

## Datenbankanbindung an einer grafischen Oberfläche

Einleitung.....	2
<b>1 Erstellen einer Datenbank im phpMyAdmin mithilfe von XAMPP</b>	
1.1 Installation von XAMPP.....	3
1.2 Öffnen des XAMPP Control-Panels.....	6
1.2.1 Starten des Apache-Servers.....	6
1.2.2 Starten des MySQL-Servers.....	7
1.3 Zugriff auf den MariaDB-Server im Internetbrowser.....	7
1.4 Anlegen einer Datenbank in phpMyAdmin.....	8
1.4.1 Festlegen eines Datenbanknamens und der Kollation.....	8
1.4.2 Erzeugen einer Datenbanktabelle mit festgelegter Spaltenanzahl.....	8
1.4.3 Festlegen von Attributen mit Daten.....	9
1.4.4 Hinzufügen eines Primärschlüssels.....	10
1.4.5 Einfügen von Attributwerten.....	11
1.4.6 Abbildung einer fertigen Datenbanktabelle.....	11
1.4.7 Importieren von Datenbanken sowie Datenbanktabellen.....	12
1.4.8 Exportieren einer Datenbank.....	13
<b>2 Einbinden einer Datenbank in das System</b>	
2.1 Installation eines MySQL-ODBC-Treibers.....	14
2.2 Erstellen einer ODBC-Datenquelle.....	15
2.2.1 Einfügen des MySQL-ODBC-ANSI-Treibers in die Datenquelle.....	15
2.2.2 Konfiguration einer Datenquelle.....	16
2.2.3 Verbindungstest mit dem System.....	16
2.2.4 Benutzerdatenquelle mitsamt integrierter Datenbank.....	16
<b>3 Erstellen einer Lazarus-GUI mit eingebundener Datenbank</b>	
3.1 Installation von Lazarus.....	17
3.2 Anfertigen des Formulars und Anbinden der Datenbank.....	18
3.3 Implementation des Programmcodes in Pascal.....	24
3.4 Ausführen des Programms und Testen von SQL-Abfragen.....	26
3.5 Beenden des Programms und mögliche Fehlermeldungen.....	28

## Einleitung

Es werden in vielen Einrichtungen und Organisationen Daten sowie Informationen über bestimmte Personen sowie Ereignisse immer häufiger gebraucht. Da es zurzeit kaum jemand analog mit Papier und Stift macht und die Daten außerdem in großen Mengen vorhanden sind sowie eine hohe Anzahl von Werten enthalten, werden sie auf Computern in Tabellen übertragen. Aus diesen Tabellen entsteht dann schließlich eine Datenbank, wo in jeder Tabelle die Teile der dort gespeicherten Informationen detailliert aufgeteilt sind. Jeder, der sich noch nicht mit der Informationstechnologie beschäftigt hat, weiß nicht, dass man eine Datenbank auf einem lokalen Server in einem Internetbrowser erstellen kann und diese dann eventuell an ein Programm anbindet. Das folgende Projekt soll als Anleitung dazu dienen, wie das alles ausgeführt werden kann. Mögliche Beispiele dazu sind Bewertungstabellen oder sportliche Ergebnisse.

Als Beispiel für eine Datenbank wählte ich die Fußball-Bundesligasaison 2019/2020, da in der Zeit, wo ich mit meinem Projekt begonnen habe, die Saison 2020/2021 noch nicht zu Ende war. Fußballdaten enthalten häufig mehrere Tabellen. Die wichtigste davon ist natürlich die Abschlusstabelle u. a. mit den Endplatzierungen und der Anzahl der Siege, Unentschieden und Niederlagen sowie der Punktzahl. Weitere Tabellen stellen den direkten Vergleich aller Mannschaften miteinander (Kreuztabelle), sowie die Liste von Torschützen, Vorlagengebern und den Punktesammlern (Scorerliste → 1 Tor bzw. 1 Torvorlage = 1 Punkt). Eine sechste Tabelle stellt das Ergebnis der Relegationsspiele (Platz 16 der Bundesliga gegen Platz 3 der 2. Liga) dar. Dafür wird erstmal XAMPP benötigt, eine Software zur Installation des Apache-Webservers und MySQL-Servers sowie zur Konfiguration des MariaDB-Servers für die Datenbank mit der Webanwendung phpMyAdmin, welche in der Skriptsprache PHP erstellt wurde. PHP wird in diesem Projekt zwar nicht benötigt, aber es gibt die Abfragesprache SQL, die für das Erstellen, Bearbeiten, Hinzufügen, Löschen sowie Anzeigen von Daten verwendet wird.

Die Datenbank soll dann mithilfe eines MySQL-ODBC-Treibers an eine in der Entwicklungsumgebung Lazarus erstellte Anwendung mit grafischer Benutzeroberfläche, welche in der Programmiersprache Pascal implementiert wird, angebunden werden. In ein großes Eingabefeld können SQL-Datenbankabfragen eingegeben werden und in einer extra dafür eingebauten Datenbanktabelle (in der grafischen Oberfläche) wird ein Ausschnitt mit den betroffenen Daten angezeigt, falls die Abfrage syntaktisch korrekt gestellt wurde, sonst erscheint eine Fehlermeldung.

# 1 Erstellen einer Datenbank im phpMyAdmin mithilfe von XAMPP

## 1.1 Installation von XAMPP

- Links zum Downloaden:

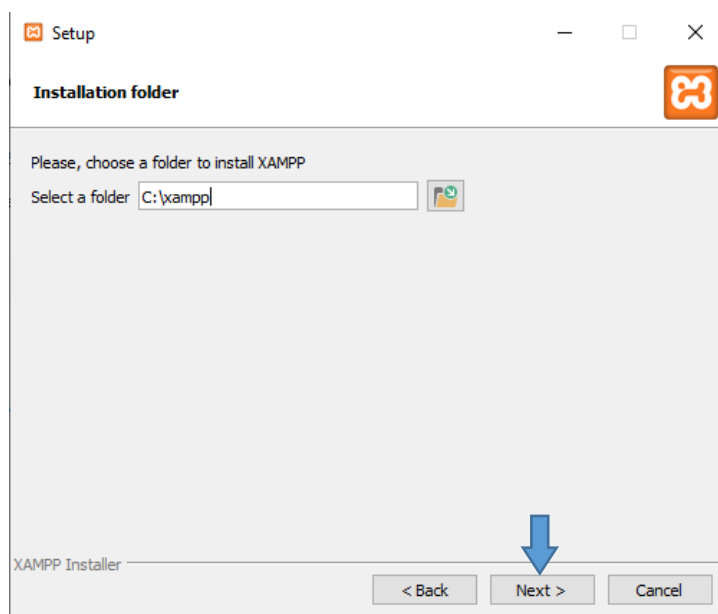
<https://www.heise.de/download/product/xampp-10929/download> (für Windows)

<https://webdev-training.de/tutorials/xampp/so-installierst-du-xampp-auf-dem-mac>  
(für macOS)

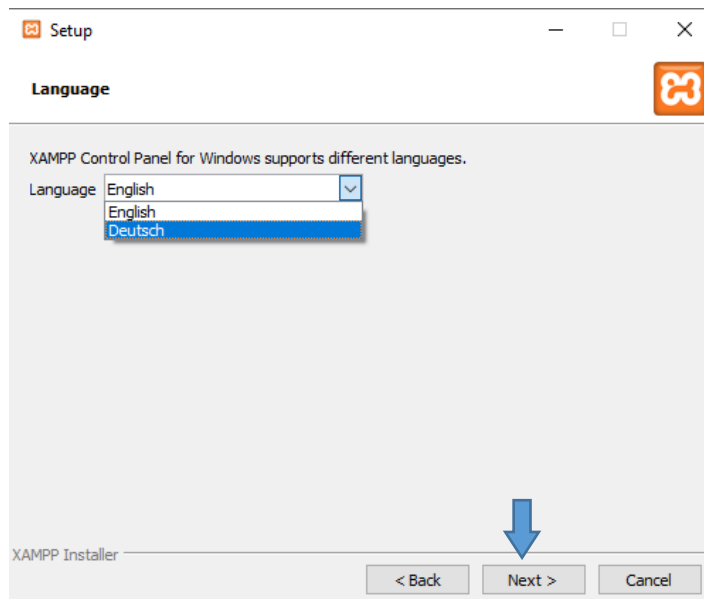
Während der Installation von XAMPP soll folgendes beachtet werden:

1. Ordner für die Installation: `C:\xampp`, anschließend Klick auf *Next*

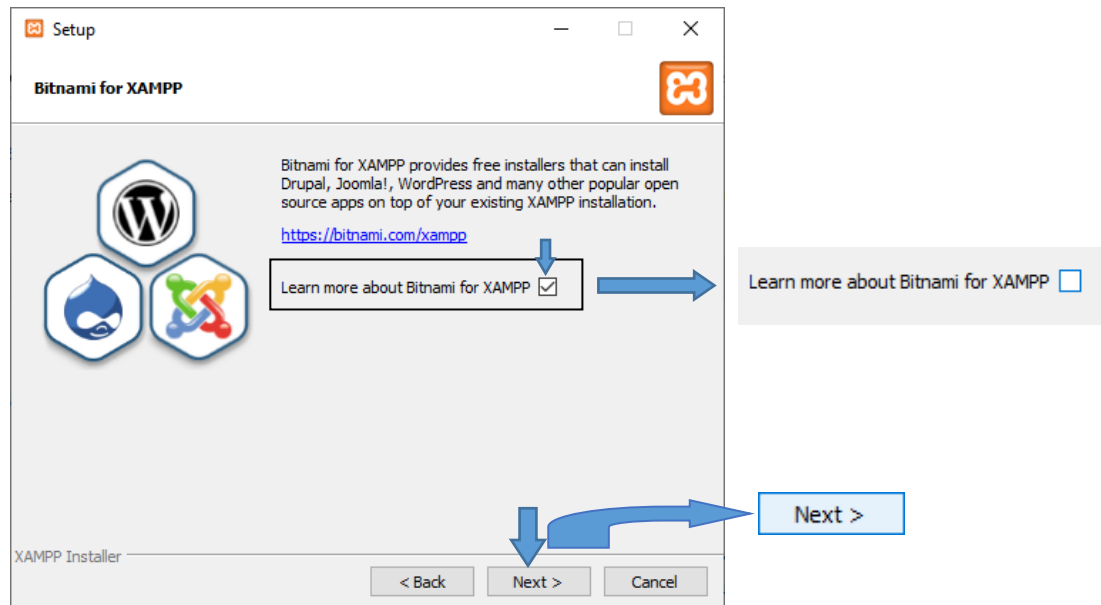
(Beim vorherigen Schritt soll einfach auf den Next-Button geklickt werden.)



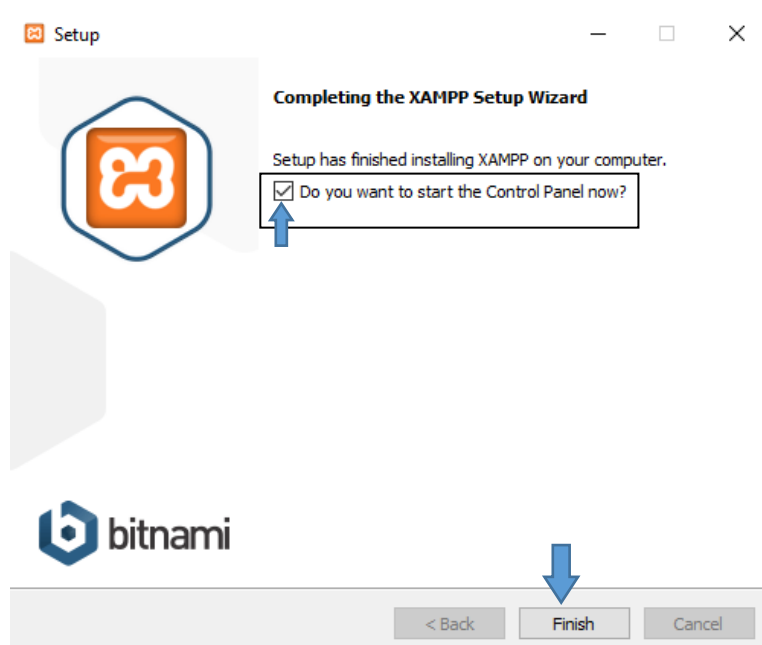
2. Danach kann man zwischen der englischen sowie deutschen Sprache wählen.



