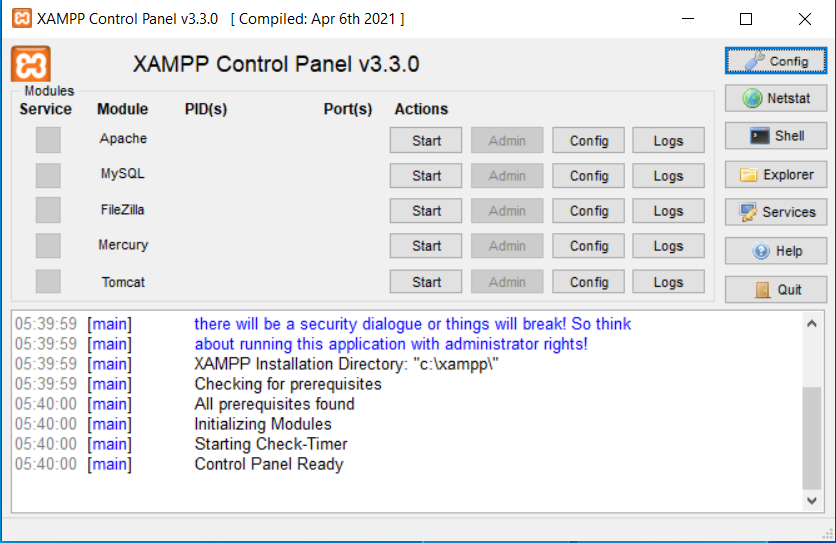
Install appropriate XAMPP from <https://www.apachefriends.org/> for your computer.

Refer <https://undsgn.com/xampp-tutorial/>

After successful completion, you will see the following screen.



Press Start button against MySQL and click on Admin button in the same row to run phpMyAdmin.

# **Create and access MySQL database**

## **Create a user with user-id as “testUser” with password as “A4min@123”.**

## **Create a database Employees.**

## **Create three MySQL database tables as per the details shown below:**

1. Employee table is created using the following command:

CREATE TABLE EMPLOYEE (

EMP\_ID CHAR(6) NOT NULL,

DEPT\_ID CHAR(4) NOT NULL,

FIRST\_NAME CHAR(20) NOT NULL,

LAST\_NAME CHAR(20),

AGE INT,

SEX CHAR(1),

INCOME FLOAT,

DOJ DATETIME);

The above field names are self-explanatory. However, DOJ field refers to Date of Joining.

1. Department table is created using the following command:

CREATE TABLE DEPARTMENT(

DEPT\_ID CHAR(4) NOT NULL,

DEPT\_NAME CHAR(20),

EMP\_ID CHAR(6) NOT NULL);

The above field names are self-explanatory.

1. Leaves table is created using the following command:

CREATE TABLE IF NOT EXISTS leaves

(

EMP\_ID CHAR(6) NOT NULL,

LEAVE\_START\_DATE DATE,

LEAVE\_END\_DATE DATE,

LEAVE\_DAYS TINYINT NOT NULL);

The above field names are self-explanatory.

# **Insert data into these tables through Python code**

## **Class file: db\_Class\_v1.py**

# -\*- coding: utf-8 -\*-

"""

Created on Sun Feb 16 00:22:25 2020

@author: P.V.SUBRAMANIAN

"""

import mysql.connector

from mysql.connector import Error, errorcode

import random

import string

class Employee:

"""

In this Class object, we establish the data base connection, create MySQL tables and insert the

records for the following tables

a. EMPLOYEE

b. DEPARTMENT

c. LEAVES

"""

import mysql.connector

from mysql.connector import Error, errorcode

import random

import string

def \_\_init\_\_(self, host, user, pwd, db):

self.host = host

self.db = db

self.user = user

self.pwd = pwd

def db\_connection(self):

self.connection = mysql.connector.connect(host = self.host,

database = self.db,

user = self.user,

password = self.pwd)

return self.connection

def db\_create\_table(self):

