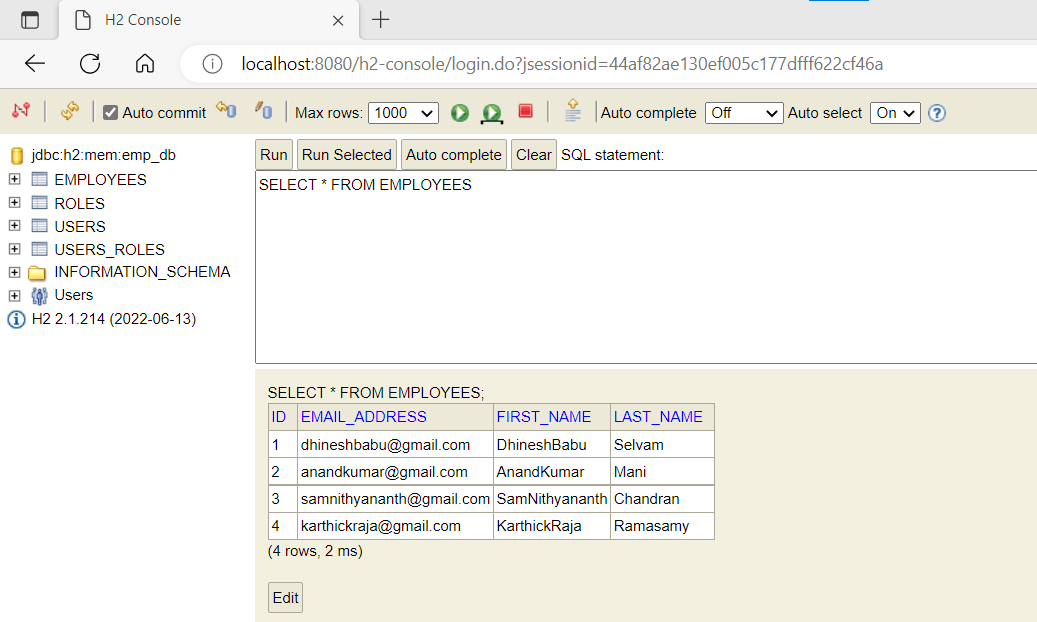
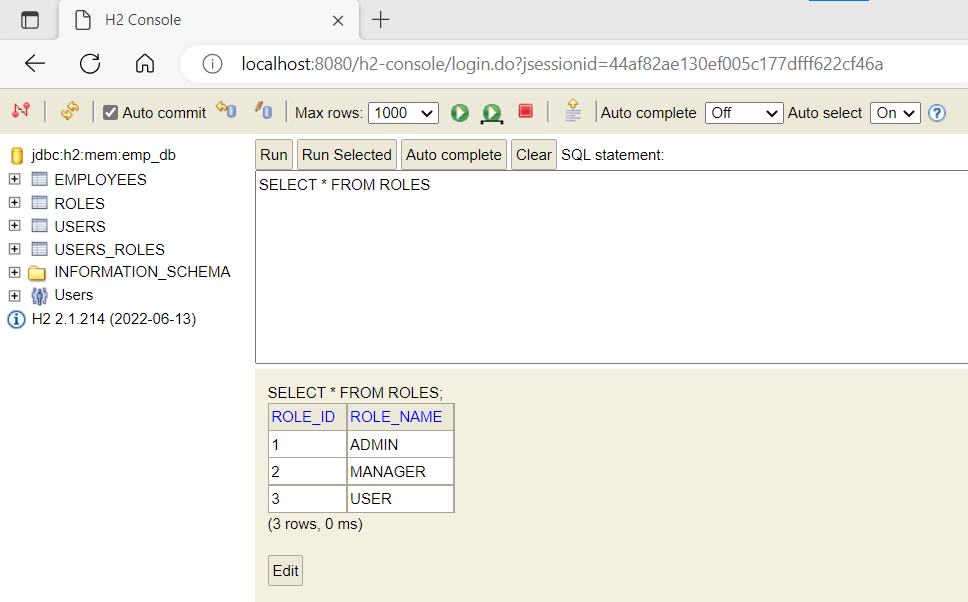
On every start of the application below default data will be loaded to H2 In memory DB.

