



# Informationssysteme - Teil 1

Sommersemester 2025

Christian Rybovic  
L-TIP-24-Do-a

«Ein Informationssystem ist eine Kombination aus Hardware, Software, Daten und Prozessen, die der Unterstützung von Informationsverarbeitung und Entscheidungsfindung dient.»



## **Inhalt**

Kompetenznachweis 1 vom 01.05.25.....	1
Kompetenznachweis 2 vom 08.05.25.....	5
Kompetenznachweis 3 vom 15.05.25.....	7
Kompetenznachweis 4 vom 22.05.25.....	8
Kompetenznachweis 5 vom 05.06.25.....	10
Kompetenznachweis 6 vom 12.06.25.....	12
Kompetenznachweis 7 vom 26.06.25.....	14

# Kompetenznachweis 1 vom 01.05.25

Um aus Daten eine Struktur zu erstellen, müssen wir verschiedene Schritte durchlaufen. Als Beispiel haben wir folgende Daten aus dem Script "Informationssysteme 1":

Daten					
name	anrede	strasse	plzundort	telefon	leseausweisnr
Fritz Müller	Herr	Hauptstr. 12	72070 Tübingen	07071-555-12312	1289234877
Susi Simmer	Frau	Herbstallee 1	72074 Tübingen	07071-555-654654	
Susi Sommer	Frau		72074 Tübingen	07071-555-64444	984613325

In einem ersten Schritt werden die Daten atomisiert:

Daten							
vorname	nachname	anrede	strasse	plz	ort	telefon	leseausweisnr
Fritz	Müller	Herr	Hauptstr. 12	72070	Tübingen	07071-555-12312	1289234877
Susi	Simmer	Frau	Herbstallee 1	72074	Tübingen	07071-555-654654	
Susi	Sommer	Frau		72074	Tübingen	07071-555-64444	984613325

Die Tabelle wird für den eindeutigen Zugriff erweitert:

Daten								
id	vorname	nachname	anrede	strasse	plz	ort	telefon	leseausweisnr
1	Fritz	Müller	Herr	Hauptstr. 12	72070	Tübingen	07071-555-12312	1289234877
2	Susi	Simmer	Frau	Herbstallee 1	72074	Tübingen	07071-555-654654	
3	Susi	Sommer	Frau		72074	Tübingen	07071-555-64444	984613325

Datenredundanzen durch Aufteilung auf mehrere Tabellen vermeiden:

Adresse							
id	vorname	nachname	akuerzel	strasse	plz	telefon	leseausweisnr
1	Fritz	Müller	m	Hauptstr. 12	72070	07071-555-12312	1289234877
2	Susi	Simmer	w	Herbstallee 1	72074	07071-555-654654	
3	Susi	Sommer	w		72074	07071-555-64444	984613325

Plz		
plz	ort	
72070	Tübingen	
72074	Tübingen	

Anrede	
akuerzel	anrede
m	Hauptstr. 12
w	Herbstallee 1

Im letzten Schritt müssen alle logisch zusammenpassenden Daten weiter ausgelagert werden:

Adresse						
<b>id</b>	<b>vorname</b>	<b>nachname</b>	<b>akuerzel</b>	<b>strasse</b>	<b>plz</b>	<b>telefon</b>
1	Fritz	Müller	m	Hauptstr. 12	72070	07071-555-12312
2	Susi	Simmer	w	Herbstallee 1	72074	07071-555-654654
3	Susi	Sommer	w		72074	07071-555-64444

Plz		Anrede		Ausweis	
<b>plz</b>	<b>ort</b>	<b>akuerzel</b>	<b>anrede</b>	<b>leseausweisnr</b>	<b>a-id</b>
72070	Tübingen	m	Hauptstr. 12	1289234877	1
72074	Tübingen	w	Herbstallee 1	984613325	3

Dieser Vorgang nennt man normalisieren und wird im nächsten Kapitel nochmal detaillierter mit anderen Beispieldaten erklärt.

# Normalisierung

Angenommen wir haben folgende Daten:

Data							
UID	Company	Address		Products			
314	Lumitrax	Hauptstraße 45, 3011 Bern		1 - Silvaron Flex (Wood) [12.90], 2 - LuminaGlass (Glass) [10.50]			
776	Novara	Lindenstrasse 27, 4051 Basel		1 - Thermora Core (Glass) [45.60], 2 - Viroflex (Wood) [12.90], 3 - QuantumMesh (Steel) [22.00]			
479	Atheronix	Marktgasse 5, 3011 Bern		1 - PyroSteel (Steel) [9.90]			

Weil die Verarbeitung und Anpassung der Daten in dieser Form sehr aufwendig ist und komplex sein kann, müssen diese zuerst normalisiert werden.

## 1. Normalform

Die 1. Normalform ist dann gegeben, wenn alle Informationen in einer Tabelle **atomar** vorliegen. Das bedeutet, dass in einer Zelle nicht mehrere (unabhängige) Informationen vorhanden sein dürfen. Diese müssen auf mehrere Spalten und Zeilen getrennt werden.

