MongoDB Aggregation Pipeline
Detailed Study Material

1 Introduction

The Aggregation Pipeline in MongoDB is a powerful framework used to process and transform data through multiple stages. It works similarly to data processing pipelines where each stage performs an operation on the data and passes the result to the next stage.

Definition: Aggregation pipeline is a sequence of stages that process documents and return computed results.

2 Basic Syntax

```
db.collection.aggregate([
   { stage1 },
   { stage2 },
   ...
])
```

Each stage starts with a **$ operator**.

3 Common Stages

3.1 $match

Filters documents (similar to WHERE in SQL)

```
{ $match: { salary: { $gt: 25000 } } }
```

3.2 $project

Selects specific fields and performs transformations

```
{ $project: { name: 1, salary: 1 } }
```

3.3 $group

Groups documents and performs aggregation

```
{
  $group: {
    _id: "$fav_brand",
    totalSalary: { $sum: "$salary" }
  }
}
```

3.4 $sort

Sorts documents

```
{ $sort: { salary: -1 } }
```

3.5 $limit

Limits number of documents

```
{ $limit: 5 }
```

3.6 $skip

Skips documents

```
{ $skip: 5 }
```

4 Example Pipeline

```
db.agnp.aggregate([
  { $match: { salary: { $gt: 20000 } } },
  { $group: {
      _id: "$fav_brand",
      avgSalary: { $avg: "$salary" }
    }
  },
  { $sort: { avgSalary: -1 } }
])
```

5 Working Concept

- Data flows through stages sequentially
- Each stage modifies or filters data
- Output of one stage becomes input of next stage

6 Advantages (Merits)

- Highly powerful for data transformation
- Supports complex analytics operations
- Reduces need for multiple queries
- Efficient for large datasets (optimized internally)
- Allows real-time data processing
- Supports indexing for performance improvement

7 Disadvantages (Demerits)

- Complex pipelines can become difficult to read
- Debugging is harder compared to simple queries
- Memory limitations (100MB limit without disk use)
- Requires deeper understanding of stages
- Performance issues if not optimized properly

8 Important Operators

- $sum
- $avg
- $min
- $max
- $count
- $push

9 Aggregation vs Find Query

Find Query	Aggregation Pipeline
Simple retrieval	Complex processing
Limited transformation	Advanced transformation
Easy to use	Slightly complex

10 Best Practices

- Use $match early to reduce data size
- Use indexes for filtering
- Avoid unnecessary stages
- Keep pipeline short and optimized
- Use $project to limit fields

11 Conclusion

Aggregation pipeline is a powerful tool in MongoDB for performing advanced data processing and analytics. While it may seem complex initially, mastering it provides significant advantages in handling large-scale data efficiently.