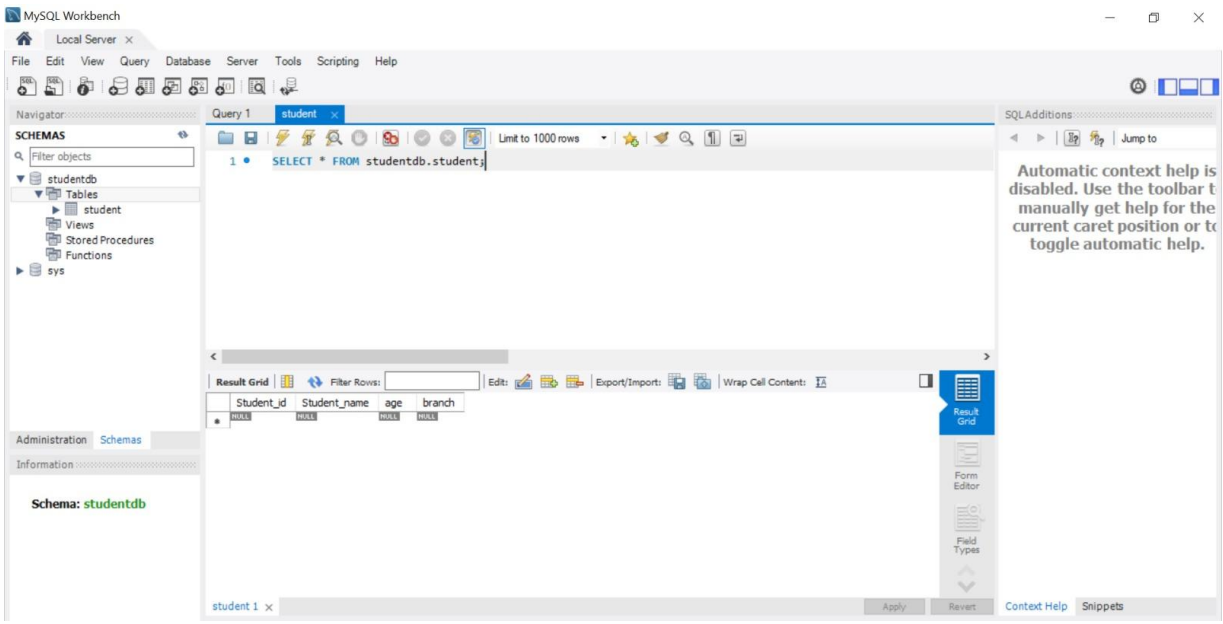
MODERN APPLICATION DEVELOPMENT (JAVA SPRING BOOT)
ASSIGNMENT-2

Arunima Mohanan
20BCE0357
VIT Vellore

1) Create, update and delete commands in mysql

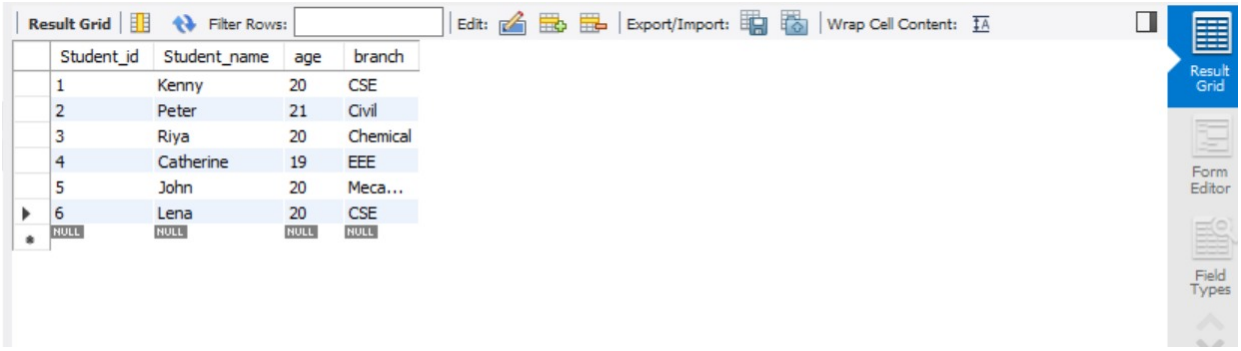
- CREATE table statement:

```
CREATE DATABASE `studentdb`;

CREATE TABLE `studentdb`.`student` (
  `Student_id` INT NOT NULL,
  `Student_name` VARCHAR(45) NOT NULL,
  `age` INT NOT NULL,
  `branch` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Student_id`));

SELECT * FROM studentdb.student;
```



