### Problem Statement: Create Collection `Social\_Media` and Perform Queries

We will create the `Social\_Media` collection in MongoDB with the fields `User\_Id`, `User\_Name`, `No\_of\_Posts`, `No\_of\_Friends`, `Friends\_List`, and `Interests`. After inserting sample data, we will execute the queries to retrieve specific information as described in the problem.

---

### Step 1: Create Collection `Social\_Media` and Insert Documents

First, create the collection and insert 20 sample documents. Here's how you can do it:

```javascript

use social\_media\_db // Switch to the desired database

db.Social\_Media.insertMany([

{ User\_Id: 1, User\_Name: "Alice", No\_of\_Posts: 150, No\_of\_Friends: 120, Friends\_List: ["Bob", "Charlie", "David"], Interests: ["Music", "Traveling"] },

{ User\_Id: 2, User\_Name: "Bob", No\_of\_Posts: 80, No\_of\_Friends: 90, Friends\_List: ["Alice", "Charlie"], Interests: ["Gaming", "Reading"] },

{ User\_Id: 3, User\_Name: "Charlie", No\_of\_Posts: 200, No\_of\_Friends: 150, Friends\_List: ["Alice", "Bob", "David"], Interests: ["Cooking", "Gaming"] },

{ User\_Id: 4, User\_Name: "David", No\_of\_Posts: 50, No\_of\_Friends: 30, Friends\_List: ["Alice", "Charlie"], Interests: ["Traveling", "Sports"] },

{ User\_Id: 5, User\_Name: "Eve", No\_of\_Posts: 120, No\_of\_Friends: 100, Friends\_List: ["Frank", "Grace"], Interests: ["Movies", "Photography"] },

{ User\_Id: 6, User\_Name: "Frank", No\_of\_Posts: 30, No\_of\_Friends: 50, Friends\_List: ["Eve", "Grace"], Interests: ["Reading", "Traveling"] },

{ User\_Id: 7, User\_Name: "Grace", No\_of\_Posts: 110, No\_of\_Friends: 110, Friends\_List: ["Eve", "Frank"], Interests: ["Music", "Cooking"] },

{ User\_Id: 8, User\_Name: "Hannah", No\_of\_Posts: 250, No\_of\_Friends: 200, Friends\_List: ["Ivy", "Jack"], Interests: ["Photography", "Reading"] },

{ User\_Id: 9, User\_Name: "Ivy", No\_of\_Posts: 70, No\_of\_Friends: 80, Friends\_List: ["Hannah", "Jack"], Interests: ["Music", "Sports"] },

{ User\_Id: 10, User\_Name: "Jack", No\_of\_Posts: 180, No\_of\_Friends: 150, Friends\_List: ["Hannah", "Ivy"], Interests: ["Traveling", "Gaming"] },

{ User\_Id: 11, User\_Name: "Ken", No\_of\_Posts: 160, No\_of\_Friends: 90, Friends\_List: ["Liam", "Mia"], Interests: ["Music", "Technology"] },

{ User\_Id: 12, User\_Name: "Liam", No\_of\_Posts: 100, No\_of\_Friends: 70, Friends\_List: ["Ken", "Mia"], Interests: ["Gaming", "Sports"] },

{ User\_Id: 13, User\_Name: "Mia", No\_of\_Posts: 200, No\_of\_Friends: 180, Friends\_List: ["Ken", "Liam"], Interests: ["Movies", "Reading"] },

{ User\_Id: 14, User\_Name: "Nina", No\_of\_Posts: 50, No\_of\_Friends: 40, Friends\_List: ["Oscar", "Paul"], Interests: ["Cooking", "Technology"] },

{ User\_Id: 15, User\_Name: "Oscar", No\_of\_Posts: 30, No\_of\_Friends: 20, Friends\_List: ["Nina", "Paul"], Interests: ["Photography", "Music"] },

{ User\_Id: 16, User\_Name: "Paul", No\_of\_Posts: 40, No\_of\_Friends: 50, Friends\_List: ["Nina", "Oscar"], Interests: ["Reading", "Gaming"] },

{ User\_Id: 17, User\_Name: "Quincy", No\_of\_Posts: 75, No\_of\_Friends: 60, Friends\_List: ["Rita", "Steve"], Interests: ["Traveling", "Sports"] },

{ User\_Id: 18, User\_Name: "Rita", No\_of\_Posts: 90, No\_of\_Friends: 110, Friends\_List: ["Quincy", "Steve"], Interests: ["Music", "Movies"] },

{ User\_Id: 19, User\_Name: "Steve", No\_of\_Posts: 65, No\_of\_Friends: 85, Friends\_List: ["Quincy", "Rita"], Interests: ["Cooking", "Gaming"] },

{ User\_Id: 20, User\_Name: "Tom", No\_of\_Posts: 130, No\_of\_Friends: 160, Friends\_List: ["Uma", "Victor"], Interests: ["Photography", "Technology"] }

]);

```

---

### Step 2: Queries

#### 1. List All Users from the `Social\_Media` Collection in Formatted Manner

We can use the `find()` method with the `pretty()` function to format the output in a readable way.

```javascript

db.Social\_Media.find().pretty();

```

- \*\*Explanation\*\*: This will display all the documents in the `Social\_Media` collection in a formatted manner, making it easier to read.

#### 2. Find All Users Having Number of Posts Greater Than 100

We will use the `$gt` (greater than) operator to filter users who have posted more than 100 posts.

```javascript

db.Social\_Media.find({ No\_of\_Posts: { $gt: 100 } }).pretty();

```

- \*\*Explanation\*\*: This query will find all users where the `No\_of\_Posts` is greater than 100.

#### 3. List the User Names and Their Respective `Friends\_List`

We want to display the `User\_Name` and their `Friends\_List` for each user.

```javascript

db.Social\_Media.find({}, { User\_Name: 1, Friends\_List: 1, \_id: 0 }).pretty();

```

- \*\*Explanation\*\*: This query retrieves only the `User\_Name` and `Friends\_List` fields from all documents in the `Social\_Media` collection. `\_id: 0` excludes the default `\_id` field from the output.

#### 4. Display the User IDs and `Friends\_List` of Users Who Have More Than 5 Friends

We will use the `$gt` operator to filter users who have more than 5 friends.

```javascript

db.Social\_Media.find({ No\_of\_Friends: { $gt: 5 } }, { User\_Id: 1, Friends\_List: 1, \_id: 0 }).pretty();

```

- \*\*Explanation\*\*: This query will display the `User\_Id` and `Friends\_List` for users who have more than 5 friends.

#### 5. Display All Users with `No\_of\_Posts` in Descending Order

We will use the `sort()` method to sort the users by `No\_of\_Posts` in descending order.

```javascript

db.Social\_Media.find().sort({ No\_of\_Posts: -1 }).pretty();

```

- \*\*Explanation\*\*: The `sort({ No\_of\_Posts: -1 })` sorts the users by `No\_of\_Posts` in descending order. The `-1` indicates descending order.

---

### Summary of Queries

1. \*\*List all users in a formatted manner\*\*: `find().pretty()`.

2. \*\*Find users with more than 100 posts\*\*: `find({ No\_of\_Posts: { $gt: 100 } })`.

3. \*\*List user names and their friends list\*\*: `find({}, { User\_Name: 1, Friends\_List: 1, \_id: 0 })`.

4. \*\*Display user ids and friends list for users with more than 5 friends\*\*: `find({ No\_of\_Friends: { $gt: 5 } }, { User\_Id: 1, Friends\_List: 1, \_id: 0 })`.

5. \*\*Display all users with posts in descending order\*\*: `find().sort({ No\_of\_Posts: -1 })`.

---

### Running the Queries

1. \*\*Open MongoDB Shell\*\* (`mongosh`).

2. \*\*Switch to the desired database\*\*:

```javascript

use social\_media\_db

```

3. \*\*Execute the queries\*\*: Run each of the queries listed above.

By following these steps, you will be able to manage your `Social\_Media` collection and extract the required data efficiently using MongoDB.