# emp = Employee(self)

# # prepare a cursor object using cursor() method

# cursor = emp.db\_connection(self)

#

# Drop table if it already exist using execute() method.

connection = self.connection

cursor = connection.cursor()

cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")

# Create table as per requirement

sql = """CREATE TABLE EMPLOYEE (

EMP\_ID CHAR(6) NOT NULL,

DEPT\_ID CHAR(4) NOT NULL,

FIRST\_NAME CHAR(20) NOT NULL,

LAST\_NAME CHAR(20),

AGE INT,

SEX CHAR(1),

INCOME FLOAT,

DOJ DATETIME)"""

cursor.execute(sql)

cursor.execute("DROP TABLE IF EXISTS DEPARTMENT")

sql\_dept = """CREATE TABLE DEPARTMENT(

DEPT\_ID CHAR(4) NOT NULL,

DEPT\_NAME CHAR(20),

EMP\_ID CHAR(6) NOT NULL)"""

cursor.execute(sql\_dept)

cursor.execute("DROP TABLE IF EXISTS LEAVES")

sql\_leaves = """CREATE TABLE IF NOT EXISTS LEAVES(

EMP\_ID CHAR(6) NOT NULL,

LEAVE\_START\_DATE DATE,

LEAVE\_END\_DATE DATE,

LEAVE\_DAYS TINYINT NOT NULL);"""

cursor.execute(sql\_leaves)

def insertVariblesIntoTable(self, EMPID, DEPTID, FN, LN, AGE, SEX, INCOME, DOJ):

connection = self.connection

try:

cursor = connection.cursor()

mySql\_insert\_query = """INSERT INTO employee (EMP\_ID, DEPT\_ID, FIRST\_NAME, LAST\_NAME, AGE, SEX, INCOME, DOJ)

VALUES (%s, %s, %s, %s, %s, %s, %s, %s) """

recordTuple = (EMPID, DEPTID, FN, LN, AGE, SEX, INCOME, DOJ)

cursor.execute(mySql\_insert\_query, recordTuple)

connection.commit()

print("Record inserted successfully into Employee table")

connection.commit()

print("Record inserted successfully into Employee table")

except mysql.connector as error:

print("Failed to insert into MySQL table %s " % error)

def insertVariblesIntoTable2(self, DEPTID, DEPTNAME, EMPID):

connection = self.connection

try:

cursor = connection.cursor()

mySql\_insert\_query = """INSERT INTO department (DEPT\_ID, DEPT\_NAME,EMP\_ID)

VALUES (%s, %s, %s) """

recordTuple = (DEPTID, DEPTNAME, EMPID)

cursor.execute(mySql\_insert\_query, recordTuple)

connection.commit()

print("Record inserted successfully into Department table")

cursor = connection.cursor()

except mysql.connector as error:

print("Failed to insert into MySQL table %s " % error)

def insertVariblesIntoTable3(self, EMPID, LEAVE\_START, LEAVE\_END, LDAYS):

connection = self.connection

try:

cursor = connection.cursor()

mySql\_insert\_query = """INSERT INTO leaves (EMP\_ID, LEAVE\_START\_DATE, LEAVE\_END\_DATE, LEAVE\_DAYS)

VALUES (%s, %s, %s, %s) """

recordTuple = (EMPID, LEAVE\_START, LEAVE\_END, LDAYS)

cursor.execute(mySql\_insert\_query, recordTuple)

connection.commit()

print("Record inserted successfully into Leaves table")

cursor = connection.cursor()

except mysql.connector as error:

print("Failed to insert into MySQL table %s " % error)

###

### End of class

###

**Insert data by using Poupluate\_employee\_tables.py**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Jun 20 07:58:40 2022

@author: PVS

"""

import random

from datetime import datetime

import sys, os

import random

import string

import mysql.connector

# pip install mysql-connector-python

folder\_name = r'D:\GL\Interview\_DSE\Mock-interview-main'

sys.path.insert(0, folder\_name)

now = datetime.now()

current\_time = now.strftime("%H:%M:%S")

from db\_Class\_v1 import Employee

error = None

emp = Employee("localhost","testUser", "A4min@123","employees" )

connection = emp.db\_connection()

emp.db\_create\_table()

