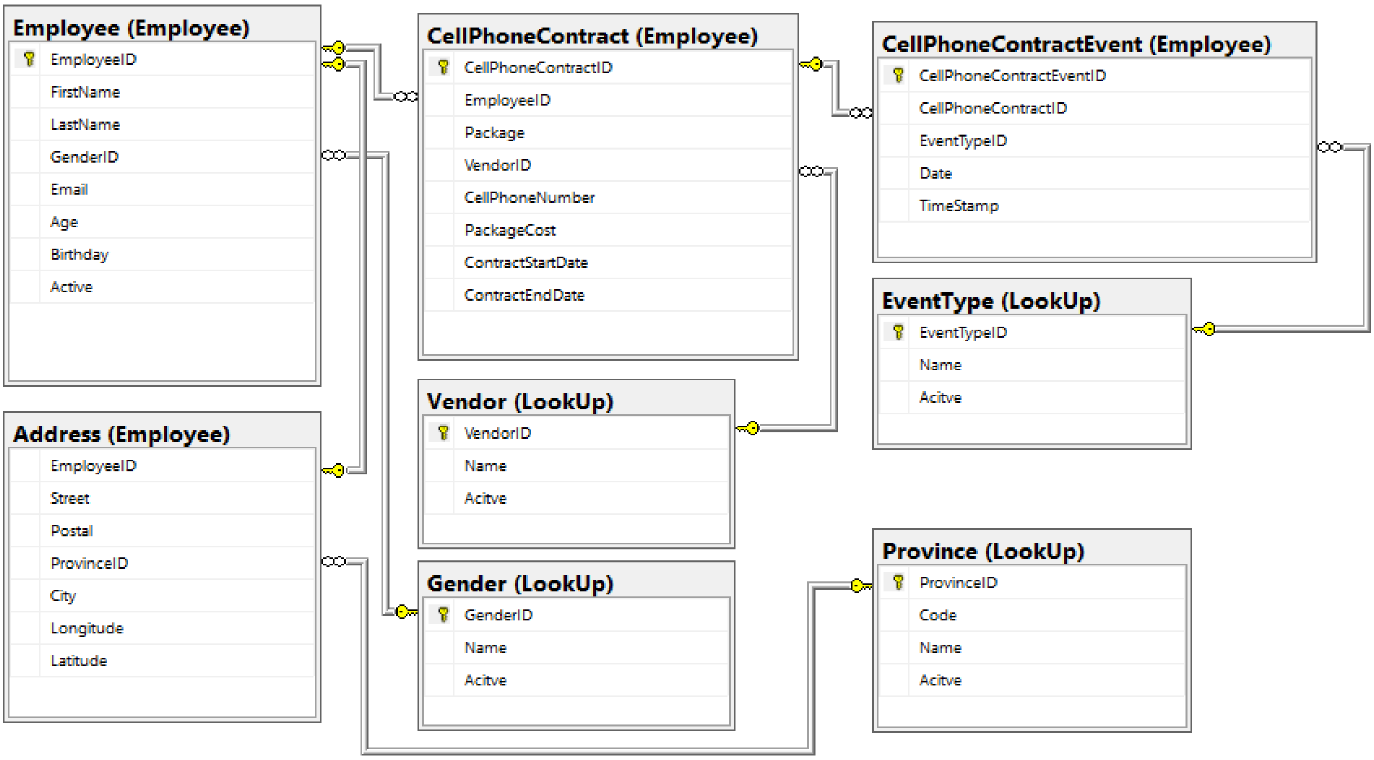
Database Challenges

Question 1 – Answer

Annexure: Database-Question1-ERD.png



Question 2 – Answer

Annexure: Database-Question2-Script.sql

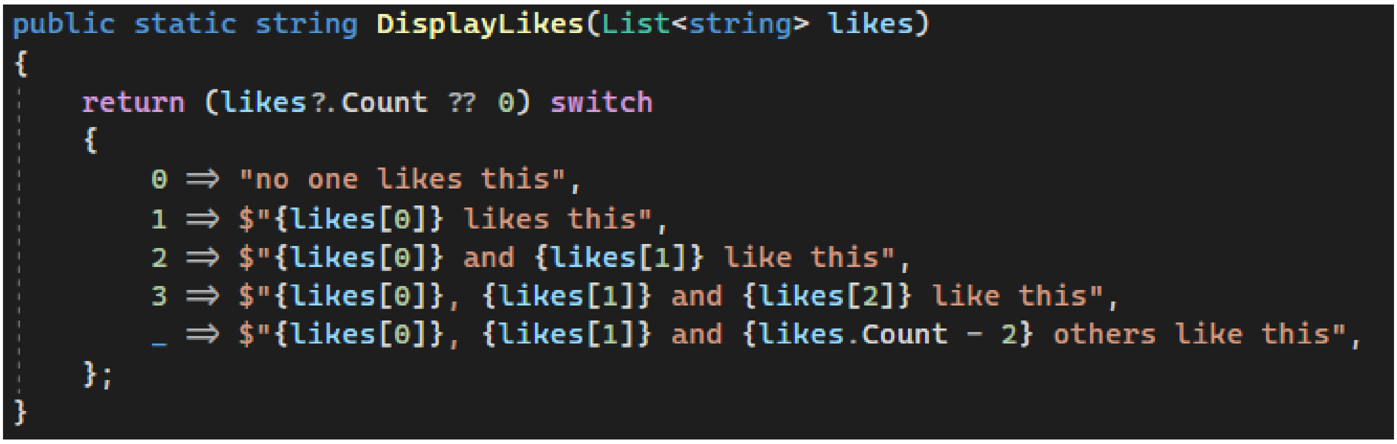
```
SELECT
Email,
LEFT(Email, CHARINDEX('@',Email)) + 'company' + RIGHT(Email, LEN(Email) - CHARINDEX('.',Email,
CHARINDEX('@',Email)) + 1) 'NewEmail'
--UPDATE E SET E.Email = LEFT(email, CHARINDEX('@',email)) + 'company' + RIGHT(email, LEN(email) -
CHARINDEX('.',email, CHARINDEX('@',email)) + 1)
FROM [dbo].[Employee] E
```

