

Modern Application Development (Java Spring Boot)

ASSIGNMENT 2

-ABHISHEK S MATTAM

20BCI0162

VIT VELLORE

I have used terminal instead of MYSQL Workbench because of some system errors

1.) CREATE, UPDATE, DELETE commands in MYSQL.

Creating a database:

```
[mysql> CREATE DATABASE vehicles;
Query OK, 1 row affected (0.03 sec)
```

Creating table:

```
[mysql> USE vehicles;
Database changed
[mysql> CREATE TABLE cars(id INT,name VARCHAR(10),brand VARCHAR(10));
Query OK, 0 rows affected (0.04 sec)

[mysql> DESCRIBE cars;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int           | YES  |     | NULL    |       |
| name  | varchar(10)   | YES  |     | NULL    |       |
| brand | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Inserting data into the table:

```
[mysql> INSERT INTO cars(id, name, brand) values (1,'swift','maruthi');
Query OK, 1 row affected (0.01 sec)

[mysql> INSERT INTO cars(id, name, brand) values (2,'verna','hyundai');
Query OK, 1 row affected (0.01 sec)

[mysql> INSERT INTO cars(id, name, brand) values (3,'nexon','tata');
Query OK, 1 row affected (0.00 sec)
```

After insertion:

```
[mysql> SELECT * FROM cars;
+-----+-----+-----+
| id    | name  | brand  |
+-----+-----+-----+
| 1     | swift | maruthi |
| 2     | verna | hyundai |
| 3     | nexon | tata    |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Updating:

```
[mysql> UPDATE cars SET name='venue' WHERE brand='hyundai';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

[mysql> SELECT * FROM cars;
+-----+-----+-----+
| id    | name  | brand  |
+-----+-----+-----+
| 1     | swift | maruthi |
| 2     | venue | hyundai |
| 3     | nexon | tata    |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Deleting:

```
[mysql> DELETE FROM cars WHERE brand='tata';
Query OK, 1 row affected (0.00 sec)

[mysql> SELECT * FROM cars;
+-----+-----+-----+
| id    | name  | brand  |
+-----+-----+-----+
| 1     | swift | maruthi |
| 2     | venue | hyundai |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

2.)CREATE TABLES AND PERFORM JOINS IN MYSQL

Creating database and tables:

```
[mysql> CREATE DATABASE BOOKSHELFF;
Query OK, 1 row affected (0.01 sec)

[mysql> USE BOOKSHELFF;
Database changed
[mysql> CREATE TABLE authors (    author_id INT PRIMARY KEY,        author_name VARCHAR(255),        nationality VARCHAR(255) );
Query OK, 0 rows affected (0.01 sec)

[mysql> show tables;
+-----+
| Tables_in_bookshelff |
+-----+
| authors               |
+-----+
1 row in set (0.00 sec)

[mysql> DESCRIBE authors;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| author_id  | int       | NO   | PRI | NULL    |       |
| author_name | varchar(255) | YES  |     | NULL    |       |
| nationality | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE books (
->     book_id INT PRIMARY KEY,
->     title VARCHAR(255),
->     author_id INT,
->     price DECIMAL(10, 2)
[   -> );
Query OK, 0 rows affected (0.02 sec)

[mysql> DESCRIBE books;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| book_id    | int       | NO   | PRI | NULL    |       |
| title      | varchar(255) | YES  |     | NULL    |       |
| author_id  | int       | YES  |     | NULL    |       |
| price      | decimal(10,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Insertion to both tables:

```

mysql> INSERT INTO books (book_id,title,author_id,price) VALUES (11,'BOOK 1',99,1000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (book_id,title,author_id,price) VALUES (12,'BOOK 2',98,2000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO authors (author_id,author_name,nationality) VALUES (99,'Abhi','India');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO authors (author_id,author_name,nationality) VALUES (98,'Arun','Nepal');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT *FROM books;
+-----+-----+-----+-----+
| book_id | title | author_id | price |
+-----+-----+-----+-----+
|      11 | BOOK 1 |        99 | 1000.00 |
|      12 | BOOK 2 |        98 | 2000.00 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> SELECT *FROM authors;
+-----+-----+-----+
| author_id | author_name | nationality |
+-----+-----+-----+
|        98 | Arun        | Nepal      |
|        99 | Abhi        | India      |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

INNER JOIN:

```

mysql> SELECT books.title, authors.author_name
      -> FROM books
      -> INNER JOIN authors ON books.author_id = authors.author_id;
+-----+-----+
| title | author_name |
+-----+-----+
| BOOK 2 | Arun        |
| BOOK 1 | Abhi        |
+-----+-----+
2 rows in set (0.00 sec)

