Page No.
Date

DBMSL 31147

Assignment B3.

Date of completion : 30/10/20 Date of submission : 12/11/20.

Title : Aggregation & Indexing in MongoDB.

Problem statement :
Implement aggregation and indexing with suitable example using MongoDB.

Learning Objectives :
To understand indexing and aggregation concept in mongoDB

Learning Outcomes :
Students will be able to :
1) Implement indexing.
2) Implement aggregation pipeline

H/W & S/W Required :
MongoDB, Windows/Ubuntu, i7 processor, 8 GB RAM.

Theory :
Indexes support the efficient execution of queries in MongoDB. Without indexes, MongoDB must perform a collection scan, i.e. scan every document in a collection to select those documents that match the query statement.

Default _id index :
MongoDB creates a unique index on the _id field during the creation of a collection. The _id index prevents clients from inserting two documents with the same value for the id field. You cannot drop this index on the _id field

Indexing Types :
Single field indexes :
It includes data from a single field of the documents in a collection
db.collectionName.createIndex({keyName : 1 or -1}).
1 for ascending sorting, -1 for descending sorting.

Scanned with CamScanner

Page No.
Date

Compound index:
It includes more than one field of the documents in a collection. The order of field listed in a compound index has a significance. For eg: if a compound index consist of {userid: 1, score: -1} the index sorts first by userid and then within each userid value sorts by score

db.collection.createIndex({<field1>: <type>, <field 2>: <type2>})

Multikey Index:
It is an index on an array field, adding an index key for each value in the array.

Geospatial Index:
It supports location based searches

Text Indexes:
It supports search of string content in documents

Hashed Index:
It maintains entries with hashes of value of indexed field and are used with sharded clusters to | to support hashed shared keys.

Unique Index:
The unique property for an index causes MongoDB to reject duplicate values for the indexed fields.

db. ~~create~~ Collection.createIndex(<key & value>, {unique: true});

Partial Index:
Partial Indexes only index the documents in a collection that meet a specified filter expression. By indexing a subset of the documents in a collection, partial indexes have a lower storage requirements and reduced performance costs.

db.collection.createIndex({key: ~~value~~ 1, key2: 1}, {partialFilterExpression: {rating: {$gt: 5}}}).

Page No.
Date

Index Display:
db.collection.getIndexes()
Return an array that hold a list of document that identify & describe the existing indexes on the collections.

Index Drop:
db.collection.dropIndex({key:1})
db.collection.dropIndexes({key:1}, {key2:-1})

Aggregation: These operations group values from multiple document together and can perform a variety of operations on the grouped data to return a result.
Syntax: db.collection.aggregate (operation).

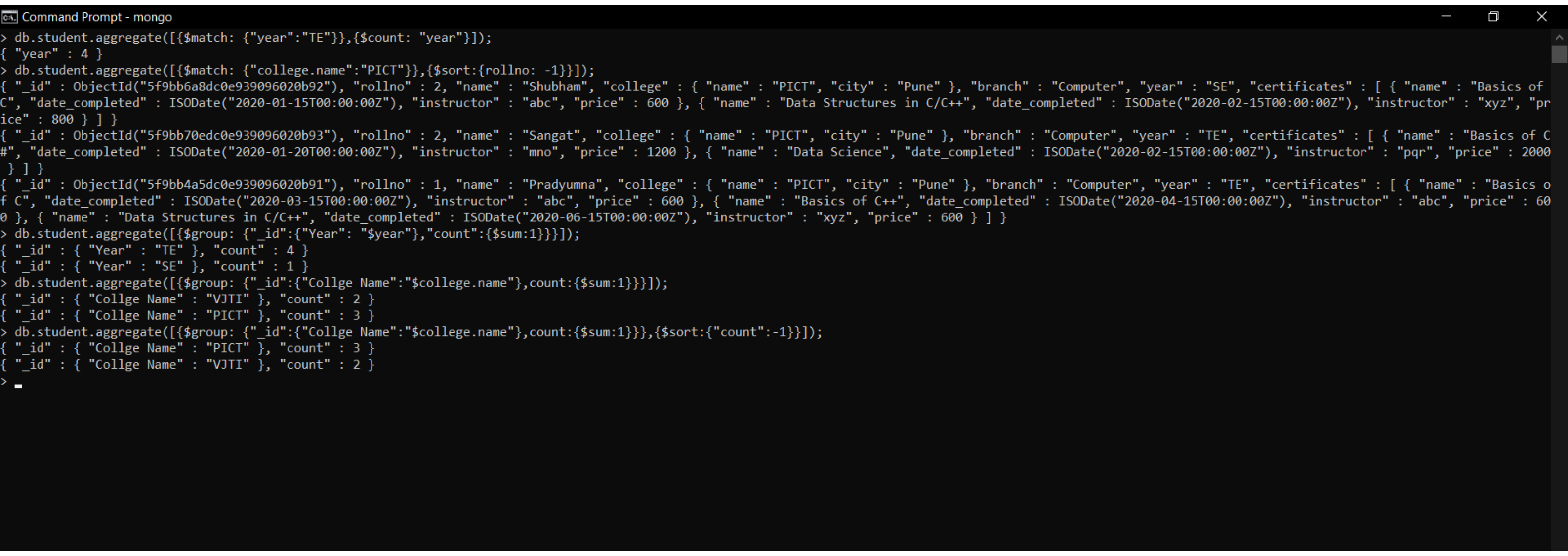
Expression		Description
$ sum	-	sum up the defined value from all documents
$ avg	-	Calculates the average of all given values from docs
$ min	-	Gets the minimum of corresponding value
$ max	-	Get the max value of corresponding field.

