

# Session 05 Database Commands

17 Tháng Mười 2025 8:03 CH

## 1 Introduction to Database Commands

Developers use MongoDB database commands to create, modify, and update databases.

(Các lập trình viên sử dụng các lệnh cơ sở dữ liệu của MongoDB để tạo, sửa đổi và cập nhật cơ sở dữ liệu.)

- Database commands là **các lệnh nội bộ** để thao tác trực tiếp với database và collection.
- Có thể chia thành 4 nhóm chính:
  1. User Management (Quản lý người dùng)
  2. Authentication (Xác thực)
  3. Write Operations (Thao tác ghi)
  4. Aggregation (Tổng hợp dữ liệu)

**Basic syntax:**

```
js  
  
db.runCommand({ <command> })
```

→ Chạy lệnh trên **database hiện tại**.

Nếu là lệnh quản trị (admin), dùng:

```
js  
  
db.adminCommand({ <command> })
```

→ Luôn chạy trên database “admin” dù bạn đang ở database nào.

## 2 User Management Commands (Lệnh quản lý người dùng)

### a) createUser

Used to create a new user in the current database. (Được dùng để tạo người dùng mới trong cơ sở dữ liệu hiện tại.)

**Syntax:**

```
js  
  
db.runCommand({  
  createUser: "User_1",  
  pwd: passwordPrompt(), // hoặc "<mật khẩu>"  
  roles: [ "readWrite" ]  
})
```

#### Giải thích:

- createUser: tên người dùng.
- pwd: mật khẩu.
- roles: quyền hạn (ví dụ: "readWrite", "read").
- customData: thêm thông tin mô tả.
- writeConcern: mức độ xác nhận ghi dữ liệu.
- authenticationRestrictions: chỉ định IP được phép truy cập.

👉 Nếu user đã tồn tại → MongoDB trả lỗi **duplicate user**.

#### ✳ b) grantRolesToUser

**Grants additional roles to an existing user.** (Cấp thêm quyền cho người dùng hiện có.)

Syntax:

```
js

db.runCommand({
  grantRolesToUser: "User_1",
  roles: [
    { role: "read", db: "sample_supplies" },
    "readWrite"
  ]
})
```

#### Giải thích:

- Có thể gán quyền trong cùng hoặc khác database.
- Nếu khác DB, phải chỉ rõ { role: "read", db: "sample\_supplies" }.

#### 🔍 c) userInfo

**Displays user information.** (Hiển thị thông tin người dùng.)

Syntax:

```
js

db.runCommand({
  userInfo: { user: "User_1", db: "sample_training" },
  showPrivileges: true
})
```

#### Giải thích:

- Có thể xem thông tin 1 user, nhiều user, hoặc tất cả DB ({ forAllDBs: true }).

#### ⓧ d) dropUser

**Removes a user from the current database.** (Xóa người dùng khỏi cơ sở dữ liệu hiện tại.)

Syntax:

```
js
```

```
db.runCommand({ dropUser: "User_2" })
```

### 🔒 3 Authentication Commands (Lệnh xác thực)

#### a) db.auth

**Authenticates a user with username and password.** (Xác thực người dùng bằng tên đăng nhập và mật khẩu.)

Syntax:

```
js
```

```
db.auth("User_1", "user1")
```

Hoặc:

```
js
```

```
db.auth({
  user: "User_1",
  pwd: passwordPrompt()
})
```

Giải thích:

- Trả về 1 nếu xác thực thành công, 0 nếu thất bại.
- Xác thực chỉ hoạt động sau khi kết nối Mongo shell.
- Hỗ trợ cơ chế mã hóa **SCRAM-SHA-1, SCRAM-SHA-256**.

### 📝 ⚡ Write Operation Commands (Các lệnh ghi dữ liệu)

#### a) insert

**Inserts one or more documents into a collection.** (Chèn một hoặc nhiều tài liệu vào collection.)

Syntax:

```
db.runCommand({
  insert: "accounts",
  documents: [{ account_id: 988877, limit: 2000, products: ["InvestmentFund"] }
}]
})
```

Giải thích:

- ordered: nếu true → dừng khi gặp lỗi; false → vẫn chèn các bản ghi khác.
- bypassDocumentValidation: bỏ qua quy tắc kiểm tra schema.

### b) find

**Selects and returns documents that match a query.** (Chọn và trả về các tài liệu phù hợp với truy vấn.)

Syntax:

```
js

db.runCommand({
  find: "accounts",
  filter: { limit: { $lt: 6000 } },
  projection: { account_id: 1, limit: 1 },
  sort: { limit: 1 }
})
```

Giải thích:

- filter: điều kiện lọc.
- projection: chọn các trường hiển thị (1=hiện, 0=ẩn).
- sort: sắp xếp.
- skip, limit: phân trang.
- collation: quy tắc so sánh chuỗi (phân biệt hoa/thường, dấu).

### c) findAndModify

**Finds, modifies, and returns a single document.** (Tìm, sửa và trả về một tài liệu.)

Syntax:

```
js

db.runCommand({
  findAndModify: "accounts",
  query: { limit: 2000 },
  update: { $set: { limit: 2500 } },
  new: true
})
```

Giải thích:

- new: true → trả về tài liệu sau khi sửa.
- Nếu không có tài liệu phù hợp → trả về null.

### d) update

**Updates one or more documents in a collection.** (Cập nhật một hoặc nhiều tài liệu trong collection.)

Syntax:

```
js

db.runCommand({
```

```
update: "accounts",
updates: [
  {
    q: { limit: 3000 },
    u: { $inc: { limit: 1000 } },
    multi: true
  }
])
})
```

#### Giải thích:

- q: điều kiện chọn tài liệu.
- u: thay đổi thực hiện.
- \$set, \$inc, \$unset → các toán tử cập nhật phổ biến.
- multi: true → cập nhật nhiều tài liệu cùng lúc.

#### e) delete

**Deletes one or more documents from a collection.** (Xóa một hoặc nhiều tài liệu khỏi collection.)

#### Syntax:

```
js

db.runCommand({
  delete: "accounts",
  deletes: [{ q: { limit: 7000 }, limit: 0 }]
})
```

#### Giải thích:

- limit: 1 → xóa 1 tài liệu đầu tiên.
- limit: 0 → xóa **tất cả** tài liệu phù hợp.

#### 5 Aggregation Commands (Lệnh tổng hợp)

##### a) count

**Counts documents that match a query.** (Đếm số tài liệu thỏa mãn điều kiện.)

#### Syntax:

```
js

db.runCommand({
  count: "accounts",
  query: { limit: { $lte: 4000 } }
})
```

### b) aggregate

Performs aggregation using pipeline stages. (Thực hiện tổng hợp dữ liệu bằng pipeline.)

Syntax:

```
js

db.runCommand({
  aggregate: "accounts",
  pipeline: [
    { $project: { account_id: 1, limit: 1 } },
    { $group: { _id: "$limit", count: { $count: {} } } }
  ],
  cursor: {}
})
```

Giải thích:

- **\$project:** chọn cột hiển thị.
- **\$group:** nhóm dữ liệu theo field.
- **\$count:** đếm số lượng nhóm.

### c) distinct

Returns distinct values for a specified field. (Trả về các giá trị duy nhất của một trường cụ thể.)

Syntax:

```
js

db.runCommand({ distinct: "accounts", key: "products" })
```

→ Kết quả: danh sách giá trị khác nhau của products.

## 6 Summary

| Command          | Function                  | Vietnamese Meaning         |
|------------------|---------------------------|----------------------------|
| createUser       | Create new user           | Tạo người dùng mới         |
| grantRolesToUser | Assign permissions        | Cấp quyền truy cập         |
| usersInfo        | Show user details         | Xem thông tin người dùng   |
| dropUser         | Remove user               | Xóa người dùng             |
| auth             | Authenticate user         | Xác thực đăng nhập         |
| insert           | Add new document          | Thêm dữ liệu mới           |
| find             | Retrieve documents        | Truy vấn dữ liệu           |
| update           | Modify multiple documents | Cập nhật nhiều dữ liệu     |
| delete           | Delete data               | Xóa dữ liệu                |
| aggregate        | Summarize / group data    | Tổng hợp, thống kê dữ liệu |
| distinct         | Unique field values       | Lấy giá trị duy nhất       |
| count            | Count records             | Đếm bản ghi                |

**Key Takeaways (Tổng kết quan trọng)**

- 1 db.runCommand() là **cách chính thống nhất để chạy các lệnh nội tại** trong MongoDB.
- 2 User Management đảm bảo an toàn và kiểm soát truy cập dữ liệu.
- 3 Write Operations (insert, update, delete) giúp thao tác dữ liệu chính xác.
- 4 Aggregation Commands dùng để phân tích, thống kê dữ liệu.
- 5 Tất cả các lệnh đều có thể kết hợp với các **toán tử logic** (\$gt, \$lt, \$set, \$sum, ...) để tạo truy vấn mạnh mẽ.