###

### Insert data

###

n = 20

EMPID\_LIST = list()

for i in range(n):

FN\_list = ["Arun", "Arockia", "Alex", "Amin", "Bhim", "Chandar", "Durvan", "Durvik", "Dhanis", "Yashnav", \

"Sheshvan", "Sowgatha", "Vijaychandra", "Vijay", "Praveen", "Arya", "Ritesh", "Piyush",\

"Arya","Xavier"]

FN\_list\_Female= ["Anika", "Parvathi Devi", "Aslesha", "Aashna", "Ahalya", "Anamika",\

"Bianca", "Binal", "Bina", "Bagya", "Saranya", "Diana", "Disha",\

"Deepti", "Esha", "Estaa", "Eshika", "Firaki", "Fanha", "Gina", "Gauri",\

"Hasina", "Hrithika", "Isha", "Iravati"]

DEPTNAME\_list = ['HR & Admin', 'Finance & Accounts' , 'Production', 'Sales & Marketing', 'Purchase']

DEPTNAME = random.choice(DEPTNAME\_list)

DEPTNAME = random.choice(DEPTNAME\_list)

DEPT\_ID = str(DEPTNAME\_list.index(DEPTNAME) + 1)

DOJ\_list = ['1990-01-31', '1998-10-31','1997-03-31','2000-04-01','2001-04-11',\

'1995-07-21', '2002-05-24','2004-01-22','2008-07-01','2010-08-01',\

'1997-11-01', '1997-11-01','1996-09-20','2014-02-03','1993-11-03',\

'2010-01-11', '2009-11-09','2017-10-20','2009-10-11','2008-03-20']

DOJ = random.choice(DOJ\_list)

EMP\_ID = str(random.randrange(1000, 9000))

FN = random.choice(FN\_list)

LN = random.choice(random.choice(string.ascii\_uppercase))

AGE = str(random.choice(list(range(21, 66))))

SEX = random.choice(['M', 'F'])

INCOME = str(round(float(random.random()) \* 100000,0))

try:

print(EMP\_ID)

EMPID\_LIST += [EMP\_ID]

except:

print('Error')

emp.insertVariblesIntoTable(EMP\_ID, DEPT\_ID, FN, LN, AGE, 'M', INCOME, DOJ) # Male employees

EMP\_ID = str(random.randrange(100, 900))

FN\_Female = random.choice(FN\_list\_Female)

LN = random.choice(random.choice(string.ascii\_uppercase))

AGE = str(random.choice(list(range(21, 65))))

SEX = random.choice(['M', 'F'])

INCOME = str(round(float(random.random()) \* 100000,0))

try: ## Error

print(EMP\_ID)

EMPID\_LIST += [EMP\_ID]

except:

print('Error')

emp.insertVariblesIntoTable(EMP\_ID, DEPT\_ID, FN\_Female, LN, AGE, 'F', INCOME, DOJ) # Female employees

###

DEPTNAME\_list = ['HR & Admin', 'Finance & Accounts' , 'Production', 'Sales & Marketing', 'Purchase']

MANAGER\_list = ['HR00', 'FA00' , 'PR00' , 'SM00', 'PU00']

NAME\_list = ['Rosy' , 'Peter', 'Arul', 'Murugan', 'Sowmya']

LN = random.choice(random.choice(string.ascii\_uppercase))

SEX\_list = ['F', 'M', 'M', 'M', 'F']

DOJ\_list\_Manager = ['2010-01-11', '2009-11-09','2017-10-20','2009-10-11','2008-03-20']

for i in range(len(DEPTNAME\_list)):

DEPTNAME = DEPTNAME\_list[i]

DEPT\_ID = str(i + 1)

EMP\_ID = MANAGER\_list[i]

FN\_MANAGER = NAME\_list[i]

LN = random.choice(random.choice(string.ascii\_uppercase))

SEX = SEX\_list[i]

