

```
// =====  
// ✅ USE DATABASE  
// =====  
use CollegeDB
```

```
// =====  
// ✅ CREATE COLLECTION  
// =====  
db.createCollection("students")
```

```
// =====  
// ✅ INSERT (CREATE)  
// =====
```

```
// insertOne  
db.students.insertOne({  
    roll: 1,  
    name: "Shritej",  
    branch: "CS",  
    marks: 85,  
    city: "Pune"  
})
```

```
// insertMany
```

```
db.students.insertMany([
  { roll: 2, name: "Atharva", branch: "IT", marks: 92, city: "Mumbai" },
  { roll: 3, name: "Kunjal", branch: "ENTC", marks: 76, city: "Pune" },
  { roll: 4, name: "Harshad", branch: "CS", marks: 45, city: "Kolhapur" },
  { roll: 5, name: "Ayush", branch: "CS", marks: 68, city: "Pune" }
])
```

```
// =====
// ✅ READ (FIND)
// =====
```

```
// find all
db.students.find()
```

```
// projection
db.students.find({}, { name: 1, marks: 1, _id: 0 })
```

```
// =====
// ✅ LOGICAL OPERATORS
// =====
```

```
// AND
db.students.find({ $and: [ { branch: "CS" }, { city: "Pune" } ] })
```

```
// OR  
db.students.find({ $or: [ { marks: { $gt: 90 } }, { city: "Mumbai" } ] })
```

```
// NOT  
db.students.find({ marks: { $not: { $gt: 80 } } })
```

```
// NOR  
db.students.find({ $nor: [ { branch: "IT" }, { marks: { $gt: 90 } } ] })
```

```
// =====  
//  COMPARISON OPERATORS  
// =====
```

```
// > , <  
db.students.find({ marks: { $gt: 70 } })  
db.students.find({ marks: { $lt: 50 } })
```

```
// range  
db.students.find({ marks: { $gte: 60, $lte: 90 } })
```

```
// in, nin  
db.students.find({ branch: { $in: ["CS", "IT"] } })  
db.students.find({ city: { $nin: ["Pune", "Mumbai"] } })
```

```
// =====  
// ✅ REGEX (Pattern Search)  
// =====
```

```
// name starts with A  
db.students.find({ name: { $regex: "^A" } })
```

```
// ends with j, ignore case  
db.students.find({ name: { $regex: "j$", $options: "i" } })
```

```
// =====  
// ✅ UPDATE  
// =====
```

```
// updateOne  
db.students.updateOne(  
  { roll: 4 },  
  { $set: { marks: 60, city: "Nagpur" } }  
)
```

```
// updateMany  
db.students.updateMany(  
  { city: "Pune" },  
  { $set: { hostel: "Yes" } }  
)
```

```
// update with $inc, $push
db.students.updateOne(
  { roll: 2 },
  {
    $inc: { marks: 5 },
    $push: { skills: "NodeJS" }
  }
)

// =====
// ✅ SAVE() ALTERNATIVE (WORKING)
// =====

// Modern replacement for save()
db.students.updateOne(
  { _id: ObjectId("690de729b4c1271f2b50eb6b") },
  {
    $set: {
      name: "Paras",
      branch: "CS",
      marks: 88
    }
  },
  { upsert: true }
```

```
)  
  
// ======  
// ✅ SAVE() (OLD MONGODB — ONLY FOR JOURNAL)  
// ======
```

```
// This WILL NOT run in modern mongosh,  
// but write it in your journal:
```

```
/*  
db.students.save({  
    _id: ObjectId("690de729b4c1271f2b50eb6b"),  
    name: "Paras",  
    branch: "CS",  
    marks: 88  
})  
*/
```

```
// =====
```

```
// ✅ DELETE  
// =====
```

```
// deleteOne  
db.students.deleteOne({ roll: 5 })
```

```
// deleteMany  
db.students.deleteMany({ marks: { $lt: 40 } })
```

```
// =====  
// ✅ SORTING  
// =====
```

```
// descending  
db.students.find().sort({ marks: -1 })
```

```
// ascending  
db.students.find().sort({ name: 1 })
```

```
// =====  
// ✅ LIMIT & SKIP  
// =====  
db.students.find().limit(3)  
db.students.find().skip(2)
```

```
// =====  
// ✅ DISTINCT  
// =====
```

```
db.students.distinct("city")

// =====
// ✅ COUNT
// =====

db.students.countDocuments()
db.students.countDocuments({ branch: "CS" })

// =====
// ✅ AGGREGATION QUERIES
// =====

// group by branch
db.students.aggregate([
  { $group: { _id: "$branch", avgMarks: { $avg: "$marks" }, totalStudents: { $sum: 1 } } }
])

// match + sort + limit
db.students.aggregate([
  { $match: { city: "Pune" } },
  { $sort: { marks: -1 } },
  { $limit: 5 }
])
```

```
// =====  
// ✅ INDEX  
// =====  
db.students.createIndex({ name: 1 })
```

```
// =====  
// ✅ DROP COLLECTION / DATABASE  
// =====  
db.students.drop()  
db.dropDatabase()
```