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

Answer :

MongoDB is a document database designed for ease of application development and scaling. it is a source-available cross platform document-oriented database program. It is classified as NoSQL databse program, MongoDB uses JSON like structured document.

unlike relational databse, Non-relational database stores data in whatever format, in which data is suitable. Non-relational databse stores unstructured data.

when we have unstructured, complex data, And when we have to handle lage amount of data and stores in as document. it is preferable to use MongoDB instead of SQL.

Q2. State and Explain the features of MongoDB.

Answer :

a) Replication :

MongoDB supports Master Slave replication.

A master can perform Reads and Writes and a Slave copies data from the master and can only be used for reads or back up (not writes)

b) Duplication of data

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.

c) Provides high performance.

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

Answer :

In [1]:
```
import pymongo
client = pymongo.MongoClient("mongodb+srv://pwskills:pwskills@cluster0.zrqbwnr
db = client.test
```

In [2]:
```
# Creating a databse in MongoDB
db = client["New_Databse"]
```

In [3]:
```
# Creating a collection in MongoDB
coll = db["New_Collection"]
```

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.

Answer :

In [5]:
```
data = { "hare":"krishna",
         "nimai":"nitai",
         "harshit":"chaudhary"}
```

In [6]:
```
coll.insert_one(data)
```

Out[6]:
```
<pymongo.results.InsertOneResult at 0x2427eccd150>
```

In [7]:
```
coll.find_one()
```

Out[7]:
```
{'_id': ObjectId('64d4eac7606a36699721e87f'),
 'hare': 'krishna',
 'nimai': 'nitai',
 'harshit': 'chaudhary'}
```

In [11]:
```
data1 = [{"hare":"krishna", "hari":"bol", "nimai":"nitai"},
         {"harshit":[{"chaudhary", "108" }], "education":"pursuing B.tech", "a
         {"bhagwat geeta":"108", "Bhagwatam":"18000"}]
```

In [9]:
```
coll.insert_many(data1)
```

Out[9]:
```
<pymongo.results.InsertManyResult at 0x2427fe50220>
```

In [10]:
```
for i in coll.find():
    print(i)
```

```
{'_id': ObjectId('64d4eac7606a36699721e87f'), 'hare': 'krishna', 'nimai': 'n
itai', 'harshit': 'chaudhary'}
{'_id': ObjectId('64d4eb57606a36699721e880'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai'}
{'_id': ObjectId('64d4eb57606a36699721e881'), 'harshit': 'chaudhary', 'educa
tion': 'pursuing B.tech', 'age': '18'}
{'_id': ObjectId('64d4eb57606a36699721e882'), 'bhagwat geeta': '108', 'Bhagw
atam': '18000'}
```

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

Answer :

In [21]:
```
random_data = [{"_id":"3", "company_name":"pwskills"},
               {"_id":"4", "company_name":"pwskills"},
               {"_id":"5", "company_name":"pwskills"}]
```

In [22]:
```
coll.insert_many(random_data)
```

Out[22]:
```
<pymongo.results.InsertManyResult at 0x2097e9a3d00>
```

In [25]:
```
for i in coll.find():
    print(i)
```

```
{'_id': ObjectId('64d4eac7606a36699721e87f'), 'hare': 'krishna', 'nimai': 'n
itai', 'harshit': 'chaudhary'}
{'_id': ObjectId('64d4eb57606a36699721e880'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai'}
{'_id': ObjectId('64d4eb57606a36699721e881'), 'harshit': 'chaudhary', 'educa
tion': 'pursuing B.tech', 'age': '18'}
{'_id': ObjectId('64d4eb57606a36699721e882'), 'bhagwat geeta': '108', 'Bhagw
atam': '18000'}
{'_id': ObjectId('64d4ee68606a36699721e883'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai', 'harshit': 'chaudhary', 'education': 'pursuing B.tec
h', 'age': '18', 'book': '18000'}
{'_id': ObjectId('64d4eed1f4176d048194f586'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai', 'harshit': 'chaudhary', 'education': 'pursuing B.tec
h', 'age': '18', 'book': '18000'}
{'_id': ObjectId('64d4ef2a95a555c8aa98295b'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai', 'harshit': 'chaudhary', 'education': 'pursuing B.tec
h', 'age': '18', 'book': '18000'}
{'_id': ObjectId('64d4efab95a555c8aa98295c'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai', 'harshit': 'chaudhary', 'education': 'pursuing B.tec
h', 'age': '18', 'book': '10000000', 'book_name': 'mahabharat'}
{'_id': ObjectId('64d4f07e95a555c8aa98295d'), 'hare': 'krishna', 'hari': 'bo
l', 'nimai': 'nitai'}
{'_id': ObjectId('64d4f07e95a555c8aa98295e'), 'harshit': 'chaudhary', 'educa
tion': 'pursuing B.tech', 'age': '18'}
{'_id': ObjectId('64d4f07e95a555c8aa98295f'), 'book': '108', 'book_name': 'B
hagwat Geeta'}
{'_id': ObjectId('64d4f07e95a555c8aa982960'), 'book': '18000', 'book_name':
'srimad bhagwatam'}
{'_id': ObjectId('64d4f07e95a555c8aa982961'), 'book': '10000000', 'book_nam
e': 'mahabharat'}
{'_id': '3', 'company_name': 'pwskills'}
{'_id': '4', 'company_name': 'pwskills'}
{'_id': '5', 'company_name': 'pwskills'}
```

In [ ]:
```
for i in coll.find({"_id", {"$gte":"4"}}):
    print(i)
```

In [28]:
```
#result will be :
#{'_id': '4', 'company_name': 'pwskills'}
#{'_id': '5', 'company_name': 'pwskills'}
```

Q6. Explain the sort() method. Give an example to

demonstrate sorting in MongoDB.

-

In [7]:
```
random_data1 = [{'_id': '3', 'company_name': 'pwskills'},
                {'_id': '4', 'company_name': 'tata'},
                {'_id': '5', 'company_name': 'haripriya'}]
```

In [ ]:
```
coll.insert_one(random_data1)
```

In [ ]:
```
coll.find().sort({"company_name":1})
```

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

Answer :

To delete one document, we use the delete_one() method.

To delete more than one document, we use the delete_many() method.

The drop() method removes collections from the database. It also removes all indexes associated with the dropped collection.