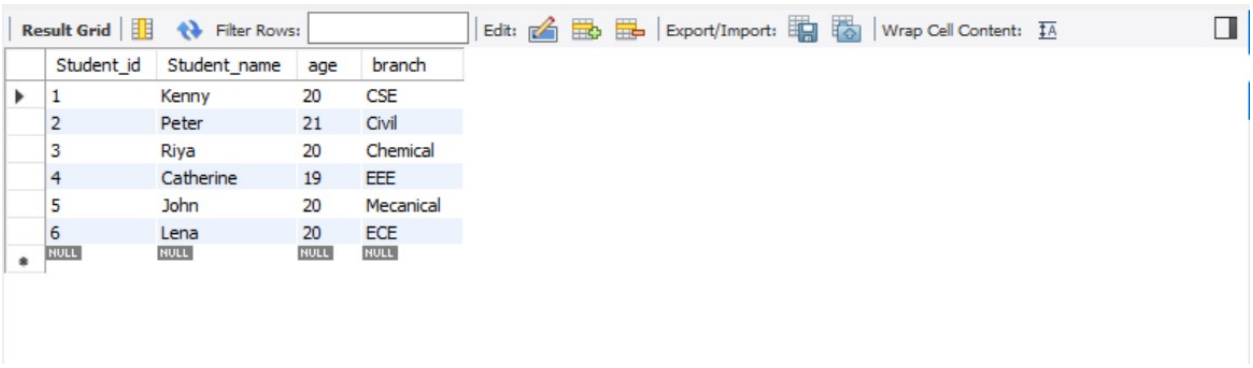
- Insert statements:

```
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('1', 'Kenny', '20', 'CSE');
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('2', 'Peter', '21', 'Civil');
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('3', 'Riya', '20', 'Chemical');
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('4', 'Catherine', '19', 'EEE');
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('5', 'John', '20', 'Mechanical');
INSERT INTO `studentdb`.`student` (`Student_id`, `Student_name`, `age`, `branch`)
VALUES ('6', 'Lena', '20', 'CSE');

SELECT * FROM studentdb.student;
```



- Update statements:

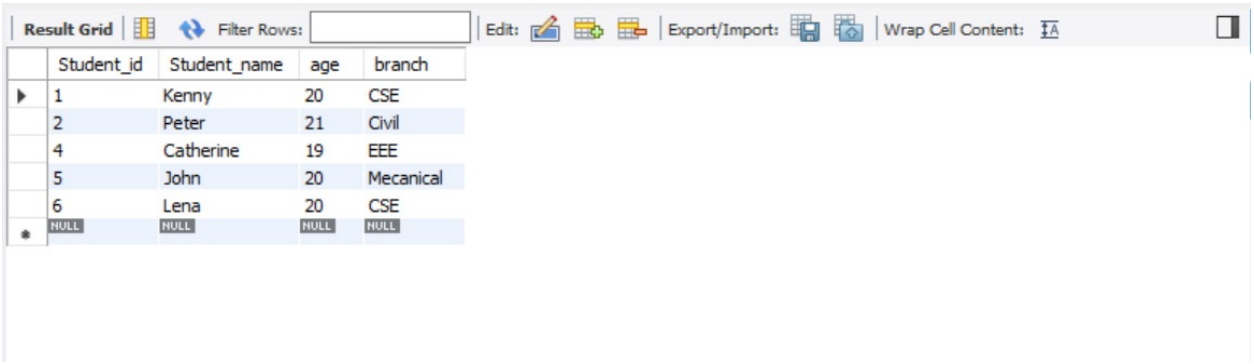
```
UPDATE  `studentdb`.`student` SET branch='ECE' WHERE Student_id='6';
SELECT* FROM `studentdb`.`student`;
```



Student_id	Student_name	age	branch
1	Kenny	20	CSE
2	Peter	21	Civil
3	Riya	20	Chemical
4	Catherine	19	EEE
5	John	20	Mecanical
6	Lena	20	ECE
NULL	NULL	NULL	NULL

- Delete statement:

```
DELETE FROM `studentdb`.`student` WHERE (`Student_id` = '3');
SELECT* FROM `studentdb`.`student`;
```



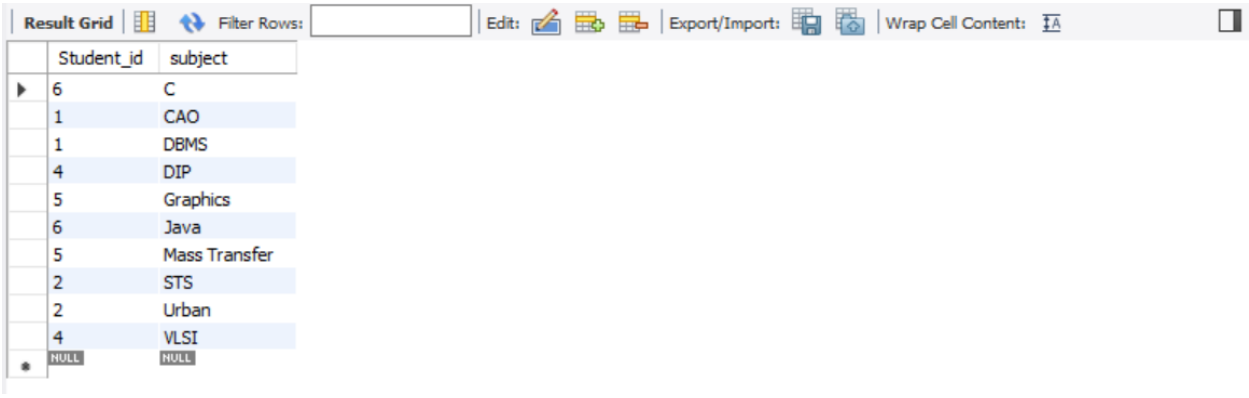
Student_id	Student_name	age	branch
1	Kenny	20	CSE
2	Peter	21	Civil
4	Catherine	19	EEE
5	John	20	Mecanical
6	Lena	20	CSE
NULL	NULL	NULL	NULL

2) Create tables and perform joins in mysql

- Table created:

```
CREATE TABLE `studentdb`.`subjects` (
`Student_id` INT NOT NULL,
`subject` VARCHAR(45) NOT NULL,
PRIMARY KEY (`Student_id`),
CONSTRAINT `Student_id`
FOREIGN KEY (`Student_id`)
REFERENCES `studentdb`.`student` (`Student_id`));
```

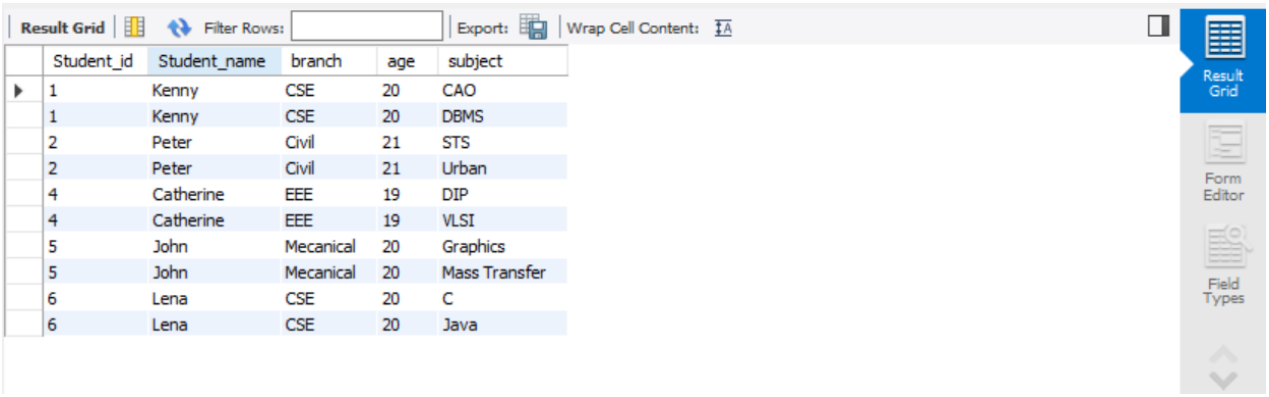
```
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('1', 'DBMS');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('1', 'CAO');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('2', 'Urban');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('2', 'STS');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('4', 'VLSI');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('4', 'DIP');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('5', 'Mass
Transfer');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('5', 'Graphics');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('6', 'C');
INSERT INTO `studentdb`.`subjects` (`Student_id`, `subject`) VALUES ('6', 'Java');

SELECT * FROM studentdb.subjects;
```



Student_id	subject
6	C
1	CAO
1	DBMS
4	DIP
5	Graphics
6	Java
5	Mass Transfer
2	STS
2	Urban
4	VLSI
NULL	NULL

- Join statement:

```
SELECT sd.Student_id,
sd.Student_name,
sd.branch,
sd.age,
s.subject
FROM studentdb.student AS sd
LEFT JOIN studentdb.subjects AS s ON sd.Student_id=s.Student_id;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student_id	Student_name	branch	age	subject
1	Kenny	CSE	20	CAO
1	Kenny	CSE	20	DBMS
2	Peter	Civil	21	STS
2	Peter	Civil	21	Urban
4	Catherine	EEE	19	DIP
4	Catherine	EEE	19	VLSI
5	John	Mecanical	20	Graphics
5	John	Mecanical	20	Mass Transfer
6	Lena	CSE	20	C
6	Lena	CSE	20	Java

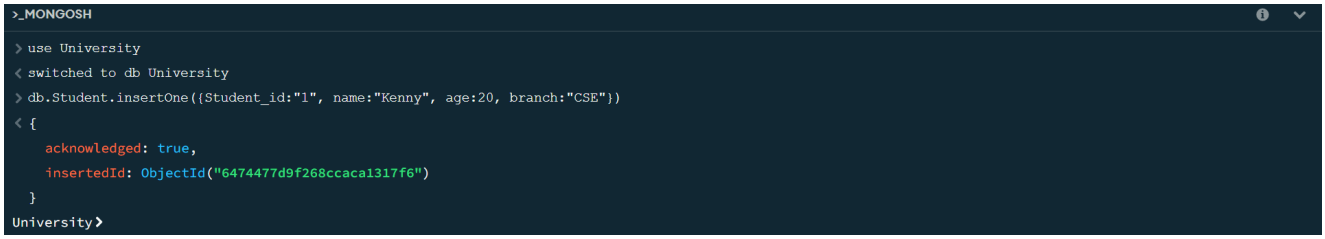
Result Grid | Form Editor | Field Types

3) Create, update and delete commands in MongoDB

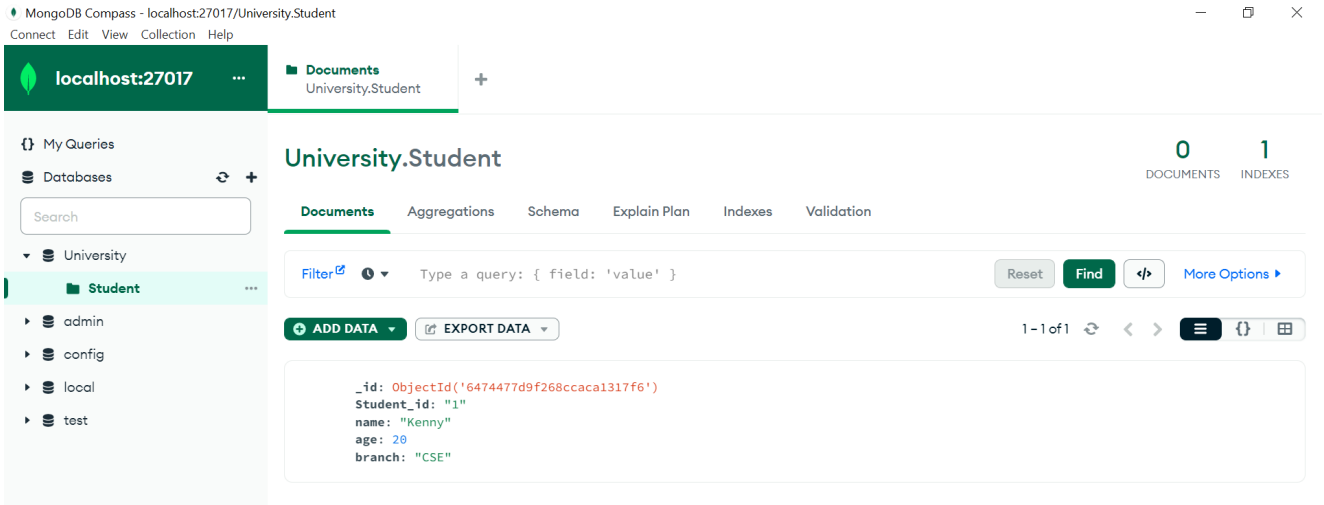
- **Insert commands**

Inserting single record:

db.Student.insertOne({Student_id:"1", name:"Kenny", age:20, branch:"CSE"})



