

# Tham Khảo Nhanh MongoDB

CRUD, truy vấn, aggregation, indexes, thiết kế schema

## CRUD Cơ Bản

### Chèn

```
db.users.insertOne({ name: "Alice", age: 30 });
db.users.insertMany([
  { name: "Bob", age: 25 },
  { name: "Carol", age: 35 }
]);
```

### Tìm Kiếm

```
db.users.find({}); // all documents
db.users.findOne({ name: "Alice" });
db.users.find({ age: { $gt: 25 } });
db.users.find({}, { name: 1, _id: 0 }); // projection
```

### Cập Nhật

```
db.users.updateOne(
  { name: "Alice" },
  { $set: { age: 31 } }
);
db.users.updateMany(
  { age: { $lt: 18 } },
  { $set: { minor: true } }
);
db.users.replaceOne({ name: "Bob" }, { name: "Bob", age: 26 });
```

### Xóa

```
db.users.deleteOne({ name: "Alice" });
db.users.deleteMany({ age: { $lt: 18 } });
```

## Toán Tử Truy Vấn

### So Sánh

<b>\$eq / \$ne</b>	Bằng / không bằng
<b>\$gt / \$gte</b>	Lớn hơn / lớn hơn hoặc bằng
<b>\$lt / \$lte</b>	Nhỏ hơn / nhỏ hơn hoặc bằng
<b>\$in / \$nin</b>	Thuộc / không thuộc tập hợp
<b>\$exists</b>	Field tồn tại hay không
<b>\$type</b>	Kiểm tra kiểu BSON

### Logic

<b>\$and</b>	Tất cả điều kiện phải đúng
<b>\$or</b>	Ít nhất một điều kiện đúng
<b>\$not</b>	Phủ định biểu thức
<b>\$nor</b>	Không điều kiện nào đúng

### Mảng & Element

<b>\$all</b>	Mảng chứa tất cả giá trị
<b>\$elemMatch</b>	Phần tử trong mảng thỏa tất cả điều kiện
<b>\$size</b>	Mảng có kích thước cụ thể
<b>\$regex</b>	Khớp biểu thức chính quy

### Truy Vấn Ví Dụ

```
db.products.find({
  price: { $gte: 10, $lte: 100 },
  tags: { $in: ["sale", "featured"] },
  "address.city": "NYC"
});
```

## Toán Tử Cập Nhật

### Cập Nhật Field

<b>\$set</b>	Đặt giá trị field
<b>\$unset</b>	Xóa field
<b>\$inc</b>	Tăng giá trị số
<b>\$mul</b>	Nhân giá trị số
<b>\$rename</b>	Đổi tên field
<b>\$currentDate</b>	Đặt field thành ngày hiện tại
<b>\$setOnInsert</b>	Chỉ đặt khi upsert tạo document mới

### Cập Nhật Mảng

<b>\$push</b>	Thêm phần tử vào mảng
<b>\$pull</b>	Xóa phần tử khỏi mảng
<b>\$addToSet</b>	Thêm nếu chưa có
<b>\$pop</b>	Xóa phần tử đầu (-1) hoặc cuối (1)
<b>\$each</b>	Dùng với \$push để thêm nhiều phần tử

### Upsert

```
db.users.updateOne(
  { email: "alice@example.com" },
  { $set: { name: "Alice" }, $setOnInsert: { createdAt: new Date() } },
  { upsert: true }
);
```

## Aggregation Pipeline

### Các Stages Phổ Biến

<b>\$match</b>	Lọc documents (như WHERE)
<b>\$group</b>	Nhóm và tính tổng hợp
<b>\$project</b>	Hình thành lại documents (chọn/đổi tên fields)
<b>\$sort</b>	Sắp xếp documents
<b>\$limit / \$skip</b>	Phân trang
<b>\$unwind</b>	Phân rã mảng thành nhiều documents
<b>\$lookup</b>	Join với collection khác
<b>\$addFields</b>	Thêm fields mới vào documents
<b>\$count</b>	Đếm tổng số documents

### Ví Dụ Aggregation

```
db.orders.aggregate([
  { $match: { status: "completed" } },
  { $group: {
    _id: "$customerId",
    total: { $sum: "$amount" },
    count: { $sum: 1 }
  }},
  { $sort: { total: -1 } },
  { $limit: 10 }
]);
```

### Lookup (Join)

```
db.orders.aggregate([
  { $lookup: {
    from: "customers",
    localField: "customerId",
    foreignField: "_id",
    as: "customer"
  }},
  { $unwind: "$customer" }
]);
```

## Indexes

### Tạo Indexes

```
db.users.createIndex({ email: 1 }); // ascending
db.users.createIndex({ age: -1 }); // descending
db.users.createIndex({ email: 1 }, { unique: true });
db.users.createIndex({ name: "text" }); // text search
db.places.createIndex({ location: "2dsphere" });
```

### Quản Lý Index

```
db.users.getIndexes();
db.users.dropIndex("email_1");
db.users.explain("executionStats").find({ email: "..." });
```

### Loại Index

<b>Single field</b>	Một trường, tăng/giảm dần
<b>Compound</b>	Nhiều trường (thứ tự quan trọng)
<b>Multikey</b>	Index trên trường mảng
<b>Text</b>	Tìm kiếm văn bản toàn văn
<b>Geospatial</b>	Truy vấn dữ liệu địa lý
<b>Hashed</b>	Dùng cho sharding

## Thiết Kế Schema

### Nhúng vs Tham Chiếu

<b>Nhúng (Embedding)</b>	Dùng khi dữ liệu luôn được đọc cùng nhau
<b>Tham chiếu (Reference)</b>	Dùng khi dữ liệu lớn hoặc được chia sẻ
<b>Mảng nhúng</b>	Tốt cho danh sách nhỏ, ổn định
<b>\$lookup</b>	Join tại query-time (hiệu suất thấp hơn)

### Mẫu Phổ Biến

```
// Embedded (one-to-few)
{ name: "Alice", address: { city: "NYC" } }

// Referenced (one-to-many)
{ _id: orderId, userId: ObjectId("...") }

// Bucket pattern (time-series)
{ sensor: "t1", date: ISODate("2024-01-01"),
  readings: [ { t: ..., v: ... } ] }
```

### Mẫu Phổ Biến

#### Phân Trang

```
const page = 2, limit = 10;
db.posts.find({})
  .sort({ createdAt: -1 })
  .skip((page - 1) * limit)
  .limit(limit);
```

### Tìm Kiếm Văn Bản

```
db.articles.createIndex({ title: "text", body: "text" });
db.articles.find({ $text: { $search: "mongodb query" } },
  { score: { $meta: "textScore" } })
  .sort({ score: { $meta: "textScore" } });
```

### Thống Kê & Thông Tin

```
db.users.countDocuments({ active: true });
db.users.estimatedDocumentCount();
db.stats();
db.users.stats();
```