# HTL Innsbruck

Anichstrasse 26-28
A-6020 Innsbruck
www.htlinn.ac.at

## Elektronik
### Telekommunikation
### Technische Informatik

FSST – 7/8. Abendschule

# *Datenbanken, MySQL-Anbindung Und ODBC mit c#*

2022-01-02

DI. Mag(FH)
Klingler Felix

# Inhalt

# 1 GRUNDLAGEN DATENBANK

- Einführung in Datenbanken
  https://openbook.rheinwerk-verlag.de/it_handbuch/12_001.html
- Normalformen
  1. NF: https://www.youtube.com/watch?v=sFG5pR5016k
  2. NF: https://www.youtube.com/watch?v=kAuZ-v-HBtA
  3. NF: https://www.youtube.com/watch?v=x183UkdsPQ8

# 2 DAS PROBLEM

Daten können in einer Datenbank gespeichert werden. Die Datenbank wiederum ist in einem Datenbank-Managementsystem DMBS eingebettet und wird von diesem manipuliert.

Ein DBMS verfügt über folgende minimale Funktionalitäten:

- Erstellen und Verwalten von Datenbanken
- Erstellen und Verwalten von Tabellen in diesen Datenbanken
- Erstellen, verwalten und ausführen von Funktionen innerhalb einer Datenbank
- Benutzermanagement mit Rechtevergabe

Eine Datenbank wiederum muss über folgende Fähigkeiten verfügen:

- Einfügen neuer Daten
- Verändern alter Daten
- Löschen alter Daten
- Suchen von Daten

MySQL ist ein DBMS, das zur freien Entwicklung zur Verfügung gestellt wird. Dabei ist zu beachten, dass sicherheitstechnische Aspekte nur rudimentär vorhanden sind, sodass sich MySQL NICHT als DBMS für kommerzielle/professionelle Anwendungen eignet.

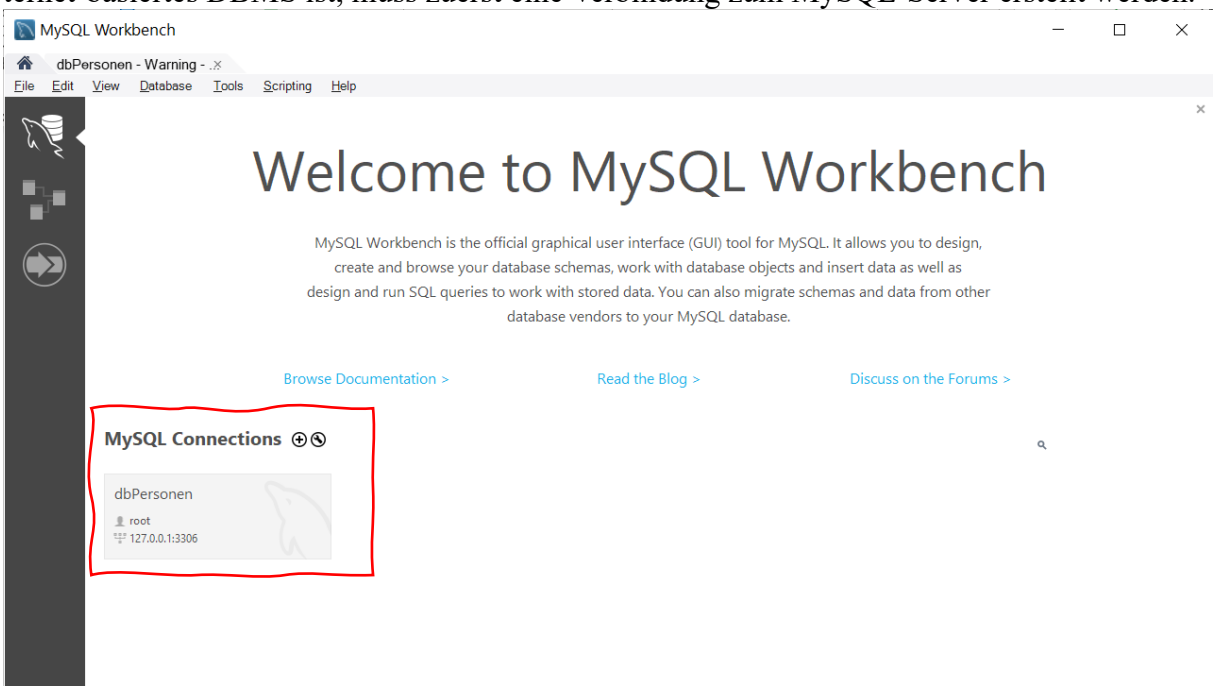
Als Frontend für MySQL wird die MySQL-Workbench verwendet. Sie stellt eine GUI für die oben aufgezählten Funktionalitäten dar.

Software-Download:

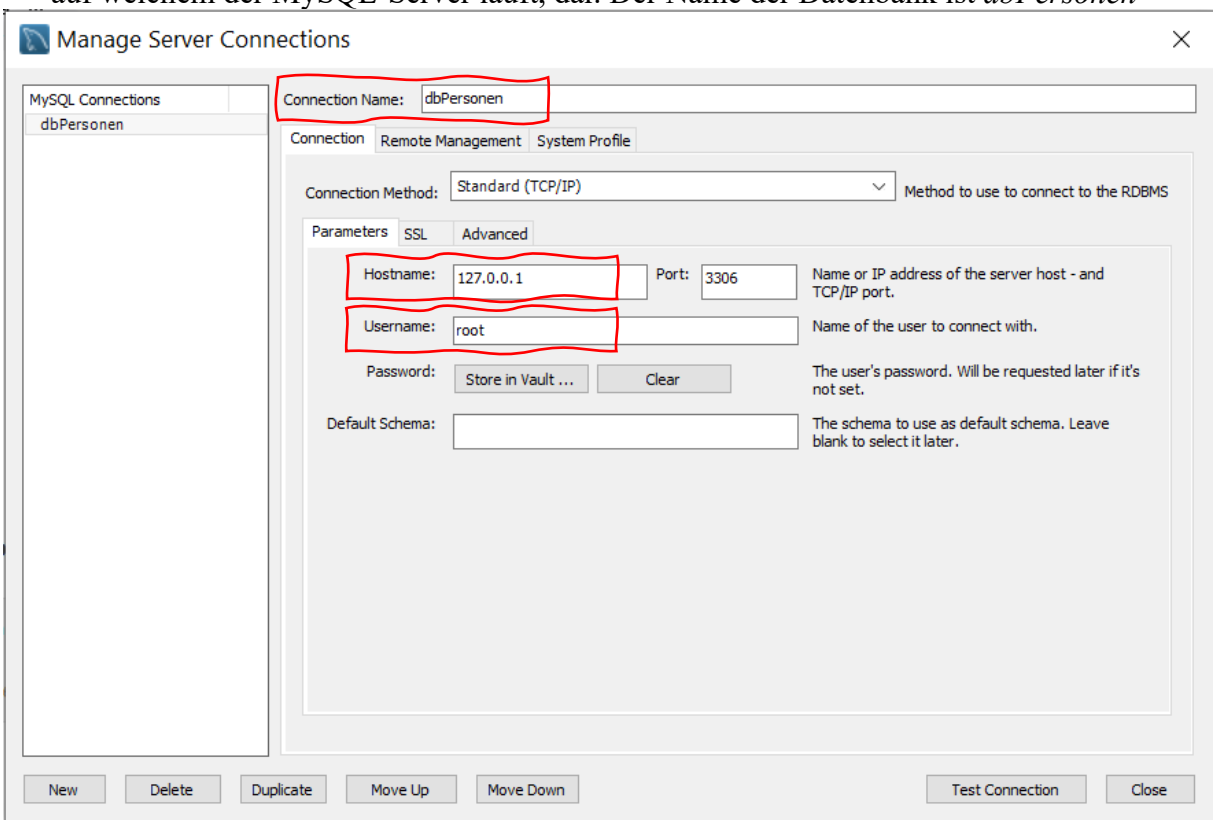
- MYSQL ist z.B. Teil von XAMPP (Web-Server) - die Portable-Version reicht
  https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/
- MySQL-Workbench
  https://dev.mysql.com/downloads/workbench/
- MySQL-ODBC-Connector (komplette Installation)
  https://dev.mysql.com/downloads/connector/odbc/
- MySQL-DotNet (komplette Installation)
  https://dev.mysql.com/downloads/connector/net/
- Dokumentation: MySQL in VS/c#
  https://downloads.mysql.com/docs/visual-studio-en.pdf
  Oder
  https://www.codeproject.com/Articles/43438/Connect-C-to-MySQL

# 3 ERSTELLUNG EINER DATENBANK IN MYSQL

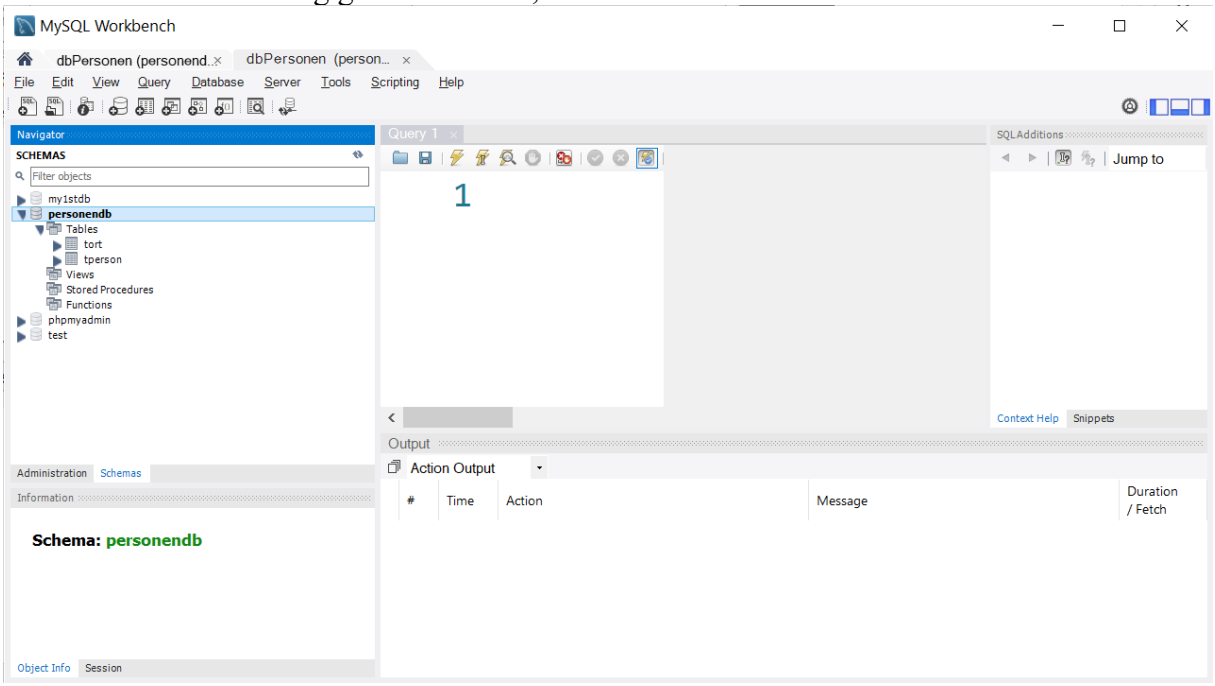
Zuerst ist eine neue Datenbank, in MySQL *Schema* genannt, zu erstellen. Da MySQL ein Internet-basiertes DBMS ist, muss zuerst eine Verbindung zum MySQL-Server erstellt werden.



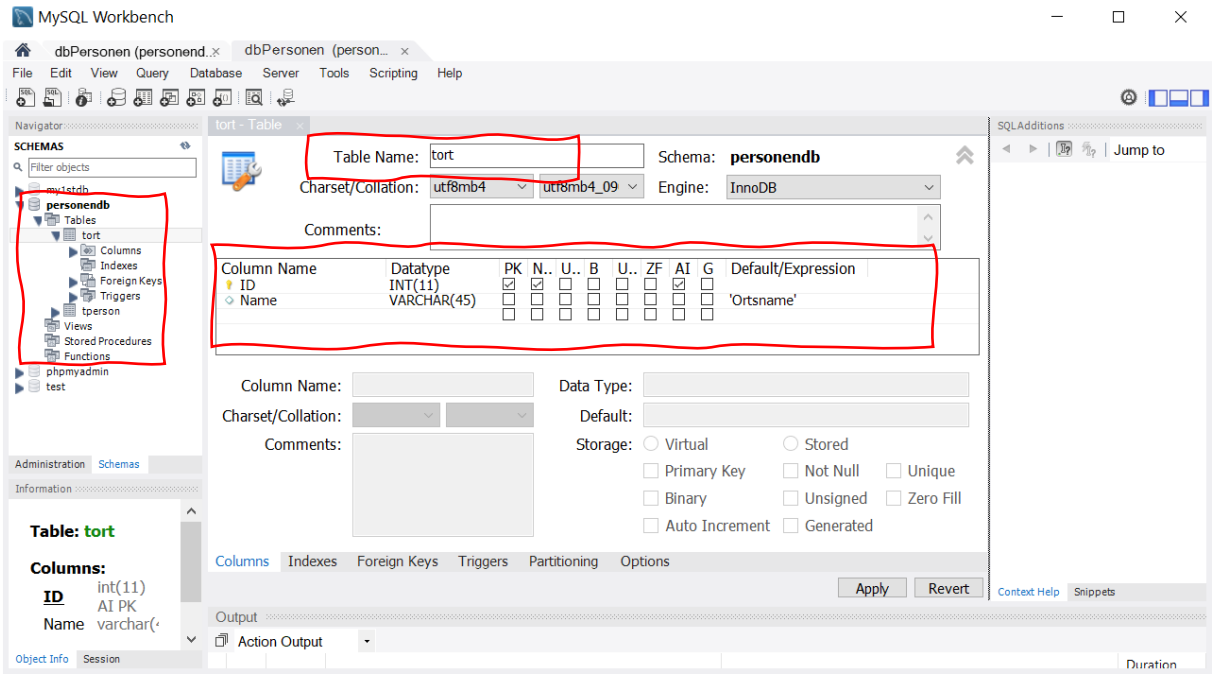

- Ist eine Verbindung schon vorhanden ➔ Doppel clicken
- Ist keine Verbindung vorhanden ➔auf (+) drücken und eine neue Verbindung mit den Verbindungsparametern erstellen. Die Internet-Adresse 127.0.0.1 stellt den eigenen Rechner, auf welchem der MySQL-Server läuft, dar. Der Name der Datenbank ist *dbPersonen*



Nachdem die Verbindung geöffnet wurde, können die Tabellen der Datenbank erstellt werden.



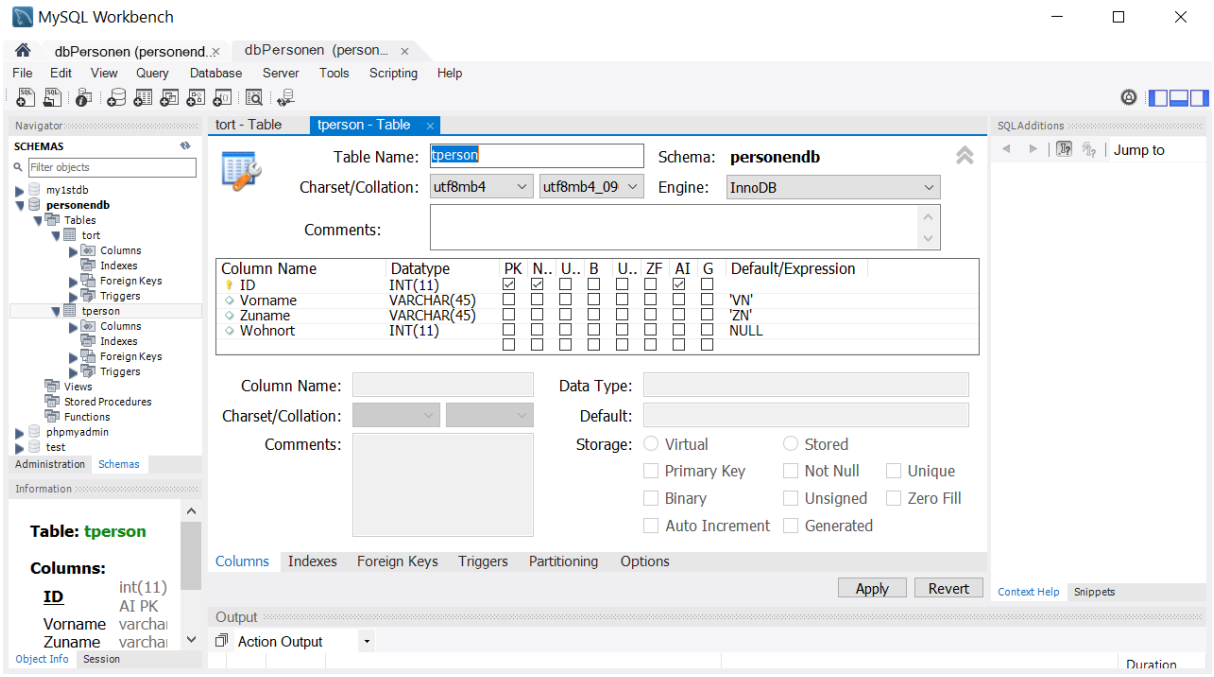

- Die Datenbank *personendb* muss neu erstellt werden.
  im Fenster *Schemas* ➔ rechte Maus ➔ *create Schema*
- Die benutzte Datenbank *personendb* wird als default-Schema aktiviert (rechte Mause)
- Erstellen der Tabelle *tort*




Hinweise:

- ✓ ID ist der Schlüssel und somit eine fortlaufende Nummer
  ➔ PK (Primary Key), nicht null (N.N) und auto inkrement (AI)
- ✓ Name ist ein String mit maximal 45 Zeichen (VARCHAR (45)) und dem Default-Wert „Ortsname“

➢ Erstellen der Tabelle *tperson*



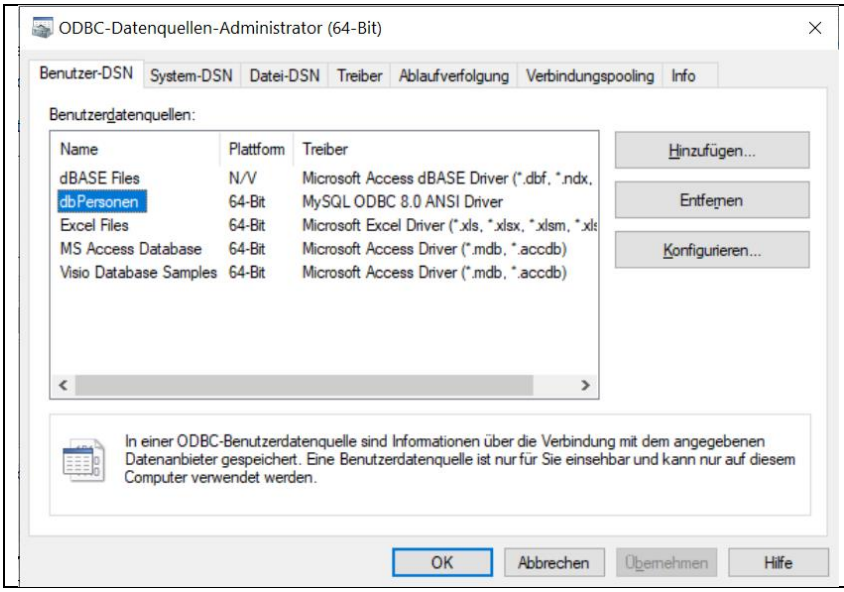

Hinweise:

- ✓ ID ist der Schlüssel und somit eine fortlaufende Nummer
  ➔ PK (Primary Key), nicht null (N) und auto inkrement (AI)
- ✓ Alle Texte sind maximal 45 Zeichen lang
- ✓ Die Verbindung zur Tabelle *tort* erfolgt über einen Integer-Wert Wohnort, der die ID des Wohnortes beinhaltet, in welchem die Person wohnt.

# 4 ODBC-CONNECTOR

Um die MySQL-Datenbank von außen zugänglich zu machen, wird der ODBC-Mechanismus (open database connectivity) verwendet.

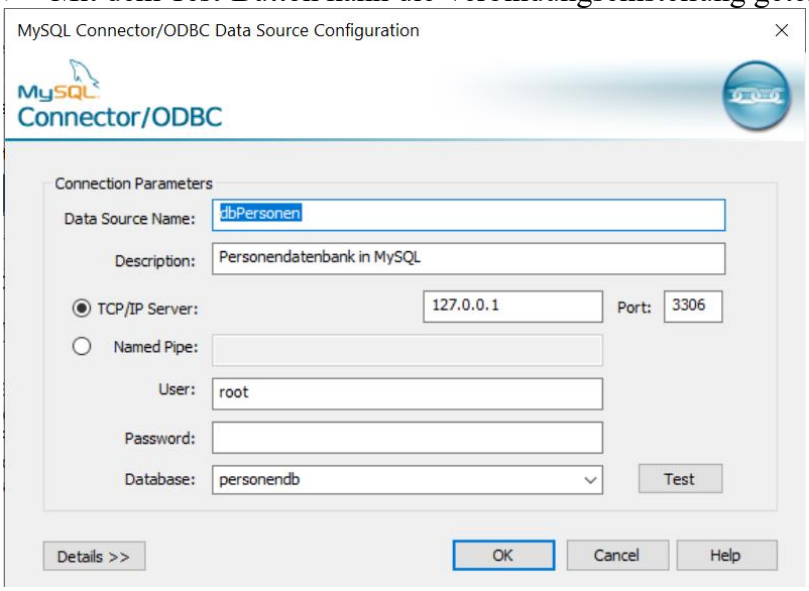
Start des ODBC-Manager
Windows+R ➔ odbcad32




Achtung:
da alle MySQL-Programme 64Bit Applikationen sind, muss auch der ODBC-Manager eine 64Bit Applikation sein. Dies wird im Titel des Managers angezeigt (auch wenn er als 32bit Applikation gestartet wurde!)

Durch *Hinzufügen* kann ein neuer ODBC-Connector erstellt werden.

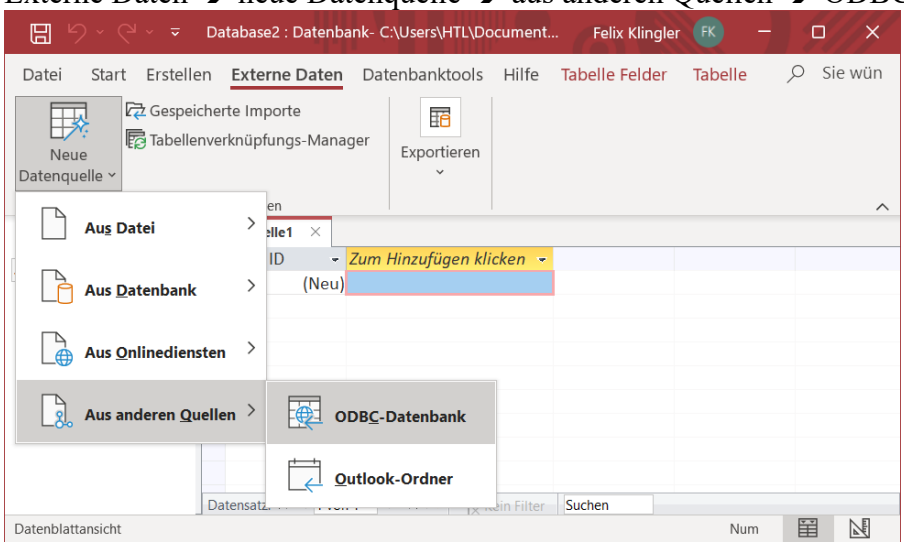
- Verbindung *dbPersonen* (wie schon bei der Anmeldung der MySQL-Workbench)
- Internetadresse 127.0.0.1 (eigener Rechner)
- Mit dem Test-Button kann die Verbindungseinstellung getestet werden.



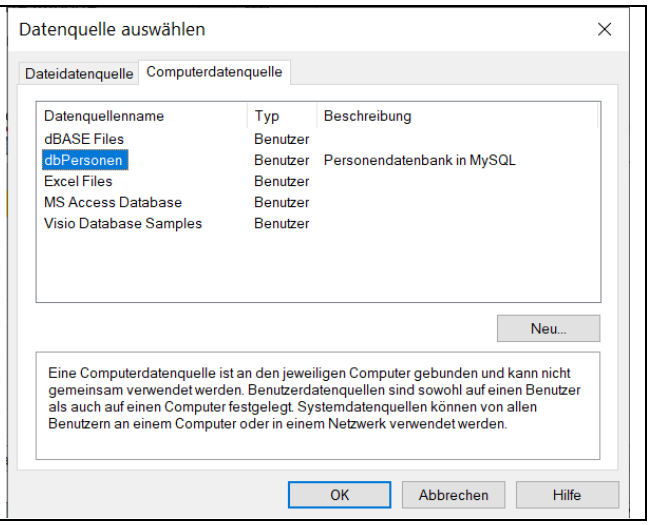

# 5 VERBINDUNGSTEST UND DATENEINGABE ÜBER MSACCESS

Starte MSAccess, erstelle eine leere Datenbank und verbinde diese über den ODBC-Connector mit der MySQL-Datenbank.

- Externe Daten ➔ neue Datenquelle ➔ aus anderen Quellen ➔ ODBC-Datenbank

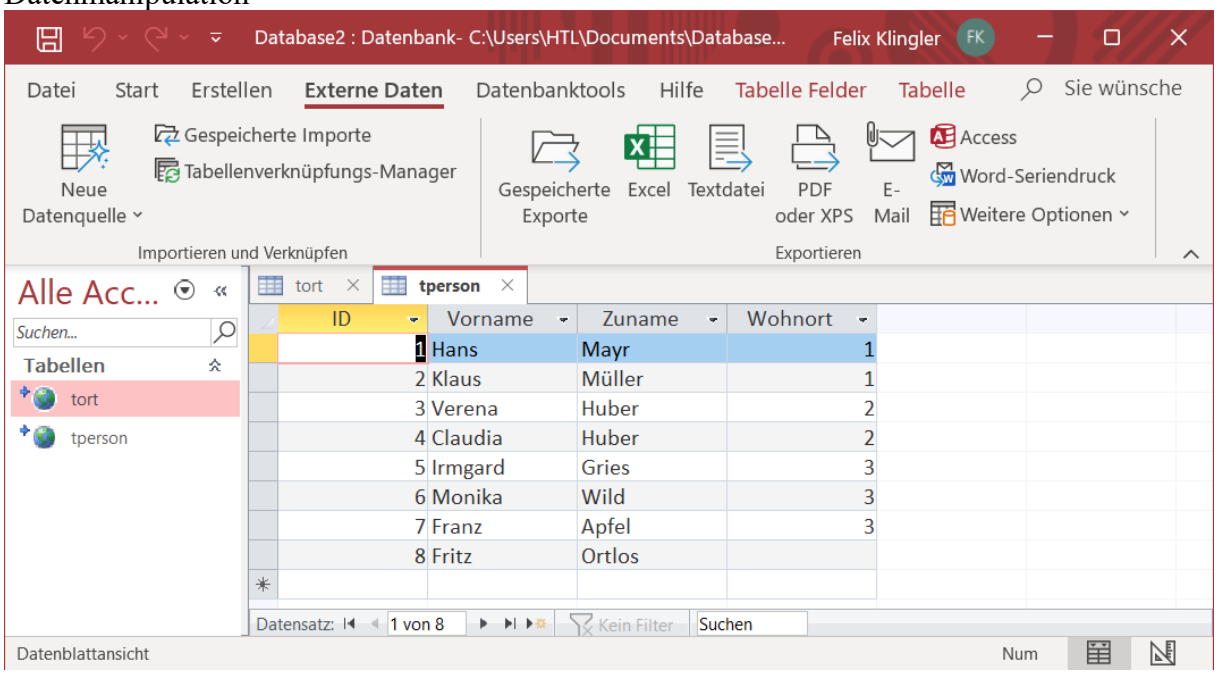


- ODBC-Connector

| | |
|---|---|
| Anbindung erfolgt über die Verknüpfung zur ODBC-Datenbank. Somit bleiben Daten und Tabellenstruktur in MySQL. Die Daten können nun von Access manipuliert werden. Eine Veränderung der Tabellenstruktur ist nicht möglich. |  |



- Tabellenverknüpfung
  beide Tabellen *tort* und *tperson* auswählen
- Datenmanipulation




Beide Tabellen sind extern (Symbol Weltkugel) und können, wie bei jeder Access-Tabelle mit Daten befüllt werden.
Achtung: Die Speicherung der Daten erfolgt weiterhin in MySQL – Access übernimmt nur die Visualisierung.

# 6 ZUGRIFF MITTELS C#

Um mit dem ODBC-Connector zu arbeiten, wird zuerst eine Verbindung über ein `OdbcConnection` Objekt erstellt. Auf Basis dieser Verbindung kann anschließend ein SQL-Statement zum Server geschickt und seine Antwort verarbeitet werden. Hier sind 3 Abfragearten zu berücksichtigen, die im Weiteren erklärt werden.

## 6.1 MYSQL-BIBLIOTHEK

Um MySQL mit/ohne ODBC nutzen zu können, ist die zugehörige Bibliothek zuerst zu installieren und anschließend in der IDE zu aktivieren.

Die Installation ist Teil des im Beginn gezeigten Software-Paketes.

### 6.1.1 ODBC

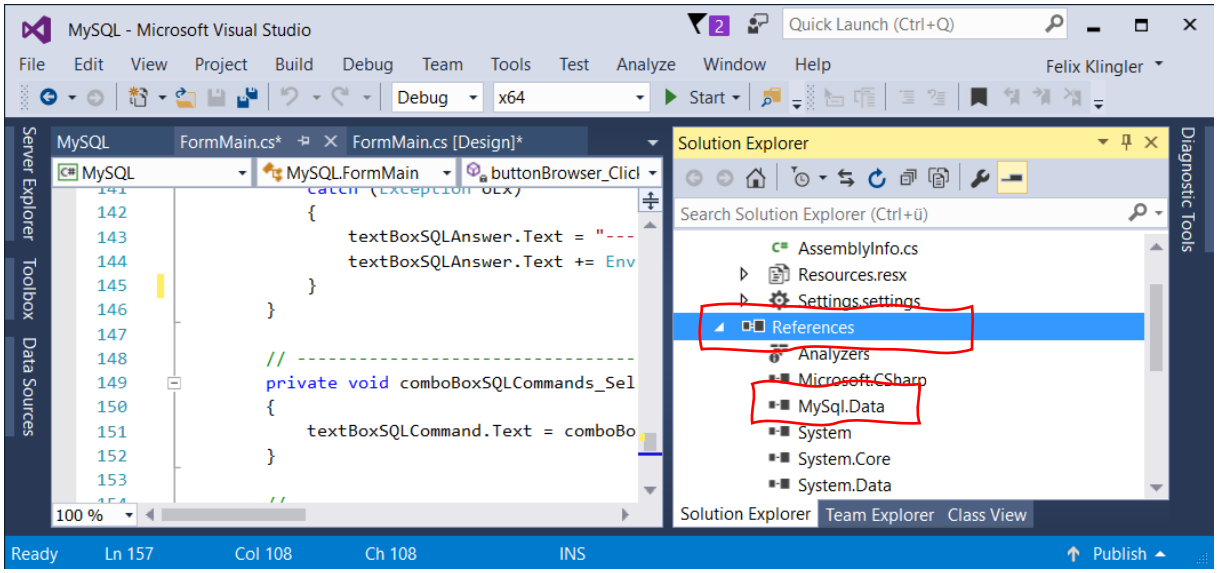
Hierzu ist nur die Direktive

```
using System.Data.Odbc;
```

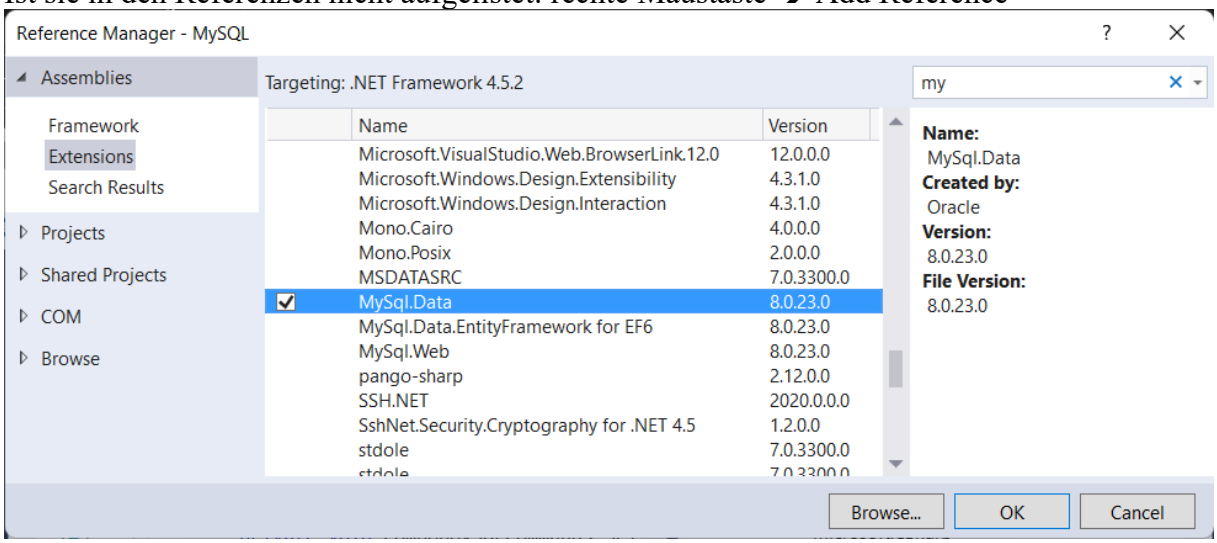
am Beginn des cs-Files anzugeben.
Der Zugriff erfolgt dann über die Klassen `OdbcConnection, OdbcCommand` und `OdbcDataReader` unabhängig vom verwendeten DBMS.

### 6.1.2 direkter Zugriff

Die MySQL- Bibliothek ist im Solution-Explorer bei den Referenzen zu aktivieren.



Ist sie in den Referenzen nicht aufgelistet: rechte Maustaste ➔ Add Reference




Der Zugriff erfolgt dann über die Klassen `MySqlConnection, MySqlCommand` und `MySqlDataReader`, wobei natürlich die Bibliothek am Beginn des c#-Files anzugeben ist.

```
using MySql.Data.MySqlClient;
```

## 6.2 VERBINDUNGSAUF/ABBAU

### 6.2.1 ODBC

Um mittels des vorher erstellten ODBC-Connectors dbpersonen eine Verbindung zum MySQL-Server zu erstellen, sind folgende Befehle nötig:

```
String sConnectionString=("dsn=dbpersonen";
OdbcConnection oODBCConnection;
oODBCConnection = new OdbcConnection(sConnectionString);
oODBCConnection.Open();
[…]
oODBCConnection.Close();
```

Da ein Verbindungsaufbau auch abgelehnt werden kann, ist die Sequenz mittels try/catch Block abzusichern.

Ist kein vorher erstellter ODBC-Connector vorhanden, können die Verbindungsparameter mittels der ODBC- `OdbcConnectionStringBuilder`-Klasse angegeben werden.

```
OdbcConnectionStringBuilder sConnectionString =new OdbcConnectionStringBuilder();
sConnectionString.Add("Uid"   ,"root");
sConnectionString.Add("Pwd"   ,"");
sConnectionString.Add("Server","127.0.0.1");
sConnectionString.Add("Port"  ,"3306");
```

### 6.2.2 direkte Verbindung

Um mittels der MySQL-Klassen auf eine MySQL-Datenbank zuzugreifen, ist eine ähnliche Vorgehensweise nötig.

```
String sConnectionString;
sConnectionString = "server=127.0.0.1;";
sConnectionString += "database=personendb;";
sConnectionString += "uid=root;";
sConnectionString += "password=;";

oMySQLConnection = new MySqlConnection(sConnectionString);
oMySQLConnection.Open();
[…]
oMySQLConnection.Close();
```

Auch hier ist ein möglicher Fehlerfall mittels try/catch abzufangen.

## 6.3 NonQuery Abfage

Diese Abfrage liefert kein Ergebnis zurück.

```
OdbcCommand oQuery;
int iAntwort;

oQuery = new OdbcCommand("insert into tperson (vorname,zuname,ort) values
("Klaus","Huber",2);",oODBCConnection);
iAntwort = Convert.ToInt32(oQuery.ExecuteNonQuery());
```

Sie kann zum Beispiel für Insert, Delete oder Update-Befehle verwendet werden.

Wird direkt zugegriffen, verwendet man statt der Klasse `OdbcCommand` eben die Klasse `MySqlCommand`.

## 6.4 Scalare Abfrage

Diese Abfrage liefert das Ergebnis der SQL-Berechnung, etwa die Anzahl der Zeilen einer SQL-Abfrage

```
OdbcCommand oQuery;
int iAntwort;

oQuery = new OdbcCommand("select count(*) from tort", oODBCConnection);
iAntwort = Convert.ToInt32(oQuery.ExecuteScalar());
```

Wird direkt zugegriffen, verwendet man statt der Klasse `OdbcCommand` eben die Klasse `MySqlCommand`.

## 6.5 ABFRAGE MIT EIN/MEHRZEILIGER ANTWORT

Diese Abfrage liefert das Ergebnis einer Select-Abfrage. Das Ergebnis wird Zeile für Zeile abgearbeitet.

```
OdbcCommand oQuery;
OdbcDataReader oDataReader;
String sDataType;
String sZeile;
int iZeile;

// Abfrage stellen
oQuery = new OdbcCommand("select * from tort", oODBCConnection);
oDataReader = oQuery.ExecuteReader();

// Antwort verarbeiten
while (oDataReader.Read ())
{
	[…]
}
```

Auf die einzelnen Spalten (und davon gibt es genau oDataReader.FieldCount) kann mit der Spaltennummer (iIndex) zum Beispiel wie folgt zugegriffen werden:

```
oDataReader.GetInt32(iIndex); // Auslesen einer 32Bit-Zahl (integer) der Spalte iIndex
oDataReader.GetInt64(iIndex); // Auslesen einer 64Bit-Zahl (bigint) der Spalte iIndex
oDataReader.GetString(iIndex);// Auslesen eines Strings (varchar) der Spalte iIndex
```

Ist ein Eintrag möglicherweise leer (null), ist dies vor dem Zugriff abzufangen:

```
if (oDataReader.IsDBNull(iIndex))
{
	iData = 0;
}
else
{
	iData = oDataReader.GetInt32(iIndex);
}
```

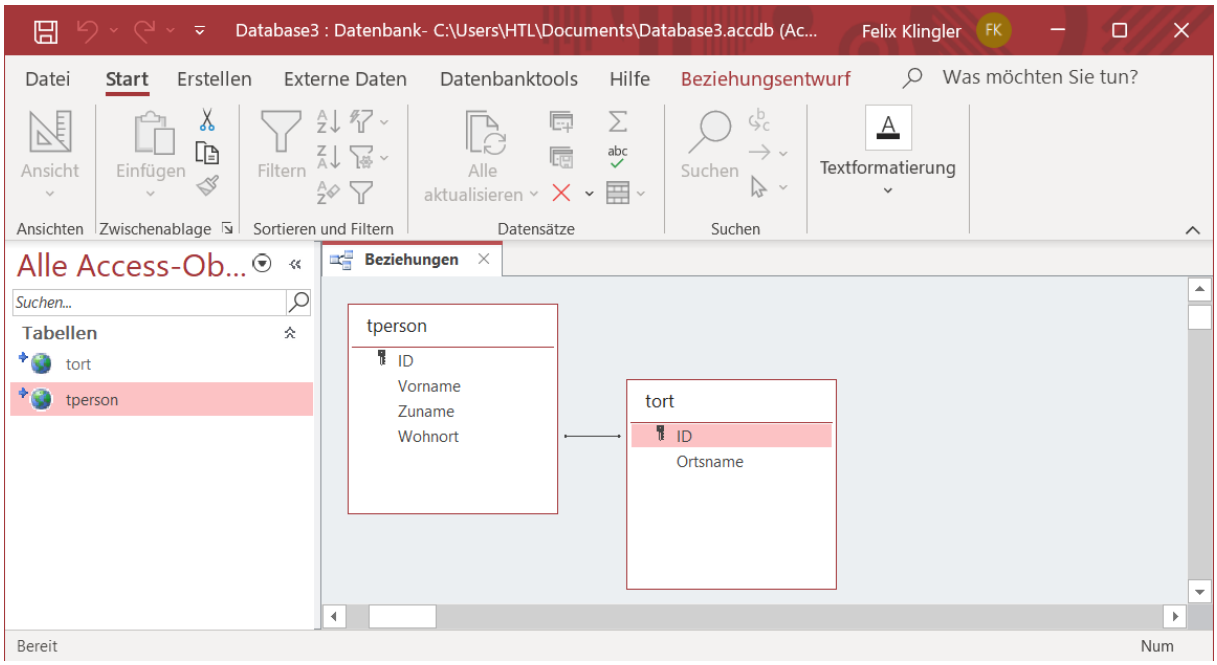
Auch hier ist der Zugriff immer mit try/catch abzusichern.

Wird direkt zugegriffen, verwendet man statt der Klasse `OdbcCommand` bzw. `OdbcDataReader` eben die Klasse `MySqlCommand` bzw. `MySQLDataReader`.

# 7 AUFGABENSTELLUNG

Erstelle in c# eine Datenbankanbindung mittels ODBC an eine MySQL-Datenbank.
In einem Textfeld sollen SQL-Statements eingegeben werden, die dann mittels entsprechender Abfrage-Type an die Datenbank weitergeleitet und deren Ergebnis in einem weiteren Textfeld angezeigt wird.

Die der Anbindung zugrunde liegende Datenbank besteht aus 2 Tabellen – tort und tperson



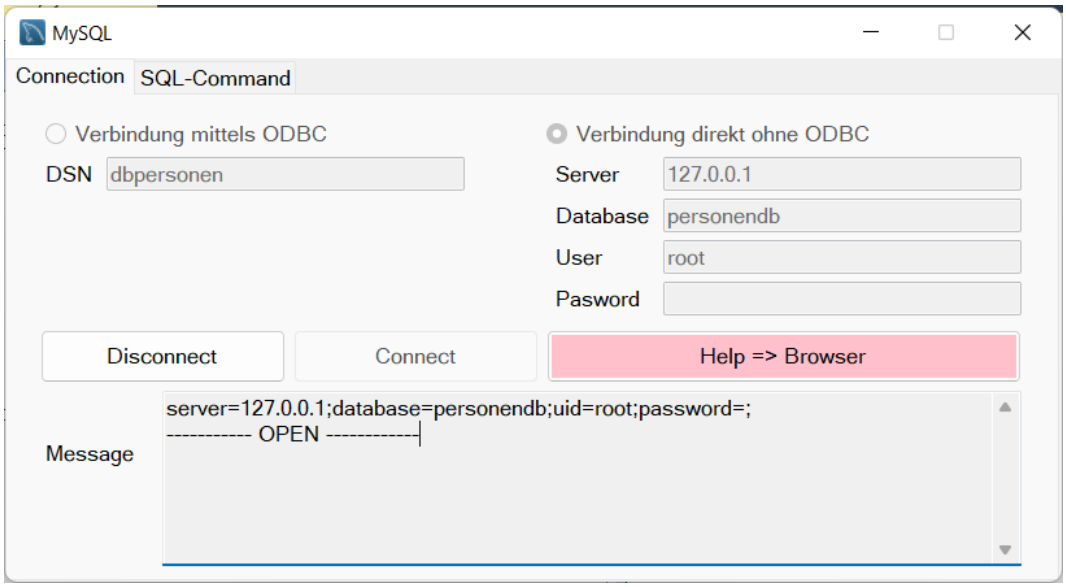

tperson
- id: integer
- Vorname, Zuname: varchar (45)
- Wohnort: integer

tort
- id: integer
- Ortsname: varchar(45)

*Relation: tperson.Wohnort=tort.id*

Für den Verbindungsauf/abbau und das Senden und Auswerten von SQL-Anfragen ist eine entsprechende GUI zu erstellen.

Verbindungsauf/abbau



SQL-Kommandos