Dbms Practical 10 Program Code:

```
db.createCollection("books");

db.books.insertMany([
  { _id: 1, title: "MongoDB Basics", author: "John", price: 500, category: "Database", copies:
30 },
  { _id: 2, title: "Node.js Guide", author: "Alex", price: 700, category: "Backend", copies: 25
},
  { _id: 3, title: "React in Depth", author: "Sarah", price: 900, category: "Frontend", copies:
40 },
  { _id: 4, title: "Python Tricks", author: "John", price: 800, category: "Programming",
copies: 35 },
  { _id: 5, title: "MongoDB Advanced", author: "Alex", price: 1200, category: "Database",
copies: 20 }
])

db.books.find()

db.books.aggregate([
  { $group: { _id: "$category", totalCopies: { $sum: "$copies" } } }
])

db.books.aggregate([
  { $group: { _id: "$author", avgPrice: { $avg: "$price" } } }
])
```

Sort books by price in descending order

```
db.books.aggregate([
  { $sort: { price: -1 } }
])
```

Sort books by price in ascending order

```
db.books.aggregate([
  { $sort: { price: 1 } }
])
```

Show only specific fields

```
db.books.aggregate([
  { $project: { _id: 0, title: 1, author: 1, price: 1 } }
])
```

Find all Database books with price > 600, sort them by price, and show only title & price

```
db.books.aggregate([
```

```
  { $match: { category: "Database", price: { $gt: 600 } } },
  { $sort: { price: -1 } },
  { $project: { _id: 0, title: 1, price: 1 } }
])
```

Create a single-field index

```
db.books.createIndex({ title: 1 })
```

Create a compound index

```
db.books.createIndex({ author: 1, price: -1 })
```

View all indexes

```
db.books.getIndexes()
```

Check Query Performance

```
db.books.find({ author: "John" }).explain("executionStats")
```

Drop an index

```
db.books.dropIndex({ title: 1 })
```