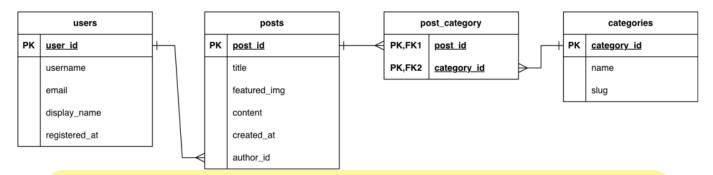
# LU08c: Tabellen mit Fremdschlüssel erstellen

**Ziel:** Wir verteilen die Daten wie bei WordPress auf mehrere Tabellen (users, posts, categories und die N:M-Zuordnung post\_category) und setzen die Fremdschlüssel direkt beim Erstellen.

### **ERD** (Überblick)

Wir gehen vom Schema aus dem Reiseblog-Beispiel aus:





Posts können mehreren Kategorien angehören (N:M). Die saubere Lösung ist eine Zwischentabelle post\_category. Das bauen wir später (s. LU08e: N:M-Beziehungen mit Zwischentabelle abbilden.

## Fremdschlüssel: Grundsyntax

**Wichtig:** Die **referenzierte Tabelle** (parent\_table) muss **bereits existieren**, sonst meldet MySQL/MariaDB einen Fehler.

# **Beispiel Reiseblog**

### 1. Tabellen anlegen

```
Tabelle posts
CREATE TABLE posts (
  post id
              INT AUTO INCREMENT PRIMARY KEY,
  title
                      VARCHAR(200) NOT NULL,
  featured_img
                     VARCHAR(512),
                      TEXT,
  content
                      DATETIME NOT NULL DEFAULT
  created at
CURRENT TIMESTAMP,
                      INT
  author id
                                  NOT NULL,
  FOREIGN KEY (author id) REFERENCES users (user id) --
Foreign Key wird hier angelegt!
);
```

### 2. Beispieldaten einfügen

Für ausführliche Erklärung zum Einfügen von Daten in Tabellen via SQL schauen Sie in "LU07 - DML: Daten einfügen, ändern und löschen" nach.

https://wiki.bzz.ch/ Printed on 2025/11/27 09:41

```
-- 1) Autor:innen
INSERT INTO users (username, email, display name) VALUES
  ('caro', 'caro@wetraveltheworld.de', 'Caro Steig'),
  ('martin', 'martin@wetraveltheworld.de',
                                            'Martin
Merten'),
  ('shaolin', 'shaolin@wetraveltheworld.de', 'Shaolin
Tran');
-- 2) Kategorien
INSERT INTO categories (name, slug) VALUES
('Städtereise', 'staedtereise'),
('Abenteuer', 'abenteuer'),
('Roadtrip', 'roadtrip'),
('Belgien', 'belgien'),
('Niederlande', 'niederlande'),
('Portugal', 'portugal'),
('Deutschland', 'deutschland'),
('Montenegro', 'montenegro'),
('Oman', 'oman'),
('USA', 'usa');
-- 3) Posts
INSERT INTO posts (title, featured img, content, created at,
author id) VALUES
('Hasselt - 10 Highlights',
'https://wetraveltheworld.de/wp-content/uploads/2025/05/hass
elt.jpg',
'Kurzguide: Die schönsten Ecken von Hasselt ...',
'2025-05-07 10:15:00', 2), -- Martin
('Utrecht - 10 Sehenswürdigkeiten',
'https://wetraveltheworld.de/wp-content/uploads/2025/06/utre
cht.jpg',
'Cafés, Grachten und Restaurant-Tipps ...',
'2025-06-05 09:30:00', 2), -- Martin
('Lissabon — 8 Tipps zu den wichtigsten Sehenswürdigkeiten',
'https://wetraveltheworld.de/wp-content/uploads/2025/03/liss
abon.ipg',
'Aussichtspunkte, Viertel und Highlights ...',
'2025-03-21 08:40:00', 1), -- Caro
('Maastricht an einem Tag',
'https://wetraveltheworld.de/wp-content/uploads/2025/06/maas
tricht.jpg',
'Spaziergang, Restaurants, Altstadt ...',
'2025-06-12 11:05:00', 1), -- Caro
('Montenegro Roadtrip — 10 Highlights',
```

```
'https://wetraveltheworld.de/wp-content/uploads/2025/06/mont enegro.jpg',
'Bucht von Kotor, Durmitor, Tara-Schlucht ...',
'2025-06-11 14:22:00', 1), -- Caro

('Oman — Top 22 Highlights',
'https://wetraveltheworld.de/wp-content/uploads/2025/06/oman.jpg',
'Nizwa, Wadis, Wüste und Roadtrip-Tipps ...',
'2025-06-11 09:00:00', 1), -- Caro

('Chicago in 3 Tagen — 17 Highlights',
'https://wetraveltheworld.de/wp-content/uploads/2024/10/chicago.jpg',
'Riverwalk, The Bean, Museen und Skyline ...',
'2024-10-07 07:50:00', 2); -- Martin
```

