NoSQL Assignment 2

Names: Swetha, Yogish, Nikhita

USNs: 1CR19CS114,086,108

SOLUTION

1. Import the required modules:

```
import mysql.connector
from pymongo import MongoClient
```

This imports the `mysql.connector` module for connecting to MySQL and the `MongoClient` class from the `pymongo` module for connecting to MongoDB.

2. Connect to MySQL:

```
   mysql_conn = mysql.connector.connect(
     host="localhost",
     user="root",
     password="Omsairam12#",
     database="mydatabase"
)
```

This creates a connection to a MySQL database called "mydatabase" running on the same machine as the Python script, using the username "username" and password "password".

3. Connect to MongoDB:

```
   # Connect to MongoDB
mongo_client = MongoClient()
mongo_db = mongo_client["mydatabase"]
mongo_coll = mongo_db["mycollection"]
```

This creates a connection to a MongoDB database called "mydatabase" running on the same machine as the Python script, and selects a collection called "mycollection".

4. Query the data from MySQL:

```
   # Query the data from MySQL
mysql_cursor = mysql_conn.cursor()
mysql_cursor.execute("SELECT * FROM student")
data = mysql_cursor.fetchall()
```

This creates a cursor object to interact with the MySQL database, executes a SQL query to select all rows from a table called "mytable", and fetches all the resulting data.

5. Transform the data to MongoDB format:

```
   # Transform the data to MongoDB format
mongo_data = []
for row in data:
    doc = {
        "id": row[0],
        "name": row[1],
        "class": row[2],
        "mark": row[3],
        "gender":row[4]
        # add more fields as needed
    }
    mongo_data.append(doc)
```

This iterates over each row of data, creates a new dictionary object for each row, and assigns each value to a corresponding field in the dictionary. The resulting list of dictionaries is stored in the `mongo_data` variable.

6. Insert the data into MongoDB:

```
   # Insert the data into MongoDB
mongo_coll.insert_many(mongo_data)
```

This inserts the list of dictionaries from `mongo_data` into the MongoDB collection specified by `mongo_coll`.

7. Close the connections:

```
# Close the connections
mysql_cursor.close()
mysql_conn.close()
mongo_client.close()
```

This closes the connections to both the MySQL and MongoDB databases.

Creating the DB in RDMS:

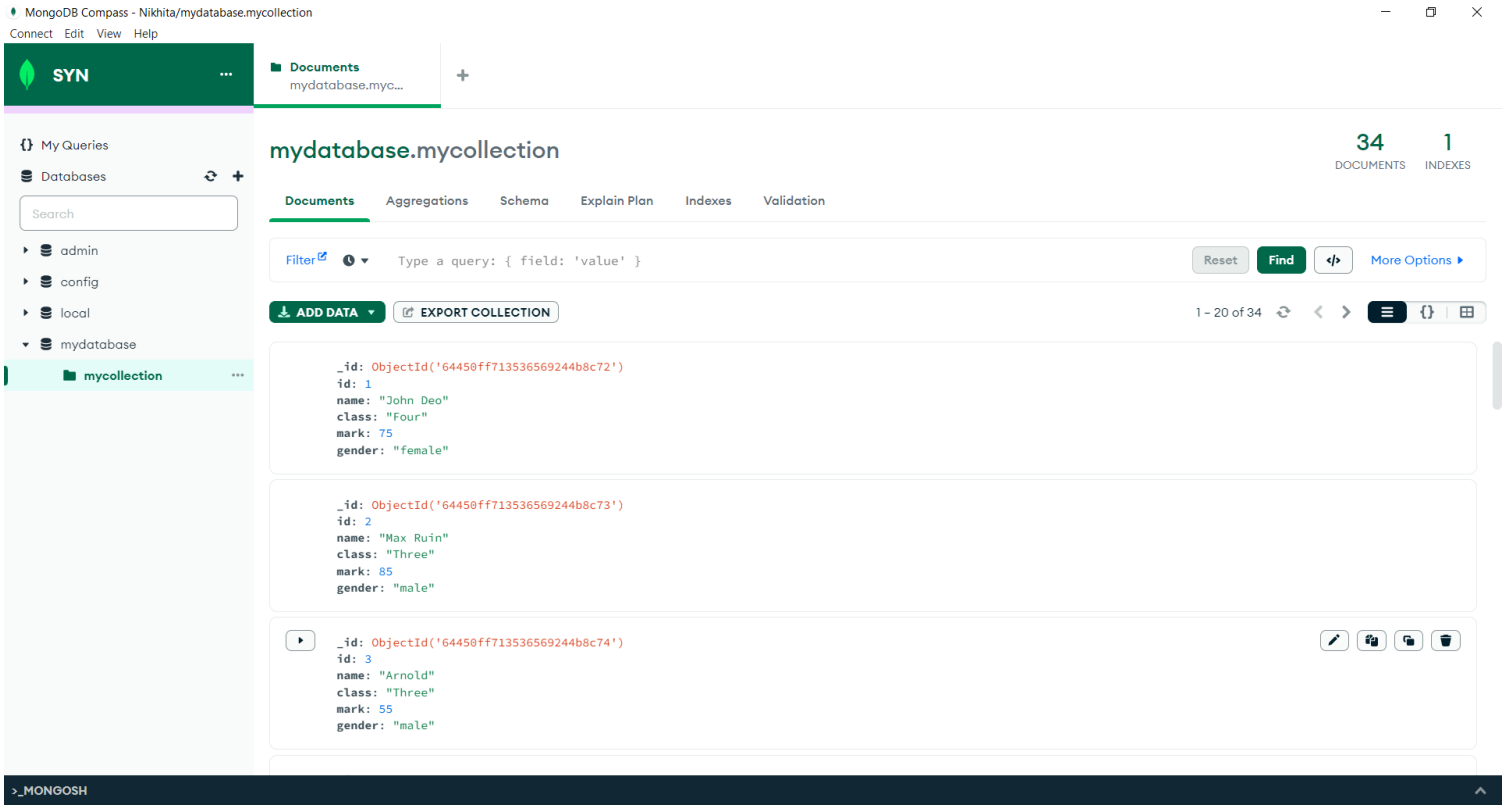
```
1   CREATE DATABASE mydatabase;
2   USE mydatabase;
3   CREATE TABLE IF NOT EXISTS `student` (
4     `id` int(2) NOT NULL AUTO_INCREMENT,
5     `name` varchar(50) CHARACTER SET utf8 NOT NULL DEFAULT '',
6     `class` varchar(10) CHARACTER SET utf8 NOT NULL DEFAULT '',
7     `mark` int(3) NOT NULL DEFAULT '0',
8     `gender` varchar(6) CHARACTER SET utf8 NOT NULL DEFAULT 'male',
9     UNIQUE KEY `id` (`id`)
10  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

```sql
INSERT INTO `student` (`id`, `name`, `class`, `mark`, `gender`) VALUES
(1, 'John Deo', 'Four', 75, 'female'),
(2, 'Max Ruin', 'Three', 85, 'male'),
(3, 'Arnold', 'Three', 55, 'male'),
(4, 'Krish Star', 'Four', 60, 'female'),
(5, 'John Mike', 'Four', 60, 'female'),
(6, 'Alex John', 'Four', 55, 'male'),
(7, 'My John Rob', 'Five', 78, 'male'),
(8, 'Asruid', 'Five', 85, 'male'),
(9, 'Tes Qry', 'Six', 78, 'male'),
(10, 'Big John', 'Four', 55, 'female'),
(11, 'Ronald', 'Six', 89, 'female'),
(12, 'Recky', 'Six', 94, 'female'),
(13, 'Kty', 'Seven', 88, 'female'),
(14, 'Bigy', 'Seven', 88, 'female'),
(15, 'Tade Row', 'Four', 88, 'male'),
(16, 'Gimmy', 'Four', 88, 'male'),
(17, 'Tumyu', 'Six', 54, 'male'),
(18, 'Honny', 'Five', 75, 'male'),
(19, 'Tinny', 'Nine', 18, 'male'),
(20, 'Jackly', 'Nine', 65, 'female'),
(21, 'Babby John', 'Four', 69, 'female'),
(22, 'Reggid', 'Seven', 55, 'female'),
(23, 'Herod', 'Eight', 79, 'male'),
(24, 'Tiddy Now', 'Seven', 78, 'male'),
(25, 'Giff Tow', 'Seven', 88, 'male'),
(26, 'Crelea', 'Seven', 79, 'male'),
(27, 'Big Nose', 'Three', 81, 'female'),
(28, 'Rojj Base', 'Seven', 86, 'female'),
(29, 'Tess Played', 'Seven', 55, 'male'),
(30, 'Reppy Red', 'Six', 79, 'female'),
(31, 'Marry Toeey', 'Four', 88, 'male'),
(32, 'Binn Rott', 'Seven', 90, 'female'),
(33, 'Kenn Rein', 'Six', 96, 'female'),
(34, 'Gain Toe', 'Seven', 69, 'male');
```

```sql
SELECT * FROM mydatabase.student;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	name	class	mark	gender
1	John Deo	Four	75	female
2	Max Ruin	Three	85	male
3	Arnold	Three	55	male
4	Krish Star	Four	60	female
5	John Mike	Four	60	female
6	Alex John	Four	55	male
7	My John Rob	Five	78	male
8	Asruid	Five	85	male
9	Tes Qry	Six	78	male
10	Big John	Four	55	female
11	Ronald	Six	89	female
12	Recky	Six	94	female
13	Kty	Seven	88	female
14	Bigy	Seven	88	female
15	Tade Row	Four	88	male
16	Gimmy	Four	88	male
17	Tumyu	Six	54	male
18	Honny	Five	75	male
19	Tinny	Nine	18	male
20	Jackly	Nine	65	female
21	Babby John	Four	69	female
22	Reggid	Seven	55	female
23	Herod	Eight	79	male
24	Tiddy Now	Seven	78	male
25	Giff Tow	Seven	88	male
26	Crelea	Seven	79	male
27	Big Nose	Three	81	female
28	Rojj Base	Seven	86	female
29	Tess Played	Seven	55	male
30	Reppy Red	Six	79	female
31	Marry Toeey	Four	88	male
32	Binn Rott	Seven	90	female
33	Kenn Rein	Six	96	female
34	Gain Toe	Seven	69	male
NULL	NULL	NULL	NULL	NULL



Advantages of using a NoSQL database like MongoDB over a relational database like MySQL include:

Scalability: NoSQL databases are designed to be horizontally scalable, which means they can handle large amounts of data across multiple servers.

Flexibility: NoSQL databases are schemaless, which means they can handle data with varying structures and can be modified without the need for schema migrations.

Performance: NoSQL databases are optimized for fast reads and writes, making them a good choice for applications that require low latency.

However, there are also some disadvantages to using a NoSQL database:

Lack of transactions: NoSQL databases typically don't support ACID transactions, which can make it more difficult to maintain data consistency in certain situations.

Limited querying capabilities: NoSQL databases often lack the sophisticated querying capabilities of relational databases, which can make it harder to extract insights from the data.

For **future work**, it would be possible to create a simple web interface to interact with the data stored in the MongoDB database. This could be achieved using a web framework like Flask or Django, along with a templating engine like Jinja. Additionally, it would be possible to add more complex querying capabilities to the application by using a search engine like Elasticsearch or a graph database like Neo4j.