Universidade do Minho
Escola de Engenharia

SHELVE - Persistent storage of arbitrary Python objects

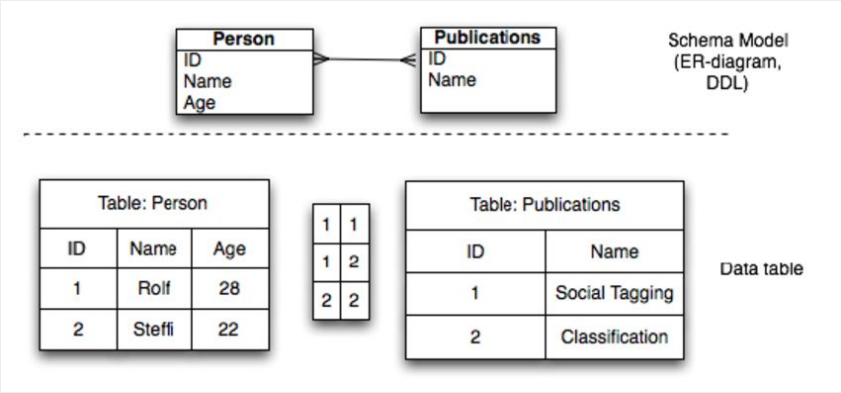
By Duarte Freitas, Rafael Silva, Ricardo Pereira

Scripting o Processamento de Linguagem Natural

Universidade do Minho, Mestrado Integrado em Engenharia Informática,
4º Ano, 2º Semestre, Abril 2020

Overview

One of the primitive forms of a database is a collection of values where each value is identified by a key, stored in disk and physical memory - called a DBM database or in a nutshell, a persistent dictionary.



Schema for relational database

What is SHELVE?

- Shelve is a python module used to store objects in a file.
- The shelve module implements persistent storage for arbitrary Python objects which can be pickled, using a dictionary-like API.
- The shelve module can be used as a simple persistent storage option for Python objects when a relational database is overkill.
- The shelf is accessed by keys, just as with a dictionary.
- The values are pickled and written to a database created and managed by anydbm.
- Python objects — anything that the pickle module can handle.
- This includes most class instances, recursive data types, and objects containing lots of shared sub-objects.
- The keys are ordinary strings.
- The shelve module in Python's standard library is a simple yet effective tool for persistent data storage when using a relational database solution is not required.

IMPORTANT

When a program has a shelf open for writing, no other program should have it open for reading or writing.

Pickle vs Shelve

pickle: is for serializing some object (or objects) as a single byte stream in a file.

```
import pickle

integers = [1, 2, 3, 4, 5]

with open('pickle-example.p', 'wb') as pfile:
    pickle.dump(integers, pfile)
```

This will dump the *integers* list to a binary file called *pickle-example.p*

Now try reading the pickled file back

```
import pickle

with open('pickle-example.p', 'rb') as pfile:
    integers = pickle.load(pfile)
    print integers
```

output [1, 2, 3, 4, 5].

Pickle vs Shelve

shelve: builds on top of pickle and implements a serialization dictionary where objects are pickled, but associated with a key (some string), so you can load your shelved data file and access your pickled objects via keys. This could be more convenient were you to be serializing many objects.

```
import shelve

integers = [1, 2, 3, 4, 5]

# If you're using Python 2.7, import contextlib and use
# the line:
# with contextlib.closing(shelve.open('shelf-example', 'c')) as shelf:
with shelve.open('shelf-example', 'c') as shelf:
    shelf['ints'] = integers
```

Notice how you add objects to the shelf via dictionary-like access.

Read the object back in with code like the following

```
import shelve

# If you're using Python 2.7, import contextlib and use
# the line:
# with contextlib.closing(shelve.open('shelf-example', 'r')) as shelf:
with shelve.open('shelf-example', 'r') as shelf:
    for key in shelf.keys():
        print(repr(key), repr(shelf[key]))
```

```
'ints', [1, 2, 3, 4, 5]
```

Example

Shelf implementation(dbm) :

```
import shelve

s = shelve.open("students_DB")
s['63129'] = {"Nome" : "D" , "Numero" : "A63129", "Curso": "MiEI"}
s['74264'] = {"Nome" : "R" , "Numero" : "A74264", "Curso": "MiEI"}
s['77045'] = {"Nome" : "R" , "Numero" : "A77045", "Curso": "MiEI"}

print(list(s.items()))
s.close()
```

```
[('63129', {'Nome': 'D', 'Numero': 'A63129', 'Curso': 'MiEI'}),
 ('74264', {'Nome': 'R', 'Numero': 'A74264', 'Curso': 'MiEI'}),
 ('77045', {'Nome': 'R', 'Numero': 'A77045', 'Curso': 'MiEI'})]
```

