

Tag 2

Inhaltsverzeichnis

- Wieso eine Datenbank?
- Neue DB-Anwendungsfelder
- Das relationale Datenmodell
- Umsetzung ER-Modell ins relationale Datenmodell
- SQL erste Schritte
- Übungen
- BYOQ

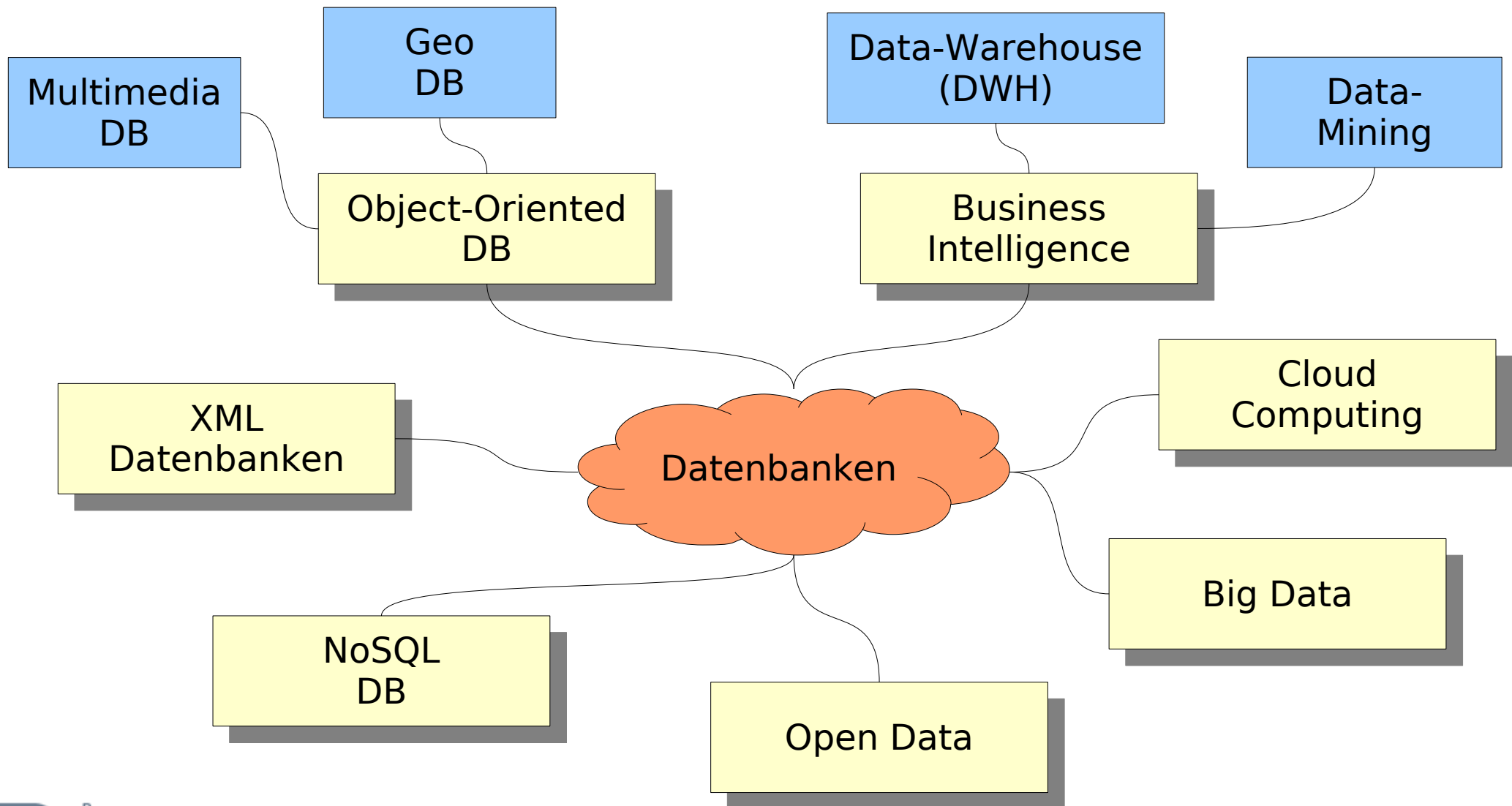
Die Datenbank-Thematik

Wieso eine Datenbank?

- Konsistenz (Redundanz...)
- Integrität (Datenverlust...)
- Mehrbenutzer-Betrieb
- Sicherheit (unkontrollierte Zugriffe...)

Die Datenbank-Thematik

Neue Anwendungsfelder

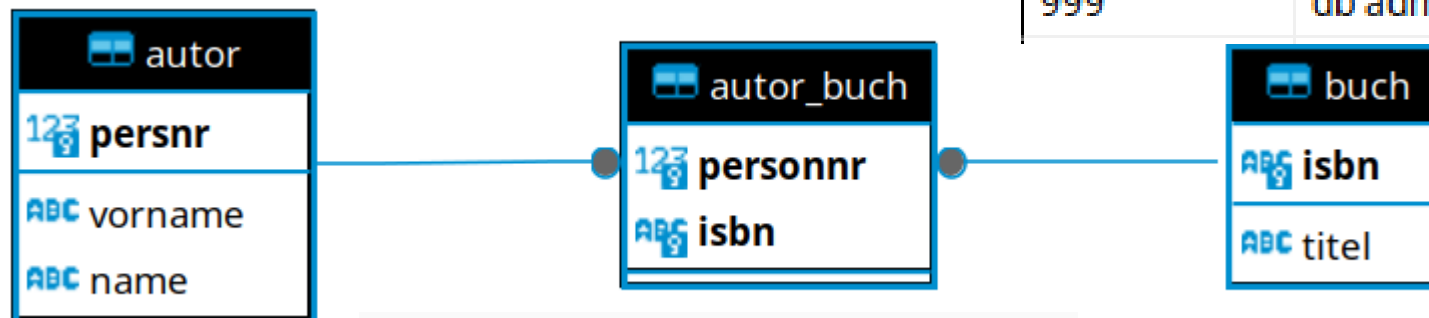


Das relationale Datenmodell

Beispiel

123 persnr	ABC vorname	ABC name
12	alfons	kemper
34	michael	kofler
56	gilles	maitre
78	larry	wall

