# LU08d - FKs per ALTER TABLE + Referenzaktionen

**Ziel:** Bereits bestehende Tabellen mit **ALTER TABLE** um Fremdschlüssel erweitern und verstehen, was **RESTRICT**, **CASCADE** und **SET NULL** bewirken.

Voraussetzung: Du verwendest die DB aus **LU08c** (Tabellen **users**, **posts**, **categories**, **post\_category** mit Beispieldaten).

# 0) ALTER TABLE - Überblick (Syntax)

Mit ALTER TABLE können wir bestehende Tabellen ändern.

```
Spalte hinzufügen
ALTER TABLE TABLE NAME
  ADD COLUMN neue_spalte DATENTYP [NULL|NOT NULL] [DEFAULT
...] [AFTER bestehende spalte];
Spalte umbenennen
ALTER TABLE TABLE NAME
  RENAME COLUMN altname TO neuename;
Datentyp ändern
ALTER TABLE TABLE_NAME
  MODIFY COLUMN spalte NEUER DATENTYP [NULL|NOT NULL]
[DEFAULT ...];
Spalte löschen
ALTER TABLE TABLE NAME
  DROP COLUMN spalte;
Fremdschlüssel hinzufügen
ALTER TABLE child_table
  ADD CONSTRAINT fk name
    FOREIGN KEY (fk spalte)
    REFERENCES parent_table(parent_pk)
    [ON DELETE {RESTRICT|CASCADE|SET NULL}]
    [ON UPDATE {RESTRICT|CASCADE|SET NULL}];
```

# ALTER TABLE TABLE\_NAME DROP FOREIGN KEY fk\_name; -- Namen via

-- herausfinden.

SHOW CREATE TABLE TABLE\_NAME;

# 1) Optionalen Verweis in POSTS ergänzen (featured category id)

Wir ergänzen eine optionale Hauptkategorie im **posts**-Tabelle und verknüpfen sie mit **categories**.

```
1.1 Spalte ergänzen und Beispielwerte setzen

-- Spalte hinzufügen (optional, falls noch nicht vorhanden)
ALTER TABLE posts
   ADD COLUMN featured_category_id INT NULL AFTER content;

-- Beispielwerte passend zu LU08c-Daten
UPDATE posts SET featured_category_id = 1 WHERE post_id = 1;
-- Schweiz
UPDATE posts SET featured_category_id = 2 WHERE post_id = 2;
-- Städtereise
UPDATE posts SET featured_category_id = 3 WHERE post_id = 3;
-- Abenteuer
```

#### 1.2 Fremdschlüssel setzen (SET NULL beim Löschen)

```
ALTER TABLE posts

ADD CONSTRAINT fk_posts_featured_category

FOREIGN KEY (featured_category_id)

REFERENCES categories (category_id)

ON DELETE SET NULL

-- Kategorie gelöscht → Post

bleibt, Verweis wird NULL

ON UPDATE RESTRICT;

-- Kategorie-ID darf nicht

"umgehängt" werden
```

https://wiki.bzz.ch/ Printed on 2025/11/27 11:03



Warum SET NULL? Die Hauptkategorie ist optional. Wird eine Kategorie gelöscht, soll der Post nicht verschwinden – nur der Verweis fällt auf NULL.

## 2) N:M-Zuordnung absichern (post\_category mit CASCADE)

Die Zuordnungstabelle **post\_category (post\_id, category\_id)** soll sich **automatisch aufräumen**, wenn ein Post oder eine Kategorie gelöscht wird.

```
2.1 FKs mit CASCADE hinzufügen (oder neu setzen)
-- Falls die FKs noch NICHT existieren:
ALTER TABLE post category
  ADD CONSTRAINT fk_pc_post
    FOREIGN KEY (post id)
    REFERENCES posts (post id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  ADD CONSTRAINT fk pc category
    FOREIGN KEY (category_id)
    REFERENCES categories (category id)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
-- Falls bereits FKs ohne CASCADE existieren:
-- 1) Namen ermitteln
SHOW CREATE TABLE post category;
-- 2) Alte FKs löschen
ALTER TABLE post_category
  DROP FOREIGN KEY <alter fk post>,
  DROP FOREIGN KEY <alter fk category>;
-- 3) Obige ALTER-Anweisung erneut ausführen (mit CASCADE)
```

### 3) Referenzaktionen - kurz & praxisnah

**RESTRICT** – schützt Elternzeilen Eltern (z. B. **users**) können nicht gelöscht/aktualisiert werden, solange Kinder (z. B. **posts**) auf sie verweisen.

Demo (soll fehlschlagen):

```
DELETE FROM users WHERE username = 'caro'; -- erwartet:
Fehler (RESTRICT)
```

**CASCADE** – räumt automatisch auf Kindzeilen in **post\_category** werden mitgelöscht, wenn der zugehörige Post/Kategorie verschwindet.

#### Demo:

```
-- Post #2 löschen → zugehörige Mappings verschwinden
automatisch
DELETE FROM posts WHERE post_id = 2;
-- Prüfung:
SELECT * FROM post_category WHERE post_id = 2; -- → keine
Zeilen
```

**SET NULL** – Verweis wird auf NULL gesetzt Kindzeile (hier: **posts**) bleibt erhalten, der optionale Verweis wird NULL.

#### Demo:

```
-- Kategorie 'staedtereise' löschen (id=2)
DELETE FROM categories WHERE slug = 'staedtereise';
-- featured_category_id wird NULL, Post bleibt bestehen
SELECT post_id, title, featured_category_id
FROM posts
ORDER BY post_id;
```

#### Beispielresultat (Ausschnitt nach den Demos):

| post_id    | title  | featured_category_id |
|------------|--|----------------------|
| 1          | Die Schweiz ist zu teuer? $\dots$ Davos & Klosters im Winter | 1                    |
| <b>∣</b> ⊀ | Usbekistan Rundreise Highlights - Die 12<br>besten           | 3                    |

Hinweis: Durch das Löschen von Post #2 (CASCADE) und Kategorie "Städtereise" (SET NULL) sind die entsprechenden Verknüpfungen bereinigt bzw. auf NULL gesetzt.

https://wiki.bzz.ch/ Printed on 2025/11/27 11:03

## 4) Merksätze

\* **RESTRICT**: Löschen/Ändern der Elternzeile nur erlaubt, wenn **keine** Kindzeilen existieren. \* **CASCADE**: Löscht/aktualisiert zugehörige Kindzeilen **automatisch** (ideal für N:M-Zuordnungen). \* **SET NULL**: Kindzeile bleibt erhalten, FK-Spalte wird **NULL** (ideal für **optionale** Verweise).



Reihenfolge beim Nachrüsten von FKs: 1) Daten prüfen/bereinigen (keine verwaisten Verweise). 2) ALTER TABLE mit passenden Referenzaktionen setzen. 3) Verhalten mit kleinen Test-Löschungen/-Änderungen nachvollziehen.



**Ausblick (LU09):** Mehrtabellen-Abfragen mit **JOINs** und **Subqueries** auf unserem Reiseblog-Schema.

From:

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

Permanent link:

 $https://wiki.bzz.ch/modul/m290\_guko/learningunits/lu08/theorie/d\_fk-alter-table?rev=1760657180$ 

Last update: 2025/10/17 01:26

