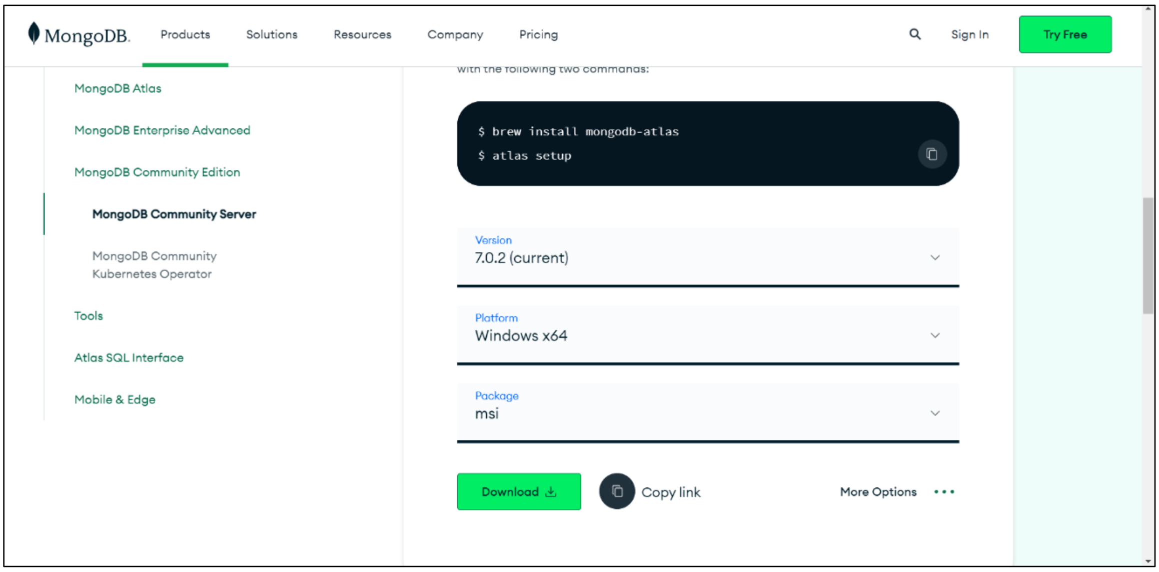
Practical 1	
Aim: To Demonstrate Installation of MongoDB	
Name: Saail Chavan	Roll No:
Performance date: 23 – 08 – 2023	Sign:

Visit the official website for MongoDB at https://www.mongodb.com/try/download/community.
On the homepage, look for the enticing "Download" button.



with the following two commands:

```
$ brew install mongodb-atlas
$ atlas setup
```

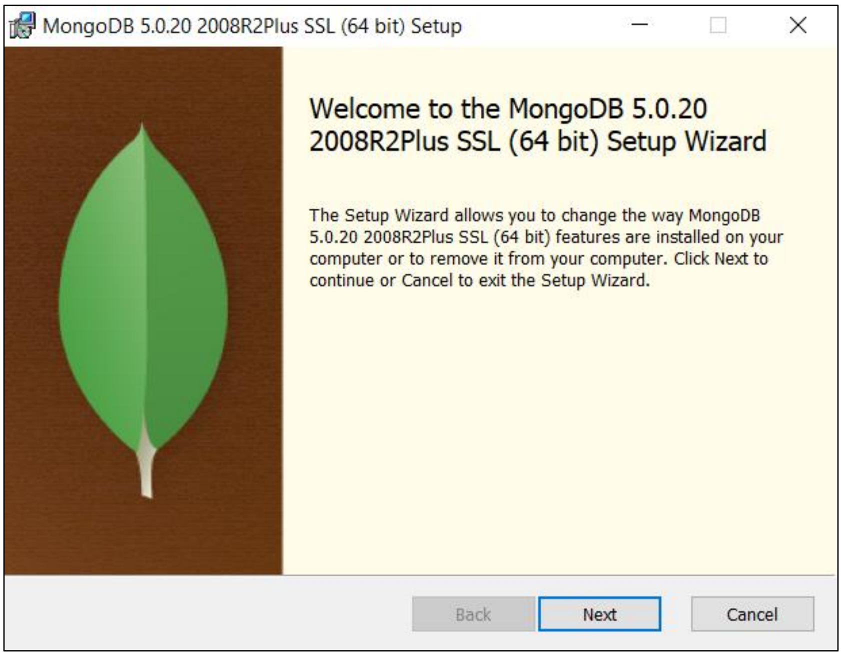
Version
7.0.2 (current)

Platform
Windows x64

Package
msi

Select the required version and platform, bellow that you'll spot a download link. Click on it to start download.

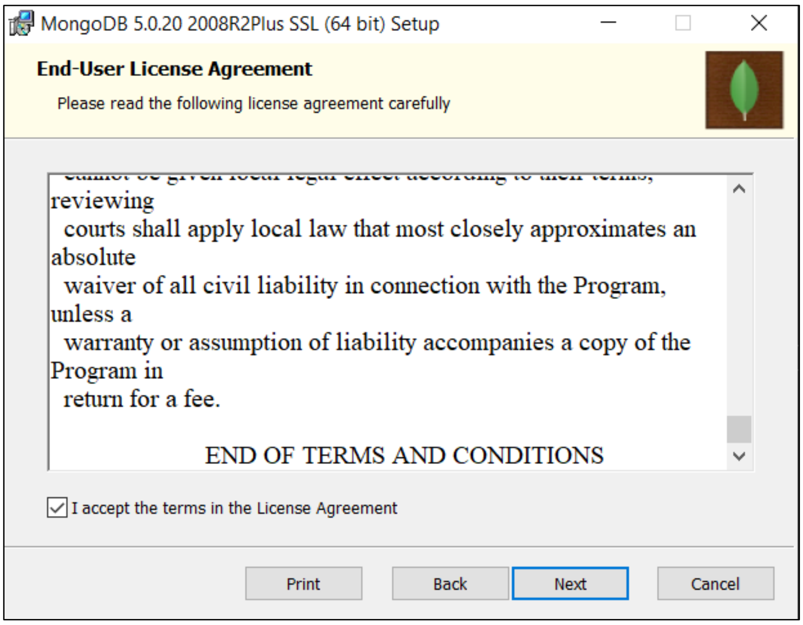
Once you have downloaded the Windows executable, start the installation.
Pick your preferred installation path and continue



Welcome to the MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard allows you to change the way MongoDB 5.0.20 2008R2Plus SSL (64 bit) features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Accept the terms in the License Agreement and select Next



MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup

End-User License Agreement

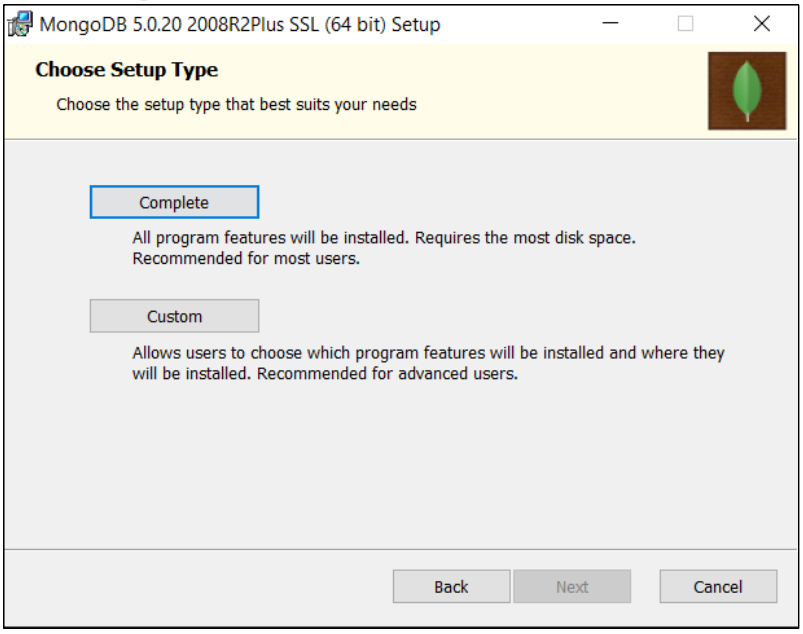
Please read the following license agreement carefully

cannot be given local legal effect according to their terms,
reviewing
 courts shall apply local law that most closely approximates an
absolute
 waiver of all civil liability in connection with the Program,
unless a
 warranty or assumption of liability accompanies a copy of the
Program in
 return for a fee.

END OF TERMS AND CONDITIONS

☑ I accept the terms in the License Agreement

Select Complete:



MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup

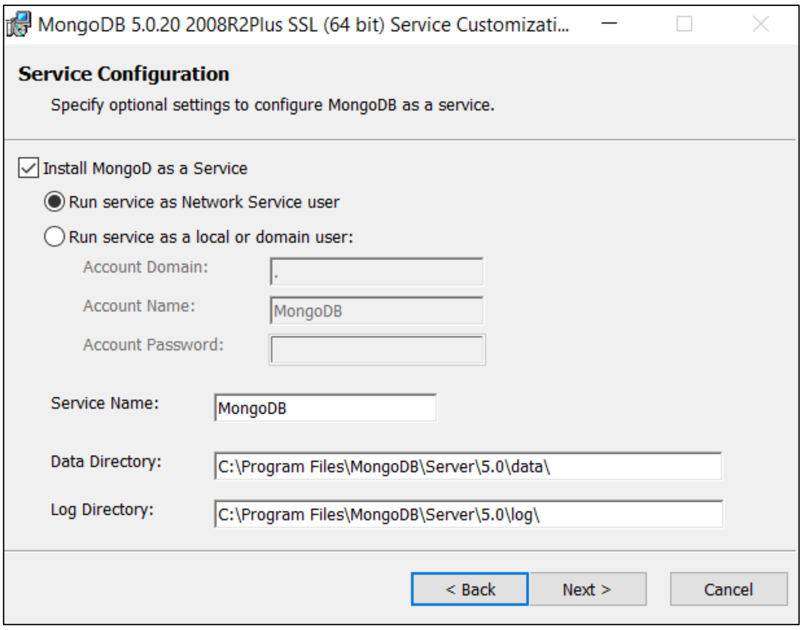
Choose Setup Type

Choose the setup type that best suits your needs

All program features will be installed. Requires the most disk space.
Recommended for most users.

Allows users to choose which program features will be installed and where they
will be installed. Recommended for advanced users.

Select the path for installation



MongoDB 5.0.20 2008R2Plus SSL (64 bit) Service Customizati...

Service Configuration

Specify optional settings to configure MongoDB as a service.

☑ Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

Account Name: MongoDB

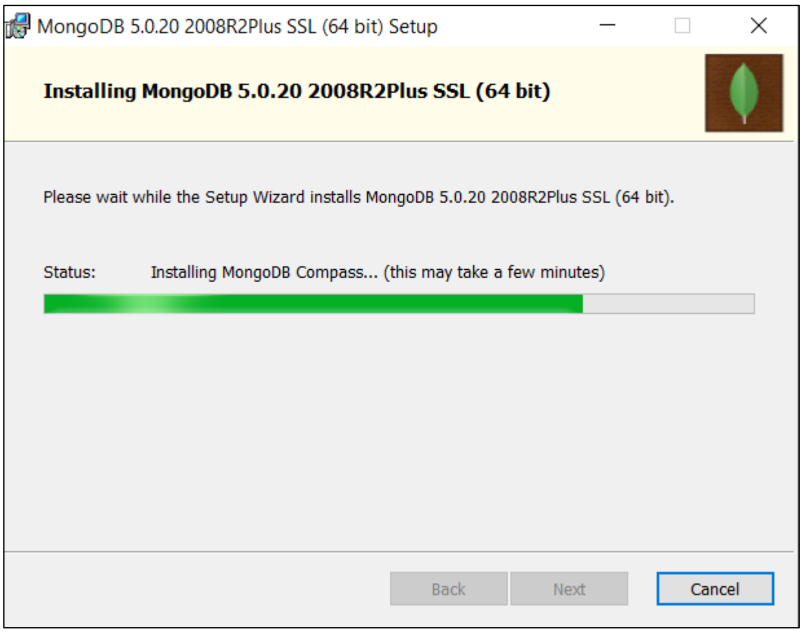
Account Password:

Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\5.0\data\

Log Directory: C:\Program Files\MongoDB\Server\5.0\log\

Then click on next and select install and wait

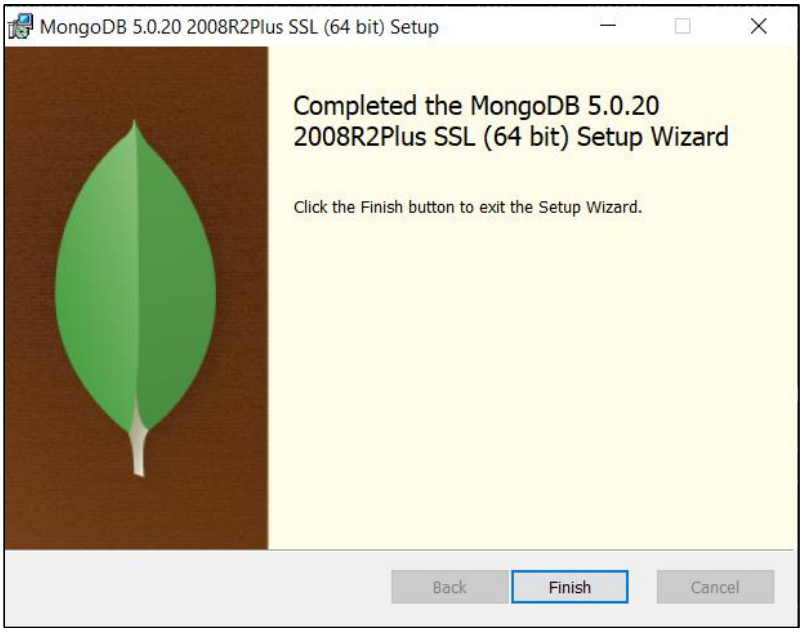

MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup

Installing MongoDB 5.0.20 2008R2Plus SSL (64 bit)

Please wait while the Setup Wizard installs MongoDB 5.0.20 2008R2Plus SSL (64 bit).

Status: Installing MongoDB Compass... (this may take a few minutes)

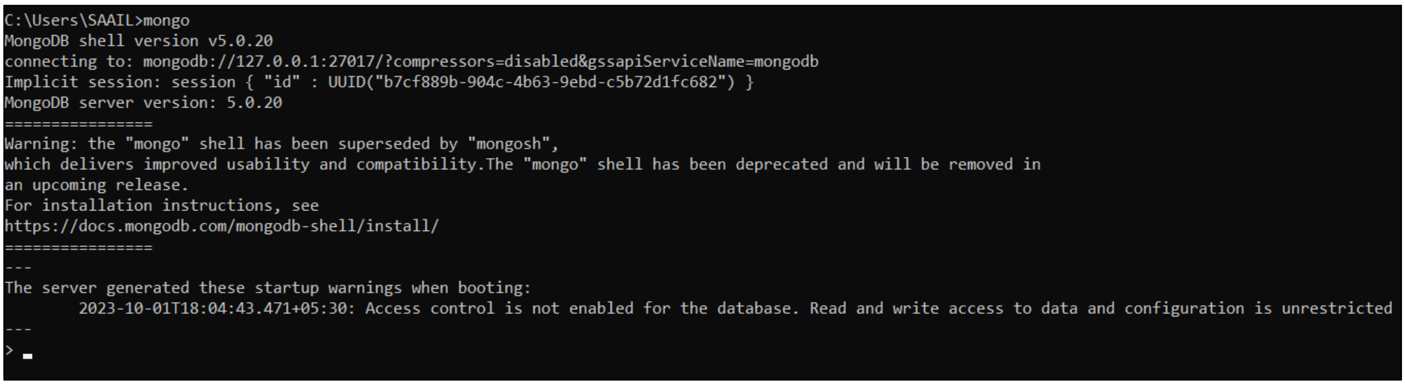
Click on Finish


MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup

Completed the MongoDB 5.0.20 2008R2Plus SSL (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

Once the installation is completed Open Command Line and type “mongo” or “mongosh” to check if its installed.


```
C:\Users\SAAIL>mongo
MongoDB shell version v5.0.20
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("b7cf889b-904c-4b63-9ebd-c5b72d1fc682") }
MongoDB server version: 5.0.20
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2023-10-01T18:04:43.471+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
> _
```

We’ve successfully installed MongoDB

<table>
<tr><td colspan="2"><b>Practical 2</b></td></tr>
<tr><td colspan="2"><b><u>Aim</u></b>: To Perform CRUD Operations on Student Data in MongoDB</td></tr>
<tr><td>Name: Saail Chavan</td><td>Roll No:</td></tr>
<tr><td>Performance date: 23 – 08 – 2023</td><td>Sign:</td></tr>
</table>

<u>Cmd (In Command Prompt) :</u>

```
C:\Users\HP>mongo MongoDB shell
version v5.0.20connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb Implicit session:
session {"id" : UUID("ef4b909f-6dce-4e52-90d8-328bdc1e9502") }MongoDB server version: 5.0.20
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecatedand will be
removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2023-08-23T15:34:38.319+05:30: Access control is not enabled for the database. Readand write
access to data and configuration is unrestricted
---
> use msc1;
switched to db msc1

> db;
msc1
```

Insert Queries:

1. **Insert a document with roll number 1, name 'Saail', and marks 800.**

   ```
   >db.msc1.insert({rno: 1, name: 'Saail', marks: 600});

   WriteResult({ "nInserted" : 1 })
   ```

2. **Insert multiple documents:**

   ```
   >db.msc1.insertMany([{rno: 2, name: 'Durgesh', marks: 750},{rno: 3, name: 'Rahul', marks:
   540},{rno: 4, name: 'Shruti', marks: 250},{rno: 5, name: 'Ruheen', marks: 900}]);
   ```

```
{  "acknowledged" : true,
        "insertedIds" : [
                ObjectId("64f75c9d81759736f5ea4341"),
                ObjectId("64f75c9d81759736f5ea4342"),
                ObjectId("64f75c9d81759736f5ea4343"),
                ObjectId("64f75c9d81759736f5ea4344")]}
```

Find Queries:

1. **Find all documents in the 'msc1' collection:**
   >db.msc1.find();

```
{ "_id" : ObjectId("64f75d0581759736f5ea4346"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f75d0581759736f5ea4347"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d0581759736f5ea4348"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d0581759736f5ea4349"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
{ "_id" : ObjectId("64f75d0c81759736f5ea434a"), "rno" : 1, "name" : "Saail", "marks" : 600 }
```

2. **Find the first document in the 'msc1' collection:**
   >db.msc1.findOne();

```
{
        "_id" : ObjectId("64f7551a81759736f5ea433c"),
        "rno" : 1,
        "name" : "Saail",
        "marks" : 600
}
```

3. **Find the second document in the 'msc1' collection.**
   >db.msc1.find().limit(2);

```
{ "_id" : ObjectId("64f7551a81759736f5ea433c"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f7556e81759736f5ea433d"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
```

4. **display all documents after skipping the first two documents**
   >db.msc1.find().skip(2);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f7556e81759736f5ea4340"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

---

5. **Skip and limit:**

   a. **displaying them in a pretty format:**
      >db.msc1.find().skip(2).pretty();

```
{       "_id" : ObjectId("64f7556e81759736f5ea433e"),
        "rno" : 3,
        "name" : "Rahul",
        "marks" : 540}
{
       "_id" : ObjectId("64f7556e81759736f5ea433f"),
        "rno" : 4,
        "name" : "Shruti",
        "marks" : 250
}
{         "_id" : ObjectId("64f7556e81759736f5ea4340"),
        "rno" : 5,
        "name" : "Ruheen",
        "marks" : 900
}
```

b. **Skip the first document and limit the result to 1 document.**
>db.msc1.find().skip(1).limit(1);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433d"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
```

c. **Skip the first document and limit the result to 2 documents.**
>db.msc1.find().skip(1).limit(2);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433d"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
```

d. **Skip the first two documents and limit the result to 2 documents.**
>db.msc1.find().skip(2).limit(2);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

e. **Skip the first three documents and limit the result to 1 document.**
>db.msc1.find().skip(3).limit(1);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

f. **Skip the first three documents and limit the result to 3 documents."**
>db.msc1.find().skip(3).limit(3);

```
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f7556e81759736f5ea4340"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

---

6. **Count: Find the count of documents in the 'msc1' collection**
>db.msc1.count();

```
5
```

---

7. **Find all documents in the 'msc1' collection with:**

a. **Roll Number 3:**
>db.msc1.find({rno: 3});

```
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
```

b. **Roll number 4:**
>db.msc1.find({rno: 4});

```
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

c. **Roll number 1 or 5:**
>db.msc1.find({$or:[{rno:{$eq:1}},{rno:{$eq:5}}]});

```
{ "_id" : ObjectId("64f7551a81759736f5ea433c"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f7556e81759736f5ea4340"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

d. roll numbers greater than or equal to 3:

```
>db.msc1.find({rno:{$gte:3}});

{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f7556e81759736f5ea4340"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

e. roll number 1 and 3:

```
>db.msc1.find({rno:{$in:[1,3]}});

{ "_id" : ObjectId("64f7551a81759736f5ea433c"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
```

f. roll numbers other than 1 and 3:

```
>db.msc1.find({rno:{$nin:[1,3]}});

{ "_id" : ObjectId("64f7551a81759736f5ea433c"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f7556e81759736f5ea433e"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
```

g. roll numbers other than 4 and 5:

```
>db.msc1.find({$nor:[{rno:{$eq:4}},{rno:{$eq:5}}]});

{ "_id":ObjectId("64f7551a81759736f5ea433c"),"rno":1,"name":"Saail","marks": 600 }
{"_id":ObjectId("64f7556e81759736f5ea433d"),"rno":2,"name":"Durgesh","marks": 750 }
{"_id":ObjectId("64f7556e81759736f5ea433e"),"rno":3,"name":"Rahul","marks": 540 }
```

---

8. Find all documents in the 'msc1' collection with MARKS:

a. between 650 and 750 (inclusive).

```
>db.msc1.find({$and:[{marks:{$gte:650}},{marks:{$lte:750}}]});

{ "_id" : ObjectId("64f7556e81759736f5ea433d"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f7556e81759736f5ea433f"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

b. less than 650?"

