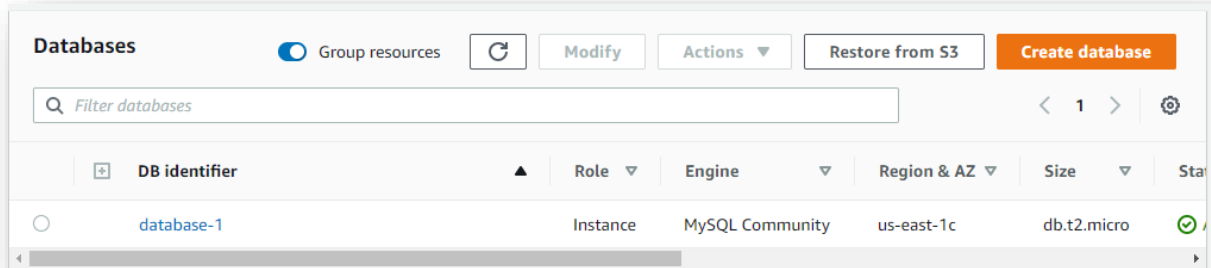
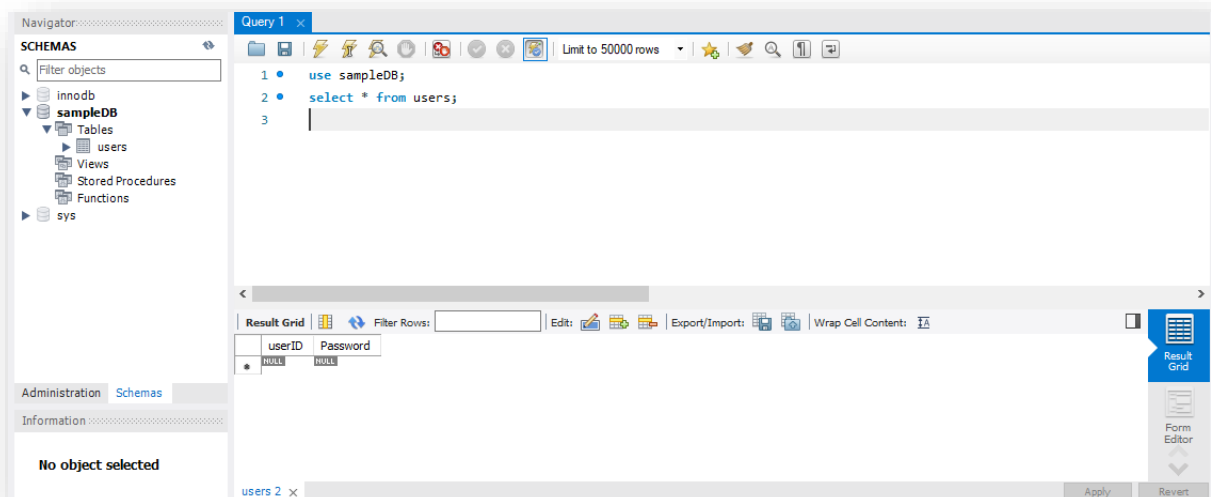


