LU08e: N:M-Beziehungen mit Zwischentabelle abbilden

Ziel: Sie modellieren eine **Many-to-Many**-Beziehung (Posts ↔ Kategorien) korrekt mit einer **Zwischentabelle** post category und füllen Daten-Zuordnungen.

Voraussetzung: Die Tabellen und Beispieldaten aus **LU08c** (*users, posts, categories*) sind vorhanden. (Die Verknüpfung **users** ↔ **posts** ist 1:N; **posts** ↔ **categories** ist N:M.)

Warum N:M auflösen?

In einer relationalen Tabelle kann ein Fremdschlüssel **immer nur auf genau eine** Zielzeile verweisen. Bei **N:M** gilt jedoch: **ein Post → mehrere Kategorien** und **eine Kategorie → mehrere Posts**. Das lässt sich mit **einem** FK-Feld nicht sauber abbilden:

* FK in posts nur als category_id: Pro Post wäre nur eine Kategorie möglich.

Mehrfachzuordnungen erzwingen Duplikate desselben Posts → Änderungen/Löschungen an mehreren Stellen, fehleranfällig. * FK in categories nur als post_id: Pro Kategorie wäre nur ein Post möglich. Für mehrere Posts müsste man Kategorien duplizieren → Eindeutigkeit geht verloren. * Kommaliste in einer Spalte (z. B. Belgien, Städtereise): In einer Zelle stehen mehrere Werte. Die DB kann nicht prüfen, ob diese Kategorien existieren (kein FK); Abfragen werden unzuverlässig und langsam. * Mehrere FK-Spalten (z. B. category_id1, category_id2 ...): Die Anzahl möglicher Zuordnungen ist künstlich begrenzt; das Schema müsste bei neuen Fällen ständig erweitert werden.

Die folgenden Tabellen-Skizzen visualisieren diese Ansätze – **alles sind schlechte Lösungen** (siehe oben):

1./2. Duplizierte Zeilen

um gleichen Post zwei Kategorien zuweisen zu können

post_id	title	featured_img	content	created_at	category_id
1	Hasselt – 10 Highlights	https://wetraveltheworld.de/wp- content/uploads/2025/05/hasselt.jpg	Kurzguide: Die schönsten Ecken von Hasselt	2025-05-07 10:15:00	1
2	Hasselt – 10 Highlights	https://wetraveltheworld.de/wp- content/uploads/2025/05/hasselt.jpg	Kurzguide: Die schönsten Ecken von Hasselt	2025-05-07 10:15:00	4
3	Utrecht – 10 Sehenswürdigkeiten	https://wetraveltheworld.de/wp- content/uploads/2025/06/utrecht.jpg	Cafés, Grachten und Restaurant-Tipps	2025-06-05 09:30:00	1
4	Utrecht – 10 Sehenswürdigkeiten	https://wetraveitheworld.de/wp- content/uploads/2025/06/utrecht.jpg	Cafés, Grachten und 2025-06-05 Restaurant-Tipps 09:30:00		5

3. Kommaliste als FK

um gleichen Post zwei Kategorien zuweisen zu können

post_id	title	featured_img	content	created_at	category_id
1	Hasselt – 10 Highlights	https://wetraveltheworld.de/wp- content/uploads/2025/05/hasselt.jpg	Kurzguide: Die schönsten Ecken von Hasselt	2025-05-07 10:15:00	1, 4
2	Utrecht – 10 Sehenswürdigkeiten	https://wetraveltheworld.de/wp- content/uploads/2025/06/utrecht.jpg	Cafés, Grachten und Restaurant-Tipps	2025-06-05 09:30:00	1, 5
3	Lissabon – 8 Tipps zu den wichtigsten Sehenswürdigkeiten	https://wetraveitheworld.de/wp- content/uploads/2025/03/lissabon.jpg	Aussichtspunkte, Viertel und Highlights	2025-03-21 08:40:00	1, 6

4. mehrere Fremdschlüssel-Spalten

um gleichen Post zwei Kategorien zuweisen zu können

post_id	title	featured_img	content	created_at	category_id1	category_id2
1	Hasselt – 10 Highlights	https://wetraveitheworld.de/wp- content/uploads/2025/05/hasseit.jpg	Kurzguide: Die schönsten Ecken von Hasselt	2025-05-07 10:15:00	1	4
2	Utrecht – 10 Sehenswürdigkeiten	https://wetraveitheworld.de/wp- content/uploads/2025/06/utrecht.jpg	Cafés, Grachten und Restaurant-Tipps	2025-06-05 09:30:00	1	5
1	Lissabon – 8 Tipps zu den wichtigsten Sehenswürdigkeiten	https://wetraveitheworld.de/wp- content/uploads/2025/03/lissabon.jpg	Aussichtspunkte, Viertel und Highlights	2025-03-21 08:40:00	1	6

