Opgave: Database & ConsoleApplikation

Du skal lave en prototype på et system til en bowlingbane. Dette indebærer; Du skal lave tre tabeller i en database, samt et lille program, der kan vise værdierne.

Lav tre tabeller

Du skal lave tre tabeller Person, Booking og BowlingAlley, Du vælger bare en database – du kan sagtens have flere tabeller i samme database.
Brug W3Scools som reference-opslag:

- Create table https://www.w3schools.com/sql/sql_create_table.asp
- Primary key https://www.w3schools.com/sql/sql_primarykey.asp
- Foreign key https://www.w3schools.com/sql/sql_foreignkey.asp

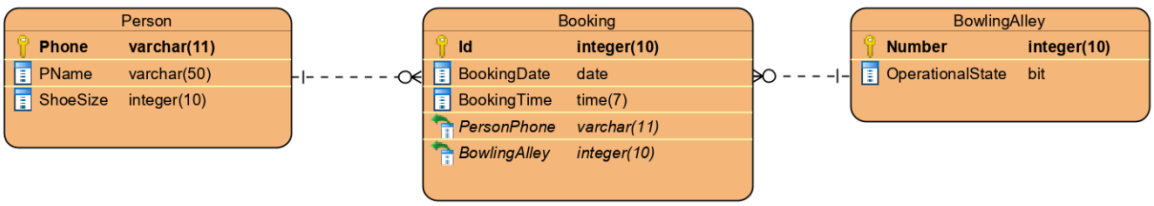
Person:
Phone (nvarchar(11)), PName (nvarchar(50)), ShoeSize (integer(10))

Booking:
Id (integer(10)), BookingDate(date), BookingTime (time(7)), *PersonPhone (nvarchar(11))*, *BowlingAlley (interger(10))*

Bowling Alley:
Number (interger(10)), OperationalState (bit)

Understreng+fed = primær nøgle

Understreg+kursiv = fremmednøgle



Extra: ID for booking skal generes automatisk i databasen

Lave et lille program

Til det skal du have model-klasser, samt selve ConsoleApp'en.

Model-klasse(r)

Du skal lave et projekt **'Class Library'**

Du skal implementere de 3 model klasser:

1. Person med properties, default konstruktør og TosString:
   Phone (string), Name (string), ShoeSize (int)

2. Booking:
   Id (int), Date(datetime), Time (timespan), PersonPhone (string), BowlingAlley (int)

3. Bowling Alley:
   Number (int), OperationalState (bool)

Console Applikation

Du skal lave endnu et projekt '**Console Application**', hvor du i første omgang skal lave en database-forbindelse som lave CRUD på Person (tabellen).

Husk at lav en reference (under dependencies) til dit library.
Installer NuGet pakken '**System.Data.SqlClient**'

Lav en klasse DBWorker og kald metoder fra programmet.

I DBWorker klassen skal du lave 5 metoder:

```
public List<Person> GetAllPersons()

public Person GetPersonByPhoneNo(string phone)

public Person AddPerson(Person person)

public Person DeletePerson(string phone)

public Person UpdatePerson(string phone, Person person)
```

Du skal implementere de fem metoder og kalde dem fra programmet.

Implementering af metoder og kald til databasen

For SELECT

Alle select metoder er opbygget efter følgende skabelon:

```
List<Person> personer = new List<Person>();
fx string queryString = "select * from Person";

using (SqlConnection connection = new SqlConnection(connectionString))
{
   connection.Open();

   SqlCommand command = new SqlCommand(queryString, connection);
   SqlDataReader reader = command.ExecuteReader();

   while (reader.Read())
   {
      Person p = ReadPerson(reader);
      personer.Add(p);
   }
}
return personer;
```

ConnectionString findes under properties for databasen.
ReadPerson er en hjælpe metode

```
private Person ReadPerson(SqlDataReader reader)
{
   Person p = new Person();

   p.Phone = reader.GetString(0);
   p.Pname = reader.GetString(1);
   p.ShoeSize = reader.GetInt32(2);

   return p;
}
```

For INSERT, UPDATE og DELETE

SQL-kommandoerne insert, update og delete (og create, alter osv) er som følger:
Den skabelon som alle 5 metoder er opbygget af er som følger:

```
fx string queryString = "insert into Person Values(@Phone,@PName,@Size)";

using (SqlConnection connection = new SqlConnection(connectionString))
{
   connection.Open();

   SqlCommand command = new SqlCommand(queryString, connection);
   command.Parameters.AddWithValue("@Phone", person.Phone);
   command.Parameters.AddWithValue("@PName", person.Pname);
   command.Parameters.AddWithValue("@Size", person.ShoeSize);


   int rows = command.ExecuteNonQuery();
   if (rows != 1)
   {
      throw new ArgumentException("Person er ikke oprettet");
   }
```

Ekstra implementer tilsvarende metoder for BowlingAlley og Bookings(lidt svær)