Oppgave 4

4.1

```sql
SELECT * FROM mydb.person
WHERE ZipCode = 5006
```

4.2

```sql
SELECT *
FROM mydb.person
WHERE datediff(now(), BirthDate) / 365.25 >= 40
```

4.3

```sql
SELECT CONCAT(Lastname," ", Firstname," ", ZipCode)
FROM mydb.person
```

4.4

```sql
SELECT avg(datediff(now(), Birthdate)) /365 as AverageInYears, MAX(BirthDate) as Max, MIN(BirthDate) as Mi
FROM mydb.person p
INNER JOIN mydb.location l
ON p.ZipCode = l.ZipCode
GROUP BY l.Location
```

4.5

```sql
SELECT location.ZipCode, location.Location
FROM location
LEFT JOIN  person ON location.ZipCode = person.ZipCode
WHERE person.person_ID IS NULL;
```

4.6

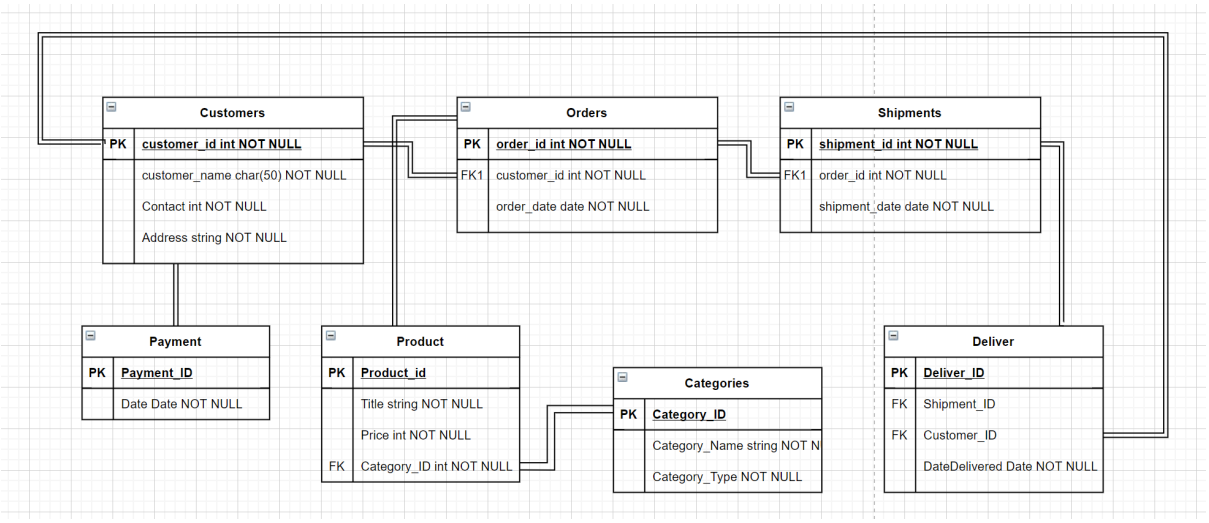
```
<configuration>
    <connectionStrings>
        <add name ="DBConnection" connectionString="server=localhost;user id=root;persistsecurityinfo=True;database=mydb"
            providerName ="System.Data.SqlClient"/>
    </connectionStrings>
```

```
public class All_users
{
    public int PersonID;
    public string Firstname;
    public string Lastname;
    public string Address;
    public string ZipCode;
    public DateTime BirthDate;
}
```

```
public class WebService1 : System.Web.Services.WebService
{
    [WebMethod]
    [ScriptMethod(ResponseFormat = ResponseFormat.Json, UseHttpGet = true)]
    0 references
    public void getAllUsers()
    {
        JavaScriptSerializer ser = new JavaScriptSerializer();
        All_users[] getStu = null;
        try
        {
            using (SqlConnection connection = new SqlConnection(DBConnect.ConnectionString))
            {
                SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM mydb.person", connection);
                sda.SelectCommand.CommandType = CommandType.Text;
                DataTable datatable = new DataTable();
                sda.Fill(datatable);
                getStu = new All_users[datatable.Rows.Count];
                int Count = 0;
                for (int i = 0; i <datatable.Rows.Count; i++){

                    getStu[Count] = new All_users();
                    getStu[Count].PersonID = Convert.ToInt32(datatable.Rows[i]["PersonID"]);
                    getStu[Count].Firstname = Convert.ToString(datatable.Rows[i]["Firstname"]);
                    getStu[Count].Lastname = Convert.ToString(datatable.Rows[i]["Lastname"]);
                    getStu[Count].Address = Convert.ToString(datatable.Rows[i]["Address"]);
                    getStu[Count].ZipCode = Convert.ToString(datatable.Rows[i]["ZipCode"]);
                    getStu[Count].BirthDate = Convert.ToDateTime(datatable.Rows[i]["BirthDate"]);
                    Count++;
                }
                datatable.Clear();
                connection.Close();
            }
        } catch(Exception ex)
        {
            var JSonData = new
            {
                getStu = getStu
            };
            HttpContext.Current.Response.Write(ser.Serialize(JSonData));
        }
    }
```

```
    public class DBConnect
    {
        1 reference
        public static string ConnectionString
        {
            get
            {
                return WebConfigurationManager.ConnectionStrings["DBConnection"].ConnectionString.ToString();
            }
        }
    }
}
```


Customers
PK customer_id int NOT NULL
customer_name char(50) NOT NULL
Contact int NOT NULL
Address string NOT NULL
Orders
PK order_id int NOT NULL
FK1 customer_id int NOT NULL
order_date date NOT NULL
Shipments
PK shipment_id int NOT NULL
FK1 order_id int NOT NULL
shipment_date date NOT NULL
Payment
PK Payment_ID
Date Date NOT NULL
Product
PK Product_id
Title string NOT NULL
Price int NOT NULL
FK Category_ID int NOT NULL
Categories
PK Category_ID
Category_Name string NOT N
Category_Type NOT NULL
Deliver
PK Deliver_ID
FK Shipment_ID
FK Customer_ID
DateDelivered Date NOT NULL

Oppgave 5.a

```
2 references
public class PersonDBAccessLayer
{
    MySqlConnection con = new MySqlConnection("server=localhost;user id=root;password=pass; database=user");
    1 reference
    public string AddPersonRecord(PersonEntities personEntities)
    {
        try
        {
            MySqlCommand cmd = new MySqlCommand("sp_Employee_Add", con);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@personID", personEntities.personID);
            cmd.Parameters.AddWithValue("@Firstname", personEntities.Firstname);
            cmd.Parameters.AddWithValue("@Lastname", personEntities.Lastname);
            cmd.Parameters.AddWithValue("@Address", personEntities.Address);
            cmd.Parameters.AddWithValue("@ZipCode", personEntities.ZipCode);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            return ("Data save Successfully");
        }
        catch (Exception ex)
        {
            if (con.State == ConnectionState.Open)
            {
                con.Close();
            }
            return (ex.Message.ToString());
        }
    }

    0 references
    internal string AddPersonRecords(PersonEntities personEntities)
    {
        throw new NotImplementedException();
    }
}
}
```

```
public class PersonController : Controller
{
    PersonDBAccessLayer perdb = new PersonDBAccessLayer();

    0 references
    public class Person12Controller : Controller
    {
        }
        [HttpPost]
        0 references
        public IActionResult Create([Bind] PersonEntities personEntities)
        {
            try
            {
                if (ModelState.IsValid)
                {
                    string resp = perdb.AddPersonRecord(personEntities);
                    TempData["msg"] = resp;
                }
            }
            catch (Exception ex)
            {
                TempData["msg"] = ex.Message;
            }
            return View();
        }
    }
}
```

```
public class PersonEntities
{
    [Required]
    1 reference
    public int personID{ get; set; }
    [Required]
    1 reference
    public string Firstname{ get; set; }
    [Required]
    1 reference
    public string Lastname { get; set; }
    [Required]
    1 reference
    public string Address { get; set; }
    [Required]
    1 reference
    public string ZipCode { get; set; }
    0 references
    public DateTime BirthDate { get; set; }

}
}
```

```
@model WebApplication3.Models.PersonEntities
@{
    ViewData["Title"] = "Create Person";
}
<h2>Create</h2>
<h4>Employees</h4>
<hr />
<form asp-action="Create" class="form-horizontal">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <div class="form-group">
        <label class="control-label">personID</label>
        <input asp-for="personID" class="form-control" />
        <span asp-validation-for="persnoID" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label class="control-label">Firstnam</label>
        <input asp-for="Firstname" class="form-control" />
        <span asp-validation-for="Firstname" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label class="control-label">Lastname</label>
        <input asp-for="Lastname" class="form-control" />
        <span asp-validation-for="Lastname" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label class="control-label">Address</label>
        <input asp-for="Address" class="form-control" />
        <span asp-validation-for="Address" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label class="control-label">ZipCode</label>
        <input asp-for="ZipCode" class="form-control" />
        <span asp-validation-for="ZipCode" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label class="control-label">BirthDate</label>
        <input asp-for="BirthDate" class="form-control" />
        <span asp-validation-for="BirthDate" class="text-danger"></span>
    </div>
    <div class="form-group">
        <input type="submit" value="Submit" class="btn bg-primary" />

    </div>
</form>
@{
    if (@TempData["Msg"] != null)
    {
        <script>

            alert('@TempData["msg"]')
        </script>

    }
}
```

Oppgave 5b

```
class Person
{
    1 reference
    public int personID
    {
        get; set;
    }

    1 reference
    public string Firstname
    {
        get; set;
    }
    1 reference
    public string Lastname
    {
        get; set;
    }
    1 reference
    public string Address
    {
        get; set;
    }
    0 references
    public string ZipCode
    {
        get; set;
    }
    1 reference
    public DateTime BirthDate
    {
        get; set;
    }


    0 references
    private List<Person> PersonList()
    {
        string connetionString;
        MySqlConnection cnn;
        connetionString = "Server=localhost;user id=root;password=pass;database=user";
        cnn = new MySqlConnection(connetionString);
        cnn.Open();
        MessageBox.Show("Connection Open  !");
        MySqlCommand command = cnn.CreateCommand();
        command.CommandText = "SELECT * FROM person WHERE ZipCode = 5006";
        MySqlDataReader reader = command.ExecuteReader();
```

```
        MySqlDataReader reader = command.ExecuteReader();
        var wsl = new List<Person>();
        if (reader.Read())
        {
            var ws = new Person();
            ws.personID = reader.GetInt32("personID");
            ws.Firstname = reader.GetString("Firstname");
            ws.Lastname = reader.GetString("Lastname");
            ws.Address = reader.GetString("Address");
            ws.BirthDate = reader.GetDateTime("BirthDate");
            wsl.Add(ws);


        }

        cnn.Close();
        Console.WriteLine(wsl);
        return wsl;

    }

}
```