MongoDB_Practical_Slips_Answers

SLIP 1

Here are five sample documents for the "Properties" collection.

```
[
  { "_id": ObjectId("64c9a1b1f6291e7d5c7a52b1"), "address": "123 Elm St", "area": "Downtown", "rate": 500000,"owner": { "name": "Mr. Patil", "contact": "1234567890" } },

  { "_id": ObjectId("64c9a1b1f6291e7d5c7a52b2"), "address": "456 Oak St", "area": "Uptown", "rate": 300000, "owner": { "name": "Ms. Sharma", "contact": "0987654321" } },

  { "_id": ObjectId("64c9a1b1f6291e7d5c7a52b3"), "address": "789 Pine St", "area": "Suburb", "rate": 200000, "owner": { "name": "Mr. Khan", "contact": "1231231231" } },

  { "_id": ObjectId("64c9a1b1f6291e7d5c7a52b4"), "address": "101 Maple St", "area": "Downtown", "rate": 150000, "owner": { "name": "Ms. Singh", "contact": "9879879879" } },

  { "_id": ObjectId("64c9a1b1f6291e7d5c7a52b5"), "address": "202 Birch St", "area": "Nashik", "rate": 100000, "owner": { "name": "Mr. Desai", "contact": "4564564564" } }
]
```

---

a. Display area wise property details.

```
db.Properties.aggregate([
  {
    $group: {
      _id: "$area",
      properties: { $push: { address: "$address", rate: "$rate", owner: "$owner" } }
    }
  }
])
```

---

b. Display property owned by 'Mr. Patil' having minimum rate.

```
db.Properties.find(
  { "owner.name": "Mr. Patil" },
  { _id: 0, address: 1, rate: 1 }
).sort({ rate: 1 }).limit(1)
```

---

c. Give the details of owner whose property is at “Nashik”.

```
db.Properties.find(
  { area: "Nashik" },
  { _id: 0, "owner.name": 1, "owner.contact": 1 }
)
```

---

d. Display area of property whose rate is less than 100000.

```
db.Properties.find(
  { rate: { $lt: 100000 } },
  { _id: 0, area: 1 }
)
```

==========

SLIP2

Here are five sample documents for the Newspapers collection.

```
[
  { "_id": ObjectId("64c9a1b1f6291e7d5c7a53b1"), "title": "Daily Nashik Times", "language": "Marathi", "publisher": { "name": "ABC Publications", "state": "Maharashtra" }, "city": "Nashik", "sales": 50000 },
  { "_id": ObjectId("64c9a1b1f6291e7d5c7a53b2"), "title": "Mumbai Mirror", "language": "English",
```

"publisher": { "name": "XYZ Media", "state": "Maharashtra" }, "city": "Mumbai", "sales": 120000 },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a53b3"), "title": "Ahmedabad Chronicle", "language": "Gujarati",
"publisher": { "name": "PQR Publishers", "state": "Gujarat" }, "city": "Ahmedabad", "sales": 30000 },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a53b4"), "title": "Pune Daily", "language": "Marathi",
"publisher": { "name": "LMN Media", "state": "Maharashtra" }, "city": "Pune", "sales": 80000 },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a53b5"), "title": "Surat News", "language": "Gujarati",
"publisher": { "name": "DEF Publications", "state": "Gujarat" }, "city": "Surat", "sales": 45000 }
]

---

a. List all newspapers available in “NASHIK” city.

```
db.Newspapers.find(
  { city: "Nashik" },
  { _id: 0, title: 1 }
)
```

---

b. List all the newspapers of “Marathi” language.

```
db.Newspapers.find(
  { language: "Marathi" },
  { _id: 0, title: 1 }
)
```

---

c. Count the number of publishers from the “Gujarat” state.

```
db.Newspapers.distinct(
  "publisher.name",
  { "publisher.state": "Gujarat" }
).length
```

---

d. Write a cursor to show newspapers with the highest sale in Maharashtra State.

```
db.Newspapers.find(
  { "publisher.state": "Maharashtra" },
  { _id: 0, title: 1, sales: 1 }
).sort({ sales: -1 }).limit(1)
```

==========

SLIP 3

Here are five sample documents for the Employees collection.

