

Методы сбора, хранения, обработки и анализа данных

Лекция 15

СУБД MongoDB – индексирование

Индекс

- Что такое индекс?
- Каким образом применяется?
- В каких случаях создается автоматически?

Индекс

- Что получаем при добавлении индекса?
- Чем за это платим?
- Для каких объектов можно создать индекс?
- Для каких столбцов создаем индекс?

Индекс

- Что такое простой индекс?
- Что такое составной индекс?
- Что такое уникальный / не уникальный индекс?
- Какой порядок значений (ASC, DESC)?
- Что такое индекс покрытия?
- Что такое фильтрующий индекс?

Интеграция индекса с таблицей

- Что такое кластеризованные индексы?
- Сколько может быть кластеризованных индексов?
- Что такое некластеризованные индексы?
- Сколько может быть некластеризованных индексов?
- Какой индекс работает быстрее – кластеризованный или некластеризованный?

Эффективность индекса

- Эффективность индекса оценивает оптимизатор запросов, основываясь на характеристиках запроса:
 - Селективность
 - Плотность
 - Распределение значений

Селективность

- Селективность – отношение числа выбираемых записей к общему числу записей
- Чем выше селективность – чем больше записей выбирается – тем хуже

Плотность

- Плотность – отношение числа дубликатов значений к общему числу значений
- Лучшая плотность – уникальные значения

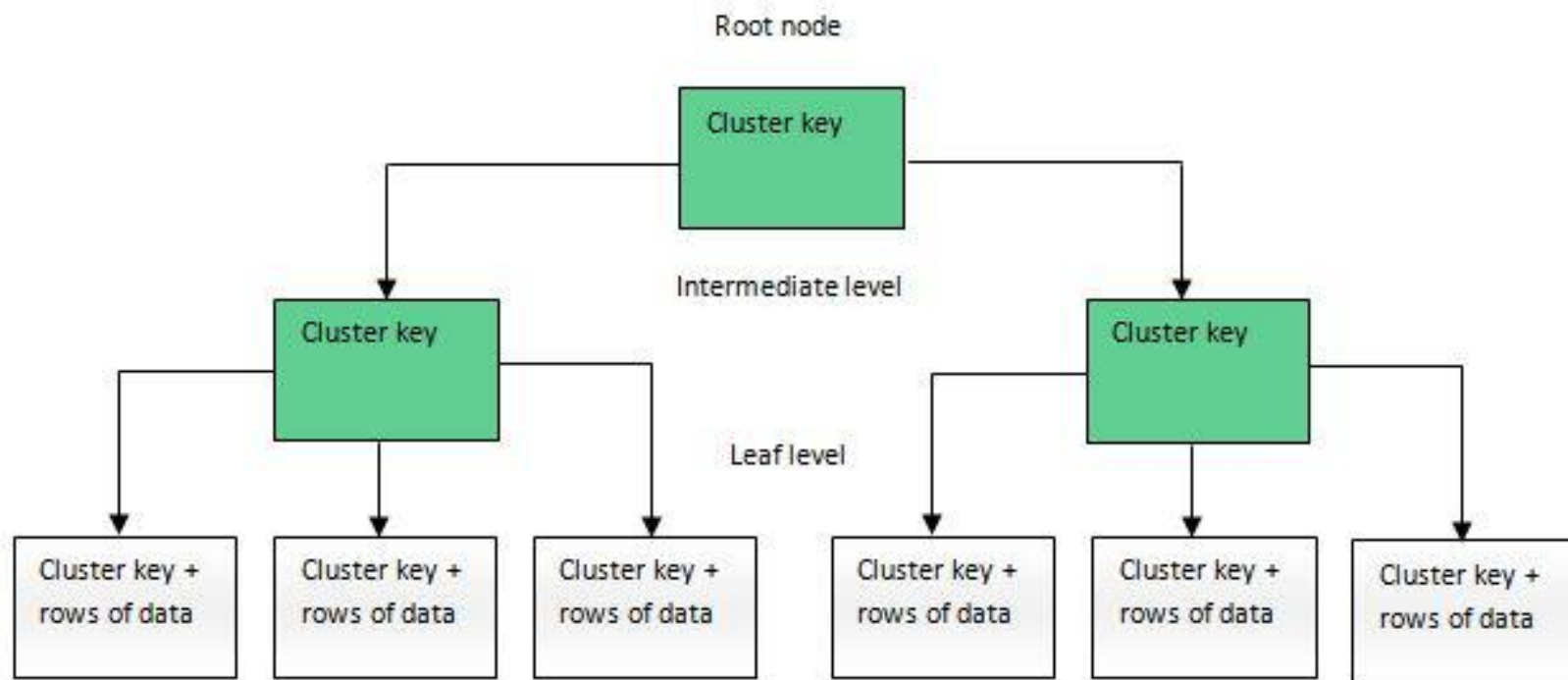
Распределение значений

- Распределение значений – показатель, как значения ключей индекса распределены по всему интервалу возможных значений
- Больше слов на букву А, чем на букву Й

