FIT9132 Introduction to Databases

Week 12 Applied Class Activities

Big Data and NOSQL

FIT Database Teaching Team

Complete the following activities listed below

FIT9132 2023 S1

FIT9132 Introduction to Databases

Author: FIT Database Teaching Team

Important

Remember before starting any lab activity which involves working with files, first use SQLDeveloper to pull from the FIT GitLab server so as to ensure your local files and the FIT GitLab server files are in sync.

12.1 MongoDB

12.1.1 The Basics - Setting Up the Environment and Getting Familiar with MongoDB

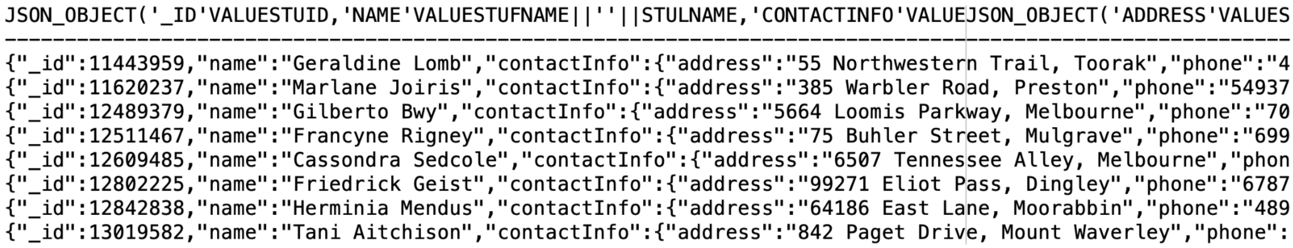
In this applied session we will focus on MongoDB, one of the popular Big Data platforms. The data in MongoDB must be stored in a specific format, for example:

```
{
    "_id": 12489379,
    "name": "Gilberto Bwy",
    "contactInfo": {
      "address": "5664 Loomis Parkway, Melbourne",
      "phone": "7037621034",
      "email": "Gilberto.Bwy@student.monash.edu"
    },
    "enrolmentInfo": [
      {
        "unitcode": "FIT1045",
        "year": "2019",
        "semester": 1,
        "mark": 40,
        "grade": "N"
      },
      {
        "unitcode": "FIT2094",
        "year": "2020",
        "semester": 1,
        "mark": 63,
        "grade": "C"
      },
      {
        "unitcode": "FIT1050",
        "year": "2019",
        "semester": 2,
        "mark": 92,
        "grade": "HD"
      },
      {
        "unitcode": "FIT1045",
        "year": "2019",
        "semester": 2,
        "mark": 89,
        "grade": "HD"
      },
      {
        "unitcode": "FIT1050",
        "year": "2019",
        "semester": 1,
        "mark": 44,
        "grade": "N"
      }
    ]
  }
```

We can generate JSON structures from relational database data and use the JSON output to create suitable documents in MongoDB. To generate JSON of the form shown on the previous page, in Oracle SQL use:

```
SET PAGESIZE 200

SELECT
    JSON_OBJECT ( '_id' VALUE stuid, 'name' VALUE stufname
                || ' '
                || stulname,
                'contactInfo' VALUE JSON_OBJECT (
                    'address' VALUE stuaddress,
                    'phone'   VALUE rtrim(stuphone),
                    'email' VALUE stuemail
                    ),
                'enrolmentInfo' VALUE JSON_ARRAYAGG(
                    JSON_OBJECT(
                        'unitcode' VALUE unitcode,
                        'year' VALUE to_char(ofyear, 'yyyy'),
                        'semester' VALUE ofsemester,
                        'mark' VALUE enrolmark,
                        'grade' VALUE enrolgrade
                        )
                    ) FORMAT JSON )
    || ','
FROM
    uni.student
    NATURAL JOIN uni.enrolment
GROUP BY
    stuid,
    stufname,
    stulname,
    stuaddress,
    stuphone,
    stuemail
ORDER BY
    stuid;
```

This will produce output of the form:

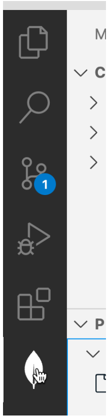


```
JSON_OBJECT('_ID'VALUESTUID,'NAME'VALUESTUFNAME||''||STULNAME,'CONTACTINFO'VALUEJSON_OBJECT('ADDRESS'VALUES
-----------------------------------------------------------------------------------------------------------
{"_id":11443959,"name":"Geraldine Lomb","contactInfo":{"address":"55 Northwestern Trail, Toorak","phone":"4
{"_id":11620237,"name":"Marlane Joiris","contactInfo":{"address":"385 Warbler Road, Preston","phone":"54937
{"_id":12489379,"name":"Gilberto Bwy","contactInfo":{"address":"5664 Loomis Parkway, Melbourne","phone":"70
{"_id":12511467,"name":"Francyne Rigney","contactInfo":{"address":"75 Buhler Street, Mulgrave","phone":"699
{"_id":12609485,"name":"Cassondra Sedcole","contactInfo":{"address":"6507 Tennessee Alley, Melbourne","phon
{"_id":12802225,"name":"Friedrick Geist","contactInfo":{"address":"99271 Eliot Pass, Dingley","phone":"6787
{"_id":12842838,"name":"Herminia Mendus","contactInfo":{"address":"64186 East Lane, Moorabbin","phone":"489
{"_id":13019582,"name":"Tani Aitchison","contactInfo":{"address":"842 Paget Drive, Mount Waverley","phone":
```

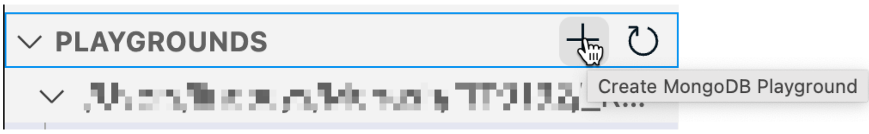
Please note we will only use the first 10 rows of the data output in our work so as to keep the task more manageable when loading data.

