



Introduction to Database Systems: CS312

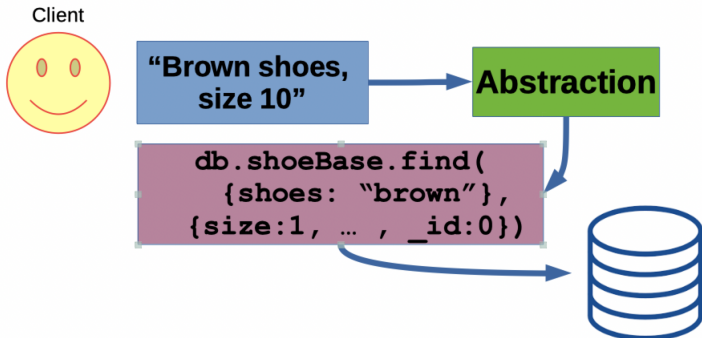
Pymongo

Oliver Bonham-Carter

3 May 2022

Abstraction

To make some process *abstract* is to hide (automate) some of the details that serve to complicate the process. The idea behind *abstraction* here is to create a more user-friendly experience by removing some of the the complexities of using Mongo databases.



Mongo Container Commands

Abstraction

Mongo
Container

Programming
Inside
Container

Start the bash

```
sudo docker exec -it mongodb bash
```

Download updated package information with apt

```
apt update
```

Install an editor, Python3, Pip

```
apt install nano  
apt install python3-pip
```

Use Pip to install *pymongo*

```
pip install pymongo
```

Use *Nano* to begin coding

```
nano pymongoDemo.py
```

After coding, exit Nano and run your code

```
python3 pymongoDemo.py
```

Main Nano Menu Items

^G	Get Help
^X	Exit

^O	WriteOut
^J	Justify

^R	Read File
^W	Where is

^Y	Prev Pg
^V	Next Pg

^K	Cut Text
^U	UnCut Text

^C	Cur Pos
^T	To Spell

- Control-O :: ^O : Save
- Control-X :: ^X : Exit

```
#!/usr/bin/env python3

# libraries
from pymongo import MongoClient
import string

# creating connections for communicating with MongoDB
client = MongoClient('localhost:27017')
db = client.mongodemo # The name of the collection is mongodemo

...

print("\t [+] Data BEFORE addition")
read()
print("\t [+] Insert some data")
insert()
print("\t [+] Data AFTER addition")
read()
print("\t [+] Update Data")
update()
print("\t [+] Data AFTER Update")
read()
```

Read Function

Abstraction

Mongo
Container

Programming
Inside
Container

```
def read():  
    """ function to read records from mongo db """  
    try:  
        empCol = db.Employee.find()  
        print("\n Found: all data from DataEmployee \n")  
        for emp in empCol:  
            print(f"\t [+] {emp}")  
    except Exception as e:  
        print(str(e))  
# end of read()
```

```
def insert():
    """ Function to insert data into mongo db """
    try:
        employeeId = input('Enter Employee id :')
        employeeFirstName = input('Enter FirstName :')
        employeeLastName = input('Enter LastName :')
        employeeAge = input('Enter age :')
        employeeCountry = input('Enter Country :')

        # insert the data into the base
        db.Employee.insert_one(
            {
                "id": employeeId,
                "firstName":employeeFirstName,
                "lastName":employeeLastName,
                "age":employeeAge,
                "country":employeeCountry
            })
        print("\nInserted data successfully\n")

    except Exception as e:
        print(str(e))
    # end of insert()
```

Update

Abstraction

Mongo
Container

Programming
Inside
Container

```
def update():
    """ Function to update record to mongo db """
    print("  Update:")
    try:
        employeeId = input('  Enter Employee id :')
        employeeFirstName = input('  Enter FirstName :')
        employeeLastName = input('  Enter LastName :')
        employeeAge = input('  Enter age :')
        employeeCountry = input('  Enter Country :')

        # update the record with the new information
        db.Employee.update_one(
            {"id": employeeId},
            {
                "$set": {
                    "firstName":employeeFirstName,
                    "lastName":employeeLastName,
                    "age":employeeAge,
                    "country":employeeCountry
                } })
        print("\nRecords updated successfully. \n")
    except Exception as e:
        print(str(e))
    # end of update()
```


Graded Class Activity

Complete This Activity for Points

Abstraction

Mongo
Container

Programming
Inside
Container



THINK

- Now go back to the above code to create a different type of DB.
- Create some data to insert using Pymongo code into your new database.
- Write sophisticated queries in your database to show it?

Graded Class Activity

Complete This Activity for Points

Abstraction

Mongo
Container

Programming
Inside
Container



THINK

Use *Nano* to begin coding in the container

```
nano pymongoDemo_activity.py
```

- <https://classroom.github.com/a/oJrLadlt>
- Copy and paste your working code from your container into your local file `src/pyMongoDemo_activity.py` for your submission.
- Complete four questions in `writing/report.md`
- Due at 11:59pm on the 5th May 2022.
- You can work with others but each person submits his or her own work to individual repository.