## Part C: AWS RDS database service experiment

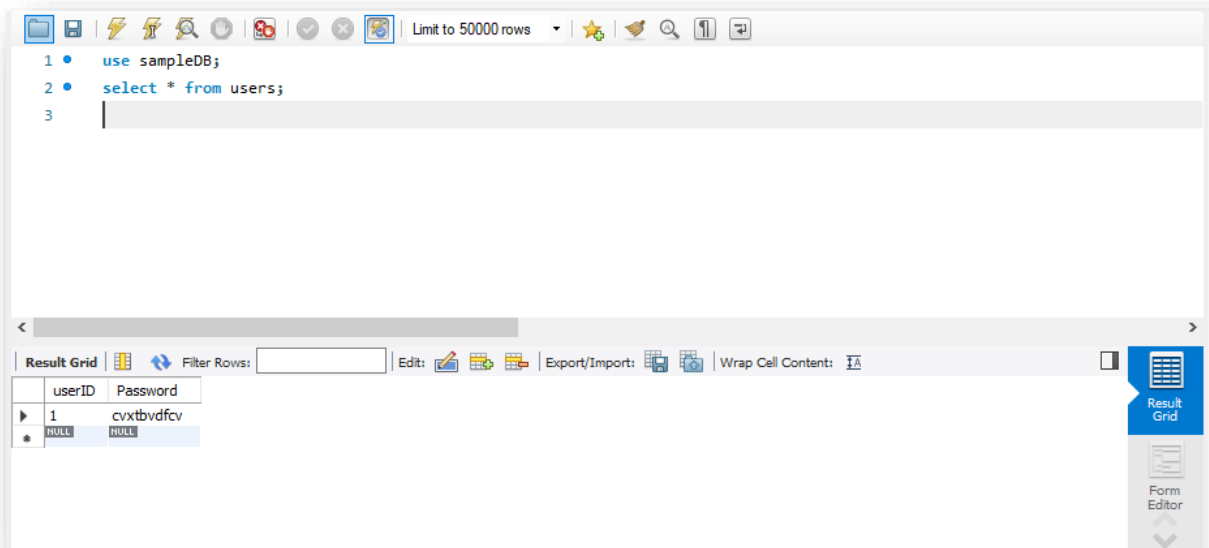
### a. Create MySQL DB instance using RDS



### b. Create single table database with two fields ('userID', 'Password')



c. Insert userID and Password into the database

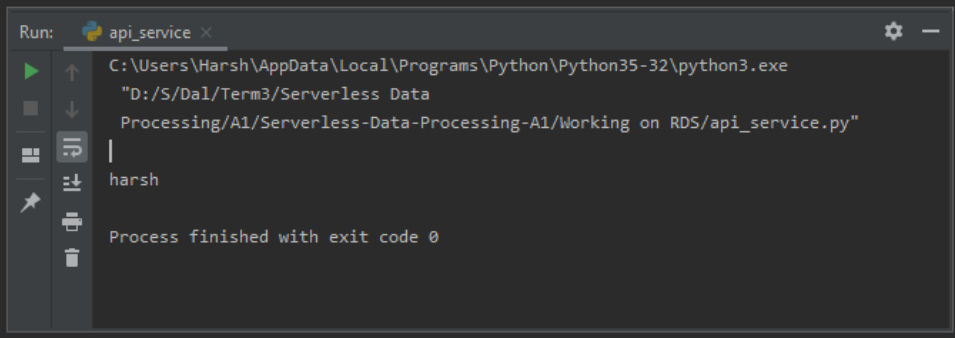


```
def addData(self,user_id,password):
    encrypted_password=generatePassword.encryptPassword(password) #encrypt_password
    object = self.connectRDS("database-1.ceofxtsta8i.us-east-1.rds.amazonaws.com", "sampleDB", "admin",
                             "amazonrds")
    cur = object.cursor()
    query = "INSERT INTO users (userID>Password) VALUES (" +str(user_id)+"," +encrypted_password+)"
    cur.execute(query)
    object.commit()
    cur.close()
    object.close()
```

d. Write a function to Retrieve password from given userID

```
def fetchData(self, userID):
    object = self.connectRDS("database-1.ceofxztsta8i.us-east-1.rds.amazonaws.com", "sampleDB", "admin",
                             "amazonrds")
    cur=object.cursor()
    query = "SELECT Password FROM users WHERE userID="+str(userID)
    cur.execute(query)
    print(generatePassword.decryptPassword(cur.fetchone()[0])) #get decrypted password
    object.commit()
    cur.close()
    object.close()

api = RdsApi()
#api.addData(1,"harsh")
api.fetchData(1)
```



```
Run: api_service x
C:\Users\Harsh\AppData\Local\Programs\Python\Python35-32\python3.exe
"D:/S/Dal/Term3/Serverless Data
Processing/A1/Serverless-Data-Processing-A1/Working on RDS/api_service.py"
|
harsh
Process finished with exit code 0
```

e. Python-mysql connector library for integration of RDS with python

```
import mysql.connector
import generatePassword
```

f. The password must be encrypted before inserting and decrypted after fetching.  
The 'Lookup5410.txt' file is uploaded on S3 bucket and then downloaded back on the local system.

