

Buchtausch-App – Phase 2

Entitätsdokumentation (CREATE +
INSERT + Screenshots)

Anton Hirsch – IU14119299

Tabelle: adresse – SQL Statements

```
CREATE TABLE adresse (
```

```
    adresse_id INT AUTO_INCREMENT PRIMARY KEY,  
    strasse VARCHAR(120),  
    hausnummer VARCHAR(10),  
    plz VARCHAR(10),  
    ort VARCHAR(120)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO adresse (strasse, hausnummer, plz, ort) VALUES  
('Hauptstraße', '12', '10115', 'Berlin'),  
('Bahnhofstraße', '7A', '80331', 'München'),  
('Musterweg', '5', '50667', 'Köln'),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: adresse – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0002 Sekunden.) [ort: AUGSBURG... - KÖLN...] [plz: 86150... - 50667...]

```
SELECT * FROM adresse ORDER BY ort, plz LIMIT 10;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

Zusätzliche Optionen

				adresse_id	strasse	hausnummer	plz	ort
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	17	Ostring	21	86150	Augsburg
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	1	Hauptstraße	12	10115	Berlin
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	9	Parkallee	3	28195	Bremen
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	19	Feldstraße	18	28195	Bremen
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	6	Schillerplatz	1	01067	Dresden
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	10	Ringstraße	27	60311	Frankfurt am Main
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	4	Lindenallee	23	20095	Hamburg
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	13	Bergstraße	9	34117	Kassel
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	8	Seestraße	8	24103	Kiel
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	3	Musterweg	5	50667	Köln

 ☐ Alle auswählen markierte:  Bearbeiten  Kopieren  Löschen  Exportieren

Tabelle: benutzer – SQL Statements

```
CREATE TABLE benutzer (
```

```
    benutzer_id INT AUTO_INCREMENT PRIMARY KEY,  
    benutzername VARCHAR(50) NOT NULL UNIQUE,  
    vorname VARCHAR(100) NOT NULL,  
    nachname VARCHAR(100) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    passwort_hash VARCHAR(255) NOT NULL,  
    adresse_id INT,  
    status VARCHAR(10) NOT NULL,      -- z. B. 'aktiv', 'gesperrt'  
    CONSTRAINT fk_benutzer_adresse  
        FOREIGN KEY (adresse_id)  
        REFERENCES adresse(adresse_id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO benutzer (benutzername, vorname, nachname, email, passwort_hash, adresse_id, status) VALUES  
('anna.k', 'Anna', 'Klein', 'anna.k@example.com', 'hash_anna', 1, 'aktiv'),  
('max.m', 'Max', 'Müller', 'max.m@example.com', 'hash_max', 2, 'aktiv'),  
('sara.s', 'Sara', 'Schmidt', 'sara.s@example.com', 'hash_sara', 3, 'aktiv'),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: benutzer – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0009 Sekunden.) [benutzer_id: 1... - 10...]

```
SELECT b.benutzer_id, b.benutzername, a.ort FROM benutzer b LEFT JOIN adresse a ON a.adresse_id = b.adresse_id WHERE b.status = 'aktiv' ORDER BY b.benutzer_id LIMIT 10;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

Zusätzliche Optionen

benutzer_id	benutzername	ort
1	anna.k	Berlin
2	max.m	München
3	sara.s	Köln
4	timo.b	Hamburg
5	lena.h	Stuttgart
6	paul.w	Dresden
7	mia.f	Nürnberg
8	jonas.r	Kiel
9	klara.s	Bremen
10	felix.w	Frankfurt am Main

Tabelle: admin – SQL Statements

```
CREATE TABLE admin (  
  
    admin_id INT PRIMARY KEY,  
    CONSTRAINT fk_admin_benutzer  
        FOREIGN KEY (admin_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO admin (admin_id) VALUES  
(11),  
(2),  
(5),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: admin – Testfall Screenshot

✓ Zeige Datensätze 0 - 2 (3 insgesamt, Die Abfrage dauerte 0,0008 Sekunden.) [admin_id: 2... - 11...]

```
SELECT ad.admin_id, b.username FROM admin ad INNER JOIN benutzer b ON b.benutzer_id = ad.admin_id ORDER BY ad.admin_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

admin_id	1	username
----------	---	----------

2	max.m
---	-------

5	lena.h
---	--------

11	admin.1
----	---------

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Tabelle: verleihher – SQL Statements

```
CREATE TABLE verleihher (  
  
    verleihher_id INT PRIMARY KEY,  
    CONSTRAINT fk_verleihher_benutzer  
        FOREIGN KEY (verleihher_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO verleihher (verleihher_id) VALUES  
(1),  
(2),  
(4),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```


Tabelle: verleiher – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0012 Sekunden.) [verleiher_id: 1... - 16...]

