

# Beginners Guide : MongoDB Basics

- **About MongoDB :**

- Mongodb is a NoSQL database, NoSQL is stand for Not Only Structured Query Language
- NoSQL databases like Mongodb is put data in JSON format and SQL databases are stored data in tabular format.
- Data is in JSON format it means it is key values pair because of this data retrieval in NOSQL is very very faster as compair to SQL
- Table in Mongodb called as Collection and record is called as document
- The primary key is automatically set to the '\_id' field
- It support geospatial query it means it return longitude and latitude
- It is easily integrated with Big Data Hadoop
- It is easy to install, scalable ,high performance
- For High transaction data Mongodb not good

- **Connecting Python and Mongo**

In [11]:

```
# import driver
import pymongo

#connect with client
client = pymongo.MongoClient('mongodb://127.0.0.1:27017')

#return client details
client
```

Out[11]: MongoClient(host=['127.0.0.1:27017'], document\_class=dict, tz\_aware=False, connect=True)

- **Create Database**

In [16]:

```
db = client['Record']
```

- our database is created but it will not present in Mongo
- After creating collection and adding document then our database is visible

In [17]:

```
client.list_database_names()
```

Out[17]: ['admin', 'config', 'demo', 'local', 'rus\_first\_mongo']

- **Create Collection**

In [101...]

```
collection = db['Student']
```

```
collection
```

```
Out[101... Collection(Database(MongoClient(host=['127.0.0.1:27017']), document_class=dict, tz_aware=False, connect=True), 'Record'), 'Student')
```

- Now we create a collection lets check our database is visible or not

```
In [102... client.list_database_names()
```

```
Out[102... ['admin', 'config', 'demo', 'local', 'rus_first_mongo']
```

- Here we see that still our database and collection is not visible

## • **Insert Single Record**

```
In [103... # create a single record
```

```
record = {  
    'Name': 'Rushikesh',  
    'Education': 'BE',  
    'Branch': 'CSE',  
    'Phone No': 9090909090,  
}
```

```
# insert it into collection name as 'Student'  
collection.insert_one(record)
```

```
Out[103... <pymongo.results.InsertOneResult at 0x1990b733b40>
```

- Now, we inserted a single record in 'Student' collection
- Let's check our database is created or not

## • **Check Database and Collection**

- Check all databases

```
In [104... client.list_database_names()
```

```
Out[104... ['Record', 'admin', 'config', 'demo', 'local', 'rus_first_mongo']
```

- Here our database is created now
- Find list of Collection in 'Record' Database

```
In [105... db.list_collection_names()
```

```
Out[105... ['Student']
```

## • **Insert Multiple Record**

```
In [106... # create a List of dictionary with multiple records
list_records = [
    {
        'Name': 'Robert',
        'Education': ['Bsc', 'Msc', 'PhD'],
        'College': 'Howard University',
        'Age': 28,
        'Email' : 'robert@haward.com'
    },
    {
        'Name': 'Subadh',
        'Education': ('BE', 'M.Tech'),
        'City': ['Mumbai', 'Dadar'],
        'Age': 31,
        'Email': 'subadhab@bhai.jao',
        'Hobby': 'cricket'
    },
    {
        'Name': 'Rushikesh',
        'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'},
        'City': 'Amravati',
        'Age': 22
    },
    {
        'Name': 'Prajwal',
        'Education': {'College': 'GCOEA', 'Branch': 'CS'},
        'City': 'Pune',
        'Age': 21
    },
    {
        'Name': 'Tushar',
        'Education': {'College': 'CEOET', 'Branch': 'EXTC'},
        'City': 'Yavatmal',
        'Age': 21
    },
    {
        'Name': 'Shubhas',
        'Education': {'College': 'GHRCOEN', 'Branch': 'EEE'},
        'City': 'Nagpur',
        'Age': 23
    },
    {
        'Name': 'Shivam',
        'Education': {'College': 'IITB', 'Branch': 'IT'},
        'City': 'Banglore',
        'Age': 22
    },
]
```

```
In [107... #insert multiple record in collection
collection.insert_many(list_records)
```

```
Out[107... <pymongo.results.InsertManyResult at 0x1990bd2f1c0>
```