In unserem Fall werden pro Produkt eine separate Zeile angelegt, wobei alle Produktinformationen in einer separaten Spalte abgefüllt werden. Auch die Adresse wird aufgeteilt in Strasse und Stadt. Die Daten in der Tabelle sehen dann folgendermassen aus:

Data							
UID	Company	Street	City	PID	Product	Price	Material
314	Lumitrax	Hauptstraße 45	3011 Bern	1	Silvaron Flex	12.90	Wood
314	Lumitrax	Hauptstraße 45	3011 Bern	2	LuminaGlass	10.50	Glass
776	Novara	Lindenstrasse 27	4051 Basel	1	Thermora Core	45.60	Glass
776	Novara	Lindenstrasse 27	4051 Basel	2	Viroflex	12.90	Wood
776	Novara	Lindenstrasse 27	4051 Basel	3	QuantumMesh	22.00	Steel
479	Atheronix	Marktgasse 5	3011 Bern	1	PyroSteel	9.90	Steel

## 2. Normalform

Wenn nun beispielsweise die Adresse einer Firma angepasst werden soll, dann muss diese in mehreren Datensätzen angepasst werden (da mehrfach vorhanden). Hierbei können die Daten schnell inkonsistent werden, was anschliessend nur mit einem enormen Mehraufwand bereinigt werden kann. Aus diesem Grund müssen die Daten auf mehrere Tabellen aufgeteilt und über die Primärschlüssel miteinander verknüpft werden.

Die Daten werden somit neu in zwei eigenständige Tabellen voneinander getrennt, die Doppelten Firmeneinträge auf jeweils einen Eintrag reduziert und Informationen über die Primärschlüssel miteinander verknüpft:

Company				Products				
UID	Company	Street	City	UID	PID	Product	Price	Material
314	Lumitrax	Hauptstraße 45	3011 Bern	314	1	Silvaron Flex	12.90	Wood
776	Novara	Lindenstrasse 27	4051 Basel	314	2	LuminaGlass	10.50	Glass
479	Atheronix	Marktgasse 5	3011 Bern	776	1	Thermora Core	45.60	Glass
				776	2	Viroflex	12.90	Wood
				776	3	QuantumMesh	22.00	Steel
				479	1	PyroSteel	9.90	Steel

## 3. Normalform

Für die dritte Normalform müssen diese Daten nun weiter aufgeteilt werden. Es darf kein Nichtschlüsselattribut von einem anderen Nichtschlüsselattribut abhängig sein.

Beispielsweise können mehrere Unternehmen in einer Stadt gemeldet sein, weshalb die Stadt und Postleitzahl ausgelagert werden sollten, genauso wie das Material, aus welchem die Produkte bestehen. Die Daten sind nun in folgenden Tabellen abgebildet:

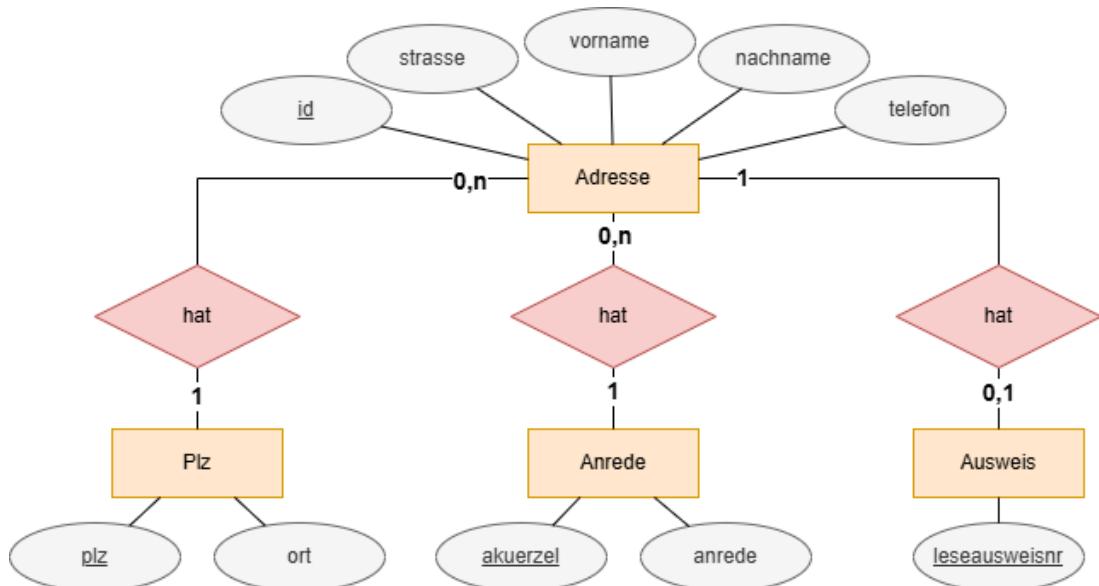
Company				Citys			Products				Material		
UID	Company	Street	CID	CID	Name	PLZ	UID	PID	Product	Price	MID	MID	Material
314	Lumitrax	Hauptstraße 45	1	1	Bern	3011	314	1	Silvaron Flex	12.90	1	1	Wood
776	Novara	Lindenstrasse 27	2	2	Basel	4051	314	2	LuminaGlass	10.50	2	2	Glass
479	Atheronix	Marktgasse 5	1				776	1	Thermora Core	45.60	2		
							776	2	Viroflex	12.90	1		
							776	3	QuantumMesh	22.00	3		
							479	1	PyroSteel	9.90	3		

# Kompetenznachweis 2 vom 08.05.25

Daten können auch als Mengenschreibweise in Textform oder als ER-Modell dargestellt werden. Der erste Teil der Mengenschreibweise besteht aus dem Tabellenname und in den runden Klammern sind anschliessend alle Attribute aufgelistet, wobei **Primärschlüssel** unterstrichen und **Fremdschlüssel** mit einer nachgestellten # dargestellt werden.