```
SELECT v.verleiher_id, b.benutzername FROM verleiher v JOIN benutzer b ON b.benutzer_id = v.verleiher_id ORDER BY v.verleiher_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

verleiher_id ▲ 1	benutzername
1	anna.k
2	max.m
4	timo.b
5	lena.h
7	mia.f
8	jonas.r
10	felix.w
12	mod.1
14	eva.l
16	luis.g

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Tabelle: ausleiher – SQL Statements

```
CREATE TABLE ausleiher (  
  
    ausleiher_id INT PRIMARY KEY,  
    CONSTRAINT fk_ausleiher_benutzer  
        FOREIGN KEY (ausleiher_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO ausleiher (ausleiher_id) VALUES  
(1),  
(3),  
(4),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: ausleiher – Testfall Screenshot

✓ Zeige Datensätze 0 - 15 (16 insgesamt, Die Abfrage dauerte 0,0006 Sekunden.) [ausleiher_id: 1... - 20...]

```
SELECT a.ausleiher_id, b.beutzername FROM ausleiher a JOIN benutzer b ON b.beutzer_id = a.ausleiher_id ORDER BY a.ausleiher_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

ausleiher_id ▲ 1	benutzername
1	anna.k
3	sara.s
4	timo.b
6	paul.w
7	mia.f
8	jonas.r
9	klara.s
10	felix.w
11	admin.1
12	mod.1
13	nils.k
15	ole.p
17	nina.b
18	yara.o
19	tim.s
20	zoe.h

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Tabelle: verlag – SQL Statements

```
CREATE TABLE verlag (  
  
    verlag_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL UNIQUE,  
    adresse_id INT,  
    CONSTRAINT fk_verlag_adresse  
        FOREIGN KEY (adresse_id)  
        REFERENCES adresse(adresse_id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO verlag (name, adresse_id) VALUES  
('Verlag Morgenstern', 3),  
('Nordlicht Verlag', 4),  
('Südwind Press', 5),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: verlag – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0010 Sekunden.) [name: ALPENBLICK VERLAG... - WESTPARK MEDIEN...]

```
SELECT v.verlag_id, v.name, a.ort FROM verlag v LEFT JOIN adresse a ON a.adresse_id = v.adresse_id ORDER BY v.name;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

verlag_id	name ▲ 1	ort
7	Alpenblick Verlag	München
10	Donau Edition	Stuttgart
9	Elbstrand Print	Hamburg
6	Hauptstadt Verlag	Berlin
2	Nordlicht Verlag	Hamburg
4	Ostsee Bücher	Kiel
8	Rheinland Books	Köln
3	Südwind Press	Stuttgart
1	Verlag Morgenstern	Köln
5	Westpark Medien	Bremen

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Tabelle: autor – SQL Statements

```
CREATE TABLE autor (  
  
    autor_id INT AUTO_INCREMENT PRIMARY KEY,  
    vorname VARCHAR(100),  
    nachname VARCHAR(100)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO autor (vorname, nachname) VALUES  
('Thomas', 'Mann'),  
('Franz', 'Kafka'),  
('Hermann', 'Hesse'),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: autor – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [titel: BUDDENBROOKS... - UNTERLEUTEN...]

```
SELECT w.titel, CONCAT(a.vorname, ' ', a.nachname) AS autor FROM werk_autor NATURAL JOIN autor a JOIN buch_werk w ON w.werk_id = werk_autor.werk_id ORDER BY w.titel;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern:

Zusätzliche Optionen

titel ▲ 1	autor
Buddenbrooks	Thomas Mann
Das Parfum	Patrick Süskind
Der Prozess	Franz Kafka
Der Steppenwolf	Hermann Hesse
Die Blechtrommel	Günter Grass
Die Physiker	Friedrich Dürrenmatt
Die Vermessung der Welt	Daniel Kehlmann
Effi Briest	Theodor Fontane
Nachtzug nach Lissabon	Pascal Mercier
Tintenherz	Cornelia Funke
Tyll	Daniel Kehlmann
Unterleuten	Juli Zeh

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern:

Tabelle: genre – SQL Statements

```
CREATE TABLE genre (  
  
    genre_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(80) UNIQUE,  
    beschreibung TEXT  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO genre (name, beschreibung) VALUES  
('Roman', 'Erzählende Prosa'),  
('Novelle', 'Kürzere Erzählform'),  
('Fantasy', 'Fantastische Welten'),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```


Tabelle: genre – Testfall Screenshot

✔ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0008 Sekunden.)

```
SELECT g.genre_id, g.name AS genre, COUNT(wg.werk_id) AS anzahl_werke FROM werk_genre wg RIGHT JOIN genre g ON g.genre_id = wg.genre_id GROUP BY g.genre_id, g.name ORDER BY anzahl_werke DESC, g.name;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen

Anzahl der Datensätze:

25 ▼

Zeilen filtern:

Zusätzliche Optionen

genre_id	genre	anzahl_werke	▼ 1
1	Roman	10	
9	Historisch	3	
10	Drama	1	
3	Fantasy	1	
11	Jugendbuch	1	
4	Krimi	1	
6	Biografie	0	
12	Essay	0	
8	Lyrik	0	
2	Novelle	0	
5	Sachbuch	0	
7	Science Fiction	0	

☐ Alles anzeigen

Anzahl der Datensätze:

25 ▼

Zeilen filtern:

Tabelle: buch_werk – SQL Statements

```
CREATE TABLE buch_werk (
```

```
    werk_id INT AUTO_INCREMENT PRIMARY KEY,  
    titel VARCHAR(255) NOT NULL,  
    isbn CHAR(13) NOT NULL UNIQUE,  
    erscheinungsjahr INT,  
    sprache VARCHAR(40),  
    verlag_id INT,  
    CONSTRAINT fk_werk_verlag  
        FOREIGN KEY (verlag_id)  
        REFERENCES verlag(verlag_id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO buch_werk (titel, isbn, erscheinungsjahr, sprache, verlag_id) VALUES  
('Buddenbrooks', '9780000000001', 1901, 'Deutsch', 1),  
('Der Prozess', '9780000000002', 1925, 'Deutsch', 2),  
('Der Steppenwolf', '9780000000003', 1927, 'Deutsch', 3),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: buch_werk – Testfall Screenshot

✓ Zeige Datensätze 0 - 4 (5 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [titel: TYLL... - TINTENHERZ...]

