Computer Science Project

Certificate

This is to certify that Krishna Bagade, a student of class XII-B, Ryan International School, Sanpada has successfully completed the coursework of Computer Science, that is "Python Project File" under the guidance of Mrs Gargly Dam during 2022-23 in partial fulfilment of physics practical examination of Central Board of Secondary Education.

Signature of Internal Examiner:

Signature of External Examiner:

Signature of Principal:

School stamp:

ACKNOWLEDGEMENT

I would like to express my sincere and special gratitude to our Principal Ma'am, Mrs. Muriel Fernandes of our esteemed Ryan International School, Sanpada for always encouraging us to excel in all that we do. I would like to thank my Computer Science teacher Mrs Gargly Dam for her continuous guidance and encouragement and immense motivation which helped me at all stages of this project. Lastly, I would like to thank my family and friends for helping me in the completion of the project.

#1 To write a program that prints a table on two columns-table that helps converting miles into kilometre

Source code:-

```
print("Miles\t | \tKilometer")

for j in range (7)
    print(10**j , end= "\t" )
    print(" | " , end= "\t")
    print( 1.6093 * 10**j)
```

Output:-

```
Miles    |  kilometer
1    |  1.6093
10   |  16.093
100  |  160.93
1000     |  1609.3
10000    |  16093.0
100000   |  160930.0
1000000  |  1609300.0
```

#2 To write a program printing a table with two columns that help convert pounds into kilograms

Source code:-

```
print("Pound\t | \tKilogram")

for j in range (0,101,5)
    print( j , end= "\t" )
    print(" | " , end="\t")
    print( 0.45* j )
```

Output:-

```
Pounds   |  Kilogram
0    |  0.0
5    |  2.25
10   |  4.5
15   |  6.75
20   |  9.0
25   |  11.25
30   |  13.5
35   |  15.75
40   |  18.0
45   |  20.25
50   |  22.5
55   |  24.75
60   |  27.0
65   |  29.25
70   |  31.5
75   |  33.75
80   |  36.0
85   |  38.25
90   |  40.5
95   |  42.75
100  |  45.0
```

#3 To write a program that reads two times in military format(0900,1730) and prints the number of hours and minutes between the two times.

Source code:-

```
first = int(input("Please Enter the
first time ="))
sec = int(input("Please Enter the
second time = " ))

a = sec - First

print(a // 100, "hours" , a
%100,"min")
```

Output:-

```
Please Enter the first time = 0900
Please Enter the sec time = 1730
8 hours 30 min
```

```
Please Enter the first time = 1115
Please Enter the sec time = 1730
6 hours 15 min
```

```
Please Enter the first time = 1214
Please Enter the sec time = 2230
10 hours 16 min
```

#4 To write a function called addDict(dict1,dict2) which computes the union of two dictionaries. It should return a new dictionary, with all the items in of both its arguments (assumed to be dictionaries). If the same key appears in both arguments,pick a value from either dictionary.

Source code:-

```
def adddict(dict1,dict2) :
    dic = {}
    for i in dict1 :
        dic  [ i ] = dict1 [i]
    for j in dict2 :
        dic [ j ] = dict2 [j]
    return dic
dict1=eval(input("Enter First
Dictionary :- "))
dic2=eval(input("Enter Second
Dictionary :-"))

print(adddict(dict1,dict2))
```

Output:-

```
Enter First Dictionary :-{"Zone":1,"Area":2,"Street":3}
Enter Second Dictionary:-{"Road":4,"Area":2,"Street":4}
{'Zone': 1, 'Area': 2, 'Street': 4, 'Road': 4}
```

```
Enter First Dictionary :-{"Name":"Lion","Scientific name":"Panthera leo","Length":"1.8m"}
Enter Second Dictionary:-{"Mass":"190kg","Lifespan":"8-10years","speed":"80km/hr"}
{'Name': 'lion', 'Scientific name': 'Panthera leo', 'Length': '1.8m', 'Mass': '190kg', 'Lifespan': '8-10years', 'speed': '80km/hr'}
```

```
Enter First Dictionary :-{"Brand name":"Dominos","Type of food":"Pizza"}
Enter Second Dictionary:-{"Average price":"300-400"}
{'Brand name': 'Dominos', 'Type of food': 'Pizza', 'Average price': '300-400'}
```

#5 To write a program to sort a dictionary's keys using Bubble sort and produce the sorted keys as a list.

