Lab Session 12

12. NO SQL AGGREGATION

Aim:

The aim of this experiment is to provide a powerful and flexible framework for performing complex data processing and analysis tasks on collections. Aggregation operations allow for the transformation, grouping, filtering, and computation of data, enabling developers to derive meaningful insights from their MongoDB data.

Description:

Aggregation operations in MongoDB provide a powerful framework for performing complex data processing and analysis tasks on collections. These operations allow developers to derive meaningful insights from their MongoDB data by transforming, grouping, filtering, and computing data in a flexible and efficient manner.

Pre-Requisites: This Section contains the list of Software/Tools or required knowledge to complete the task under the Laboratory Session.

Pre lab:

1. **What is the purpose of the $redact operator in MongoDB aggregation?**

$redact is used to filter or restrict fields or documents during aggregation based on conditions (e.g., access control).

2. **What is the purpose of the $out operator in MongoDB aggregation?**

$out writes the aggregation result to a specified collection (overwrites if it exists).

3. **What is the purpose of the $lookup pipeline in MongoDB?**

$lookup performs a left outer join to combine documents from two collections based on a related field.

Inlab:

To demonstrate aggregation operations such as $avg, $min, $max, $push, and $addToSet in MongoDB, we will use a Sales collection. This collection will contain documents representing sales transactions.

📦 Sample Sales Data

```
[
{ item: "Laptop", price: 1000, quantity: 2, region: "North" }
{ item: "Tablet", price: 500, quantity: 5, region: "South" }
{ item: "Phone",  price: 700, quantity: 4, region: "East" }
]
```

🔁 Aggregation Operations

- **$avg – Average price by item:**

```
{ $group: { _id: "$item", avgPrice: { $avg: "$price" } } }
```

- **$min – Minimum price by item:**

```
{ $group: { _id: "$item", minPrice: { $min: "$price" } } }
```

- **$max – Max quantity by region:**

```
{ $group: { _id: "$region", maxQty: { $max: "$quantity" } } }
```

- **$push – List all prices per item:**

```
{ $group: { _id: "$item", prices: { $push: "$price" } } }
```

- **$addToSet – Unique items per region:**

```
{ $group: { _id: "$region", items: { $addToSet: "$item" } } }
```

Execute Aggregation Pipeline and its operations (pipeline must contain match, group, sort, project, $skip etc.)

📦 **Sample Sales Collection**

```
{ item: "Laptop", price: 1000, quantity: 2, region: "North" }
{ item: "Tablet", price: 500, quantity: 5, region: "South" }
{ item: "Laptop", price: 950, quantity: 1, region: "North" }
{ item: "Phone",  price: 700, quantity: 4, region: "East" }
{ item: "Tablet", price: 520, quantity: 3, region: "South" }
```

🔁 **Aggregation Pipeline**

```
db.Sales.aggregate([
  { $match: { region: "South" } },
  {
    $group: {
      _id: "$item",
      totalQuantity: { $sum: "$quantity" },
      avgPrice: { $avg: "$price" }
    }
  },
  { $sort: { totalQuantity: -1 } },
  {
    $project: {
      _id: 0,
      item: "$_id",
      totalQuantity: 1,
      avgPrice: 1
    }
  },
  { $skip: 1 },
  { $limit: 1 }
])
```

Viva

1. **How is data stored in collections in MongoDB?**

Data is stored as BSON documents in collections (similar to JSON format).

2. **What is the default port on which MongoDB listens?**

MongoDB listens on port 27017 by default.

3. **How does MongoDB provide high availability and disaster recovery?**

Using replica sets with automatic failover and data replication.

4. Describe how to read data from a MongoDB collection.

Use the find() method:

db.collection.find()

Post lab:

To display a summary of reviews in an e-commerce collection, we can assume the ecommerce database contains a products collection with documents structured to include reviews. Each product document could have a reviews array with review details such as rating, comment, and user.

Assumed Product Structure:

```
{
  "name": "Product Name",
  "reviews": [
    { "user": "u1", "rating": 5, "comment": "Great!" },
    { "user": "u2", "rating": 4, "comment": "Good!" }
  ]
}
```

MongoDB Aggregation:

```
db.products.aggregate([
  {
    $project: {
      name: 1,
      averageRating: { $avg: "$reviews.rating" },
      totalReviews: { $size: "$reviews" },
      ratingBreakdown: {
        $arrayToObject: {
          $map: {
            input: [1, 2, 3, 4, 5],
            as: "star",
            in: {
              k: { $toString: "$$star" },
              v: {
                $size: {
                  $filter: {
                    input: "$reviews",
                    as: "review",
                    cond: { $eq: ["$$review.rating", "$$star"] }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
])
```

Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: