# Session 1

1. Check directory

dir

1. Đi vào thư mục

cd <folder\_name>

1. Thoát ra thư mục

cd ..

1. Short key cho #

Ctrl + /

1. Input

**VD: Với data dạng string**

n = input()

m = input()

d = n + m

print(d)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python hello.py

hello

hoho

hellohoho

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python hello.py

3

4

34

**VD: Với data dạng integer**

n = input()

m = input()

d = int(n) + int(m)

print(d)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python hello.py

3

4

7

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python hello.py

50

50

100

yob = input()

age = 2020 - int(yob)

print(age)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

2010

10

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

1975

45

yob = input("enter your yob here")

age = 2020 - int(yob)

print(age)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

enter your yob here1996

24

yob = input("enter your yob here")

age = 2020 - int(yob)

print("this is your age")

print(age)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

enter your yob here2000

this is your age

20

yob = input("enter your yob here")

age = 2020 - int(yob)

print("this is your age", age)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

enter your yob here2002

this is your age 18

yob = input("enter your yob here")

age = 2020 - int(yob)

print("you are", age, "years old")

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python yob.py

enter your yob here2001

you are 19 years old

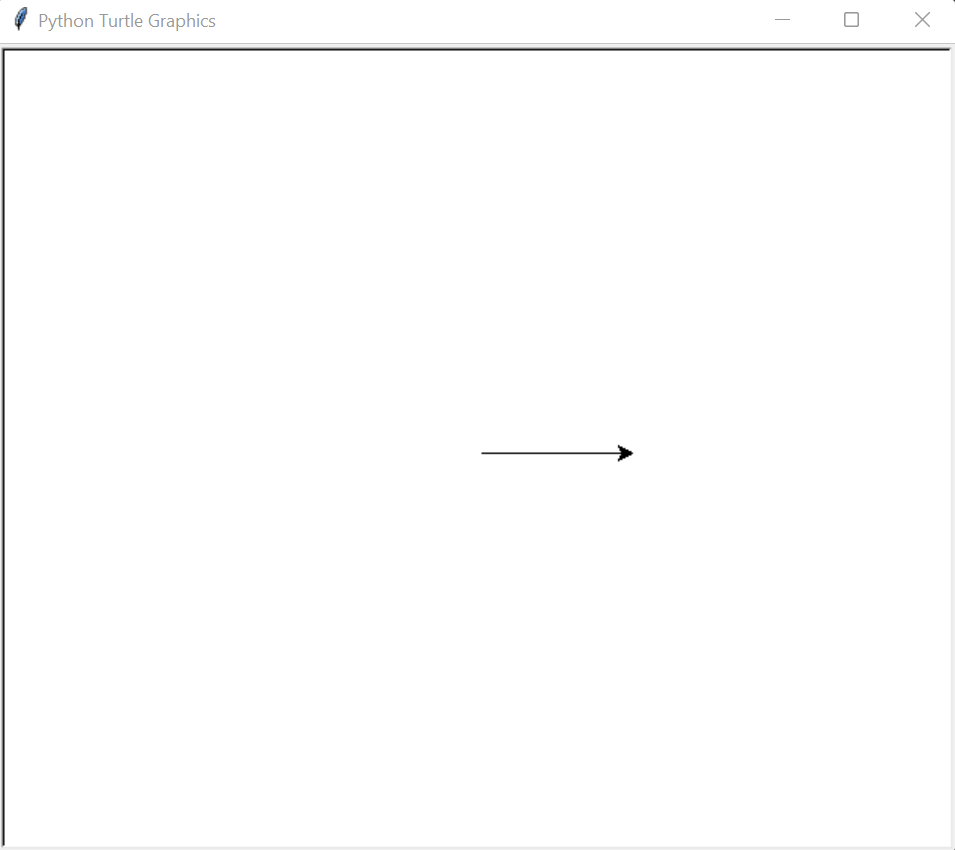
1. Draw

from turtle import \*

forward(100)

mainloop()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session1> python draw.py

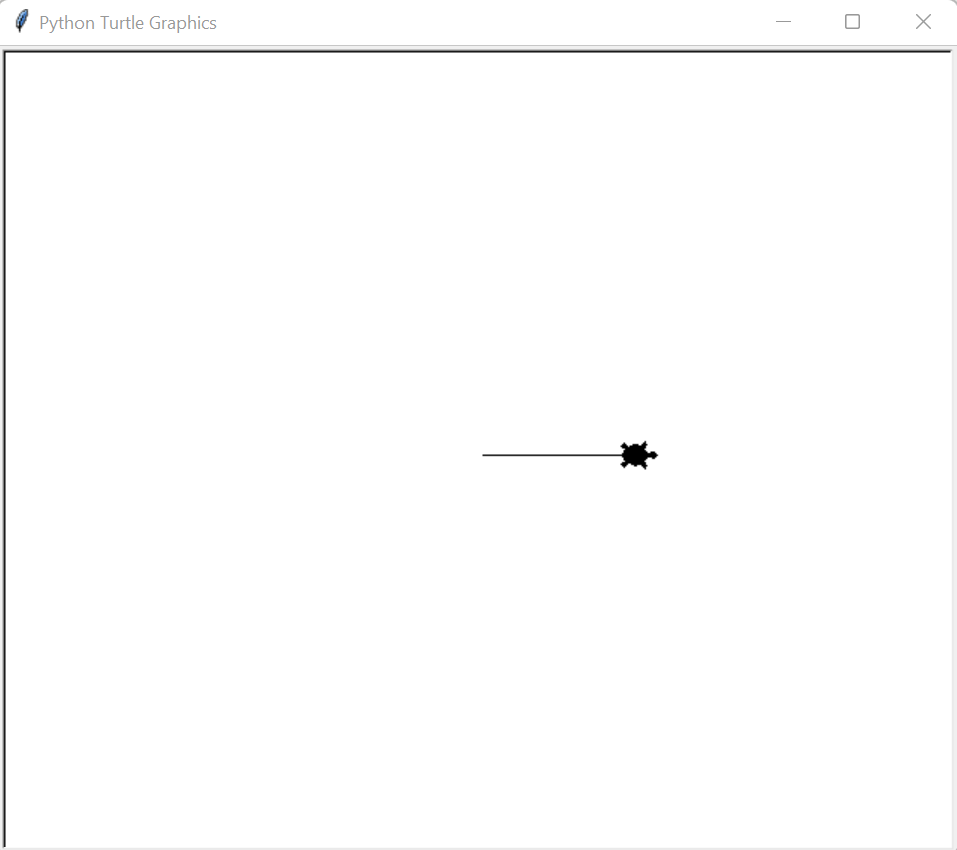


from turtle import \*

shape("turtle")   # Hinh dang dau mui ten

forward(100)

mainloop()



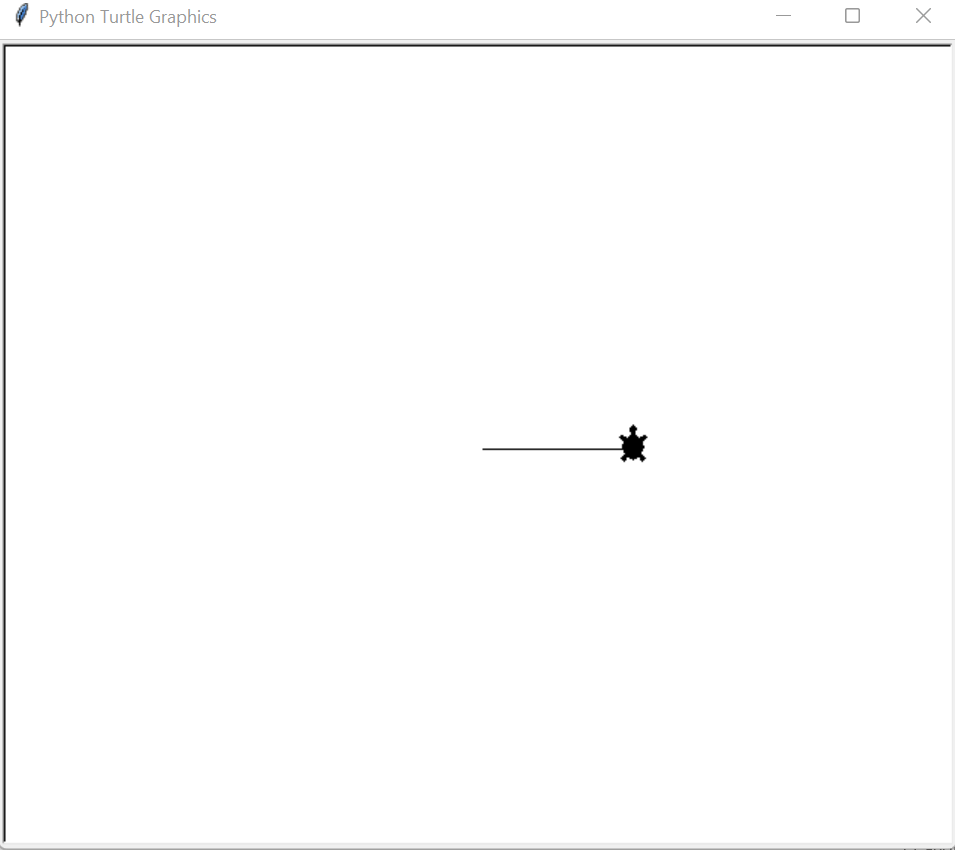
from turtle import \*

shape("turtle")

forward(100)

left(90)

mainloop()



from turtle import \*

shape("turtle")

forward(100)

left(90)

forward(100)

left(90)

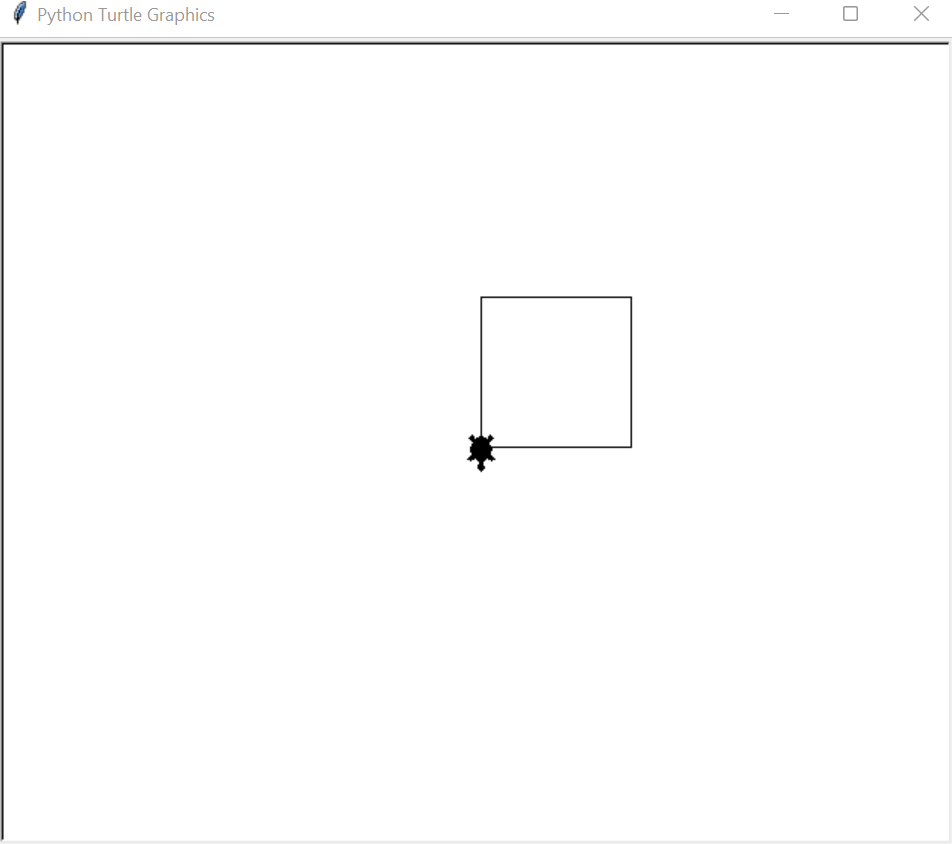
forward(100)

left(90)

forward(100)

left(90)

mainloop()



from turtle import \*

shape("turtle")

speed(-1)

forward(100)

left(90)

forward(100)

left(90)

forward(100)

left(90)

forward(100)

left(90)

left(30)

forward(100)

left(90)

forward(100)

left(90)

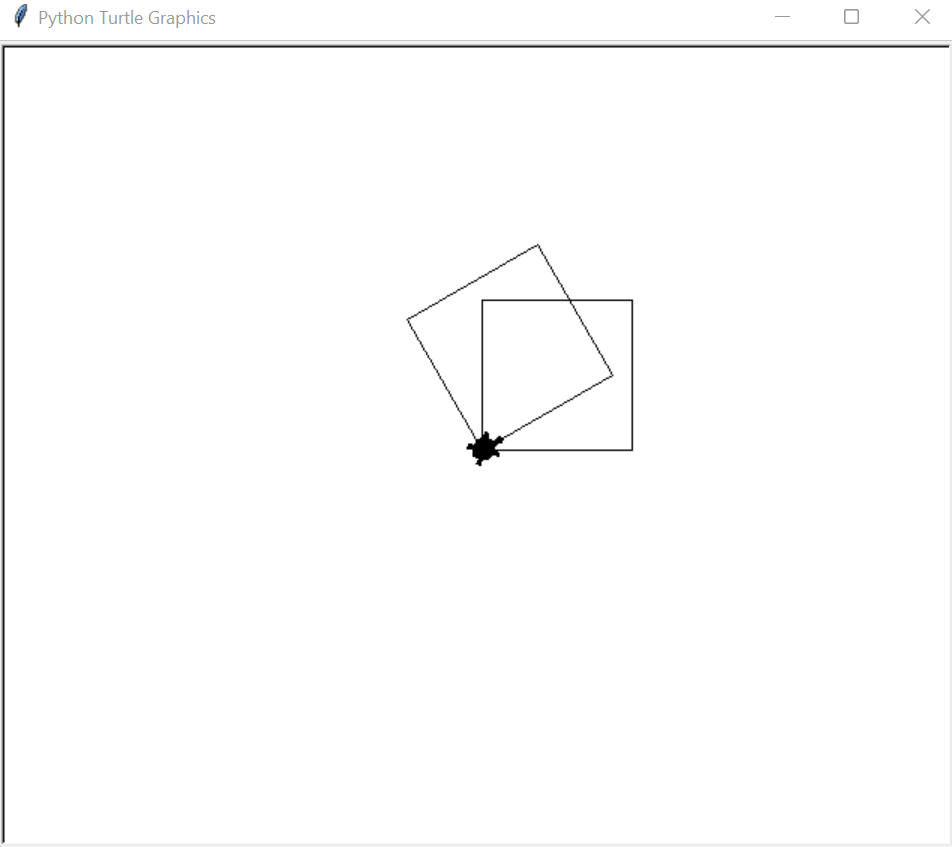
forward(100)

left(90)

forward(100)

left(90)

mainloop()



from turtle import \*

shape("turtle")

speed(-1)

for i in range(8):