```
>db.msc1.find({marks:{$not:{$gte:650}}});

{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }

>db.msc1.find({marks:{$lte:650}});

{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

c. divisible by 200.

```
>db.msc1.find({marks:{$mod:[200,0]}});

{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
```

d. **that exist and are either 600 or 750.**
>db.msc1.find({marks:{$exists:true,$in:[600,750]}});

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
```

e. **that exist and are either 900 or 1750**
>db.msc1.find({marks:{$exists:true,$in:[900,1750]}});

```
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

---

9. **REGEX: Find all documents in the 'msc1' collection with names:**

a. **Starting with the letter 'S':**
>db.msc1.find({name:{$regex:/^S/}});

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

b. **Ending with the letter 'n':**
>db.msc1.find({name:{$regex:/n$/}});

```
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

c. **Ending with the letter 'h':**
>db.msc1.find({name:{$regex:/h$/}});

```
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
```

---

10. **Sorting: Find all documents in the 'msc1' collection, sorted by:**

a. **name in ascending order:**
>db.msc1.find().sort({name:1});

```
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

b. **roll number in ascending order:**
>db.msc1.find().sort({rno:1});

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

c. **marks in descending order:**
>db.msc1.find().sort({marks:-1});

```
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

d. roll number in descending order:

```
>db.msc1.find().sort({rno:-1});
```

```
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Durgesh", "marks" : 750 }
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
```

---

11. UPDATE Query :

a. Update the document with roll number 2 to change the name to 'Vikram':

```
>db.msc1.update({rno:2},{$set:{name:"Vikram"}});
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.msc1.find();
```

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Vikram", "marks" : 750 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

b. Update all documents with marks 750 to change their marks to 770:

```
>db.msc1.update({marks:750},{$set:{marks:770}});
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.msc1.find();
```

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 600 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Vikram", "marks" : 770 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 900 }
```

c. Update all documents with marks 900 or 600 to change their marks to 500:

```
>db.msc1.updateMany({marks:{$in:[750,600]}},{$set:{marks:500}});
```

```
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
```

```
>db.msc1.find();
```

```
{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 500 }
{ "_id" : ObjectId("64f75d5281759736f5ea434c"), "rno" : 2, "name" : "Vikram", "marks" : 770 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 500 }
```

---

12. Delete Query:

a. **Delete the document with roll number 2:**

```
>db.msc1.deleteOne({rno:2});
{ "acknowledged" : true, "deletedCount" : 1 }

>db.msc1.find();

{ "_id" : ObjectId("64f75d4c81759736f5ea434b"), "rno" : 1, "name" : "Saail", "marks" : 500 }
{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
{ "_id" : ObjectId("64f75d5281759736f5ea434f"), "rno" : 5, "name" : "Ruheen", "marks" : 500 }
```

b. **Delete all documents with marks 500:**

```
>db.msc1.deleteMany({marks:500});

{ "acknowledged" : true, "deletedCount" : 2 }
>db.msc1.find();

{ "_id" : ObjectId("64f75d5281759736f5ea434d"), "rno" : 3, "name" : "Rahul", "marks" : 540 }
{ "_id" : ObjectId("64f75d5281759736f5ea434e"), "rno" : 4, "name" : "Shruti", "marks" : 250 }
```

Insert Queries:

```
>db.msc1.insert({rno:5,name:'Rohan',marks:'750',hobbies:['writing','reading','singing']});
>db.msc1.insert({rno:1,name:'Apurva',marks:'700',hobbies:['gaming','sleeping','singing']});
>db.msc1.insert({rno:2,name:'Saail',marks:'750',hobbies:['gaming','Gym','dancing']});
>db.msc1.insert({rno:3,name:'Sahid',marks:'760',hobbies:['gaming','singing','dancing']});
```

Find Queries: Find all documents in the 'msc1' collection where:

1. **hobbies include both 'gaming' and 'dancing':**

```
>db.msc1.find({hobbies:{$all:['gaming', 'dancing']}});

{ "_id" : ObjectId("64f7610081759736f5ea4355"), "rno" : 2, "name" : "Saail", "marks" : "750",
"hobbies" : [ "gaming", "Gym", "dancing" ] }
{ "_id" : ObjectId("64f7610881759736f5ea4356"), "rno" : 3, "name" : "Sahid", "marks" : "760",
"hobbies" : [ "gaming", "singing", "dancing" ] }
```

2. **marks are greater than 750:**

```
>db.msc1.find({$expr:{$gt:["$marks","750"]}});

{ "_id" : ObjectId("64f7610881759736f5ea4356"), "rno" : 3, "name" : "Sahid", "marks" : "760",
"hobbies" : [ "gaming", "singing", "dancing" ] }
```

3. **greater than or equal to 750:**

```
>db.msc1.find({$expr:{$gte:["$marks","750"]}});

{ "_id" : ObjectId("64f760ef81759736f5ea4353"), "rno" : 5, "name" : "Rohan", "marks" : "750",
"hobbies" : [ "writing", "reading", "singing" ] }
{ "_id" : ObjectId("64f7610081759736f5ea4355"), "rno" : 2, "name" : "Saail", "marks" : "750",
"hobbies" : [ "gaming", "Gym", "dancing" ] }
{ "_id" : ObjectId("64f7610881759736f5ea4356"), "rno" : 3, "name" : "Sahid", "marks" : "760",
"hobbies" : [ "gaming", "singing", "dancing" ] }
```

```
>use uni
>db.uni.insert([{country : 'Spain', city : 'Salamanca', name : 'USAL', location : { type : 'Point', coordinates : [ -5.6722512, 40.9607792 ] }, students : [ { year : 2014, number : 24774 }, { year : 2015, number : 23166 }, { year : 2016, number : 21913 },{ year : 2017, number : 21715] }}, { country : 'Spain', city : 'Salamanca', name : 'UPSA', location : { type : 'Point', coordinates : [ -5.6691191, 40.9631732 ] }, students : [ { year : 2014, number : 4788 }, { year : 2015, number : 4821 }, { year : 2016, number : 6550 }, { year : 2017, number : 6125 }]}]);

BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
```

```
//New DB courses;

>use courses
>db.courses.insert([ { university : 'USAL', name : 'Computer Science', level : 'Excellent' }, { university : 'USAL', name : 'Electronics', level : 'Intermediate' }, {university : 'USAL', name : 'Communication', level : 'Excellent' }]);

BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 3,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})

>db.courses.find();

{ "_id" : ObjectId("65198934627e32f113e6e7d9"), "university" : "USAL", "name" : "Computer Science", "level" : "Excellent" }
{ "_id" : ObjectId("65198934627e32f113e6e7da"), "university" : "USAL", "name" : "Electronics", "level" : "Intermediate" }
{ "_id" : ObjectId("65198934627e32f113e6e7db"), "university" : "USAL", "name" : "Communication", "level" : "Excellent" }
```

Aggregation Queries:

```
>db.uni.find();
```

```
{ "_id" : ObjectId("65198819627e32f113e6e7d7"), "country" : "Spain", "city" : "Salamanca",
"name" : "USAL", "location" : { "type" : "Point", "coordinates" : [ -5.6722512, 40.9607792 ] },
"students" : [ { "year" : 2014, "number" : 24774 }, { "year" : 2015, "number" : 23166 }, { "year" :
2016, "number" : 21913 }, { "year" : 2017, "number" : 21715 } ] }
{ "_id" : ObjectId("65198819627e32f113e6e7d8"), "country" : "Spain", "city" : "Salamanca",
"name" : "UPSA", "location" : { "type" : "Point", "coordinates" : [ -5.6691191, 40.9631732 ] },
"students" : [ { "year" : 2014, "number" : 4788 }, { "year" : 2015, "number" : 4821 }, { "year" :
2016, "number" : 6550 }, { "year" : 2017, "number" : 6125 } ] }
```

1. Aggregate to find documents matching the country 'Spain' and city 'Salamanca':

```
>db.uni.aggregate([{$match:{country:'Spain',city:'Salamanca'}}]);
```

```
{ "_id" : ObjectId("65198819627e32f113e6e7d7"), "country" : "Spain", "city" : "Salamanca",
"name" : "USAL", "location" : { "type" : "Point", "coordinates" : [ -5.6722512, 40.9607792 ] },
"students" : [ { "year" : 2014, "number" : 24774 }, { "year" : 2015, "number" : 23166 }, { "year" :
2016, "number" : 21913 }, { "year" : 2017, "number" : 21715 } ] }
{ "_id" : ObjectId("65198819627e32f113e6e7d8"), "country" : "Spain", "city" : "Salamanca",
"name" : "UPSA", "location" : { "type" : "Point", "coordinates" : [ -5.6691191, 40.9631732 ] },
"students" : [ { "year" : 2014, "number" : 4788 }, { "year" : 2015, "number" : 4821 }, { "year" :
2016, "number" : 6550 }, { "year" : 2017, "number" : 6125 } ] }
```

2. Aggregate to project only the 'country' field:

```
>db.uni.aggregate([{$project:{country:1}}]);
```

```
{ "_id" : ObjectId("65198819627e32f113e6e7d7"), "country" : "Spain" }
{ "_id" : ObjectId("65198819627e32f113e6e7d8"), "country" : "Spain" }
```

3. Aggregate to project 'country' and 'city' fields:

```
>db.uni.aggregate([{$project:{country:1,city:1}}]);
```

```
{ "_id" : ObjectId("65198819627e32f113e6e7d7"), "country" : "Spain", "city" : "Salamanca" }
{ "_id" : ObjectId("65198819627e32f113e6e7d8"), "country" : "Spain", "city" : "Salamanca" }
```

4. Aggregate to project 'country', 'city', and 'name' fields:

```
>db.uni.aggregate([{$project:{country:1,city:1,name:1}}]);
```

```
{ "_id" : ObjectId("65198819627e32f113e6e7d7"), "country" : "Spain", "city" : "Salamanca",
"name" : "USAL" }
{ "_id" : ObjectId("65198819627e32f113e6e7d8"), "country" : "Spain", "city" : "Salamanca",
"name" : "UPSA" }
```

5. Aggregate to group documents by 'name' and calculate the total number of documents for each group:

```
>db.uni.aggregate([{$group:{_id:'$name',totalDocs:{$sum:1}}}]);
```

```
{ "_id" : "USAL", "totalDocs" : 1 }
{ "_id" : "UPSA", "totalDocs" : 1 }
```

6. Aggregate to project 'country', 'city', and 'name' fields while excluding the '_id' field:

```
>db.uni.aggregate([{$project:{_id:0,country:1,city:1,name:1}}]);
```

```
{ "country" : "Spain", "city" : "Salamanca", "name" : "USAL" }
{ "country" : "Spain", "city" : "Salamanca", "name" : "UPSA" }
```

7. Aggregate to group documents by 'name', calculate the total number of documents for each group, and output the result to a new collection 'mydata':

```
>db.uni.aggregate([{$group:{_id:'$name',totalDocs:{$sum:1}}},{$out:'mydata'}]);
>db.mydata.find();
```

```
{ "_id" : "UPSA", "totalDocs" : 1 }
{ "_id" : "USAL", "totalDocs" : 1 }
```

8. Aggregate to match documents with 'name' equal to 'USAL', unwind the 'students' array, and pretty print the result:

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'}]).pretty();
```

```
{
        "_id" : ObjectId("65198819627e32f113e6e7d7"),
        "country" : "Spain",
        "city" : "Salamanca",
        "name" : "USAL",
        "location" : {
                "type" : "Point",
                "coordinates" : [
                        -5.6722512,
                        40.9607792
                ]
        },
        "students" : {
                "year" : 2014,
                "number" : 24774
        }
}
{
        "_id" : ObjectId("65198819627e32f113e6e7d7"),
        "country" : "Spain",
        "city" : "Salamanca",
        "name" : "USAL",
        "location" : {
                "type" : "Point",
                "coordinates" : [
                        -5.6722512,
                        40.9607792
                ]
        },
        "students" : {
                "year" : 2015,
                "number" : 23166
        }
}
{
        "_id" : ObjectId("65198819627e32f113e6e7d7"),
    "country" : "Spain",
    "city" : "Salamanca",
    "name" : "USAL",
    "location" : {
        "type" : "Point",
        "coordinates" : [
            -5.6722512,
            40.9607792
        ]
    },
    "students" : {
        "year" : 2016,
        "number" : 21913
    }
}
{
    "_id" : ObjectId("65198819627e32f113e6e7d7"),
    "country" : "Spain",
    "city" : "Salamanca",
    "name" : "USAL",
    "location" : {
        "type" : "Point",
        "coordinates" : [
            -5.6722512,
            40.9607792
        ]
    },
    "students" : {
        "year" : 2017,
        "number" : 21715
    }
}
```

9. Aggregate to match documents with 'name' equal to 'USAL', unwind the 'students' array, project the 'year' and 'number' fields excluding '_id', and pretty print the result:

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'},{$project:{_id:0,'students.year':1,'students.number':1}}]).pretty();
```

```
{ "students" : { "number" : 24774 } }
{ "students" : { "number" : 23166 } }
{ "students" : { "number" : 21913 } }
{ "students" : { "number" : 21715 } }
```

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'},{$project:{_id:0,'students.year':1,'students.number':1}},{$sort:{'students.number':-1}}]).pretty();
```

```
{ "students" : { "number" : 24774 } }
{ "students" : { "number" : 23166 } }
{ "students" : { "number" : 21913 } }
{ "students" : { "number" : 21715 } }
```

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'},{$project:{_id:0,'students.year':1,'students.number':1}},{$sort:{'students.number':1}}]).pretty();
```

```
{ "students" : { "number" : 21715 } }
{ "students" : { "number" : 21913 } }
{ "students" : { "number" : 23166 } }
{ "students" : { "number" : 24774 } }
```

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'},{$project:{_id:0,'students.year':1,'students.number':1}},{$sort:{'students.number':1}},{$limit:1}]).pretty();
```

```
{ "students" : { "number" : 21715 } }
```

```
>db.uni.aggregate([{$match:{name:'USAL'}},{$unwind:'$students'},{$project:{_id:0,'students.year':1,'students.number':1}},{$sort:{'students.number':1}},{$skip:2},{$limit:1}]).pretty();
```

```
{ "students" : { "number" : 23166 } }
```

```
>db.uni.aggregate([{$unwind:'$students'},{$count:'totalDocs'}]);
```

```
{ "totalDocs" : 8 }
```

```
>db.uni.aggregate([{$project:{_id:0,name:1,marks:1}},{$group:{_id:'$name',totalDocs:{$sum:'$marks'}}}]);
```

```
{ "_id" : "UPSA", "totalDocs" : 0 }
{ "_id" : "USAL", "totalDocs" : 0 }
```

```
>db.uni.aggregate([{$project:{_id:0,name:1,marks:1}},{$group:{_id:'$name',totalDocs:{$sum:1}}}]);
```

```
{ "_id" : "UPSA", "totalDocs" : 1 }
{ "_id" : "USAL", "totalDocs" : 1 }
```

```
>db.uni.aggregate([{$project:{_id:0,name:1,marks:1}},{$group:{_id:'$name',totalDocs:{$min:'$marks'}}}]);
```

```
{ "_id" : "UPSA", "totalDocs" : null }
{ "_id" : "USAL", "totalDocs" : null }
```

```
>db.uni.aggregate([{$project:{_id:0,name:1,marks:1}},{$group:{_id:'$name',totalDocs:{$max:'$marks'}}}]);
```

```
{ "_id" : "UPSA", "totalDocs" : null }
{ "_id" : "USAL", "totalDocs" : null }
```

Practical 3	
Aim: To demonstrate Aggregation Pipeline in MongoDB.	
Name: Saail Chavan	Roll No:
Performance date: 02 – 09 – 2023	Sign:

Create a mongodb schema with name inventory and contains the following attributes.

1. Item name
2. Quantity
3. Size (height, width and unit of measure)
4. Quality (having grade from A – E)
5. Instock (warehouse [Eg: w1, w2], quantity)

Insert queries:

```
>db.inventory.insert({iname:"Airbrush", quant:12, size:{height:12,width:12,unit:"cm"},
qual:"A",instock:[{whouse:"W1",quantity:8},{whouse:"W4",quantity:4}]});
```

```
>db.inventory.insert({iname:"Ball", quant:30, size:{height:2,width:2,unit:"cm"},
qual:"D",instock:[{whouse:"W4",quantity:20},{whouse:"W2",quantity:10}]});
```

```
>db.inventory.insertMany([{iname:"Chairs", quant:6, size:{height:0.5,width:0.5,unit:"m"},
qual:"C",instock:[{whouse:"W3",quantity:3},{whouse:"W7",quantity:2},{whouse:
"W10",quantity:1}]} , {iname:"journal", quant:100, size:{height:24,width:16,unit:"cm"},
qual:"B",instock:[{whouse:"W1",quantity:10},{whouse:"W2",quantity:60},{whouse:
"W5",quantity:10},{whouse: "W10",quantity:20}]} , {iname:"Erasers ", quant:500,
size:{height:40,width:20,unit:"mm"},qual:"D",instock:[{whouse:"W4",quantity:100},{whouse:"W3"
,quantity:100},{whouse: "W5",quantity:300}]}]);
```

```
>db.inventory.insertMany([{iname:"Duster", quant:40, size:{height:0.2,width:0.1,unit:"m"},
qual:"E",instock:[{whouse:"W1",quantity:12},{whouse:"W2",quantity:12},{whouse:
"W10",quantity:26}]} , {iname:"Papers", quant:1000, size:{height:24,width:16,unit:"cm"},
qual:"C",instock:[{whouse:"W4",quantity:500},{whouse:"W6",quantity:500}]} , {iname:"Mouse",
quant:30, size:{height:10,width:5,unit:"cm"},
qual:"A",instock:[{whouse:"W10",quantity:10},{whouse:"W9",quantity:10},{whouse:
"W8",quantity:10}]}]);
```

```
>db.inventory.insertMany([{iname:"Keyboard", quant:42, size:{height:0.2,width:0.5,unit:"m"},
qual:"A",instock:[{whouse:"W8",quantity:12},{whouse:"W6",quantity:13},{whouse:
"W3",quantity:17}]} , {iname:"Stand", quant:350, size:{height:24,width:30,unit:"cm"},
qual:"B",instock:[{whouse:"W4",quantity:250},{whouse:"W6",quantity:100}]}]);
```

Questions:

1. **Find all items having quantity less than or equal to 10**

```
> db.inventory.find({quant:{$lte:10}});
```

```
[{
   _id: ObjectId("639ad50a60dc5989073cca6a"),iname: 'Chairs',quant: 6,
   size: { height: 0.5, width: 0.5, unit: 'm' },qual: 'C',
   instock: [{ whouse: 'W3', quantity: 3 },{ whouse: 'W7', quantity: 2 },{ whouse: 'W10', quantity:
1 }]}]
```

2. **Find all items having quality as A and display name and quality only**

```
> db.inventory.find({qual:"A"},{iname:1,qual:1});
```

```
[{_id: ObjectId("639ad32160dc5989073cca68"), iname: 'Airbrush',qual: 'A'},
  {_id: ObjectId("639ad5d660dc5989073cca6f"),iname: 'Mouse',qual: 'A'},
  {_id: ObjectId("639ad66e60dc5989073cca70"),iname: 'Keyboard',qual: 'A'}] OR
```

