Cài đặt hàm Python LoadRatings()

1. **Cài đặt máy ảo Ubuntu**

* Bản Ubuntu 22.04 LTS: <https://releases.ubuntu.com/jammy/ubuntu-22.04.5-desktop-amd64.iso>
* Mở VMware Workstation / Player → “Create a New Virtual Machine”.
* Chọn file .iso của Ubuntu vừa tải.
* Cấu hình máy ảo:

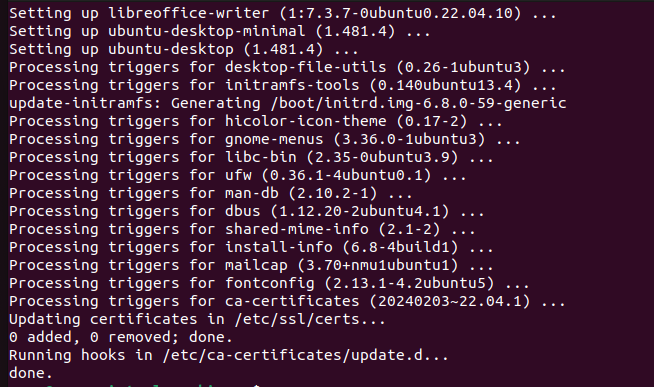
RAM: tối thiểu **4GB** (nên để 8GB nếu máy bạn đủ).

CPU: 2 cores.

Ổ cứng: ≥ 20GB (dùng SSD thì nhanh hơn).

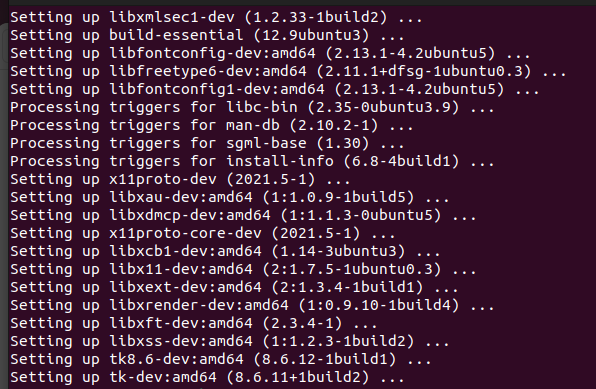
* Chạy lệnh cập nhật hệ thống:

sudo apt upadate && sudo apt upgrade -y



* Cài python 3.12
* Cài thư viện phụ thuộc:

sudo apt install -y wget build-essential libssl-dev zlib1g-dev \libbz2-dev libreadline-dev libsqlite3-dev curl libncursesw5-dev \xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev



* Tải mã nguồn Python 3.12.3

cd /usr/src

sudo wget https://www.python.org/ftp/python/3.12.3/Python-3.12.3.tgz

sudo tar xzf Python-3.12.3.tgz

cd Python-3.12.3

* Biên dịch và cài đặt

sudo ./configure --enable-optimizations

sudo make -j$(nproc)

sudo make altinstall

* Đặt làm phiên bản sử dụng trong dự án

python3.12 –version

Kết quả:



1. **Cài đặt PostgreSQL**

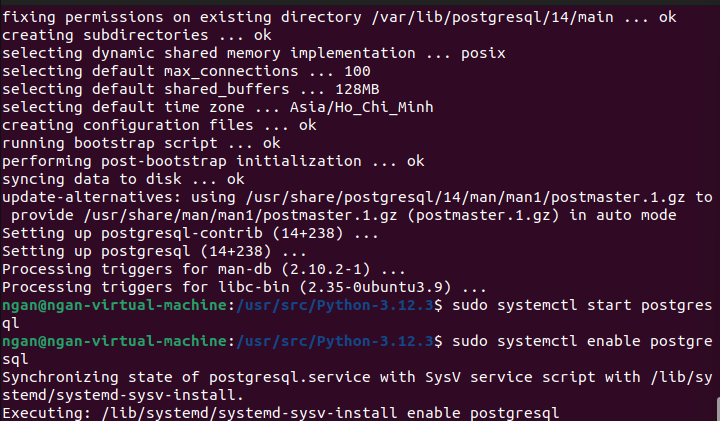
* Cài đặt PostgreSQL

sudo apt install postgresql postgresql-contrib -y

* Khởi động dịch vụ PostgreSQL

sudo systemctl start postgresql

sudo systemctl enable postgresql



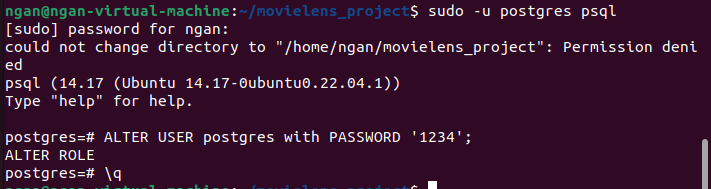
* Tạo user và password cho dự án:

sudo -u postgres psql

Trong giao diện PostgreSQL

ALTER USER postgres WITH PASSWORD '1234';