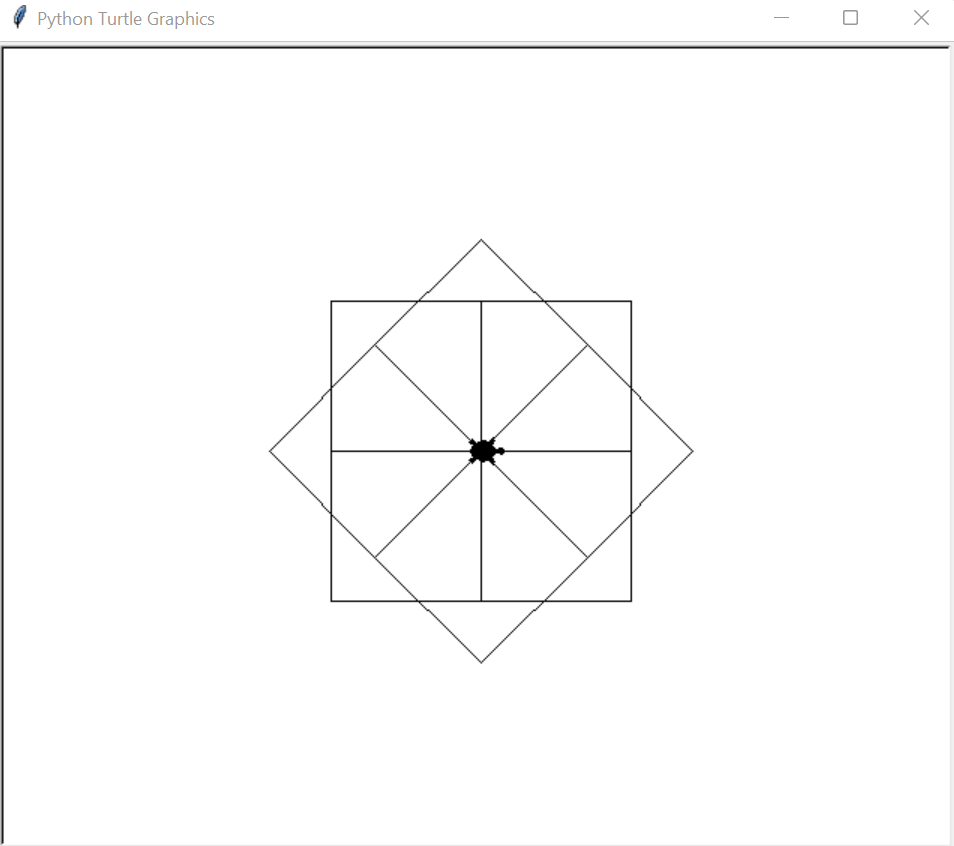
    for i in range(4):

        forward(100)

        left(90)

    left(45)

mainloop()



# Session 2

1. Lặp lại

for i in range(5):

    print('Hello')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

Hello

Hello

Hello

Hello

Hello

1. In các biến trong dãy
   1. for in in range(…):

for i in range(5):

    print(i)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

0

1

2

3

4

print(\*range(5))

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

0 1 2 3 4

print(\*range(1, 5))

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

1 2 3 4

print(\*range(1, 10, 2))   # Start, Stop, Step

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

1 3 5 7 9

print(\*range(1, 10, 3))

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

1 4 7

**BT: In các số chạy từ 0 -> 25**

for i in range(26):

    print(i)

**BT: In các số chẵn chạy từ 0 -> 100**

for i in range(0, 101, 2):

    print(i)

**BT: In các số chạy ngược lại từ 100 -> 1**

for i in range(100, 0, -1):

    print(i)

* 1. for i in […]

for i in ['a', 'b', 'c']:

print(i)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python .\loop\_intro.py

a

b

c

**BT: Tính tổng các số từ 1 đến n.**

n = int(input('enter a number'))

total = 0

for i in range(n+1):

    total = total + i

print(total)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python total.py

enter a number3

6

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python total.py

enter a number5

15

1. Vẽ hình với turtle

from turtle import \*

for i in range(3):

    forward(100)

    left(360/3)

for i in range(4):

    forward(100)

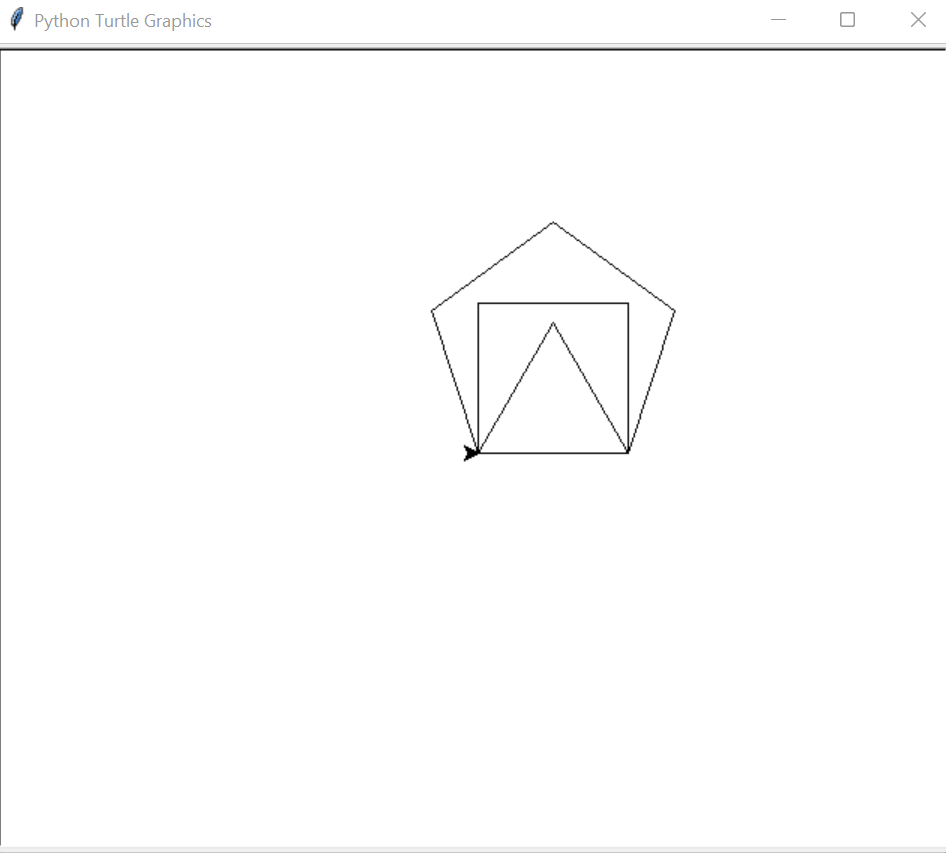
    left(360/4)

for i in range(5):

    forward(100)

    left(360/5)

mainloop()



from turtle import \*

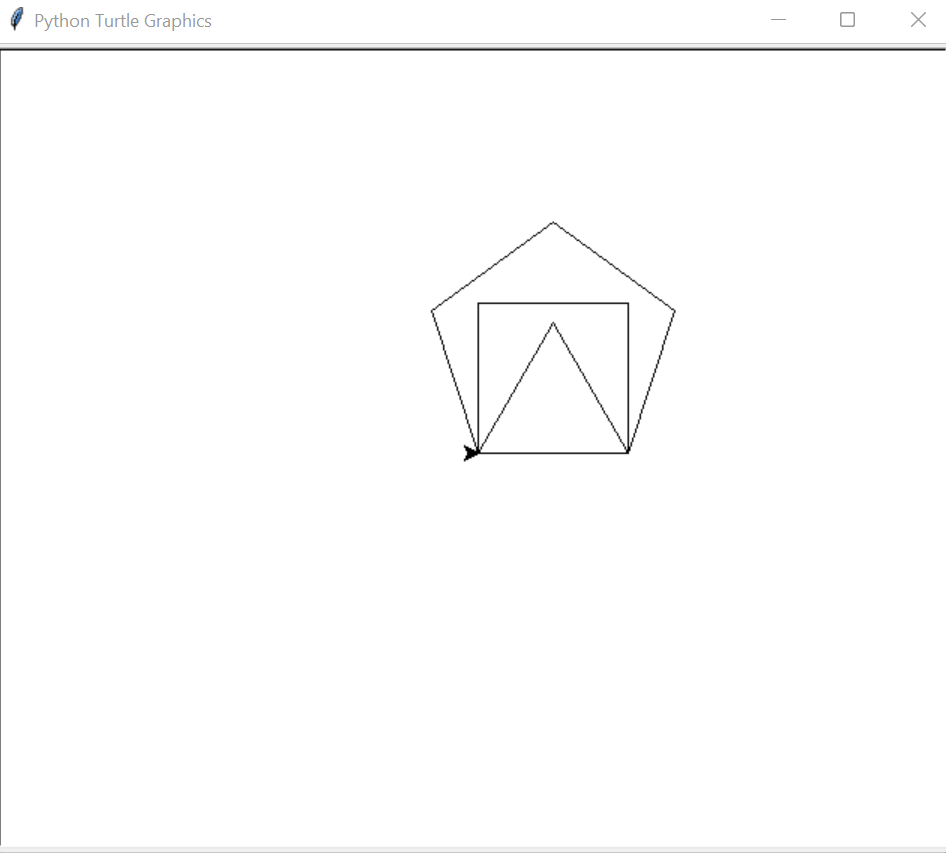
for edge in range(3, 6):   # Neu khong dat start = 3, edge se chay tu 0

    for i in range(edge):

        forward(100)

        left(360/edge)

mainloop()



from turtle import \*

n = int(input('enter number of edge '))

speed(-1)

for edge in range(3, n+1):

    for i in range(edge):

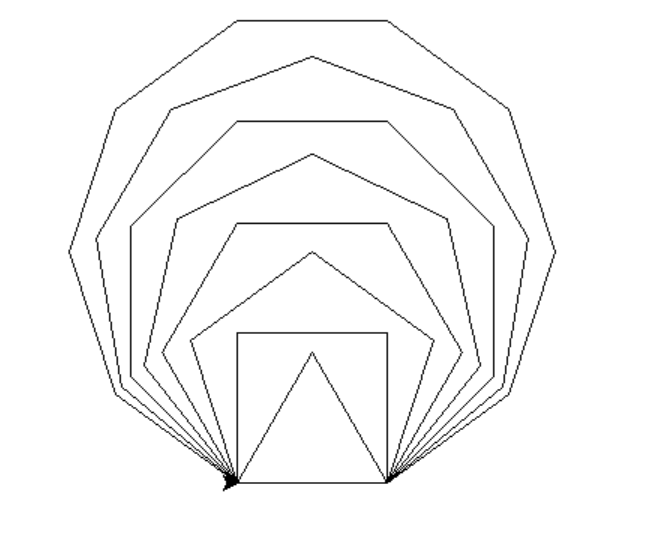
        forward(100)

        left(360/edge)

mainloop()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python rua.py

enter number of edge 10



1. Conditional

age = 14

if age > 18:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

You are not allowed here

age = 20

if age > 18:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

Welcome

age = 20

print(age > 18)

if age > 18:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

True

Welcome

age = 14

print(age > 18)

if age > 18:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

False

You are not allowed here

age = 20

print(age > 18)

if True:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

True

Welcome

age = 15

print(age > 18)

if False:

    print('Welcome')

else:

    print('You are not allowed here')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

False

You are not allowed here

score = 8

if score > 9:

    print('hs giỏi')

elif score > 7:

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs khá

score = 9.5

if score > 9:

    print('hs giỏi')

elif score > 7:

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs giỏi

score = 4

if score > 9:

    print('hs giỏi')

elif score > 7:

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs bình thường

*Note: 3 điều kiện* ***if, elif, else 🡪*** *Khi đã đạt điều kiện đầu tiên, python sẽ in ra cái đầu tiên*

score = 4

if score < 9:

    print('hs giỏi')

elif score < 7:

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs giỏi

score = 4

if score < 9:

    print('hs giỏi')

if score < 7:

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs giỏi

hs khá

*🡪 Trong trường hợp này, biến* ***if*** *đã được tách riêng chạy 2 lần nên python sẽ in cả 2 kết quả.*

score = 4

att = 'good'

if score > 9 and att == 'good':

    print('hs giỏi')

elif score > 7 or att == 'good':

    print('hs khá')

else:

    print('hs bình thường')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python conditional.py

hs khá

a = int(input('a= '))

b = int(input('b= '))

c = int(input('c= '))

delta = b\*\*2 - 4\*a\*c

print(delta)

if delta < 0:

    print('no x')

elif delta == 0:

    result = (-b) / (2\*a)

    print(result)

else:

    x1 = (-b + delta\*\*(1/2)) / (2\*a)

    x2 = (-b - delta\*\*(1/2)) / (2\*a)

    print('x1=', x1, 'x2=', x2)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python math.py

a= 1

b= 2

c= 1

0

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python math.py

a= 1

b= 5

c= 6

1

x1= -2.0 x2= -3.0

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python math.py

a= 1

b= 2

c= 3

-8

no x

1. Lặp lại với lệnh while, break, count
   1. Lặp lại liên tục

while True:   # Khi TRUE, in ra "you can't stop me"

    print("you can't stop me")   # Nhan to hop phim Ctrl C de dung lai

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

* 1. Chỉ chạy kết quả 1 lần

while True:

    print("you can't stop me")

    break   # chi in ket qua ra mot lan dung lai o lenh break

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

you can't stop me

* 1. Đếm số lần lặp lại count

count = 0

while True:

    if count == 5:

        break

    print("you can't stop me")

    count = count + 1

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

***Note*: Kết quả tương tự với lệnh for in in range (5)**

for i in range (5):

    print("you can't stop me")

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

Note: Khi dùng False để kiểm soát vòng while, python sẽ chạy nốt thêm một lần khi nhận được lệnh False 🡪 6 dòng thay vì 5 dòng

count = 0

running = True

while running:

    if count == 5:

        running = False

    print("you can't stop me")

    count = count + 1

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

you can't stop me

**BT: Viết code cho phép đăng nhập trong vòng 8 lần với username = ‘mindx’ và password = ‘password’**

from getpass import getpass

username = 'mindx'

password = 'password'

count = 0

while True:

    if count > 7:

        print('hết lần thử rồi bạn eii')

        break

    username\_input = input('username = ')

    password\_input = input('password = ')

    if username\_input == username and password\_input == password:

        print('Welcome')

        break

    else:

        count += 1

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

username = minds

password = dasdad

username = mkdsfs

password = geger

username = asdasda

password = adssadasd

username = adssadsa

password = rtetete

username = aadsad

password = dfsdfdfsd

username = sfdffd

password = sfsdfdfd

username = sfddsfsdf

password = sfdfdf

username = sfdfds

password = sdffsfd

hết lần thử rồi bạn eii

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session2> python loop\_intro.py

username = mindx

password = password

Welcome

# Session 3

1. Kiểu dữ liệu list []

monan1 = 'phở'

monan2 = 'bún chả'

monan3 = 'trứng rán'

monan4 = 'thịt chó'

monan5 = 'cơm'

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print(monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

***Note*: Có thể chứa nhiều kiểu dữ liệu khác nhau trong list**

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print(monan[0])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

phở

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print(monan[-1])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

cơm

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print(monan[-4])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

bún chả

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

for i in range (5):

    print(monan[i])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

phở

bún chả

trứng rán

thịt chó

cơm

1. Độ dài của list: len()

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

for i in range (len(monan)):

    print(monan[i])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

phở

bún chả

trứng rán

thịt chó

cơm

1. Thêm biến mới cho list: <tên list>.append()

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

monan.append('bơ')   # CREATE

for i in range (len(monan)):

    print(monan[i])   # READ

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

phở

bún chả

trứng rán

thịt chó

cơm

bơ

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

a = input()

monan.append(a)   # CREATE

for i in range (len(monan)):

    print(monan[i])   # READ

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

hihi

phở

bún chả

trứng rán

thịt chó

cơm

hihi

1. Sửa list: <tên list>[i] =

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó']

monan[0] = 'cơm'   # UPDATE

print(monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

['cơm', 'bún chả', 'trứng rán', 'thịt chó']

1. Kiểm tra vị trí của biến trong list: <tên list>.index()

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print(monan.index('trứng rán'))

print(monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

2

['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

1. Kiểm tra sự tồn tại của biến trong list

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó', 'cơm']

print('trứng' in monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

False

BT: Cho phép người dùng sửa list menu

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó']

for i in range(len(monan)):

    print(i+1, monan[i])

update\_value = input('nhập tên món ăn muốn đổi: ')

index = monan.index(update\_value)

monan[index] = input('nhập tên món ăn mới: ')

print(monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

1 phở

2 bún chả

3 trứng rán

4 thịt chó

nhập tên món ăn muốn đổi: bún chả

nhập tên món ăn mới: bún đậu mắm tép

['phở', 'bún đậu mắm tép', 'trứng rán', 'thịt chó']

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó']

for i in range(len(monan)):

    print(i+1, monan[i])

update\_value = input('nhập tên món ăn muốn đổi: ')

if update\_value in monan:

    index = monan.index(update\_value)

    monan[index] = input('nhập tên món ăn mới: ')

    print(monan)

else:

    print('Không tìm thấy món đó đâu mannn')

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

1 phở

2 bún chả

3 trứng rán

4 thịt chó

nhập tên món ăn muốn đổi: cơm

Không tìm thấy món đó đâu mannn

1. Xóa một phần tử trong list: <tên list>.pop(i); <tên list>.remove(‘ ‘)

monan = ['phở', 'bún chả', 'trứng rán', 'thịt chó']

for i in range(len(monan)):

    print(monan[i])   # READ

monan[0] = 'cơm'   # UPDATE

print(monan)

monan.pop(0)   # DELETE by index

print(monan)

monan.remove('bún chả')   # DELETE by value

print(monan)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session3> python menu.py

phở

bún chả

trứng rán

thịt chó

['cơm', 'bún chả', 'trứng rán', 'thịt chó']

['bún chả', 'trứng rán', 'thịt chó']

['trứng rán', 'thịt chó']

***Note*: Thứ tự các phần tử trong list bị xáo trộn, độ dài của list ngắn lại**

# Session 4

1. Kiểu dữ liệu dictionary (object, map,…) {}

# dictionary, object, map

person = {

    'name': 'Đạt',

    'yob': 96,

    'company': 'Viettel'

}

print(person)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

{'name': 'Đạt', 'yob': 96, 'company': 'Viettel'}

person = {

    'name': 'Đạt',

    'yob': 96,

    'company': 'Viettel',

    'key': None

}

name = person['name']

print(person['name'])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

Đạt

person = {

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None

}

name = person['name']

print(person['yob']['year'])

print(len(person['company']))

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

1996

2

1. Đọc, tạo, update, xóa dữ liệu trong Dictionary

person = {

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None

}

name = person['name']

for key in person:   # READ

    print(key)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

name

yob

company

key

person = {

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None

}

name = person['name']

for key in person:   # READ

    print(person[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

Đạt

{'year': 1996, 'month': 1, 'day': 1}

['Viettel', 'Vinaphone']

None

person = {

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None

}

name = person['name']

# for key in person:

#     print(person[key])

for key in person:   # READ

    print(key, person[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

name Đạt

yob {'year': 1996, 'month': 1, 'day': 1}

company ['Viettel', 'Vinaphone']

key None

***Note*: Nếu có hai key giống nhau, python sẽ lấy key cuối cùng làm kết quả cuối. VD với key ‘name’: ‘Đạt’ và key ‘name’: ‘Đức’ sau đây:**

person = {

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None,

    'name': 'Đức'

}

name = person['name']

for key in person:   # READ

    print(key, person[key])

person['relationship'] = True   # CREATE

person['yob']['month'] = 4   # UPDATE

del person['key']   # DELETE

for key in person:   # READ

    print(key, person[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

name Đức

yob {'year': 1996, 'month': 1, 'day': 1}

company ['Viettel', 'Vinaphone']

key None

name Đức

yob {'year': 1996, 'month': 4, 'day': 1}

company ['Viettel', 'Vinaphone']

relationship True

person = {

    'name': 'Đức',

    'name': 'Đạt',

    'yob': {

        'year': 1996,

        'month': 1,

        'day': 1

    },

    'company': ['Viettel', 'Vinaphone'],

    'key': None

}

name = person['name']

for key in person:   # READ

    print(key, person[key])

person['relationship'] = True   # CREATE

person['yob']['month'] = 4   # UPDATE

del person['key']   # DELETE

for key in person:   # READ

    print(key, person[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python person.py

name Đạt

yob {'year': 1996, 'month': 1, 'day': 1}

company ['Viettel', 'Vinaphone']

key None

name Đạt

yob {'year': 1996, 'month': 4, 'day': 1}

company ['Viettel', 'Vinaphone']

relationship True

1. Tra từ trong dictionary

teen\_code = {

    'hc': 'học',

    'nc': 'nói chuyện',

    'vk': 'vũ khí',

    'ck': 'chuyển khoản',

}

while True:

    for key in teen\_code:

        print(key, end='\t')

    print()

    print('\*'\*30)

    input\_key = input('enter the word you want to search: ')

    if input\_key in teen\_code:

        print('it`s mean: ', teen\_code[input\_key])

    else:

        prompt = input('word not found, would u like to contribute it`s meaning? ')

        if prompt == 'y':

            teen\_code[input\_key] = input('enter meaning: ')

        else:

            print('bye')

            break

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python teencode.py

hc nc vk ck

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

enter the word you want to search: ck

it`s mean: chuyển khoản

hc nc vk ck

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

enter the word you want to search: cltmca

word not found, would u like to contribute it`s meaning? y

enter meaning: có làm thì mới có ăn

hc nc vk ck cltmca

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

enter the word you want to search: cltmca

it`s mean: có làm thì mới có ăn

hc nc vk ck cltmca

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

enter the word you want to search: sdadas

word not found, would u like to contribute it`s meaning? no

bye

1. Đếm kí tự count = count + 1

from itertools import count

sentence = 'adsewwfsfdgery'

count\_dictionary = {}

for char in sentence:

    if char in count\_dictionary:

        count\_dictionary[char] = count\_dictionary[char] + 1

    else:

        count\_dictionary[char] = 1

print(count\_dictionary)

for key in count\_dictionary:

    print(key, count\_dictionary[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python count\_letter.py

{'a': 1, 'd': 2, 's': 2, 'e': 2, 'w': 2, 'f': 2, 'g': 1, 'r': 1, 'y': 1}

a 1

d 2

s 2

e 2

w 2

f 2

g 1

r 1

y 1

1. Sắp xếp lại kí tự sorted()

Cách 1:

sentence = 'adsewwfsfdgery'

sorted\_sentence = sorted(sentence)

count\_dictionary = {}

for char in sentence:

    if char in count\_dictionary:

        count\_dictionary[char] = count\_dictionary[char] + 1

    else:

        count\_dictionary[char] = 1

print(count\_dictionary)

for key in count\_dictionary:

    print(key, count\_dictionary[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python count\_letter.py

{'a': 1, 'd': 2, 's': 2, 'e': 2, 'w': 2, 'f': 2, 'g': 1, 'r': 1, 'y': 1}

a 1

d 2

s 2

e 2

w 2

f 2

g 1

r 1

y 1

Cách 2:

sentence = 'adsewwfsfdgery'

count\_dictionary = {}

for char in sentence:

    if char in count\_dictionary:

        count\_dictionary[char] = count\_dictionary[char] + 1

    else:

        count\_dictionary[char] = 1

items = count\_dictionary.items()

print(sorted(items))

for key in count\_dictionary:

    print(key, count\_dictionary[key])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python count\_letter.py

[('a', 1), ('d', 2), ('e', 2), ('f', 2), ('g', 1), ('r', 1), ('s', 2), ('w', 2), ('y', 1)]

a 1

d 2

s 2

e 2

w 2

f 2

g 1

r 1

y 1

sentence = 'adsewwfsfdgery'

count\_dictionary = {}

for char in sentence:

    if char in count\_dictionary:

        count\_dictionary[char] = count\_dictionary[char] + 1

    else:

        count\_dictionary[char] = 1

key\_value\_list = sorted(count\_dictionary.items())

for key\_value in key\_value\_list:

    print(key\_value)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python count\_letter.py

('a', 1)

('d', 2)

('e', 2)

('f', 2)

('g', 1)

('r', 1)

('s', 2)

('w', 2)

('y', 1)

sentence = 'adsewwfsfdgery'

count\_dictionary = {}

for char in sentence:

    if char in count\_dictionary:

        count\_dictionary[char] = count\_dictionary[char] + 1

    else:

        count\_dictionary[char] = 1

key\_value\_list = sorted(count\_dictionary.items())

for key\_value in key\_value\_list:

    print(key\_value[0], key\_value[1])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python count\_letter.py

a 1

d 2

e 2

f 2

g 1

r 1

s 2

w 2

y 1

1. Bóc tách dữ liệu

Đóng rồi mở từng tầng để xem key

Kiểu dữ liệu là gì? List/ dictionary

movies = data["results"]

for movie in movies:

    print(movie['title'])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

Joker

The Old Man & the Gun

Spider-Man: Far from Home

Rambo: Last Blood

movies = data["results"]

input\_pop = int(input('enter the popularity '))

for movie in movies:

    if movie['popularity'] > input\_pop:

        print(movie['title'])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

enter the popularity 500

Joker

movies = data["results"]

input\_pop = int(input('enter the year of the movie '))

for movie in movies:

    year = movie['release\_date'].split('-')

    print(year)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

enter the year of the movie 2019

['2019', '10', '04']

['2018', '09', '28']

['2019', '07', '02']

['2019', '09', '20']

movies = data["results"]

input\_pop = int(input('enter the year of the movie '))

for movie in movies:

    year = int(movie['release\_date'].split('-')[0])

    print(year)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

enter the year of the movie 2019

2019

2018

2019

2019

movies = data["results"]

input\_year = int(input('enter the year of the movie '))

for movie in movies:

    year = int(movie['release\_date'].split('-')[0])

    if year == input\_year:

        print(movie['title'])

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

enter the year of the movie 2019

Joker

Spider-Man: Far from Home

Rambo: Last Blood

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

enter the year of the movie 2018

The Old Man & the Gun

movies = data["results"]

total = 0

for movie in movies:

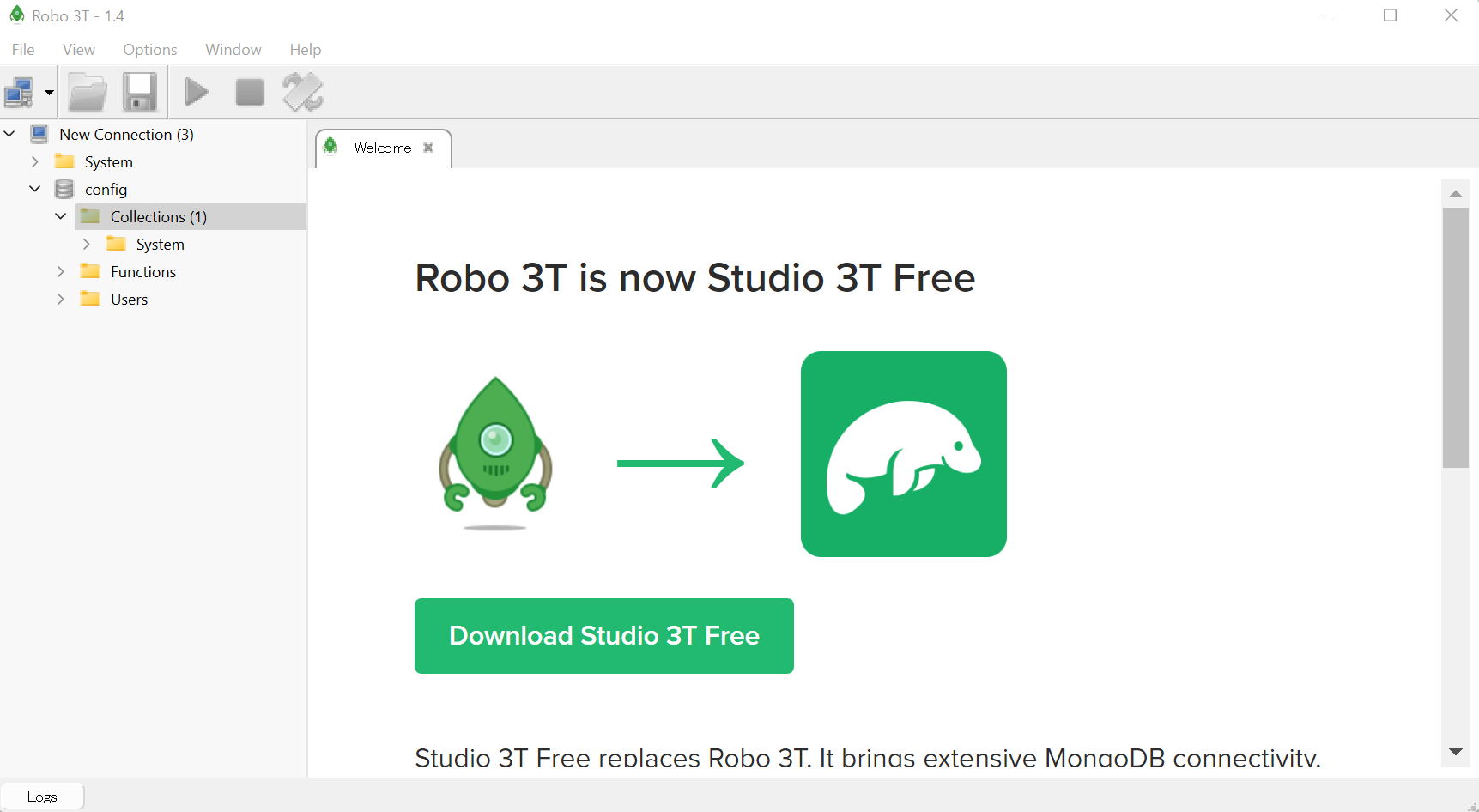
    total = total + movie['vote\_count']

print(total)

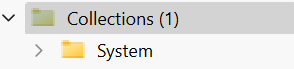
PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session4> python movie.py

5520

# Session 5



Database trong MongoDB 

Trong database có chứa collections, trong collections chứa các bản ghi (records) 

1. Cài đặt pymongo để import Mongo

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11> pip install pymongo

1. Kết nối python với MongoDB

from pymongo import MongoClient

client = MongoClient()   # Tao ra ket noi voi MongoDB qua bien client

db = client.get\_database('d4e11')   # Tu bien client/hoac nhung gi co san de tao ra database

collection = db.get\_collection('members')   # tu database da lay/tao ra de lay/tao ra collection

1. Tạo dữ liệu CREATE
   1. Tạo một bản ghi **collection.insert\_one({ })**

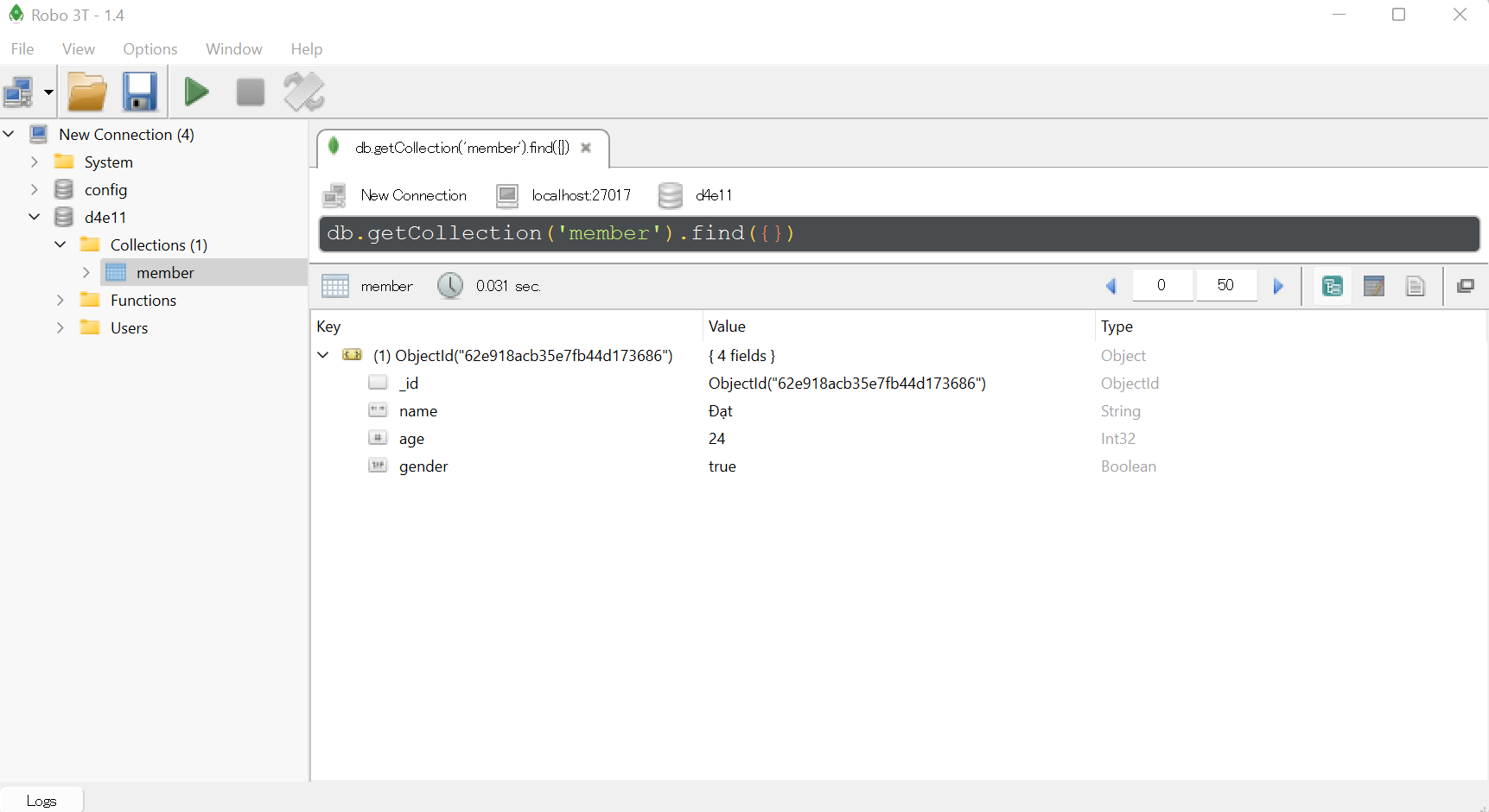
collection.insert\_one({   # CREATE

    'name': 'Đạt',

    'age': 24,

    'gender': True

})



* 1. Tạo nhiều bản ghi **collection.insert\_many([{ }])**

collection.insert\_many(

  [

    {

    'name': 'Đạt1',

    'age': 24,

    'gender': True

    },

    {

    'name': 'Đạt2',

    'age': 24,

    'gender': True

    },

    {

    'name': 'Đạt3',

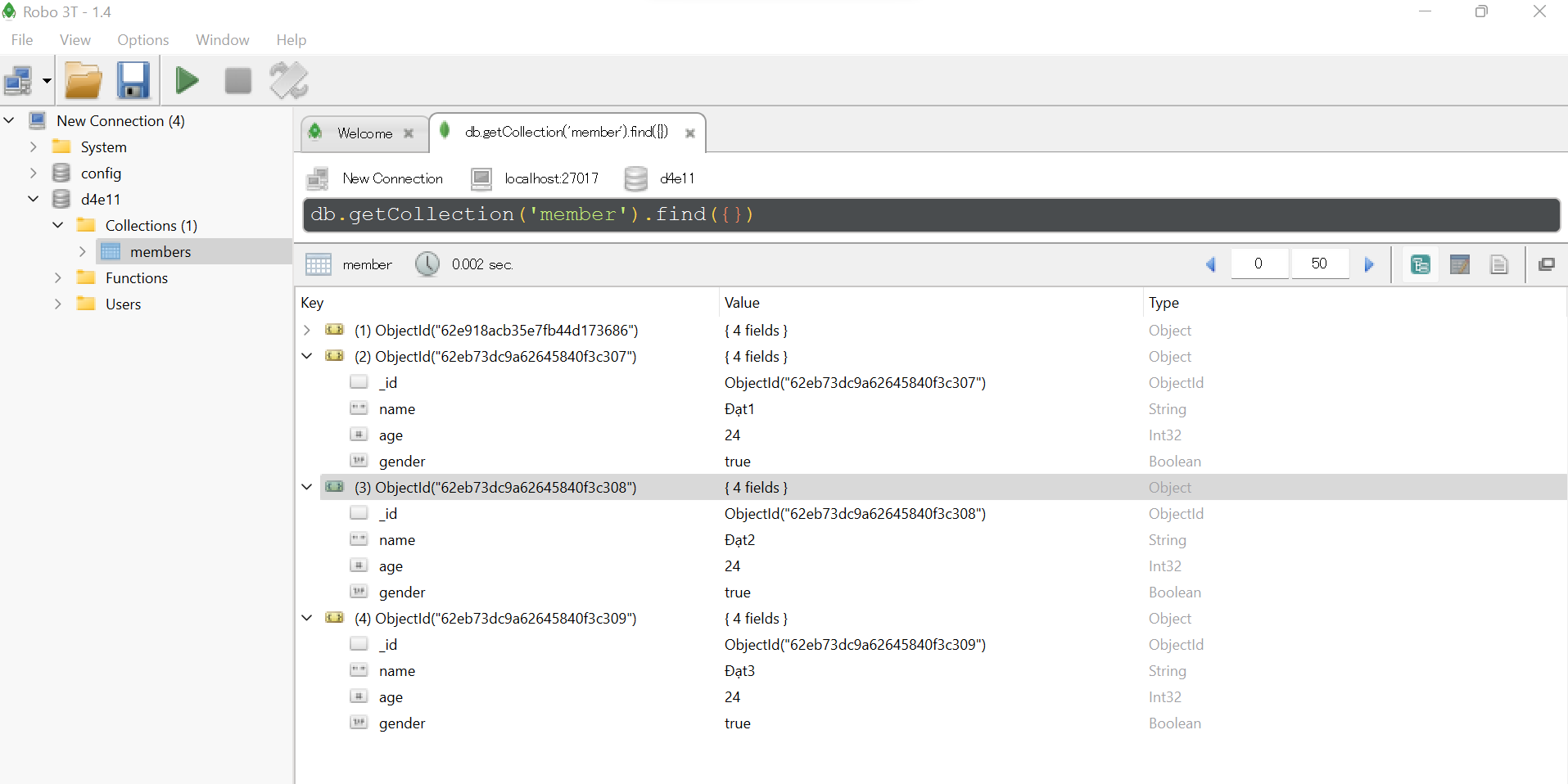
    'age': 24,

    'gender': True

    },

  ]

)

****

1. Truy vấn dữ liệu QUERY
   1. **collection.find ({ })**

members = collection.find()

for member in members:

    print(member)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python db\_intro.py

{'\_id': ObjectId('62e918acb35e7fb44d173686'), 'name': 'Đạt', 'age': 24, 'gender': True}

{'\_id': ObjectId('62eb73dc9a62645840f3c307'), 'name': 'Đạt1', 'age': 24, 'gender': True}

{'\_id': ObjectId('62eb73dc9a62645840f3c308'), 'name': 'Đạt2', 'age': 24, 'gender': True}

{'\_id': ObjectId('62eb73dc9a62645840f3c309'), 'name': 'Đạt3', 'age': 24, 'gender': True}

* 1. Tìm dữ liệu theo trường thông tin

from pymongo import MongoClient

client = MongoClient()

db = client.get\_database('d4e11')

collection = db.get\_collection('candidates')

data = {

  "data": [

    {

      "name": "Tiger Nixon",

      "hr": {

        "position": "System Architect",

        "salary": "$320,800",

        "start\_date": "2011/04/25"

      },

…

    {

      "name": "Donna Snider",

      "hr": {

        "position": "Customer Support",

        "salary": "$112,000",

        "start\_date": "2011/01/25"

      },

      "contact": [

        "New York",

        "4226"

      ]

    }

  ]

}

collection.insert\_many(data['data'])

members = collection.find({   # QUERY

  'hr.position': 'Accountant'

})

for member in members:

  print(member)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c1'), 'name': 'Garrett Winters', 'hr': {'position': 'Accountant', 'salary': '$170,750', 'start\_date': '2011/07/25'}, 'contact': ['Tokyo', '8422']}

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c4'), 'name': 'Airi Satou', 'hr': {'position': 'Accountant', 'salary': '$162,700', 'start\_date': '2008/11/28'}, 'contact': ['Tokyo', '5407']}

members = collection.find({   # QUERY

  'hr.position': 'Senior Javascript Developer'

})

for member in members:

  print(member)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c3'), 'name': 'Cedric Kelly', 'hr': {'position': 'Senior Javascript Developer', 'salary': '$433,060', 'start\_date': '2012/03/29'}, 'contact': ['Edinburgh', '6224']}

1. Đếm số lượng bản ghi chứa dữ liệu cần tìm **collection.count\_documents ({ })**

members = collection.count\_documents({

  'hr.position': 'Accountant'