```
> db.inventory.find({qual:"A"},{iname:1,qual:1,_id:0});
```

```
[{ iname: 'Airbrush', qual: 'A' },{ iname: 'Mouse', qual: 'A' },{ iname: 'Keyboard', qual: 'A' }]
```

---

3. **Having instock warehouse as 1 and display names, instock warehouse and instock quantity**

```
> db.inventory.find({"instock.whouse":{$eq:"W1"}},{iname:1,_id:0, "instock.whouse":1});
```

```
[{ iname: 'Airbrush',instock: [ { whouse: 'W1' }, { whouse: 'W4' } ]},
{iname: 'journal',instock: [{ whouse: 'W1' },{ whouse: 'W2' },{ whouse: 'W5' },{ whouse: 'W10' }
]},{iname: 'Duster',instock: [ { whouse: 'W1' }, { whouse: 'W2' }, { whouse: 'W10' } ] }]
```

---

4. **Having quality as B and E and display their names and quality**

```
> db.inventory.find({$or:[{qual:"B"}, {qual:"E"}]},{iname:1,_id:0, qual:1});
```

```
[{ iname: 'journal', qual: 'B' },{ iname: 'Duster', qual: 'E' },{ iname: 'Stand', qual: 'B' }]
```

---

5. **Having instock quantity between 20 and 40 and display names, quantity and instock data**

```
> db.inventory.find({$and:[{"instock.quantity":{$lte:40}},
{"instock.quantity":{$gte:20}}]},{iname:1,_id:0, qual:1, "instock":1});
```

```
[{iname: 'Ball', qual: 'D', instock: [ { whouse: 'W4', quantity: 20 }, { whouse: 'W2', quantity: 10 } ] }, {
iname: 'journal', qual: 'B', instock: [ { whouse: 'W1', quantity: 10 }, { whouse: 'W2', quantity: 60 }, {
whouse: 'W5', quantity: 10 }, { whouse: 'W10', quantity: 20 } ] }, { iname: 'Duster', qual: 'E', instock: [ {
whouse: 'W1', quantity: 12 }, { whouse: 'W2', quantity: 12 }, { whouse: 'W10', quantity: 26 } ] }
]
```

---

6. **Find all items starting with S.**

```
> db.inventory.find({iname:{$regex:'S'}}, {iname:1, quant:1, _id:0});
```

```
[ { iname: 'Stand', quant: 350 } ]
```

---

7. **Find all inventory item names having instock warehouse as w2 and w5 and w7, show name, warehouse number, except id**

```
> db.inventory.find({"instock.whouse":{$in:["W2", "W5", "W7"]}},{iname:1, _id:0, "instock.whouse":1});
```

```
[
  { iname: 'Ball', instock: [ { whouse: 'W4' }, { whouse: 'W2' } ] },
  {
    iname: 'Chairs',
    instock: [ { whouse: 'W3' }, { whouse: 'W7' }, { whouse: 'W10' } ]
  },
  {
    iname: 'journal',
    instock: [
      { whouse: 'W1' },
      { whouse: 'W2' },
      { whouse: 'W5' },
      { whouse: 'W10' }
    ]
  },
  {
    iname: 'Erasers ',
    instock: [ { whouse: 'W4' }, { whouse: 'W3' }, { whouse: 'W5' } ]
  },
  {
    iname: 'Duster',
    instock: [ { whouse: 'W1' }, { whouse: 'W2' }, { whouse: 'W10' } ]
  }
]
```

---

8. **Find all inventory item names having instock warehouse not as w2 and w5 and w7, show name, warehouse number, except id**

```
> db.inventory.find({"instock.whouse":{$nin:["W2", "W5", "W7"]}},{iname:1, _id:0, "instock.whouse":1});
```

```
[
  {iname: 'Airbrush',instock: [ { whouse: 'W1' }, { whouse: 'W4' } ]},
  { iname: 'Papers', instock: [ { whouse: 'W4' }, { whouse: 'W6' } ] },
  {iname: 'Mouse',instock: [ { whouse: 'W10' }, { whouse: 'W9' }, { whouse: 'W8' } ]},
  {iname: 'Keyboard',instock: [ { whouse: 'W8' }, { whouse: 'W6' }, { whouse: 'W3' } ]},
  { iname: 'Stand', instock: [ { whouse: 'W4' }, { whouse: 'W6' } ] }
]
```

---

9. Item names not having instock quantity greater than equal to 50

```
>db.inventory.find({"instock.quantity":{$not:{$gte:50}}},{iname:1,_id:0, qual:1, "instock":1});
```

```
[
  { iname: 'Airbrush', qual: 'A', instock: [ { whouse: 'W1', quantity: 8 }, { whouse: 'W4', quantity: 4 } ] },
{ iname: 'Ball', qual: 'D', instock: [ { whouse: 'W4', quantity: 20 }, { whouse: 'W2', quantity: 10 } ] },
{ iname: 'Chairs', qual: 'C', instock: [ { whouse: 'W3', quantity: 3 }, { whouse: 'W7', quantity: 2 }, {
whouse: 'W10', quantity: 1 } ] },
{ iname: 'Duster', qual: 'E', instock: [ { whouse: 'W1', quantity: 12 }, { whouse: 'W2', quantity: 12 }, {
whouse: 'W10', quantity: 26 } ] },
{ iname: 'Mouse', qual: 'A', instock: [ { whouse: 'W10', quantity: 10 }, { whouse: 'W9', quantity: 10 }, {
whouse: 'W8', quantity: 10 } ] },
{ iname: 'Keyboard', qual: 'A', instock: [ { whouse: 'W8', quantity: 12 }, { whouse: 'W6', quantity: 13 }, {
whouse: 'W3', quantity: 17 } ] }
]
```

---

10. Quality as E and quantity greater than equal to 50

```
> db.inventory.find({$and:[{qual:"E"}, {quant:{$gte:20}}]},{iname:1,_id:0, qual:1, quant:1});
```

```
[ { iname: 'Duster', quant: 40, qual: 'E' } ]
```

---

11. Count of inventory having unit of measure as metre

```
> db.inventory.find({"size.unit":"m"}).count();
```

```
3
```

---

12. Display first 3 documents

```
> db.inventory.find().limit(3);
```

```
[
  { _id: ObjectId("639ad32160dc5989073cca68"), iname: 'Airbrush', quant: 12, size: { height: 12, width:
12, unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W1', quantity: 8 }, { whouse: 'W4', quantity: 4 } ] },
{ _id: ObjectId("639ad50260dc5989073cca69"), iname: 'Ball', quant: 30, size: { height: 2, width: 2, unit:
'cm' }, qual: 'D', instock: [ { whouse: 'W4', quantity: 20 }, { whouse: 'W2', quantity: 10 } ] },
{ _id: ObjectId("639ad50a60dc5989073cca6a"), iname: 'Chairs', quant: 6, size: { height: 0.5, width: 0.5,
unit: 'm' }, qual: 'C', instock: [ { whouse: 'W3', quantity: 3 }, { whouse: 'W7', quantity: 2 }, { whouse:
'W10', quantity: 1 } ] }]
```

Aggregation functions: match-> group-> project-> sort-> limit

1. **Display the total quantity of all items.**

   ```
   > db.inventory.aggregate([{$group:{_id:null, "Sum of quantity":{$sum:"$quant"}}}]);
   ```

   ```
   [ { _id: null, 'Sum of quantity': 2110 } ]
   ```

---

2. **Display the average quantity of items present in each document within the collection.**

   ```
   > db.inventory.aggregate([{$group:{_id:null, "Average of quantity":{$avg:"$quant"}}}]);
   ```

   ```
   [ { _id: null, 'Average of quantity': 211 } ]
   ```

---

3. **Display which item has the lowest quantity within the collection.**

   ```
   > db.inventory.aggregate([{$group:{_id:null, "Minimum quantity":{$min:"$quant"}}}]);
   ```

   ```
   [ { _id: null, 'Minimum quantity': 6 } ]
   ```

---

4. **Display which item has the highest quantity within the collection.**

   ```
   > db.inventory.aggregate([{$group:{_id:null, "Maximum quantity":{$max:"$quant"}}}]);
   ```

   ```
   [ { _id: null, 'Minimum quantity': 6 } ]
   ```

---

5. **Display all the items having unit in cm using $match stage.**

   ```
   > db.inventory.aggregate([{$match:{"size.unit":"cm"}}, {$project:{iname:1, quant:1}}]);
   ```

   ```
   { "_id" : ObjectId("63a01c2d8065354d1932e7d1"), "iname" : "Stapler", "quant" : 100 }
   { "_id" : ObjectId("63a01c2d8065354d1932e7d2"), "iname" : "Paper", "quant" : 100 }
   { "_id" : ObjectId("63a01c2d8065354d1932e7d3"), "iname" : "Eraser", "quant" : 30 }
   { "_id" : ObjectId("63a01c2d8065354d1932e7d4"), "iname" : "Calculator", "quant" : 5 }
   ```

Match items having quantity greater than equal to 40 quantity

> db.inventory.aggregate([{$match: {quant: {$gte:40}}}]);

```
[{ _id: ObjectId("639ad50a60dc5989073cca6b"),
    iname: 'journal',
    quant: 100,
    size: { height: 24, width: 16, unit: 'cm' },
    qual: 'B',
    instock: [
      { whouse: 'W1', quantity: 10 },
      { whouse: 'W2', quantity: 60 },
      { whouse: 'W5', quantity: 10 },
      { whouse: 'W10', quantity: 20 }]},
  {    _id: ObjectId("639ad50a60dc5989073cca6c"),
    iname: 'Erasers ',
    quant: 500,
    size: { height: 40, width: 20, unit: 'mm' },
    qual: 'D',
    instock: [
      { whouse: 'W4', quantity: 100 },
      { whouse: 'W3', quantity: 100 },
      { whouse: 'W5', quantity: 300 }
    ]},
  {    _id: ObjectId("639ad5d660dc5989073cca6e"),
    iname: 'Papers',
    quant: 1000,
    size: { height: 24, width: 16, unit: 'cm' },
    qual: 'C',
    instock: [
      { whouse: 'W4', quantity: 500 },
      { whouse: 'W6', quantity: 500 }
    ]},
  { _id: ObjectId("639ad66e60dc5989073cca70"),
    iname: 'Keyboard',
    quant: 42,
    size: { height: 0.2, width: 0.5, unit: 'm' },
    qual: 'A',
    instock: [
      { whouse: 'W8', quantity: 12 },
      { whouse: 'W6', quantity: 13 },
      { whouse: 'W3', quantity: 17 }
    ]  },
  {    _id: ObjectId("639ad66e60dc5989073cca71"),
    iname: 'Stand',
    quant: 350,
    size: { height: 24, width: 30, unit: 'cm' },
    qual: 'B',
    instock: [
      { whouse: 'W4', quantity: 250 },
      { whouse: 'W6', quantity: 100 }]}]
```

Items having Unit of Measure as metre and display item name and size

```
> db.inventory.aggregate([{$match: {"size.unit":"m"}}, {$project: {iname:1,_id:0, "size.unit":1}}]);
```

```
[  { iname: 'Chairs', size: { unit: 'm' } },
  { iname: 'Duster', size: { unit: 'm' } },
  { iname: 'Keyboard', size: { unit: 'm' } }]
```

Display all inventory items using $match in aggregation

```
> db.inventory.aggregate([{$match: {}}]);
```

```
[  { _id: ObjectId("639ad32160dc5989073cca68"), iname: 'Airbrush', quant: 12, size: { height: 12, width: 12,
unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W1', quantity: 8 }, { whouse: 'W4', quantity: 4 } ] },
{ _id: ObjectId("639ad50260dc5989073cca69"), iname: 'Ball', quant: 30, size: { height: 2, width: 2, unit:
'cm' }, qual: 'D', instock: [ { whouse: 'W4', quantity: 20 }, { whouse: 'W2', quantity: 10 } ] },
{ _id: ObjectId("639ad50a60dc5989073cca6a"), iname: 'Chairs', quant: 6, size: { height: 0.5, width: 0.5,
unit: 'm' }, qual: 'C', instock: [ { whouse: 'W3', quantity: 3 }, { whouse: 'W7', quantity: 2 }, { whouse: 'W10',
quantity: 1 } ] }, { _id: ObjectId("639ad50a60dc5989073cca6b"), iname: 'journal', quant: 100, size: { height:
24, width: 16, unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W1', quantity: 10 }, { whouse: 'W2', quantity: 60 },
{ whouse: 'W5', quantity: 10 }, { whouse: 'W10', quantity: 20 } ] },
{ _id: ObjectId("639ad5d660dc5989073cca6d"), iname: 'Duster', quant: 40, size: { height: 0.2, width: 0.1,
unit: 'm' }, qual: 'E', instock: [ { whouse: 'W1', quantity: 12 }, { whouse: 'W2', quantity: 12 }, { whouse:
'W10', quantity: 26 } ] }, { _id: ObjectId("639ad5d660dc5989073cca6e"), iname: 'Papers', quant: 1000, size:
{ height: 24, width: 16, unit: 'cm' }, qual: 'C', instock: [ { whouse: 'W4', quantity: 500 }, { whouse: 'W6',
quantity: 500 } ] }, { _id: ObjectId("639ad5d660dc5989073cca6f"), iname: 'Mouse', quant: 30, size: {
height: 10, width: 5, unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W10', quantity: 10 }, { whouse: 'W9',
quantity: 10 }, { whouse: 'W8', quantity: 10 } ] }, { _id: ObjectId("639ad66e60dc5989073cca70"), iname:
'Keyboard', quant: 42, size: { height: 0.2, width: 0.5, unit: 'm' }, qual: 'A', instock: [ { whouse: 'W8',
quantity: 12 }, { whouse: 'W6', quantity: 13 }, { whouse: 'W3', quantity: 17 } ] }, { _id:
ObjectId("639ad66e60dc5989073cca71"), iname: 'Stand', quant: 350, size: { height: 24, width: 30, unit: 'cm'
}, qual: 'B', instock: [ { whouse: 'W4', quantity: 250 }, { whouse: 'W6', quantity: 100 } ] } ]
```

Item having names: Chairs, journal, duster and display its: Name quantity unit

```
> db.inventory.aggregate([{$match: {iname: {$in: ["Chairs","journal","Duster"]}}}, {$project:
{iname:1,_id:0, quant:1, "size.unit":1}}]);
```

```
[{ iname: 'Chairs', quant: 6, size: { unit: 'm' } },{ iname: 'journal', quant: 100, size: { unit: 'cm' } },{ iname:
'Duster', quant: 40, size: { unit: 'm' } }]
```

Item names not having: Chairs, journal, duster and display its: Name quantity unit

```
> db.inventory.aggregate([{$match: {iname: {$nin: ["Chairs","journal","Duster"]}}}, {$project:
{iname:1,_id:0, quant:1, "size.unit":1}}]);
```

```
[  { iname: 'Airbrush', quant: 12, size: { unit: 'cm' } },
  { iname: 'Ball', quant: 30, size: { unit: 'cm' } },
  { iname: 'Erasers ', quant: 500, size: { unit: 'mm' } },
  { iname: 'Papers', quant: 1000, size: { unit: 'cm' } },
  { iname: 'Stand', quant: 350, size: { unit: 'cm' } }]
```

Display all items having instock quantity between 50 and 100, display item name, unit of measure and instock details.

```
> db.inventory.aggregate([{$match: {$and:[{"instock.quantity":{$lte:100}},
{"instock.quantity":{$gte:50}}]}}, {$project: {iname:1,_id:0,"size.unit":1,"instock":1}}]);
```

```
[{ iname: 'journal', size: { unit: 'cm' }, instock: [ { whouse: 'W1', quantity: 10 }, { whouse: 'W2', quantity: 60
}, { whouse: 'W5', quantity: 10 }, { whouse: 'W10', quantity: 20 } ] },
{ iname: 'Erasers ', size: { unit: 'mm' }, instock: [ { whouse: 'W4', quantity: 100 }, { whouse: 'W3', quantity:
100 }, { whouse: 'W5', quantity: 300 } ] },
{ iname: 'Stand', size: { unit: 'cm' }, instock: [ { whouse: 'W4', quantity: 250 }, { whouse: 'W6', quantity: 100
} ] }]
```

Display all items sorted on item names

```
> db.inventory.aggregate([{$match: {}}, {$sort: {iname:1}}]);
```

```
[  { _id: ObjectId("639ad32160dc5989073cca68"), iname: 'Airbrush', quant: 12, size: { height: 12, width: 12,
unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W1', quantity: 8 }, { whouse: 'W4', quantity: 4 } ] }, { _id:
ObjectId("639ad50260dc5989073cca69"), iname: 'Ball', quant: 30, size: { height: 2, width: 2, unit: 'cm' },
qual: 'D', instock: [ { whouse: 'W4', quantity: 20 }, { whouse: 'W2', quantity: 10 } ] }, { _id:
ObjectId("639ad50a60dc5989073cca6a"), iname: 'Chairs', quant: 6, size: { height: 0.5, width: 0.5, unit: 'm'
}, qual: 'C', instock: [ { whouse: 'W3', quantity: 3 }, { whouse: 'W7', quantity: 2 }, { whouse: 'W10',
quantity: 1 } ] }, { _id: ObjectId("639ad5d660dc5989073cca6d"), iname: 'Duster', quant: 40, size: { height:
0.2, width: 0.1, unit: 'm' }, qual: 'E', instock: [ { whouse: 'W1', quantity: 12 }, { whouse: 'W2', quantity: 12 },
{ whouse: 'W10', quantity: 26 } ] }, { _id: ObjectId("639ad50a60dc5989073cca6c"), iname: 'Erasers ', quant:
500, size: { height: 40, width: 20, unit: 'mm' }, qual: 'D', instock: [ { whouse: 'W4', quantity: 100 }, {
whouse: 'W3', quantity: 100 }, { whouse: 'W5', quantity: 300 } ] }, { _id:
ObjectId("639ad66e60dc5989073cca70"), iname: 'Keyboard', quant: 42, size: { height: 0.2, width: 0.5, unit:
'm' }, qual: 'A', instock: [ { whouse: 'W8', quantity: 12 }, { whouse: 'W6', quantity: 13 }, { whouse: 'W3',
quantity: 17 } ] }, { _id: ObjectId("639ad5d660dc5989073cca6f"), iname: 'Mouse', quant: 30, size: { height:
10, width: 5, unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W10', quantity: 10 }, iname: 'Papers', quant: 1000,
size: { height: 24, width: 16, unit: 'cm' }, qual: 'C', instock: [ { whouse: 'W4', quantity: 500 }, { whouse:
'W6', quantity: 500 } ] }, { _id: ObjectId("639ad66e60dc5989073cca71"), iname: 'Stand', quant: 350, size: {
height: 24, width: 30, unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W4', quantity: 250 }, { whouse: 'W6',
quantity: 100 } ] }, { _id: ObjectId("639ad50a60dc5989073cca6b"), iname: 'journal', quant: 100, size: {
height: 24, width: 16, unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W1', quantity: 10 }, { whouse: 'W2',
quantity: 60 }, { whouse: 'W5', quantity: 10 }, { whouse: 'W10', quantity: 20 } ] }]
```

Display all items having quantity greater than equal to 70 sorted by their quantity in descending order and display their iname and quantity

```
> db.inventory.aggregate([{$match: {quant:{$gte:70}}}, {$sort: {quant:-1}}, {$project:{iname:1,
quant:1, _id:0}}]);
```

```
[  { iname: 'Papers', quant: 1000 },
  { iname: 'Erasers ', quant: 500 },
  { iname: 'Stand', quant: 350 },
  { iname: 'journal', quant: 100 }]
```

Adding additional records to perform group aggregation on:

```
>db.inventory.insertMany([{ iname: 'Airbrush', quant: 55, size: { height: 52, width: 26, unit: 'm' },
qual: 'D', instock: [ { whouse: 'W2', quantity: 65 }, { whouse: 'W4', quantity: 8 } ] }, { iname: 'Ball',
quant: 39, size: { height: 7, width: 5, unit: 'cm' }, qual: 'D', instock: [ { whouse: 'W4', quantity: 82 }, {
whouse: 'W2', quantity: 12 } ] }, { iname: 'Chairs', quant: 47, size: { height: 48, width: 20, unit: 'm' },
qual: 'C', instock: [ { whouse: 'W3', quantity: 25 }, { whouse: 'W7', quantity: 3 }, { whouse: 'W10',
quantity: 8 } ] }, { iname: 'journal', quant: 120, size: { height: 24, width: 16, unit: 'cm' }, qual: 'B',
instock: [ { whouse: 'W1', quantity: 85 }, { whouse: 'W2', quantity: 20 }, { whouse: 'W5', quantity: 18
}, { whouse: 'W10', quantity: 23 } ] }, { iname: 'Erasers ', quant: 58, size: { height: 45, width: 70, unit:
'mm' }, qual: 'D', instock: [ { whouse: 'W4', quantity: 550 }, { whouse: 'W3', quantity: 140 }, { whouse:
'W5', quantity: 340 } ] }, { iname: 'Duster', quant: 40, size: { height: 0.2, width: 0.1, unit: 'm' }, qual:
'E', instock: [ { whouse: 'W1', quantity: 12 }, { whouse: 'W2', quantity: 12 }, { whouse: 'W10',
quantity: 26 } ] }, { iname: 'Papers', quant: 1250, size: { height: 24, width: 16, unit: 'cm' }, qual: 'C',
instock: [ { whouse: 'W4', quantity: 700 }, { whouse: 'W6', quantity: 570 } ] }, { iname: 'Mouse', quant:
30, size: { height: 10, width: 5, unit: 'cm' }, qual: 'A', instock: [ { whouse: 'W10', quantity: 10 }, {
whouse: 'W9', quantity: 10 }, { whouse: 'W8', quantity: 10 } ] }, { iname: 'Keyboard', quant: 42, size: {
height: 0.2, width: 0.5, unit: 'm' }, qual: 'A', instock: [ { whouse: 'W8', quantity: 12 }, { whouse: 'W6',
quantity: 13 }, { whouse: 'W3', quantity: 17 } ] }, { iname: 'Stand', quant: 350, size: { height: 24,
width: 30, unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W4', quantity: 250 }, { whouse: 'W6', quantity:
100 } ] } ]);
```

Display all items with name journals

```
> db.inventory.aggregate([{$match: {iname:"journal"}}]);
```

```
[{ _id: ObjectId("639ad50a60dc5989073cca6b"), iname: 'journal', quant: 100, size: { height: 24, width: 16,
unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W1', quantity: 10 }, { whouse: 'W2', quantity: 60 }, { whouse:
'W5', quantity: 10 }, { whouse: 'W10', quantity: 20 } ] }, { _id: ObjectId("63a018397235907b7ffe3859"),
iname: 'journal', quant: 120, size: { height: 24, width: 16, unit: 'cm' }, qual: 'B', instock: [ { whouse: 'W1',
quantity: 85 }, { whouse: 'W2', quantity: 20 }, { whouse: 'W5', quantity: 18 }, { whouse: 'W10', quantity: 23
}] }]
```

Display all inventory item names grouped by their name

```
> db.inventory.aggregate([{$group:{_id:"$iname"}}]);
```

```
[  { _id: 'Ball' }, { _id: 'Duster' },  { _id: 'Chairs' },  { _id: 'Airbrush' },  { _id: 'Erasers ' },  { _id: 'Mouse' },
{ _id: 'Papers' },  { _id: 'journal' },  { _id: 'Keyboard' },  { _id: 'Stand' }]
```

Count the number of items in a particular group, by item name

```
> db.inventory.aggregate([{$group:{_id: "$iname", "Total number of items are: ": {$sum:1}}}]);
```

```
[
  { _id: 'Chairs', 'Total number of items are: ': 2 },
  { _id: 'Papers', 'Total number of items are: ': 2 },
  { _id: 'journal', 'Total number of items are: ': 2 },
  { _id: 'Airbrush', 'Total number of items are: ': 2 },
  { _id: 'Keyboard', 'Total number of items are: ': 2 }]
```

Count the number of items in a particular group, by item name and sort them in ascending order

```
> db.inventory.aggregate([{$group:{_id: "$iname", count: {$sum:1}}}, {$sort: {_id:1}}]);
```

```
[
  { _id: 'Airbrush', count: 2 },
  { _id: 'Ball', count: 2 },
  { _id: 'Chairs', count: 2 },
  { _id: 'Duster', count: 2 },
  { _id: 'Erasers ', count: 2 },
  { _id: 'Keyboard', count: 2 },
  { _id: 'Mouse', count: 2 },
  { _id: 'Papers', count: 2 },
  { _id: 'Stand', count: 2 },
  { _id: 'journal', count: 2 }
]
```

Count the number of items in a particular group, by item name and sort them in descending order.

```
> db.inventory.aggregate([{$group:{_id: "$iname", count: {$sum:1}}}, {$sort: {_id:-1}}]);
```

```
[
  { _id: 'journal', count: 2 },
  { _id: 'Stand', count: 2 },
  { _id: 'Papers', count: 2 },
  { _id: 'Mouse', count: 2 },
  { _id: 'Keyboard', count: 2 },
  { _id: 'Erasers ', count: 2 },
  { _id: 'Duster', count: 2 },
  { _id: 'Chairs', count: 2 },
  { _id: 'Ball', count: 2 },
  { _id: 'Airbrush', count: 2 }
]
```

Display total quantities of all items grouped by item name and sort ascendingly

```
> db.inventory.aggregate([{$group:{_id: "$iname", count: {$sum:"$quant"}}}, {$sort: {_id:1}}]);
```

```
[
  { _id: 'Airbrush', count: 67 },
  { _id: 'Ball', count: 69 },
  { _id: 'Chairs', count: 53 },
  { _id: 'Duster', count: 80 },
  { _id: 'Erasers ', count: 558 },
  { _id: 'Keyboard', count: 84 },
  { _id: 'Mouse', count: 60 },
  { _id: 'Papers', count: 2250 },
  { _id: 'Stand', count: 700 },
  { _id: 'journal', count: 220 }
]
```

Display total quantities of all items grouped by item name and sort ascendingly having count gte 80

```
> db.inventory.aggregate([{$group:{_id: "$iname", count: {$sum:"$quant"}}}, {$sort: {_id:1}},
{$match: {count:{$gte:80}}}]);
```

```
[
  { _id: 'Duster', count: 80 },
  { _id: 'Erasers ', count: 558 },
  { _id: 'Keyboard', count: 84 },
  { _id: 'Papers', count: 2250 },
  { _id: 'Stand', count: 700 },
  { _id: 'journal', count: 220 }
]
```

Display all names with their quantity, using project first and then match gte 50

```
> db.inventory.aggregate([{$project: {_id:0, iname:1, quant:1}}, {$match: {quant:{$gte:50}}}])
```

```
[
  { iname: 'journal', quant: 100 },
  { iname: 'Erasers ', quant: 500 },
  { iname: 'Papers', quant: 1000 },
  { iname: 'Stand', quant: 350 },
  { iname: 'Airbrush', quant: 55 },
  { iname: 'journal', quant: 120 },
  { iname: 'Erasers ', quant: 58 },
  { iname: 'Papers', quant: 1250 },
  { iname: 'Stand', quant: 350 }
]
```

Display all names with their quantity, using project first and then match gte 50, group and count the number of items by item name sorted on their total count

```
> db.inventory.aggregate([{$project: {_id:0, iname:1, quant:1}}, {$match: {quant:{$gte:50}}},
{$group:{_id: "$iname", count: {$sum:1}}}, {$sort:{count:1}}]);
```

```
[
  { _id: 'Airbrush', count: 1 },
  { _id: 'journal', count: 2 },
  { _id: 'Erasers ', count: 2 },
  { _id: 'Papers', count: 2 },
  { _id: 'Stand', count: 2 }
]
```

Display all names with their quantity, using project first and then match gte 50, group and calculate the number of items by item name sorted on their total quantity

```
> db.inventory.aggregate([{$project: {_id:0, iname:1, quant:1}}, {$match: {quant:{$gte:50}}},
{$group:{_id: “$iname”, count: {$sum:”$quant”}}}, {$sort:{count:1}}]);
```

```
[  { _id: 'Airbrush', count: 55 },
  { _id: 'journal', count: 220 },
  { _id: 'Erasers ', count: 558 },
  { _id: 'Stand', count: 700 },
  { _id: 'Papers', count: 2250 }]
```

Group by name as well as quantity

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname",
Quantity:"$quant"},totalcount:{$sum:1}}}]);
```

```
[
  { _id: { Name: 'Chairs', Quantity: 6 }, totalcount: 1 },
  { _id: { Name: 'Stand', Quantity: 350 }, totalcount: 3 },
  { _id: { Name: 'journal', Quantity: 120 }, totalcount: 2 },
  { _id: { Name: 'Erasers ', Quantity: 500 }, totalcount: 1 },
  { _id: { Name: 'Duster', Quantity: 40 }, totalcount: 3 },
  { _id: { Name: 'Airbrush', Quantity: 55 }, totalcount: 2 },
  { _id: { Name: 'journal', Quantity: 100 }, totalcount: 1 },
  { _id: { Name: 'Ball', Quantity: 30 }, totalcount: 1 },
  { _id: { Name: 'Papers', Quantity: 1250 }, totalcount: 2 },
  { _id: { Name: 'Erasers ', Quantity: 58 }, totalcount: 2 },
  { _id: { Name: 'Papers', Quantity: 1000 }, totalcount: 1 },
  { _id: { Name: 'Mouse', Quantity: 30 }, totalcount: 3 },
  { _id: { Name: 'Keyboard', Quantity: 42 }, totalcount: 3 },
  { _id: { Name: 'Airbrush', Quantity: 12 }, totalcount: 1 },
  { _id: { Name: 'Ball', Quantity: 39 }, totalcount: 2 },
  { _id: { Name: 'Chairs', Quantity: 47 }, totalcount: 2 }
]
```

Calculate total quantity of items grouped by item name and quantity

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname", Quantity:"$quant"},
count:{$sum:1},totalquantity:{$sum:"$quant"}}}]);
```

```
[
  { _id: { Name: 'Chairs', Quantity: 6 }, count: 1, totalquantity: 6 },
  { _id: { Name: 'Stand', Quantity: 350 },count: 3,totalquantity: 1050},
  {_id: { Name: 'journal', Quantity: 120 },count: 2,totalquantity: 240 },
  {_id: { Name: 'Erasers ', Quantity: 500 },count: 1,totalquantity: 500},
  {_id: { Name: 'Duster', Quantity: 40 },count: 3,totalquantity: 120},
  {_id: { Name: 'Airbrush', Quantity: 55 },count: 2,totalquantity: 110},
  {_id: { Name: 'journal', Quantity: 100 },count: 1,totalquantity: 100},
  { _id: { Name: 'Ball', Quantity: 30 }, count: 1, totalquantity: 30 },
  {_id: { Name: 'Papers', Quantity: 1250 }, count: 2,totalquantity: 2500},
  {_id: { Name: 'Erasers ', Quantity: 58 },count: 2,totalquantity: 116}]
```

Calculate total quantity of items grouped by item name and quantity and sort by iname

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname",
Quantity:"$quant"},totalquantity:{$sum:"$quant"}}}, {$sort:{"_id.Name":1}}]);
```

```
[
  { _id: { Name: 'Airbrush', Quantity: 12 }, totalquantity: 12 },
  { _id: { Name: 'Airbrush', Quantity: 55 }, totalquantity: 110 },
  { _id: { Name: 'Ball', Quantity: 39 }, totalquantity: 78 },
  { _id: { Name: 'Ball', Quantity: 30 }, totalquantity: 30 },
  { _id: { Name: 'Chairs', Quantity: 6 }, totalquantity: 6 },
  { _id: { Name: 'Chairs', Quantity: 47 }, totalquantity: 94 },
  { _id: { Name: 'Duster', Quantity: 40 }, totalquantity: 120 },
  { _id: { Name: 'Erasers ', Quantity: 58 }, totalquantity: 116 },
  { _id: { Name: 'Erasers ', Quantity: 500 }, totalquantity: 500 },
  { _id: { Name: 'Keyboard', Quantity: 42 }, totalquantity: 126 },
  { _id: { Name: 'Mouse', Quantity: 30 }, totalquantity: 90 },
  { _id: { Name: 'Papers', Quantity: 1000 }, totalquantity: 1000 },
  { _id: { Name: 'Papers', Quantity: 1250 }, totalquantity: 2500 },
  { _id: { Name: 'Stand', Quantity: 350 }, totalquantity: 1050 },
  { _id: { Name: 'journal', Quantity: 120 }, totalquantity: 240 },
  { _id: { Name: 'journal', Quantity: 100 }, totalquantity: 100 }
]
```

Calculate total quantity of items grouped by item name and quantity and sort by iname, display all records except first 2 records.

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname",
Quantity:"$quant"},totalquantity:{$sum:"$quant"}}}, {$sort:{"_id.Name":1}}, {$skip:2}]);
```

```
[  { _id: { Name: 'Ball', Quantity: 30 }, totalquantity: 30 },
  { _id: { Name: 'Ball', Quantity: 39 }, totalquantity: 78 },
  { _id: { Name: 'Chairs', Quantity: 6 }, totalquantity: 6 },
  { _id: { Name: 'Chairs', Quantity: 47 }, totalquantity: 94 },
  { _id: { Name: 'Duster', Quantity: 40 }, totalquantity: 120 },
  { _id: { Name: 'Erasers ', Quantity: 500 }, totalquantity: 500 },
  { _id: { Name: 'Erasers ', Quantity: 58 }, totalquantity: 116 },
  { _id: { Name: 'Keyboard', Quantity: 42 }, totalquantity: 126 },
  { _id: { Name: 'Mouse', Quantity: 30 }, totalquantity: 90 },
  { _id: { Name: 'Papers', Quantity: 1250 }, totalquantity: 2500 },
  { _id: { Name: 'Papers', Quantity: 1000 }, totalquantity: 1000 },
  { _id: { Name: 'Stand', Quantity: 350 }, totalquantity: 1050 },
  { _id: { Name: 'journal', Quantity: 120 }, totalquantity: 240 },
  { _id: { Name: 'journal', Quantity: 100 }, totalquantity: 100 }]
```

Calculate total quantity of items grouped by item name and quantity and sort by iname, display all records except first 2 records and limit upto 3.

```
inventory> db.inventory.aggregate([{$group:{_id: {Name:"$iname",
Quantity:"$quant"},totalquantity:{$sum:"$quant"}}}, {$sort:{"_id.Name":1}}, {$skip:2},
{$limit:3}]);
```

```
[ { _id: { Name: 'Ball', Quantity: 30 }, totalquantity: 30 },
  { _id: { Name: 'Ball', Quantity: 39 }, totalquantity: 78 },
  { _id: { Name: 'Chairs', Quantity: 47 }, totalquantity: 94 }]
```

Average:

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname", Quantity:"$quant"}, count:
{$sum:1},totalquantity:{$avg:"$quant"}}}]);
```

```
[{ _id: { Name: 'Chairs', Quantity: 6 }, count: 1, totalquantity: 6 },
{ _id: { Name: 'journal', Quantity: 120 }, count: 2, totalquantity: 120 },
{ _id: { Name: 'Airbrush', Quantity: 12 }, count: 1, totalquantity: 12 },
{ _id: { Name: 'Mouse', Quantity: 30 }, count: 3, totalquantity: 30 },
{ _id: { Name: 'Keyboard', Quantity: 42 }, count: 3, totalquantity: 42 },
{ _id: { Name: 'Papers', Quantity: 1000 }, count: 1, totalquantity: 1000 },
{ _id: { Name: 'Ball', Quantity: 39 }, count: 2, totalquantity: 39 },
{ _id: { Name: 'Chairs', Quantity: 47 }, count: 2, totalquantity: 47 },
{ _id: { Name: 'Erasers ', Quantity: 58 }, count: 2, totalquantity: 58 },
{ _id: { Name: 'Papers', Quantity: 1250 }, count: 2, totalquantity: 1250 },
{ _id: { Name: 'Ball', Quantity: 30 }, count: 1, totalquantity: 30 },
{ _id: { Name: 'journal', Quantity: 100 }, count: 1, totalquantity: 100 },
{ _id: { Name: 'Erasers ', Quantity: 500 }, count: 1, totalquantity: 500 },
{ _id: { Name: 'Duster', Quantity: 40 }, count: 3, totalquantity: 40 },
{ _id: { Name: 'Airbrush', Quantity: 55 }, count: 2, totalquantity: 55 },
{ _id: { Name: 'Stand', Quantity: 350 }, count: 3, totalquantity: 350 }]
```

Grouped by name and average

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname"}, count: {$sum:1},
Average:{$avg:"$quant"}}}]);
```

```
[
  { _id: { Name: 'Ball' }, count: 3, Average: 36 },
  { _id: { Name: 'Mouse' }, count: 3, Average: 30 },
  { _id: { Name: 'Stand' }, count: 3, Average: 350 },
  { _id: { Name: 'Duster' }, count: 3, Average: 40 },
  { _id: { Name: 'Keyboard' }, count: 3, Average: 42 },
  { _id: { Name: 'Chairs' }, count: 3, Average: 33.333333333333336 },
  { _id: { Name: 'Papers' }, count: 3, Average: 1166.6666666666667 },
  { _id: { Name: 'journal' }, count: 3, Average: 113.33333333333333 },
  { _id: { Name: 'Erasers ' }, count: 3, Average: 205.33333333333334 },
  { _id: { Name: 'Airbrush' }, count: 3, Average: 40.666666666666664 }]
```

OR

```
> db.inventory.aggregate([{$group:{_id: {Name:"$iname"}, count: {$sum:1}, TotalQuantity:
{$sum:"$quant"}, Average:{$avg:"$quant"}}}]);
```

```
[  { _id: { Name: 'Ball' }, count: 3, TotalQuantity: 108, Average: 36 },
   {   _id: { Name: 'Stand' },    count: 3,    TotalQuantity: 1050,    Average: 350},
   {    _id: { Name: 'Airbrush' },    count: 3,    TotalQuantity: 122,    Average: 40.666666666666664},
   {    _id: { Name: 'journal' },count: 3,TotalQuantity: 340,    Average: 113.33333333333333  },
   {    _id: { Name: 'Erasers ' }, count: 3,    TotalQuantity: 616,    Average: 205.33333333333334  },
   {    _id: { Name: 'Papers' }, count: 3,    TotalQuantity: 3500,    Average: 1166.6666666666667 },
{    _id: { Name: 'Chairs' },  count: 3,    TotalQuantity: 100,    Average: 33.333333333333336  },
   {    _id: { Name: 'Keyboard' },  count: 3,    TotalQuantity: 126,    Average: 42  },
   {    _id: { Name: 'Duster' }, count: 3,    TotalQuantity: 120,    Average: 40  },
   { _id: { Name: 'Mouse' }, count: 3, TotalQuantity: 90, Average: 30 }]
```

<table>
<tr><th colspan="2">Practical 4</th></tr>
<tr><td colspan="2"><b><u>Aim</u></b>: To demonstrate the Usage of MongoDB in Python.</td></tr>
<tr><td>Name: Saail Chavan</td><td>Roll No: KCTBCS059</td></tr>
<tr><td>Performance date: 02 – 09 – 2023</td><td>Sign:</td></tr>
</table>

Q.1 Download and install pymongo driver to access mongodb database and test the same.

