

17 Feb Ass

March 9, 2023

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

[]: ANS -

[]: MongoDB **is** a non-relational document database 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.

[]: 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 database use a
↳ storage model that **is** optimized
for the specific requirements of the **type** of data being stored.

[]: Organizations of **all** sizes are adopting MongoDB , especially **as** a cloud
↳ database, because it enables them to build
applications faster, handle highly diverse data types, **and** manage applications
↳ more efficiently at scale.

[]:

[]:

[]: Q2. State **and** Explain the features of MongoDB.

[]: ANS -

[]: MongoDB **is** an **open**-source document-oriented database that **is** designed to store
↳ a large scale of data **and** also allows
you to work **with** that very efficiently. It **is** categorized under the nonSQL
↳ database because the storage.

[]: Schema-less Database :

It is the great feature provided by the MongoDB. a
→ schema-less database means one collection
can hold different types of documents init. or in other words, in the MongoDB
→ base, a single collection can hold multiple
documents and these documents consist of the different number of
→ field, content, and the size.

[]: Document Oriented:

In MongoDB, all the data stored in the document instead of
→ the table like in RDBMS. In these documents
the data is stored in field instead in rows and columns which make the data
→ more flexible in comparison to RDBMS.

[]: Indexing :

In MongoDB database, every field in the documents is indexed with
→ primary and secondary indices this makes easier and
takes less time to get or search data from the pool of the data. If the data is
→ not indexed, then database search each
documents with the specified query which takes lots of time and not so
→ efficient.

[]: Scalability :

MongoDB provides horizontal scalability with the help of sharding.
→ sharding means a distributed data on the
multiple servers, here a large amount of data is partitioned into data chunks
→ using a shard key, it will also add new machines to a
running database.

[]: High Performance:

The high performance of MongoDB is very high and data
→ persistence as compared to another database due to its
features like scalability, indexing, replication.

[]: Replication :

MongoDB provides high availability and redundancy with the help of
→ replication, it creates multiple copies of the data
and sends these copies to a different server so that if one server fails, then the
→ data is retrieved from another server.

[]:

[]:

```
[ ]: Q3. Write a code to connect MongoDB to Python. Also, create a database and a
    ↪collection in MongoDB.
```

```
[ ]: ANS -
```

```
[24]: import pymongo

client = pymongo.MongoClient("mongodb+srv://shadabkhan:10000001@cluster0.
    ↪phuo5yf.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```

```
[25]: pip install pymongo
```

Requirement already satisfied: pymongo in /opt/conda/lib/python3.10/site-packages (4.3.3)

Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in /opt/conda/lib/python3.10/site-packages (from pymongo) (2.3.0)

Note: you may need to restart the kernel to use updated packages.

```
[26]: client
```

```
[26]: MongoClient(host=['ac-n52dmqv-shard-00-00.phuo5yf.mongodb.net:27017',
    'ac-n52dmqv-shard-00-02.phuo5yf.mongodb.net:27017', 'ac-n52dmqv-shard-00-01.phuo5yf.mongodb.net:27017'], document_class=dict, tz_aware=False, connect=True, retrywrites=True, w='majority', authsource='admin', replicaset='atlas-11yhrw-shard-0', tls=True)
```

```
[27]: client = pymongo.MongoClient("mongodb+srv://shadabkhan:10000001@cluster0.
    ↪phuo5yf.mongodb.net/?retryWrites=true&w=majority")
```

```
[28]: db = client['assignment']
```

```
[29]: coll_create = db["my_record"]
```

```
[30]: data = {"name" : "shadab",
    "class" : "data science master",
    "timing" : "flexi"
}
```

```
[31]: coll_create.insert_one(data)
```

```
[31]: <pymongo.results.InsertOneResult at 0x7fdd7068a380>
```

```
[32]: data1 = {"mail_id" : "shadab@gmail.com",
    "phone_number" : 23455665
}
```

```
[33]: coll_create.insert_one(data1)
```

```
[33]: <pymongo.results.InsertOneResult at 0x7fdd90671780>
```

```
[34]: data2 = {"list_course" : ["data science masters " , "web dev " , "java with_
↳DSA"],
            "mentor" : ["shadab" , "zishan"]}
```

```
[35]: coll_create.insert_one(data2)
```

```
[35]: <pymongo.results.InsertOneResult at 0x7fdd706a2b00>
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: 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 -
```

