



合肥大學  
HEFEI UNIVERSITY



## Datenbanken

# 18. Fabrik-Datenbank: LibreOffice Base mit PostgreSQL verbinden

Thomas Weise (汤卫思)  
[tweise@hfuu.edu.cn](mailto:tweise@hfuu.edu.cn)

Institute of Applied Optimization (IAO)  
School of Artificial Intelligence and Big Data  
Hefei University  
Hefei, Anhui, China

应用优化研究所  
人工智能与大数据学院  
合肥大学  
中国安徽省合肥市

# Databases



Dies ist ein Kurs über Datenbanken an der Universität Hefei (合肥大学).

Die Webseite mit dem Lehrmaterial dieses Kurses ist <https://thomasweise.github.io/databases> (siehe auch den QR-Kode unten rechts). Dort können Sie das Kursbuch (in Englisch) und diese Slides finden. Das Repository mit den Beispielen finden Sie unter <https://github.com/thomasWeise/databasesCode>.



# Outline



1. Einleitung
2. Mit der Datenbank verbinden
3. Zusammenfassung





# Einleitung



# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.



# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.
- Oder vielleicht schreiben wir auch ein Python-Programm, das die Daten unsere Datenbank mit Methoden der Artificial Intelligence (AI) und Machine Learning (ML) analysiert, um zukünftige Verkäufe vorherzusehen.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.
- Oder vielleicht schreiben wir auch ein Python-Programm, das die Daten unsere Datenbank mit Methoden der Artificial Intelligence (AI) und Machine Learning (ML) analysiert, um zukünftige Verkäufe vorherzusehen.
- Vielleicht läuft unser Python-Kode auch in einem Webserver und bietet uns Web-basierte Formulare zur Dateneingabe an.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.
- Oder vielleicht schreiben wir auch ein Python-Programm, das die Daten unsere Datenbank mit Methoden der Artificial Intelligence (AI) und Machine Learning (ML) analysiert, um zukünftige Verkäufe vorherzusehen.
- Vielleicht läuft unser Python-Kode auch in einem Webserver und bietet uns Web-basierte Formulare zur Dateneingabe an.
- Alle das sind eher große Applikationen.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.
- Oder vielleicht schreiben wir auch ein Python-Programm, das die Daten unsere Datenbank mit Methoden der Artificial Intelligence (AI) und Machine Learning (ML) analysiert, um zukünftige Verkäufe vorherzusehen.
- Vielleicht läuft unser Python-Kode auch in einem Webserver und bietet uns Web-basierte Formulare zur Dateneingabe an.
- Alle das sind eher große Applikationen.
- Sie werden für spezielle Use-Cases von Softwareingenieuren entwickelt und von Programmierern implementiert und gewartet ... und sind daher teuer.

# Datenbankzugriff mit Python



- Wir haben bereits diskutiert, wie wir uns mit Python auf unser DBMS verbinden können.
- Wir könnten eine Applikation mit einer schönen Benutzeroberfläche programmieren.
- Wir könnten auch ein Python-Programm schreiben, das Daten aus verschiedenen Quellen, z. B. Webseiten, Webservices, anderen Datenbanken oder Sensoren in Maschinen zieht und in unsere DB einfügt.
- Oder vielleicht schreiben wir auch ein Python-Programm, das die Daten unsere Datenbank mit Methoden der Artificial Intelligence (AI) und Machine Learning (ML) analysiert, um zukünftige Verkäufe vorherzusehen.
- Vielleicht läuft unser Python-Kode auch in einem Webserver und bietet uns Web-basierte Formulare zur Dateneingabe an.
- Alle das sind eher große Applikationen.
- Sie werden für spezielle Use-Cases von Softwareingenieuren entwickelt und von Programmierern implementiert und gewartet ... und sind daher teuer.
- Sie werden also eher selten in kleinen Firmen auftauchen.

# Billige Datenbankanwendungen



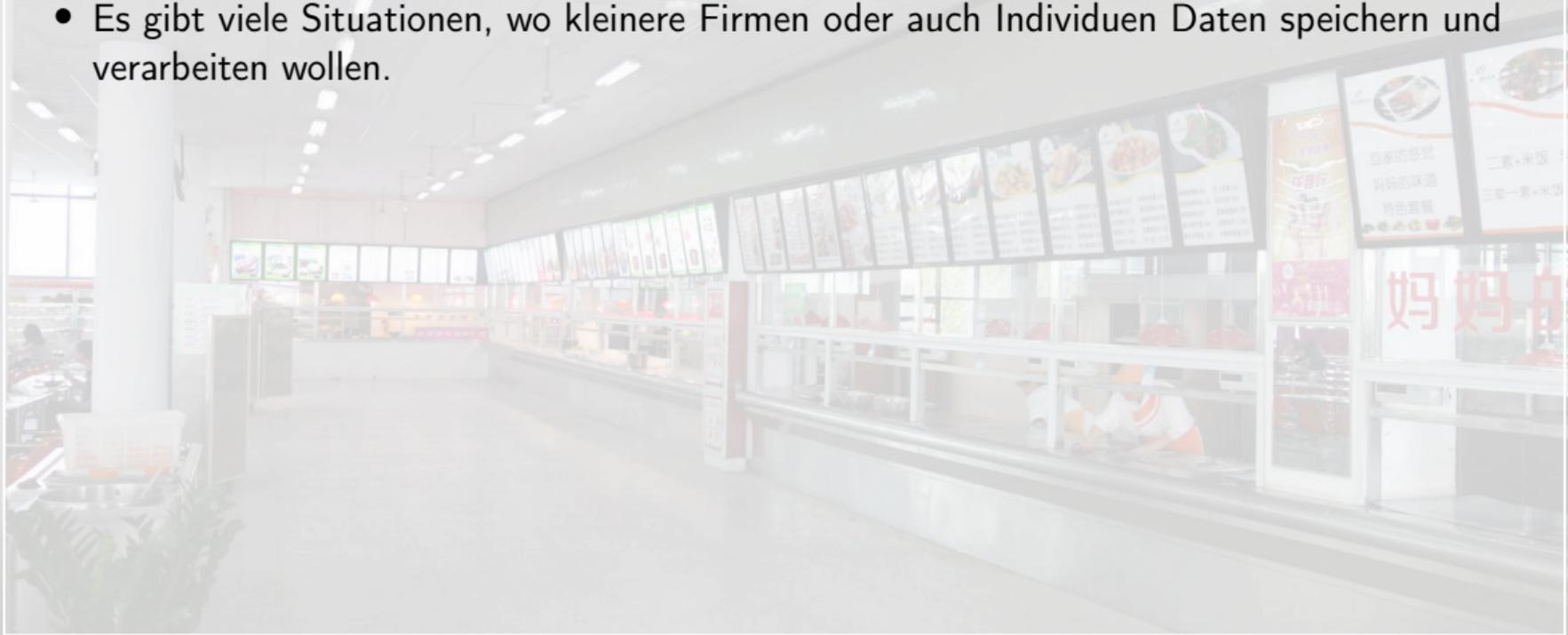
- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.



# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.



# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.
- Diese können dann oft nach kommerziellen Lösungen suchen, aber die gibt es nicht für jeden Anwendungsfall und manchmal sind sie einfach zu teuer.



# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.
- Diese können dann oft nach kommerziellen Lösungen suchen, aber die gibt es nicht für jeden Anwendungsfall und manchmal sind sie einfach zu teuer.
- Manchmal sind wir auch in einer frühen Phase des Ausrollens einer Datenbank in einer großen Organisation.

# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.
- Diese können dann oft nach kommerziellen Lösungen suchen, aber die gibt es nicht für jeden Anwendungsfall und manchmal sind sie einfach zu teuer.
- Manchmal sind wir auch in einer frühen Phase des Ausrollens einer Datenbank in einer großen Organisation.
- Vielleicht ist ein großes Investment in der Zukunft geplant.

# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.
- Diese können dann oft nach kommerziellen Lösungen suchen, aber die gibt es nicht für jeden Anwendungsfall und manchmal sind sie einfach zu teuer.
- Manchmal sind wir auch in einer frühen Phase des Ausrollens einer Datenbank in einer großen Organisation.
- Vielleicht ist ein großes Investment in der Zukunft geplant.
- Aber wir müssen erstmal die Organisation davon überzeugen, dass unser Projekt sinnvoll ist.

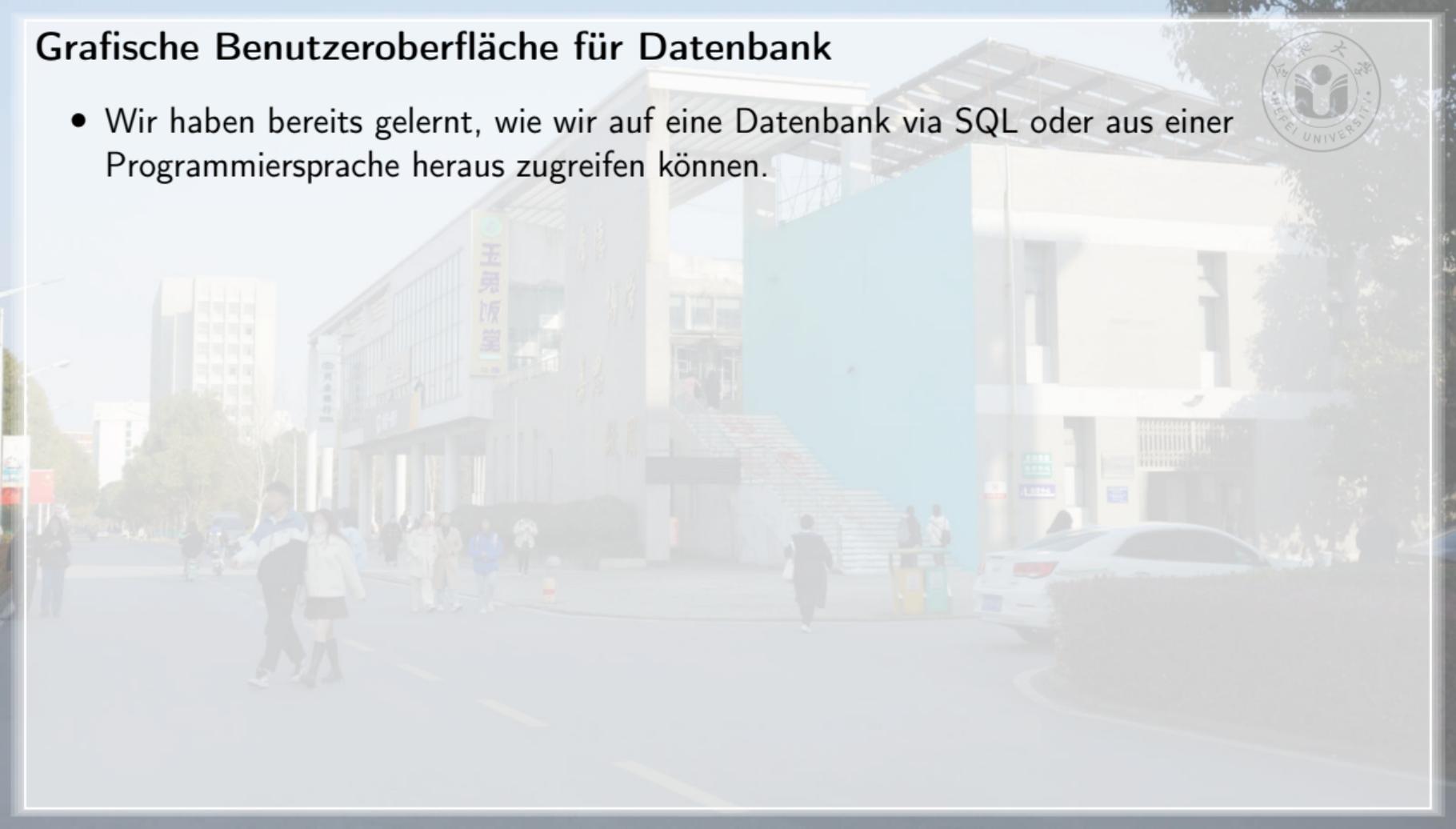
# Billige Datenbankanwendungen



- Aber selbst in Umgebungen mit weniger großen Budgets ist es oft sinnvoll Datenbanken zu benutzen.
- Es gibt viele Situationen, wo kleinere Firmen oder auch Individuen Daten speichern und verarbeiten wollen.
- Diese können dann oft nach kommerziellen Lösungen suchen, aber die gibt es nicht für jeden Anwendungsfall und manchmal sind sie einfach zu teuer.
- Manchmal sind wir auch in einer frühen Phase des Ausrollens einer Datenbank in einer großen Organisation.
- Vielleicht ist ein großes Investment in der Zukunft geplant.
- Aber wir müssen erstmal die Organisation davon überzeugen, dass unser Projekt sinnvoll ist.
- We wollen dann einen schnellen und billigen Prototyp unserer Applikation, mit dem wir zumindest Daten erstellen und auslesen können.

