

# Week 2 Assessment

## 1. Created A new Web API project in Visual Studio:

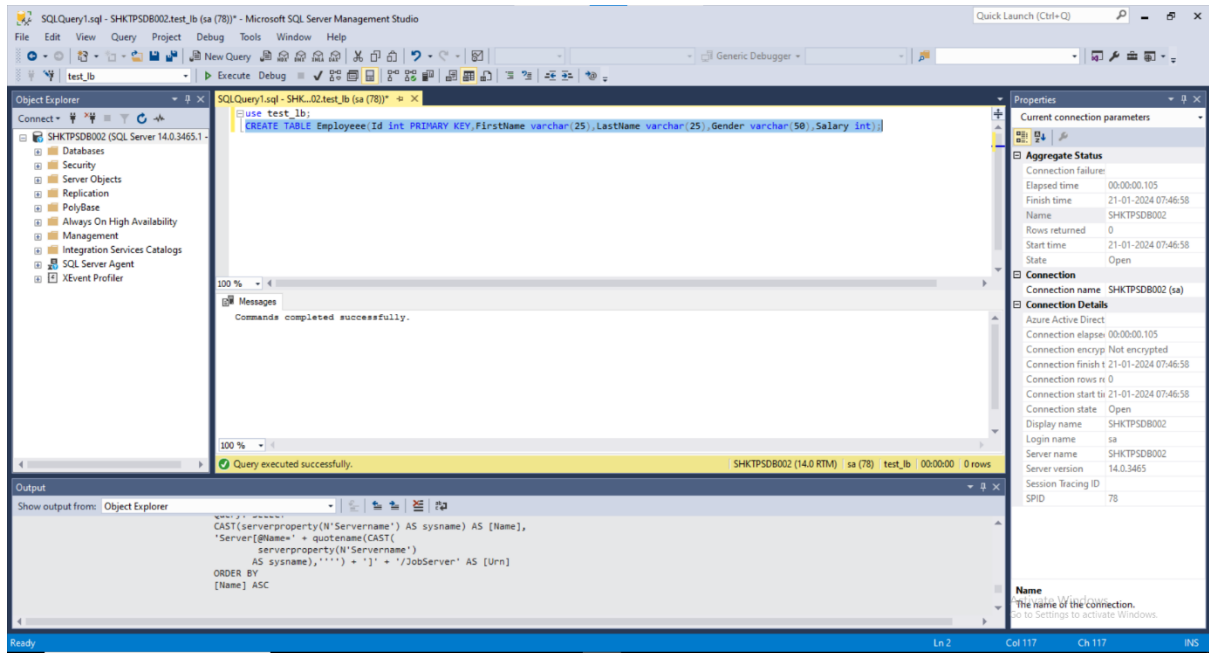
### (Overall Process)

After opening a new webapi project create table and complete the database creation work and then goto solution explorer and select the explorer and add new project and choose class library(.net framework).

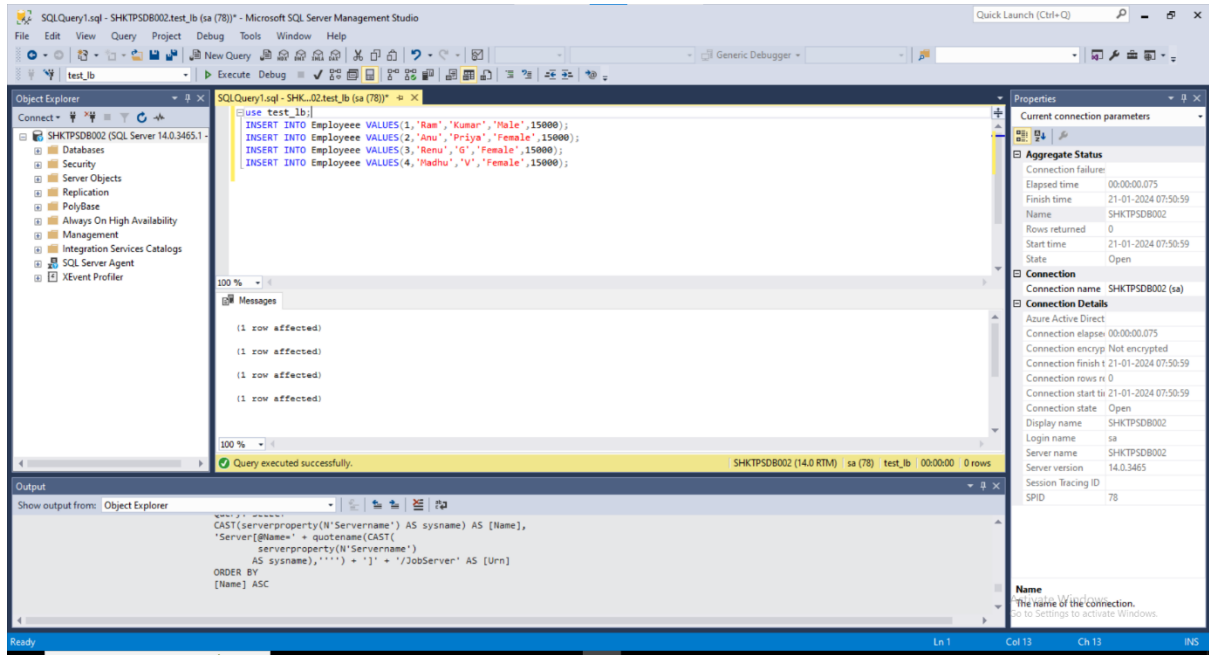
Delete the class that is auto-generated and select class library in project and ADO.NET Entity Model and then perform connection to the database.

Now Select the original project and add the reference of ADO.NET Object and then create controller and write code and execute the API.

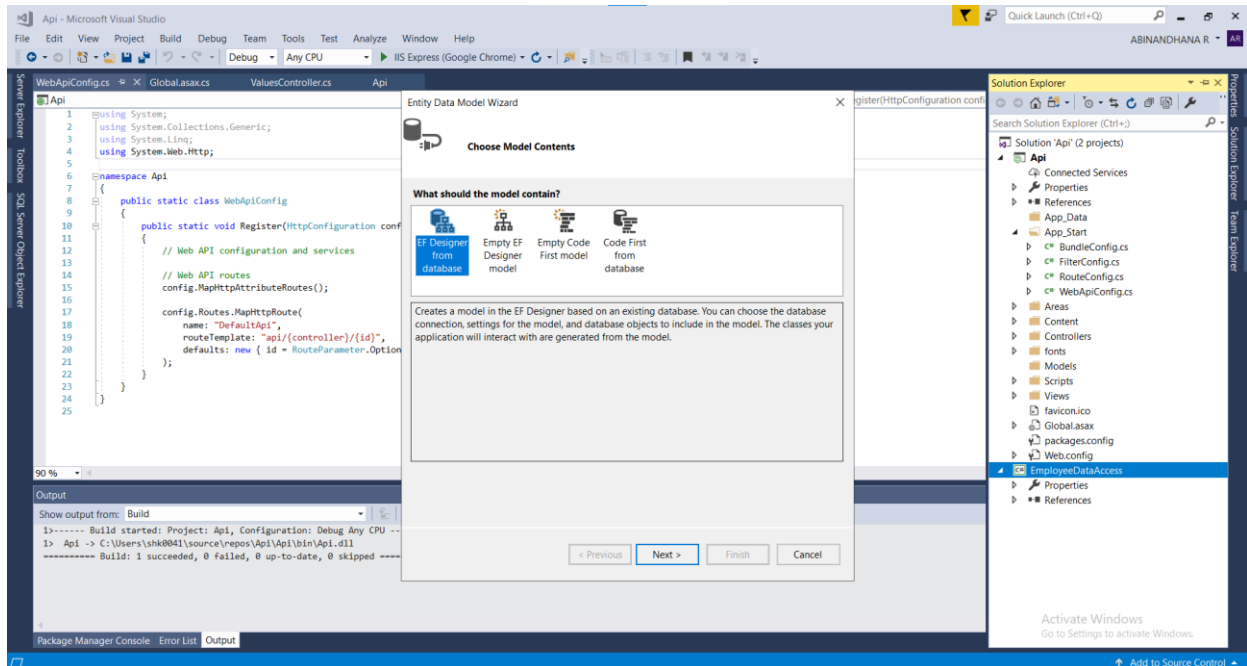
## 2. Created A Table In SQL :



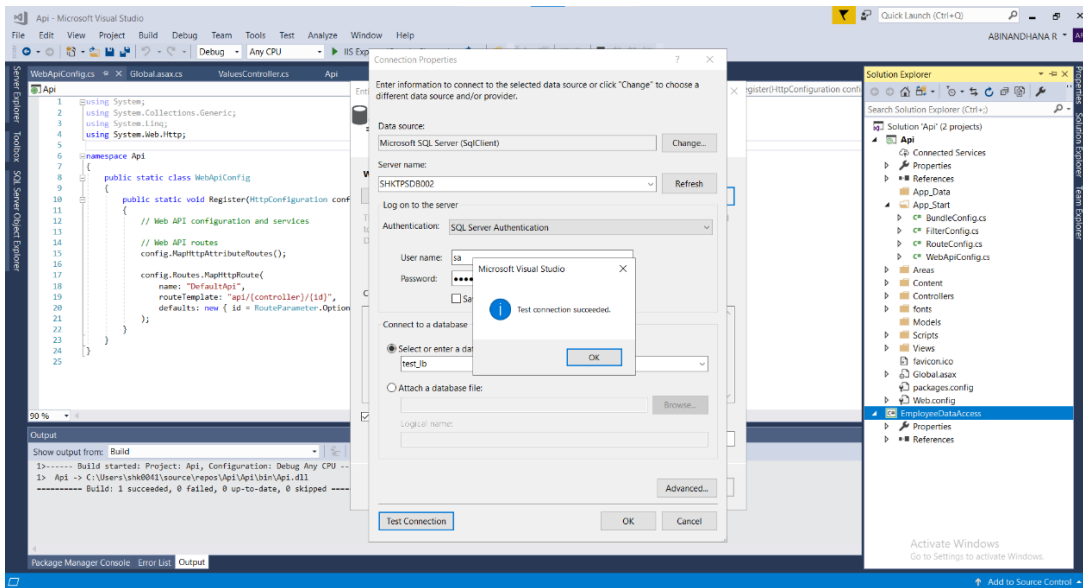
### 3. Inserted Values In The Table:



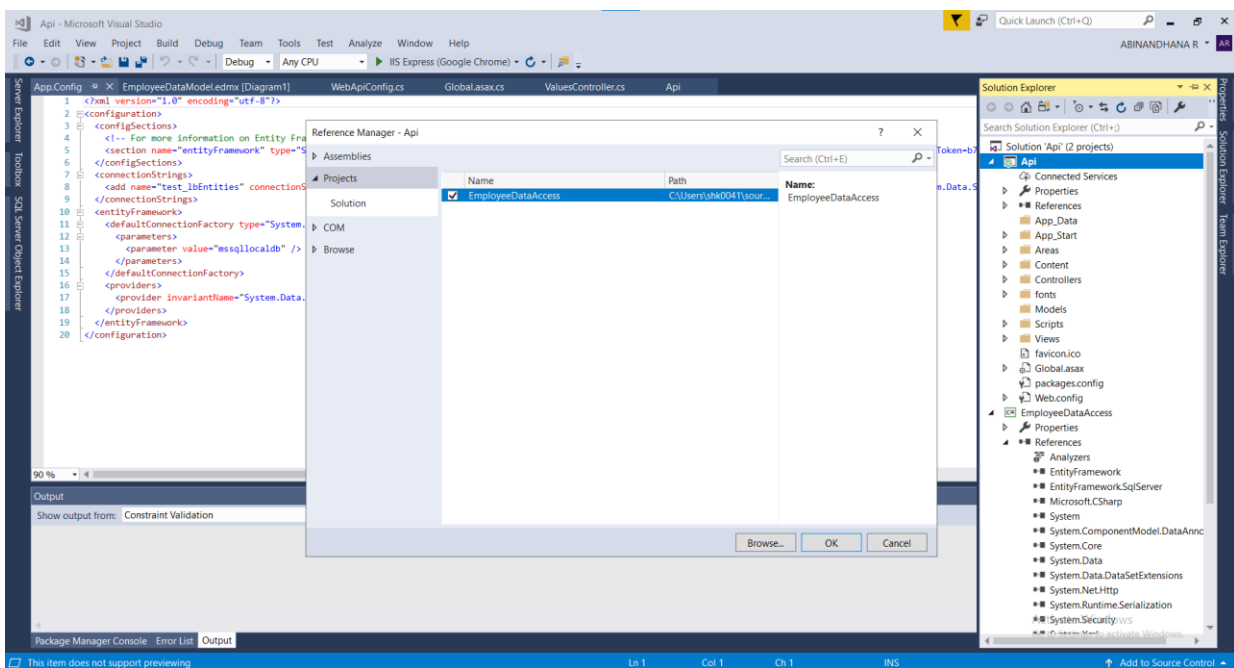
### 4. Opened Visual Studio And Connected The Database:



## 5. Tested Database connection:

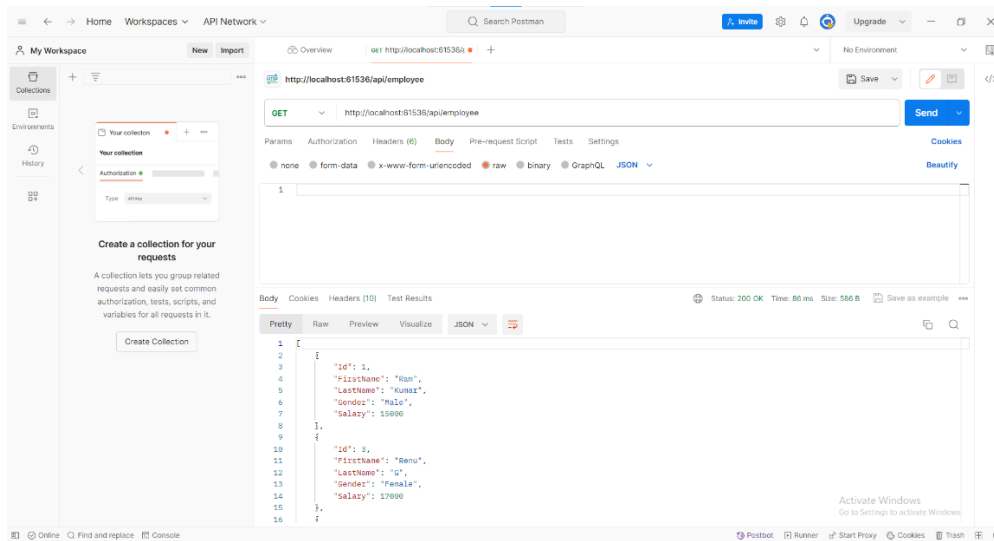


## 6. Adding ADO.NET Object as reference:

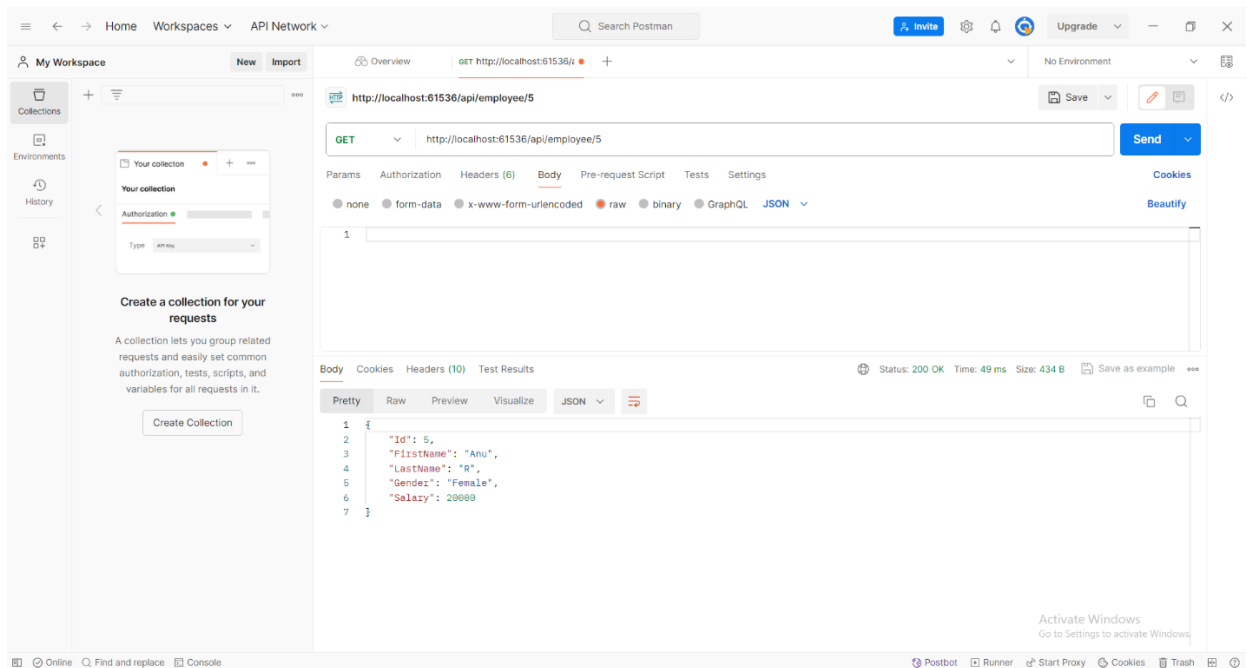


## 7.Add New Controller To the Controller Folder Sand Check Whether The Connection String Is Correct In Web.Config And App.Config Files.