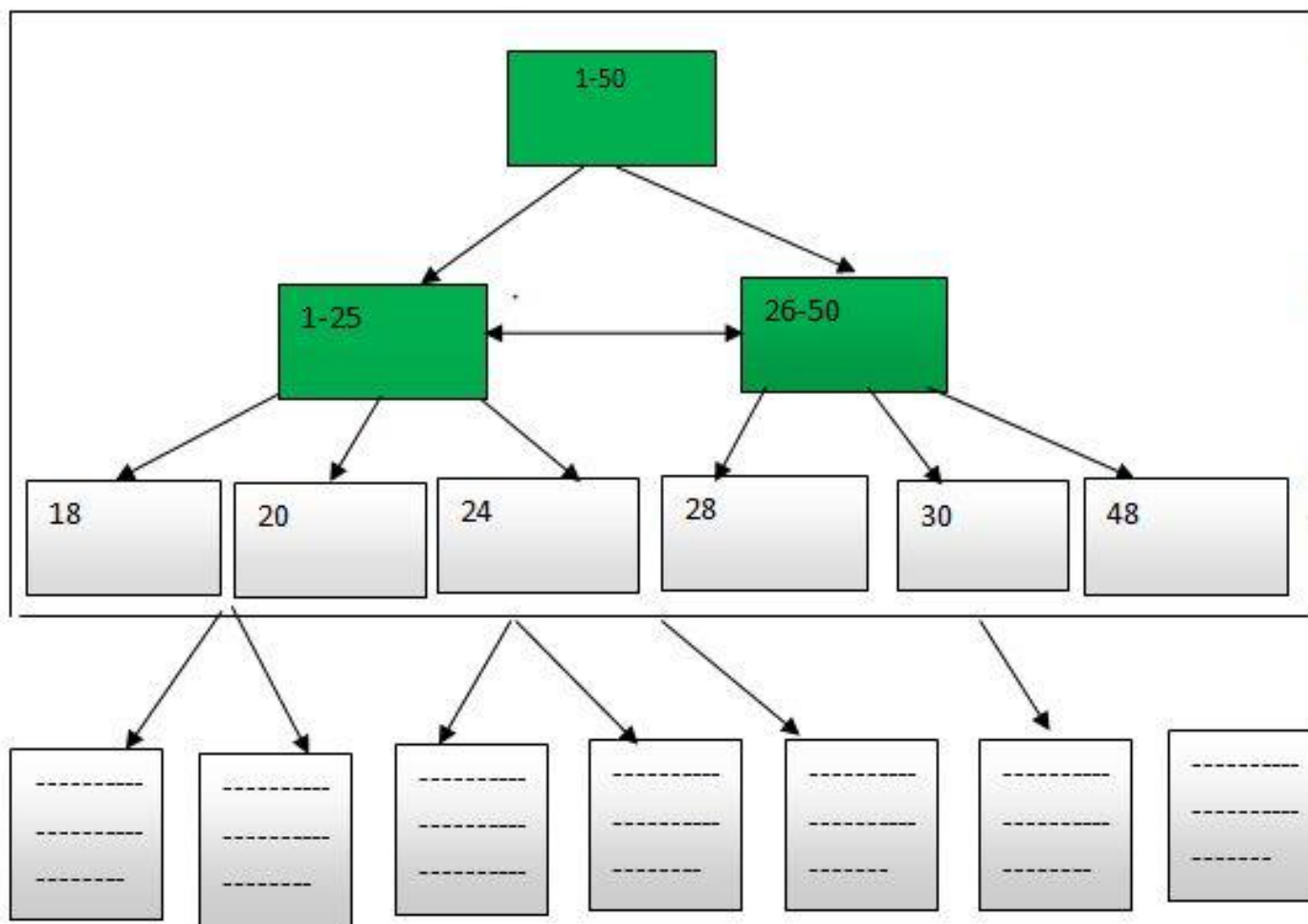
Индекс

- Структура индекса:
 - деревья (tree-index)
 - частичные
 - функциональные
 - двоичные таблицы (bitmap, Oracle)
 - пространственные индексы (spatial)
 - полнотекстовые индексы (full text)
 - XML-индексы
 - колоночные индексы (для OLAP-приложений)

B-tree кластеризованный индекс



B-tree некластеризованный индекс

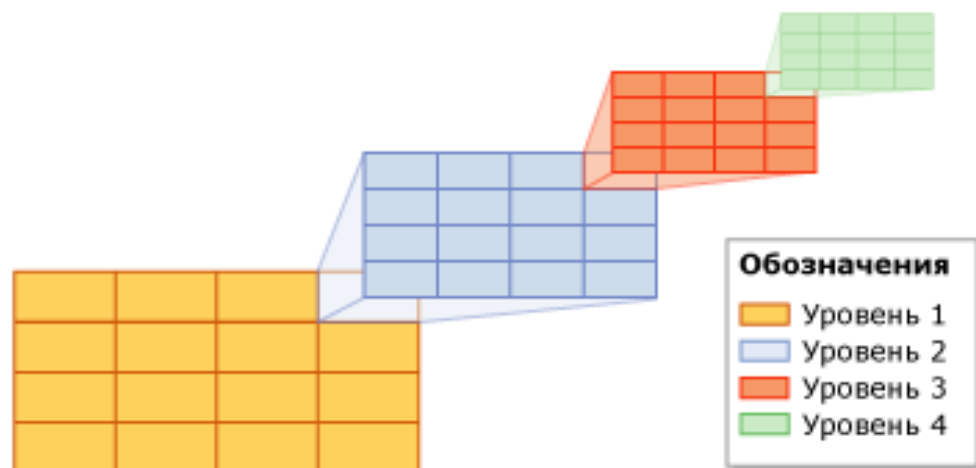


Битовые индексы в Oracle

- Битовые карты:
 - Строк — столько, сколько значений
 - Длина строки — столько, сколько строк в таблице
- Предназначен для индексирования столбцов с набором значений
- Не подходит для таблиц с частым обновлением
- Хорошо подходят для хранилищ данных

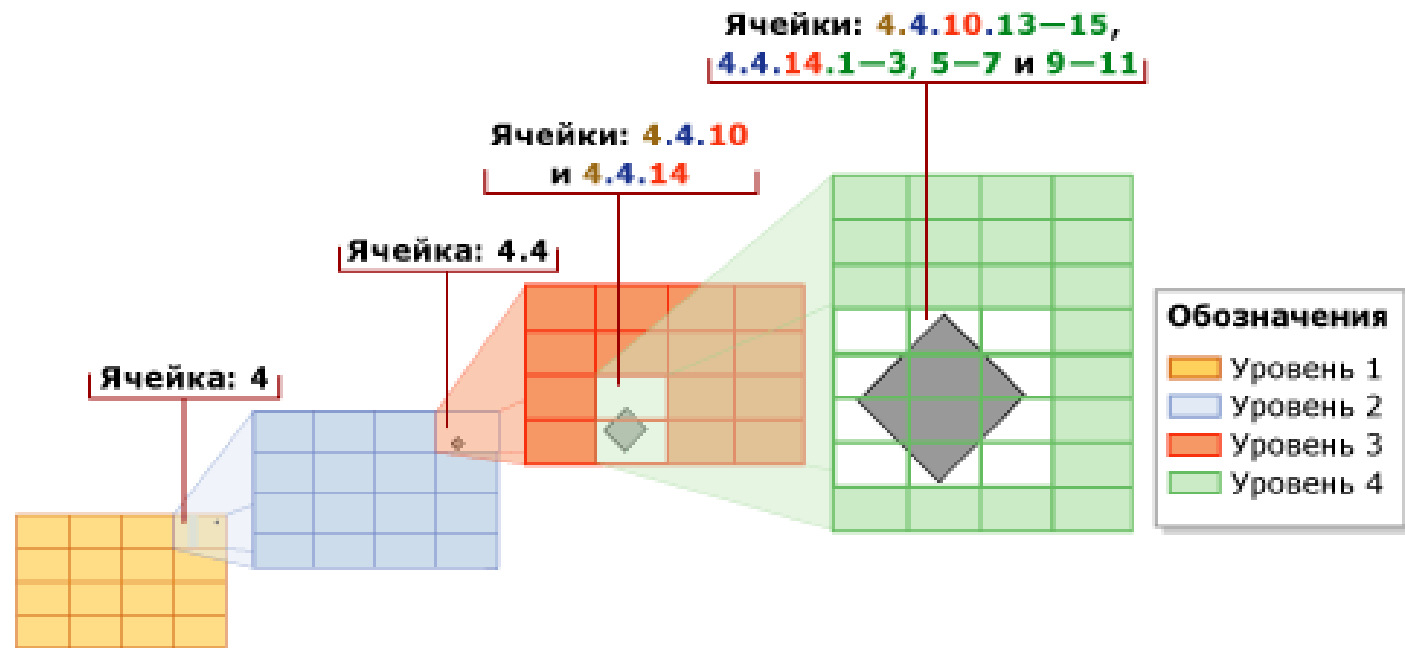
Пространственные индексы в MS SQL Server

```
CREATE SPATIAL INDEX i_spatial_shape  
ON market(shape)  
USING GEOMETRY_GRID  
WITH (BOUNDING_BOX = (xmin=0, ymin=0, xmax=500, ymax=200),  
GRIDS = (LOW, LOW, MEDIUM, HIGH),  
PAD_INDEX = ON);
```