[
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a54c1"), "name": "Alice", "salary": 75000, "department": { "name": "Sales", "employees_count": 10 } },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a54c2"), "name": "Bob", "salary": 80000, "department": { "name": "Marketing", "employees_count": 8 } },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a54c3"), "name": "Charlie", "salary": 95000, "department": { "name": "Sales", "employees_count": 10 } },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a54c4"), "name": "David", "salary": 65000, "department": { "name": "HR", "employees_count": 5 } },
{ "_id": ObjectId("64c9a1b1f6291e7d5c7a54c5"), "name": "Eve", "salary": 55000, "department": { "name": "Sales", "employees_count": 10 } }
]

---

a. Display name of the employee who has the highest salary.

```
db.Employees.find(
  {},
  { _id: 0, name: 1, salary: 1 }
).sort({ salary: -1 }).limit(1)
```

---

b. Display the department with the maximum number of employees.

```
db.Employees.aggregate([
  {
```

```
    $group: {
      _id: "$department.name",
      total_employees: { $sum: 1 }
    }
  },
  {
    $sort: { total_employees: -1 }
  },
  {
    $limit: 1
  }
])
```

---

c. Write a cursor which shows department-wise employee information.

```
db.Employees.aggregate([
  {
    $group: {
      _id: "$department.name",
      employees: {
        $push: {
          name: "$name",
          salary: "$salary"
        }
      }
    }
  }
])
```

---

d. List all the employees who work in the Sales department and have a salary greater than 50000.

```
db.Employees.find(
  { "department.name": "Sales", salary: { $gt: 50000 } },
  { _id: 0, name: 1, salary: 1 }
)
```

==========

SLIP4

Here are ten sample documents for the Hospitals collection.

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a55c1"),
    "name": "Nashik General Hospital",
    "city": "Nashik",
    "specializations": ["Pediatric", "Orthopedic"],
    "rating": 4.5,
    "doctors": ["Dr. Deshmukh", "Dr. Patel"],
    "reviews": [
      { "reviewer_name": "John Doe", "review_text": "Excellent service!" },
      { "reviewer_name": "Jane Smith", "review_text": "Very professional staff." }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a55c2"),
    "name": "Nashik Women’s Hospital",
    "city": "Nashik",
    "specializations": ["Gynaec"],
    "rating": 4.0,
    "doctors": ["Dr. Rao"],
    "reviews": [
      { "reviewer_name": "Alice Johnson", "review_text": "Great care during my visit." }
```

```
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c3"),
  "name": "Nashik Orthopedic Center",
  "city": "Nashik",
  "specializations": ["Orthopedic"],
  "rating": 4.2,
  "doctors": ["Dr. Deshmukh", "Dr. Kumar"],
  "reviews": [
    { "reviewer_name": "Bob Brown", "review_text": "Very knowledgeable doctors." }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c4"),
  "name": "Central Hospital Nashik",
  "city": "Nashik",
  "specializations": ["Pediatric", "Gynaec", "Orthopedic"],
  "rating": 3.9,
  "doctors": ["Dr. Sharma"],
  "reviews": [
    { "reviewer_name": "Emily Davis", "review_text": "Good hospital but waiting time is long." }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c5"),
  "name": "Eastside Health Clinic",
  "city": "Nashik",
  "specializations": ["Pediatric"],
  "rating": 4.1,
  "doctors": ["Dr. Singh"],
  "reviews": [
    { "reviewer_name": "Michael Wilson", "review_text": "Very caring staff." }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c6"),
  "name": "West End Hospital",
  "city": "Pune",
  "specializations": ["Orthopedic"],
  "rating": 4.3,
  "doctors": ["Dr. Patel"],
  "reviews": [
    { "reviewer_name": "Sophia Lee", "review_text": "Highly recommend this hospital." }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c7"),
  "name": "Southside Medical Center",
  "city": "Nashik",
  "specializations": ["Gynaec"],
  "rating": 4.4,
  "doctors": ["Dr. Deshmukh", "Dr. Verma"],
  "reviews": [
    { "reviewer_name": "Liam Martinez", "review_text": "Excellent facilities and services." }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a55c8"),
  "name": "North Valley Clinic",
  "city": "Nashik",
```