Example 2

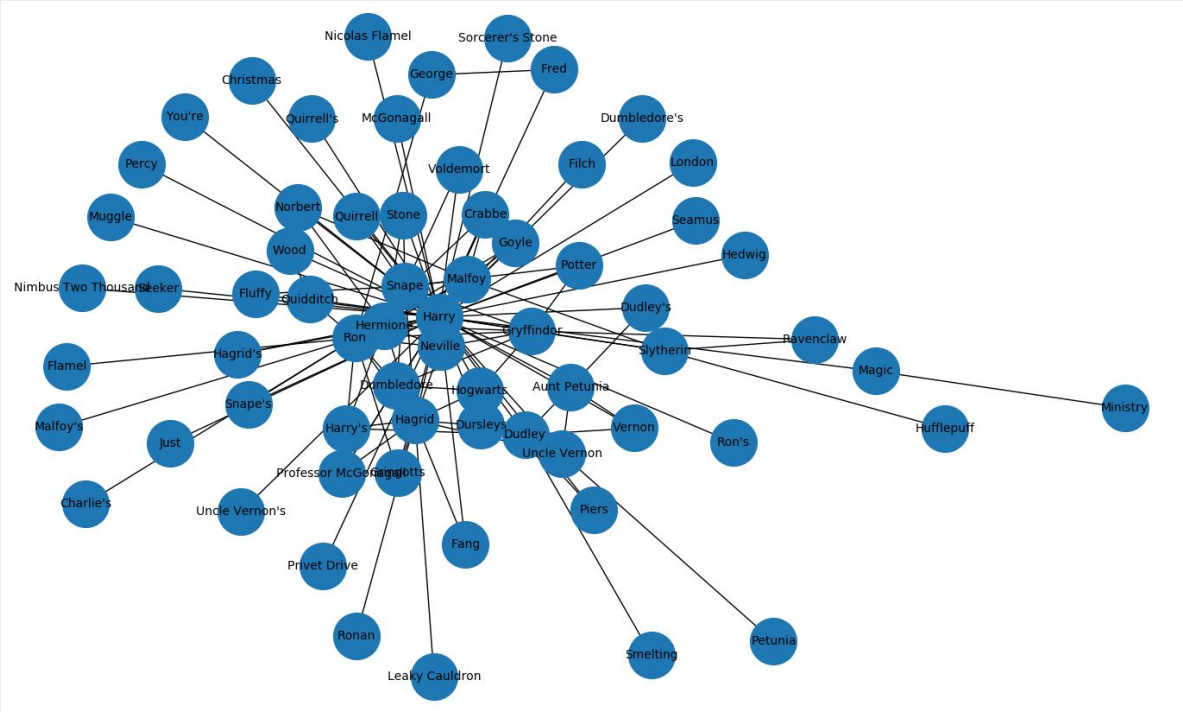
Shelf implementation (dbm):

```
import shelve
s = shelve.open("students_DB")
s['63129'] = {"Nome" : "D" , "Numero" : "A63129", "Curso": "MiEI"}
s['74264'] = {"Nome" : "R" , "Numero" : "A74264", "Curso": "MiEI"}
s['77045'] = {"Nome" : "R" , "Numero" : "A77045", "Curso": "MiEI"}
print(list(s.items()))
s.close()
```

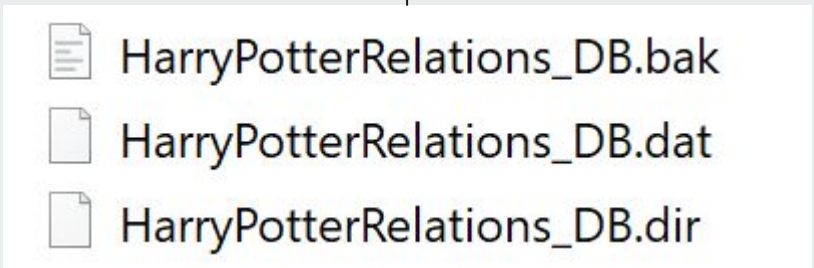
```
[('63129', {'Nome': 'D', 'Numero': 'A63129', 'Curso': 'MiEI'}),
 ('74264', {'Nome': 'R', 'Numero': 'A74264', 'Curso': 'MiEI'}),
 ('77045', {'Nome': 'R', 'Numero': 'A77045', 'Curso': 'MiEI'})]
```

```
s = shelve.open("students_DB")
s['63129']= {"Nome" : "D" , "Numero" : "X", "Curso": "MiEI"}
s.close()
s=shelve.open("students_DB")
print(list(s.items()))
s.close()
```