})

print(members)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

2

members = collection.count\_documents({

  # 'hr.position': 'Accountant'

})

print(members)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

57

1. Truy vấn dữ liệu **KHÔNG PHẢI** là giá trị nào đấy **$ne**

members = collection.find({   # QUERY

  'hr.position': {'$ne': 'Accountant'}   # ne = not equal

})

for member in members:

  print(member)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c0'), 'name': 'Tiger Nixon', 'hr': {'position': 'System Architect', 'salary': '$320,800', 'start\_date': '2011/04/25'}, 'contact': ['Edinburgh', '5421']}

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c2'), 'name': 'Ashton Cox', 'hr': {'position': 'Junior Technical Author', 'salary': '$86,000', 'start\_date': '2009/01/12'}, 'contact': ['San Francisco', '1562']}

…

{'\_id': ObjectId('62eb81236fbc1f01cc91a1f8'), 'name': 'Donna Snider', 'hr': {'position': 'Customer Support', 'salary': '$112,000', 'start\_date': '2011/01/25'}, 'contact': ['New York', '4226']}

members = collection.find\_one({   # QUERY

  'hr.position': {'$ne': 'Accountant'}   # ne = not equal

})

print(members)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c0'), 'name': 'Tiger Nixon', 'hr': {'position': 'System Architect', 'salary': '$320,800', 'start\_date': '2011/04/25'}, 'contact': ['Edinburgh', '5421']}

members = collection.find\_one({   # QUERY

  'hrtio': 'asda'   # Du lieu khong ton tai

})

print(members)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

None

* 1. Sắp lại dữ liệu được truy vấn **collection.find ({ }).sort(‘ ‘: 1/-1)**

### **sort(‘ ‘: 1) từ bé đến lớn, A đến Z**

## SAP XEP DU LIEU TRUY VAN ##

members = collection.find({   # QUERY

}).sort('name', 1)

for member in members:

  print(member)

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c4'), 'name': 'Airi Satou', 'hr': {'position': 'Accountant', 'salary': '$162,700', 'start\_date': '2008/11/28'}, 'contact': ['Tokyo', '5407']}

…

{'\_id': ObjectId('62eb81236fbc1f01cc91a1f0'), 'name': 'Zorita Serrano', 'hr': {'position': 'Software Engineer', 'salary': '$115,000', 'start\_date': '2012/06/01'}, 'contact': ['San Francisco', '4389']}

### **sort(‘ ‘: -1) từ lớn đến bé, Z đến A**

members = collection.find({   # QUERY

}).sort('name', -1)

for member in members:

  print(member)

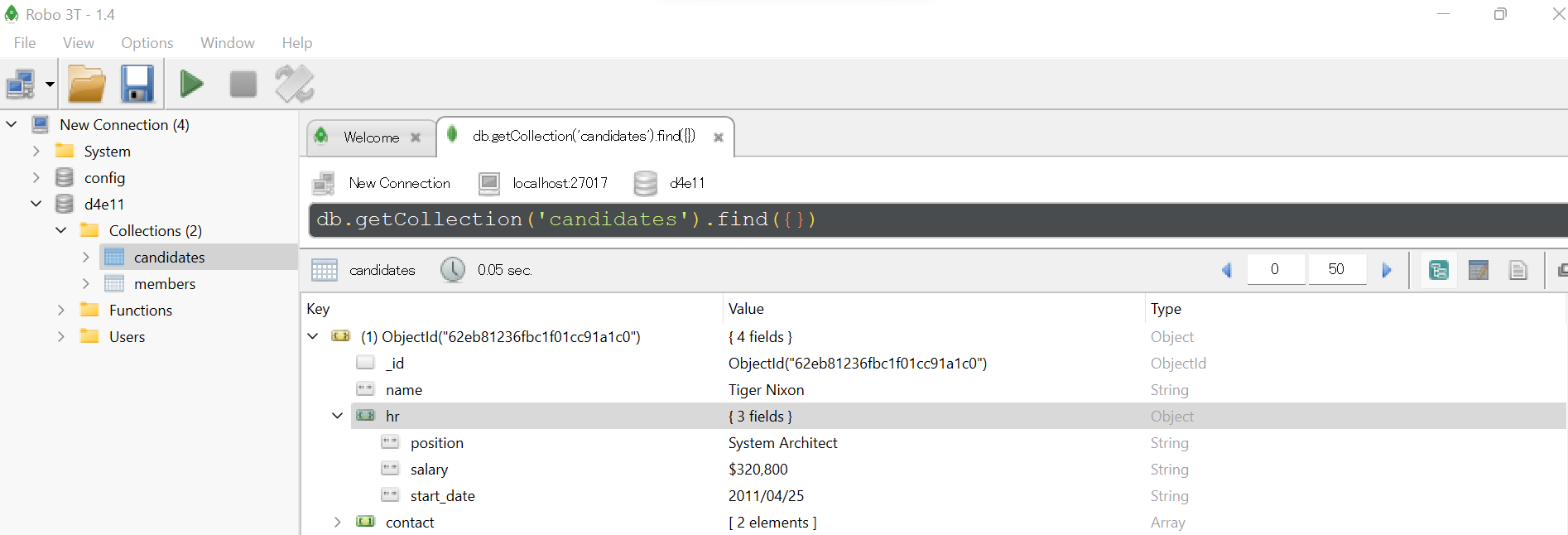
PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> python hr.py

{'\_id': ObjectId('62eb81236fbc1f01cc91a1f0'), 'name': 'Zorita Serrano', 'hr': {'position': 'Software Engineer', 'salary': '$115,000', 'start\_date': '2012/06/01'}, 'contact': ['San Francisco', '4389']}

…

{'\_id': ObjectId('62eb81236fbc1f01cc91a1c4'), 'name': 'Airi Satou', 'hr': {'position': 'Accountant', 'salary': '$162,700', 'start\_date': '2008/11/28'}, 'contact': ['Tokyo', '5407']}

1. Update bản ghi



## UPDATE DU LIEU BAN GHI ##

query = {'name': 'Tiger Nixon'}

update = {

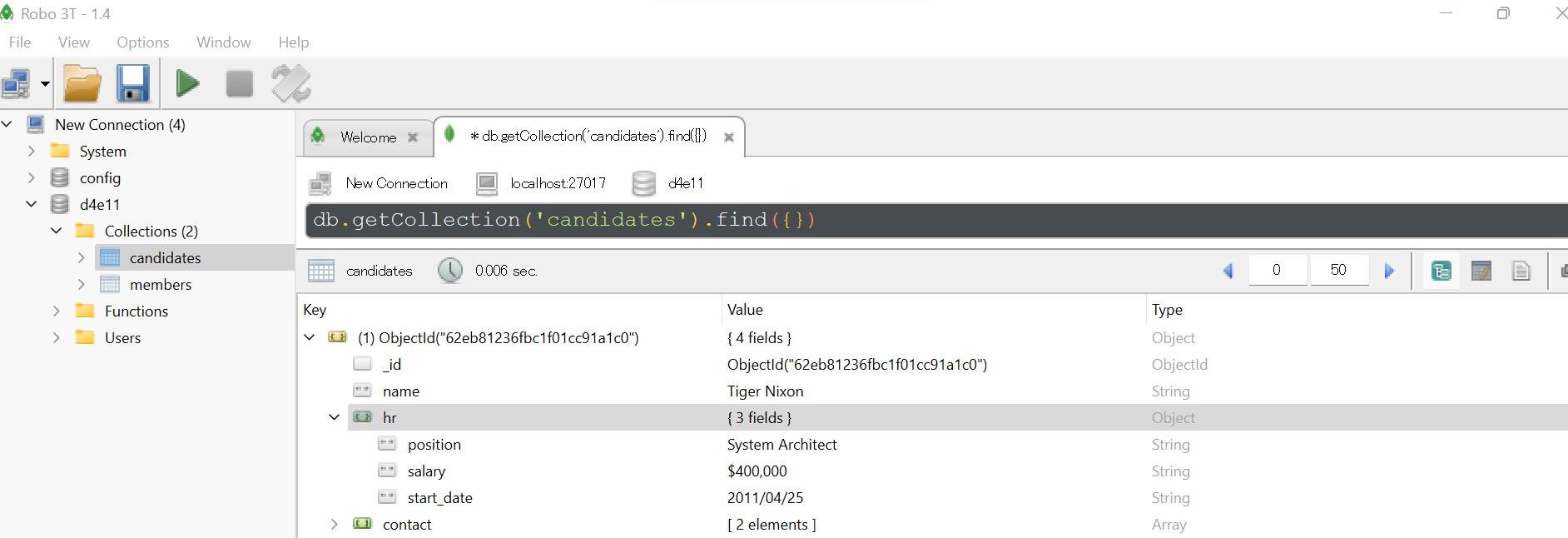
  '$set': {

    'hr.salary': '$400,000'

  }

}

members = collection.update\_one(query, update)



## THEM TRUONG THONG TIN KHI UPDATE ##

query = {'name': 'Tiger Nixon'}

update = {

  '$set': {

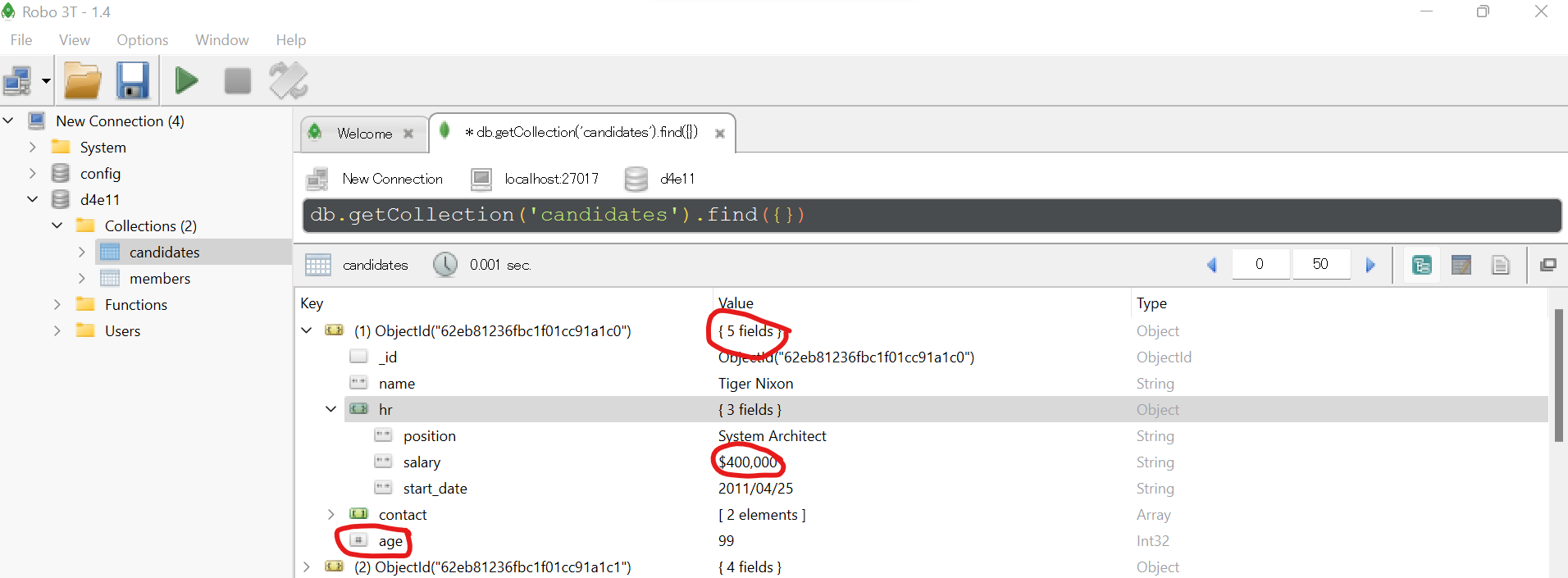
    'hr.salary': '$400,000',

    'age': 99

  }

}

members = collection.update\_one(query, update)



Xem hết session 5 part 3

1. Tải dữ liệu http về máy model requests

Cài đặt pip install requests

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session5> pip install requests

import requests

from pymongo import MongoClient

mongo\_client = MongoClient()

db = mongo\_client.get\_database('d4e11')

collection = db.get\_collection('police\_call')

client = requests.get('https://data.baltimorecity.gov/resource/xviu-ezkt.json')

data = client.json()

collection.insert\_many()

total\_false\_call = collection.count\_documents({'priority': 'Non-Emergency'})

total\_call = collection.count\_documents()

print(total\_false\_call \* 100/ total\_call)

import requests

from pymongo import MongoClient

mongo\_client = MongoClient()

db = mongo\_client.get\_database('d4e11')

collection = db.get\_collection('police\_call')

all\_district = collection.distinct('district')

count\_call\_by\_district = {}

for district in all\_district:

    count\_call\_by\_district[district] = collection.count\_documents({district}: district)

print(count\_call\_by\_district)

max = 0

name = ''

for key in count\_call\_by\_district:

    if count\_call\_by\_district[key] > max:

        max = count\_call\_by\_district[key]

        name = key

print(name, max)

# Session 6

**Mongo Practice**

## Query / Find Documents

query the movies collection to

1. get all documents
2. get all documents with writer set to "Quentin Tarantino"
3. get all documents where actors include "Brad Pitt"
4. get all documents with franchise set to "The Hobbit"
5. get all movies released in the 90s
6. get all movies released before the year 2000 or after 2010

## Update Documents

1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."
2. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."
3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

## Text Search

1. find all movies that have a synopsis that contains the word "Bilbo"
2. find all movies that have a synopsis that contains the word "Gandalf"
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"
4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"
5. find all movies that have a synopsis that contains the word "gold" and "dragon"

## Delete Documents

1. delete the movie "Pee Wee Herman's Big Adventure"
2. delete the movie "Avatar"

# Session 7

1. Cài đặt pymyslq để import MySQL

C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> pip install pymysql

1. Kết nối python với MySQL và tạo database

import pymysql

client = pymysql.connect(

    host='localhost',

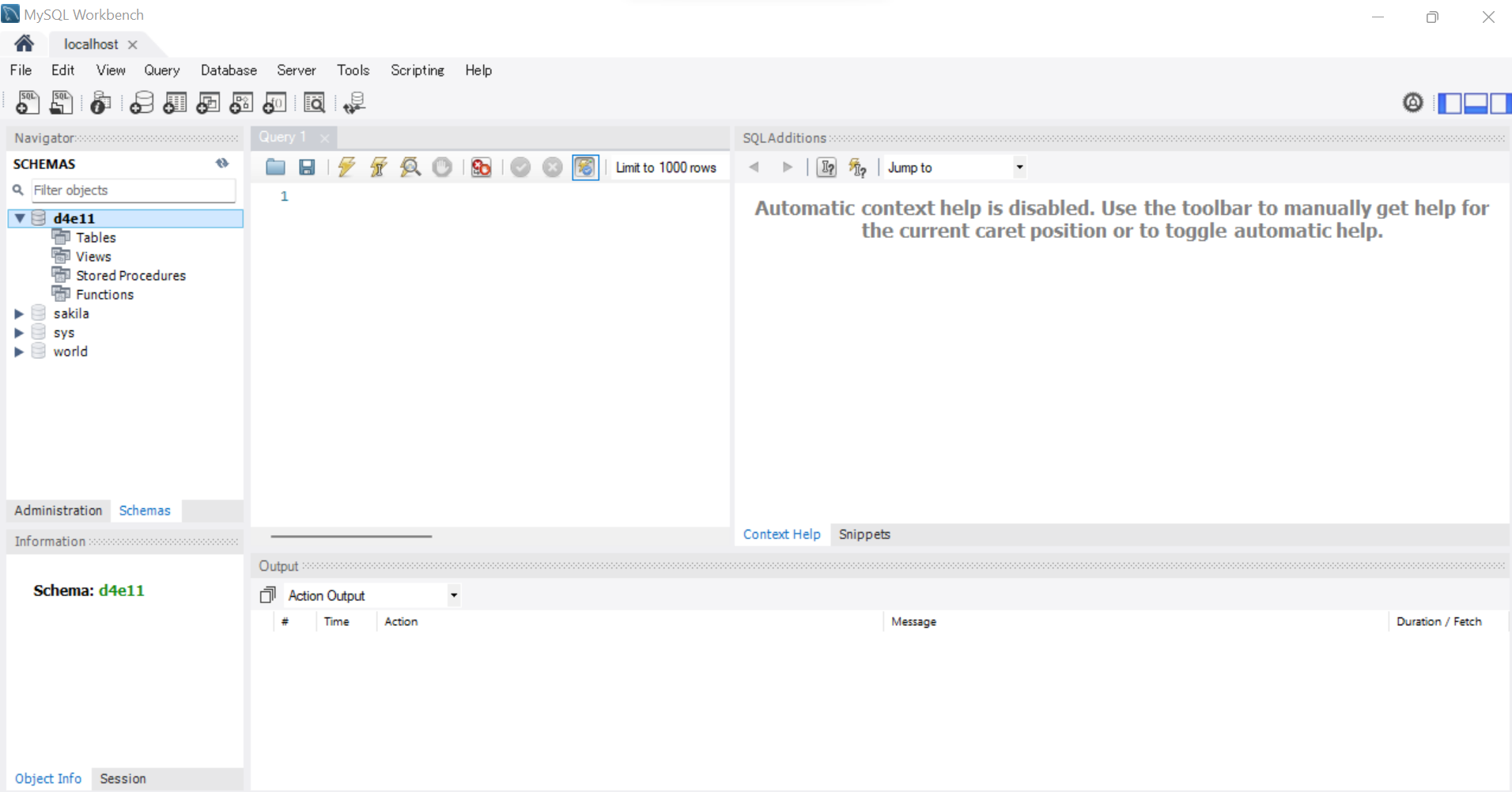
    user='root',

    password='Inuka4682!'

)

cursor = client.cursor()

cursor.execute('CREATE DATABASE d4e11')



1. Tạo bảng Table trong database

Note: Khi tạo bảng cần tạo cả id

## CREATE bang trong MySQL ##

cursor.execute('''

    CREATE TABLE d4e11.user (

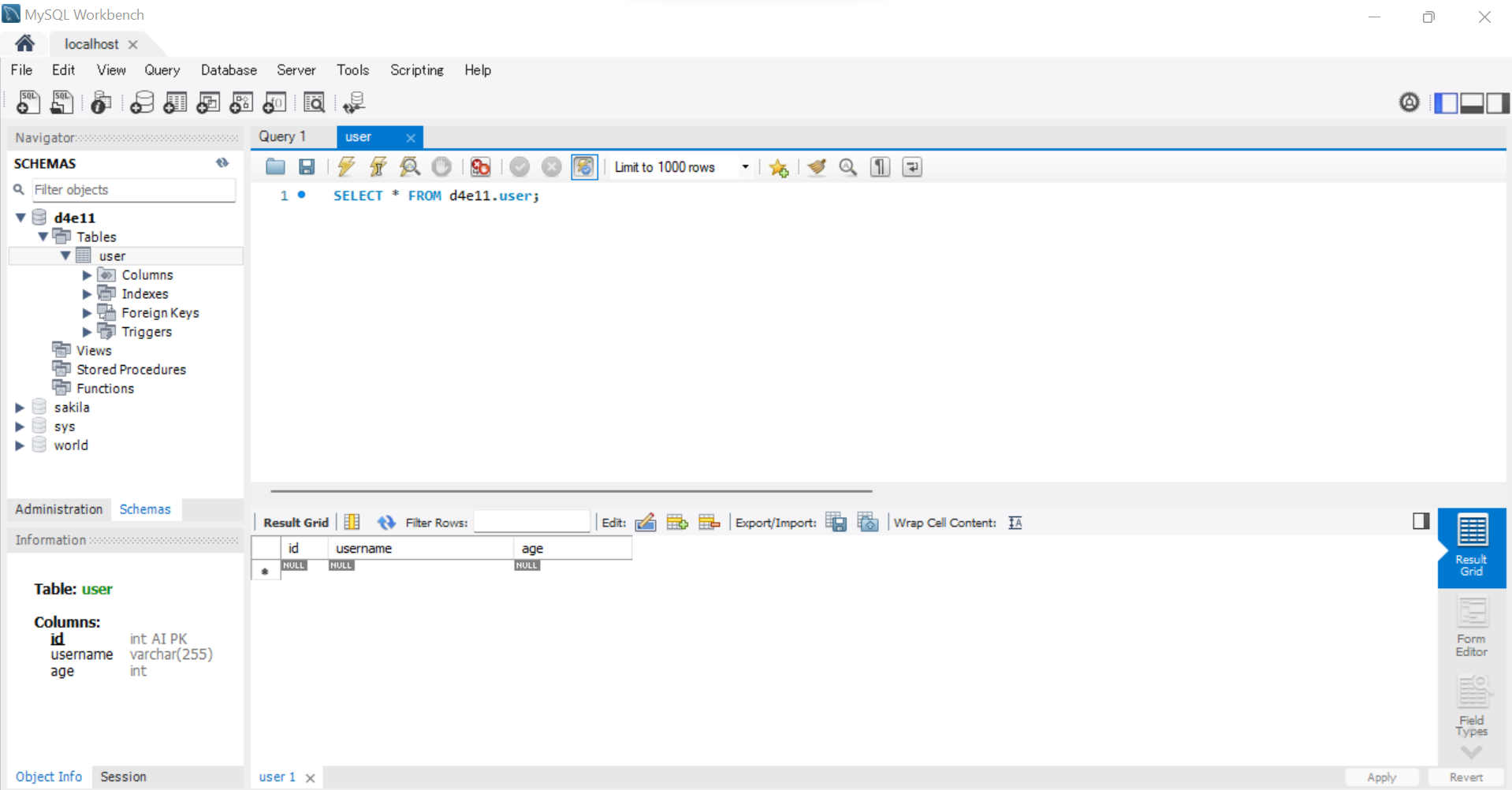
        id INT(11) AUTO\_INCREMENT PRIMARY KEY,

        username VARCHAR(255),

        age INT(11)

    )

''')



1. INSERT dữ liệu vào bảng

## INSERT du lieu vao bang ##

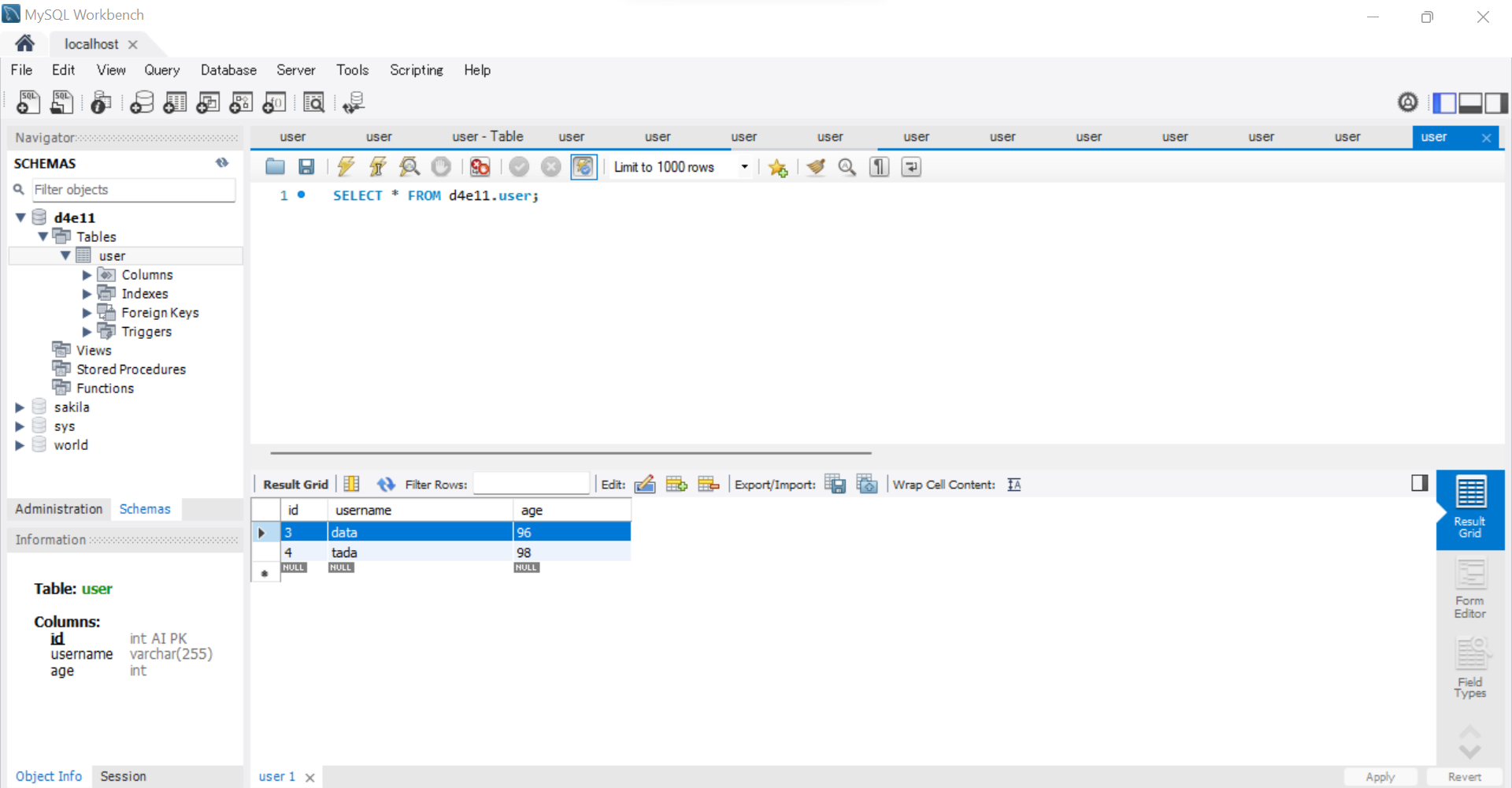
cursor.execute('''

INSERT INTO d4e11.user (username, age)

VALUES ('data', '96'), ('tada', '98');

''')

client.commit()



1. Lấy giữ liệu ra từ bảng MySQL
   1. cursor.fetchall() lấy tất cả dữ liệu đáp ứng nhu cầu

## Lay giu lieu ra tu MySQL ##

cursor.execute(''' SELECT \* FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

((3, 'data', 96), (4, 'tada', 98))

## Lay het tat ca du lieu trong bang ##

client = pymysql.connect(

    host='localhost',

    user='root',

    password='Inuka4682!',

    cursorclass=pymysql.cursors.DictCursor

)

cursor = client.cursor()

cursor.execute(''' SELECT \* FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'id': 3, 'username': 'data', 'age': 96}, {'id': 4, 'username': 'tada', 'age': 98}]

## Chi lay du lieu trong truong chi dinh ##

cursor.execute(''' SELECT username FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'username': 'data'}, {'username': 'tada'}]

cursor.execute(''' SELECT username, id FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'username': 'data', 'id': 3}, {'username': 'tada', 'id': 4}]

cursor.execute(''' SELECT SUM(id) FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'SUM(id)': Decimal('7')}]

cursor.execute(''' SELECT COUNT(id) FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'COUNT(id)': 2}]

cursor.execute(''' SELECT COUNT(id) AS count FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'count': 2}]

cursor.execute(''' SELECT MAX(age) AS max FROM d4e11.user ''')

data = cursor.fetchall()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

[{'max': 98}]

* 1. cursor.fetchone() chỉ lấy một dữ liệu đáp ứng nhu cầu

cursor.execute(''' SELECT \* FROM d4e11.user ''')

data = cursor.fetchone()

print(data)

client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

{'id': 3, 'username': 'data', 'age': 96}

cursor.execute('''SELECT MAX(age) AS max FROM d4e11.user''')

data = cursor.fetchone()

print(data)

print(data['max'])

client.commit()