Pipeline concept:
The aggregation pipeline is a framework for data aggregation modeled on the concept of data processing pipeline. Documents enter a multi-stage pipeline that transforms the documents into aggregate results.

Possible stages in aggregation framework:
$project - used to select some specific fields.
$match - filtering operating, reduces the amount of documents that are given to as input to the next stage.
$group: Does the actual aggregation.
$sort - Sorts the documents.
Eg:
```
db.orders.aggregate([
        $match: {status: 'A'},
        {$group: {_id: {cust_id: "$cust_id"}}, total: {$sum: "$amt"}},
        }]);
```

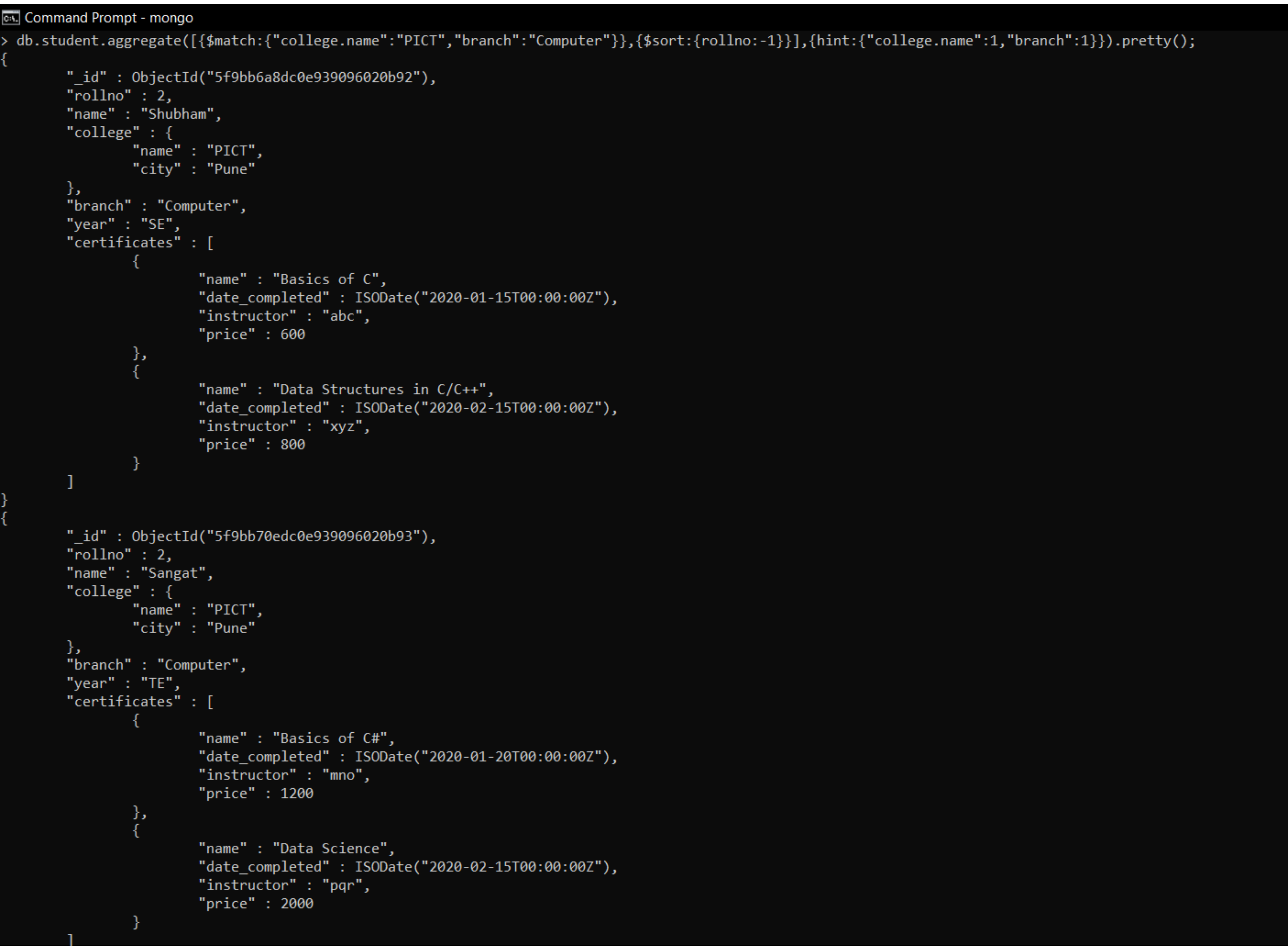
Page No.
Date

Conclusion:
1) Understand different types of indexs and implemented them
2) Understand the concept of aggregation pipeline & successfully implemented it.



