

Einführung in SQL

Karel Štípek

Einleitung

In PCNEWS 53 (Juni 1997) habe ich auf Seite 82 ein Clipper-Programm veröffentlicht, das zwei Tabellen miteinander vergleicht. Die Tabellen sind sequentiell gelesen und dabei die Unterschiede in eine Ergebnistabelle geschrieben worden.

In modernen Datenbanksystemen erfolgt die Datenbearbeitung meistens mittels SQL. SQL (*Structured Query Language*) ist eine standardisierte Sprache, mit der man eigentlich das gewünschte Ergebnis beschreibt ohne die Bearbeitung der einzelnen Datensätze programmieren zu müssen.

Im folgenden Artikel wird die Lösung des Tabellenvergleichs mit SQL in MS-Access 97 dargestellt.

Grundlagen der SQL-Selektion

Die einfachste Tabellenabfrage hat folgende Form:

```
SELECT <Feldliste> FROM <Tabelle oder Tabellenverknüpfung>
```

Weiters können Selektionskriterien und Sortierung mit

```
WHERE <Bedingung>
ORDER BY <Feldliste>
```

angegeben werden.

Tabellenverknüpfungen - JOINS

Mit nur einer Tabelle kann man gewöhnlich nicht viel anfangen. Mehrere Tabellen werden mittels Schlüsselfeldern verknüpft. Diese Verknüpfungen werden in SQL mit der Anweisung JOIN kodiert.

Die Grundform ist:

```
SELECT <Feldliste> FROM <Tab1> JOIN <Tab2>
ON <Sch11> = <Sch12>
```

Dieses Befehl liefert die ausgewählten Tabellenfelder nur aus den Datensätzen, für die in beiden Tabellen die Werte der Schlüsselfelder <Sch11> und <Sch12> gleich sind. Die Verknüpfung wird auch INNER JOIN genannt.

Es ist aber manchmal auch notwendig, aus einer der Tabellen alle Datensätze zu selektieren, unabhängig davon, ob der entsprechende Schlüssel in der anderen Tabelle existiert oder nicht. Die eventuell nicht existierenden verknüpften Felder haben den undefinierten Wert NULL.

Das wird mit einem OUTER JOIN realisiert. Die Anweisung hat zwei Varianten:

- LEFT JOIN - aus der auf der linken Seite angeführten Tabelle werden alle Datensätze selektiert

- RIGHT JOIN - das gleiche mit der anderen Tabelle.

Tabellenvergleich mit SQL

Die Erklärung der SQL Syntax wird durch ein praktisches Beispiel anschaulicher. Gehen wir davon aus, dass wir zwei Tabellen "Tab1", "Tab2" mit den Schlüsselfeldern "Sch1" und Nicht-Schlüsselfeldern "Wert" vergleichen. Es werden die für die Teilaufgaben eines Tabellenvergleichs notwendigen SQL-Anweisungen aufgelistet.

- 1 Welche Schlüsselwerte sind in der ersten, nicht aber in der zweiten Tabelle ?

```
SELECT Tab1.Sch1 FROM Tab1 LEFT JOIN Tab2 ON
Tab1.Sch1 = Tab2.Sch12 WHERE Tab2.Sch1 IS
NULL
```

- 2 Welche Schlüsselwerte sind in der zweiten, nicht aber in der ersten Tabelle ?

```
SELECT Tab2.Sch1 FROM Tab1 RIGHT JOIN Tab2
ON Tab1.Sch1 = Tab2.Sch12 WHERE Tab1.Sch1 IS
NULL
```

- 3 Welche Nicht-Schlüsselwerte sind gleich (bei gleichen Schlüsseln) ?

```
SELECT Tab1.Sch1, Tab1.Wert FROM Tab1 INNER
JOIN Tab2 ON Tab1.Sch1 = Tab2.Sch12 WHERE
Tab1.Wert = Tab2.Wert
```

- 4 Welche Nicht-Schlüsselwerte sind unterschiedlich (bei gleichen Schlüsseln) ?

```
SELECT Tab1.Sch1, Tab1.Wert, Tab2.Wert
FROM Tab1 INNER JOIN Tab2 ON Tab1.Sch1 =
Tab2.Sch12 WHERE Tab1.Wert <> Tab2.Wert
```

Was kann SQL noch ?

Mit der bereits erklärten SELECT-Anweisung sind die Möglichkeiten von SQL bei weitem nicht ausgeschöpft. Mit SQL können nicht nur Daten selektiert, sondern auch die Tabellen erstellt (CREATE TABLE), die Datensätze geändert (UPDATE), hinzugefügt (INSERT) oder gelöscht (DELETE) werden, u.a.

Implementierung in MS-Access

- 1 Die SQL-Anweisung wird am einfachsten mit dem eingebauten Abfrage-Generator erstellt. Auf der graphischen Oberfläche werden die Tabellen einfach verknüpft und die Felder ausgewählt. Das Ergebnis der Abfrage und der generierte SQL-Ausdruck kann angeschaut werden.
- 2 Der SQL-Ausdruck kann anstelle eines Tabellen- oder Abfragenamen direkt in die Eigenschaften der Formulare, Berichte oder Steuerelemente eingetragen werden.
- 3 Eine SQL-Anweisung kann aus dem Programmcode durchgeführt werden mit

```
DoCmd.RunSQL <SQL-Ausdruck>
```

- 4 Die flexibelste Möglichkeit, eine Datenabfrage zu definieren ist es, den SQL-Ausdruck einer Abfrage erst wäh-

rend der Laufzeit zusammenzulegen und ihn einer Abfrage zuzuordnen.

Programm CompTab

Die bisher besprochenen Techniken bilden den Kern eines Access-Tools, das ich sehr oft bei der Entwicklung verwende. Die Funktionalität ähnelt der des oben genannten Clipper-Programms, die Verbesserung besteht darin, dass man auch Tabellen mit unterschiedlichen Strukturen vergleichen kann und die ganze Bedienung sehr einfach ist.

Die einzige Maske des Programms ist auf folgendem Bild dargestellt.



Aus zwei Access-Datenbanken können zwei Tabellen ausgewählt werden. Die vorhandenen Felder werden in den großen Listboxen aufgelistet. Eins oder mehrere verknüpften Schlüsselfelder und verglichene Felder können mit den Schaltflächen mit Pfeilen in die kleineren Listboxen übernommen werden.

Nach dem Aktivieren der unteren Schaltflächen werden die SQL-Ausdrücke für die Abfragen dynamisch erstellt und die Ergebnisse angezeigt.

Standards und Abweichungen

Auch wenn die SQL-Sprache normiert ist, gibt es doch bei diversen Datenbanksystemen kleine Unterschiede.



Einen guten Überblick kann man mit folgendem Buch gewinnen.

The SQL Programmer's Reference von Wayne S. Freeze; Verlag: Ventana 1998, ISBN:

1-56604-760-9

Der Inhalt des Buches ist auch auf der beigelegten CD enthalten.

Das Buch ist kein Lehrbuch für Anfänger, sondern eine umfangreiche Sprachreferenz. Die genaue Syntax und spezifische Unterschiede werden für folgende Normen, bzw. Datenbanken aufgelistet: ANSI SQL-92, IBM DB-2 Version 5, Informix Online Server Version 7.22, Microsoft Access 97, Microsoft SQL-Server 6.5, Oracle Version 8, Sybase SQL Server Version 11.1