Die Struktur aus dem Script "Informationssysteme 1" als Mengenschreibweise und ER-Modell könnte folgendermassen ausschauen:

```
Adresse (id, vorname, nachname, akuerzel#, strasse, plz#, telefon)
Plz (plz, ort)
Anrede (akuerzel, anrede)
Ausweis (leseausweisnr, a-id#)
```

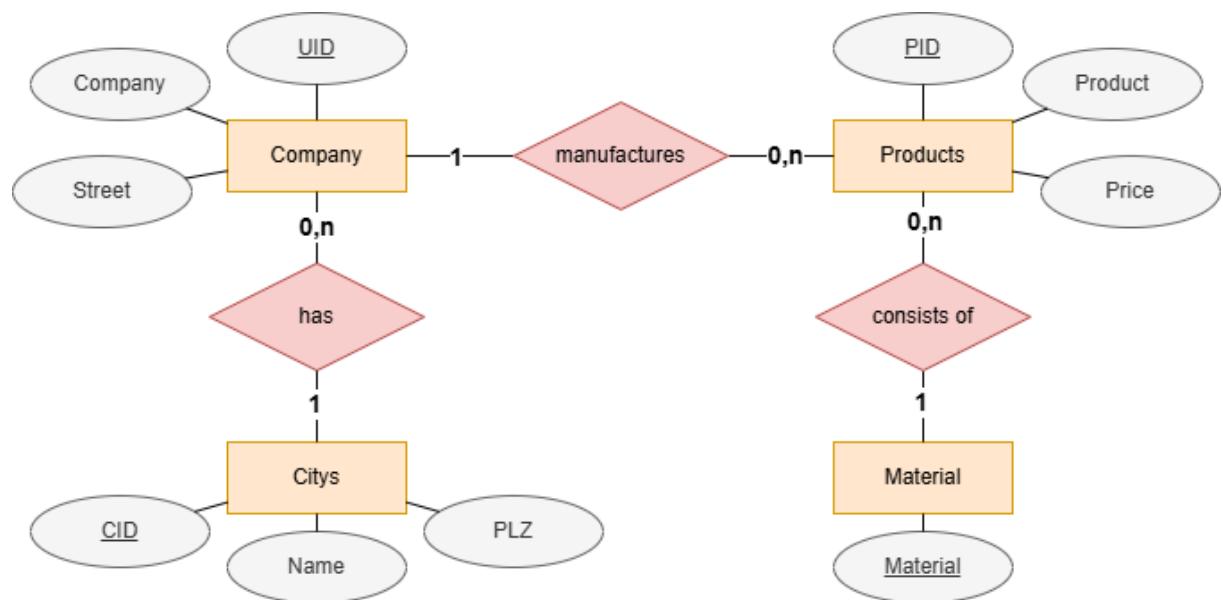


Die Mengenschreibweise und das ER-Modell für das zweite Beispiel würden dann folgendermassen ausschauen:

```

Company (UID, Company, Street, CID#)
Citys (CID, Name, PLZ)
Products (UID#, PID, Product, Price, MID#)
Material (MID, Material)

```



# Kompetenznachweis 3 vom 15.05.25

Ein ER-Modell kann nun weiter in ein ER-Diagramm umgewandelt werden. Angenommen wir möchten eine Bibliothek als ERD abbilden. Die Bibliothek hat Kunden und Autoren mit einer Adresse und Bücher, welche durch die Kunden bewertet und ausgeliehen werden können.

Für die Bewertung und für die Ausleihe muss jeweils eine weitere Tabelle angelegt werden, weil es sich dort um eine m:n-Beziehung handelt. Zusätzlich werden die zwei Tabellen noch mit weiteren Attributen angereichert (z.B. bei der Ausleihe mit einem Datum, von wann bis wann ein Buch ausgeliehen wird).

