

Kapitel 7: Referentielle Integrität

- ▶ Im Allgemeinen sind nur solche Instanzen einer Datenbank erlaubt, deren Relationen die der Datenbank bekannten *Integritätsbedingungen* (IB) erfüllen.
- ▶ Integritätsbedingungen sind von ihrer Natur aus *deklarativ*: sie definieren die zulässigen Instanzen, ohne auszudrücken, wie eine Gewährleistung der Integrität implementiert werden kann.

Eine wichtige Klasse von deklarativen Integritätsbedingungen sind *Fremdschlüsselbedingungen*, die gewährleisten, dass keine *dangling* Referenzen zwischen den Tupeln in den Tabellen bestehen. Man redet auch von *referentieller Integrität*.

- ▶ Komplementär zu den deklarativen Bedingungen bieten Datenbanksysteme einen *Trigger-Mechanismus* an, mit dem in Form von Regeln definiert werden kann, welche Aktionen zur Gewährleistung der Integrität vorgenommen werden sollen, bzw., wie Verletzungen behandelt werden sollen.

- ▶ *Fremdschlüsselbedingungen* werden als Teil der CREATE TABLE-Klausel definiert: zu jedem von null verschiedenen Fremdschlüsselwert in einer Zeile der *referenzierenden* Tabelle, der C- (child-) Tabelle, existiert ein entsprechender Schlüsselwert in einer Zeile der *referenzierten* Tabelle, der P- (parent-) Tabelle.
- ▶ Man redet hier auch von *referentieller* Integrität. Zur Definition von Fremdschlüsselbedingungen steht die FOREIGN KEY-Klausel zur Verwendung in der C-Tabelle zur Verfügung.

Die Spalte LCode innerhalb der Tabelle Provinz enthält Werte des Schlüssels LCode der Tabelle Land. Zu jeder Zeile in Provinz muss eine Zeile in Land existieren, deren Schlüsselwert gleich dem Fremdschlüsselwert ist.

```
CREATE TABLE Provinz (  
    PName    VARCHAR(35),  
    LCode    VARCHAR(4),  
    Fläche   NUMBER  
    PRIMARY KEY (PName, LCode),  
    FOREIGN KEY (LCode) REFERENCES Land (LCode) )
```

Die Zeilen der Tabelle Grenze enthalten jeweils zwei unterschiedliche Fremdschlüssel, die beide Werte des Schlüssels LCode der Tabelle Land annehmen.

```
CREATE TABLE Grenze (  
    LCode1    VARCHAR(4),  
    LCode2    VARCHAR(4),  
    Länge     INTEGER,  
    PRIMARY KEY (LCode1, LCode2),  
    FOREIGN KEY (LCode1) REFERENCES Land (LCode),  
    FOREIGN KEY (LCode2) REFERENCES Land (LCode) )
```

Für die Tabelle Stadt sind zwei Fremdschlüsselbeziehungen relevant. Einmal müssen die referenzierten Länder in der Tabelle zu Land existieren, und zum andern entsprechend die Provinzen. Letzterer Fremdschlüssel besteht aus zwei Spalten. Die Zuordnung der einzelnen Spalten des Fremd- und Primärschlüssels ergeben sich aus der Reihenfolge des Hinschreibens.

```
CREATE TABLE Stadt (  
    :  
    PRIMARY KEY (SName, LCode, PName),  
    FOREIGN KEY (LCode) REFERENCES Land (LCode),  
    FOREIGN KEY (LCode, PName) REFERENCES Provinz (LCode, PName) )
```

referentielle Aktionen im Überblick

Zur Gewährleistung der referentiellen Integrität werden sogenannte *referentielle Aktionen* zur Ausführung bezüglich der C-Tabellen definiert. Aufgabe dieser Aktionen ist die Kompensierung von durch DELETE- und UPDATE-Operationen auf der zugehörigen P-Tabelle verursachten Verletzungen der Integrität.