ABC isbn	ABC titel
111	visualbasic 2008
123	datenbanksysteme
222	mathematica
456	mysql 5
789	linux
999	db administration



123 personnr	ABC isbn
34 ↗	↗ 111
12 ↗	↗ 123
34 ↗	↗ 222
34 ↗	↗ 456
34 ↗	↗ 789

Das relationale Datenmodell

Pro und Kontra

Pro

- Mathematisch basiert
- Weit verbreitet
- Lange Erfahrung
- Guter programmatischer Support

Kontra

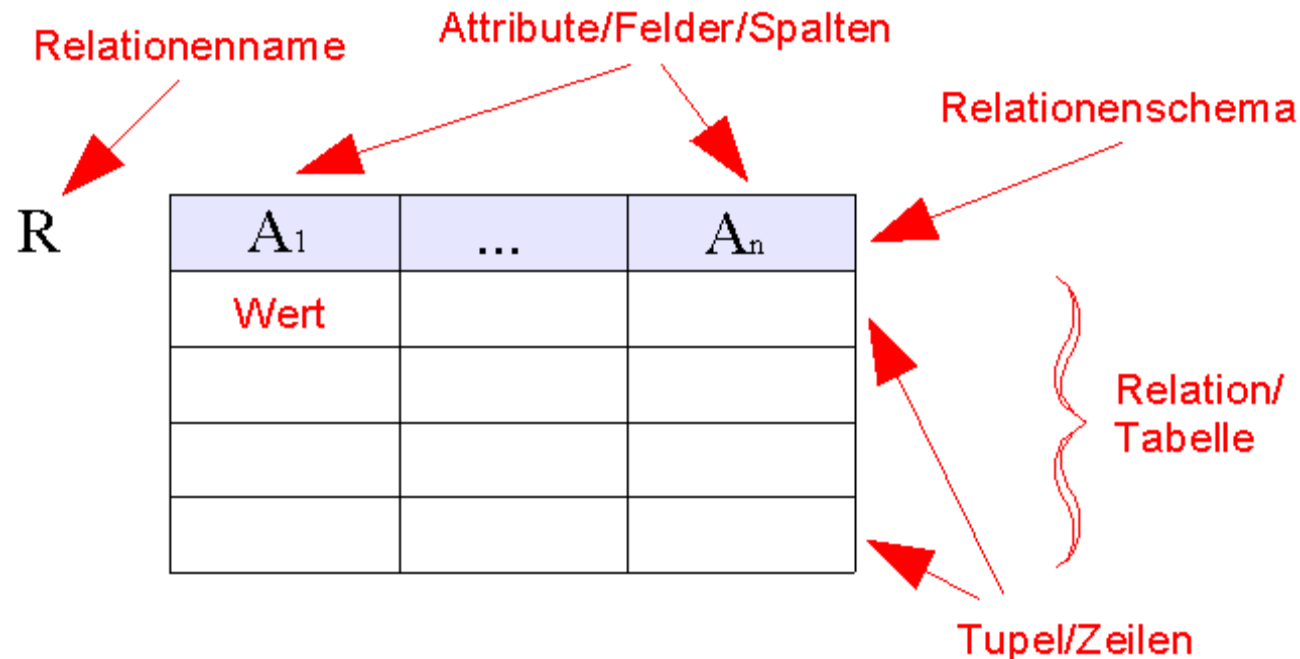
- Künstliche Datensegmentierung

Produkte (*Achtung: alle sind verschieden*):
Oracle, MySQL, PostgreSQL, Ingres, MS SQL Server,
Sybase, DB2, (MS Access?), ...

Applikationen: zahlreiche...

Das relationale Datenmodell

Relation / Tabelle



- Relationale Datenbank: Sammlung von **Tabellen**
- Jede Tabelle: Sammlung von **Zeilen und Spalten**
- Jede Zeile: Menge von gleichen unstrukturierten **Spalten**
- Freie Ordnung von Zeilen und Attributen
- Abfragen, um Daten zu bearbeiten

Das relationale Datenmodell

Primär- und Fremdschlüssel

Primärschlüssel



Personal

PersNr	Vorname	Name	...
...	...	...	...
234	Frieda	Müller	...
993	Horst	Winter	...
670	Hans	Sauer	...
...	...	...	...

IstChefVon

Vorgesetzter	Untergebener
...	...
234	993
234	670
...	...



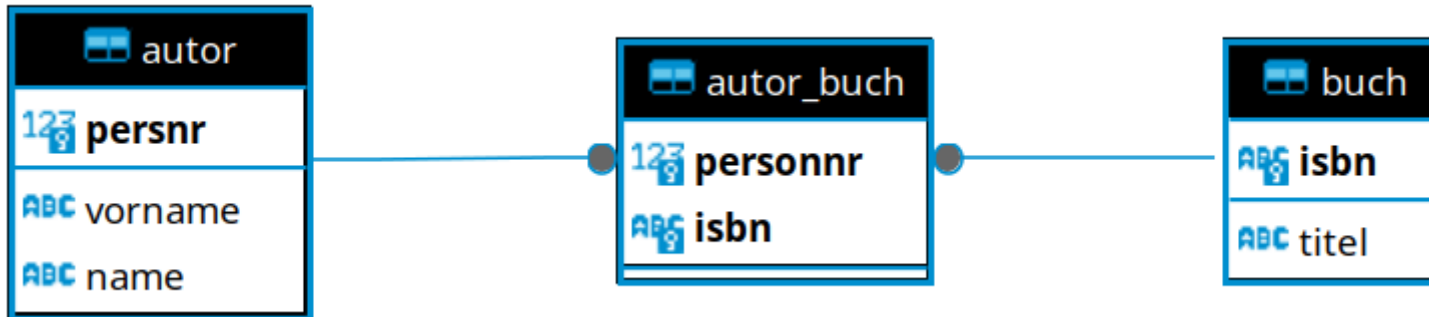
Fremdschlüssel

Primärschlüssel von IstChefVon

- *Primärschlüssel*: Minimale Menge von Attributen, die einen Datensatz in einer Tabelle **eindeutig identifiziert**.
(Primary-Key, PK)
- *Fremdschlüssel*: Attribut einer Relation, welches den Primärschlüssel einer (anderen) Relation **referenziert**.
(Foreign-Key, FK)

