

# Extended Practice for Session 5 \_ MongoDB

## Database Commands

17 Tháng Mười 2025 8:01 CH

### 🌟 Extended Practice – Database Commands in MongoDB

#### ✳️ Exercise 1 – Create a Secure Database Environment

Task (Yêu cầu):

Create a database named company\_db, add a collection employees, and create two users:

Tạo database company\_db, thêm collection employees, và tạo hai người dùng:

1 admin\_user – full access (readWrite on all databases)

2 viewer\_user – read-only access

Commands:

```
js

use company_db
db.createCollection("employees")

db.runCommand({
  createUser: "admin_user",
  pwd: "admin123",
  roles: [{ role: "root", db: "admin" }]
})

db.runCommand({
  createUser: "viewer_user",
  pwd: "viewer123",
  roles: [{ role: "read", db: "company_db" }]
})
```

Explanation:

- "root" role gives full privileges on all databases.
- "read" role allows only reading data from the assigned database.  
("root" cho phép toàn quyền truy cập mọi database; "read" chỉ cho phép xem dữ liệu.)

👉 Đây là **cấu hình phân quyền chuẩn** giữa người quản trị và người xem dữ liệu.

#### 📝 Exercise 2 – Insert Sample Employee Records (Bài 2 – Chèn dữ liệu nhân viên mẫu)

Commands:

```
db.employees.insertMany([
  { emp_id: 1001, name: "Alice", dept: "Sales", salary: 5000, age: 28 },
  { emp_id: 1002, name: "Bob", dept: "IT", salary: 8000, age: 32 },
  { emp_id: 1003, name: "Charlie", dept: "HR", salary: 6000, age: 26 },
  { emp_id: 1004, name: "David", dept: "IT", salary: 9000, age: 30 },
  { emp_id: 1005, name: "Eva", dept: "Sales", salary: 5500, age: 29 }]
```

])

#### Explanation:

This adds 5 employee documents representing ID, name, department, salary, and age.

(Lệnh này thêm 5 bản ghi nhân viên với ID, tên, phòng ban, lương và tuổi.)

→ Bạn sẽ dùng các bản ghi này để luyện tập find, update, delete, và aggregate.

#### 🔍 Exercise 3 – Display Specific Fields and Sort Results

(Bài 3 – Hiển thị trường cụ thể và sắp xếp kết quả)

#### Commands:

```
js

db.runCommand({
  find: "employees",
  projection: { name: 1, dept: 1, salary: 1, _id: 0 },
  sort: { salary: -1 }
})
```

#### Explanation:

- projection chọn chỉ các trường cần thiết (name, dept, salary).
- sort: {salary: -1} sắp xếp giảm dần theo lương.  
→ Giúp bạn **xem danh sách nhân viên lương cao đến thấp**.

#### 📝 Exercise 4 – Update Employee Salaries (Bài 4 – Cập nhật lương nhân viên)

#### Task:

Increase salary by **10%** for all employees in **Sales department**.

(Tăng 10% lương cho tất cả nhân viên phòng Sales.)

#### Command:

```
js

db.runCommand({
  update: "employees",
  updates: [
    {
      q: { dept: "Sales" },
      u: { $mul: { salary: 1.1 } },
      multi: true
    }
  ]
})
```

#### Explanation:

- `$mul` multiplies salary by 1.1 (tăng 10%).
- `multi: true` cập nhật nhiều document cùng lúc.  
→ Giúp bạn hiểu **toán tử số học `$mul`** trong cập nhật hàng loạt.

### Exercise 5 – Delete Employees Below Salary Threshold (Bài 5 – Xóa nhân viên có lương thấp)

#### Task:

Delete all employees with salary **less than 5500**. (Xóa tất cả nhân viên có lương dưới 5500.)

#### Command:

```
js

db.runCommand({
  delete: "employees",
  deletes: [
    { q: { salary: { $lt: 5500 } }, limit: 0 }
  ]
})
```

#### Explanation:

- `$lt` = less than (nhỏ hơn).
- `limit: 0` xóa tất cả document khớp điều kiện.  
→ Đây là **lệnh delete toàn bộ có điều kiện**, tương tự DELETE FROM WHERE trong SQL.

### Exercise 6 – findAndModify Example (Bài 6 – Ví dụ lệnh findAndModify)

#### Task:

Find employee **Bob**, increase his salary by 15%, and return the updated document.

(Tìm nhân viên **Bob**, tăng lương 15%, và hiển thị kết quả mới.)

#### Command:

```
js

db.runCommand({
  findAndModify: "employees",
  query: { name: "Bob" },
  update: { $mul: { salary: 1.15 } },
  new: true
})
```

#### Explanation:

- `findAndModify` vừa tìm, vừa cập nhật, vừa trả kết quả sau khi cập nhật.
- `$mul` tăng giá trị trường salary theo tỷ lệ.  
→ Lệnh mạnh mẽ để **thực hiện thao tác CRUD trong một bước duy nhất**.

### Exercise 7 – Aggregate Employee Data (Bài 7 – Tổng hợp dữ liệu nhân viên)

**Task:**

Display total salary by department. (Hiển thị tổng lương theo từng phòng ban.)

**Command:**

```
js                                     ⌂ Sao chép mã

db.runCommand({
  aggregate: "employees",
  pipeline: [
    { $group: { _id: "$dept", total_salary: { $sum: "$salary" } } },
    { $sort: { total_salary: -1 } }
  ],
  cursor: {}
})
```

**Explanation:**

- \$group nhóm theo phòng (dept) và tính tổng (\$sum).
- \$sort sắp xếp theo tổng lương giảm dần.  
→ Kết quả giống SELECT dept, SUM(salary) FROM employees GROUP BY dept ORDER BY SUM DESC trong SQL.

 **Exercise 8 – Authenticate and View Privileges** (Bài 8 – Xác thực người dùng và xem quyền hạn)**Command:**

```
db.auth("viewer_user", "viewer123")
db.runCommand({ userInfo: { user: "viewer_user", db: "company_db" },
  showPrivileges: true })
```

**Explanation:**

- db.auth() xác thực tài khoản người dùng.
- userInfo hiển thị quyền hiện có.  
→ Giúp kiểm tra xem user có đúng quyền “read-only” hay không.

 **Exercise 9 – Count Employees by Condition** (Bài 9 – Đếm nhân viên theo điều kiện)**Task:**

Count employees older than 28. (Đếm số nhân viên trên 28 tuổi.)

**Command:**

```
js

db.runCommand({
  count: "employees",
  query: { age: { $gt: 28 } }
})
```

#### Explanation:

- \$gt = greater than.
- Kết quả trả về dạng { n: <count> }. → Dễ dùng cho thống kê hoặc báo cáo nhân sự.

#### ⌚ Exercise 10 – Distinct Departments (Bài 10 – Lấy danh sách phòng ban duy nhất)

#### Command:

```
js

db.runCommand({
  distinct: "employees",
  key: "dept"
})
```

#### Explanation:

- distinct trả về danh sách các giá trị **không trùng lặp** trong một trường.  
→ Giống như SELECT DISTINCT dept FROM employees.

#### ✓ Summary (Tóm tắt ôn tập nâng cao)

Command	Purpose	Giải thích
createUser	Create user	Tạo tài khoản người dùng
grantRolesToUser	Assign roles	Cấp quyền cho user
auth	Authenticate	Xác thực đăng nhập
insert	Add data	Thêm dữ liệu mới
update	Modify data	Cập nhật dữ liệu
delete	Remove data	Xóa dữ liệu
findAndModify	Find + Update	Tìm và chỉnh sửa
aggregate	Data grouping	Tổng hợp dữ liệu
count	Count results	Đếm bản ghi
distinct	Unique values	Lấy giá trị duy nhất

#### 💡 Practice Tips (Mẹo luyện tập)

- ◆ Luôn dùng use <database> trước khi chạy db.runCommand()
- ◆ Khi test quyền user, hãy logout hoặc mở **MongoDB shell mới** rồi db.auth()
- ◆ Khi thao tác nhiều, dùng db.getUsers() để xem toàn bộ user hiện có
- ◆ Đừng quên kiểm tra kết quả bằng db.collection.find() sau mỗi thao tác