# Grafische Benutzeroberfläche für Datenbank

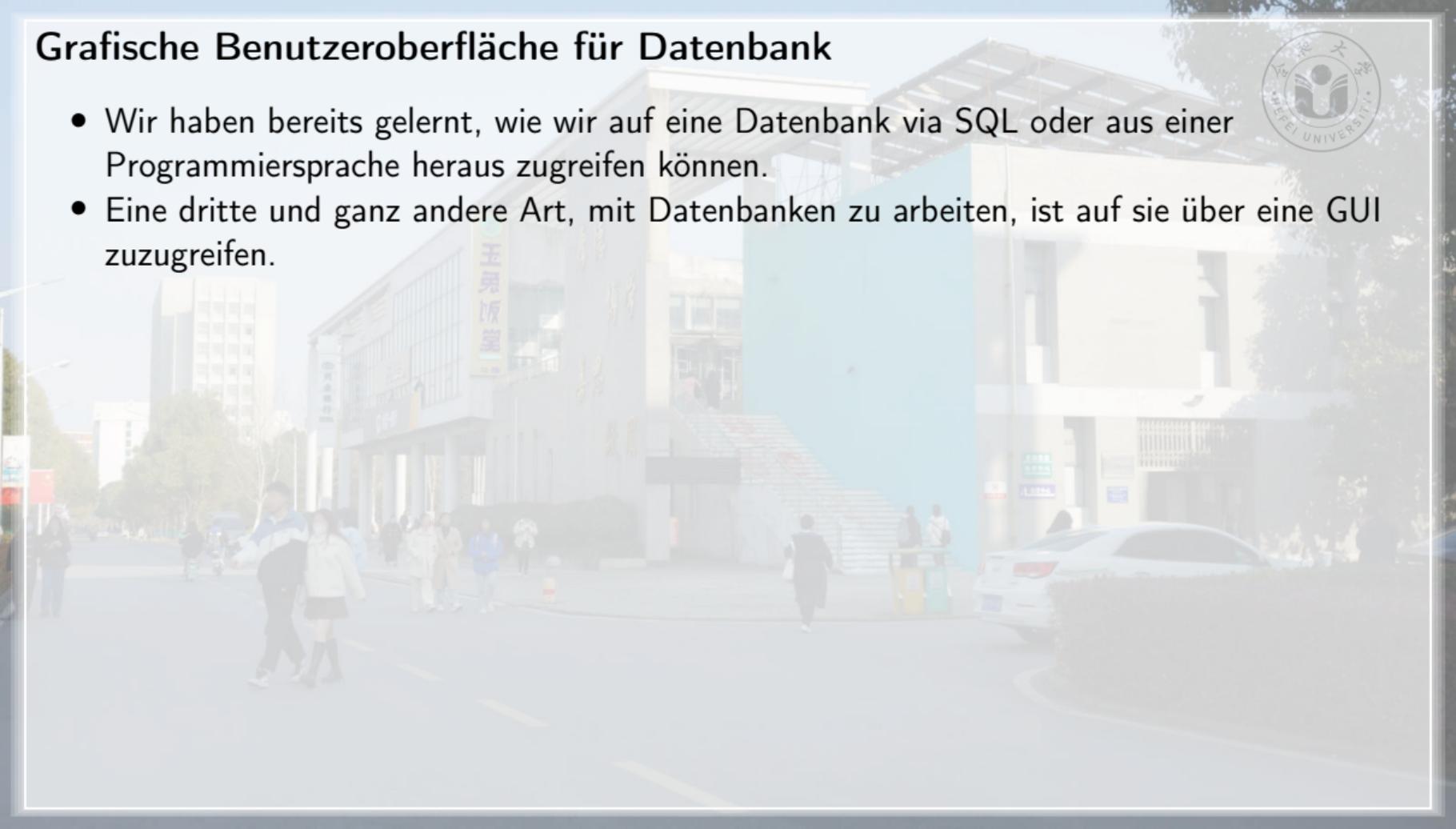
- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.



# Grafische Benutzeroberfläche für Datenbank



- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.
- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.



# Grafische Benutzeroberfläche für Datenbank



- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.
- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.
- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.

# Grafische Benutzeroberfläche für Datenbank



- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.
- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.
- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.
- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.

# Grafische Benutzeroberfläche für Datenbank



- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.
- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.
- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.
- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.
- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.

# Grafische Benutzeroberfläche für Datenbank



- Wir haben bereits gelernt, wie wir auf eine Datenbank via SQL oder aus einer Programmiersprache heraus zugreifen können.
- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.
- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.
- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.
- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.
- Wie Sie gelernt haben, können „echte DBMSe“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanageten Daten zu.

# Grafische Benutzeroberfläche für Datenbank



- Eine dritte und ganz andere Art, mit Datenbanken zu arbeiten, ist auf sie über eine GUI zuzugreifen.
- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.
- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.
- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.
- Wie Sie gelernt haben, können „echte DBMS“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanageten Daten zu.
- Daher können weder Microsoft Access noch LibreOffice Base als DBMS für Applikationen empfohlen werden, die über Hobbies und einfache Aufgaben aus kleinen Büros hinausgehen.

# Grafische Benutzeroberfläche für Datenbank



- Typische Beispiele so einer GUI sind das kommerzielle Microsoft Access<sup>6,20,88</sup> und die kostenfreie Open Source Software LibreOffice Base<sup>35,74</sup>.
- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.
- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.
- Wie Sie gelernt haben, können „echte DBMS“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanageten Daten zu.
- Daher können weder Microsoft Access noch LibreOffice Base als DBMS für Applikationen empfohlen werden, die über Hobbies und einfache Aufgaben aus kleinen Büros hinausgehen.
- Sie haben aber beide coole Werkzeuge, wie Berichte und Formulare.

# Grafische Benutzeroberfläche für Datenbank



- Beide erlauben es, Datenbanken in einzelnen Dateien zu erstellen und auf dem lokalen Computer auf sie zuzugreifen.
- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.
- Wie Sie gelernt haben, können „echte DBMS“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanageten Daten zu.
- Daher können weder Microsoft Access noch LibreOffice Base als DBMS für Applikationen empfohlen werden, die über Hobbies und einfache Aufgaben aus kleinen Büros hinausgehen.
- Sie haben aber beide coole Werkzeuge, wie Berichte und Formulare.
- Anstatt sie als DBMS zu verwenden, können wir sie als GUI benutzen, die auf eine Datenbank in einem *anderen* DBMS liegt!

# Grafische Benutzeroberfläche für Datenbank



- Sie realisieren solche Datenbanken als einzelne lokale Dateien auf denen der aktuelle Benutzer arbeiten kann.
- Wie Sie gelernt haben, können „echte DBMS“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanageten Daten zu.
- Daher können weder Microsoft Access noch LibreOffice Base als DBMS für Applikationen empfohlen werden, die über Hobbies und einfache Aufgaben aus kleinen Büros hinausgehen.
- Sie haben aber beide coole Werkzeuge, wie Berichte und Formulare.
- Anstatt sie als DBMS zu verwenden, können wir sie als GUI benutzen, die auf eine Datenbank in einem *anderen* DBMS liegt!
- Und dann können wir diese coolen Werkzeuge benutzen, um mit einer Datenbank in einem professionellen DBMS zu arbeiten-

# Grafische Benutzeroberfläche für Datenbank



- Wie Sie gelernt haben, können „echte DBMS“ Datenbanken auf einem Computer speichern und Benutzer greifen von anderen Computern auf die zentral gemanagerten Daten zu.
- Daher können weder Microsoft Access noch LibreOffice Base als DBMS für Applikationen empfohlen werden, die über Hobbies und einfache Aufgaben aus kleinen Büros hinausgehen.
- Sie haben aber beide coole Werkzeuge, wie Berichte und Formulare.
- Anstatt sie als DBMS zu verwenden, können wir sie als GUI benutzen, die auf eine Datenbank in einem *anderen* DBMS liegt!
- Und dann können wir diese coolen Werkzeuge benutzen, um mit einer Datenbank in einem professionellen DBMS zu arbeiten-
- Und genau das machen wir nun, mit dem kostenlosen LibreOffice Base.



Mit der Datenbank verbinden



## Mit der Datenbank verbinden

- Wir wollen uns nun von LibreOffice Base aus auf unsere Fabrik-Datenbank im PostgreSQL DBMS verbinden.



## Mit der Datenbank verbinden



- Wir wollen uns nun von LibreOffice Base aus auf unsere Fabrik-Datenbank im PostgreSQL DBMS verbinden.
- Dazu öffnen wir zuerst LibreOffice Base, so wie wir das in der letzten Einheit gelernt haben.

# Mit der Datenbank verbinden



- Wir wollen uns nun von LibreOffice Base aus auf unsere Fabrik-Datenbank im PostgreSQL DBMS verbinden.
- Dazu öffnen wir zuerst LibreOffice Base, so wie wir das in der letzten Einheit gelernt haben.
- Der PostgreSQL Server muss auch laufen.

# Mit der Datenbank verbinden



- Im Startbildschirm von LibreOffice Base wählen wir „Connect to existing database“ aus.

Select database

Save and proceed

**Welcome to the LibreOffice Database Wizard**

Use the Database Wizard to create a new database, open an existing database file, or connect to a database stored on a server.

What do you want to do?

Create a new database

Embedded database:

Open an existing database file

Recently used:

**Connect to an existing database**

# Mit der Datenbank verbinden

- Wir klicken auf as Drop-Down Menü für Datenbankarten.



Select database

Connection settings

Set up user authentication

Save and proceed

### Welcome to the LibreOffice Database Wizard

Use the Database Wizard to create a new database, open an existing database file, or connect to a database stored on a server.

What do you want to do?

