LS4-4: Referentielle Integrität

# Situation

Nach den ersten SQL-Befehlen und kleinen Funktionstests sind erste Problemfälle aufgetreten.

Liebe Kolleginnen und Kollegen – bei den ersten SQL-Statements zum Löschen von Tabellen und Datensätze gab es größere Schwierigkeiten. Z. B. konnte die Tabelle „kunden“ bzw. einzelne Datensätze daraus nicht gelöscht werden:





Bitte klären Sie, was diese Fehlermeldungen zu bedeuten haben. Machen Sie auch einen Vorschlag, wie das Problem gefixt werden kann.

Welches Problem/Aufgabe stellt sich dar?

Welche Lösungsmöglichkeit(n) besteht(en)?

Referentielle Integrität – Praxisbeispiel

Auszug aus der Höllental-Senfmühle-Datenbank – **logisches Datenmodell**:

kunden

**kunde\_id (PK)**

anrede

vorname\_id

nachname

strasse

plz\_id

bestellung

**bestellung\_id (PK)**

datum

zeit

status

**kunden\_id (FK)**

1

n

email

Auszug aus der Höllental-Senfmühle -Datenbank - **Datensätze**:

|  |
| --- |
|  Eingabeaufforderung -mysql -u root ꟷ □ 🞨 |
| [..]MariaDB [jamando]> select \* from kunden;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| kunde\_id | anrede | vorname | nachname | strasse | plz\_id |
| 1 | Frau | Irene | Ahrens | Heidemarie-Bock-Allee 28 | 04886 |
| 2 | Frau | Irmtraut | Kraemer | Reuterring 0/8 | 31737 |
| 3 | Frau | Erna | Schneider | Greinerallee 21c | 39110 |
| 4 | Herr | Andreas | Vogt | Ullrichstrasse 2/0 | 61169 |
| 5 | Herr | Eckerhard | Fleisch | Hanns-Gross-Platz 1 | 54354 |

MariaDB [jamando]> select \* from bestellung;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bestellung\_id | datum | zeit | status | kunde\_id |
| 1 | 2020-03-01 | 13:04:32 | bezahlt | 4 |
| 13 | 2020-07-03 | 18:44:47 | bezahlt | 2 |
| 53 | 2021-05-27 | 09:02:07 | bezahlt | 1 |
| 76 | 2021-10-03 | 00:06:04 | offen | 3 |
| 86 | 2021-12-05 | 19:53:21 | offen | 3 |

MariaDB [jamando]> |

**Frage: Was passiert, wenn die Tabelle „kunden“ oder einzelne Datensätze der Tabelle „kunden“ gelöscht werden?**

In der Tabelle „bestellung“ bestehen Verknüpfungen zur Tabelle „kunden“. Wenn die Tabelle „kunden“ oder einzelnen Datensätze davon gelöscht wird gehen auch die notwendigen Informationen für die Bestellung verloren. Z. B. können dann die Bestellungen „76“ und „86“ keinem Kunden mehr zugeordnet werden. In der Folge kann auch der noch offene Rechnungsbetrag nicht mehr eingefordert werden.

**Mögliche Lösungsansätze**

**Keine Referenzkontrolle**: Ohne Referenzkontrolle kann ein Kunde in der Tabelle „kunde“ (Elterntabelle) gelöscht werden, ohne dass dies Änderungen in der Tabelle „bestellung“ (Kindtabelle) zur Folge hätte.

Wenn bspw. Erna Schneider (kunden\_id = 3) aus der Tabelle „kunde“ gelöscht wird, würde in der Tabelle „bestellung“ die kunden\_id = 3 zugeordnet bleiben, den es dann aber nicht mehr gibt (Inkonsistente Referenzen).

**RI-Kontrolle**: Das Löschen einer kunden\_id (z. B. kunden\_id = 3) in der Tabelle kunden wird verhindert, bis Erna Schneider in der Tabelle bestellung gelöscht wird. (= optimale Lösung).

