## **MongoDB Aggregation Queries – Culinary Skills Academy**

### **1. Filtering – $match**

**Q1: Get all courses with duration more than 4 hours**

db.courses.aggregate([

{ $match: { duration: { $gt: "4h" } } }

])

**Q2: Find users from city "Chennai"**

db.users.aggregate([

{ $match: { "address.city": "Chennai" } }

])

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

**Q3: Show only course title and duration**

db.courses.aggregate([

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

])

**Q4: Add a taxAdjustedPrice field (assume 18% tax)**

db.recipes.aggregate([

{

$addFields: {

taxAdjustedPrice: { $multiply: ["$price", 1.18] }

}

}

])

### **3. Aggregation – $group**

**Q5: Count number of users per skill level**

db.users.aggregate([

{

$group: {

\_id: "$skill\_level",

userCount: { $sum: 1 }

}

}

])

**Q6: Get average feedback score per user**

db.assignments.aggregate([

{

$group: {

\_id: "$user\_id",

avgScore: { $avg: "$feedback.score" }

}

}

])

### **4. Array Handling – $unwind**

**Q7: List every enrolled course per user**

db.users.aggregate([

{ $unwind: "$enrolled\_courses" },

{ $project: { user\_id: 1, course: "$enrolled\_courses" } }

])

**Q8: List each module of a course**

db.courses.aggregate([

{ $unwind: "$modules" },

{ $project: { course\_id: "$course\_id", module: "$modules" } }

])

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

**Q9: Show top 5 longest courses**

db.courses.aggregate([

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

{ $limit: 5 }

])

**Q10: Get page 2 with 5 recipes per page**

db.recipes.aggregate([

{ $skip: 5 },

{ $limit: 5 }

])

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

**Q11: Count total users**

db.users.aggregate([

{ $count: "totalUsers" }

])

**Q12: Bucket recipes into difficulty levels**

db.recipes.aggregate([

{

$bucket: {

groupBy: "$difficulty",

boundaries: ["Easy", "Medium", "Hard"],

default: "Unknown",

output: { count: { $sum: 1 } }

}

}

])

**Q13: Separate expensive and cheap recipes**

db.recipes.aggregate([

{

$facet: {

expensive: [{ $match: { price: { $gt: 500 } } }],

cheap: [{ $match: { price: { $lt: 200 } } }]

}

}

])

### **7. Joins – $lookup**

**Q14: Join recipes with related course**

db.recipes.aggregate([

{

$lookup: {

from: "courses",

localField: "linked\_course",

foreignField: "course\_id",

as: "courseInfo"

}

}

])

**Q15: Join assignments with user info**

db.assignments.aggregate([

{

$lookup: {

from: "users",

localField: "user\_id",

foreignField: "user\_id",

as: "userInfo"

}

}

])

### **8. Advanced – $merge**

**Q16: Merge user certification summary to new collection**

db.certifications.aggregate([

{

$group: {

\_id: "$user\_id",

totalCerts: { $sum: 1 }

}

},

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

])