Source Code:-

```
dic = eval (input("Enter a dictionary : "))
key = list(dic.keys())

For j in range (len(key)):
    For i in range ( len ( key ) - 1) :
        If  key [ i ] > key [ i +1 ] :
            key [ i ] , key [ i + 1 ] =
key [ i +1 ] , key [ i ]

print(key)
```

Output:-

```
Enter a dictionary:{1:"Violet",2:"Indigo",3:"blue"}
[1, 2, 3]
```

```
Enter a dictionary:{5:"red",4:"green",7:"yellow"}
[4, 5, 7]
```

#6 Write a function that receives 2 numbers and generates a random number from the range. Using this function the main program should be able to print 3 numbers randomly.

Source Code:-

```
import random
def generate_random(a,b):
    randNum=[]
    randNum.append(random.randrange(a, b))
    randNum.append(random.randrange(a, b))
    randNum.append(random.randrange(a, b))
    print(randNum)
    return randNum


#__main__
first_number=int(input("Enter the first number:"))
second_number=int(input("Enter the second
number:"))
generate_random(first_number,second_number)
```

Output:-

```
/usr/bin/python3 /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXII/ch3Q.4.py
Enter the first number:4
Enter the second number:56
[47, 8, 11]

Process finished with exit code 0
```

```
/usr/bin/python3 /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXII/ch3Q.4.py
Enter the first number:4
Enter the second number:56
[47, 8, 11]

Process finished with exit code 0
```

#7 Write a function that receives 2 string arguments and check whether they are same length strings(return true otherwise false)

Source Code:-

```
def string_length(string1,string2):
    return len(string1)==len(string2)

#__main__
string1=input("Enter the first string: ")
string2=input("Enter the second string: ")
print(string_length(string1,string2))
```

Output:-

```
/usr/bin/python3 /Users/krishnabagade/PycharmProjects
/demo/ch3.Q5.py
Enter the first string: redhead
Enter the second string: redbed
False
Enter the first string: luffy
Enter the second string: flufy
True
```

#8 Write a function that takes two numbers and returns the number that has minimum 1's digit.

Source Code:-

```
def minimumcheck(num1,num2):
    uni1=num1%10
    uni2=num2%10
    if uni1<uni2:
        return num1
    else:
        return num2


#__main__
num1=int(input("Enter the first number: "))
num2=int(input("Enter the second number: "))
print("Minimum one's
digit",minimumcheck(num1,num2))
```

Output:-

```
/usr/bin/python3 /Users/krishnabagade/PycharmProjects
/demo/ch3.Q8.py
Enter the first number: 107
Enter the second number: 454
Minimum one's digit 454
Enter the first number: 10000
Enter the second number: 9999
Minimum one's digit 10000

Process finished with exit code 0
```

#9 Write a function that receives 2 string arguments and check whether they are same length strings(return true otherwise false)

Source Code:-

```
def string_length(string1,string2):
    return len(string1)==len(string2)

#__main__
string1=input("Enter the first string: ")
string2=input("Enter the second string: ")
print(string_length(string1,string2))
```

Output :-

```
/usr/bin/python3 /Users/krishnabagade/PycharmProjects
/demo/ch3.Q5.py
Enter the first string: redhead
Enter the second string: redbed
False
Enter the first string: luffy
Enter the second string: flufy
True
```

