```
Please enter a MongoDB connection string (Default: mongodb://localhost/): mongosh
mongosh
Current Mongosh Log ID: 68725b859ac0cd5216718dc3
Connecting to:          mongodb://127.0.0.1:27017/mongosh?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.5
Using MongoDB:          8.0.11
Using Mongosh:          2.5.5

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


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

------
   The server generated these startup warnings when booting
   2025-07-12T12:12:25.696+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

mongosh> show dbs
admin   40.00 KiB
config  60.00 KiB
local   72.00 KiB
mongosh> use facultySystemDB
switched to db facultySystemDB
facultySystemDB> db.createCollection("Student")
{ ok: 1 }
```

```
facultySystemDB> db.Student.insertOne({FirstName:"Mrihan", LastName:"Mohamed", age:26, Faculty:{name:"FCI", address:"Minia"}, grades:[{C_name:"Mongo DB", gr
ade:60, pass:true}, {c_name:"JS", grade:100, pass:true}], isFired:false})
{
  acknowledged: true,
  insertedId: ObjectId('68726b1a9ac0cd5216718dc4')
}
facultySystemDB> db.Student.insertMany([{FirstName:”semon”, LastName:”hany”, age:22, Faculty:{name:”FCI”, address:”Qena”}, grades:[{C_name:”Mongo DB”, grade
:85, pass:true}, {C_name:”Angular”, grade:70, pass:true}], isFired:false},
... {FirstName:”abear”, LastName:”ahmed”, age:23, Faculty:{name:”FCI”, address:”Assuit”}, grades:[{C_name:”React”, gradegrade:40, pass:false}, {C_name:”JS”,
 grade:70, pass:true}], isFired:true}])
...
facultySystemDB> db.Student.insertMany([{FirstName:”semon”, LastName:”hany”, age:22, Faculty:{name:”FCI”, address:”Qena”facultySystemDB> db.Student.insertMa
ny([{FirstName:"semon", LastName:"hany", age:22, Faculty:{name:"FCI", address:"Qena"}, grades:[{C_name:"Mongo DB", grade:85, pass:true}, {C_name:"Angular",
grade:70, pass:true}], isFired:false},{FirstName:"abear", LastName:"ahmed", age:23, Faculty:{name:"FCI", address:"Assuit"}, grades:[{C_name:"React", grade:4
0, pass:false}, {C_name:"JS", grade:70, pass:true}], isFired:true}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68726c3d9ac0cd5216718dc5'),
    '1': ObjectId('68726c3d9ac0cd5216718dc6')
  }
}
facultySystemDB> db.Student.find()
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
```

```
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc6'),
    FirstName: 'abear',
    LastName: 'ahmed',
    age: 23,
    Faculty: { name: 'FCI', address: 'Assuit' },
    grades: [
      { C_name: 'React', grade: 40, pass: false },
      { C_name: 'JS', grade: 70, pass: true }
    ],
    isFired: true
  }
]
facultySystemDB> db.Student.find({FirstName:"semon"})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  }
]
```

```
facultySystemDB> db.Student.find({$or:[{FirstName:"Mrihan"},{LastName:"ahmed"}]})
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc6'),
    FirstName: 'abear',
    LastName: 'ahmed',
    age: 23,
    Faculty: { name: 'FCI', address: 'Assuit' },
    grades: [
      { C_name: 'React', grade: 40, pass: false },
      { C_name: 'JS', grade: 70, pass: true }
    ],
    isFired: true
  }
]
facultySystemDB> db.Student.find({FirstName:{$ne:"semon"}})
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
```

```
facultySystemDB> db.Student.find({FirstName:{$ne:"semon"}})
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc6'),
    FirstName: 'abear',
    LastName: 'ahmed',
    age: 23,
    Faculty: { name: 'FCI', address: 'Assuit' },
    grades: [
      { C_name: 'React', grade: 40, pass: false },
      { C_name: 'JS', grade: 70, pass: true }
    ],
    isFired: true
  }
]
facultySystemDB> db.Student.find({age:{$lt:25}})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  },
```

```
]
facultySystemDB> db.Student.find({isFired:{$exists:true}})
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc6'),
    FirstName: 'abear',
    LastName: 'ahmed',
    age: 23,
    Faculty: { name: 'FCI', address: 'Assuit' },
    grades: [
      { C_name: 'React', grade: 40, pass: false },
      { C_name: 'JS', grade: 70, pass: true }
    ],
    isFired: true
  }
]
```

```
facultySystemDB> db.Student.find({$and:[{age:{$gte:25}},{Faculty:{$ne:null}}]})
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'Mohamed',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  }
]
facultySystemDB> db.Student.find({FirstName:"semon"},{FirstName:1, LastName:1, Faculty:0, age:40}
MongoServerError[Location31254]: Cannot do exclusion on field Faculty in inclusion projection
facultySystemDB> db.Student.find({FirstName:"semon"},{FirstName:1, LastName:1, age:40})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22
  }
]
facultySystemDB> db.Student.find({FirstName:"semon"},{FirstName:1, LastName:1, GPA:3.5})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany'
  }
]
```

```
  }
]
facultySystemDB> db.Student.find({FirstName:"semon"},{isFired:0, Faculty:0})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ]
  }
]
facultySystemDB> db.Student.updateOne({FirstName:"abear"},{$set:{LastName:"mostafa"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultySystemDB> db.Student.updateMany({$or:[{FirstName:"abear"},{FirstName:"Mrihan"}]}, {$set:{LastName:"mansour"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