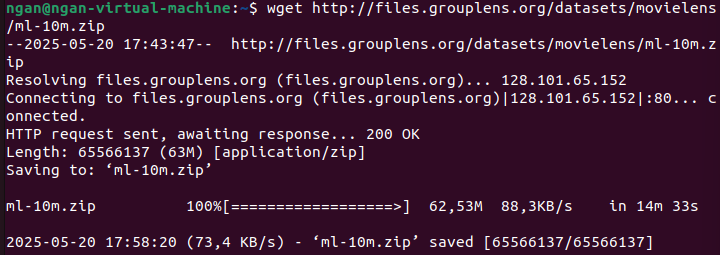
\q



1. **Tệp rating.dat**

* Tải file zip

wget http://files.grouplens.org/datasets/movielens/ml-10m.zip

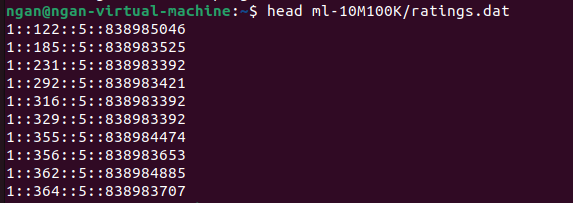


* Giải nén

unzip ml-10m.zip

* Kiểm tra dữ liệu

head ml-10M100K/ratings.dat



1. **Cài đặt hàm LoadRatings()**

* Tạo thư mục làm project:

cd ~

mkdir movielens\_project

cd movielens\_project

* Tạo file Interface.py. Đảm bảo trong thư mục có chứa đủ 3 file test



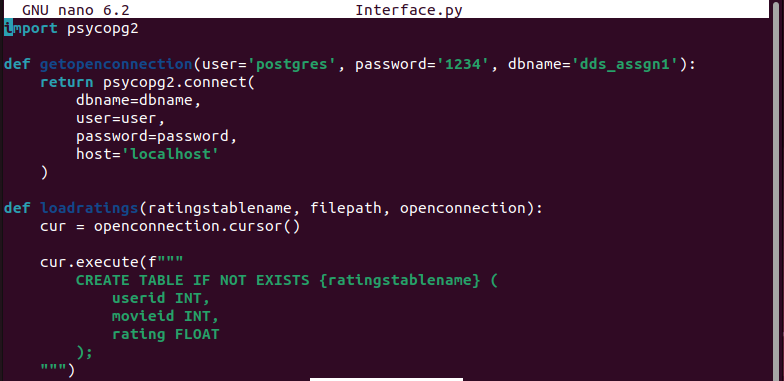
* CÀI THƯ VIỆN psycopg2

sudo apt install libpq-dev python3-psycopg2 -y

pip install psycopg2

* Mở file Interface.py và dán đoạn code chứa hàm load\_ratings() vào:

nano Interface.py



**import** psycopg2

**def** getopenconnection(user='postgres', password='1234', dbname='dds\_assgn1'):

**return** psycopg2.connect(

dbname=dbname,

user=user,

password=password,

host='localhost'

)

**def** loadratings(ratingstablename, filepath, openconnection):

cur = openconnection.cursor()

cur.execute(f"""

CREATE TABLE IF NOT EXISTS {ratingstablename} (

userid INT,

movieid INT,

rating FLOAT

);

""")

openconnection.commit()

**with** open(filepath, 'r') **as** file:

**for** line **in** file:

parts = line.strip().split("::")

**if** len(parts) == 4:

userid, movieid, rating = int(parts[0]), int(parts[1]), float(parts[2])

cur.execute(f"""

INSERT INTO {ratingstablename} (userid, movieid, rating)

VALUES (%s, %s, %s)

""", (userid, movieid, rating))

openconnection.commit()

cur.close()

* Hàm test\_loadratings\_only.py ( nếu muốn test)

touch test\_loadratings\_only.py

nano test\_loadratings\_only.py

Dán đoạn code sau vào file nano:

**import** testHelper

**import** Interface **as** MyAssignment # Phải đặt tên y chang như trong Assignment1Tester.py

DATABASE\_NAME = 'dds\_assgn1'

RATINGS\_TABLE = 'ratings'

INPUT\_FILE\_PATH = 'test\_data.dat'

ACTUAL\_ROWS\_IN\_INPUT\_FILE = 20 # file test\_data.dat có 20 dòng

**if** \_\_name\_\_ == '\_\_main\_\_':

**try**:

testHelper.createdb(DATABASE\_NAME)

**with** testHelper.getopenconnection(dbname=DATABASE\_NAME) **as** conn:

conn.set\_isolation\_level(0)

testHelper.deleteAllPublicTables(conn)

[result, e] = testHelper.testloadratings(

MyAssignment,

RATINGS\_TABLE,

INPUT\_FILE\_PATH,

conn,

ACTUAL\_ROWS\_IN\_INPUT\_FILE

)

**if** result:

**print**("✅ Hàm loadratings() PASS.")

**else**:

**print**("❌ Hàm loadratings() FAIL.")

**print**(e)

**except** Exception **as** ex:

**print**("❌ Lỗi khi test:")

**print**(ex)