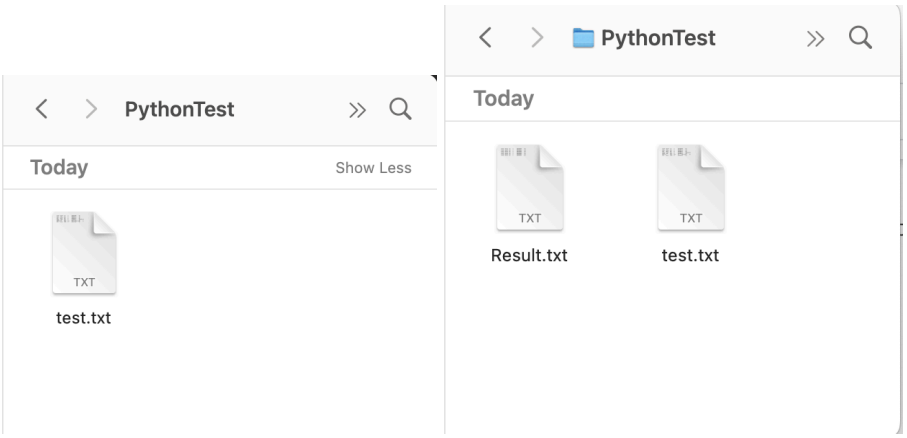
#10 Write a program that copies one file to another. Have the program read filenames from the user.

Source Code:-

```
def readFile(filepath):
    myfile = open(filepath, "r")
    s=myfile.readlines()
    print("File output",s)
    writeFile(s)
    myfile.close()


def writeFile(s):

fileout=open("/Users/krishnabagade/Documents/Python
Test/Result.txt", "w")
    for sen in range(len(s)):
        fileout.write(s[0])
    fileout.close()

#/Users/krishnabagade/Documents/PythonTest/test.txt
userip=input("Enter file path:")
readFile(userip)
```

Output:

-




```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/filehandling.py
Enter file path:/Users/krishnabagade/Documents/PythonTest/test.txt
File output ['Fluffy\tHarold\tcat\tf\t1993-02-04\t\\N\n', 'Claws\tGwen\tcat\tm\t1994-03-17\t\\N\n', 'Buffy\tHarold\tdog\tf\t1989-05-13\t\\N\n', 'Fang\tBenny\tdog\

Process finished with exit code 0
```

Result.txt

```
Fluffy  Harold  cat     f       1993-02-04      \N
Claws   Gwen    cat     m       1994-03-17      \N
Buffy   Harold  dog     f       1989-05-13      \N
Fang    Benny   dog     m       1990-08-27      \N
Bowser  Diane   dog     m       1979-08-31      1995-07-29
Chirpy  Gwen    bird    f       1998-09-11      \N
Whistler        Gwen    bird    \N      1997-12-09      \N
Slim    Benny   snake   m       1996-04-29      \N
```

test.txt

```
Fluffy  Harold  cat     f       1993-02-04      \N
Claws   Gwen    cat     m       1994-03-17      \N
Buffy   Harold  dog     f       1989-05-13      \N
Fang    Benny   dog     m       1990-08-27      \N
Bowser  Diane   dog     m       1979-08-31      1995-07-29
Chirpy  Gwen    bird    f       1998-09-11      \N
Whistler        Gwen    bird    \N      1997-12-09      \N
Slim    Benny   snake   m       1996-04-29      \N
```

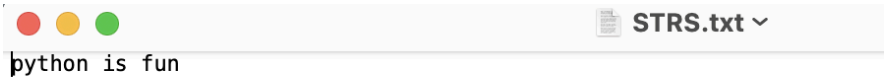
#11 Write a program that will create an object called fallout for writing , associate it with the filename STRS.txt. The code should keep on writing strings to it as long as the user wants.