```
}
facultySystemDB> db.Student.find()
[
  {
    _id: ObjectId('68726b1a9ac0cd5216718dc4'),
    FirstName: 'Mrihan',
    LastName: 'mansour',
    age: 26,
    Faculty: { name: 'FCI', address: 'Minia' },
    grades: [
      { C_name: 'Mongo DB', grade: 60, pass: true },
      { c_name: 'JS', grade: 100, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc6'),
    FirstName: 'abear',
    LastName: 'mansour',
    age: 23,
    Faculty: { name: 'FCI', address: 'Assuit' },
    grades: [
      { C_name: 'React', grade: 40, pass: false },
      { C_name: 'JS', grade: 70, pass: true }
    ],
    isFired: true
  }
]
```

```
facultySystemDB> db.Student.replaceOne({FirstName:"abear"},{FirstName:"aya", LastName:"saed", age:26})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultySystemDB> db.Student.findOneAndUpdate({},{$unset:{isFired:false}}, {returnDocument:after})
ReferenceError: after is not defined
facultySystemDB> db.Student.findOneAndUpdate({},{$unset:{isFired:false}}, {returnDocument:"after"})
{
  _id: ObjectId('68726b1a9ac0cd5216718dc4'),
  FirstName: 'Mrihan',
  LastName: 'mansour',
  age: 26,
  Faculty: { name: 'FCI', address: 'Minia' },
  grades: [
    { C_name: 'Mongo DB', grade: 60, pass: true },
    { c_name: 'JS', grade: 100, pass: true }
  ]
}
facultySystemDB> db.Student.find({isFired:{$exist:true}})
MongoServerError[BadValue]: unknown operator: $exist
facultySystemDB> db.Student.find({isFired:{$exists:true}})
[
  {
    _id: ObjectId('68726c3d9ac0cd5216718dc5'),
    FirstName: 'semon',
    LastName: 'hany',
    age: 22,
    Faculty: { name: 'FCI', address: 'Qena' },
    grades: [
      { C_name: 'Mongo DB', grade: 85, pass: true },
      { C_name: 'Angular', grade: 70, pass: true }
    ],
    isFired: false
  }
]
facultySystemDB> db.Student.UpdateMany({},{$inc:{age:3}})
```

```
facultySystemDB> db.Student.UpdateMany({},{$inc:{age:3}})
TypeError: db.Student.UpdateMany is not a function
facultySystemDB> db.Student.updateMany({},{$inc:{age:3}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
facultySystemDB> db.Student.findOneAndDelete({age:{$e:29}}, {returnDocument:"after"})
MongoServerError[BadValue]: unknown operator: $e
facultySystemDB> db.Student.findOneAndDelete({age:{$gte:29}}, {returnDocument:"after"})
{
  _id: ObjectId('68726b1a9ac0cd5216718dc4'),
  FirstName: 'Mrihan',
  LastName: 'mansour',
  age: 29,
  Faculty: { name: 'FCI', address: 'Minia' },
  grades: [
    { C_name: 'Mongo DB', grade: 60, pass: true },
    { c_name: 'JS', grade: 100, pass: true }
  ]
}
facultySystemDB> db.Student.drop()
true
facultySystemDB> show dbs
admin    40.00 KiB
config  108.00 KiB
local    72.00 KiB
facultySystemDB> db.dropDatabase()
{ ok: 1, dropped: 'facultySystemDB' }
```

