**ASSIGNMENT**

**Q1. Insert a single document into a collection named students:**

**QUERY**

db.students.insertOne({ name: "John", age: 20, course: "Math" })

**Q2. Insert multiple documents into the employees collection:**

**QUERY**

db.employees.insertMany([

{ name: "Alice", salary: 35000, department: "HR" },

{ name: "Bob", salary: 40000, department: "IT" }

])

**Q3. Find all documents from the products collection:**

**QUERY**

db.products.find()

**Q4. Find users where age > 25:**

**QUERY**

db.users.find({ age: { $gt: 25 } })

**Q5. Find orders where status is "pending" or "shipped":**

**QUERY**

db.orders.find({ status: { $in: ["pending", "shipped"] } })

**Q6. Update email of user with username "john\_doe":**

**QUERY**

db.users.updateOne(

{ username: "john\_doe" },

{ $set: { email: "newemail@example.com" } }

)

**Q7. Delete a student where roll is 101:**

**QUERY**

db.students.deleteOne({ roll: 101 })

**Q8. Find employees with salary ≥ 30000:**

**QUERY**

db.employees.find({ salary: { $gte: 30000 } })

**Q9. Find books by "Chetan Bhagat" and published after 2010:**

**QUERY**

db.books.find({ author: "Chetan Bhagat", publishedYear: { $gt: 2010 } })

**Q10. Count customers in "Delhi":**

**QUERY**

db.customers.countDocuments({ city: "Delhi" })

**Q11. Find first 5 users using limit():**

**QUERY**

db.users.find().limit(5)

**Q12. Skip first 10, retrieve next 5 from logs:**

**QUERY**

db.logs.find().skip(10).limit(5)

**Q13. Sort products by price ascending:**

**QUERY**

db.products.find().sort({ price: 1 })

**Q14. Sort users by createdAt descending:**

**QUERY**

db.users.find().sort({ createdAt: -1 })

**Q15. Retrieve only name and email fields (hide \_id):**

**QUERY**

db.users.find({}, { name: 1, email: 1, \_id: 0 })

**Q16. Find students with marks between 60 and 90:**

**QUERY**

db.students.find({ marks: { $gte: 60, $lte: 90 } })

**Q17. Find sales where amount < 500 or > 5000:**

**QUERY**

db.sales.find({

$or: [

{ amount: { $lt: 500 } },

{ amount: { $gt: 5000 } }

]

})

**Q18. Update status to "completed" where deliveryDate is not null:**

**QUERY**

db.orders.updateMany(

{ deliveryDate: { $ne: null } },

{ $set: { status: "completed" } }

)

**Q19. Delete inactive users (active = false):**

**QUERY**

db.users.deleteMany({ active: false })

**Q20. Find users from "Bangalore" or age > 30:**

**QUERY**

db.users.find({

$or: [

{ city: "Bangalore" },

{ age: { $gt: 30 } }

]

})