3. Im nächsten Arbeitsschritt soll beachtet werden, dass das Häkchen neben dem Label „*Learn more about Bitnami for XAMPP*“ entfernt werden soll, damit keine überflüssige Installation von Bitnami stattfindet. Darauf folgt ein erneuter Klick auf die Schaltfläche *Next* und es erscheint ein Hinweisfenster, wo drinsteht, dass die Installation stattfinden kann, nachdem man noch einmal auf *Next* klickt.

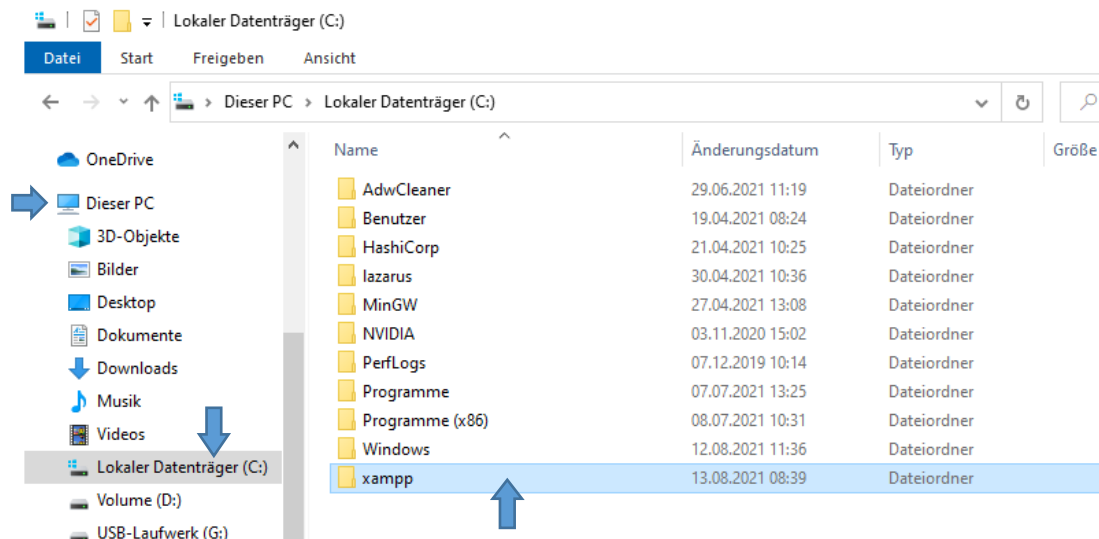


4. Anschließend findet die Installation statt. Obwohl im darauffolgenden Fenster bei der Frage zum sofortigen Start des XAMPP Control-Panels ein Häkchen gesetzt ist, könnte man es eventuell entfernen, um es dann stattdessen vom Dateipfad des Panels aus (*C:\xampp\xampp-control.exe*) über den Windows-Explorer zu starten. Mit einem Klick auf *Finish* ist die Installation beendet.



Der Dateipfad des XAMPP Control-Panels ist auf dem PC im Windows-Explorer folgendermaßen zu erreichen:

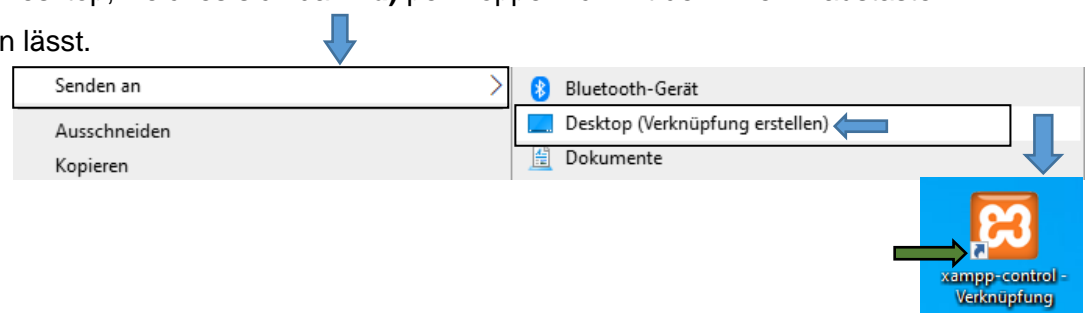
1. Man wählt im Verzeichnis *Dieser PC* den *lokalen Datenträger (C:)*, welcher die (Haupt-)Festplatte darstellt. Anschließend wird dort der Ordner *xampp* geöffnet.



2. Im Ordner *xampp* soll der Scroll-Balken bis ganz nach unten gescrollt werden, wo die Anwendung *xampp-control.exe* als drittletzte Datei zu finden ist.

	xampp-control	05.06.2019 14:10	Anwendung	3.290 KB
--	---------------	------------------	-----------	----------

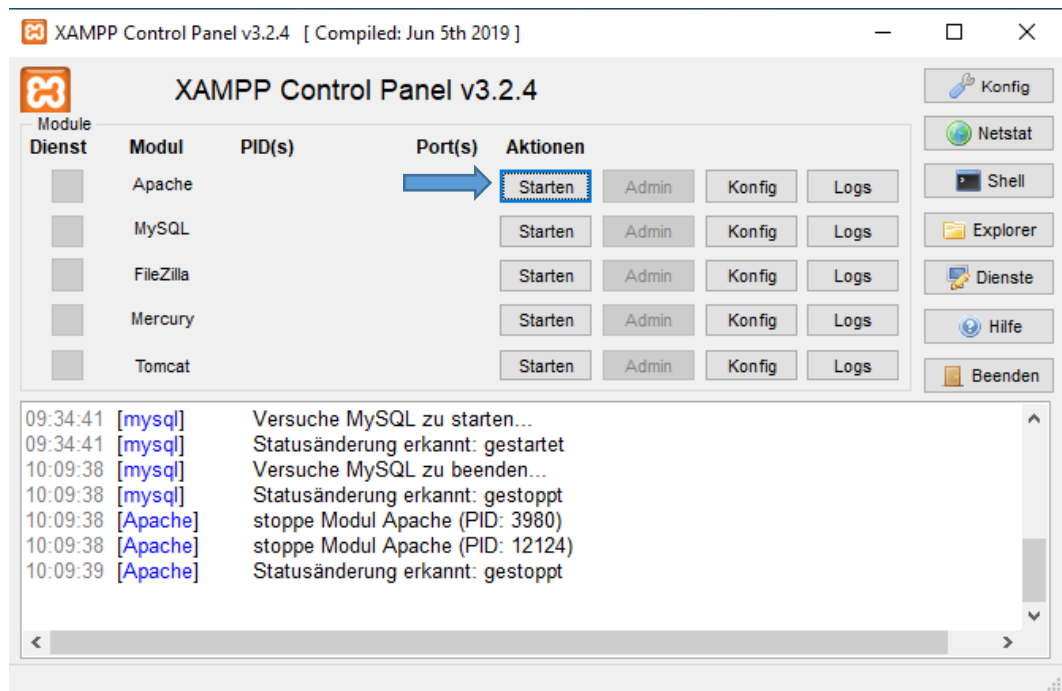
3. Um auf dem Desktop eine Verknüpfung zum XAMPP Control-Panel zu erstellen, klickt man **a)** mit der rechten Maustaste auf *xampp-control.exe*. Es erscheint ein Kontextmenü, wo die Option *Senden an* mit einem Pfeil rechts neben sich ausgewählt werden soll, indem man einfach **b)** die Maus auf den Menüpunkt richtet ohne darauf zu klicken, sodass sich rechts vom Kontextmenü ein Untermenü öffnet. Dort soll anschließend **c)** die Option *Desktop (Verknüpfung erstellen)* mit der linken Maustaste angeklickt werden und es erscheint ein Verknüpfungssymbol für das XAMPP Control-Panel auf dem Desktop, welches sich dann **d)** per Doppelklick mit der linken Maustaste öffnen lässt.



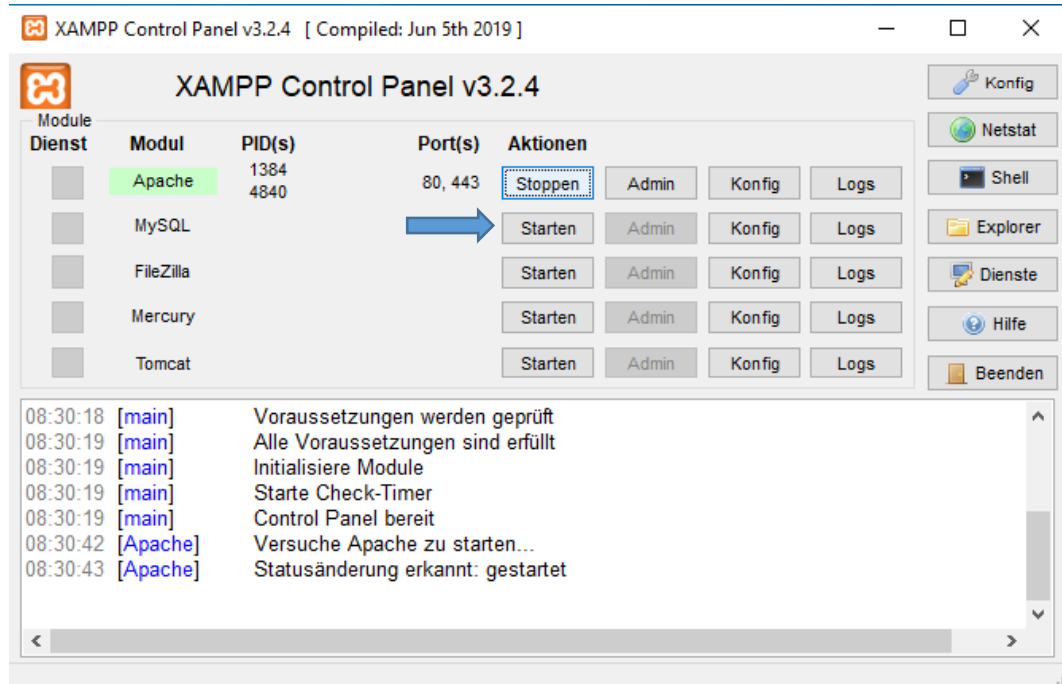
## 1.2 Öffnen des XAMPP Control-Panels

### 1.2.1 Starten des Apache-Servers

- Klick auf die Aktion *Starten* im Apache-Modul (1. Zeile)



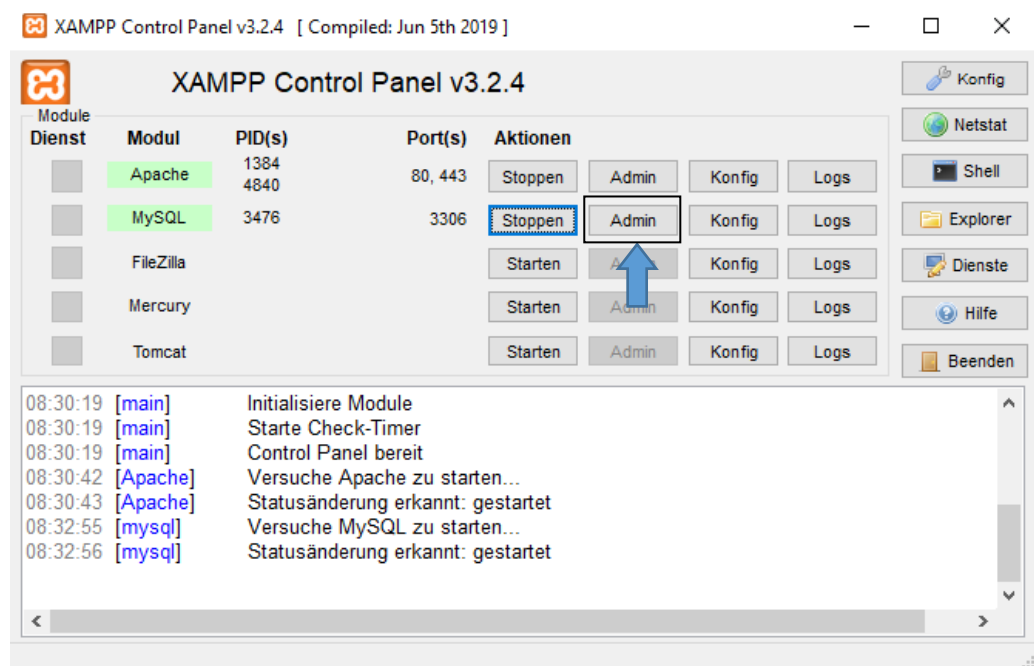
### 1.2.2 Starten des MySQL-Servers (Klick auf *Starten* bei MySQL – Zeile 2)