```
SELECT werk_id, titel, isbn FROM buch_werk WHERE erscheinungsjahr >= 2000 ORDER BY erscheinungsjahr DESC, titel;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

				werk_id	titel	isbn
<input type="checkbox"/>		Bearbeiten	Kopieren	Löschen	6 Tyll	9780000000006
<input type="checkbox"/>		Bearbeiten	Kopieren	Löschen	9 Unterleuten	9780000000009
<input type="checkbox"/>		Bearbeiten	Kopieren	Löschen	8 Die Vermessung der Welt	9780000000008
<input type="checkbox"/>		Bearbeiten	Kopieren	Löschen	10 Nachtzug nach Lissabon	9780000000010
<input type="checkbox"/>		Bearbeiten	Kopieren	Löschen	12 Tintenherz	9780000000012

↑ ☐ Alle auswählen markierte: Bearbeiten Kopieren Löschen Exportieren

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▼

Tabelle: werk_autor – SQL Statements

```
CREATE TABLE werk_autor (
```

```
    werk_id INT NOT NULL,  
    autor_id INT NOT NULL,  
    rolle VARCHAR(100),  
    PRIMARY KEY (werk_id, autor_id),  
    CONSTRAINT fk_wa_werk  
        FOREIGN KEY (werk_id)  
        REFERENCES buch_werk(werk_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_wa_autor  
        FOREIGN KEY (autor_id)  
        REFERENCES autor(autor_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO werk_autor (werk_id, autor_id, rolle) VALUES  
(1, 1, 'Autor'),  
(2, 2, 'Autor'),  
(3, 3, 'Autor'),  
... ;
```

-- Testfall siehe Screenshot nächste Folie --

Tabelle: werk_autor – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0010 Sekunden.) [titel: BUDDENBROOKS... - UNTERLEUTEN...] [nachname: MANN... - ZEH...]

```
SELECT wa.werk_id, w.titel, a.nachname FROM werk_autor wa INNER JOIN buch_werk w ON w.werk_id = wa.werk_id INNER JOIN autor a ON a.autor_id = wa.autor_id ORDER BY w.titel, a.nachname;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern:

Zusätzliche Optionen

werk_id	titel ▲ 1	nachname
1	Buddenbrooks	Mann
5	Das Parfum	Süskind
2	Der Prozess	Kafka
3	Der Steppenwolf	Hesse
4	Die Blechtrommel	Grass
11	Die Physiker	Dürrenmatt
8	Die Vermessung der Welt	Kehlmann
7	Effi Briest	Fontane
10	Nachtzug nach Lissabon	Mercier
12	Tintenherz	Funke
6	Tyll	Kehlmann
9	Unterleuten	Zeh

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern:

Tabelle: werk_genre – SQL Statements

```
CREATE TABLE werk_genre (  
  
    werk_id INT NOT NULL,  
    genre_id INT NOT NULL,  
    PRIMARY KEY (werk_id, genre_id),  
    CONSTRAINT fk_wg_werk  
        FOREIGN KEY (werk_id)  
        REFERENCES buch_werk(werk_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_wg_genre  
        FOREIGN KEY (genre_id)  
        REFERENCES genre(genre_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO werk_genre (werk_id, genre_id) VALUES  
(1, 1),  
(2, 1),  
(3, 1),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: werk_genre – Testfall Screenshot

✓ Zeige Datensätze 0 - 16 (17 insgesamt, Die Abfrage dauerte 0,0005 Sekunden.) [titel: BUDDENBROOKS... - UNTERLEUTEN...]

```
SELECT w.titel, g.name AS genre FROM werk_genre NATURAL JOIN genre g JOIN buch_werk w ON w.werk_id = werk_genre.werk_id ORDER BY w.titel, g.name;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen

Anzahl der Datensätze: 25 ▼

Zeilen filtern: Diese Tabelle durchsuchen

Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

titel ▲ 1	genre
Buddenbrooks	Roman
Das Parfum	Krimi
Das Parfum	Roman
Der Prozess	Roman
Der Steppenwolf	Roman
Die Blechtrommel	Historisch
Die Blechtrommel	Roman
Die Physiker	Drama
Die Vermessung der Welt	Historisch
Die Vermessung der Welt	Roman
Effi Briest	Roman
Nachtzug nach Lissabon	Roman
Tintenherz	Fantasy
Tintenherz	Jugendbuch
Tyll	Historisch
Tyll	Roman
Unterleuten	Roman

☐ Alles anzeigen

Anzahl der Datensätze: 25 ▼

Zeilen filtern: Diese Tabelle durchsuchen

Nach Schlüssel sortieren: keine ▼

Tabelle: zustand – SQL Statements

```
CREATE TABLE zustand (  
  
    zustand_id INT AUTO_INCREMENT PRIMARY KEY,  
    stufe VARCHAR(30) NOT NULL,  
    beschreibung TEXT  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO zustand (stufe, beschreibung) VALUES  
('neu', 'Wie frisch aus dem Laden'),  
('sehr gut', 'Minimale Gebrauchsspuren'),  
('gut', 'Leichte Gebrauchsspuren'),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```


Tabelle: zustand – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0002 Sekunden.) [stufe: **ABGENUTZT...** - SEHR GUT...]