### a. File upload:

first-bucket-csci-5410

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

US East (N. Virginia)

Viewing 1 to 1

Name	Last modified	Size	Storage class
Lookup5410.txt	May 23, 2020 2:40:11 AM GMT-0300	150.0 B	Standard

Viewing 1 to 1

```
""" Upload a file """
def fileUpload(self, bucket_name, source_file_name, file_name):
    s3Resource = boto3.resource('s3')
    s3Resource.Object(bucket_name, source_file_name).upload_file(Filename=file_name)
    print('file uploaded')
```

### b. File download

```
""" Downloading a file from the bucket """
def downloadFile(self, bucket_name, obj_name, dest_file_name):
    s3Client = boto3.client('s3')
    s3Client.download_file(bucket_name, obj_name, dest_file_name)
    print('file downloaded')

s3 = s3Api()
s3.downloadFile("first-bucket-csci-5410", "Lookup5410.txt", "Lookup.txt")
```

### c. Password encryption and decryption function

```
def encryptPassword(password):
    password=password.lower()
    f = open('Lookup.txt')
    string=''
    for i in password:
        f.seek(0,0)
        for j in f.readlines():
            if(i==j[0]):
                j=j.rstrip()
                string+=j[2:]
    return string

def decryptPassword(password):
    f = open('Lookup.txt')
    string=''
    count=0
    while(count<len(password)):
        encryptedLetter=str(password[count])+str(password[count+1])
        count+=2
        f.seek(0,0)
        for j in f.readlines():
            j = j.rstrip()
            if(encryptedLetter==j[2:]):
                string+=j[0]
    return string
```

**Note:** The function for uploading the lookup file to the s3 bucket and downloading it to the local system is provided in the source code of **Part B**

# Source code

## api\_service.py

```
import mysql.connector
import generatePassword

class RdsApi:

    def connectRDS(self,host,database,user,password):
        return mysql.connector.connect(
            host=host,
            database=database,
            user=user,
            password=password)

    def addData(self,user_id,password):
        encrypted_password=generatePassword.encryptPassword(password) #encrypt password
        object = self.connectRDS("database-1.ceofxztsta8i.us-east-1.rds.amazonaws.com", "sampleDB",
"admin", "amazonrds")
        cur = object.cursor()
        query = 'INSERT INTO users (userID>Password) VALUES
({0},{1})'.format(str(user_id),encrypted_password)
        cur.execute(query)
        object.commit()
        cur.close()
        object.close()

    def fetchData(self,userID):
        object = self.connectRDS("database-1.ceofxztsta8i.us-east-1.rds.amazonaws.com", "sampleDB",
"admin",
                                "amazonrds")
        cur=object.cursor()
        query = "SELECT Password FROM users WHERE userID="+str(userID)
        cur.execute(query)
        print(generatePassword.decryptPassword(cur.fetchone()[0]))    #get decrypted password
        object.commit()
        cur.close()
        object.close()

api = RdsApi()
api.addData(1,"harsh")
api.fetchData(1)
```

## generatePassword.py

```
def encryptPassword(password):
    password=password.lower()
    f = open('Lookup.txt')
    string=""
    for i in password:
        f.seek(0,0)
        for j in f.readlines():
            if(i==j[0]):
                j=j.rstrip()
                string+=j[2:]
    return string

def decryptPassword(password):
    f = open('Lookup.txt')
    string=""
    count=0
    while(count<len(password)):
        encryptedLetter=str(password[count])+str(password[count+1])
        count+=2
        f.seek(0,0)
        for j in f.readlines():
            j = j.rstrip()
            if(encryptedLetter==j[2:]):
                string+=j[0]
    return string
```

## References

- [1]*Pynative.com*, 2020. [Online]. Available: <https://pynative.com/python-mysql-database-connection/>. [Accessed: 25- May- 2020].