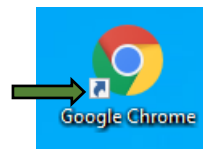
- Ports (4. Spalte): 80 – HTTP, 443 – HTTPS
- PIDs (**P**rocess **I**dentifiers) immer 4-stellig, jederzeit anders (3. Spalte)
- 3306 (Zeile 2, Spalte 4) – Port für MySQL, nur ein PID

### 1.3 Zugriff auf den MariaDB-Server im Internetbrowser

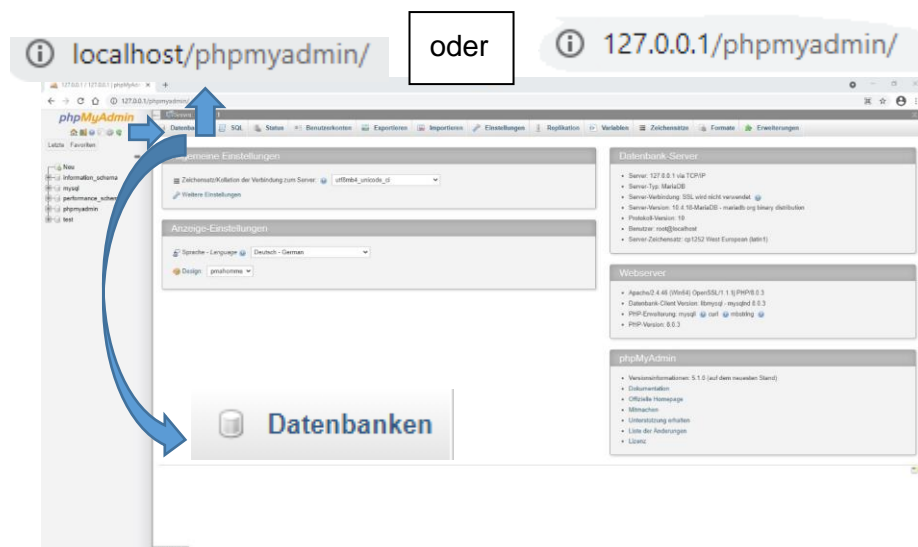
- a) Öffnen des Webbrowsers durch Klick auf *Admin* in der MySQL-Spalte (einfache, effektive Variante)



- b) Eingabe der lokalen Host-IP-Adresse sowie phpMyAdmin in den Browser: (umständlichere Variante)



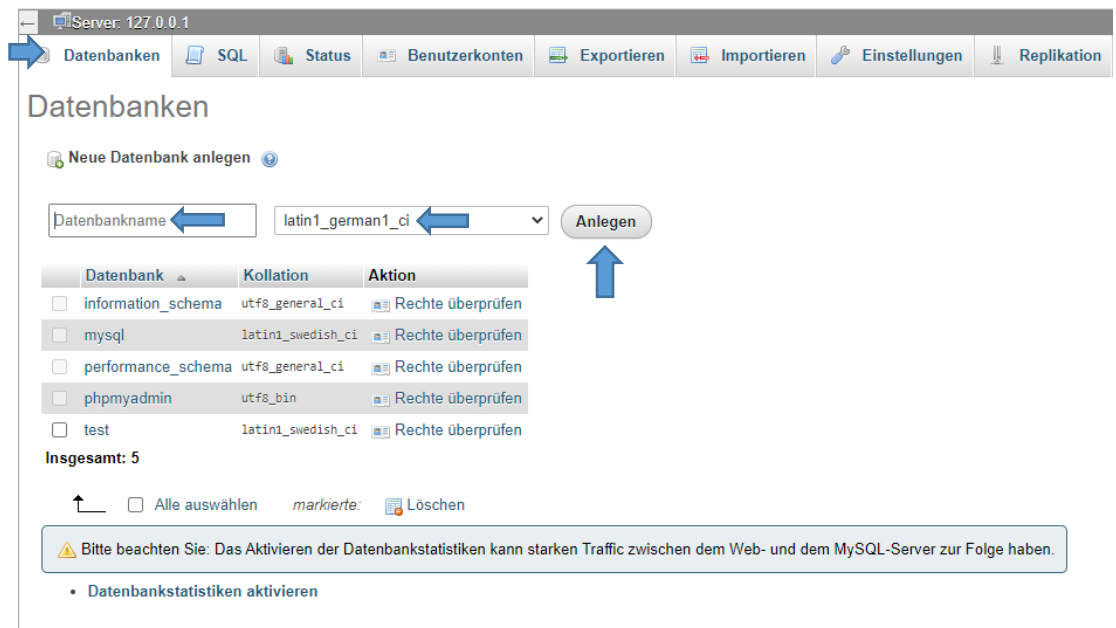
Man öffnet einen Webbrowser (z. B.: Google Chrome) manuell und gibt in dessen Adresszeile den Domännennamen (localhost) oder die lokale IP-Adresse (127.0.0.1) und anschließend */phpmyadmin* ein, um auf den MariaDB-Server zugreifen zu können.



## 1.4 Anlegen einer Datenbank in phpMyAdmin

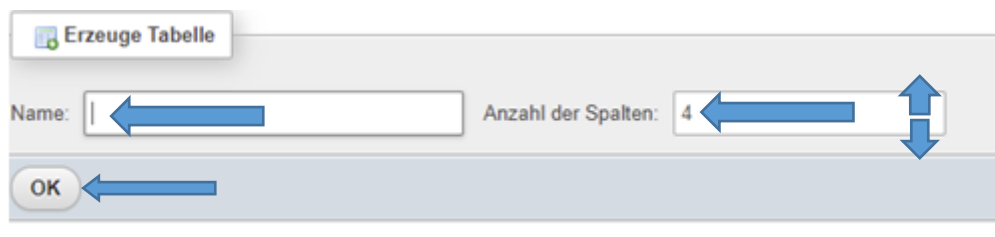
### 1.4.1 Festlegen eines Datenbanknamens und der Kollation

- Klick auf Registerkarte *Datenbanken*
- Datenbankname: *Fussball\_Bundesliga\_2019\_20*
- Kollation: *latin\_german1\_ci*
- Klick auf Schaltfläche *Anlegen*



### 1.4.2 Erzeugen einer Datenbanktabelle mit festgelegter Spaltenanzahl

- Tabellenname: *Abschlusstabelle*
- Anzahl der Spalten: 9
- Klick auf *OK*





### 1.4.3 Festlegen von Attributen mit Daten

Name	Typ	Länge/Werte	Standard
Platz	INT	2	Kein(e)
Aus zentralen Spalten wählen			
Verein	VARCHAR	30	Kein(e)
Aus zentralen Spalten wählen			
Spiele	INT	2	Wie definiert:
Aus zentralen Spalten wählen			
			34
Siege	INT	2	Kein(e)
Aus zentralen Spalten wählen			
Unentschieden	INT	2	Kein(e)
Aus zentralen Spalten wählen			
Niederlagen	INT	2	Kein(e)
Aus zentralen Spalten wählen			
Tore_und_Gegentore	VARCHAR	6	Kein(e)
Aus zentralen Spalten wählen			
Tordifferenz	INT	3	Kein(e)
Aus zentralen Spalten wählen			
Punkte	INT	2	Kein(e)
Aus zentralen Spalten wählen			
<b>Struktur</b>			
<input type="button" value="SQL Vorschau"/> <input type="button" value="Speichern"/>			

INT ist der Datentyp für ganzzahlige Werte und VARCHAR ist der Datentyp für Zeichenketten aus Buchstaben, Ziffern und Sonderzeichen aus der beim Erstellen der gesamten Datenbank ausgewählten Kollation *latin1\_german\_ci* (Zeichensatz für die deutsche Sprache). Am Ende wird das Erstellen der Datenbanktabelle mit einem Klick auf die Schaltfläche „Speichern“ beendet. Da jeder Verein am Ende der Saison jeweils 34 Spiele absolviert haben muss (Insgesamt hat die deutsche 1. Bundesliga 18 Vereine, welche gegen jeden der 17 Gegner jeweils ein Hin- und Rückspiel bzw. ein Heim- und Auswärtsspiel zu absolvieren haben und damit 2 Mal gegen jeden Gegner spielen müssen.), wird für das Attribut bzw. die Spalte „Spiele“ (Anzahl der Spiele) ein Wert von 34 vordefiniert. Mit der

Länge der Werte (3. Spalte in der Abbildung) wird die maximale Zeichenanzahl für die Attributwerte (alle Zellen außerhalb der Kopfzeile) festgelegt.

#### 1.4.4 Hinzufügen eines Primärschlüssels

Server: 127.0.0.1 » Datenbank: fussball\_bundesliga\_2019\_20 » Tabelle: abschlusstabelle

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Rechte Operationen N

Tabellenstruktur Beziehungsansicht

#	Name	Typ	Kollation	Attribute	Null	Standard
<input type="checkbox"/>	1 Platz	int(2)			Nein	kein(e)
<input type="checkbox"/>	2 Verein	varchar(30)	latin1_german1_ci		Nein	kein(e)
<input type="checkbox"/>	3 Spiele	int(2)			Nein	34
<input type="checkbox"/>	4 Siege	int(2)			Nein	kein(e)
<input type="checkbox"/>	5 Unentschieden	int(2)			Nein	kein(e)
<input type="checkbox"/>	6 Niederlagen	int(2)			Nein	kein(e)
<input type="checkbox"/>	7 Tore_und_Gegentore	varchar(6)	latin1_german1_ci		Nein	kein(e)
<input type="checkbox"/>	8 Tordifferenz	int(3)			Nein	kein(e)
<input type="checkbox"/>	9 Punkte	int(2)			Nein	kein(e)

☐ Alle auswählen
 markierte:
 ☐ Anzeigen
 ☐ Bearbeiten
 ☐ Löschen
 ☒ Primärschlüssel
 ☐ Unique
 ☐ Index
 ☐ Räum

Um die Relation (Tabelle) eindeutig zu sortieren, wird ein Primärschlüssel benötigt. Als solcher wird in der Regel ein Attribut mit ganzzahligen Werten festgelegt. In diesem Beispiel in Form der Relation „Abschlusstabelle“ sowie in allen anderen Tabellen außer der Tabelle „Relegation“ ist es das Attribut „Platz“ mit dem Datentyp INT, wo die Punkte für die Ordnungszahlen in den Attributwerten dadurch weggelassen werden müssen. In der Tabelle „Relegation“ ist es das Datum, wo die Spiele stattgefunden haben („Spiele“), welchem der Datentyp VARCHAR zugewiesen wurde, damit es genauso dargestellt wird, wie in der Vorgabe, denn mit dem Datentyp DATE wäre es umständlicher gewesen, da man dort noch das Format für das Datum einstellen müsste. In den VARCHAR-Zeichenketten werden die Werte, die mit einer Ziffer beginnen, in deren Reihenfolge aufsteigend (entsprechend der alphabetischer Sortierung der aus Buchstaben bestehenden Zeichenketten) sortiert.

<input checked="" type="checkbox"/>	1 Platz	int(2)		<input type="checkbox"/>	1 Platz	
-------------------------------------	---------	--------	--	--------------------------	---------	--

Man wählt das Attribut „Platz“ aus, indem man im kleinen Quadrat links davor ein Häkchen setzt und klickt dann auf den Link „Primärschlüssel“.

- Primärschlüssel für die Tabelle „Relegation“:

Name	Typ	Kollation
Spiele	varchar(12)	latin1_german1_ci

- Spalte des Attributs „Spiele“ in der Tabelle „Relegation“:

Spiele
2. Juli 2020
6. Juli 2020
Gesamt:

#### 1.4.5 Einfügen von Attributwerten

- Klick auf Registerkarte *Einfügen*



**Quelle:** [https://de.wikipedia.org/wiki/Fu%C3%9Fball-Bundesliga\\_2019/20](https://de.wikipedia.org/wiki/Fu%C3%9Fball-Bundesliga_2019/20)

Server: 127.0.0.1 » Datenbank: fußball\_bundesliga\_2019\_20 » Tabelle: abschlusstabelle

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren

Spalte	Typ	Funktion	Null	Wert
Platz	int(2)			
Verein	varchar(30)			
Spiele	int(2)			34
Siege	int(2)			
Unentschieden	int(2)			
Niederlagen	int(2)			
Tore_und_Gegentore	varchar(6)			
Tordifferenz	int(3)			
Punkte	int(2)			

Es werden nur in die 5. Spalte „Wert“ Daten eingegeben, die den Attributen in der 1. Spalte entsprechen.






(z. B.: Platz – 3; Verein – RB\_Leipzig; Spiele – 34; Siege – 18; Unentschieden – 12; Niederlagen – 4; Tore\_und\_Gegentore – 81:37; Tordifferenz – 44; Punkte – 66).

Bei den Bezeichnungen dürfen insgesamt nur 63 Zeichen verwendet werden (genauso wie in der Programmierung):

26 Groß- sowie Kleinbuchstaben des englischen Alphabets (A – Z bzw. a – z), alle zehn Ziffern aus dem dezimalen Zahlensystem, welches das mit Abstand gewöhnlichste und am häufigsten im Alltag verwendete Zahlensystem ist (0 - 9), sowie der Unterstrich (\_) als einziges Sonderzeichen, ansonsten keine weiteren Zeichen (**einschließlich Leerzeichen**).

Es sollen nur für die Abschlusstabelle die Werte eingefügt werden, der Rest soll danach importiert werden.

Man kann zwar eine beliebige Anzahl von Datensätzen (Tupeln) auswählen. Für die Relation *Abschlusstabelle* werden 18 Tupel benötigt, aber irgendwie ist es so, dass maximal 14 Tupel mit den jeweiligen Attributwerten auf einmal gespeichert werden können und somit muss der Vorgang noch mal ausgeführt werden, dieses Mal aber mit den 4 verbliebenen Datensätzen.

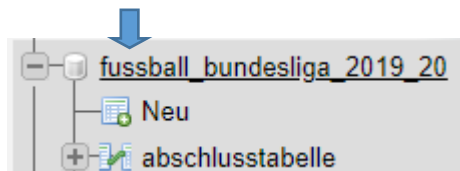

 Einfügen mit  
 Datensätzen fortfahren 
 


#### 1.4.6 Abbildung einer fertigen Datenbanktabelle

Platz	Verein	Spiele	Siege	Unentschieden	Niederlagen	Tore_und_Gegentore	Tordifferenz	Punkte
1	FC_Bayern_Muenchen	34	26	4	4	100:32	68	82
2	Borussia_Dortmund	34	21	6	7	84:41	43	69
3	RB_Leipzig	34	18	12	4	81:37	44	66
4	Borussia_Moenchengladbach	34	20	5	9	66:40	26	65
5	Bayer_04_Leverkusen	34	19	6	9	61:44	17	63
6	TSG_1899_Hoffenheim	34	15	7	12	53:53	0	52
7	VfL_Wolfsburg	34	13	10	11	48:46	2	49
8	SC_Freiburg	34	13	9	12	48:47	1	48
9	Eintracht_Frankfurt	34	13	6	15	59:60	-1	45
10	Hertha_BSC	34	11	8	15	48:59	-11	41
11	1_FC_Union_Berlin	34	12	5	17	41:58	-17	41
12	FC_Schalke_04	34	9	12	13	38:58	-20	39
13	1_FSV_Mainz_05	34	11	4	19	44:65	-21	37
14	1_FC_Koeln	34	10	6	18	51:69	-18	36
15	FC_Augsburg	34	9	9	16	45:63	-18	36
16	Werder_Bremen	34	8	7	19	42:69	-27	31
17	Fortuna_Duesseldorf	34	6	12	16	36:67	-31	30
18	SC_Paderborn_07	34	4	8	22	37:74	-37	20

#### 1.4.7 Importieren von Datenbanken sowie Datenbanktabellen

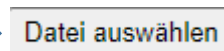
1. Klick auf die Datenbank *fussball\_bundesliga\_2019\_20*



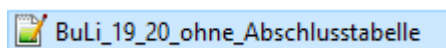
2. Klick auf die Registerkarte *Importieren*



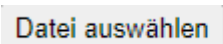
3. Schaltfläche „Datei auswählen“ anklicken



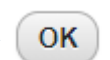
4. SQL-Datei „BuLi\_19\_20\_ohne\_Abschlusstabelle.sql“ auswählen



(Abschlusstabelle soll selbstständig mit phpMyAdmin erstellt werden, während die restlichen Tabellen als SQL-Datenbank importiert werden.)

5. Damit ist die SQL-Datei ausgewählt.  BuLi\_19\_...tabelle.sql

6. Zum Schluss erfolgt ein Klick auf den OK-Button rechts unten.

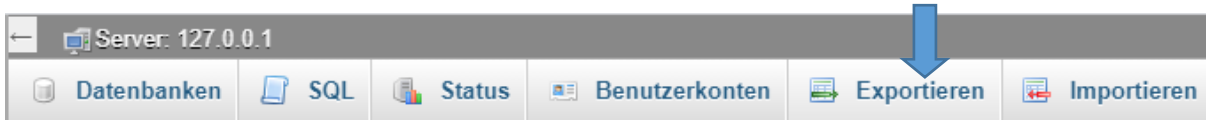


## 1.4.8 Exportieren einer Datenbank

1. Klick auf die Startseite des Servers  
(Haus-Symbol links oben)

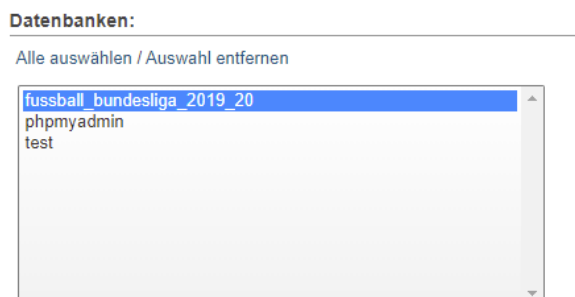


2. Klick auf die Registerkarte *Exportieren*



3. angepasste Exportmethode auswählen (statt der schnellen Methode)

4. Datenbank in der Liste auswählen  
(*fussball\_bundesliga\_2019\_20*)



5. Dateinamen von „@SERVER@“ (= 127.0.0.1) in = „BuLi\_19\_20“ umbenennen  
und auf den OK-Button rechts unten zum Bestätigen klicken.

### Ausgabe:

- ☐ Exportierte Datenbanken/Tabellen/Spalten umbenennen
- ☒ Speichere Ausgabe in Datei

Vorlage für den Dateinamen: ?

BuLi\_19\_20



OK

## 2 Einbinden einer Datenbank in das System

### 2.1 Installation eines MySQL-ODBC-Treibers

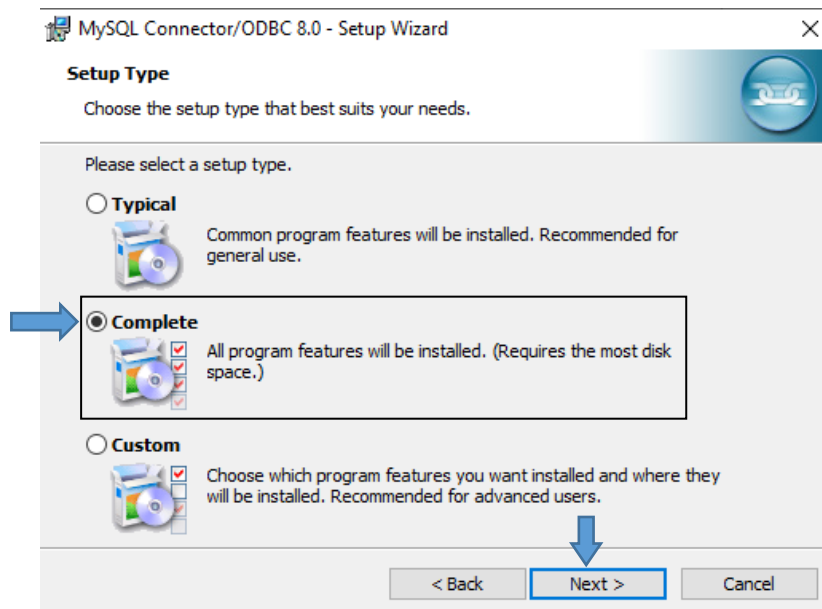
(Quelle für Windows und macOS: <https://dev.mysql.com/downloads/connector/odbc/>)

Bei zwei Arbeitsschritten soll folgendes beachtet werden:

- Lizenzvereinbarung akzeptieren, durch Klick auf *Next* bestätigen



- **Complete** auswählen, anschließend erneuter Klick auf *Next*-Button



- weitere Arbeitsschritte einfach nur mit *Next* bzw. *OK* bestätigen

ODBC-Connector-Download für macOS (dieselbe Quelle wie für Windows):

#### Connector/ODBC 5.3.14

Select Operating System:

macOS

Looking for the latest GA version?

- macOS statt Microsoft Windows als Betriebssystem auswählen

! Packages for Mojave (10.14) are compatible with High Sierra (10.13)

macOS 10.14 (x86, 64-bit), DMG Archive	5.3.14	7.2M	<a href="#">Download</a>
(mysql-connector-odbc-5.3.14-macos10.14-x86-64bit.dmg)			
		MD5: ee4c5711b6bfbac7612a35beb42bab0   Signature	
macOS 10.14 (x86, 64-bit), Compressed TAR Archive	5.3.14	6.9M	<a href="#">Download</a>
(mysql-connector-odbc-5.3.14-macos10.14-x86-64bit.tar.gz)			
		MD5: c25d52a2e544d06bf4b8c32792226c69   Signature	

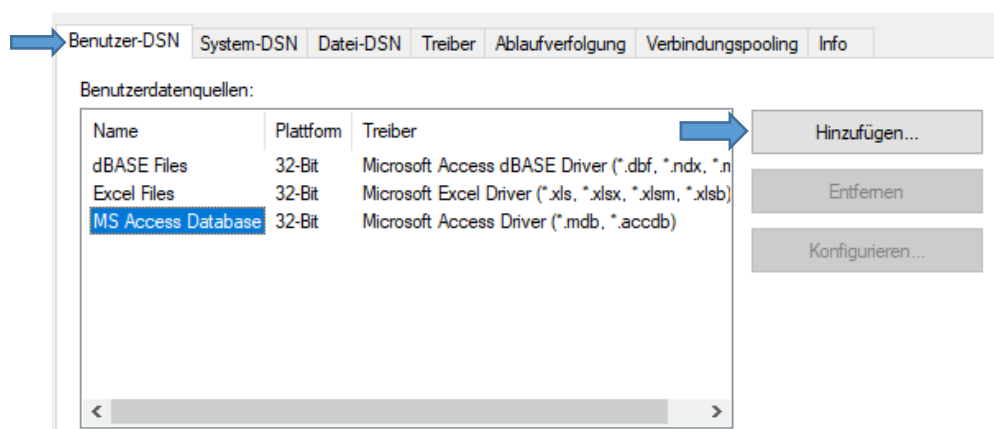
## 2.2 Erstellen einer ODBC-Datenquelle

- Menü *Systemsteuerung* im Startmenü auswählen
- dort Menüpunkt *Verwaltung* und anschließend *ODBC-Datenquellen (64-Bit)*

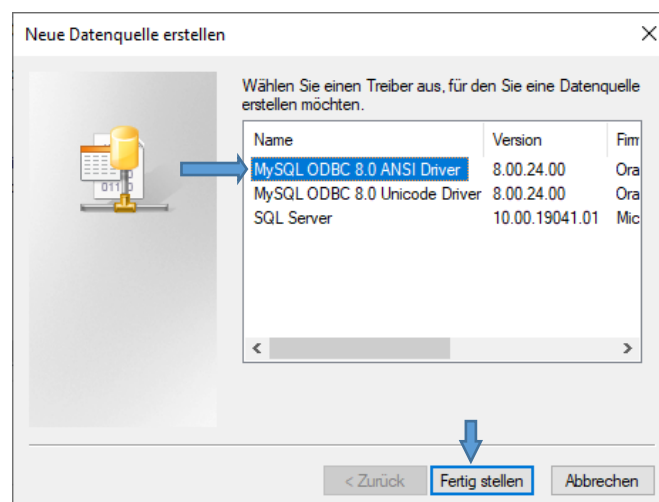


### 2.2.1 Einfügen des MySQL-ODBC-ANSI-Treibers in die Datenquelle

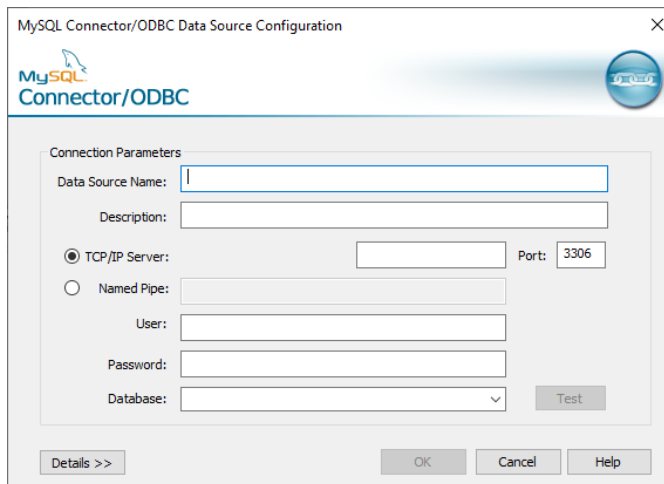
- Klick auf die Schaltfläche *Hinzufügen* (Registerkarte *Benutzer-DSN* bereits ausgewählt)



- anschließend den **MYSQL ODBC 8.0 ANSI Driver** auswählen und fertig stellen



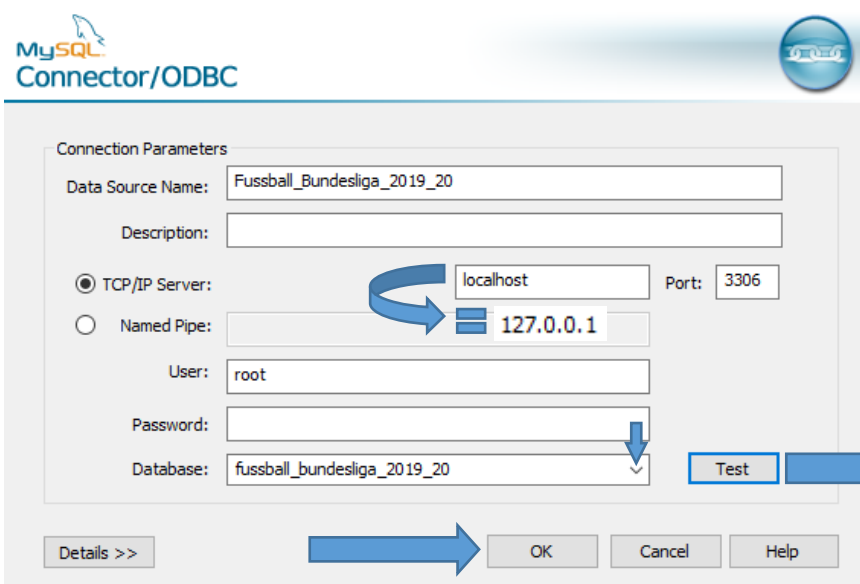
## 2.2.2 Konfiguration einer Datenquelle



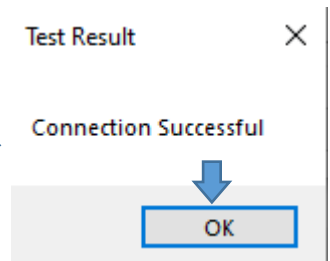
- Data Source Name (DSN, 1. Zeile):  
*Fussball\_Bundesliga\_2019\_20*
- TCP/IP-Server: *localhost* (= 127.0.0.1)  
(3. Zeile)
- User (Benutzer): *root* (kein Passwort!)  
(5. Zeile)      (6. Zeile bleibt leer)
- Database (ausgewählte Datenbank):  
*fussball\_bundesliga\_2019\_20*  
(7. Zeile)

## 2.2.3 Verbindungstest mit dem System

MySQL Connector/ODBC Data Source Configuration



- Klick auf den *Test*-Button nach Dateneingabe
- alle Daten richtig → Verbindung erfolgreich,
- beide Fenster mit *OK* bestätigen

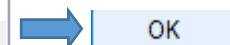


## 2.2.4 Benutzerdatenquelle mitsamt integrierter Datenbank

Benutzerdatenquellen:

Name	Plattform	Treiber
dBASE Files	32-Bit	Microsoft Access dBASE Driver (*.dbf, ...)
Excel Files	32-Bit	Microsoft Excel Driver (*.xls, *.xlsx, *.xls)
<b>Fussball_Bundesliga_2019_20</b>	64-Bit	MySQL ODBC 8.0 ANSI Driver
MS Access Database	32-Bit	Microsoft Access Driver (*.mdb, *.accdb)

Bestätigung mit *OK*





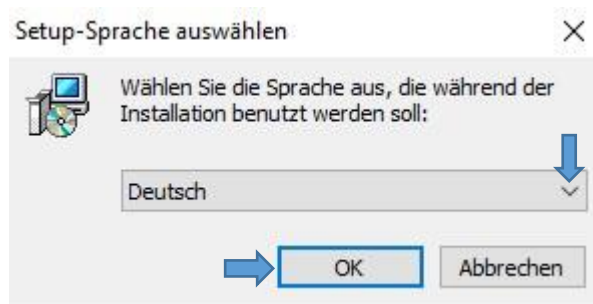
### 3 Erstellen einer Lazarus-GUI mit eingebundener Datenbank

#### 3.1 Installation von Lazarus

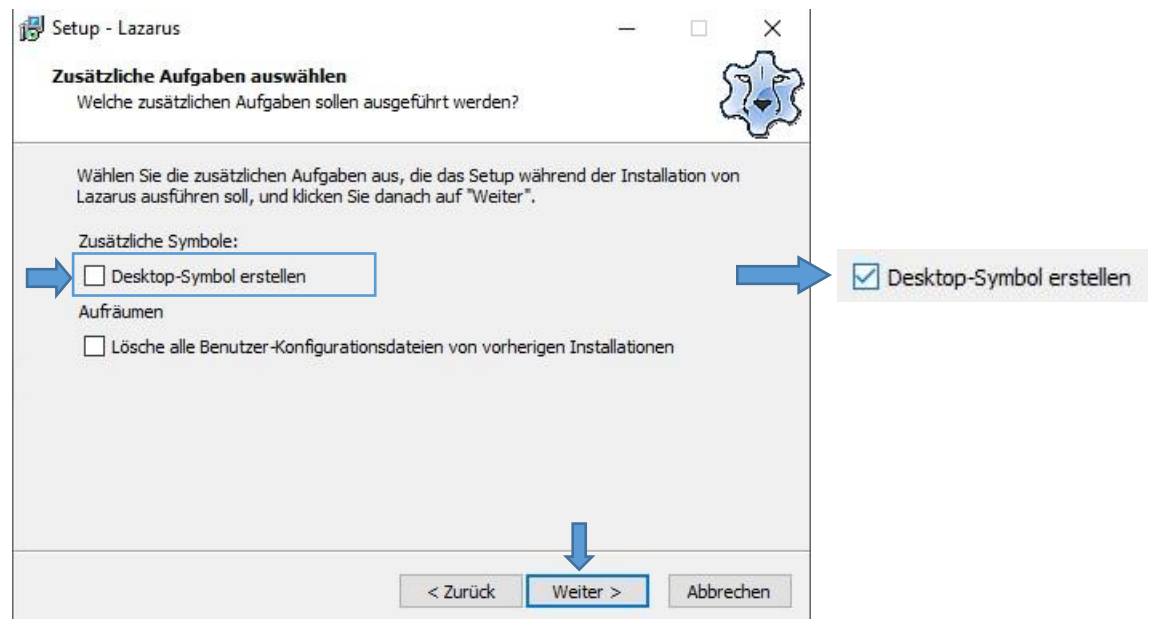
Quelle für Windows: <https://www.heise.de/download/product/lazarus-59651/download>  
(64-Bit)

Bei einigen Installationsschritten von Lazarus muss folgendes beachtet werden:

1. deutsche Sprache auswählen (bereits eingestellt) und mit **OK** bestätigen



2. Häkchen bei „Desktop-Symbol erstellen“ setzen und auf **Weiter** klicken



Bei den anderen Installationsschritten soll einfach mit „OK“ bzw. „Weiter“ bestätigt werden.

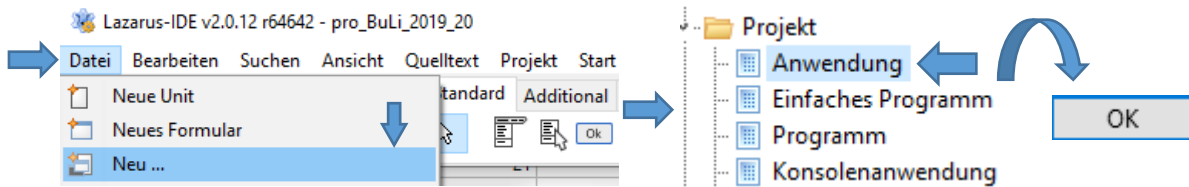
Quelle für macOS:

<https://sourceforge.net/projects/lazarus/files/Lazarus%20macOS%20x8664/Lazarus%202.0.12/> (64-Bit)

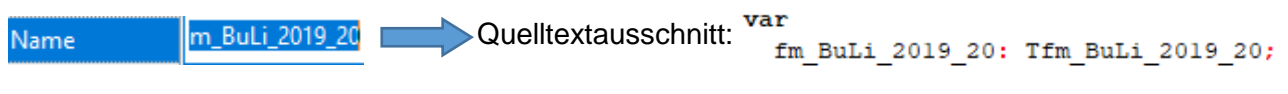
### 3.2 Anfertigen des Formulars und Anbinden der Datenbank

Sollte beim Starten von Lazarus eine fertige Anwendung starten, sollte eine neue Anwendung mit einem leeren Formular erstellt werden.

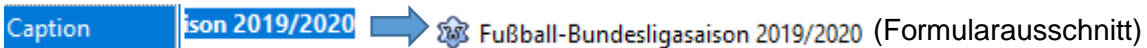
(Datei → Neu ... → Anwendung → Bestätigung mit Klick auf OK-Button)



Ändern des Formularnamens im Objektinspektor (linkes Fenster, sichtbarmachen mit F11):

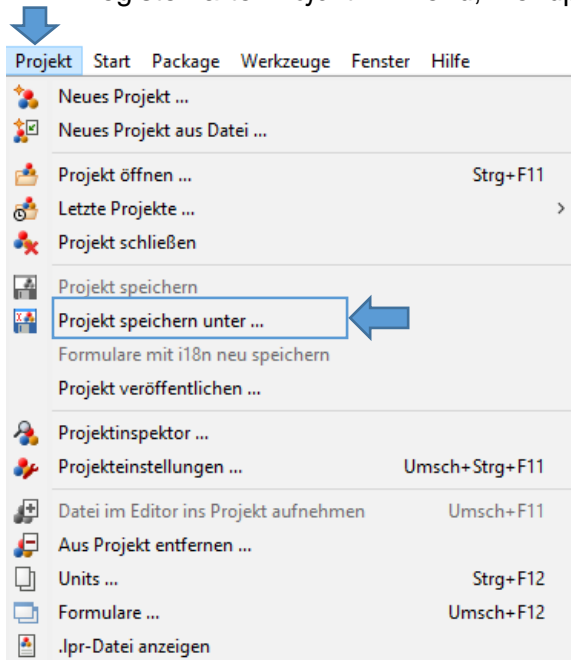


Ändern der Überschrift des Formulars (Bei *Caption* sind alle möglichen Zeichen erlaubt.):

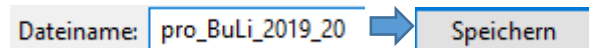


Speichern des Projekts:

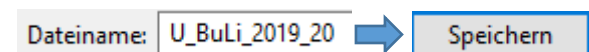
- Registerkarte *Projekt* im Menü, Menüpunkt *Projekt speichern unter ...*



Speichern der Projekt-Datei (öffnet sich nur dann, wenn ein Fehler beim Übersetzen des Programms auftritt.)

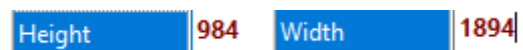


Speichern der Unit-Datei (Hauptdatei):



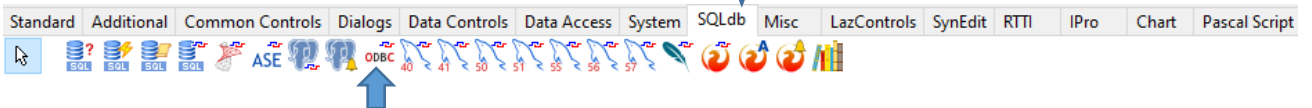
Anpassen der Höhe und Breite der GUI

Im Objektinspektor:



## Einfügen unsichtbarer Komponenten und Änderung deren Eigenschaften im Objektspektor

### a) ODBC-Datenbankverbindung (Registerkarte SQLdb)



a.1. Bezeichnung ändern: ODBC

Name Bundesliga\_2019\_20

a.2. Datenbankbezeichnung:

DatabaseName Fussball\_Bundesliga\_2019\_20

a.3. Verbindung zur DB aktivieren (Connected auf True setzen):



Formularausschnitt:



### b) SQLTransaction (Registerkarte SQLdb)



b.1. Bezeichnung ändern:

Name Trans\_BuLi\_19\_20

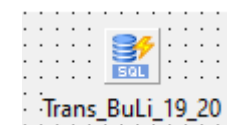
b.2. ODBC als Datenbank:

Database Bundesliga\_2019\_20

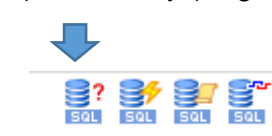
b.3. Aktivierung auf True setzen:



Formularausschnitt:



### c) SQLQuery (Registerkarte SQLdb)



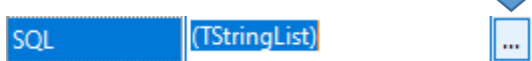
c.1. Query umbenennen:

Name qu\_Abschlusstabelle

c.2. ODBC als Datenbank:

Database Bundesliga\_2019\_20

c.3. SQL-Abfrage bearbeiten:

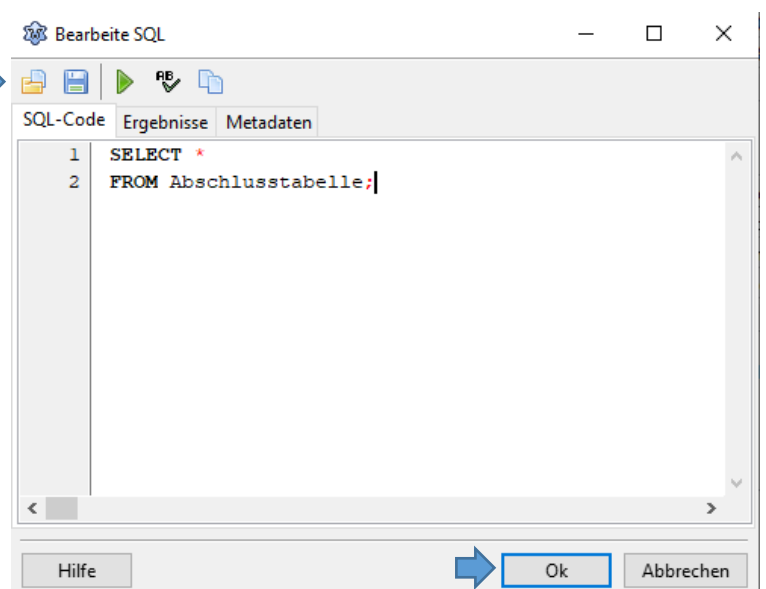
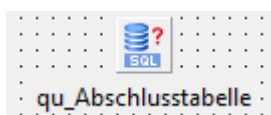


Auswählen aller Daten aus der Abschlusstabelle:

c.4. Query aktivieren (auf True setzen):



Formularausschnitt:



d) DataSource (Registerkarte *Data Access*)

Standard Additional Common Controls Dialogs Data Controls Data Access System SQLdb Misc LazControls SynEdit RTTI IPro Chart Pascal Script

d.1. umbenennen nach der Tabelle:  
Name DS\_Abschlusstabelle

d.2. Query als Datensatz:  
DataSet qu\_Abschlusstabelle

Formularausschnitt:  
DS\_Abschlusstabelle

Sichtbare Komponenten in der Anwendungs-GUI (GUI=Graphical User Interface)

e) DBGrid (Registerkarte *Data Controls*)

Standard Additional Common Controls Dialogs Data Controls Data Access System

leeres DBGrid:

Änderung des Namens und Einbauen der Datenbank als Datenquelle:

Name DBG\_Abschlusstabelle DataSource Abschlusstabelle

Formularausschnitt (DBG\_Abschlusstabelle mit gefüllten Werten):

	Platz	Verein	Spiele	Siege	entschieden	Niederlagen	Tore_und_C	Tordifferenz	Punkte
▶	1	FC_Bayern_	34	26	4	4	100:32	68	82
	2	Borussia_D	34	21	6	7	84:41	43	69
	3	RB_Leipzig	34	18	12	4	81:37	44	66

Genauso sollen jeweils eine SQLQuery, eine DataSource sowie ein DBGrid für die anderen fünf Tabellen (siehe Datenbank) mit entsprechenden Bezeichnungen und SQL-Abfragen (im Falle von SQLQuerys) eingerichtet werden.

f) Memofeld für Abfragen (Registerkarte *Standard*):

Umbenennen in Memo\_SQL:

Standard Additional Common Controls

Inhalt des Memofelds leeren:  
Lines (TStrings)

Formularausschnitt:

Zeichenketteneditor

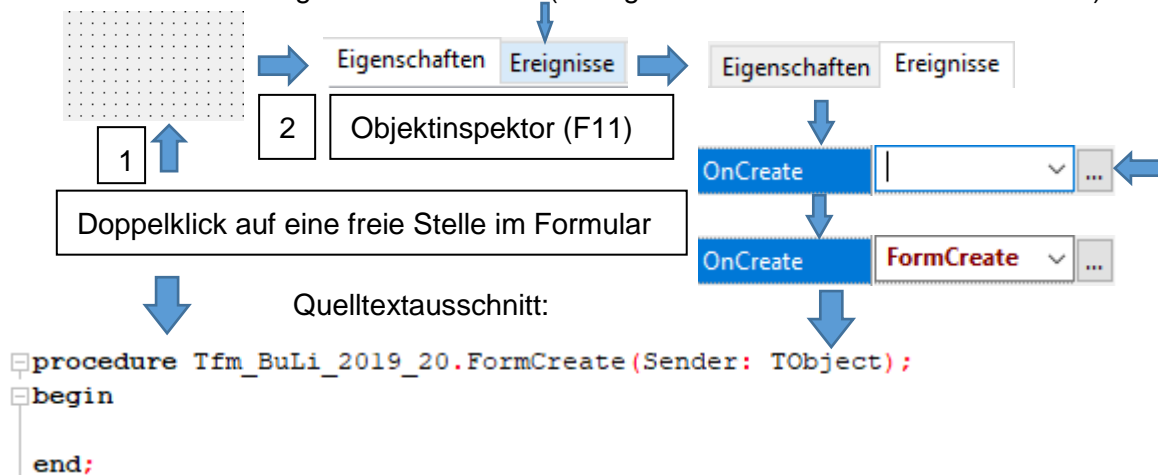
Sortieren  
Löschen  
1

0 Zeilen, 0 Zeichen

2 Ok Abbrechen

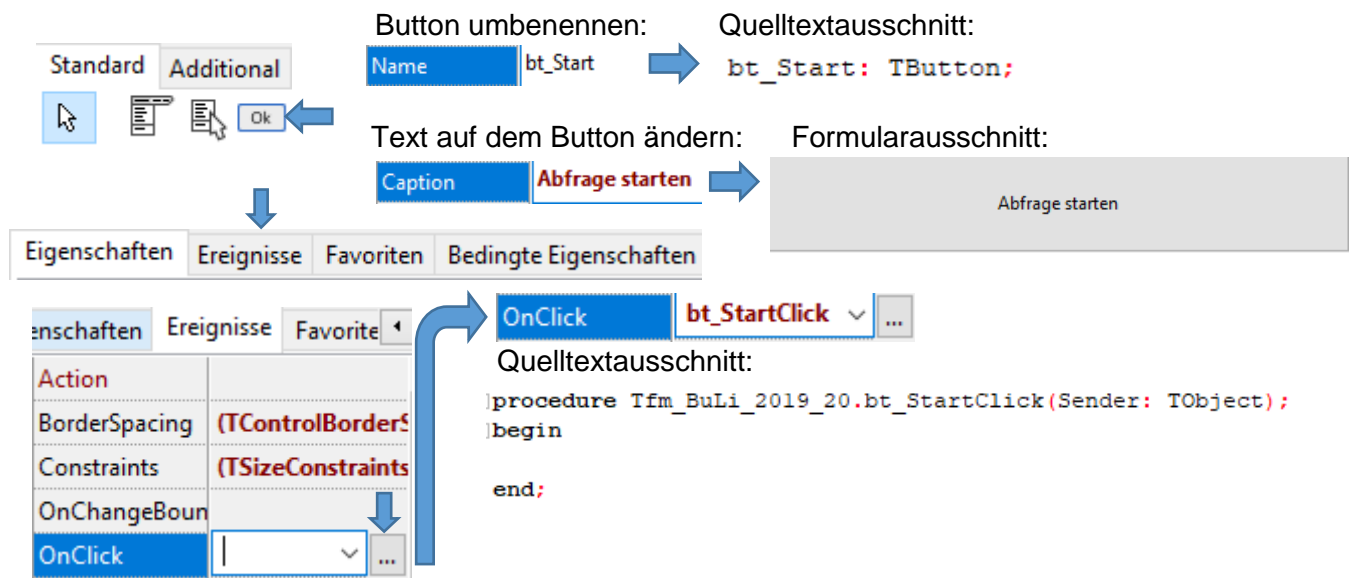
g) das Formular selbst

- Prozedur zum Erzeugen des Formulars (2 Möglichkeiten nach Klick aufs Formular):

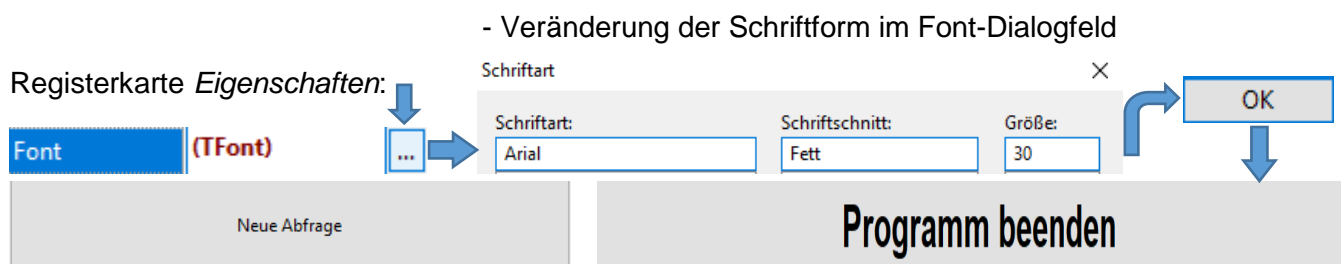


h) Buttons (Registerkarte *Standard*)

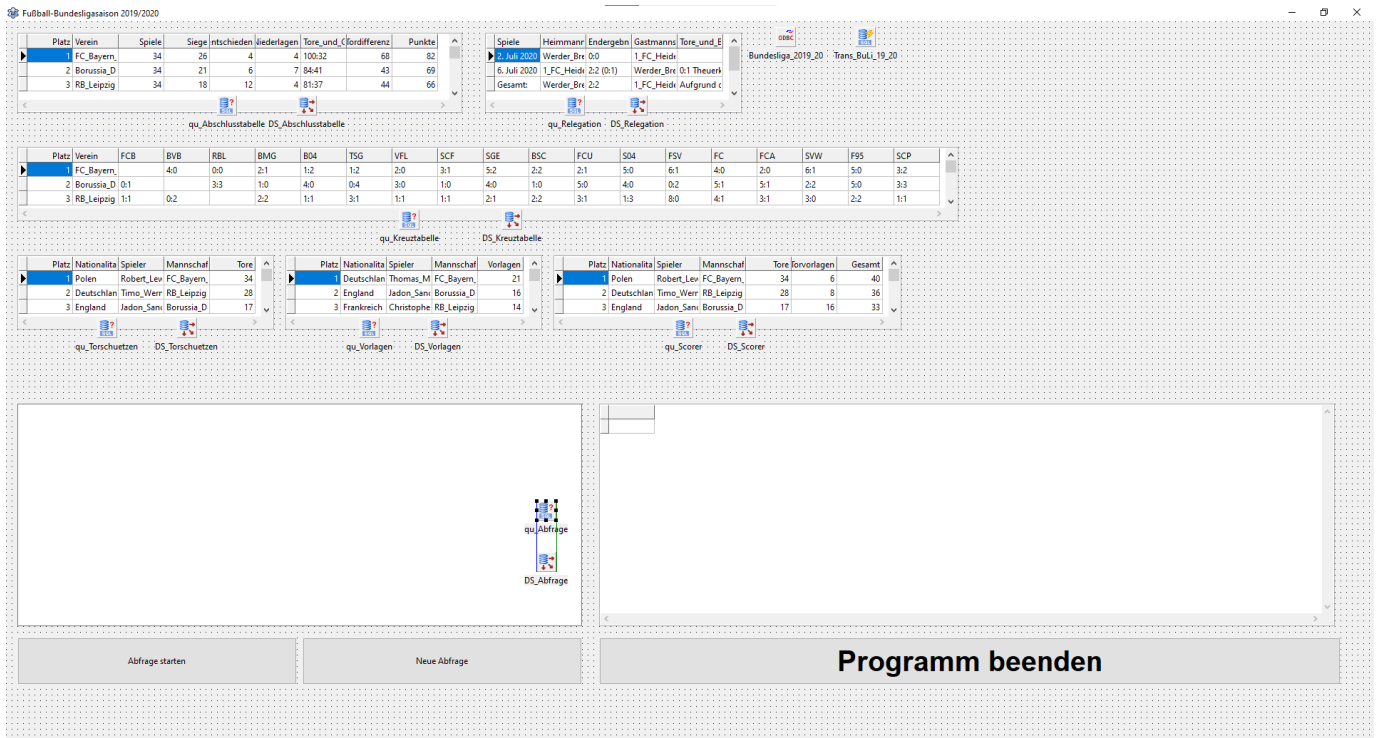
- Prozedur zum Klick auf den Button *bt\_Start* (Registerkarte *Ereignisse* im Objektinspektor):



Neben *bt\_Start* (rechts) soll außerdem ein Button *bt\_neu* mit der Aufschrift *Neue Abfrage* erzeugt werden, für den ebenfalls eine Prozedur zum Klicken (*bt\_neuClick*) erstellt werden. Rechts davon, unter der Abfragetabelle *DBG\_Loesung* (welche dann noch miteingebaut wird) soll außerdem der Button *bt\_beenden* (Aufschrift: *Neue Abfrage*) zum Beenden des Programms erstellt werden, bei dem zusätzlich noch die Schrift auf 30 Punkt (pt) vergrößert und fett gestellt werden soll.



Am Ende soll die zusammengebaute grafische Benutzeroberfläche (Schnittstelle) des Formulars *fm\_BuLi\_2019\_20* folgendermaßen aussehen (Darstellung deutlich verkleinert):



Die ODBC-Datenbankverbindung *Bundesliga\_2019\_20* und die SQLTransaction *Trans\_BuLi\_2019\_20* (siehe rechts oben) sind die einzigen beiden unsichtbaren Komponenten (unsichtbar, da sie bei der Ausführung des Programms im Formular nicht zu sehen sind und somit auch nicht verwendet bzw. bearbeitet werden können), die auf die gesamte Datenbank *Fussball\_Bundesliga\_2019\_20* bezogen sind.

Für jede der sechs Datenbanktabellen (*Abschlusstabelle*, *Relegation*, *Kreuztabelle*, *Torschuetzenliste*, *meiste\_Torvorlagen* sowie *Scorerliste*) wurden jeweils eine SQLQuery, eine DataSource (unsichtbar) sowie ein DBGrid (sichtbar) als zuständige Komponenten mit entsprechenden Bezeichnungen (*qu\_Tabellenname*, *DS\_Tabellenname* bzw. *DBG\_Tabellenname*) ins Formular eingebaut.

Innerhalb des Memofeldes *Memo\_SQL* befinden sich außerdem eine SQLQuery (*qu\_Abfrage*) sowie eine DataSource (*DS\_Abfrage*). An *qu\_Abfrage* soll erstmal bis auf den Namen nichts verändert werden. Bei *DS\_Abfrage* soll *qu\_Abfrage* als Datensatz (DataSet) eingestellt werden. Rechts daneben über dem Button *bt\_beenden* befindet sich ein DBGrid (*DBG\_Loesung*) mit *DS\_Abfrage* als Datenquelle (DataSource). Alle drei Komponenten erhalten noch keine Daten, da sie erst bei der Ausführung des Programms zum Testen von SQL-Abfragen, die dann vom Nutzer ins Memofeld *Memo\_SQL* eingegeben werden, gebraucht werden. *DBG\_Loesung* dient dabei zur Anzeige für die Lösung der SQL-Abfrage.

Der Code des Programms wurde in der Programmiersprache Pascal geschrieben

(// bzw. {...} in blauer Fettschrift = Kommentare bzw. Erklärungen, werden nicht mitgelesen)

Quelltexteditor

```

1  unit U_BuLi_2019_20;
.  //Programmname
.  {$mode objfpc}{$H+}
.  //grafische Oberfläche (Schnittstelle)
5  interface
.
.  //eingebundene Programmbibliothek, Arten von Komponenten, Objekten und
.  //Klassen, die in diesem Programm verwendet werden
.  uses
10   Classes, SysUtils, odbcconn, sqldb, db, FileUtil, Forms, Controls, Graphics,
.   Dialogs, DBGrids, StdCtrls;
12
.  type
.
15  { Tfm_BuLi_2019_20 }
.  //Formular mit grafischer Oberfläche und deren Komponenten
.  Tfm_BuLi_2019_20 = class(TForm)
.   bt_neu: TButton; //Schaltfläche für neue SQL-Abfragen
.   Bundesliga_2019_20: TODBConnection; //ODBC-Datenbankverbindung mit Datenbank
20   bt_Start: TButton; //Schaltfläche zum Start der SQL-Abfrage
.   bt_beenden: TButton; //Schaltfläche zum Schließen des Formulars
.   DBG_Loesung: TDBGrid; //Datenbanktabelle für Lösungen zu den SQL-Abfragen
.   //Relationen (Tabellen) aus der eingebundenen Datenbank
.   DBG_Scorer: TDBGrid;
25   DBG_Vorlagen: TDBGrid;
.   DBG_Torschuetzen: TDBGrid;
.   DBG_Relegation: TDBGrid;
.   DBG_Kreuztabelle: TDBGrid;
.   DBG_Abschlusstabelle: TDBGrid;
30   //Datenquellen
.   DS_Abfrage: TDataSource;
.   DS_Scorer: TDataSource;
.   DS_Vorlagen: TDataSource;
.   DS_Torschuetzen: TDataSource;
35   DS_Relegation: TDataSource;
.   DS_Kreuztabelle: TDataSource;
.   DS_Abschlusstabelle: TDataSource;
.   //Eingabefeld für SQL-Abfragen
.   Memo_SQL: TMemo;
40   //Querys mit den Datenbankabfragen für jede Relation sowie Lösungstabelle
.   qu_Abschlusstabelle: TSQLQuery;
.   qu_Kreuztabelle: TSQLQuery;
.   qu_Relegation: TSQLQuery;
.   qu_Torschuetzen: TSQLQuery;
45   qu_Vorlagen: TSQLQuery;
.   qu_Scorer: TSQLQuery;
.   qu_Abfrage: TSQLQuery;
.   //Transaktion der in die ODBC_Datenquelle eingebundenen Datenbank
.   Trans_BuLi_19_20: TSQLTransaction;
50   //Prozeduren für das Klicken auf die jeweilige Schaltfläche (Button)
.   //Registerkarte "Ereignisse" im Objektinspektor => OnClick
.   procedure bt_beendenClick(Sender: TObject);
.   procedure bt_neuClick(Sender: TObject);
.   procedure bt_StartClick(Sender: TObject);

```



## 3.3 Implementation des Programmcodes in Pascal

```

55  //Prozedur zum Erzeugen des Formulars
.  procedure FormCreate(Sender: TObject);
.  private
.  { private declarations }
.  public
60  { public declarations }
.  end;

.
.  var //Hier wird das Formular immer als globale Variable deklariert.
.      fm_BuLi_2019_20: Tfm_BuLi_2019_20;
65
.  implementation
.  //Bereich, wo der Anwender die Befehle selbst eingibt
.  {$R *.lfm}
.
.
70 { Tfm_BuLi_2019_20 }
.
.  procedure Tfm_BuLi_2019_20.FormCreate(Sender: TObject);
.  begin
.  //Formular erzeugen, durch Doppelklick aufs Formular aufgerufen
75  //weitere Option: Registerkarte "Ereignisse" im Objektinspektor => OnCreate
.  //Ausrichtung: Abstände von oben sowie links
.  top:=10;
.  left:=10;
.  Memo_SQL.Lines.Add('SELECT '); //Erste Zeile mit "SELECT " wird eingefügt.
80  Memo_SQL.Lines.Add('FROM '); //Zweite Zeile mit "FROM " wird eingefügt.
.  Memo_SQL.SelStart:=7; {Cursor steht auf der 8. Stelle in der ersten Zeile
.  nach dem Wort "SELECT ", welches mitsamt dem darauffolgenden Leerzeichen
.  7 Stellen einnimmt}
.  //Datenbanktabelle (DataBaseGrid) für Abfragen unsichtbar machen (ausblenden)
85  DBG_Loesung.Hide; //DBG_Loesung.Visible:=false;
.  end;

.
.  procedure Tfm_BuLi_2019_20.bt_StartClick(Sender: TObject);
.  begin
90  //Klick auf die Schaltfläche (Button) "Abfrage Starten"
.
.  qu_Abfrage.Active:=false; //Query für Datenbankabfrage deaktivieren
.  //Text in qu_Abfrage wird ins Eingabefeld Memo_SQL übertragen
.  qu_Abfrage.SQL.Text:=Memo_SQL.Text;
95  qu_Abfrage.Active:=true; //Query für Datenbankabfragen wieder aktivieren
.
.  //Alle 6 Datenbanktabellen werden ausgeblendet (unsichtbar gemacht).
.  DBG_Abschlusstabelle.Hide; //Visible:=false;
.  DBG_Kreuztabelle.Hide; //Visible:=false;
100  DBG_Relegation.Hide; //Visible:=false;
.  DBG_Torschuetzen.Hide; //Visible:=false;
.  DBG_Vorlagen.Hide; //Visible:=false;
.  DBG_Scorer.Hide; //Visible:=false;
.  //Ausblenden des Eingabefeldes Memo_SQL und der Schaltfläche "Abfrage starten"
105  Memo_SQL.Hide; //Visible:=false;
.  bt_start.Hide; //Visible:=false;
.  //Datenbanktabelle für SQL-Lösungen wird eingeblendet (sichtbar gemacht)
.  DBG_Loesung.Show; //DBG_Loesung.Visible:=true;