```
->python –m Pip install pymongo
```

Q.2 Create a database “customer” in mongodb using python and perform CRUD operations on the same.
Input and Output:

```
import pymongo
#Creating connection
dbobj = pymongo.MongoClient("mongodb://127.0.0.1:27017/")
#Create database
dbase = dbobj["Customer"]
print("Database Created")
```

#Output

```
Database Created
```

```
#Add data and check
data = dbase["customers"]
#List all dbs in mongodb
print("All Databases:"+str(dbobj.list_database_names()))
```

#Output

```
All Databases:['admin', 'cars', 'config', 'inventory', 'kcfycs', 'local', 'mydb', 'test1']
```

```
#Check if database already exists
dblist =  dbobj.list_database_names()
if "Customer" in dblist:
   print("Database exists")
else:
   print("Not existing")
```

#Output

```
Not existing
```

```
#Insert 1 record
dict1 = {"Name":"Jack","address":"12 Pivet Drive","Cid":"101"}
ins1 = data.insert_one(dict1)
print("\nInserted Record: "+str(ins1.inserted_id))
```

#Output

```
Inserted Record: 63b80215a3412dd89c9ca66e
```

```
#Insert multiple records
l1= [{"Name":"Smith","address":"Dorsey Lane","Cid":"102"},
    {"Name":"Keanu","address":"Anderson Alley","Cid":"103"},
    {"Name":"Thomas","address":"Gotham Crossroad","Cid":"104"},
    {"Name":"Percy","address":"Kings Cross Corner","Cid":"105"},
    {"Name":"Creed","address":"Doughnut Avenue","Cid":"106"},
    {"Name":"Griffin","address":"Doughnut Avenue","Cid":"107"},
    {"Name":"Jim","address":"Scranton PA","Cid":"104"}]
c = data.insert_many(l1)
print("Records inserted: "+str(c.inserted_ids))
```

```
#Output
Records inserted: [ObjectId('63b80215a3412dd89c9ca66f'), ObjectId('63b80215a3412dd89c9ca670'),
ObjectId('63b80215a3412dd89c9ca671'), ObjectId('63b80215a3412dd89c9ca672'),
ObjectId('63b80215a3412dd89c9ca673'), ObjectId('63b80215a3412dd89c9ca674'),
ObjectId('63b80215a3412dd89c9ca675')]
```

```
#Print all docs in collection using find method
print("Records:\n\n")
for data1 in data.find():
   print(data1)
```

```
#Output
Records:
{'_id': ObjectId('63b80215a3412dd89c9ca66e'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b80215a3412dd89c9ca66f'), 'Name': 'Smith', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b80215a3412dd89c9ca670'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b80215a3412dd89c9ca671'), 'Name': 'Thomas', 'address': 'Gotham Crossroad', 'Cid':
'104'}
{'_id': ObjectId('63b80215a3412dd89c9ca672'), 'Name': 'Percy', 'address': 'Kings Cross Corner', 'Cid': '105'}
{'_id': ObjectId('63b80215a3412dd89c9ca673'), 'Name': 'Creed', 'address': 'Doughnut Avenue', 'Cid': '106'}
{'_id': ObjectId('63b80215a3412dd89c9ca674'), 'Name': 'Griffin', 'address': 'Doughnut Avenue', 'Cid': '107'}
{'_id': ObjectId('63b80215a3412dd89c9ca675'), 'Name': 'Jim', 'address': 'Scranton PA', 'Cid': '104'}
```

```
#Find document with address
myq = {"address":"Dorsey Lane"}
mydoc1 = data.find(myq)
print("\n\nRecord with Matched Address: ")
for i in mydoc1:
   print(i)
```

```
#Output
Record with Matched Address:
{'_id': ObjectId('63b80215a3412dd89c9ca66f'), 'Name': 'Smith', 'address': 'Dorsey Lane', 'Cid': '102'}
```

```
#Sort based on name
sorted1 = data.find().sort("Name")
print("\n\nSorted Records: ")
for i in sorted1:
   print(i)
```

#Output

```
Sorted Records:
{'_id': ObjectId('63b80215a3412dd89c9ca673'), 'Name': 'Creed', 'address': 'Doughnut Avenue', 'Cid': '106'}
{'_id': ObjectId('63b80215a3412dd89c9ca674'), 'Name': 'Griffin', 'address': 'Doughnut Avenue', 'Cid': '107'}
{'_id': ObjectId('63b80215a3412dd89c9ca66e'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b80215a3412dd89c9ca675'), 'Name': 'Jim', 'address': 'Scranton PA', 'Cid': '104'}
{'_id': ObjectId('63b80215a3412dd89c9ca670'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b80215a3412dd89c9ca672'), 'Name': 'Percy', 'address': 'Kings Cross Corner', 'Cid': '105'}
{'_id': ObjectId('63b80215a3412dd89c9ca66f'), 'Name': 'Smith', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b80215a3412dd89c9ca671'), 'Name': 'Thomas', 'address': 'Gotham Crossroad', 'Cid':
'104'}
```

```
#Delete one
del1 = {"Name":"Percy"}
d1 = data.delete_one(del1)
var = data.find()
print("\n\nRecords after Deletion: ")
for i in var:
   print(i)

print("Records deleted: "+str(d1.deleted_count))
```

#Output

```
Records after Deletion:
{'_id': ObjectId('63b80215a3412dd89c9ca66e'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b80215a3412dd89c9ca66f'), 'Name': 'Smith', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b80215a3412dd89c9ca670'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b80215a3412dd89c9ca673'), 'Name': 'Creed', 'address': 'Doughnut Avenue', 'Cid': '106'}
{'_id': ObjectId('63b80215a3412dd89c9ca674'), 'Name': 'Griffin', 'address': 'Doughnut Avenue', 'Cid': '107'}
{'_id': ObjectId('63b80215a3412dd89c9ca675'), 'Name': 'Jim', 'address': 'Scranton PA', 'Cid': '104'}
Records deleted: 1
```

```
#Delete many
del1 = {"address":"Doughnut Avenue"}
d1 = data.delete_many(del1)
var = data.find()
print("\n\nRecords after Deletion: ")
for i in var:
   print(i)
print("Records deleted: "+str(d1.deleted_count))
```

#Output

```
Records after Deletion:
{'_id': ObjectId('63b80215a3412dd89c9ca66e'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b80215a3412dd89c9ca66f'), 'Name': 'Smith', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b80215a3412dd89c9ca670'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b80215a3412dd89c9ca675'), 'Name': 'Jim', 'address': 'Scranton PA', 'Cid': '104'}
Records deleted: 2
```

```
#Update one record
up1 = {"Cid":"102"}
set1 = {"$set":{"Name":"Michael"}}
u1 = data.update_one(up1, set1)
var = data.find()
for i in var:
    print(i)
print("\n\nUpdated record count: "+str(u1.modified_count))
```

```
#Output
{'_id': ObjectId('63b80215a3412dd89c9ca66e'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b80215a3412dd89c9ca670'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b80215a3412dd89c9ca675'), 'Name': 'Jim', 'address': 'Scranton PA', 'Cid': '104'}
Updated record count: 1
```

```
#Update many
up2 = {"Cid":"104"}
set2 = {"$set":{"address":"Philly USA"}}
u2 = data.update_many(up2, set2)
var = data.find()
for i in var:
    print(i)
print("\n\nUpdated record count: "+str(u2.modified_count))
```

```
#Output
{'_id': ObjectId('63b802dda3412dd89c9ca677'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b802dda3412dd89c9ca678'), 'Name': 'Michael', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b802dda3412dd89c9ca679'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
{'_id': ObjectId('63b802dda3412dd89c9ca67e'), 'Name': 'Jim', 'address': 'Philly USA', 'Cid': '104'}
Updated record count: 2
```

```
#limit
var = data.find().limit(3)
print("\n\nLimited Records: ")
for i in var:
    print(i)
```

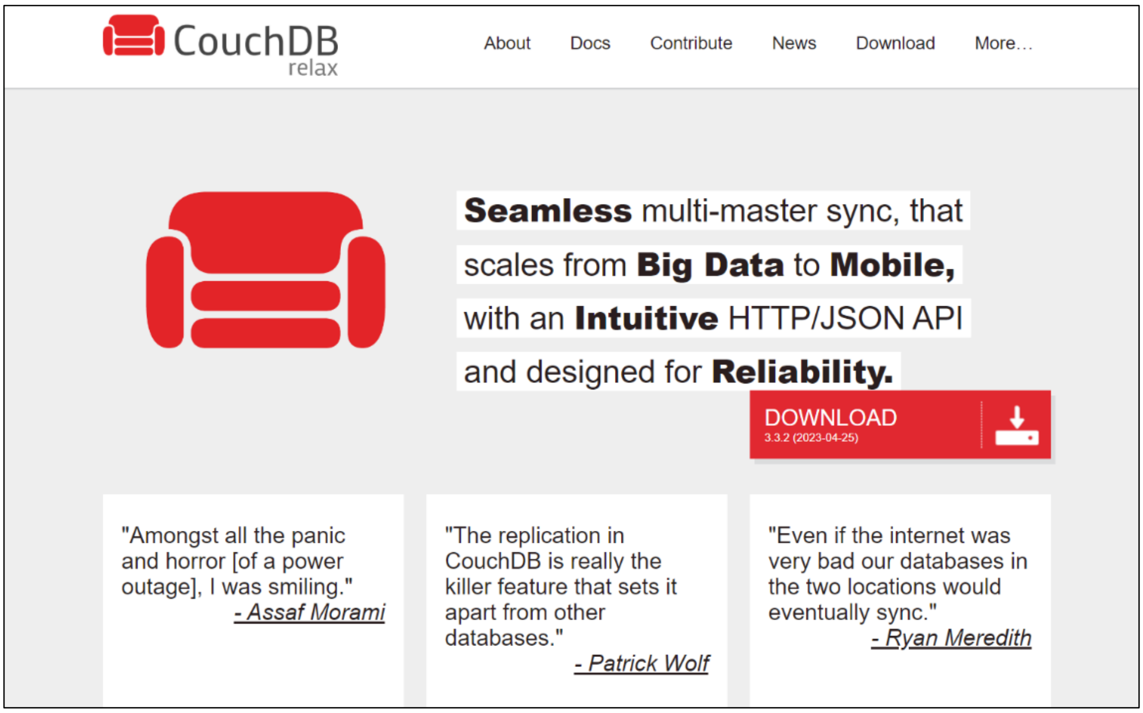
```
#Output
Limited Records:
{'_id': ObjectId('63b802dda3412dd89c9ca677'), 'Name': 'Jack', 'address': '12 Pivet Drive', 'Cid': '101'}
{'_id': ObjectId('63b802dda3412dd89c9ca678'), 'Name': 'Michael', 'address': 'Dorsey Lane', 'Cid': '102'}
{'_id': ObjectId('63b802dda3412dd89c9ca679'), 'Name': 'Keanu', 'address': 'Anderson Alley', 'Cid': '103'}
```

```
#Drop collection
var = data.drop()
print("Dropped Collection")
```

```
#Output
Dropped Collection
```

Practical 5	
Aim: To demonstrate Installation of CouchDB	
Name: Saail Chavan	Roll No:
Performance date:	Sign:

Visit the official website for CouchDB at https://couchdb.apache.org/.
On the homepage, look for the enticing "Download" button.



CouchDB relax

About Docs Contribute News Download More…

Seamless multi-master sync, that scales from **Big Data** to **Mobile,** with an **Intuitive** HTTP/JSON API and designed for **Reliability.**

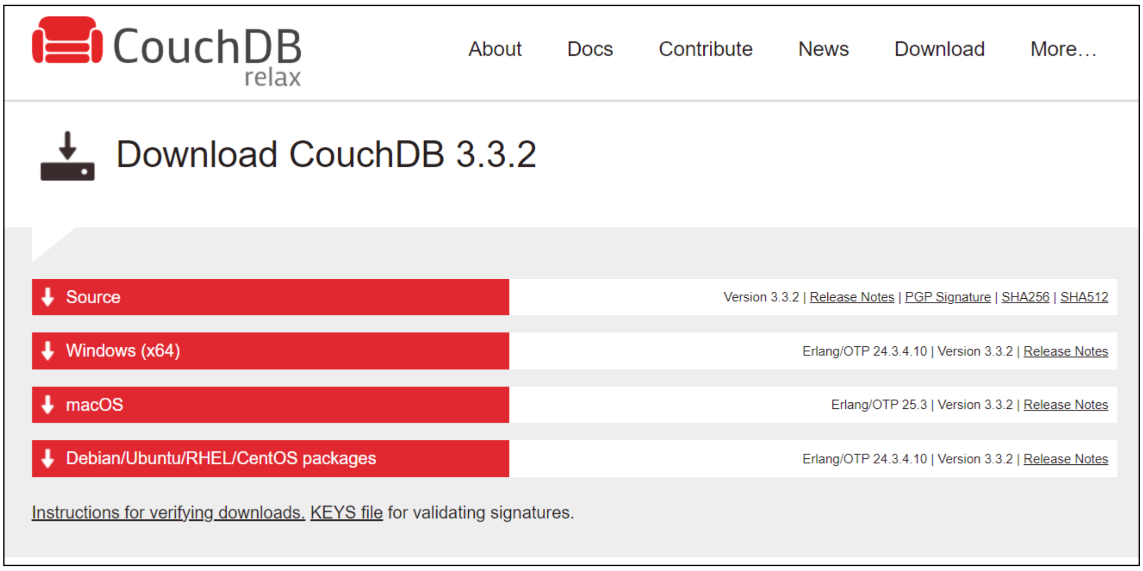
DOWNLOAD
3.3.2 (2023-04-25)

"Amongst all the panic and horror [of a power outage], I was smiling."
- Assaf Morami

"The replication in CouchDB is really the killer feature that sets it apart from other databases."
- Patrick Wolf

"Even if the internet was very bad our databases in the two locations would eventually sync."
- Ryan Meredith

Clicking this button will whisk you away to a page where you will find a collection of download links for CouchDB in various formats.



CouchDB relax

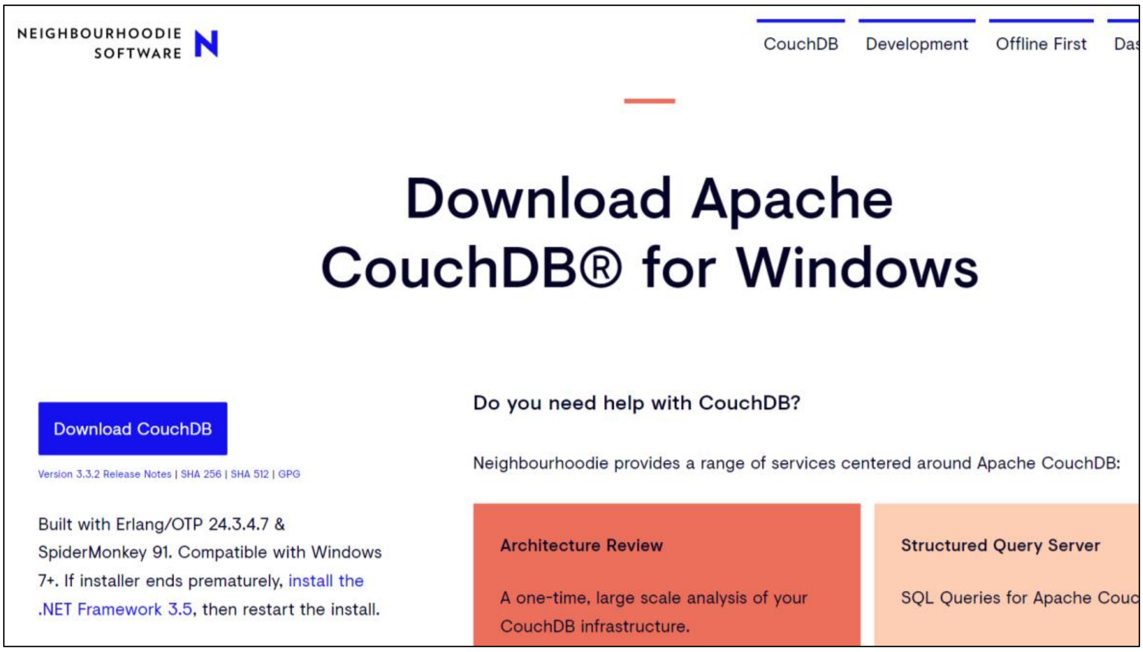
About Docs Contribute News Download More…

Download CouchDB 3.3.2

Source	Version 3.3.2 | Release Notes | PGP Signature | SHA256 | SHA512
Windows (x64)	Erlang/OTP 24.3.4.10 | Version 3.3.2 | Release Notes
macOS	Erlang/OTP 25.3 | Version 3.3.2 | Release Notes
Debian/Ubuntu/RHEL/CentOS packages	Erlang/OTP 24.3.4.10 | Version 3.3.2 | Release Notes

Instructions for verifying downloads. KEYS file for validating signatures.

Among the options, you'll spot a download link tailor-made for Windows systems. Click on it to start download



Download Apache CouchDB® for Windows

Built with Erlang/OTP 24.3.4.7 & SpiderMonkey 91. Compatible with Windows 7+. If installer ends prematurely, install the .NET Framework 3.5, then restart the install.

Do you need help with CouchDB?

Neighbourhoodie provides a range of services centered around Apache CouchDB:

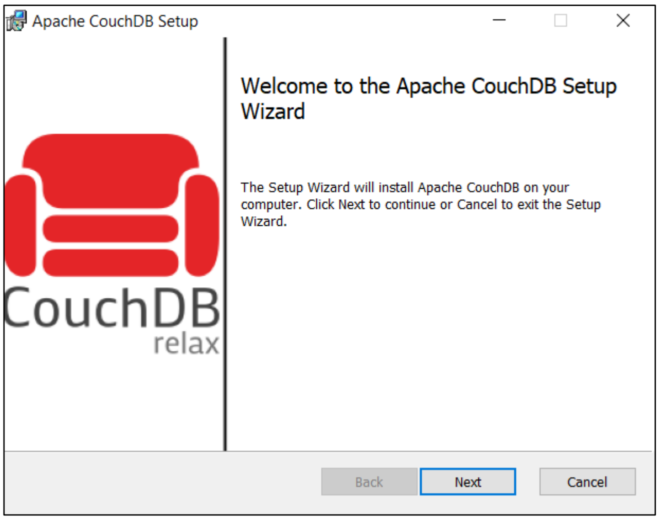
Architecture Review

A one-time, large scale analysis of your CouchDB infrastructure.

Structured Query Server

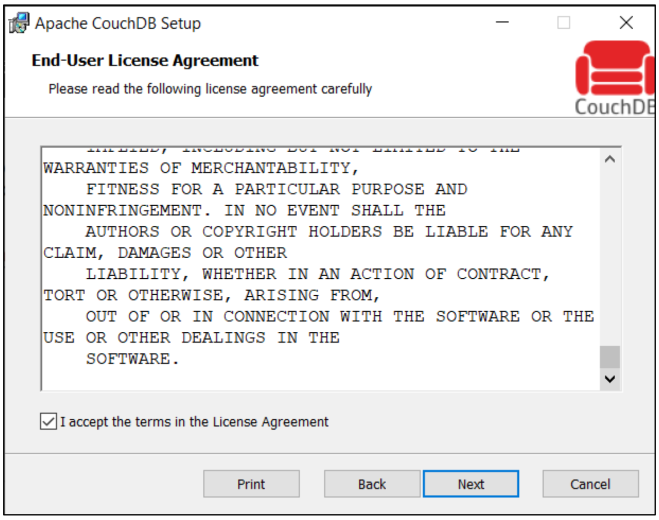
SQL Queries for Apache Couc

Installing CouchDB:



Welcome to the Apache CouchDB Setup Wizard

The Setup Wizard will install Apache CouchDB on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



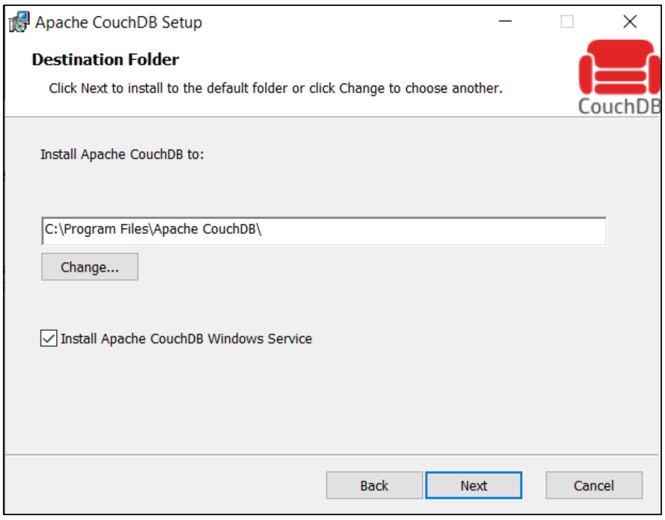
End-User License Agreement

Please read the following license agreement carefully