```
facultySystemDB> use facultySystemV2
switched to db facultySystemV2
facultySystemV2> db.createCollection("Students")
{ ok: 1 }
facultySystemV2> db.Student.insertOne({
...   firstName: "remon",
...   lastName: "beshara",
...   isFired: false,
...   FacultyID: 2023576,
...   courses: [
...     { course_id: "CS45", grade: 78 },
...     { course_id: "IT76", grade: 59 }
...   ]
... })
...
{
  acknowledged: true,
  insertedId: ObjectId('687278469ac0cd5216718dc7')
}
facultySystemV2> db.Faculty.insertOne({name:"FCI", address:"sohag"})
{
  acknowledged: true,
  insertedId: ObjectId('687278b09ac0cd5216718dc8')
}
facultySystemV2> show dbs
admin             40.00 KiB
config           108.00 KiB
facultySystemV2   56.00 KiB
local             72.00 KiB
facultySystemV2> db.createCollection("Courses")
{ ok: 1 }
facultySystemV2> db.Course.insertMany([
...   { courseName: "web technology", final_mark: 100 },
...   { courseName: "database", final_mark: 70 },
...   { courseName: "history of computers", final_mark: 150 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('687279189ac0cd5216718dc9'),
```

```
}
facultySystemV2> db.Courses.createIndex({id:1},{unique:true})
id_1
facultySystemV2> db.Courses.createIndex({TTL_id:1}, {expireAfterSeconds:240})
TTL_id_1
facultySystemV2> db.Courses.find()

facultySystemV2> db.Courses.find({courseName:"database"}).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'facultySystemV2.Courses',
    parsedQuery: { courseName: { '$eq': 'database' } },
    indexFilterSet: false,
    queryHash: 'FC7860A0',
    planCacheShapeHash: 'FC7860A0',
    planCacheKey: 'B1FF6A80',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { courseName: { '$eq': 'database' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  queryShapeHash: '1AAC9C29C882894E4236B80586C418F3E0E7DFB1BE26A4067C6E0759D6FCA596',
  command: {
    find: 'Courses',
    filter: { courseName: 'database' },
    '$db': 'facultySystemV2'
  },
  serverInfo: {
    host: 'semon-hany',
    port: 27017,
    version: '8.0.11',
    gitVersion: 'bed99f699da6cb2b74262aa6d473446c41476643'
```

```
]
facultySystemV2> db.Courses.createIndex({TTL_id:1}, {expireAfterSeconds:240})
TTL_id_1
facultySystemV2>  db.Courses.find()
[
  {
    _id: ObjectId('687279189ac0cd5216718dc9'),
    courseName: 'web technology',
    final_mark: 100
  },
  {
    _id: ObjectId('687279189ac0cd5216718dca'),
    courseName: 'database',
    final_mark: 70
  },
  {
    _id: ObjectId('687279189ac0cd5216718dcb'),
    courseName: 'history of computers',
    final_mark: 150
  }
]
facultySystemV2> db.Courses.createIndex({TTL_id:1}, {expireAfterSeconds:500})
MongoServerError[IndexOptionsConflict]: An equivalent index already exists with the same name but different options. Requested index: { v: 2, key: { TTL_id:
 1 }, name: "TTL_id_1", expireAfterSeconds: 500 }, existing index: { v: 2, key: { TTL_id: 1 }, name: "TTL_id_1", expireAfterSeconds: 240 }
facultySystemV2> db.Courses.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { TTL_id: 1 },
    name: 'TTL_id_1',
    expireAfterSeconds: 240
  }
]
facultySystemV2> db.Courses.dropIndex("TTL_id_1")
{ nIndexesWas: 2, ok: 1 }
facultySystemV2> db.Courses.createIndex({TTL_id:1}, {expireAfterSeconds:500})
TTL_id_1
facultySystemV2>
```