Pymongo

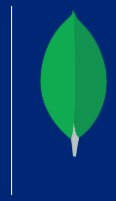
Commands

MongoDB

CRUD

In any database there are four fundamental operations which represents **CRUD**.

C → Create
R → Read
U → Update
D → Delete



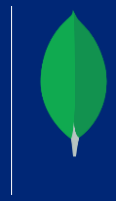
Create

MongoDB Atlas, PyMongo

MongoDB

Create Commands

S.No	Operation	MongoDB Shell (mongosh)	PyMongo (Python)
1	Show all databases	`show dbs`	`client.list_database_names()`
2	Use / switch database	`use school`	`db = client["school"]`
3	Show current database	`db`	`db.name`
4	Show collections	`show collections`	`db.list_collection_names()`
5	Create collection	`db.students.insertOne({})`	`db.students.insert_one({})`
6	Insert one document	`db.students.insertOne({name:"Amit"})`	`db.students.insert_one({"name":"Amit"})`
7	Insert many documents	`db.students.insertMany([{name:"A"}, {name:"B"}])`	`db.students.insert_many([{name:"A"}, {name:"B"}])`



Read

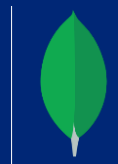
MongoDB Atlas, PyMongo

MongoDB

Read Commands

```
db.collection.find(
<filter>,
<projection>
)
```

S.No	Operation	MongoDB Shell (mongosh)	PyMongo (Python)
1	Show all databases	`show dbs`	`client.list_database_names()`
2	Show collections	`show collections`	`db.list_collection_names()`
3	Find all documents	`db.students.find()`	`db.students.find()`
4	Find one document	`db.students.findOne({name:"Amit"})`	`db.students.find_one({"name":"Amit"})`
5	Filter documents	`db.students.find({age: 20})`	`db.students.find({"age":20})`
6	Count documents	`db.students.countDocuments({})`	`db.students.count_documents({})`
7	Sort documents	`db.students.find().sort({age:1})` `1 → ascending order, -1 → descending order`	`db.students.find().sort({age:1})` `1 → ascending order, -1 → descending order`
8	Limit documents	`db.students.find().limit(5)`	`db.students.find().limit(5)`
9	Projection Select on specific fields	`db.students.find({},{name: 1, age: 1})`	`db.students.find({},{name: 1, age: 1})`



Read

Comparative Operators

MongoDB

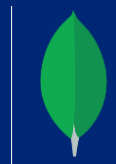
Comparative Operators

```
db.collection.find(
    {
        field: { $operator : value}
    }
)
```

MongoDB

Comparative Operators

Operator	Meaning	Example Query
$eq	Equal to	`db.emp.find({ age: { $eq: 35 } })`
$ne	Not equal to	`db.emp.find({ salary: { $ne: 75000 } })`
$gt	Greater than	`db.emp.find({ salary: { $gt: 40000 } })`
$gte	Greater than or equal to	`db.emp.find({ age: { $gte: 30 } })`
$lt	Less than	`db.emp.find({ age: { $lt: 30 } })`
$lte	Less than or equal to	`db.emp.find({ salary: { $lte: 50000 } })`
$in	In	`db.emp.find({ name: { $in: ["Ram", "Sita"] } })`
$nin	Not in	`db.emp.find({ name: { $nin: ["Ram", "Sita"] } })`



Read

Logical Operators

MongoDB

Logical Operators

```
db.collection.find(
    {
        $operator:[
                    condition1,
                    condition2,
                    …
                ]
    }
)
```

MongoDB

Logical Operators

Operator	Meaning	Description	Example Query
$and	Logical AND	Matches documents where **all conditions** are true	`db.emp.find({ $and: [ { age: { $gt: 30 }}, { salary: { $gte: 50000 }} ]})`
$or	Logical OR	Matches documents where **at least one condition** is true	`db.emp.find({ $or: [ { age: { $lt: 25 }}, { salary: { $gt: 80000 }} ]})`
$nor	Logical NOR	Matches documents where **none of the conditions** are true	`db.emp.find({ $nor: [ { age: { $lt: 25 } }, { salary: { $gt: 80000 }} ]})`
$not	Logical NOT	Matches documents where the condition is **not true**	`db.emp.find({ age: { $not: { $gte: 30 } } })`

Update commands

Modify existing documents in a collection

Update Commands in MongoDB

Update operations are used to **modify existing documents** in collection.

MongoDB mainly provides three update methods:

```
db.collection.update_one()

db.collection.update_many()

db.collection.replace_one()
```

MongoDB

Update One

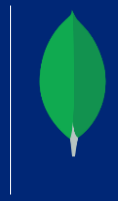
```
db.collection.update_one(
      <filter>,
      <update>,
      <options>…
)
```

MongoDB

Common Update Operators

Operator	Purpose	Example Query
$set	Set or update a field	db.emp.update_one({ name: { $eq: “Amit” } }, { $set: { age: 21} })
$unset	Remove a field	db.emp.update_one({ name: “Amit”}, { $unset: {phone: “”} })
$inc	Increment a numeric value	db.emp.update_one({ name: “Amit”}, { $inc: {age: 1} })
$push	Add value to array	db.enm.update_one({ name: "Amit" }, { $push: { skills: "Python" } })
$pull	Remove value from array	db.enm.update_one({ name: "Amit" }, { $pull: { skills: "Python" } })

`$set`	→	add / update field
`$unset`	→	remove field
`$inc`	→	increment number
`$push`	→	add to array
`$pull`	→	remove from array



update_many, replace_one

Modify existing documents in a collection

Update Commands in MongoDB

Update operations are used to **modify existing documents** in collection.

MongoDB mainly provides three update methods:

```
db.collection.update_one()

db.collection.update_many()

db.collection.replace_one()
```

Delete Commands

Remove documents or entire collection/databases from MongoDB

Delete Commands in MongoDB

Delete operations are used to **remove documents or entire collections/databases** in collection.

MongoDB mainly provides two main methods:

```
db.collection.delete_one()

db.collection.delete_many()
```

MongoDB

Delete One

```
db.collection.delete_one(
      <filter>,
      <options>…
)
```

MongoDB

Common Delete Operations

S.No	Operation	PyMongo
1	Delete one document	db.emp.delete_one({ "name": "Rohit" })
2	Delete many documents	db.emp.delete_many({ "currency": "inr" })
3	Delete by condition	db.emp.delete_many({ "salary": { "$lt": 30000 } })
4	Delete all documents	db.emp.delete_many({})
5	Delete using _id	db.emp.delete_one({ "_id": ObjectId("…") })
6	Drop collection	db.emp.drop()
7	Drop database	client.drop_database("employee")

VEDICSKILL.