

<https://www.mongodb.com/try/download/community>

<https://docs.mongodb.com/manual/reference/sql-comparison/>

What is MongoDB

[MongoDB](https://www.javatpoint.com/mongodb-tutorial) is an open-source document database that provides high performance, high availability, and automatic scaling.

In simple words, you can say that - Mongo DB is a document-oriented database. It is an open source product, developed and supported by a company named 10gen.

MongoDB is available under General Public license for free, and it is also available under Commercial license from the manufacturer.

The manufacturing company 10gen has defined MongoDB as:

"MongoDB is a scalable, open source, high performance, document-oriented database." - 10gen

MongoDB was designed to work with commodity servers. Now it is used by the company of all sizes, across all industry.

Advantages of NoSQL

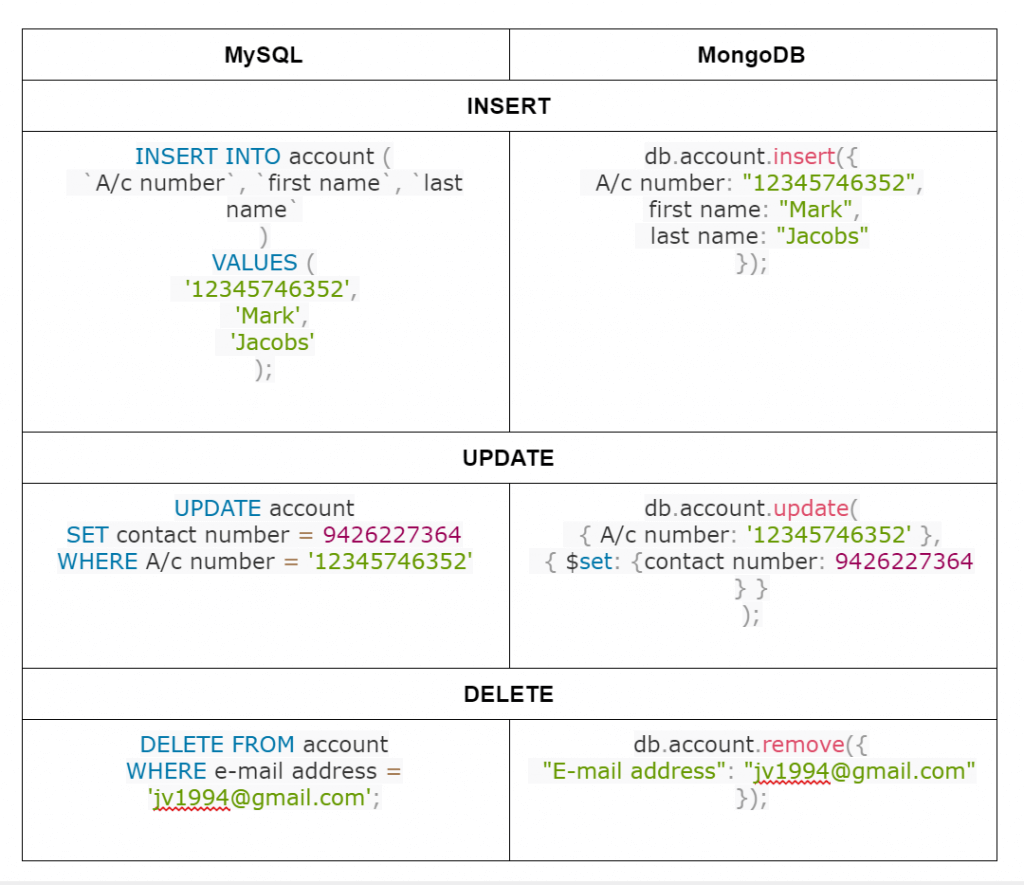
* It supports query language.
* It provides fast performance.
* It provides horizontal scalability.

MongoDB Advantages

* **MongoDB is schema less**. It is a document database in which one collection holds different documents.
* There may be **difference between number of fields, content and size of the document** from one to other.
* **Structure of a single object is clear** in MongoDB.
* There are **no complex joins** in MongoDB.
* MongoDB provides the **facility of deep query** because it supports a powerful dynamic query on documents.
* It is very **easy to scale**.
* It **uses internal memory for storing working sets** and this is the reason of its fast access.

