FIT2094-FIT3171 Databases

Week 12 Applied Class Activities

Big Data and NO SQL

FIT Database Teaching Team

Complete the following activities listed below

FIT2094-FIT3171 2022 S1

FIT2094-FIT3171 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 - Session Demonstration

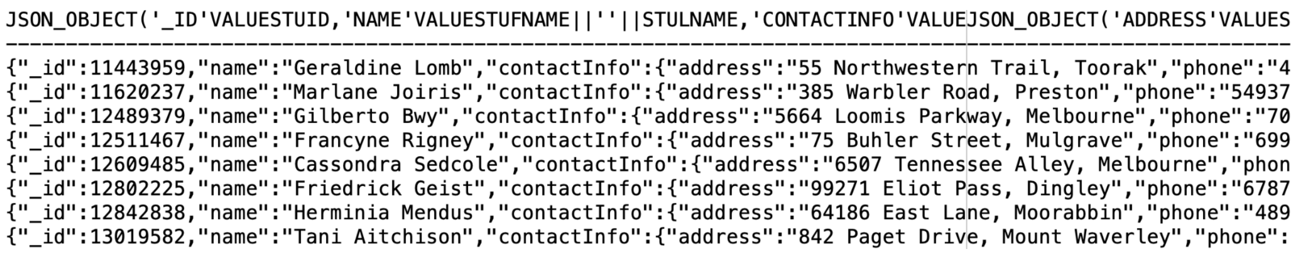
In this applied session we will focus on MongoDB, one of the popular Big Data platforms. The data in MongoDB must be stored in a specific format, for example:

```
{
    "_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;
```

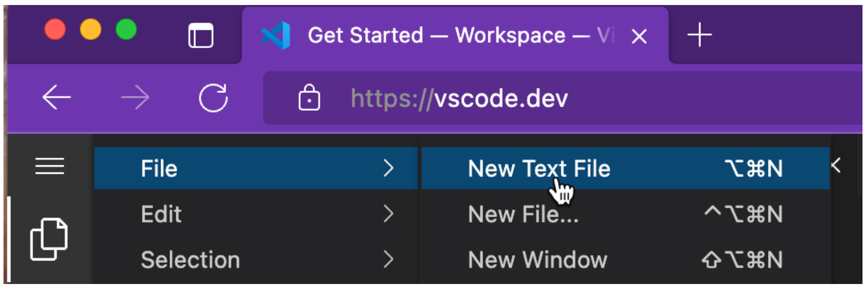
This will produce output of the form:



```
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.

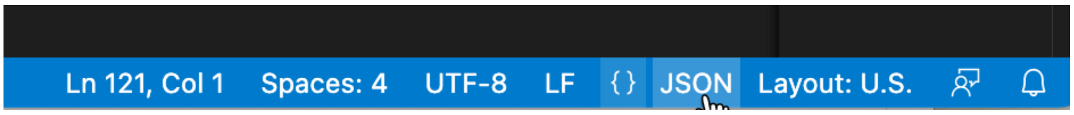
Open Microsoft Visual Studio Code for the Web, select the three vertical bars (Application Menu) in the top left and then File - New Text File:



Copy **the first 10 lines of the generated SQL output** into Microsoft Visual Studio Code.

Remove the comma symbol on the right hand end of the last line.

On the right bottom corner **change** the document type from **plain text to json (if required):**



Then right click on your text and select **Format Document**. Save the file via the Application Menu, as demo_enrolment.json in your local repo.

Open the MongoDB shell https://docs.mongodb.com/manual/tutorial/getting-started/, then click the working window to connect to the database.

Alternatively, you can install your own local copy of MongoDB by following the steps listed at: https://docs.mongodb.com/manual/administration/install-community/

This is an **OPTIONAL installation**, it is not required as part of this unit - you can complete all of your required activities via the online MongoDB shell at the URL above.

Note **for MacOS** the install of the community edition occurs via brew and requires:

- A 64-bit Intel CPU or Apple Silicon CPU
- macOS Mojave (10.14) (or higher)
- Command Line Tools (CLT) for Xcode:
    - xcode-select --install
    - or install the full Xcode via the App Store (note this is a large download (11Gb) and will take some time)
- A Bourne-compatible shell for installation (e.g. bash or zsh)

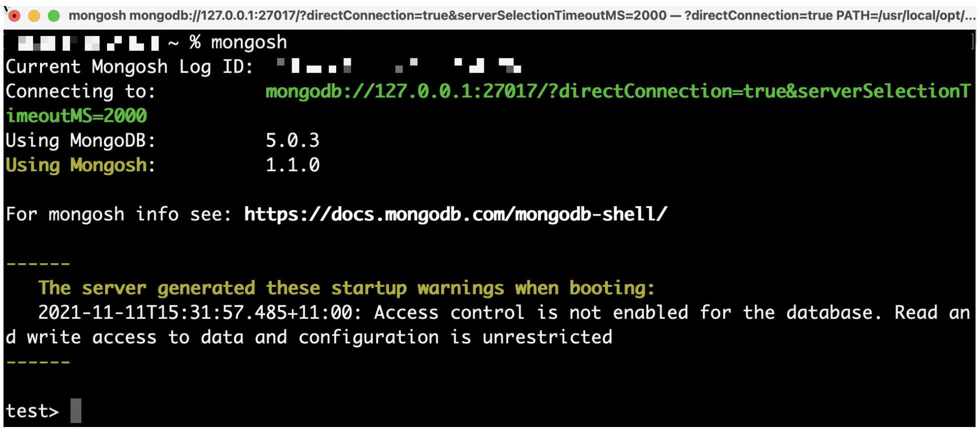
Once installation is complete you will access MongoDB from the terminal app by typing:

```
brew services start mongodb-community
```

to start the monogdb service (database) followed by:

```
mongosh
```

to connect to mongo


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000 — ?directConnection=true PATH=/usr/local/opt/...

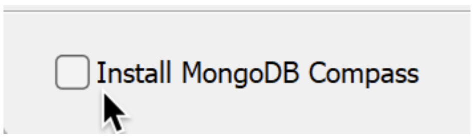
```
[redacted] ~ % mongosh
Current Mongosh Log ID: [redacted]
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionT
imeoutMS=2000
Using MongoDB:          5.0.3
Using Mongosh:          1.1.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting:
   2021-11-11T15:31:57.485+11:00: Access control is not enabled for the database. Read an
d write access to data and configuration is unrestricted
------

test>
```

Once you have completed your work and closed the mongo shell (mongosh) via exit type, in the terminal:

```
brew services stop mongodb-community
```

to stop the mongodb service.

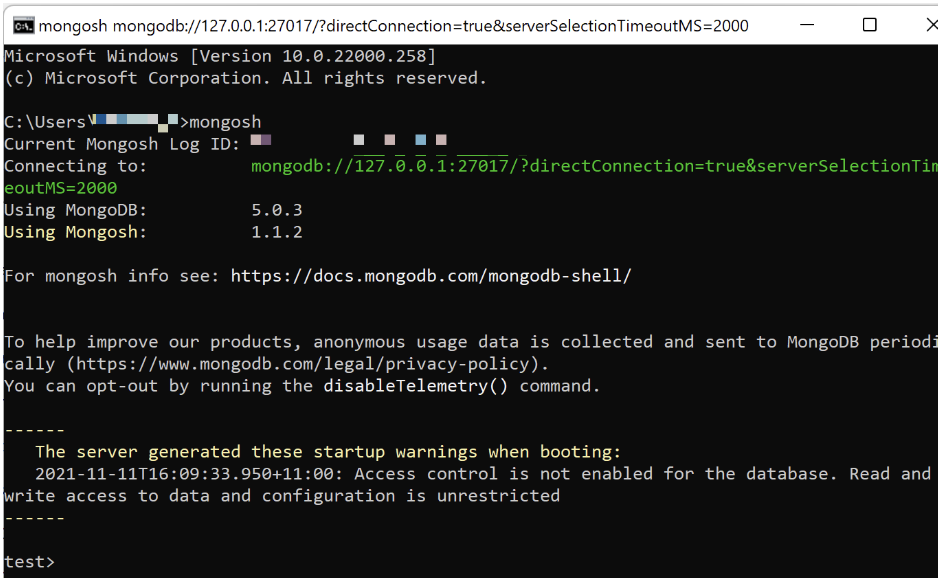
For **MS Windows** you need to download, and run the msi installer to set up MongoDB. We suggest you install the DB as a Windows service. Note you ***do not*** need to install Compass. Please uncheck the option:



during the installation.

After mongoDB has been installed you will then need to then install the Mongo DB Shell (mongosh), so that you can access the database.

Once installation is complete you will access MongoDB from a command window by typing: `mongosh`


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Microsoft Windows [Version 10.0.22000.258]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>mongosh
Current Mongosh Log ID: [redacted]
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTim
eoutMS=2000
Using MongoDB:          5.0.3
Using Mongosh:          1.1.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodi
cally (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting:
   2021-11-11T16:09:33.950+11:00: Access control is not enabled for the database. Read and
write access to data and configuration is unrestricted
------

test>
```

