

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 TABLE_NAME
  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}];
```

Fremdschlüssel/Constraint löschen

```
ALTER TABLE TABLE_NAME
  DROP FOREIGN KEY fk_name;
-- Namen via
SHOW CREATE TABLE TABLE_NAME;
-- herausfinden.
```

1) Optionalen Verweis in POSTS ergänzen

Wir ergänzen in **posts** eine **optionale** verantwortliche Redaktor:in (**editor_id**) und verknüpfen sie mit **users**.

1.1 Spalte ergänzen und Beispielwerte setzen

```
-- Spalte hinzufügen
ALTER TABLE posts
  ADD COLUMN editor_id INT NULL AFTER author_id;

-- Beispielwerte passend zu LU08c-Daten (user_id: 1=caro,
2=martin, 3=shaolin)
UPDATE posts SET editor_id = 2 WHERE post_id = 1; -- Post
#1: Editor = martin
UPDATE posts SET editor_id = 1 WHERE post_id = 2; -- Post
#2: Editor = caro
UPDATE posts SET editor_id = NULL WHERE post_id = 3; --
Post #3: (noch) kein Editor
```

1.2 Fremdschlüssel setzen (SET NULL beim Löschen)

```
ALTER TABLE posts
  ADD CONSTRAINT fk_posts_editor
    FOREIGN KEY (editor_id)
    REFERENCES users (user_id)
    ON DELETE SET NULL -- Editor gelöscht → Post bleibt,
Verweis wird NULL
    ON UPDATE RESTRICT; -- user_id soll nicht geändert
werden können (User-PK bleibt stabil; man ändert den Editor
immer in posts)
```



Warum SET NULL? Die Redaktor:in ist **optional**. Wird ein User gelöscht, soll der Post **nicht** mitgelöscht werden – der optionale Verweis fällt einfach 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? ... Davos & Klosters im Winter	1
3	Usbekistan Rundreise Highlights - Die 12 besten ...	3

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

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=1760657679

Last update: **2025/10/17 01:34**