`SELECT * FROM zustand ORDER BY stufe;`

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: Zeilen filtern: Nach Schlüssel sortieren:

Zusätzliche Optionen

<div><div><div></div><div></div></div><div></div></div>			zustand_id	stufe	beschreibung
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	6	abgenutzt	Stärker beansprucht, voll lesbar
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	10	antik	Sehr alt, empfindlich
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	8	bibliotheksex	Stempel/Signatur vorhanden
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	5	gebraucht	Gut lesbar, sichtbare Nutzung
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	3	gut	Leichte Gebrauchsspuren
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	7	mit Notizen	Eintragungen vorhanden
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	1	neu	Wie frisch aus dem Laden
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	4	ordentlich	Einige Knicke oder Markierungen
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	9	sammelwürdig	Besonders gepflegt
<input type="checkbox"/>	 Bearbeiten	 Kopieren  Löschen	2	sehr gut	Minimale Gebrauchsspuren

☐ Alle auswählen markierte: Bearbeiten Kopieren Löschen Exportieren

Tabelle: abholort – SQL Statements

```
CREATE TABLE abholort (  
  
    abholort_id INT AUTO_INCREMENT PRIMARY KEY,  
    verleih_id INT NOT NULL,  
    adresse_id INT NOT NULL,  
    bezeichnung VARCHAR(120) NOT NULL, -- z. B. 'Haustür', 'Packstation'  
    hinweis TEXT,  
    CONSTRAINT fk_abholort_verleih  
        FOREIGN KEY (verleih_id)  
        REFERENCES verleih(verleih_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_abholort_adresse  
        FOREIGN KEY (adresse_id)  
        REFERENCES adresse(adresse_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO abholort (verleih_id, adresse_id, bezeichnung, hinweis) VALUES  
(1, 1, 'Haustür', 'Bitte klingeln'),  
(2, 2, 'Packstation', 'DHL-Box 102'),  
(4, 4, 'Haustür', 'Nach 18 Uhr'),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: abholort – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0005 Sekunden.) [ort: BERLIN... - STUTTGART...][bezeichnung: HAUSTÜR... - ARBEITSPLATZ...]

```
SELECT ab.abholort_id, b.benutzername AS verleiherv, a.ort, ab.bezeichnung FROM abholort ab JOIN verleiherv v ON v.verleiher_id = ab.verleiher_id JOIN benutzer b ON b.benutzer_id = v.verleiher_id JOIN adresse a ON a.adresse_id = ab.adresse_id ORDER BY a.ort, ab.bezeichnung;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine

Zusätzliche Optionen

abholort_id	verleiher	ort	bezeichnung
1	anna.k	Berlin	Haustür
12	lena.h	Dresden	Garage
7	felix.w	Frankfurt am Main	Haustür
3	timo.b	Hamburg	Haustür
6	jonas.r	Kiel	Nachbar
11	max.m	Köln	Nachbar
8	mod.t	Leipzig	Packstation
10	luis.g	Magdeburg	Haustür
2	max.m	München	Packstation
5	mia.f	Nürnberg	Garage
9	eva.l	Potsdam	Haustür
4	lena.h	Stuttgart	Arbeitsplatz

☐ Alles anzeigen | Anzahl der Datensätze: 25 | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine

Tabelle: zeitfenster – SQL Statements

```
CREATE TABLE zeitfenster (  
  
    zeitfenster_id INT AUTO_INCREMENT PRIMARY KEY,  
    abholort_id INT NOT NULL,  
    wochentag INT,          -- 0–6 (So–Sa)  
    von_uhrzeit TIME,  
    bis_uhrzeit TIME,  
    CONSTRAINT fk_zeitfenster_abholort  
        FOREIGN KEY (abholort_id)  
        REFERENCES abholort(abholort_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT chk_zeitfenster_tag CHECK (wochentag BETWEEN 0 AND 6)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO zeitfenster (abholort_id, wochentag, von_uhrzeit, bis_uhrzeit) VALUES  
(1, 1, '17:00:00', '20:00:00'),  
(2, 2, '08:00:00', '12:00:00'),  
(3, 3, '18:00:00', '21:00:00'),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: zeitfenster – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [wochentag: 1... - 5...] [von_uhrzeit: 17:00:00... - 17:30:00...]