Create a new database

Embedded database:

Open an existing database file

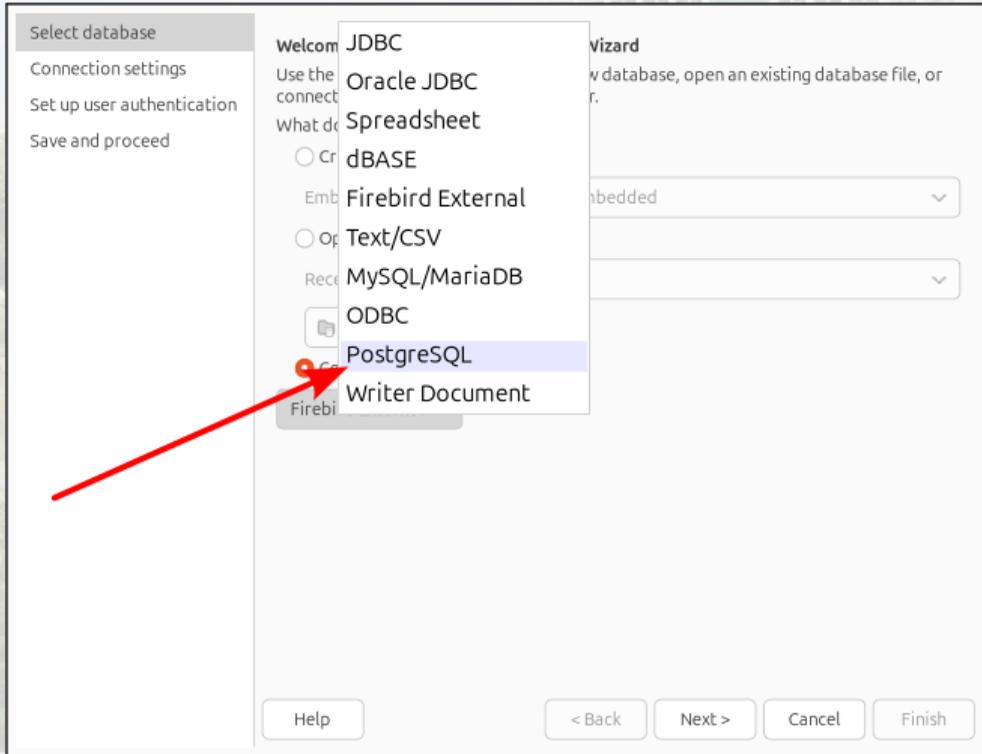
Recently used:

Connect to an existing database

# Mit der Datenbank verbinden



- Wir klicken auf as Drop-Down Menü für Datenbankarten.
- Und wir wählen PostgreSQL als Datenbankart aus.



# Mit der Datenbank verbinden

- Wir klicken auf **Next**.



Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Welcome to the LibreOffice Database Wizard

Use the Database Wizard to create a new database, open an existing database file, or connect to a database stored on a server.

What do you want to do?

Create a new database

Embedded database:

Open an existing database file

Recently used:

Connect to an existing database

A red arrow points from the bottom center of the wizard window towards the 'Next >' button.

# Mit der Datenbank verbinden

- Nun müssen wir die Informationen über unsere Datenbank eingeben.



Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Set up a connection to a PostgreSQL database

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string.  
Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

Three red arrows originate from a single point on the left and point to the 'Database name', 'Server', and 'Port number' input fields. A fourth red arrow points from the 'Next >' button to the right.

# Mit der Datenbank verbinden



- Nun müssen wir die Informationen über unsere Datenbank eingeben
  - Der Name der Datenbank ist `factory`.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Set up a connection to a PostgreSQL database

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string.  
Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

# Mit der Datenbank verbinden



- Nun müssen wir die Informationen über unsere Datenbank eingeben
  - Als Server wählen wir localhost aus, weil user DBMS auf unserem lokalen Computer läuft.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Set up a connection to a PostgreSQL database

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string.  
Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



- Nun müssen wir die Informationen über unsere Datenbank eingeben
  - Wenn unser DBMS auf einem anderen Computer laufen würde, dann würden wir dessen IP-Adresse eingeben.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Set up a connection to a PostgreSQL database

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string. Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



- Nun müssen wir die Informationen über unsere Datenbank eingeben
  - Als Port wählen wir den Standard-PostgreSQL-Port 5321 aus.

The screenshot shows a dialog box titled "Set up a connection to a PostgreSQL database". On the left, there is a sidebar with four options: "Select database", "Set up PostgreSQL connection" (which is highlighted), "Set up user authentication", and "Save and proceed". The main area contains the following text and fields:

**Set up a connection to a PostgreSQL database**

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string.  
Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

At the bottom, there are five buttons: "Help", "< Back", "Next >", "Cancel", and "Finish".

Red arrows point from a common origin on the left to the "Set up PostgreSQL connection" sidebar item, the "Database name" field, the "Server" field, the "Port number" field, and the "Next >" button.

# Mit der Datenbank verbinden



- Nun müssen wir die Informationen über unsere Datenbank eingeben
  - Dann klicken wir **Next**.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

### Set up a connection to a PostgreSQL database

Please enter the required information to connect to a PostgreSQL database, either by entering the host name, port number and server, or by entering the connection string.  
Please contact your system administrator if you are unsure

Database name:

Server:

Port number:  - + Default: 5432

Alternatively, enter the driver-specific connection string here

# Mit der Datenbank verbinden

- Nun geben wir die Authentikationsinformationen ein.



Select database

Set up PostgreSQL connection

**Set up user authentication**

Save and proceed

**Set up the user authentication**

Some databases require you to enter a user name.

User name

Password required

Three red arrows originate from the left side of the window. One points to the 'Set up user authentication' section header, another points to the 'Password required' checkbox, and the third points to the 'Test Connection' button.

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Mit diesen Informationen wird sich LibreOffice Base in unser DBMS einloggen.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

**Set up the user authentication**

Some databases require you to enter a user name.

User name

Password required

Test Connection

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Wir können den Benutzernamen eingeben, nämlich `boss`.

The screenshot shows a dialog box titled "Set up the user authentication". On the left, a sidebar contains the following options: "Select database", "Set up PostgreSQL connection", "Set up user authentication" (which is highlighted), and "Save and proceed". The main area of the dialog has the heading "Set up the user authentication" and the text "Some databases require you to enter a user name." Below this, there is a "User name" input field containing the text "boss". To the right of the input field is a checked checkbox labeled "Password required". A "Test Connection" button is located to the right of the input field. At the bottom of the dialog, there are five buttons: "Help", "< Back", "Next >", "Cancel", and "Finish". Three red arrows are overlaid on the image: one points to the "Set up user authentication" option in the sidebar, another points to the "Password required" checkbox, and a third points to the "Test Connection" button.

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Wir hatten das Passwort `superboss123` für diesen Benutzer gesetzt, aber wir können es nicht eingeben.

The screenshot shows a dialog box titled "Set up the user authentication". On the left, there is a sidebar with four options: "Select database", "Set up PostgreSQL connection", "Set up user authentication" (which is highlighted), and "Save and proceed". The main area of the dialog has the heading "Set up the user authentication" and a sub-heading "Some databases require you to enter a user name.". Below this, there is a "User name" input field containing the text "boss". To the right of the input field is a checked checkbox labeled "Password required". At the bottom right of the main area is a "Test Connection" button. At the bottom of the dialog, there are five buttons: "Help", "< Back", "Next >", "Cancel", and "Finish". Three red arrows originate from the left side of the dialog: one points to the "User name" input field, another points to the "Password required" checkbox, and a third points to the "Test Connection" button.

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Es muss jedesmal eingetippt werden, wenn wir diese Verbindung benutzen.

The screenshot shows a dialog box for setting up a PostgreSQL connection. On the left, a sidebar lists steps: 'Select database', 'Set up PostgreSQL connection', 'Set up user authentication' (highlighted), and 'Save and proceed'. The main area is titled 'Set up the user authentication' and contains the text 'Some databases require you to enter a user name.' Below this, there is a 'User name' field with the value 'boss' and a checked checkbox labeled 'Password required'. A 'Test Connection' button is located to the right. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Cancel', and 'Finish'. Three red arrows point from the left towards the 'Set up user authentication' step, the 'Password required' checkbox, and the 'Test Connection' button.

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Das ist wahrscheinlich damit keine sensitiven Daten wie Passwörter in einem LibreOffice Base-Dokument gespeichert werden können.

Select database

Set up PostgreSQL connection

Set up user authentication

Save and proceed

**Set up the user authentication**

Some databases require you to enter a user name.

User name

Password required

Test Connection

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - So können keine Zugangsdaten verloren gehen oder aus Versehen mit dem Dokument an andere Benutzer weitergegeben werden.

The screenshot shows a dialog box titled "Set up the user authentication". On the left, a sidebar lists steps: "Select database", "Set up PostgreSQL connection", "Set up user authentication" (highlighted), and "Save and proceed". The main area contains the text "Some databases require you to enter a user name." followed by a "User name" field containing "boss". Below this is a checked checkbox labeled "Password required". A "Test Connection" button is located to the right. At the bottom, there are buttons for "Help", "< Back", "Next >", "Cancel", and "Finish". Three red arrows point from the left towards the "Set up user authentication" step, the "Password required" checkbox, and the "Test Connection" button.

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - So oder so, wir müssen **Password required** auswählen, denn der Benutzer **boss** muss sich mit einem Passwort anmelden.

Select database

Set up PostgreSQL connection

**Set up user authentication**

Save and proceed

**Set up the user authentication**

Some databases require you to enter a user name.

User name

Password required

# Mit der Datenbank verbinden



- Nun geben wir die Authentikationsinformationen ein:
  - Dann klicken wir **Test Connection**.

The screenshot shows a dialog box titled "Set up the user authentication". On the left, a sidebar contains four steps: "Select database", "Set up PostgreSQL connection", "Set up user authentication" (which is highlighted), and "Save and proceed". The main area of the dialog is titled "Set up the user authentication" and contains the text "Some databases require you to enter a user name." Below this, there is a "User name" field with the text "boss" entered. To the right of the field is a checked checkbox labeled "Password required". At the bottom right of the main area is a button labeled "Test Connection". At the bottom of the dialog are five buttons: "Help", "< Back", "Next >", "Cancel", and "Finish". Three red arrows originate from the left side of the dialog: one points to the "Set up user authentication" step in the sidebar, another points to the "Password required" checkbox, and a third points to the "Test Connection" button.

## Mit der Datenbank verbinden



- Das Authentifikationsfenster ploppt auf, wir tippen das Passwort `superboss123` ein und drücken `OK`.

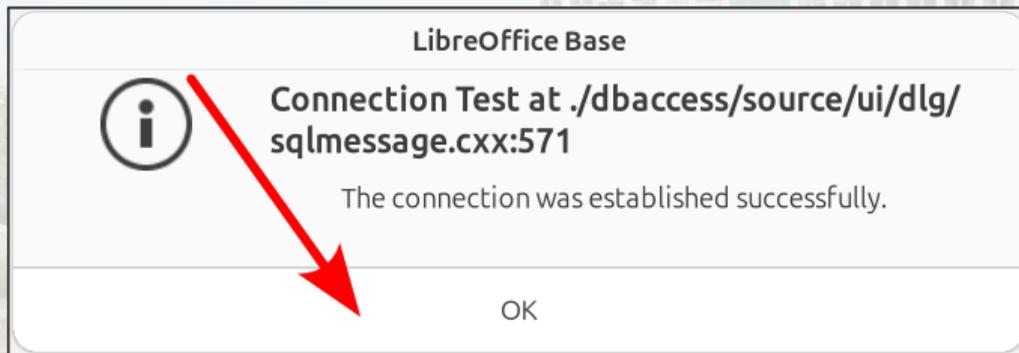
Message from server:  
A password is needed to connect to the data source .  
Enter user name and password for:

User name:

Password:

## Mit der Datenbank verbinden

- Der Test war erfolgreich. Wir klicken .