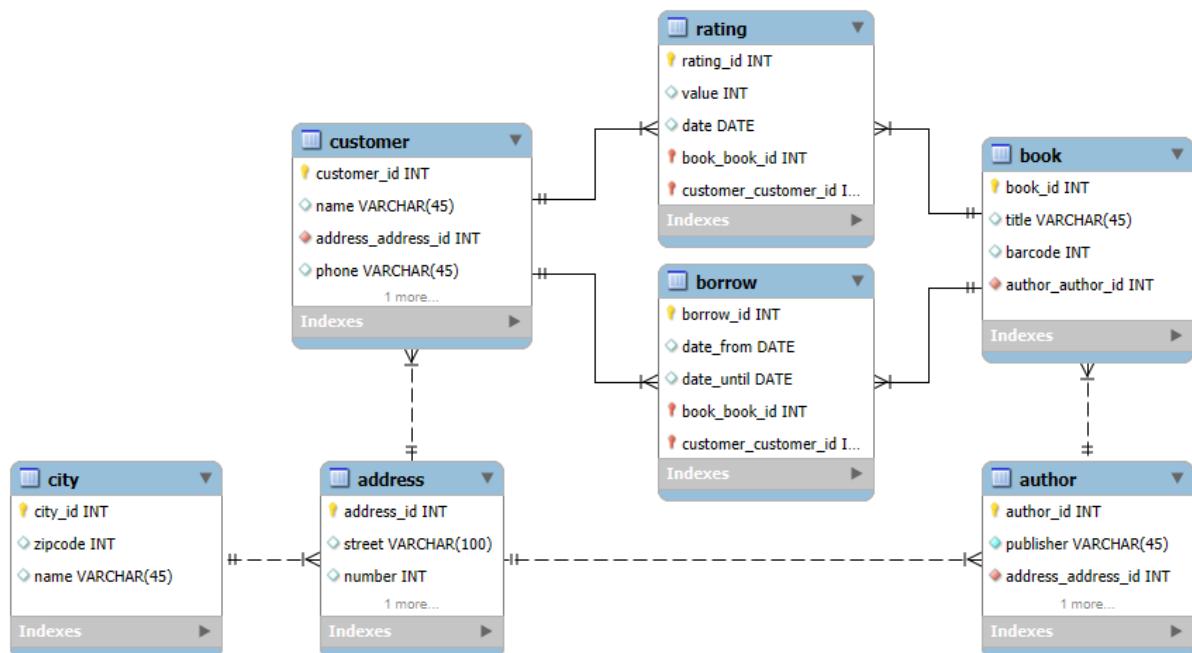
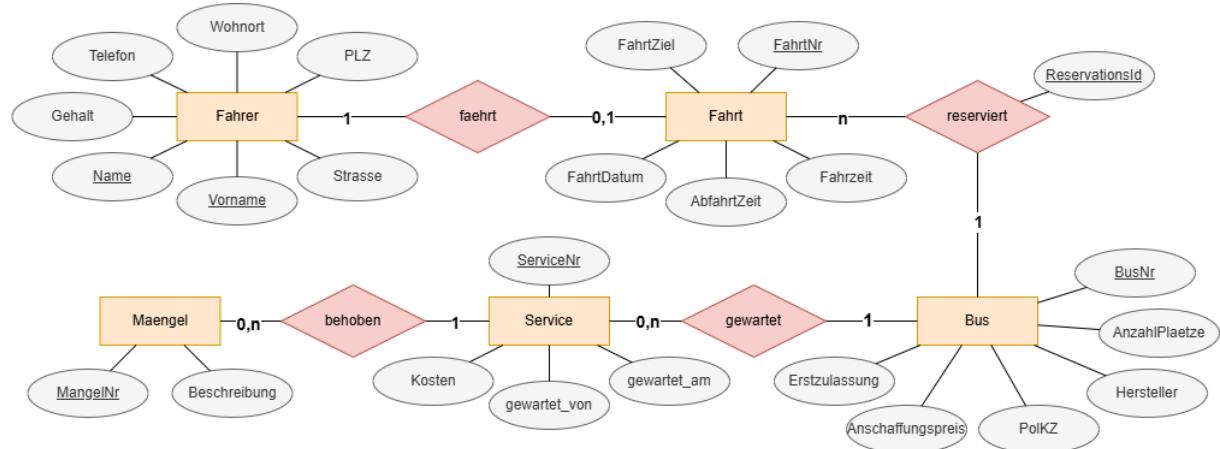


Diagramm mit MySQL Workbench erstellt

# Kompetenznachweis 4 vom 22.05.25

## Aufgabe 4.5.1 aus dem Script "Informationssysteme 1"

Die Daten des Busreiseunternehmens wurden in folgendem ER-Modell abgebildet:



Für das Busreiseunternehmen sind hier einige Beispieldaten abgebildet:

Fahrer								
FID	Name	Vorname	Strasse	Wohnort	PLZ	Telefon	Lohn	
1	Meier	Markus	Parkweg	Luzern	6003	0735845689	4800	
2	Braun	Hans	Bünten	Bern	3007	0733255874	4700	
3	Müller	Andreas	Bach	Sursee	6210	0732255512	5100	

Fahrt						
RID	Ziel	Datum	Zeit	Fahrzeit	FID	
1	Basel	05.03.25	07:30	2.0	1	
2	Chur	11.07.25	11:45	1.5	3	

Reservation		
RID	BID	Sitzplatz
1	2	A7
2	2	A3
1	1	B2

Bus					
BID	AnzahlPlaetze	Hersteller	PolKZ	Anschaffungspreis	Erstzulassung
1	30	Ford	LU08983	98000	22.11.21
2	40	Chevrolet	LU07758	112000	05.03.24

Service				
SID	BID	Kosten	Gewartet_am	Gewartet_von
1	2	3400	05.04.24	Thomas
2	1	1700	18.05.23	Markus
3	2	550	17.07.24	Simon

Maengel		
MID	SID	Beschreibung
1	1	Ölwechsel
2	1	Austausch Stossstange
3	2	Kraftstoffpumpe
4	3	Sicherheitsgurt

### Aufgabe 4.5.2 aus dem Script "Informationssysteme 1"

Die ERM Skizze wurde in ein ERD transformiert. Hierbei wurden für die Berufsabschlüsse und Standorte eine neue Tabelle angelegt. Das Alter sowie die Anzahl Angestellter in einer Abteilung werden nicht gespeichert, sondern anhand des Datums und der Anzahl Einträge bei der Abfrage berechnet. Im MySQL Workbench kann die Datenbank anschliessend mit der Option "Forward Engineer SQL Script" direkt generiert werden.