Ключевое слово	Конфигурация сетки	Число ячеек
LOW	4X4	16
MEDIUM	8X8	64
HIGH	16X16	256

Тесселяция



Полнотекстовые индексы в MS SQL Server

1	Полнотекстовый индекс строится на основании
2	Пространственный индекс используется
3	Битовый индекс применяется
4	Кластеризованный и некластеризованный индекс

1	полнотекстовый
2	пространственный
3	битовый
4	кластеризованный
4	некластеризованный
1, 2, 3, 4	индекс
1	строится
2	применяется

Полнотекстовые индексы в MS SQL Server

- Средства разбиения по словам и парадигматические модули
- Списки стоп-слов
- Файлы тезауруса
- Фильтры

XML индексы в MS SQL Server

- Первичный XML-индекс:
 - Индексируются все теги, значения и пути
 - Используется для возвращения скалярных значений или поддеревьев
- Вторичные XML-индексы:
 - FOR PATH — по структуре документа
 - FOR VALUE — по значениям элементов и атрибутов столбца XML
 - FOR PROPERTY — для поиска по свойствам
- Не могут быть составными
- Не могут быть кластеризованными

Индексы XML в MS SQL Server

```
CREATE PRIMARY XML INDEX index_xml_column ON xmltab(xml_column);
```

```
CREATE XML INDEX i_xmlcolumn_path ON xmltab(xml_column)  
USING XML INDEX index_xml_column FOR PATH;
```

Колоночные индексы в MS SQL Server

- Данные хранятся по столбцам
- В индекс включаются столбцы, по которым будет производиться поиск
- Позволяют получить значительный выигрыш в производительности для больших массивов данных



Индекс

- Что такое фрагментация индекса?
- В чем заключается обслуживание индекса?

Индекс

- Как принимается решение о применении индекса?
- Каковы критерии принятия решения?
- Как обеспечить использование индекса?
- Что такое hints?
- Как оценить использование индекса?

Hints

- Hints – подсказки оптимизации являются частью инструкции SELECT, которые указывают оптимизатору запросов, что нужно выполнять данную инструкцию определенным образом
- Использовать определенный индекс, способ соединения или для специфической выдачи данных

```
SELECT * FROM AUDITORIUM WITH (INDEX(cs_index1))  
WHERE AUDITORIUM = '461-2';
```

```
SELECT * FROM FACULTY JOIN PULPIT  
ON FACULTY.FACULTY = PULPIT.FACULTY  
AND FACULTY.FACULTY = 'ИдиП'  
OPTION (FORCE ORDER);
```

```
SELECT * FROM TEACHER OPTION (FAST 10);
```

Добавление данных

- Добавим в коллекцию 1 000 000 элементов

```
for (i=0; i<1000000; i++) {  
    db.dev_subjects.insertOne(  
        {  
            "id" : i,  
            "subject_name" : "Subject_" + i,  
            "semester" : Math.floor(Math.random()*8),  
            "number_of_lectures" : Math.floor(Math.random()*18),  
            "number_of_labs" : Math.floor(Math.random()*28)}});  
}
```

```
{ acknowledged: true,  
  insertedId: ObjectId("67d549039cf06cf6985c75a5") }
```

```
> db.dev_subjects.countDocuments()  
< 1000006
```


Добавление данных

- Добавим в коллекцию 100 000 элементов:
 - Подготовить js-файл
 - Запустить консольное приложение mongosh
 - Загрузить файл в mongosh

insert.js – Блокнот

Файл Правка Формат Вид Справка

```
for (i=0; i<100000; i++) {  
    db.db_subjects.insertOne(  
        {  
            "subject_id" : i,  
            "subject_name" : "DB_Subject_" + i,  
            "semester" : Math.floor(Math.random()*7),  
            "number_of_lectures" : Math.floor(Math.random()*28),  
            "number_of_labs" : Math.floor(Math.random()*36)}});  
}
```

```
all_subjects> load("C:/MongoDb/bin/insert.js")  
true
```

```
> use all_subjects  
< 'switched to db all_subjects'  
> db.db_subjects.countDocuments()  
< 100000  
all_subjects> |
```

Просмотр плана запроса без индекса

```
< { explainVersion: '1',
  queryPlanner:
    { namespace: 'all_subjects.dev_subjects',
      indexFilterSet: false,
      parsedQuery: { subject_name: { '$eq': 'Subject_199' } },
      queryHash: '21F2FA6E',
      planCacheKey: '21F2FA6E',
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
      maxScansToExplodeReached: false,
      winningPlan:
        { stage: 'COLLSCAN',
          filter: { subject_name: { '$eq': 'Subject_199' } },
          direction: 'forward' },
      rejectedPlans: [] },
```

```
executionStats:
  { executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 500,
    totalKeysExamined: 0,
    totalDocsExamined: 1000006,
    executionStages:
      { stage: 'COLLSCAN',
        filter: { subject_name: { '$eq': 'Subject_199' } },
        nReturned: 1,
        executionTimeMillisEstimate: 17,
        works: 1000008,
        advanced: 1,
        needTime: 1000006,
        needYield: 0,
        saveState: 1000,
        restoreState: 1000,
        isEOF: 1,
        direction: 'forward',
        docsExamined: 1000006 } },
```

```
> db.dev_subjects.find({"subject_name": "Subject_199"}).explain("executionStats")
```