Open Microsoft Visual Studio (please install the current version if you do not have it installed) and then follow the instructions in Appendix 1 to set up your MongoDB connection. Please note, like Oracle, you **MUST be on campus or running the Monash VPN** to be able to reach the MongoDB server. *If you are on campus and cannot connect please run the VPN before retrying, unfortunately it appears at the moment that some on-campus locations are blocked from the server without the VPN.*

Next select MongoDB in the left menu panel



Once selected you can click on your connection to connect to the MongoDB database, after connection select to add a new PLAYGROUND

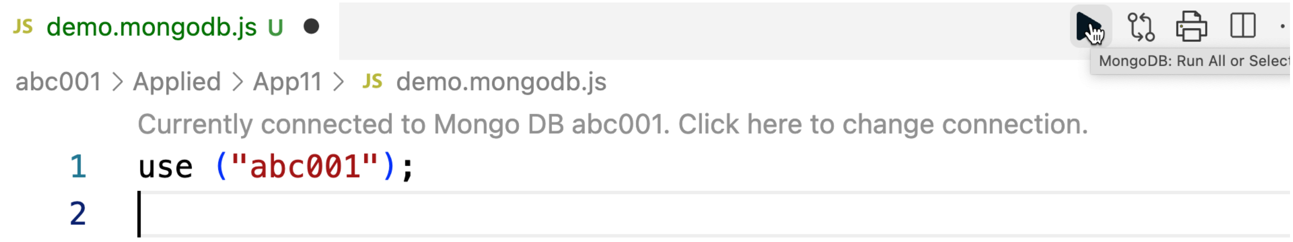


As soon as the playground has been created - use File - Save As and save the playground into your Week 12 Applied folder as demo.mongodb.js (the extension must be **.mongodb.js**).

Type the following into your new playground:

```
use ("abc001");
```

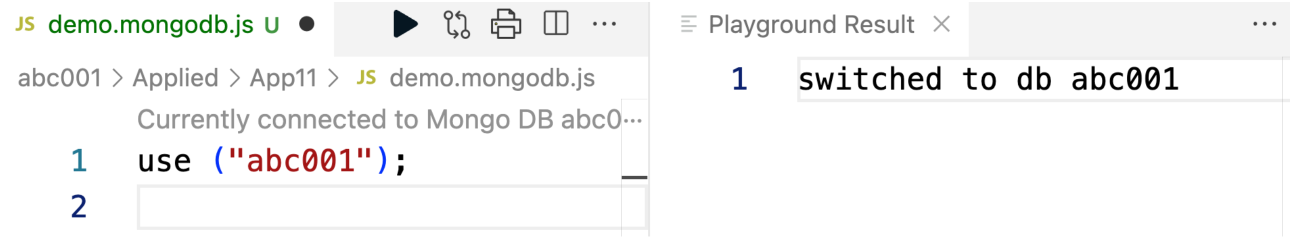
where abc001 is your **authcate** username, and then run the command using the play icon on the top right


demo.mongodb.js U
MongoDB: Run All or Select
abc001 > Applied > App11 > demo.mongodb.js

Currently connected to Mongo DB abc001. Click here to change connection.

```
use ("abc001");
```

Your result will be displayed:



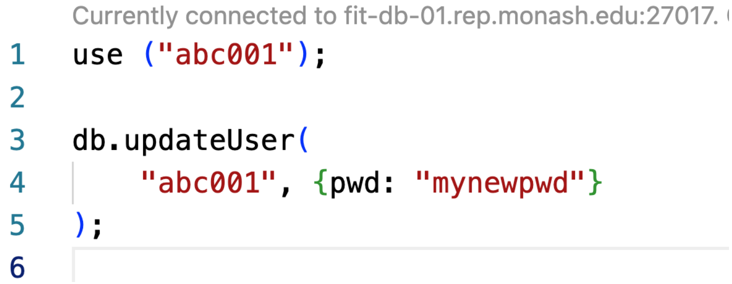
Currently connected to Mongo DB abc0…

```
use ("abc001");
```

```
switched to db abc001
```

Change of Password

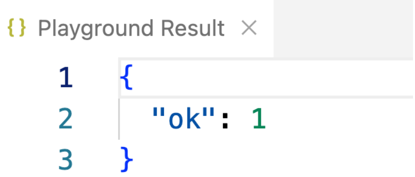
The **first time** you connect to the server, as above, **you must change your password** using the following on a private i.e. non public screen (please replace *mynewpwd* with the password you wish to use):



Currently connected to fit-db-01.rep.monash.edu:27017.

```
use ("abc001");

db.updateUser(
    "abc001", {pwd: "mynewpwd"}
);

```

Run this in Playground and the result you should get will be:



```
{
  "ok": 1
}
```

OK indicating your password has successfully been changed. Please immediately erase the db.updateUser command since it has your new password in clear text.