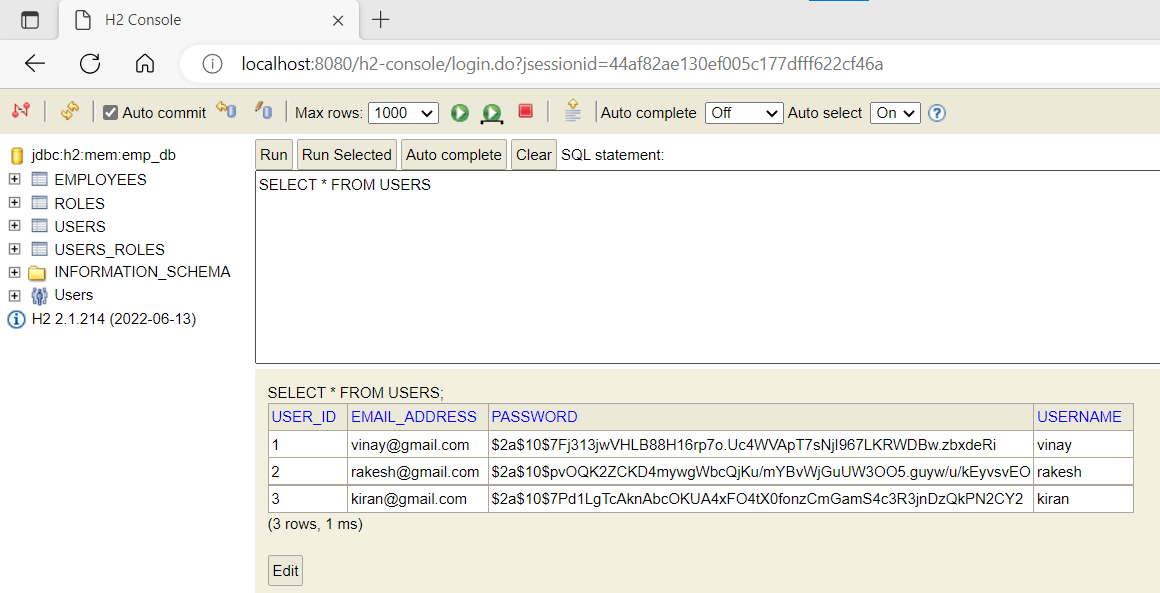
**Employee Table:**



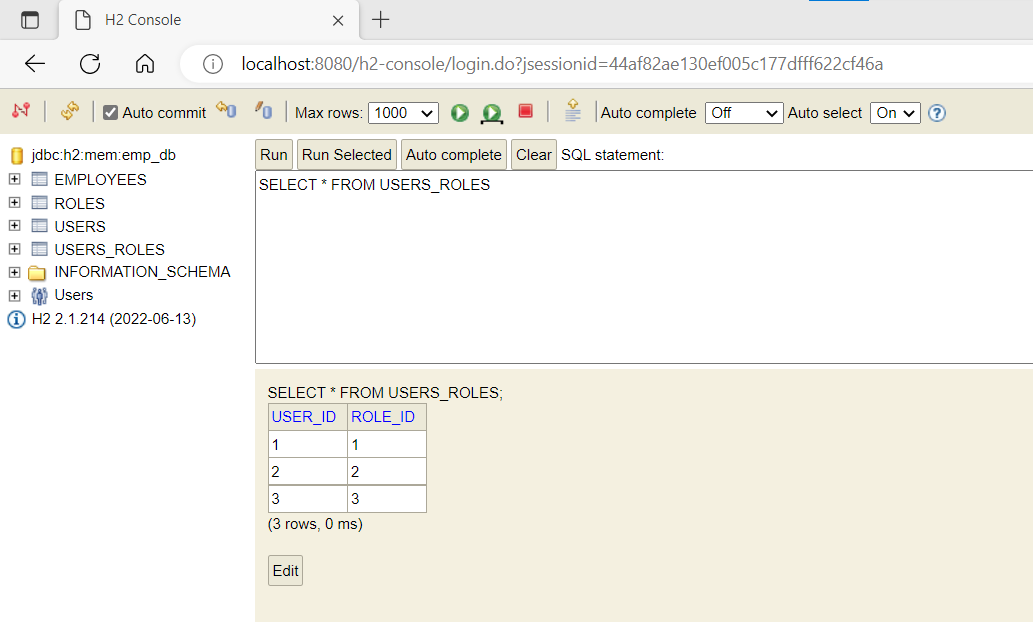
**Roles Table:**



**Users Table:**



**User\_Roles Table:**



**Scenarios to be Tested:**

**Scenario 1:** Your application should be able to add roles in the database dynamically in the db.

**Api Url:** http://localhost:8080/api/roles/

**Http Method:** POST

**Authorization Type:** No Auth

**Request Json:**

{

    "roleName": "ADMIN"

}

**Response Json:**

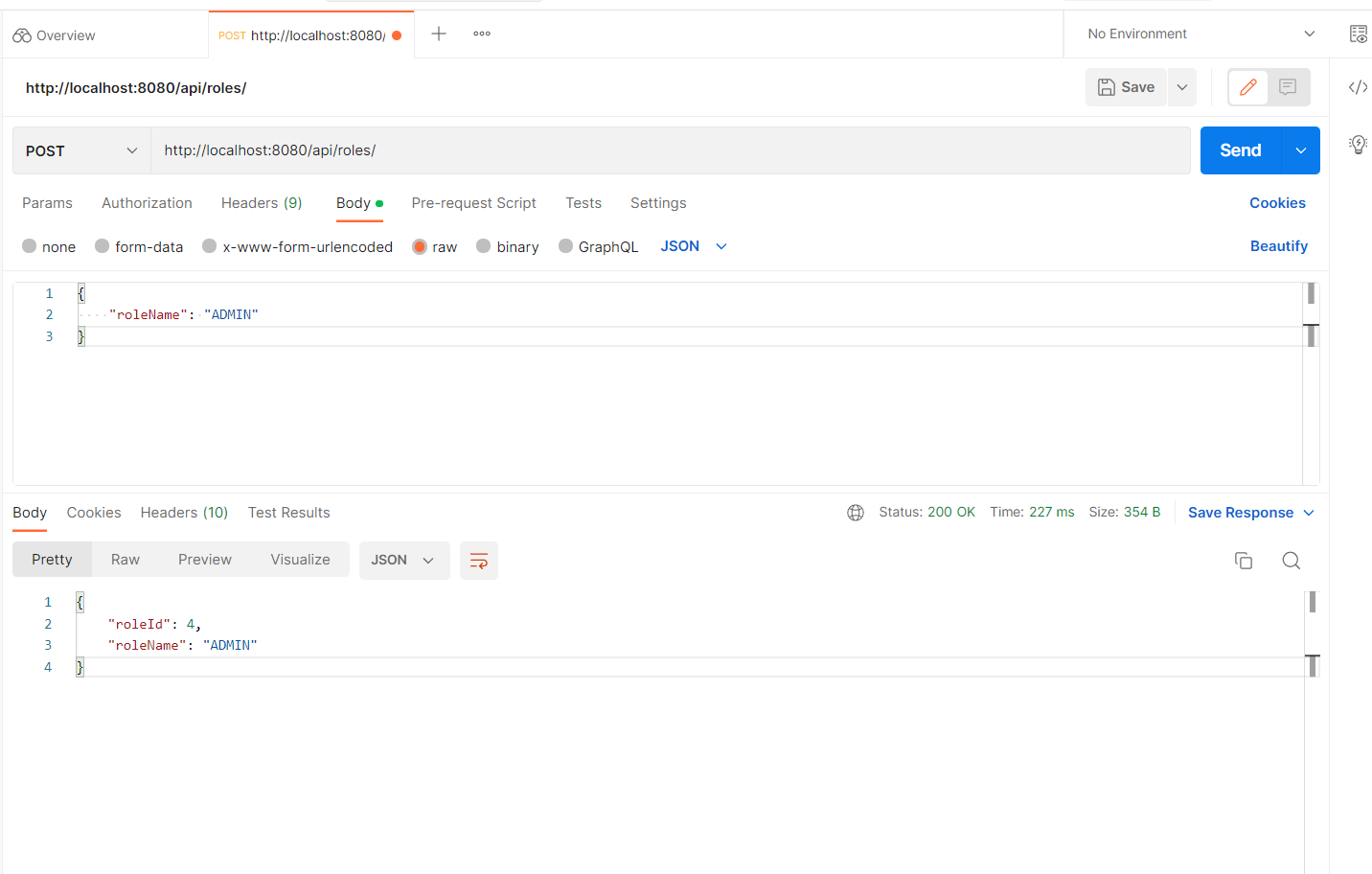
{

    "roleId": 4,

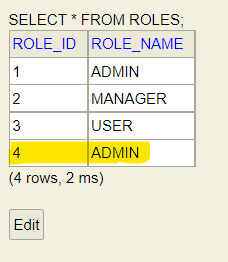
    "roleName": "ADMIN"

}

Postman SS:



H2 DB SS:



**Scenario 2:** Your application should be able to add Users in the db which can be used for authentication purposes.

**Api Url:** http://localhost:8080/api/users/

**Http Method:** POST

**Authorization Type:** No Auth

**Request Json:**

{

    "userId": 4,

    "username": "somu",

    "password": "$2a$10$jhdiHbbnE6ifHZC7w1Z0Ke22GZY.NB2ORgSrPQBXOgxso2..LfUPK",

    "emailAddress": "somu@gmail.com",

    "roles": [

    {

            "roleId": 4,

            "roleName": "ADMIN"

        }

    ]

