```
db.users.insertMany([

  { "user_id": 1, "name": "John", "age": 30, "city": "New York", "interests": ["coding", "sports"],
"salary": 70000 },

  { "user_id": 2, "name": "Alice", "age": 25, "city": "Los Angeles", "interests": ["reading", "music"],
"salary": 50000 },

  { "user_id": 3, "name": "Bob", "age": 35, "city": "Chicago", "interests": ["traveling", "sports"],
"salary": 90000 },

  { "user_id": 4, "name": "Eve", "age": 28, "city": "San Francisco", "interests": ["cooking",
"reading"], "salary": 60000 },

  { "user_id": 5, "name": "Charlie", "age": 40, "city": "New York", "interests": ["music",
"gardening"], "salary": 100000 },

  { "user_id": 6, "name": "Sophia", "age": 30, "city": "Chicago", "interests": ["sports", "traveling"],
"salary": 75000 }

])
```

1. Count

Count the number of users who live in Chicago and are interested in "sports."

```
db.users.countDocuments({ city: "Chicago", interests: "sports" })
```

2. Sort

Sort users by `salary` in descending order, and then by `age` in ascending order (in case of ties).

```
db.users.find().sort({ salary: -1, age: 1 })
```

3. Limit and Skip

Fetch the third and fourth highest-paid users:

```
db.users.find().sort({ salary: -1 }).skip(2).limit(2)
```

4. Aggregation

Query 1: Average Salary by City

Find the average salary of users grouped by city:

```
db.users.aggregate([
  { $group: { _id: "$city", avgSalary: { $avg: "$salary" } } },
  { $sort: { avgSalary: -1 } }
])
```

Query 2: Users Interested in "Sports" with Total Salary

Find users who are interested in "sports" and calculate their total salary:

```
db.users.aggregate([
 { $match: { interests: "sports" } },
 { $group: { _id: null, totalSalary: { $sum: "$salary" } } }
])
```

Query 3: List Users with Selected Fields

Return only the name, age, and salary fields of users, sorted by age in ascending order:

```
db.users.aggregate([
 { $project: { _id: 0, name: 1, age: 1, salary: 1 } },
 { $sort: { age: 1 } }
])
```

5. Complex Filter: Multiple Conditions

Find users who are:

- Above 30 years old
- Living in "Chicago" or "New York"
- Interested in "sports"

```
db.users.find({
 age: { $gt: 30 },
 city: { $in: ["Chicago", "New York"] },
 interests: "sports"
})
```

6. Add a New Field to Users

- Add a new field called `status` that labels users as either "High Earner" (salary >= 75000) or "Low Earner":

```
db.users.aggregate([
 {
  $addFields: {
   status: { $cond: { if: { $gte: ["$salary", 75000] }, then: "High Earner", else: "Low Earner" } }
  }
 }
])
```