P

ractical – 07

Aim: Data Storage in Redis

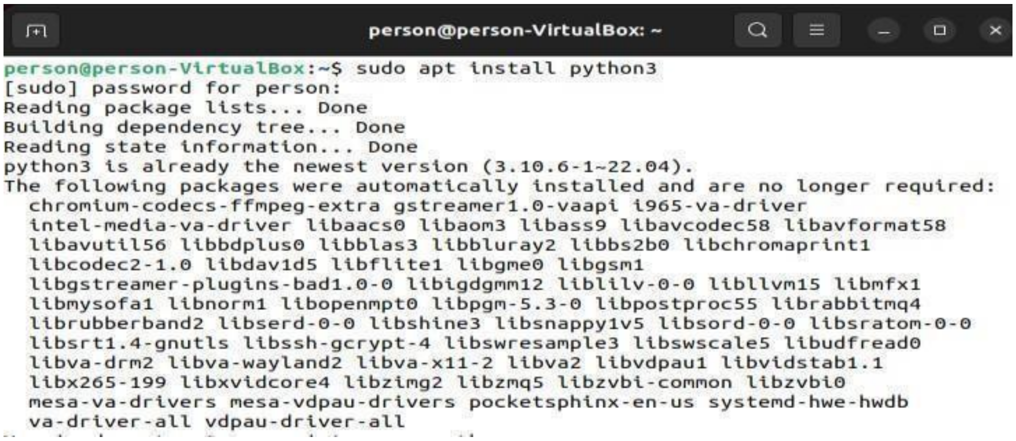
a) Implement caching functionality using Redis as a cache Store.

b) Store and retrieve data from Redis cache using appropriate commands.

Description: - Implement caching functionality using Redis in a hypothetical programming language. Keep in mind that the exact implementation details may vary depending on the programming language you are using, as different languages may have different Redis client libraries.

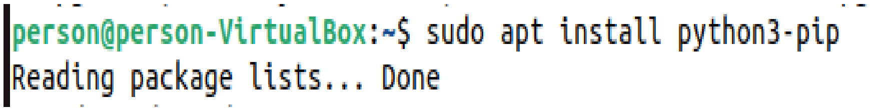
Execution:

1) Implement caching functionality using Redis as a cache store using python.
2) To install Python on Ubuntu, you can use the following steps:
   - Sudo apt install python3



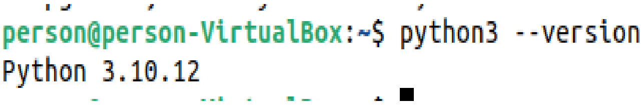
```
person@person-VirtualBox:~$ sudo apt install python3
[sudo] password for person:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libdav1d5 libflite1 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm15 libmfx1
  libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
  libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1
  libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us systemd-hwe-hwdb
  va-driver-all vdpau-driver-all
```

3) sudo apt install python3-pip



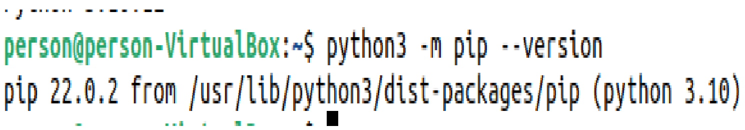
```
person@person-VirtualBox:~$ sudo apt install python3-pip
Reading package lists... Done
```

