MongoDB Pipeline: Hướng dẫn và Case Study

Tác giả: Đặng Kim Thi

Giới thiệu về Pipeline trong MongoDB

Pipeline trong MongoDB là một công cụ mạnh mẽ của Aggregation Framework, cho phép thực hiện các thao tác xử lý và phân tích dữ liệu theo từng giai đoạn (stages). Mỗi giai đoạn thực hiện một nhiệm vụ cụ thể như lọc dữ liệu, chuyển đổi định dạng, nhóm dữ liệu, tính toán hoặc sắp xếp. Pipeline hoạt động theo mô hình **đầu vào - đầu ra**, trong đó đầu ra của một giai đoạn là đầu vào cho giai đoạn tiếp theo.

Các giai đoạn chính trong Pipeline

1. **$match**: Lọc dữ liệu theo điều kiện.
2. **$project**: Chọn trường cần hiển thị hoặc tạo trường mới từ dữ liệu hiện có.
3. **$group**: Nhóm dữ liệu dựa trên một hoặc nhiều trường, đồng thời tính toán các giá trị như tổng, trung bình, đếm số lượng.
4. **$sort**: Sắp xếp dữ liệu theo thứ tự tăng dần hoặc giảm dần.
5. **$limit**: Giới hạn số lượng tài liệu trong kết quả.
6. **$skip**: Bỏ qua một số tài liệu đầu tiên.
7. **$unwind**: Tách các phần tử trong mảng thành các tài liệu riêng biệt.
8. **$lookup**: Thực hiện phép nối (join) giữa các collection.

Case Study 1: Quản lý điểm sinh viên

1. Tạo cơ sở dữ liệu và chèn dữ liệu

```
// Tạo cơ sở dữ liệu và collection
use Student;

db.Stud_mark.insertMany([
    {
        "name": "Adam",
        "gender": "M",
        "subjects": ["Java", "C", "Python"],
        "marks": [89, 78, 90],
        "average": 85.6
    },
    {
        "name": "Franklin",
        "gender": "M",
        "subjects": ["C", "VB", "Python"],
        "marks": [78, 85, 89],
        "average": 84
    },
    {
```

```
        "name": "Michael",
        "gender": "M",
        "subjects": ["Java", "PHP"],
        "marks": [88, 89],
        "average": 88.5
    },
    {
        "name": "Amelia",
        "gender": "F",
        "subjects": ["Ruby", "C++"],
        "marks": [86, 87],
        "average": 86.5
    }
]);
```

2. Các truy vấn minh họa

Tìm kiếm dữ liệu

```
// 1. Tìm sinh viên có điểm trung bình là 84
console.log("Sinh viên có điểm trung bình là 84:");
db.Stud_mark.find({ average: 84 }).pretty();

// 2. Tìm sinh viên có điểm trung bình lớn hơn 85
console.log("Sinh viên có điểm trung bình lớn hơn 85:");
db.Stud_mark.find({ average: { $gt: 85 } }).pretty();

// 3. Tìm sinh viên có điểm trung bình trong khoảng từ 87 đến 90
console.log("Sinh viên có điểm trung bình từ 87 đến 90:");
db.Stud_mark.find({ average: { $gte: 87, $lte: 90 } }).pretty();
```

Cập nhật dữ liệu

```
// 4. Thêm trường "Date_of_exam" cho Amelia
console.log("Thêm trường Date_of_exam cho Amelia:");
db.Stud_mark.updateOne(
    { name: "Amelia" },
    { $set: { Date_of_exam: new Date() } }
);

// 5. Tăng điểm trung bình của Franklin thêm 2
console.log("Tăng điểm trung bình của Franklin thêm 2:");
db.Stud_mark.updateOne(
    { name: "Franklin" },
    { $inc: { average: 2 } }
);

// 6. Đổi tên trường "Date_of_exam" thành "Examination_date"
```

```
console.log("Đổi tên trường Date_of_exam thành Examination_date:");
db.Stud_mark.updateMany(
    {},
    { $rename: { Date_of_exam: "Examination_date" } }
);
```