```

LEFT JOIN:

```

mysql> SELECT books.title, authors.author_name FROM books LEFT JOIN authors ON books.author_id = authors.author_id;
+-----+-----+
| title | author_name |
+-----+-----+
| BOOK 1 | Abhi        |
| BOOK 2 | Arun        |
+-----+-----+
2 rows in set (0.00 sec)

```

RIGHT JOIN:

```

mysql> SELECT books.title, authors.author_name FROM books RIGHT JOIN authors ON books.author_id = authors.author_id;
+-----+-----+
| title | author_name |
+-----+-----+
| BOOK 2 | Arun        |
| BOOK 1 | Abhi        |
+-----+-----+
2 rows in set (0.01 sec)

```

CROSS JOIN:

```
mysql> SELECT books.book_id,authors.author_id, books.title, authors.author_name FROM books CROSS JOIN authors;
```

book_id	author_id	title	author_name
12	98	BOOK 2	Arun
11	98	BOOK 1	Arun
12	99	BOOK 2	Abhi
11	99	BOOK 1	Abhi

```
4 rows in set (0.00 sec)
```

FULL JOIN:

```
mysql> SELECT books.title, authors.author_name FROM books LEFT JOIN authors ON books.author_id = authors.author_id UNION SELECT books.title, authors.author_name FROM books RIGHT JOIN authors ON books.author_id = authors.author_id WHERE books.title IS NULL;
```

title	author_name
BOOK 1	Abhi
BOOK 2	Arun

```
2 rows in set (0.00 sec)
```

3.)CREATE, UPDATE, DELETE COMMANDS IN MONGO

Using MONGODB COMPASS,

Creating collection:

Create Collection

Collection Name

personaldata

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)



Cancel

Create Collection

Inserting data :