```
SELECT z.zeitfenster_id, ab.bezeichnung, z.wochentag, z.von_uhrzeit, z.bis_uhrzeit FROM zeitfenster z JOIN abholort ab ON ab.abholort_id = z.abholort_id WHERE z.wochentag BETWEEN 1 AND 5 ORDER BY z.wochentag, z.von_uhrzeit;
```

☐ Messen [\[Inline bearbeiten\]](#) [\[Bearbeiten\]](#) [\[SQL erklären\]](#) [\[PHP-Code erzeugen\]](#) [\[Aktualisieren\]](#)

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

zeitfenster_id	bezeichnung	wochentag	von_uhrzeit	bis_uhrzeit
1	Haustür	1	17:00:00	20:00:00
8	Packstation	1	19:00:00	21:00:00
9	Haustür	2	07:30:00	09:00:00
2	Packstation	2	08:00:00	12:00:00
10	Haustür	3	12:00:00	14:00:00
3	Haustür	3	18:00:00	21:00:00
4	Arbeitsplatz	4	09:00:00	11:00:00
11	Nachbar	4	15:00:00	18:00:00
5	Garage	5	16:00:00	19:00:00
12	Garage	5	17:30:00	19:30:00

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine ▼

Tabelle: buch_exemplar – SQL Statements – Teil 1

```
CREATE TABLE buch_exemplar (  
  
    exemplar_id INT AUTO_INCREMENT PRIMARY KEY,  
    werk_id INT NOT NULL,  
    verleih_id INT NOT NULL,  
    zustand_id INT,  
    verleihstatus VARCHAR(20),    -- 'verfügbar', 'ausgeliehen', 'reserviert'  
    max_ausleihdauer INT,        -- in Tagen  
    abholort_id INT,  
    versandoption BOOLEAN,  
    CONSTRAINT fk_exemplar_werk  
        FOREIGN KEY (werk_id)  
        REFERENCES buch_werk(werk_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_exemplar_verleiher  
        FOREIGN KEY (verleiher_id)  
        REFERENCES verleih(verleiher_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_exemplar_zustand  
        FOREIGN KEY (zustand_id)  
        REFERENCES zustand(zustand_id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_exemplar_abholort  
        FOREIGN KEY (abholort_id)  
        REFERENCES abholort(abholort_id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
-- Insert Statement, Testfall siehe Screenshot nächste Folie --
```

Tabelle: buch_exemplar – SQL Statements Teil 2 – Testfall Screenshot

```
INSERT INTO buch_exemplar (werk_id, verleih_id, zustand_id, verleihstatus, max_ausleihdauer, abholort_id, versandooption) VALUES
(1, 1, 1, 'verfügbar', 21, 1, TRUE),
(2, 2, 3, 'verfügbar', 14, 2, FALSE),
(3, 4, 2, 'ausgeliehen', 28, 3, TRUE),
... ;
```

✓ Zeige Datensätze 0 - 19 (20 insgesamt, Die Abfrage dauerte 0,0008 Sekunden.) [titel: BUDDENBROOKS... - UNTERLEUTEN...] [exemplar_id: 1... - 9...]

```
SELECT e.exemplar_id, w.titel, b.benutzername AS verleih, COALESCE(z.stufe, 'unbekannt') AS zustand, e.verleihstatus FROM buch_exemplar e JOIN buch_werk w ON w.werk_id = e.werk_id JOIN verleih v ON v.verleiher_id = e.verleiher_id JOIN benutzer b ON b.benutzer_id = v.verleiher_id LEFT JOIN zustand z ON z.zustand_id = e.zustand_id LEFT JOIN abholort ab ON ab.abholort_id = e.abholort_id ORDER BY w.titel, e.exemplar_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

exemplar_id	titel	verleiher	zustand	verleihstatus
1	Buddenbrooks	anna.k	neu	verfügbar
13	Buddenbrooks	timo.b	ordentlich	verfügbar
5	Das Parfum	mia.f	ordentlich	reserviert
17	Das Parfum	felix.w	gut	verfügbar
2	Der Prozess	max.m	gut	verfügbar
14	Der Prozess	lena.h	neu	ausgeliehen
3	Der Steppenwolf	timo.b	sehr gut	ausgeliehen
15	Der Steppenwolf	mia.f	gebraucht	verfügbar
4	Die Blechtrommel	lena.h	gebraucht	verfügbar
16	Die Blechtrommel	jonas.r	sehr gut	reserviert
11	Die Physiker	max.m	sammelwürdig	verfügbar
8	Die Vermessung der Welt	mod.1	mit Notizen	verfügbar
20	Die Vermessung der Welt	luis.g	sehr gut	verfügbar
7	Effi Briest	felix.w	abgenutzt	verfügbar
19	Effi Briest	eva.l	mit Notizen	verfügbar
10	Nachtzug nach Lissabon	luis.g	gut	verfügbar
12	Tintenherz	lena.h	sehr gut	verfügbar
6	Tyll	jonas.r	sehr gut	verfügbar
18	Tyll	mod.1	abgenutzt	verfügbar
9	Unterleuten	eva.l	bibliotheksex	ausgeliehen

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ | Zeilen filtern: Diese Tabelle durchsuchen | Nach Schlüssel sortieren: keine ▼