Просмотр плана запроса с ИНДЕКСОМ

```
> db.dev_subjects.createIndex({"subject_name" : 1})  
< 'subject_name_1'
```

```
db.dev_subjects.find({"subject_name": "Subject_199"}).explain("executionStats")  
< { explainVersion: '1',  
  queryPlanner:  
    { namespace: 'all_subjects.dev_subjects',  
      indexFilterSet: false,  
      parsedQuery: { subject_name: { '$eq': 'Subject_199' } },  
      queryHash: '21F2FA6E',  
      planCacheKey: 'B61E8823',  
      maxIndexedOrSolutionsReached: false,  
      maxIndexedAndSolutionsReached: false,  
      maxScansToExplodeReached: false,  
      winningPlan:  
        { stage: 'FETCH',  
          inputStage:  
            { stage: 'IXSCAN',  
              keyPattern: { subject_name: 1 },  
              indexName: 'subject_name_1',  
              isMultiKey: false,  
              multiKeyPaths: { subject_name: [ ] },  
              isUnique: false,  
              isSparse: false,  
              isPartial: false,  
              indexVersion: 2,  
              direction: 'forward',  
              indexBounds: { subject_name: [ '['Subject_199', 'Subject_199']' ] } },  
              projectedFields: [ ] } } } } }
```

```
{ executionSuccess: true,  
  nReturned: 1,  
  executionTimeMillis: 0,  
  totalKeysExamined: 1,  
  totalDocsExamined: 1,  
  executionStages:  
    { stage: 'FETCH',  
      nReturned: 1,  
      executionTimeMillisEstimate: 0,  
      works: 2,  
      advanced: 1,  
      needTime: 0,  
      needYield: 0,  
      saveState: 0,  
      restoreState: 0,  
      isEOF: 1,  
      docsExamined: 1,  
      alreadyHasObj: 0,  
      inputStage:
```

Просмотр плана запроса с индексом

```
{ executionSuccess: true,
  nReturned: 1,
  executionTimeMillis: 0,
  totalKeysExamined: 1,
  totalDocsExamined: 1,
  executionStages:
    { stage: 'FETCH',
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      works: 2,
      advanced: 1,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      docsExamined: 1,
      alreadyHasObj: 0,
      inputStage:
```

```
{ stage: 'IXSCAN',
  nReturned: 1,
  executionTimeMillisEstimate: 0,
  works: 2,
  advanced: 1,
  needTime: 0,
  needYield: 0,
  saveState: 0,
  restoreState: 0,
  isEOF: 1,
  keyPattern: { subject_name: 1 },
  indexName: 'subject_name_1',
  isMultiKey: false,
  multiKeyPaths: { subject_name: [] },
  isUnique: false,
  isSparse: false,
  isPartial: false,
  indexVersion: 2,
  direction: 'forward',
  indexBounds: { subject_name: [ '"Subject_199"', '"Subject_199"' ] },
  keysExamined: 1,
  seeks: 1,
  dupsTested: 0,
  dupsDropped: 0 } }
```

Сравнение планов

- Без индекса:
 - nReturned = 1
 - totalDocsExamined = 1 000 006
 - executionTimeMillis = 500
- С индексом:
 - nReturned = 1
 - totalDocsExamined = 1
 - executionTimeMillis = 0

Просмотр индексов

all_subjects.dev_subjects

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

CREATE INDEX

Name and Definition ^

Type

id

_id ↑

REGULAR ⓘ

subject_name_1

subject_name ↑

REGULAR ⓘ

Виды индексов

- Простые индексы
- Составные индексы
- Уникальные индексы
- Индексы покрытия
- Частичные индексы

Простые индексы

- Простые индексы – на один атрибут
- Используются при поиске и сортировке
 - nReturned = 113332
 - totalDocsExamined = 113332
 - executionTimeMillis = 130

```
db.dev_subjects.find({"subject_name" : {$gt : "Subject_101", $lt : "Subject_203" }}).ex
```


Создание индекса

Create Index

Choose an index name

semester

Configure the index definition







semester

1 (asc)



ADD ANOTHER FIELD

Options

- ☐ Build index in the background 
- ☐ Create unique index 
- ☐ Create TTL 
- ☐ Partial Filter Expression 
- ☐ Use Custom Collation 
- ☐ Wildcard Projection 

CANCEL

CREATE INDEX

Создание индекса

all_subjects.dev_subjects

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

CREATE INDEX

Name and Definition ^

Type

id

id ↑

REGULAR ⓘ

semester

semester ↑

REGULAR ⓘ

subject_name_1

subject_name ↑

REGULAR ⓘ

Простые индексы

```
> db.dev_subjects.find({"subject_name" : {$gt : "Subject_109", $lt : "Subject_119"} }).sort({"semester" : 1}).explain("executionStats")
```

- Поиск или сортировка по нескольким атрибутам – простой индекс не помогает
 - nReturned = 11110
 - totalDocsExamined = 1000006
 - executionTimeMillis = 1646

Составные индексы

all_subjects.dev_subjects

Documents

Aggregations

Schema

E

CREATE INDEX

Name and Definition ^

_id

_id

semester

semester

subject_semester

subject_name

semester

subject_name_1

subject_name

Составные индексы

```
> db.dev_subjects.find({"subject_name" : {$gt : "Subject_109", $lt : "Subject_119"} }).sort({"semester" : 1}).explain("execu
```

- nReturned = 11110
- totalDocsExamined = 1000006
- executionTimeMillis = 1362

Составные индексы

all_subjects.dev_subjects

Documents Aggregations Schema E

CREATE INDEX

Name and Definition ^

id

_id ↑

semester

semester ↑

semester_subject

semester ↑ subject_name ↑

subject_name_1

subject_name ↑

subject_semester

subject_name ↑ semester ↑

Составные индексы

```
db.dev_subjects.find({"subject_name" : {$gt : "Subject_101", $lt : "Subject_203"}}).sort({"semester" : 1}).explain()
```

- nReturned = 11110
- totalDocsExamined = 1000006
- executionTimeMillis = 1473

Составные индексы

- Ключи для равенства должны быть первыми
- Ключи для сортировки должны находиться перед полями с многими значениями, причем должен учитываться порядок сортировки
- Ключи для полей с многими значениями должны быть последними