}

**Response Json:**

{

    "userId": 4,

    "username": "somu",

    "password": "$2a$10$jhdiHbbnE6ifHZC7w1Z0Ke22GZY.NB2ORgSrPQBXOgxso2..LfUPK",

    "emailAddress": "somu@gmail.com",

    "roles": [

        {

            "roleId": 4,

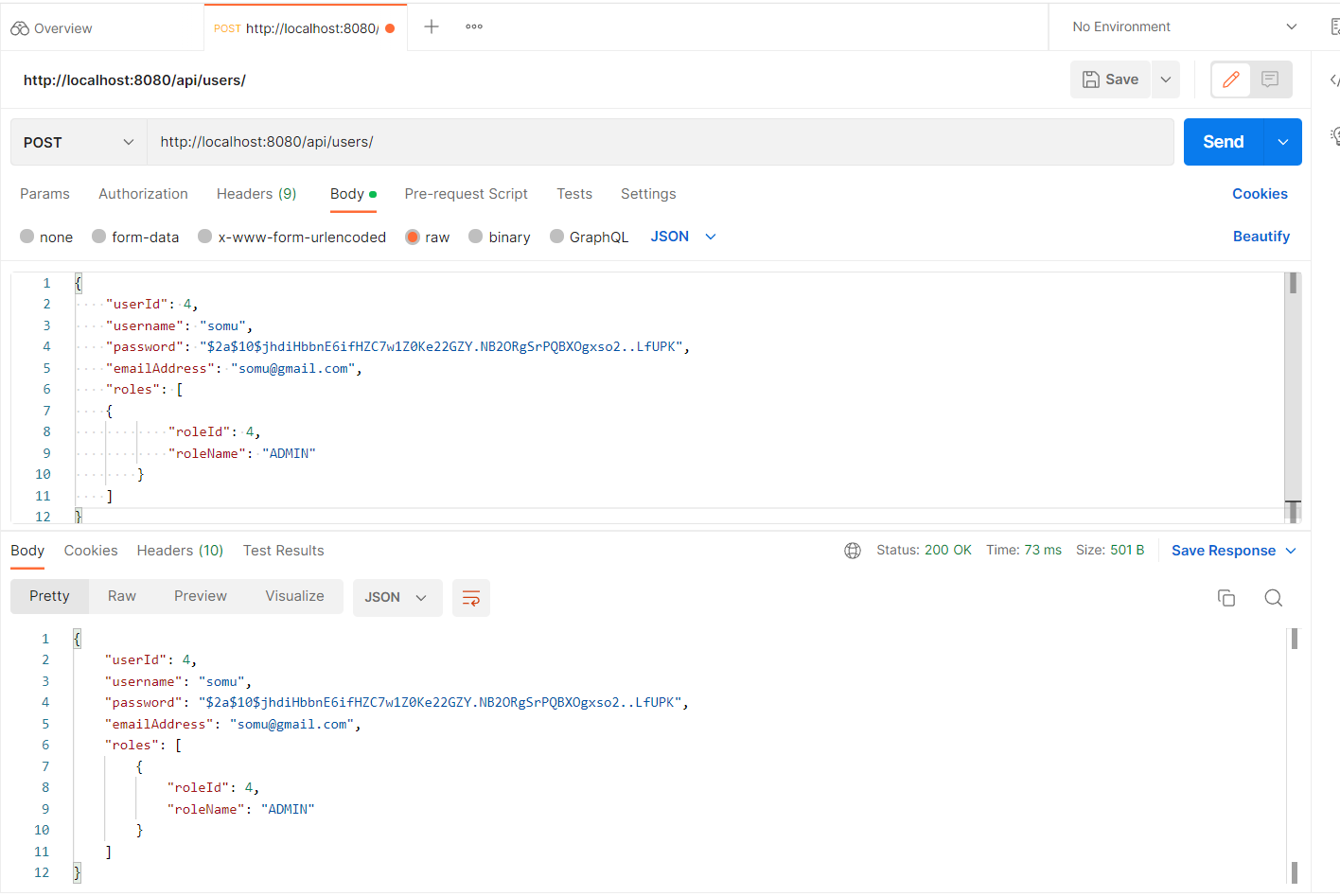
            "roleName": "ADMIN"

        }

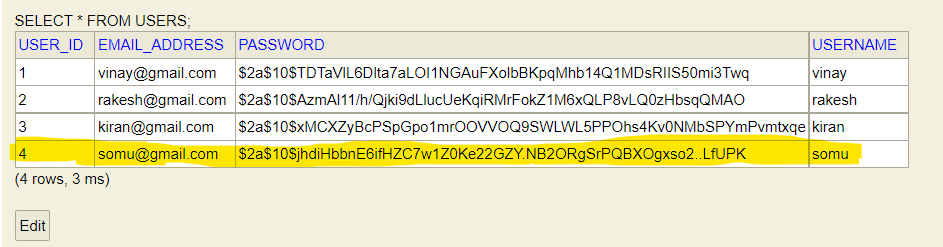
    ]

}

Postman SS:



H2 DB SS:



**Scenario 3:** Now Your application should be able to add employees data in the db if and only if the authenticated user is **ADMIN**.

**3a Negative Scenario:**

**Api Url:** http://localhost:8080/api/employees/

**Http Method:** POST

**Authorization Type:** No Auth

**Request Json:**

{

    "firstName": "Prabhu",

    "lastName": "Chinnasamy",

    "emailAddress": "prabhu\_tc@yahoo.com"

