

LU08a - Tabellen in Beziehung setzen - Fremdschlüssel & Referenzaktionen

Lernziel: Du kannst Fremdschlüssel (FOREIGN KEY) definieren, referentielle Integrität erklären und die Referenzaktionen (**RESTRICT/NO ACTION**, **CASCADE**, **SET NULL**) anhand von Tests unterscheiden.

Zeitbedarf: 2×45 Minuten

Voraussetzungen: CREATE TABLE (PK, NOT NULL, AUTO_INCREMENT), INSERT/UPDATE/DELETE, SELECT einer einzelnen Tabelle.

Inhalt:

- 1) Fremdschlüssel & referentielle Integrität
- 2) Fremdschlüssel direkt beim Erstellen (CREATE TABLE)
- 3) Fremdschlüssel nachträglich hinzufügen (ALTER TABLE)
- 4) Referenzaktionen testen (RESTRICT, CASCADE, SET NULL)
- 5) Aufträge & Checkliste

ERD (Mini-Filmdatenbank)

Wir bleiben beim bekannten Film-Beispiel und ergänzen es um Personen und Besetzungen:

```
[person] (person_id PK, name UNIQUE, ...)
  1      \
          \      (M:N via film_cast)
            >-----[film_cast] (film_id PK/FK, person_id PK/FK, role)
            /
[film] (film_id PK, title, released_year, director_id FK ->
person.person_id)
```

Regeln:

- Ein Film kann 0..n Cast-Mitglieder haben (film_cast).
- Eine Person kann 0..n Rollen gespielt haben (film_cast).
- Ein Film hat optional eine/n Regisseur:in (director_id), darf aber auch ohne gesetzt sein.

Hinweis: In modernen MySQL-Installationen ist **InnoDB** Standard – wir müssen es nicht explizit angeben.

Start jetzt: Wenn nötig, wiederhole kurz die Basistabelle:

```
CREATE TABLE favourite_film (
  film_id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(100) NOT NULL,
  director VARCHAR(50),
```

```
released_year YEAR,  
star1 VARCHAR(50)  
);  
  
INSERT INTO favourite_film (title, director, released_year, star1)  
VALUES ('Lost in Translation', 'Sofia Coppola', 2003, 'Scarlett Johansson');
```

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m290_guko/learningunits/lu08/theorie/a_intro?rev=1760165247

Last update: **2025/10/11 08:47**