```
    IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY,
    FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE
    AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER
    LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM,
    OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE
    SOFTWARE.
```

☑ I accept the terms in the License Agreement

Once you've downloaded the Windows executable, start the installation.
Pick your preferred installation path and continue



Destination Folder

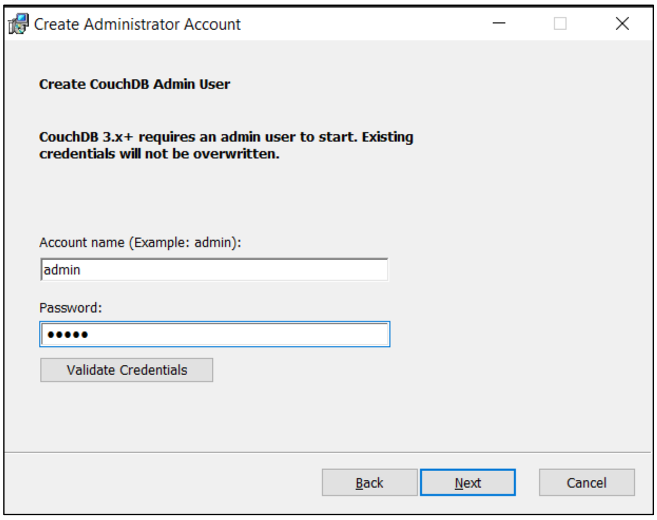
Click Next to install to the default folder or click Change to choose another.

Install Apache CouchDB to:

C:\Program Files\Apache CouchDB\

☑ Install Apache CouchDB Windows Service

Create CouchDB admin credentials for login Set the cookie value to a random

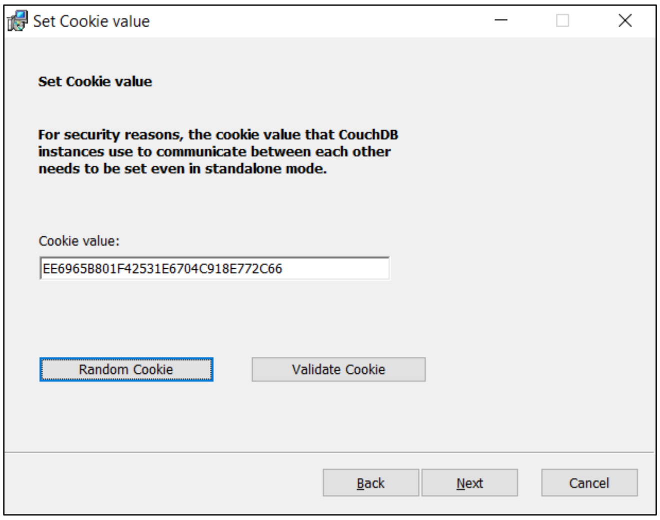


Create CouchDB Admin User

CouchDB 3.x+ requires an admin user to start. Existing credentials will not be overwritten.

Account name (Example: admin): admin

Password: •••••

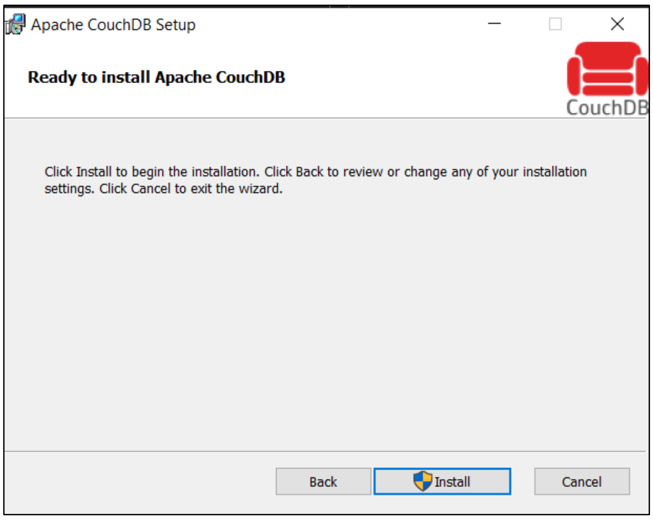


Set Cookie value

For security reasons, the cookie value that CouchDB instances use to communicate between each other needs to be set even in standalone mode.

Cookie value: EE6965B801F42531E6704C918E772C66

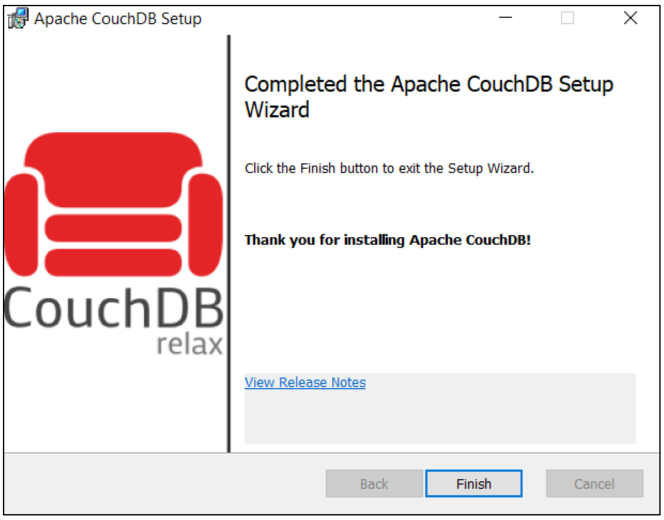
And click on install:



Ready to install Apache CouchDB

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Once the installation is complete click on finish



Completed the Apache CouchDB Setup Wizard

Click the Finish button to exit the Setup Wizard.

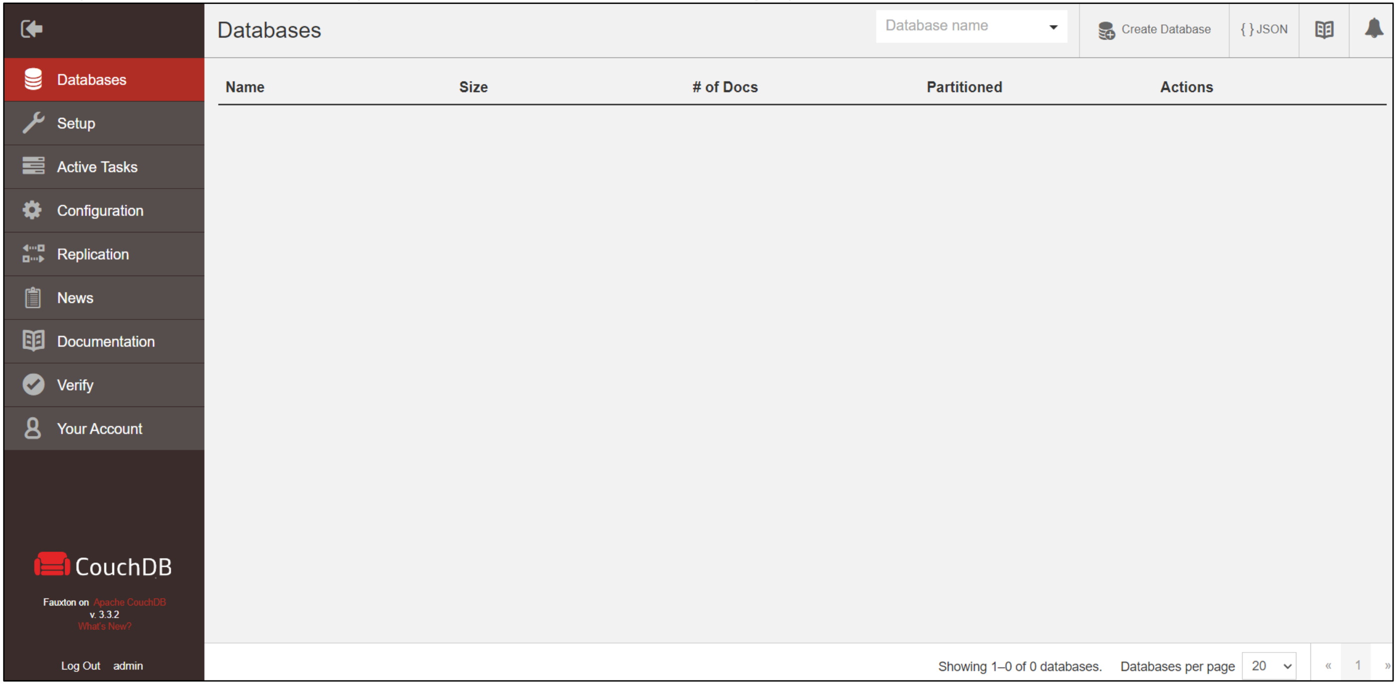
Thank you for installing Apache CouchDB!

Now that you've successfully installed CouchDB, After installation, open built-in web interface of CouchDB by visiting the following link: http://127.0.0.1:5984/

If everything goes fine, this will give you a web page, which have the following output:

```
{
   "couchdb":"Welcome",
   "version":"3.3.2",
   "git_sha":"11a234070",
   "uuid":"52a206d18afee5c0a8c72b2dbebaff75",
   "features":["access-ready","partitioned","pluggable-storage engines","reshard","scheduler"],
   "vendor":
   {
      "name":"The Apache Software Foundation"
   }
}
```

Open your web browser and enter the following URL: http://127.0.0.1:5984/_utils/ .
And login using the given credentials.
Here, you can interact with your CouchDB instance, manage your databases, and explore its features.



Name	Size	# of Docs	Partitioned	Actions

Showing 1–0 of 0 databases.   Databases per page 20

CURL:

If you have version 1803 or later of Windows 10, curl is installed by default. To try it out, open your terminal and type curl –version . If it is installed, then you'll see the version number and which features are enabled.

If you don’t have curl installed or have a version of Windows earlier than Windows 10, you can download and install curl as follows.

1. In Windows, create a folder called **curl** in your C: drive.

2. Go to https://curl.se/download.html and download one of the following zip files:
    - If you have a Windows 64 system, scroll to the **Win64 - Generic** section and look for the latest **Win64 ia64 zip** version with SSL support. It's normally second in the list. Click the version number to start the download.
    - If you have a Windows 32 system, scroll to the **Win32 - Generic** section and look for the latest **Win32 zip** version with SSL support. It's normally second in the list. Click the version number to start the download.

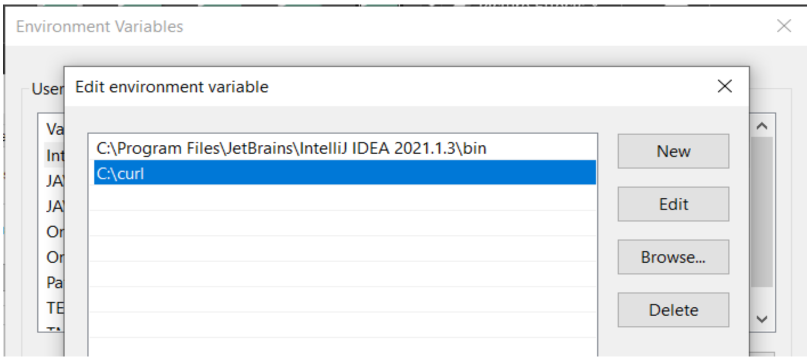
3. Unzip the downloaded file and move the **curl.exe** file to your **C:\curl** folder.

4. Go to https://curl.se/docs/caextract.html and download the digital certificate file named **cacert.pem**.

    (The PEM file contains a bundle of valid digital certificates. The certificates are used to verify the authenticity of secure websites. They're distributed by certificate authority (CA) companies such as GlobalSign and VeriSign. The PEM file allows curl to connect securely to the Zendesk API using the Secure Sockets Layer (SSL) protocol.)

5. Move the **cacert.pem** file to your **C:\curl** folder and rename it **curl-ca-bundle.crt**.

6. Add the curl folder path to your Windows PATH environment variable so that the curl command is available from any location at the command prompt. Update the variable as follows:
    - In the **Start** menu, right-click **This PC** and select **More > Properties**.
      Note: In Windows 7, right-click **Computer** and select **Properties**.
    - Click **Advanced System Settings**.
    - In the **Advanced** tab, click the **Environment Variables** button on the lower right side.
    - Select the "Path" variable in **System Variables**, and click **Edit**.
    - In the **Edit environment variable** dialog box, click **New** and add the path to the **curl.exe** file. Example: C: \curl.



C:\Program Files\JetBrains\IntelliJ IDEA 2021.1.3\bin
C:\curl

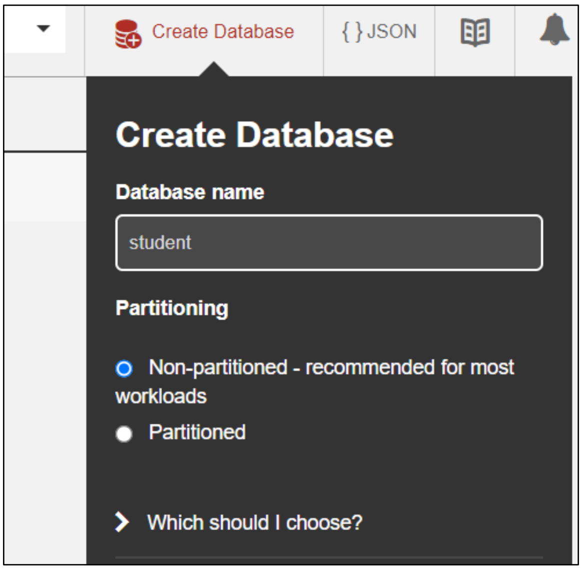
  Windows 7: In the **Variable Value** textbox, append a semicolon to the value, followed by the path to the **curl.exe** file. Example: ;C:\curl

    - Keep clicking OK to accept the change and close the dialog box.

Practical 6	
Aim: To Perform CRUD Operations in CouchDB	
Name: Saail Chavan	Roll No:
Performance date:	Sign:

1. Creating:

Click of the Create Database to manually create a database.



Create Database

Database name

student

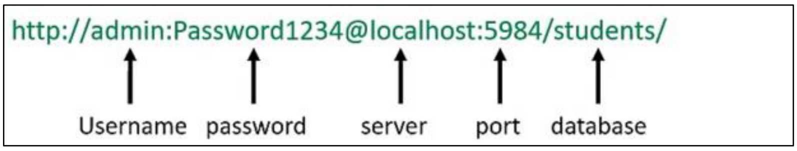
Partitioning

● Non-partitioned - recommended for most workloads

○ Partitioned

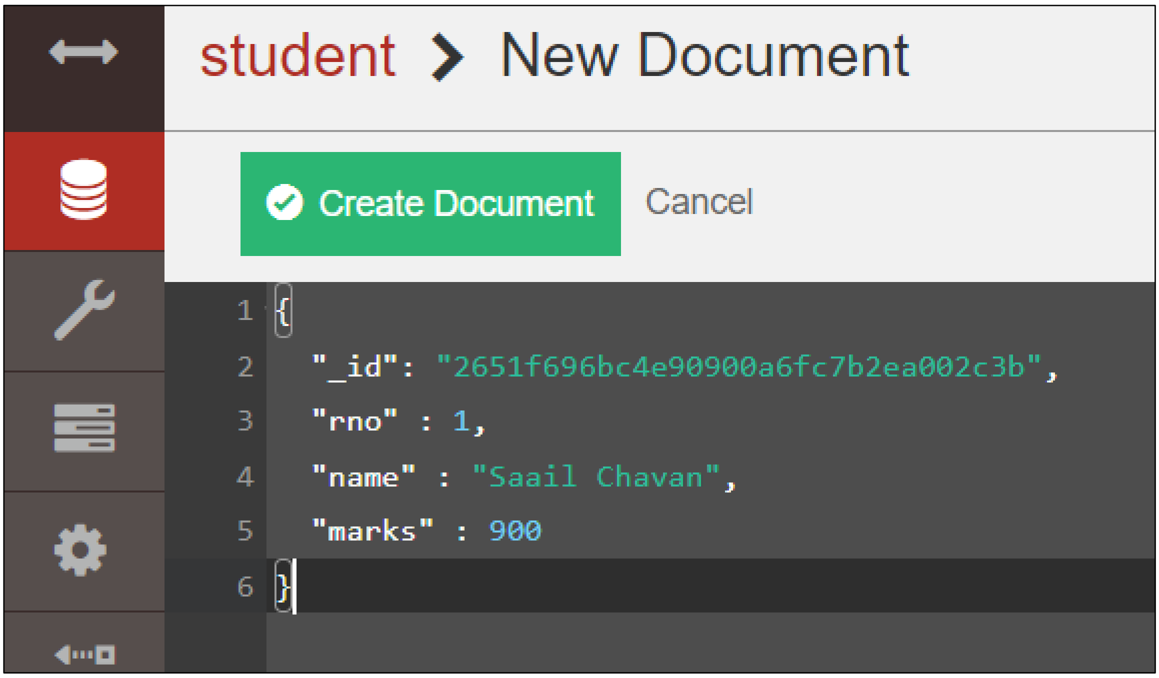
> Which should I choose?

we can also create a database from Command Prompt using:
curl -X PUT http://yourusername:yourpassword@localhost:5984/student

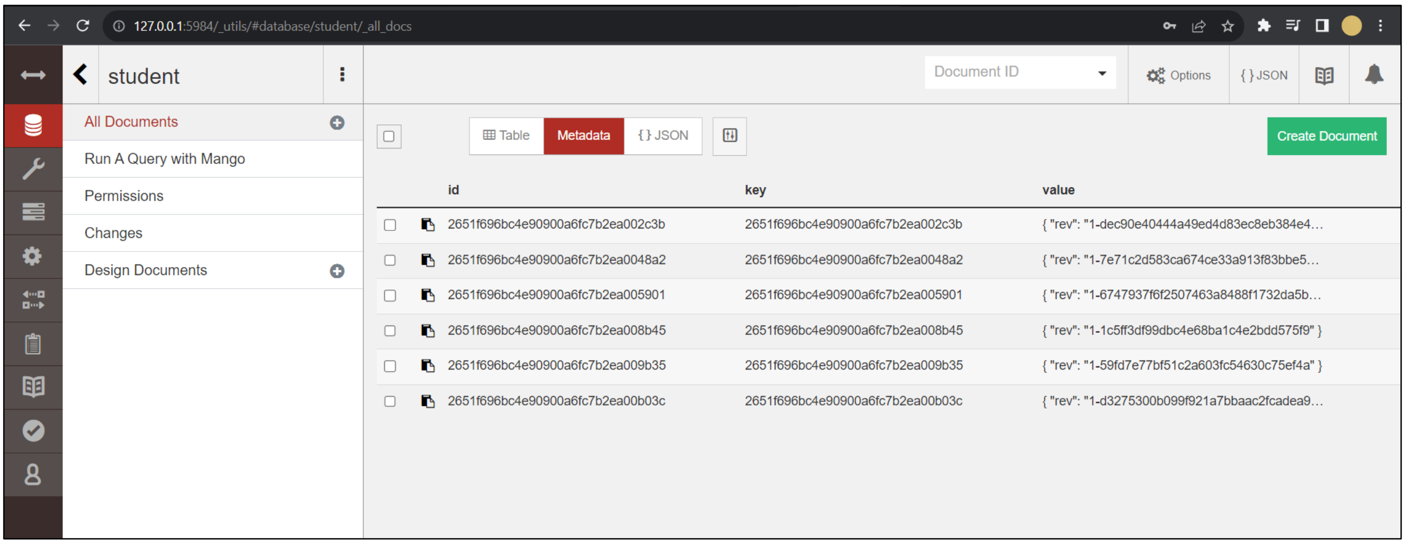


```
http://admin:Password1234@localhost:5984/students/
```

To Create a document click on the database you created and then click on "create database"



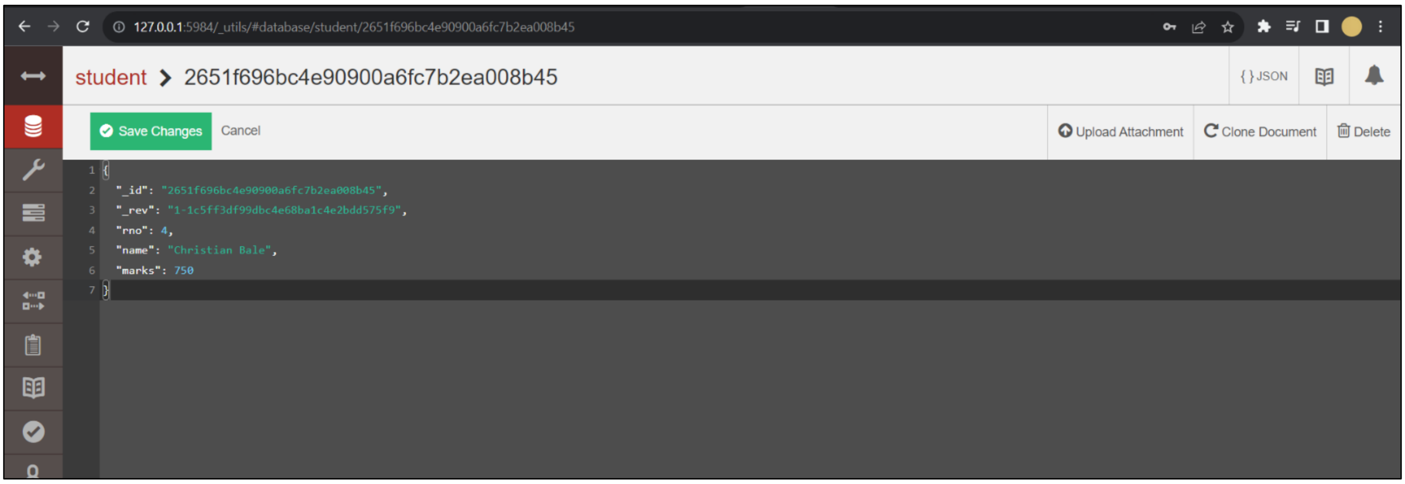
```
{
  "_id": "2651f696bc4e90900a6fc7b2ea002c3b",
  "rno" : 1,
  "name" : "Saail Chavan",
  "marks" : 900
}
```



id	key	value
2651f696bc4e90900a6fc7b2ea002c3b	2651f696bc4e90900a6fc7b2ea002c3b	{ "rev": "1-dec90e40444a49ed4d83ec8eb384e4…
2651f696bc4e90900a6fc7b2ea0048a2	2651f696bc4e90900a6fc7b2ea0048a2	{ "rev": "1-7e71c2d583ca674ce33a913f83bbe5…
2651f696bc4e90900a6fc7b2ea005901	2651f696bc4e90900a6fc7b2ea005901	{ "rev": "1-6747937f6f2507463a8488f1732da5b…
2651f696bc4e90900a6fc7b2ea008b45	2651f696bc4e90900a6fc7b2ea008b45	{ "rev": "1-1c5ff3df99dbc4e68ba1c4e2bdd575f9" }
2651f696bc4e90900a6fc7b2ea009b35	2651f696bc4e90900a6fc7b2ea009b35	{ "rev": "1-59fd7e77bf51c2a603fc54630c75ef4a" }
2651f696bc4e90900a6fc7b2ea00b03c	2651f696bc4e90900a6fc7b2ea00b03c	{ "rev": "1-d3275300b099f921a7bbaac2fcadea9…

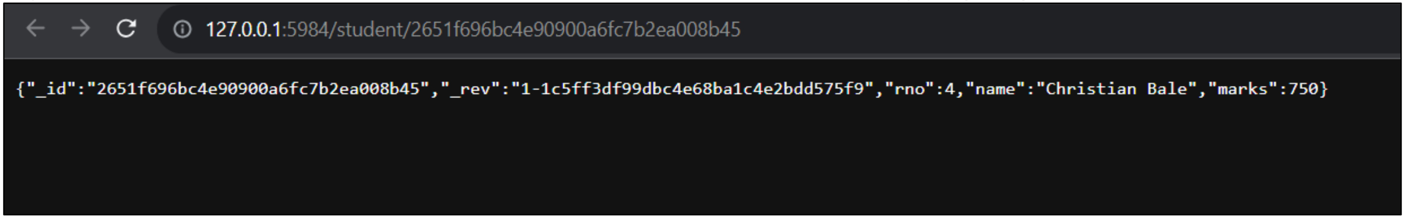
2. Reading:

You can read the document just by clicking on it :


127.0.0.1:5984/_utils/#database/student/2651f696bc4e90900a6fc7b2ea008b45
student > 2651f696bc4e90900a6fc7b2ea008b45
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete

```
{
  "_id": "2651f696bc4e90900a6fc7b2ea008b45",
  "_rev": "1-1c5ff3df99dbc4e68ba1c4e2bdd575f9",
  "rno": 4,
  "name": "Christian Bale",
  "marks": 750
}
```

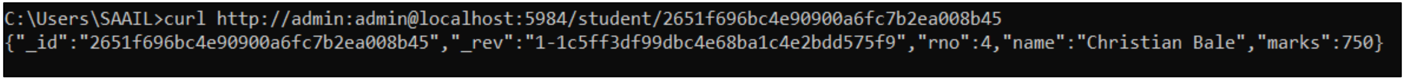
If you click on JSON it will open a new window and show what exactly is going to be sent to the client:



```
{"_id":"2651f696bc4e90900a6fc7b2ea008b45","_rev":"1-1c5ff3df99dbc4e68ba1c4e2bdd575f9","rno":4,"name":"Christian Bale","marks":750}
```

To read the document using curl you will only need the id of the document, In this case:
Id= “2651f696bc4e90900a6fc7b2ea008b45”

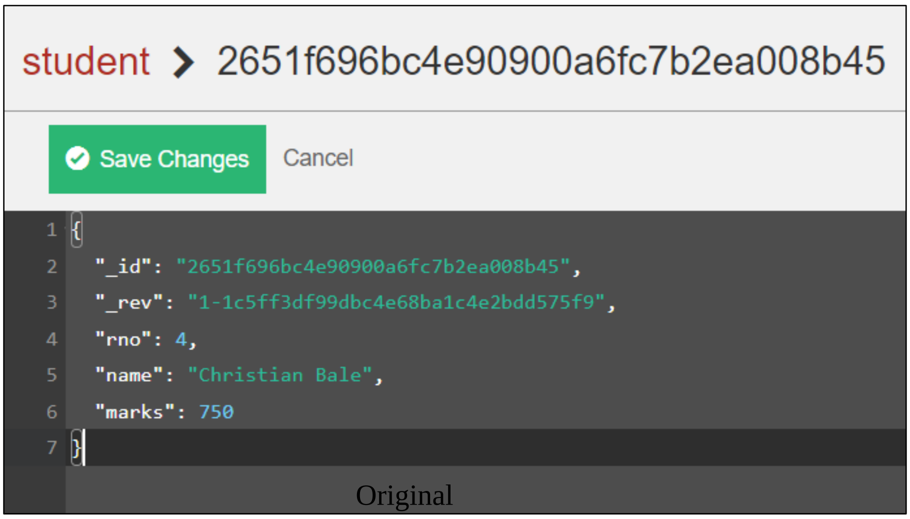
curl http://yourusername:yourpassword@localhost:5984/student/id



```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/2651f696bc4e90900a6fc7b2ea008b45
{"_id":"2651f696bc4e90900a6fc7b2ea008b45","_rev":"1-1c5ff3df99dbc4e68ba1c4e2bdd575f9","rno":4,"name":"Christian Bale","marks":750}
```

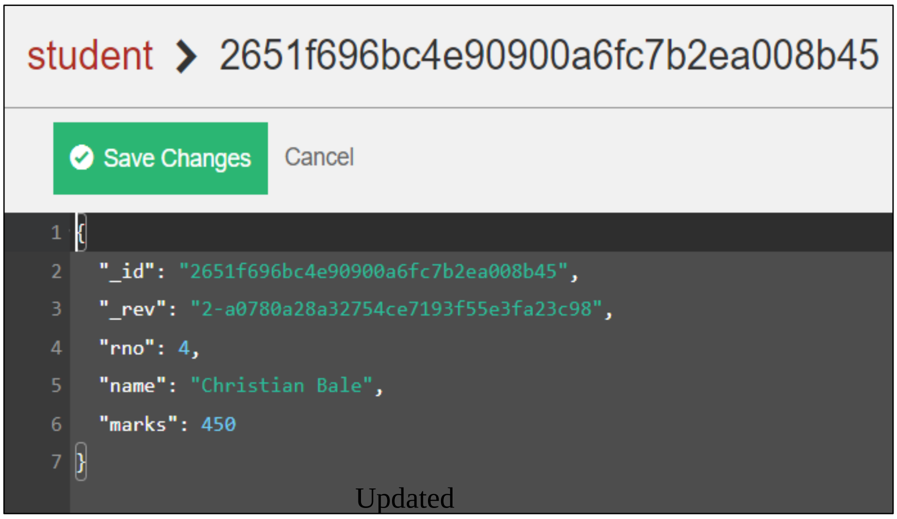
3. Updating:

To Change the values of the document you can open the document, do the changes and then hit save changes:



```
{
  "_id": "2651f696bc4e90900a6fc7b2ea008b45",
  "_rev": "1-1c5ff3df99dbc4e68ba1c4e2bdd575f9",
  "rno": 4,
  "name": "Christian Bale",
  "marks": 750
}
```

Original



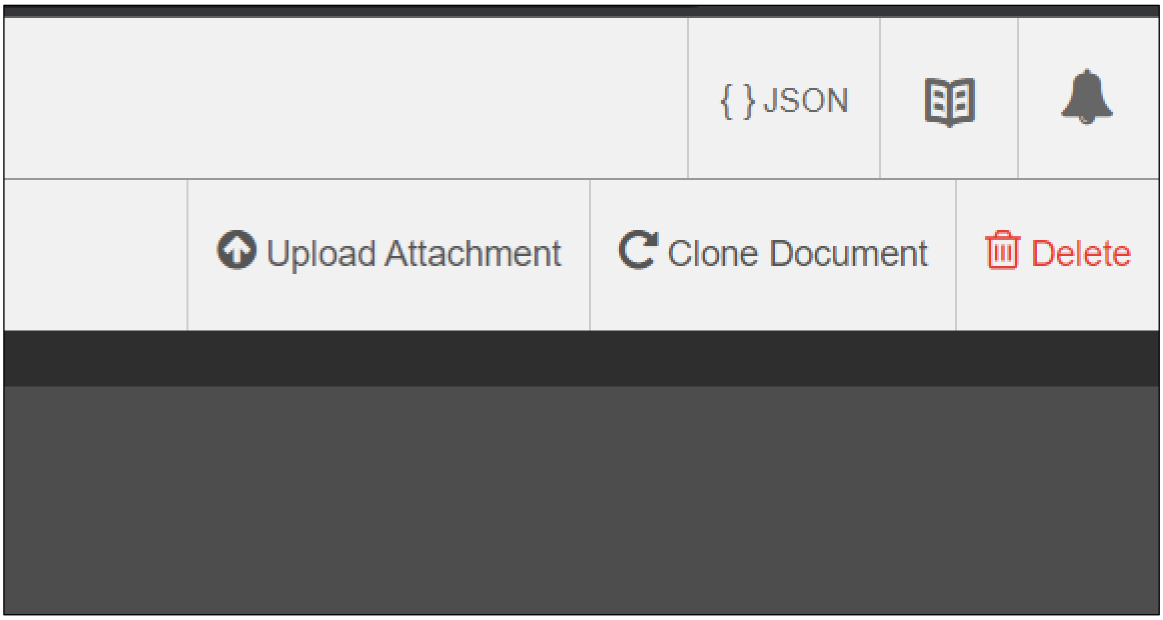
```
{
  "_id": "2651f696bc4e90900a6fc7b2ea008b45",
  "_rev": "2-a0780a28a32754ce7193f55e3fa23c98",
  "rno": 4,
  "name": "Christian Bale",
  "marks": 450
}
```

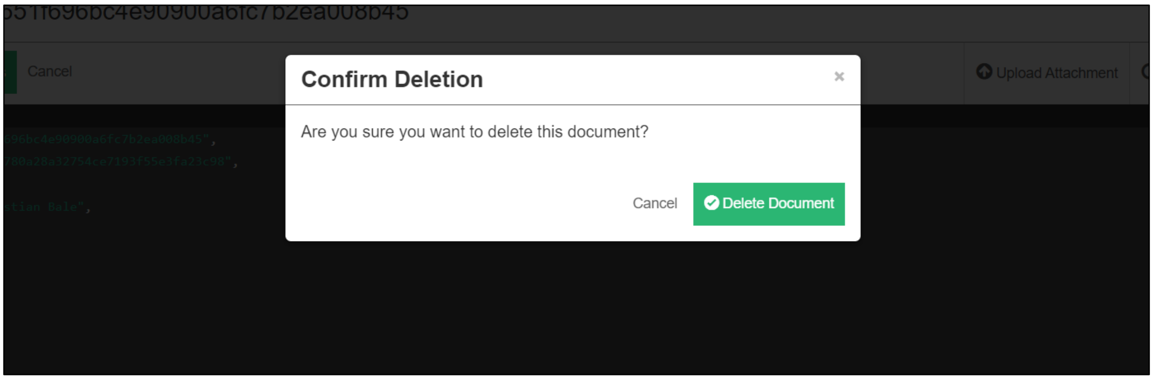
Updated

Here I changed the Marks from 750 to 450 simply by just editing the document.
You can notice that every time you update the document the rev value of the document also changes,
The first number of rev indicates how many revisions there have been
i.e. how many times it has been updated.

4. Deleting:

To delete a document you just have to click on delete on right side



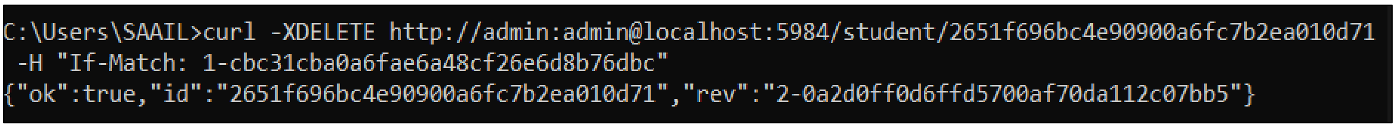

Confirm Deletion

Are you sure you want to delete this document?

And the document will be deleted.

To delete a document using curl you will need the ID and revision value of the document:

Curl -XDELETE http://admin:admin@localhost@5984/student/2651f696bc4e90900a6fc7b2ea010d71 -H “If-Match: 1-cbc31cba0a6fae6a48cf26e6d8b76dbc”



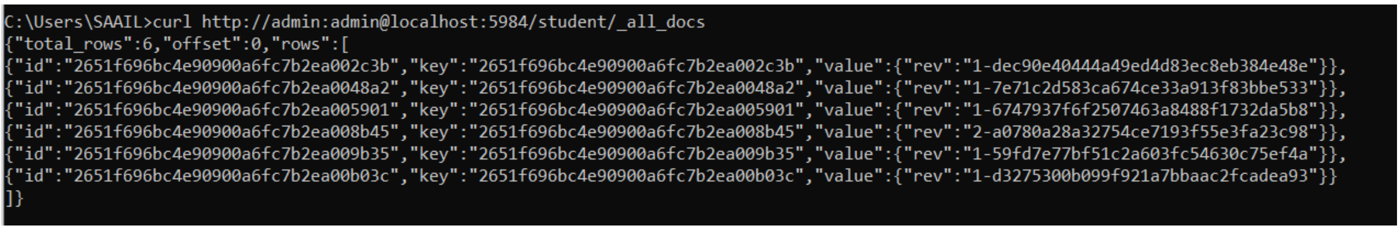
```
C:\Users\SAAIL>curl -XDELETE http://admin:admin@localhost:5984/student/2651f696bc4e90900a6fc7b2ea010d71
 -H "If-Match: 1-cbc31cba0a6fae6a48cf26e6d8b76dbc"
{"ok":true,"id":"2651f696bc4e90900a6fc7b2ea010d71","rev":"2-0a2d0ff0d6ffd5700af70da112c07bb5"}
```

The document will be deleted

Practical 7	
Aim: To Create and Utilize Views in CouchDB	
Name: Saail Chavan	Roll No:
Performance date:	Sign:

There is one default view in CouchDB called _all_docs

- curl http://<user>:<password>@localhost:5984/<database>/_all_docs
- **curl http://admin:admin@localhost:5984/student/_all_docs**

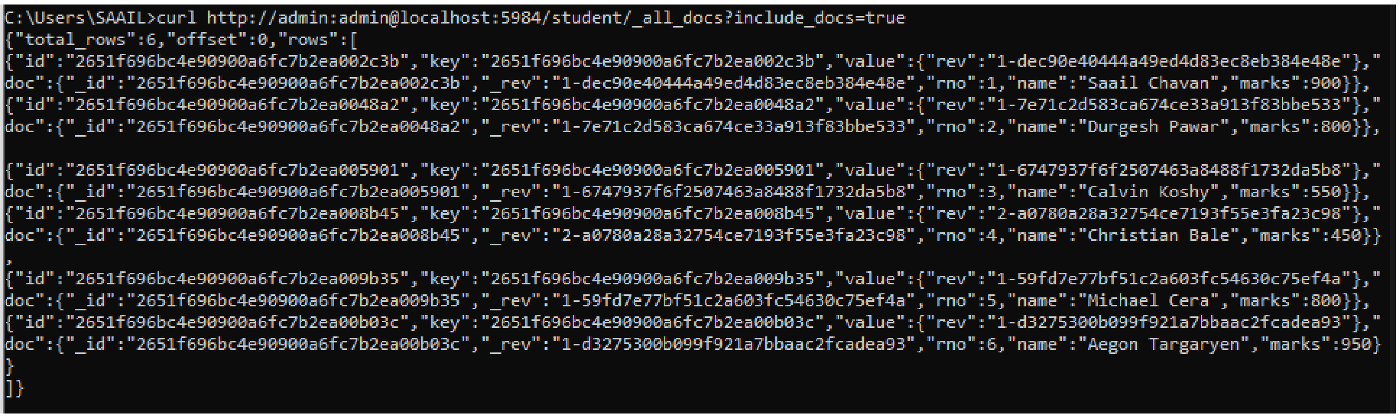


```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_all_docs
{"total_rows":6,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea002c3b","key":"2651f696bc4e90900a6fc7b2ea002c3b","value":{"rev":"1-dec90e40444a49ed4d83ec8eb384e48e"}},
{"id":"2651f696bc4e90900a6fc7b2ea0048a2","key":"2651f696bc4e90900a6fc7b2ea0048a2","value":{"rev":"1-7e71c2d583ca674ce33a913f83bbe533"}},
{"id":"2651f696bc4e90900a6fc7b2ea005901","key":"2651f696bc4e90900a6fc7b2ea005901","value":{"rev":"1-6747937f6f2507463a8488f1732da5b8"}},
{"id":"2651f696bc4e90900a6fc7b2ea008b45","key":"2651f696bc4e90900a6fc7b2ea008b45","value":{"rev":"2-a0780a28a32754ce7193f55e3fa23c98"}},
{"id":"2651f696bc4e90900a6fc7b2ea009b35","key":"2651f696bc4e90900a6fc7b2ea009b35","value":{"rev":"1-59fd7e77bf51c2a603fc54630c75ef4a"}},
{"id":"2651f696bc4e90900a6fc7b2ea00b03c","key":"2651f696bc4e90900a6fc7b2ea00b03c","value":{"rev":"1-d3275300b099f921a7bbaac2fcadea93"}}
]}
```

- _all_docs returns the value of _id and _rev, the two values you need for update and delete

To get the contents of the documents in _all_docs view. Add **?include_docs=true** to the end of the URL,
 ? indicates we are passing argument

curl http://admin:admin@localhost:5984/student/_all_docs?include_docs=true



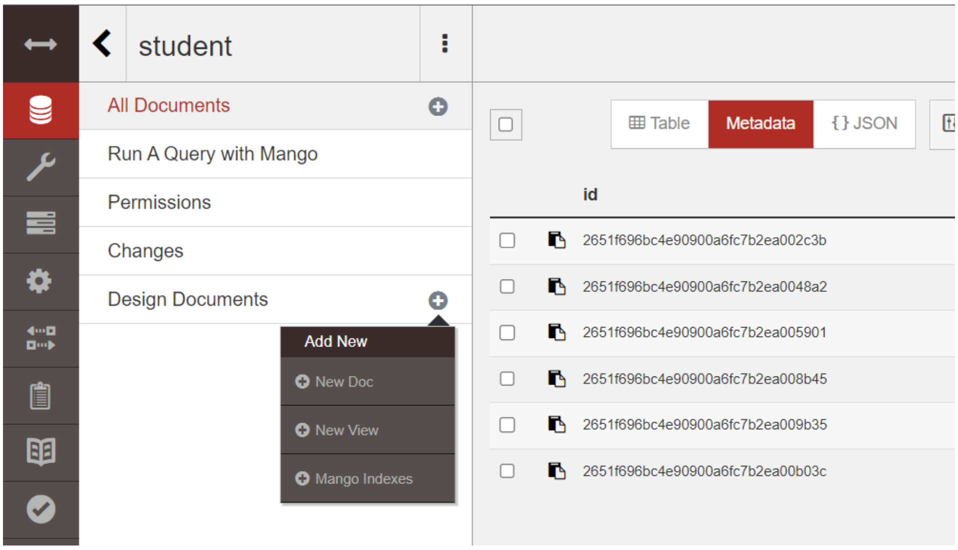
```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_all_docs?include_docs=true
{"total_rows":6,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea002c3b","key":"2651f696bc4e90900a6fc7b2ea002c3b","value":{"rev":"1-dec90e40444a49ed4d83ec8eb384e48e"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea002c3b","_rev":"1-dec90e40444a49ed4d83ec8eb384e48e","rno":1,"name":"Saail Chavan","marks":900}},
{"id":"2651f696bc4e90900a6fc7b2ea0048a2","key":"2651f696bc4e90900a6fc7b2ea0048a2","value":{"rev":"1-7e71c2d583ca674ce33a913f83bbe533"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea0048a2","_rev":"1-7e71c2d583ca674ce33a913f83bbe533","rno":2,"name":"Durgesh Pawar","marks":800}},