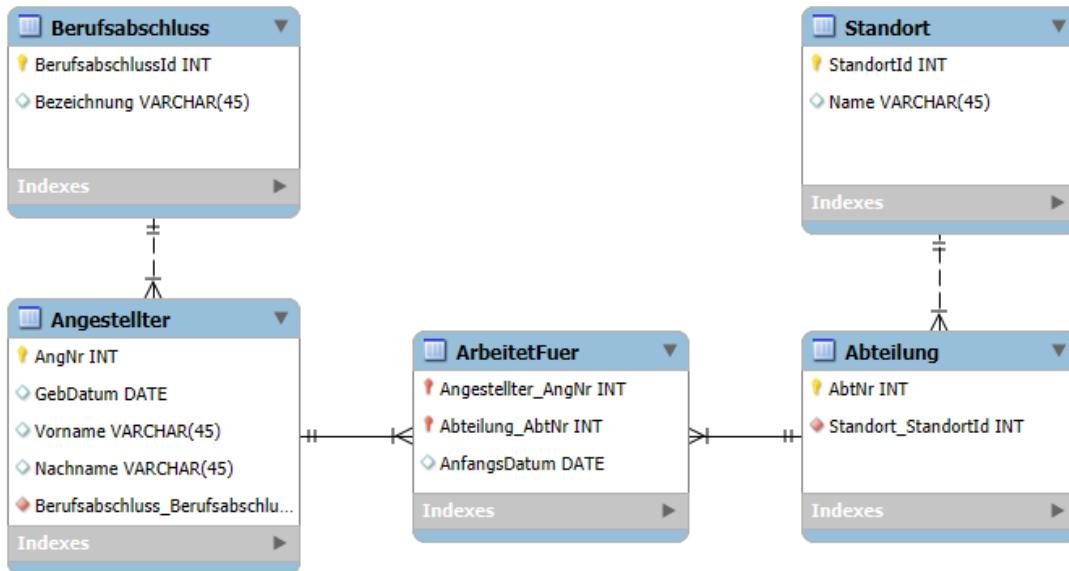
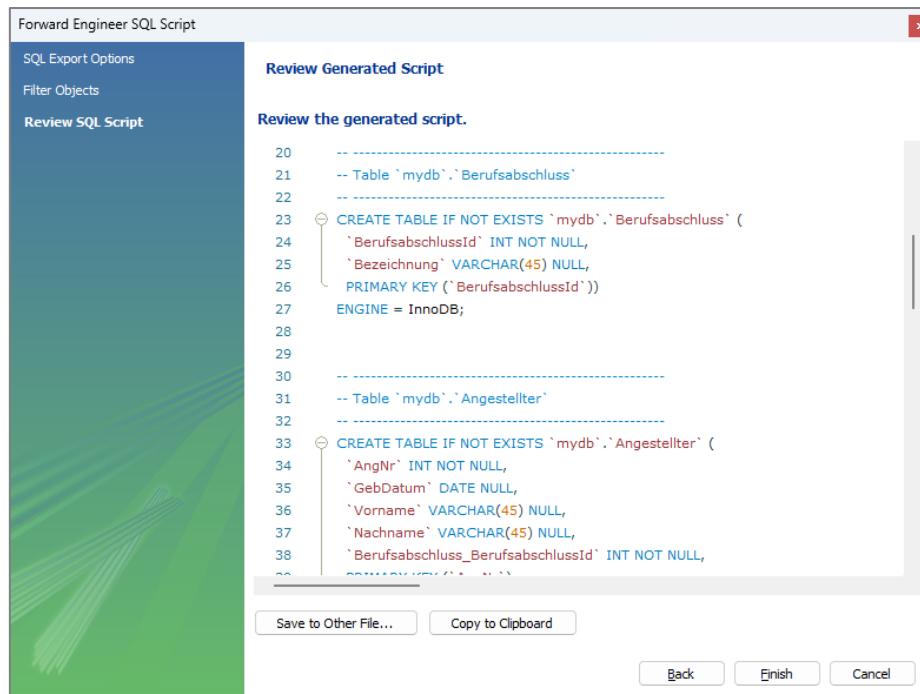


Diagramm mit MySQL Workbench erstellt



Datenbank direkt im MySQL-Workbench generieren

# Kompetenznachweis 5 vom 05.06.25

Für diese Übung mit SQL-Befehlen wird eine vorgefertigte Datenbank mit Ortschaften verwendet. Die Datenbank hat total 5'316 Datensätze bestehend aus dem Ort, dem Kanton und der Postleitzahl. Als Datenbank-Editor wird "DB Browser for SQLite" verwendet.

## SELECT

```
1 SELECT PLZ, ort
2 FROM ort;
3
4
5
```

PLZ	ort
1000	Lausanne
1000	Lausanne 1 Dépôt
1000	Lausanne 2
1000	Lausanne 3
1000	Lausanne 6
1000	Lausanne 7

## WHERE

```
1 SELECT PLZ, ort
2 FROM ort
3 WHERE PLZ > 3000;
4
5
6
```

PLZ	ort
3001	Bern
3002	Bern Postfinance
3003	Bern
3003	Bern 3 Nationalrat
3003	Bern 3 Ständerat
3004	Bern

Mit SELECT werden Daten von einer Tabelle abgefragt.  
Wenn alle Spalten abgefragt werden sollen, kann man ein Asterix (\*) verwenden.