# Mit der Datenbank verbinden



- Zurück in der Eingabemaske können wir nun **Next** klicken.

Select database

Set up PostgreSQL connection

**Set up user authentication**

Save and proceed

### Set up the user authentication

Some databases require you to enter a user name.

User name

Password required

# Mit der Datenbank verbinden



- Das bringt uns in das letzte Fenster des Dokument-Erstellungs-Dialogs.

The screenshot shows a dialog box with a sidebar on the left and a main content area on the right. The sidebar contains four options: 'Select database', 'Set up PostgreSQL connection', 'Set up user authentication', and 'Save and proceed'. The 'Save and proceed' option is selected and highlighted. The main content area is titled 'Decide How to Proceed After Saving the Database' and contains the following text and options:

Do you want the wizard to register the database in LibreOffice?

- Yes, register the database for me
- No, do not register the database

After the database file has been saved, what do you want to do?

- Open the database for editing
- Create tables using the table wizard

Click 'Finish' to save the database.

At the bottom of the dialog, there are five buttons: 'Help', '< Back', 'Next >', 'Cancel', and 'Finish'. Three red arrows originate from a single point on the left side of the dialog. One arrow points to the 'Save and proceed' button in the sidebar. Another arrow points to the 'No, do not register the database' radio button. The third arrow points to the 'Finish' button at the bottom right.

# Mit der Datenbank verbinden



- Das bringt uns in das letzte Fenster des Dokument-Erstellungs-Dialogs.
- Wir wählen aus, dass wir die Datenbank nicht registrieren wollen.

Select database

Set up PostgreSQL connection

Set up user authentication

**Save and proceed**

**Decide How to Proceed After Saving the Database**

Do you want the wizard to register the database in LibreOffice?

Yes, register the database for me

No, do not register the database

After the database file has been saved, what do you want to do?

Open the database for editing

Create tables using the table wizard

Click 'Finish' to save the database.

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



- Wir wählen aus, dass wir die Datenbank nicht registrieren wollen.
- Wir wählen aus, dass wir die Datenbank zum Bearbeiten öffnen wollen.

The screenshot shows a dialog box titled "Decide How to Proceed After Saving the Database". On the left, a sidebar lists steps: "Select database", "Set up PostgreSQL connection", "Set up user authentication", and "Save and proceed" (which is highlighted). The main area contains the following text and options:

**Decide How to Proceed After Saving the Database**

Do you want the wizard to register the database in LibreOffice?

Yes, register the database for me

No, do not register the database

After the database file has been saved, what do you want to do?

Open the database for editing

Create tables using the table wizard

Click 'Finish' to save the database.

At the bottom, there are four buttons: "Help", "< Back", "Next >", "Cancel", and "Finish". Three red arrows originate from a single point on the left and point to the "Save and proceed" button, the "No, do not register the database" radio button, and the "Finish" button.

# Mit der Datenbank verbinden



- Wir wählen aus, dass wir die Datenbank zum Bearbeiten öffnen wollen.
- Schlussendlich können wir auf **Finish** klicken.

Select database

Set up PostgreSQL connection

Set up user authentication

**Save and proceed**

**Decide How to Proceed After Saving the Database**

Do you want the wizard to register the database in LibreOffice?

Yes, register the database for me

No, do not register the database

After the database file has been saved, what do you want to do?

Open the database for editing

Create tables using the table wizard

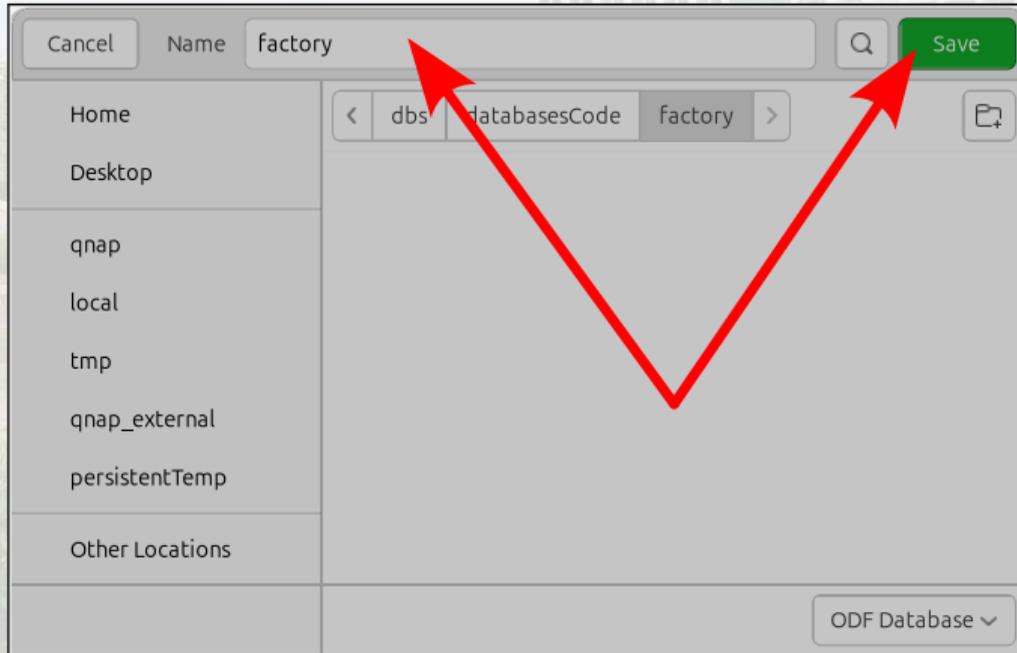
Click 'Finish' to save the database.

Help < Back Next > Cancel Finish

# Mit der Datenbank verbinden



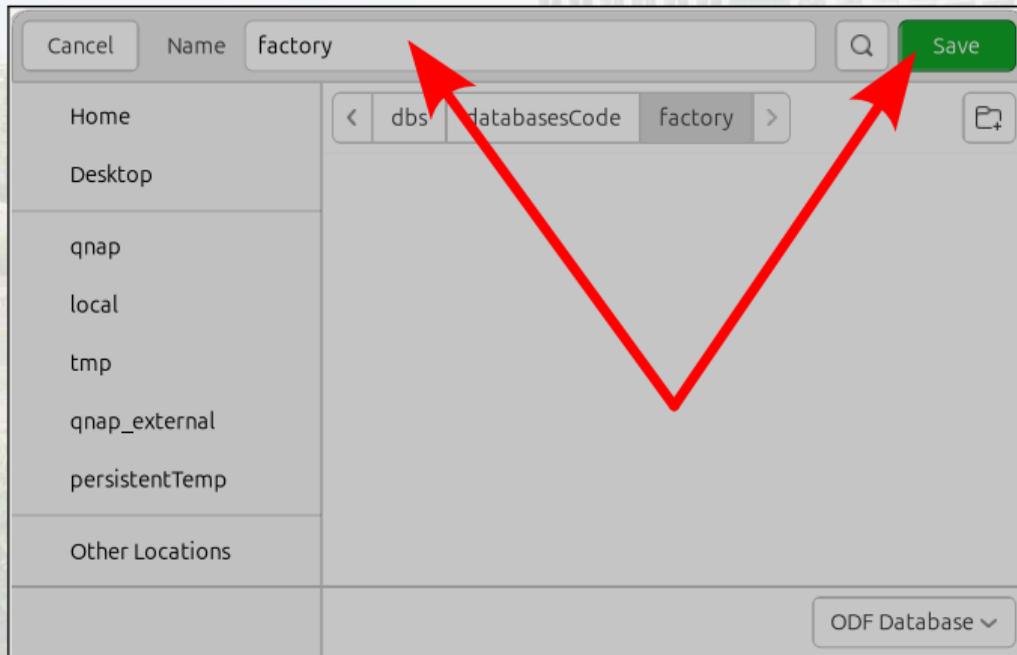
- Wir müssen das LibreOffice Base Dokument nun in einer passenden Datei speichern.



# Mit der Datenbank verbinden



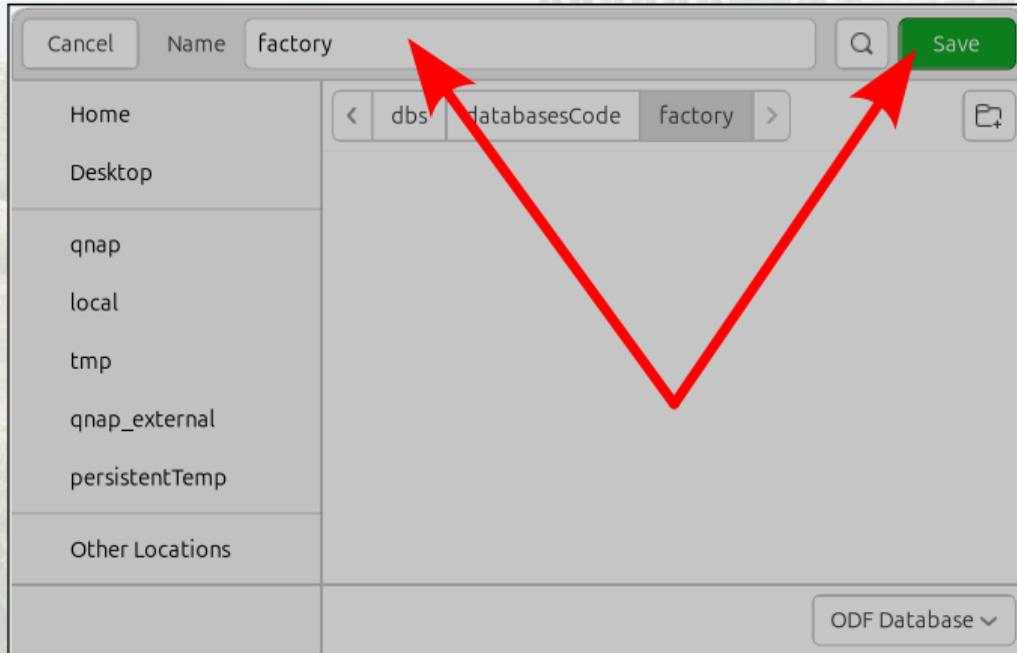
- Wir müssen das LibreOffice Base Dokument nun in einer passenden Datei speichern.
- Der Dateityp ist `odf`, was im Grunde einer ZIP-komprimierten Kollektion von XML-Doumenten entspricht.