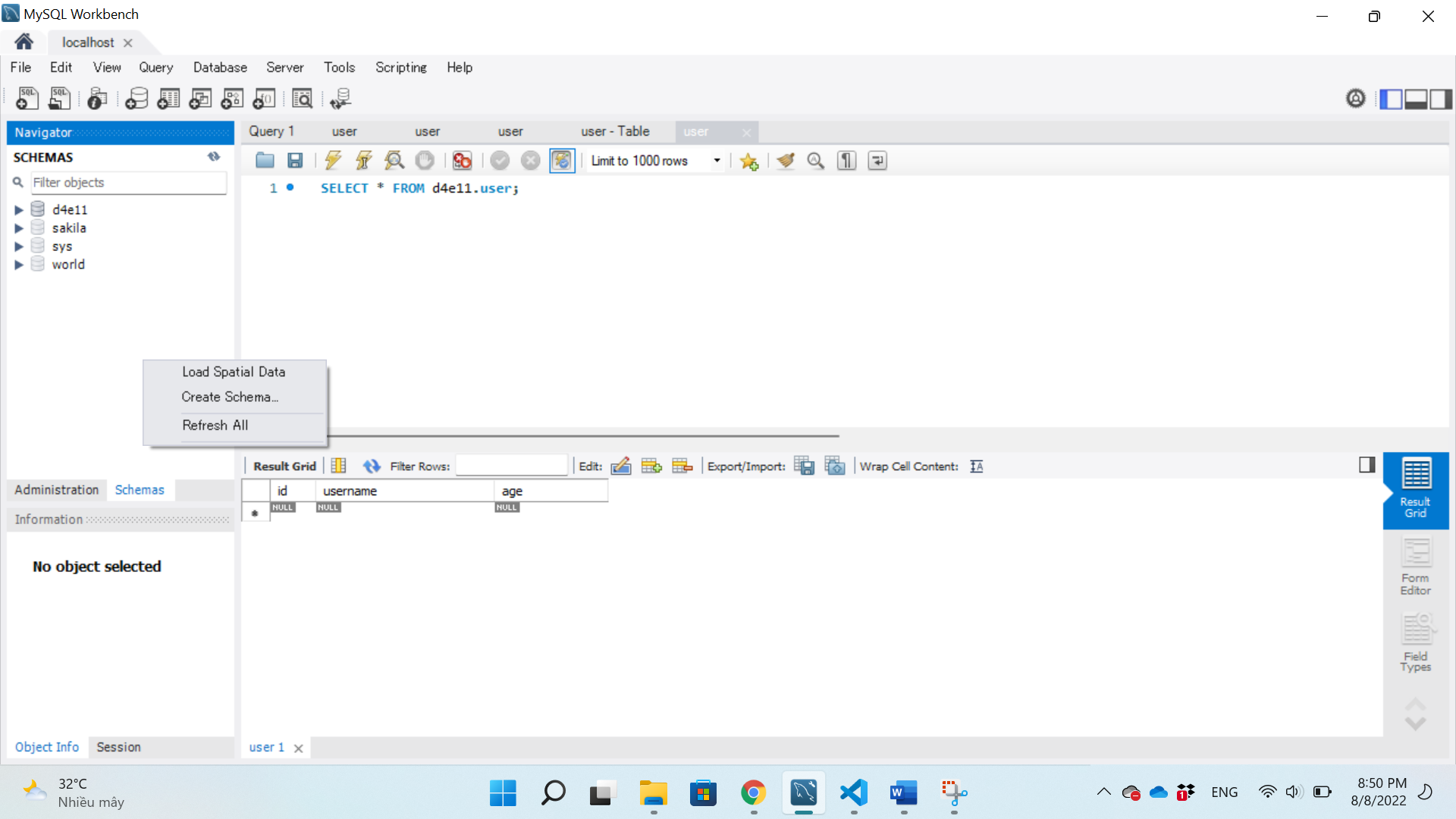
PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python mysql\_intro.py

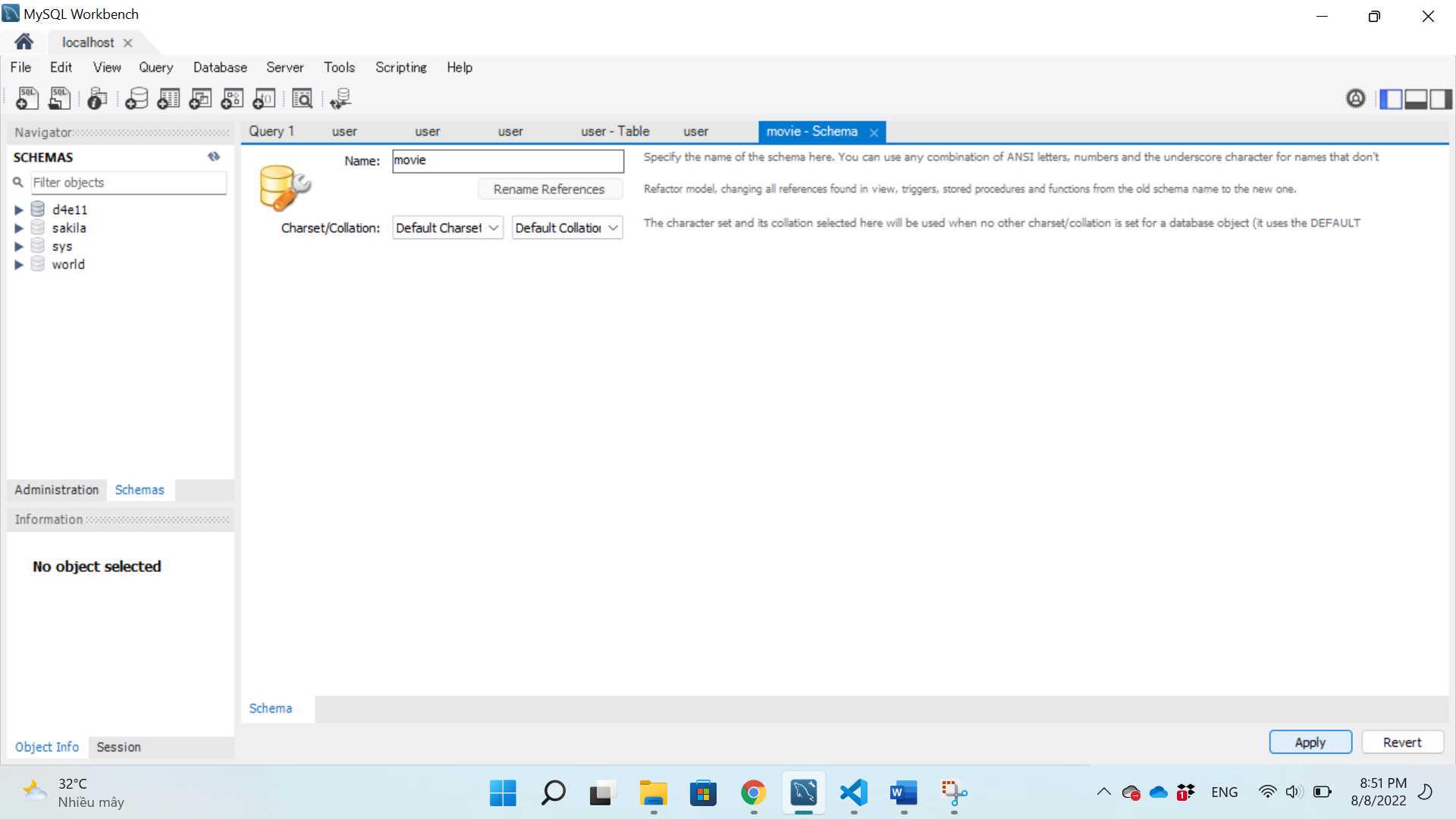
{'max': 98}

98

# Session 8

1. Tạo database mới CREATE schema





1. Tạo bảng movie trong database movie

import pymysql

client = pymysql.connect(

    host='localhost',

    user='root',

    password='Inuka4682!',

    cursorclass=pymysql.cursors.DictCursor

)

cursor = client.cursor()

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie(

        id VARCHAR(255) PRIMARY KEY,

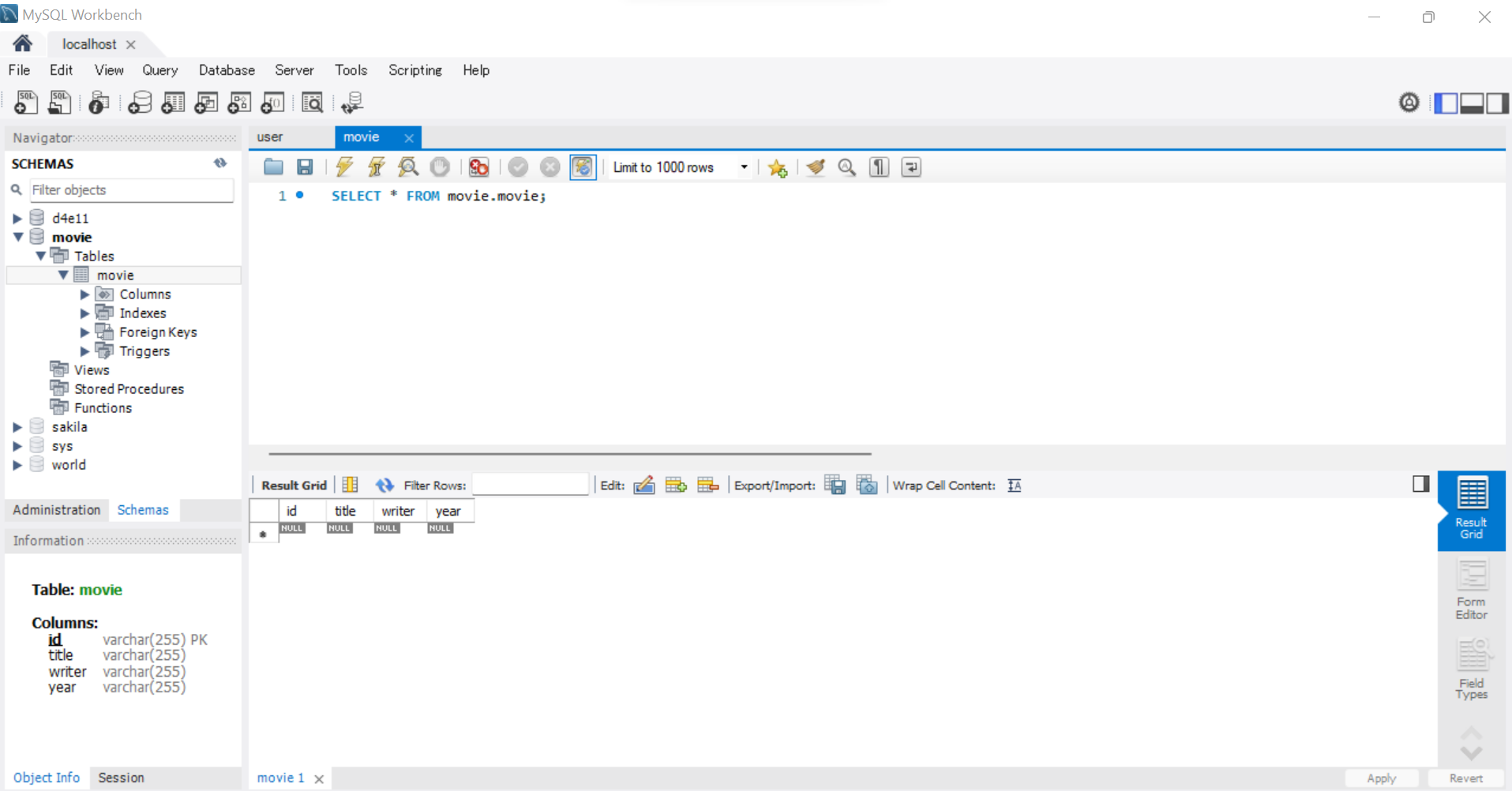
        title VARCHAR(255),

        writer VARCHAR(255),

        year VARCHAR(255)

    )

''')



cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie(

        id VARCHAR(255) PRIMARY KEY,

        title VARCHAR(255),

        writer VARCHAR(255),

        year VARCHAR(255)

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.actor(

        name VARCHAR(255) PRIMARY KEY

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie\_actor(

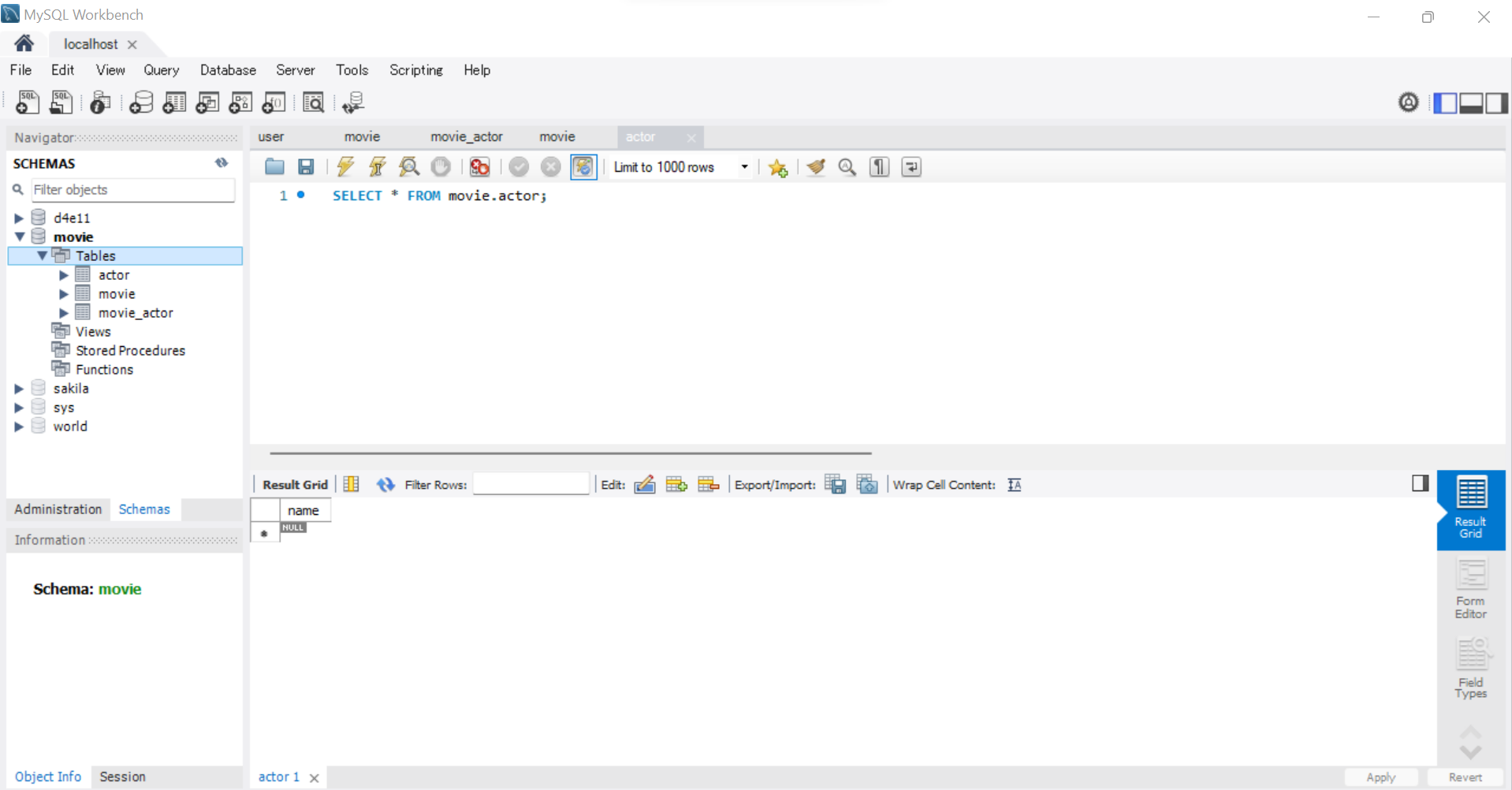
        movie\_id VARCHAR(255),

        actor\_name VARCHAR(255),

        PRIMARY KEY(movie\_id, actor\_name)