Die Einträge können mir der WHERE-Klausen eingeschränkt werden.

## AND / OR

```
1 SELECT PLZ, ort, kanton
2 FROM ort
3 WHERE (PLZ = 1040 OR PLZ = 1054)
4 OR PLZ > 9600;
5
6
```

PLZ	ort	kanton
1040	Echallens	VD
1040	Villars-le-Terroir	VD
1040	St-Barthélemy	VD
1040	Echallens Dist	VD
1054	Morrens	VD
9601	Lütisburg Station	SG

## IN

```
1 SELECT PLZ, ort, kanton
2 FROM ort
3 WHERE ort
4 IN ('Luzern','Bern');
5
6
```

PLZ	ort	kanton
3000	Bern	BE
3001	Bern	BE
3003	Bern	BE
3004	Bern	BE
3005	Bern	BE
3006	Bern	BE

Mit AND und OR können mehrere Einschränkungen in einer Abfrage vereint werden.

Es werden alle Datensätze für die Orte "Luzern" und "Bern" ausgegeben. Wildcards sind hier nicht erlaubt.

## BETWEEN

```
1 SELECT PLZ, ort, kanton
2 FROM ort
3 WHERE PLZ
4 BETWEEN 6060 AND 6070;
5
6
```

PLZ	ort	kanton
6060	Sarnen	OW
6060	Sarnen 1	OW
6060	Ramersberg	OW
6060	Sarnen 2 Büntemp.	OW
6060	Sarnen 2	OW
6060	Sarnen 2 Zust	OW

Mit BETWEEN findet man alle Einträge, welche sich dazwischen befinden.

## LIKE

```
1 SELECT PLZ, ort, kanton
2 FROM ort
3 WHERE ort
4 LIKE "B%e_";
5
6
```

PLZ	ort	kanton
1038	Bercher	VD
1136	Bussy-Chardonney	VD
1195	Bursinel	VD
1233	Bernex	GE
1277	Borex	VD
1279	Bogis-Bossey	VD

Die LIKE-Klausel ist sehr mächtig. Hier können mit Platzhalter gearbeitet werden. Underline (\_) steht dabei für ein beliebiges Zeichen und Prozent (%) für beliebig viele.

### ORDER BY

```
1  SELECT PLZ, ort, kanton
2  FROM ort
3  ORDER BY ort DESC;
4
5
6
```

	PLZ	ort	kanton
1	8080	Zürich West SPS	ZH
2	8074	Zürich Voice Pub	ZH
3	8085	Zürich Versich.	ZH
4	8008	Zürich Tbrunnen	ZH
5	8010	Zürich SPS AG	ZH
6	8086	Zürich R Digest	ZH

Daten können mit ORDER BY sortiert werden, wobei standardmäßig aufsteigend und mit dem Zusatz DESC absteigend sortiert werden kann.

### GROUP BY

```
1  SELECT kanton, count(ort)
2  FROM ort
3  GROUP BY kanton;
4
5
6
```

	kanton	count(ort)
1	AG	355
2	AI	17
3	AR	33
4	BE	713
5	BL	126
6	BS	70

Es können auch Daten zusammengefasst bzw. gruppiert werden. Zusätzlich wurden hier mit COUNT die Ortschaften pro Kanton gezählt.

### HAVING

```
1  SELECT PLZ, count(ort)
2  FROM ort
3  GROUP BY PLZ
4  HAVING PLZ > 6000 AND PLZ < 6100;
5
6
```

	PLZ	count(ort)
1	6002	2
2	6003	1
3	6004	1
4	6005	2
5	6006	1
6	6007	1

Die Gruppierung kann mit HAVING weiter eingeschränkt werden. Hier werden die Anzahl Ortschaften pro Postleitzahl angezeigt, welche sich zwischen 6000 und 6100 befinden.

### COUNT

```
1  SELECT COUNT(kanton) as ANZAHL
2  FROM ort
3  WHERE kanton IN ('LU');
4
5
6
```

	ANZAHL
1	210

Mit COUNT können die Zeilen gezählt werden, welche mit der Abfrage übereinstimmen.

# Kompetenznachweis 6 vom 12.06.25

Als Aufgabe wurde im Script ein ERM bereitgestellt, welches in ein ERD in der 3. Normalform transformiert werden soll. Das ERD wurde wie nachfolgend abgebildet umgesetzt.