}

**Response Json:**

{

    "timestamp": "2022-10-25T14:28:39.653+00:00",

    "status": 401,

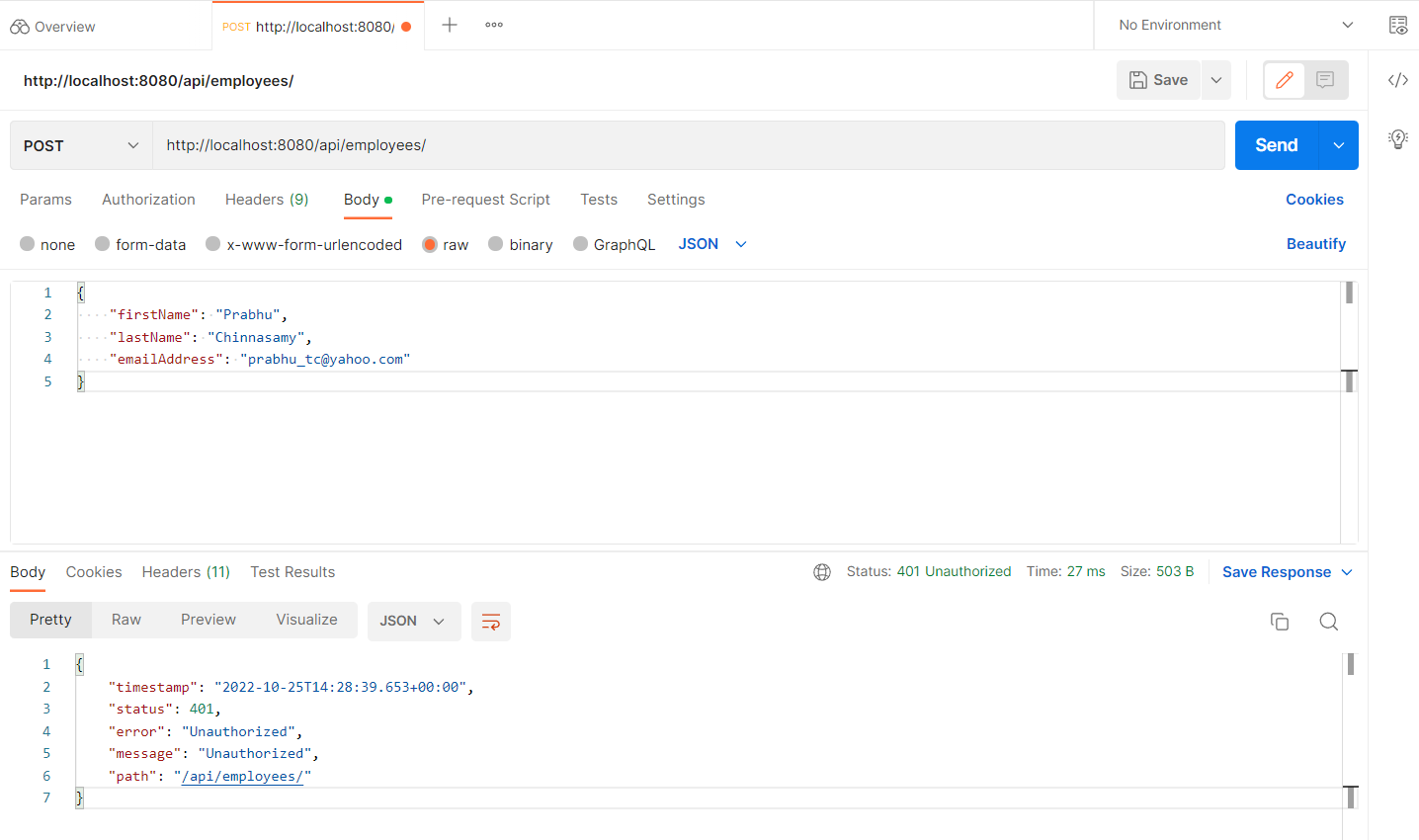
    "error": "Unauthorized",

    "message": "Unauthorized",

    "path": "/api/employees/"

}

Postman SS:



**3b Positive Scenario:**

**Api Url:** http://localhost:8080/api/employees/

**Http Method:** POST

**Authorization Type:** Basic Auth

**username:** somu

**password:** welcome

**Request Json:**

{

    "firstName": "Prabhu",

    "lastName": "Chinnasamy",

    "emailAddress": "prabhu\_tc@yahoo.com"

}

**Response Json:**

{

    "id": 5,

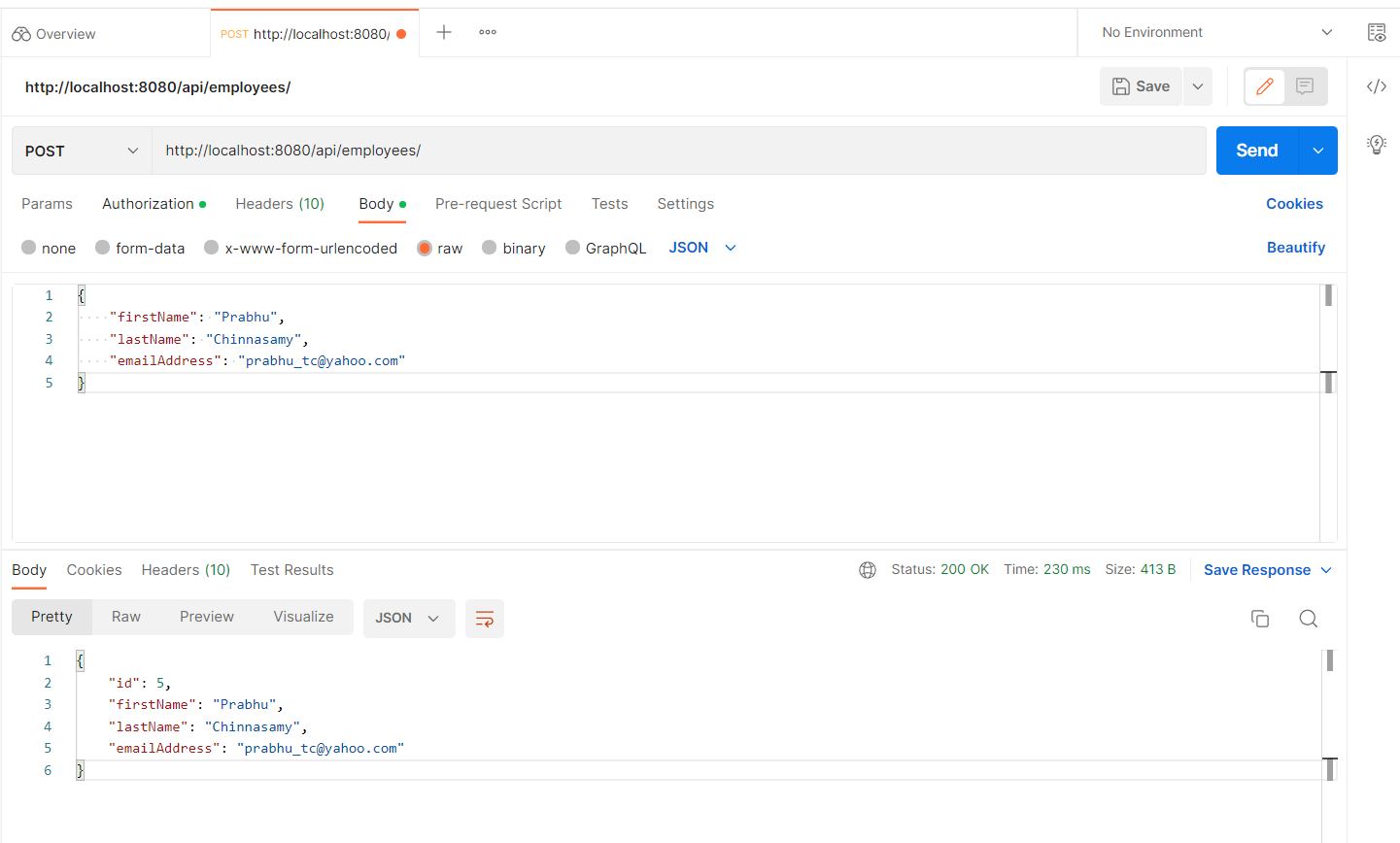
    "firstName": "Prabhu",

    "lastName": "Chinnasamy",

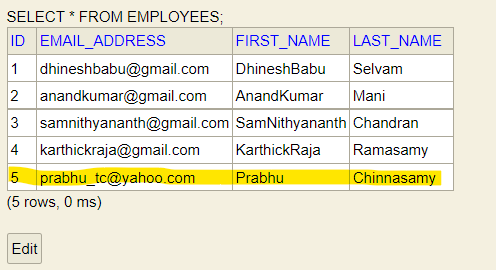
    "emailAddress": "prabhu\_tc@yahoo.com"

}

Postman SS:



H2 DB SS:



**Scenario 4:** Your application should provide an endpoint to list all the employees stored in the database.

**Api Url:** http://localhost:8080/api/employees/

**Http Method:** GET

**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

[

    {

        "id": 1,

        "firstName": "DhineshBabu",

        "lastName": "Selvam",

        "emailAddress": "dhineshbabu@gmail.com"

    },

    {

        "id": 2,

        "firstName": "AnandKumar",

        "lastName": "Mani",

        "emailAddress": "anandkumar@gmail.com"

    },

    {

        "id": 5,

        "firstName": "Prabhu",

        "lastName": "Chinnasamy",

        "emailAddress": "prabhu\_tc@yahoo.com"

    },

    {

        "id": 3,

        "firstName": "SamNithyananth",

        "lastName": "Chandran",

        "emailAddress": "samnithyananth@gmail.com"

