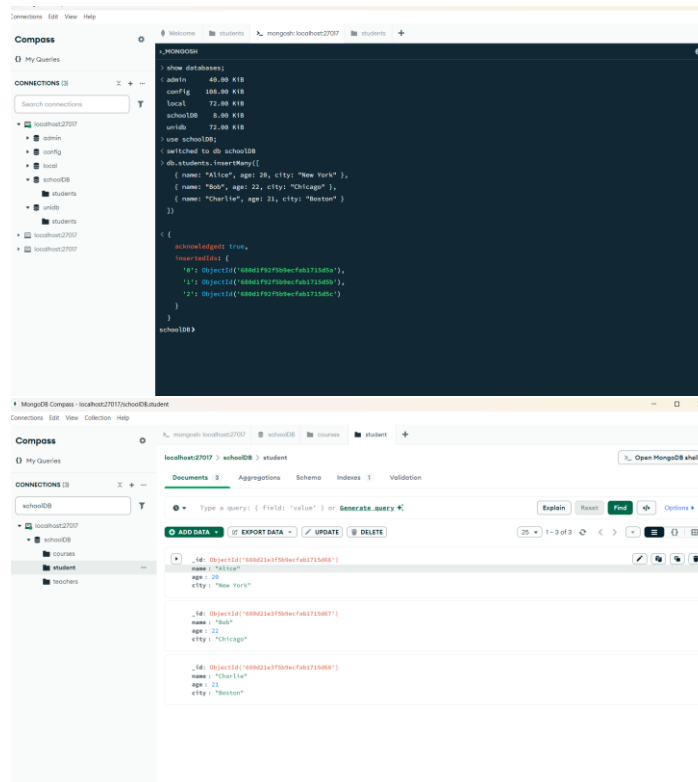


IT2234 (p) Web Service and Server Technologies.
University of Vavuniya
Faculty of Applied science
Department of physical science
2021/ICT/30

1. Create the database schoolDB
2. Create collections students, courses and teachers. Insert documents

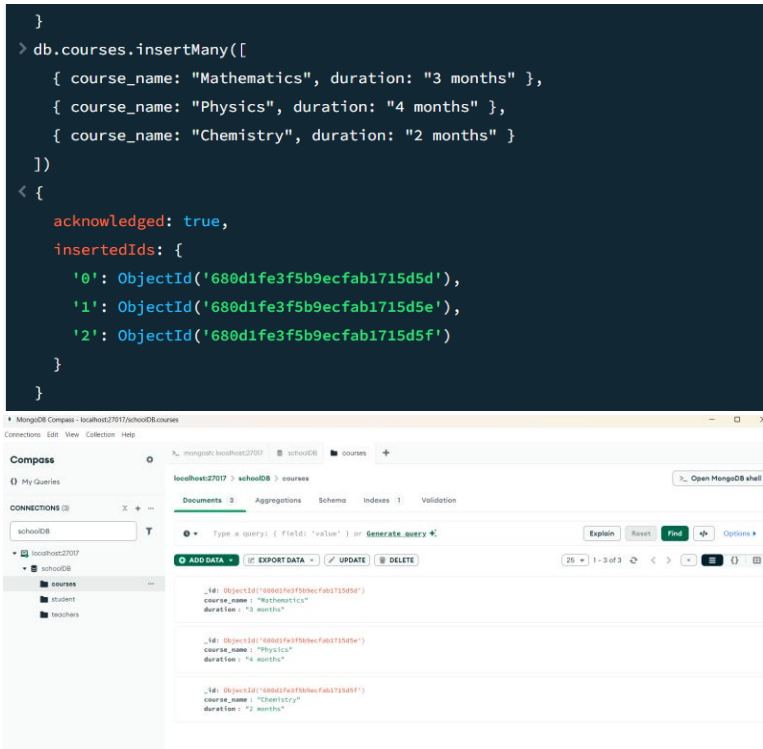
Students collection

```
db.students.insertMany([
  { name: "Alice", age: 20, city: "New York" },
  { name: "Bob", age: 22, city: "Chicago" },
  { name: "Charlie", age: 21, city: "Boston" }
])
```



