LU08: Tabellen in Beziehung setzen – Fremdschlüssel & Referenzaktionen

Lernziel: Du kannst Fremdschlüssel (FOREIGN KEY) definieren, Referenzaktionen (RESTRICT/NO ACTION, CASCADE, SET NULL) erklären und ihre Wirkung praktisch testen. (Modul 290: HNK 1.5, 2.2, 5.1/5.2)

0) Vorbereitung: Lieblingsfilm wiederholen

```
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)
) ENGINE=InnoDB;

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

1) Ziel: Auf mehrere Tabellen normalisieren

Wir ergänzen sinnvolle Tabellen:

- **person** (Schauspieler:innen und Regie)
- **film** (ohne person-bezogene Strings)
- **film_cast** (M:N zwischen film und person)
- **director** (Optional: separate Tabelle) oder FK direkt in film

2) Variante A: FK schon beim Erstellen (CREATE TABLE)

```
DROP TABLE IF EXISTS film_cast;
DROP TABLE IF EXISTS film;
DROP TABLE IF EXISTS person;

CREATE TABLE person (
  person_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  UNIQUE(name)
) ENGINE=InnoDB;

CREATE TABLE film (
  film_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
  title VARCHAR(100) NOT NULL,
  released_year YEAR,
  -- Regie: Wenn Regisseur:in gelöscht wird, bleibt Film erhalten -> SET
NULL
  director_id INT NULL,
  CONSTRAINT fk_film_director
    FOREIGN KEY (director_id)
    REFERENCES person(person_id)
    ON DELETE SET NULL
    ON UPDATE RESTRICT
) ENGINE=InnoDB;

CREATE TABLE film_cast (
  film_id INT NOT NULL,
  person_id INT NOT NULL,
  ROLE VARCHAR(50) DEFAULT 'Actor',
  PRIMARY KEY (film_id, person_id),
  CONSTRAINT fk_cast_film
    FOREIGN KEY (film_id)
    REFERENCES film(film_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT fk_cast_person
    FOREIGN KEY (person_id)
    REFERENCES person(person_id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT
) ENGINE=InnoDB;
```

Warum diese Aktionen?

- **SET NULL** bei *director_id*: Film bleibt, Regie-FK wird nullbar.
- **CASCADE** bei *film_cast.film_id*: Wird ein Film gelöscht, verschwindet seine Besetzung.
- **RESTRICT** bei *film_cast.person_id*: Eine Person mit Cast-Einträgen darf nicht einfach gelöscht werden.

3) Variante B: FK nachträglich hinzufügen (ALTER TABLE)

Du kennst bereits Tabellen ohne FK. So fügst du später FKs hinzu:

```
ALTER TABLE film
  ADD CONSTRAINT fk_film_director
  FOREIGN KEY (director_id)
  REFERENCES person(person_id)
  ON DELETE SET NULL
  ON UPDATE RESTRICT;

ALTER TABLE film_cast
```

```
  ADD CONSTRAINT fk_cast_film
  FOREIGN KEY (film_id)
  REFERENCES film(film_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

4) Testdaten & Effekte prüfen

```
-- Personen
INSERT INTO person(name) VALUES ('Sofia Coppola'), ('Scarlett Johansson'),
('Bill Murray');

-- Filme
INSERT INTO film(title, released_year, director_id)
VALUES ('Lost in Translation', 2003,
        (SELECT person_id FROM person WHERE name='Sofia Coppola'));

-- Besetzung
INSERT INTO film_cast(film_id, person_id, ROLE)
SELECT f.film_id, p.person_id, 'Lead'
FROM film f JOIN person p ON p.name IN ('Scarlett Johansson','Bill Murray')
WHERE f.title='Lost in Translation';

-- TEST 1 (RESTRICT): Person mit Cast löschen -> sollte scheitern
DELETE FROM person WHERE name='Scarlett Johansson';

-- TEST 2 (SET NULL): Regie löschen -> director_id wird NULL
DELETE FROM person WHERE name='Sofia Coppola';
SELECT film_id, title, director_id FROM film;

-- TEST 3 (CASCADE): Film löschen -> zugehörige Cast-Reihen verschwinden
DELETE FROM film WHERE title='Lost in Translation';
SELECT * FROM film_cast;
```

Hinweis: Namen der automatisch erzeugten Constraints findest du mit:

```
SHOW CREATE TABLE film\G
SHOW CREATE TABLE film_cast\G
```

5) Aufträge (ca. 30′)

A1 – RESTRICT verstehen: Erzeuge eine neue Person „Giovanni Test“ und verknüpfe sie mit einem Film. Versuche, die Person zu löschen. Erkläre die Fehlermeldung in 1–2 Sätzen.

A2 – SET NULL im Einsatz: Stelle director_id auf NOT NULL um. Was passiert beim Löschen der Regie-Person nun? Dokumentiere die Fehlermeldung und begründe.

```
ALTER TABLE film MODIFY director_id INT NOT NULL;  -- danach DELETE erneut testen
```

A3 – CASCADE beobachten: Lege 2 Filme mit derselben Cast an. Lösche einen Film. Beweise via SELECT, dass nur die entsprechenden Cast-Zeilen verschwunden sind.

A4 – Bonus: Erweitere dein früheres ERD „Eishockeyverein“ um echte Fremdschlüssel (ON DELETE/UPDATE passend wählen). Erstelle die Tabellen mit ENGINE=InnoDB und teste mindestens 1 RESTRICT- und 1 CASCADE-Fall.

6) Merksätze

- FKs erzwingen **Referentielle Integrität** zwischen Parent- und Child-Tabelle.
- Ohne passende Parent-Zeile ist ein non-NULL FK im Child **nicht** erlaubt.
- Wähle Referenzaktionen nach Geschäftsregel: **bewahren**, **mitlöschen**, **entkoppeln**.