Tabelle: ausleihe – SQL Statements

```
CREATE TABLE ausleihe (
```

```
    ausleihe_id INT AUTO_INCREMENT PRIMARY KEY,  
    exemplar_id INT NOT NULL,  
    verleih_id INT NOT NULL,  
    ausleiher_id INT NOT NULL,  
    startdatum DATE,  
    enddatum DATE,  
    status VARCHAR(20),          -- z. B. 'aktiv', 'abgeschlossen'  
    CONSTRAINT fk_ausleihe_exemplar  
        FOREIGN KEY (exemplar_id)  
        REFERENCES buch_exemplar(exemplar_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_ausleihe_verleiher  
        FOREIGN KEY (verleiher_id)  
        REFERENCES verleih(verleiher_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_ausleihe_ausleiher  
        FOREIGN KEY (ausleiher_id)  
        REFERENCES ausleiher(ausleiher_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO ausleihe (exemplar_id, verleih_id, ausleiher_id, startdatum, enddatum, status) VALUES  
(1, 1, 3, '2025-07-01', '2025-07-20', 'abgeschlossen'),  
(2, 2, 4, '2025-07-10', '2025-07-24', 'abgeschlossen'),  
(3, 4, 6, '2025-08-01', '2025-08-22', 'aktiv'),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```


Tabelle: ausleihe – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0005 Sekunden.) [ausleihe_id: 1... - 12...]

```
SELECT ausleihe_id, titel, verleih_name, ausleiher_name, status FROM v_ausleihe_detail WHERE status IN ('aktiv','abgeschlossen') ORDER BY ausleihe_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: v Zeilen filtern:

Zusätzliche Optionen

ausleihe_id	titel	verleiher_name	ausleiher_name	status
1	Buddenbrooks	anna.k	sara.s	abgeschlossen
2	Der Prozess	max.m	timo.b	abgeschlossen
3	Der Steppenwolf	timo.b	paul.w	aktiv
4	Die Blechtrommel	lena.h	mia.f	abgeschlossen
5	Das Parfum	mia.f	jonas.r	aktiv
6	Tyll	jonas.r	klara.s	aktiv
7	Effi Briest	felix.w	admin.1	abgeschlossen
8	Die Vermessung der Welt	mod.1	mod.1	aktiv
9	Unterleuten	eva.l	nils.k	abgeschlossen
10	Nachtzug nach Lissabon	luis.g	ole.p	abgeschlossen
11	Die Physiker	max.m	nina.b	aktiv
12	Tintenherz	lena.h	yara.o	aktiv

☐ Alles anzeigen | Anzahl der Datensätze: v Zeilen filtern:

Tabelle: reservierung – SQL Statements

```
CREATE TABLE reservierung (
```

```
    reservierung_id INT AUTO_INCREMENT PRIMARY KEY,  
    exemplar_id INT NOT NULL,  
    ausleiher_id INT NOT NULL,  
    von_datum DATE,  
    bis_datum DATE,  
    status VARCHAR(20),      -- z. B. 'aktiv', 'storniert', 'abgelaufen'  
    CONSTRAINT fk_reserv_exemplar  
        FOREIGN KEY (exemplar_id)  
        REFERENCES buch_exemplar(exemplar_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_reserv_ausleiher  
        FOREIGN KEY (ausleiher_id)  
        REFERENCES ausleiher(ausleiher_id)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO reservierung (exemplar_id, ausleiher_id, von_datum, bis_datum, status) VALUES  
(1, 19, '2025-09-10', '2025-09-20', 'aktiv'),  
(2, 20, '2025-09-05', '2025-09-12', 'aktiv'),  
(4, 3, '2025-09-15', '2025-09-25', 'aktiv'),  
... ;
```

-- Testfall siehe Screenshot nächste Folie --

Tabelle: reservierung – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [von_datum: 2025-09-01... - 2025-09-25...] [titel: TYLL... - TINTENHERZ...]

```
SELECT r.reservierung_id, w.titel, bu.benutzername AS ausleiher, r.von_datum, r.bis_datum, r.status FROM reservierung r JOIN buch_exemplar e ON e.exemplar_id = r.exemplar_id JOIN buch_werk w ON w.werk_id = e.werk_id JOIN ausleiher au ON au.ausleiher_id = r.ausleiher_id JOIN benutzer bu ON bu.benutzer_id = au.ausleiher_id WHERE r.von_datum >= '2025-09-01' ORDER BY r.von_datum, w.titel;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern:

[Zusätzliche Optionen](#)

reservierung_id	titel	ausleiher	von_datum <small>▲ 1</small>	bis_datum	status
5	Tyll	paul.w	2025-09-01	2025-09-08	aktiv
4	Das Parfum	timo.b	2025-09-02	2025-09-09	storniert
8	Unterleuten	klara.s	2025-09-03	2025-09-13	abgelaufen
2	Der Prozess	zoe.h	2025-09-05	2025-09-12	aktiv
12	Der Steppenwolf	ole.p	2025-09-06	2025-09-16	aktiv
7	Die Vermessung der Welt	jonas.r	2025-09-07	2025-09-14	aktiv
1	Buddenbrooks	tim.s	2025-09-10	2025-09-20	aktiv
9	Nachtzug nach Lissabon	felix.w	2025-09-11	2025-09-18	aktiv
3	Die Blechtrommel	sara.s	2025-09-15	2025-09-25	aktiv
6	Effi Briest	mia.f	2025-09-18	2025-09-28	aktiv
10	Die Physiker	admin.1	2025-09-20	2025-09-30	aktiv
11	Tintenherz	mod.1	2025-09-25	2025-10-02	aktiv

☐ Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern:

Tabelle: bewertung – SQL Statements

