MongoDB Sorting and Limiting Examples

use yourDatabaseName; // Replace with your actual DB name

# Sample Data

Assuming you already inserted the following sample data into the `employees` collection:

db.employees.insertMany([  
 {  
 "\_id": 1,  
 "name": "Alice",  
 "age": 30,  
 "department": "HR",  
 "salary": 5000,  
 "skills": ["communication", "management"],  
 "active": true  
 },  
 {  
 "\_id": 2,  
 "name": "Bob",  
 "age": 25,  
 "department": "IT",  
 "salary": 7000,  
 "skills": ["javascript", "nodejs", "mongodb"],  
 "active": false  
 },  
 {  
 "\_id": 3,  
 "name": "Charlie",  
 "age": 35,  
 "department": "Finance",  
 "salary": 6500,  
 "skills": ["excel", "accounting"],  
 "active": true  
 },  
 {  
 "\_id": 4,  
 "name": "Diana",  
 "age": 28,  
 "department": "IT",  
 "salary": 7200,  
 "skills": ["python", "mongodb"],  
 "active": true  
 }  
]);

# Sorting in MongoDB

## Sort by Salary Ascending

db.employees.find().sort({ salary: 1 });

Explanation: 1 indicates ascending order.

## Sort by Salary Descending

db.employees.find().sort({ salary: -1 });

Explanation: -1 indicates descending order.

## Sort by Multiple Fields

db.employees.find().sort({ department: 1, salary: -1 });

Explanation: Sort by department ascending, then salary descending.

## Sort by Name Alphabetically

db.employees.find().sort({ name: 1 });

Explanation: Alphabetical (A-Z) sort.

## Sort by Boolean Field (active)

db.employees.find().sort({ active: -1 });

Explanation: Sorts with true (active) employees first.

# Limit and Skip in MongoDB

## Limit Results (Top N Records)

db.employees.find().limit(2);

Explanation: Returns only the first 2 documents.

## Skip First N Records

db.employees.find().skip(2);

Explanation: Skips the first 2 records and returns the rest.

## Combine Sort + Limit

db.employees.find().sort({ salary: -1 }).limit(1);

Explanation: Returns the employee with the highest salary.

## Pagination (Skip + Limit)

db.employees.find().sort({ name: 1 }).skip(2).limit(2);

Explanation: Skip first 2 sorted by name, return next 2.