# Mit der Datenbank verbinden

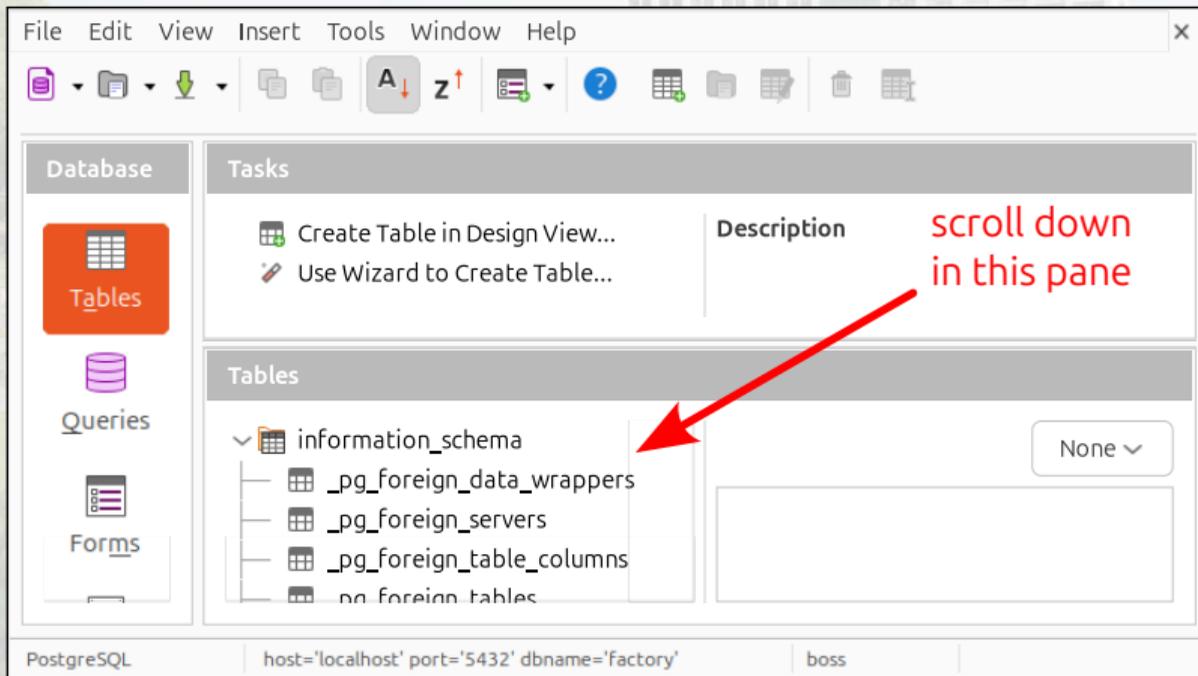


- Der Dateityp ist `odb`, was im Grunde einer ZIP-komprimierten Kollektion von XML-Doumenten entspricht.
- Egal. Wir wählen `factory.odb` als Dateiname und klicken `Save`.



# Mit der Datenbank verbinden

- Nun öffnet sich die Datenbank-GUI.



# Mit der Datenbank verbinden

- Nun öffnet sich die Datenbank-GUI.
- Wir sehen ganz viele Informationen.



File Edit View Insert Tools Window Help

Database

Tasks

Tables

Queries

Forms

information\_schema

\_pg\_foreign\_data\_wrappers

\_pg\_foreign\_servers

\_pg\_foreign\_table\_columns

\_pg\_foreign\_tables

None v

PostgreSQL | host='localhost' port='5432' dbname='factory' | boss

scroll down in this pane

# Mit der Datenbank verbinden



- Wir sehen ganz viele Informationen.
- Schauen wir erstmal nach den Datenbankobjekten, die wir selbst bereits erstellt haben.

File Edit View Insert Tools Window Help

Database

Tasks

Tables

information\_schema

- \_pg\_foreign\_data\_wrappers
- \_pg\_foreign\_servers
- \_pg\_foreign\_table\_columns
- \_pg\_foreign\_tables

PostgreSQL | host='localhost' port='5432' dbname='factory' | boss

scroll down in this pane

# Mit der Datenbank verbinden



- Schauen wir erstmal nach den Datenbankobjekten, die wir selbst bereits erstellt haben.
- Wir gehen in die **Tables**-Ansicht und skrollen nach unten.

The screenshot shows the Microsoft Access application window. The menu bar includes File, Edit, View, Insert, Tools, Window, and Help. The toolbar contains various icons for file operations and database actions. The interface is divided into several panes:

- Database:** A vertical sidebar on the left with icons for Tables (selected), Queries, and Forms.
- Tasks:** A pane with two options: "Create Table in Design View..." and "Use Wizard to Create Table...".
- Tables:** A tree view showing the database structure. The "information\_schema" folder is expanded, revealing sub-items: "\_pg\_foreign\_data\_wrappers", "\_pg\_foreign\_servers", "\_pg\_foreign\_table\_columns", and "\_pg\_foreign\_tables".
- Description:** A pane on the right with the text "scroll down in this pane" in red and a red arrow pointing to the "information\_schema" folder in the Tables pane.

At the bottom of the window, the status bar displays: PostgreSQL | host='localhost' port='5432' dbname='factory' | boss



# Mit der Datenbank verbinden

- Wir gehen in die `Tables`-Ansicht und skrollen nach unten.
- Wenn wir bis in den `public`-Knoten skrollen, dann finden wir unsere drei Tabellen und die Sicht `sale`.

File Edit View Insert Tools Window Help

Database

Tasks

Create Table in Design View...  
Use Wizard to Create Table...

Description we find our tables under point "public"

Tables

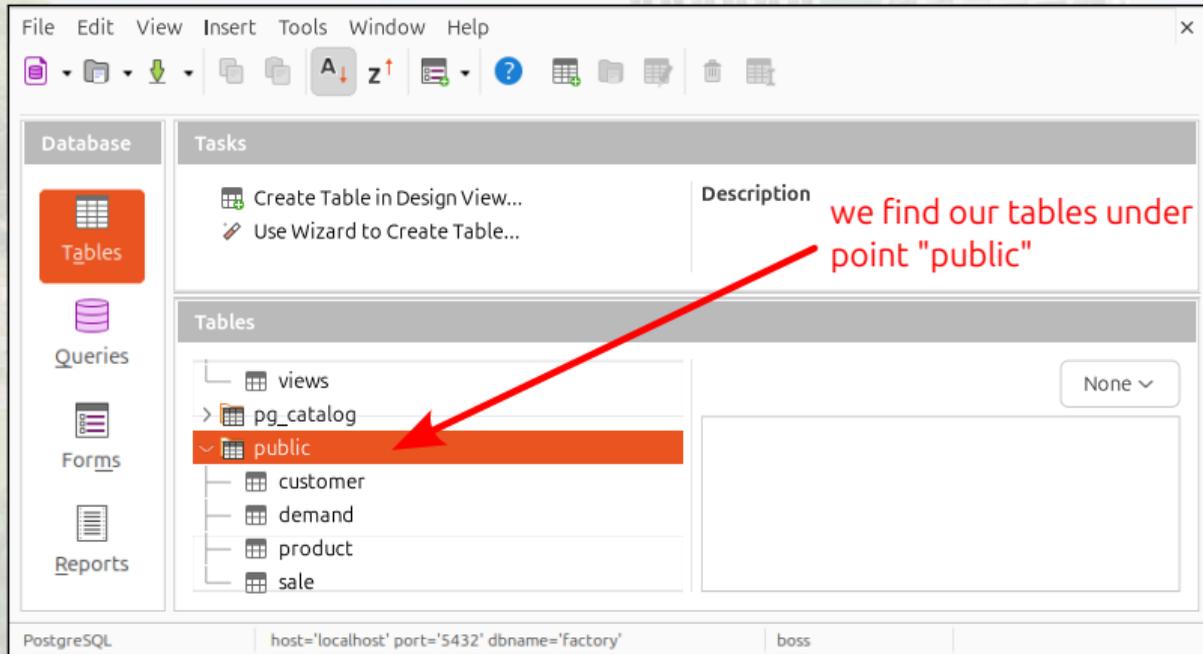
views  
pg\_catalog  
public  
customer  
demand  
product  
sale

PostgreSQL host='localhost' port='5432' dbname='factory' boss

# Mit der Datenbank verbinden



- Wenn wir bis in den `public`-Knoten skrollen, dann finden wir unsere drei Tabellen und die Sicht `sale`.
- LibreOffice Base kann unsere Fabrik-Datenbank und die Objekte darin sehen.





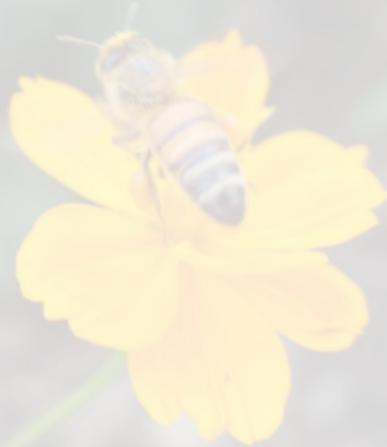
# Zusammenfassung



# Zusammenfassung



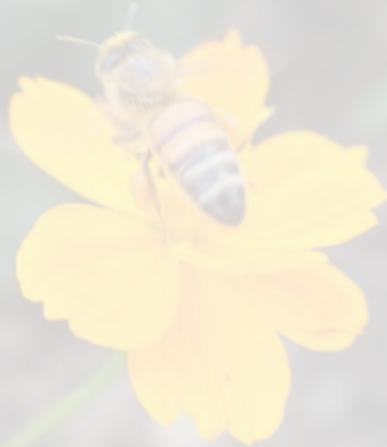
- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.



# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.



# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.
- Das würde so ähnlich auch mit anderen DBMSen funktionieren.



# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.
- Das würde so ähnlich auch mit anderen DBMSen funktionieren.
- Wir hätten auch Microsoft Access anstatt von LibreOffice Base benutzen können.

# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.
- Das würde so ähnlich auch mit anderen DBMSen funktionieren.
- Wir hätten auch Microsoft Access anstatt von LibreOffice Base benutzen können.
- Der wichtige Punkt ist, dass es Werkzeuge wie Microsoft Access und LibreOffice Base von Drittanbietern gibt und das diese auch auf Datenbanken verbinden können, die von Servern wie PostgreSQL, MySQL, Oracle Database, oder Microsoft SQL Server gemanaged werden.

# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.
- Das würde so ähnlich auch mit anderen DBMSen funktionieren.
- Wir hätten auch Microsoft Access anstatt von LibreOffice Base benutzen können.
- Der wichtige Punkt ist, dass es Werkzeuge wie Microsoft Access und LibreOffice Base von Drittanbietern gibt und das diese such auf Datenbanken verbinden können, die von Servern wie PostgreSQL, MySQL, Oracle Database, oder Microsoft SQL Server gemanaged werden.
- Aber was werden wir damit alles machen können?

# Zusammenfassung



- Wir haben nun LibreOffice Base erfolgreich mit unserem PostgreSQL DBMS verbunden.
- LibreOffice Base kann sich nun bei dem PostgreSQL Server einloggen.
- Das würde so ähnlich auch mit anderen DBMSen funktionieren.
- Wir hätten auch Microsoft Access anstatt von LibreOffice Base benutzen können.
- Der wichtige Punkt ist, dass es Werkzeuge wie Microsoft Access und LibreOffice Base von Drittanbietern gibt und das diese auch auf Datenbanken verbinden können, die von Servern wie PostgreSQL, MySQL, Oracle Database, oder Microsoft SQL Server gemanaged werden.
- Aber was werden wir damit alles machen können?
- Bald werden wir es wissen.



谢谢您门！

Thank you!

Vielen Dank!