After you have connected to MongoDB via the web shell (or locally), type

```
use results
```

and hit enter. You are now using the results database on MongoDB.

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

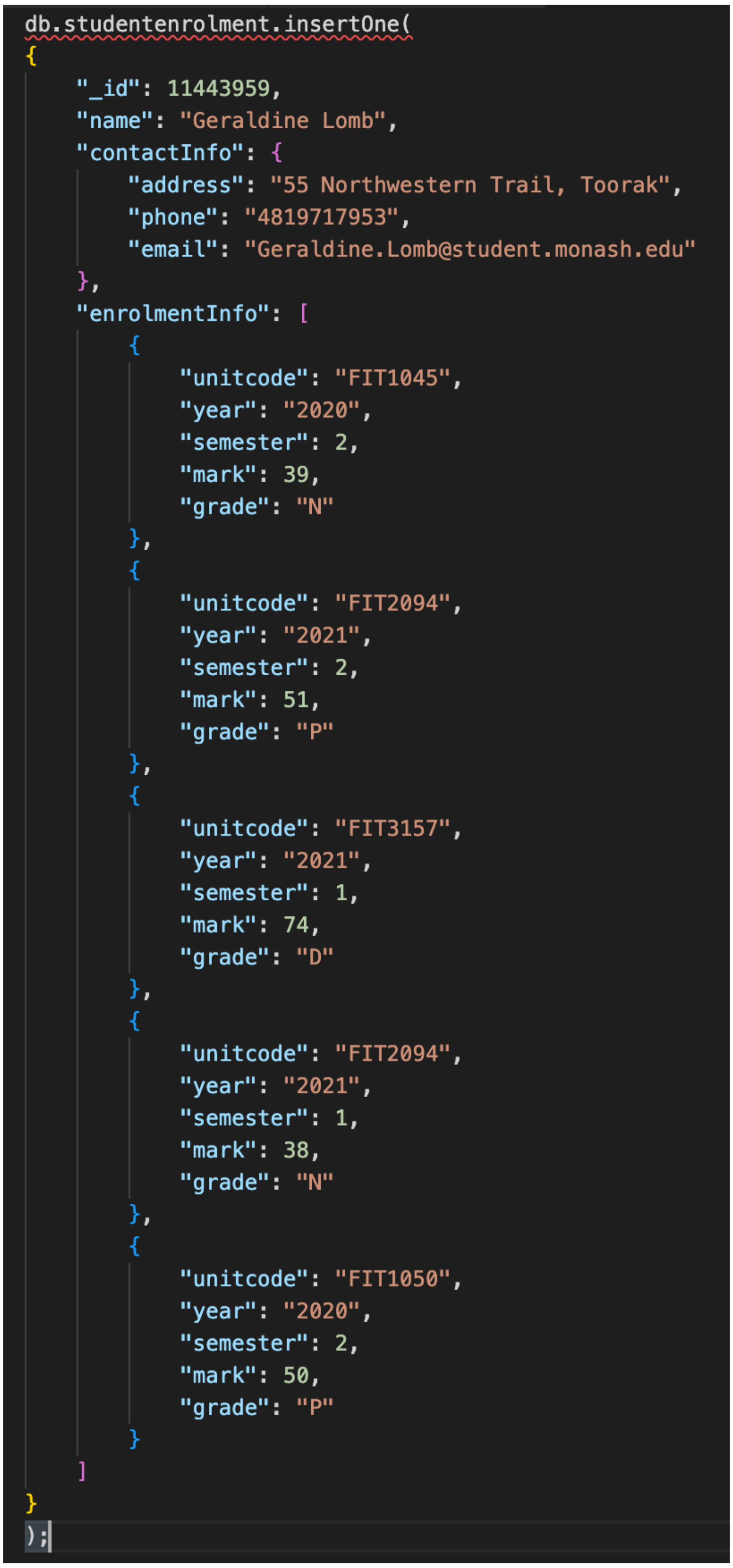
```
show collections
```

and hit enter.

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

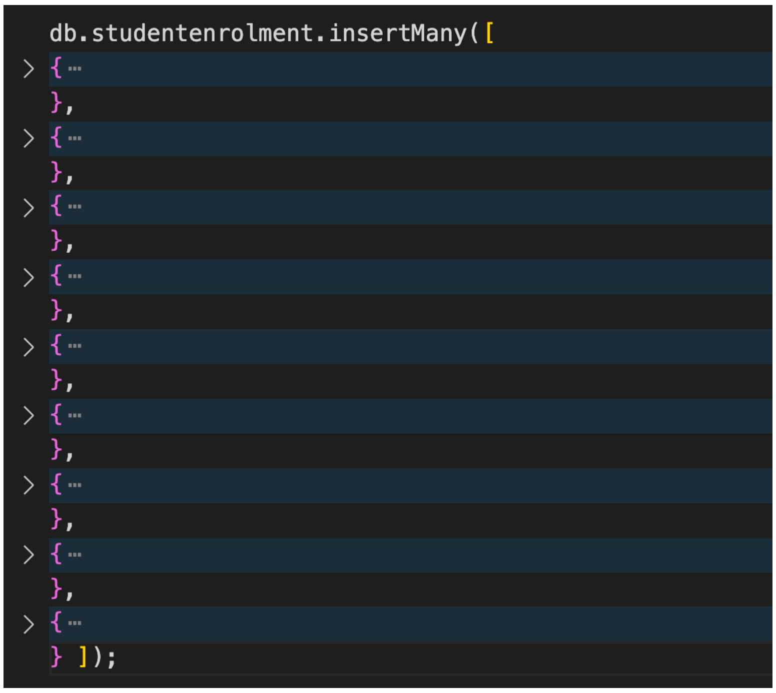
Now, let’s create, read, update and delete (CRUD) documents on MongoDB

1. Create a collection by inserting one document into the collection. For this activity, we will call the collection studentenrolment



```
db.studentenrolment.insertOne(
{
    "_id": 11443959,
    "name": "Geraldine Lomb",
    "contactInfo": {
        "address": "55 Northwestern Trail, Toorak",
        "phone": "4819717953",
        "email": "Geraldine.Lomb@student.monash.edu"
    },
    "enrolmentInfo": [
        {
            "unitcode": "FIT1045",
            "year": "2020",
            "semester": 2,
            "mark": 39,
            "grade": "N"
        },
        {
            "unitcode": "FIT2094",
            "year": "2021",
            "semester": 2,
            "mark": 51,
            "grade": "P"
        },
        {
            "unitcode": "FIT3157",
            "year": "2021",
            "semester": 1,
            "mark": 74,
            "grade": "D"
        },
        {
            "unitcode": "FIT2094",
            "year": "2021",
            "semester": 1,
            "mark": 38,
            "grade": "N"
        },
        {
            "unitcode": "FIT1050",
            "year": "2020",
            "semester": 2,
            "mark": 50,
            "grade": "P"
        }
    ]
}
);
```

2. Insert the remaining 9 documents into the collection using (documents shown collapsed). Note that since we are now inserting many documents they must be enclosed in an array [...]



```
db.studentenrolment.insertMany([
{…
},
{…
},
{…
},
{…
},
{…
},
{…
},
{…
},
{…
},
{…
} ]);
```

3. Read data

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

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

   or for the latest version of mongoDB

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

   b. Read all documents

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

   c. Show the data for student id = 13119134

   ```
   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.find().count();

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

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

12.1.2 MongoDB Create Update and Delete

First, download the week12_bigdata.txt file from Moodle, place this file in your working directory in your App12 folder. Write the necessary SQL statements and MongoDB scripts for **12.1.2 and 12.1.3** questions in that file.

1. Use 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"
      }
    ]
  }
```

2. Name the collection as **enrolment** (you may use any suitable name for your database) and insert the first 10 documents generated by the select statement into MongoDB (*if it returns an error, try to add a maximum 5 documents at any one time*).
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 enrolment, 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

Write db.find() commands for 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 SETU 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 FIT2094-FIT3171 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