Insert Document

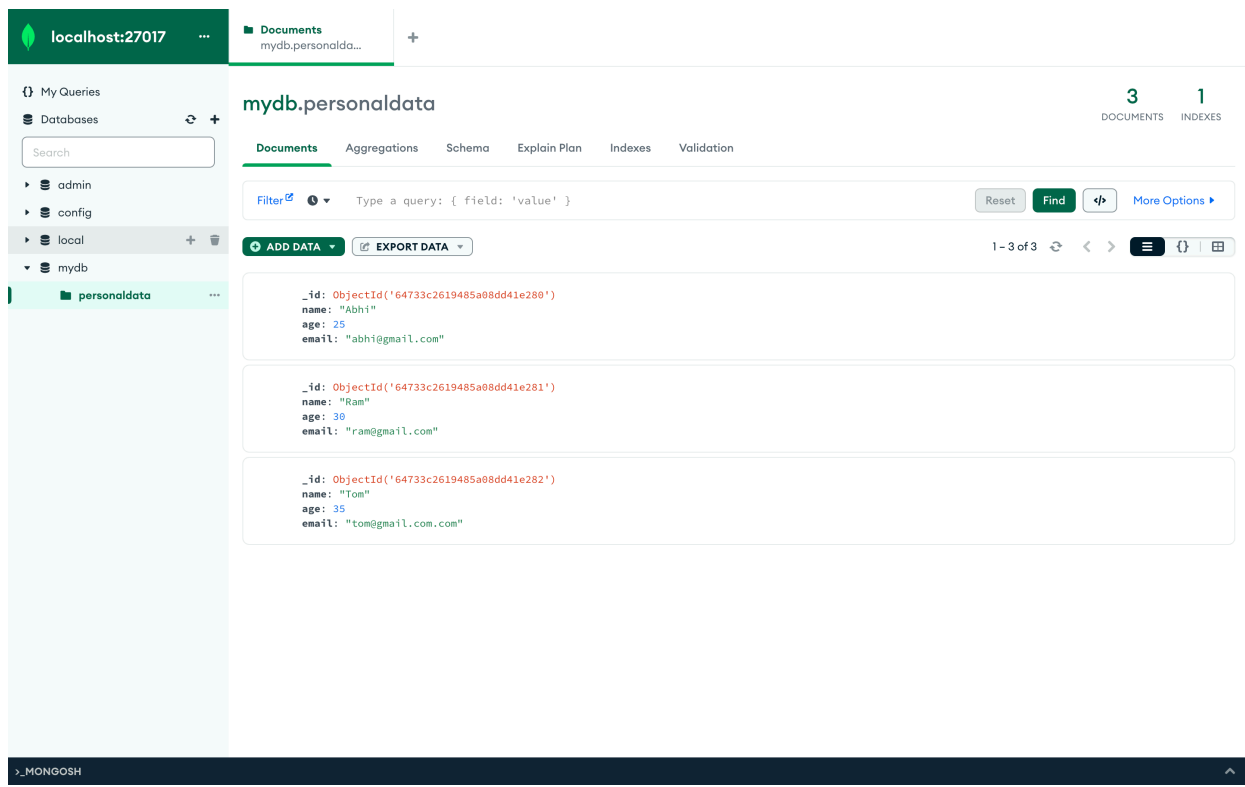
To collection mydb.personaldata

VIEW  

```
1  [
2    { "name": "Abhi", "age": 25, "email": "abhi@gmail.com" },
3    { "name": "Ram", "age": 30, "email": "ram@gmail.com" },
4    { "name": "Tom", "age": 35, "email": "tom@gmail.com" }
5  ]
6
```

Cancel

Insert



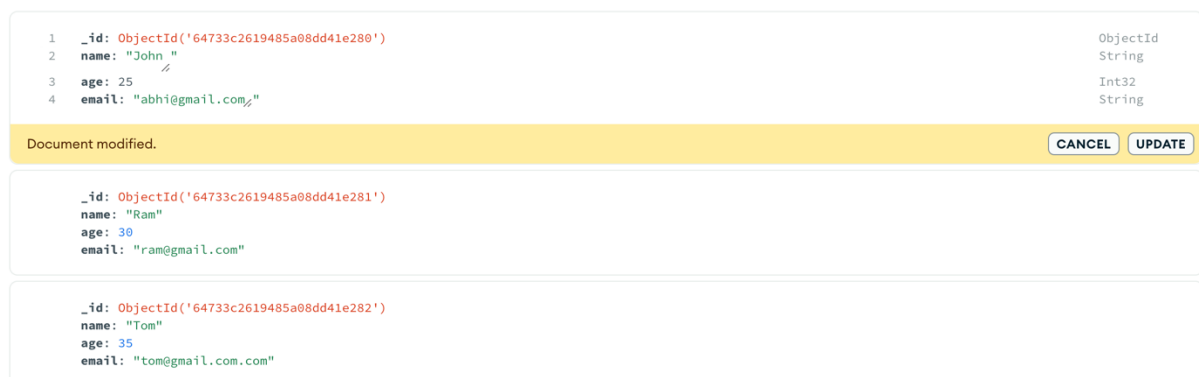
Updating the document:

Locate the document in the collection view.

Click on the "Edit Document" button (pencil icon) next to the document.

Modify the fields you want to update.

Updated the name to “John” from “Abhi”:

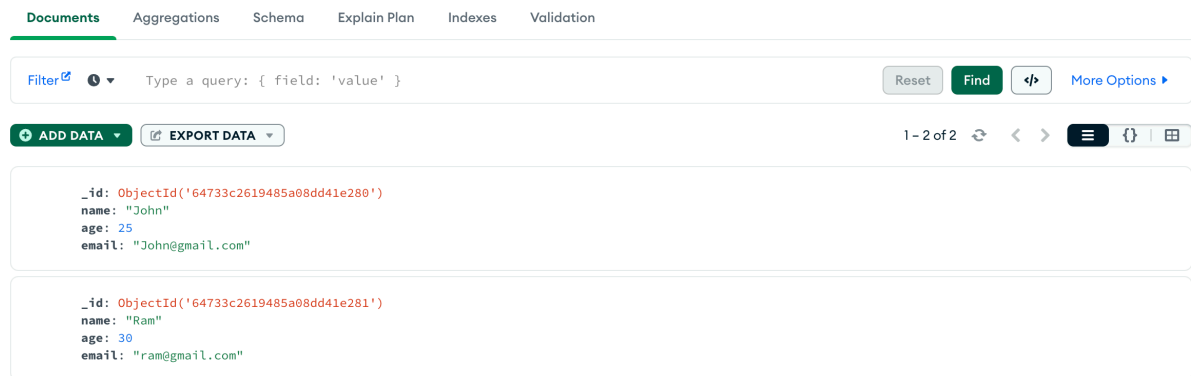


Deleting the document:

Locate the document in the collection view.

Click on the "Delete Document" button (trash bin icon) next to the document.
Confirm the deletion when prompted.

Deleted the last record ("Tom"):



Instead of using MONGODB COMPASS, we can use commands as well:

Creating a document:

```
// Syntax: db.collection.insertOne(document)
db.users.insertOne({ name: "John", age: 25, email: "john@example.com" });
```

Updating a document:

```
// Syntax: db.collection.updateOne(filter, update, options)
db.users.updateOne({ name: "John" }, { $set: { age: 26 } });
```

Deleting a document:

```
// Syntax: db.collection.deleteOne(filter)
db.users.deleteOne({ name: "John" });
```