# References I



- [1] Dirk Angermann. *T-SQL-Abfragen für Microsoft SQL-Server 2022*. Blaufelden, Schwäbisch Hall, Baden-Württemberg, Germany: mitp Verlags GmbH & Co. KG, Juni 2024. ISBN: **978-3-7475-0633-2** (siehe S. 91).
- [2] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume I: Introduction to SQL Queries*. Tetras Publishing, Okt. 2018. ISBN: **978-1-9996172-4-0**. See also<sup>3</sup> (siehe S. 79, 91).
- [3] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume II: In-Depth Querying*. Tetras Publishing, Okt. 2018. ISBN: **978-1-9996172-5-7**. See also<sup>2</sup> (siehe S. 79, 91).
- [4] Daniel J. Barrett. *Efficient Linux at the Command Line*. Sebastopol, CA, USA: O'Reilly Media, Inc., Feb. 2022. ISBN: **978-1-0981-1340-7** (siehe S. 90, 93).
- [5] Daniel Bartholomew. *Learning the MariaDB Ecosystem: Enterprise-level Features for Scalability and Availability*. New York, NY, USA: Apress Media, LLC, Okt. 2019. ISBN: **978-1-4842-5514-8** (siehe S. 91).
- [6] Ben Beitler. *Hands-On Microsoft Access 2019*. Birmingham, England, UK: Packt Publishing Ltd, März 2020. ISBN: **978-1-83898-747-3** (siehe S. 20–27, 90, 91).
- [7] Peter Belknap, John Beresiewicz, Benoît Dageville, Karl Dias, Yakov Shafranovich und Khaled Yagoub. "A Decade of Oracle Database Manageability". 34(4):20–27, Dez. 2011. URL: [http://sites.computer.org/debull/A11dec/DODM%5C\\_V2.pdf](http://sites.computer.org/debull/A11dec/DODM%5C_V2.pdf) (siehe S. 91).
- [8] Tim Berners-Lee. *Re: Qualifiers on Hypertext links...* Geneva, Switzerland: World Wide Web project, European Organization for Nuclear Research (CERN) und Newsgroups: alt.hypertext, 6. Aug. 1991. URL: <https://www.w3.org/People/Berners-Lee/1991/08/art-6484.txt> (besucht am 2025-02-05) (siehe S. 90, 93).
- [9] Tim Berners-Lee, Roy T. Fielding und Henrik Frystyk Nielsen. *Hypertext Transfer Protocol -- HTTP/1.0. Request for Comments (RFC) 1945*. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Mai 1996. URL: <https://www.ietf.org/rfc/rfc1945.txt> (besucht am 2025-02-05) (siehe S. 90).
- [10] Alex Berson. *Client/Server Architecture*. 2. Aufl. Computer Communications Series. New York, NY, USA: McGraw-Hill, 29. März 1996. ISBN: **978-0-07-005664-0** (siehe S. 89).

# References II



- [11] Bernard Obeng Boateng. *Data Modeling with Microsoft Excel*. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2023. ISBN: 978-1-80324-028-2 (siehe S. 91).
- [12] Silvia Botros und Jeremy Tinley. *High Performance MySQL*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Nov. 2021. ISBN: 978-1-4920-8051-0 (siehe S. 91).
- [13] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: 978-0-13-769132-6 (siehe S. 91).
- [14] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: 978-1-78862-936-2 (siehe S. 89).
- [15] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. Request for Comments (RFC) 8259. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Dez. 2017. URL: <https://www.ietf.org/rfc/rfc8259.txt> (besucht am 2025-02-05) (siehe S. 90).
- [16] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen und Eve Maler, Hrsg. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation. Wakefield, MA, USA: World Wide Web Consortium (W3C), 26. Nov. 2008–7. Feb. 2013. URL: <http://www.w3.org/TR/2008/REC-xml-20081126> (besucht am 2024-12-15) (siehe S. 93).
- [17] Jason Cannon. *High Availability for the LAMP Stack*. Shelter Island, NY, USA: Manning Publications, Juni 2022 (siehe S. 90, 92).
- [18] Donald D. Chamberlin. "50 Years of Queries". *Communications of the ACM (CACM)* 67(8):110–121, Aug. 2024. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/3649887. URL: <https://cacm.acm.org/research/50-years-of-queries> (besucht am 2025-01-09) (siehe S. 91, 92).
- [19] Stuart Cheshire und Marc Krochmal. *Special-Use Domain Names*. Request for Comments (RFC) 6761. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Feb. 2013. URL: <https://www.ietf.org/rfc/rfc4253.txt> (besucht am 2025-02-27) (siehe S. 91).
- [20] Christmas, FL, USA: Simon Sez IT. *Microsoft Access 2021 – Beginner to Advanced*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2023. ISBN: 978-1-83546-911-8 (siehe S. 20–27, 90, 91).
- [21] David Clinton und Christopher Negus. *Ubuntu Linux Bible*. 10. Aufl. Bible Series. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 10. Nov. 2020. ISBN: 978-1-119-72233-5 (siehe S. 93).

# References III



- [22] Edgar Frank „Ted“ Codd. “A Relational Model of Data for Large Shared Data Banks”. *Communications of the ACM (CACM)* 13(6):377–387, Juni 1970. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/362384.362685. URL: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> (besucht am 2025-01-05) (siehe S. 92).
- [23] Coding Gears und Train Your Brain. *YAML Fundamentals for DevOps, Cloud and IaC Engineers*. Birmingham, England, UK: Packt Publishing Ltd, März 2022. ISBN: 978-1-80324-243-9 (siehe S. 93).
- [24] Timothy W. Cole und Myung-Ja K. Han. *XML for Catalogers and Metadata Librarians (Third Millennium Cataloging)*. 1. Aufl. Dublin, OH, USA: Libraries Unlimited, 23. Mai 2013. ISBN: 978-1-59884-519-8 (siehe S. 93).
- [25] “`csv` – CSV File Reading and Writing”. In: *Python 3 Documentation. The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library/csv.html> (besucht am 2024-11-14) (siehe S. 89).
- [26] *Database Language SQL*. Techn. Ber. ANSI X3.135-1986. Washington, D.C., USA: American National Standards Institute (ANSI), 1986 (siehe S. 92).
- [27] Matt David und Blake Barnhill. *How to Teach People SQL*. San Francisco, CA, USA: The Data School, Chart.io, Inc., 10. Dez. 2019–10. Apr. 2023. URL: <https://dataschool.com/how-to-teach-people-sql> (besucht am 2025-02-27) (siehe S. 92).
- [28] *Database Language SQL*. International Standard ISO 9075-1987. Geneva, Switzerland: International Organization for Standardization (ISO), 1987 (siehe S. 92).
- [29] Paul Deitel, Harvey Deitel und Abbey Deitel. *Internet & World Wide WebW[?]: How to Program*. 5. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Nov. 2011. ISBN: 978-0-13-299045-5 (siehe S. 93).
- [30] Pooyan Doozandeh und Frank E. Ritter. “Some Tips for Academic Writing and Using Microsoft Word”. *XRDS: Crossroads, The ACM Magazine for Students* 26(1):10–11, Herbst 2019. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 1528-4972. doi:10.1145/3351470 (siehe S. 91).
- [31] Ingy döt Net, Tina Müller, Pantelis Antoniou, Eemeli Aro, Thomas Smith, Oren Ben-Kiki und Clark C. Evans. *YAML Ain’t Markup Language (YAML™) version 1.2*. Revision 1.2.2. Seattle, WA, USA: YAML Language Development Team, 1. Okt. 2021. URL: <https://yaml.org/spec/1.2.2> (besucht am 2025-01-05) (siehe S. 93).

# References IV



- [32] Russell J.T. Dyer. *Learning MySQL and MariaDB*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2015. ISBN: 978-1-4493-6290-4 (siehe S. 91).
- [33] Donald E. Eastlake 3rd und Aliza R. Panitz. *Reserved Top Level DNS Names*. Request for Comments (RFC) 2606. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Juni 1999. URL: <https://www.ietf.org/rfc/rfc2606.txt> (besucht am 2025-02-27) (siehe S. 91).
- [34] *ECMAScript Language Specification*. Standard ECMA-262, 3rd Edition. Geneva, Switzerland: Ecma International, Dez. 1999. URL: [https://ecma-international.org/wp-content/uploads/ECMA-262\\_3rd\\_edition\\_december\\_1999.pdf](https://ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf) (besucht am 2024-12-15) (siehe S. 90).
- [35] Steve Fanning, Vasudev Narayanan, „flywire“, Olivier Hallot, Jean Hollis Weber, Jenna Sargent, Pulkit Krishna, Dan Lewis, Peter Schofield, Jochen Schiffers, Robert Großkopf, Jost Lange, Martin Fox, Hazel Russman, Steve Schwettman, Alain Romedenne, Andrew Pitonyak, Jean-Pierre Ledure, Drew Jensen und Randolph Gam. *Base Guide 7.3. Revision 1. Based on LibreOffice 7.3 Community*. Berlin, Germany: The Document Foundation, Aug. 2022. URL: <https://books.libreoffice.org/en/BG73/BG73-BaseGuide.pdf> (besucht am 2025-01-13) (siehe S. 20–27, 90).
- [36] Luca Ferrari und Enrico Pirozzi. *Learn PostgreSQL*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: 978-1-83763-564-1 (siehe S. 92).
- [37] Roy T. Fielding, Jim Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen und Tim Berners-Lee. *Hypertext Transfer Protocol -- HTTP/1.1*. Request for Comments (RFC) 2068. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Jan. 1997. URL: <https://www.ietf.org/rfc/rfc2068.txt> (besucht am 2025-02-05) (siehe S. 90).
- [38] Roy T. Fielding, Mark Nottingham und Julian F. Reschke. *HTTP Semantics*. Request for Comments (RFC) 9110. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Juni 2022. URL: <https://www.ietf.org/rfc/rfc9110.txt> (besucht am 2025-02-05) (siehe S. 90).
- [39] Jonas Gamalielsson und Björn Lundell. "Long-Term Sustainability of Open Source Software Communities beyond a Fork: A Case Study of LibreOffice". In: *8th IFIP WG 2.13 International Conference on Open Source Systems: Long-Term Sustainability OSS'2012*. 10.–13. Sep. 2012, Hammamet, Tunisia. Hrsg. von Imed Hammouda, Björn Lundell, Tommi Mikkonen und Walt Scacchi. Bd. 378. IFIP Advances in Information and Communication Technology (IFIPAICT). Berlin/Heidelberg, Germany: Springer-Verlag GmbH Germany, 2012, S. 29–47. ISSN: 1868-4238. ISBN: 978-3-642-33441-2. doi:10.1007/978-3-642-33442-9\_3 (siehe S. 90).

# References V



- [40] Dawn Griffiths. *Excel Cookbook – Recipes for Mastering Microsoft Excel*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2024. ISBN: [978-1-0981-4332-9](#) (siehe S. 91).
- [41] Ilya Grigorik. *HTTP Protocols*. Sebastopol, CA, USA: O'Reilly Media, Inc., Dez. 2017. ISBN: [978-1-4920-3046-1](#) (siehe S. 90).
- [42] Terry Halpin und Tony Morgan. *Information Modeling and Relational Databases*. 3. Aufl. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Juli 2024. ISBN: [978-0-443-23791-1](#) (siehe S. 92).
- [43] Jan L. Harrington. *Relational Database Design and Implementation*. 4. Aufl. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Apr. 2016. ISBN: [978-0-12-849902-3](#) (siehe S. 92).
- [44] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: [978-1-0981-0894-6](#) (siehe S. 90).
- [45] Christian Heimes. "[defusedxml](#) 0.7.1: XML Bomb Protection for Python stdlib Modules". In: 8. März 2021. URL: <https://pypi.org/project/defusedxml> (besucht am 2024-12-15) (siehe S. 93).
- [46] Matthew Helmke. *Ubuntu Linux Unleashed 2021 Edition*. 14. Aufl. Reading, MA, USA: Addison-Wesley Professional, Aug. 2020. ISBN: [978-0-13-668539-5](#) (siehe S. 90, 93).
- [47] Manuel Hoffmann, Frank Nagle und Yanuo Zhou. *The Value of Open Source Software*. Working Paper 24-038. Boston, MA, USA: Harvard Business School, 1. Jan. 2024. URL: [https://www.hbs.edu/ris/Publication%20Files/24-038\\_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf](https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf) (besucht am 2025-06-04) (siehe S. 92).
- [48] John Hunt. *A Beginners Guide to Python 3 Programming*. 2. Aufl. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2023. ISBN: [978-3-031-35121-1](#). doi:[10.1007/978-3-031-35122-8](#) (siehe S. 92).
- [49] *Information Technology – Database Languages – SQL – Part 1: Framework (SQL/Framework), Part 1*. International Standard ISO/IEC 9075-1:2023(E), Sixth Edition, (ANSI X3.135). Geneva, Switzerland: International Organization for Standardization (ISO) und International Electrotechnical Commission (IEC), Juni 2023. URL: [https://standards.iso.org/ittf/PubliclyAvailableStandards/ISO\\_IEC\\_9075-1\\_2023\\_ed\\_6\\_-\\_id\\_76583\\_Publication\\_PDF\\_\(en\).zip](https://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_9075-1_2023_ed_6_-_id_76583_Publication_PDF_(en).zip) (besucht am 2025-01-08). Consists of several parts, see <https://modern-sql.com/standard> for information where to obtain them. (Siehe S. 92).