Das relationale Datenmodell

Schlüssel- und Schema-Definition



Wo sind welche Schlüssel? PK **rot**, FK **blau**

autor : { persnr: int, vorname: String, name: String }

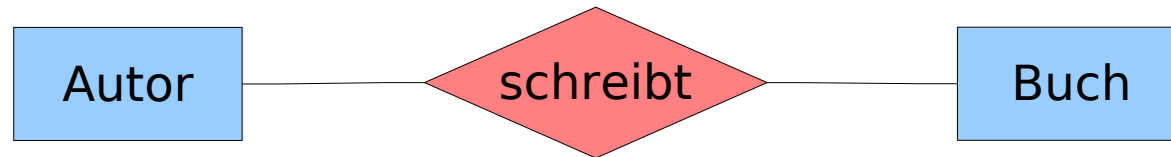
autor_buch : { personnr: int, isbn: String }

buch : { isbn: String, titel: String }

personnr + isbn bilden zusammen einen PK

Umsetzung ER-Modell in rel. Datenmodell

Entitäten



autor : { persnr: int, vorname: String, name: String }

<u>persnr</u>	vorname	name
12	alfons	kemper
34	michael	kofler
56	gilles	maitre
78	larry	wall

Meine Notation:
Tabelle: {Feld: Typ, ...}

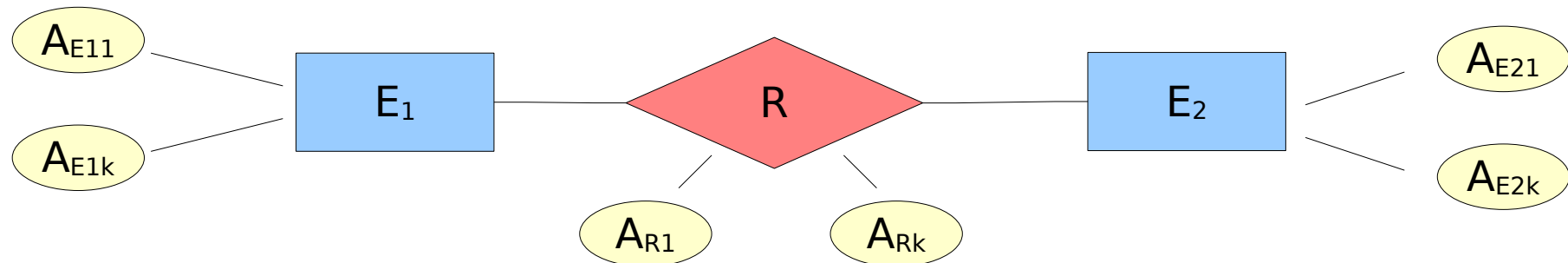
buch : { isbn: String, titel: String }

<u>isbn</u>	titel
111	visualbasic 2008
123	datenbanksysteme
222	mathematica
456	mysql 5
789	linux
999	db administration

Meine Notation:
PK wird unterstrichen.

Umsetzung ER-Modell in rel. Datenmodell

Beziehungen (1)



$R : \{ A_{E11}, A_{E21}, A_{R1} .. A_{Rk} \}$

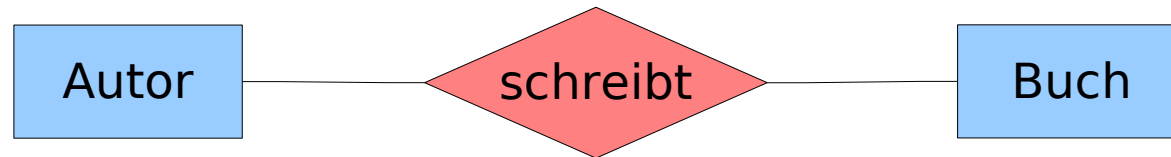
E1 Primärschlüssel → A_{E11}

E2 Primärschlüssel → A_{E21}

Eigene Attribute von R → $A_{R1} .. A_{Rk}$

Umsetzung ER-Modell in rel. Datenmodell

Beziehungen (2)



`schreibt : { personnr: int, isbn: String }`

  personnr	  isbn
34 	 111
12 	 123
34 	 222
34 	 456
34 	 789

oder eher

`autor_buch : { personnr: int, isbn: String }`

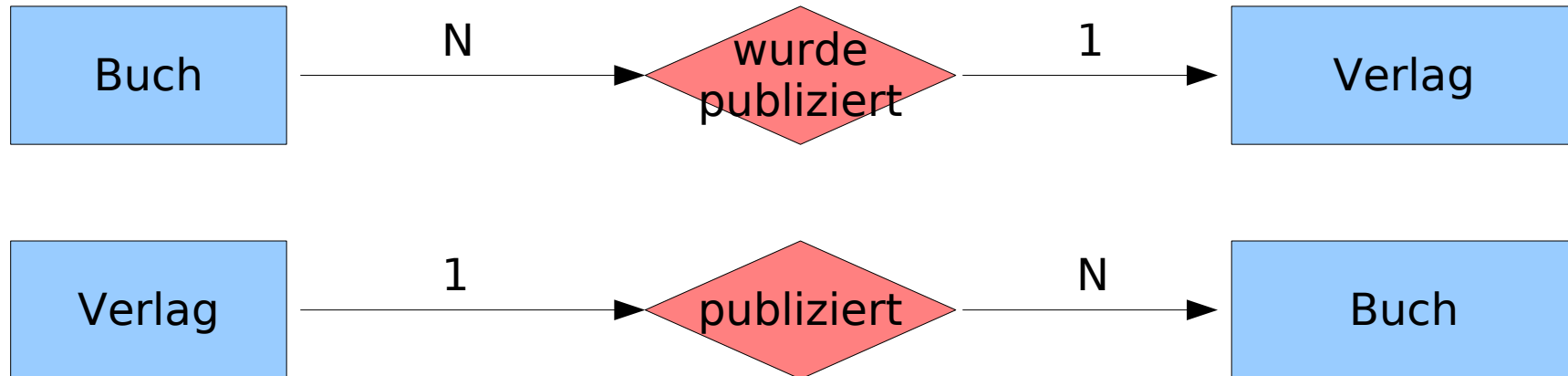
Umsetzung ER-Modell in rel. Datenmodell

Verfeinerung (1)