Distinctive features of MongoDB

* Easy to use
* Light Weight
* Extremely faster than RDBMS



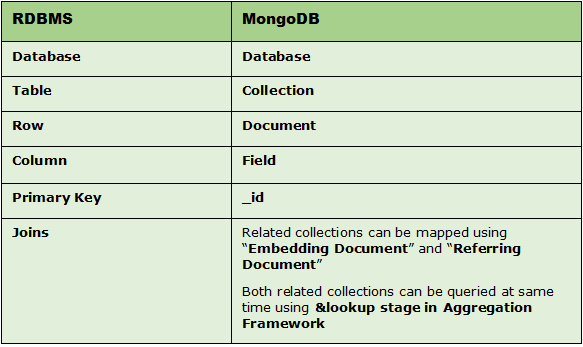
## PyMongo

Python needs a MongoDB driver to access the MongoDB database.

In this tutorial we will use the MongoDB driver "PyMongo".

We recommend that you use PIP to install "PyMongo".

PIP is most likely already installed in your Python environment.



pip install pymongo

## Creating a Database

## import pymongo myclient = pymongo.MongoClient("mongodb://localhost:27017/") mydb = myclient["mydatabase"]

## Insert Into Collection

import pymongo  
  
myclient = pymongo.MongoClient(**'mongodb://localhost:27017/'**)  
mydb = myclient[**'mydatabase'**]  
mycol = mydb[**"customers"**]  
  
mydict = { **"name"**: **"John"**, **"address"**: **"Highway 37"** }  
  
x = mycol.insert\_one(mydict)  
  
print(x)

## Return the \_id Field

mydict = { **"name"**: **"Peter"**, **"address"**: **"Lowstreet 27"** }  
x = mycol.insert\_one(mydict)  
print(x.inserted\_id)

## Insert Multiple Documents

import pymongo  
  
myclient = pymongo.MongoClient(**"mongodb://localhost:27017/"**)  
mydb = myclient[**"mydatabase"**]  
mycol = mydb[**"customers"**]  
  
mylist = [  
 { **"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"**}  
]  
  
x = mycol.insert\_many(mylist)  
  
*#print list of the \_id values of the inserted documents:*print(x.inserted\_ids)

## Insert Multiple Documents, with Specified IDs

import pymongo  
  
myclient = pymongo.MongoClient(**"mongodb://localhost:27017/"**)  
mydb = myclient[**"mydatabase"**]  
mycol = mydb[**"customers"**]  
  
mylist = [  
 { **"\_id"**: 1, **"name"**: **"John"**, **"address"**: **"Highway 37"**},  
 { **"\_id"**: 2, **"name"**: **"Peter"**, **"address"**: **"Lowstreet 27"**},  
 { **"\_id"**: 3, **"name"**: **"Amy"**, **"address"**: **"Apple st 652"**},  
 { **"\_id"**: 4, **"name"**: **"Hannah"**, **"address"**: **"Mountain 21"**},  
 { **"\_id"**: 5, **"name"**: **"Michael"**, **"address"**: **"Valley 345"**},  
 { **"\_id"**: 6, **"name"**: **"Sandy"**, **"address"**: **"Ocean blvd 2"**},  
 { **"\_id"**: 7, **"name"**: **"Betty"**, **"address"**: **"Green Grass 1"**},  
 { **"\_id"**: 8, **"name"**: **"Richard"**, **"address"**: **"Sky st 331"**},  
 { **"\_id"**: 9, **"name"**: **"Susan"**, **"address"**: **"One way 98"**},  
 { **"\_id"**: 10, **"name"**: **"Vicky"**, **"address"**: **"Yellow Garden 2"**},  
 { **"\_id"**: 11, **"name"**: **"Ben"**, **"address"**: **"Park Lane 38"**},  
 { **"\_id"**: 12, **"name"**: **"William"**, **"address"**: **"Central st 954"**},  
 { **"\_id"**: 13, **"name"**: **"Chuck"**, **"address"**: **"Main Road 989"**},  
 { **"\_id"**: 14, **"name"**: **"Viola"**, **"address"**: **"Sideway 1633"**}  
]  
  
x = mycol.insert\_many(mylist)  
  
*#print list of the \_id values of the inserted documents:*print(x.inserted\_ids)

# Python MongoDB Find

import pymongo  
  
myclient = pymongo.MongoClient(**"mongodb://localhost:27017/"**)  
mydb = myclient[**"mydatabase"**]  
mycol = mydb[**"customers"**]  
  
x = mycol.find\_one()  
print(x)

find all

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
for x in mycol.find():  
  print(x)

## Return Only Some Fields

The second parameter of the find() method is an object describing which fields to include in the result.

This parameter is optional, and if omitted, all fields will be included in the result.

### Example

Return only the names and addresses, not the \_ids:

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
for x in mycol.find({},{ "\_id": 0, "name": 1, "address": 1 }):  
  print(x)

## Filter the Result

import pymongo  
  
myclient = pymongo.MongoClient(**"mongodb://localhost:27017/"**)  
mydb = myclient[**"mydatabase"**]  
mycol = mydb[**"customers"**]  
  
myquery = { **"address"**: **"Park Lane 38"** }  
  
mydoc = mycol.find(myquery)  
  
for x in mydoc:  
 print(x)

## Advanced Query

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
myquery = { "address": { "$gt": "S" } }  
  
mydoc = mycol.find(myquery)  
  
for x in mydoc:  
  print(x)

sort

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
mydoc = mycol.find().sort("name")  
  
for x in mydoc:  
  print(x)

## Delete Document

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

The first parameter of the delete\_one() method is a query object defining which document to delete.

## import pymongo myclient = pymongo.MongoClient("mongodb://localhost:27017/") mydb = myclient["mydatabase"] mycol = mydb["customers"] myquery = { "address": "Mountain 21" } mycol.delete\_one(myquery)

## Delete Many Documents

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
myquery = { "address": {"$regex": "^S"} }  
  
x = mycol.delete\_many(myquery)  
  
print(x.deleted\_count, " documents deleted.")

## Delete All Documents in a Collection

To delete all documents in a collection, pass an empty query object to the delete\_many() method:

### Example

Delete all documents in the "customers" collection:

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
x = mycol.delete\_many({})  
  
print(x.deleted\_count, " documents deleted.")

## Delete Collection

You can delete a table, or collection as it is called in MongoDB, by using the drop() method.

### Example

Delete the "customers" collection:

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
mycol.drop()

# Python MongoDB Update

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
myquery = { "address": "Valley 345" }  
newvalues = { "$set": { "address": "Canyon 123" } }  
  
mycol.update\_one(myquery, newvalues)  
  
#print "customers" after the update:  
for x in mycol.find():  
  print(x)

## Update Many

To update all documents that meets the criteria of the query, use the update\_many() method.

### Example

Update all documents where the address starts with the letter "S":

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
myquery = { "address": { "$regex": "^S" } }  
newvalues = { "$set": { "name": "Minnie" } }  
  
x = mycol.update\_many(myquery, newvalues)  
  
print(x.modified\_count, "documents updated.")

## Update Many

To update all documents that meets the criteria of the query, use the update\_many() method.

### Example

Update all documents where the address starts with the letter "S":

import pymongo  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
mydb = myclient["mydatabase"]  
mycol = mydb["customers"]  
  
myquery = { "address": { "$regex": "^S" } }  
newvalues = { "$set": { "name": "Minnie" } }  
  
x = mycol.update\_many(myquery, newvalues)  
  
print(x.modified\_count, "documents updated.")

## Limit the Result

## import pymongo myclient = pymongo.MongoClient("mongodb://localhost:27017/") mydb = myclient["mydatabase"] mycol = mydb["customers"] myresult = mycol.find().limit(5) #print the result: for x in myresult:   print(x)