Lösung: die Zwischentabelle post_category. Wir trennen die Objekte (posts, categories) und speichern **jede einzelne Zuordnung** als eigene Zeile in post_category – genau **ein Paar** (post_id, category_id) pro Zeile. So bleibt jede Beziehung **eindeutig** modelliert, Primärschlüssel bleiben eindeutig, Fremdschlüssel sind **prüfbar**, und Abfragen bleiben **klar** und **performant**.

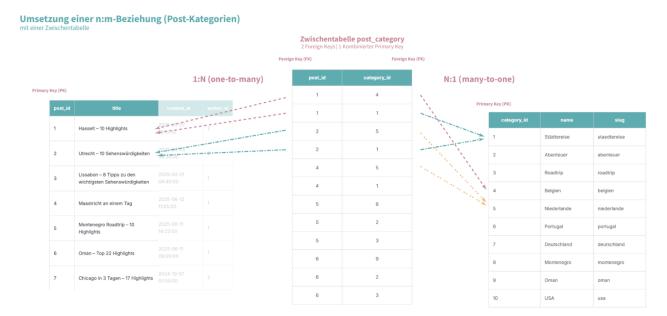
1:N vs N:M mit Zwischentabelle

users ↔ posts (1:N)

https://wiki.bzz.ch/ Printed on 2025/12/10 12:10



posts ↔ categories (N:M) - korrekt über Zwischentabelle post_category



1) Zwischentabelle anlegen (mit Fremdschlüsseln)

Eine Zeile = **genau eine Zuordnung** (post_id, category_id). **CASCADE** sorgt dafür, dass Zuordnungen automatisch gelöscht werden, wenn ein Post oder eine Kategorie gelöscht wird.

2) N:M-Zuordnungen einfügen

(IDs gemäss **LU08c**)

```
INSERT INTO post_category (post_id, category_id) VALUES
(1, 4), -- Hasselt → Belgien
(1, 1), -- Hasselt → Städtereise
(2, 5), -- Utrecht → Niederlande
(2, 1), -- Utrecht → Städtereise
(3, 6), -- Lissabon → Portugal
(3, 1), -- Lissabon → Städtereise
(4, 5), -- Maastricht → Niederlande
(4, 1), -- Maastricht → Städtereise
(5, 8), -- Montenegro Roadtrip → Montenegro
(5, 2), -- Montenegro Roadtrip → Abenteuer
(5, 3), -- Montenegro Roadtrip → Roadtrip
(6, 9), -- Oman \rightarrow Oman
(6, 2), -- Oman → Abenteuer
(6, 3), -- Oman → Roadtrip
(7, 10), -- Chicago → USA
(7, 1); -- Chicago → Städtereise
```

3) Praxis: CASCADE

Beim Löschen eines Posts (oder einer Kategorie) werden die **Zuordnungen in post_category automatisch entfernt**.

```
-- Demo: Post #2 löschen

DELETE FROM posts WHERE post_id = 2;

-- Kontrolle: Mappings zu Post #2 sind weg

SELECT * FROM post_category WHERE post_id = 2;
```

Erwartung: Die letzte Abfrage liefert **keine Zeilen** (CASCADE hat die Zuordnungen bereinigt).

https://wiki.bzz.ch/ Printed on 2025/12/10 12:10

4) Aufgaben

- Aufgabe A: Fügen Sie eine neue Kategorie `Benelux` (slug: benelux) ein und ordnen Sie
 Hasselt 10 Highlights zusätzlich dieser Kategorie zu. Prüfen Sie danach, ob drei Kategorien für
 den Post angezeigt werden.
- **Aufgabe B:** Löschen Sie die Kategorie *roadtrip*. Prüfen Sie, dass die Posts weiterhin existieren und nur die entsprechenden *post category-*Zeilen verschwunden sind (CASCADE).

Zusammenfassung

- N:M wird immer über eine Zwischentabelle mit einer Paarung pro Zeile modelliert: (post_id, category_id).
- **Fremdschlüssel** sichern die Datenqualität; mit **ON DELETE CASCADE** werden Zuordnungen automatisch aufgeräumt.



Ausblick auf LU09: Wir formulieren dieselben Abfragen für mehrere Tabellen

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m290_guko/learningunits/lu08/theorie/e_many-to-many?rev=1760907676

Last update: 2025/10/19 23:01