- Ziel: **Zusammenfassung** von Relationen
- Möglich nur für 1:1, 1:n oder n:1 Beziehungen
- **Geht nicht für n:m** Beziehungen

Umsetzung ER-Modell in rel. Datenmodell

Verfeinerung (2)



Nicht ideale Lösung (wäre korrekt für "n:m" Bez.)

buch : { isbn: String, titel: String }
verlag : { verlagid: Integer, kategorie: String }
buch_verlag : { isbn: String, **verlagid**: Integer }

Vereinfachte Lösung

buch : { isbn: String, titel: String; verlagid: Integer }
verlag : { **verlagid**: Integer, kategorie: String }

Tipp aus der Praxis

Umsetzung rel. Datenmodell ins ER-Modell

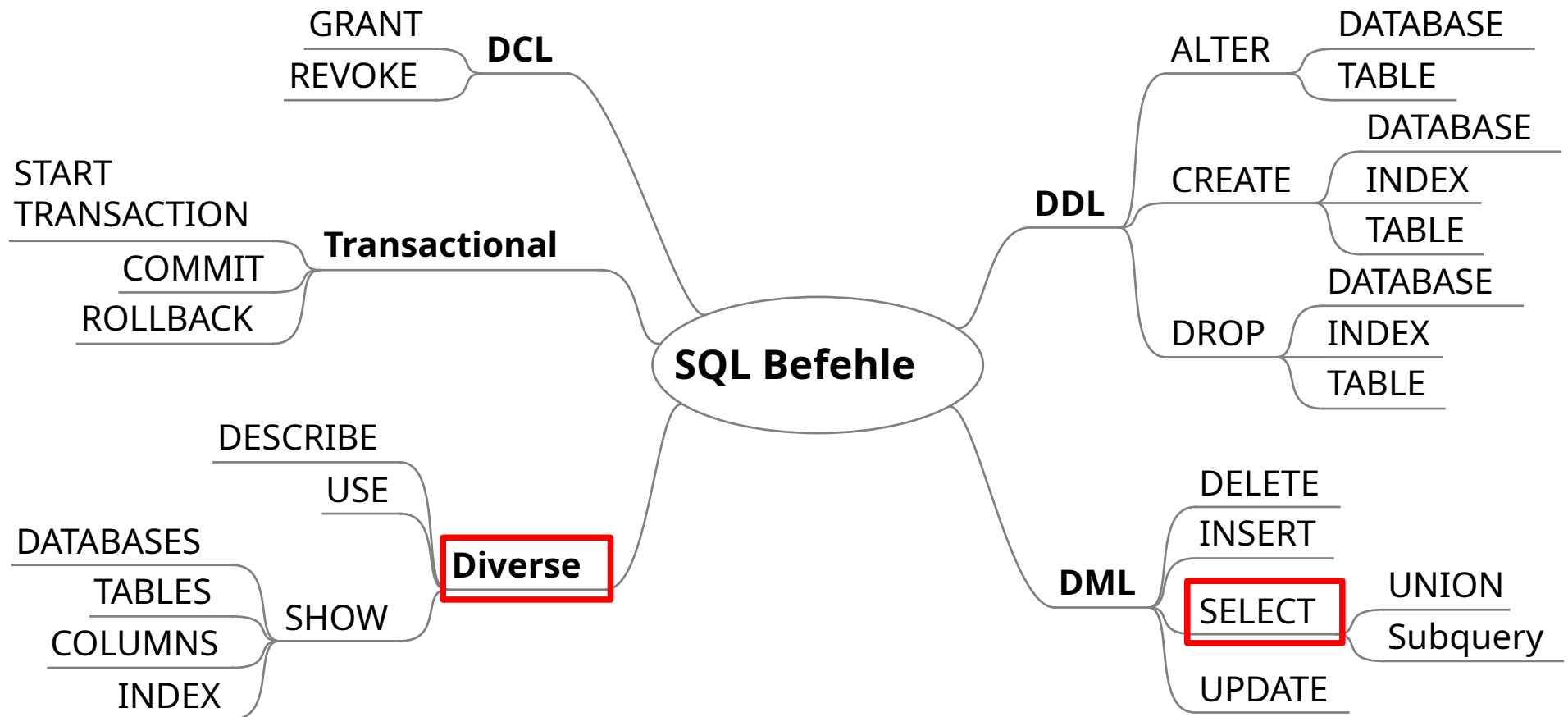
- Ziel: Ein komplexes DB-Schema verstehen
- Tipp: Das ER-Modell davon *selber von Hand* zeichnen
- Tools
 - Blatt Papier (A4-A...)
 - Bleistift
 - Radiergummi
 - Geduld ;-)
- Abstrahieren (*nicht alle Details zeichnen*)

