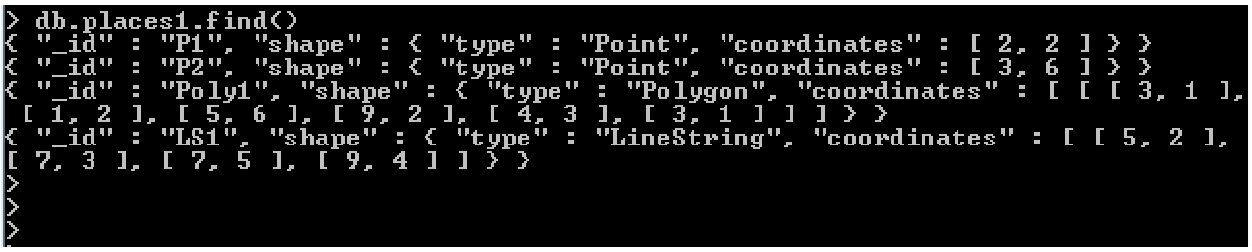
Module 5

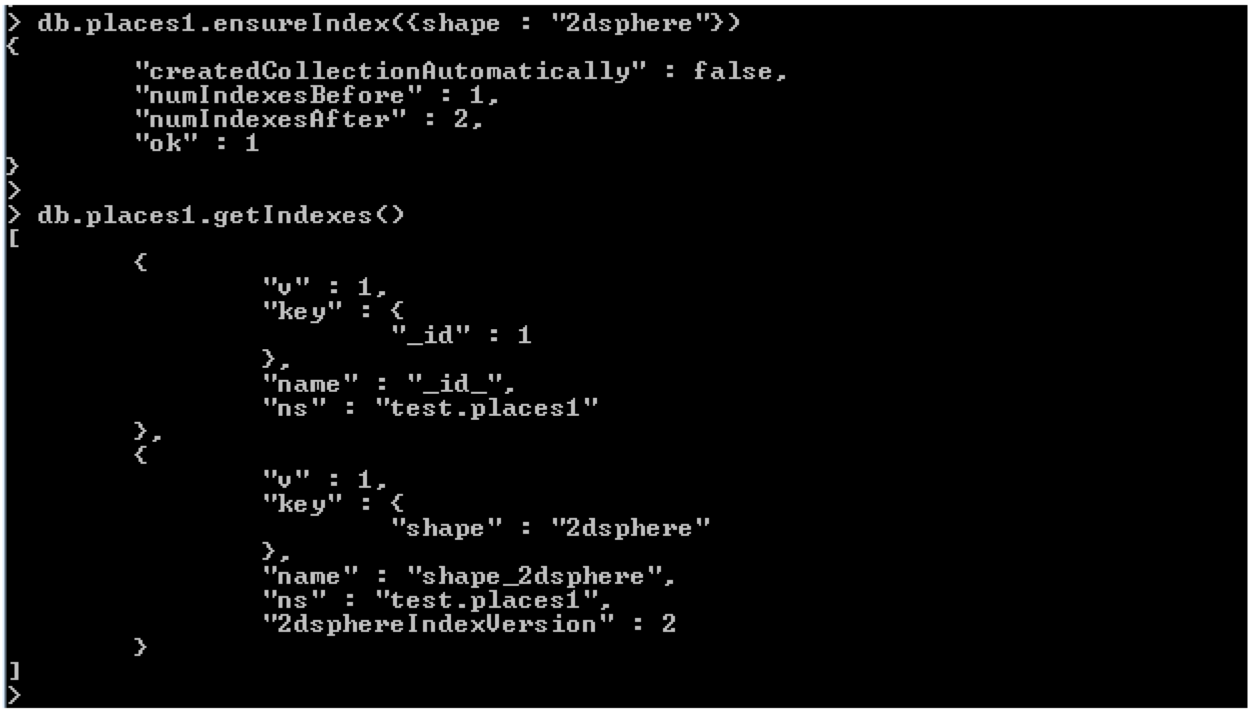
2dsphere indexes

Suppose we have following collection in our database:



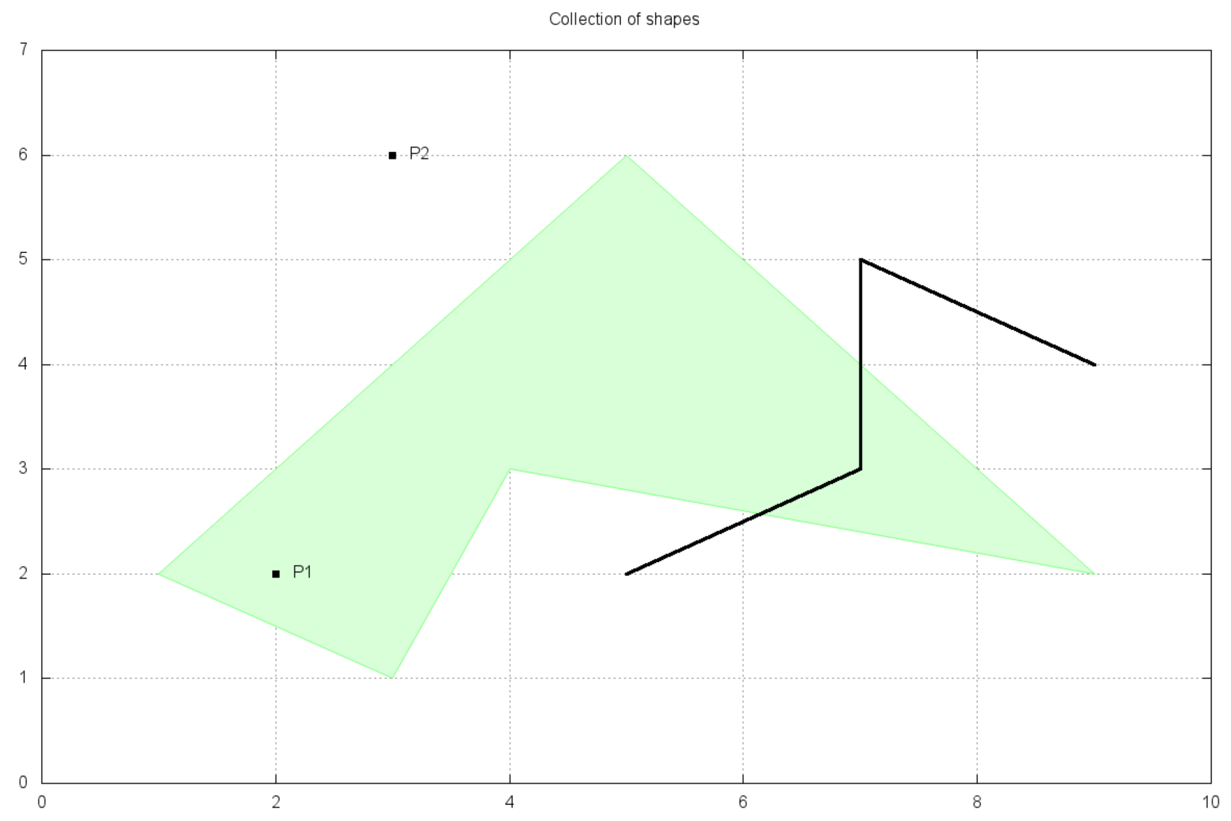
```
> db.places1.find()
{ "_id" : "P1", "shape" : { "type" : "Point", "coordinates" : [ 2, 2 ] } }
{ "_id" : "P2", "shape" : { "type" : "Point", "coordinates" : [ 3, 6 ] } }
{ "_id" : "Poly1", "shape" : { "type" : "Polygon", "coordinates" : [ [ [ 3, 1 ],
 [ 1, 2 ], [ 5, 6 ], [ 9, 2 ], [ 4, 3 ], [ 3, 1 ] ] ] } }
{ "_id" : "LS1", "shape" : { "type" : "LineString", "coordinates" : [ [ 5, 2 ],
[ 7, 3 ], [ 7, 5 ], [ 9, 4 ] ] } }
>
>
>
```

We indexed on shape field, and we see the indexes using getIndexes() method:



```
> db.places1.ensureIndex({shape : "2dsphere"})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
>
> db.places1.getIndexes()
[
        {
                "v" : 1,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "test.places1"
        },
        {
                "v" : 1,
                "key" : {
                        "shape" : "2dsphere"
                },
                "name" : "shape_2dsphere",
                "ns" : "test.places1",
                "2dsphereIndexVersion" : 2
        }
]
>
```

Here the figure shows you that how we can plot the four different types of different documents:

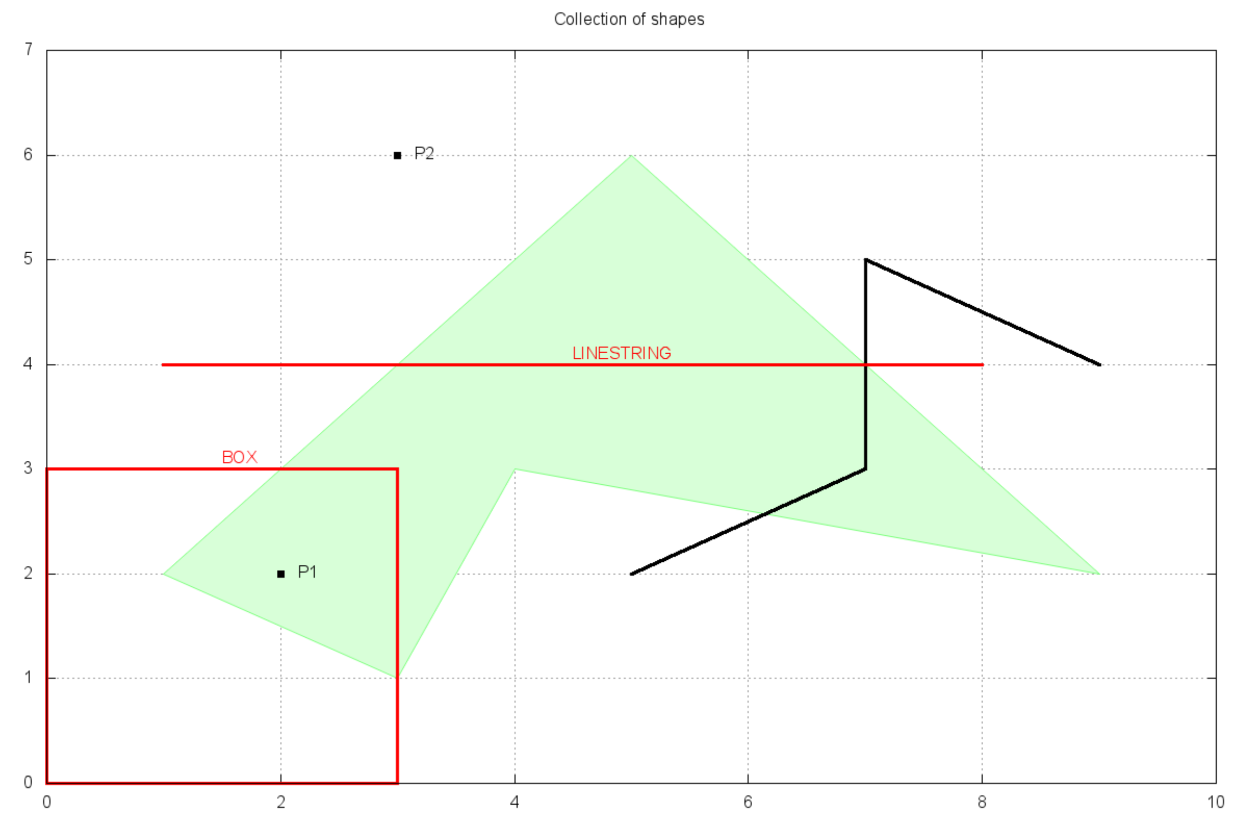


To perform queries with GeoJSON geometries, you can use some new query operators $geoIntersects and $geometry. First of all, we define some query geometries:

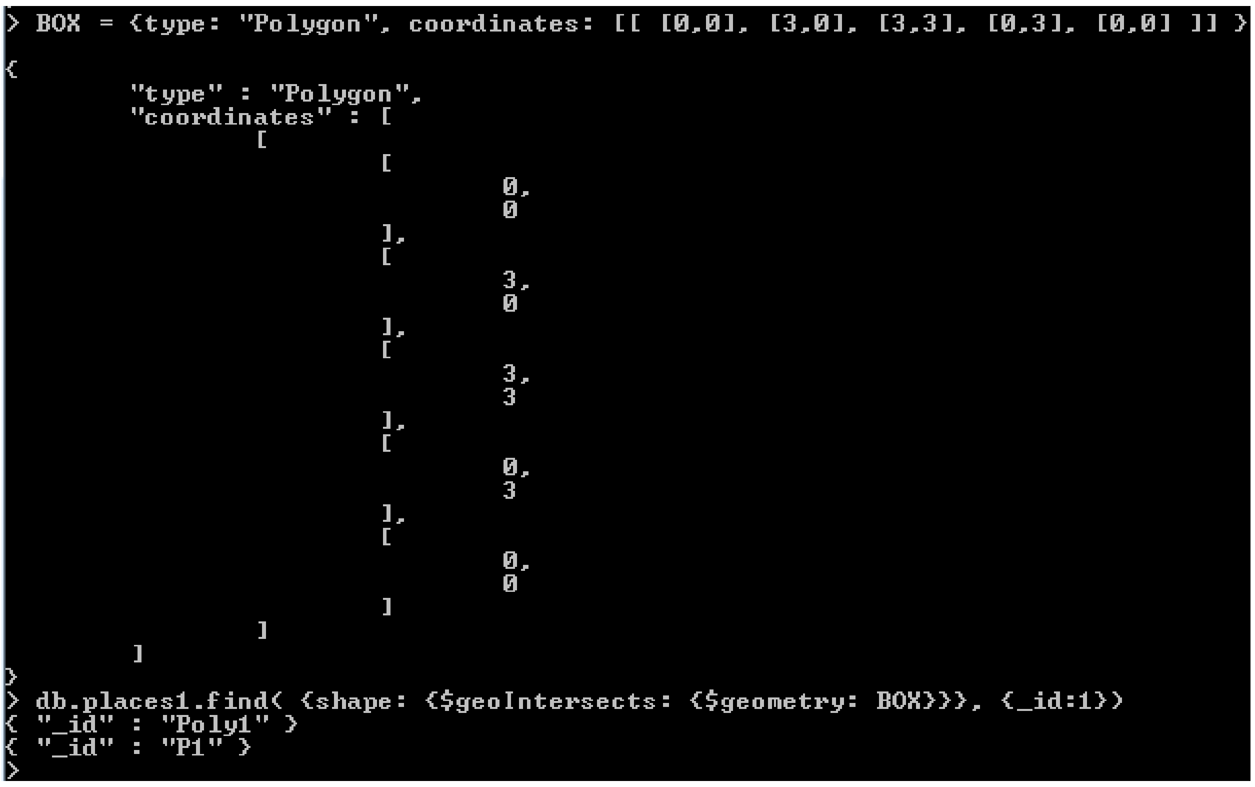
Like:

```
> BOX = {type: "Polygon", coordinates: [[ [0,0], [3,0], [3,3], [0,3], [0,0] ]] }
> LINESTRING = {type: "LineString", coordinates: [[1,4], [8,4]]}
```