Покрывающий индекс

```
> db.dev_subjects.find({"subject_name" : {$gt : "Subject_109", $lt : "Subject_119"}} , {"subject_name" : 1, "semester" : 2}).sort({"semester" : 1})
```

- Покрывающий индекс – индекс содержит в себе все необходимые атрибуты

Использование индексов с \$

- Не использовать отрицание
- Использовать вначале равенство, потом диапазоны

Hints

```
> db.db_subjects.find({semester:{$gt:2}, number_of_lectures:20}).sort({semester:1})
```

	number_of_lectures	semester	hint
nReturned	2089	2089	2089
executionTimeMillis	17	117	14
totalDocsExamined	3588	57288	3588
totalKeysExamined	3588	57288	3588

```
db.db_subjects.find({semester:{$gt:2}, number_of_lectures:20}).sort({semester:1}).hint({number_of_lectures:1}).
```

Hints

- Используем коллекцию `db_subjects`
- Создадим индекс по `number_of_lectures` и посмотрим план
- Создадим индекс по `semester` и посмотрим план
- Стоимость увеличилась
- Как вернуться к предыдущему индексу?

Уникальные индексы

- Уникальные индексы – одно значение в индексе

Name and Definition ^	Type	Size	Usage	Properties
<div><div>_id_</div><div><div>_id_</div><div></div></div></div>	REGULAR ⓘ	1.6 MB	14 since Tue Mar 18 2025	UNIQUE ⓘ
<div><div>number_of_lectures_1</div><div><div>number_of_lectures_1</div><div></div></div></div>	REGULAR ⓘ	466.9 KB	0 since Tue Mar 18 2025	
<div><div>semester</div><div><div>semester</div><div></div></div></div>	REGULAR ⓘ	458.8 KB	0 since Tue Mar 18 2025	

Частичные индексы

- Частичные индексы – индексируют часть элементов коллекции
- Используются для отсеечения ненужных значений

```
> db.db_subjects.createIndex({"number_of_labs" : 1}, {"partialFilterExpression" :{ number_of_labs: { $gt: 16 } }})  
< 'number_of_labs_1'
```

number_of_labs_1

number_of_labs ↑

partialFilterExpression: {"number_of_labs":{"\$gt":16}}

PARTIAL ⓘ

Управление индексами

- Периодически требуется проверять, как работают индексы – с ростом коллекции
- Индекс можно удалить DropIndex
- Если требуется доступ к коллекции во время создания индекса – выставлять свойство **background**

Управление индексами

```
> db.db_subjects.getIndices()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { number_of_lectures: 1 },
    name: 'number_of_lectures_1',
    background: false
  },
  { v: 2, key: { semester: 1 }, name: 'semester', background: false },
  {
    v: 2,
    key: { number_of_labs: 1 },
    name: 'number_of_labs_1',
    partialFilterExpression: { number_of_labs: [Object] }
  }
]
```


Пространственные индексы

- 2dsphere:
 - В сферической геометрии, SRID = WGS84
- 2d
 - На плоской модели
- Элементы задаются в формате GeoJSON
- Точка, линия, полигон
- Массив [долгота, широта]

Добавление коллекции 2d

Create Collection

Collection Name

locations

▼ **Advanced Collection Options** (e.g. Time-Series, Capped, Clustered collections)

☐ **Capped Collection**

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [Learn More](#)

☐ **Use Custom Collation**

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [Learn More](#)

☐ **Time-Series**

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Cancel

Create Collection

Добавление данных

```
db.locations.insertOne( {"name": "Minsk", "loc": {"type": "Point", "coordinates": [ 150, 150 ] } })  
< { acknowledged: true,  
  insertedId: ObjectId("67d94036c17e3f603c59c4cc") }
```

```
> db.locations.insertOne( {"name": "Svisloch", "loc": {"type": "LineString", "coordinates": [ [ 150 , 10 ], [ 120, 20 ], [ 160, 120 ] ] } })  
< { acknowledged: true,  
  insertedId: ObjectId("67d94040c17e3f603c59c4cd") }
```

```
> db.locations.insertOne({"name": "Belarus", "loc": { "type": "Polygon","coordinates" : [[10,10], [10,200], [200,200]] } })  
< { acknowledged: true,  
  insertedId: ObjectId("67d94090c17e3f603c59c4ce") }
```

```
> db.locations.createIndex({"loc.coordinates" : "2d"}, {"min" : -1000, "max" : 1000})  
< 'loc.coordinates_2d'
```

Поиск данных

- `$geoWithin` – нахождение объектов внутри фигуры
 - `$box` – внутри прямоугольника
 - `$center` – внутри круга
 - `$polygon` – внутри полигона
- `$near` - нахождение объектов поблизости
 - Объекты сортируются по расстоянию
 - По умолчанию, возвращается 100 объектов

Поиск данных внутри прямоугольника

```
> db.locations.find({"loc.coordinates": {$geoWithin: {$box: [[10, 10], [400, 700]]} } })
< { _id: ObjectId("67d94036c17e3f603c59c4cc"),
  name: 'Minsk',
  loc: { type: 'Point', coordinates: [ 150, 150 ] } }
{ _id: ObjectId("67d94040c17e3f603c59c4cd"),
  name: 'Svisloch',
  loc:
    { type: 'LineString',
      coordinates: [ [ 150, 10 ], [ 120, 20 ], [ 160, 120 ] ] } }
{ _id: ObjectId("67d94090c17e3f603c59c4ce"),
  name: 'Belarus',
  loc:
    { type: 'Polygon',
      coordinates: [ [ 10, 10 ], [ 10, 200 ], [ 200, 200 ] ] } }
```

Поиск данных внутри круга

```
> db.locations.find({"loc.coordinates": {$geoWithin: {$center: [[10, 60] , 125] } } })
< { _id: ObjectId("67d94040c17e3f603c59c4cd"),
  name: 'Svisloch',
  loc:
    { type: 'LineString',
      coordinates: [ [ 150, 10 ], [ 120, 20 ], [ 160, 120 ] ] } }
{ _id: ObjectId("67d94090c17e3f603c59c4ce"),
  name: 'Belarus',
  loc:
    { type: 'Polygon',
      coordinates: [ [ 10, 10 ], [ 10, 200 ], [ 200, 200 ] ] } }
```