SQL

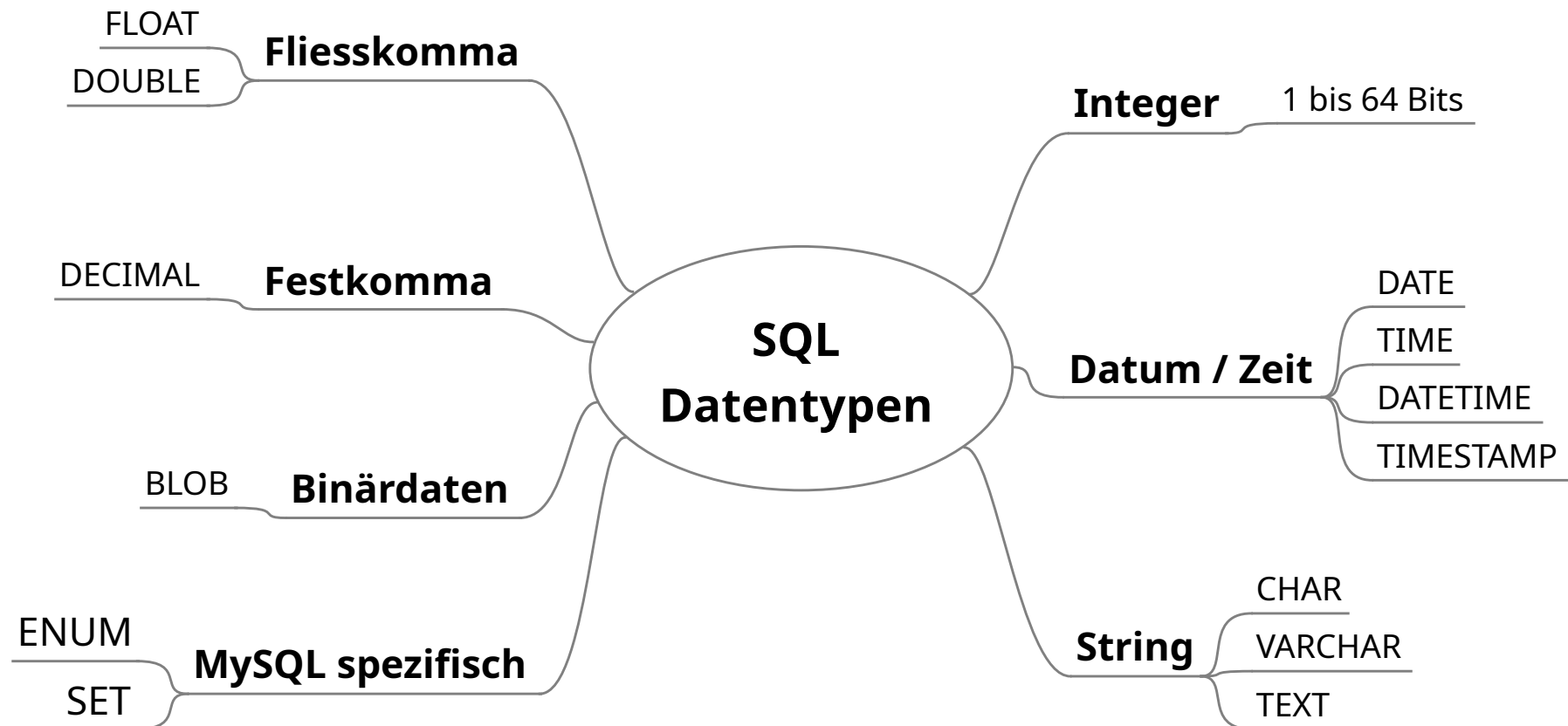
Wieso hier? Gründe dafür

- Um Daten zu modellieren und auf Daten zugreifen können.
- Viele Daten sind heute in relationalen DB => SQL ist **DIE** standard Sprache
- **SQL Datenzugriffsmodell** auch in anderen Gebieten präsent (z.B. Pandas DataFrame, für Join, Group, etc...)
- Hier beschränken wir uns auf den Datenzugriff, lesend
=> ***SQL Survival Kit***