Unfortunately we are not aware of any simple way to change your MongoDB connection password in VS Code. The simplest approach is to right click on your current connection, select remove connection and then add your connection back with the new password you have just set (see Appendix 1 again to set up the connection).

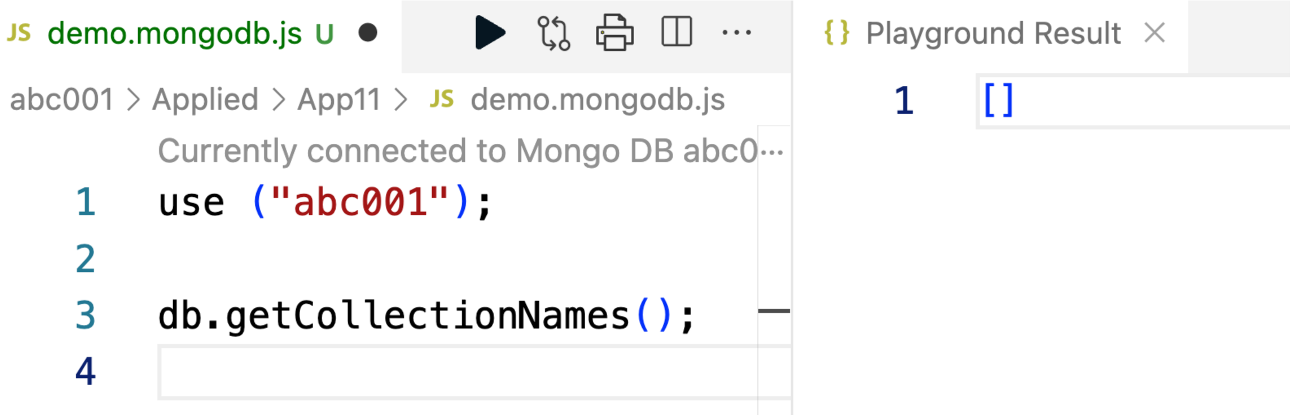
A MongoDB database is a set of collections, and a collection is a set of documents. To show collections in your database, type

```
db.getCollectionNames();
```

and run the commands

At the moment, there is no collection in the results database as you have not added any documents as yet.

The returned document is empty:

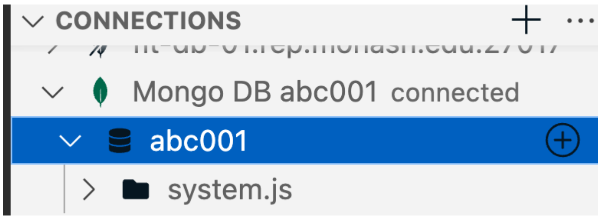


Currently connected to Mongo DB abc0···

```
use ("abc001");

db.getCollectionNames();

```

```
[]
```

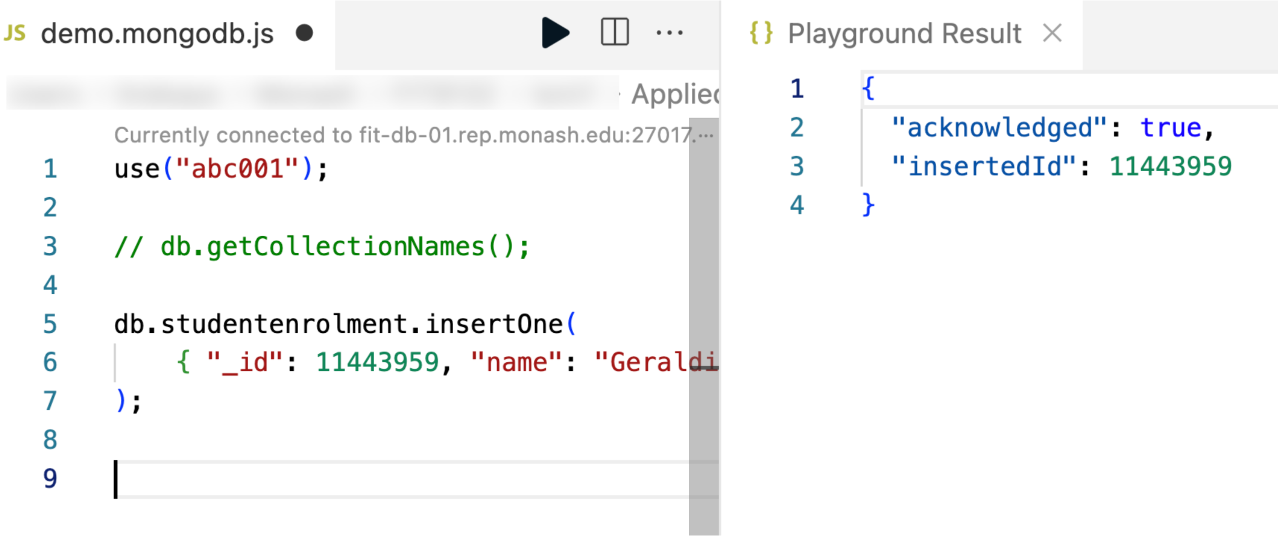
You can also navigate you collections/documents from the CONNECTIONS window - here we currently have no collections



Now, let's create, read, update and delete (CRUD) documents on MongoDB

1. Create a collection by inserting one document into the collection. For this activity, we will call the collection studentenrolment, be sure to remove the comma from the end of the first line of the SQL output.