```
>_MONGOSH
> use University
< switched to db University
> db.Student.insertOne({Student_id:"1", name:"Kenny", age:20, branch:"CSE"})
< {
    acknowledged: true,
    insertedId: ObjectId("6474477d9f268ccaca1317f6")
  }
University>
```



MongoDB Compass - localhost:27017/University.Student

Connect Edit View Collection Help

localhost:27017

Documents
University.Student

My Queries
Databases
Search
University
Student
admin
config
local
test

University.Student

0 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

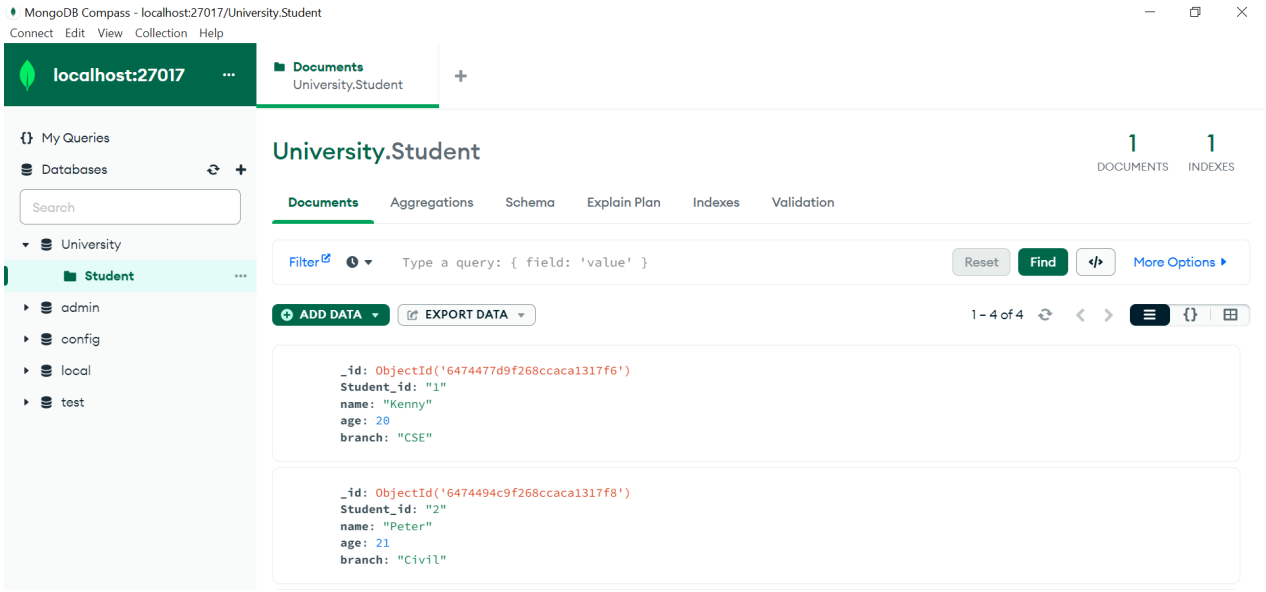
Filter | Type a query: { field: 'value' } | Reset | Find | More Options

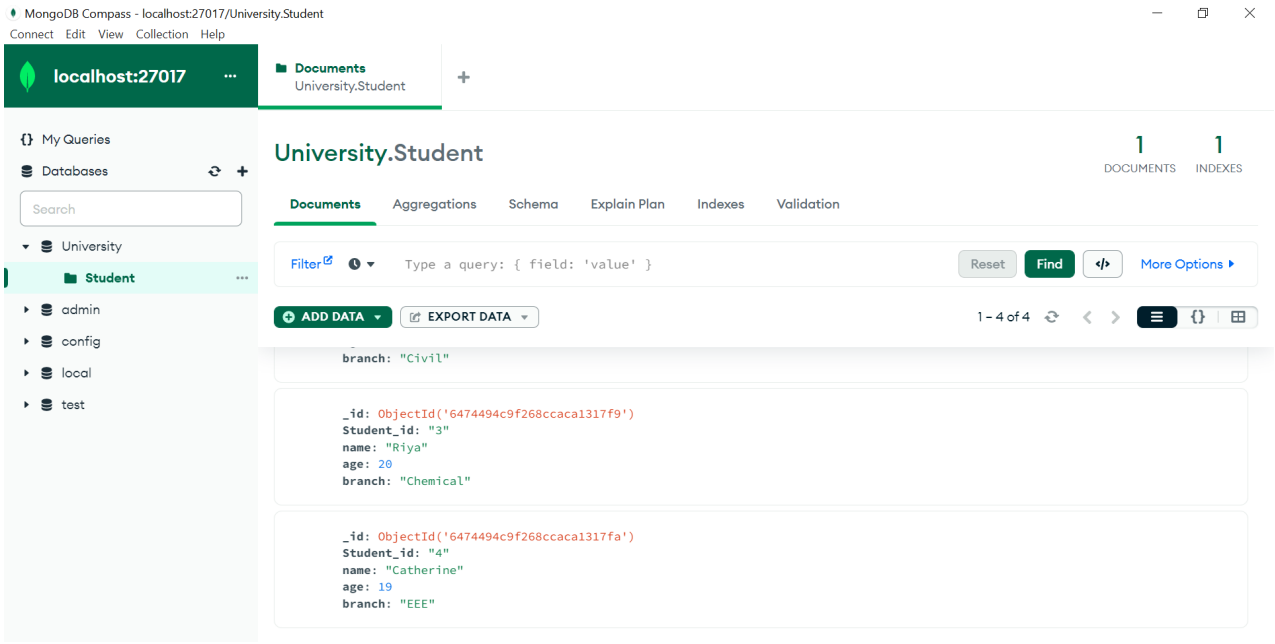
ADD DATA | EXPORT DATA | 1-1 of 1