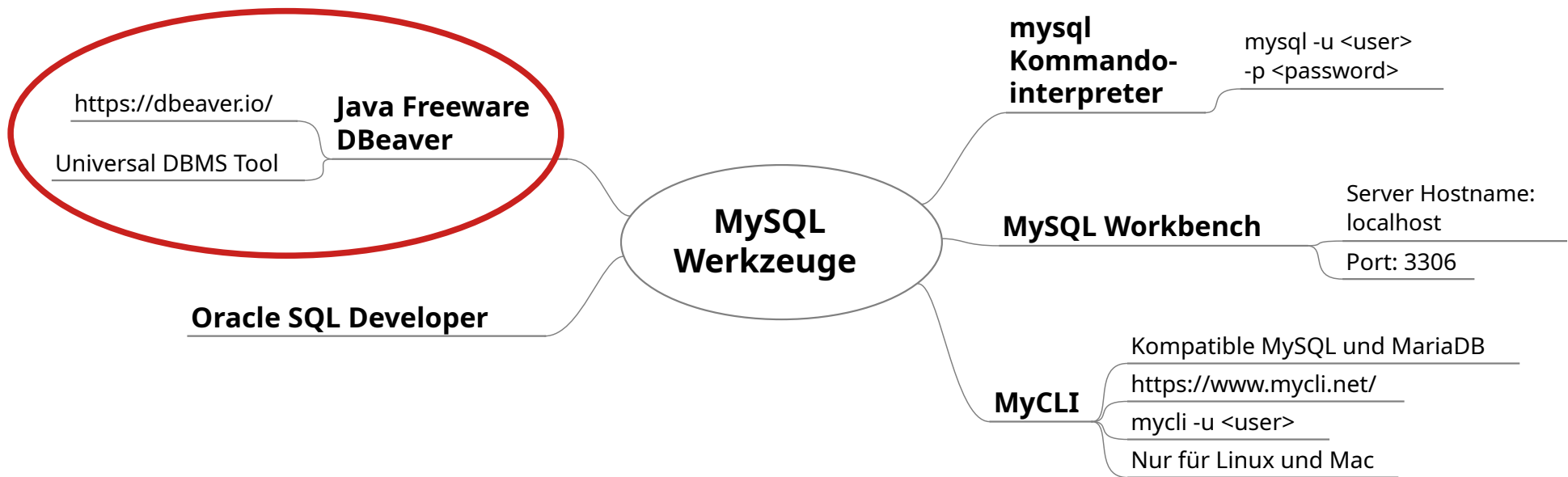
SQL, Übersicht aller Befehle



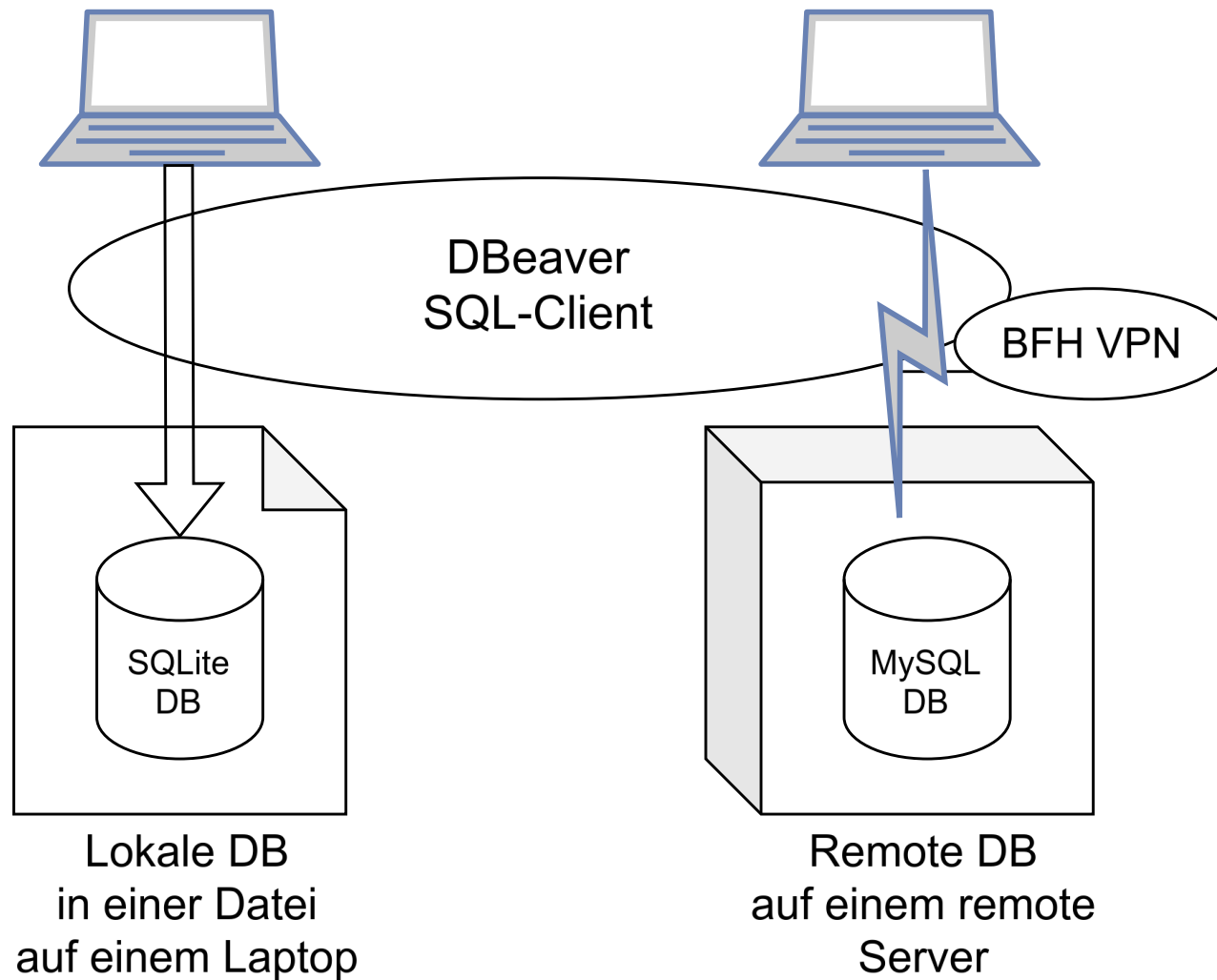
SQL, Datentypen



(My)SQL Admin Client-Programme



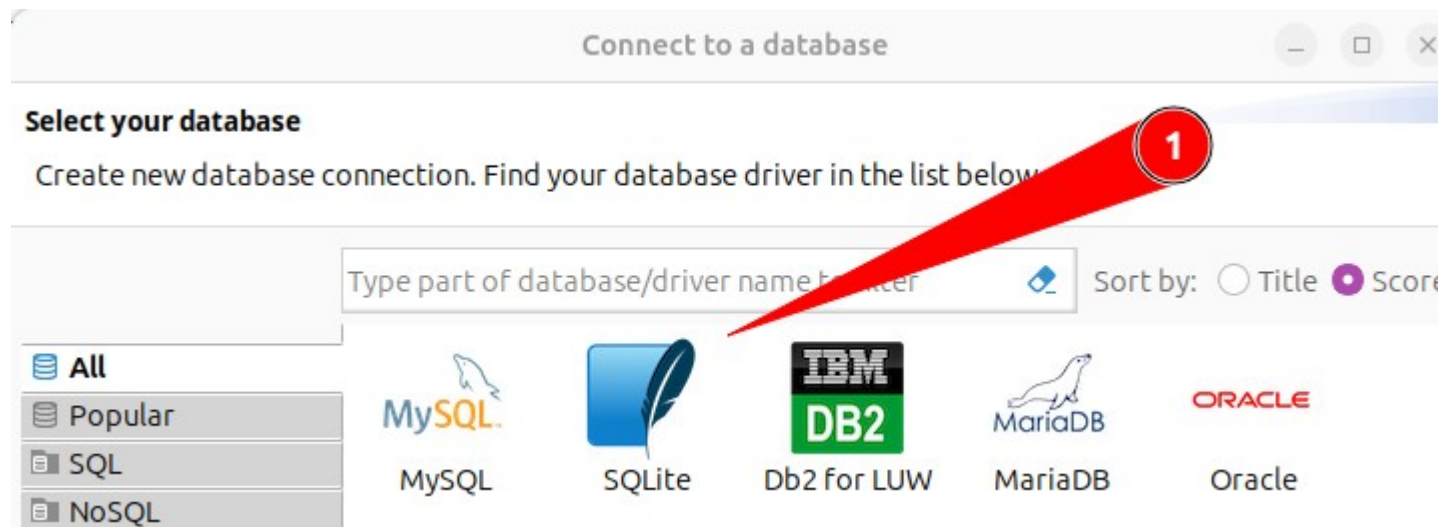
DB-Zugriff mit DBeaver Lokal und Remote



Zugriff auf lokale SQLite-Datenbank

DBeaver-Verbindung erstellen (1)