```
    "specializations": ["Orthopedic"],
    "rating": 4.1,
    "doctors": ["Dr. Rao"],
    "reviews": [
      { "reviewer_name": "Olivia Moore", "review_text": "Good experience overall." }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a55c9"),
    "name": "Healthcare Hub",
    "city": "Nashik",
    "specializations": ["Pediatric", "Gynaec"],
    "rating": 4.0,
    "doctors": ["Dr. Kumar"],
    "reviews": [
      { "reviewer_name": "James Anderson", "review_text": "Great pediatric care." }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a55c10"),
    "name": "City Care Hospital",
    "city": "Nashik",
    "specializations": ["Orthopedic"],
    "rating": 3.8,
    "doctors": ["Dr. Deshmukh"],
    "reviews": [
      { "reviewer_name": "Ava Jackson", "review_text": "Good service, but can improve." }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a55c11"),
    "name": "Nashik Medical Clinic",
    "city": "Nashik",
    "specializations": ["Pediatric"],
    "rating": 4.6,
    "doctors": ["Dr. Singh"],
    "reviews": [
      { "reviewer_name": "Noah Harris", "review_text": "Outstanding care and support." }
    ]
  }
]
```

---

3. Queries

a. List the names of hospitals with a specific specialization (e.g., "Orthopedic").

```
db.Hospitals.find(
  { specializations: "Orthopedic" },
  { _id: 0, name: 1 }
)
```

---

b. List the names of all hospitals located in a specific city (e.g., "Nashik").

```
db.Hospitals.find(
  { city: "Nashik" },
  { _id: 0, name: 1 }
)
```

---

c. List the names of hospitals where Dr. Deshmukh visits.

```
db.Hospitals.find(
  { doctors: "Dr. Deshmukh" },
```

```
  { _id: 0, name: 1 }
)
```

---

d. List the names of hospitals whose rating is greater than or equal to 4.

```
db.Hospitals.find(
  { rating: { $gte: 4 } },
  { _id: 0, name: 1 }
)
```

==========

SLIP 5

Here are five sample documents for the Projects collection.

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a56c1"),
    "name": "Project Alpha",
    "type": "Development",
    "duration_months": 6,
    "employees": [
      { "name": "Mr. Patil", "role": "Developer" },
      { "name": "Ms. Sharma", "role": "Tester" }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a56c2"),
    "name": "Project Beta",
    "type": "Research",
    "duration_months": 4,
    "employees": [
      { "name": "Mr. Deshmukh", "role": "Researcher" },
      { "name": "Mr. Patil", "role": "Analyst" }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a56c3"),
    "name": "Project Gamma",
    "type": "Development",
    "duration_months": 2,
    "employees": [
      { "name": "Ms. Singh", "role": "Developer" }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a56c4"),
    "name": "Project Delta",
    "type": "Marketing",
    "duration_months": 5,
    "employees": [
      { "name": "Mr. Patil", "role": "Marketing Manager" },
      { "name": "Ms. Rao", "role": "Marketing Specialist" }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a56c5"),
    "name": "Project Epsilon",
    "type": "Development",
    "duration_months": 3,
    "employees": [
      { "name": "Ms. Sharma", "role": "Developer" },
      { "name": "Mr. Kumar", "role": "Tester" }
```

```
    ]
  }
]
```
---
a. List all names of projects where type is a specific value (e.g., "Development").
```
db.Projects.find(
  { type: "Development" },
  { _id: 0, name: 1 }
)
```
---
b. List all the projects with a duration greater than 3 months.
```
db.Projects.find(
  { duration_months: { $gt: 3 } },
  { _id: 0, name: 1 }
)
```
---
c. Count the number of employees working on a specific project (e.g., "Project Alpha").
```
db.Projects.findOne(
  { name: "Project Alpha" },
  { _id: 0, employees: 1 }
).employees.length
```
---
d. List the names of projects on which Mr. Patil is working.
```
db.Projects.find(
  { "employees.name": "Mr. Patil" },
  { _id: 0, name: 1 }
)
```
==========

SLIP 6

Here are five sample documents for the Policies collection.
```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a57c1"),
    "customer_name": "John Doe",
    "policy_name": "Komal Jeevan",
    "policy_type": "Monthly",
    "premium_amount": 5000,
    "company": "ABC Insurance"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a57c2"),
    "customer_name": "Jane Smith",
    "policy_name": "Health Guard",
    "policy_type": "Yearly",
    "premium_amount": 15000,
    "company": "XYZ Insurance"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a57c3"),
    "customer_name": "Alice Johnson",
    "policy_name": "Komal Jeevan",
    "policy_type": "Half Yearly",
    "premium_amount": 10000,
    "company": "ABC Insurance"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a57c4"),
    "customer_name": "Bob Brown",
    "policy_name": "Life Secure",
    "policy_type": "Monthly",
    "premium_amount": 6000,
```

```
    "company": "DEF Insurance"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a57c5"),
    "customer_name": "Emily Davis",
    "policy_name": "Komal Jeevan",
    "policy_type": "Yearly",
    "premium_amount": 12000,
    "company": "ABC Insurance"
  }
]
```

