

LU08A1 - Lösungen: Fremdschlüssel per ALTER TABLE

Die Lösungen gehen vom Datenstand aus, den die LU06-Aufträge hinterlassen. Haben Sie in LU05 oder LU06 andere Daten eingetragen, unterscheiden sich IDs und Namen. Das Verhalten der Datenbank ist trotzdem dasselbe.

Was bei allen Cases passiert

Schritt	Test	Ergebnis	Warum
1	Fremdschlüssel setzen	erlaubt	Alle vorhandenen Werte in der Kindtabelle sind gültig
2a	Gültigen Wert in die Fremdschlüssel-Spalte eintragen	erlaubt	Den Elterneintrag gibt es
2b	Ungültigen Wert eintragen (z.B. 999)	Fehler 1452	Den Elterneintrag gibt es nicht. Die Kindzeile würde ins Leere zeigen.
3	Elterneintrag löschen, auf den verwiesen wird	Fehler 1451	Die Kindzeilen hätten danach keinen gültigen Elterneintrag mehr
4a	Primärschlüssel ändern, auf den nicht verwiesen wird	erlaubt	Keine Kindzeile zeigt auf diesen Eintrag
4b	Primärschlüssel ändern, auf den verwiesen wird	Fehler 1451	Die Kindzeilen würden auf eine ID zeigen, die es nicht mehr gibt

Die beiden Fehlermeldungen unterscheiden sich darin, von welcher Seite der Befehl kommt:

Fehler 1452 kommt aus der **Kindtabelle**: *Cannot add or update a child row: a foreign key constraint fails*

Ein neuer oder geänderter Wert in der Fremdschlüssel-Spalte hat keinen passenden Elterneintrag.

Fehler 1451 kommt aus der **Elterntabelle**: *Cannot delete or update a parent row: a foreign key constraint fails*

Ein Elterneintrag soll gelöscht oder geändert werden, auf den noch Kindzeilen zeigen.

Regel formulieren

Der Fremdschlüssel blockiert jeden Befehl, nach dem eine Kindzeile auf einen Elterneintrag zeigen würde, den es nicht gibt.

Ähnliche Formulierungen sind ebenfalls richtig, zum Beispiel: *Ein Elterneintrag ist geschützt, solange*

Kindzeilen auf ihn verweisen. Diese Regel deckt aber nur die Schritte 3 und 4 ab, nicht Schritt 2b.

Eishockeyverein

Fremdschlüssel: *spieler_in.block_id* → *block*

Verwiesen wird auf Block 1 «1. Linie». Nicht verwiesen wird auf Block 2 «2. Linie».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE spieler_in
ADD FOREIGN KEY (block_id) REFERENCES block(block_id);

-- Schritt 2a: gültiger Wert -> erlaubt
INSERT INTO spieler_in (name, rueckennummer, spielposition,
block_id)
VALUES ('Lara Stalder', 10, 'Stürmerin', 1);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO spieler_in (name, rueckennummer, spielposition,
block_id)
VALUES ('Lara Stalder', 10, 'Stürmerin', 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM block WHERE block_id = 1;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE block SET block_id = 20 WHERE block_id = 2;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
1451
UPDATE block SET block_id = 10 WHERE block_id = 1;
```

Postzustellung

Fremdschlüssel: *briefkasten.gebiet_id* → *postgebiet*

Verwiesen wird auf Gebiet 1 «Kreis 1». Nicht verwiesen wird auf Gebiet 2 «Kreis 4».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE briefkasten
ADD FOREIGN KEY (gebiet_id) REFERENCES
postgebiet(gebiet_id);

-- Schritt 2a: gültiger Wert -> erlaubt
```

```
INSERT INTO briefkasten (standort, leerungszeit, gebiet_id)
VALUES ('Paradeplatz', '17:45:00', 1);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO briefkasten (standort, leerungszeit, gebiet_id)
VALUES ('Paradeplatz', '17:45:00', 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM postgebiet WHERE gebiet_id = 1;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE postgebiet SET gebiet_id = 20 WHERE gebiet_id = 2;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
1451
UPDATE postgebiet SET gebiet_id = 10 WHERE gebiet_id = 1;
```

Öffentlicher Verkehr

Fremdschlüssel: *fahrt.linien_id* → *linie*

Verwiesen wird auf Linie 1 «Linie 4». Nicht verwiesen wird auf Linie 2 «Linie 11».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE fahrt
ADD FOREIGN KEY (linien_id) REFERENCES linie(linien_id);