```
CREATE TABLE bewertung (
```

```
    bewertung_id INT AUTO_INCREMENT PRIMARY KEY,  
    benutzer_id INT NOT NULL,  
    werk_id INT NOT NULL,  
    sterne INT,          -- 1–5  
    kommentar VARCHAR(255),  
    erstellt_am TIMESTAMP,  
    CONSTRAINT fk_bewertung_benutzer  
        FOREIGN KEY (benutzer_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_bewertung_werk  
        FOREIGN KEY (werk_id)  
        REFERENCES buch_werk(werk_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT uq_bewertung_einmal UNIQUE (benutzer_id, werk_id),  
    CONSTRAINT chk_bewertung_sterne CHECK (sterne BETWEEN 1 AND 5)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO bewertung (benutzer_id, werk_id, sterne, kommentar, erstellt_am) VALUES  
(1, 1, 5, 'Großer Klassiker.', '2025-08-01 10:00:00'),  
(3, 2, 4, 'Fesselnd und klug.', '2025-08-02 11:20:00'),  
(4, 3, 3, 'Teilweise sperrig.', '2025-08-03 09:15:00'),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: bewertung – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [titel: BUDDENBROOKS... - EFFI BRIEST...]

```
SELECT w.titel, ROUND(AVG(bw.sterne),2) AS avg_sterne, COUNT(*) AS anzahl FROM bewertung bw JOIN buch_werk w ON w.werk_id = bw.werk_id GROUP BY w.werk_id, w.titel ORDER BY avg_sterne DESC, anzahl DESC, w.titel;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: Zeilen filtern:

Zusätzliche Optionen

titel	avg_sterne ▾ 1	anzahl ▾ 2
Buddenbrooks	5.00	1
Die Blechtrommel	5.00	1
Die Vermessung der Welt	5.00	1
Nachtzug nach Lissabon	5.00	1
Tintenherz	5.00	1
Das Parfum	4.00	1
Der Prozess	4.00	1
Die Physiker	4.00	1
Tyll	4.00	1
Unterleuten	4.00	1
Der Steppenwolf	3.00	1
Effi Briest	3.00	1

☐ Alles anzeigen | Anzahl der Datensätze: Zeilen filtern:

Tabelle: chat_thread – SQL Statements

```
CREATE TABLE chat_thread (
```

```
    thread_id INT AUTO_INCREMENT PRIMARY KEY,  
    erstellt_am TIMESTAMP,  
    thema VARCHAR(255)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO chat_thread (erstellt_am, thema) VALUES  
( '2025-08-01 09:00:00', 'Abholung Buddenbrooks'),  
( '2025-08-02 10:30:00', 'Reservierung Der Prozess'),  
( '2025-08-03 12:00:00', 'Zustand Die Blechtrommel'),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: chat_thread – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0002 Sekunden.) [erstellt_am: 2025-08-10 15:45:00... - 2025-08-01 09:00:00...]

```
SELECT thread_id, thema, erstellt_am FROM chat_thread ORDER BY erstellt_am DESC LIMIT 10;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

Zusätzliche Optionen

				thread_id	thema	erstellt_am ▾ 1
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	10	Allgemeine Fragen	2025-08-10 15:45:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	9	Chat-Test	2025-08-09 13:05:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	8	Suche nach Klassikern	2025-08-08 11:20:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	7	Bewertung diskutieren	2025-08-07 08:55:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	6	Abholzeit anpassen	2025-08-06 18:10:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	5	Rückgabe Tyll	2025-08-05 16:40:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	4	Versandoptionen	2025-08-04 14:15:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	3	Zustand Die Blechtrommel	2025-08-03 12:00:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	2	Reservierung Der Prozess	2025-08-02 10:30:00
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	1	Abholung Buddenbrooks	2025-08-01 09:00:00

☐ Alle auswählen markierte: Bearbeiten Kopieren Löschen Exportieren

Tabelle: chat_nachricht – SQL Statements

```
CREATE TABLE chat_nachricht (  
  
    nachricht_id INT AUTO_INCREMENT PRIMARY KEY,  
    thread_id INT NOT NULL,  
    sender_id INT NOT NULL,  
    text TEXT NOT NULL,  
    gesendet_am TIMESTAMP,  
    gelesen BOOLEAN DEFAULT FALSE,  
    CONSTRAINT fk_msg_thread  
        FOREIGN KEY (thread_id)  
        REFERENCES chat_thread(thread_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_msg_sender  
        FOREIGN KEY (sender_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
INSERT INTO chat_nachricht (thread_id, sender_id, text, gesendet_am, gelesen) VALUES  
(1, 1, 'Hallo, kann ich heute um 19 Uhr abholen?', '2025-08-01 09:10:00', FALSE),  
(1, 2, 'Ja, passt. Bitte klingeln.', '2025-08-01 09:15:00', TRUE),  
(2, 3, 'Ist die Reservierung noch aktiv?', '2025-08-02 10:40:00', FALSE),  
... ;  
  
-- Testfall siehe Screenshot nächste Folie --
```


Tabelle: chat_nachricht – Testfall Screenshot

✓ Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0,0065 Sekunden.) [thread_id: 1... - 10...] [nachricht_id: 1... - 12...]

