**MongoDB – Aggregation Queries for Fitness App (FITIFYX)**

# **1. users**

### **Q1: Most recently joined user**

db.users.aggregate([

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

{ $limit: 1 }

])

### **Q2: List verified users with email domains**

db.users.aggregate([

{ $match: { isVerified: true } },

{ $project: {

username: 1,

emailDomain: { $arrayElemAt: [{ $split: ["$email", "@"] }, 1] }

}}

])

### **Q3: Count users grouped by email domain**

db.users.aggregate([

{ $project: {

domain: { $arrayElemAt: [{ $split: ["$email", "@"] }, 1] }

}},

{ $group: { \_id: "$domain", count: { $sum: 1 } } }

])

# **2. profiles**

### **Q4: Distribution of users by age group**

db.profiles.aggregate([

{ $project: {

age: { $dateDiff: { startDate: "$dob", endDate: "$$NOW", unit: "year" } }

}},

{ $bucket: {

groupBy: "$age",

boundaries: [0, 18, 25, 35, 50, 100],

default: "Unknown",

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

}}

])

### **Q5: Profiles created and updated on the same day**

db.profiles.aggregate([

{ $project: {

sameDay: {

$eq: [

{ $dateToString: { format: "%Y-%m-%d", date: "$createdAt" } },

{ $dateToString: { format: "%Y-%m-%d", date: "$updatedAt" } }

]

}

}},

{ $match: { sameDay: true } }

])

### **Q6: Count of users by city**

db.profiles.aggregate([

{ $group: { \_id: "$location", count: { $sum: 1 } } }

])

# **3. challenges**

### **Q7: Challenges active today**

db.challenges.aggregate([

{ $match: {

startDate: { $lte: new Date() },

endDate: { $gte: new Date() }

}}

])

### **Q8: Longest duration challenges**

db.challenges.aggregate([

{ $project: {

title: 1,

durationDays: {

$dateDiff: { startDate: "$startDate", endDate: "$endDate", unit: "day" }

}

}},

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

{ $limit: 5 }

])

### **Q9: Challenges grouped by goal type**

db.challenges.aggregate([

{ $group: { \_id: "$goalType", count: { $sum: 1 } } }

])

# **4. posts**

### **Q10: Posts containing media**

db.posts.aggregate([

{ $match: { mediaURL: { $ne: null } } },

{ $count: "mediaPosts" }

])

### **Q11: Users who have commented on their own post**

db.posts.aggregate([

{ $match: { $expr: {

$in: ["$userId", "$comments.userId"]

}}}

])

### **Q12: Extract users with most comments across posts**

db.posts.aggregate([

{ $unwind: "$comments" },

{ $group: {

\_id: "$comments.userId",

totalComments: { $sum: 1 }

}},

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

{ $limit: 5 }

])

# **5. leaderboards**

### **Q13: Challenges with more than 5 users ranked**

db.leaderboards.aggregate([

{ $project: {

challengeId: 1,

rankCount: { $size: "$rankings" }

}},

{ $match: { rankCount: { $gt: 5 } } }

])

### **Q14: User rank distribution**

db.leaderboards.aggregate([

{ $unwind: "$rankings" },

{ $group: {

\_id: "$rankings.rank",

frequency: { $sum: 1 }

}}

])

### **Q15: Score variance within each leaderboard**

db.leaderboards.aggregate([

{ $unwind: "$rankings" },

{ $group: {

\_id: "$challengeId",

maxScore: { $max: "$rankings.score" },

minScore: { $min: "$rankings.score" }

}},

{ $project: {

scoreVariance: { $subtract: ["$maxScore", "$minScore"] }

}}

])

# **6. teams**

### **Q16: Teams with more than 3 members**

db.teams.aggregate([

{ $project: {

name: 1,

memberCount: { $size: "$members" }

}},

{ $match: { memberCount: { $gt: 3 } } }

])

### **Q17: Total calories grouped by challenge**

db.teams.aggregate([

{ $group: {

\_id: "$challengeId",

totalCalories: { $sum: "$totalCalories" }

}}

])

### **Q18: Average team size**

db.teams.aggregate([

{ $project: { memberCount: { $size: "$members" } } },

{ $group: { \_id: null, avgSize: { $avg: "$memberCount" } } }

])

# **7. progress\_logs**

### **Q19: User with highest average steps per log**

db.progress\_logs.aggregate([

{ $group: {

\_id: "$userId",

avgSteps: { $avg: "$steps" }

}},

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

{ $limit: 1 }

])

### **Q20: Total distance walked per challenge**

db.progress\_logs.aggregate([

{ $group: {

\_id: "$challengeId",

totalDistance: { $sum: "$distance" }

}}

])

### **Q21: Daily log counts by date**

db.progress\_logs.aggregate([

{ $group: {

\_id: { $dateToString: { format: "%Y-%m-%d", date: "$date" } },

count: { $sum: 1 }

}},

{ $sort: { \_id: 1 } }

])

# **8. rewards**

### **Q22: Rewards with no users yet**

db.rewards.aggregate([

{ $match: { earnedBy: { $size: 0 } } }

])

### **Q23: Rewards grouped by points required range**

db.rewards.aggregate([

{ $bucket: {

groupBy: "$pointsRequired",

boundaries: [0, 500, 1000, 2000, 5000],

default: "5000+",

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

}}

])

### **Q24: Most recently updated rewards**

db.rewards.aggregate([

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

{ $limit: 3 }

])

# **9. notifications**

### **Q25: Notification types sent in the past 7 days**

db.notifications.aggregate([

{ $match: { createdAt: { $gte: new Date(Date.now() - 7\*24\*60\*60\*1000) } } },

{ $group: { \_id: "$type", count: { $sum: 1 } } }

])

### **Q26: Average notifications per user**

db.notifications.aggregate([

{ $group: {

\_id: "$userId",

total: { $sum: 1 }

}},

{ $group: {

\_id: null,

avgPerUser: { $avg: "$total" }

}}

])

### **Q27: Percentage of read vs unread**

db.notifications.aggregate([

{ $group: {

\_id: "$isRead",

count: { $sum: 1 }

}},

{ $group: {

\_id: null,

total: { $sum: "$count" },

data: { $push: { isRead: "$\_id", count: "$count" } }

}},

{ $unwind: "$data" },

{ $project: {

isRead: "$data.isRead",

percent: { $multiply: [{ $divide: ["$data.count", "$total"] }, 100] }

}}

])

# **10. activity\_trackers**

### **Q28: Devices that haven't synced in last 24 hours**

db.activity\_trackers.aggregate([

{ $match: { lastSynced: { $lt: new Date(Date.now() - 24\*60\*60\*1000) } } }

])

### **Q29: Average calories per sync session**

db.activity\_trackers.aggregate([

{ $group: {

\_id: null,

avgCalories: { $avg: "$syncData.totalCalories" }

}}

])

### **Q30: Providers used across all users**

db.activity\_trackers.aggregate([

{ $group: { \_id: "$provider", count: { $sum: 1 } } }

])