---

3. Queries

a. List the details of customers who have taken the “Komal Jeevan” Policy.

```
db.Policies.find(
  { policy_name: "Komal Jeevan" }
)
```

---

b. Display the average premium amount.

```
db.Policies.aggregate([
  {
    $group: {
      _id: null,
      average_premium: { $avg: "$premium_amount" }
    }
  }
])
```

---

c. Increase the premium amount by 5% for policy type “Monthly”.

```
db.Policies.updateMany(
  { policy_type: "Monthly" },
  { $mul: { premium_amount: 1.05 } }
)
```

---

d. Count the number of customers who have taken a policy type “Half Yearly”.

```
db.Policies.countDocuments(
  { policy_type: "Half Yearly" }
)
```

==========

SLIP 7

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a51a1"),
    "first_name": "Sam",
    "last_name": "Smith",
    "address": "123 Elm St",
    "contact": "1234567890",
    "accounts": [
      {
        "account_number": "ACC123",
        "account_type": "Saving",
        "branch": "Main Branch",
        "opened_on": ISODate("2020-01-01")
      }
    ]
  },
```

```
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a51a2"),
    "first_name": "Sarah",
    "last_name": "Johnson",
    "address": "456 Oak St",
    "contact": "0987654321",
    "accounts": [
      {
        "account_number": "ACC124",
        "account_type": "Checking",
        "branch": "Main Branch",
        "opened_on": ISODate("2020-01-01")
      }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a51a3"),
    "first_name": "Mike",
    "last_name": "Brown",
    "address": "789 Pine St",
    "contact": "1231231231",
    "accounts": [
      {
        "account_number": "ACC125",
        "account_type": "Saving",
        "branch": "West Branch",
        "opened_on": ISODate("2021-02-15")
      }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a51a4"),
    "first_name": "Sally",
    "last_name": "Green",
    "address": "101 Maple St",
    "contact": "9879879879",
    "accounts": [
      {
        "account_number": "ACC126",
        "account_type": "Loan",
        "branch": "East Branch",
        "opened_on": ISODate("2019-11-30")
      }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a51a5"),
    "first_name": "John",
    "last_name": "Doe",
    "address": "202 Birch St",
    "contact": "4564564564",
    "accounts": [
      {
        "account_number": "ACC127",
        "account_type": "Saving",
        "branch": "Main Branch",
        "opened_on": ISODate("2020-01-01")
      }
    ]
  }
]
```

—----------

a. List names of all customers whose first name starts with a “S”.

```
db.Customers.find(
  { first_name: { $regex: '^S', $options: 'i' } },
  { _id: 0, first_name: 1, last_name: 1 }
)
```

—----------

b. List all customers who opened an account on 1/1/2020 in a specific branch (e.g., "Main Branch").

```
db.Customers.find(
  {
    "accounts": {
      $elemMatch: {
        opened_on: ISODate("2020-01-01"),
        branch: "Main Branch"
      }
    }
  },
  { _id: 0, first_name: 1, last_name: 1 }
)
```

—----------

c. List the names of customers where account_type = "Saving".

```
db.Customers.find(
  { "accounts.account_type": "Saving" },
  { _id: 0, first_name: 1, last_name: 1 }
)
```

—----------

d. Count total number of loan account holders in a specific branch (e.g., "East Branch").