To make sure that I'm happy with my update statement and WHERE clause, I like to do a select first. After the safety check I then select the commented UPDATE clause and execute the query.

Back-end Code Challenges

Question 1 – Answer

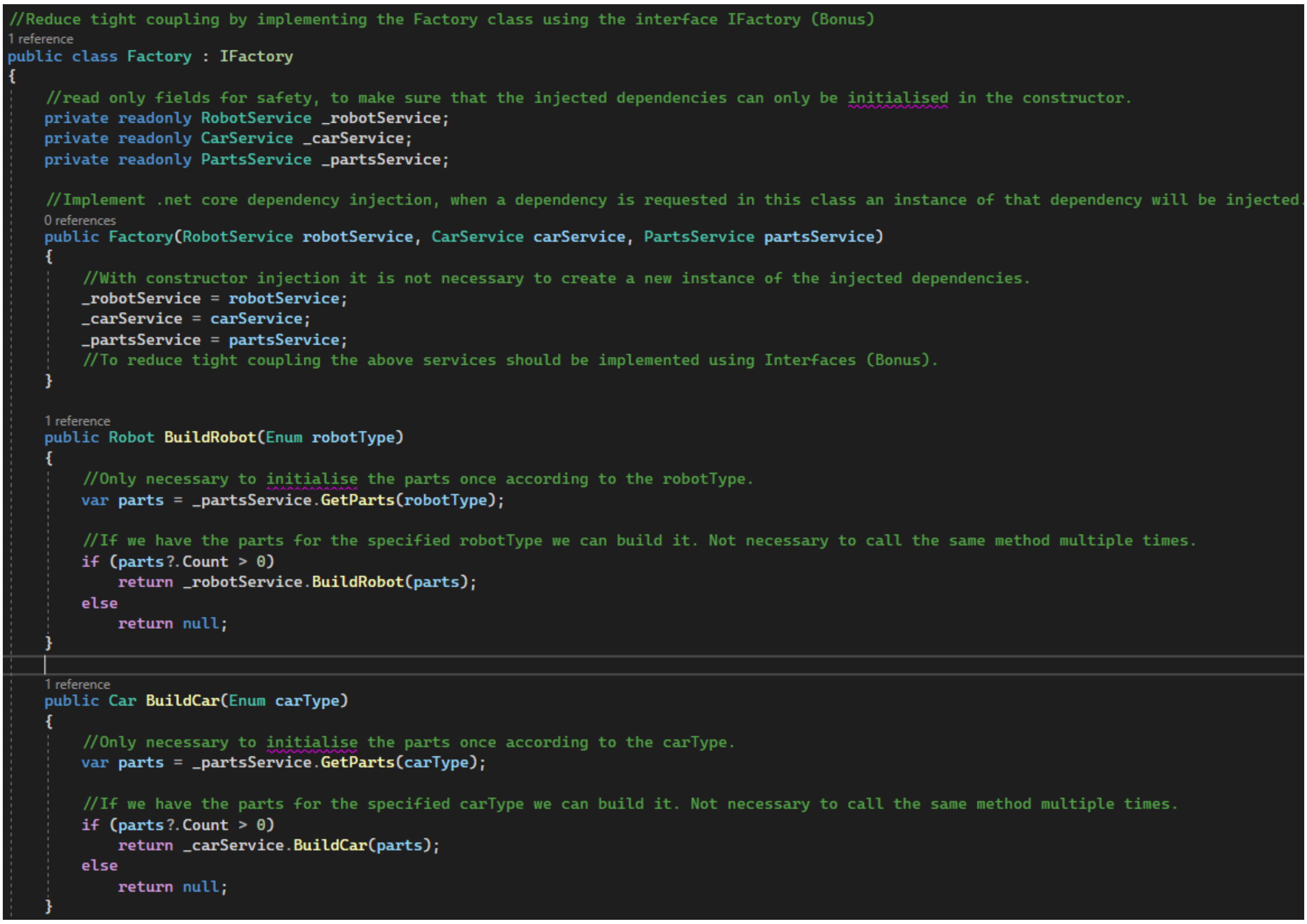
Annexure: Like.cs



```
public static string DisplayLikes(List<string> likes)
{
    return (likes?.Count ?? 0) switch
    {
        0 => "no one likes this",
        1 => $"{likes[0]} likes this",
        2 => $"{likes[0]} and {likes[1]} like this",
        3 => $"{likes[0]}, {likes[1]} and {likes[2]} like this",
        _ => $"{likes[0]}, {likes[1]} and {likes.Count - 2} others like this",
    };
}
```

