In [1]:

```
!pip install pymongo
```

```
Collecting pymongo
  Downloading pymongo-4.3.3-cp39-cp39-win_amd64.whl (382 kB)
     ---------------------------------------- 382.5/382.5 kB 1.3 MB/s eta 0:
00:00
Collecting dnspython<3.0.0,>=1.16.0
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
     ---------------------------------------- 283.7/283.7 kB 2.9 MB/s eta 0:
00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.3.0 pymongo-4.3.3
```

In [2]:

```
import pymongo
```

In [5]:

```
#connect to a client
client = pymongo.MongoClient("mongodb+srv://Euphor:mongodb@cluster0l.s2a7sxh.mongodb.net/
db = client.test
#now it will be showing 1 connection in mongodb atlas
#we can create as many clusters
```

Type *Markdown* and LaTeX: α^2

MONGO DB

> **It is a document based database i.e we are able to store data in a dictionary format or JSON(Java Script Object Notation) format.**
>
> **In SQL a table is created, But in MONGODB a collection is created inside which documents are stores(note: documents are not structured i.e it can store any king do data)**

Creating DATABASE

In [15]:

```
#creating db
db = client['First_db']
```

In [16]:

```
#creating collection inside dbcrea
collection =db["First_coll"]
```

In [17]:

```
#creating data for adding into class
data = {"name" : "asit",
        "class" : "data",
        "timing " : "flexi"
}
```

In [18]:

```
#inssertin above data into the collection
collection.insert_one(data)
```

Out[18]:

```
<pymongo.results.InsertOneResult at 0x1d4c35dda00>
```

In [19]:

```
#proof that MONGO db in unstructured
#data2 given below has diff structure than the above data
data2 = {"mail_id" : "sudh@gmail.com",
         "phone_number" :2343243242
}
collection.insert_one(data2)
```

Out[19]:

```
<pymongo.results.InsertOneResult at 0x1d4c2e9da90>
```

In [20]:

```
data3 = {"list_course" : ["data science masters " , "web dev " , "java with DSA"],
        "mentor" : ["sudhanshu" , "anurag" , "hyder"]}
collection.insert_one(data3)
```

Out[20]:

```
<pymongo.results.InsertOneResult at 0x1d4c39e24f0>
```

In [22]:

```
#As data4 is a list of data(i.e dictionaries) therfore use: insert_many
data4 = [
  { "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" },
  { "name": "Richard", "address": "Sky st 331" },
  { "name": "Susan", "address": "One way 98" },
  { "name": "Vicky", "address": "Yellow Garden 2" },
  { "name": "Ben", "address": "Park Lane 38" },
  { "name": "William", "address": "Central st 954" },
  { "name": "Chuck", "address": "Main Road 989" },
  { "name": "Viola", "address": "Sideway 1633" }
]
collection.insert_many(data4)
```

Out[22]:

```
<pymongo.results.InsertManyResult at 0x1d4c34fbd60>
```

In [23]:

```
data5 = {
 "name": "notebook",
 "qty": 50,
 "rating": [ { "score": 8 }, { "score": 9 } ],
 "size": { "height": 11, "width": 8.5, "unit": "in" },
 "status": "A",
 "tags": [ "college-ruled", "perforated"]
}
collection.insert_one(data5)
```

Out[23]:

```
<pymongo.results.InsertOneResult at 0x1d4c3f4fdf0>
```

In [25]:

```
list_of_records = [
    {'companyName': 'iNeuron',
     'product': 'Affordable AI',
     'courseOffered': 'Machine Learning with Deployment'},

    {'companyName': 'iNeuron',
     'product': 'Affordable AI',
     'courseOffered': 'Deep Learning for NLP and Computer vision'},

    {'companyName': 'iNeuron',
     'product': 'Master Program',
     'courseOffered': 'Data Science Masters Program'}
]
collection.insert_many(list_of_records)
```

Out[25]:

```
<pymongo.results.InsertManyResult at 0x1d4c3c68490>
```

In [26]:

```
for i in collection.find():
    print(i)
```

```
{'_id': ObjectId('641c22f0e955c7002d0f8794'), 'name': 'asit', 'class': 'da
ta', 'timing ': 'flexi'}
{'_id': ObjectId('641c23bfe955c7002d0f8795'), 'mail_id': 'sudh@gmail.com',
'phone_number': 2343243242}
{'_id': ObjectId('641c2435e955c7002d0f8796'), 'list_course': ['data scienc
e masters ', 'web dev ', 'java with DSA'], 'mentor': ['sudhanshu', 'anura
g', 'hyder']}
{'_id': ObjectId('641c24c0e955c7002d0f8797'), 'name': 'Amy', 'address': 'A
pple st 652'}
{'_id': ObjectId('641c24c0e955c7002d0f8798'), 'name': 'Hannah', 'address':
'Mountain 21'}
{'_id': ObjectId('641c24c0e955c7002d0f8799'), 'name': 'Michael', 'addres
s': 'Valley 345'}
{'_id': ObjectId('641c24c0e955c7002d0f879a'), 'name': 'Sandy', 'address':
'Ocean blvd 2'}
{'_id': ObjectId('641c24c0e955c7002d0f879b'), 'name': 'Betty', 'address':
'Green Grass 1'}
{'_id': ObjectId('641c24c0e955c7002d0f879c'), 'name': 'Richard', 'addres
s': 'Sky st 331'}
{'_id': ObjectId('641c24c0e955c7002d0f879d'), 'name': 'Susan', 'address':
'One way 98'}
{'_id': ObjectId('641c24c0e955c7002d0f879e'), 'name': 'Vicky', 'address':
'Yellow Garden 2'}
{'_id': ObjectId('641c24c0e955c7002d0f879f'), 'name': 'Ben', 'address': 'P
ark Lane 38'}
{'_id': ObjectId('641c24c0e955c7002d0f87a0'), 'name': 'William', 'addres
s': 'Central st 954'}
{'_id': ObjectId('641c24c0e955c7002d0f87a1'), 'name': 'Chuck', 'address':
'Main Road 989'}
{'_id': ObjectId('641c24c0e955c7002d0f87a2'), 'name': 'Viola', 'address':
'Sideway 1633'}
{'_id': ObjectId('641c24e0e955c7002d0f87a3'), 'name': 'notebook', 'qty': 5
0, 'rating': [{'score': 8}, {'score': 9}], 'size': {'height': 11, 'width':
8.5, 'unit': 'in'}, 'status': 'A', 'tags': ['college-ruled', 'perforate
d']}
{'_id': ObjectId('641c2500e955c7002d0f87a4'), 'companyName': 'iNeuron', 'p
roduct': 'Affordable AI', 'courseOffered': 'Machine Learning with Deployme
nt'}
{'_id': ObjectId('641c2500e955c7002d0f87a5'), 'companyName': 'iNeuron', 'p
roduct': 'Affordable AI', 'courseOffered': 'Deep Learning for NLP and Comp
uter vision'}
{'_id': ObjectId('641c2500e955c7002d0f87a6'), 'companyName': 'iNeuron', 'p
roduct': 'Master Program', 'courseOffered': 'Data Science Masters Progra
m'}
```

In [ ]:

```
```