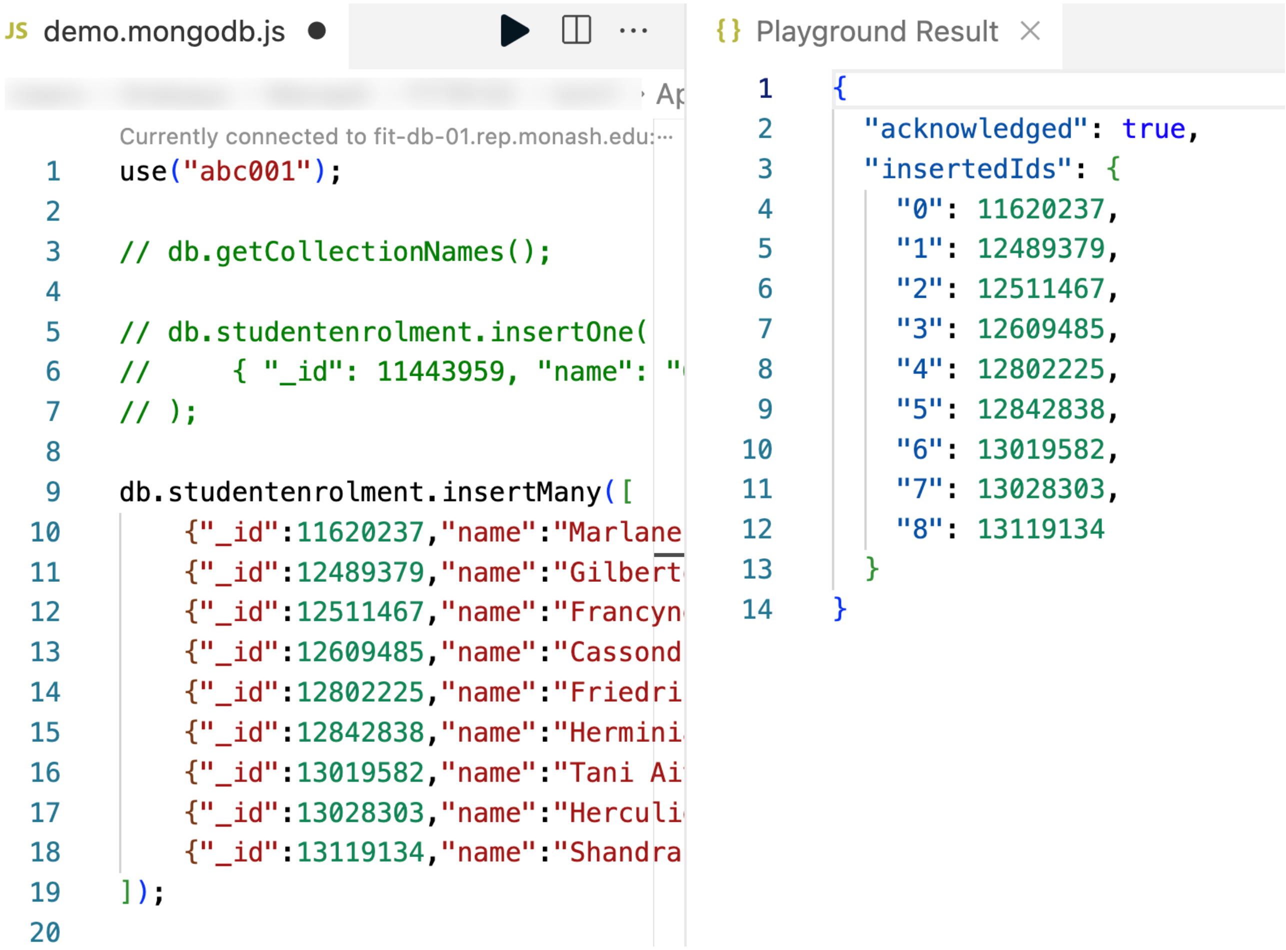
   When running successive commands in Playground using it as a shell type environment comment out the lines you do not wish to run using // on each line (highlight the lines to comment and use F1- Add line comment OR the keyboard shortcut). Your playground result will show the data successfully added:



```
use("abc001");

// db.getCollectionNames();

db.studentenrolment.insertOne(
    { "_id": 11443959, "name": "Geraldi
);

```

```
{
  "acknowledged": true,
  "insertedId": 11443959
}
```

2. Insert the remaining 9 documents into the collection. Note that since we are now inserting many documents they must be enclosed in an array [...] and remember to remove the last comma in the data (leave the ones at each line ONLY remove the very last one):



