Practical 3: Create with given database and performed with following operators: Database: Student Collection: example Document: two documents that contain the details of the shapes in the form of field-value pairs.

a. $cmp b. $gt c. $lt d. $eq e. $lte f. $gte

Step 1: Create Database and Collection

```
> use Students
< already on db Students
> db.createCollection("Example")
```

Step 2: Insert Documents

```
> db.Example.insertMany([
    {_id:1, shape:'circle', side:0, area:78.5, color:'red'},
    {_id:2, shape:'Square', side:4, area:64, color:'Blue'}])
< {
    acknowledged: true,
    insertedIds: {
      '0': 1,
      '1': 2
    }
  }
```

Step 3: Use MongoDB Comparison Operators

a. $cmp — Compare two values

```
> db.Example.aggregate([{
    $project:{shape:1, compareArea:{$cmp:["$area",70]}}}])
< {
    _id: 1,
    shape: 'circle',
    compareArea: 1
  }
  {
    _id: 2,
    shape: 'Square',
    compareArea: -1
  }
```

b. $gt — Greater Than

```
> db.Example.find({area:{$gt:70}})
< {
    _id: 1,
    shape: 'circle',
    side: 0,
    area: 78.5,
    color: 'red'
  }
```

c. $lt — Less Than

```
> db.Example.find({area:{$lt:70}})
< {
    _id: 2,
    shape: 'Square',
    side: 4,
    area: 64,
    color: 'Blue'
  }
```

d. $eq — Equal

```
> db.Example.find({side:{$eq:4}})
< {
    _id: 2,
    shape: 'Square',
    side: 4,
    area: 64,
    color: 'Blue'
  }
```

e. $lte — Less Than or Equal To

```
> db.Example.find({area:{$lte:64}})
< {
    _id: 2,
    shape: 'Square',
    side: 4,
    area: 64,
    color: 'Blue'
  }
```

f. $gte — Greater Than or Equal To

```
> db.Example.find({area:{$gte:78.5}})
< {
    _id: 1,
    shape: 'circle',
    side: 0,
    area: 78.5,
    color: 'red'
  }
```

Practical 4: Create database using Department table and performed following CRUD operations: Create Operations, Read Operations, Update Operations, Delete Operations.

Step 1: Create / Insert Data (C → Create Operation)

```
> use College
< switched to db College
> db.createCollection("Department")
< { ok: 1 }
```

```
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('692312d480c791abe818d686'),
      '1': ObjectId('692312d480c791abe818d687'),
      '2': ObjectId('692312d480c791abe818d688')
    }
  }
```

Step 2: Read Data (R → Read Operation)

```
> db.Department.find()
< {
    _id: ObjectId('692312d480c791abe818d686'),
    dept_id: 1,
    dept_name: 'Computer Science',
    head: 'Dr. Sharma',
    total_staff: 25,
    location: 'Block A'
  }
  {
    _id: ObjectId('692312d480c791abe818d687'),
    dept_id: 2,
    dept_name: 'Mathematics',
    head: 'Dr. Meena',
    total_staff: 18,
    location: 'Block B'
  }
  {
    _id: ObjectId('692312d480c791abe818d688'),
    dept_id: 3,
    dept_name: 'Commerce',
    head: 'Dr. Verma',
    total_staff: 20,
    location: 'Block C'
  }
```

```
> db.Department.find({}, { dept_name: 1, head: 1, _id: 0 })
< {
    dept_name: 'Computer Science',
    head: 'Dr. Sharma'
  }
  {
    dept_name: 'Mathematics',
    head: 'Dr. Meena'
  }
  {
    dept_name: 'Commerce',
    head: 'Dr. Verma'
  }
```

```
> db.Department.find({ total_staff: { $gt: 18 } })
< {
    _id: ObjectId('692312d480c791abe818d686'),
    dept_id: 1,
    dept_name: 'Computer Science',
    head: 'Dr. Sharma',
    total_staff: 25,
    location: 'Block A'
  }
  {
    _id: ObjectId('692312d480c791abe818d688'),
    dept_id: 3,
    dept_name: 'Commerce',
    head: 'Dr. Verma',
    total_staff: 20,
    location: 'Block C'
  }
```

```
> db.Department.findOne({ dept_id: 2 })
< {
    _id: ObjectId('692312d480c791abe818d687'),
    dept_id: 2,
    dept_name: 'Mathematics',
    head: 'Dr. Meena',
    total_staff: 18,
    location: 'Block B'
  }
```

Step 3: Update Data (U → Update Operation)

```
> db.Department.updateOne(
    { dept_name: "Mathematics" },
    { $set: { head: "Dr. Ramesh" } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
> db.Department.updateMany(
    {},
    { $inc: { total_staff: 2 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 3,
    modifiedCount: 3,
    upsertedCount: 0
  }
```

Step 4: Delete Data (D → Delete Operation)

```
> db.Department.deleteOne({ dept_name: "Commerce" })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

```
> db.Department.deleteMany({ total_staff: { $lt: 20 } })
< {
    acknowledged: true,
    deletedCount: 0
  }
```

```
> db.Department.deleteMany({})
< {
    acknowledged: true,
    deletedCount: 2
  }
```

Practical 5: Create table called Customer Table.

a. **Insert records into the table.**

b. **Add salary column to the table.**

c. **Alter the table column domain.**

d. **Drop salary column of the customer table.**

e. **Delete the rows of customer table whose cust_city is “hyd”.**

Step 1: Create Collection: Customer

```
> use Customer
< switched to db Customer
> db.createCollection("Customer")
< { ok: 1 }
```

Step 2: Insert records into the table.

```
> db.Customer.insertOne({
      cust_id: 1,
      cust_name: "Riya",
      cust_city: "pune"
  })
< {
    acknowledged: true,
    insertedId: ObjectId('692318b180c791abe818d689')
  }
```

```
> db.Customer.insertMany([
      { cust_id: 1, cust_name: "Riya", cust_city: "pune" },
      { cust_id: 2, cust_name: "Amit", cust_city: "hyd" },
      { cust_id: 3, cust_name: "Sara", cust_city: "mumbai" },
      { cust_id: 4, cust_name: "Vikram", cust_city: "hyd" },
      { cust_id: 5, cust_name: "Nita", cust_city: "delhi" }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('692318bc80c791abe818d68a'),
      '1': ObjectId('692318bc80c791abe818d68b'),
      '2': ObjectId('692318bc80c791abe818d68c'),
      '3': ObjectId('692318bc80c791abe818d68d'),
      '4': ObjectId('692318bc80c791abe818d68e')
    }
  }
```

Step 3: Add salary column to the table.

```
> db.Customer.updateMany(
      {},
      { $set: { salary: 0 } }  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 6,
    modifiedCount: 6,
    upsertedCount: 0
  }
```

Step 4: Alter the table column domain.

```
> db.Customer.updateMany(
     {},
     [
         { $set: { salary: { $toString: "$salary" } } }
     ]
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 6,
    modifiedCount: 6,
    upsertedCount: 0
  }
```

Step 5: Drop salary column of the customer table.

```
> db.Customer.updateMany(
     {},
     { $unset: { salary: "" } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 6,
    modifiedCount: 6,
    upsertedCount: 0
  }
```

Step 6: Delete the rows of customer table whose cust_city is “hyd”.

```
> db.Customer.deleteMany({ cust_city: "hyd"
< {
    acknowledged: true,
    deletedCount: 2
  }
```