



MongoDb Databases

Trainer: Mr. Nilesh Ghule



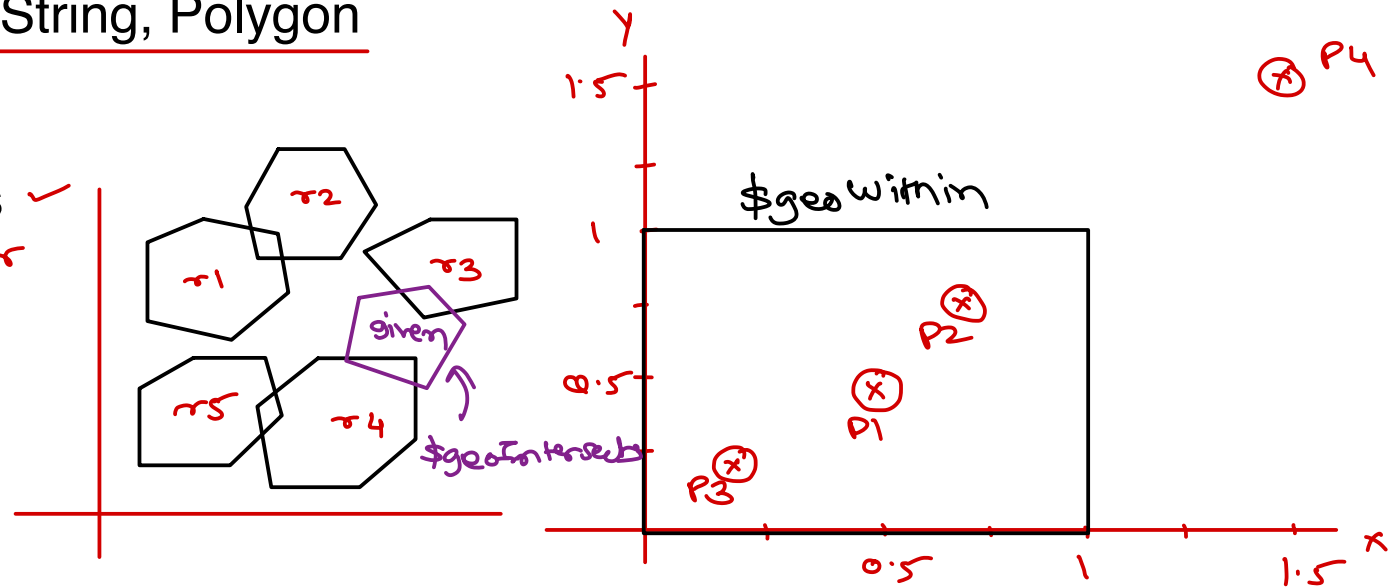
Mongo - Indexes

- `db.books.find({ "subject" : "C" }).explain(true);`
- `explain()` → explains the query execution plan.
- Above query by default does full collection scan, hence slower.
- `db.books.createIndex({ "subject" : 1 });`
- Searching on indexed columns reduces query execution time.
- Options can be provided (2nd arg): `{ unique : true }`
 - Duplicate values are not allowed in that field.
- By default `"_id"` field is indexed in mongodb (unique index).
- `db.books.getIndexes();`
- `db.books.dropIndex({ "subject" : 1 });`



Mongo - GeoSpatial Queries

- mongodb support three types of geo-spatial queries: *indexes*
 - 2-d index: traditional long-lat. used in older mongodb (2.2-). *→ not geojson.*
 - 2-d sphere index: data can be stored as GeoJSON. *→ required to find nearby services.*
 - geo-haystack: query on very small area. not much used.
- GeoJSON stores geometry and coordinates: <http://geojsonlint.com/> ✓
 - { type: "geometry", coordinates: [long, lat] };
 - { type: "Point", coordinates: [73.86704859999998, 18.4898445] };
- Possible geometry types are:
 - Point, LineString, MultiLineString, Polygon
- allowed queries:
 - inclusion - \$geoWithin ✓
 - intersection - \$geoIntersects ✓
 - proximity - ~~\$near~~ *\$geoNear*



Mongo - GeoSpatial Queries

- For faster execution create geo-index :
 - `db.busstops.createIndex({ location : "2dsphere" });`
- Example proximity query:

```
db.busstops.find( { location : { $near : {  
    $geometry : { type : "Point" ,  
        coordinates : [73.86704859999998, 18.4898445]  
    },  
    $maxDistance : 200  
} } } );
```



Mongo - Capped Collections

- Capped collections are fixed sized collections for high-throughput insert and retrieve operations.
- They maintain the order of insertion without any indexing overhead.
- The oldest documents are auto-removed to make a room for new records. The size of collection should be specified while creation.
- The update operations should be done with index for better performance. If update operation change size, then operation fails.
- Cannot delete records from capped collections. Can drop collection.
- Capped collections cannot be sharded.
- `db.createCollection("logs", { capped: true, size: 4096 });` → if size is below 4096, 4096 is considered. Higher sizes are roundup by 256.





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