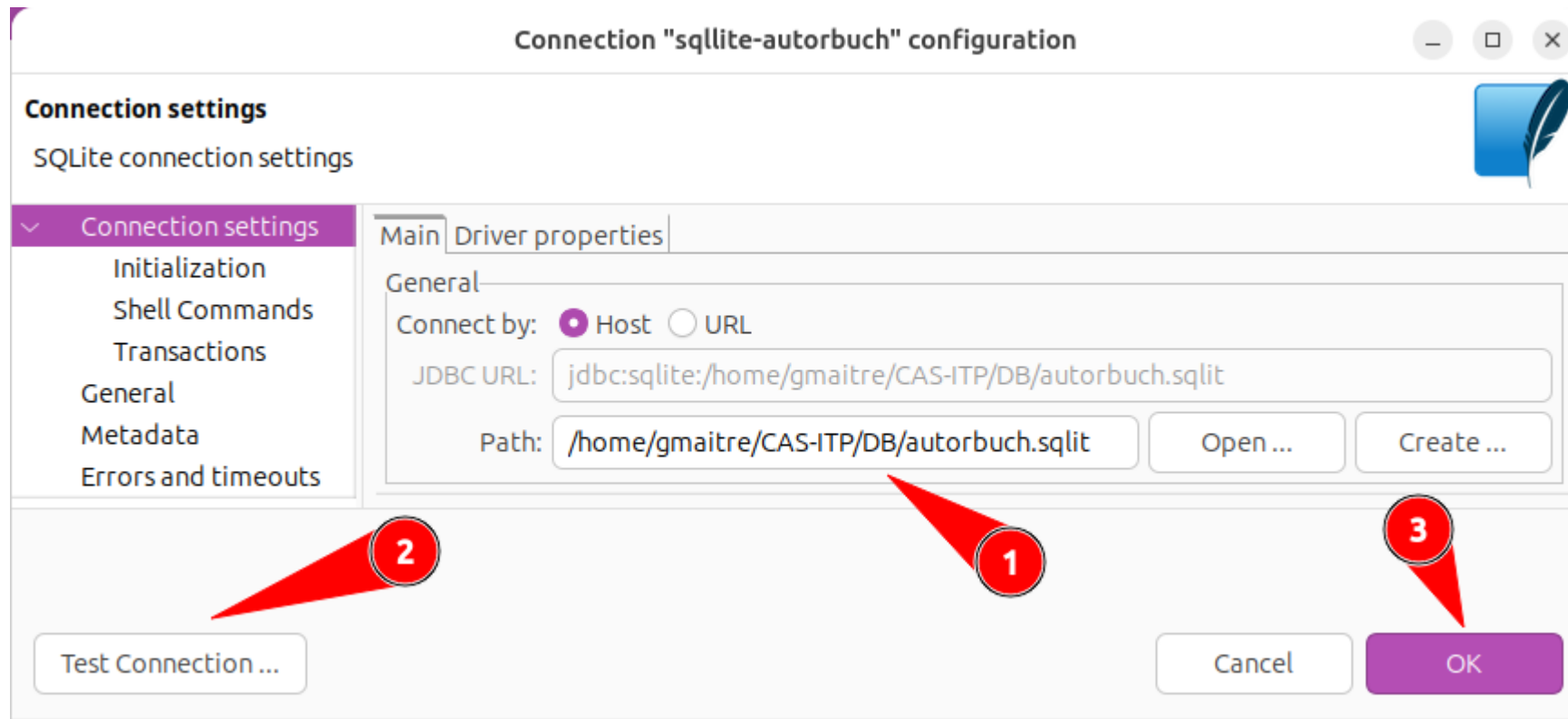
- Was ist SQLite? [Siehe hier](#).
- Aus Moodle die DB *autorbuch.sqlite* Datei ins Ihren ITP-DMOD Unterverzeichnis herunterladen.
- Neue Verbindung mit *DBeaver / Database / New database connection* erstellen



Zugriff auf lokale SQLite-Datenbank

DBeaver-Verbindung erstellen (2)

- Zuerst Pfad zur lokalen *autorbuch.sqlit* DB-Datei angeben, die vorher heruntergeladen wurde.



- Dann, unter "General" ein Verbindungsname angeben, z.B. *sqlite-autorbuch*

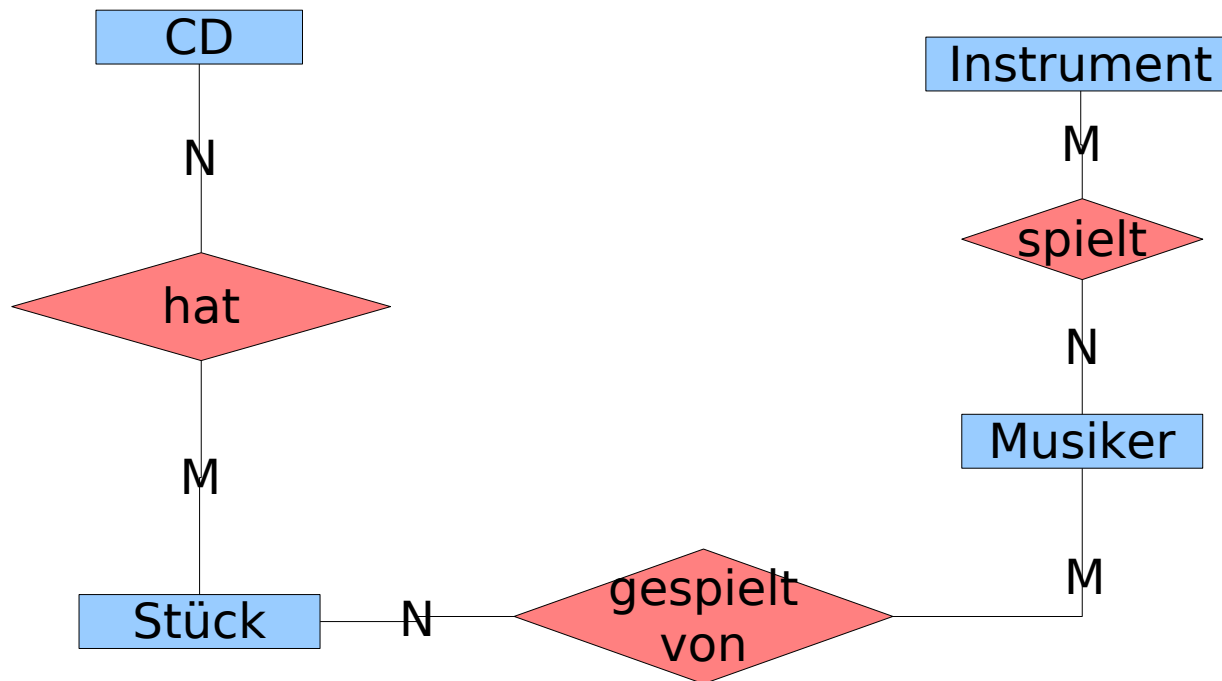
Zugriff auf lokale SQLite-Datenbank DBeaver-Verbindung erstellen (3)

- Dann mit einem rechten Mausklick auf die *sqlite-autorbuch* Verbindung, *SQL-Editor / New SQL-Script* anklicken (Punkt 1).
- Und den Befehl *select * from autor* angeben (Punkt 2).