    },

    {

        "id": 4,

        "firstName": "KarthickRaja",

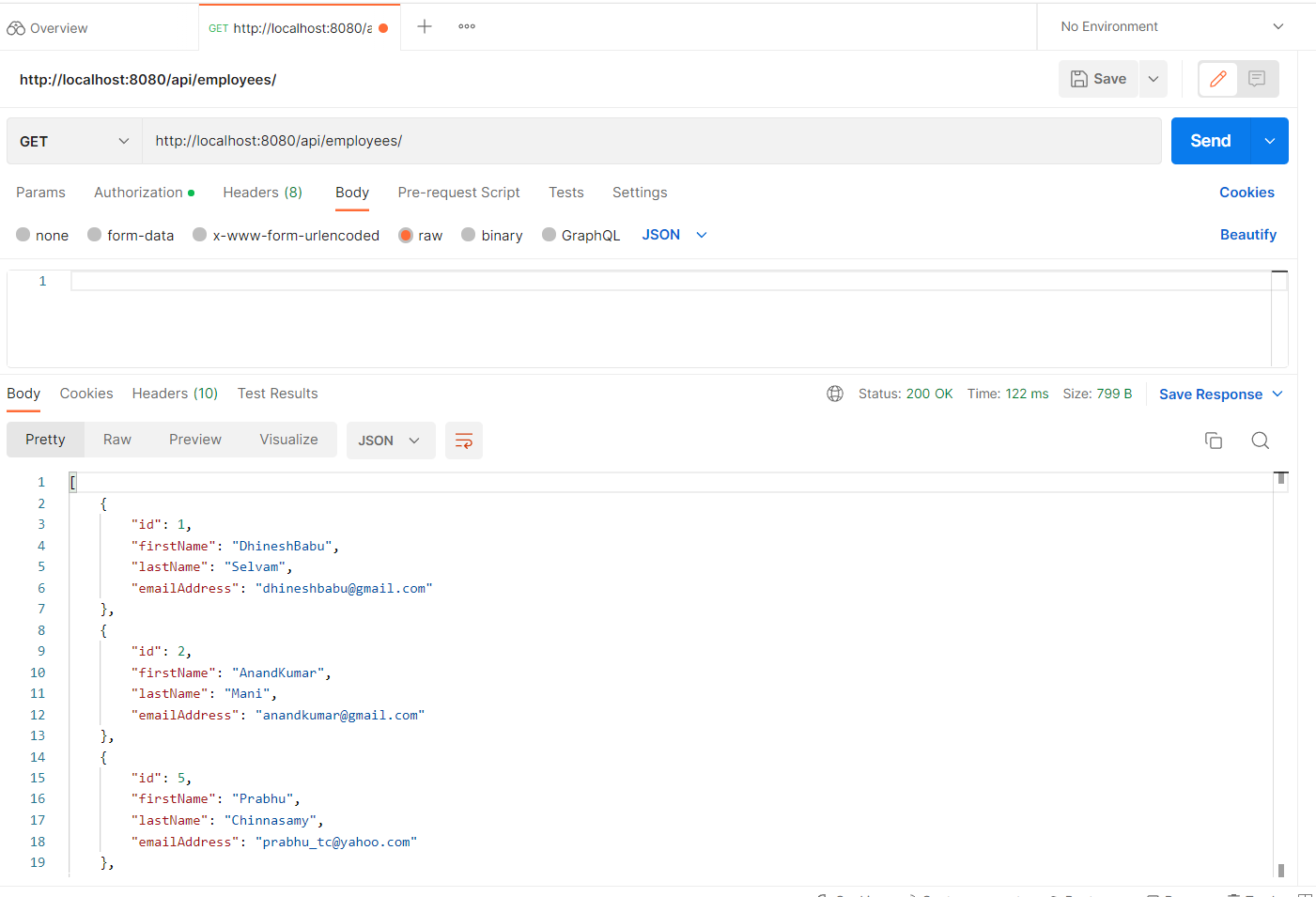
        "lastName": "Ramasamy",

        "emailAddress": "karthickraja@gmail.com"

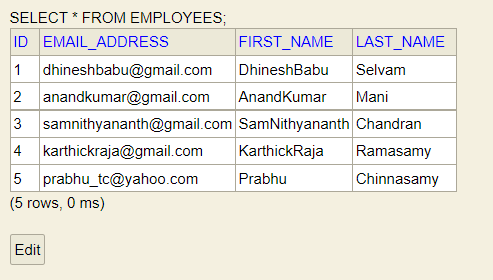
    }

]

Postman SS:



H2 DB SS:



**Scenario 5:** Your application should provide endpoint to fetch or get an employee record specifically based on the id of that employee.

