 FM            | New Delhi      | 1003 |
| 1002 | Bath Soap       | ABC         | 50    | 6904 | Appolo             | Madhya Pradesh | 1005 |
| 1003 | Shampoo         | COP         | 65    | 6901 | DreamZ Disney      | New Delhi      | 1001 |
| 1003 | Shampoo         | COP         | 65    | 6902 | Life Line Hospital | Mumbai         | 1004 |
| 1003 | Shampoo         | COP         | 65    | 6903 | 98.7 FM            | New Delhi      | 1003 |
| 1003 | Shampoo         | COP         | 65    | 6904 | Appolo             | Madhya Pradesh | 1005 |
| 1004 | Lens Solution   | TAP         | 350   | 6901 | DreamZ Disney      | New Delhi      | 1001 |
| 1004 | Lens Solution   | TAP         | 350   | 6902 | Life Line Hospital | Mumbai         | 1004 |
| 1004 | Lens Solution   | TAP         | 350   | 6903 | 98.7 FM            | New Delhi      | 1003 |
| 1004 | Lens Solution   | TAP         | 350   | 6904 | Appolo             | Madhya Pradesh | 1005 |
| 1005 | Sanitizer       | COP         | 35    | 6901 | DreamZ Disney      | New Delhi      | 1001 |
| 1005 | Sanitizer       | COP         | 35    | 6902 | Life Line Hospital | Mumbai         | 1004 |
| 1005 | Sanitizer       | COP         | 35    | 6903 | 98.7 FM            | New Delhi      | 1003 |
| 1005 | Sanitizer       | COP         | 35    | 6904 | Appolo             | Madhya Pradesh | 1005 |

20 rows in set (0.00 sec)

50. Display the client id manufacture and city from product and client table where client ID is 6903

```
mysql> select c_id, manufacture, city from product p, client c where c.c_id=6903;
```

| c_id | manufacture | city      |
|------|-------------|-----------|
| 6903 | XYZ         | New Delhi |
| 6903 | ABC         | New Delhi |
| 6903 | COP         | New Delhi |
| 6903 | TAP         | New Delhi |
| 6903 | COP         | New Delhi |

5 rows in set (0.00 sec)

51. Display client name and manufacture details from the product and client table where price is greater than 45

```
mysql> select client_name, manufacture from product p, client c where p.p_id=c.p_id and price>45;
```

| client_name        | manufacture |
|--------------------|-------------|
| Life Line Hospital | TAP         |
| 98.7 FM            | COP         |

2 rows in set (0.00 sec)

**52. Display the product name and city details from the client and product table whose city is New Delhi**

```
mysql> select product_name, city from product, client where  
client.city='new delhi';
```

| product_name    | city      |
|-----------------|-----------|
| Plastic Bottles | New Delhi |
| Plastic Bottles | New Delhi |
| Bath Soap       | New Delhi |
| Bath Soap       | New Delhi |
| Shampoo         | New Delhi |
| Shampoo         | New Delhi |
| Lens Solution   | New Delhi |
| Lens Solution   | New Delhi |
| Sanitizer       | New Delhi |
| Sanitizer       | New Delhi |

10 rows in set (0.00 sec)