Practical 9: ORM Assignment

Aim: Use any one ORM from Sequelize ORM, Type ORM and Prism ORM and implement following task.

1. **Create Connection**
2. **Create Database**
3. **Create Table**
4. **Insert Record**
5. **Display Record**

Code:

App.js

```
require("dotenv").config();
const express=require("express");
let app=express();


 require("./model");
 const userCtrl=require("./controller/studentController");

app.use(express.json());

app.get("/",(req,res)=>{
    res.send({title:'Welcome to nodeJS, Squilize and MySql tutorial'});
})

//Create database connection
app.get("/crreate-db",(req,res)=>{
    const {Sequelize, DataTypes}=require('sequelize');

    const sequelize=new
Sequelize(process.env.MYSQL_DATABASE,'root',process.env.MYSQL_PASSWORD,{
        host:process.env.MYSQL_HOST,
        dialect:process.env.DIALECT,
        logging:false,
        pool:{max:5,min:0,idle:1000}
    });

    sequelize.authenticate()
    .then(()=>{
        console.log("Connected successfully with Database")
    })
    .catch(err=>{
        console.log("Issue with DB"+err)
    })})

//Create student table
app.get("/create-table",(req,res)=>{
    const db={}
    db.Sequelize=Sequelize;
    db.sequelize=sequelize;

    db.student=require('./student.js')(sequelize, DataTypes);

    db.sequelize.sync({force:false})

    .then(()=>{
        console.log("Database is Sync")
    })
    module.exports=db;})

//CRUD operations
app.get("/insert", userCtrl.addStudent)//insert
app.get("/insertStudents", userCtrl.insertStudents)//insert multiple
app.get("/update", userCtrl.updateStudent)//update
app.get("/delete", userCtrl.deleteStudent)//delete
app.get("/findStudent", userCtrl.findStudent)//find




const port=process.env.APP_PORT || 4000;
app.listen(port,()=>{
    console.log("server listining on port: ",port);
})
```

studentController.js

```
const db=require("../model");
const Students=db.student;
const addStudent=async(req,res)=>{
    let data=await
Students.create({name:"Kalindi",email:"kalindi@gmail.com",gender:"Female"})
    console.log(data.dataValues);
    await data.save();
    let response={
        data:"inserted"
    }
    res.status(200).json(response);
```

```
}
const insertStudents=async(req,res)=>{
    let data=await Students.bulkCreate([
        {name:"Jeel",email:"jeel@gmail.com",gender:"Male"},
        {name:"Meet",email:"meet@gmail.com",gender:"Male"},
        {name:"Krupa",email:"krupa@gmail.com",gender:"Female"},

    ])
     let response={
        data:"inserted"
    }
    res.status(200).json(response);
}
const updateStudent=async(req,res)=>{
    let data=await Students.update({name:"VIvek Mehta"},
    {where:{
        id:1
    }
});

    console.log(data.dataValues);
    await data.save();
    let response={
        data:"updated"
    }
    res.status(200).json(response);
}

const deleteStudent=async(req,res)=>{
    let data=await Students.destroy({
        where :{
        id:1
        }
        //truncate:true;
    });
    let response={
        data:"deleted"
    }
    res.status(200).json(response);

}
const findStudent=async(req,res)=>{
    let data=await Students.findAll({});
    let response={
        data:data
    }
    res.status(200).json(response);
```

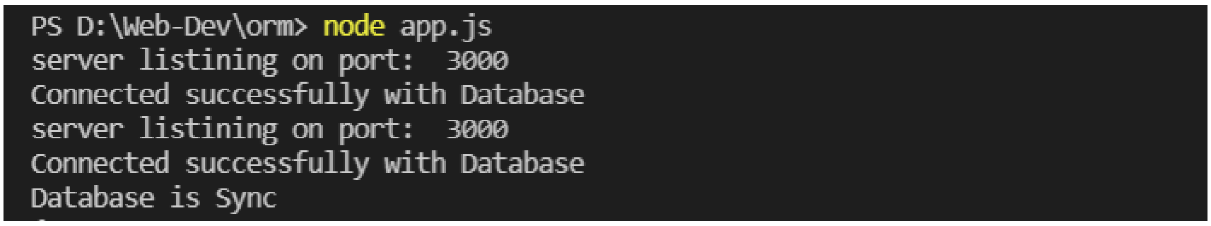
```
}

module.exports={addStudent,updateStudent,deleteStudent,insertStudents,findStud
ent}
```

Student.js

```
module.exports=(sequelize,DataTypes)=>{
    const student=sequelize.define("student",{
        name:DataTypes.STRING,
        email:{
            type:DataTypes.STRING,
            defaultValue:'abc@gmail.com'
        },
        gender:{
            type:DataTypes.STRING
        }
    },{
    })
     return student;
}
```

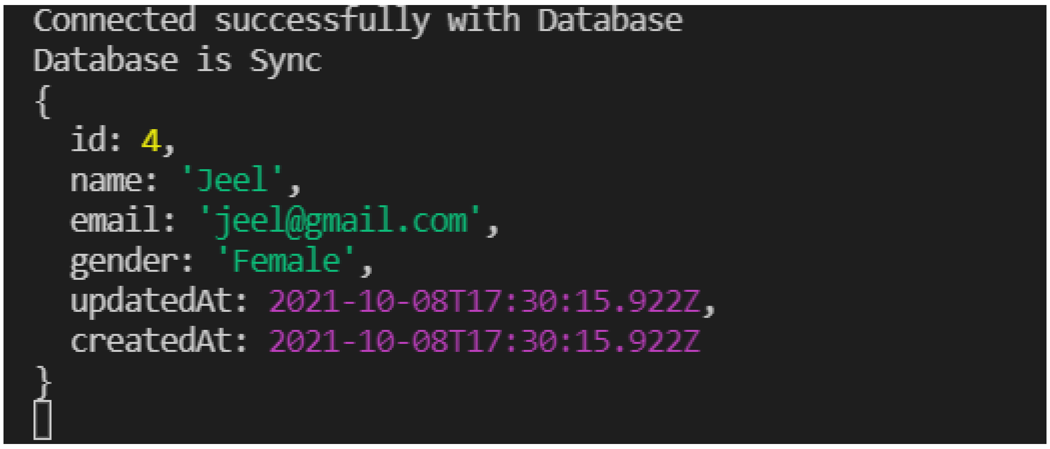
Output:

Database Connection:

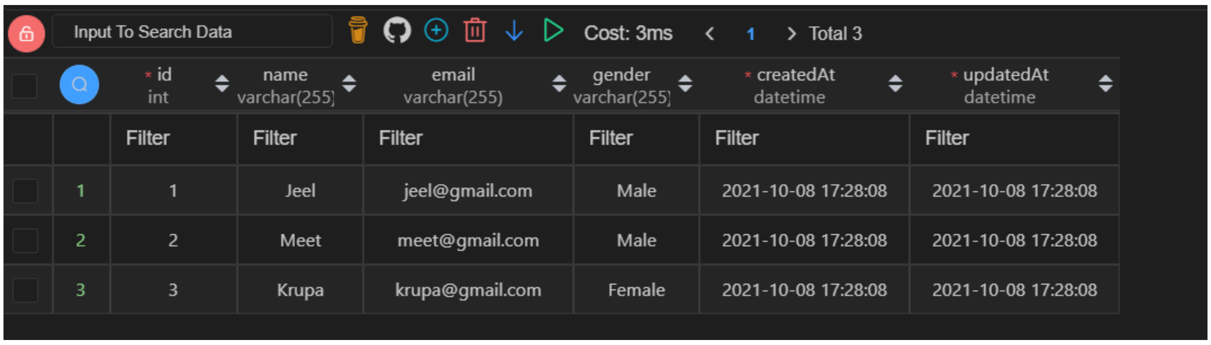


```
PS D:\Web-Dev\orm> node app.js
server listining on port:  3000
Connected successfully with Database
server listining on port:  3000
Connected successfully with Database
Database is Sync
```

Insert Data-



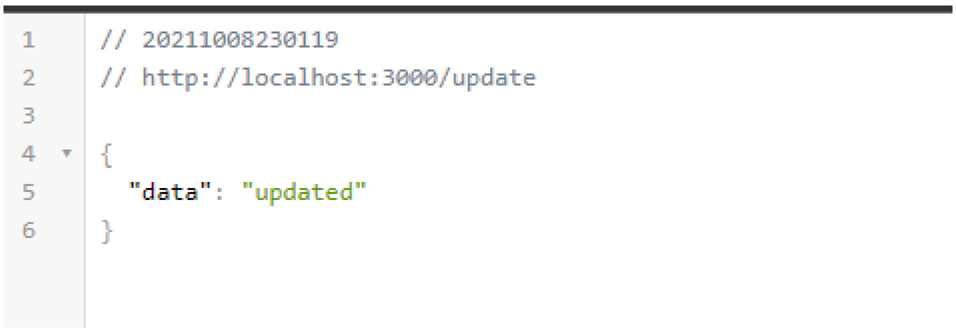
```
Connected successfully with Database
Database is Sync
{
  id: 4,
  name: 'Jeel',
  email: 'jeel@gmail.com',
  gender: 'Female',
  updatedAt: 2021-10-08T17:30:15.922Z,
  createdAt: 2021-10-08T17:30:15.922Z
}
```