# References VI



- [50] Katie Kodes. *Intro to XML, JSON, & YAML*. London, England, UK: Payhip, 2019–4. Sep. 2020 (siehe S. 93).
- [51] Darl Kuhn und Thomas Kyte. *Expert Oracle Database Architecture: Techniques and Solutions for High Performance and Productivity*. 4. Aufl. New York, NY, USA: Apress Media, LLC, Nov. 2021. ISBN: 978-1-4842-7499-6 (siehe S. 91).
- [52] James F. Kurose und Keith Ross. *Computer Networking: A Top Down Approach*. 8. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., 30. März 2020. ISBN: 978-0-13-668155-7 (siehe S. 92).
- [53] Jay LaCroix. *Mastering Ubuntu Server*. 4. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2022. ISBN: 978-1-80323-424-3 (siehe S. 92).
- [54] Joan Lambert und Curtis Frye. *Microsoft Office Step by Step (Office 2021 and Microsoft 365)*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Juni 2022. ISBN: 978-0-13-754493-6 (siehe S. 91).
- [55] Kent D. Lee und Steve Hubbard. *Data Structures and Algorithms with Python*. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2015. ISBN: 978-3-319-13071-2. doi:10.1007/978-3-319-13072-9 (siehe S. 92).
- [56] *LibreOffice – The Document Foundation*. Berlin, Germany: The Document Foundation, 2024. URL: <https://www.libreoffice.org> (besucht am 2024-12-12) (siehe S. 90, 91).
- [57] Gloria Lotha, Aakanksha Gaur, Erik Gregersen, Swati Chopra und William L. Hosch. "Client-Server Architecture". In: *Encyclopaedia Britannica*. Hrsg. von The Editors of Encyclopaedia Britannica. Chicago, IL, USA: Encyclopædia Britannica, Inc., 3. Jan. 2025. URL: <https://www.britannica.com/technology/client-server-architecture> (besucht am 2025-01-20) (siehe S. 89).
- [58] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 92).
- [59] *MariaDB Server Documentation*. Milpitas, CA, USA: MariaDB, 2025. URL: <https://mariadb.com/kb/en/documentation> (besucht am 2025-04-24) (siehe S. 91).
- [60] Ron McFadyen und Cindy Miller. *Relational Databases and Microsoft Access*. 3. Aufl. Palatine, IL, USA: Harper College, 2014–2019. URL: <https://harpercollege.pressbooks.pub/relationaldatabases> (besucht am 2025-04-11) (siehe S. 91).

# References VII



- [61] Jim Melton und Alan R. Simon. *SQL: 1999 – Understanding Relational Language Components*. The Morgan Kaufmann Series in Data Management Systems. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Juni 2001. ISBN: 978-1-55860-456-8 (siehe S. 92).
- [62] *Microsoft Word*. Redmond, WA, USA: Microsoft Corporation, 2024. URL: <https://www.microsoft.com/en-us/microsoft-365/word> (besucht am 2024-12-12) (siehe S. 91).
- [63] Cameron Newham und Bill Rosenblatt. *Learning the Bash Shell – Unix Shell Programming: Covers Bash 3.0*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., 2005. ISBN: 978-0-596-00965-6 (siehe S. 89).
- [64] Regina O. Obe und Leo S. Hsu. *PostgreSQL: Up and Running*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Okt. 2017. ISBN: 978-1-4919-6336-4 (siehe S. 92).
- [65] Robert Orfali, Dan Harkey und Jeri Edwards. *Client/Server Survival Guide*. 3. Aufl. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 25. Jan. 1999. ISBN: 978-0-471-31615-2 (siehe S. 89).
- [66] Yasset Pérez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe da Veiga Leprevost, Christian Fufezan, Tobias Ternent, Stephen J. Eglén, Daniel S. Katz, Tom J. Pollard, Alexander Kononov, Robert M. Flight, Kai Blin und Juan Antonio Vizcaíno. "Ten Simple Rules for Taking Advantage of Git and GitHub". *PLOS Computational Biology* 12(7), 14. Juli 2016. San Francisco, CA, USA: Public Library of Science (PLOS). ISSN: 1553-7358. doi:10.1371/JOURNAL.PCBI.1004947 (siehe S. 90).
- [67] Dušan Petković. *Microsoft SQL Server 2019: A Beginner's Guide*. 7. Aufl. New York, NY, USA: McGraw-Hill, Jan. 2020. ISBN: 978-1-260-45888-6 (siehe S. 91).
- [68] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. 92).
- [69] Abhishek Ratan, Eric Chou, Pradeeban Kathiravelu und Dr. M.O. Faruque Sarker. *Python Network Programming*. Birmingham, England, UK: Packt Publishing Ltd, Jan. 2019. ISBN: 978-1-78883-546-6 (siehe S. 89).
- [70] Federico Razzoli. *Mastering MariaDB*. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2014. ISBN: 978-1-78398-154-0 (siehe S. 91).

# References VIII



- [71] Mike Reichardt, Michael Gundall und Hans D. Schotten. "Benchmarking the Operation Times of NoSQL and MySQL Databases for Python Clients". In: *47th Annual Conference of the IEEE Industrial Electronics Society (IECON'2021)*. 13.–15. Okt. 2021, Toronto, ON, Canada. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE), 2021, S. 1–8. ISSN: 2577-1647. ISBN: 978-1-6654-3554-3. doi:10.1109/IECON48115.2021.9589382 (siehe S. 91).
- [72] Mark Richards und Neal Ford. *Fundamentals of Software Architecture: An Engineering Approach*. Sebastopol, CA, USA: O'Reilly Media, Inc., Jan. 2020. ISBN: 978-1-4920-4345-4 (siehe S. 89).
- [73] Stuart J. Russell und Peter Norvig. *Artificial Intelligence: A Modern Approach (AIMA)*. 4. Aufl. Hoboken, NJ, USA: Pearson Education, Inc. ISBN: 978-1-292-40113-3. URL: <https://aima.cs.berkeley.edu> (besucht am 2024-06-27) (siehe S. 89).
- [74] Winfried Seimert. *LibreOffice 7.3 – Praxiswissen für Ein- und Umsteiger*. Blaufelden, Schwäbisch Hall, Baden-Württemberg, Germany: mitp Verlags GmbH & Co. KG, Apr. 2022. ISBN: 978-3-7475-0504-5 (siehe S. 20–27, 90, 91).
- [75] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Request for Comments (RFC) 4180. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Okt. 2005. URL: <https://www.ietf.org/rfc/rfc4180.txt> (besucht am 2025-02-05) (siehe S. 89).
- [76] Shai Shalev-Shwartz und Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, England, UK: Cambridge University Press (CUP), Juli 2014. ISBN: 978-1-107-05713-5. URL: <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning> (besucht am 2024-06-27) (siehe S. 91).
- [77] Ellen Siever, Stephen Figgins, Robert Love und Arnold Robbins. *Linux in a Nutshell*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2009. ISBN: 978-0-596-15448-6 (siehe S. 90).
- [78] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: 978-1-0981-3391-7 (siehe S. 89).
- [79] John Miles Smith und Philip Yen-Tang Chang. "Optimizing the Performance of a Relational Algebra Database Interface". *Communications of the ACM (CACM)* 18(10):568–579, Okt. 1975. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/361020.361025 (siehe S. 92).

# References IX



- [80] "SQL Commands". In: *PostgreSQL Documentation*. 17.4. The PostgreSQL Global Development Group (PGDG), 20. Feb. 2025. Kap. Part VI. Reference. URL: <https://www.postgresql.org/docs/17/sql-commands.html> (besucht am 2025-02-25) (siehe S. 92).
- [81] Ryan K. Stephens und Ronald R. Plew. *Sams Teach Yourself SQL in 21 Days*. 4. Aufl. Sams Tech Yourself. Indianapolis, IN, USA: SAMS Technical Publishing und Hoboken, NJ, USA: Pearson Education, Inc., Okt. 2002. ISBN: 978-0-672-32451-2 (siehe S. 87, 92).
- [82] Ryan K. Stephens, Ronald R. Plew, Bryan Morgan und Jeff Perkins. *SQL in 21 Tagen. Die Datenbank-Abfragesprache SQL vollständig erklärt (in 14/21 Tagen)*. 6. Aufl. Burghthann, Bayern, Germany: Markt+Technik Verlag GmbH, Feb. 1998. ISBN: 978-3-8272-2020-2. Translation of<sup>81</sup> (siehe S. 92).
- [83] Allen Taylor. *Introducing SQL and Relational Databases*. New York, NY, USA: Apress Media, LLC, Sep. 2018. ISBN: 978-1-4842-3841-7 (siehe S. 92).
- [84] Alkin Tezuysal und Ibrar Ahmed. *Database Design and Modeling with PostgreSQL and MySQL*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2024. ISBN: 978-1-80323-347-5 (siehe S. 91, 92).
- [85] *The JSON Data Interchange Syntax*. Standard ECMA-404, 2nd Edition. Geneva, Switzerland: Ecma International, Dez. 2017. URL: <https://ecma-international.org/publications-and-standards/standards/ecma-404> (besucht am 2024-12-15) (siehe S. 90).
- [86] Linus Torvalds. "The Linux Edge". *Communications of the ACM (CACM)* 42(4):38–39, Apr. 1999. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/299157.299165 (siehe S. 90).
- [87] Mariot Tsitoara. *Beginning Git and GitHub: Version Control, Project Management and Teamwork for the New Developer*. New York, NY, USA: Apress Media, LLC, März 2024. ISBN: 979-8-8688-0215-7 (siehe S. 89, 90, 93).
- [88] Laurie A. Ulrich und Ken Cook. *Access For Dummies*. Hoboken, NJ, USA: For Dummies (Wiley), Dez. 2021. ISBN: 978-1-119-82908-9 (siehe S. 20–27, 90, 91).
- [89] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: 978-0-13-792931-3 (siehe S. 90).

