Иногда требуется определить, с какими документами - геометрическими фигурами - из коллекции пересекается интересующая нас фигура. Для решения такой задачи в MongoDB используется запрос $geoIntersects.

Этот запрос выбирает из коллекции документы, у которых геопространственные (geospatial) данные пересекаются с указанным объектом GeoJSON, то есть те документы, у которых пересечение геоданных с указанным объектом не является пустым. Включаются также случаи, когда данные и указанный объект имеют общую сторону.

Пример использования запроса $geoIntersects:

db.places.find(

{

'geometry':{

'$geoIntersects':{

'$geometry' : {

"type": "Polygon",

"coordinates": [

[ [ 0, 0 ], [ 3, 6 ], [ 6, 1 ], [ 0, 0 ] ]

]

}

}

}

}

)

Особенности:

При указании координат долготы и широты, сначала следует указать долготу, а затем широту.

Значения долготы должны быть в диапазоне [-180, 180]; значения широты - в диапазоне [-90, 90].

Запрос $geoIntersects использует сферическую геометрию. Он не требует геопространственного индекса. Однако геопространственный индекс улучшит выполнение запроса.

Запрос $geoIntersects поддерживается только геопространственным индексом 2dsphere.

GeoJSON

GeoJSON — открытый формат, предназначенный для хранения географических структур данных, основан на JSON.

Формат может хранить примитивные типы для описания географических объектов, такие как: точки (адреса и местоположения), линии (улицы, шоссе, границы), многоугольники (страны, штаты, участки земли). Также могут храниться так называемые мультитипы, которые представляют собой объединение нескольких примитивных типов.

Объекты GeoJSON:

Примитивные типы: Point, LineString, Polygon.

Мультитипы: MultiPoint, MultiLineString, MultiPolygon.

Пример:

{ "type": "FeatureCollection",

"features": [

{ "type": "Feature",

"geometry": {"type": "Point", "coordinates": [102.0, 0.5]},

"properties": {"prop0": "value0"}

},

{ "type": "Feature",

"geometry": {

"type": "LineString",

"coordinates": [

[102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]

]

},

"properties": {

"prop0": "value0",

"prop1": 0.0

}

},

{ "type": "Feature",

"geometry": {

"type": "Polygon",

"coordinates": [

[ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],

[100.0, 1.0], [100.0, 0.0] ]

]

},

"properties": {

"prop0": "value0",

"prop1": {"this": "that"}

}

}

]

}

$geoWithin

Запрос $geoWithin выбирает из коллекции документы, содержащие геопространственные данные, которые полностью находятся внутри указанной фигуры. При этом граница фигуры считается частью фигуры.

Фигура, внутри которой должны находиться искомые объекты, может быть следующих типов: GeoJSON Polygon, GeoJSON MultiPolygon или фигурой, определенной парами координат.

Особенности:

При указании координат долготы и широты, сначала следует указать долготу, а затем широту.

Запрос $geoWithin поддерживается геопространственными индексами 2dsphere и 2d.

Оператор $geoWithin возвращает неотсортированные результаты. Поэтому этот запрос работает быстрее, чем геопространственные запросы $near и $nearSphere, которые сортируют результаты.

Пример:

db.places.find(

{

'geometry':{

'$geoWithin':{

'$geometry':{

"type": "Polygon",

"coordinates": [ [ [ 0, 0 ], [ 3, 6 ], [ 6, 1 ], [ 0, 0 ] ] ]

}

}

}

}

)

$near

Когда требуется найти географические объекты, которые находятся рядом с указанной точкой, удобно использовать запрос $near. В этом запросе также можно указать минимальное и максимальное расстояние от выбранной точки до объекта.

Для запроса $near можно указывать точку как в формате GeoJSON, так и парой ее координат.

Запрос $near требует наличие геопространственного индекса:

индекса 2dsphere, если указывается точка в формате GeoJSON

индекса 2d, если точка указывается парой координат.

При указании точки GeoJSON можно (не обязательно) использовать характеристики $minDistance и $maxDistance. Эти характеристики ограничивают результаты выполнения запроса расстоянием в метрах.

$minDistance ограничивает результаты до тех документов, которые находятся как минимум на указанном расстоянии от выбранной точки (то есть не ближе). Опция $minDistance доступна только при использовании индекса 2dsphere.

$maxDistance ограничивает результаты до тех документов, которые находятся не дальше указанного расстояния от выбранной точки.

Запрос $near сортирует результаты по расстоянию от указанной точки.

Пример (с указанием точки GeoJSON):

db.places.find(

{

'loc':

{ $near :

{

$geometry: {

type: "Point",

coordinates: [ -73.9667, 40.78 ]

},

$minDistance: 1000,

$maxDistance: 5000

}

}

}

)