```
db.Customers.find({
  "accounts": {
    $elemMatch: {
      branch: "East Branch",
      account_type: "Loan"
    }
  }
}).count();
```

==========

SLIP 8:

Here are five sample documents for the Inventory collection.

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58c1"),
    "item_name": "Planner",
    "tags": ["office", "stationery"],
    "quantity": 400,
    "status": "A",
    "height": 10,
    "warehouses": [
      { "warehouse_name": "Warehouse A", "stock_quantity": 15 },
      { "warehouse_name": "Warehouse B", "stock_quantity": 25 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58c2"),
    "item_name": "Notebook",
    "tags": ["office", "stationery", "education"],
    "quantity": 200,
    "status": "B",
    "height": 8,
    "warehouses": [
      { "warehouse_name": "Warehouse A", "stock_quantity": 10 },
```

```
      { "warehouse_name": "Warehouse C", "stock_quantity": 30 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58c3"),
    "item_name": "Desk Lamp",
    "tags": ["office", "furniture"],
    "quantity": 50,
    "status": "A",
    "height": 12,
    "warehouses": [
      { "warehouse_name": "Warehouse B", "stock_quantity": 5 },
      { "warehouse_name": "Warehouse C", "stock_quantity": 15 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58c4"),
    "item_name": "Chair",
    "tags": ["furniture", "office"],
    "quantity": 100,
    "status": "C",
    "height": 18,
    "warehouses": [
      { "warehouse_name": "Warehouse A", "stock_quantity": 50 },
      { "warehouse_name": "Warehouse C", "stock_quantity": 50 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58c5"),
    "item_name": "Table",
    "tags": ["furniture", "office"],
    "quantity": 300,
    "status": "B",
    "height": 30,
    "warehouses": [
      { "warehouse_name": "Warehouse A", "stock_quantity": 20 },
      { "warehouse_name": "Warehouse B", "stock_quantity": 50 }
    ]
  }
]
```

---

3. Queries

a. List all the items where quantity is greater than 300.

```
db.Inventory.find(
  { quantity: { $gt: 300 } },
  { _id: 0, item_name: 1, quantity: 1 }
)
```

---

b. List all items which have tags less than 5.

```
db.Inventory.find(
  { "tags.4": { $exists: false } },
  { _id: 0, item_name: 1, tags: 1 }
)
```

---

c. List all items having status equal to "B" or having quantity less than 50 and height of the product should be greater than 8.

```
db.Inventory.find(
  {
```

```
    $or: [
      { status: "B" },
      { $and: [ { quantity: { $lt: 50 } }, { height: { $gt: 8 } } ] }
    ]
  },
  { _id: 0, item_name: 1, status: 1, quantity: 1, height: 1 }
)
```

---

d. Find all warehouses that keep item “Planner” and have stock quantity less than 20.

```
db.Inventory.find(
  {
    item_name: "Planner",
    "warehouses.stock_quantity": { $lt: 20 }
  },
  { _id: 0, "warehouses.$": 1 }
)
```

==========

SLIP 9:

Here are ten sample documents for the CustomerLoans collection.

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d1"),
    "customer_name": "David Smith",
    "loan_type": "Home",
    "loan_amount": 250000,
    "city": "Pimpri",
    "address": "123 Main St, Pimpri"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d2"),
    "customer_name": "Daniel Brown",
    "loan_type": "Car",
    "loan_amount": 30000,
    "city": "Pimpri",
    "address": "456 Elm St, Pimpri"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d3"),
    "customer_name": "Sarah Johnson",
    "loan_type": "Personal",
    "loan_amount": 50000,
    "city": "Pimpri",
    "address": "789 Oak St, Pimpri"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d4"),
    "customer_name": "Michael Williams",
    "loan_type": "Home",
    "loan_amount": 150000,
    "city": "Pune",
    "address": "101 Pine St, Pune"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d5"),
    "customer_name": "Diana Wilson",
    "loan_type": "Car",
    "loan_amount": 20000,
    "city": "Pimpri",
    "address": "202 Maple St, Pimpri"
  },
```

```
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d6"),
    "customer_name": "David Lee",
    "loan_type": "Personal",
    "loan_amount": 75000,
    "city": "Pune",
    "address": "303 Birch St, Pune"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d7"),
    "customer_name": "Emma Davis",
    "loan_type": "Home",
    "loan_amount": 200000,
    "city": "Pimpri",
    "address": "404 Cedar St, Pimpri"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d8"),
    "customer_name": "Daniel Green",
    "loan_type": "Car",
    "loan_amount": 25000,
    "city": "Pimpri",
    "address": "505 Spruce St, Pimpri"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58d9"),
    "customer_name": "Derek White",
    "loan_type": "Personal",
    "loan_amount": 60000,
    "city": "Pune",
    "address": "606 Fir St, Pune"
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58da"),
    "customer_name": "Diana Brown",
    "loan_type": "Home",
    "loan_amount": 180000,
    "city": "Pimpri",
    "address": "707 Willow St, Pimpri"
  }
]
```

---

3. Queries

a. List all customers whose name starts with the character 'D'.