The screenshot shows the DBeaver 25.0.1 interface. The Database Navigator on the left shows the 'sqlite-autorbuch' connection selected. The SQL Editor in the center contains the query `select * from autor;`. Below the editor, the results are displayed in a table grid.

	persnr	vorname	name
1	12	alfons	kemper
2	34	michael	kofer
3	56	gilles	maitre
4	78	larry	wall

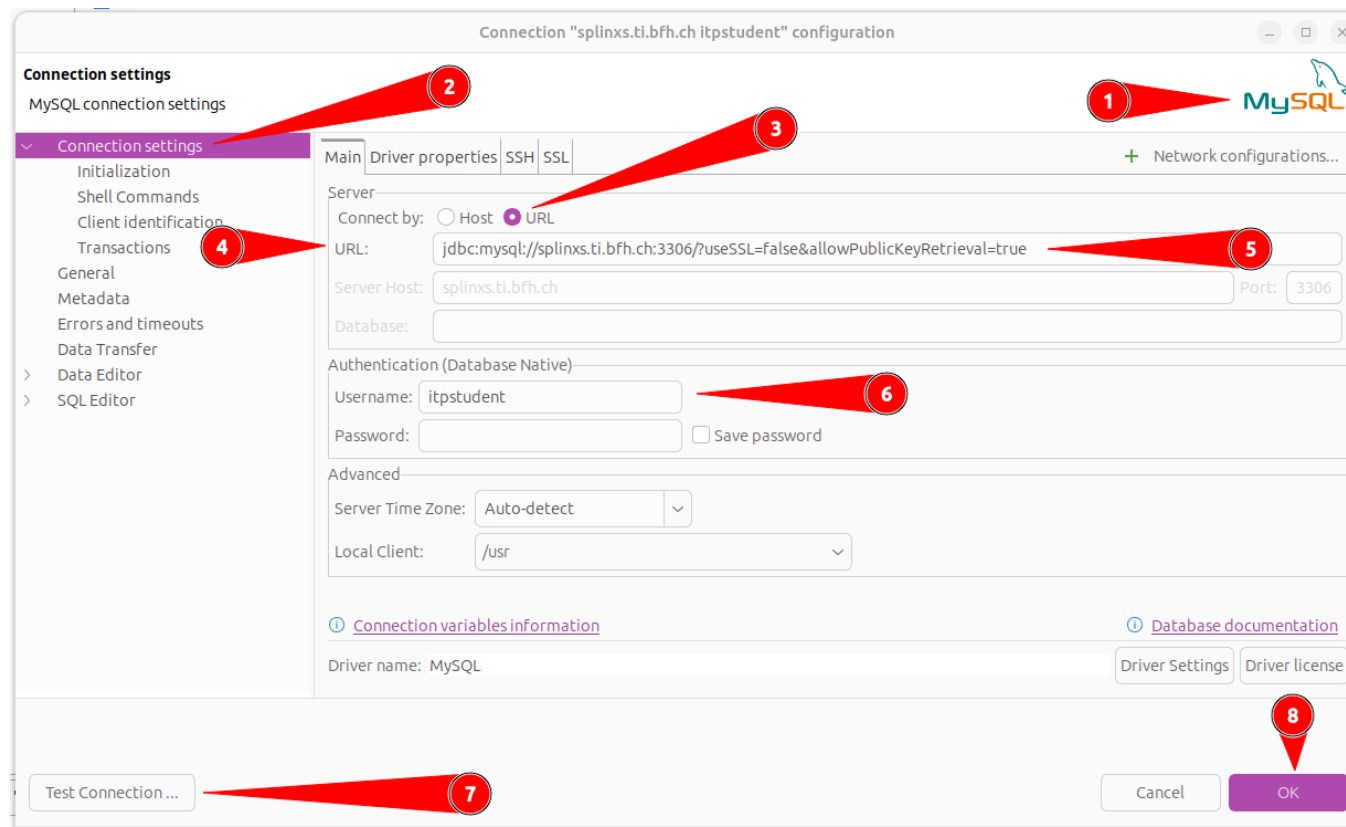
Übungen (basierend auf CD Sammlung)



- 1) Erstellen Sie mit DBeaver eine lokale Verbindung auf die AutorBuch SQLite-Datenbank (siehe die vorigen Slides).
- 2) Das CD-Sammlung ER-Modell hier oben in ein relationale Datenmodell umwandeln (auf Papier oder im Notepad-Editor).
- 3) Eine remote Verbindung auf meine MySQL AutorBuch Datenbank erstellen (siehe nächste Slides).

Zugriff auf remote meine MySQL-Datenbank DBeaver-Verbindung erstellen

- Bedingung: Zugriff auf das BFH-Netzwerk
 - VPN verwenden. Beschreibung → [hier im BFH-Intranet](#).
- Verbindung mit DBeaver / Database / New database connection



URL: `jdbc:mysql://splinxs.ti.bfh.ch:3306/?useSSL=false&allowPublicKeyRetrieval=true`

Das Passwort wird während dem Unterricht mitgeteilt

DMOD 2 - 24
Version 2.6

Zugriff auf meine remote MySQL-Datenbank

Daten abfragen

Die SQL-Befehle einzeln eingeben und mit <shift>-<enter> ausführen.

The screenshot shows the DBeaver 25.0.1 interface. The main window is titled "DBeaver 25.0.1 - <splinxs.ti.bfh.ch itpstudent> Script-1". The left sidebar shows a tree view of the database structure, with "splinxs.ti.bfh.ch itpstudent" selected. The right pane shows a SQL script with the following commands:

```
show databases;
use autorbuch;
show tables;
select * from autorbuch.autor;
```

Below the script, a table titled "autor 1" displays the results of the query. The table has four columns: "persnr", "vorname", "name", and an empty column. The data is as follows:

	persnr	vorname	name	
1	12	alfons	kemper	
2	34	michael	kofler	
3	56	gilles	maitre	
4	78	larry	wall	

The bottom status bar indicates "4 row(s) fetched - 0.009s, on 2025-04-02 at 07:53:57".