DOJ = DOJ\_list\_Manager[i]

emp.insertVariblesIntoTable(EMP\_ID, DEPT\_ID, FN\_MANAGER, LN, AGE, SEX, INCOME, DOJ)

emp.insertVariblesIntoTable2(DEPT\_ID, DEPTNAME, EMP\_ID)

###

### Insert records into leaves table

###

from datetime import datetime, timedelta

print(EMPID\_LIST)

for i in range(5):

EMP\_ID = random.choice(EMPID\_LIST)

DATE\_delt = random.randint(10, 365)

start = datetime.now() + timedelta(days = -DATE\_delt)

start\_date = start.strftime('%Y-%m-%d')

LDAYS = random.randint(1, 10)

end = start + timedelta(days= (LDAYS-1))

end\_date = end.strftime('%Y-%m-%d')

emp.insertVariblesIntoTable3(EMP\_ID, start, end, str(LDAYS))

###

sql\_select = """SELECT CONVERT(count(\*), CHAR) FROM EMPLOYEE"""

cursor = connection.cursor()

cursor.execute(sql\_select)

count = cursor.fetchone()

print("\n Total Employee records inserted is %s" % count)

sql\_select1 = """SELECT CONVERT(count(\*), CHAR) FROM DEPARTMENT"""

cursor = connection.cursor()

cursor.execute(sql\_select1)

count = cursor.fetchone()

print("\n Total Department records inserted is %s" % count)

sql\_select = """SELECT CONVERT(count(\*), CHAR) FROM LEAVES"""

cursor = connection.cursor()

cursor.execute(sql\_select)

count = cursor.fetchone()

print("\n Total Leaves records inserted is %s" % count)

# **Retrieve data from these tables through Python code**

**set1\_Q.py**

# -\*- coding: utf-8 -\*-

"""

Created on Fri Oct 30 12:08:56 2020

@author: PVS

"""

###

### Set 1

###

import mysql.connector, sys

from datetime import datetime

folder\_name = r'D:\GL\_DSE\_Interview\Skill\_Assessment\_CSI'

sys.path.insert(0, folder\_name)

now = datetime.now()

current\_time = now.strftime("%H:%M:%S")

from db\_Class import Employee

error = None

emp = Employee("localhost","testUser", "A4min@123","employees" )

connection = emp.db\_connection()

cursor = connection.cursor()

file\_name\_out = 'SQL\_Test at ' + str(current\_time) + ".txt"

f = open("SQL\_test\_log\_SET1.txt", "w")

f.write(file\_name\_out)

f.write("\n== SET 1 ==\n")

###

f.write("\nA) Query involving a single database table\n")

###

### Q1

###

f.write("\nQ1) Give the count of male employees\n")

sql\_query1 = """SELECT COUNT(\*) FROM EMPLOYEE WHERE SEX = 'M';"""

cursor.execute(sql\_query1)

results\_M\_count = cursor.fetchone()

try:

M\_count = "\nMale employees count : " + str(results\_M\_count[0])

print(M\_count)

f.write(M\_count)

Explain\_txt1 = "\nThe WHERE clause is used to compare the given value with the field value available in the database table."

f.write(Explain\_txt1)

Explain\_txt2 = "\nThe COUNT() function returns the number of records returned by a SELECT query.\n\n"

f.write(Explain\_txt2)

except:

print("\nError")

###

### Q2

###

f.write("\n\nQ2) Give the count of employees who joined 10 years ago (i.e. experience in the company is 10 years or more) \n")

sql\_query2 = """SELECT COUNT(\*) FROM EMPLOYEE WHERE DOJ < DATE\_SUB(NOW(), INTERVAL 10 YEAR);"""

cursor.execute(sql\_query2)

results\_10y\_count = cursor.fetchone()

try:

count\_10y = "\nCount of Employees who joined 10 years ago : " + str(results\_10y\_count[0])

f.write(count\_10y)

Explain\_txt1 = "\n\nThe DATE\_SUB() function subtracts a time / date interval from a date and then returns the date."

f.write(Explain\_txt1)

except:

print("\nError")

### ========================================================================

f.write("\n\n\nB) Query involving two database tables")