```
db.CustomerLoans.find(
  { customer_name: { $regex: /^D/ } },
  { _id: 0, customer_name: 1 }
)
```

---

b. List the names of customers in descending order who have taken a loan from Pimpri city.

```
db.CustomerLoans.find(
  { city: "Pimpri" },
  { _id: 0, customer_name: 1 }
).sort({ customer_name: -1 })
```

---

c. Display customer details having the maximum loan amount.

```
db.CustomerLoans.find()
  .sort({ loan_amount: -1 })
```

```
  .limit(1)
```

---

d. Update the address of the customer whose name is “Mr. Patil” and loan_amount is greater than 100000.

```
db.CustomerLoans.updateOne(
  { customer_name: "Mr. Patil", loan_amount: { $gt: 100000 } },
  { $set: { address: "New Address, City" } }
)
```

==========

SLIP10


```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e1"),
    "product_name": "Smartphone X",
    "brand_name": "TechBrand",
    "warranty_period": 1,
    "rating": 4.5,
    "customer_name": "Alice Johnson",
    "purchase_date": ISODate("2023-08-15T00:00:00Z"),
    "city": "Nashik",
    "bill_amount": 70000
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e2"),
    "product_name": "Laptop Pro",
    "brand_name": "CompuTech",
    "warranty_period": 2,
    "rating": 4.7,
    "customer_name": "Bob Smith",
    "purchase_date": ISODate("2023-08-16T00:00:00Z"),
    "city": "Nashik",
    "bill_amount": 120000
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e3"),
    "product_name": "Bluetooth Headphones",
    "brand_name": "SoundWave",
    "warranty_period": 1,
    "rating": 4.2,
    "customer_name": "Charlie Brown",
    "purchase_date": ISODate("2023-08-15T00:00:00Z"),
    "city": "Pune",
    "bill_amount": 5000
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e4"),
    "product_name": "Smartwatch 3",
    "brand_name": "WearTech",
    "warranty_period": 1,
    "rating": 4.8,
    "customer_name": "David Wilson",
    "purchase_date": ISODate("2023-08-17T00:00:00Z"),
    "city": "Nashik",
    "bill_amount": 15000
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e5"),
    "product_name": "Tablet Mini",
    "brand_name": "Gizmos",
    "warranty_period": 1,
```

```
    "rating": 4.0,
    "customer_name": "Emma Davis",
    "purchase_date": ISODate("2023-08-18T00:00:00Z"),
    "city": "Mumbai",
    "bill_amount": 25000
  }
]
```

---

3. Queries

a. List the names of products whose warranty period is one year.

```
db.OnlineShopping.find(
  { warranty_period: 1 },
  { _id: 0, product_name: 1 }
)
```

---

b. List the customers who have made a purchase on “15/08/2023”.

```
db.OnlineShopping.find(
  { purchase_date: ISODate("2023-08-15T00:00:00Z") },
  { _id: 0, customer_name: 1 }
)
```

---

c. Display the names of products with the brand which has the highest rating.

```
db.OnlineShopping.aggregate([
  {
    $group: {
      _id: "$brand_name",
      max_rating: { $max: "$rating" }
    }
  },
  {
    $lookup: {
      from: "OnlineShopping",
      localField: "max_rating",
      foreignField: "rating",
      as: "product_details"
    }
  },
  {
    $unwind: "$product_details"
  },
  {
    $project: {
      _id: 0,
      product_name: "$product_details.product_name",
      brand_name: "$product_details.brand_name",
      rating: "$product_details.rating"
    }
  }
])
```

---

d. Display customers who stay in a specific city (e.g., "Nashik") and have a bill amount greater than 50000.

