**1. Creating the MongoDB Collection:**

JavaScript

// Assuming you're using the MongoDB Node.js driver (mongodb)

const MongoClient = require('mongodb').MongoClient;

const uri = "mongodb://localhost:27017"; // Replace with your MongoDB connection string

async function createCollection() {

const client = new MongoClient(uri);

try {

await client.connect();

const database = client.db("your\_database\_name");

const collection = database.collection("employees");

// Insert sample data

const data = [

{ employee\_id: 1, first\_name: "John", last\_name: "Doe", department: "IT", salary: 60000, hire\_date: "2021-05-15", position: "Software Engineer" },

// ... other employee data

];

const result = await collection.insertMany(data);

console.log(`${result.insertedCount} documents were inserted`);

} finally {

await client.close();

}

}

createCollection();

Use code [with caution.](/faq#coding)

**2. MongoDB Queries:**

JavaScript

// Assuming you have a MongoDB client connected to the database

**// 1. Find All Employees**

const allEmployees = await collection.find().toArray();

**// 2. Find Employees in the IT department**

const itEmployees = await collection.find({ department: "IT" }).toArray();

**// 3. Find Employees in Finance department with salary greater than 85000**

const financeEmployeesHighSalary = await collection.find({ department: "Finance", salary: { $gt: 85000 } }).toArray();

**// 4. Count the Number of Employees in Each Department**

const departmentCounts = await collection.aggregate([

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

]).toArray();

**// 5. Calculate the Average Salary in Each Department**

const averageSalaries = await collection.aggregate([

{ $group: { \_id: "$department", averageSalary: { $avg: "$salary" } } }

]).toArray();

**// 6. Find Employees Hired After a Certain Date**

const recentHired = await collection.find({ hire\_date: { $gt: new Date("2021-01-01") } }).toArray();

**// 7. Update the Salary of All Employees in the IT Department by Adding 50000**

const updateITSalaries = await collection.updateMany({ department: "IT" }, { $inc: { salary: 50000 } });

**// 8. Delete an Employee Record by employee\_id=6**

const deleteEmployee = await collection.deleteOne({ employee\_id: 6 });

**// 9. Find the Highest Salary in Each Department**

const highestSalaries = await collection.aggregate([

{ $group: { \_id: "$department", highestSalary: { $max: "$salary" } } }

]).toArray();

**// 10. Count the Number of Employees in Each Department with More Than 1 Employee**

const departmentsWithMultipleEmployees = await collection.aggregate([

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

{ $match: { count: { $gt: 1 } } }

]).toArray();