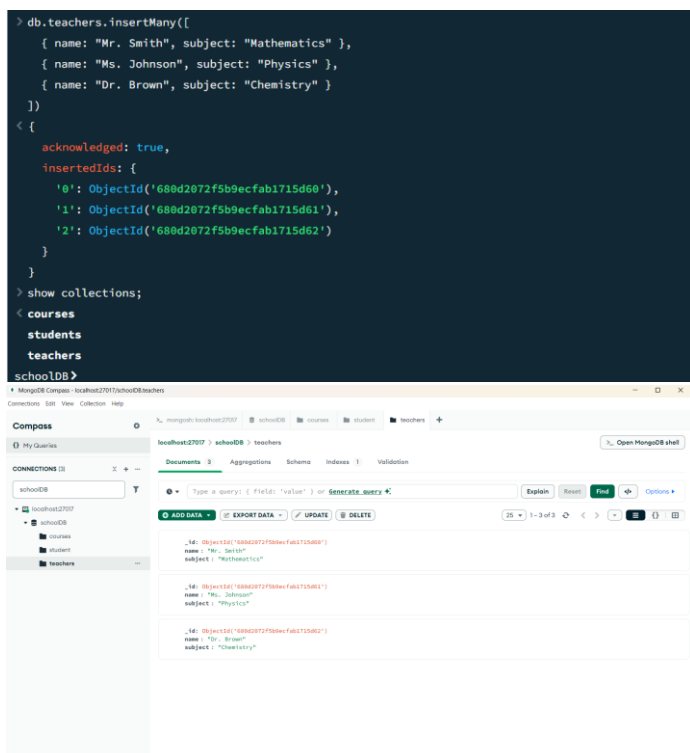
Courses collection

```
db.courses.insertMany([
  { course_name: "Mathematics", duration: "3 months" },
  { course_name: "Physics", duration: "4 months" },
  { course_name: "Chemistry", duration: "2 months" }
])
```



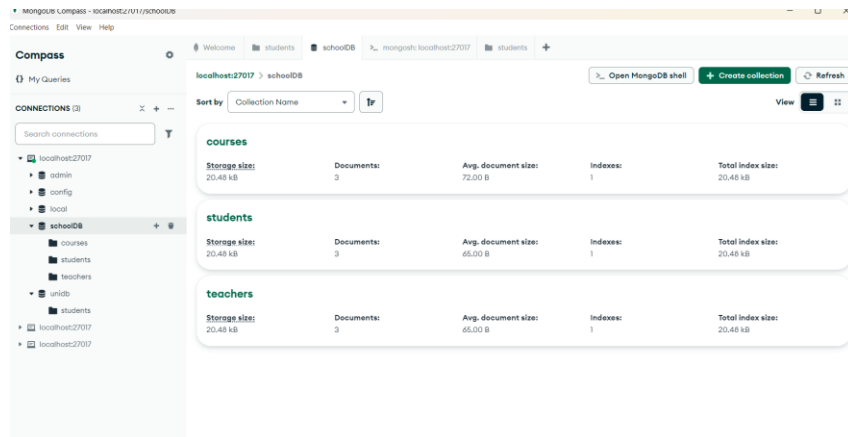
Teachers collections

```
db.teachers.insertMany([
  { name: "Mr. Smith", subject: "Mathematics" },
  { name: "Ms. Johnson", subject: "Physics" },
  { name: "Dr. Brown", subject: "Chemistry" }
])
```



Questions.

i) Show all students



ii) Find students who live in "Chicago"

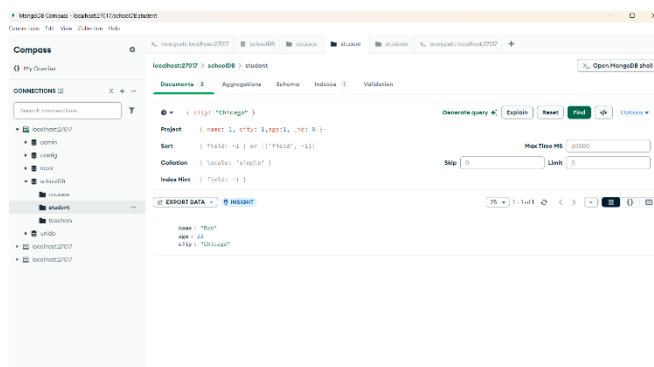
`db.student.find({ city: "Chicago" })`

```
> db.student.find({ city: "Chicago" })
< {
  _id: ObjectId('680d21e3f5b9ecfab1715d67'),
  name: 'Bob',
  age: 22,
  city: 'Chicago'
}
schoolDB>
```

