# Projection trong MongoDB

Tác giả: **Đặng Kim Thi**

Projection trong MongoDB là kỹ thuật được sử dụng để **chọn các trường hoặc phần tử cần hiển thị trong kết quả truy vấn**, thay vì trả về toàn bộ tài liệu. Điều này giúp tối ưu hóa hiệu năng khi làm việc với dữ liệu lớn.

## Các toán tử Projection chính

| Toán tử | Ý nghĩa | Ví dụ minh họa | Kết quả |
|---|---|---|---|
| `$` | Trả về phần tử đầu tiên trong mảng thỏa mãn điều kiện trong truy vấn. Nếu không có điều kiện, trả về phần tử đầu tiên của mảng. | **Truy vấn:** `db.collection.find({ field: value }, { arrayField: { $: 1 } })` **Dữ liệu:** `arrayField: [1, 2, 3, 4]` | `arrayField: [1]` |
| `$elemMatch` | Trả về phần tử đầu tiên trong mảng thỏa mãn điều kiện được xác định bằng toán tử `$elemMatch`. | **Truy vấn:** `db.collection.find({}, { arrayField: { $elemMatch: { field: condition } } })` | Trả về **phần tử đầu tiên** phù hợp với điều kiện trong mảng. |
| `$meta` | Lấy metadata liên quan đến tài liệu, như điểm tìm kiếm văn bản. | **Truy vấn:** `db.collection.find({}, { score: { $meta: "textScore" } })` | Trả về điểm tìm kiếm của các tài liệu, ví dụ: `score: 3.25`. |
| `$slice` | Giới hạn số phần tử trả về từ mảng trong tài liệu. | **Truy vấn:** `db.collection.find({}, { arrayField: { $slice: 2 } })` | Trả về 2 phần tử đầu tiên của mảng `arrayField`. |

## Ví dụ minh họa

1. Toán tử `$`

- **Dữ liệu mẫu**:

```
{ "name": "John", "scores": [90, 85, 78, 92] }
```

- **Truy vấn**:

```
db.students.find({ name: "John" }, { scores: { $: 1 } });
```

- **Kết quả**:

```
{ "name": "John", "scores": [90] }
```

2. Toán tử `$elemMatch`

- **Dữ liệu mẫu**:

```
{ "name": "Jane", "scores": [{ "subject": "Math", "score": 85 }, {
"subject": "English", "score": 90 }] }
```

- **Truy vấn**:

```
db.students.find({ name: "Jane" }, { scores: { $elemMatch: { subject:
"Math" } } });
```

- **Kết quả**:

```
{ "name": "Jane", "scores": [{ "subject": "Math", "score": 85 }] }
```

3. Toán tử `$meta`

- **Dữ liệu mẫu**: Dùng trong tìm kiếm văn bản.
- **Truy vấn**:

```
db.articles.find({ $text: { $search: "MongoDB" } }, { score: { $meta:
"textScore" } });
```

- **Kết quả**:

```
{ "title": "Introduction to MongoDB", "score": 3.5 }
```

4. Toán tử `$slice`

- **Dữ liệu mẫu**:

```
{ "name": "Sara", "tasks": ["Task1", "Task2", "Task3", "Task4"] }
```

- **Truy vấn**:

```
db.employees.find({ name: "Sara" }, { tasks: { $slice: 2 } });
```

- **Kết quả**:

```
{ "name": "Sara", "tasks": ["Task1", "Task2"] }
```

---

# Bài tập thực hành

## Bài tập 1

Dữ liệu mẫu:

```
[
  { "name": "Alice", "hobbies": ["Reading", "Traveling", "Swimming"] },
  { "name": "Bob", "hobbies": ["Gaming", "Cooking", "Running"] }
]
```

**Yêu cầu**:

1. Sử dụng toán tử `$` để lấy sở thích đầu tiên của Alice.
2. Sử dụng toán tử `$slice` để lấy 2 sở thích đầu tiên của Bob.

---

## Bài tập 2

Dữ liệu mẫu:

```
[
  { "title": "MongoDB Basics", "tags": ["database", "NoSQL", "MongoDB"] },
  { "title": "Advanced MongoDB", "tags": ["database", "MongoDB",
"performance"] }
]
```

**Yêu cầu**:

1. Sử dụng toán tử `$elemMatch` để tìm bài viết có tag `"NoSQL"`.
2. Sử dụng toán tử `$slice` để giới hạn kết quả tags chỉ còn 2 phần tử.

---

### Bài tập 3

Dữ liệu mẫu:

```
[
  { "name": "Mike", "scores": [80, 85, 90] },
  { "name": "Lucy", "scores": [70, 75, 80] }
]
```

**Yêu cầu**:

1. Sử dụng toán tử `$` để lấy điểm đầu tiên của Mike.
2. Sử dụng toán tử `$slice` để lấy 2 điểm đầu tiên của Lucy.

---

Nếu cần giải chi tiết cho các bài tập trên, bạn có thể hỏi Cô giáo thêm nhé! 😊