- Working on Query

- Find single record

```
In [108... collection.find_one()
```

```
Out[108... {'_id': ObjectId('60b08e255c8cb2e89ed95b35'),
 'Name': 'Rushikesh',
 'Education': 'BE',
 'Branch': 'CSE',
 'Phone No': 9090909090}
```

- Find all records

```
In [109... for i in collection.find():
    print(i,end='\n\n')
```

```
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 28, 'Email': 'robert@haward.com'}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 31, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket'}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b38'), 'Name': 'Rushikesh', 'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'}, 'City': 'Amravati', 'Age': 22}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b39'), 'Name': 'Prajwal', 'Education': {'College': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune', 'Age': 21}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3a'), 'Name': 'Tushar', 'Education': {'College': 'CEOET', 'Branch': 'EXTC'}, 'City': 'Yavatmal', 'Age': 21}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EEE'}, 'City': 'Nagpur', 'Age': 23}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3c'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore', 'Age': 22}
```

- Find top 4 record

```
In [122... for i in collection.find().limit(4):
    print(i)
```

```
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090, 'Graduation': 'Under Grad'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 25, 'Email': 'robert@haward.com'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 25, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket', 'Graduation': 'Under Grad'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b38'), 'Name': 'Rushikesh', 'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'}, 'City': 'Amravati', 'Age': 22}
```

- Skip first 4 record and return reemaining

```
In [135...]:  
for i in collection.find().skip(4):  
    print(i)  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b39'), 'Name': 'Prajwal', 'Education': {'College': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune', 'Age': 21}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3a'), 'Name': 'Tushar', 'Education': {'College': 'CEOET', 'Branch': 'EXTC'}, 'City': 'Yavatmal', 'Age': 21}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EE'}, 'City': 'Nagpur', 'Age': 23, 'lastModified': datetime.datetime(2021, 5, 28, 6, 32, 41, 236000)}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3c'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore', 'Age': 22}
```

- Find record with name 'Rushikesh'

```
In [110...]:  
for i in collection.find({'Name': 'Rushikesh'}):  
    print(i,end='\n\n')  
  
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b38'), 'Name': 'Rushikesh', 'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'}, 'City': 'Amravati', 'Age': 22}
```

- **Query operator:**

- \$in => In
- \$lt => less than
- \$gt => greater than
- \$nlt => not less than
- \$ngt => not greater than
- \$ne => No equal to
- \$nin => not in
- \$and => Binary and
- \$or => Binbary or
- \$set => use for updating

## - \$unset => use to deleting

- Find record whose education in [ BE , Bsc ]

```
In [111...: for i in collection.find({'Education':{'$in':['BE','Bsc']} }):  
      print(i,end='\n\n')  
  
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090}  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 28, 'Email': 'robert@haward.com'}  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 31, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket'}
```

- find record whose age is less than 35 and education is BE

```
In [112...: for i in collection.find({'Education':'BE','Age':{'$lt':35}}):  
      print(i)  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 31, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket'}
```

```
In [113...: for i in collection.find({'$and':[{'Education':'BE'},{'Age':{'$lt':35}}]}):  
      print(i)  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 31, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket'}
```

- Find record whose age is greater than equal to 23 and education is M.Tech

```
In [114...: for i in collection.find({'$or':[{'Education':'M.Tech'},{'Age':{'$gte':23}}]}):  
      print(i)  
  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 28, 'Email': 'robert@haward.com'}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 31, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket'}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EEE'}, 'City': 'Nagpur', 'Age': 23}
```