**Api Url:** http://localhost:8080/api/employees/3

**Http Method:** GET

**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

{

    "id": 3,

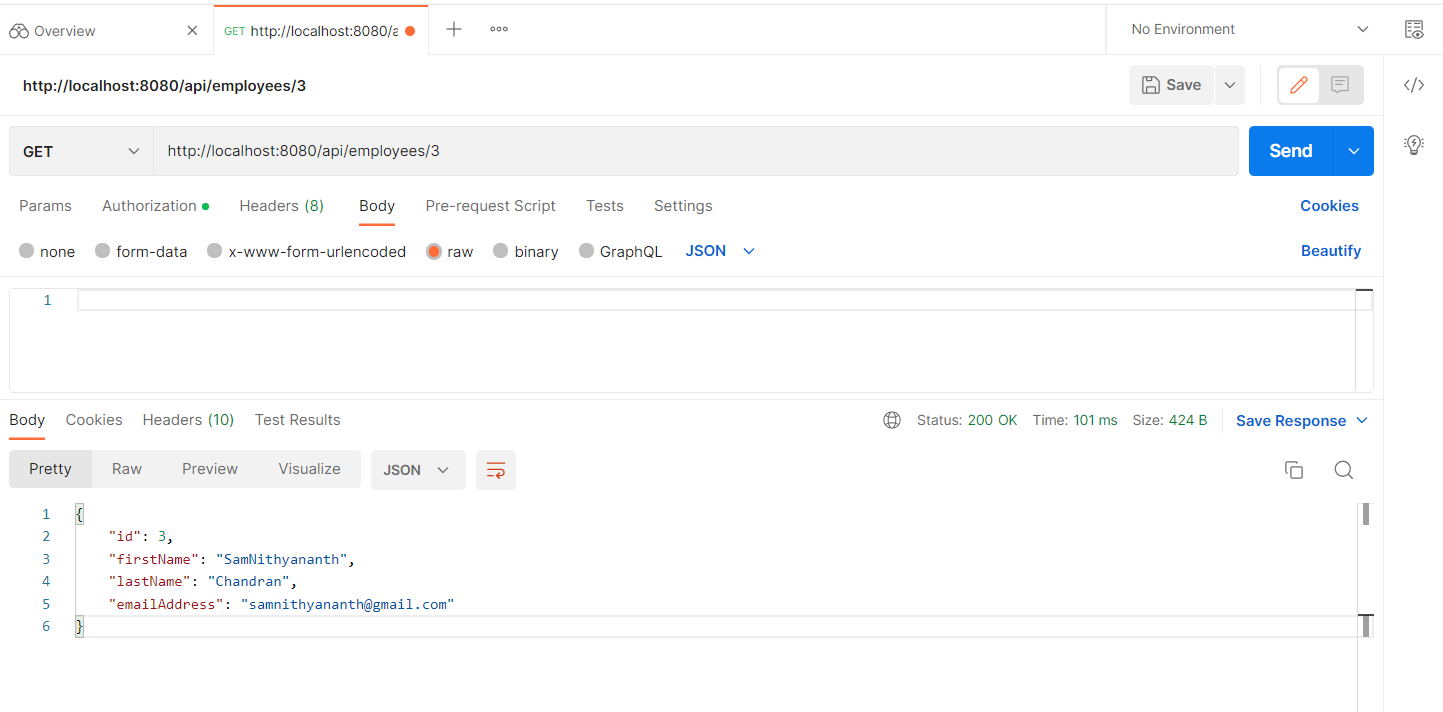
    "firstName": "SamNithyananth",

    "lastName": "Chandran",

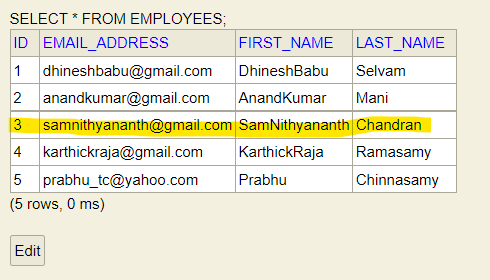
    "emailAddress": "samnithyananth@gmail.com"

}

Postman SS:



H2 DB SS:



**Scenario 6:** Your application should provide an endpoint to update an existing employee record with the given updated json object.

**Api Url:** http://localhost:8080/api/employees/

**Http Method:** POST

**Authorization Type:** Basic Auth

**username:** somu

**password:** welcome

**Request Json:**

{

    "id": 5,

    "firstName": "Ravi",

    "lastName": "Kumar",

    "emailAddress": "ravikumar@yahoo.com"

}

**Response Json:**

{

    "id": 5,

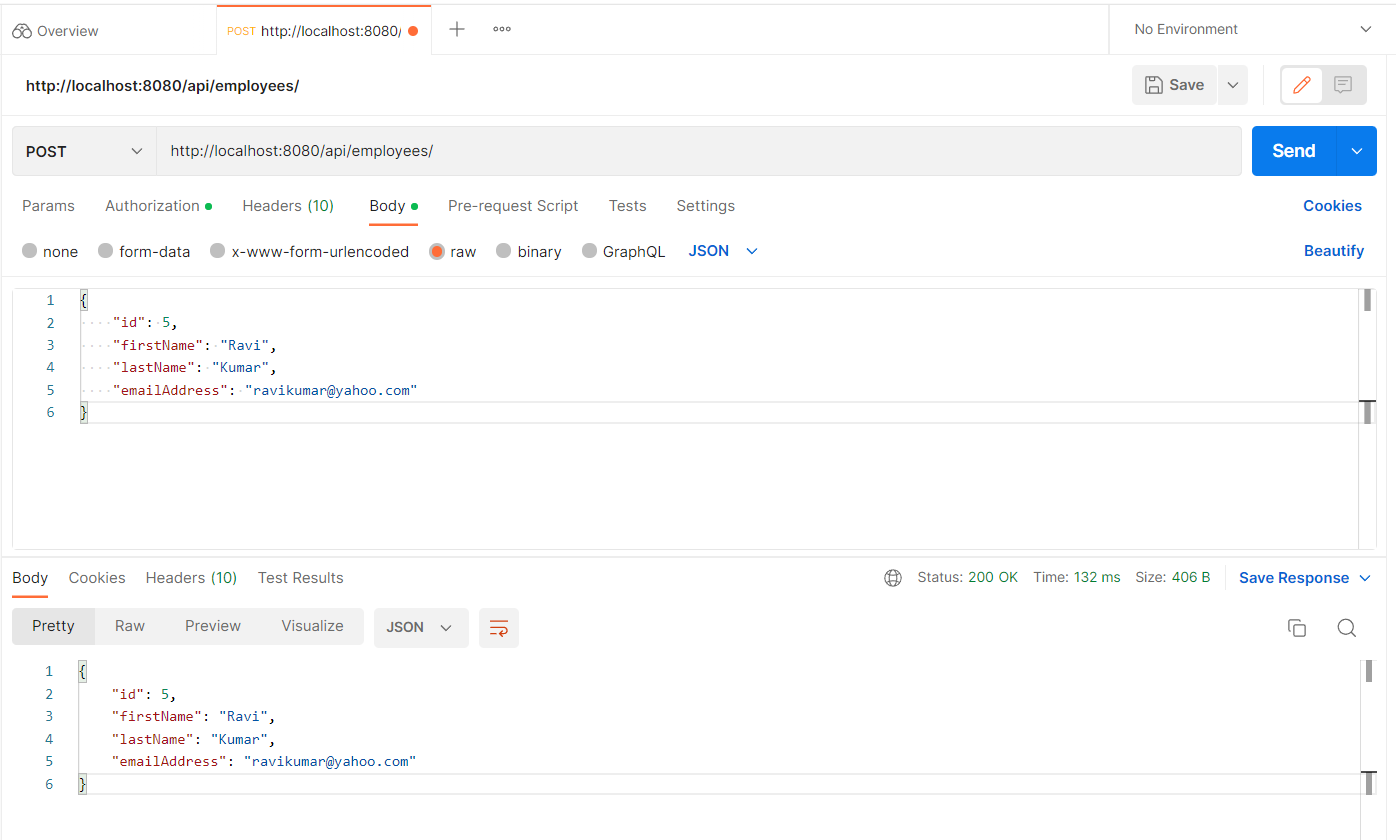
    "firstName": "Ravi",

    "lastName": "Kumar",

    "emailAddress": "ravikumar@yahoo.com"

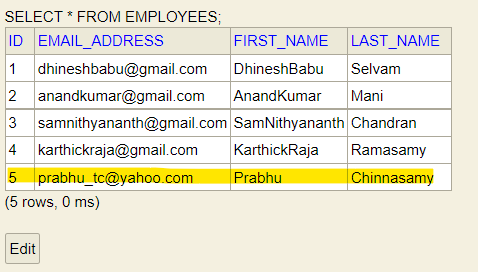
}

Postman SS:

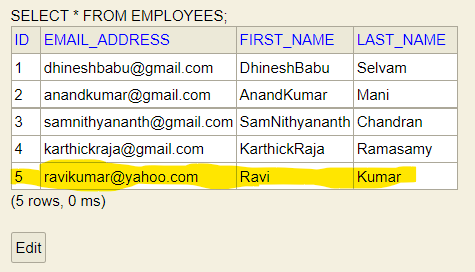


H2 DB SS:

(Before Update):



After Update:



**Scenario 7:** Your application should also provide an endpoint to delete an existing employee record based on the id of the employee.

**Api Url:** http://localhost:8080/api/employees/5

**Http Method:** DELETE

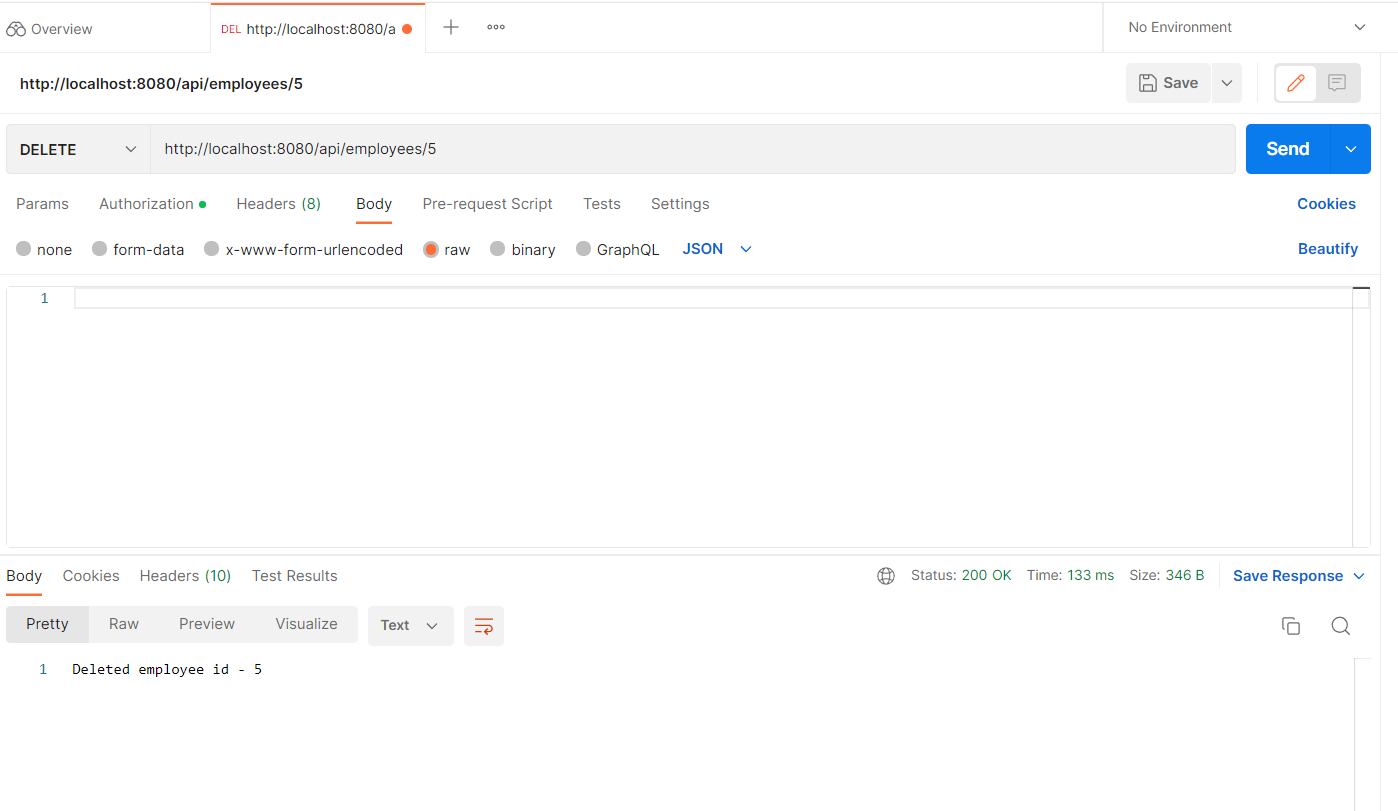
**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

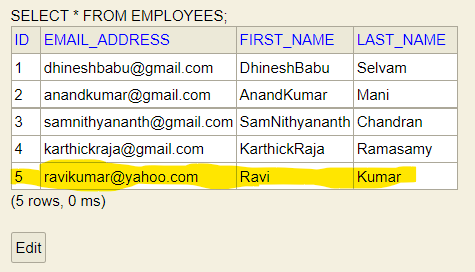
Deleted employee id - 5

Postman SS:

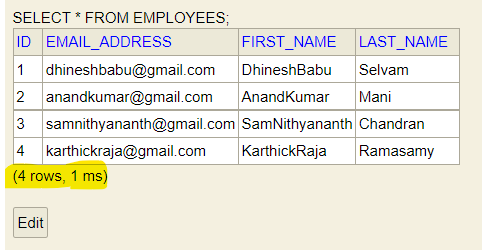


H2 DB SS:

(Before Update):



After Update:



**Scenario 8:** Your application should provide an endpoint to fetch an employee by his/her first name and if found more than one record then list them all.