4) python3 –version



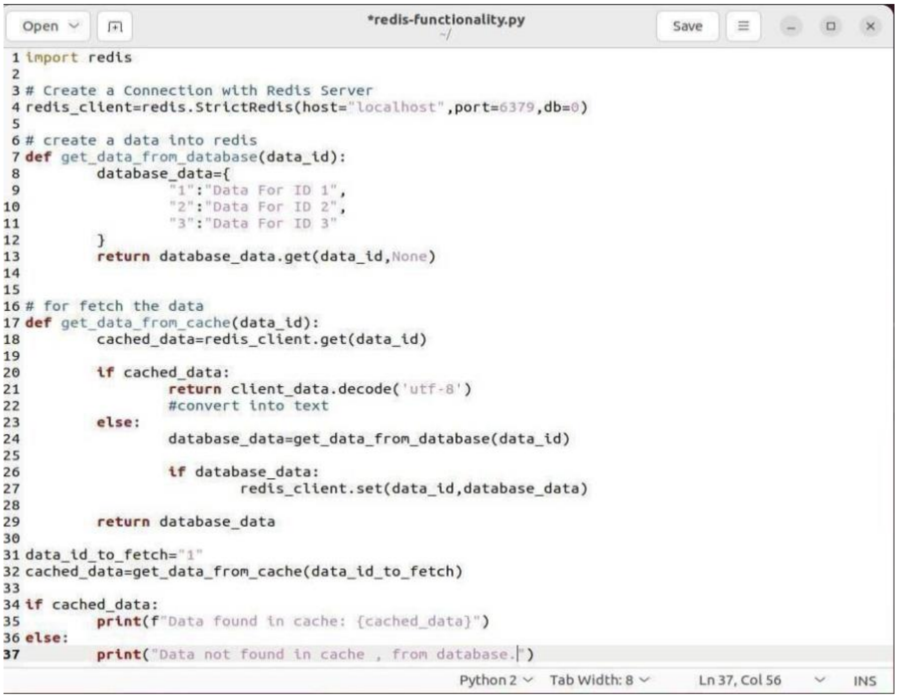
```
person@person-VirtualBox:~$ python3 --version
Python 3.10.12
```

5) python3 -m pip –version.



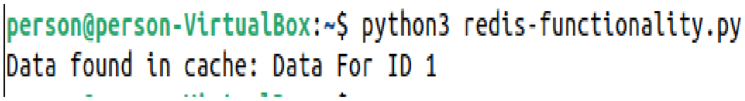
```
person@person-VirtualBox:~$ python3 -m pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

6) Use a text editor to write your Python script and save the file as redis-functionality.py



```python
import redis

# Create a Connection with Redis Server
redis_client=redis.StrictRedis(host="localhost",port=6379,db=0)

# create a data into redis
def get_data_from_database(data_id):
        database_data={
                "1":"Data For ID 1",
                "2":"Data For ID 2",
                "3":"Data For ID 3"
        }
        return database_data.get(data_id,None)


# for fetch the data
def get_data_from_cache(data_id):
        cached_data=redis_client.get(data_id)

        if cached_data:
                return client_data.decode('utf-8')
                #convert into text
        else:
                database_data=get_data_from_database(data_id)

                if database_data:
                        redis_client.set(data_id,database_data)

        return database_data

data_id_to_fetch="1"
cached_data=get_data_from_cache(data_id_to_fetch)

if cached_data:
        print(f"Data found in cache: {cached_data}")
else:
        print("Data not found in cache , from database.")
```

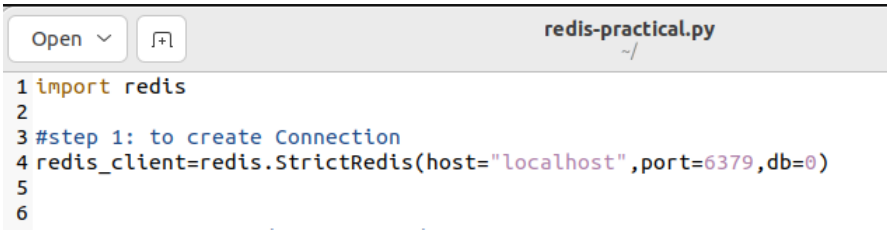
7) To turn the file in the command line type python3 redis-functionality.py(make sure you are in the right folder where you have saved redis- functionality.py).



```
person@person-VirtualBox:~$ python3 redis-functionality.py
Data found in cache: Data For ID 1
```

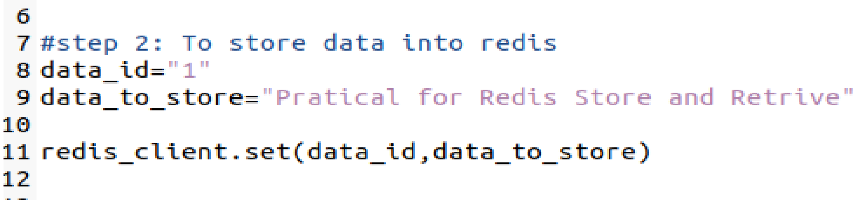
8) Store and retrieve data from Redis cache using appropriate commands.

□ Connecting to Redis



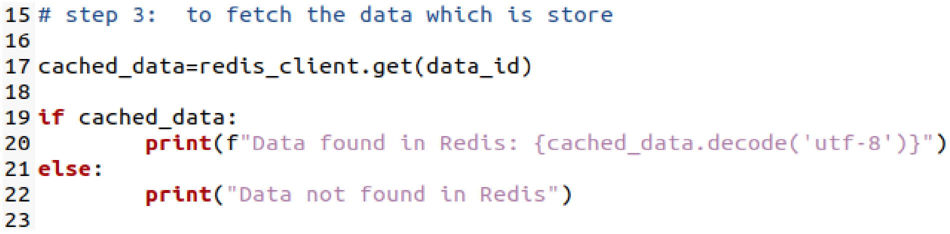
```python
import redis