Question 2 – Answer

Annexure: Factory.cs



```
//Reduce tight coupling by implementing the Factory class using the interface IFactory (Bonus)
1 reference
public class Factory : IFactory
{
    //read only fields for safety, to make sure that the injected dependencies can only be initialised in the constructor.
    private readonly RobotService _robotService;
    private readonly CarService _carService;
    private readonly PartsService _partsService;

    //Implement .net core dependency injection, when a dependency is requested in this class an instance of that dependency will be injected.
    0 references
    public Factory(RobotService robotService, CarService carService, PartsService partsService)
    {
        //With constructor injection it is not necessary to create a new instance of the injected dependencies.
        _robotService = robotService;
        _carService = carService;
        _partsService = partsService;
        //To reduce tight coupling the above services should be implemented using Interfaces (Bonus).
    }

    1 reference
    public Robot BuildRobot(Enum robotType)
    {
        //Only necessary to initialise the parts once according to the robotType.
        var parts = _partsService.GetParts(robotType);

        //If we have the parts for the specified robotType we can build it. Not necessary to call the same method multiple times.
        if (parts?.Count > 0)
            return _robotService.BuildRobot(parts);
        else
            return null;
    }

    1 reference
    public Car BuildCar(Enum carType)
    {
        //Only necessary to initialise the parts once according to the carType.
        var parts = _partsService.GetParts(carType);

        //If we have the parts for the specified carType we can build it. Not necessary to call the same method multiple times.
        if (parts?.Count > 0)
            return _carService.BuildCar(parts);
        else
            return null;
    }
```

Front-end Code Challenges

Question 1.1 – Answer

firstDiv color = red

secondDiv color = orange

But only the secondDiv will be visible, because the firstDiv has no closing tag, the secondDiv will overlap with the firstDiv. You will only see an orange block.

Question 1.2 – Answer

```
    <script>
        document.getElementById('firstDiv').style.backgroundColor = "pink";
    </script>
```

Question 1.3 – Answer

```
    <script>
        document.getElementById('secondDiv').classList.add("yellow-card");
    </script>
```

Question 2.1 – Answer

In JavaScript ‘==’ operator converts the operands to a common type before comparison. In this example the string will be converted to a number and then compared (5 == 5 = true).

Question 2.2 – Answer

Use the ‘===’ operator instead of the ‘==’ operator

```
    <script>
        function compareIt(num1, num2) {
            return num1 === num2;
        }
    </script>
```

Question 3.1 – Answer

With the use of CSS media queries and CSS Flexbox.

But I would personally use Bootstrap as I have a lot of experience with it. This framework has a grid system that is built with flexbox and is fully responsive.

Question 3.2 – Answer

It improves the performance of web applications, by reducing the number of HTTP request the browser needs to make to load a page. The page will then load faster.

Question 3.3 – Answer

Sass styling needs to be complied to a CSS file using a compiler. You must then link to the generated CSS file in the HTML page.

Question 3.4 – Answer

Use a polyfill library to add functions that is not available in older versions and use a transpiler tool to translate the new code to an older version like ES5 that is widely supported.