

## Hintergrund:

In der Ausbildung zum Fachinformatiker wird im Christophorus-Werk Lingen e.V. unter anderem das Thema SQL (Structured Query Language) thematisiert. Die Auszubildenden bekommen vom Ausbilder gezeigt, wie die Datenbanksprache SQL aufgebaut ist und wie einfache Abfragen auf einer SQL-Datenbank abgesetzt werden können.

Nach der theoretischen Einleitung sollen die Auszubildenden dann eigenständig an einer Datenbank das Gelernte umsetzen.

## Aufgabe:

Deine Aufgabe ist es, ein geeignetes "Lernprogramm SQL" für die Abzubildenden (in Win-Forms) zu erstellen. Das Programm soll in C# geschrieben sein. Die Datenbank soll mit SQLite angesprochen werden und als *Datenbank.db* im Verzeichnis des Lernprogramms liegen.

## Grober Aufbau des Programms:

Das Programm soll eine Textbox (Multiline) als "Input" haben. Dort sollen dann SQL-Statements eingetragen werden können. Mit einem Klick auf einen "Absenden" Button soll dieser per SQLite verarbeitet werden. Unter dem "Input"-Feld soll eine "Ausgabe" als DataGridView vorhanden sein. Die GridView soll die Ergebnisse bei einer Selectabfrage enthalten.

Das Programm soll folgende Aufgaben erfüllen:

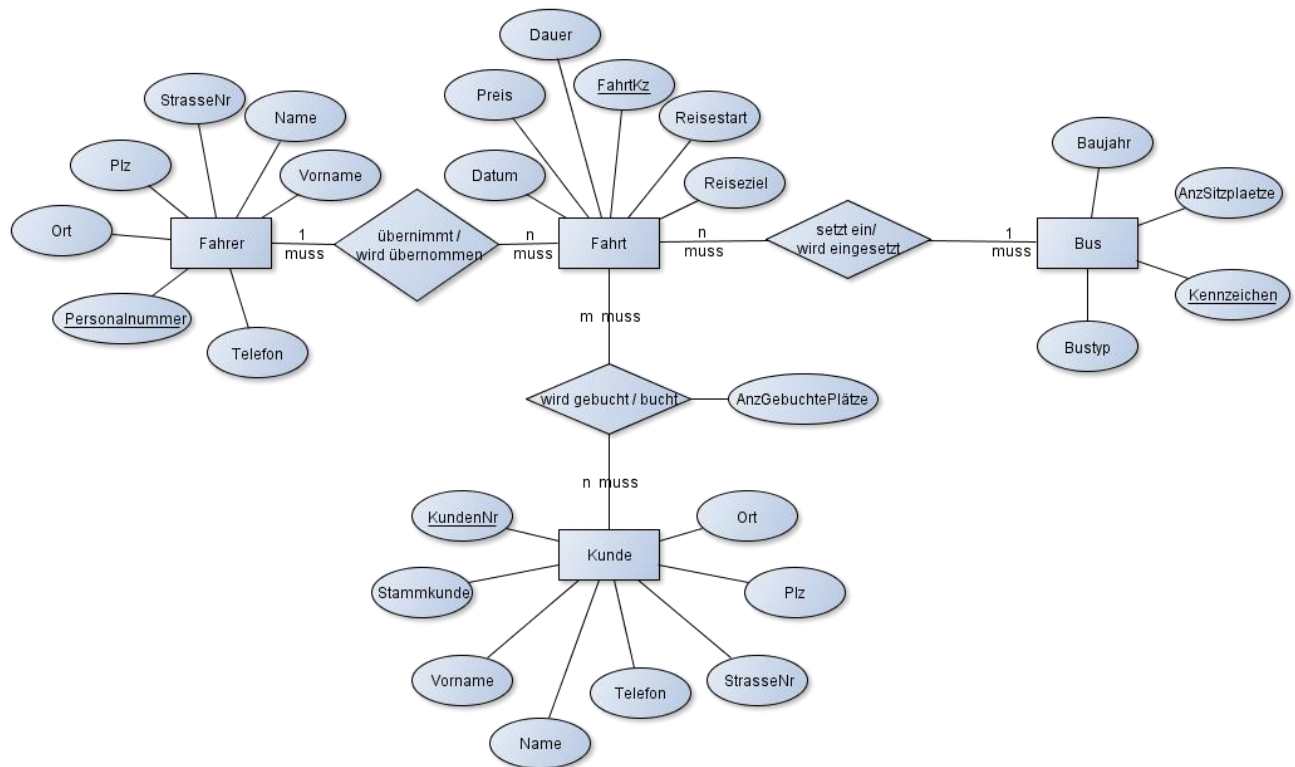
- Beim Starten soll vom Programm eine geeignete Datenbank erstellt werden
  - Mehrere Tabellen mit Primärschlüsseln und Beziehungen zueinander
  - Wenn dir nichts einfällt, nimm das Beispiel von unten (Busunternehmen)
- Es sollen die hinterlegten Tabellen ausgegeben werden können. (Als MessageBox, als extra Window oder in einer TreeView)
- Es soll die Grundfunktionen: SELECT, INSERT, UPDATE, DELETE beherrschen
- Es soll automatisch entschieden werden, ob es ein Query oder ein NonQuery Statement ist.
  - Bei einem Query soll es eine Ausgabe mit der entsprechenden Antwort geben. (Benutze ein DataGridView // Eine Möglichkeit zur Umsetzung siehe ganz unten)
  - Bei Nonquery soll es ein Feedback ("X-Zeilen verändert") geben (Benutze den Rückgabewert von -> SqlCommand.ExecuteNonQuery());
    - *The number of rows inserted, updated, or deleted. -1 for SELECT statements.* <https://learn.microsoft.com/en-us/dotnet/api/microsoft.data.sqlite.sqlitecommand.executenonquery?view=msdata-sqlite-7.0.0>