Currently connected to fit-db-01.rep.monash.edu:…

```
use("abc001");

// db.getCollectionNames();

// db.studentenrolment.insertOne(
//     { "_id": 11443959, "name": "
// );

db.studentenrolment.insertMany([
    {"_id":11620237,"name":"Marlane
    {"_id":12489379,"name":"Gilbert
    {"_id":12511467,"name":"Francyn
    {"_id":12609485,"name":"Cassond
    {"_id":12802225,"name":"Friedri
    {"_id":12842838,"name":"Hermini
    {"_id":13019582,"name":"Tani Ai
    {"_id":13028303,"name":"Herculi
    {"_id":13119134,"name":"Shandra
]);
```

```
{
  "acknowledged": true,
  "insertedIds": {
    "0": 11620237,
    "1": 12489379,
    "2": 12511467,
    "3": 12609485,
    "4": 12802225,
    "5": 12842838,
    "6": 13019582,
    "7": 13028303,
    "8": 13119134
  }
}
```

3. Now time for you to retreive/update/delete some data - add each of the following to a new line in your playground and run the command in MongoDB to test the output:

   a. Check how many documents have been inserted into the collection

   ```
   db.studentenrolment.countDocuments()
   ```

   b. Read all documents

   ```
   db.studentenrolment.find();
   ```

   c. Show the data for student id = 13119134

   ```
   db.studentenrolment.find({"_id":{"$eq":13119134}})
   ```

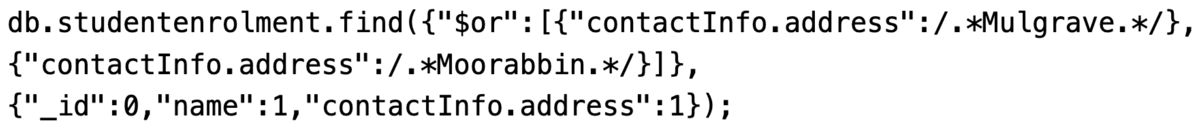
   or in shorthand form (omitting "$eq"):

   ```
   db.studentenrolment.find({"_id":13119134});
   ```

   d. Show id and name of students who were enrolled in FIT3157

   ```
   db.studentenrolment.find({"enrolmentInfo.unitcode":"FIT3157"},
   {"_id":1,"name":1});
   ```

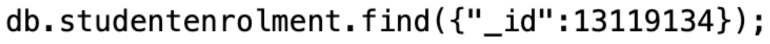
   e. Show the name and address of students who live in Mulgrave or Moorabbin



```
db.studentenrolment.find({"$or":[{"contactInfo.address":/.*Mulgrave.*/},
{"contactInfo.address":/.*Moorabbin.*/}]},
{"_id":0,"name":1,"contactInfo.address":1});
```

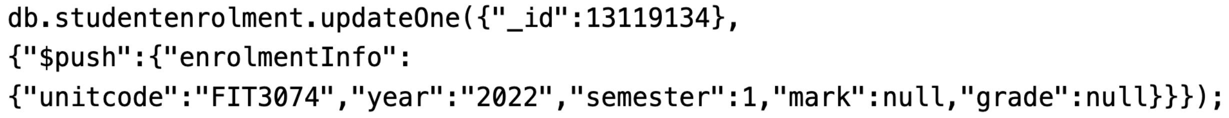
4. Add/remove data from an array

   a. Show the data for student id = 13119134



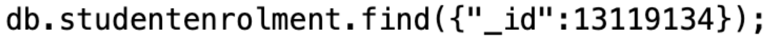
```
db.studentenrolment.find({"_id":13119134});
```

   b. Add a new enrolment for student id = 13119134, the student was enrolled in FIT3074 in semester 1 2022. Set the mark and grade as null.



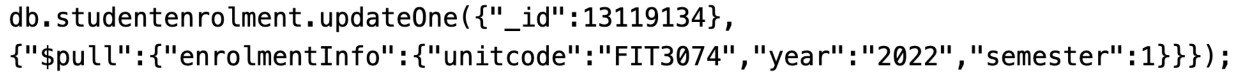
```
db.studentenrolment.updateOne({"_id":13119134},
{"$push":{"enrolmentInfo":
{"unitcode":"FIT3074","year":"2022","semester":1,"mark":null,"grade":null}}});
```

   c. Check if the data is correctly inserted



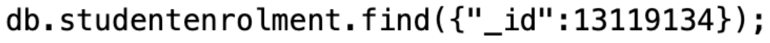
```
db.studentenrolment.find({"_id":13119134});
```

   d. Remove the enrolment data for student id = 13119134 in FIT3074 for semester 1 2022.



```
db.studentenrolment.updateOne({"_id":13119134},
{"$pull":{"enrolmentInfo":{"unitcode":"FIT3074","year":"2022","semester":1}}});
```

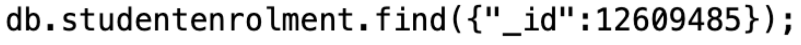
   e. Check if the data is correctly removed



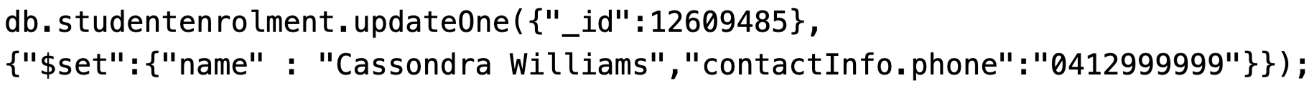
```
db.studentenrolment.find({"_id":13119134});
```

