

Modern Application Development (Java Spring Boot)

Assignment-2

REG.NO: 20BCE2752

NAME: AATHIRAINATHAN P

1) create, update and delete commands in mysql

Create:

Schema:

create schema myschema;

use myschema;

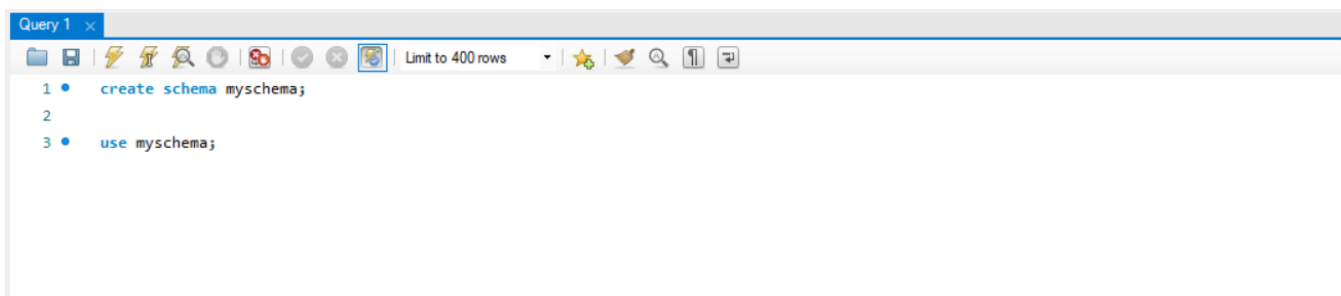
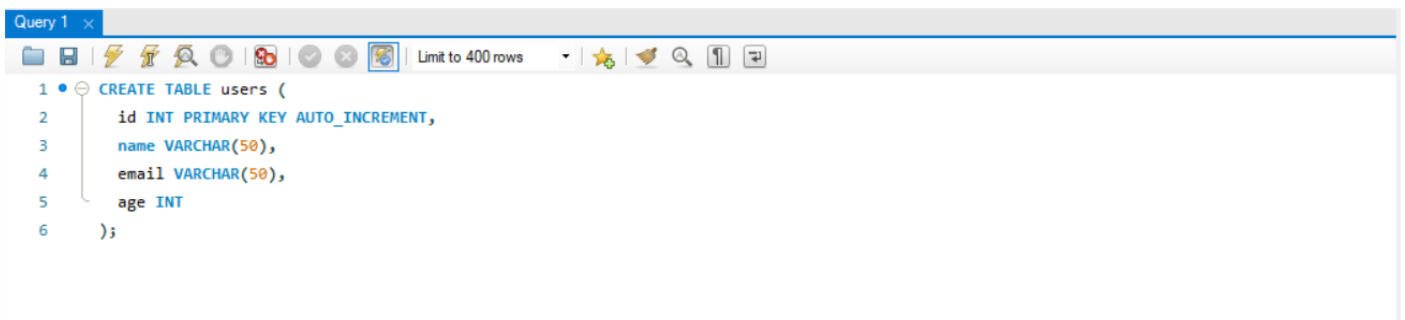


Table:

CREATE TABLE users (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(50),
email VARCHAR(50),
age INT
);



Insert:

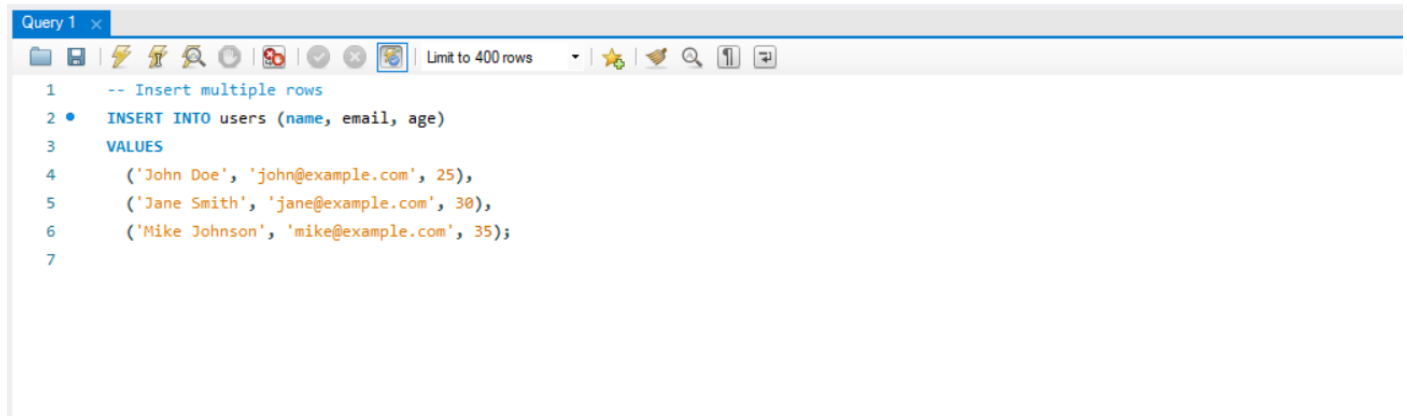
INSERT INTO users (name, email, age)

VALUES

('John Doe', 'john@example.com', 25),

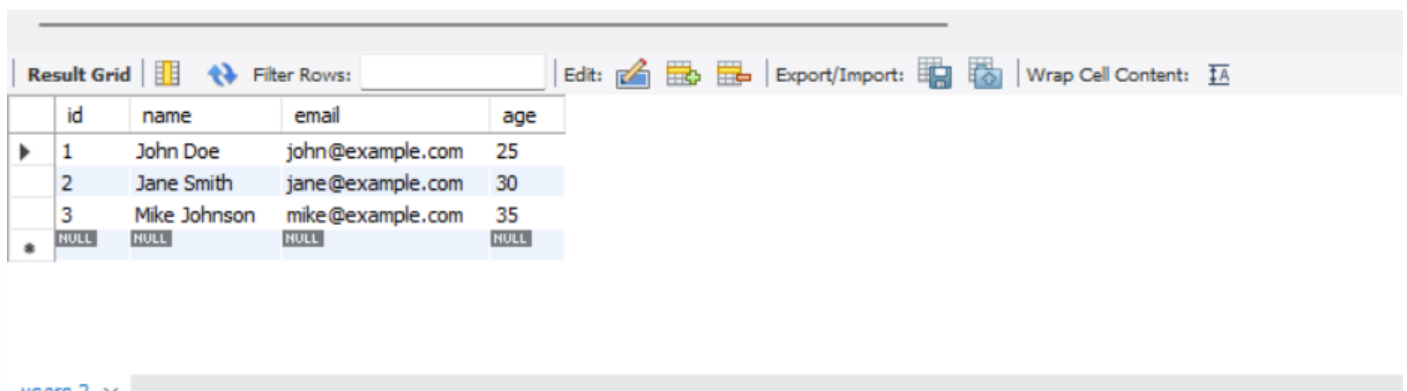
('Jane Smith', 'jane@example.com', 30),

('Mike Johnson', 'mike@example.com', 35);



The screenshot shows a SQL query editor window titled "Query 1". The query is as follows:

```
1  -- Insert multiple rows
2  •  INSERT INTO users (name, email, age)
3     VALUES
4     ('John Doe', 'john@example.com', 25),
5     ('Jane Smith', 'jane@example.com', 30),
6     ('Mike Johnson', 'mike@example.com', 35);
7
```



The screenshot shows a database result grid with the following data:

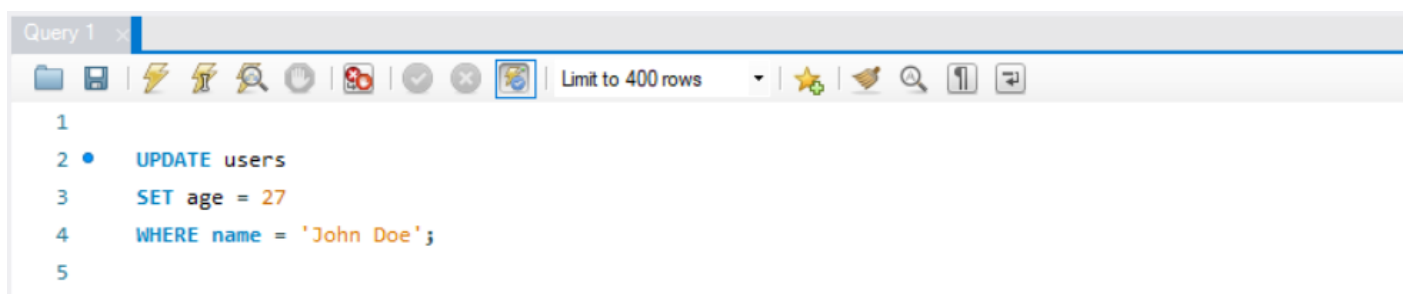
	id	name	email	age
▶	1	John Doe	john@example.com	25
	2	Jane Smith	jane@example.com	30
	3	Mike Johnson	mike@example.com	35
*	NULL	NULL	NULL	NULL

Update:

UPDATE users

SET age = 27

WHERE name = 'John Doe';



The screenshot shows a SQL query editor window titled "Query 1". The query is as follows:

```
1
2  •  UPDATE users
3     SET age = 27
4     WHERE name = 'John Doe';
5
```


Result Grid					Filter Rows:					Edit:					Export/Import:					Wrap Cell Content:				
	id	name	email	age																				
▶	1	John Doe	john@example.com	27																				
	2	Jane Smith	jane@example.com	30																				
	3	Mike Johnson	mike@example.com	35																				
*	NULL	NULL	NULL	NULL																				

Delete:

DELETE FROM users

WHERE age > 30;

Query 1 x



1

2 • DELETE FROM users

3 WHERE age > 30;

4

Result Grid					Filter Rows:					Edit:					Export/Import:					Wrap Cell Content:				
	id	name	email	age																				
▶	1	John Doe	john@example.com	27																				
	2	Jane Smith	jane@example.com	30																				
*	NULL	NULL	NULL	NULL																				

2) Create tables and perform join operation:

Step 1: Create the students table

```
Query 1 x
Limit to 400 rows
1 CREATE TABLE students (
2     student_id INT PRIMARY KEY,
3     name VARCHAR(50),
4     age INT,
5     gender VARCHAR(10)
6 );
```

Step 2: Insert data into the students table

```
Query 1 x
Limit to 400 rows
1 INSERT INTO students (student_id, name, age, gender)
2 VALUES
3     (1, 'John Doe', 20, 'Male'),
4     (2, 'Jane Smith', 22, 'Female'),
5     (3, 'Mike Johnson', 21, 'Male');
```

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

	student_id	name	age	gender
▶	1	John Doe	20	Male
	2	Jane Smith	22	Female
	3	Mike Johnson	21	Male
*	NULL	NULL	NULL	NULL

Step 3: Create the courses table

```
Query 1 x
Limit to 400 rows
1
2
3 CREATE TABLE courses (
4     course_id INT PRIMARY KEY,
5     name VARCHAR(50),
6     instructor VARCHAR(50),
7     credits INT
8 );
```

Step 4: Insert data into the courses table

Query 1

```
1 • INSERT INTO courses (course_id, name, instructor, credits)
2 VALUES
3     (1, 'Mathematics', 'Professor Smith', 4),
4     (2, 'History', 'Professor Johnson', 3),
5     (3, 'English', 'Professor Davis', 3);
```

Result Grid

	course_id	name	instructor	credits
▶	1	Mathematics	Professor Smith	4
	2	History	Professor Johnson	3
	3	English	Professor Davis	3
*	NULL	NULL	NULL	NULL

Step 5: Perform a join operation to retrieve data from both tables

Query 1

```
1 • SELECT students.name, students.age, students.gender, courses.name AS course_name, courses.instructor
2 FROM students
3 JOIN courses ON students.student_id = courses.course_id;
```

Result Grid

	name	age	gender	course_name	instructor
▶	John Doe	20	Male	Mathematics	Professor Smith
	Jane Smith	22	Female	History	Professor Johnson
	Mike Johnson	21	Male	English	Professor Davis

3) Create, insert, update, delete in mongodb

Create:

```
db.createCollection("users");
```

```
>_MONGOSH
> db.createCollection("users");
< { ok: 1 }
test>
```

Insert:

```
db.users.insertMany([
  { name: "Jane Smith", age: 30, email: "jane@example.com" },
  { name: "Mike Johnson", age: 35, email: "mike@example.com" }
]);
```

```
>_MONGOSH
> db.createCollection("users");
< { ok: 1 }
> db.users.insertMany([
  { name: "Jane Smith", age: 30, email: "jane@example.com" },
  { name: "Mike Johnson", age: 35, email: "mike@example.com" }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("647329ec72c3f9e37d4c966b"),
    '1': ObjectId("647329ec72c3f9e37d4c966c")
  }
}
test>
```

Update:

```
db.users.updateOne(  
  { name: "Jane Smith" },  
  { $set: { age: 27, email: "updated@example.com" } }  
);
```

```
> db.users.updateOne(  
  { name: "Jane Smith" },  
  { $set: { age: 27, email: "updated@example.com" } }  
);  
  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
test>
```

```
> db.users.find()  
  
< {  
  _id: ObjectId("647329ec72c3f9e37d4c966b"),  
  name: 'Jane Smith',  
  age: 27,  
  email: 'updated@example.com'  
}  
{  
  _id: ObjectId("647329ec72c3f9e37d4c966c"),  
  name: 'Mike Johnson',  
  age: 35,  
  email: 'mike@example.com'  
}  
test>
```

Delete:

```
db.users.deleteOne({ name: "Jane Smith" });
```

```
> db.users.deleteOne({ name: "Jane Smith" });  
< {  
  acknowledged: true,  
  deletedCount: 1  
}  
test>
```

```
> db.users.find()  
< {  
  _id: ObjectId("647329ec72c3f9e37d4c966c"),  
  name: 'Mike Johnson',  
  age: 35,  
  email: 'mike@example.com'  
}  
test>
```

20BCE2752

AATHIRAINATHAN P