Name : Haris **Roll # 23P-0573**

Mongo sh

Use schoolDB

```
C:\Users\User>mongosh
Current Mongosh Log ID: 681d91e72b4512fdcc0d818f
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.3
Using MongoDB:          8.0.9
Using Mongosh:          2.3.3
mongosh 2.5.1 is available for download: https://www.mongodb.com/try/download/shell

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

------
   The server generated these startup warnings when booting
   2025-05-09T10:14:43.248+05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin   40.00 KiB
config  60.00 KiB
local   72.00 KiB
test> use SchoolDB
```

2. Create two collections:
   Students
   Courses

   db.createCollection("Students")
   db.createCollection("Courses")

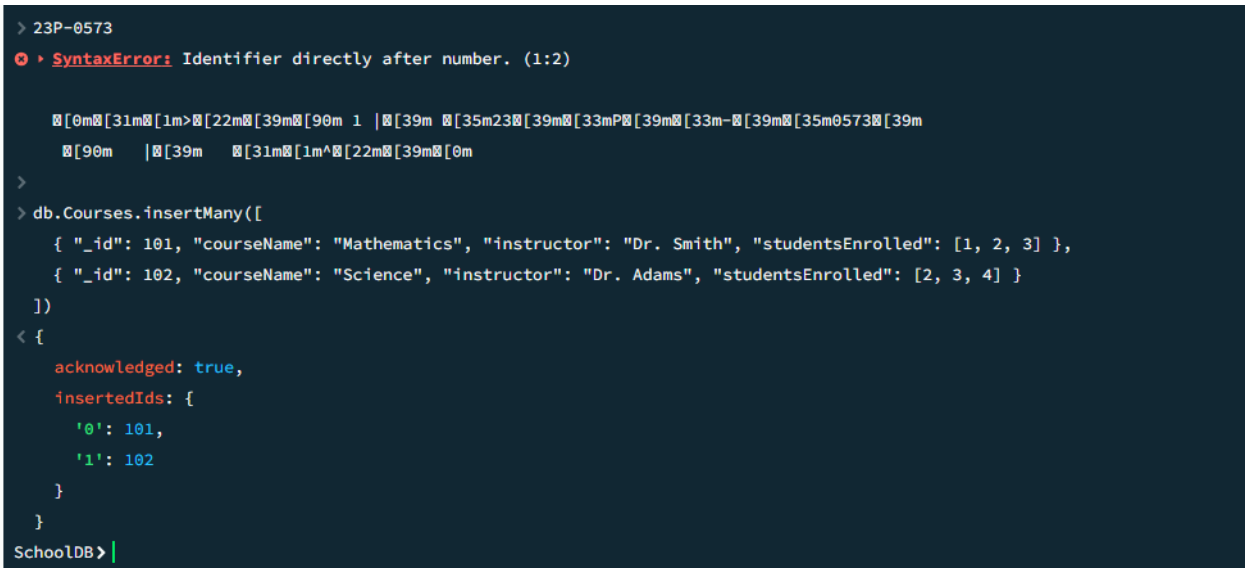
```
SchoolDB> db.createCollection("Students")
{ ok: 1 }
SchoolDB> 23P-0573
Uncaught:
SyntaxError: Identifier directly after number. (1:2)

> 1 | 23P-0573
    |   ^
  2 |

SchoolDB> db.createCollection("Courses")
{ ok: 1 }
SchoolDB>
```

3. Insert the following documents into the Students collection:

```
db.Students.insertMany([
 { "_id": 1, "name": "Alice", "age": 20, "scores": { "math": 85, "science": 90 } },
 { "_id": 2, "name": "Bob", "age": 22, "scores": { "math": 78, "science": 82 } },
 { "_id": 3, "name": "Charlie", "age": 21, "scores": { "math": 92, "science": 88 } },
 { "_id": 4, "name": "Daisy", "age": 23, "scores": { "math": 68, "science": 74 } }
])
```

```
> db.Students.insertMany([
    { "_id": 1, "name": "Alice", "age": 20, "scores": { "math": 85, "science": 90 } },
    { "_id": 2, "name": "Bob", "age": 22, "scores": { "math": 78, "science": 82 } },
    { "_id": 3, "name": "Charlie", "age": 21, "scores": { "math": 92, "science": 88 } },
    { "_id": 4, "name": "Daisy", "age": 23, "scores": { "math": 68, "science": 74 } }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': 1,
      '1': 2,
      '2': 3,
      '3': 4
    }
  }
SchoolDB> 23P-0573
```