###

### Q3

###

sql\_select\_y1 = """SELECT CONCAT(A.FIRST\_NAME, " ", A.LAST\_NAME) AS NAME """

sql\_select\_y2 = """FROM EMPLOYEE A, DEPARTMENT B """

sql\_select\_y3 = """WHERE A.EMP\_ID = B.EMP\_ID AND A.DOJ = ALL(SELECT MAX(DOJ) FROM employee);"""

sql\_query = sql\_select\_y1 + sql\_select\_y2 + sql\_select\_y3

cursor.execute(sql\_query)

results\_yM = cursor.fetchone()

print('\nManager who joined last : {}'.format(results\_yM[0]))

f.write("\n\nQ3) Give the name of the manager who joined last, i.e. later than all other managers")

try:

M\_last = "\n\nManager who joined last : " + results\_yM[0]

f.write(M\_last )

except:

print("\nError")

###

### Q4

###

sql\_select\_sal1 = """SELECT COUNT(\*), A.DEPT\_ID, B.DEPT\_NAME """

sql\_select\_sal2 = """FROM EMPLOYEE A, DEPARTMENT B """

sql\_select\_sal3 = """WHERE INCOME > ALL(SELECT avg(income) FROM employee) and a.DEPT\_ID = b.DEPT\_ID GROUP by DEPT\_ID;"""

sql\_query = sql\_select\_sal1 + sql\_select\_sal2 + sql\_select\_sal3

cursor.execute(sql\_query)

results\_sal = cursor.fetchall()

f.write("\n\nQ4) Give the count, department ID and department name of the employees drawing more than the average salary")

for r in results\_sal:

print('\nCount of Employees drawing more salary department-wise : {}'.format(str(r[0])))

try:

Salary\_more = "\n\nCount of employees drawing more salary than average : " + str(r[0]) + " Dept ID : " + \

r[1] + " Name : " + r[2]

f.write(Salary\_more)

except:

print("\nError")

# disconnect from server

cursor.close()

f.close()

### ========================================================================

**set2\_Q.py**

# -\*- coding: utf-8 -\*-

"""

Created on Fri Oct 30 12:08:56 2020

@author: PVS

"""

###

### Set 1

###

import mysql.connector, sys

from datetime import datetime

folder\_name = r'E:\DSE-MOCK-INTERVIEW'

sys.path.insert(0, folder\_name)

now = datetime.now()

current\_time = now.strftime("%H:%M:%S")

from db\_Class\_220621 import Employee

error = None

emp = Employee("localhost","testUser", "A4min@123","hr" )

connection = emp.db\_connection()

cursor = connection.cursor()

file\_name\_out = 'SQL\_Test at ' + str(current\_time) + ".txt"

f = open("SQL\_test\_log\_SET2.txt", "w")

f.write(file\_name\_out)

f.write("\n== SET 2 ==\n")

###

f.write("\nA) Query involving a single database table\n")

###

### Q1

###

sql\_query1 = """SELECT COUNT(\*) FROM EMPLOYEE WHERE SEX = 'F';"""

cursor.execute(sql\_query1)

results\_F\_count = cursor.fetchone()

f.write("\nQ1) Give the count of female employees\n")

f.write("\n\nAnwers\n\n")

f.write(sql\_query1)

try:

F\_count = "\nFemale employees count : " + str(results\_F\_count[0])

print(F\_count)

f.write(F\_count)

Explain\_txt1 = "\nThe WHERE clause is used to compare the given value with the field value available in the database table."

f.write(Explain\_txt1)

Explain\_txt2 = "\nThe COUNT() function returns the number of records returned by a SELECT query.\n\n"

f.write(Explain\_txt2)

except:

print("\nError")

###

### Q2

###

sql\_query2 = """SELECT COUNT(\*) FROM EMPLOYEE WHERE DOJ < DATE\_SUB(NOW(), INTERVAL 5 YEAR);"""

cursor.execute(sql\_query2)

results\_5y\_count = cursor.fetchone()

f.write("\n\nQ2) Give the count of employees who joined 5 years ago (i.e. experience in the company is 5 years or more) \n")

f.write("\n\nAnwers\n\n")

f.write(sql\_query2)

try:

count\_5y = "\nCount of Employees who joined 5 years ago : " + str(results\_5y\_count[0])

f.write(count\_5y)