**Api Url:** http://localhost:8080/api/employees/search/AnandKumar

**Http Method:** GET

**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

[

    {

        "id": 2,

        "firstName": "AnandKumar",

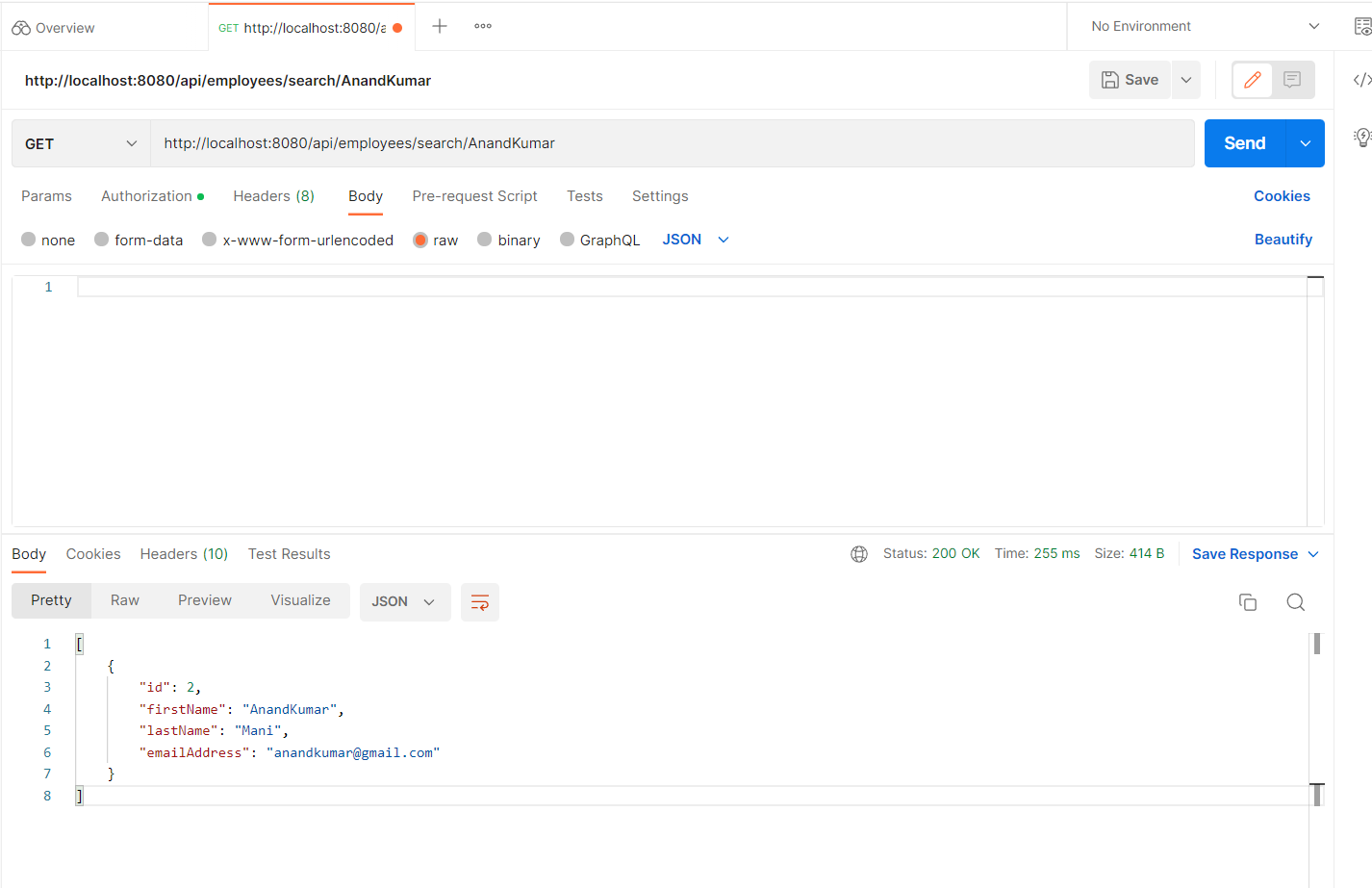
        "lastName": "Mani",

        "emailAddress": "anandkumar@gmail.com"

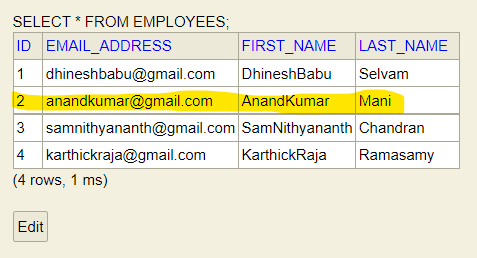
    }

]

Postman SS:



H2 DB SS:



**Scenario 9:** Your application should be able to list all employee records sorted on their first name in either ascending order or descending order.

**9a Ascending Order:**

**Api Url:** http://localhost:8080/api/employees/sort?order="asc"

**Http Method:** GET

**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

[

    {

        "id": 2,

        "firstName": "AnandKumar",

        "lastName": "Mani",

        "emailAddress": "anandkumar@gmail.com"

    },

    {

        "id": 1,

        "firstName": "DhineshBabu",

        "lastName": "Selvam",

        "emailAddress": "dhineshbabu@gmail.com"

    },

    {

        "id": 4,

        "firstName": "KarthickRaja",

        "lastName": "Ramasamy",

        "emailAddress": "karthickraja@gmail.com"

    },

    {

        "id": 3,

        "firstName": "SamNithyananth",

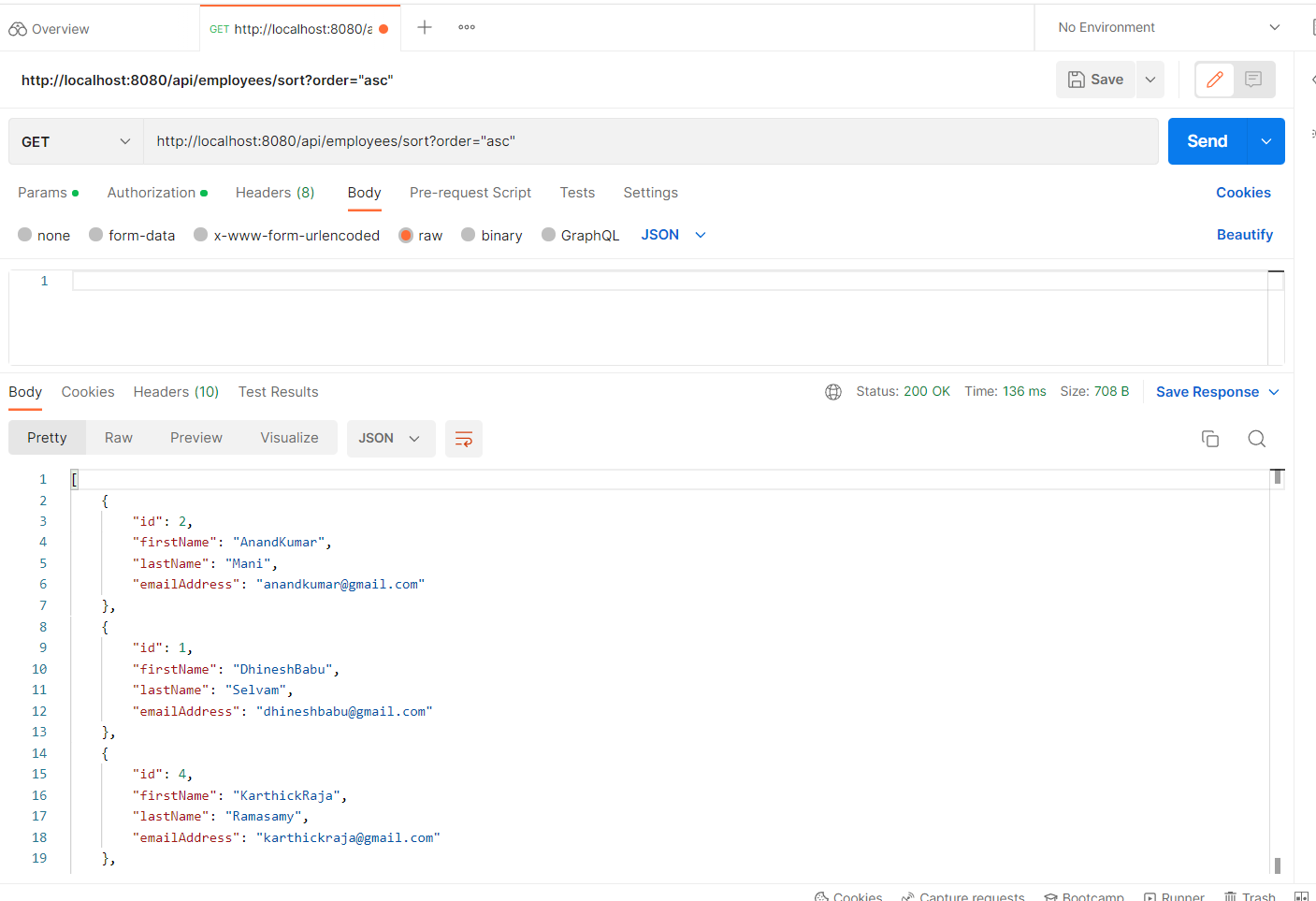
        "lastName": "Chandran",

        "emailAddress": "samnithyananth@gmail.com"

    }

]

Postman SS:



**9a Descending Order:**

**Api Url:** http://localhost:8080/api/employees/sort?order="desc"

**Http Method:** GET

**Authorization Type:** No Auth

**Request Json:** Empty

**Response Json:**

[

    {

        "id": 3,

        "firstName": "SamNithyananth",

        "lastName": "Chandran",

        "emailAddress": "samnithyananth@gmail.com"

    },

    {

        "id": 4,

        "firstName": "KarthickRaja",

        "lastName": "Ramasamy",

        "emailAddress": "karthickraja@gmail.com"

    },

    {

        "id": 1,

        "firstName": "DhineshBabu",

        "lastName": "Selvam",

        "emailAddress": "dhineshbabu@gmail.com"

    },

    {

        "id": 2,

        "firstName": "AnandKumar",

        "lastName": "Mani",

        "emailAddress": "anandkumar@gmail.com"

    }

]

Postman SS:

