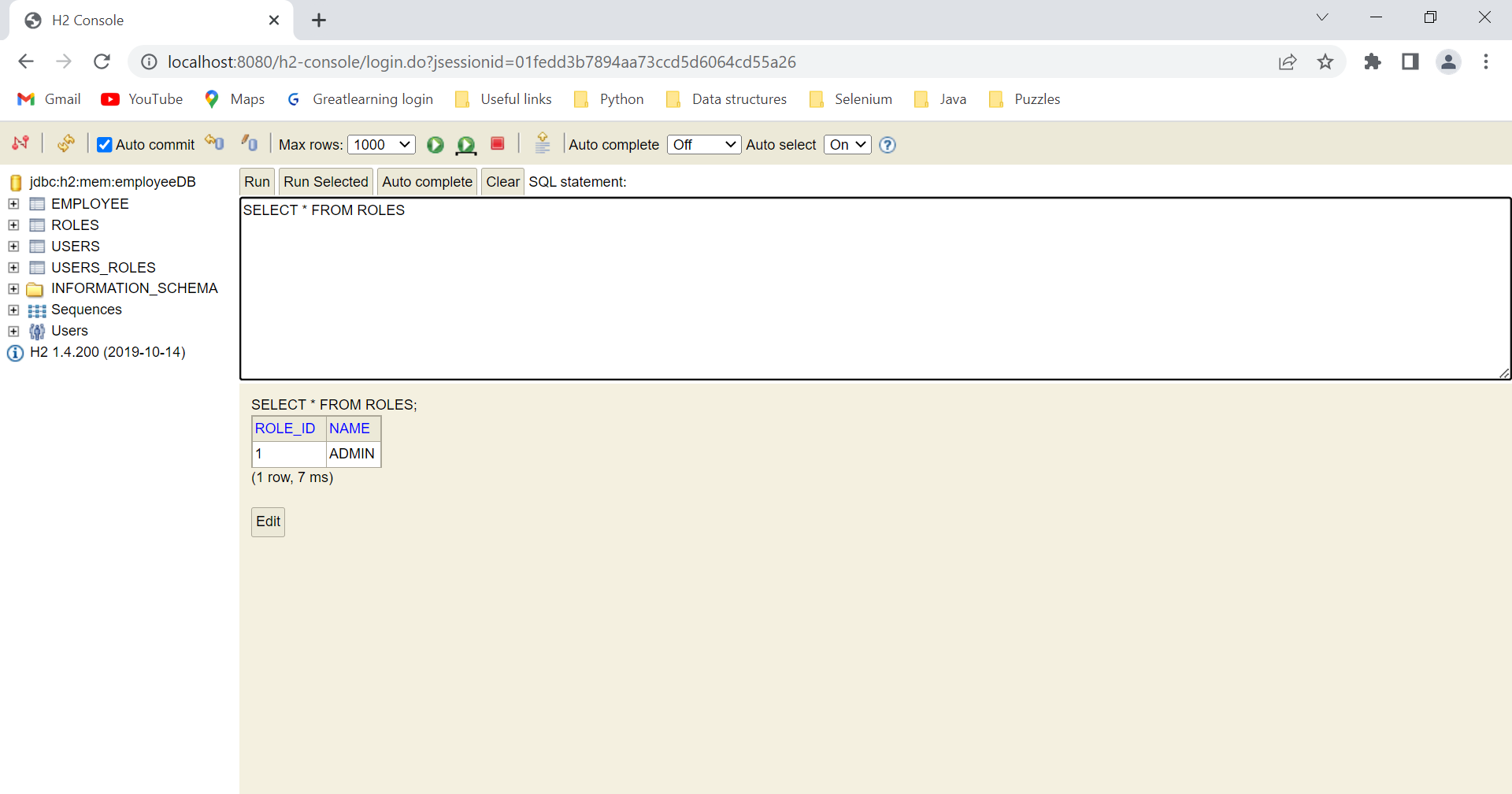
**Add New Role with Authorized User:**

Username/Password: admin/superadmin

**Before Adding new role:**

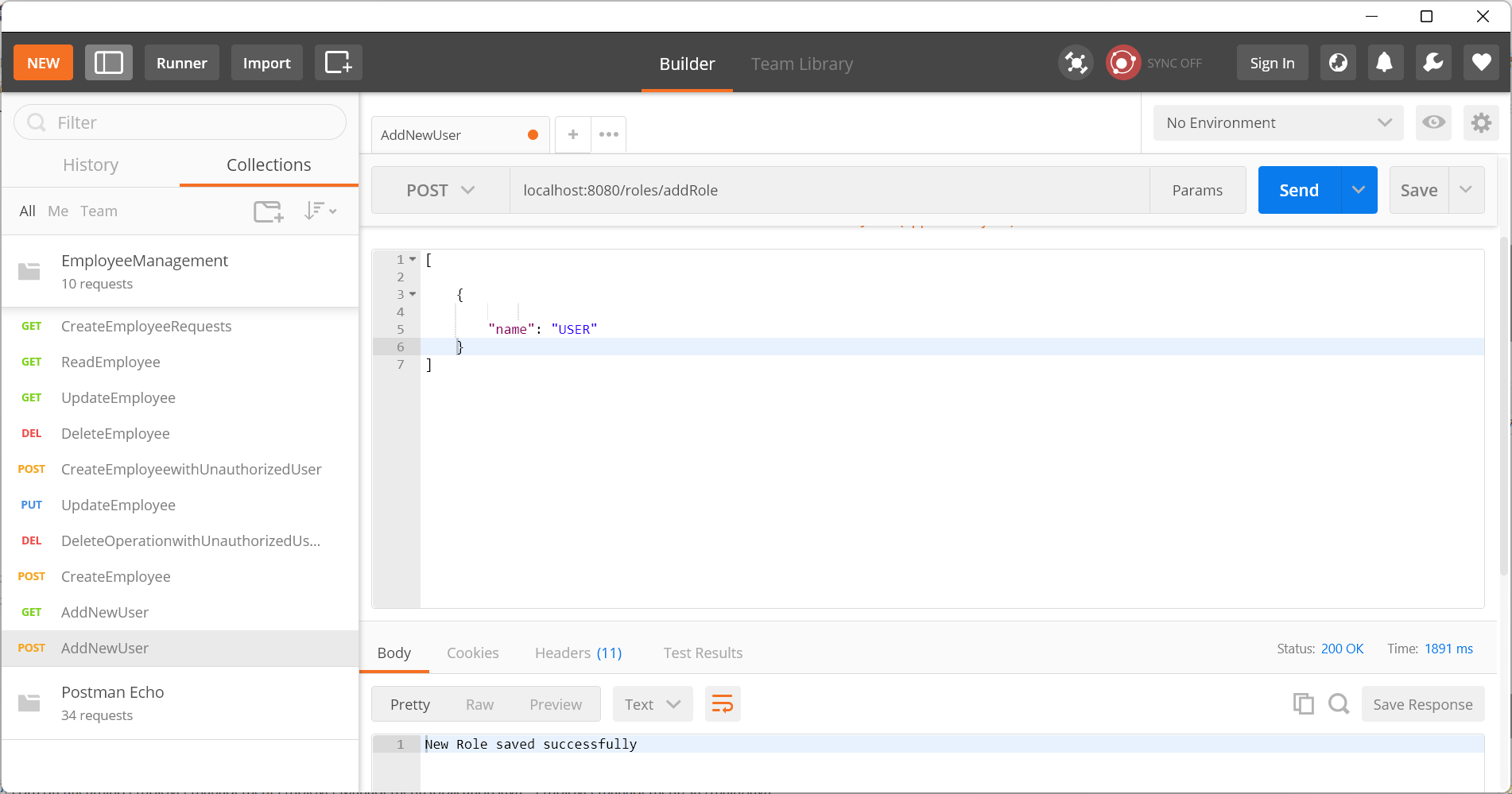
**Data in DB:**



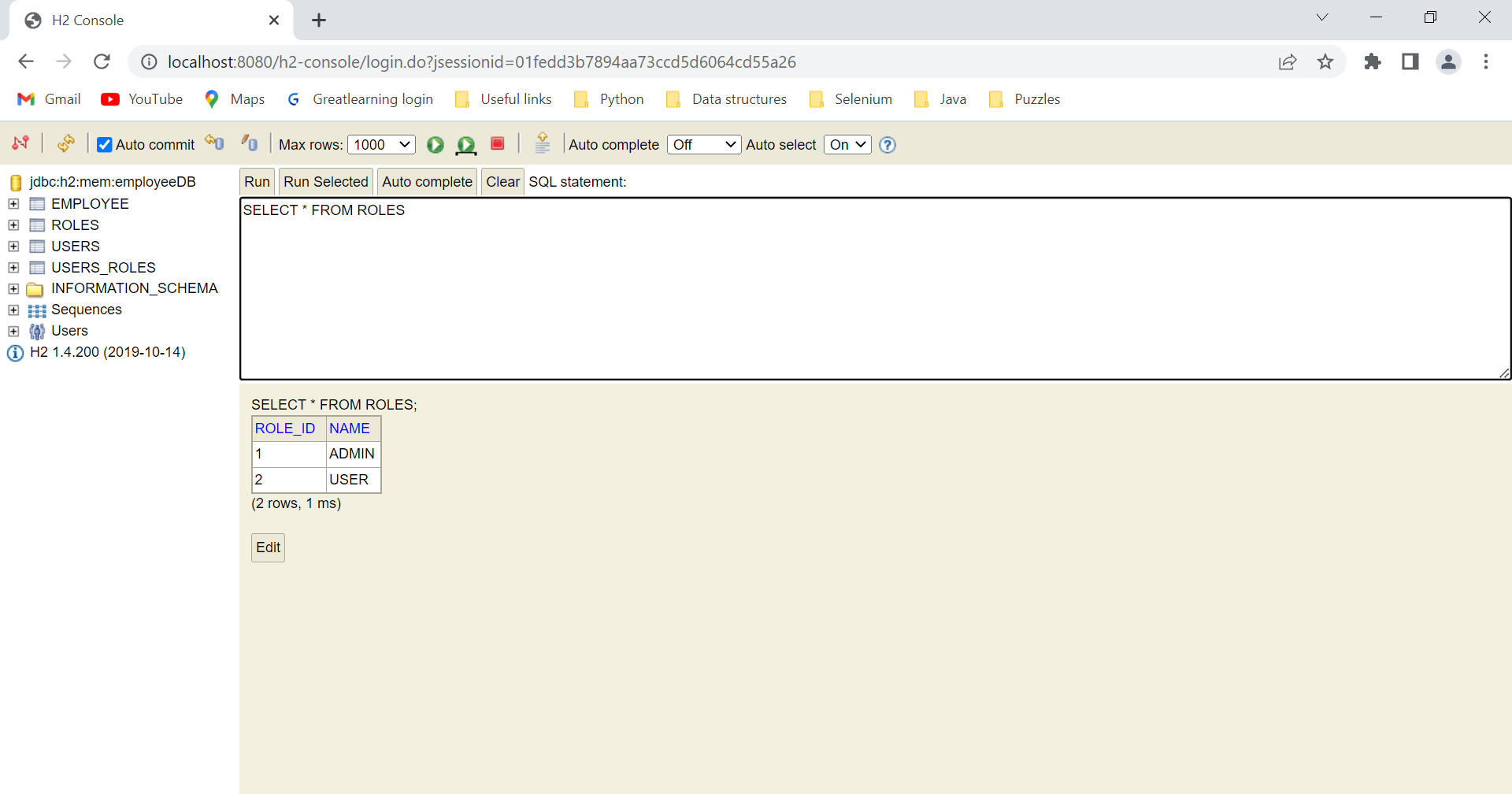
**After Adding:**

**Postman:**

**Endpoint:** http://localhost:8080/roles/addRole

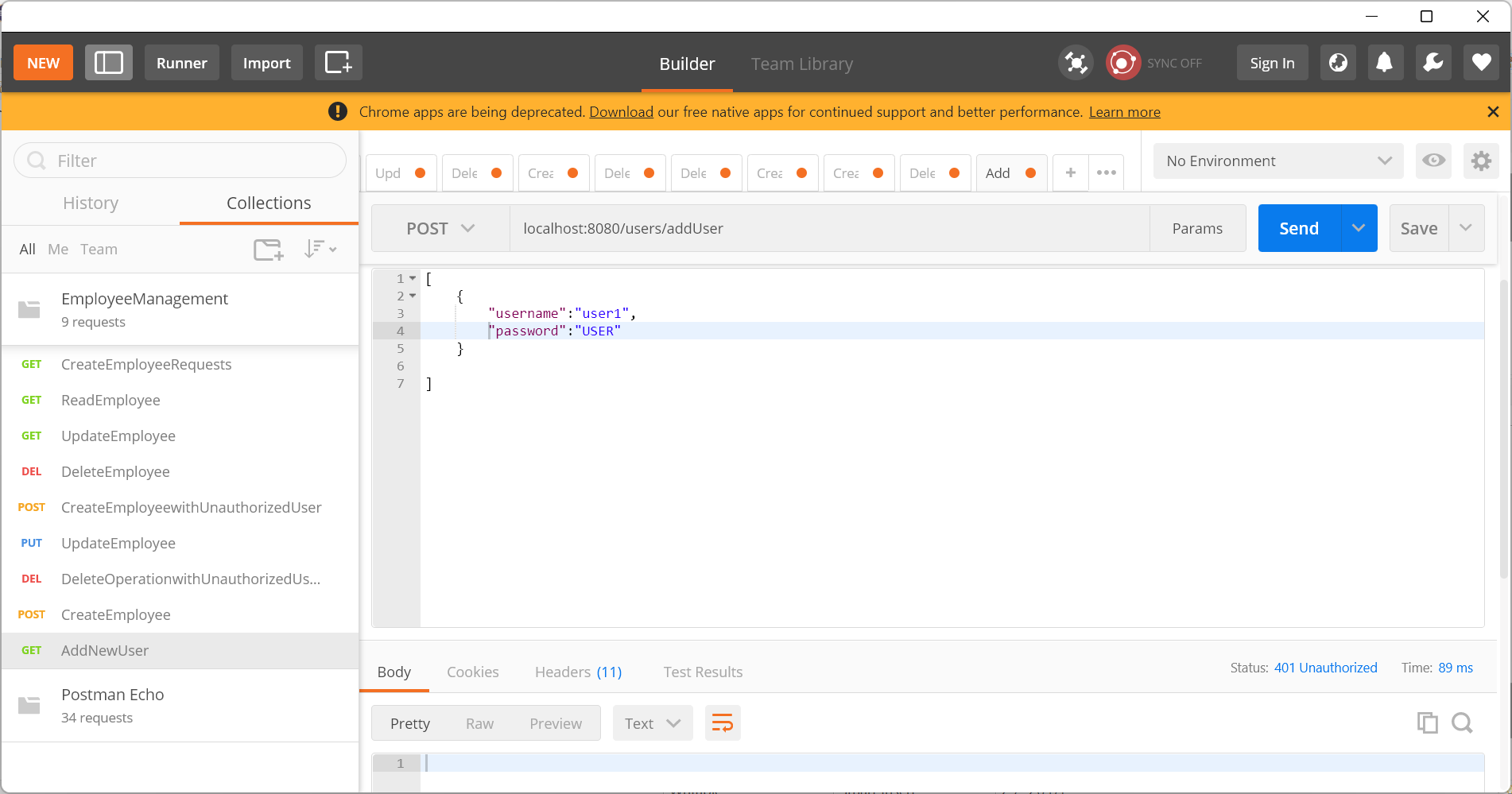


**Data in DB:**



**Add New User without Authorization:**

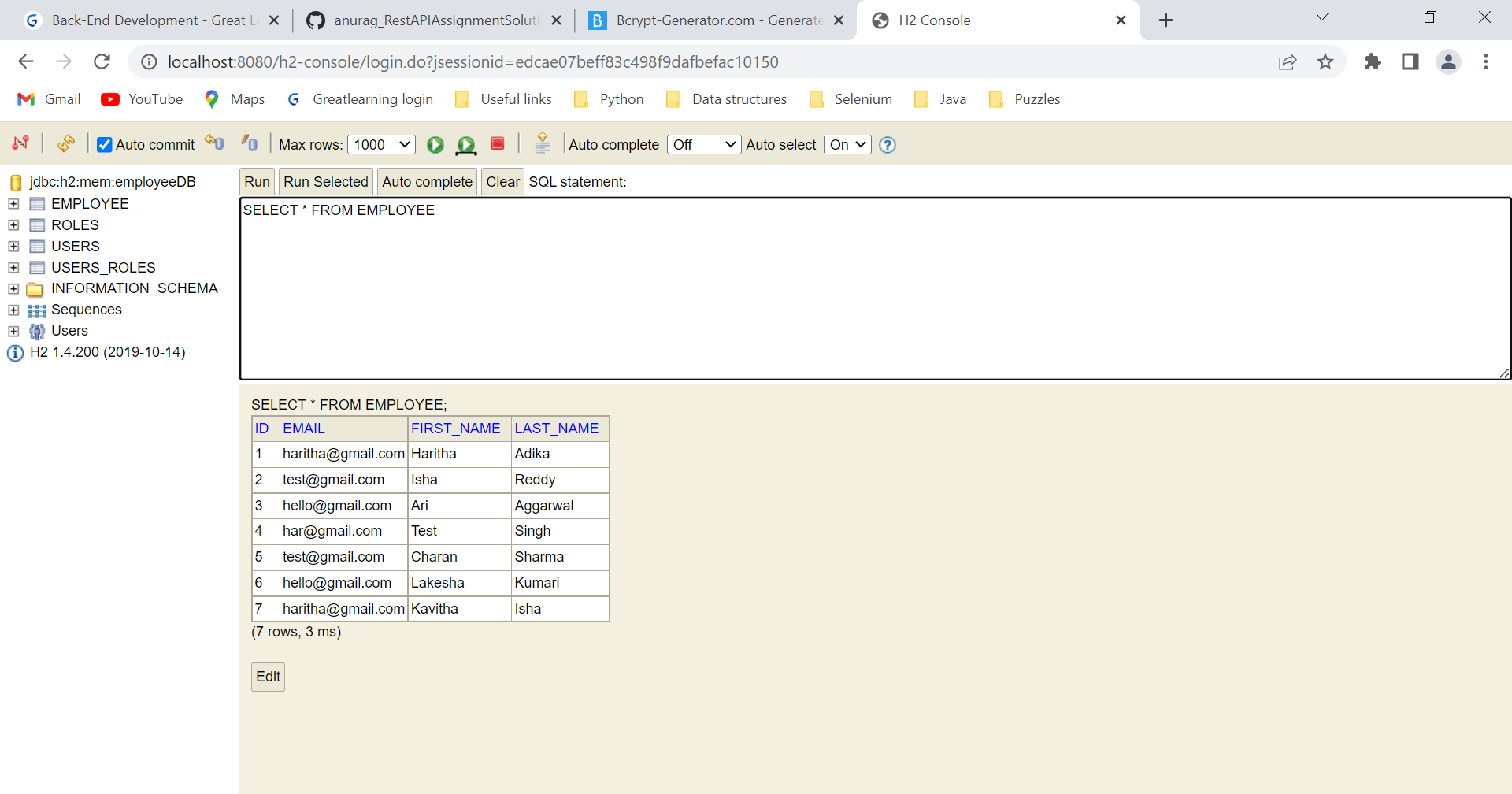
**Postman:** Giving 401 Unauthorized error, if we try to add a new user without using ADMIN credentials