    )

''')



1. Kết nối với dữ liệu MongoDB để lấy dữ liệu

import pymysql

## Ket noi voi du lieu MongoClient ##

from pymongo import MongoClient

mongo\_client = MongoClient()

mongo\_db = mongo\_client.get\_database('mongo\_practice')

movie\_collection = mongo\_db.get\_collection('movies')

## Ket noi voi du lieu MySQL ##

mysql\_client = pymysql.connect(

    host='localhost',

    user='root',

    password='Inuka4682!',

    cursorclass=pymysql.cursors.DictCursor

)

cursor = mysql\_client.cursor()

## Tao bang trong MySQL ##

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie(

        id VARCHAR(255) PRIMARY KEY,

        title VARCHAR(255),

        writer VARCHAR(255),

        year VARCHAR(255)

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.actor(

        name VARCHAR(255) PRIMARY KEY

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie\_actor(

        movie\_id VARCHAR(255),

        actor\_name VARCHAR(255),

        PRIMARY KEY(movie\_id, actor\_name)

    )

''')

query = {

    'title': {'$ne': None},

    'writer': {'$ne': None},

    'year': {'$ne': None},

    'actors': {'$ne': None}

}

for movie in movie\_collection.find(query):   # EXTRACT

    print(movie)

mysql\_client.commit()

PS C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11\session7> python movie\_sql.py

{'\_id': ObjectId('62ef6c3bf3380fc685e60930'), 'title': 'Fight Club', 'writer': 'Chuck Palahniuk', 'year': 1999, 'actors': ['Brad Pitt', 'Edward Norton']}

{'\_id': ObjectId('62ef6c3bf3380fc685e60931'), 'title': 'Pulp Fiction', 'writer': 'Quentin Tarantino', 'year': 1994, 'actors': ['Brad Pitt', 'Edward Norton']}

{'\_id': ObjectId('62ef6c3bf3380fc685e60932'), 'title': 'Inglorious Basterds', 'writer': 'Chuck Palahniuk', 'year': 1999, 'actors': ['Brad Pitt', 'Edward Norton']}

{'\_id': ObjectId('62ef6c3bf3380fc685e60933'), 'title': 'The Hobbit: An Unexpected Journey', 'writer': 'Chuck Palahniuk', 'year': 1999, 'actors': ['Brad Pitt', 'Edward Norton']}

{'\_id': ObjectId('62ef6c3bf3380fc685e60934'), 'title': 'The Hobbit: The Desolation of Smaug', 'writer': 'Chuck Palahniuk', 'year': 1999, 'actors': ['Brad Pitt', 'Edward Norton']}

{'\_id': ObjectId('62ef6c3bf3380fc685e60935'), 'title': 'The Hobbit: The Battle of the Five Armies', 'writer': 'Chuck Palahniuk', 'year': 1999, 'actors': ['Brad Pitt', 'Edward Norton']}

1. Upload dữ liệu từ MongoDB sang MySQL

import pymysql

## Ket noi voi du lieu MongoClient ##

from pymongo import MongoClient

mongo\_client = MongoClient()

mongo\_db = mongo\_client.get\_database('mongo\_practice')

movie\_collection = mongo\_db.get\_collection('movies')

## Ket noi voi du lieu MySQL ##

mysql\_client = pymysql.connect(

    host='localhost',

    user='root',

    password='Inuka4682!',

    cursorclass=pymysql.cursors.DictCursor

)

cursor = mysql\_client.cursor()

## Tao bang trong MySQL ##

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie(

        id VARCHAR(255) PRIMARY KEY,

        title VARCHAR(255),

        writer VARCHAR(255),

        year VARCHAR(255)

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.actor(

        name VARCHAR(255) PRIMARY KEY

    )

''')

cursor.execute('''

    CREATE TABLE IF NOT EXISTS movie.movie\_actor(

        movie\_id VARCHAR(255),

        actor\_name VARCHAR(255),

        PRIMARY KEY(movie\_id, actor\_name)

    )

''')

## Upload du lieu tu MongoDB sang MySQL ##

query = {

    'title': {'$ne': None},

    'writer': {'$ne': None},

    'year': {'$ne': None},

    'actors': {'$ne': None}

}

for movie in movie\_collection.find(query):   # EXTRACT

    # TRANSFORM

    movie\_id = str(movie['\_id'])

    movie\_title = movie['title']

    movie\_writer = movie['writer']

    movie\_year = movie['year']

    # LOAD

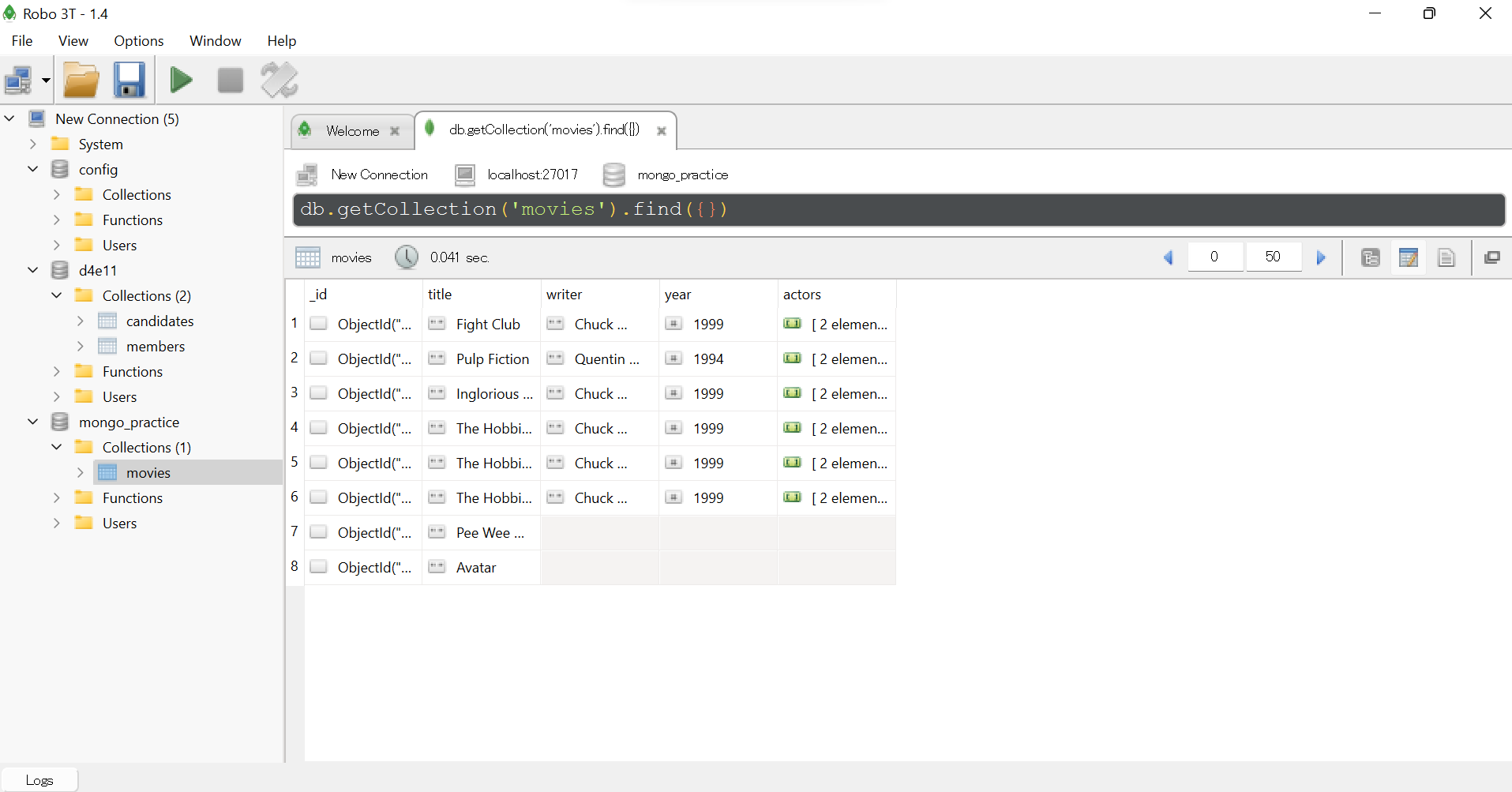
    cursor.execute(f'''

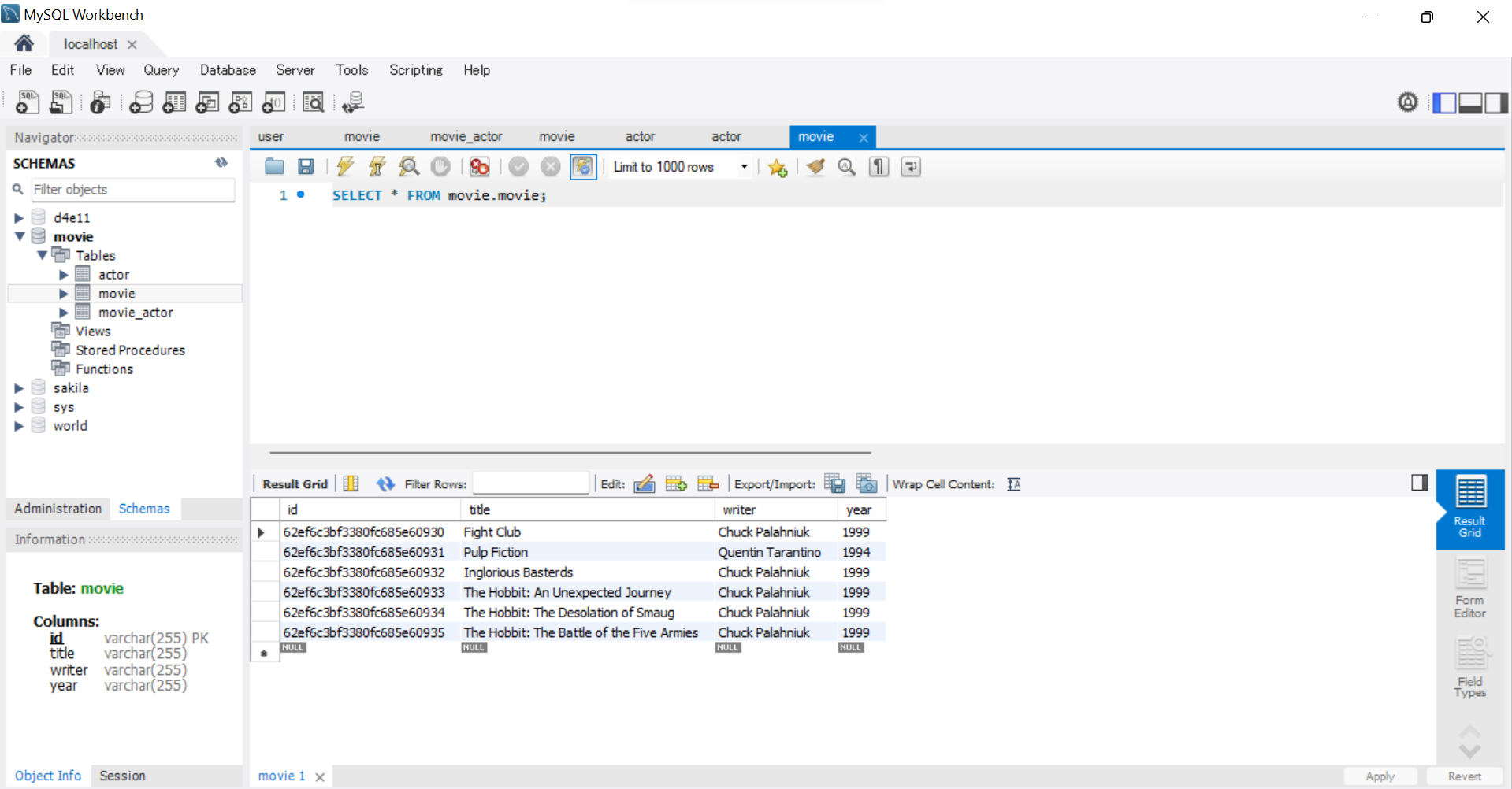
        INSERT INTO movie.movie(id, title, writer, year)

        VALUES ('{movie\_id}', '{movie\_title}', '{movie\_writer}', '{movie\_year}')

    ''')

mysql\_client.commit()





1. Đẩy code lên github

C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11>git stage .

C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11>git commit -m "initial commit"

C:\Users\lethi\Data Self-study\D4E11\Practice\D4E11>git push