Giới thiệu về 2dsphere Index trong MongoDB

Tác giả: Đặng Kim Thi

1. 2dsphere Index là gì?

2dsphere Index là một loại index trong MongoDB, thiết kế để hỗ trợ truy vấn không gian hai chiều (2D) trên dữ liệu địa lý. Loại index này cho phép MongoDB xác định một cách nhanh chóng các điểm (địa điểm, đường, vùng) trên bề mặt hình cầu (như Trái đất).

MongoDB sử dụng 2dsphere Index để:

1. **Tìm kiếm địa lý (Geospatial Queries):** Tìm các điểm gần nhau hoặc nằm trong một khu vực.
2. **Kiểm tra không gian (Spatial Intersections):** Xác định xem một đối tượng địa lý nào đang giao nhau hoặc nằm trong một khu vực nhất định.
3. **Tối ưu truy vấn dữ liệu địa lý:** Giảm thiểu thời gian xử lý truy vấn trong các bộ sưu tập lớn.

---

2. Tại sao cần 2dsphere Index?

Dữ liệu địa lý khác biệt với dữ liệu tường minh (tabular). Khi truy vấn trên dữ liệu địa lý:

- **Truy vấn thông thường:** MongoDB phải duyệt qua tất cả các tài liệu, dẫn đến hiệu suất thấp.
- **Truy vấn không gian:** Tối ưu truy vấn theo không gian đòi hỏi một công cục có thể hiểu được các đối tượng địa lý như điểm, đường, vùng. **2dsphere Index** cung cấp các khả năng này, giúp MongoDB hiểu và xử lý nhanh các truy vấn địa lý phức tạp.

---

3. Cách tạo 2dsphere Index

Bước 1: Chuẩn bị dữ liệu

MongoDB hỗ trợ định dạng GeoJSON cho dữ liệu địa lý. Cấu trúc dữ liệu mẫu:

```
{
  "name": "Location A",
  "location": {
    "type": "Point",
    "coordinates": [longitude, latitude]
  }
}
```

- `type`: Kiểu hình học (điểm, đường, vùng, v.v.).
- `coordinates`: Mảng gồm kinh độ (*longitude*) và vĩ độ (*latitude*).

Bước 2: Tạo Index

Tạo index 2dsphere trên trường địa lý:

```
db.collectionName.createIndex({ location: "2dsphere" });
```

- `collectionName`: Tên bộ sưu tập.
- `location`: Tên trường chứa dữ liệu địa lý.
- `"2dsphere"`: Loại index dành cho dữ liệu địa lý.

Bước 3: Chạy truy vấn địa lý

1. **Tìm các điểm gần nhất**:

```
db.collectionName.find({
  location: {
    $near: {
      $geometry: { type: "Point", coordinates: [longitude, latitude]
},
      $maxDistance: 1000 // Khoảng cách tối đa (mét)
    }
  }
});
```

2. **Tìm các điểm trong một khu vực**:

```
db.collectionName.find({
  location: {
    $geoWithin: {
      $geometry: {
        type: "Polygon",
        coordinates: [
          [
            [long1, lat1],
            [long2, lat2],
            [long3, lat3],
            [long1, lat1] // Điểm đầu phải trùng điểm cuối
          ]
        ]
      }
    }
  }
});
```

---

4. Ví dụ minh họa

Dữ liệu mẫu

Bộ sưu tập `places` chứa dữ liệu địa lý:

```
db.places.insertMany([
  { name: "Park", location: { type: "Point", coordinates: [106.6297,
10.8231] } }, // TPHCM
  { name: "Mall", location: { type: "Point", coordinates: [106.6652,
10.7626] } }, // Landmark 81
  { name: "Cafe", location: { type: "Point", coordinates: [106.7009,
10.7758] } }  // Nhà thờ Đức Bà
]);
```

Tạo Index

```
db.places.createIndex({ location: "2dsphere" });
```

Truy vấn gần nhất

Tìm địa điểm gần Nhà thờ Đức Bà trong bán kính 2 km:

```
db.places.find({
  location: {
    $near: {
      $geometry: { type: "Point", coordinates: [106.7009, 10.7758] },
      $maxDistance: 2000 // 2 km
    }
  }
});
```

Truy vấn trong khu vực

Tìm địa điểm nằm trong một tam giác:

```
db.places.find({
  location: {
    $geoWithin: {
      $geometry: {
        type: "Polygon",
        coordinates: [
          [
            [106.6, 10.7],
            [106.7, 10.8],
            [106.8, 10.7],
            [106.6, 10.7] // Đóng vòng
          ]
        ]
      }
    }
```

```
    }
});
```

---

5. Lưu ý khi sử dụng 2dsphere Index

1. **Tuân thủ chuẩn GeoJSON:** Trường địa lý phải có định dạng `type` và `coordinates` đúng.
2. **Tổng quan tọđộ:** Kinh độ trong khoảng `[-180, 180]`, vĩ độ trong khoảng `[-90, 90]`.
3. **Cân nhắc chi phí:** Sử dụng index gia tăng khả năng truy vấn nhưng cũng tăng chi phí bảo trì.
4. **Truy vấn đúng mục đích:** Dữ liệu không gian hai chiều nên không thịch hợp với bản đồ phẳng.

---

6. Ứng dụng thực tế

- **Ứng dụng gọi xe:** Tìm tài xế hoặc hành khách gần vị trí hiện tại.
- **Thương mại điện tử:** Tìm kho hàng hoặc cửa hàng gần nhất.
- **Bất động sản:** Xác định tài sản trong một khu vực.
- **Ứng dụng bản đồ:** Tính toán khoảng cách và đường đi.

2dsphere Index mang đến khả năng tích hợp linh hoạt các tính năng địa lý vào các ứng dụng MongoDB, đáp ứng yêu cầu khá phức tạp trong thực tế.