#### Resultate (nach dem Einfügen):

#### **Verknüpfung Tabelle users & posts (one-to-many)**



Tabelle categories (wird später mit 'posts' verknüpft)

https://wiki.bzz.ch/ Printed on 2025/11/27 09:41

category_id	name	slug
1	Städtereise	staedtereise
2	Abenteuer	abenteuer
3	Roadtrip	roadtrip
4	Belgien	belgien
5	Niederlande	niederlande
6	Portugal	portugal
7	Deutschland	deutschland
8	Montenegro	montenegro
9	Oman	oman
10	USA	usa

### 3. Fremdschlüssel in Aktion (Standard: RESTRICT)

Beim setzen von Fremdschlüsseln macht MySQL etwas im Hintergrund: es sichert die zwei verbundenen Tabellen bzw. die Datensätzen deren gegen änderungen und Löschungen ab. Das nennt man Referenzaktion und standartmässig setzt hier MySQL Restrict ein. Was heisst das für unser Reiseblog-Beispiel. Wir haben zwei Tabellen users und posts. Der Fremdschlüssel author\_id wurde bei posts gesetzt und verweist auf die user\_id in users. Das heisst die Eltern-Tabelle ist hier users während die Kind-Tabelle posts nun abhängig ist von dessen Eltern. Die Elterntabelle schützt nun MySQL gegen gewisse Änderungen und Löschungen: Löschen/Ändern der Elternzeile ist nur erlaubt, wenn keine Kindzeilen darauf verweisen.

```
-- Sicherheit: Welche FKs sind gesetzt?
SHOW CREATE TABLE posts;
```

Demo 1 - User ohne Posts löschen (erlaubt):

```
DELETE FROM users
WHERE username = 'shaolin'; -- hat keine Posts zugewiesen
SELECT user_id, username FROM users;
```

Erwartung: Die Zeile wird gelöscht (keine Posts verweisen auf den User).

Demo 2 - User mit Posts löschen (blockiert):

```
DELETE FROM users
WHERE username = 'martin';
```

Erwartete Fehlermeldung (sinngemäss):



[23000][1451] Cannot delete or update a parent row: a foreign key constraint fails (travel\_blog.posts, CONSTRAINT posts\_ibfk\_1 FOREIGN KEY (author\_id) REFERENCES users(user id))

Grund: In posts.author\_id gibt es Kindzeilen (z.B. "Hasselt – 10 Highlights"), die auf users.user\_id von martin verweisen → RESTRICT verhindert das Löschen.

Demo 3 - Unkritisches Attribut ändern (erlaubt):

```
UPDATE users
SET username = 'martin_travels'
WHERE user_id = 2; -- OK: FKs verweisen auf user_id, nicht
auf username
```

Demo 4 - Primärschlüssel ändern (blockiert, RESTRICT):

```
UPDATE users
SET user_id = 5
WHERE user_id = 2; -- erwartet: Fehler (RESTRICT), da
posts.author_id -> users.user_id
```

Erklärung: posts.author\_id → users.user\_id ist ein FK mit Standard ON UPDATE RESTRICT. Solange Posts existieren, darf der user id-Wert nicht verändert werden.

Demo 5 - Kategorie-ID ändern: Was passiert?

```
UPDATE categories
SET category_id = 11
```

https://wiki.bzz.ch/ Printed on 2025/11/27 09:41

WHERE category\_id = 10; -- USA  $\rightarrow$  11: funktioniert - Primary Keys dürfen geändert werden.

#### Warum ist das so?

RESTRICT (Standard) schützt die Elternzeile (z. B. users): Solange Kindzeilen (z. B. posts) auf sie zeigen, sind DELETE/UPDATE am referenzierten Primärschlüssel blockiert.

Änderungen an nicht referenzierten Spalten (z. B. users.username) sind frei möglich – FKs verweisen hier nicht darauf.

Ob eine Änderung blockiert oder mitgezogen (CASCADE) wird, hängt von der ON DELETE/ON UPDATE-Einstellung im FK ab.

Merke: Fremdschlüssel geben dir Datensicherheit:



- Verhindern verwaiste Daten (z. B. Posts ohne gültigen Autor),
- definieren klares Verhalten bei Löschen/Ändern (RESTRICT, CASCADE, SET NULL),
- halten die Datenbank konsistent.

Ausblick: In **LU08d** fügen wir FKs per **ALTER TABLE** nachträglich hinzu und testen die Referenzaktionen **RESTRICT**, **CASCADE** und **SET NULL** gezielt.

From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

 $https://wiki.bzz.ch/modul/m290\_guko/learningunits/lu08/theorie/c\_fk-create-table?rev=1760903468$ 

Last update: 2025/10/19 21:51

