MongoDB Interview Case Studies & Real-World Scenarios (Part 3)

🔹 Section 12: Query Scenarios

Q56. Find users with age between 20 and 30.

```
db.users.find({ age: { $gte: 20, $lte: 30 } })
```

Q57. Retrieve top 3 highest salaries from employees.

```
db.employees.find().sort({ salary: -1 }).limit(3)
```

Q58. Find all users whose name starts with “A”.

```
db.users.find({ name: { $regex: /^A/, $options: "i" } })
```

Q59. Find all users who don’t have an email field.

```
db.users.find({ email: { $exists: false } })
```

🔹 Section 13: Aggregation Case Studies

Q60. Find average salary per department.

```
db.employees.aggregate([
  { $group: { _id: "$department", avgSalary: { $avg: "$salary" } } }
])
```

Q61. Count number of orders placed by each customer.

```
db.orders.aggregate([
  { $group: { _id: "$customerId", totalOrders: { $count: {} } } }
])
```

Q62. Find total sales per month.

```
db.sales.aggregate([
  { $group: { _id: { $month: "$orderDate" }, total: { $sum: "$amount" } }
}
])
```

Q63. Find top 5 most sold products.

```
db.sales.aggregate([
  { $group: { _id: "$product", totalSold: { $sum: 1 } } },
  { $sort: { totalSold: -1 } },
  { $limit: 5 }
])
```

🔹 Section 14: Join & Lookup Scenarios

- **Q64. List all orders with customer info.**

```
db.orders.aggregate([
  { $lookup: {
    from: "customers",
    localField: "customerId",
    foreignField: "_id",
    as: "customerInfo"
  }}
])
```

- Q65. Find customers who placed orders but haven’t made payments.

```
db.customers.aggregate([
 { $lookup: {
  from: "payments",
  localField: "_id",
  foreignField: "customerId",
  as: "payments"
 }},
 { $match: { payments: { $eq: [] } } }
])
```

🔹 Section 15: Real-World Scenarios

- Q66. Find users who logged in during the last 7 days.

```
db.users.find({
 lastLogin: { $gte: new Date(Date.now() - 7*24*60*60*1000) }
})
```

- Q67. Detect duplicate email IDs in users collection.

```
db.users.aggregate([
 { $group: { _id: "$email", count: { $sum: 1 } } },
 { $match: { count: { $gt: 1 } } }
])
```

- Q68. Find gaps in invoice numbers (missing sequence).

```
db.invoices.aggregate([
  { $sort: { invoiceNo: 1 } },
  { $group: {
    _id: null,
    missing: { $push: "$invoiceNo" }
  }}
])
```

- Q69. Get total revenue per region (pivot style).

```
db.sales.aggregate([
  { $group: {
    _id: "$region",
    totalRevenue: { $sum: "$amount" }
  }}
])
```