```



```

.   end;
110
.   procedure Tfm_BuLi_2019_20.bt_neuClick(Sender: TObject);
.   begin
.       //Klick auf die Schaltfläche (Button) "Neue Abfrage"
.
.       //Datenbanktabellen, Eingabefeld und bt_start wieder einblenden
115   DBG_Abschlusstabelle.Show;//Visible:=true;
.       DBG_Kreuztabelle.Show;//Visible:=true;
.       DBG_Relegation.Show;//Visible:=true;
.       DBG_Torschuetzen.Show;//Visible:=true;
120   DBG_Vorlagen.Show;//Visible:=true;
.       DBG_Scorer.Show;//Visible:=true;
.       Memo_SQL.Show;//Visible:=true;
.       bt_start.Show;//Visible:=true;
.       //Datenbanktabelle für Lösungen wird unsichtbar gemacht (ausgeblendet)
125   DBG_Loesung.Hide;//Visible:=false;
.       Memo_SQL.Clear;// Inhalt des Eingabefeldes für SQL-Abfragen wird gelöscht
.       qu_Abfrage.Active:=false;//Query für die SQL-Abfragen wird deaktiviert
.       //SELECT = Auswahl des Datenbank-/Tabellenabschnitts mit dessen Inhalt
.       //FROM = betroffene Tabelle/-n, wo der gesuchte Inhalt sich befindet
130   Memo_SQL.Lines.Add('SELECT ');//Erste Zeile mit "SELECT " wird eingefügt.
.       Memo_SQL.Lines.Add('FROM ');//Zweite Zeile mit "FROM " wird eingefügt.
.       Memo_SQL.SetFocus;//Cursor wird ins Eingabefeld Memo_SQL gesetzt
.       Memo_SQL.SelStart:=7;
.       //Cursor steht auf der 8. Stelle in der ersten Zeile nach dem Wort "SELECT ",
135   //welches mitsamt dem darauffolgenden Leerzeichen 7 Stellen einnimmt
.   end;
.
.   procedure Tfm_BuLi_2019_20.bt_beendenClick(Sender: TObject);
.   begin
140   fm_BuLi_2019_20.Close;//Schließen des Formulars -> Programm wird beendet
.   end;
.
.   end.
144

```

Hiermit ist der Programmcode erstellt und nun kann das Programm mit einem Klick auf das grüne dreieckige Wiedergabesymbol im Menü (siehe nächstes Bild) oder durchs Drücken der Funktionstaste F9 (auf der Tastatur) gestartet werden.



Der Eingabebereich des Nutzers befindet sich zwischen den orangenen Streifen.

Mit zwei Schrägstrichen (Slashes: „//“) sind einzeilige Kommentare gekennzeichnet, während mehrzeilige Kommentare von geschweiften Klammern „{...}“ umschlossen werden.

Kommentare sind bei der Kompilierung (Übersetzung) bzw. Ausführung des Programms kein Bestandteil vom Pascal-Quellcode des Programms.

### 3.4 Ausführen des Programms und Testen von SQL-Abfragen

- verkleinertes Abbild der Benutzeroberfläche nach dem Programmstart

Abschlusstabelle (links oben):

Platz	Verein	Spiele	Siege	Unentsch
1	FC_Bayern_Muenchen	34	26	
2	Borussia_Dortmund	34	21	
3	RB_Leipzig	34	18	

Relegation (rechts oben):

Spiele	Heimmannschaft	Enderg
2. Juli 2020	Werder_Bremen	0:0
6. Juli 2020	1.FC_Heidenheim	2:2 (0:1)
Gesamt:	Werder_Bremen	2:2

Kreuztabelle (Mitte):

Platz	Verein	FCB	BVB	RBL	BMG	B04	TSG	VFL	SCF	SGE	BSC	FCU	S04	FSV	FC	FCA	SVW	F95	SCP
1	FC_Bayern_Muenchen		4:0	0:0	2:1	1:2	1:2	2:0	3:1	5:2	2:2	2:1	5:0	6:1	4:0	2:0	6:1	5:0	3:2
2	Borussia_Dortmund	0:1		3:3	1:0	4:0	0:4	3:0	1:0	4:0	1:0	5:0	4:0	0:2	5:1	5:1	2:2	5:0	3:3
3	RB_Leipzig	1:1	0:2		2:2	1:1	3:1	1:1	1:1	2:1	2:2	3:1	1:3	8:0	4:1	3:1	3:0	2:2	1:1

v. l. n. r.: Torschuetzenliste, meiste\_Torvorlagen, Scorerliste (unten)

Platz	Nationalitaet	Spieler	Platz	Nationalitaet	Spieler	Platz	Nationalitaet	Spieler	Mannscha
1	Polen	Robert_Lewandowski	1	Deutschland	Thomas_Mueller	1	Polen	Robert_Lewandowski	FC_Bayern
2	Deutschland	Timo_Werner	2	England	Jadon_Sancho	2	Deutschland	Timo_Werner	RB_Leipzig
3	England	Jadon_Sancho	3	Frankreich	Christopher_Nkunku	3	England	Jadon_Sancho	Borussia_D

SQL-Abfragefeld (Memo\_SQL):

Button zum Ausführen der Abfrage

```
SELECT
FROM
```

(bt\_start):

Abfrage starten

Neue Abfrage

Neue-Abfrage-Button (bt\_neu)

Button zum Schließen des Programms (bt\_beenden)

**Programm beenden**

SQL-Abfragen - Bsp. 1: (\*- Auswahl aller Daten aus der Tabelle „Torschuetzenliste“)

```
SELECT *
FROM Torschuetzenliste;
```

Abfrage starten

Lösungstabelle *DBG\_Loesung* wird sichtbar. Dort wird die *Abschlusstabelle* eingeblendet.

Platz	Nationalitaet	Spieler	Mannschaft	Tore
1	Polen	Robert_Lewandowski	FC_Bayern_Muenchen	34
2	Deutschland	Timo_Werner	RB_Leipzig	28
3	England	Jadon_Sancho	Borussia_Dortmund	17
4	Niederlande	Wout_Weghorst	VfL_Wolfsburg	16
5	Deutschland	Rouwen_Hennings	Fortuna_Duesseldorf	15
6	Kolumbien	Jhon_Cordoba	1_FC_Koeln	13
7	Norwegen	Erling_Haaland	Borussia_Dortmund	13
8	Deutschland	Florian_Niederlechner	FC_Augsburg	13
9	Schweden	Robin_Quaison	1_FSV_Mainz_05	13
10	Schweden	Sebastian_Andersson	1_FC_Union_Berlin	12
11	Deutschland	Serge_Gnabry	FC_Bayern_Muenchen	12
12	Deutschland	Kai_Havertz	Bayer_04_Leverkusen	12
13	Kroatien	Andrej_Kramaric	TSG_1899_Hoffenheim	12

**Programm beenden**Klick auf *bt\_neu*, um eine neue SQL-Abfrage einzugeben (*DBG\_Loesung* wieder unsichtbar):

Neue Abfrage

```
SELECT
FROM
```

SQL-Abfragen Bsp. 2:

```
SELECT S.Nationalitaet AS Land, S.Spieler, S.Mannschaft
```

```
FROM Scorerliste S INNER JOIN Torschuetzenliste T ON
S.Spieler=T.Spieler INNER JOIN meiste_Torvorlagen V ON
T.Spieler=V.Spieler
```

```
WHERE S.Nationalitaet='Deutschland';
```

Auflisten der Nationalität (als Land ausgegeben), des Namens und der Mannschaft der Spieler, die in allen 3 Listen (Torschützen, Vorlagengeber und Scorer) vorhanden sind und aus Deutschland kommen.

Lösungsausschnitt aus *DBG\_Loesung* (nur ein Datensatz trifft in diesem Fall zu):

Land	Spieler	Mannschaft
Deutschland	Serge_Gnabry	FC_Bayern_Muenchen

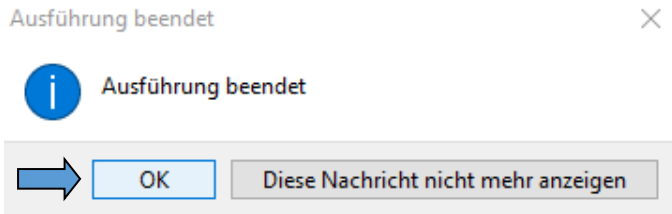
### 3.5 Beenden des Programms und mögliche Fehlermeldungen



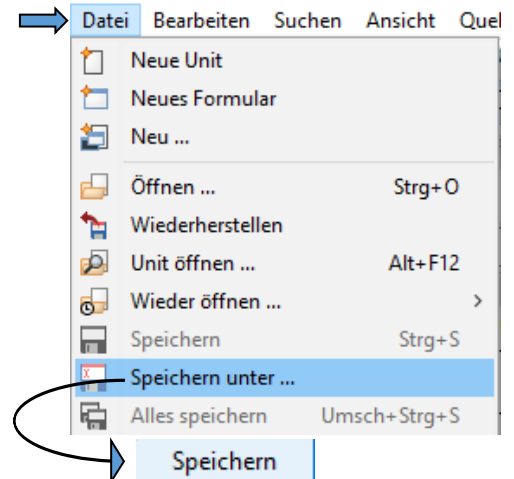
- Klick auf bt\_beenden oder aufs Kreuz in der rechten, oberen Ecke



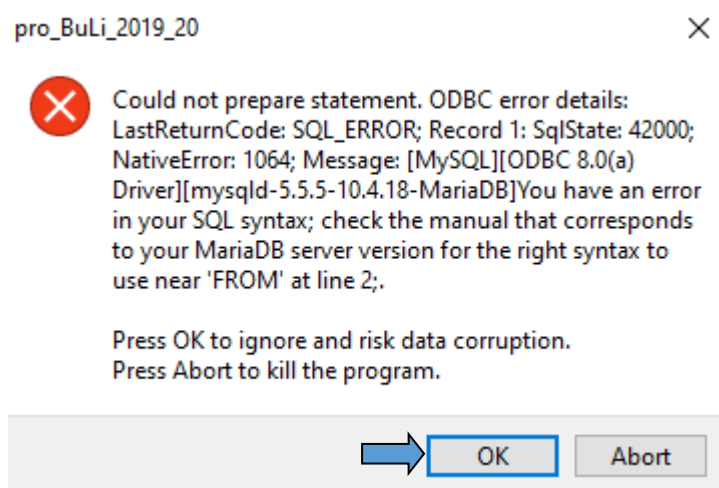
Hinweisfenster (nach Schließung des Programms):



Quelltext-Datei speichern:



Fehler in der SQL-Abfrage oder fehlende Abfrage:



Starten des Programms ohne die Server Apache und MySQL gestartet zu haben:

