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;

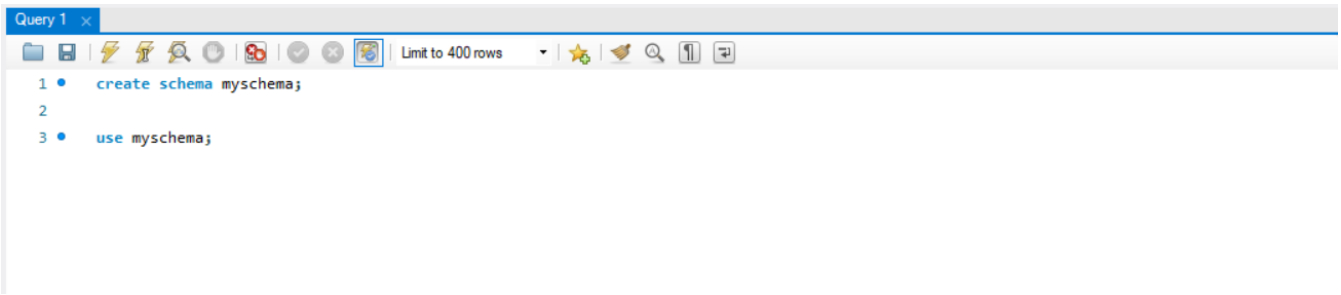
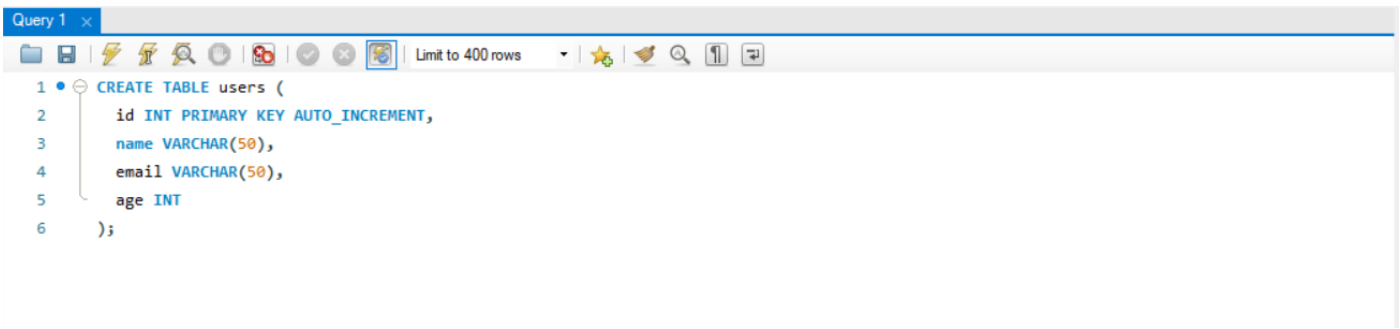


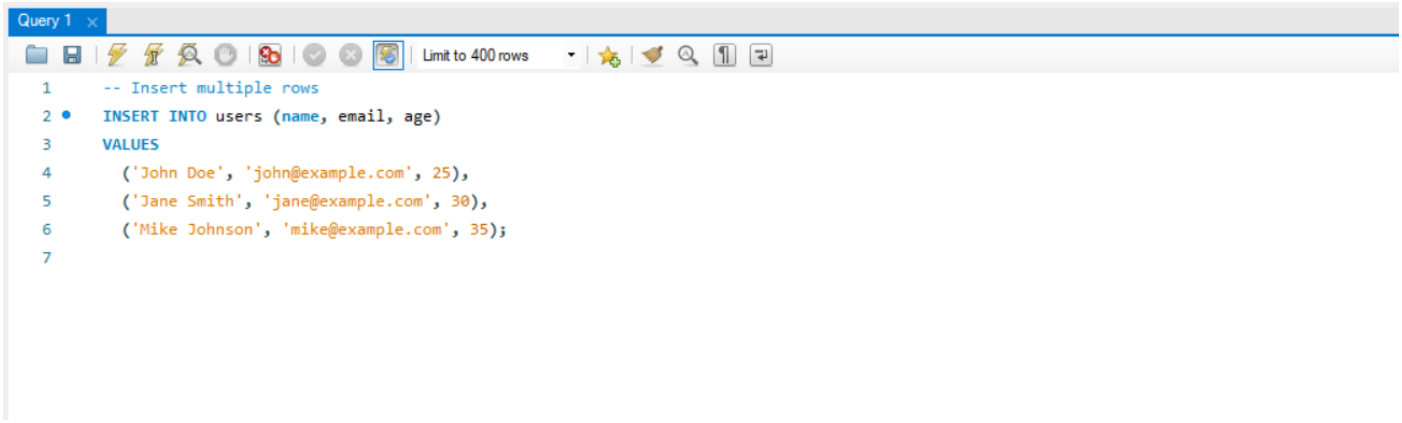
Table:

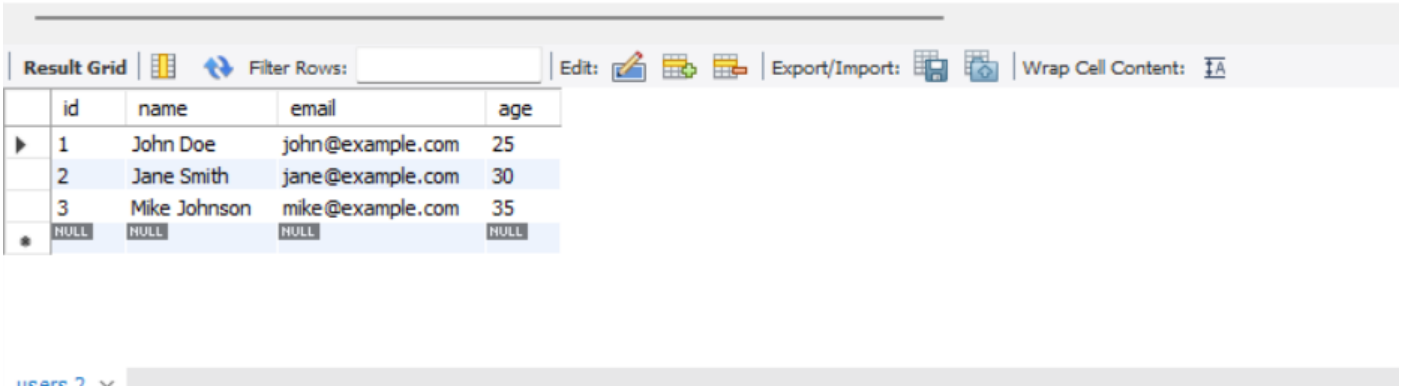
```
CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50),
  email VARCHAR(50),
  age INT
);
```



Insert:

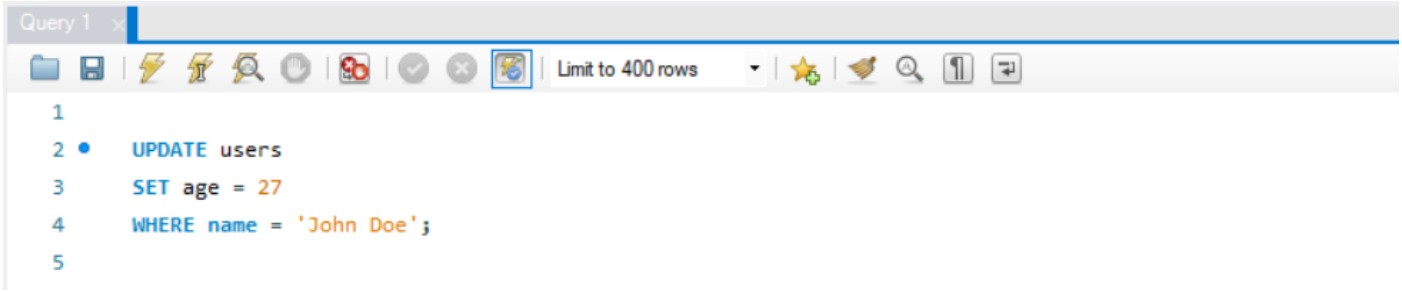
```
INSERT INTO users (name, email, age)
VALUES
  ('John Doe', 'john@example.com', 25),
  ('Jane Smith', 'jane@example.com', 30),
  ('Mike Johnson', 'mike@example.com', 35);
```

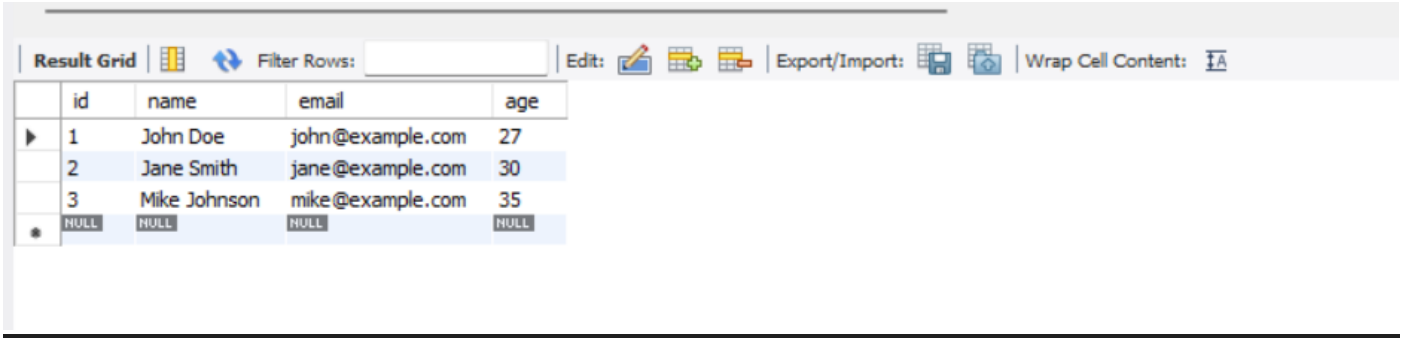




Update:

```
UPDATE users
SET age = 27
WHERE name = 'John Doe';
```

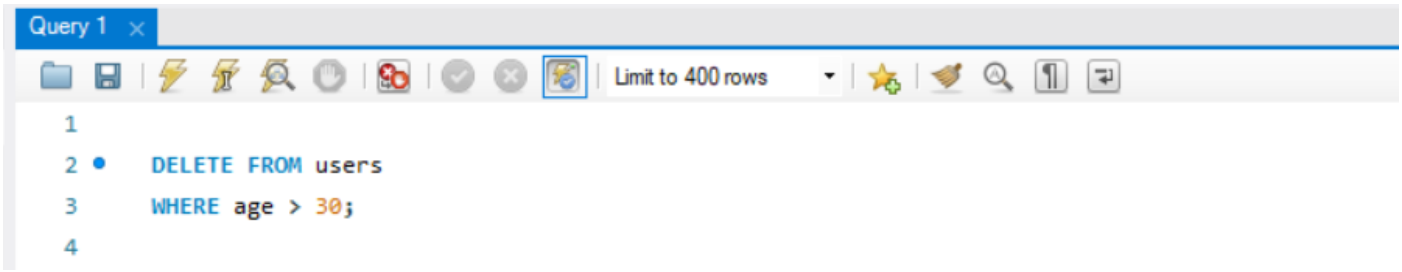


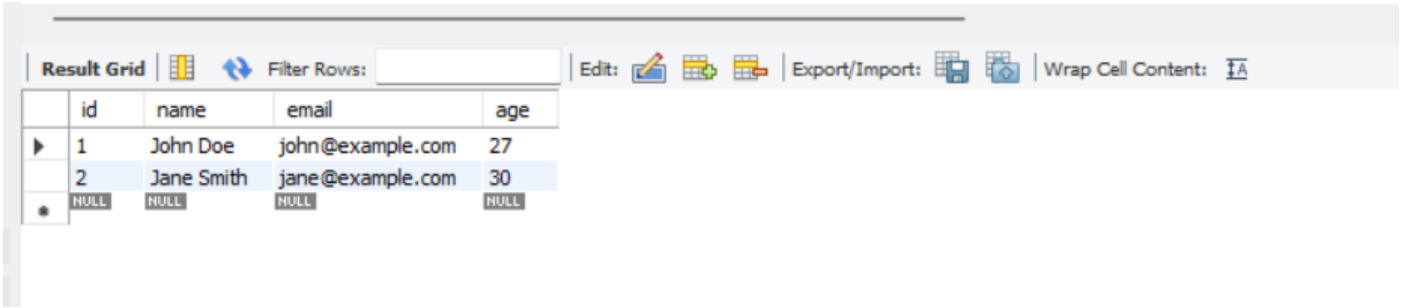


Delete:

DELETE FROM users

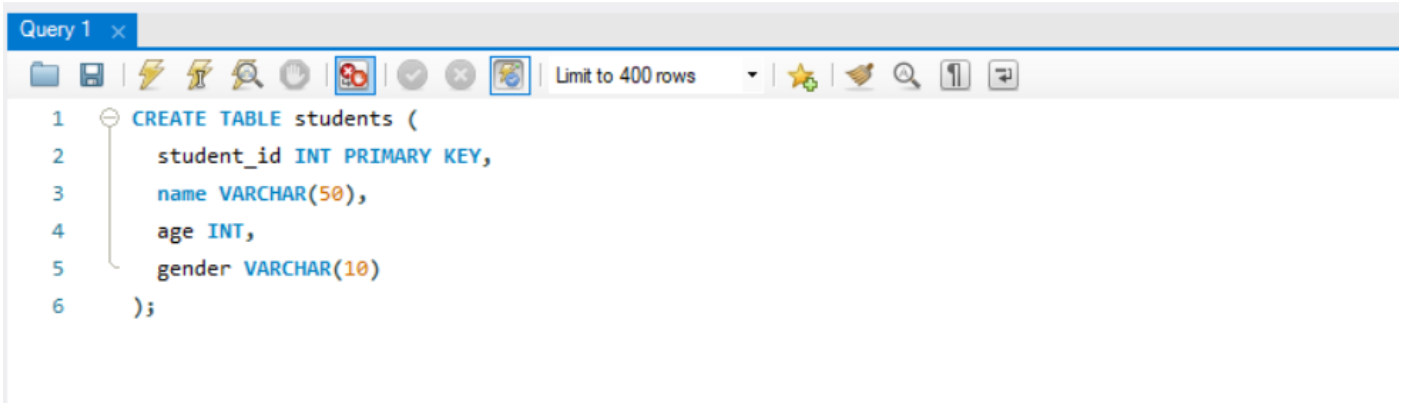
WHERE age > 30;



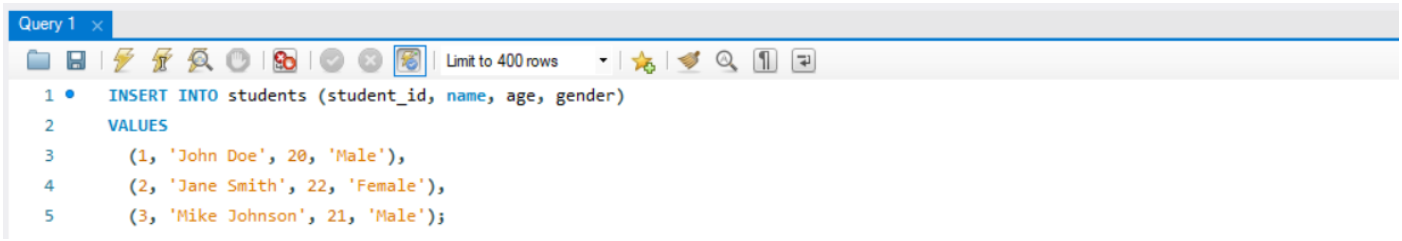


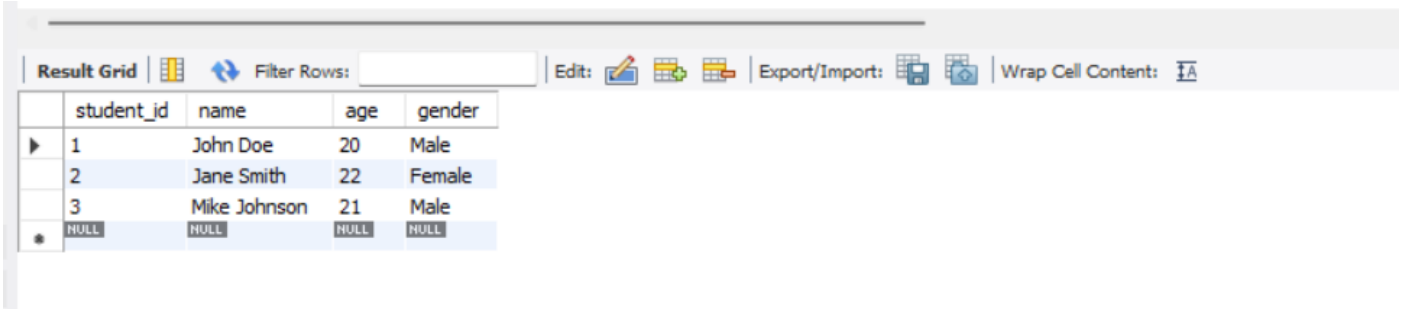
2) Create tables and perform join operation:

Step 1: *Create the students table*

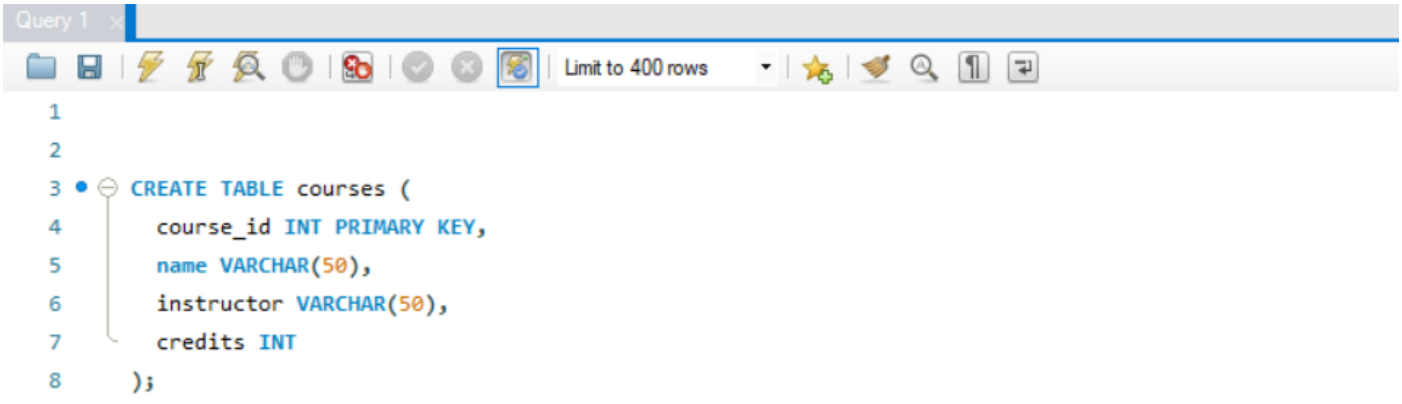


Step 2: *Insert data into the students table*





Step 3: *Create the courses table*



Step 4: *Insert data into the courses table*

Query 1

Limit to 400 rows

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

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

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* ***<u>a join operation</u>*** *to retrieve data from both tables*

Query 1

Limit to 400 rows

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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