Explain\_txt1 = "\n\nThe DATE\_SUB() function subtracts a time / date interval from a date and then returns the date."

f.write(Explain\_txt1)

print(count\_5y)

except:

print("\nError")

### =======================================================================

###

### Q3

###

sql\_select\_old1 = """SELECT FIRST\_NAME, LAST\_NAME, CONVERT(AGE, CHAR) AS AGE """

sql\_select\_old2 = """FROM EMPLOYEE """

sql\_select\_old\_male3 = """WHERE AGE = ALL(SELECT MAX(AGE) FROM EMPLOYEE WHERE SEX = 'M') AND SEX = 'M';"""

sql\_select\_old\_male = sql\_select\_old1 + "\n" + sql\_select\_old2 + "\n" + sql\_select\_old\_male3

cursor.execute(sql\_select\_old\_male)

results\_sal = cursor.fetchall()

for results\_M in results\_sal:

print('\nOldest Male : {}'.format(results\_M ))

f.write("\n\nQ3) Select the Oldest Male employee")

f.write("\n\nAnwers\n\n")

f.write(sql\_select\_old\_male)

M\_name = "\nMale employee name : " + results\_M[0] + " " + results\_M[1] + " Age : " + results\_M[2]

f.write(M\_name )

###

f.write("\n\n\nB) Query involving two database tables")

###

### Q4

###

sql\_select\_sal1 = """SELECT COUNT(\*), A.DEPT\_ID, B.DEPT\_NAME """

sql\_select\_sal2 = """FROM EMPLOYEE A, DEPARTMENT B """

sql\_select\_sal3 = """WHERE INCOME < ALL(SELECT avg(income) FROM employee) and a.DEPT\_ID = b.DEPT\_ID GROUP by DEPT\_ID;"""

sql\_query = sql\_select\_sal1 + sql\_select\_sal2 + sql\_select\_sal3

cursor.execute(sql\_query)

results\_sal = cursor.fetchall()

f.write("\n\nQ4) Give the count, department ID and department name of the employees drawing Less than the average salary")

f.write("\n\nAnwers\n\n")

f.write(sql\_query)

for r in results\_sal:

print('\nCount of Employees drawing less salary department-wise : {}'.format(str(r[0])))

try:

Salary\_less = "\n\nCount of employees drawing less salary than average : " + str(r[0]) + " Dept ID : " + \

r[1] + " Name : " + r[2]

f.write(Salary\_less)

except:

print("\nError")

###

### Q5

###

sql\_select\_leave1 = """SELECT CONCAT(A.FIRST\_NAME, " ", A.LAST\_NAME) AS NAME, B.LEAVE\_DAYS """

sql\_select\_leave2 = """FROM EMPLOYEE A, LEAVES B """

sql\_select\_leave3 = """WHERE A.EMP\_ID = B.EMP\_ID AND B.LEAVE\_DAYS = ALL(SELECT MAX(LEAVE\_DAYS) FROM LEAVES);"""

sql\_query = sql\_select\_leave1 + sql\_select\_leave2 + sql\_select\_leave3

cursor.execute(sql\_query)

results\_L = cursor.fetchone()

print('\nEmployee who take max leave : {}'.format(results\_L[0]))

try:

f.write("\n\nQ5) Give the count of employees who availed leave")

f.write("\n\nAnwers\n\n")

f.write(sql\_query)

Name\_leave = "\n\nEmployee name who took max. leave : " + results\_L[0]

f.write(Name\_leave)

except:

print("\nError")

# disconnect from server

cursor.close()

f.close()

### =======================================================================

**set3\_Q.py**

# -\*- coding: utf-8 -\*-

"""

Created on Fri Oct 30 12:08:56 2020

@author: PVS

"""