#step 1: to create Connection
redis_client=redis.StrictRedis(host="localhost",port=6379,db=0)


```

□ Storing Data in Redis

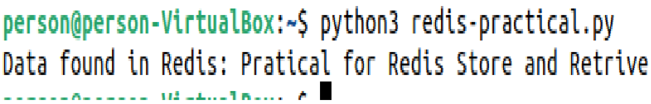


```python

#step 2: To store data into redis
data_id="1"
data_to_store="Pratical for Redis Store and Retrive"

redis_client.set(data_id,data_to_store)

```

□ Retrieving Data from Redis

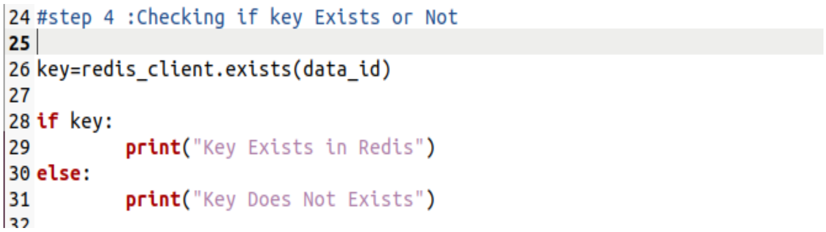


```python
# step 3:  to fetch the data which is store

cached_data=redis_client.get(data_id)

if cached_data:
        print(f"Data found in Redis: {cached_data.decode('utf-8')}")
else:
        print("Data not found in Redis")

```

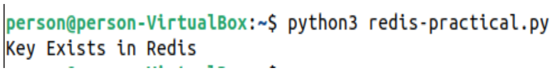


```
person@person-VirtualBox:~$ python3 redis-practical.py
Data found in Redis: Pratical for Redis Store and Retrive
```

□ Checking if a Key Exists

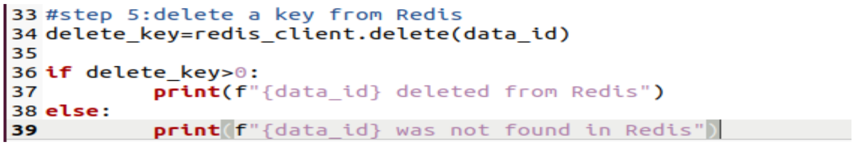


```python
#step 4 :Checking if key Exists or Not

key=redis_client.exists(data_id)

if key:
        print("Key Exists in Redis")
else:
        print("Key Does Not Exists")
```

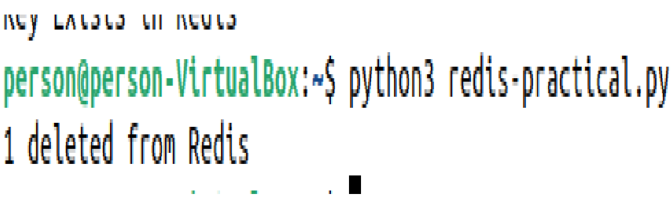


```
person@person-VirtualBox:~$ python3 redis-practical.py
Key Exists in Redis
```

□ Deleting a Key from Redis



```python
#step 5:delete a key from Redis
delete_key=redis_client.delete(data_id)

if delete_key>0:
        print(f"{data_id} deleted from Redis")
else:
        print(f"{data_id} was not found in Redis")
```



```
Key Exists in Redis
person@person-VirtualBox:~$ python3 redis-practical.py
1 deleted from Redis
```