PRACTICAL 4

AIM :- Indexing ,Aggregation and Map Reduce in NoSQL-DB.

1. INDEXING :-

- Default _ID Index :-

Ans :-

```
db.Asset_055.getIndexes()
```

```
jeni> db.Asset_055.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
jeni>
```

- **Single Field Index :-**

```
db.Asset_055.createIndex({Rank:1})
```

```
jeni> db.Asset_055.createIndex({Rank:1})
Rank_1
jeni> db.Asset_055.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { Rank: 1 }, name: 'Rank_1' }
]
jeni>
```

- **Compound Index :-**

```
db.Asset_055.createIndex({"Name":1,"Desc":1}
```

```
jeni> db.Asset_055.createIndex({"Name":1,"Desc":1}
... )
Name_1_Desc_1
jeni> db.Asset_055.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { Rank: 1 }, name: 'Rank_1' },
  { v: 2, key: { Name: 1, Desc: 1 }, name: 'Name_1_Desc_1' }
]
jeni>
```

- **Multi-Key Index :-**

```
db.Asset_055.createIndex({"Language":1})

db.Asset_055.explain(true).find({"Language":"English"})
```

```
jeni> db.Asset_055.createIndex({"Language":1})
Language_1
jeni> db.Asset_055.explain(true).find({"Language":"English"})
{
  queryPlanner: {
    plannerVersion: 1,
    namespace: 'jeni.Asset_055',
    indexFilterSet: false,
    parsedQuery: { Language: { '$eq': 'English' } },
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { Language: 1 },
        indexName: 'Language_1',
        isMultiKey: true,
        multiKeyPaths: { Language: [ 'Language' ] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { Language: [ '["English", "English"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 39442,
    executionTimeMillis: ##
```

AndContinue…

- **Special Index : Text Index**

```
db.Asset_055.createIndex({Desc:"text"})

db.Asset_055.find({$text : {$search:"college"}})
```

```
jeni> db.Asset_055.createIndex({Desc:"text"})
Desc_text
jeni> db.Asset_055.find({$text : {$search:"college"}})

jeni>
```

2.Aggregration

- **Sum :-**

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])
[
  { _id: 'aman', num_tutorial: 9 },
  { _id: 'guru', num_tutorial: 2620 },
  { _id: 'divy', num_tutorial: 710 },
  { _id: 'het', num_tutorial: 8400 }
]
jeni>
```

- Avg:-

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])
[
  { _id: 'aman', num_tutorial: 9 },
  { _id: 'guru', num_tutorial: 655 },
  { _id: 'divy', num_tutorial: 355 },
  { _id: 'het', num_tutorial: 2800 }
]
jeni>
```

- Min:-

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])
[
  { _id: 'aman', num_tutorial: 9 },
  { _id: 'guru', num_tutorial: 20 },
  { _id: 'divy', num_tutorial: 10 },
  { _id: 'het', num_tutorial: 100 }
]
jeni>
```

- Max:-

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])
[
  { _id: 'divy', num_tutorial: 700 },
  { _id: 'het', num_tutorial: 8000 },
  { _id: 'guru', num_tutorial: 1500 },
  { _id: 'aman', num_tutorial: 9 }
]
jeni>
```

- First:-

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", first_url : {$first : "$url"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", first_url : {$first : "$url"}}}])
[
  { _id: 'aman', first_url: 'http://www.virtualdj.com' },
  { _id: 'guru', first_url: 'http://www.ganpatuniversity.ac.in' },
  { _id: 'divy', first_url: 'http://www.gnu.ac.in' },
  { _id: 'het', first_url: 'http://www.python.org' }
]
jeni>
```

- Last:-

```
db.gnu_055.aggregate([{$group: {_id : "$by_user", last_url : {$last : "$url"}}}])
```

```
jeni> db.gnu_055.aggregate([{$group: {_id : "$by_user", last_url : {$last : "$url"}}}])
[
  { _id: 'aman', last_url: 'http://www.virtualdj.com' },
  { _id: 'guru', last_url: 'https://www.mongodb.com/' },
  { _id: 'divy', last_url: 'https://learn.microsoft.com/' },
  { _id: 'het', last_url: 'http://www.DBMS.com' }
]
jeni>
```

- Exercise On Aggregation

```
db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$sum : "$total"}}}])
```

```
jeni> db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$sum : "$total"}}}])
[
  { _id: 'guru', money: 199 },
  { _id: 'tom', money: 199.99 },
  { _id: 'parth', money: 130 },
  { _id: 'divy', money: 59 },
  { _id: 'vinit', money: 20 },
  { _id: 'jay', money: 80 },
  { _id: 'rushi', money: 90 },
  { _id: 'brandy', money: 19 }
]
jeni>
```

```
db.purchase_orders_055.aggregate([{$group: {_id : "$product", money : {$avg : "$total"}}}])
```