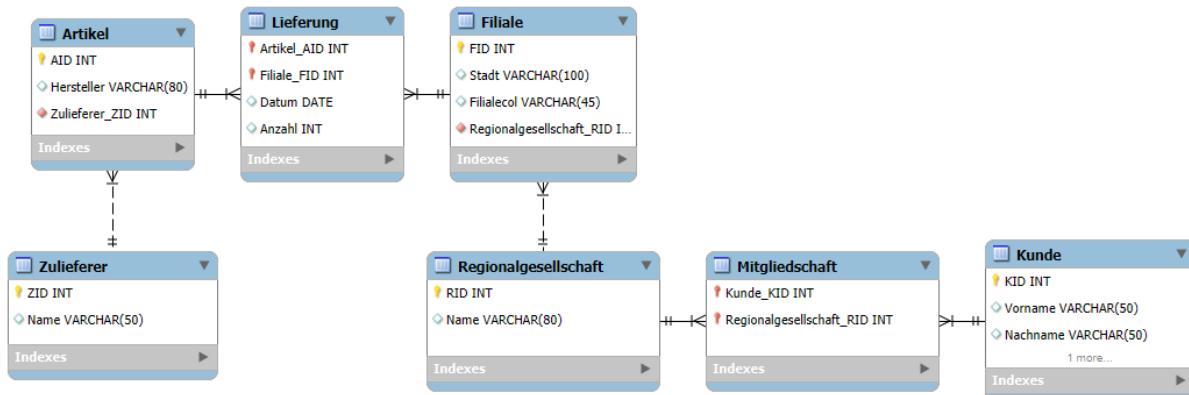
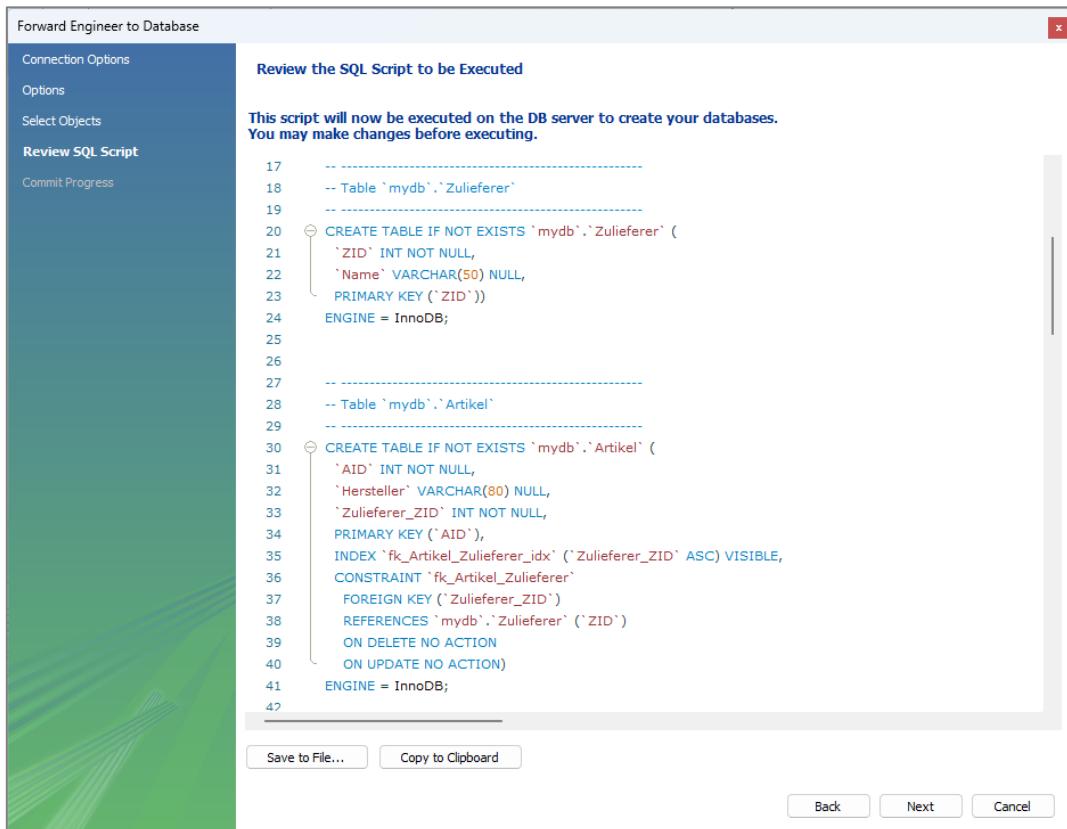


Diagramm mit MySQL Workbench erstellt

Die Datenbank mit den zugehörigen Tabellen konnte dann mit wenigen Klicks direkt erstellt werden. Im MySQL Workbench wird diese Funktion «Forward Engineer» genannt.



Datenbank und Tabellen generieren

Anschliessend wurden noch Zufallsdaten in die Datenbank eingepflegt. Beim Editieren über das MySQL Workbench wird vor dem Ausführen immer noch der SQL-Code angezeigt. Wenn die Daten fehlerhaft sind, erscheint eine Meldung. Beim manuellen Eingeben kann z.B. das Datum im falschen Format sein oder dass ein Fremdschlüssel verwendet wird, welcher eigentlich gar nicht existiert. Diese Fehler muss man dann zuerst bereinigen.