5. Update Value

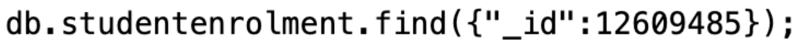
   Update student id 12609485's name from Cassondra Sedcole to Cassondra Williams. She also changed her phone number to 0412999999



```
db.studentenrolment.find({"_id":12609485});
```

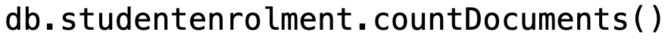


```
db.studentenrolment.updateOne({"_id":12609485},
{"$set":{"name" : "Cassondra Williams","contactInfo.phone":"0412999999"}});
```

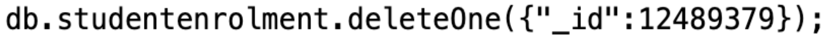


```
db.studentenrolment.find({"_id":12609485});
```

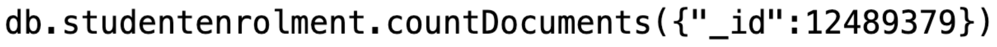
6. Delete all details (document) of student id 12489379



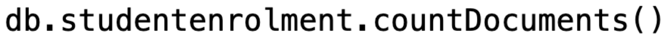
```
db.studentenrolment.countDocuments()
```



```
db.studentenrolment.deleteOne({"_id":12489379});
```



```
db.studentenrolment.countDocuments({"_id":12489379})
```



```
db.studentenrolment.countDocuments()
```

12.1.2 MongoDB Create Update and Delete

1. **Download the sql file from the Week 12 Moodle page (week12_bigdata.sql)** and save it into your week 12 Applied folder. Use the file to write an SQL select statement to generate a collection of documents using the following structure/format from the UNI database.

```
{
    "_id": 12489379,
    "name": "Gilberto Bwy",
    "contactInfo": {
      "address": "5664 Loomis Parkway, Melbourne",
      "phone": "7037621034",
      "email": "Gilberto.Bwy@student.monash.edu"
    },
    "dob": "30-08-1992",
    "enrolmentInfo": [
      {
        "unitcode": "FIT1045",
        "unitname": "Algorithms and programming fundamentals in python",
        "year": "2019",
        "semester": 1,
        "mark": 40,
        "grade": "N"
      },
      {
        "unitcode": "FIT2094",
        "unitname": "Databases",
        "year": "2020",
        "semester": 1,
        "mark": 63,
        "grade": "C"
      },
      {
        "unitcode": "FIT1050",
        "unitname": "Web fundamentals",
        "year": "2019",
        "semester": 2,
        "mark": 92,
        "grade": "HD"
      },
      {
        "unitcode": "FIT1045",
        "unitname": "Algorithms and programming fundamentals in python",
        "year": "2019",
        "semester": 2,
        "mark": 89,
        "grade": "HD"
      },
      {
        "unitcode": "FIT1050",
        "unitname": "Web fundamentals",
        "year": "2019",
        "semester": 1,
        "mark": 44,
        "grade": "N"
      }
    ]
  }
```

Download the playground file from the Week 12 Moodle page (week12_bigdata.mongodb.js), and save it into your week 12 Applied folder. Using this playground, carry out the following:

2. Create a new collection named as **enrolment** and insert the first 10 documents generated by the select statement from 1 above into MongoDB
3. Create a new enrolment for studid 12489379, the student is enrolled in FIT2002 (IT Project Management) in semester 1 2022. Since this is a new enrollment, set the mark and the grade as null.
4. Update this enrolment for studid 12489379 in FIT2002, set the mark to 65 and grade to C
5. Delete this enrolment for student id 12489379 in FIT2002

12.1.3 MongoDB Read

Continue working in your week12_bigdata.mongodb.js playground and write db.find() commands for the following questions:

1. Retrieve the document for student id = 12802225
2. Show the id and name of students who have any mark greater than 95 in any enrolment (hint: use $gt:95)
3. Retrieve the name and contact info of students who enrolled in any unit which has "web design" as part of its name
4. Retrieve the id and name of any students who have grades WH or N

12.2 SETU

Please complete your SET for this unit (it provides important feedback to the teaching staff)

Important

You need to get into the habit of establishing this as a standard FIT9132 workflow - Pull at the start of your working session, work on the activities you wish to/are able to complete during this session, add (stage), commit changes and then Push the changes back to the FIT GitLab server

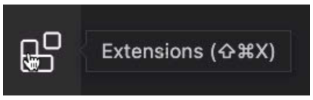
Appendix 1 Configuration of Visual Studio Code

MongoDB provides a Visual Studio Code extension for access to a MongoDB database. The full documentation is available here:

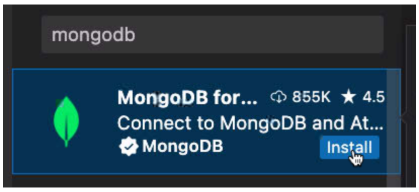
https://www.mongodb.com/docs/mongodb-vscode/

Step 1 Add the VS Extension for MongoDB

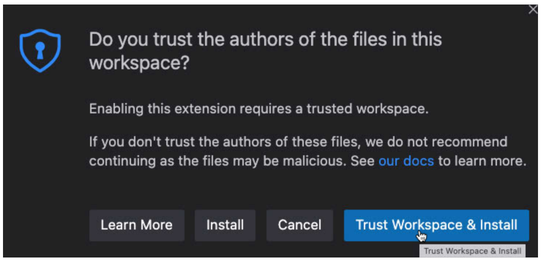
Select Extensions in the left menu:



Search for the extension MongoDB for VSCode and then click on Install



Click on Trust Workspace & Install



Do you trust the authors of the files in this workspace?