4. Insert the following documents into the `Courses` collection:



Above error just for roll number below queury is right

5. Use findOne to retrieve:

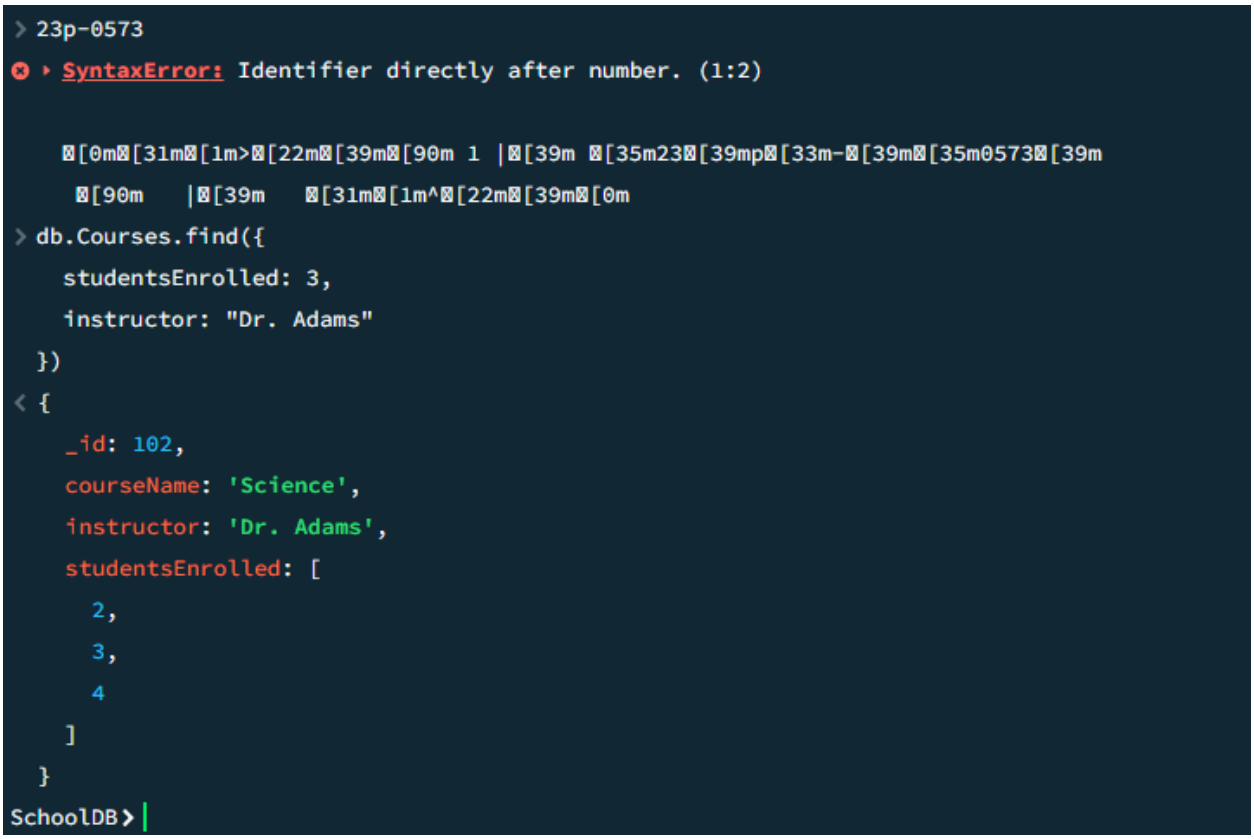
```
db.Students.find({
  "scores.math": { $gte: 85 },
  age: { $lt: 22 }
})
```

db.Courses.find({

studentsEnrolled: 3,

instructor: "Dr. Adams"

})



6. Use find to retrieve:

o Students with math score >= 80 and science score < 90.

```
> db.Students.find({
    "scores.math": { $gte: 80 },
    "scores.science": { $lt: 90 }
  })
< {
    _id: 3,
    name: 'Charlie',
    age: 21,
    scores: {
      math: 92,
      science: 88
    }
  }
SchoolDB> 23p0573
```

o Students whose age is < 23 or have a math score >= 85.

```
>_MONGOSH
> db.Students.find({
    "scores.science": { $gte: 80 },
    $or: [
      { "scores.math": { $lt: 75 } },
      { age: { $gt: 22 } }
    ]
  })
<
SchoolDB> 23p0573
```

o Students with science score >= 80 and (either math score < 75 or age > 22).

```
>_MONGOSH
    scores: {
      math: 85,
      science: 90
    }
  }
  {
    _id: 2,
    name: 'Bob',
    age: 22,
    scores: {
      math: 78,
      science: 82
    }
  }
  {
    _id: 3,
    name: 'Charlie',
    age: 21,
    scores: {
      math: 92,
      science: 88
    }
  }
SchoolDB>
```