We can check the results respectively, that how the box and linestring is displayed in the graph below:

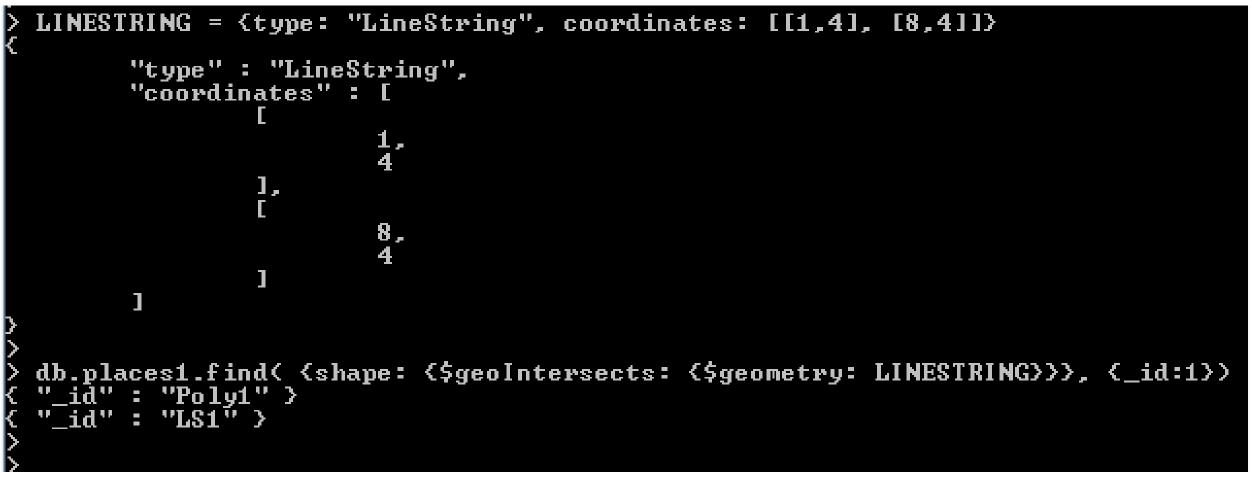


Accordingly we can check the conditions, that this is for “BOX”. That this query will display the id’s which intersects the BOX.



```
> BOX = {type: "Polygon", coordinates: [[ [0,0], [3,0], [3,3], [0,3], [0,0] ]] }
{
        "type" : "Polygon",
        "coordinates" : [
                [
                        [
                                0,
                                0
                        ],
                        [
                                3,
                                0
                        ],
                        [
                                3,
                                3
                        ],
                        [
                                0,
                                3
                        ],
                        [
                                0,
                                0
                        ]
                ]
        ]
}
> db.places1.find( {shape: {$geoIntersects: {$geometry: BOX}}}, {_id:1})
{ "_id" : "Poly1" }
{ "_id" : "P1" }
>
```

For linestring, all those id’s are displayed which intersect a linestring:



```
> LINESTRING = {type: "LineString", coordinates: [[1,4], [8,4]]}
{
        "type" : "LineString",
        "coordinates" : [
                [
                        1,
                        4
                ],
                [
                        8,
                        4
                ]
        ]
}
>
> db.places1.find( {shape: {$geoIntersects: {$geometry: LINESTRING}}}, {_id:1})
{ "_id" : "Poly1" }
{ "_id" : "LS1" }
>
>
```