The screenshot shows the MySQL Workbench interface. The left pane is the Navigator, displaying the database structure. The 'SCHEMAS' section shows the 'mydb' schema with tables: artikel, filiale, kunde, lieferung, and mitgliedschaft. The 'kunde' table is selected. The right pane shows the results of a query: 'SELECT \* FROM mydb.kunde;'. The results are displayed in a grid:

	KID	Vorname	Nachname	Geburtsdatum
1	1	Kaelen	Viroth	1995-02-05
2	2	Liora	Maris	1981-05-11
3	3	Taren	Solvik	1998-01-17
4	4	Mirae	Jorlan	2001-04-04
*	NULL	NULL	NULL	NULL

Tabelle «Kunde» mit Beispieldaten

# Kompetenznachweis 7 vom 26.06.25

In diesem Beispiel wurde eine rudimentäre Projektverwaltungsdatenbank erstellt. Folgende Einstellungen wurde dabei implementiert:

- Grundsätzlich wurde für alle Attribute die Option «NOT NULL» aktiviert, ausser für das Attribut «Employee/Salary».
- Ein «Department» kann einem «Employee» und auch einem «Project» zugewiesen werden.
- Die Zuweisung der Mitarbeiter an einem Projekt passiert über eine separate Tabelle, welche als Primärschlüssel «EID» und «PID» hat.
- Wird ein Projekt oder ein Mitarbeiter gelöscht, so wird automatisch auch die Tabelle der Mitarbeiter bereinigt («CONSTRAINT ON UPDATE CASCADE»).
- Für alle Primärschlüssel wurde die Option «Auto Increment» aktiviert. Dadurch erhalten diese Datensätze immer eine neue, fortlaufende Nummerierung.

Hier sieht man, dass auch bereits kleine Datenbanken etwas Logik und Komplexität erfordern. Das nachfolgend Abgebildete ERD wurde anschliessend erstellt und mit Daten befüllt.

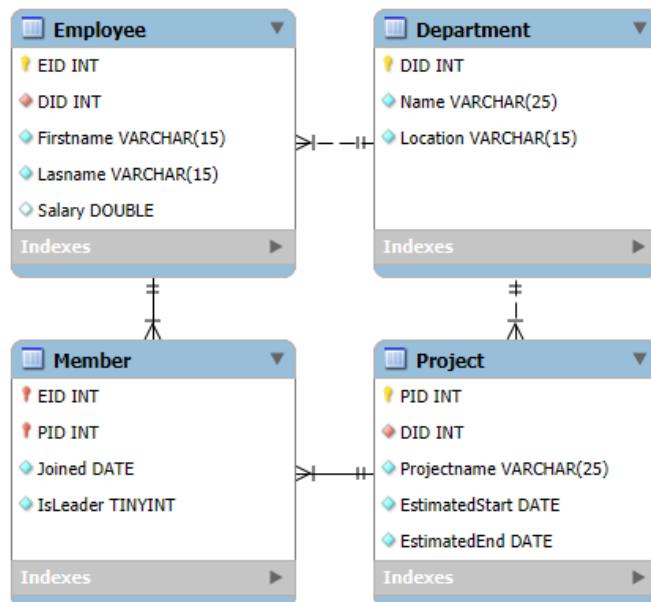
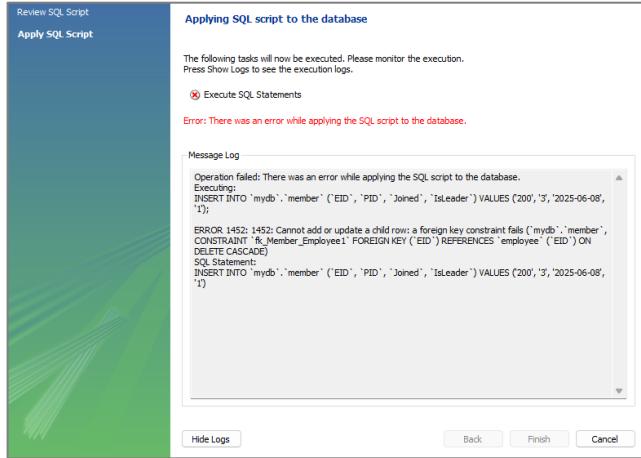


Diagramm mit MySQL Workbench erstellt

Nachdem die Datenbank mit Beispieldaten befüllt wurde, sind noch 2 Tests durchgeführt worden. Mit diesen wurde sichergestellt, dass das Schema korrekt ist und wie gewünscht funktioniert.

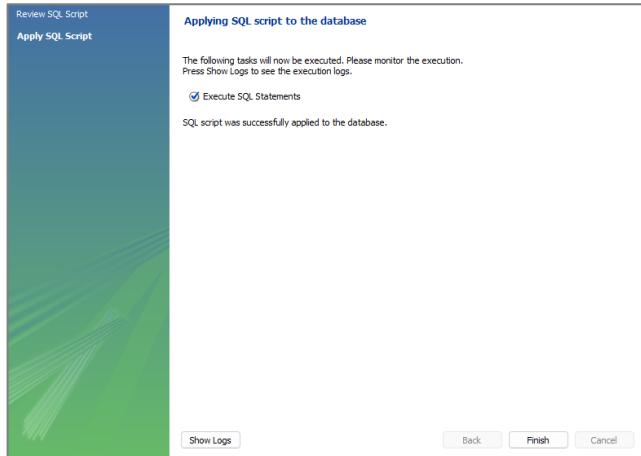
### Test 1 - Ungültiger Fremdschlüssel verwenden



ERROR 1452: 1452: Cannot add or update a child row: a foreign key constraint fails.

➔ Dieser Test war erfolgreich.

### Test 2 – Mitarbeiter löschen und prüfen, ob die Projektmitglieder bereinigt werden



SQL Statement executed successfully. No errors.

➔ Dieser Test war erfolgreich.