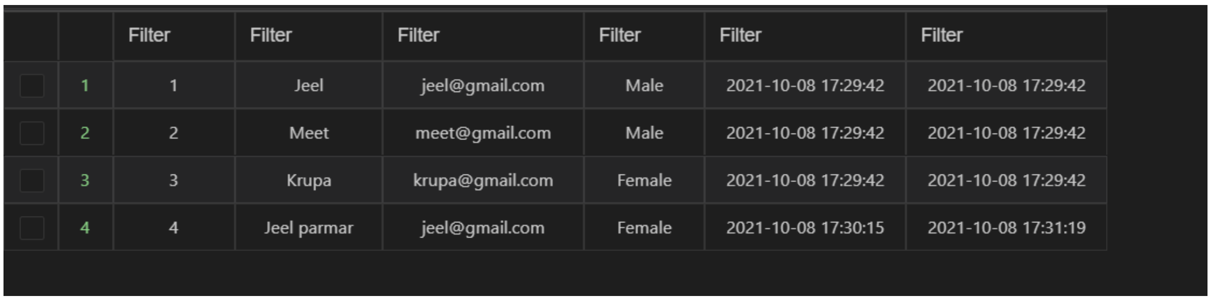
Input To Search Data Cost: 3ms 1 Total 3

	id int	name varchar(255)	email varchar(255)	gender varchar(255)	createdAt datetime	updatedAt datetime
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Jeel	jeel@gmail.com	Male	2021-10-08 17:28:08	2021-10-08 17:28:08
2	2	Meet	meet@gmail.com	Male	2021-10-08 17:28:08	2021-10-08 17:28:08
3	3	Krupa	krupa@gmail.com	Female	2021-10-08 17:28:08	2021-10-08 17:28:08

Update Data

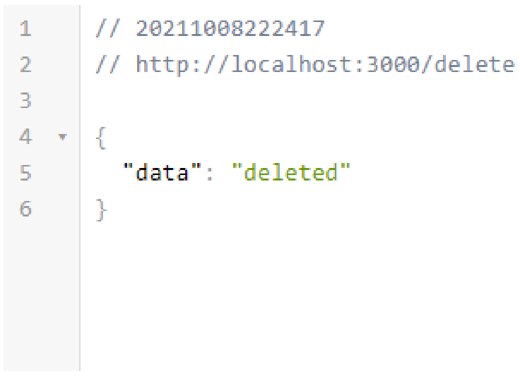


```
// 20211008230119
// http://localhost:3000/update

{
  "data": "updated"
}
```



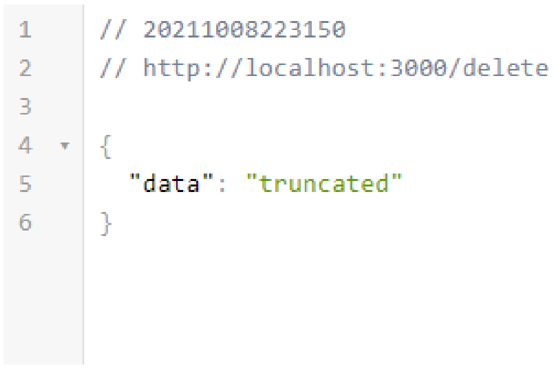
		Filter	Filter	Filter	Filter	Filter	Filter
	1	1	Jeel	jeel@gmail.com	Male	2021-10-08 17:29:42	2021-10-08 17:29:42
	2	2	Meet	meet@gmail.com	Male	2021-10-08 17:29:42	2021-10-08 17:29:42
	3	3	Krupa	krupa@gmail.com	Female	2021-10-08 17:29:42	2021-10-08 17:29:42
	4	4	Jeel parmar	jeel@gmail.com	Female	2021-10-08 17:30:15	2021-10-08 17:31:19

Delete Data:



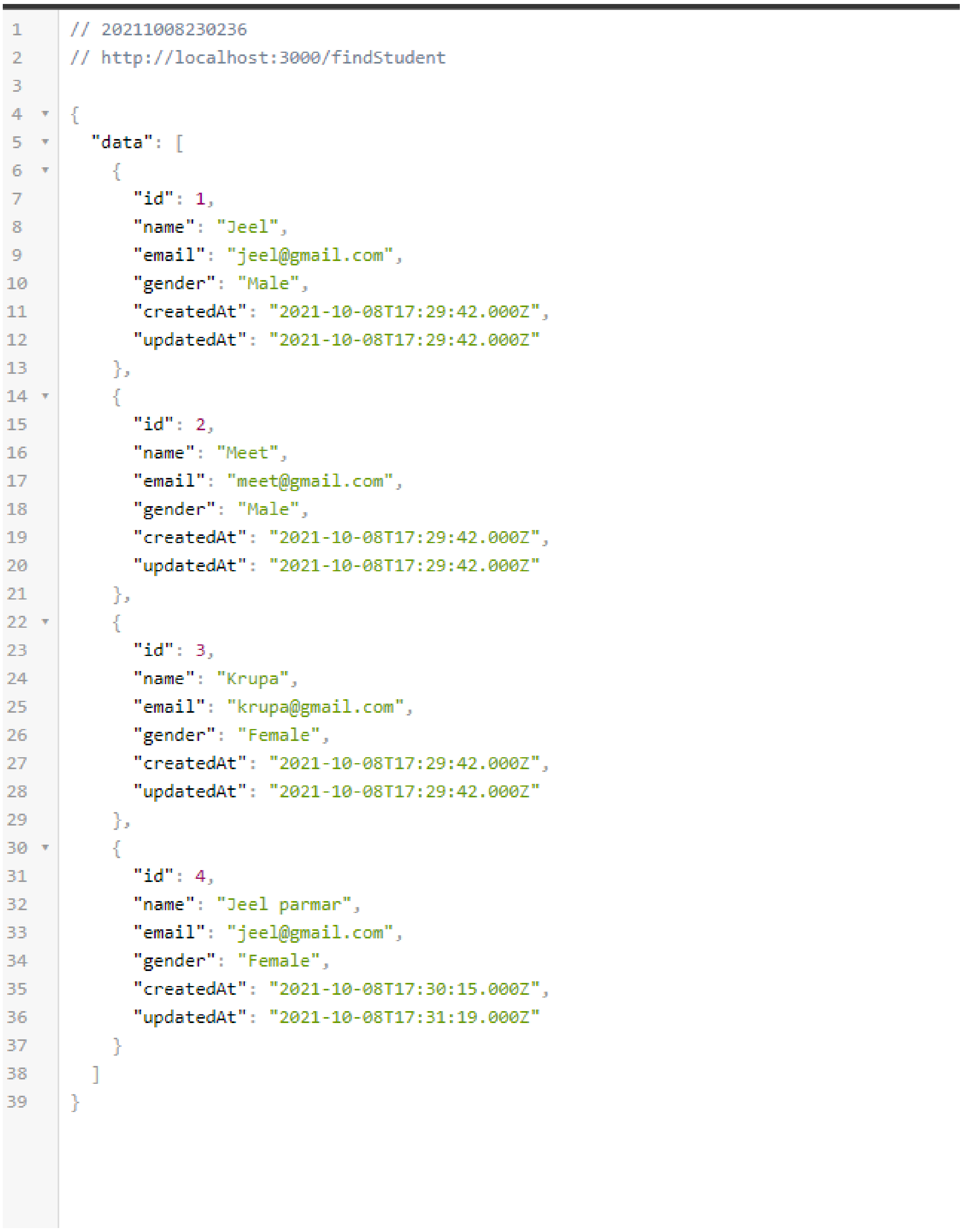
```
// 20211008222417
// http://localhost:3000/delete

{
  "data": "deleted"
}
```

Truncate data:



```
// 20211008223150
// http://localhost:3000/delete

{
  "data": "truncated"
}
```

Find Students



```
// 20211008230236
// http://localhost:3000/findStudent

{
  "data": [
    {
      "id": 1,
      "name": "Jeel",
      "email": "jeel@gmail.com",
      "gender": "Male",
      "createdAt": "2021-10-08T17:29:42.000Z",
      "updatedAt": "2021-10-08T17:29:42.000Z"
    },
    {
      "id": 2,
      "name": "Meet",
      "email": "meet@gmail.com",
      "gender": "Male",
      "createdAt": "2021-10-08T17:29:42.000Z",
      "updatedAt": "2021-10-08T17:29:42.000Z"
    },
    {
      "id": 3,
      "name": "Krupa",
      "email": "krupa@gmail.com",
      "gender": "Female",
      "createdAt": "2021-10-08T17:29:42.000Z",
      "updatedAt": "2021-10-08T17:29:42.000Z"
    },
    {
      "id": 4,
      "name": "Jeel parmar",
      "email": "jeel@gmail.com",
      "gender": "Female",
      "createdAt": "2021-10-08T17:30:15.000Z",
      "updatedAt": "2021-10-08T17:31:19.000Z"
    }
  ]
}
```