- find record whose Branch is [CS,IT] and age is greater than 20

```
In [99]: for i in collection.find({'Education.Branch':{'$in':['CS','IT']} }):  
      print(i)  
  
{'_id': ObjectId('60b07ece5c8cb2e89ed95b2a'), 'Name': 'Prajwal', 'Education': {'College': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune'}
```

```
{'_id': ObjectId('60b07ece5c8cb2e89ed95b2d'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore'}
{'_id': ObjectId('60b0865e5c8cb2e89ed95b31'), 'Name': 'Prajwal', 'Education': {'College': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune', 'Age': 21}
{'_id': ObjectId('60b0865e5c8cb2e89ed95b34'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore', 'Age': 22}
```

- Update Branch of 'Shubhas' to EE

```
In [115...]: collection.update_one(
    {'Name': 'Shubhas'},
    {'$set': {'Education.Branch': 'EE'},
     '$currentDate': {'lastModified': True}}
)
```

```
Out[115...]: <pymongo.results.UpdateResult at 0x1990bd8ca80>
```

```
In [116...]: for i in collection.find({'Name': 'Shubhas'}):
    print(i)
```

```
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EE'}, 'City': 'Nagpur', 'Age': 23, 'lastModified': datetime.datetime(2021, 5, 28, 6, 32, 41, 236000)}
```

- Check Age is greater than 25 then set it to 25

```
In [118...]: collection.update_many(
    {'Age': {'$gt': 25}},
    {'$set': {'Age': 25}}
)
```

```
Out[118...]: <pymongo.results.UpdateResult at 0x1990bdeba00>
```

```
In [120...]: for i in collection.find({}):
    print(i)
```

```
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090, 'Graduation': 'Under Grad'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 25, 'Email': 'robert@haward.com'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 25, 'Email': 'subadhab@bhai.jao', 'Hobby': 'Cricket', 'Graduation': 'Under Grad'}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b38'), 'Name': 'Rushikesh', 'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'}, 'City': 'Amravati', 'Age': 22}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b39'), 'Name': 'Prajwal', 'Education': {'College': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune', 'Age': 21}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3a'), 'Name': 'Tushar', 'Education': {'College': 'CEOET', 'Branch': 'EXTC'}, 'City': 'Yavatmal', 'Age': 21}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EE'}, 'City': 'Nagpur', 'Age': 23, 'lastModified': datetime.datetime(2021, 5, 28, 6, 32, 41, 236000)}
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3c'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore', 'Age': 22}
```

- If Education is BE then create a Graduation key and set Under Grad as values

```
In [119... collection.update_many(  
    {'Education': 'BE'},  
    {'$set': {'Graduation': 'Under Grad'}}  
)
```

```
Out[119... <pymongo.results.UpdateResult at 0x1990d1c3340>
```

```
In [121... for i in collection.find({'Education': 'BE'}):  
    print(i)
```

```
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090, 'Graduation': 'Under Grad'}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 25, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket', 'Graduation': 'Under Grad'}
```

- **Aggregate Operator**

- avg
- count
- sum
- min
- max

- **Find all distinct branches in collection documents**

```
In [131... db.Student.distinct('Education.Branch')
```

```
Out[131... ['CS', 'CSE', 'EE', 'EXTC', 'IT']
```

- **Find the count of person whose age is less than 23**

```
In [128... collection.find({'Age': {'$lt': 23}}).count()
```

```
<ipython-input-128-2a9aef5f7b61>:1: DeprecationWarning: count is deprecated. Use Collection.count_documents instead.  
collection.find({'Age': {'$lt': 23}}).count()
```

```
Out[128... 4
```

