MongoDB Pipeline: Hướng dẫn chi tiết

Tác giả: Đặng Kim Thi

Giới thiệu

Pipeline trong MongoDB là một công cụ mạnh mẽ trong **Aggregation Framework**, cho phép xử lý và phân tích dữ liệu theo từng bước (giai đoạn). Mỗi giai đoạn trong pipeline thực hiện một thao tác cụ thể, chẳng hạn như lọc, nhóm, sắp xếp hoặc chuyển đổi dữ liệu. Pipeline hoạt động như một chuỗi xử lý, trong đó đầu ra của một giai đoạn là đầu vào của giai đoạn tiếp theo. Điều này giúp bạn thực hiện các phân tích phức tạp một cách hiệu quả và linh hoạt.

Lợi ích của Pipeline

- **Xử lý nhiều bước trong một truy vấn duy nhất**: Giảm chi phí I/O bằng cách xử lý dữ liệu hoàn toàn tại server.
- **Khả năng tích hợp mạnh mẽ**: Cho phép thực hiện các thao tác phức tạp như tính toán, nhóm, sắp xếp và xử lý dữ liệu nhúng hoặc mảng.
- **Hiệu suất cao**: Kết hợp nhiều giai đoạn để tối ưu hóa quy trình phân tích dữ liệu.
- **Tương thích với dữ liệu phức tạp**: Hỗ trợ các cấu trúc như mảng và tài liệu nhúng.

Các giai đoạn chính trong Pipeline

1. **$match**: Lọc tài liệu theo điều kiện (tương tự truy vấn `find`).
2. **$project**: Định hình dữ liệu, chọn trường hoặc tạo trường mới.
3. **$group**: Nhóm dữ liệu theo khóa và thực hiện các phép tính tích lũy.
4. **$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. **$unwind**: Phân tách các mảng thành tài liệu riêng biệt.
8. **$lookup**: Thực hiện phép nối (join) giữa các collection.
9. **$addFields**: Thêm các trường mới vào kết quả.
10. **$out**: Ghi kết quả vào một collection khác.

Ví dụ minh họa

Bài toán

Bạn có một collection `stocks` chứa dữ liệu giao dịch chứng khoán như sau:

```
{
  "_id": 1,
  "trader": "Minh",
  "transactions": [
```

```
    {"symbol": "VNM", "price": 75, "quantity": 100},
    {"symbol": "FPT", "price": 85, "quantity": 50}
  ],
  "tradeDate": "2025-01-01"
},
{
  "_id": 2,
  "trader": "Lan",
  "transactions": [
    {"symbol": "VCB", "price": 95, "quantity": 30},
    {"symbol": "BID", "price": 40, "quantity": 100}
  ],
  "tradeDate": "2025-01-02"
}
```

Yêu cầu

1. **Tính tổng giá trị giao dịch của mỗi nhà giao dịch**: Mỗi giao dịch sẽ được tính bằng công thức `price * quantity`. Tổng giá trị giao dịch của nhà giao dịch là tổng giá trị của tất cả các giao dịch của họ.
2. **Lọc các nhà giao dịch có tổng giá trị giao dịch lớn hơn 5.000 VNĐ**: Chỉ quan tâm đến các nhà giao dịch có giá trị giao dịch cao.
3. **Sắp xếp kết quả theo tổng giá trị giao dịch giảm dần**: Hiển thị nhà giao dịch nào giao dịch nhiều nhất trước.

Pipeline

```
db.stocks.aggregate([
  {
    $unwind: "$transactions" // Tách từng giao dịch trong mảng
transactions thành tài liệu riêng biệt
  },
  {
    $group: {
      _id: "$trader",
      totalValue: { $sum: { $multiply: ["$transactions.price",
"$transactions.quantity"] } }
    }
  },
  {
    $match: { totalValue: { $gt: 5000 } } // Lọc các nhà giao dịch có tổng
giá trị > 5000
  },
  {
    $sort: { totalValue: -1 } // Sắp xếp theo tổng giá trị giao dịch giảm
dần
  }
]);
```

Kết quả

```
[
  {
    "_id": "Minh",
    "totalValue": 9250
  },
  {
    "_id": "Lan",
    "totalValue": 6200
  }
]
```

Mô tả nghiệp vụ từng bước

1. **Tách từng giao dịch riêng biệt**:
   - Dữ liệu ban đầu có cấu trúc mảng `transactions`, mỗi nhà giao dịch có nhiều giao dịch.
   - `$unwind` tách các phần tử trong mảng `transactions` thành các tài liệu riêng lẻ.
2. **Tính tổng giá trị giao dịch của từng nhà giao dịch**:
   - Sử dụng `$group` để nhóm các giao dịch theo `trader`.
   - Với mỗi nhà giao dịch, tính tổng giá trị bằng `$sum` và toán tử `$multiply` để nhân `price` và `quantity` của từng giao dịch.
3. **Lọc các nhà giao dịch có tổng giá trị giao dịch lớn hơn 5.000 VNĐ**:
   - Sử dụng `$match` để chỉ giữ lại các tài liệu có `totalValue` lớn hơn 5.000.
4. **Sắp xếp theo tổng giá trị giao dịch**:
   - Sử dụng `$sort` để sắp xếp kết quả theo `totalValue` giảm dần, nhà giao dịch có giá trị giao dịch lớn nhất sẽ xuất hiện trước.

Kết luận

Pipeline trong MongoDB là một công cụ mạnh mẽ để phân tích dữ liệu phức tạp. So với các truy vấn cơ bản, nó không chỉ mạnh mẽ hơn mà còn linh hoạt và hiệu quả hơn. Bằng cách tận dụng các giai đoạn như `$match`, `$group`, `$unwind` và `$project`, bạn có thể dễ dàng xây dựng các quy trình phân tích dữ liệu toàn diện chỉ với một truy vấn duy nhất.