Source Code:-

```
def writeFile():

fileout=open("/Users/krishnabagade/Documents/Python Test/STRS.txt", "w")
    while True:
        print("Enter file operation:")
        print("1.To write sentences in file")
        print("2. to close the file")
        ch=int(input("Enter choice:"))
        if ch==1:
            s=input("Enter text to be added:")
            fileout.write(s)
        elif ch==2:
            fileout.close()
            break
        else:
            print("Please enter a valid number:")

#__main__
writeFile()
```

Output:

-



#12 Consider the following definition of a dictionary member, write a method in python to write the content in a pickled file member.dat.

Member = {'MemberNo.' :_____, 'Name' : _____ }

Source Code:-

```
import pickle

def dictionary_wb(mem1):
    memberfile =
open('/Users/krishnabagade/Documents/PythonTest/mem
ber.dat','wb')
    pickle.dump(mem1, memberfile)
    memberfile.close()


mem_no = input("Enter dictionary member number to
be recorded in file memberfile:")
mem_name = input("Enter dictionary member name to
be recorded in file memberfile:")
mem1 = {mem_no: mem_name}
dictionary_wb(mem1)
```

Output :-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/filehandling3.py
Enter dictionary member number to be recorded in file memberfile:3267463759038503952948297420480 2
Enter dictionary member name to be recorded in file memberfile:Tara Singh Honey Yo Yo

Process finished with exit code 0
```

Äï@

#13 Write a program that depending on the user's choice, either pushes or pops an element in a stack. The elements are shifted towards the right so that the top always remains at the 0th (zeroth)index.

Source Code:-

```
def POP(Arr):
    if len(Arr) >0:
        v = Arr.pop(0)
        return v
    else:
        print("Stack is empty underflow occurred")


def push(Arr, a):
    Arr.insert(0,a)


Arr = []
Top = None
while True:
    print("stack operations")
    print("1.Push")
    print("2.Pop")
    print("3.Exit")
    ch = int(input("Enter your choice 1-3:"))
    if ch == 1:
        a = input("Enter number to be entered in
stack:")
        push(Arr, a)
        print(Arr)
    elif ch == 2:
        h=POP(Arr)
        print(h)
    elif ch == 3:
        break
    else:
```

```
print("Enter a valid number:")
```

Output:-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/stackspf.py
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:1
Enter number to be entered in stack:rge
['rge']
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:1
Enter number to be entered in stack:121
['121', 'rge']
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:2
121
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:
```

#14 Write a function in python POP(Arr), where Arr is a stack implemented by a list of numbers. The function returns values deleted from the stack.

Source Code:-

```
def POP(Arr):
    if len(Arr) >0:
        v=Arr.pop()
        return v
    else:
        print("Stack is empty underflow occurred")


def push(Arr,a):
    Arr.append(a)
    top=len(Arr)-1


Arr = []
Top = None
while True :
    print("stack operations")
    print("1.Push")
    print("2.Pop")
    print("3.Exit")
    ch=int(input("Enter your choice 1-3:"))
    if ch ==1:
        a=int(input("Enter number to be entered in
stack:"))
        push(Arr,a)
    elif ch ==2:
        h=POP(Arr)
        print(h)
    elif ch ==3:
        break
    else:
        print("Enter a valid number:")
```

Output:

-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/stackspf.py
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:1
Enter number to be enetred in stack:23
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:2
23
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:2
Stack is empty underflow occurred
None
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:1
Enter number to be enetred in stack:3
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:3

Process finished with exit code 0
```

#15 A line of text is read from the input terminal into a stack. Write a program to output the string in the reverse order, each character appearing twice.

Source Code:-

```
def POP(Arr):
    res=[]
    while Arr:
        cur_char=Arr.pop()
        res.extend([cur_char,cur_char])
    return res


def push(Arr,a):
    for chr in a:
        Arr.append(chr)


Arr = []
while True :
    print("stack operations")
    print("1.Push")
    print("2.Pop")
    print("3.Exit")
    ch=int(input("Enter your choice 1-3:"))
    if ch == 1:
        sentence = input("Enter sentence to be
entered in stack:")
        push(Arr, sentence)
    elif ch == 2:
        h=POP(Arr)
        print(h)
    elif ch ==3:
        break
```

Output:

-

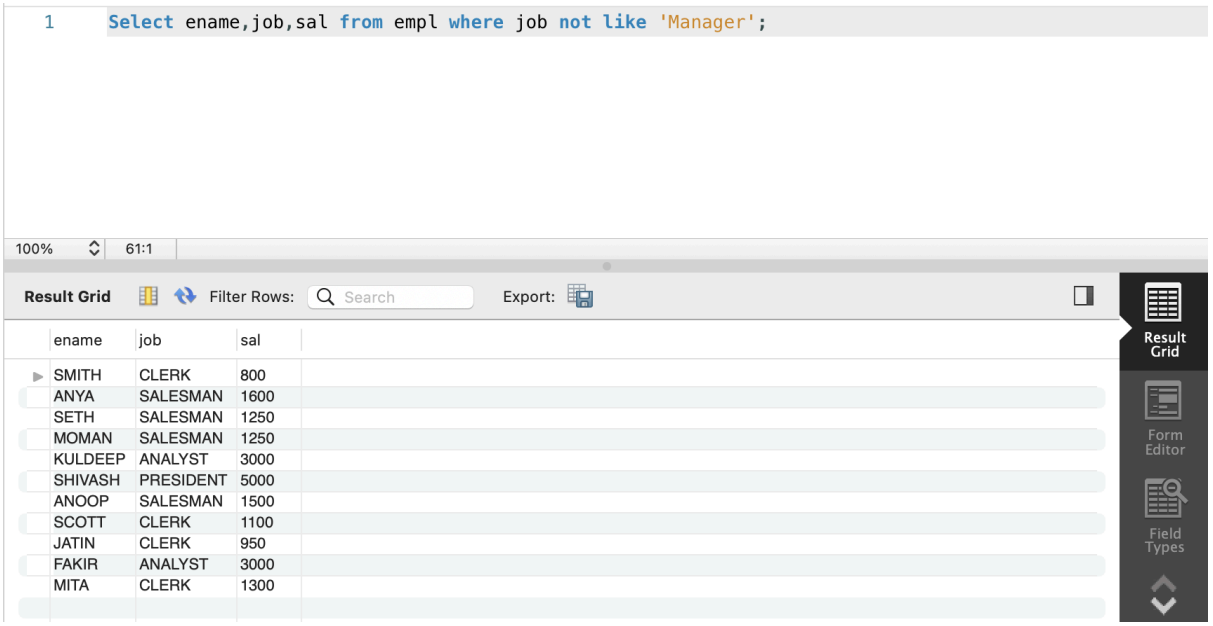
```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/stackspf.py
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:1
Enter sentence to be entered in stack:python is fun
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:2
['n', 'n', 'u', 'u', 'f', 'f', ' ', ' ', 's', 's', 'i', 'i', ' ', ' ', 'n', 'n', 'o', 'o', 'h', 'h', 't', 't', 'y', 'y', 'p', 'p']
stack operations
1.Push
2.Pop
3.Exit
Enter your choice 1-3:
```

#16 Write a query to display the name,job title and salary of those employees who do not have manager.

Source Code:-

```
Select ename,job,sal from empl where job not like 'Manager';
```

Output:-

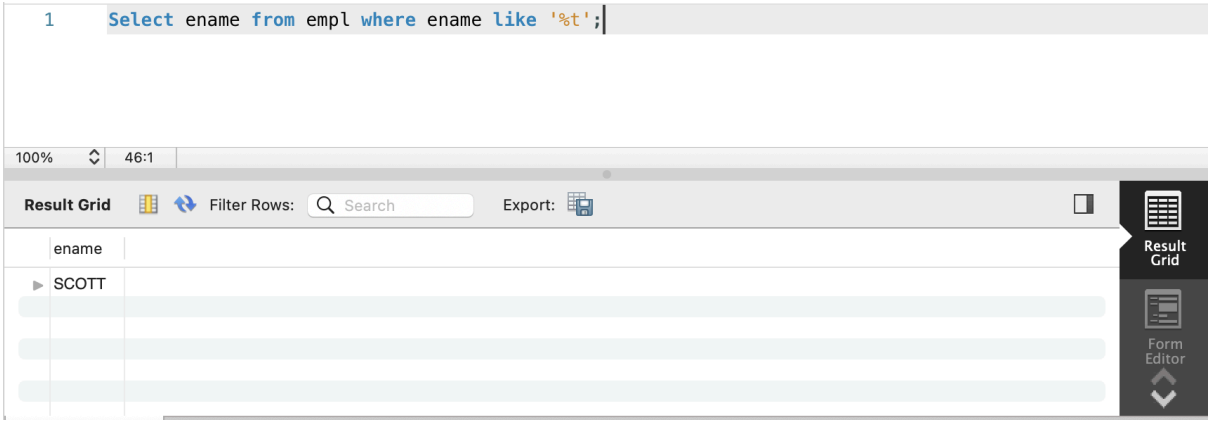


#17 Write a query to display the name of the employee whose name contains ‘T’ as the last alphabet.

Source Code:-

```
Select ename from empl where ename like '%t';
```

Output:-

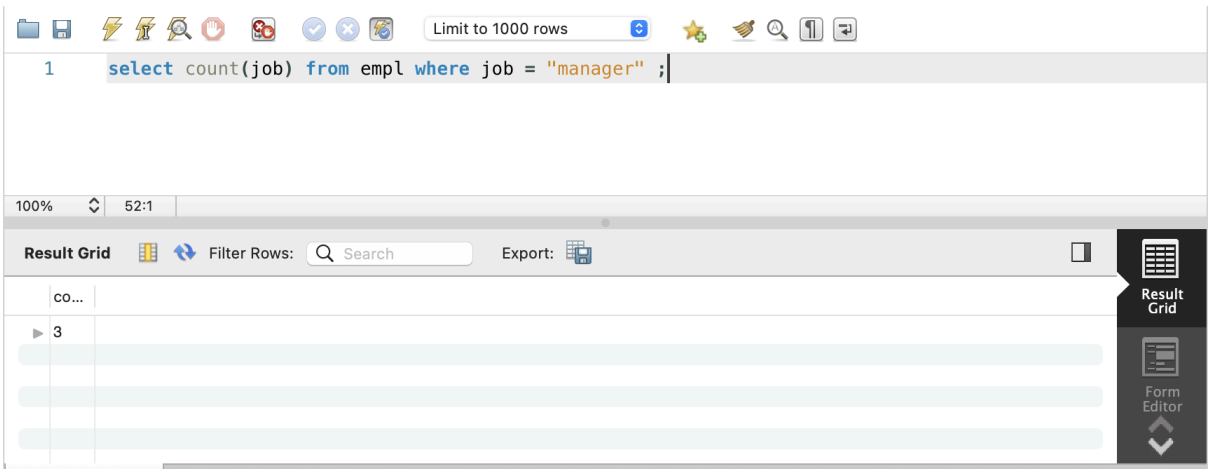


#18 Find out number of employees having “Manager” as Job.

Source Code:-

