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

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

Logic

`$and`	Tất cả điều kiện phải đúng
`$or`	Ít nhất một điều kiện đúng
`$not`	Phủ định biểu thức
`$nor`	Không điều kiện nào đúng

Mảng & Element

`$all`	Mảng chứa tất cả giá trị
`$elemMatch`	Phần tử trong mảng thỏa tất cả điều kiện
`$size`	Mảng có kích thước cụ thể
`$regex`	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

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

Cập Nhật Mảng

`$push`	Thêm phần tử vào mảng
`$pull`	Xóa phần tử khỏi mảng
`$addToSet`	Thêm nếu chưa có
`$pop`	Xóa phần tử đầu (-1) hoặc cuối (1)
`$each`	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

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

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

Thiết Kế Schema

Nhúng vs Tham Chiếu

`Nhúng (Embedding)`	Dùng khi dữ liệu luôn được đọc cùng nhau
`Tham chiếu (Reference)`	Dùng khi dữ liệu lớn hoặc được chia sẻ
`Mảng nhúng`	Tốt cho danh sách nhỏ, ổn định
`$lookup`	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();
```