Поиск данных внутри многоугольника

```
> db.locations.find({"loc.coordinates": {$geoWithin: {$polygon: [[10, 10], [30, 600], [60, 100]]} } })  
< { _id: ObjectId("67d94090c17e3f603c59c4ce"),  
  name: 'Belarus',  
  loc:  
    { type: 'Polygon',  
      coordinates: [ [ 10, 10 ], [ 10, 200 ], [ 200, 200 ] ] } }  
all_subjects > |
```

Поиск данных поблизости от точки

```
> db.locations.find({"loc.coordinates": {$near: [150, 160] } })
< { _id: ObjectId("67d94036c17e3f603c59c4cc"),
  name: 'Minsk',
  loc: { type: 'Point', coordinates: [ 150, 150 ] } }
{ _id: ObjectId("67d94040c17e3f603c59c4cd"),
  name: 'Svisloch',
  loc:
    { type: 'LineString',
      coordinates: [ [ 150, 10 ], [ 120, 20 ], [ 160, 120 ] ] } }
{ _id: ObjectId("67d94090c17e3f603c59c4ce"),
  name: 'Belarus',
  loc:
    { type: 'Polygon',
      coordinates: [ [ 10, 10 ], [ 10, 200 ], [ 200, 200 ] ] } }
```


Добавление коллекции 2d sphere

```
db.places.insertMany([{"type":"Feature","properties":{"Location ":"El", "Description ":"2 Stands","latitude":"53.6847795","longitude":"-1.83661","type":"Point","coordinates":[-1.83661,53.684779]}}, {"type":"Feature","properties":{"Location ":"Coronation Street Car park","Town":"Elland ","Description ":"5 Stands Shelter","latitude":"53.685762","longitude":"-1.8380105"},"geometry":{"type":"Point","coordinates":[-1.83801,53.685762]}}, {"type":"Feature","properties":{"Location ":"Elland ","Description ":"5 Stands & Shelter","latitude":"53.6812903","longitude":"-1.847618","type":"Point","coordinates":[-1.847618,53.68129]}}, {"type":"Feature","properties":{"Location ":"Brooksbank School","Town":"Elland ","Description ":"2 Stands","latitude":"53.6812903","longitude":"-1.8480018"},"geometry":{"type":"Point","coordinates":[-1.848002,53.68129]}}, {"type":"Feature","properties":{"Location ":"Halifax Swimming Pool","Town":"Halifax","Description ":"2 Stands","latitude":"53.7188378","longitude":"-1.8598759"},"geometry":{"type":"Point","coordinates":[-1.859876,53.718838]}}, {"type":"Feature","properties":{"Location ":"Halifax Station","Town":"Halifax","Description ":"3 Lockers, 7 stands & Shelter","latitude":"53.7212217","longitude":"-1.8538919"},"geometry":{"type":"Point","coordinates":[-1.853892,53.721222]}}, {"type":"Feature","properties":{"Location ":"Halifax Street","Town":"Halifax","Description ":"2 Stands","latitude":"53.7212217","longitude":"-1.856539"},"geometry":{"type":"Point","coordinates":[-1.856539,53.721221]}}
```

Добавление коллекции 2d sphere

all_subjects.places

70
DOCUMENTS

1
INDEXES

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER { field: 'value' }

OPTIONS

FIND

RESET

↺

⋮

ADD DATA

⬇

VIEW

☰

{ }

📊

Displaying documents 1 - 20 of 70

<

>

REFRESH

```
_id: ObjectId('67d989718237e3af3dca2699')
type: "Feature"
> properties: Object
> geometry: Object
```

```
_id: ObjectId('67d989ae8237e3af3dca269a')
type: "Feature"
> properties: Object
> geometry: Object
```

```
> db.places.createIndex({"geometry.coordinates" : "2dsphere"})
< 'geometry.coordinates_2dsphere'
```

Добавление индекса 2d sphere

```
> db.places.createIndex({"geometry" : "2dsphere"})  
< 'geometry_2dsphere'
```

all_subjects.places

70

DOCUMENTS

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

CREATE INDEX

Name
and
Definition

Type

Size

Usage

Prope

id

id

REGULAR

36.9 KB

5

since Tue Mar 18 2025

UNIQ

geometry.coordin...

geometry....

GEOSPATIAL

20.5 KB

0

since Tue Mar 18 2025

Поиск данных

- \$geoIntersects – пересечение объектов
- \$near - нахождение объектов поблизости
- \$geoWithin – нахождение объектов внутри фигуры
 - \$geometry – внутри какого-либо объекта
 - \$centerSphere – внутри круга на поверхности

Поиск пересечения

```
> db.places.find({geometry:{$geoIntersects:{$geometry:{type:"Point",coordinates:[-1.83661, 53.684779]}}}})  
< { _id: ObjectId("67d98c46ec3dde719eb898d6"),  
  type: 'Feature',  
  properties:  
    { 'Location ': 'Elland Timber Street',  
      Town: 'Elland ',  
      'Description ': '2 Stands',  
      latitude: '53.6847795',  
      longitude: '-1.8366104' },  
  geometry: { type: 'Point', coordinates: [ -1.83661, 53.684779 ] } }
```

Поиск данных внутри круга

```
> db.places.find({geometry:{$geoWithin: { $centerSphere: [[-1.83661, 53.684779], 1/3963.2] } } })
< { _id: ObjectId("67d98a198237e3af3dca269d"),
  type: 'Feature',
  properties:
    { 'Location ': 'Elland Morrisons Lower Car park',
      Town: 'Elland ',
      'Description ': '10 Stands',
      latitude: '53.6875807',
      longitude: '-1.8357843' },
  geometry: { type: 'Point', coordinates: [ -1.835784, 53.687581 ] } }
{ _id: ObjectId("67d98a498237e3af3dca269e"),
  type: 'Feature',
  properties:
    { 'Location ': 'Elland Morrisons Briggate entrance',
      Town: 'Elland ',
      'Description ': '9 Stands & 6 Bikeaway lockers',
      latitude: '53.687047',
      longitude: '-1.8366399' },
  geometry: { type: 'Point', coordinates: [ -1.83664, 53.687047 ] } }
```