**Create New Employee using Unauthorized User:**

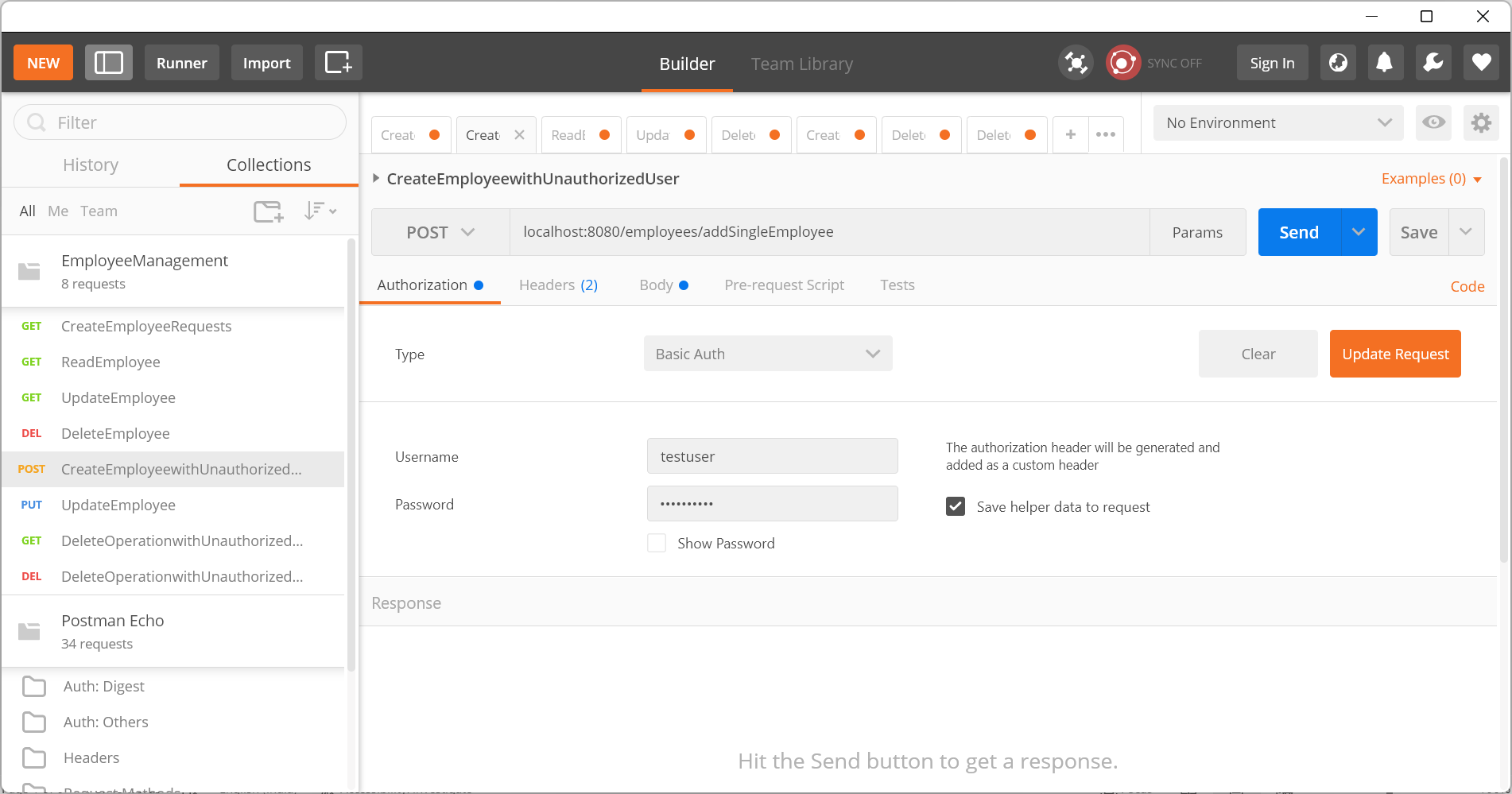
**Before:**

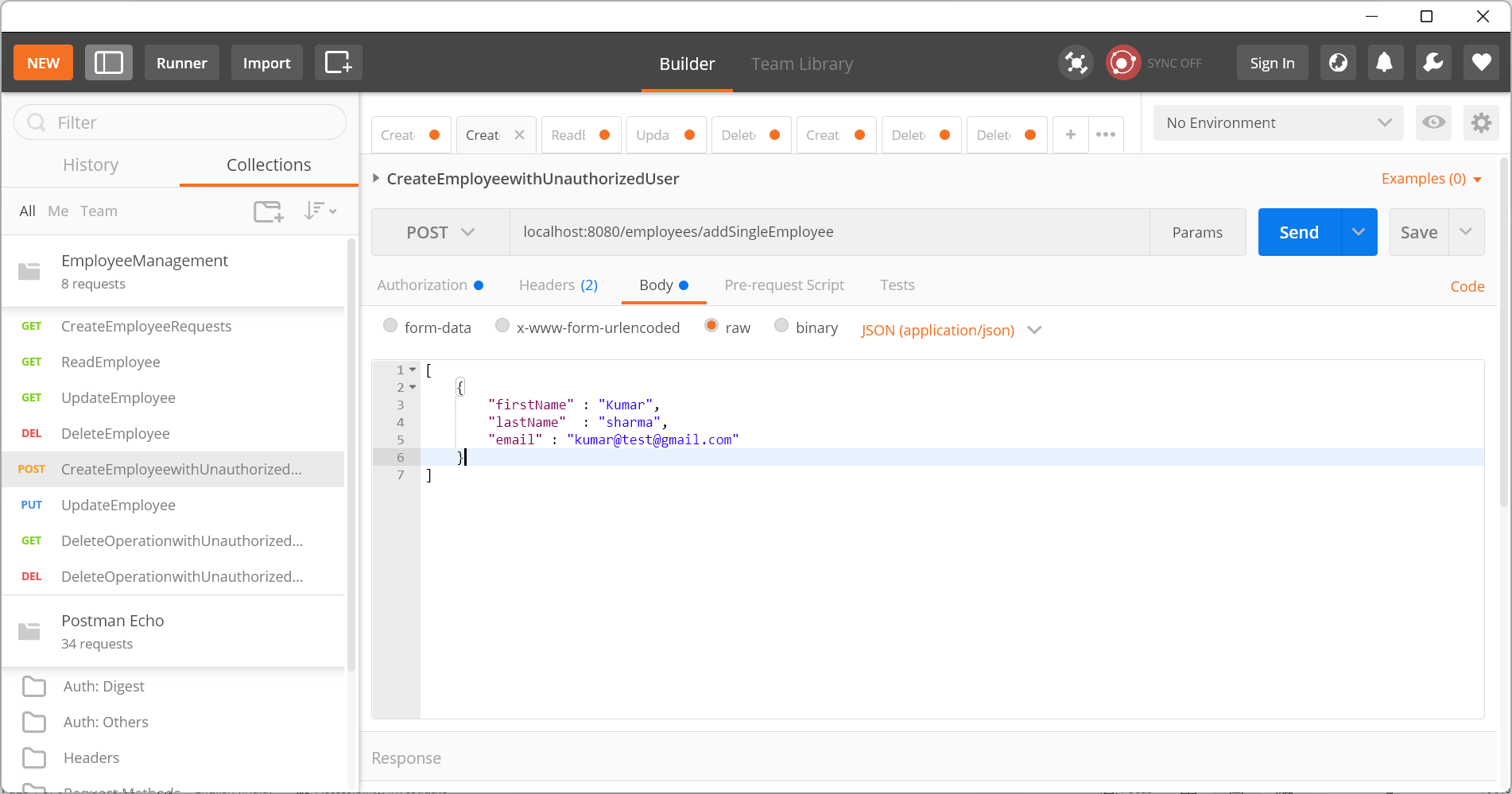
**Data In DB:**



**Postman:**

**Endpoint :** http://localhost:8080/employees/addSingleEmployee

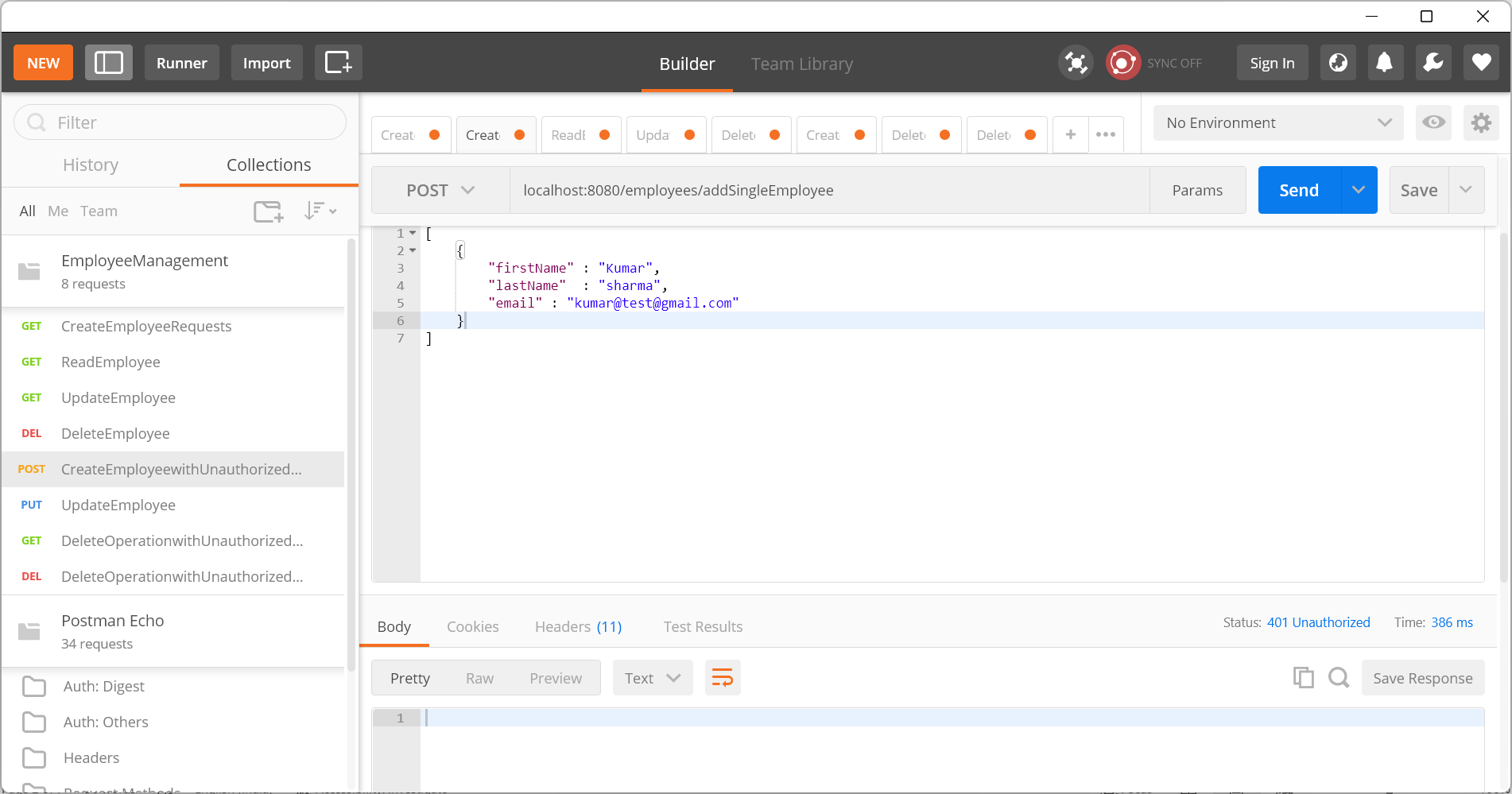




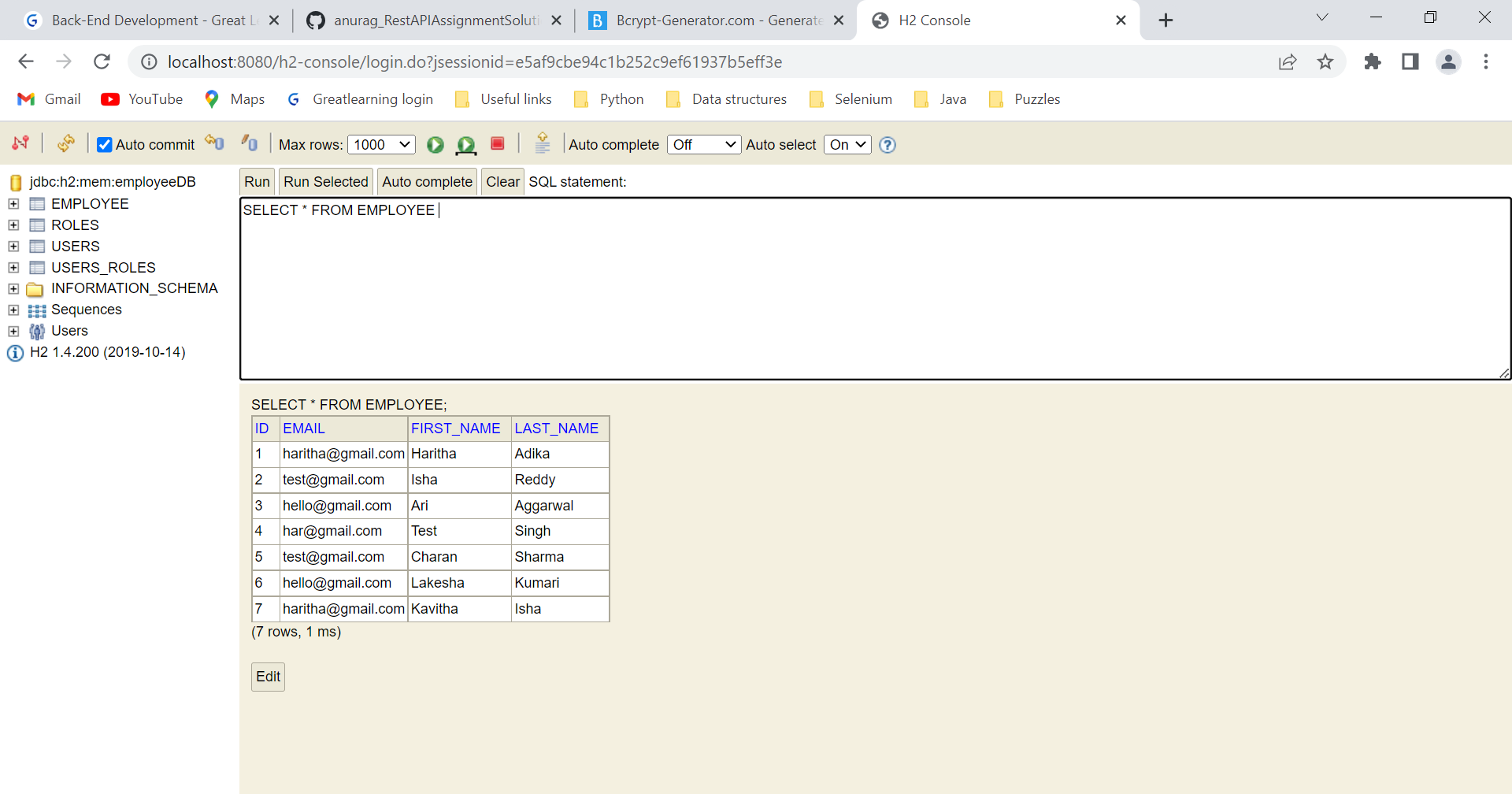
**After:**

**Postman:**

**Endpoint :** http://localhost:8080/employees/addSingleEmployee



**Data in Db is unchanged:**

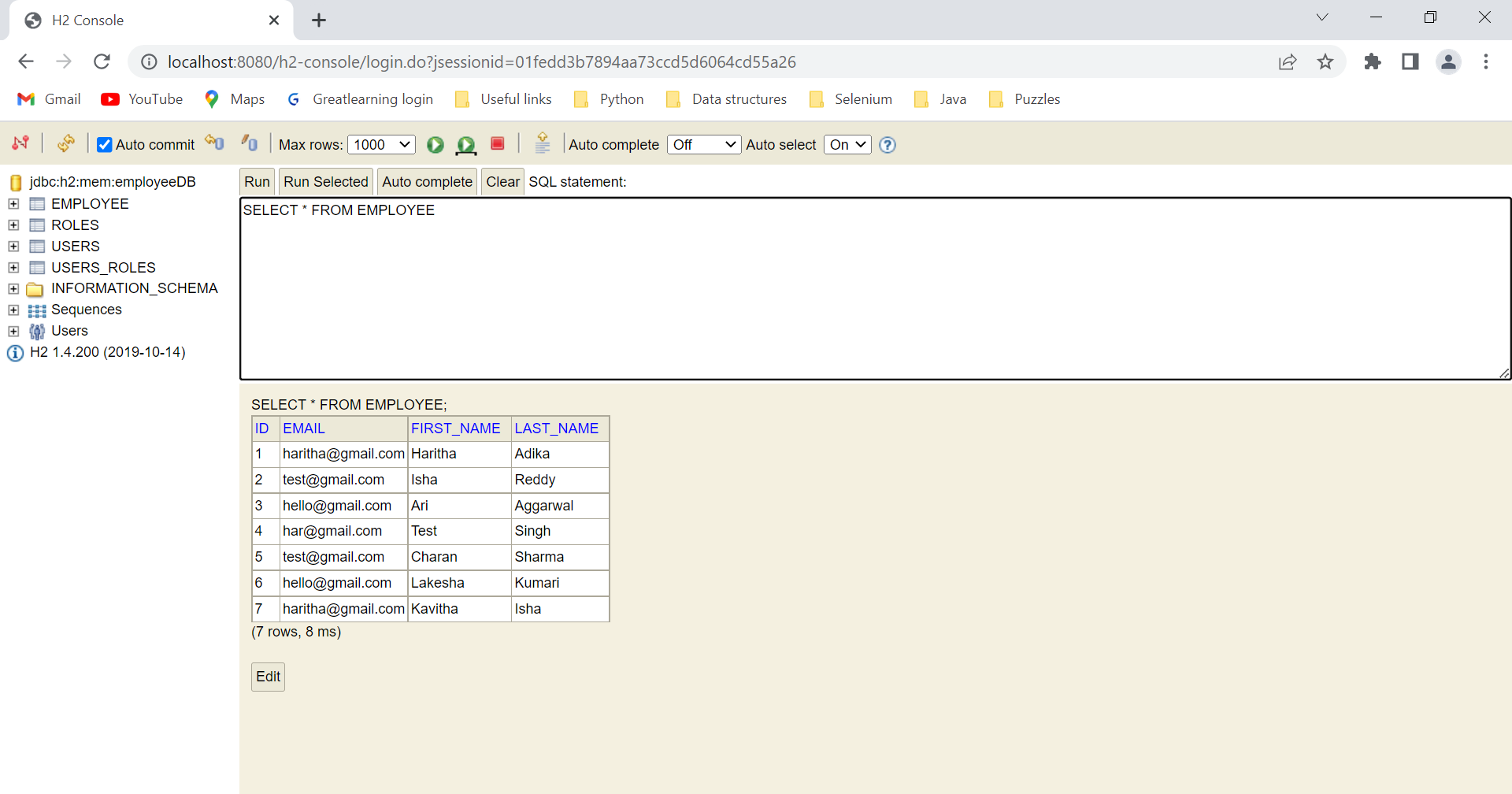


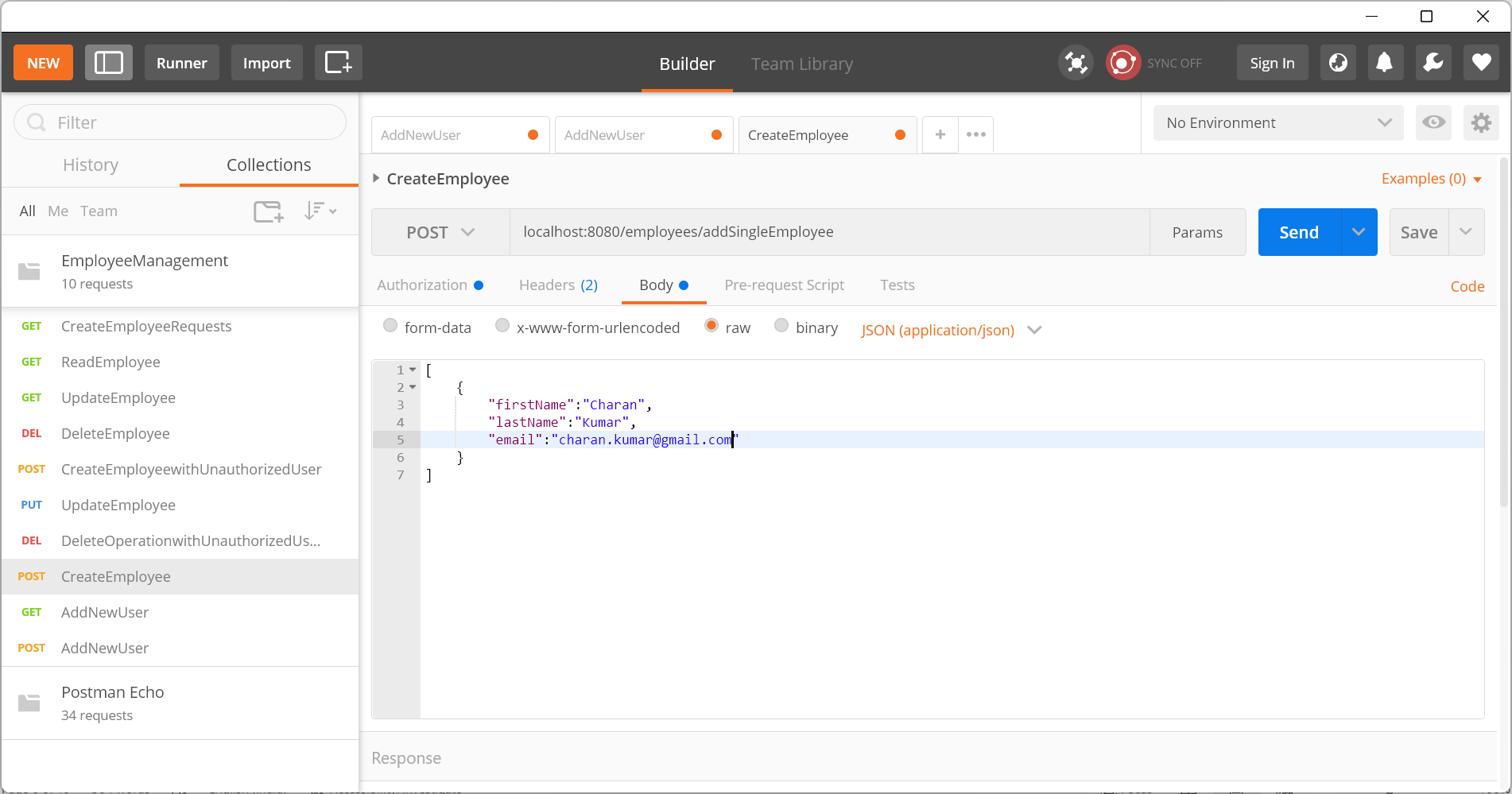
**Create New Employee with Authorized user:**

Username: admin

Password: superadmin

**Data in DB before adding new employee details:**

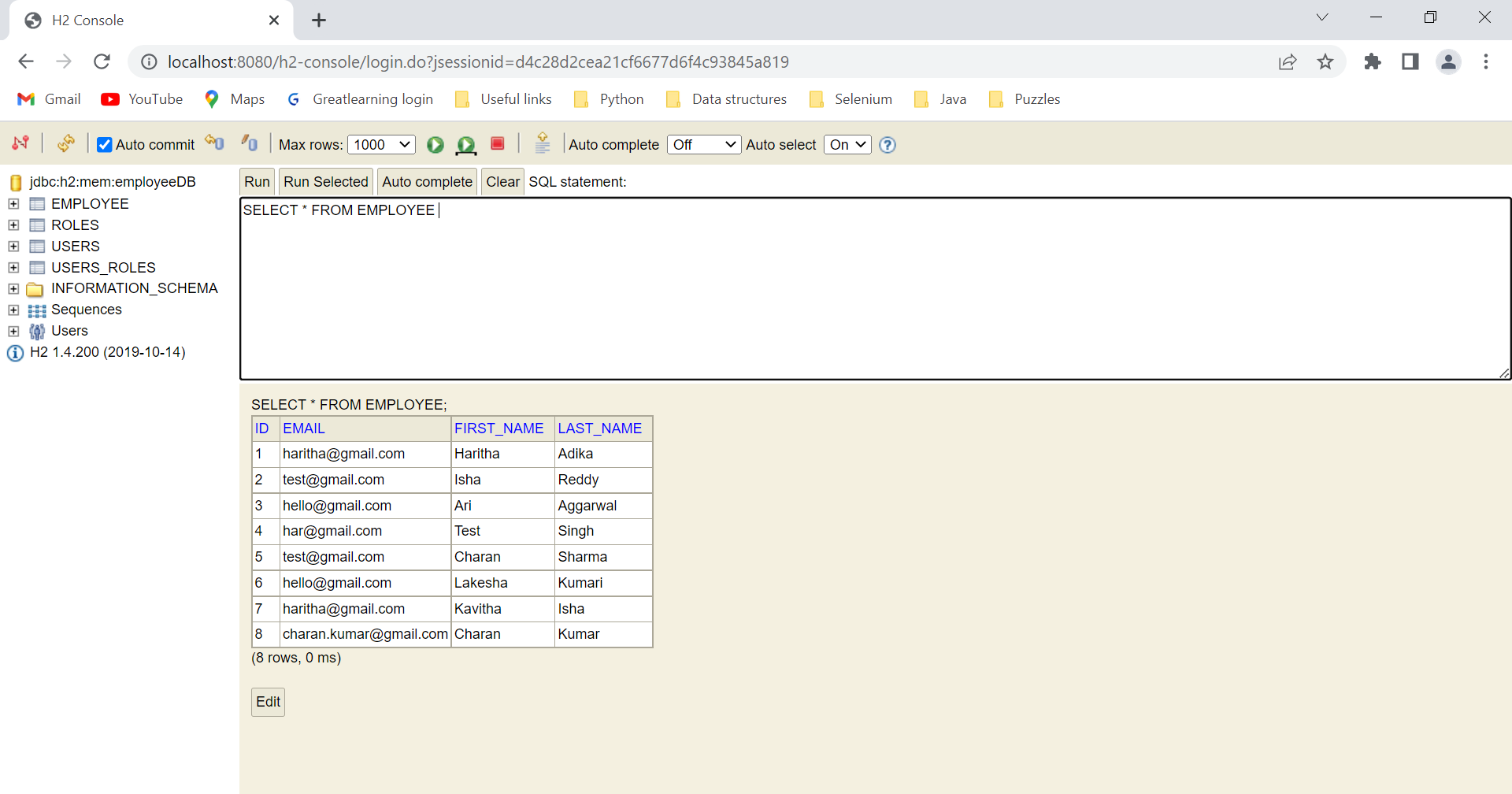




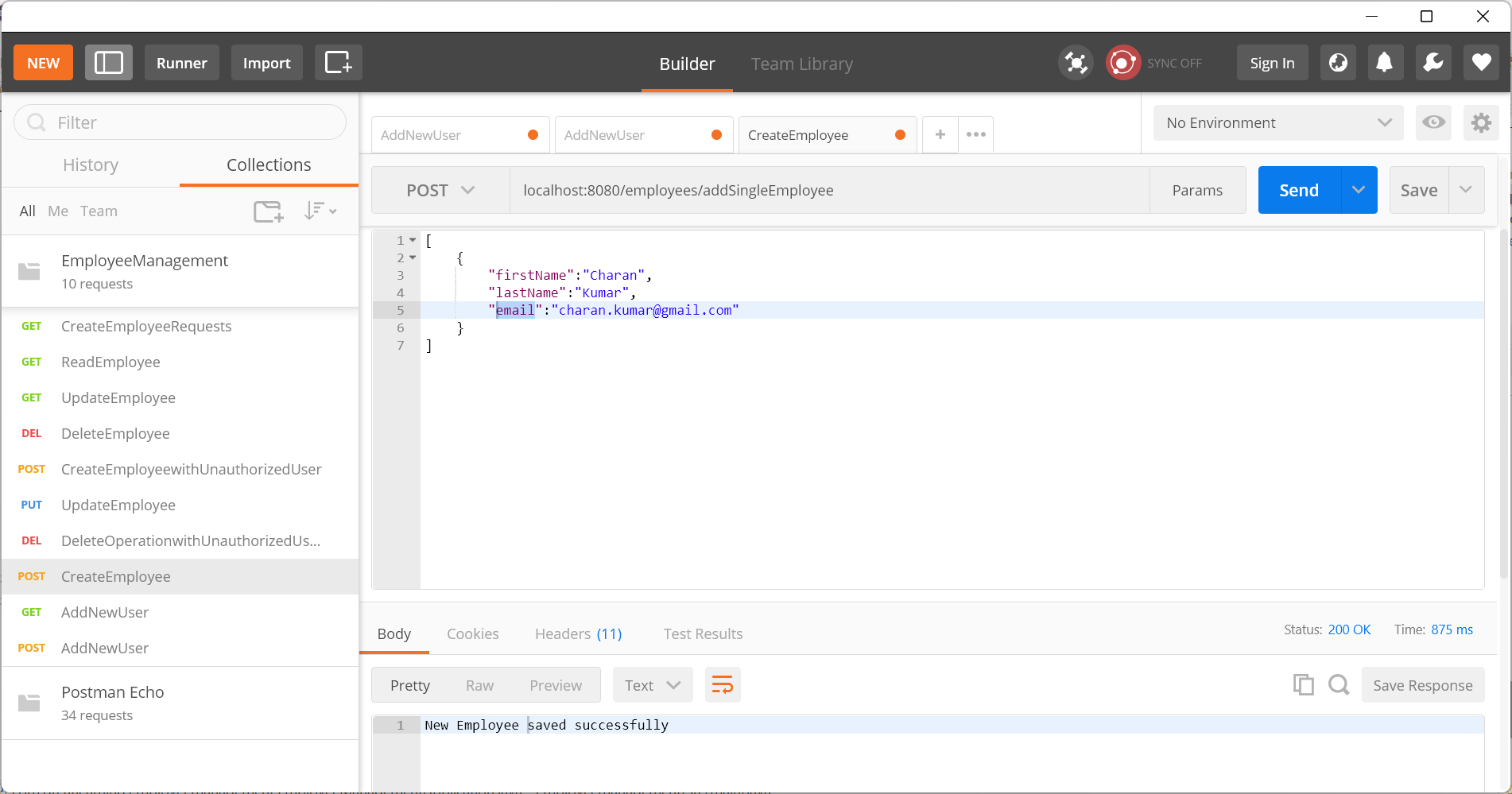
**After:**

**Endpoint :** <http://localhost:8080/employees/addSingleEmployee>

**Data In DB:**



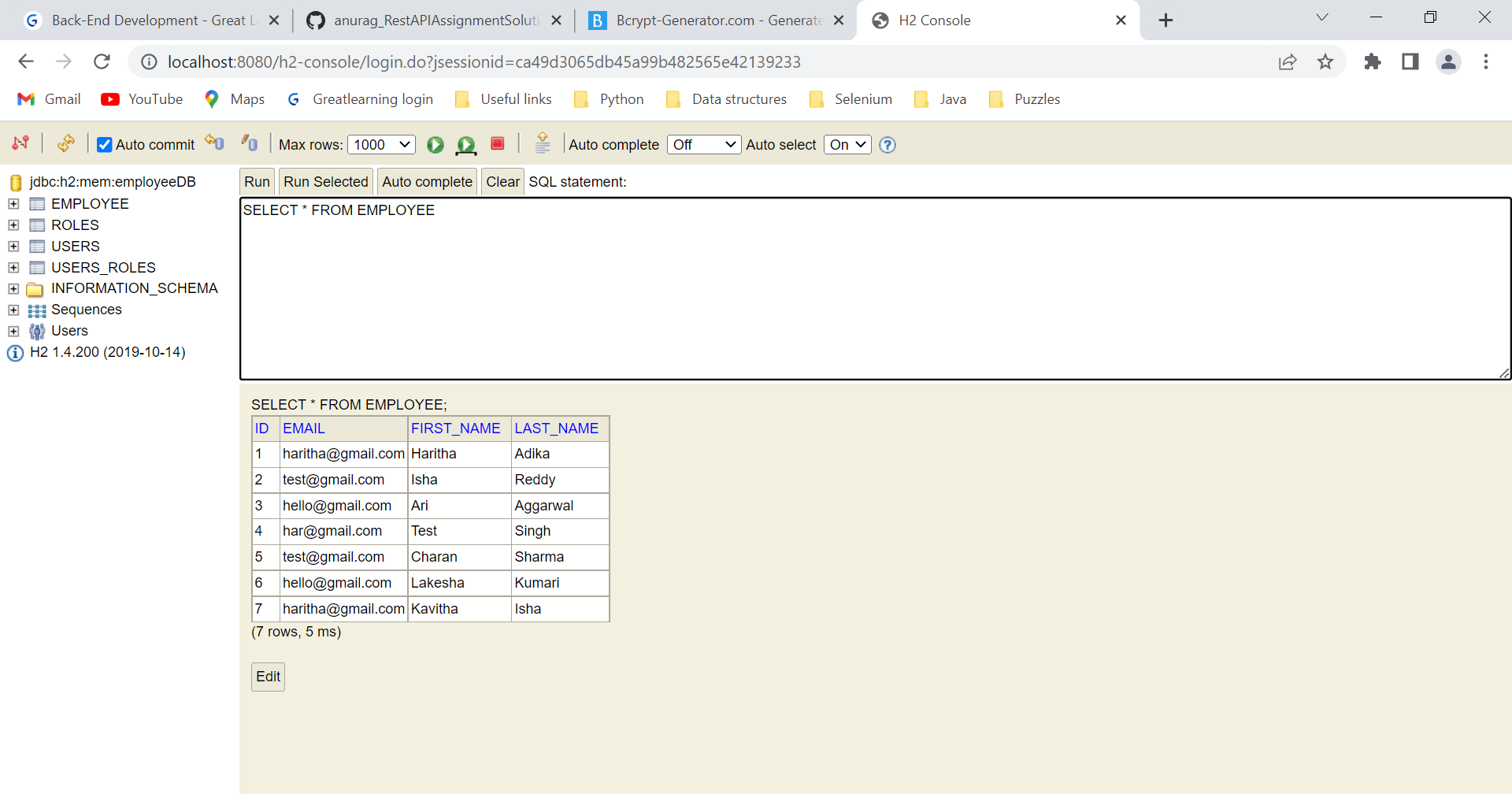
**Postman:**



**Read Employee:**

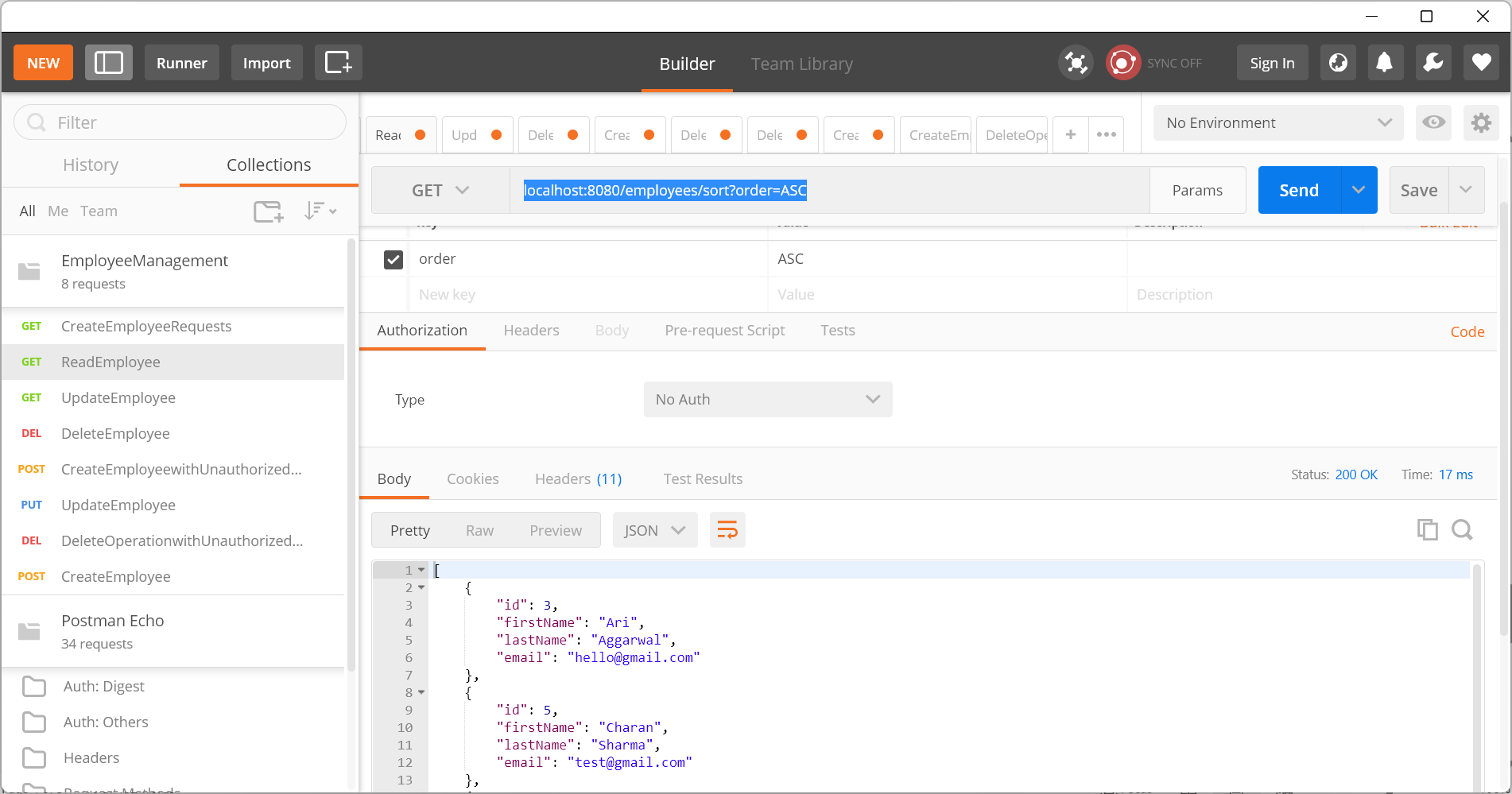
1. list all employee records sorted on their first name in either ascending order or descending order

**Data in DB:**



**Endpoint for sort in Ascending order:**

Endpoint : http://localhost:8080/employees/sort?order=ASC



**Response Body:**

[

{

"id": 3,

"firstName": "Ari",

"lastName": "Aggarwal",

"email": "hello@gmail.com"

},

{

"id": 5,

"firstName": "Charan",

"lastName": "Sharma",

"email": "test@gmail.com"

},

{

"id": 1,

"firstName": "Haritha",

"lastName": "Adika",

"email": "haritha@gmail.com"

},

{

"id": 2,

"firstName": "Isha",

"lastName": "Reddy",

"email": "test@gmail.com"

},

{

"id": 7,

"firstName": "Kavitha",

"lastName": "Isha",

"email": "haritha@gmail.com"

},

{

"id": 6,

"firstName": "Lakesha",

"lastName": "Kumari",

"email": "hello@gmail.com"

},

{

"id": 4,

"firstName": "Test",

"lastName": "Singh",

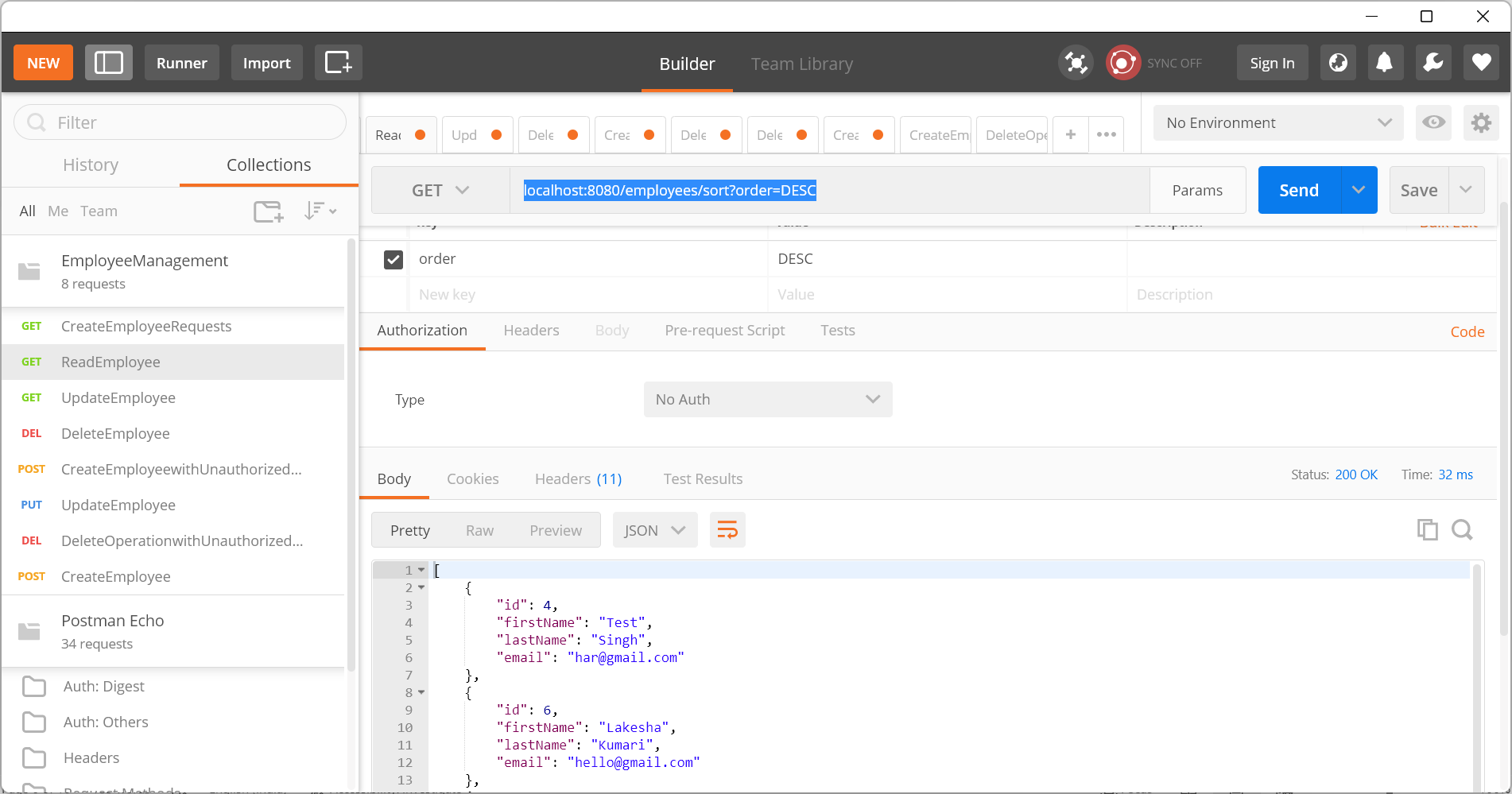
"email": "har@gmail.com"

}

]

**Endpoint to sort employees in Descending order:**

Endpoint : http://localhost:8080/employees/sort?order=DESC



**Response Body:**

[

{

"id": 4,

"firstName": "Test",

"lastName": "Singh",

"email": "har@gmail.com"

},

{

"id": 6,

"firstName": "Lakesha",

"lastName": "Kumari",

"email": "hello@gmail.com"

},

{

"id": 7,

"firstName": "Kavitha",

"lastName": "Isha",

"email": "haritha@gmail.com"

},

{

"id": 2,

"firstName": "Isha",

"lastName": "Reddy",

"email": "test@gmail.com"

},

{

"id": 1,

"firstName": "Haritha",

"lastName": "Adika",

"email": "haritha@gmail.com"

},

{

"id": 5,

"firstName": "Charan",

"lastName": "Sharma",

"email": "test@gmail.com"

},

{

"id": 3,

"firstName": "Ari",

"lastName": "Aggarwal",

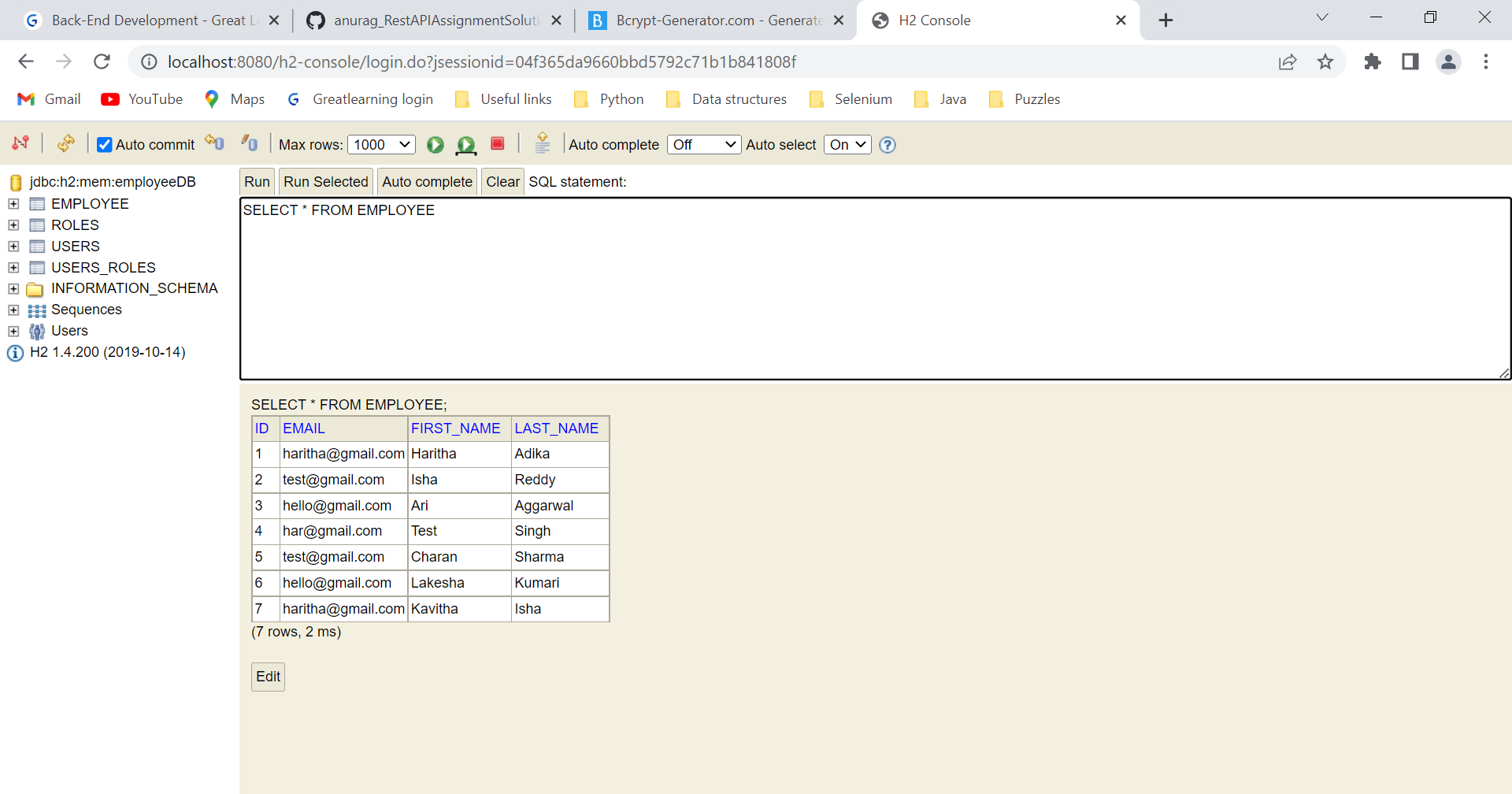
"email": "hello@gmail.com"

}

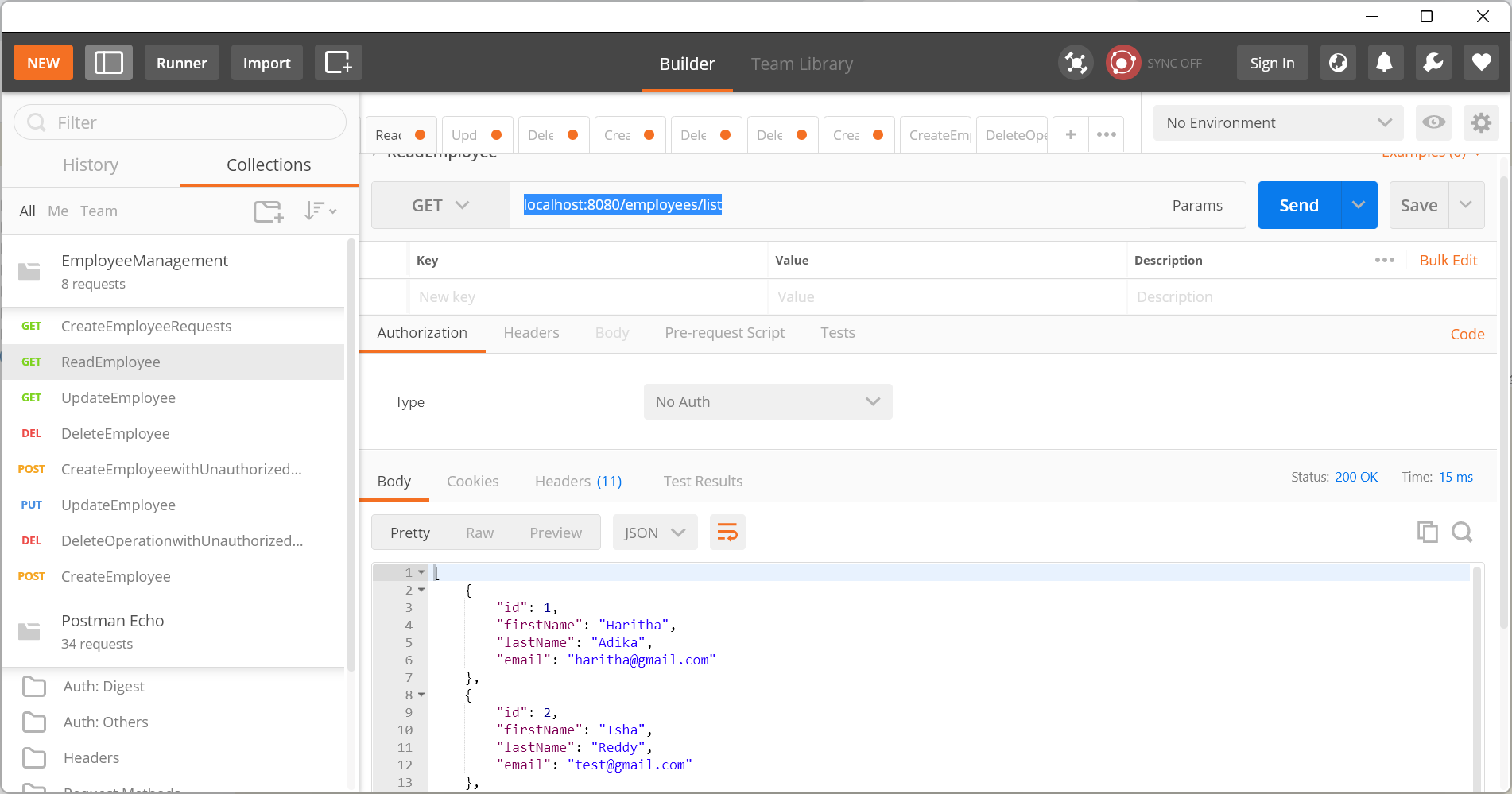
]

1. Your application should provide an endpoint to list all the employees stored in the database

**Data in DB:**



**Endpoint for fetching all the employee records:**

Endpoint : http://localhost:8080/employees/list

**Response Body:**

[

{

"id": 1,

"firstName": "Haritha",

"lastName": "Adika",

"email": "haritha@gmail.com"

},

{

"id": 2,

"firstName": "Isha",

"lastName": "Reddy",

"email": "test@gmail.com"

},

{

"id": 3,

"firstName": "Ari",

"lastName": "Aggarwal",

"email": "hello@gmail.com"

},

{

"id": 4,

"firstName": "Test",

"lastName": "Singh",

"email": "har@gmail.com"

},

{

"id": 5,

"firstName": "Charan",

"lastName": "Sharma",

"email": "test@gmail.com"

},

{

"id": 6,

"firstName": "Lakesha",

"lastName": "Kumari",

"email": "hello@gmail.com"

},

{

"id": 7,

"firstName": "Kavitha",

"lastName": "Isha",

"email": "haritha@gmail.com"

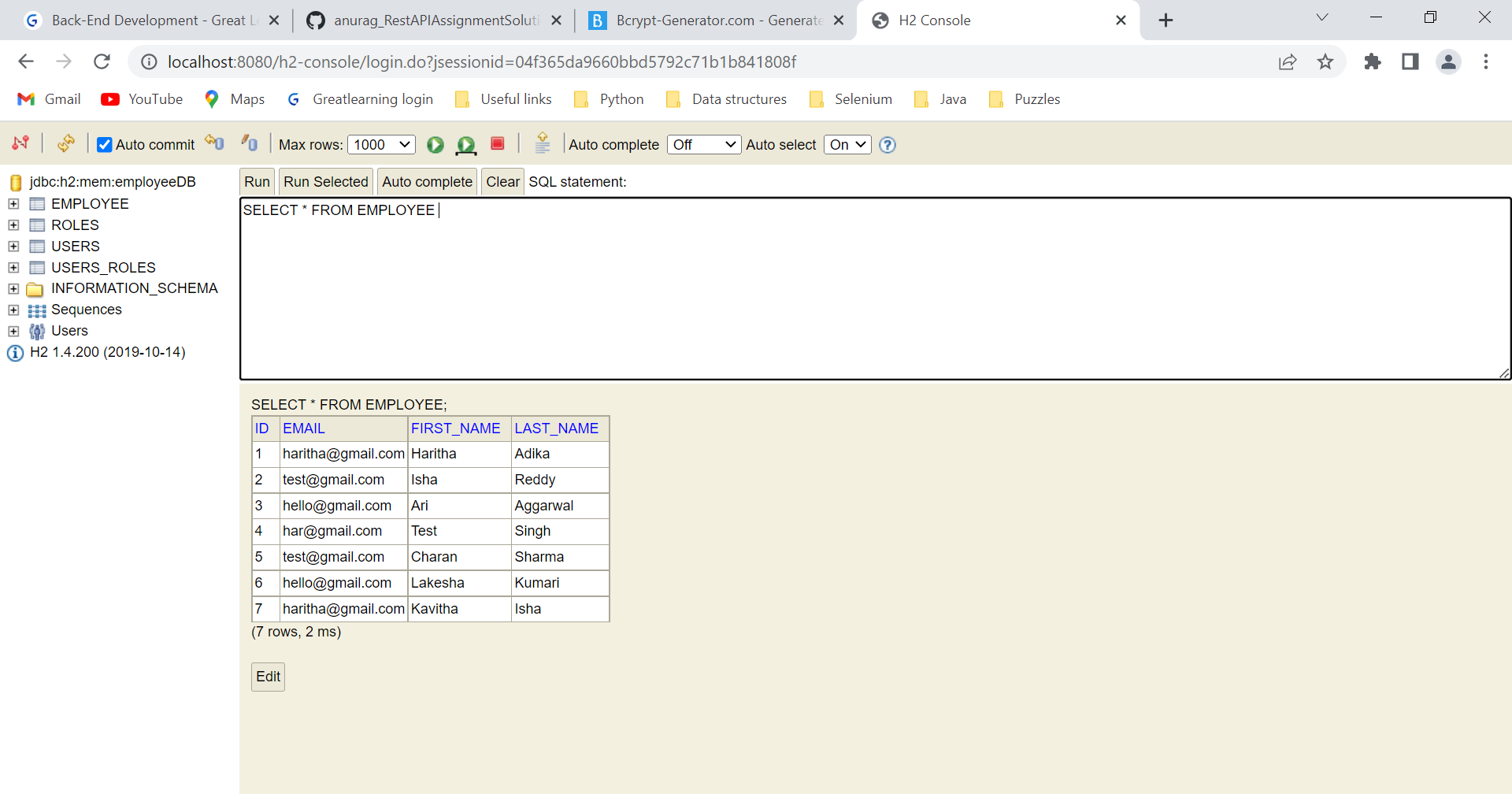
}

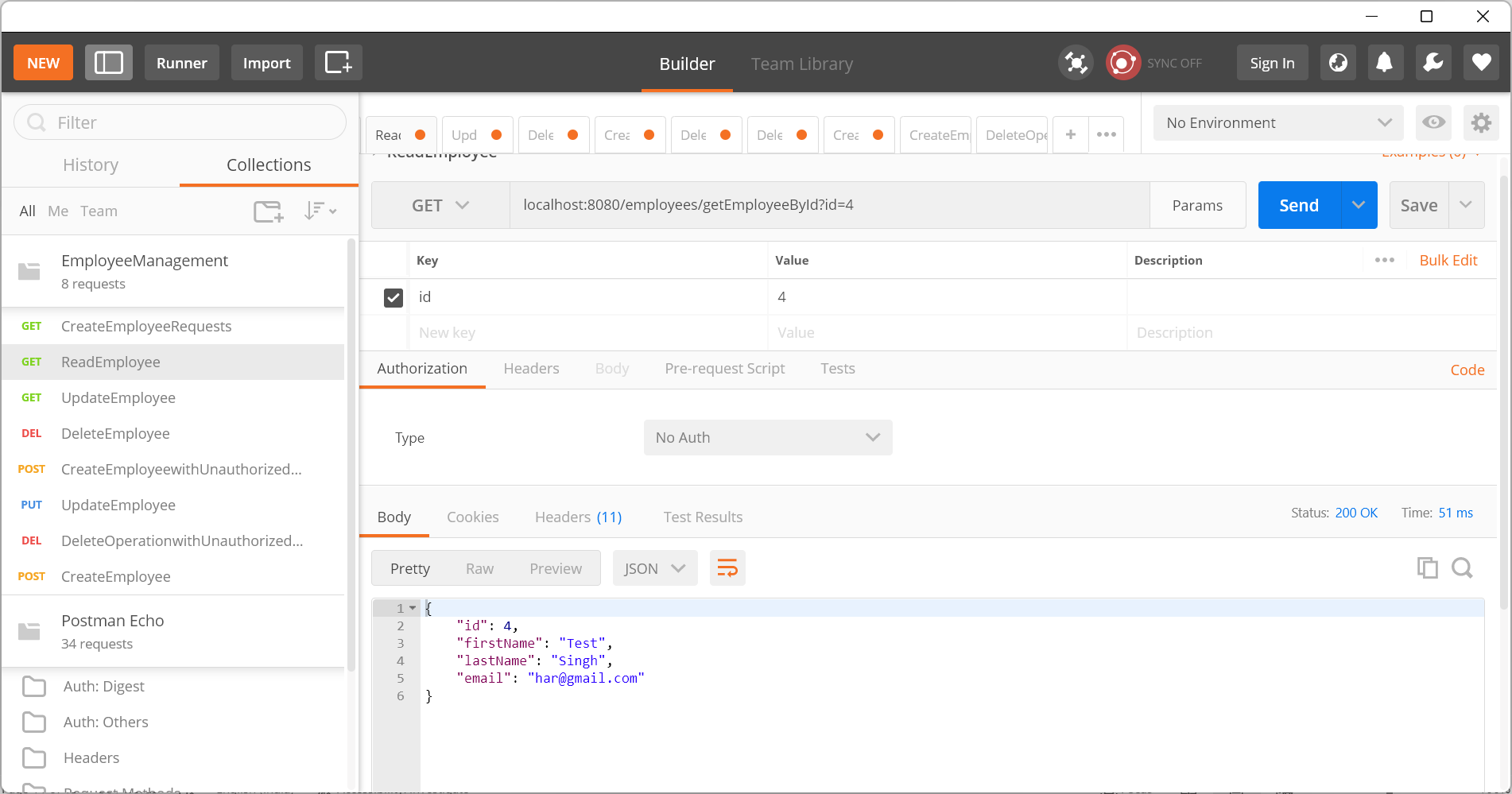
]

1. Endpoint to fetch or get an employee record specifically based on the id of that employee

Endpoint: http://ocalhost:8080/employees/getEmployeeById?id=4

**Data in DB:**





**Response Body:**

{

"id": 4,

"firstName": "Test",

"lastName": "Singh",

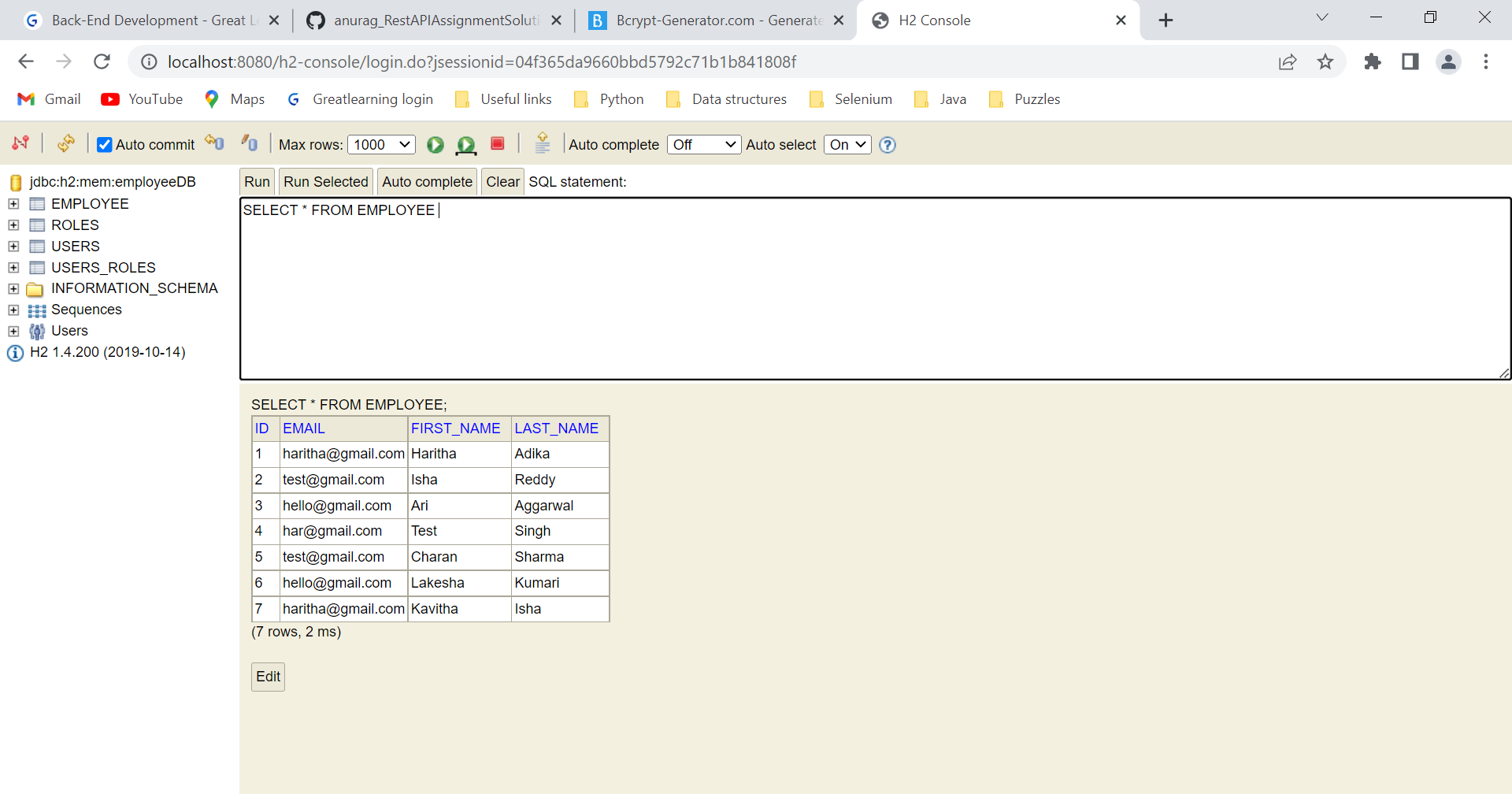
"email": "har@gmail.com"

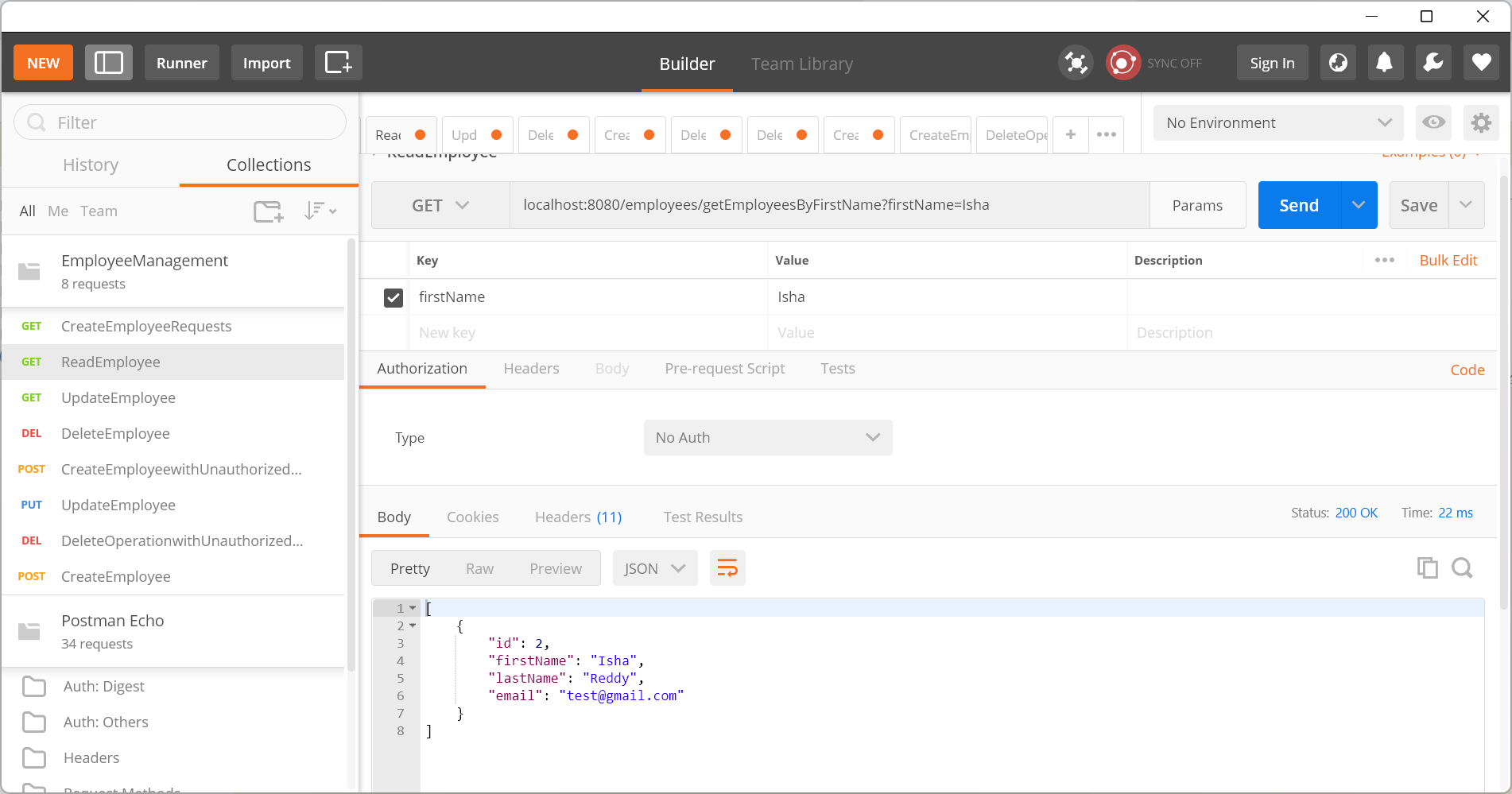
}

1. Endpoint to fetch an employee by his/her first name and if found more than one record then list them all

**Endpoint:** http://ocalhost:8080/employees/getEmployeesByFirstName?firstName=Isha

**Data in DB:**





**Response Body:**

[

{

"id": 2,

"firstName": "Isha",

"lastName": "Reddy",

"email": "test@gmail.com"

}

]

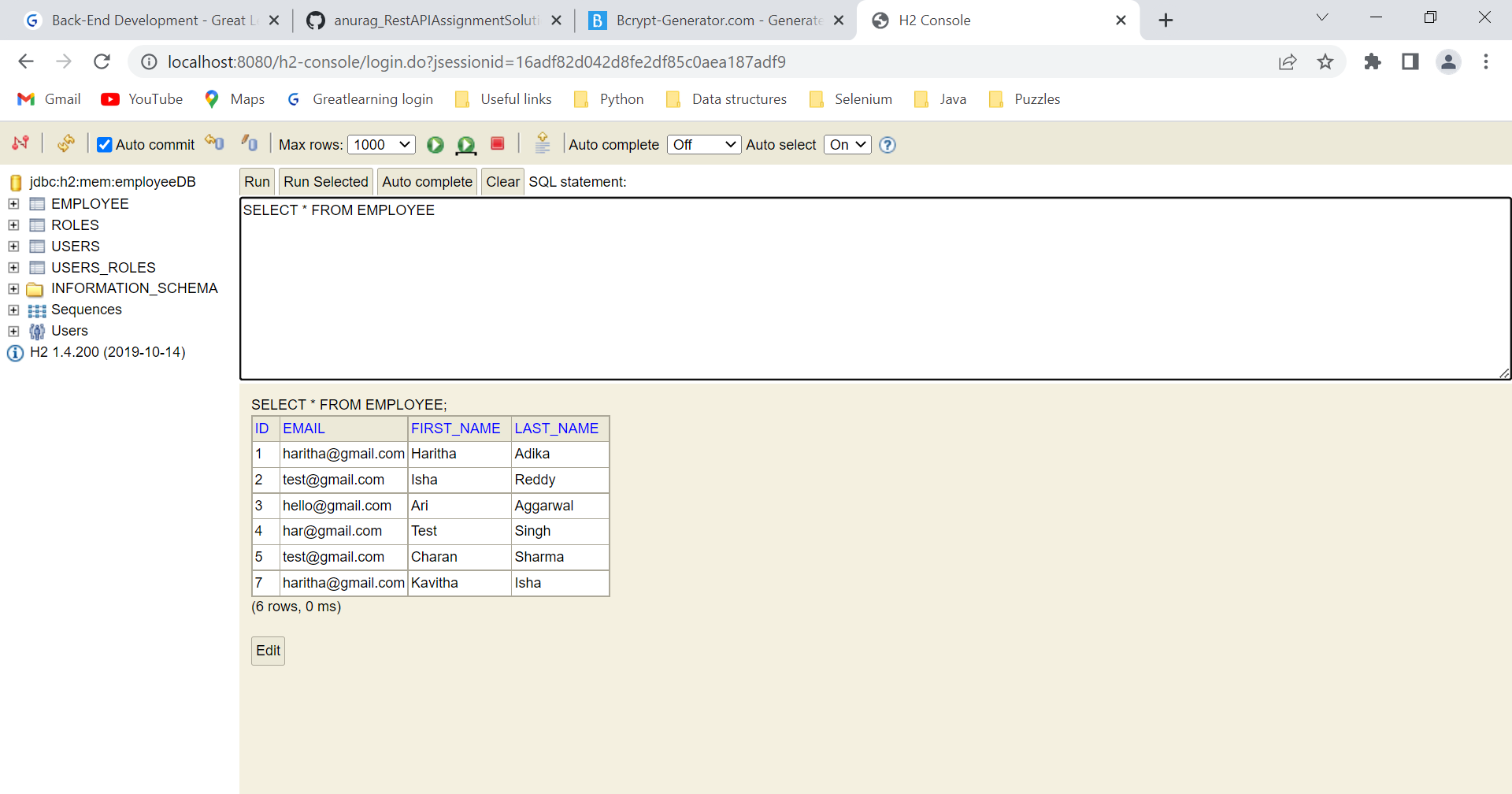
**Delete using Authorized user credentials:**

Username: admin

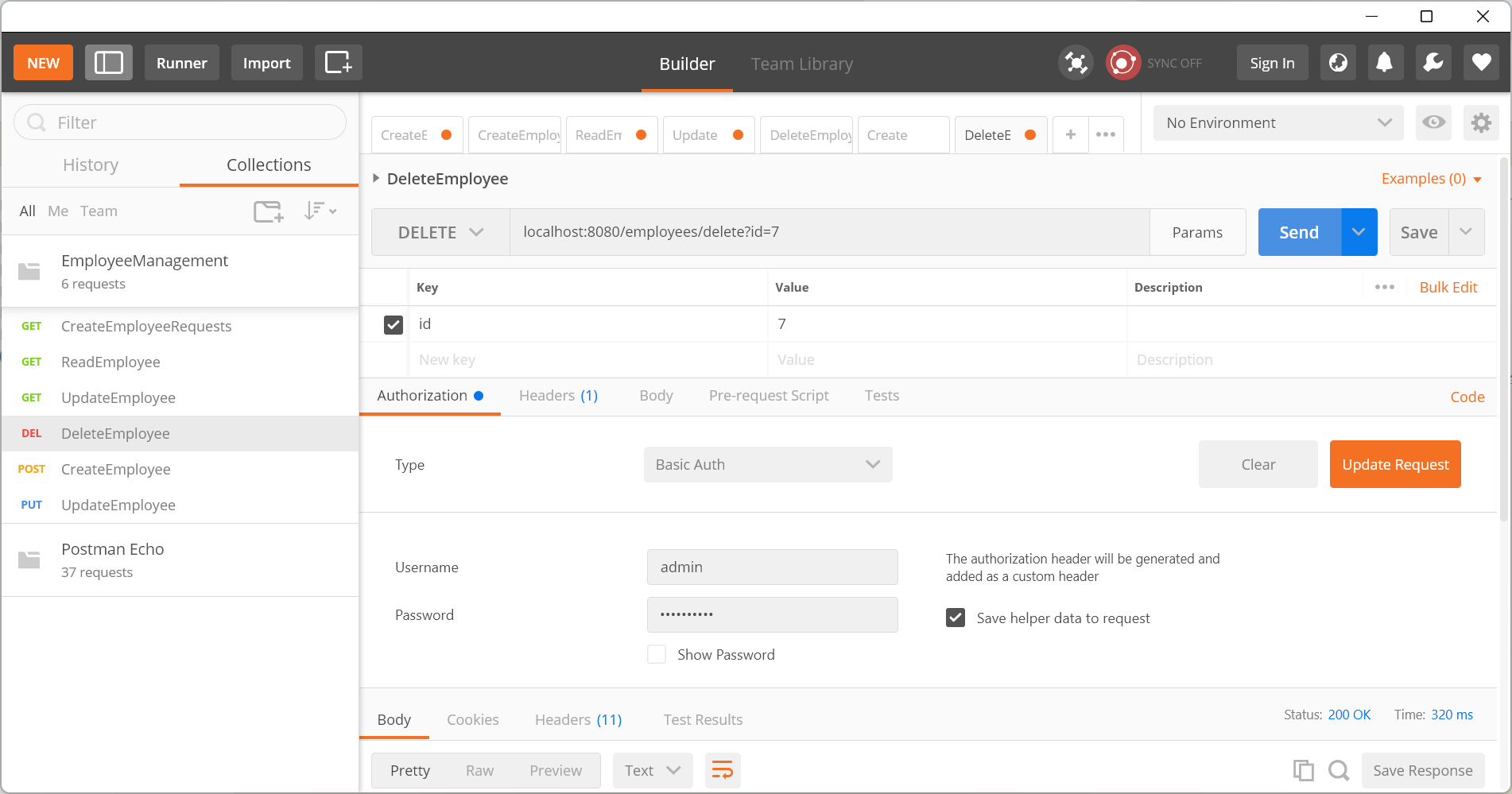
Password: superadmin

**Before Delete:**

**Data in DB before Deleting:**

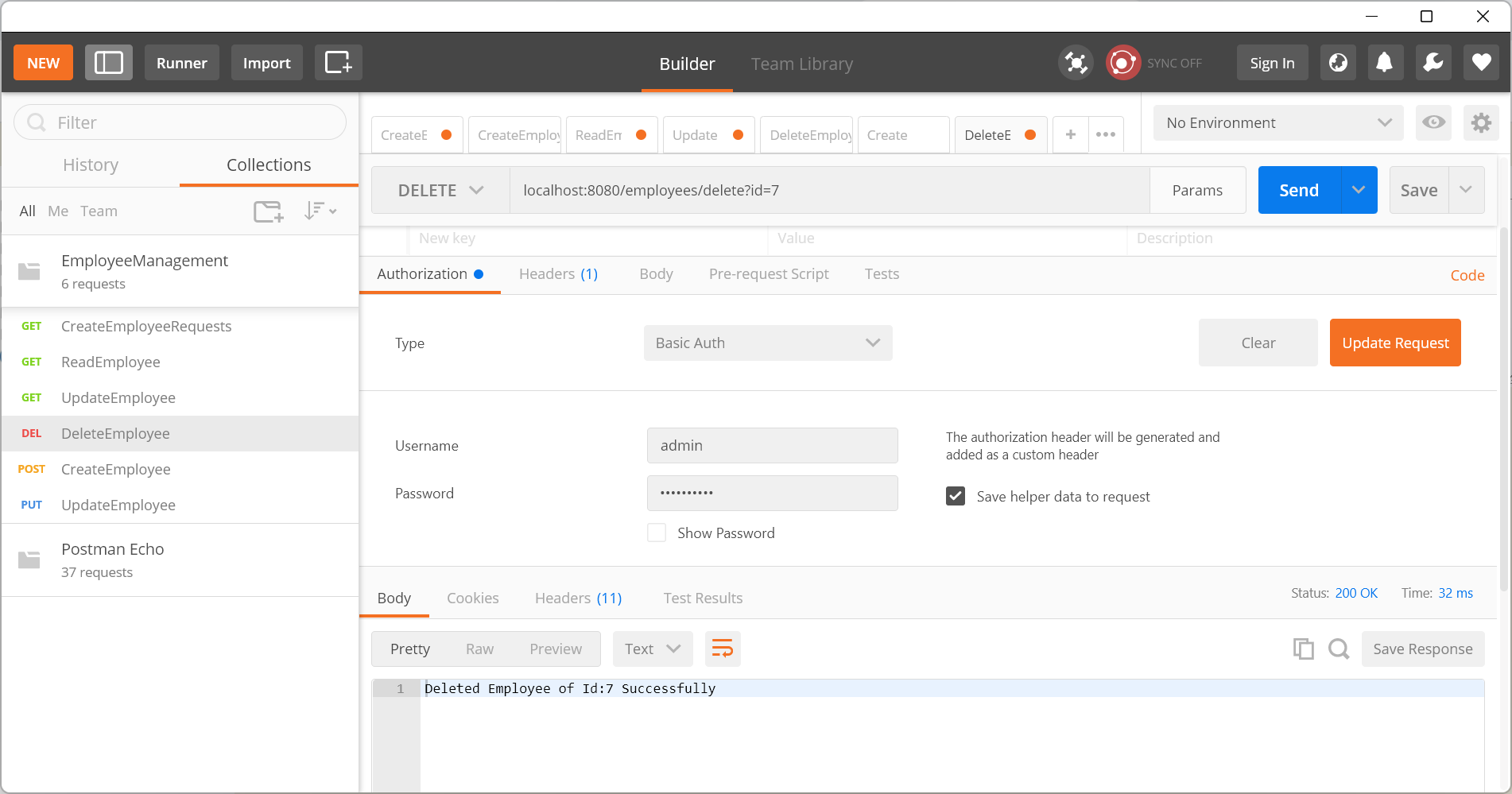


**Postman:**

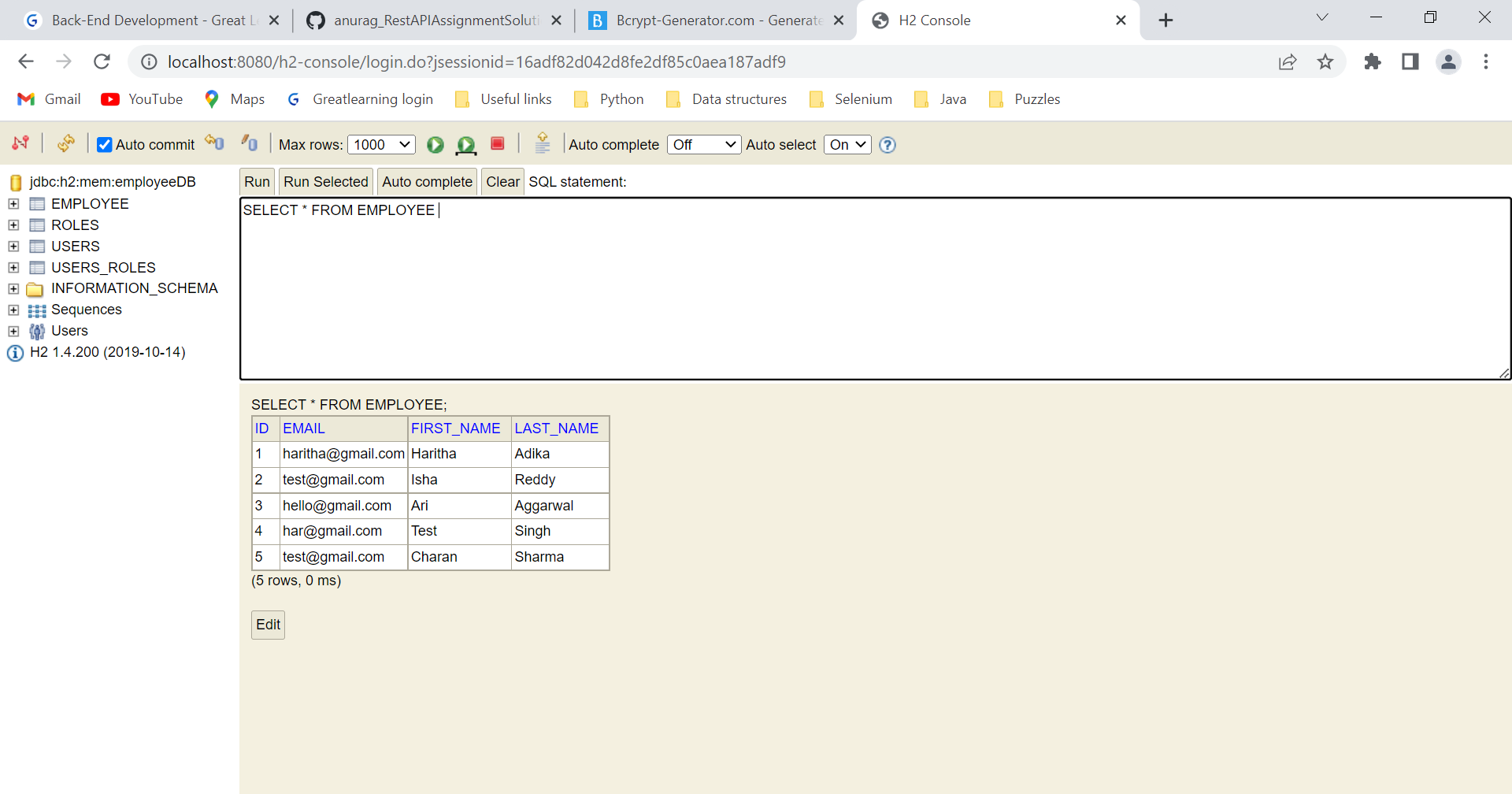


**After Delete:**

**Endpoint :** http://localhost:8080/employees/delete?id=7



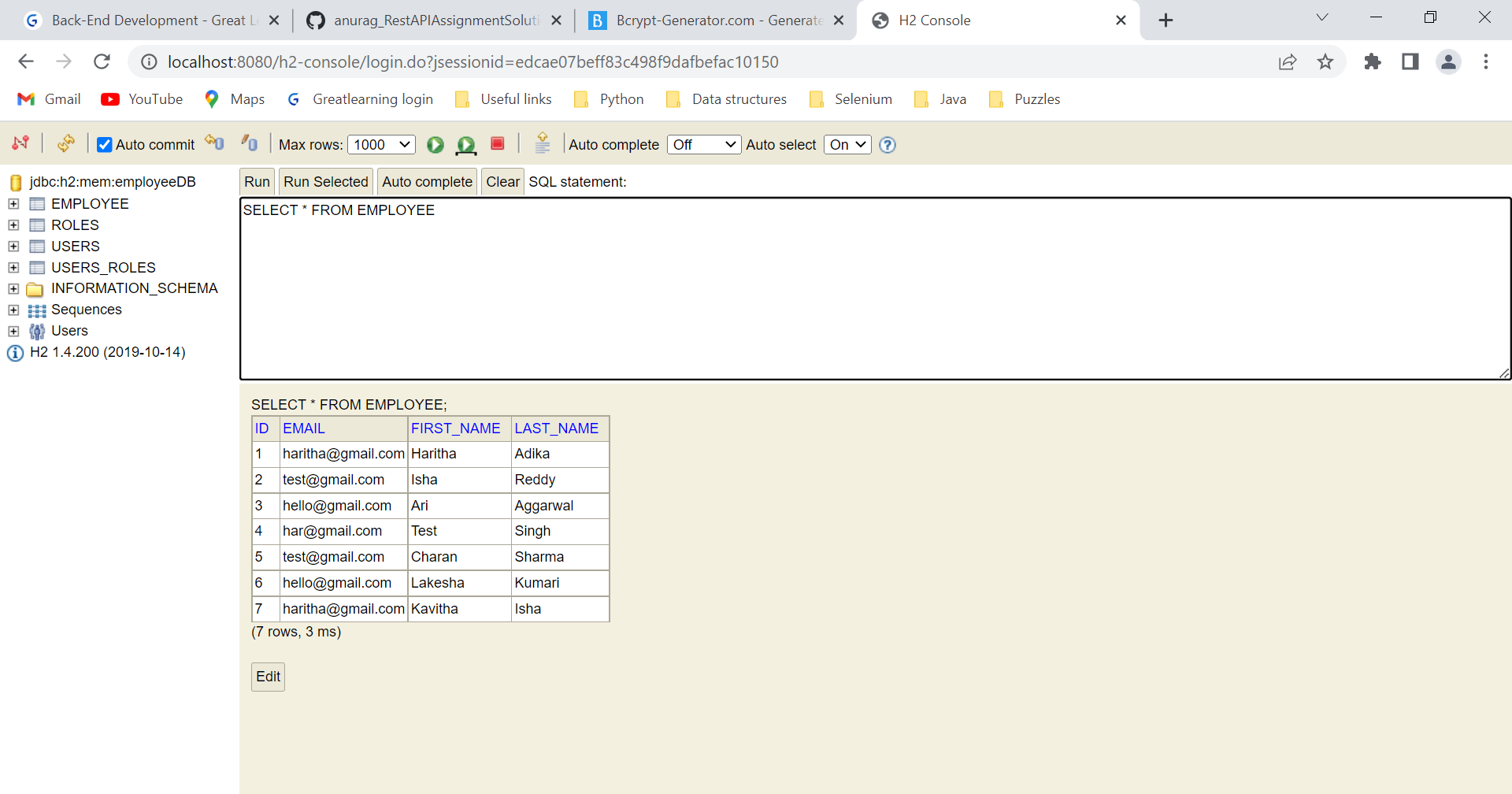