```
[37]: import pymongo
```

```
client = pymongo.MongoClient("mongodb+srv://shadabkhan:10000001@cluster0.
↳phuo5yf.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```

```
[38]: pip install pymongo
```

```
Requirement already satisfied: pymongo in /opt/conda/lib/python3.10/site-
packages (4.3.3)
```

```
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in
/opt/conda/lib/python3.10/site-packages (from pymongo) (2.3.0)
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
[39]: client
```

```
[39]: MongoClient(host=['ac-n52dmqv-shard-00-00.phuo5yf.mongodb.net:27017',
'ac-n52dmqv-shard-00-02.phuo5yf.mongodb.net:27017', 'ac-n52dmqv-
shard-00-01.phuo5yf.mongodb.net:27017'], document_class=dict, tz_aware=False,
connect=True, retrywrites=True, w='majority', authsource='admin',
replicaset='atlas-11yhrw-shard-0', tls=True)
```

```
[40]: client = pymongo.MongoClient("mongodb+srv://shadabkhan:10000001@cluster0.
↳phuo5yf.mongodb.net/?retryWrites=true&w=majority")
```

```
[41]: db = client['assignment']
```

```
[42]: coll_create = db["my_record"]
```

```
[43]: data = {"name" : "shadab",  
             "class" : "data science master",  
             "timing" : "flexi"  
}
```

```
[44]: coll_create.insert_one(data)
```

```
[44]: <pymongo.results.InsertOneResult at 0x7fdd706c3c40>
```

```
[45]: data1 = {"mail_id" : "shadab@gmail.com",  
              "phone_number" : 23455665  
}
```

```
[46]: coll_create.insert_one(data1)
```

```
[46]: <pymongo.results.InsertOneResult at 0x7fdd706c0d60>
```

```
[47]: data2 = {"list_course" : ["data science masters " , "web dev " , "java with_  
↪DSA"],  
              "mentor" : ["shadab" , "zishan"]}
```

```
[48]: coll_create.insert_one(data2)
```

```
[48]: <pymongo.results.InsertOneResult at 0x7fdd706a2bc0>
```

```
[49]: data3 = {  
      "name" : "notebook",  
      "qty" : 60,  
      "rating" : [ { "score" : 5 } , { "score" : 8 } ],  
      "size" : { "height" : 34, "width" : 5.6, "unit" : "in" },  
      "status" : "B",  
      "tags" : [ "college-ruled", "perforated"]  
}
```

```
[50]: coll_create.insert_one(data3)
```

```
[50]: <pymongo.results.InsertOneResult at 0x7fdd706c1000>
```

```
[52]: data4 = [  
      { "name": "shadab", "address": "adarsh nagar456" },  
      { "name": "zishan", "address": "mountain 43" },  
      { "name": "zoya", "address" : "lotus 45" },  
      { "name": "afsar", "address" : "nobody 56" },  
]
```

```

{ "name": "ayesha", "address" : "tree 3" },
{ "name": "aftab", "address" : "garden67" },
{ "name": "yesha", "address" : "city09" },
{ "name": "kishan", "address" : "tokyo" },
{ "name": "jay", "address" : "london" }
]

```

```
[53]: coll_create.insert_many(data4)
```

```
[53]: <pymongo.results.InsertManyResult at 0x7fdd7050be50>
```

```
[54]: for i in coll_create.find():
      print(i)
```

```

{'_id': ObjectId('640a13c3b8ff9f5686fc2deb'), 'name': 'shadab', 'class': 'data
science master', 'timing': 'flexi'}
{'_id': ObjectId('640a14b3b8ff9f5686fc2dec'), 'mail_id': 'shadab@gmail.com',
'phone_number': 23455665}
{'_id': ObjectId('640a159db8ff9f5686fc2ded'), 'list_course': ['data science
masters ', 'web dev ', 'java with DSA'], 'mentor': ['shadab', 'zishan']}
{'_id': ObjectId('640a1a08b8ff9f5686fc2df0'), 'name': 'shadab', 'class': 'data
science master', 'timing': 'flexi'}
{'_id': ObjectId('640a1a38b8ff9f5686fc2df1'), 'mail_id': 'shadab@gmail.com',
'phone_number': 23455665}
{'_id': ObjectId('640a1a6ab8ff9f5686fc2df2'), 'list_course': ['data science
masters ', 'web dev ', 'java with DSA'], 'mentor': ['shadab', 'zishan']}
{'_id': ObjectId('640a1c31b8ff9f5686fc2df3'), 'name': 'notebook', 'qty': 60,
'rating': [{'score': 5}, {'score': 8}], 'size': {'height': 34, 'width': 5.6,
'unit': 'in'}, 'status': 'B', 'tags': ['college-ruled', 'perforated']}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df4'), 'name': 'shadab', 'address':
'adarsh nagar456'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df5'), 'name': 'zishan', 'address':
'mountain 43'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df6'), 'name': 'zoya', 'address': 'lotus
45'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df7'), 'name': 'afsar', 'address':
'nobody 56'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df8'), 'name': 'ayesha', 'address': 'tree
3'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2df9'), 'name': 'aftab', 'address':
'garden67'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2dfa'), 'name': 'yesha', 'address':
'city09'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2dfb'), 'name': 'kishan', 'address':
'tokyo'}
{'_id': ObjectId('640a20a0b8ff9f5686fc2dfc'), 'name': 'jay', 'address':
'london'}

```

```
[55]: coll_create.find_one()
```

```
[55]: {'_id': ObjectId('640a13c3b8ff9f5686fc2deb'),  
      'name': 'shadab',  
      'class': 'data science master',  
      'timing': 'flexi'}
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q5. Explain how you can use the find() method to query the MongoDB database.␣  
      ↳Write a simple code to  
      demonstrate this.
```

```
[ ]: ANS -
```

```
[ ]: The find() method in MongoDB selects documents in a collection or view and␣  
      ↳return a cursor to the selected  
      documents. It has two parameters : query and projection.  
      - The first query or filter parameter specifies the search  
      - The second optional projection parameter specifies what parts of each␣  
      ↳matching documents are returned
```

```
[ ]: db.Customers.find()  
  
      //return all documents in the collection
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q6. Explain the sort() method. Give an example to demonstrate sorting in␣  
      ↳MongoDB.
```

```
[ ]: ANS -
```

```
[ ]: In MongoDB , sorting is done by the sort() methods. The sort() method consist of␣  
      ↳two basic building blocks. These building blocks  
      are field to be sorted and the sort order  
      The sorting order in MongoDB is defined by either a one or a minus here the␣  
      ↳positive one represents the ascending order, while the negative  
      one represents the descending order.
```

```
[ ]: db.collection_name.find().sort( { field_name: sort order })
```

[]:

[]:

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

[]: ANS -

[]: Delete methods are used to remove records from the collection, delete one
↳ deletes the first record that matches the filter
the filter and delete many removes every record that matches the filter

[]: The drop() function is used to remove A SET OF labels from a row or column. we
↳ may exclude rows or column by defining label name
and matching axes or directly defining index or column names. label on various
↳ levels may be removed by using a multi-index.

[]: