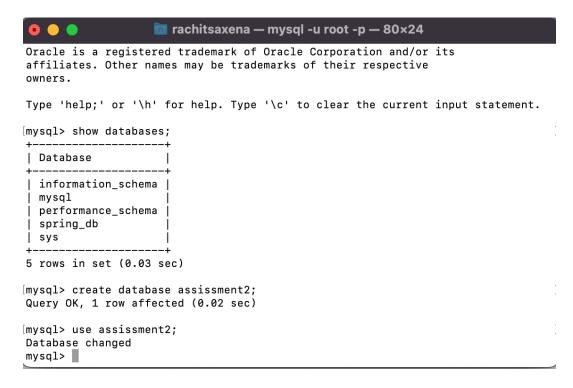
Assessment 2

Name - Rachit Saxena Reg - 20BCE2322 Campus - Vellore

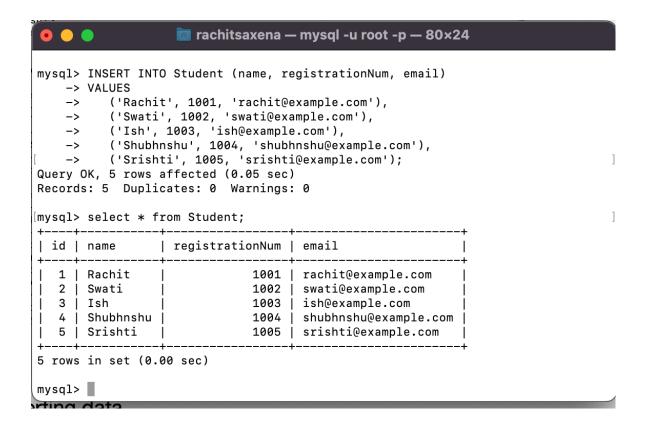
- 1. create, update, and delete commands in MySQL
 - a. Creating the database -



b. Create table

```
🔟 rachitsaxena — mysql -u root -p — 80×24
[mysql> create table Student(id int primary key auto_increment,name varchar(50) n
ot null, registrationNum int unique, email varchar(100) unique)
Query OK, 0 rows affected (0.03 sec)
[mysql> show tables;
| Tables_in_assissment2 |
| Student
1 row in set (0.00 sec)
[mysql> describe Student;
| Field
                  | Type
                                 | Null | Key | Default | Extra
| id
                  | int
                                        | PRI | NULL
                                                         | auto_increment
                                               I NULL
                  | varchar(50)
                                 l NO
| registrationNum | int
                                   YES
          | varchar(100) | YES | UNI | NULL
4 rows in set (0.03 sec)
```

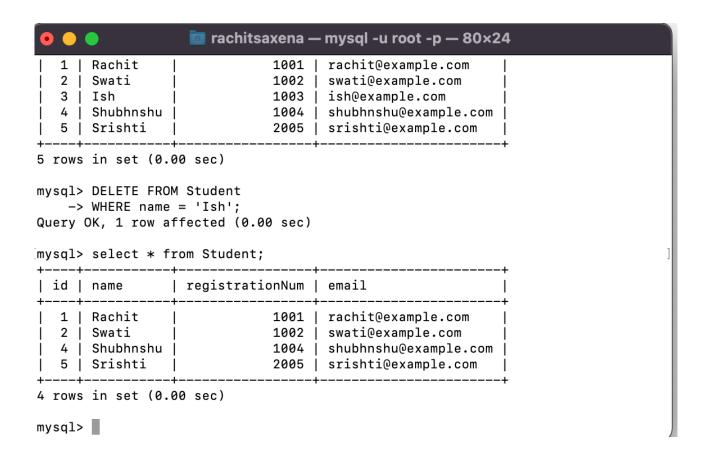
c. Inserting data



d. Updating the tables

```
🔟 rachitsaxena — mysql -u root -p — 80×24
   4 | Shubhnshu |
                              1004 | shubhnshu@example.com |
   5 | Srishti |
                              1005 | srishti@example.com
5 rows in set (0.00 sec)
mysql> UPDATE Student
    -> SET registrationNum = 2005
    -> WHERE name = 'Srishti';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
[mysql> select * from Student;
| id | name | registrationNum | email
  1 | Rachit | 1001 | rachit@example.com
| 2 | Swati |
| 3 | Ish |
                           1002 | swati@example.com
1003 | ish@example.com
| 4 | Shubhnshu |
| 5 | Srishti |
                             1004 | shubhnshu@example.com |
                             2005 | srishti@example.com
5 rows in set (0.00 sec)
mysql>
```

e. Deleting a row



f. create tables and perform joins in mysql

```
🔟 rachitsaxena — mysql -u root -p — 92×34
mysql> CREATE TABLE Course (
    -> courseID INT PRIMARY KEY,
          courseName VARCHAR(50)
   ->
   -> );
Query OK, 0 rows affected (0.03 sec)
mysql> INSERT INTO Course (courseID, courseName)
   -> VALUES
   -> (1001, 'Computer Science '),
        (1002, 'Mechanical Engineering '),
(1003, 'Human Computer Interface');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> CREATE TABLE JoinedData AS
   -> SELECT Student.*, Course.courseName
   -> FROM Student
   -> JOIN Course ON Student.registrationNum = Course.courseID;
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> show JoinedData;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use near 'JoinedData' at line 1
mysql> select * from JoinedData;
| id | name | registrationNum | email
                                                   | courseName
| 1 | Rachit |
                 1001 | rachit@example.com | Computer Science
| 2 | Swati |
                          1002 | swati@example.com | Mechanical Engineering
2 rows in set (0.00 sec)
mysql>
```

2. a. Creating and inserting a table in mongoDB

```
1. mongo (mongo)
> db.dropDatabase()
{ "dropped" : "hr", "ok" : 1 }
> show dbs
admin
       0.000GB
config 0.000GB
local 0.000GB
test
       0.000GB
> show dbs
admin 0.000GB
config 0.000GB
local
       0.000GB
test
        0.000GB
> use hr
switched to db hr
> db.employees.insert
db.employees.insert(
                          db.employees.insertMany( db.employees.insertOne(
> db.employees.insert({
      "employeeNumber": 2,
      "firstName": "Alexis",
      "lastName": "Bull",
      "company": "Some Company, LLC.",
      "age": 30
WriteResult({ "nInserted" : 1 })
```

b. Deleting a record from the document

```
1. mongo (mongo)
       "_id" : ObjectId("5c6470318d81086dd22649be"),
       "employeeNumber" : 2,
       "firstName" : "Alexis",
       "lastName" : "Bull",
       "company": "Some Company, LLC.",
       "age" : 30
> db.employees.findOne({"firstName": "Alexis"}).pretty()
2019-02-13T13:31:14.713-0600 E QUERY [js] TypeError: db.employees.findOne(...
).pretty is not a function :
@(shell):1:1
> db.employees.findOne({"firstName": "Alexis"})
       "_id" : ObjectId("5c646f9e8d81086dd22649bc"),
       "employeeNumber" : 2,
       "firstName" : "Alexis",
       "lastName" : "Bull",
       "company": "Some Company, LLC.",
       "age" : 30
 db.employees.deleteOne({"employeeNumber": 2})
 "acknowledged" : true, "delet adCount" : 1 }
```

c. Updating a record in a document

```
1. mongo (mongo)
        "nModified" : 0,
        "writeError" : {
                "code": 9,
                "errmsg": "Modifiers operate on fields but we found type string
 instead. For example: {$mod: {<field>: ...}} not {$set: \"IT Versity, Inc.\"}"
})
> db.employees.update({"_id": 2}, {$set: {"company": "IT Versity, Inc."}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.employees.find().pretty()
       "_id" : 2,
       "firstName" : "Alexis",
       "lastName" : "Bull",
       "company" : "IT Versity, Inc.",
       "age" : 30
> db.employees.update(
     { "_id": 2 },
      { "company" : "ITVersity, Inc." }
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModiffied" : 1 })
```