```
SELECT t.thread_id, t.thema, n.nachricht_id, u.benutzername AS sender, n.text, n.gelesen FROM chat_thread t LEFT JOIN chat_nachricht n ON n.thread_id = t.thread_id LEFT JOIN benutzer u ON u.benutzer_id = n.sender_id ORDER BY t.thread_id, n.nachricht_id;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Nach Schlüssel sortieren: keine ▼

Zusätzliche Optionen

thread_id	thema	nachricht_id	sender	text	gelesen
1	Abholung Buddenbrooks	1	anna.k	Hallo, kann ich heute um 19 Uhr abholen?	0
1	Abholung Buddenbrooks	2	max.m	Ja, passt. Bitte klingeln.	1
2	Reservierung Der Prozess	3	sara.s	Ist die Reservierung noch aktiv?	0
3	Zustand Die Blechtrommel	4	lena.h	Das Buch ist in gutem Zustand.	1
4	Versandoptionen	5	mia.f	Bietet ihr Versand an?	0
4	Versandoptionen	6	jonas.r	Ja, gegen Portübernahme.	1
5	Rückgabe Tyll	7	felix.w	Ich bringe es morgen zurück.	1
6	Abholzeit anpassen	8	mod.1	Können wir auf 20 Uhr verschieben?	0
7	Bewertung diskutieren	9	klara.s	Warum nur 3 Sterne?	0
8	Suche nach Klassikern	10	admin.1	Suche Empfehlungen für Klassiker.	1
9	Chat-Test	11	nils.k	Testnachricht 1	1
10	Allgemeine Fragen	12	ole.p	Wie funktioniert die App?	1

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▼ Zeilen filtern: Nach Schlüssel sortieren: keine ▼

Tabelle: hilfe_ticket – SQL Statements

```
CREATE TABLE hilfe_ticket (
```

```
    ticket_id INT AUTO_INCREMENT PRIMARY KEY,  
    ersteller_id INT NOT NULL,  
    betreff VARCHAR(255),  
    beschreibung VARCHAR(2000),  
    status VARCHAR(20),          -- 'offen', 'in Bearbeitung', 'geschlossen'  
    erstellt_am TIMESTAMP,  
    geschlossen_am TIMESTAMP,  
    CONSTRAINT fk_ticket_benutzer  
        FOREIGN KEY (ersteller_id)  
        REFERENCES benutzer(benutzer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO hilfe_ticket (ersteller_id, betreff, beschreibung, status, erstellt_am, geschlossen_am) VALUES  
(1, 'Login-Problem', 'Passwort zurücksetzen schlägt fehl.', 'geschlossen', '2025-08-01 08:00:00', '2025-08-01 10:00:00'),  
(3, 'Datenkorrektur', 'Falscher Nachname in meinem Profil.', 'in Bearbeitung', '2025-08-02 09:30:00', NULL),  
(4, 'Spam-Nachrichten', 'Erhalte viele Nachrichten von Unbekannten.', 'offen', '2025-08-03 10:45:00', NULL),  
... ;
```

```
-- Testfall siehe Screenshot nächste Folie --
```

Tabelle: hilfe_ticket – Testfall Screenshot

✓ Zeige Datensätze 0 - 9 (10 insgesamt, Die Abfrage dauerte 0,0003 Sekunden.) [erstellt_am: 2025-08-10 17:45:00... - 2025-08-01 08:00:00...]

```
SELECT h.ticket_id, u.benutzername, h.betreff, h.status, h.erstellt_am FROM hilfe_ticket h JOIN benutzer u ON u.benutzer_id = h.ersteller_id WHERE h.status IN ('offen','in Bearbeitung','geschlossen') ORDER BY FIELD(h.status,'offen','in Bearbeitung','geschlossen'), h.erstellt_am DESC;
```

☐ Messen [[Inline bearbeiten](#)] [[Bearbeiten](#)] [[SQL erklären](#)] [[PHP-Code erzeugen](#)] [[Aktualisieren](#)]

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▾ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▾

Zusätzliche Optionen

ticket_id	benutzername	betreff	status	erstellt_am
10	ole.p	Allgemeine Frage	offen	2025-08-10 17:45:00
5	mia.f	Ausleihe-Fehler	offen	2025-08-05 12:10:00
3	timo.b	Spam-Nachrichten	offen	2025-08-03 10:45:00
8	admin.1	Chat lädt nicht	in Bearbeitung	2025-08-08 15:25:00
6	jonas.r	Adresse ändern	in Bearbeitung	2025-08-06 13:55:00
2	sara.s	Datenkorrektur	in Bearbeitung	2025-08-02 09:30:00
9	mod.1	Rolle anpassen	geschlossen	2025-08-09 16:05:00
7	felix.w	Bewertung löschen	geschlossen	2025-08-07 14:40:00
4	paul.w	Reservierung hängt	geschlossen	2025-08-04 11:20:00
1	anna.k	Login-Problem	geschlossen	2025-08-01 08:00:00

☐ Alles anzeigen | Anzahl der Datensätze: 25 ▾ Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine ▾