# References X



- [90] Thomas Weise (汤卫思). *Databases*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), Institute of Applied Optimization (应用优化研究所, IAO), 2025. URL: <https://thomasweise.github.io/databases> (besucht am 2025-01-05) (siehe S. 89, 90, 92).
- [91] Thomas Weise (汤卫思). *Programming with Python*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), Institute of Applied Optimization (应用优化研究所, IAO), 2024–2025. URL: <https://thomasweise.github.io/programmingWithPython> (besucht am 2025-01-05) (siehe S. 92).
- [92] *What is a Relational Database?* Armonk, NY, USA: International Business Machines Corporation (IBM), 20. Okt. 2021–12. Dez. 2024. URL: <https://www.ibm.com/think/topics/relational-databases> (besucht am 2025-01-05) (siehe S. 92).
- [93] Peter Whyte. *Microsoft SQL Server DBA Blog*. Edinburgh, Scotland, UK, 2018–2025. URL: <https://peter-whyte.com/sql-dba-blog> (besucht am 2025-06-03) (siehe S. 91).
- [94] Ulf Michael „Monty“ Widenius, David Axmark und Uppsala, Sweden: MySQL AB. *MySQL Reference Manual – Documentation from the Source*. Sebastopol, CA, USA: O'Reilly Media, Inc., 9. Juli 2002. ISBN: 978-0-596-00265-7 (siehe S. 91).
- [95] Kinza Yasar und Craig S. Mullins. *Definition: Database Management System (DBMS)*. Newton, MA, USA: TechTarget, Inc., Juni 2024. URL: <https://www.techtarget.com/searchdatamanagement/definition/database-management-system> (besucht am 2025-01-11) (siehe S. 89).
- [96] Pavlo V. Zahorodko und Pavlo V. Merzlykin. “An Approach for Processing and Document Flow Automation for Microsoft Word and LibreOffice Writer File Formats”. In: *4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW'2021)*. 18. Dez. 2021, Virtual Event and Kryvyi Rih, Ukraine. Hrsg. von Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Soloviev und Andrii M. Striuk. Bd. 3077 der Reihe CEUR Workshop Proceedings (CEUR-WS.org). Aachen, Nordrhein-Westfalen, Germany: CEUR-WS Team, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, 2022, S. 66–82. ISSN: 1613-0073. URL: <https://ceur-ws.org/Vol-3077/paper12.pdf> (besucht am 2025-10-04) (siehe S. 90, 91).
- [97] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: 978-1-78439-687-9 (siehe S. 89).

# Glossary (in English) I



AI Artificial Intelligence, see, e.g.,<sup>73</sup>

Bash is a the shell used under Ubuntu Linux, i.e., the program that „runs“ in the terminal and interprets your commands, allowing you to start and interact with other programs<sup>14,63,97</sup>. Learn more at <https://www.gnu.org/software/bash>.

client In a client-server architecture, the client is a device or process that requests a service from the server. It initiates the communication with the server, sends a request, and receives the response with the result of the request. Typical examples for clients are web browsers in the internet as well as clients for database management systems (DBMSes), such as `psql`.

client-server architecture is a system design where a central server receives requests from one or multiple clients<sup>10,57,65,69,72</sup>. These requests and responses are usually sent over network connections. A typical example for such a system is the World Wide Web (WWW), where web servers host websites and make them available to web browsers, the clients. Another typical example is the structure of database (DB) software, where a central server, the DBMS, offers access to the DB to the different clients. Here, the client can be some terminal software shipping with the DBMS, such as `psql`, or the different applications that access the DBs.

CSV *Comma-Separated Values* is a very common and simple text format for exchanging tabular or matrix data<sup>75</sup>. Each row in the text file represents one row in the table or matrix. The elements in the row are separated by a fixed delimiter, usually a comma („,"), sometimes a semicolon („;"). Python offers some out-of-the-box CSV support in the `csv` module<sup>25</sup>.

DB A *database* is an organized collection of structured information or data, typically stored electronically in a computer system. Databases are discussed in our book *Databases*<sup>90</sup>.

DBA A *database administrator* is the person or group responsible for the effective use of database technology in an organization or enterprise.

DBMS A *database management system* is the software layer located between the user or application and the DB. The DBMS allows the user/application to create, read, write, update, delete, and otherwise manipulate the data in the DB<sup>95</sup>.

Git is a distributed Version Control Systems (VCS) which allows multiple users to work on the same code while preserving the history of the code changes<sup>78,87</sup>. Learn more at <https://git-scm.com>.

# Glossary (in English) II



- GitHub** is a website where software projects can be hosted and managed via the Git VCS<sup>66,87</sup>. Learn more at <https://github.com>.
- GUI** graphical user interface
- HTTP** The Hyper Text Transfer Protocol (HTTP) is the protocol linking web browsers to web servers in the WWW<sup>8,9,37,38,41</sup>.
- IT** information technology
- JavaScript** JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers<sup>34</sup>.
- JSON** *JavaScript Object Notation* is a data interchange format<sup>15,85</sup> based on JavaScript<sup>34</sup> syntax.
- LAMP Stack** A system setup for web applications: Linux, Apache (a web server), MySQL, and the server-side scripting language PHP<sup>17,46</sup>.
- LibreOffice** is an open source office suite<sup>39,56,74</sup> which is a good and free alternative to Microsoft Office. It offers software such as LibreOffice Writer, LibreOffice Calc, and LibreOffice Base. See<sup>90</sup> for more information and installation instructions.
- LibreOffice Base** is a DBMS that can work on stand-alone files but also connect to other popular relational databases<sup>35,74</sup>. It is part of LibreOffice<sup>39,56,74</sup> and has functionality that is comparable to Microsoft Access<sup>6,20,88</sup>.
- LibreOffice Calc** is a spreadsheet software that allows you to arrange and perform calculations with data in a tabular grid. It is a free and open source spreadsheet software<sup>56,74</sup>, i.e., an alternative to Microsoft Excel. It is part of LibreOffice<sup>39,56,74</sup>.
- LibreOffice Writer** is a free and open source text writing program<sup>96</sup> and part of LibreOffice<sup>39,56,74</sup>. It is a good alternative to Microsoft Word.
- Linux** is the leading open source operating system, i.e., a free alternative for Microsoft Windows<sup>4,44,77,86,89</sup>. We recommend using it for this course, for software development, and for research. Learn more at <https://www.linux.org>. Its variant Ubuntu is particularly easy to use and install.

# Glossary (in English) III



**localhost** is the hostname of the current computer<sup>19,33</sup>. It is equivalent to the IP address `127.0.0.1`. Any message or package sent to localhost will be sent to the current computer itself.

**MariaDB** An open source relational database management system that has forked off from MySQL<sup>2,3,5,32,59,70</sup>. See <https://mariadb.org> for more information.

**Microsoft Access** is a DBMS that can work on DBs stored in single, stand-alone files but also connect to other popular relational databases<sup>6,20,60,88</sup>. It is part of Microsoft Office. A free and open source alternative to this commercial software is LibreOffice Base.

**Microsoft Excel** is a spreadsheet program that allows users to store, organize, manipulate, and calculate data in tabular structures<sup>11,40,54</sup>. It is part of Microsoft Office. A free alternative to this commercial software is LibreOffice Calc<sup>56,74</sup>.

**Microsoft Office** is a commercial suite of office software, including Microsoft Excel, Microsoft Word, and Microsoft Access<sup>54</sup>. LibreOffice is a free and open source alternative.

**Microsoft SQL Server** The Microsoft SQL Server is a successful commercial relational/Structured Query Language (SQL)-based DBMS<sup>1,67,93</sup>. Learn more at <https://www.microsoft.com/sql-server> and <https://learn.microsoft.com/en-us/sql>.

**Microsoft Windows** is a commercial proprietary operating system<sup>13</sup>. It is widely spread, but we recommend using a Linux variant such as Ubuntu for software development and for our course. Learn more at <https://www.microsoft.com/windows>.

**Microsoft Word** is one of the leading text writing programs<sup>30,62,96</sup> and part of Microsoft Office. A free alternative to this commercial software is the LibreOffice Writer.

**ML** Machine Learning, see, e.g.,<sup>76</sup>

**MySQL** An open source relational database management system<sup>12,32,71,84,94</sup>. MySQL is famous for its use in the LAMP Stack. See <https://www.mysql.com> for more information.

**Oracle Database** The Oracle Database was the first commercial SQL-based relational database<sup>18</sup>. It is still a highly successful proprietary product with many features<sup>7,51</sup>. Learn more at <https://www.oracle.com/database>.

# Glossary (in English) IV



**OSS** Open source software, i.e., software that can freely be used, whose source code is made available in the internet, and which is usually developed cooperatively over the internet as well<sup>47</sup>. Typical examples are Python, Linux, Git, and PostgreSQL.

**port** A port in networking is a software-defined number associated to a network protocol that receives or transmits communication for a specific service<sup>52</sup>. In the client-server architecture, the server listens for incoming communication connections at a specific port. The clients connect to the network address and port number of the server to establish such connections. port numbers range from 0 to 65535, where the port numbers from 0 to 1023 are so-called well-known ports corresponding to the most common services, e.g., Hyper Text Transfer Protocol (HTTP), the protocol underpinning of the WWW, uses normally port 80.

**PostgreSQL** An open source object-relational DBMS<sup>36,64,68,84</sup>. See <https://postgresql.org> for more information.  
**psql** is the client program used to access the PostgreSQL DBMS server.

**Python** The Python programming language<sup>48,55,58,91</sup>, i.e., what you will learn about in our book<sup>91</sup>. Learn more at <https://python.org>.

**relational database** A relational DB is a database that organizes data into rows (tuples, records) and columns (attributes), which collectively form tables (relations) where the data points are related to each other<sup>22,42,43,79,83,90,92</sup>.

**server** In a client-server architecture, the server is a process that fulfills the requests of the clients. It usually waits for incoming communication carrying the requests from the clients. For each request, it takes the necessary actions, performs the required computations, and then sends a response with the result of the request. Typical examples for servers are web servers<sup>17</sup> in the internet as well as DBMSes. It is also common to refer to the computer running the server processes as server as well, i.e., to call it the „server computer“<sup>53</sup>.

**SQL** The *Structured Query Language* is basically a programming language for querying and manipulating relational databases<sup>18,26–28,49,61,80–83</sup>. It is understood by many DBMSes. You find the SQL commands supported by PostgreSQL in the reference<sup>80</sup>.

# Glossary (in English) V



- terminal** A terminal is a text-based window where you can enter commands and execute them<sup>4,21</sup>. Knowing what a terminal is and how to use it is very essential in any programming- or system administration-related task. If you want to open a terminal under Microsoft Windows, you can **Druck auf**  + **R**, **dann Schreiben von** `cmd`, **dann Druck auf** . Under Ubuntu Linux, **Ctrl** + **Alt** + **T** opens a terminal, which then runs a Bash shell inside.
- Ubuntu** is a variant of the open source operating system Linux<sup>21,46</sup>. We recommend that you use this operating system to follow this class, for software development, and for research. Learn more at <https://ubuntu.com>. If you are in China, you can download it from <https://mirrors.ustc.edu.cn/ubuntu-releases>.
- VCS** A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code<sup>87</sup>. A distributed VCS allows multiple users to work on the same code and upload their changes to the server, which then preserves the change history. The most popular distributed VCS is Git.
- WWW** World Wide Web<sup>8,29</sup>
- XML** The *Extensible Markup Language* is a text-based language for storing and transporting of data<sup>16,24,50</sup>. It allows you to define elements in the form `<myElement myAttr="x">...text...</myElement>`. Different from comma-separated values (CSV), elements in XML can be hierarchically nested, like `<a<b><c>test</c></b><b>bla</b></a>`, and thus easily represent tree structures. XML is one of most-used data interchange formats. To process XML in Python, use the `defusedxml` library<sup>45</sup>, as it protects against several security issues.
- YAML** *YAML Ain't Markup Language™* is a human-friendly data serialization language for all programming languages<sup>23,31,50</sup>. It is widely used for configuration files in the DevOps environment. See <https://yaml.org> for more information.