###

### Set 1

###

import mysql.connector, sys

from datetime import datetime

folder\_name = r'E:\DSE-MOCK-INTERVIEW'

sys.path.insert(0, folder\_name)

now = datetime.now()

current\_time = now.strftime("%H:%M:%S")

from db\_Class\_220621 import Employee

error = None

emp = Employee("localhost","testUser", "A4min@123","hr" )

connection = emp.db\_connection()

cursor = connection.cursor()

file\_name\_out = 'SQL\_Test at ' + str(current\_time) + ".txt"

f = open("SQL\_test\_log\_SET3.txt", "w")

f.write(file\_name\_out)

f.write("\n== SET 3 ==\n")

###

f.write("\nA) Query involving a single database table\n")

###

### Q1

###

f.write("\nQ1) Give the count of employees having income between 20000 AND 40000\n")

sql\_query1 = """SELECT COUNT(\*) FROM employee WHERE INCOME BETWEEN 20000 AND 40000;"""

cursor.execute(sql\_query1)

results\_INC\_count = cursor.fetchone()

try:

INC\_count = "\nFemale employees count : " + str(results\_INC\_count[0])

print(INC\_count)

f.write(INC\_count)

Explain\_txt1 = "\nThe WHERE clause is used to compare the given value with the field value available in the database table."

f.write(Explain\_txt1)

Explain\_txt2 = "\nThe COUNT() function returns the number of records returned by a SELECT query.\n\n"

f.write(Explain\_txt2)

except:

print("\nError")

###

### Q2

###

f.write("\n\nQ2) Give the count of employees who will attain 60 years from 5 years from now \n")

sql\_query2 = """SELECT COUNT(\*) FROM EMPLOYEE WHERE AGE < 60 AND AGE + 5 > 60;"""

cursor.execute(sql\_query2)

results\_5y\_count = cursor.fetchone()

try:

count\_5y = "\nCount of Employees who will attain 60 years from 5 years from now : " + str(results\_5y\_count[0])

f.write(count\_5y)

except:

print("\nError")

### ========================================================================

###

### Q3

###

sql\_select\_old1 = """SELECT FIRST\_NAME, LAST\_NAME, CONVERT(AGE, CHAR) AS AGE """

sql\_select\_old2 = """FROM EMPLOYEE """

sql\_select\_old\_female3 = """WHERE AGE = ALL(SELECT MAX(AGE) FROM EMPLOYEE WHERE SEX = 'M') AND SEX = 'F';"""

sql\_select\_old\_female = sql\_select\_old1 + "\n" + sql\_select\_old2 + "\n" + sql\_select\_old\_female3

cursor.execute(sql\_select\_old\_female)

results\_F = cursor.fetchone()

print('\nOldest Male : {}'.format(results\_F))

try:

f.write("\n\nQ3) Select the Oldest Female employee")

F\_name = "\nFemale employee name : " + results\_F[0] + " " + results\_F[1] + " Age : " + results\_F[2]

f.write(F\_name )

except:

print("\nError")

f.write("\n\n\nB) Query involving two database tables")

###

### Q4

###

sql\_select\_sal1 = """SELECT COUNT(\*), A.DEPT\_ID, B.DEPT\_NAME """

sql\_select\_sal2 = """FROM EMPLOYEE A, DEPARTMENT B """

sql\_select\_sal3 = """WHERE INCOME < 20000 and a.DEPT\_ID = b.DEPT\_ID GROUP by DEPT\_ID;"""

sql\_query = sql\_select\_sal1 + sql\_select\_sal2 + sql\_select\_sal3

cursor.execute(sql\_query)

results\_sal = cursor.fetchall()

f.write("\n\nQ4) Give the count, department ID and department name of the employees drawing Less than 2000")

for r in results\_sal:

print('\nCount of Employees drawing less than 20000 as income department-wise : {}'.format(str(r[0])))

try:

Salary\_less = "\n\nCount of employees drawing less than 20k as income : " + str(r[0]) + " Dept ID : " + \

r[1] + " Name : " + r[2]

f.write(Salary\_less)

except:

print("\nError")

# disconnect from server

cursor.close()

f.close()

### ========================================================================