- ▶ Änderungen der P-Tabelle werden auf die C-Tabelle übertragen (Aktion CASCADE).
- ▶ Die Änderung der P-Tabelle wird im Falle einer Verletzung der referentiellen Integrität einer C-Tabelle abgebrochen (Aktion NO ACTION und Aktion RESTRICT).
- ▶ Der Fremdschlüsselwert der C-Tabelle wird angepaßt (Aktion SET NULL und Aktion SET DEFAULT).

Hinweis: Oracle unterstützt nur DELETE-Operationen. Als referentielle Aktionen kann CASCADE und SET NULL, SET DEFAULT gewählt werden - keine Angabe entspricht NO ACTION.

Wird der Code eines Landes geändert oder das Land gelöscht, so sollen die neuen Codes bei den zugehörigen Provinzen nachgezogen werden, bzw. auch die Provinzen des gelöschten Landes gelöscht werden.

```
CREATE TABLE Provinz (  
    :  
    FOREIGN KEY (LCode) REFERENCES Land (LCode)  
    ON DELETE CASCADE ON UPDATE CASCADE )
```

Werden Provinzen gelöscht, so sollen ihre Städte weiter in der Datenbank bestehen bleiben, wobei der betreffende Fremdschlüsselwert Nullwerte erhält. Änderungen eines Provinzschlüssels sollen auf die betroffenen Städte übertragen werden.

```
CREATE TABLE Stadt (  
    :  
    PRIMARY KEY (SNAME)  
    FOREIGN KEY (LCode, PName)  
    REFERENCES Provinz (LCode, PName)  
    ON DELETE SET NULL ON UPDATE CASCADE )
```

Warum werden nur DELETE- und UPDATE-Operationen auf den zugehörigen P-Tabellen betrachtet?

- ▶ Einfügen bezüglich der P-Tabelle oder Löschen bezüglich der C-Tabelle ist für die referentielle Integrität immer unkritisch.
- ▶ Einfügen bezüglich der C-Tabelle oder Ändern bezüglich der C-Tabelle, die einen Fremdschlüsselwert erzeugen, zu dem kein Schlüssel in der P-Tabelle existiert, sind immer primär unzulässig, da von Änderungen in den C-Tabellen im Allgemeinen kein sinnvoller Rückschluss auf Änderungen der P-Tabellen möglich ist; anderenfalls sind die Änderungen unkritisch.

referentielle Aktionen

- NO ACTION:** Die Operation auf der P-Tabelle wird zunächst ausgeführt; ob Dangling References in der C-Tabelle entstanden sind wird erst nach Abarbeitung aller durch die Operation auf der P-Tabelle direkt oder indirekt ausgelösten referentiellen Aktionen überprüft.
- RESTRICT:** Die Operation auf der P-Tabelle wird nur dann ausgeführt, wenn durch ihre Anwendung keine Dangling References in der C-Tabelle entstehen.
- CASCADE:** Die Operation auf der P-Tabelle wird ausgeführt. Erzeugt die DELETE/UPDATE-Operation Dangling References in der C-Tabelle, so werden die entsprechenden Zeilen der C-Tabelle ebenfalls mittels DELETE entfernt, bzw. mittels UPDATE geändert. Ist die C-Tabelle selbst P-Tabelle bezüglich einer anderen Bedingung, so wird das DELETE/UPDATE bezüglich der dort festgelegten Löschr/Änderungs-Regel weiter behandelt.
- SET DEFAULT:** Die Operation auf der P-Tabelle wird ausgeführt. In der C-Tabelle wird der entsprechende Fremdschlüsselwert durch die für die betroffenen Spalten in der C-Tabelle festgelegten DEFAULT-Werte ersetzt; es muss jedoch gewährleistet sein, daß entsprechende Schlüsselwerte in den P-Tabellen existieren.
- SET NULL:** Die Operation auf der P-Tabelle wird ausgeführt. In der C-Tabelle wird der entsprechende Fremdschlüsselwert spaltenweise durch NULL ersetzt. Voraussetzung ist hier, daß Nullwerte zulässig sind.