Case Study 2: Phân tích doanh thu bán hàng theo tháng

1. Tạo cơ sở dữ liệu và chèn dữ liệu

```
// Tạo cơ sở dữ liệu và collection
use Sales;

db.sales.insertMany([
    {
        "_id": 1,
        "date": "2025-01-01",
        "product": "Product A",
        "category": "Electronics",
        "quantity": 2,
        "price": 500
    },
    {
        "_id": 2,
        "date": "2025-01-02",
        "product": "Product B",
        "category": "Furniture",
        "quantity": 1,
        "price": 1500
    },
    {
        "_id": 3,
        "date": "2025-01-03",
        "product": "Product C",
        "category": "Clothing",
        "quantity": 5,
        "price": 200
    },
    {
        "_id": 4,
        "date": "2025-02-01",
        "product": "Product D",
        "category": "Electronics",
        "quantity": 1,
        "price": 1000
    },
    {
        "_id": 5,
        "date": "2025-02-02",
        "product": "Product E",
```

```
        "category": "Furniture",
        "quantity": 2,
        "price": 800
    },
    {
        "_id": 6,
        "date": "2025-02-03",
        "product": "Product F",
        "category": "Clothing",
        "quantity": 10,
        "price": 150
    }
]);
```

2. Phân tích dữ liệu bằng pipeline

```
// Pipeline phân tích doanh thu bán hàng theo tháng
console.log("Phân tích doanh thu bán hàng theo tháng:");
db.sales.aggregate([
    // Bước 1: Tạo trường "month" và "year" từ "date"
    {
        $project: {
            year: { $year: { $dateFromString: { dateString: "$date" } } },
            month: { $month: { $dateFromString: { dateString: "$date" } }
},
            category: 1,
            revenue: { $multiply: ["$quantity", "$price"] }
        }
    },
    // Bước 2: Nhóm theo "year", "month", "category" để tính tổng doanh
thu
    {
        $group: {
            _id: { year: "$year", month: "$month", category: "$category"
},
            totalRevenue: { $sum: "$revenue" }
        }
    },
    // Bước 3: Sắp xếp doanh thu giảm dần
    {
        $sort: { "totalRevenue": -1 }
    },
    // Bước 4: Nhóm theo "year" và "month" để lấy top 3 danh mục
    {
        $group: {
            _id: { year: "$_id.year", month: "$_id.month" },
            topCategories: {
                $push: {
                    category: "$_id.category",
                    revenue: "$totalRevenue"
                }
```

```
            }
        }
    },
    // Bước 5: Chỉ giữ top 3 danh mục
    {
        $project: {
            _id: 0,
            year: "$_id.year",
            month: "$_id.month",
            topCategories: { $slice: ["$topCategories", 3] }
        }
    }
]).pretty();
```

3. Mô tả từng bước của pipeline

1. **Tạo trường mới**:
    - Sử dụng `$project` để tạo các trường `year` và `month` từ trường `date`, đồng thời tính toán doanh thu (`revenue`) bằng cách nhân `quantity` với `price`.
2. **Nhóm dữ liệu**:
    - Dùng `$group` để nhóm theo `year`, `month`, và `category`, đồng thời tính tổng doanh thu (`totalRevenue`).
3. **Sắp xếp**:
    - Sắp xếp các danh mục theo `totalRevenue` giảm dần để chuẩn bị cho việc lấy top 3 danh mục.
4. **Nhóm theo tháng**:
    - Sử dụng `$group` để nhóm dữ liệu theo `year` và `month`, đồng thời lưu danh sách danh mục cùng doanh thu.
5. **Giữ top 3 danh mục**:
    - Sử dụng `$project` và `$slice` để chỉ giữ lại 3 danh mục có doanh thu cao nhất mỗi tháng.

Kết luận

Pipeline trong MongoDB giúp xử lý và phân tích dữ liệu một cách trực quan và hiệu quả, đặc biệt khi làm việc với dữ liệu phức tạp. Hai case study trên minh họa rõ ràng cách áp dụng các giai đoạn của pipeline vào các bài toán thực tế.