Assignment 16: 17 Feb 2023

Q1. What is MongoDB? Explain non-relational databases in short. In which scenarios it is preferred to use MongoDB over SQL databases?
Q2. State and Explain the features of MongoDB.
Q3. Write a code to connect MongoDB to Python. Also, create a database and a collection in MongoDB.
Q4. Using the database and the collection created in question number 3, write a code to insert one record, and insert many records. Use the find() and find_one() methods to print the inserted record.
Q5. Explain how you can use the find() method to query the MongoDB database. Write a simple code to demonstrate this.
Q6. Explain the sort() method. Give an example to demonstrate sorting in MongoDB.
Q7. Explain why delete_one(), delete_many(), and drop() is used.

Q1. What is MongoDB? Explain non-relational databases in short. In which scenarios, is it preferred to use MongoDB over SQL databases?

Ans:

- MongoDB is an open source NoSQL database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.
- A non-relational database is a database that does not use the tabular schema of rows and columns found in most traditional database systems. Instead, non-relational databases use a storage model that is optimized for the specific requirements of the type of data being stored.
- NoSQL databases like MongoDB are a good choice when your data is document-centric and doesn't fit well into the schema of a relational database, when you need to accommodate massive scale, when you are rapidly prototyping, and a few other use cases.

Q2. State and Explain the features of MongoDB.

Ans: MongoDB is a non-relational document database (NoSql) that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs.

Features of MongoDB

1. Support ad hoc queries
    a. In MongoDB, you can search by field, range query and it also supports regular expression searches.
2. Indexing
    a. You can index any field in a document.
3. Replication
    a. MongoDB supports Master Slave replication.
4. Duplication of data
    a. MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.
5. Load balancing
    a. It has an automatic load balancing configuration because of data placed in shards.
6. Supports map reduce and aggregation tools.
7. Uses JavaScript instead of Procedures.
8. It is a schema-less database written in C++.

9. Provides high performance.
10. Stores files of any size easily without complicating your stack.
11. Easy to administer in the case of failures.
12. It also supports:
    a. JSON data model with dynamic schemas
    b. Auto-sharding for horizontal scalability
    c. Built in replication for high availability

Q3. Write a code to connect MongoDB to Python. Also, create a database and a collection in MongoDB.

Ans:

1) Install Driver

```
$ pip install pymongo
```

2) Create Python File

// connect.py

```
from pymongo import MongoClient # import mongo client to connect
import pprint
# Creating instance of mongoclient
client = MongoClient()
# Creating database
db = client.javatpoint
employee = {"id": "101",
"name": "Peter",
"profession": "Software Engineer",
}
# Creating document
employees = db.employees
# Inserting data
employees.insert_one(employee)
# Fetching data
pprint.pprint(employees.find_one())
```

3) Execute Python Script

This script prints the inserted record on the console.

4) Enter into Mongo Shell

Now, enter into the MongoDB database shell to see the created database and collection.

```
$ mongo
```

5) Check Databases

The following command is used to show available databases.

```
> show dbs
```

6) Check Collection

The following command is used to show available collections into the database.

```
> show collections
```

7) Access Records

We can see the stored records in particular collections. The following command is used to show the record.

```
> db.employees.find()
```

Q4. Using the database and the collection created in question number 3, write a code to insert one record, and insert many records. Use the find() and find_one() methods to print the inserted record.

Ans:

```
data1 = {
                "mail_id" : "sudh@gmail.com",
                "phone_number " : 2323424324,
                "addr" : "bangalore"
}
coll_pwskills.insert_one(data1)
data2 = [
  { "name": "Amy", "address": "Apple st 652" },
  { "name": "Hannah", "address": "Mountain 21" },
  { "name": "Michael", "address": "Valley 345" },
  { "name": "Sandy", "address": "Ocean blvd 2" },
  { "name": "Betty", "address": "Green Grass 1" },
  ]
coll_pwskills.insert_many(data2)
coll_pwskills.find_one()
{
                "mail_id" : "sudh@gmail.com",
                "phone_number " : 2323424324,
                "addr" : "bangalore"
}
for i in coll_pwskills.find():
    print(i)
{ "name": "Amy", "address": "Apple st 652" },
  { "name": "Hannah", "address": "Mountain 21" },
  { "name": "Michael", "address": "Valley 345" },
  { "name": "Sandy", "address": "Ocean blvd 2" },
  { "name": "Betty", "address": "Green Grass 1" },
```

Q5. Explain how you can use the find() method to query the MongoDB database. Write a simple code to demonstrate this.

Ans:

To find documents that match a set of selection criteria, call find() with the <criteria> parameter.

Syntax:

db.Collection_name.find(selection_criteria, projection,options)

Eg-

```
> db.student.find({})
{ "_id" : ObjectId("60227eaff19652db63812e8d"), "name" : "Akshay", "age" : 18 }
{ "_id" : ObjectId("60227eaff19652db63812e8e"), "name" : "Bablue", "age" : 17, "
score" : { "math" : 230, "science" : 234 } }
{ "_id" : ObjectId("60227eaff19652db63812e8f"), "name" : "Chandhan", "age" : 18
}
>
```

Q6. Explain the sort() method. Give an example to demonstrate sorting in MongoDB.

Ans: The sort() method can be used to sort the metadata values for a calculated metadata field.
Syntax:
db.Collection_Name.sort({filed_name:1 or -1})
Eg-

```
> db.student.find().sort({age:1})
{ "_id" : ObjectId("6015ba124dabc381f81e53ae"), "name" : "Bablue", "age" : 18 }
{ "_id" : ObjectId("6015ba124dabc381f81e53ad"), "name" : "Akshay", "age" : 19 }
{ "_id" : ObjectId("6015ba124dabc381f81e53b0"), "name" : "Gourav", "age" : 20 }
{ "_id" : ObjectId("6015ba124dabc381f81e53af"), "name" : "Rakesh", "age" : 21 }
>
```

Q7. Explain why delete_one(), delete_many(), and drop() is used.

Ans:
deleteOne() deletes the first document that matches the filter. Use a field that is part of a unique index such as _id for precise deletions.
delete_many() is used when one needs to delete more than one document. A query object containing which document to be deleted is created and is passed as the first parameter to the delete_many().
Syntax:
collection.delete_many(filter, collation=None, hint=None, session=None)
drop() method drops the specified collection and any internal collections related to encrypted fields.
syntax
db. collection. drop()