```
jeni> db.purchase_orders_055.aggregate([{$group: {_id : "$product", money : {$avg : "$total"}}}])
[
  { _id: 'pizza', money: 60 },
  { _id: 'choco', money: 10 },
  { _id: 'cycle', money: 199 },
  { _id: 'toothpaste', money: 63 },
  { _id: 'guitar', money: 199.99 },
  { _id: 'mouse', money: 19 }
]
jeni>
```

```
db.purchase_orders_055.aggregate([{$match: {product: {$in: ["toothbrush", "pizza"]} } },{$group: {_id: "$product", total: { $sum: "$total"} } }])
```

```
jeni> db.purchase_orders_055.aggregate([{$match: {product: {$in: ["toothbrush", "pizza"]} } },{$group: {_id: "$product", total: { $sum: "$total"} } }])
[ { _id: 'pizza', total: 180 } ]
jeni>
```

```
db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$min : "$total"}}}])

db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$max : "$total"}}}])
```

```
jeni> db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$min : "$total"}}}])
[
  { _id: 'brandy', money: 19 },
  { _id: 'guru', money: 199 },
  { _id: 'tom', money: 199.99 },
  { _id: 'parth', money: 50 },
  { _id: 'vinit', money: 20 },
  { _id: 'divy', money: 59 },
  { _id: 'rushi', money: 10 },
  { _id: 'jay', money: 80 }
]
jeni>
```

```
jeni> db.purchase_orders_055.aggregate([{$group: {_id : "$customer", money : {$max : "$total"}}}])
[
  { _id: 'brandy', money: 19 },
  { _id: 'guru', money: 199 },
  { _id: 'tom', money: 199.99 },
  { _id: 'parth', money: 80 },
  { _id: 'vinit', money: 20 },
  { _id: 'divy', money: 59 },
  { _id: 'rushi', money: 80 },
  { _id: 'jay', money: 80 }
]
jeni>
```

```
db.purchase_orders_055.aggregate([{$group: {_id : "$product", money : {$sum : "$total"}}},{$sort:{total : 1}}])
```

```
jeni> db.purchase_orders_055.aggregate([{$group: {_id : "$product", money : {$sum : "$total"}}},{$sort:{total : 1}}])
[
  { _id: 'guitar', money: 199.99 },
  { _id: 'mouse', money: 19 },
  { _id: 'toothpaste', money: 189 },
  { _id: 'choco', money: 10 },
  { _id: 'cycle', money: 199 },
  { _id: 'pizza', money: 180 }
]
jeni>
```

- MAP REDUCE

```
db.runCommand({ mapReduce: "books_055", map: function () { for (var index = 0; index < this.authors.length; ++index) { var author = this.authors[index]; emit(author.firstName + " " + author.lastName, 1); } }, reduce: function (author, counters) { count = 0; for (var index = 0; index < counters.length; ++index) { count += counters[index]; } return count; }, out: { inline: 1 } })
```

```
jeni> db.runCommand( {
... mapReduce: "books_055",
... map: function() {
... for (var index = 0; index < this.authors.length; ++index) {
... var author = this.authors[ index ];
... emit( author.firstName + " " + author.lastName, 1 );
... }
... },
... reduce: function(author, counters) {
... count = 0;
... for (var index = 0; index < counters.length; ++index) {
... count += counters[index];
... }
... return count;
... },
... out: { inline: 1 }
... } )
{
  results: [
    { _id: 'Kyle Banker', value: 2 },
    { _id: 'Peter Bakkum', value: 1 },
    { _id: 'Kristina Chodorow', value: 2 },
    { _id: 'Rick Copeland', value: 1 },
    { _id: 'Tim Hawkins', value: 1 },
    { _id: 'Pramod J. Sadalage', value: 1 },
    { _id: 'Brad Dayley', value: 1 },
    { _id: 'Martin Fowler', value: 1 }
  ],
  ok: 1
}
jeni>
```

- Map Reduce Exercise

```
var city_max = function () { emit(this.city, this.temp); }

var reduce_max = function (city, temp) { var max = 0; for (var i = 0; i < temp.length; i++) { if (temp[i] > max) { max = temp[i]; } } return max; }
```

```
db.cities_055.mapReduce(city_max,reduce_max,{out:{inline:1}});
```

```
jeni> var city_max = function () { emit(this.city, this.temp); }

jeni> var reduce_max = function (city, temp) { var max = 0; for (var i = 0; i < temp.length; i++) { if (temp[i] > max) { max = temp[i]; } } return max; }

jeni> db.cities_055.mapReduce(city_max,reduce_max,{out:{inline:1}});
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{
  results: [
    { _id: 'Ahemdabad', value: 45 },
    { _id: 'Ontario', value: 0 },
    { _id: 'NewYork', value: 0 },
    { _id: 'Kitchner', value: 0 },
    { _id: 'Mehsana', value: 50 },
    { _id: 'Washington', value: 0 },
    { _id: 'Baroda', value: 48 }
  ],
  ok: 1
}
jeni>
```