```
db.OnlineShopping.find(
  { city: "Nashik", bill_amount: { $gt: 50000 } },
  { _id: 0, customer_name: 1 }
)
```

==========

SLIP 11

Here are five sample documents for the Sales collection.

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e1"),
    "product_name": "Laptop",
    "product_id": "P001",
    "customer_name": "Mr. Rajiv",
    "order_id": "O001",
    "order_value": 25000,
    "invoice_id": "I001",
    "invoice_date": ISODate("2023-08-15T00:00:00Z")
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e2"),
    "product_name": "Smartphone",
    "product_id": "P002",
    "customer_name": "Ms. Priya",
    "order_id": "O002",
    "order_value": 15000,
    "invoice_id": null,
    "invoice_date": null
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e3"),
    "product_name": "Headphones",
    "product_id": "P003",
    "customer_name": "Mr. Rajiv",
    "order_id": "O003",
    "order_value": 5000,
    "invoice_id": "I002",
    "invoice_date": ISODate("2023-08-16T00:00:00Z")
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e4"),
    "product_name": "Tablet",
    "product_id": "P004",
    "customer_name": "Mrs. Anjali",
    "order_id": "O004",
    "order_value": 20000,
    "invoice_id": null,
    "invoice_date": null
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e5"),
    "product_name": "Smartwatch",
    "product_id": "P005",
    "customer_name": "Mr. Rajiv",
    "order_id": "O005",
    "order_value": 30000,
    "invoice_id": "I003",
    "invoice_date": ISODate("2023-08-17T00:00:00Z")
  }
]
```

---

3. Queries

a. List all products in the inventory.

```
db.Sales.find(
  {},
  { _id: 0, product_name: 1, product_id: 1 }
```

```
)
```
---

b. List the details of orders with a value > 20000.

```
db.Sales.find(
  { order_value: { $gt: 20000 } },
  { _id: 0, order_id: 1, order_value: 1, customer_name: 1 }
)
```
---

c. List all the orders which have not been processed (invoice not generated).

```
db.Sales.find(
  { invoice_id: null },
  { _id: 0, order_id: 1, customer_name: 1, order_value: 1 }
)
```
---

d. List all the orders along with their invoice for “Mr. Rajiv”.

```
db.Sales.find(
  { customer_name: "Mr. Rajiv" },
  { _id: 0, order_id: 1, product_name: 1, order_value: 1, invoice_id: 1, invoice_date: 1 }
)
```

==========

SLIP 12

Here are five sample documents for the Movies collection:

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e1"),
    "movie_name": "Action Blast",
    "budget": 50000000,
    "actors": [
      { "actor_name": "Akshay Kumar", "role": "Hero" },
      { "actor_name": "John Doe", "role": "Villain" }
    ],
    "producers": ["Producer A", "Producer B"],
    "release_year": 2023
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e2"),
    "movie_name": "Romantic Journey",
    "budget": 20000000,
    "actors": [
      { "actor_name": "Priyanka Chopra", "role": "Heroine" },
      { "actor_name": "Akshay Kumar", "role": "Lead" }
    ],
    "producers": ["Producer B"],
    "release_year": 2022
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e3"),
    "movie_name": "Thriller Night",
    "budget": 30000000,
    "actors": [
      { "actor_name": "Akshay Kumar", "role": "Hero" },
```

```
    { "actor_name": "Emma Watson", "role": "Lead" }
   ],
   "producers": ["Producer C", "Producer D"],
   "release_year": 2023
  },
  {
   "_id": ObjectId("64c9a1b1f6291e7d5c7a58e4"),
   "movie_name": "Drama Queen",
   "budget": 15000000,
   "actors": [
    { "actor_name": "Emma Watson", "role": "Heroine" },
    { "actor_name": "John Doe", "role": "Supporting" }
   ],
   "producers": ["Producer A", "Producer B"],
   "release_year": 2022
  },
  {
   "_id": ObjectId("64c9a1b1f6291e7d5c7a58e5"),
   "movie_name": "Action Reloaded",
   "budget": 60000000,
   "actors": [
    { "actor_name": "John Doe", "role": "Hero" },
    { "actor_name": "Akshay Kumar", "role": "Lead" }
   ],
   "producers": ["Producer E"],
   "release_year": 2023
  }
]
```

---

3. Queries

a. List the names of movies with the highest budget.

```
db.Movies.find(
  {},
  { _id: 0, movie_name: 1, budget: 1 }
).sort({ budget: -1 }).limit(1)
```

---

b. Display the details of producers who have produced more than one movie in a year.

```
db.Movies.aggregate([
  { $unwind: "$producers" },
  { $group: {
     _id: { year: "$release_year", producer: "$producers" },
     movie_count: { $sum: 1 }
   }
  },
  { $match: { movie_count: { $gt: 1 } } },
  { $group: {
     _id: "$_id.producer",
     years: { $addToSet: "$_id.year" }
   }
```

```
  },
  { $project: { _id: 0, producer: "$_id", years: 1 } }
])
```

---

c. List the names of actors who have acted in at least one movie in which ‘Akshay Kumar’ has acted.