{"id":"2651f696bc4e90900a6fc7b2ea005901","key":"2651f696bc4e90900a6fc7b2ea005901","value":{"rev":"1-6747937f6f2507463a8488f1732da5b8"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea005901","_rev":"1-6747937f6f2507463a8488f1732da5b8","rno":3,"name":"Calvin Koshy","marks":550}},
{"id":"2651f696bc4e90900a6fc7b2ea008b45","key":"2651f696bc4e90900a6fc7b2ea008b45","value":{"rev":"2-a0780a28a32754ce7193f55e3fa23c98"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea008b45","_rev":"2-a0780a28a32754ce7193f55e3fa23c98","rno":4,"name":"Christian Bale","marks":450}}
,
{"id":"2651f696bc4e90900a6fc7b2ea009b35","key":"2651f696bc4e90900a6fc7b2ea009b35","value":{"rev":"1-59fd7e77bf51c2a603fc54630c75ef4a"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea009b35","_rev":"1-59fd7e77bf51c2a603fc54630c75ef4a","rno":5,"name":"Michael Cera","marks":800}},
{"id":"2651f696bc4e90900a6fc7b2ea00b03c","key":"2651f696bc4e90900a6fc7b2ea00b03c","value":{"rev":"1-d3275300b099f921a7bbaac2fcadea93"},"
doc":{"_id":"2651f696bc4e90900a6fc7b2ea00b03c","_rev":"1-d3275300b099f921a7bbaac2fcadea93","rno":6,"name":"Aegon Targaryen","marks":950}
}
]}
```

The most important part of view is emit() function
emit() take 2 arguments:

- Key: A value you provide to descrive the document you want
- Value: the value (a JSON object) you want CouchDB to return

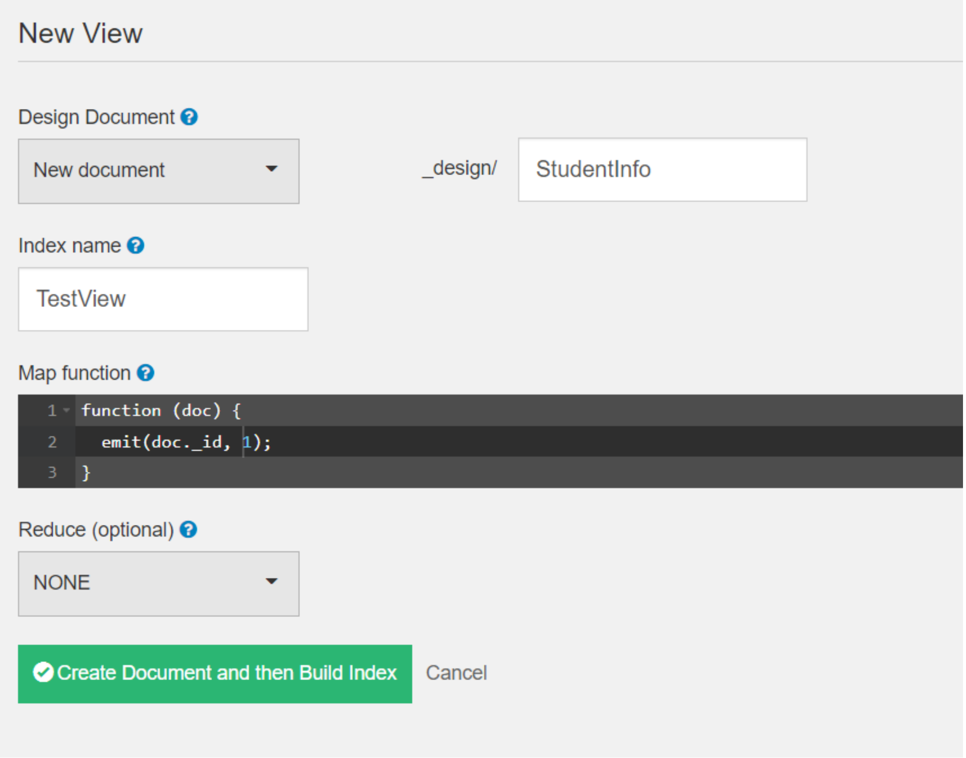
Creating a new view:

- Views are stored in design documents, which are very similar to any other document in CouchDB
- To Create View:
  - Open the database
  - Click on the + next to design document
  - New View



```
2651f696bc4e90900a6fc7b2ea002c3b
2651f696bc4e90900a6fc7b2ea0048a2
2651f696bc4e90900a6fc7b2ea005901
2651f696bc4e90900a6fc7b2ea008b45
2651f696bc4e90900a6fc7b2ea009b35
2651f696bc4e90900a6fc7b2ea00b03c
```

- you can add view to an existing design document or create a new one
  - lets create a new one called: "StudentInfo"
  - and view called: "TestView"



New View

Design Document: New document _design/ StudentInfo

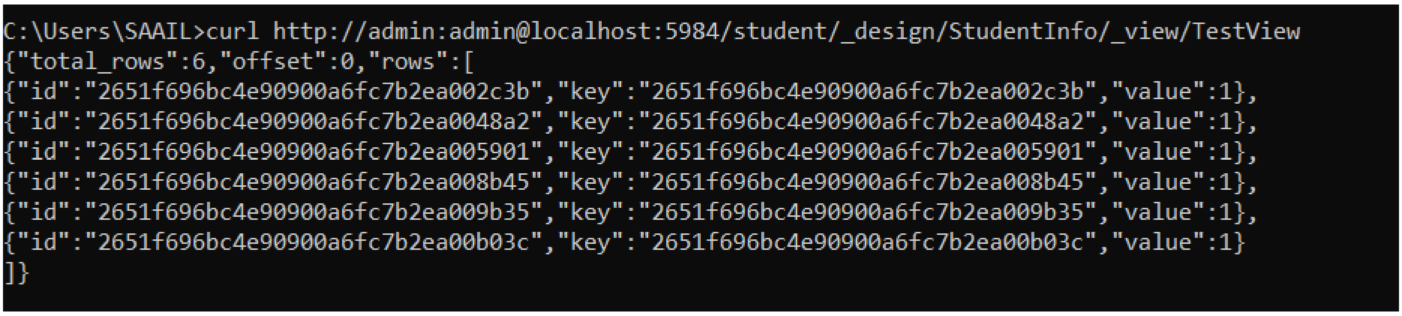
Index name: TestView

Map function:

```
function (doc) {
  emit(doc._id, 1);
}
```

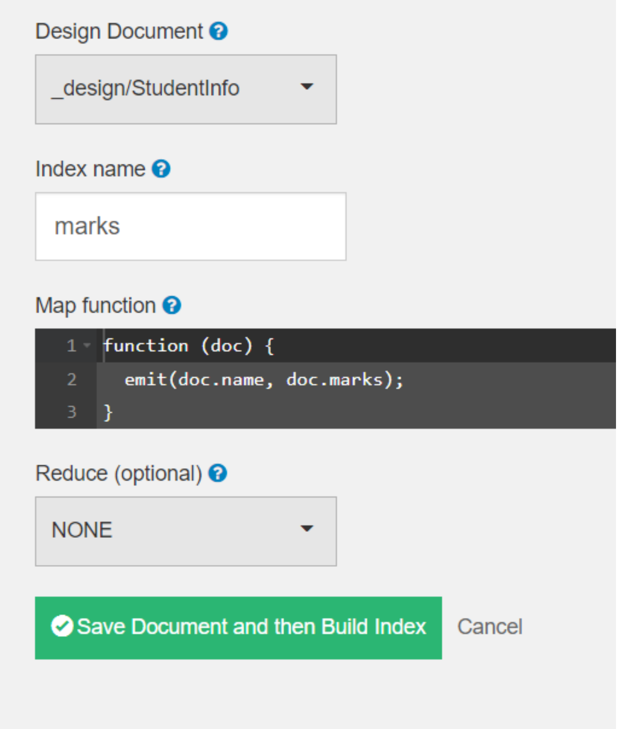
Reduce (optional): NONE

Viewing the created View:
curl http://admin:admin@localhost:5984/student/_design/StudentInfo/_view/TestView



```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_design/StudentInfo/_view/TestView
{"total_rows":6,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea002c3b","key":"2651f696bc4e90900a6fc7b2ea002c3b","value":1},
{"id":"2651f696bc4e90900a6fc7b2ea0048a2","key":"2651f696bc4e90900a6fc7b2ea0048a2","value":1},
{"id":"2651f696bc4e90900a6fc7b2ea005901","key":"2651f696bc4e90900a6fc7b2ea005901","value":1},
{"id":"2651f696bc4e90900a6fc7b2ea008b45","key":"2651f696bc4e90900a6fc7b2ea008b45","value":1},
{"id":"2651f696bc4e90900a6fc7b2ea009b35","key":"2651f696bc4e90900a6fc7b2ea009b35","value":1},
{"id":"2651f696bc4e90900a6fc7b2ea00b03c","key":"2651f696bc4e90900a6fc7b2ea00b03c","value":1}
]}
```

Similarly we can create a view just to display the name and marks of the student:



Design Document

_design/StudentInfo

Index name

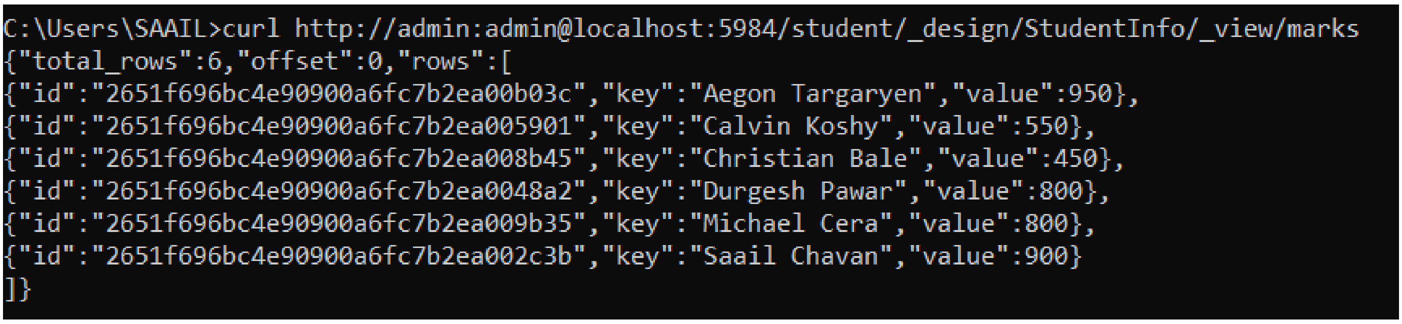
marks

Map function

```
function (doc) {
  emit(doc.name, doc.marks);
}
```

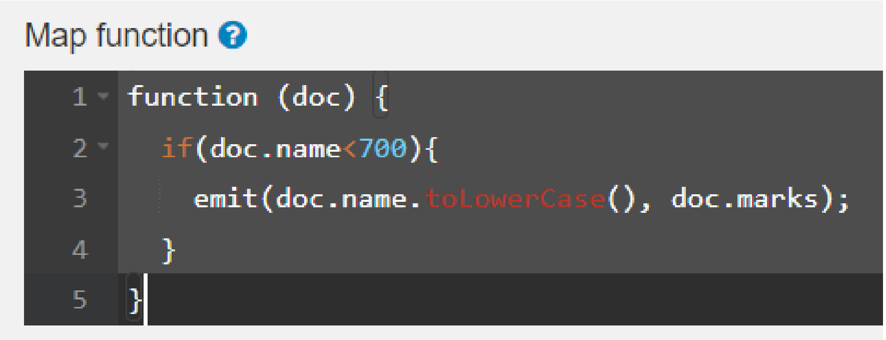
Reduce (optional)

NONE



```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_design/StudentInfo/_view/marks
{"total_rows":6,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea00b03c","key":"Aegon Targaryen","value":950},
{"id":"2651f696bc4e90900a6fc7b2ea005901","key":"Calvin Koshy","value":550},
{"id":"2651f696bc4e90900a6fc7b2ea008b45","key":"Christian Bale","value":450},
{"id":"2651f696bc4e90900a6fc7b2ea0048a2","key":"Durgesh Pawar","value":800},
{"id":"2651f696bc4e90900a6fc7b2ea009b35","key":"Michael Cera","value":800},
{"id":"2651f696bc4e90900a6fc7b2ea002c3b","key":"Saail Chavan","value":900}
]}
```

Display Marks less than 700:



Map function

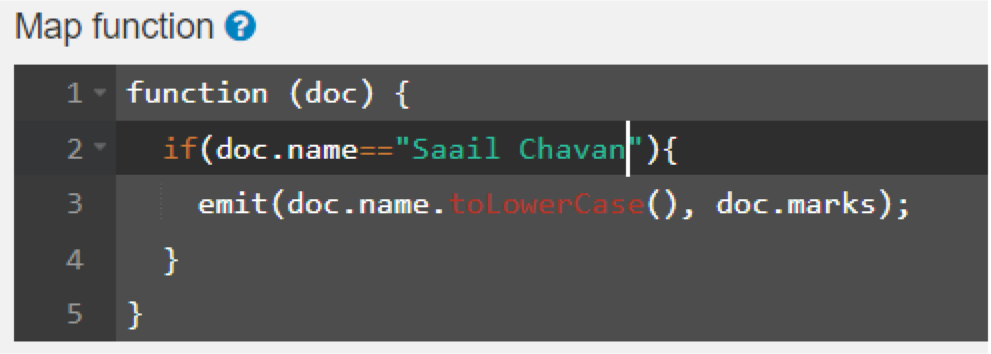
```
function (doc) {
  if(doc.name<700){
    emit(doc.name.toLowerCase(), doc.marks);
  }
}
```

(toLowerCase() is used to display string in lower case)

C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_design/StudentInfo/_view/marks

```
{"total_rows":2,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea005901","key":"calvin koshy","value":550},
{"id":"2651f696bc4e90900a6fc7b2ea008b45","key":"christian bale","value":450}]}
```

Display Marks of specific person:



```
function (doc) {
  if(doc.name=="Saail Chavan"){
    emit(doc.name.toLowerCase(), doc.marks);
  }
}
```

```
C:\Users\SAAIL>curl http://admin:admin@localhost:5984/student/_design/StudentInfo/_view/marks
```

```
{"total_rows":2,"offset":0,"rows":[
{"id":"2651f696bc4e90900a6fc7b2ea002c3b","key":"saail chavan","value":900},
{"id":"d79a1352778f5edb906e3acddf002ff8","key":"saail chavan","value":700}
]}
```

Practical 8	
Aim: HTTP URL Paths and Commands Using CURL to Interact with CouchDB for CRUD Operations.	
Name: Saail Chavan	Roll No:
Performance date: 01 – 10 – 23	Sign:

1. **Check CouchDB Server Status:**

```
>curl http://admin:admin@localhost:5984/
```

```
{"couchdb":"Welcome","version":"3.3.2","git_sha":"11a234070","uuid":"52a206d18afee5c0a8c72b2dbebaff75","features":["access-ready","partitioned","pluggable-storage-engines","reshard","scheduler"],"vendor":{"name":"The Apache Software Foundation"}}
```

---

2. **Create Employee Document:**

```
>curl -X PUT http://admin:admin@localhost:5984/emp/001 -H "Content-Type: application/json" -d "{\"empid\":\"101\",\"empname\":\"saail\",\"salary\":\"3000\"}"
```

```
{"ok":true,"id":"001","rev":"1-099964a79e3feca99feb1e3f1de2bff9"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/001 -H "Content-Type: application/json" -d "{\"empid\":\"102\",\"empname\":\"Durgesh\",\"salary\":\"2000\"}"
```

```
{"error":"conflict","reason":"Document update conflict."}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/002 -H "Content-Type: application/json" -d "{\"empid\":\"102\",\"empname\":\"Durgesh\",\"salary\":\"2000\"}"
```

```
{"ok":true,"id":"002","rev":"1-7e9054fe1e2d9bdb90704d9c3587e962"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/003 -H "Content-Type: application/json" -d "{\"empid\":\"103\",\"empname\":\"Apurva\",\"salary\":\"7000\"}"
```

```
{"ok":true,"id":"003","rev":"1-90861440dd0ae471a16a0eb0bf71c53a"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/004 -H "Content-Type: application/json" -d "{\"empid\":\"104\",\"empname\":\"Hrisabh\",\"salary\":\"1000\"}"
```

```
{"ok":true,"id":"004","rev":"1-73655c91814039cac3597a7bc64edba2"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/005 -H "Content-Type: application/json" -d "{\"empid\":\"105\",\"empname\":\"Parag\",\"salary\":\"10000\"}"
```

```
{"ok":true,"id":"005","rev":"1-261fe29149ce994a6d945f2b1cbdade5"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/006 -H "Content-Type: application/json" -d "{\"empid\":\"106\",\"empname\":\"Sarthak\",\"salary\":\"7000\"}"
```

```
{"ok":true,"id":"006","rev":"1-a6590385e16c4a00568f74584c4b6aa9"}
```

---

3. **List All Databases:**

```
>curl -X GET http://admin:admin@localhost:5984/_all_dbs

["emp","student","testdb"]
```

---

4. **Retrieve Employee Document by ID:**

```
>curl -X GET http://admin:admin@localhost:5984/emp/001

{"_id":"001","_rev":"1-
099964a79e3feca99feb1e3f1de2bff9","empid":"101","empname":"saail","salary":"3000"}

>curl -X GET http://admin:admin@localhost:5984/emp/006

{"_id":"006","_rev":"1-
a6590385e16c4a00568f74584c4b6aa9","empid":"106","empname":"Sarthak","salary":"7000"}
```

---

5. **List All Documents in the "emp" Database:**

```
>curl -X GET http://admin:admin@localhost:5984/emp/_all_docs

{"total_rows":6,"offset":0,"rows":[
{"id":"001","key":"001","value":{"rev":"1-099964a79e3feca99feb1e3f1de2bff9"}},
{"id":"002","key":"002","value":{"rev":"1-7e9054fe1e2d9bdb90704d9c3587e962"}},
{"id":"003","key":"003","value":{"rev":"1-90861440dd0ae471a16a0eb0bf71c53a"}},
{"id":"004","key":"004","value":{"rev":"1-73655c91814039cac3597a7bc64edba2"}},
{"id":"005","key":"005","value":{"rev":"1-261fe29149ce994a6d945f2b1cbdade5"}},
{"id":"006","key":"006","value":{"rev":"1-a6590385e16c4a00568f74584c4b6aa9"}}
]}
```

---

6. **Delete Employee Document with Revision:**

```
>curl -X DELETE http://admin:admin@localhost:5984/emp/004?rev=1-
73655c91814039cac3597a7bc64edba2

{"ok":true,"id":"004","rev":"2-b0b531db5b24345655c187462e1ea7a5"}

>curl -X GET http://admin:admin@localhost:5984/emp/_all_docs

{"total_rows":5,"offset":0,"rows":[
{"id":"001","key":"001","value":{"rev":"1-099964a79e3feca99feb1e3f1de2bff9"}},
{"id":"002","key":"002","value":{"rev":"1-7e9054fe1e2d9bdb90704d9c3587e962"}},
{"id":"003","key":"003","value":{"rev":"1-90861440dd0ae471a16a0eb0bf71c53a"}},
{"id":"005","key":"005","value":{"rev":"1-261fe29149ce994a6d945f2b1cbdade5"}},
{"id":"006","key":"006","value":{"rev":"1-a6590385e16c4a00568f74584c4b6aa9"}}
]}
```

---

7. **Update Employee Document with Revision:**

```
>curl -X PUT http://admin:admin@localhost:5984/emp/001 -H "Content-Type: application/json"
-d "{\"empid\":\"101\",\"empname\":\"saail\",\"salary\":\"5000\",\"yop\":\"5\",\"_rev\":\"1-
099964a79e3feca99feb1e3f1de2bff9\"}"
```

```
{"ok":true,"id":"001","rev":"2-48f49990abb89176a3f40b0a1aabcf80"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/002 -H "Content-Type: application/json"
-d "{\"empid\":\"102\",\"empname\":\"Durgesh\",\"salary\":\"2000\",\"yop\":\"5\",\"_rev\":\"1-
7e9054fe1e2d9bdb90704d9c3587e962\"}"
```

```
{"ok":true,"id":"002","rev":"2-45c3da844221f4f6056c5a474ffa96ad"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/003 -H "Content-Type: application/json"
-d "{\"empid\":\"103\",\"empname\":\"Apurva\",\"salary\":\"7000\",\"yop\":\"2\",\"_rev\":\"1-
90861440dd0ae471a16a0eb0bf71c53a\"}"
```

```
{"ok":true,"id":"003","rev":"2-009fe8daf06a268be48913662b50aa83"}
```

```
>curl -X PUT http://admin:admin@localhost:5984/emp/006 -H "Content-Type: application/json"
-d "{\"empid\":\"106\",\"empname\":\"Sarthak\",\"salary\":\"7000\",\"yop\":\"6\",\"_rev\":\"2-
0ae8afc910dcdab83da578de02cf97a2\"}"
```

```
{"ok":true,"id":"006","rev":"3-518a24481156bd92917f37abda5f854b"}
```