**MongoDB – Aggregation**

**E-Learning Platform – Gamification Based**

**1. Filtering – $match**

Use $match early in the pipeline to reduce the data size and improve performance.

**Q1: Get all blogs with more than 100 views.**

db.blogs.aggregate([

{ $match: { views: { $gt: 100 } } }

])

**Q2: Find users with badge "Gold".**

db.users.aggregate([

{ $match: { badge: "Gold" } }

])

**2. Field Shaping – $project, $addFields**

Restructure documents, add or rename fields, or compute new fields.

**Q3: Show only blog title and view count.**

db.blogs.aggregate([

{ $project: { title: 1, views: 1, \_id: 0 } }

])

**Q4: Add a field for doublePoints (2x points) for users.**

db.users.aggregate([

$addFields: {

doublePoints: { $multiply: ["$points", 2] }

}

}

])

**3. Aggregation – $group**

Use to group documents and calculate values like count, sum, avg, etc.

**Q5: Count number of blogs per author.**

db.blogs.aggregate([

{

$group: {

\_id: "$authorId",

blogCount: { $sum: 1 }

}

}

])

**Q6: Get average rating per blog post.**

db.comments.aggregate([

{

$group: {

\_id: "$blogId",

avgRating: { $avg: "$rating" }

}

}])

**4. Array Handling – $unwind**

Use $unwind to flatten arrays before further processing.

**Q7: List each suggested blog per user (AI suggestions).**

db.ai\_suggestions.aggregate([

{ $unwind: "$suggestedBlogs" },

{

$project: {

userId: 1,

blogId: "$suggestedBlogs"

}

}

])

**Q8: List every blog in user learning paths.**

db.learning\_paths.aggregate([

{ $unwind: "$blogs" },

{

$project: {

userId: 1,

blogId: "$blogs"

}

}

])

**5. Pagination & Sorting – $sort, $skip, $limit**

**Q9: Show top 5 most viewed blogs.**

db.blogs.aggregate([

{ $sort: { views: -1 } },

{ $limit: 5 }

])

**Q10: Get page 2 of blogs with 5 per page.**

db.blogs.aggregate([

{ $skip: 5 },

{ $limit: 5 }

])

**6. Analytics – $count, $bucket, $facet**

**Q11: Count number of readers.**

db.users.aggregate([

{ $match: { role: "reader" } },

{ $count: "totalReaders" }

])

**Q12: Bucket blogs by view count.**

db.blogs.aggregate([

{

$bucket: {

groupBy: "$views",

boundaries: [0, 50, 100, 200, 500],

default: "Above 500",

output: {

count: { $sum: 1 }

}

}

}

])

**Q13: Run facet for trending and new blogs.**

db.blogs.aggregate([

{

$facet: {

trending: [{ $match: { views: { $gt: 100 } } }],

recent: [{ $sort: { createdAt: -1 } }, { $limit: 5 }]

}

}

])

**7. Joins – $lookup**

**Q14: Join blogs with author details.**

db.blogs.aggregate([

{

$lookup: {

from: "users",

localField: "authorId",

foreignField: "\_id",

as: "authorDetails"

}

}

])

**Q15: Join leaderboard with user profile.**

db.leaderboards.aggregate([

{

$lookup: {

from: "users",

localField: "userId",

foreignField: "\_id",

as: "userProfile"

}

}

])

**8. Advanced – $merge, $replaceRoot, $setWindowFields**

**Q16: Merge user scores into a new collection.**

db.leaderboards.aggregate([

{

$group: {

\_id: "$userId",

totalPoints: { $sum: "$points" }

}

},

{ $merge: "user\_score\_summary" }

])