7. Use updateOne to:

Increase the science score of Bob where math score ≥ 75:

```
> db.Students.updateOne(
    { name: "Bob", "scores.math": { $gte: 75 } },
    { $inc: { "scores.science": 1 } }  // Increment science by 1 (adjust value as needed)
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
SchoolDB> 23p0573
```

8. **`updateMany` Increase math score by 5 where science < 80 and age > 22:**

```
> db.Students.updateMany(
    { "scores.science": { $lt: 80 }, age: { $gt: 22 } },
    { $inc: { "scores.math": 5 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
SchoolDB> 23p0573
```

9. **`deleteOne` Remove student named "Daisy" with science score < 80:**

```
> db.Students.deleteOne({
    name: "Daisy",
    "scores.science": { $lt: 80 }
  })
< {
    acknowledged: true,
    deletedCount: 1
  }
SchoolDB> 23p-0573
```

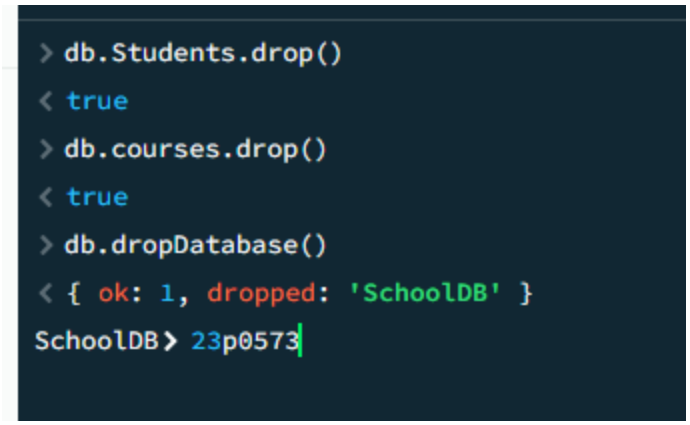
10. **`deleteMany` Remove courses where `studentsEnrolled` includes 2 or instructor is "Dr. Smith"**

```
> db.Courses.deleteMany({
    $or: [
      { studentsEnrolled: 2 },
      { instructor: "Dr. Smith" }
    ]
  })
< {
    acknowledged: true,
    deletedCount: 2
  }
SchoolDB> 23p0573
```

11. **Drop the `Students` collection:**

12. **Drop the `Courses` collection:**

13. **Delete the `SchoolDB` database:**



Task which we have to done on manual :

1) What is the MongoDB query to display all the documents in the collection `restaurants`?

db.restaurants.find()

2. What is the MongoDB query to display the fields `restaurant_id`, `name`, `borough` and `cuisine` for all the documents in the collection `restaurants`?

db.restaurants.find({}, { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 })

3. What is the MongoDB query to display the fields `restaurant_id`, `name`, `borough` and `cuisine`, but exclude the field `_id` for all the documents in the collection `restaurants`?

db.restaurants.find({}, { _id: 0, restaurant_id: 1, name: 1, borough: 1, cuisine: 1 })

4. What is the MongoDB query to display the fields `restaurant_id`, `name`, `borough` and `zip code`, but exclude the field `_id` for all the documents in the collection `restaurants`?

```
db.restaurants.find({}, { _id: 0, restaurant_id: 1, name: 1, borough: 1, "address.zipcode": 1 })
```

5. What is the MongoDB query to display all the restaurants which are in the borough `Bronx`?

```
db.restaurants.find({ borough: "Bronx" })
```

6. What is the MongoDB query to display the first 5 restaurants which are in the borough `Bronx`?

```
db.restaurants.find({ borough: "Bronx" }).limit(5)
```

7. What is the MongoDB query to find the restaurants which achieved a score more than 90?

```
db.restaurants.find({ "grades.score": { $gt: 90 } })
```

8. What is the MongoDB query to find the restaurants that achieved a score more than 80 but less than 100?

```
db.restaurants.find({ "grades.score": { $gt: 80, $lt: 100 } })
```

9. What is the MongoDB query to find the restaurants which are located at a latitude value less than `-95.754168`?

```
db.restaurants.find({ "address.coord.0": { $lt: -95.754168 } })
```