- **Sort Record base on Age**

```
In [138... for i in collection.find().sort('Age'):  
    print(i, end='\n\n')
```

```
{'_id': ObjectId('60b08e255c8cb2e89ed95b35'), 'Name': 'Rushikesh', 'Education': 'BE', 'Branch': 'CSE', 'Phone No': 9090909090, 'Graduation': 'Under Grad'}  
{'_id': ObjectId('60b08e2e5c8cb2e89ed95b39'), 'Name': 'Prajwal', 'Education': {'College': 'IIT Jodhpur', 'Degree': 'B.Tech', 'Field': 'Computer Science', 'GPA': 9.0}, 'Branch': 'ECE', 'Phone No': 9898989898, 'Graduation': 'Graduate'}
```

```

e': 'GCOEA', 'Branch': 'CS'}, 'City': 'Pune', 'Age': 21}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3a'), 'Name': 'Tushar', 'Education': {'College': 'CEOET', 'Branch': 'EXTC'}, 'City': 'Yavatmal', 'Age': 21}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b38'), 'Name': 'Rushikesh', 'Education': {'College': 'PRMCEAM', 'Branch': 'CSE'}, 'City': 'Amravati', 'Age': 22}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3c'), 'Name': 'Shivam', 'Education': {'College': 'IITB', 'Branch': 'IT'}, 'City': 'Banglore', 'Age': 22}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b3b'), 'Name': 'Shubhas', 'Education': {'College': 'GHRCOEN', 'Branch': 'EE'}, 'City': 'Nagpur', 'Age': 23, 'lastModified': datetime.datetime(2021, 5, 28, 6, 32, 41, 236000)}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b36'), 'Name': 'Robert', 'Education': ['Bsc', 'Msc', 'PhD'], 'College': 'Howard University', 'Age': 25, 'Email': 'robert@haward.com'}

{'_id': ObjectId('60b08e2e5c8cb2e89ed95b37'), 'Name': 'Subadh', 'Education': ['BE', 'M.Tech'], 'City': ['Mumbai', 'Dadar'], 'Age': 25, 'Email': 'subadhab@bhai.jao', 'Hobby': 'cricket', 'Graduation': 'Under Grad'}

```

- Find the record whose age is max

```
In [150...]: for i in collection.find({'_id': '$Name'}):
    print(i, end='\n\n')
```

- Return Name and count of City

```
In [169...]: a = collection.aggregate([
    [{'$group':
        {'_id': '$Name',
         'Total City': {"$sum": 1}}
    }
])
```

```
In [170...]: for i in a:
    print(i)

{'_id': 'Shivam', 'Total City': 1}
{'_id': 'Robert', 'Total City': 1}
{'_id': 'Tushar', 'Total City': 1}
{'_id': 'Rushikesh', 'Total City': 2}
{'_id': 'Subadh', 'Total City': 1}
{'_id': 'Prajwal', 'Total City': 1}
```

- Find Average of Age

```
In [188...]: age_sum = collection.aggregate([
    {
        '$group':
            {'_id': '$Name',
             "Age is": {"$avg": '$Age'}}
    }
])
```

```
for i in age_sum:  
    print(i)  
  
{'_id': 'Shivam', 'Age is': 22.0}  
{'_id': 'Robert', 'Age is': 25.0}  
{'_id': 'Tushar', 'Age is': 21.0}  
{'_id': 'Rushikesh', 'Age is': 22.0}  
{'_id': 'Subadh', 'Age is': 25.0}  
{'_id': 'Prajwal', 'Age is': 21.0}
```

- Delete Record whose Age is 23

```
In [189...]  
print('Before : ')  
for i in collection.find({'Age': 23}):  
    print(i)  
collection.delete_many({'Age': 23})  
print("After : ")  
for i in collection.find({'Age': 23}):  
    print(i)
```

Before :

After :

- Drop collection name as Student

```
In [190...]  
collection.drop()
```

- Check Collection in Record Database

```
In [191...]  
db.collection_names(0)
```

```
<ipython-input-191-128f7006629a>;1: DeprecationWarning: collection_names is deprecated.  
Use list_collection_names instead.  
db.collection_names(0)
```

Out[191...]

- Drop Database 'Record'

- No need to drop Database if there is no Collection in database then it drops automatically

- Check database

```
In [192...]  
client.list_database_names()
```

Out[192...]

```
['admin', 'config', 'demo', 'local', 'rus_first_mongo']
```

**Thank You !**

- Rushikesh