

INFOVEST ENTERPRISES

MILESTONE – NoSQL Implementation

Group 7

Hardik Upadhyay

Raghav Chitlangia

(857)6938717

(857)6938716

Upadhyay.har@northeastern.edu

Chitlangia.r@northeastern.edu

Percentage of effort contributed by Student 1: 50%

Percentage of effort contributed by Student 2: 50%

Signature of Student 1: Hardik Upadhyay

Signature of Student 2: Raghav Chitlangia

Submission Date: December 3, 2022

Q1.

```
db.getCollection("Investor").find({}, {"Investor_id": "$Investor_id", "area_of_interest": "$area_of_interest", "Ventures_to_invest": "$Ventures_to_invest", "_id" : NumberInt(0)});
```

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure, including the 'investor' collection. The main panel shows a query in the IntelliShell tab: `db.getCollection("Investor").find({}, {"Investor_id": "$Investor_id", "area_of_interest": "$area_of_interest", "Ventures_to_invest": "$Ventures_to_invest", "_id" : NumberInt(0)});`. The 'Raw Shell Output' tab shows the results of the query, displaying a table with columns: _id, Investor_id, area_of_interest, and Ventures_to_invest. The table contains 16 rows of data. The 'Operations' panel on the left shows the import progress for the 'investor' collection, indicating that 140 documents were imported successfully.

_id	Investor_id	area_of_interest	Ventures_to_invest
638bc992135b350ba4db66fd	1	ROBOTICS	1
638bc992135b350ba4db66fe	2	ROBOTICS	4
638bc992135b350ba4db66ff	3	information tech	4
638bc992135b350ba4db6701	4	information tech	6
638bc992135b350ba4db6702	5	Pharmaceutical	2
638bc992135b350ba4db6703	6	Pharmaceutical	7
638bc992135b350ba4db6704	7	Automobiles	4
638bc992135b350ba4db6705	8	Automobiles	3
638bc992135b350ba4db6706	9	aeroneutics	4
638bc992135b350ba4db6707	10	aeroneutics	3
638bc992135b350ba4db6708	11	Insurance	5
638bc992135b350ba4db6709	12	Insurance	8
638bc992135b350ba4db670a	13	Realstate	4
638bc992135b350ba4db670b	14	Realstate	2
638bc992135b350ba4db670c	15	Financial Techn	2
638bc992135b350ba4db670d	16	Financial Techn	3

Q2.

```
db.getCollection("Expert_info").find({}, {"Expert_name": 1.0}).sort({"area_of_expertise": 1.0});
```

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure, including the 'expert_info' collection. The main panel shows a query in the IntelliShell tab: `db.getCollection("Expert_info").find({}, {"Expert_name": 1.0}).sort({"area_of_expertise": 1.0});`. The 'Raw Shell Output' tab shows the results of the query, displaying a table with columns: _id and Expert_name. The table contains 10 rows of data. The 'Operations' panel on the left shows the import progress for the 'expert_info' collection, indicating that 140 documents were imported successfully.

_id	Expert_name
638bc992135b350ba4db66fd	Connor Morse
638bc992135b350ba4db66fe	Brittanni Bender
638bc992135b350ba4db66ff	Cora Zimmerman
638bc992135b350ba4db6701	Elvis Mueller
638bc992135b350ba4db6702	Maile Le
638bc992135b350ba4db6703	Reagan Gonzalez
638bc992135b350ba4db6704	Phillip Stafford
638bc992135b350ba4db6705	Ulric Kent
638bc992135b350ba4db6706	Shad O'donnell
638bc992135b350ba4db6707	Althea Lloyd

Q3

```
db.getCollection("client").aggregate([{"$group": {"_id": {},"COUNT(*)": {"$sum": NumberInt(1)}}}, {"$project": {"COUNT(*)": "$COUNT(*)","_id": NumberInt(0)}}]);
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Collections' list is expanded, showing 'client'. The main editor displays an aggregate query in the 'IntelliShell' tab:

```
1 aggregate([{"$group": {"_id": {}, "COUNT(*)": {"$sum": NumberInt(1)}}}, {"$project": {"COUNT(*)": "$COUNT(*)", "_id": NumberInt(0)}}]);
```

The 'Raw Shell Output' tab shows the result of the query:

_id	COUNT(*)
0	140

The 'Operations' panel at the bottom shows a successful import of 140 documents from a JSON file.

Q4

```
db.getCollection("Invest_database").find({});
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Collections' list is expanded, showing 'Invest_database'. The main editor displays a find query in the 'IntelliShell' tab:

```
1 db.getCollection("Invest_database").find({});
```

The 'Raw Shell Output' tab shows the result of the query, displaying a list of documents from the 'Invest_database' collection. The first document is:

```
{ "_id": "638bc58d135b...", "matching_ID": "1", "client_id": "1", "investor_id": "1", "category_number": "C1", "connection_procedure": "rectification_procedure", "categorization_procedure": "compliance_status", "Pi": "matchin..." }
```

The 'Operations' panel at the bottom shows a successful import of 140 documents from a JSON file.