Поиск данных внутри фигуры

```
> db.places.find({geometry:{$geoWithin: { $geometry : {type:"Polygon","coordinates":[[[-1.9, 53.68], [-1.9, 53.78], [1.8, 53.8], [-1.9, 53.68]]]} } } })
< { _id: ObjectId("67d989188237e3af3dca2698"),
  type: 'Feature',
  properties:
    { 'Location ': 'Brighouse Railway Station Westbound',
      Town: 'Brighouse',
      'Description ': '3 Lockers',
      latitude: '53.6980959',
      longitude: '-1.7790804' },
  geometry: { type: 'Point', coordinates: [ -1.77908, 53.698096 ] } }
{ _id: ObjectId("67d989718237e3af3dca2699"),
  type: 'Feature',
  properties:
    { 'Location ': 'Brighouse Sainsbury\'s',
      Town: 'Brighouse',
      'Description ': '4 Stands',
```

```
> db.places.find({geometry:{$geoWithin: { $geometry : {type:"Polygon","coordinates":[[[-1.9, 53.68], [-1.9, 53.78], [1.8, 53.8], [-1.9, 53.68]]]} } } }).count()
< 31
```

Поиск данных возле фигуры

```
> db.places.find({ geometry: {$near: {$geometry: {type: "Point", coordinates: [-1.83662, 53.684778]} } } }).limit(5)
< { _id: ObjectId("67d98c46ec3dde719eb898d6"),
  type: 'Feature',
  properties:
    { 'Location ': 'Elland Timber Street',
      Town: 'Elland ',
      'Description ': '2 Stands',
      latitude: '53.6847795',
      longitude: '-1.8366104' },
  geometry: { type: 'Point', coordinates: [ -1.83661, 53.684779 ] } }
{ _id: ObjectId("67d98c46ec3dde719eb898d7"),
  type: 'Feature',
  properties:
    { 'Location ': 'Elland Coronation Street Car park',
      Town: 'Elland ',
      'Description ': '5 Stands & Smashed Shelter',
      latitude: '53.685762',
      longitude: '-1.8380105' },
  geometry: { type: 'Point', coordinates: [ -1.83801, 53.685762 ] } }
{ _id: ObjectId("67d98a498237e3af3dca269e"),
  type: 'Feature',
  properties:
    { 'Location ': 'Elland Morrisons Briggate entrance',
      Town: 'Elland ',
      'Description ': '9 Stands & 6 Bikeaway lockers'
```


Полнотекстовые индексы

- Проблемы поиска по шаблону:
 - Медленно
 - Проблемы морфологии слов
- Полнотекстовый индекс:
 - Быстрый поиск текста
 - Поддержка токенизации, стоп-слов, стемминга
 - Релевантность

Полнотекстовые индексы

- Коллекция из 5555 объявлений о сдаче жилья внаем с отзывами

```
_id: "10006546"  
listing_url: "https://www.airbnb.com/rooms/10006546"  
name: "Ribeira Charming Duplex"  
summary: "Fantastic duplex apartment with three bedrooms, located in the histori..."  
space: "Privileged views of the Douro River and Ribeira square, our apartment ..."  
description: "Fantastic duplex apartment with three bedrooms, located in the histori..."  
neighborhood_overview: "In the neighborhood of the river, you can find several restaurants as ..."  
notes: "Lose yourself in the narrow streets and staircases zone, have lunch in..."  
transit: "Transport: • Metro station and S. Bento railway 5min; • Bus stop a 50 ..."  
access: "We are always available to help guests. The house is fully available t..."  
interaction: "Cot - 10 € / night Dog - € 7,5 / night"  
house_rules: "Make the house your home..."  
property_type: "House"  
room_type: "Entire home/apt"  
bed_type: "Real Bed"  
minimum_nights: "2"  
maximum_nights: "30"  
cancellation_policy: "moderate"  
last_scraped: 2019-02-16T05:00:00.000+00:00  
calendar_last_scraped: 2019-02-16T05:00:00.000+00:00  
first_review: 2016-01-03T05:00:00.000+00:00  
last_review: 2019-01-20T05:00:00.000+00:00  
accommodates: 8
```

Полнотекстовые индексы

- Создание индекса с указанием полей
- Создание индекса на любые поля в коллекции
- Создание индекса с весами

Создание индекса с указанием полей

```
db.reviews.createIndex({"name": "text", "summary" : "text", "space": "text",  
"description" : "text", "neighborhood_overview": "text", "notes" : "text",  
"transit": "text", "access" : "text", "interaction": "text",  
"house_rules" : "text", "access" : "text"})
```

all_subjects.reviews

Documents Aggregations Schema Explain Plan **Indexes** Validation

CREATE INDEX

Name and Definition ^

Type

Size

Usage

id

id ↕

REGULAR ⓘ

98.3 KB

1 since Mon Mar 31 2025

name_text_summary_text_space_text_description_text_neighborhood_overview_text_notes_text_transit_text_...

_fts text _ftsx ↕

TEXT ⓘ

11.9 MB

0 since Mon Mar 31 2025

Создание индекса на любые поля в коллекции

Drop Index

⚠ Type the index name

name_text_summary_text_space_text_description_text_neighborhood_overview_text_not
to drop

Cancel

Drop

```
> db.reviews.createIndex({"$**" : "text"})  
< '$**_text'  
all_subjects> |
```

Indexes
all_subjects.reviews



all_subjects.reviews

Documents Aggregations Schema Explain Plan **Indexes** Validation

CREATE INDEX

Name and Definition ^

Type

Size

\$**_text

_fts text _ftsx ↕

TEXT ⓘ

53.7 MB

Использование индекса

```
> db.reviews.find({$text: {$search: "duplex"}}).count()
< 147
> db.reviews.find({$text: {$search: "duplex"}}).limit(5)
< { _id: '4138543',
  summary: 'oOo SPACE Room only oOo Set in Shatin\'s natural countryside, our duplex is hidden between quiet
{ _id: '11223389',
  summary: 'Enjoy New York at this comfortable duplex. Only a few blocks from the main subway lines and with
{ _id: '846854',
  summary: 'Large duplex on prime Park block in West 80ies. Bedrooms sleep 4 adults and 1 or 2 toddlers plus
{ _id: '582364', summary: '' }
{ _id: '19877430',
  summary: 'Welcome to paradise! Our house is nestled in the heart of Kaneohe with magnificent views of the
all_subjects>
```

Создание индекса с весами

×

Drop Index

⚠ Type the index name `$**_text` to drop

Cancel

Drop

```
> db.reviews.createIndex({"name": "text", "summary" : "text"},{ "weights": { "name": 1,"summary": 7, "description" : 3}})
< 'name_text_summary_text'
all_subjects> |
```