Command Prompt - mongo

```
> db.student.aggregate([{$match: {"year":"TE"}},{$count: "year"}]);
{ "year" : 4 }
> db.student.aggregate([{$match: {"college.name":"PICT"}},{$sort:{rollno: -1}}]);
{ "_id" : ObjectId("5f9bb6a8dc0e939096020b92"), "rollno" : 2, "name" : "Shubham", "college" : { "name" : "PICT", "city" : "Pune" }, "branch" : "Computer", "year" : "SE", "certificates" : [ { "name" : "Basics of 
C", "date_completed" : ISODate("2020-01-15T00:00:00Z"), "instructor" : "abc", "price" : 600 }, { "name" : "Data Structures in C/C++", "date_completed" : ISODate("2020-02-15T00:00:00Z"), "instructor" : "xyz", "pr
ice" : 800 } ] }
{ "_id" : ObjectId("5f9bb70edc0e939096020b93"), "rollno" : 2, "name" : "Sangat", "college" : { "name" : "PICT", "city" : "Pune" }, "branch" : "Computer", "year" : "TE", "certificates" : [ { "name" : "Basics of C
#", "date_completed" : ISODate("2020-01-20T00:00:00Z"), "instructor" : "mno", "price" : 1200 }, { "name" : "Data Science", "date_completed" : ISODate("2020-02-15T00:00:00Z"), "instructor" : "pqr", "price" : 2000
 } ] }
{ "_id" : ObjectId("5f9bb4a5dc0e939096020b91"), "rollno" : 1, "name" : "Pradyumna", "college" : { "name" : "PICT", "city" : "Pune" }, "branch" : "Computer", "year" : "TE", "certificates" : [ { "name" : "Basics o
f C", "date_completed" : ISODate("2020-03-15T00:00:00Z"), "instructor" : "abc", "price" : 600 }, { "name" : "Basics of C++", "date_completed" : ISODate("2020-04-15T00:00:00Z"), "instructor" : "abc", "price" : 60
0 }, { "name" : "Data Structures in C/C++", "date_completed" : ISODate("2020-06-15T00:00:00Z"), "instructor" : "xyz", "price" : 600 } ] }
> db.student.aggregate([{$group: {"_id":{"Year": "$year"},"count":{$sum:1}}}]);
{ "_id" : { "Year" : "TE" }, "count" : 4 }
{ "_id" : { "Year" : "SE" }, "count" : 1 }
> db.student.aggregate([{$group: {"_id":{"Collge Name":"$college.name"},count:{$sum:1}}}]);
{ "_id" : { "Collge Name" : "VJTI" }, "count" : 2 }
{ "_id" : { "Collge Name" : "PICT" }, "count" : 3 }
> db.student.aggregate([{$group: {"_id":{"Collge Name":"$college.name"},count:{$sum:1}}},{$sort:{"count":-1}}]);
{ "_id" : { "Collge Name" : "PICT" }, "count" : 3 }
{ "_id" : { "Collge Name" : "VJTI" }, "count" : 2 }
>
```


Command Prompt - mongo

```
> db.student.aggregate([{$match:{"college.name":"PICT","branch":"Computer"}},{$sort:{rollno:-1}}],{hint:{"college.name":1,"branch":1}}).pretty();
{
        "_id" : ObjectId("5f9bb6a8dc0e939096020b92"),
        "rollno" : 2,
        "name" : "Shubham",
        "college" : {
                "name" : "PICT",
                "city" : "Pune"
        },
        "branch" : "Computer",
        "year" : "SE",
        "certificates" : [
                {
                        "name" : "Basics of C",
                        "date_completed" : ISODate("2020-01-15T00:00:00Z"),
                        "instructor" : "abc",
                        "price" : 600
                },
                {
                        "name" : "Data Structures in C/C++",
                        "date_completed" : ISODate("2020-02-15T00:00:00Z"),
                        "instructor" : "xyz",
                        "price" : 800
                }
        ]
}
{
        "_id" : ObjectId("5f9bb70edc0e939096020b93"),
        "rollno" : 2,
        "name" : "Sangat",
        "college" : {
                "name" : "PICT",
                "city" : "Pune"
        },
        "branch" : "Computer",
        "year" : "TE",
        "certificates" : [
                {
                        "name" : "Basics of C#",
                        "date_completed" : ISODate("2020-01-20T00:00:00Z"),
                        "instructor" : "mno",
                        "price" : 1200
                },
                {
                        "name" : "Data Science",
                        "date_completed" : ISODate("2020-02-15T00:00:00Z"),
                        "instructor" : "pqr",
                        "price" : 2000
                }
        ]
```