Example 3


Nicolas Flamel
Sorcerer's Stone
George
Fred
Christmas
You're
Quirrell's
McGonagall
Dumbledore's
Percy
Voldemort
Filch
London
Muggle
Norbert
Quirrell
Stone
Crabbe
Seamus
Wood
Goyle
Hedwig
Potter
Nimbus Two Thousand
Seeker
Fluffy
Quidditch
Snape
Malfoy
Dudley's
Harry
Hermione
Ron
Gryffindor
Ravenclaw
Neville
Slytherin
Hagrid's
Flamel
Magic
Dumbledore
Hogwarts
Aunt Petunia
Ministry
Snape's
Malfoy's
Harry's
Hagrid
Dursleys
Dudley
Vernon
Hufflepuff
Just
Uncle Vernon
Ron's
Professor McGonagall
Gringotts
Charlie's
Uncle Vernon's
Piers
Fang
Privet Drive
Ronan
Smelting
Petunia
Leaky Cauldron

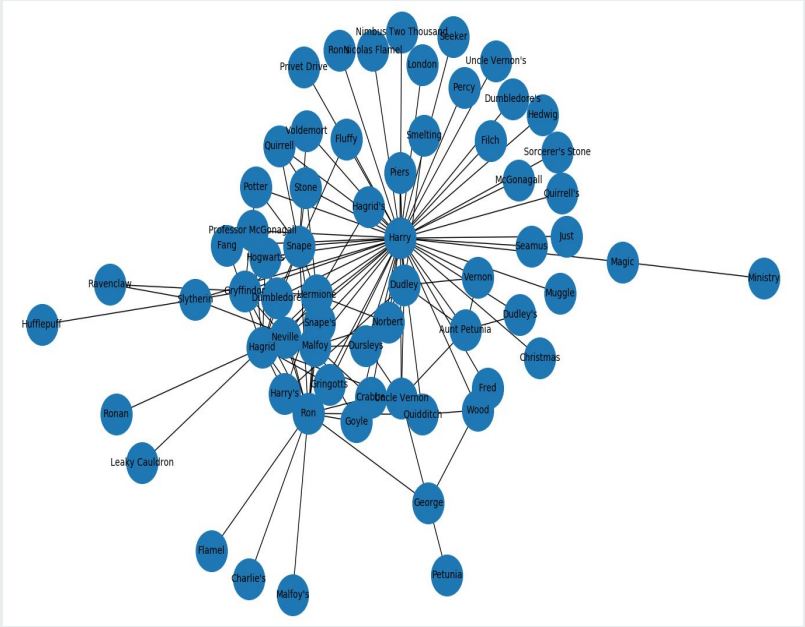
```
with shelve.open("HarryPotterRelations_DB") as hp:
    k = 0
    for i in listaPares:
        hp[str(k)] = { "Name": i[0], "Friend Name": i[1] }
        k += 1
```



```
import shelve

with shelve.open("HarryPotterRelations_DB") as hp:
    print(list(hp.items()))
```

```
[('118', {'Name': 'Dumbledore', 'Friend Name': 'Snape'}), ('119', {'Name': 'Quirrell', 'Friend Name': 'Harry'}), ('120', {'Name': 'Snape', 'Friend Name': 'Fluffy'}), ('121', {'Name': 'Stone', 'Friend Name': 'Quirrell'}), ('122', {'Name': 'Harry', 'Friend Name': "Quirrell's"}), ('123', {'Name': 'Harry', 'Friend Name': 'Fluffy'}), ('124', {'Name': 'Stone', 'Friend Name': 'Snape'}), ('125', {'Name': 'Harry', 'Friend Name': 'Norbert'}), ('126', {'Name': 'Norbert', 'Friend Name': 'Hermione'}), ('127', {'Name': 'Norbert', 'Friend Name': 'Malfoy'}), ('128', {'Name': 'Ronan', 'Friend Name': 'Hagrid'}), ('129', {'Name': 'Voldemort', 'Friend Name': 'Snape'})]
```

```
with shelve.open("HarryPotterRelations_DB") as hp:
    print(list(hp.items()))

    G = nx.Graph()
    for x in list(hp.items()):
        G.add_edge(x[1]["Name"], x[1]["Friend Name"])

    plt.figure(figsize = (15, 15))
    nx.draw(G, with_labels = True, node_size = 1500,
            font_size = 10)
    plt.show()
```



Bibliography

- https://pythontic.com/modules/shelve/introduction
- https://docs.python.org/3/library/shelve.html
- https://www.tutorialspoint.com/python-object-persistence-shelve
- https://pythonpedia.com/en/tutorial/10629/shelve
- https://pymotw.com/2/shelve/
- https://github.com/python/cpython/blob/master/Lib/shelve.py
- https://books.google.pt/books?id=q8W3WQbNWmkC&pg=PA1316&lpg=PA1316&dq
- https://books.google.pt/books?id=KtcnDwAAQBAJ&pg=PT575&lpg=PT575&dq

Universidade do Minho
Escola de Engenharia

SHELVE - Persistent storage of arbitrary Python objects

By Duarte Freitas, Rafael Silva, Ricardo Pereira

Scripting o Processamento de Linguagem Natural

Universidade do Minho, Mestrado Integrado em Engenharia Informática,
4º Ano, 2º Semestre, Abril 2020