Box
Filter
Project

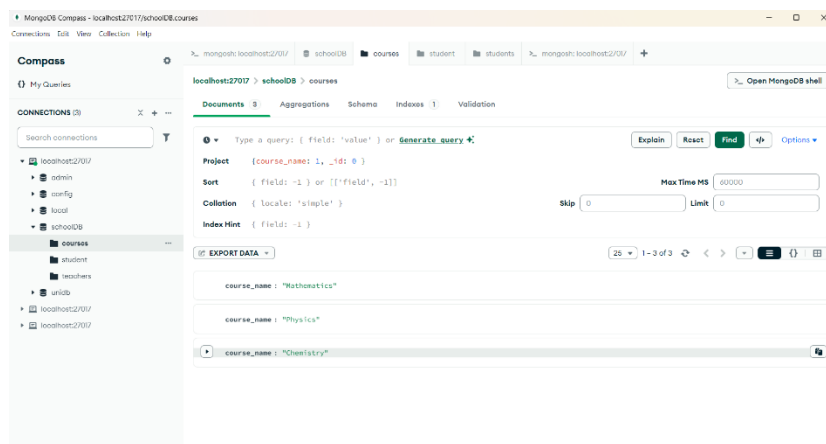
Code
`{ city: "Chicago" }`
`{ name: 1, city: 1, age: 1, _id: 0 }`

- **Filter** = "which documents you want"
- **Project** = "which fields you want to show"



- iii) List all course names
`db.courses.find({}, { course_name: 1, _id: 0 })`

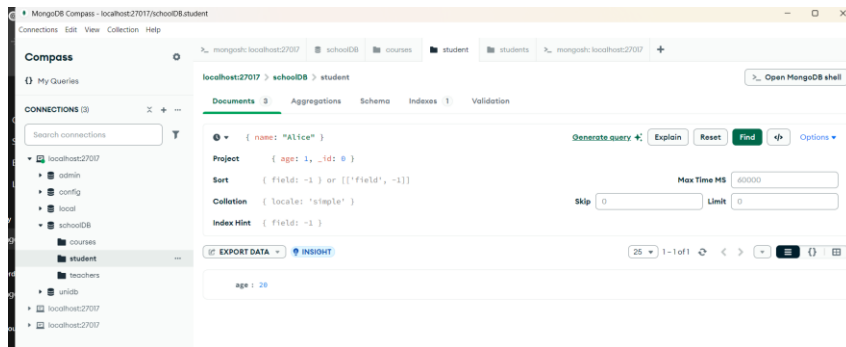
```
> db.courses.find({}, { course_name: 1, _id: 0 })
< {
  course_name: 'Mathematics'
}
{
  course_name: 'Physics'
}
{
  course_name: 'Chemistry'
}
schoolDB> |
```



- iv) Find the age of Alice
`db.students.find({ name: "Alice" }, { age: 1, _id: 0 })`

```
> db.courses.find({}, { course_name: 1, _id: 0 })
< {
  course_name: 'Mathematics'
}
{
  course_name: 'Physics'
}
{
  course_name: 'Chemistry'
}
> db.students.find({ name: "Alice" }, { age: 1, _id: 0 })
<
> db.student.find({ name: "Alice" }, { age: 1, _id: 0 })
< {
  age: 20
}
```

Box	What to write
Filter	<code>{ name: "Alice" }</code>
Project	<code>{ age: 1, _id: 0 }</code>



v) Update Bob's city to "San Francisco"

```
db.students.updateOne(
  { name: "Bob" },
  { $set: { city: "San Francisco" } }
)
```