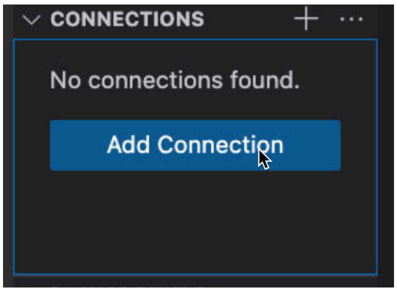
Enabling this extension requires a trusted workspace.

If you don't trust the authors of these files, we do not recommend continuing as the files may be malicious. See our docs to learn more.

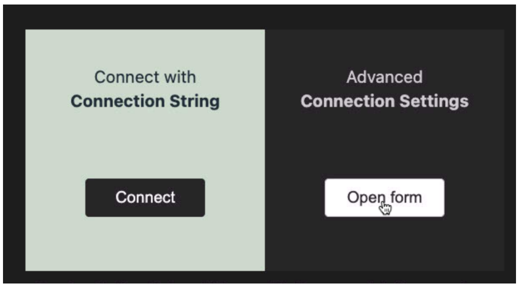
After the install has completed you will see a MongoDB leaf icon in the left menu - click on this icon to enter the MongoDB extension space installed above

Step 2 Make a connection to the MongoDB server (fit-db-01.rep.monash.edu)

Click on Add Connection



Select Advanced Connection Settings - Open Form



Complete the relevant form details:

In the General Tab,

Connection Type - select Standalone

Hostname: enter fit-db-01.rep.monash.edu

leave port at 27017

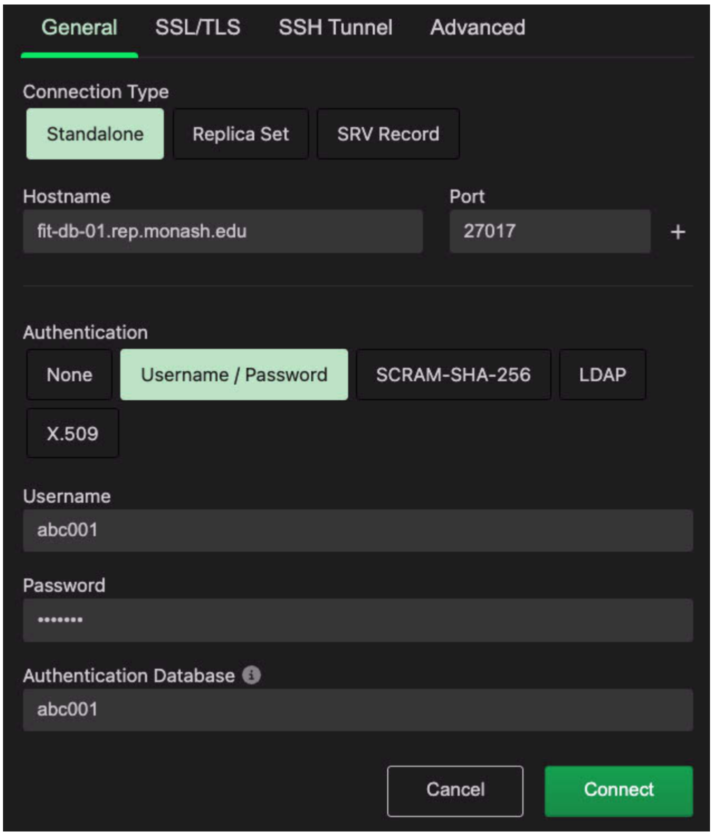
Authentication select Username/Password

Username is your authcate username

Password is: student

Authentication Database is your authcate username

DO NOT select CONNECT



Connection Type: Standalone

Hostname: fit-db-01.rep.monash.edu

Port: 27017

Authentication: Username / Password

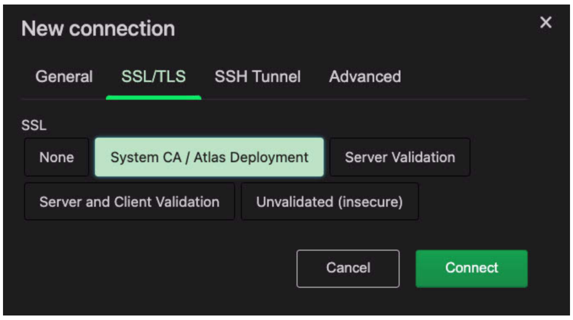
Username: abc001

Password: •••••••

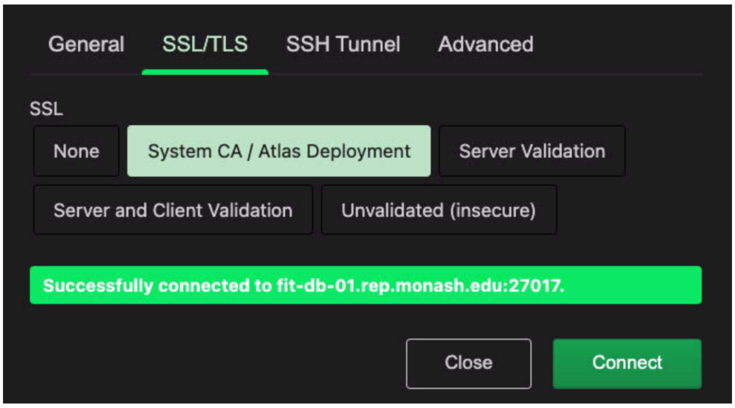
Authentication Database: abc001

Select SSL/TLS Tab

Under SSL Select SYSTEM CA / Atlas Deployment



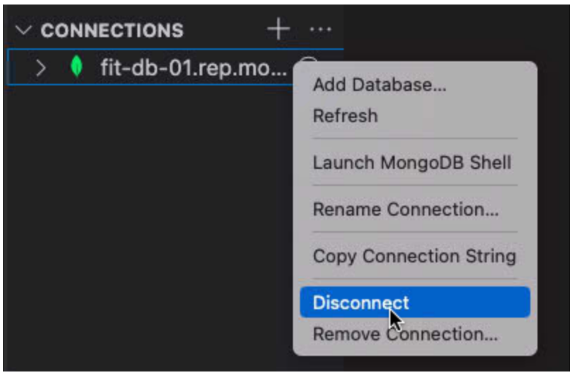
Then select the Connect button, you should then see the following message:



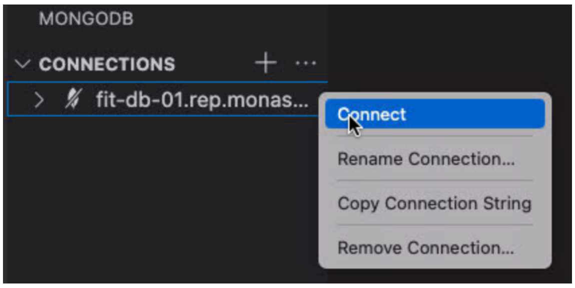
Successfully connected to fit-db-01.rep.monash.edu:27017.

You can then close this window - your connection is now visible in the mongoDB extension under CONNECTIONS:

To disconnect - right click and select Disconnect



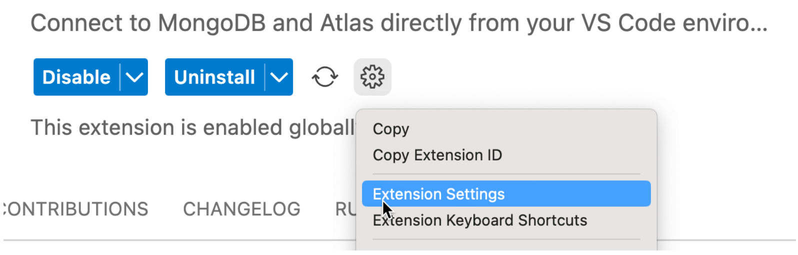
To reconnect to MongoDB, right click and select Connect



When connected the icon will be green.

Step 3 Suggested Configuration

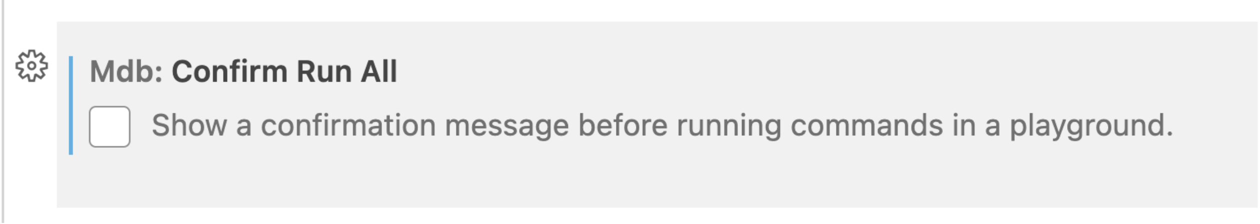
Select the MongoDB extension in Extensions (as in step 1 above), then select the Extension Settings:



Connect to MongoDB and Atlas directly from your VS Code enviro...

This extension is enabled globall

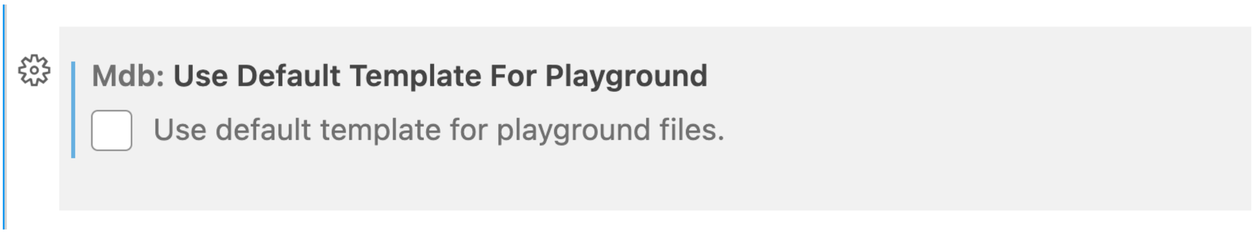
It is suggested you deselect the following two items:



Mdb: Confirm Run All

☐ Show a confirmation message before running commands in a playground.

and



Mdb: Use Default Template For Playground

☐ Use default template for playground files.