**Data in DB:**



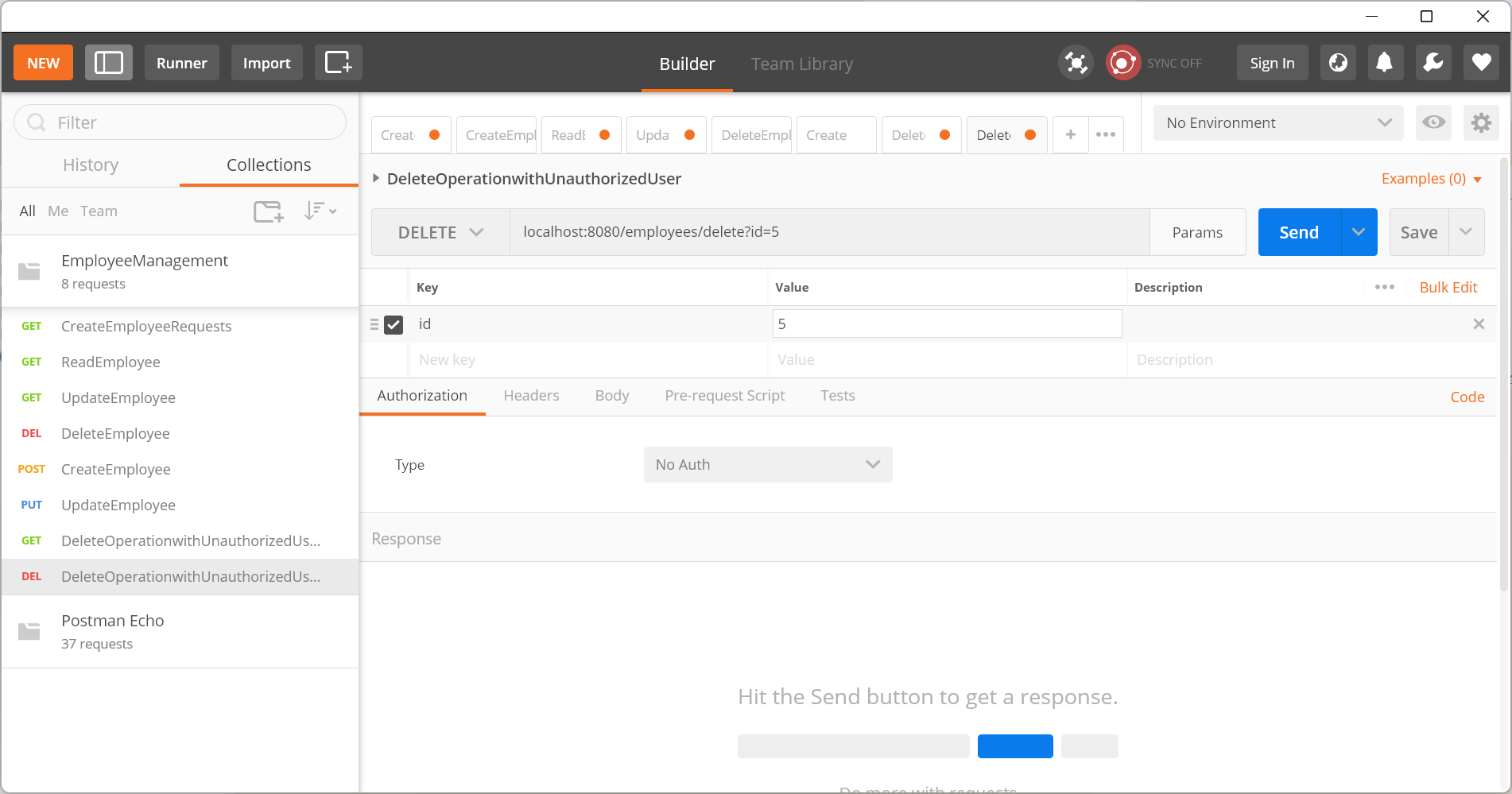
**Delete with UnAuthorized User:**

**Before:**

**Data in DB:**



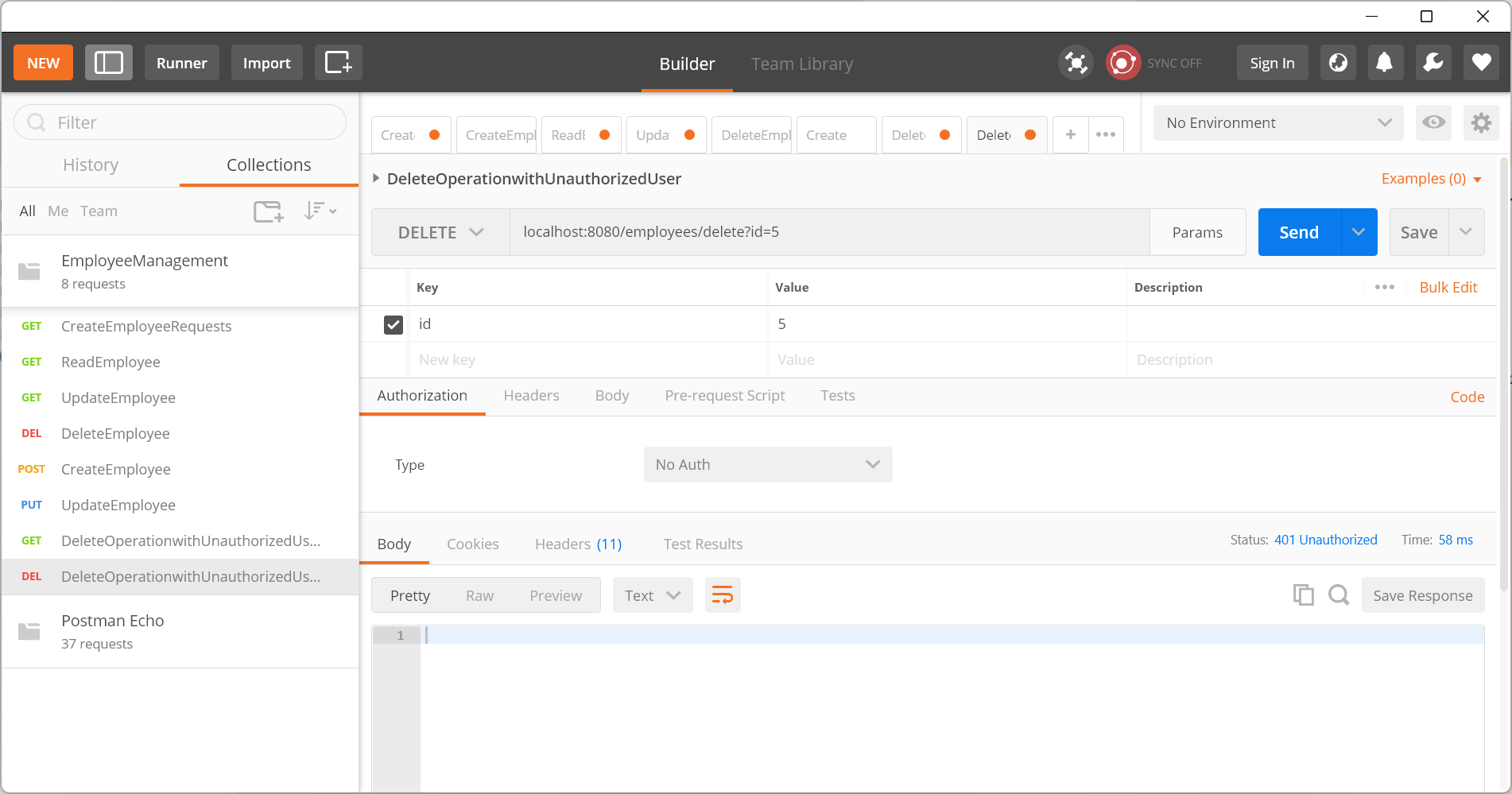
**Postman:**



**After:**

**Postman:**

**Endpoint :** http://localhost:8080/employees/delete?id=5



**Data in Db is unchanged:**