```
select count(job) from empl where job = "manager" ;
```

Output:-

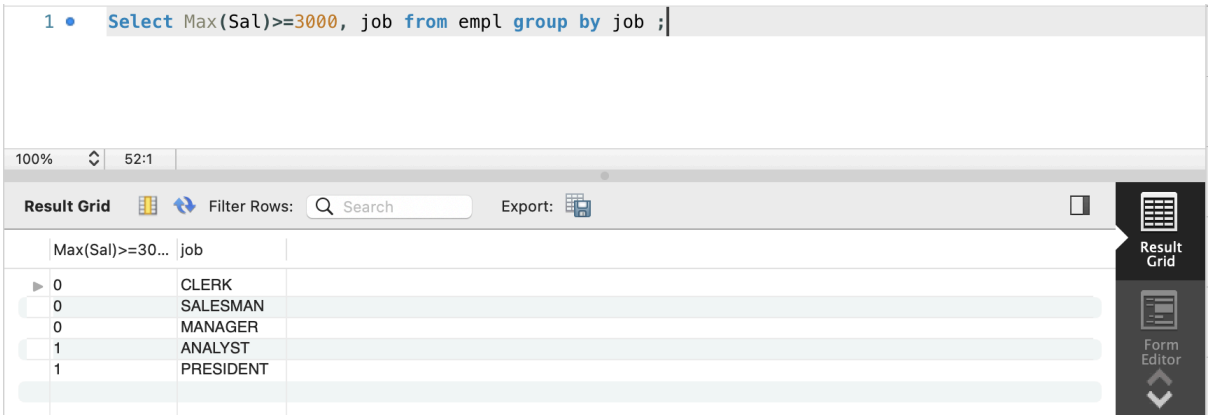


#19 Display only the jobs with maximum salary greater than or equal to 3000.

Source Code:-

```
Select Max(Sal)>=3000, job from empl group by job
;
```

Output:
-



#20 Show the average salary for all the departments with more than 3 people for a job.

Source Code:-

```
Select Avg(Sal), deptno from empl group by deptno
having count(job)>3
```

Output:

-

#21 Write a Python database connectivity script that fetches match one table of database empl that has dept no = ‘20’.

Source Code:-

```
import mysql.connector as slqr
def search_criteria():

mycon=slqr.connect(host="localhost",user="root",p
asswd="kingler79",database="menagerie",buffered=T
rue)
    if mycon.is_connected():
        print("Successfully connected to mysql")
        cursor=mycon.cursor()
        strbuilder="Select * from empl where
deptno in ('20');"
        query= cursor.execute(strbuilder)
        values=cursor.fetchmany(3)
        for v in values:
            print(v)


# __main__
print(search_criteria())
```

Output:-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/interface1.py
Successfully connected to mysql
(8369, 'SMITH', 'CLERK', 8902, datetime.date(1990, 12, 18), 800.0, None, 20)
(8566, 'MAHADEVAN', 'MANAGER', 8839, datetime.date(1991, 4, 2), 2975.0, None, 20)
(8788, 'KULDEEP', 'ANALYST', 8566, datetime.date(1991, 9, 8), 3000.0, None, 20)
None

Process finished with exit code 0
```

#22 Write a Python database connectivity script that fetches one table of database empl that has job = ‘Salesman’.

Source Code:-

```
import mysql.connector as slqr
def search_criteria():

mycon=slqr.connect(host="localhost",user="root",pas
swd="kingler79",database="menagerie",buffered=True)
    if mycon.is_connected():
        print("Successfully connected to mysql")
        cursor=mycon.cursor()
        strbuilder="Select * from empl where job in
('salesman');"
        query= cursor.execute(strbuilder)
        values=cursor.fetchone()
        print(values)
        values = cursor.fetchone()
        print(values)


 # __main__
print(search_criteria())
```

Output :-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/interface1.py
Successfully connected to mysql
(8499, 'ANYA', 'SALESMAN', 8698, datetime.date(1991, 2, 20), 1600.0, 300.0, 30)
(8521, 'SETH', 'SALESMAN', 8698, datetime.date(1991, 2, 22), 1250.0, 500.0, 30)
None

Process finished with exit code 0
```

#23 Design a Python application that fetches all the records from Pet table of menagerie database.

Source Code:-

