

Upload the Austin Animal Center (AAC) Outcomes data set into MongoDB by importing a CSV file using the appropriate MongoDB import tool.

```
brystonjensen_snhu@78e20b13519a: /usr/local/datasets
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\windows\system32> docker start cs340Mongo-brystonjensen_snhu
cs340Mongo-brystonjensen_snhu
PS C:\windows\system32> docker exec -it cs340Mongo-brystonjensen_snhu /bin/bash
root@78e20b13519a:/# su - brystonjensen_snhu
brystonjensen_snhu@78e20b13519a:~$ cd /usr/local/datasets/
brystonjensen_snhu@78e20b13519a: /usr/local/datasets$ mongoimport --username="${MONGO_USER}" --password="${MONGO_PASS}" -
-port=${MONGO_PORT} --host=${MONGO_HOST} --db=AAC --collection=animals --authenticationDatabase=admin --type=csv --heade
rline --drop ./aac_shelter_outcomes.csv
2024-02-06T18:05:01.967+0000    connected to: mongodb://localhost:27017/
2024-02-06T18:05:01.967+0000    dropping: AAC.animals
2024-02-06T18:05:02.113+0000    10000 document(s) imported successfully. 0 document(s) failed to import.
brystonjensen_snhu@78e20b13519a: /usr/local/datasets$
```

Next, you must develop a Python module in a PY file using object-oriented programming methodology to enable create and read functionality for the database.

A method that inserts a document into a specified MongoDB database and collection.

A method that queries for documents from a specified MongoDB database and collection.

```
34     # Method to implement the C in CRUD.
35     def create(self, data):
36         if data is not None:
37             validate = self.database.animals.insert_one(data) # data should be dictionary
38
39             # Checks to make sure it was successful.
40             if validate == 0:
41                 return True
42             return False
43         else:
44             raise Exception("Nothing to save, data parameter is empty")
45
46     # Method to implement the R in CRUD.
47     def read(self, dataS): # dataS is data to search.
48         if dataS is not None:
49             data = self.database.animals.find(dataS, {"_id": False})
50         else:
51             raise Exception("Nothing to read, dataS parameter is empty.")
52         return data
53
54     # Method to implement the U in CRUD.
55     def update(self, dataS, dataU): # dataS is data to search, dataU is data to update.
56         if dataS and dataU is not None:
57             updated = self.database.animals.update_many(dataS, {"$set": dataU})
58         else:
59             raise Exception("Nothing to update, dataS or data U parameters are empty.")
60         return updated
61
62     # Method to implement the D in CRUD
63     def delete(self, dataD): # dataD is data to delete.
64         if dataD is not None:
65             deleted = self.database.animals.delete_many(dataD)
66         else:
67             raise Exception("nothing to delete, dataD parameter is empty")
68         return deleted
```

Finally, create a Python testing script in Jupyter Notebooks that imports your CRUD Python module to call and test the create and read instances of CRUD functionality:

animalShelter Test

Create a Python testing script in Jupyter Notebooks that imports your CRUD Python module to call and test the create and read instances of CRUD functionality.

```
from animalShelter import AnimalShelter  
  
CRUD = AnimalShelter()
```

Connection Successful

One where create() passes:

One where create() should pass.

```
newAnimalTestData = {  
    'age_upon_outcome': '10 years',  
    'animal_id': 'BGJ0715',  
    'animal_type': 'Dog',  
    'breed': 'Border Collie',  
    'color': 'Black and White',  
    'date_of_birth': '2013-11-19',  
    'datetime': '2024-02-06 18:21:36',  
    'name': 'Doug',  
    'outcome_subtype': 'SCRIP',  
    'outcome_type': 'Adopted',  
    'sex_upon_outcome': 'Neutered Male',  
    'location_lat': 30.6525984560228,  
    'location_long': -97.7419963476444,  
    'age_upon_outcome_in_weeks': 533  
}
```

```
created = CRUD.create(newAnimalTestData)  
print(f"Item created: {created}")
```

Item created: True

One where create() fails:

```
One where it should fail.

[6]: badAnimalTestData = "This should fail."

created = CRUD.create(badAnimalTestData)

-----
TypeError                                 Traceback (most recent call last)
Cell In[6], line 3
      1 badAnimalTestData = "This should fail."
----> 3 created = CRUD.create(badAnimalTestData)

File ~\OneDrive - SNHU\CS-340\Module4\CS340_4-1\animalShelter.py:40, in AnimalShelter.create(self, data)
     37 """Method to implement the C in CRUD."""
     39 if data is not None:
--> 40     inserted = self.database.animals.insert_one(data) # data should be dictionary
     42     # Checks to make sure it was successful.
     43     if inserted != 0:

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pymongo\collection.py:663, in Collection.insert_one(self, document, bypass_document_validation, session, comment)
     615 def insert_one(
     616     self,
     617     document: Union[_DocumentType, RawBSONDocument],
     (...)
     620     comment: Optional[Any] = None,
     621 ) -> InsertOneResult:
     622     """Insert a single document.
     623
     624     >>> db.test.count_documents({'x': 1})
     (...)
     661     .. versionadded:: 3.0
     662     """
--> 663     common.validate_is_document_type("document", document)
     664     if not (isinstance(document, RawBSONDocument) or "_id" in document):
     665         document["_id"] = ObjectId() # type: ignore[index]

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pymongo\common.py:552, in validate_is_document_type(option, value)
     550 """Validate the type of method arguments that expect a MongoDB document."""
     551 if not isinstance(value, (abc.MutableMapping, RawBSONDocument)):
--> 552     raise TypeError(
     553         f"{option} must be an instance of dict, bson.son.SON, "
     554         "bson.raw_bson.RawBSONDocument, or "
     555         "a type that inherits from "
     556         "collections.MutableMapping"
     557     )

TypeError: document must be an instance of dict, bson.son.SON, bson.raw_bson.RawBSONDocument, or a type that inherits from collections.MutableMapping
```

One where read() passes:

```
One where the read() should pass.

read = CRUD.read({"name": "Doug"})
for animal in read:
    print(animal)

{'rec_num': 6966, 'age_upon_outcome': '8 years', 'animal_id': 'A739141', 'animal_type': 'Dog', 'breed': 'Pomeranian Mix', 'color': 'Tan/White', 'date_of_birth': '2008-11-28', 'datetime': '2016-12-02 17:41:00', 'monthyear': '2016-12-02T17:41:00', 'name': 'Doug', 'outcome_subtype': 'Partner', 'outcome_type': 'Transfer', 'sex_upon_outcome': 'Neutered Male', 'location_lat': 30.5256727280042, 'location_long': -97.3983427268139, 'age_upon_outcome_in_weeks': 418.105257936508}
{'age_upon_outcome': '10 years', 'animal_id': 'BGJ0715', 'animal_type': 'Dog', 'breed': 'Border Collie', 'color': 'Black and White', 'date_of_birth': '2013-11-19', 'datetime': '2024-02-06 18:21:36', 'name': 'Doug', 'outcome_subtype': 'SCRIP', 'outcome_type': 'Adopted', 'sex_upon_outcome': 'Neutered Male', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_upon_outcome_in_weeks': 533}
```

One where read() fails:

```
read = CRUD.read("bad": "data")

Cell In[11], line 1
      read = CRUD.read("bad": "data")
                        ^
SyntaxError: invalid syntax
```