



B	Lehrgang: Datenbanken	Arbeitsblatt Nr. 20
S	Thema: Integritätsbedingungen	Datum:
G	Name:	Seite 1 von 2

Integrität von Daten

Eine Datenbank soll für den abzubildenden Ausschnitt der Realität korrekte Daten liefern.

Eine Art der Integrität von Daten ist sicherlich, ob die gespeicherten Werte wahr sind. Diese Form der Integrität lässt sich nur durch einen Vergleich mit der Wirklichkeit überprüfen!

Beispiel

Beträgt der Lagerbestand des Artikels mit der ID 23 "Toaster Superbraun" wirklich noch zehn Stück? Dazu muss jemand in das Lager gehen und nachsehen!

Eine andere Art von Integrität der Daten betrifft die Gestalt von Tabellen und deren Beziehungen.

Beispiel

Ein Kunde will den Artikel "Toaster Superbraun" bestellen. Man kann nun fragen, ob es in der Artikeltabelle einen solchen Eintrag mit der ID 23 gibt.

Die inhaltliche Korrektheit der gespeicherten Daten lässt sich in einer Datenbank nicht wirklich sichern. Aber die Einhaltung von bestimmten logischen Grundvoraussetzungen zu den Daten einer Datenbank lassen sich formulieren.

Die hierzu erforderlichen Regeln nennt man **Integritätsbedingungen**.

Integritätsbedingungen bei relationalen Datenbanken

Die Integrität der Daten kann durch die folgenden drei Bedingungen¹ gewährleistet werden:

1. Jeder Datensatz einer Tabelle hat einen eindeutigen Primärschlüsselwert. Man nennt diese Bedingung auch **Entity-Integrität**.
2. Zu jedem Wert eines Fremdschlüsselattributes in einer Tabelle T_1 gibt es in der referenzierten Tabelle T_2 einen Datensatz dessen Primärschlüsselattribut den gleichen Wert hat. Diese Form der Integrität nennt man **referenzielle Integrität**.
3. Weitere Bedingungen, die beim Anlegen von Tabellen formuliert wurden, sind erfüllt. Man nennt solche Bedingungen auch **Constraints** (Einschränkungen). Ein Constraint könnte sein, dass der Wert eines Attributes "Lagerbestand" nicht kleiner Null werden darf.

Einhalten der Integritätsbedingungen

Die **Entity-Integrität** wird dadurch erreicht, in dem man beim Anlegen einer Tabelle ein Attribut oder eine Attributkombination als Primärschlüssel definiert. Im Beispiel ist das Attribut **ID** als Primärschlüssel definiert.

In der Tabelle könnte nun noch ein Fremdschlüsselattribut **Plz_ID** hinzugefügt werden, das eine Beziehung zu einer weiteren Tabelle **Plz** herstellt.

Hierdurch wird die Einhaltung der **referenziellen Integrität** erzwungen.

Constraints können ebenfalls formuliert werden: Es wird eine Gültigkeitsregel definiert, die besagt, dass das Kreditlimit kleiner 10000 sein muss.

```
CREATE TABLE KUNDEN
(
  ID Counter PRIMARY KEY,
  Name Text(50),
  Adresse Text(50),
  Kredit_Limit Currency
)
```

```
CREATE TABLE KUNDEN
(
  ID Counter PRIMARY KEY,
  Name Text(50),
  Adresse Text(50),
  Plz_ID Integer,
  Foreign Key (Plz_ID) References Plz(ID),
  Kredit_Limit Currency
)
```

¹ Vgl. Schubert, Matthias: Datenbanken, B.G. Teubner Verlag, Wiesbaden 2004, S. 76
© Uwe Homm Version vom 17. Februar 2009



Bei MS Access lässt sich dies nicht angeben! Bei MS Access kann dies nur über die Benutzeroberfläche im Modus "Erstellen einer Tabelle in der Entwurfsansicht" erfolgen.

Referenzielle Integrität detaillierter

```
CREATE TABLE KUNDEN
(
  ID Counter PRIMARY KEY,
  Name Text(50),
  Adresse Text(50),
  Plz_ID Integer,
  Foreign Key (Plz_ID) References PLZ(ID),
  Kredit_Limit Currency CHECK (Kredit_Limit <10000)
)
```

Die referenzielle Integrität kann prinzipiell beim Ändern oder beim Löschen von Datensätzen verletzt werden. Wird z.B. bei dem bereits genannten Toaster Superbraun dessen Primärschlüsselwert (ID=23) geändert, zeigen alle bisherigen Bestellungen, die auf diesen Artikel verweisen, "ins Leere". Das Gleiche gilt natürlich für das Löschen dieses Artikels.

Um die referenzielle Integrität zu wahren, sind zwei Varianten möglich:

1. Entweder verhindert das DBMS eine solche Änderung oder Löschung, oder
2. das DBMS ändert bzw. löscht alle zugehörigen Fremdschlüsselwerte.

Im Beispiel bedeutet das für den Fall 2, dass in allen bisherigen Bestellungen nun entweder der neue Fremdschlüsselwert verwendet wird, oder die Bestellungen werden gelöscht!

Für den Fall 1 kann hinter der Foreign Key-Anweisung der Zusatz **ON DELETE RESTRICTED** bzw. **ON UPDATE RESTRICTED** angegeben werden.

Für den Fall 2 kann hinter der Foreign Key-Anweisung der Zusatz **ON DELETE CASCADED** bzw. **ON UPDATE CASCADED** angegeben werden.

Bei MS Access wird zur Wahrung der referenziellen Integrität standardmäßig die Option **RESTRICTED** verwendet. Diese lässt sich auch nur über die Benutzeroberfläche einstellen bzw. ändern. In den SQL-Befehlen wird ein Syntaxfehler gemeldet.

In anderen DBMS stehen weitere Optionen zur Verfügung; z.B. kann auch der Zusatz **ON DELETE** bzw. **ON UPDATE SET NULL** verwendet werden. Hierbei würden die betroffenen Fremdschlüsselwerte den Wert **NULL** erhalten. Dies ist jedoch keine gute Vorgehensweise. **NULL**-Werte sollten in Datenbanken prinzipiell vermieden werden!

Übungen

Erstellen Sie für die nachfolgenden Übungen eine Kopie der Datenbank **Allerhand.mdb**!

1. Öffnen Sie im Menü "Extras" den Menüeintrag "Beziehungen..." und aktivieren Sie bei allen Beziehungen die Kontrollkästchen "Aktualisierungsweitergabe" und "Löschweitergabe". Ändern Sie in der Tabelle Artikelgruppe die ID 1 auf 11. Welche Auswirkung hat dies auf die Datensätze in der Tabelle Artikel?
2. Löschen Sie in der Tabelle Artikelgruppe den Datensatz mit der ID 11. Welche Auswirkung hat dies auf die Tabellen Artikel und Bestellungen?