-- Schritt 2a: gültiger Wert -> erlaubt
INSERT INTO fahrt (datum, abfahrtszeit, preis, linien_id)
VALUES ('2026-10-20', '10:00:00', 4.80, 1);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO fahrt (datum, abfahrtszeit, preis, linien_id)
VALUES ('2026-10-20', '10:00:00', 4.80, 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM linie WHERE linien_id = 1;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE linie SET linien_id = 20 WHERE linien_id = 2;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
1451
UPDATE linie SET linien_id = 10 WHERE linien_id = 1;
```

Online-Kleidershop

Fremdschlüssel: *kleidungsstueck.kategorie_id* → *kategorie*

Verwiesen wird auf Kategorie 1 «Hoodies». Nicht verwiesen wird auf Kategorie 2 «T-Shirts».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE kleidungsstueck
ADD FOREIGN KEY (kategorie_id) REFERENCES
kategorie(kategorie_id);

-- Schritt 2a: gültiger Wert -> erlaubt
INSERT INTO kleidungsstueck (name, preis, groesse,
zielgruppe, marken_id, kategorie_id)
VALUES ('ZRCL Zip Hoodie', 159.00, 'M', 'Unisex', 1, 1);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO kleidungsstueck (name, preis, groesse,
zielgruppe, marken_id, kategorie_id)
VALUES ('ZRCL Zip Hoodie', 159.00, 'M', 'Unisex', 1, 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM kategorie WHERE kategorie_id = 1;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE kategorie SET kategorie_id = 20 WHERE kategorie_id =
2;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
1451
UPDATE kategorie SET kategorie_id = 10 WHERE kategorie_id =
1;
```

Schweizer Regionen

Fremdschlüssel: *gemeinde.anlage_id* → *kva*

Verwiesen wird auf Anlage 5 «KVA Winterthur». Nicht verwiesen wird auf Anlage 1 «KVA Hagenholz».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE gemeinde
ADD FOREIGN KEY (anlage_id) REFERENCES kva(anlage_id);

-- Schritt 2a: gültiger Wert -> erlaubt
INSERT INTO gemeinde (name, einwohnerzahl, plz,
```

```
kantons_kennzeichen, anlage_id)
VALUES ('Schlieren', 20000, '8952', 'ZH', 2);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO gemeinde (name, einwohnerzahl, plz,
kantons_kennzeichen, anlage_id)
VALUES ('Schlieren', 20000, '8952', 'ZH', 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM kva WHERE anlage_id = 5;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE kva SET anlage_id = 20 WHERE anlage_id = 1;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
1451
UPDATE kva SET anlage_id = 50 WHERE anlage_id = 5;
```

Tierheim

Fremdschlüssel: *tier.art_id* → *tierart*

Verwiesen wird auf Tierart 3 «Kaninchen». Nicht verwiesen wird auf Tierart 1 «Hund».

```
-- Schritt 1: Fremdschlüssel setzen -> erlaubt
ALTER TABLE tier
ADD FOREIGN KEY (art_id) REFERENCES tierart(art_id);

-- Schritt 2a: gültiger Wert -> erlaubt
INSERT INTO tier (name, geburtsdatum, geschlecht, gehege_id,
art_id)
VALUES ('Nala', '2024-03-01', 'w', 2, 3);

-- Schritt 2b: ungültiger Wert -> Fehler 1452
INSERT INTO tier (name, geburtsdatum, geschlecht, gehege_id,
art_id)
VALUES ('Nala', '2024-03-01', 'w', 2, 999);

-- Schritt 3: Elterneintrag mit Verweis löschen -> Fehler
1451
DELETE FROM tierart WHERE art_id = 3;

-- Schritt 4a: Primärschlüssel ohne Verweis ändern ->
erlaubt
UPDATE tierart SET art_id = 20 WHERE art_id = 1;

-- Schritt 4b: Primärschlüssel mit Verweis ändern -> Fehler
```

1451

```
UPDATE tierart SET art_id = 30 WHERE art_id = 3;
```

Falls Schritt 1 scheitert

Meldet MySQL schon beim Setzen des Fremdschlüssels den Fehler 1452, steht in der Kindtabelle ein Wert, den es in der Elterntabelle nicht gibt. Die Meldung nennt dann eine Tabelle `#sql-alter-....`. Das ist eine Kopie, die MySQL während des Änderns anlegt. Gemeint ist die Kindtabelle.

So finden Sie den falschen Wert, hier am Beispiel Tierheim:

```
SELECT DISTINCT art_id FROM tier; -- welche Werte stehen in
der Kindtabelle?
SELECT art_id FROM tierart;      -- welche gibt es in der
Elterntabelle?
```

Jeder Wert aus der ersten Liste, der in der zweiten fehlt, ist verwaist. NULL zählt nicht dazu, NULL ist erlaubt. Den verwaisten Eintrag mit UPDATE auf eine bestehende ID (oder NULL) setzen oder mit DELETE löschen, danach Schritt 1 wiederholen.

M290-LU08

 Guido Koch

From:

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

Permanent link:

https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu08/loesungen/a_fk_in_lu05

Last update: **2026/10/25 20:27**