**RI mit Löschweitergabe**: Das Löschen der kunden\_id = 3 in der Tabelle kunden führt automatisch zum Löschen von Datensätze mit der kunden\_id = 3 in der Tabelle bestellung. Dabei entsteht i. d. R. ungewollter Datenverlust.

Referentielle Integrität - INFO

Referentielle Integrität (auch Referenzielle Integrität, Beziehungsintegrität oder kurz RI) sorgt für Korrektheit und Konsistenz der Beziehungen zwischen den Relationen.

Die Bedeutung des Wortes „integer“ wird im Duden mit unbescholten oder unversehrt beschrieben. Datenkonsistenz bedeutet, dass die Daten vollständig, korrekt, widerspruchsfrei und aktuell sind.

# Beschreibung

Referentielle Integrität ist ein Regelsystem, mit dessen Hilfe ein Datenbankmanagementsystem (DBMS) automatisch sicherstellt, dass Beziehungen zwischen Datensätzen in Eltern- und Kindtabellen gültig sind und dass verknüpfte Daten nicht versehentlich gelöscht oder geändert werden.

Das DBMS überwacht also die Gültigkeit der Beziehung!

# Regeln

**Regel 1:** Die Eingabe von Werten in das Fremdschlüsselfeld der Kindtabelle ist nur möglich, wenn diese im Primärschlüsselfeld der Elterntabelle enthalten sind.

Abbildung 1: Screenshot aus MS Access
mit der Option „referentielle Integrität

**Regel 2:** Ein Datensatz aus der Elterntabelle kann nicht gelöscht werden, wenn übereinstimmende Datensätze in einer Kindtabelle enthalten sind.

**Regel 3:** Ein Primärschlüsselwert in der Elterntabelle kann nicht geändert werden, wenn es zu diesem Datensatz abhängige Datensätze in der Kindtabelle gibt.

Damit verhindert die referentielle Integrität **Änderungs-, Einfüge und Löschanomalien!**

Dem Datenbanksystem wird eine Primär-Fremdschlüssel-Beziehung mit referentieller Integrität ein einziges Mal bekannt gegeben (Setzen des Hakens in Abbildung 1), ab dann überwacht es selbständig die Einhaltung der Integritätsregel.

Das Datenbanksystem nimmt damit Entwicklern von DB-Frontends und anderen Datenbank-Usern ein Stück Verantwortung ab. Deshalb sollte ein Datenbank-Entwickler alle zum Zeitpunkt der Datendefinition bekannten Integritätsbedingungen in seiner Datenbankmodellierung berücksichtigen.

Heutige Datenbankmanagementsysteme sollten alle über eine referentielle Integrität verfügen.

Erweiterungen und Besonderheiten der referentiellen Integrität

Während die RI grundsätzlich vor inkonsistenten Datenaktionen schützt, bieten viele Datenbanksysteme Zusatzfunktionen an, die bei Updates von Eltern-Datensätzen nützlich sein können:

Abbildung 2: Screenshot aus MS Access
mit der Zusatzfunktion: Aktualisierungs- und Löschweitergabe

1. **Änderungsweitergabe (ÄW)**

Wenn der eindeutige Schlüssel eines Datensatzes geändert wird, kann das DBMS die Fremdschlüssel in allen abhängigen Datensätzen anpassen – anstatt die Änderung abzulehnen. Änderungsweitergabe wird insbesondere dann benutzt, wenn natürliche Schlüssel (die sich ändern können; Familienname bei Heirat) verwendet werden; denn künstliche Schlüssel sind i. d. R. unveränderlich und eine Änderungsweitergabe nicht erforderlich.

1. **Löschweitergabe (LW)**

In bestimmten Fällen ergibt es einen Sinn, abhängige Datensätze bei Löschung des Elterndatensatzes mit zu löschen.

Diese Funktionen können in der RI-Spezifikation optional gesetzt und (je nach DBMS) durch zusätzliche Bedingungen erweitert/präzisiert werden. Sie wirken nur bei Updates von Elterndatensätzen, Kinddaten können jederzeit gelöscht oder anderen (vorhandenen) Elterndatensätzen zugeordnet werden.