## 8.API TO GET USING POSTMAN:



## GETBYID:



## 9. POST METHOD:

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:61536/api/employee`. The request body is a JSON object:

```
1 {
2   "FirstName": "Anu",
3   "Gender": "Female",
4   "Id": "5",
5   "LastName": "R",
6   "Salary": "20000"
7 }
```

The response is displayed in the bottom pane, showing a JSON object:

```
1 {"Id": "5", "FirstName": "Anu", "LastName": "R", "Gender": "Female", "Salary": "20000"}
```

The status bar indicates a 201 Created response with a time of 10.86 s and a size of 484 B.

## 10. PUT METHOD:

The screenshot shows the Postman interface for a PUT request. The URL is `http://localhost:61536/api/employee/3`. The request body is a JSON object:

```
1 {
2   "FirstName": "Renu",
3   "LastName": "G",
4   "Gender": "Female",
5   "Salary": "17000"
6 }
```

The response is displayed in the bottom pane, showing a JSON object with the text:

```
1 "Updated Data"
```

The status bar indicates a 200 OK response with a time of 13.51 s and a size of 374 B.

## SQL UPDATED:

Microsoft SQL Server Enterprise Manager interface showing a successful SQL query execution. The query executed is:

```
use test_db;
INSERT INTO Employee VALUES(1,'Ram','Kumar','Male',15000);
INSERT INTO Employee VALUES(2,'Anu','Priya','Female',15000);
INSERT INTO Employee VALUES(3,'Renu','G','Female',15000);
INSERT INTO Employee VALUES(4,'Madhu','V','Female',15000);
select from Employee;
```

The results table displays the following data:

Id	FirstName	LastName	Gender	Salary
1	Ram	Kumar	Male	15000
2	Anu	G	Female	17000
3	Madhu	V	Female	15000
4	Anu	R	Female	20000

The output pane shows the following SQL script:

```
CAST(serverproperty(N'Servername') AS sysname) AS [Name],
'Server:[Name]' + quotename(CAST(
serverproperty(N'Servername')
AS sysname),''') + ']' + '/JobServer' AS [Unr]
ORDER BY
[Name] ASC
```

## 11.DELETE METHOD:

Postman interface showing a DELETE request to `http://localhost:61536/api/employee/4`. The response is a JSON object:

```
{\"deleted\": true}
```

The status is 200 OK, Time: 68 ms, Size: 368 B.

## **CODING:**

### **CODE FOR GET METHOD:**

```
public IEnumerable<Employee> Get()
{
    using (test_lbEntities entities = new test_lbEntities())
    {
        return entities.Employees.ToList();
    }
}
```

### **CODE FOR GETBYID METHOD:**

```
public Employee Get(int id)
{
    using (test_lbEntities entities = new test_lbEntities())
    {
        return entities.Employees.FirstOrDefault(e => e.Id == id);
    }
}
```

### **CODE FOR POST METHOD:**

```
public HttpResponseMessage Post(Employee employee)
{
    try
    {
        using (test_lbEntities entities = new test_lbEntities())
        {
            entities.Employees.Add(employee);
            entities.SaveChanges();

            var response = Request.CreateResponse(HttpStatusCode.Created, employee);
            response.Headers.Location = new Uri(Url.Link("DefaultApi", new { id =
employee.Id }));
            return response;
        }
    }
    catch (Exception ex)
    {
        return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ex);
    }
}
```

### CODE FOR PUT METHOD:

```
public IHttpActionResult Put(int id, [FromBody] Employee updatedEmployee)
{
    if (updatedEmployee == null)
    {
        return BadRequest("Employee object is null");
    }

    using (test_IbEntities entities = new test_IbEntities())
    {
        var existingEmployee = entities.Employees.FirstOrDefault(e => e.Id == id);

        if (existingEmployee == null)
        {
            return NotFound();
        }

        existingEmployee.FirstName = updatedEmployee.FirstName;
        existingEmployee.LastName = updatedEmployee.LastName;
        existingEmployee.Gender = updatedEmployee.Gender;
        existingEmployee.Salary = updatedEmployee.Salary;
        entities.SaveChanges();
    }

    return Ok("Updated Data");
}
```

### CODE FOR DELETE METHOD:

```
public IHttpActionResult Delete(int id)
{
    using (test_IbEntities entities = new test_IbEntities())
    {
        var employeeToDelete = entities.Employees.FirstOrDefault(e => e.Id == id);

        if (employeeToDelete == null)
        {
            return NotFound();
        }

        entities.Employees.Remove(employeeToDelete);
        entities.SaveChanges();
    }

    return Ok("Deleted");
}
```