```
db.Movies.aggregate([
  { $match: { "actors.actor_name": "Akshay Kumar" } },
  { $unwind: "$actors" },
  { $group: {
     _id: "$actors.actor_name"
    }
  },
  { $project: { _id: 0, actor_name: "$_id" } }
])
```

---

d. List the names of movies produced by more than one producer.

```
db.Movies.find(
  { "producers.1": { $exists: true } },
  { _id: 0, movie_name: 1 }
)
```

==========

SLIP 13

Here are ten sample documents for the Competitions collection:

```
[
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e1"),
    "competition_name": "Programming Contest",
    "students": [
      { "student_name": "Alice", "class": "FY", "position": 1 },
      { "student_name": "Bob", "class": "SY", "position": 2 },
      { "student_name": "Charlie", "class": "TY", "position": 3 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e2"),
    "competition_name": "E-Rangoli",
    "students": [
      { "student_name": "Daisy", "class": "FY", "position": 1 },
      { "student_name": "Eva", "class": "SY", "position": 2 },
      { "student_name": "Fay", "class": "TY", "position": 3 },
      { "student_name": "Gina", "class": "FY", "position": 4 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58e3"),
    "competition_name": "Science Quiz",
    "students": [
```

```
    { "student_name": "Hank", "class": "FY", "position": 1 },
    { "student_name": "Ivy", "class": "SY", "position": 2 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e4"),
  "competition_name": "Math Olympiad",
  "students": [
    { "student_name": "Jack", "class": "TY", "position": 1 },
    { "student_name": "Karen", "class": "FY", "position": 2 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e5"),
  "competition_name": "Art Exhibition",
  "students": [
    { "student_name": "Leo", "class": "FY", "position": 1 },
    { "student_name": "Mona", "class": "SY", "position": 2 },
    { "student_name": "Nina", "class": "TY", "position": 3 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e6"),
  "competition_name": "Dance Battle",
  "students": [
    { "student_name": "Oscar", "class": "FY", "position": 1 },
    { "student_name": "Paul", "class": "SY", "position": 2 },
    { "student_name": "Quinn", "class": "TY", "position": 3 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e7"),
  "competition_name": "Debate",
  "students": [
    { "student_name": "Rachel", "class": "FY", "position": 1 },
    { "student_name": "Sam", "class": "SY", "position": 2 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e8"),
  "competition_name": "Spelling Bee",
  "students": [
    { "student_name": "Tina", "class": "FY", "position": 1 },
    { "student_name": "Uma", "class": "SY", "position": 2 },
    { "student_name": "Vera", "class": "TY", "position": 3 }
  ]
},
{
  "_id": ObjectId("64c9a1b1f6291e7d5c7a58e9"),
  "competition_name": "History Quiz",
```

```
    "students": [
      { "student_name": "Walt", "class": "FY", "position": 1 },
      { "student_name": "Xena", "class": "SY", "position": 2 }
    ]
  },
  {
    "_id": ObjectId("64c9a1b1f6291e7d5c7a58ea"),
    "competition_name": "Science Fair",
    "students": [
      { "student_name": "Yara", "class": "TY", "position": 1 },
      { "student_name": "Zane", "class": "FY", "position": 2 }
    ]
  }
]
```

---

3. Queries

a. Display the average number of students participating in each competition.

```
db.Competitions.aggregate([
  { $project: { competition_name: 1, student_count: { $size: "$students" } } },
  { $group: {
      _id: null,
      average_students: { $avg: "$student_count" }
    }
  }
])
```

---

b. Find the number of students for the "Programming Contest".

```
db.Competitions.find(
  { competition_name: "Programming Contest" },
  { _id: 0, student_count: { $size: "$students" } }
)
```

---

c. Display the names of the first three winners of each competition.

```
db.Competitions.find(
  {},
  { _id: 0, competition_name: 1, "students": { $slice: 3 } }
)
```

---

d. Display students from class 'FY' who participated in the 'E-Rangoli' competition.

```
db.Competitions.find(
  { competition_name: "E-Rangoli", "students.class": "FY" },
  { _id: 0, "students.$": 1 }
)
```

==========