```
import mysql.connector as slqr
mycon=slqr.connect(host="localhost",user="root"
,passwd="kingler79",database="menagerie",buffer
ed=True)
if mycon.is_connected():
    print("Successfully connected to mysql")
    cursor=mycon.cursor()
    query= cursor.execute("Select * from pet")
    values=cursor.fetchall()
    for v in values:
        print(v)
```

Output:-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/interface1.py
Successfully connected to mysql
('Puffball', 'Diane', 'hamster', 'f', datetime.date(1999, 3, 30), None)
('Fluffy', 'Harold', 'cat', 'f', datetime.date(1993, 2, 4), None)
('Claws', 'Gwen', 'cat', 'm', datetime.date(1994, 3, 17), None)
('Buffy', 'Harold', 'dog', 'f', datetime.date(1989, 5, 13), None)
('Fang', 'Benny', 'dog', 'm', datetime.date(1990, 8, 27), None)
('Bowser', 'Diane', 'dog', 'm', datetime.date(1979, 8, 31), None)
('Chirpy', 'Gwen', 'bird', 'f', datetime.date(1998, 9, 11), None)
('Whistler', 'Gwen', 'bird', 'f', datetime.date(1997, 12, 9), None)
('Slim', 'Benny', 'snake', 'm', datetime.date(1996, 4, 29), None)
```

#24 Design a Python application that fetches only those records from Event table menagerie database where type is Kennel.

Source Code:-

```
import mysql.connector as slqr
mycon=slqr.connect(host="localhost",user="r
oot",passwd="kingler79",database="menagerie
",buffered=True)
if mycon.is_connected():
    print("Successfully connected to
mysql")
    cursor=mycon.cursor()
    query= cursor.execute("Select * from
event where type='kennel'")
    values=cursor.fetchall()
    for v in values:
        print(v)
```

Output:-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/interface1.py
Successfully connected to mysql
('Bowser', datetime.date(1991, 10, 12), 'kennel', None)
('Fang', datetime.date(1991, 10, 12), 'kennel', None)

Process finished with exit code 0
```

#25 Design a Python application to obtain a search criteria from the user and then fetch records based on that from empl table.

Source Code:-

```
import mysql.connector as slqr
def search_criteria(string1, string2):

mycon=slqr.connect(host="localhost",user="root",pa
sswd="kingler79",database="menagerie",buffered=Tru
e)
    if mycon.is_connected():
        print("Successfully connected to mysql")
        cursor=mycon.cursor()
        strbuilder="Select * from empl where
"+string1+" in ('"+string2+"');"
        query= cursor.execute(strbuilder)
        values=cursor.fetchall()
        for v in values:
            print(v)


    # __main__
string1 = input("Enter the Column Name from
[EMPNO,ENAME,JOB,MGR,HIREDATA,SAL,COMM,DEPTNO]: ")
string2 = input("Enter the Filter value : ")
print(search_criteria(string1, string2))
```

Output:-

```
/Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/venv/bin/python /Users/krishnabagade/PycharmProjects/Oct2022/pythonProjectXIIB/interface1.py
Enter the Column Name from [EMPNO,ENAME,JOB,MGR,HIREDATA,SAL,COMM,DEPTNO]: JOB
Enter the Filter value : clerk
Successfully connected to mysql
(8369, 'SMITH', 'CLERK', 8902, datetime.date(1990, 12, 18), 800.0, None, 20)
(8876, 'SCOTT', 'CLERK', 8788, datetime.date(1992, 12, 9), 1100.0, None, 20)
(8900, 'JATIN', 'CLERK', 8698, datetime.date(1991, 12, 3), 950.0, None, 30)
(8934, 'MITA', 'CLERK', 8782, datetime.date(1992, 1, 23), 1300.0, None, 10)
None

Process finished with exit code 0
```

Bibliography

1. Computer Science with Python, Sumita Arora, Dhanpat Rai & Co.