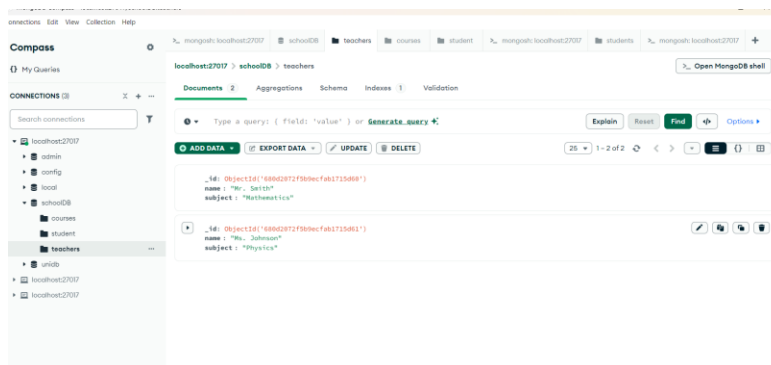
```
> db.students.updateOne(
  { name: "Bob" },
  { $set: { city: "San Francisco" } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

vi) Delete the teacher "Dr. Brown".

```
db.teachers.deleteOne({ name: "Dr. Brown" })
```

```
> db.teachers.deleteOne({ name: "Dr. Brown" })
< {
  acknowledged: true,
  deletedCount: 1
}
```

Refresh



- vii) Find teachers who teach "Physics"

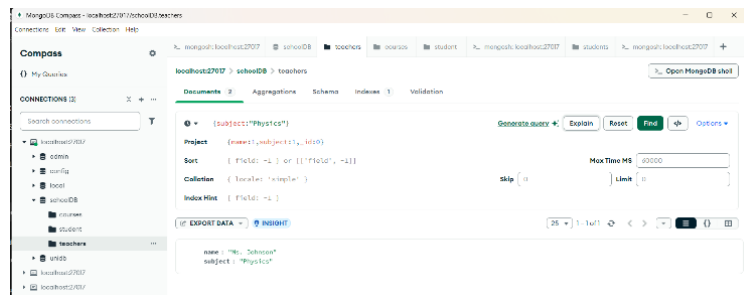
```
db.teachers.find({ subject: "Physics" })
```

```
}
> db.teachers.find({ subject: "Physics" })
< {
  _id: ObjectId('680d2072f5b9ecfab1715d61'),
  name: 'Ms. Johnson',
  subject: 'Physics'
}
```

schoolDB>

Filter - {subject:"Physics"}

Project - {name:1,subject:1,_id:0}



- viii) Add a new student named "David" aged 23 from "Miami"

```
db.students.insertOne({ name: "David", age: 23, city: "Miami" })
```

```
> db.student.insertOne({ name: "David", age: 23, city: "Miami" })
< {
  acknowledged: true,
  insertedId: ObjectId('680de13b971813dca281b137')
}
```

schoolDB>

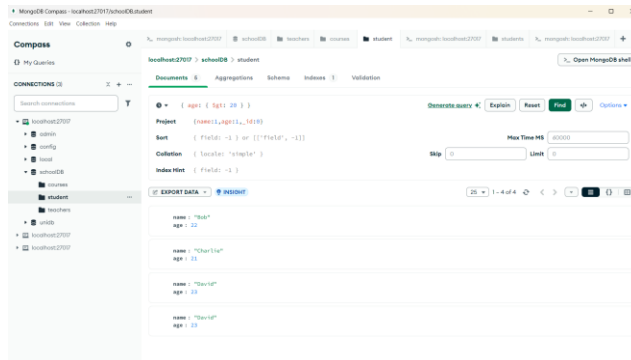
- ix) Find students whose age is greater than 20

```
db.students.find({ age: { $gt: 20 } })
```

```
> db.student.find({ age: { $gt: 20 } })
< {
  _id: ObjectId('680d21e3f5b9ecfab1715d67'),
  name: 'Bob',
  age: 22,
  city: 'Chicago'
},
{
  _id: ObjectId('680d21e3f5b9ecfab1715d68'),
  name: 'Charlie',
  age: 21,
  city: 'Boston'
},
{
  _id: ObjectId('680de13b971813dca281b137'),
  name: 'David',
  age: 23,
  city: 'Miami'
}
```

Filter - { age: { \$gt: 20 } }

Project - { name: 1, age: 1, _id: 0 }



x) Sort students by age ascending

`db.students.find().sort({ age: 1 })`