- Wenn es einen Fehler in einem Statement gibt, soll eine geeignete Fehlermeldung zurückgegeben werden. (Setze einen Breakpoint und schau dir im "Debugger" die Fehlermeldung an. Entscheide dann, welche Nachricht am besten an den User zurückgegeben werden sollte.) Schaue dir Internetquellen dazu an (<https://www.sqlite.org/rescode.html> )
- Es soll eine Funktion geben, die die Datenbank wieder auf "Werkseinstellung" zurücksetzt.
- Alternativ soll es möglich sein, eine Datenbank zu importieren und zu exportieren -> SQLiteConnection.BackupDatabase // Löschen der "alten" Datenbank.db" und ersetzen mit der zu importierenden Datenbank.
- Das Programm soll ein Logfile als .log Datei erstellen. Alle abgesetzten Befehle sowie die Antworten (verkürzt auf maximal 20 Antwortzeilen) sollen im Log protokolliert werden.
- Erweitere das Programm mit eigenen Ideen

Query von Sqlite als Winforms DataGridView Ausgabe:

```
private void button1_Click_1(object sender, EventArgs e)
{
    conn.Open();
    SQLiteCommand comm = new SQLiteCommand("Select * From Patients", conn);
    using (SQLiteDataReader read = comm.ExecuteReader())
    {
        while (read.Read())
        {
            dataGridView1.Rows.Add(new object[] {
                read.GetValue(0), // U can use column index
                read.GetValue(read.GetOrdinal("PatientName")), // Or column name like this
                read.GetValue(read.GetOrdinal("PatientAge")),
                read.GetValue(read.GetOrdinal("PhoneNumber"))
            });
        }
    }
}
```

<https://stackoverflow.com/questions/29107096/fill-datagridview-from-sqlite-db-c>



[https://sql-tutorial.de/home/start.php?include=db\\_busunternehmen](https://sql-tutorial.de/home/start.php?include=db_busunternehmen)