```
_id: ObjectId('6474477d9f268ccaca1317f6')
Student_id: "1"
name: "Kenny"
age: 20
branch: "CSE"
```

Inserting multiple records:

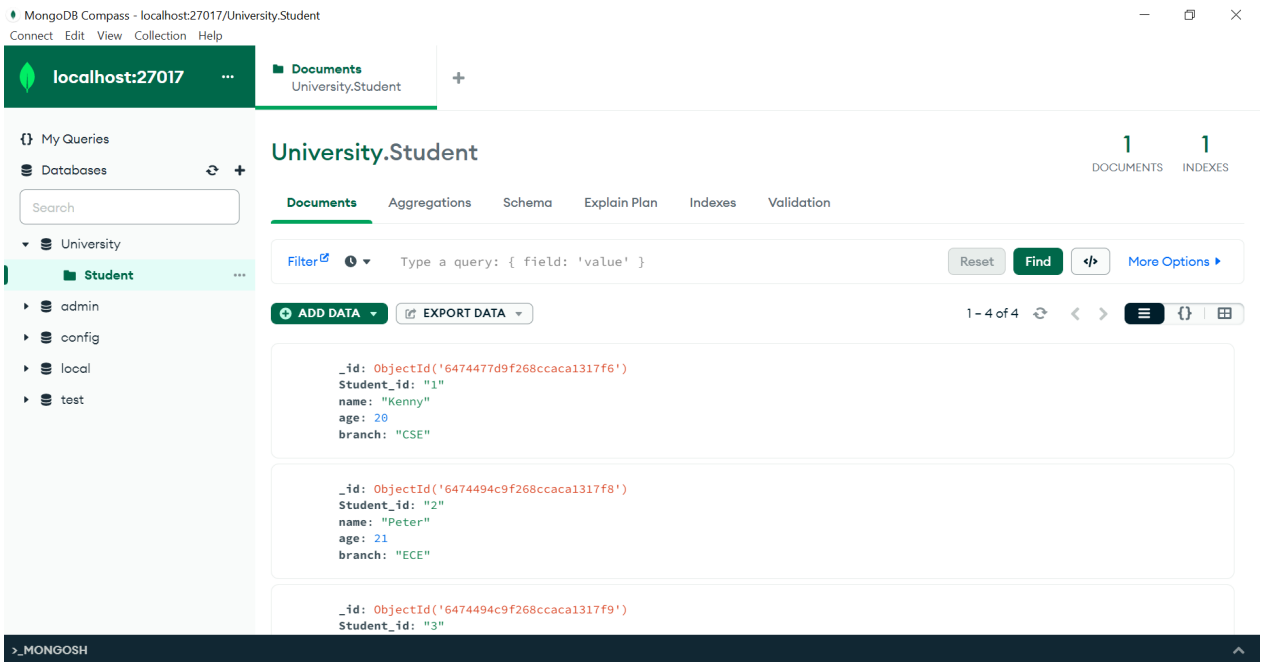
db.Student.insertMany([{Student_id:"2", name:"Peter", age:21, branch:"Civil"}, {Student_id:"3", name:"Riya", age:20, branch:"Chemical"},{Student_id:"4", name:"Catherine", age:19, branch:"EEE"}])





- **Update commands**

db.Student.updateOne({Student_id: "2"}, {$set: {branch: "ECE"}})

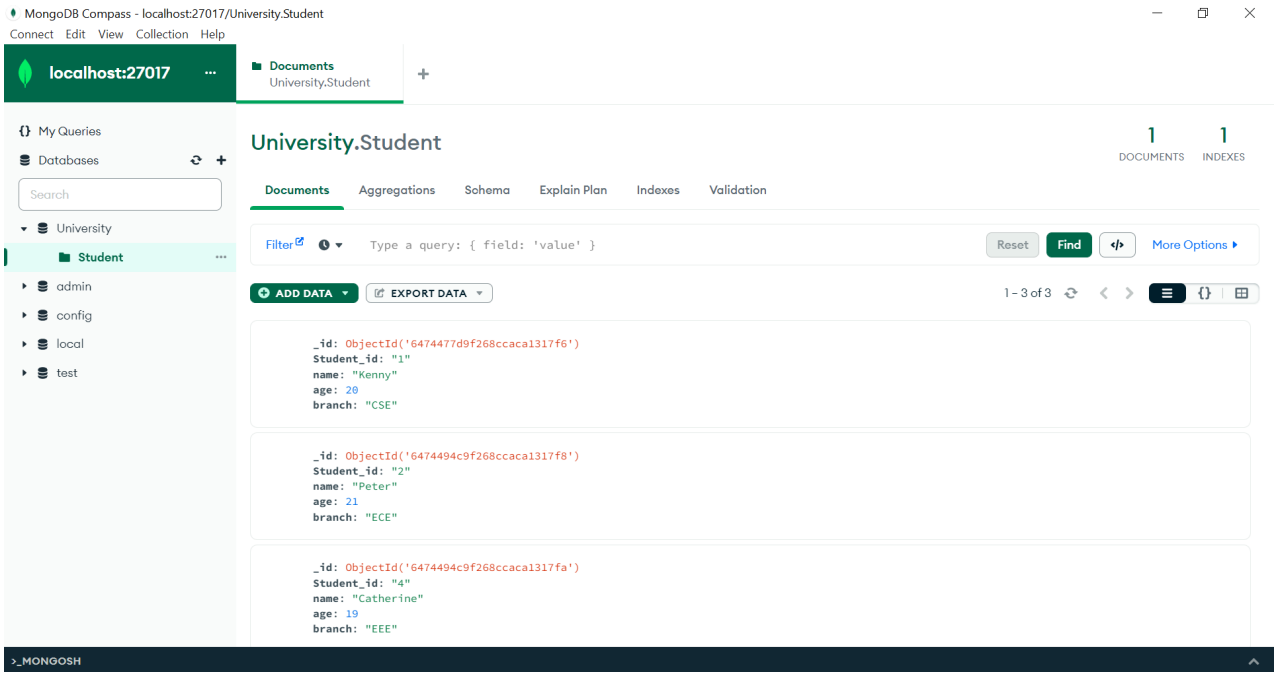


- **Delete commands**

Delete with condition:

db.Student.remove({name: "Riya"})

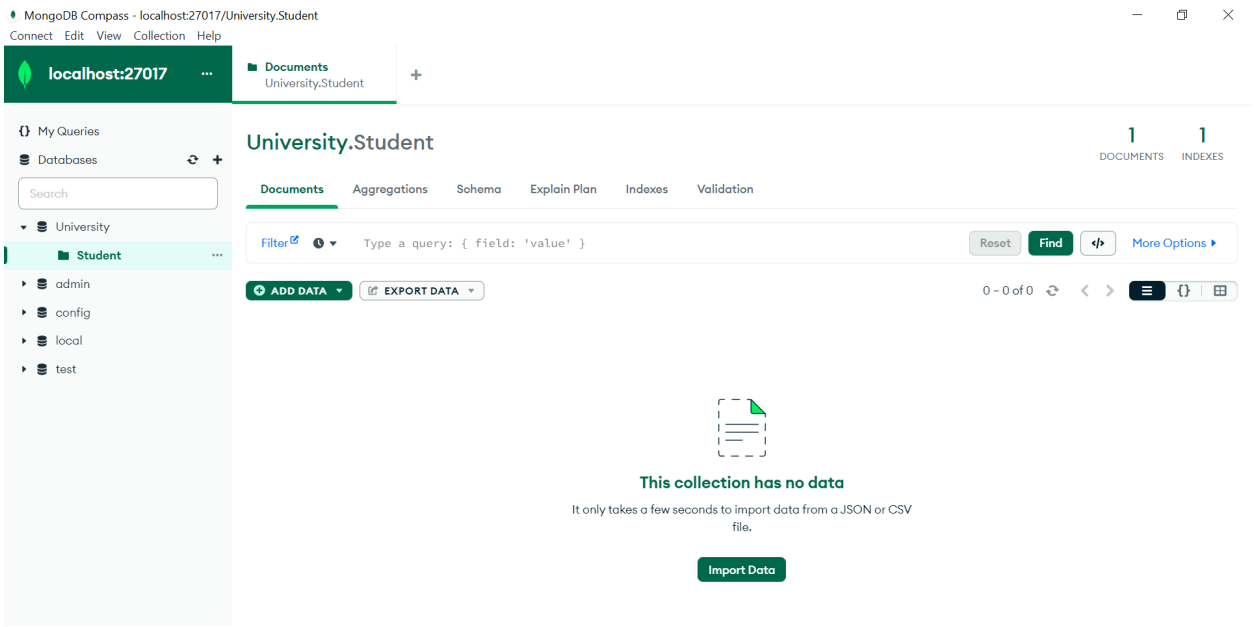
```
> db.Student.remove({name: "Riya"})
< DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
< {
    acknowledged: true,
    deletedCount: 1
  }
University>
```



Delete all records:

db.Student.remove({})

```
> db.Student.remove({})
< {
    acknowledged: true,
    deletedCount: 3
  }
University>
```


MongoDB Compass - localhost:27017/University.Student
Connect Edit View Collection Help
localhost:27017
Documents
University.Student
My Queries
Databases
Search
University
Student
admin
config
local
test
University.Student
1
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
Reset
Find
More Options
ADD DATA
EXPORT DATA
0 - 0 of 0
This collection has no data
It only takes a few seconds to import data from a JSON or CSV file.
Import Data