Создание индекса с весами

Indexes
all_subjects.reviews



all_subjects.reviews

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

CREATE INDEX

Name and Definition ^

Type

Size

id

id ↑

REGULAR ⓘ

98.3 KB

name_text_summary_text

_fts text

_ftsx ↑

TEXT ⓘ

6.7 MB

Использование индекса

```
> db.reviews.find({$text: {$search: "duplex"}} , {summary :1}).limit(5)
< { _id: '32456345',
  summary: 'An luxury duplex close the famous Plaza España and the Fira center. Our duplex is decorated in a modern a tasteful way to max
{ _id: '3905849',
  summary: 'This Modern beautiful duplex loft is located in a private and coveted boutique condominium in the prestigious and centrally
{ _id: '28795425',
  summary: 'Nice bright and sunny spacious "Acomodis" duplex apartment located in central Barcelona, in Poble Sec in a very nice and safe
{ _id: '1925519',
  summary: '$490/week. Huge 1 bedroom / 2bdr duplex in LES - fully furnished, beautiful apt - spiral staircase duplex (2 levels) - 2 bedr
{ _id: '19820634',
  summary: 'This fantastic and modern duplex apartment is situated just next to Park Güell, one of the biggest attractions in Barcelona
all_subjects>
```

Создание индекса

Indexes
all_subjects.reviews



all_subjects.reviews

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

CREATE INDEX

Name and Definition ^

Type

Size

\$**_text

_fts text _ftsx ↑

TEXT ⓘ

53.7 MB

id

id ↑

REGULAR ⓘ

98.3 KB

Использование индекса

- Поиск ИЛИ
- Поиск И
- Исключение
- Стемминг
- Релевантность

Поиск ИЛИ

```
> db.reviews.find({$text: {$search: "seaside"}}, {summary :1}).count()
```

```
< 85
```

```
> db.reviews.find({$text: {$search: "lake"}}, {summary :1}).count()
```

```
< 79
```

```
> db.reviews.find({$text: {$search: "seaside lake"}}, {summary :1}).count()
```

```
< 162
```

Поиск И

```
> db.reviews.find({$text: {$search: "\"seaside\" \"pool\" \"breakfast\" \"\"}}, {summary :1}).count()  
< 12  
  
> db.reviews.find({$text: {$search: "seaside"}}, {summary :1}).count()  
< 85  
  
> db.reviews.find({$text: {$search: "pool"}}, {summary :1}).count()  
< 1065  
  
> db.reviews.find({$text: {$search: "breakfast"}}, {summary :1}).count()  
< 1624
```

Исключение

```
> db.reviews.find({$text: {$search: "lake"}}, {summary :1}).count()  
< 79  
> db.reviews.find({$text: {$search: "seaside"}}, {summary :1}).count()  
< 85  
> db.reviews.find({$text: {$search: "seaside lake"}}, {summary :1}).count()  
< 162  
> db.reviews.find({$text: {$search: "seaside -lake"}}, {summary :1}).count()  
< 83
```

Стемминг

```
> db.reviews.find({$text: {$search: "bedrooms"}} , {summary :1}).count()  
< 3689  
> db.reviews.find({$text: {$search: "bedroom"}} , {summary :1}).count()  
< 3689  
> db.reviews.find({$text: {$search: "BEDROOM"}} , {summary :1}).count()  
< 3689  
all_subjects> |
```

Релевантность

```
> db.reviews.find({$text: {$search: "\"villa\" \"pool\" \"Barcelona\" \"\"}}, {summary : 1})
< { _id: '1396721', summary: '' }
  { _id: '582364', summary: '' }
  { _id: '15341505',
    summary: 'A double room in a unique villa located on top of Tibidabo, with a lovely garden, an amazing swimming pool
  { _id: '1410802',
    summary: 'Single-family home in Barcelona with lots of charm. Living room, fully furnished kitchen, 2 triple bedrooms
  { _id: '25233509',
    summary: 'Luxury 4-bedroom apartment with a terrace at highly sought-after Diagonal Mar area of Barcelona. This beau
  { _id: '31941328',
    summary: 'Two apartments, near to each other in Village Olympic area. Accommodation for 12 people! Really good for fa
  { _id: '21848831',
    summary: 'Sie wohnen in einer alten Landhaus Villa in Alleinlage, in einem mit viel Liebe zum Detail eingerichtetes
> db.reviews.find({$text: {$search: "\"villa\" \"pool\" \"Barcelona\" \"\"}}, {summary : 1}).count()
< 7
```


Релевантность

```
> db.reviews.find({$text: {$search: "\"villa\" \"pool\" \"Barcelona\" "}}, {summary : 1}).count()
< 7
> db.reviews.find({$text: {$search: "\"villa\" \"pool\" \"Barcelona\" "}}, {summary :1, score : {$meta : "textScore"}}).sort({score : {$meta : "textScore"}})
< { _id: '1396721', summary: '', score: 50.53132559095703 }
  { _id: '582364', summary: '', score: 49.608515754304136 }
  { _id: '15341505',
    summary: 'A double room in a unique villa located on top of Tibidabo, with a lovely garden, an amazing swimming pool and spectacular views over the city.
    score: 40.38641474410655 }
  { _id: '1410802',
    summary: 'Single-family home in Barcelona with lots of charm. Living room, fully furnished kitchen, 2 triple bedrooms, 2 double bedrooms, and 2 bathrooms
    score: 19.620680196279658 }
  { _id: '25233509',
    summary: 'Luxury 4-bedroom apartment with a terrace at highly sought-after Diagonal Mar area of Barcelona. This beautiful apartment in the exclusive Illa
    score: 11.34411247025692 }
  { _id: '21848831',
    summary: 'Sie wohnen in einer alten Landhaus Villa in Alleinlage, in einem mit viel Liebe zum Detail eingerichtetes Motto Zimmer " Afrika". Ein Doppelz
    score: 7.155733055685618 }
  { _id: '31941328',
    summary: 'Two apartments, near to each other in Village Olympic area. Accommodation for 12 people! Really good for families and for groups of friends. Tou
    score: 3.5333333333333337 }
```

Релевантность

- Суммируется количество входов, перемноженных на веса
- Применяется коэффициент точности слова
- Учитывается количество слов поиска

```
> db.reviews.createIndex({"name": "text", "summary" : "text"},{ "weights": {"name": 1,"summary": 7}})
< 'name_text_summary_text'
> db.reviews.find({$text: {$search: "\"villa\" \"pool\" \"Barcelona\" "}}, {summary :1, score : {$met
< { _id: '15341505',
  summary: 'A double room in a unique villa located on top of Tibidabo, with a lovely garden, an am
  score: 11.475274725274726 }
```

```
all_subjects>
```

Вопросы?