

Tipps & Tricks: Mai 2008

Bereich:	PL/SQL	Erstellung:	02/2008 HA
Versionsinfo:	11.1	Letzte Überarbeitung:	04/2009 HA

 [Als PDF Downloaden!](#)

Compound Trigger zur Vermeidung von Mutating Tables in 11g

Es kommt immer wieder vor, dass bei einem DML-Trigger für Überprüfungen oder Protokollierungen Daten aus der zu ändernden Tabelle benötigt werden. Ein normaler Row Trigger scheidet dann in der Regel aus, weil man sonst in das Problem mit den [Mutating Tables](#) läuft.

Ein kleines Beispiel dazu: Stellen Sie sich vor, eine Geschäftsregel besagt, dass niemand innerhalb einer Berufsgruppe mehr als 50% über dem Durchschnittsgehalt verdienen darf. Der einfachste Ansatz dazu sähe folgendermaßen aus:

```
CREATE OR REPLACE TRIGGER check_sal
AFTER UPDATE OF sal ON emp
FOR EACH ROW
DECLARE
    v_avgsal NUMBER;
BEGIN
    SELECT avg(sal)
        INTO v_avgsal
        FROM emp
        WHERE job = :NEW.job;

    IF :NEW.sal > 1.5 * v_avgsal THEN
        RAISE_APPLICATION_ERROR(-20000, 'Zu weit über Durchschnitt');
    END IF;
END;
/
```

Das geht aber leider schief, sobald Sie ein Update ausführen:

```
UPDATE emp
    SET sal = 1500
    WHERE empno = 7934
Error at line 3
ORA-04091: Tabelle SCOTT.EMP wird gerade geändert, Trigger/Funktion sieht dies
möglicherweise nicht
ORA-06512: in "SCOTT.CHECK_SAL", Zeile 4
ORA-04088: Fehler bei der Ausführung von Trigger 'SCOTT.CHECK_SAL'
```

Das Problem mit den Mutating Tables hat zugeschlagen! Ein Workaround dazu könnte so aussehen, dass Sie über einen BEFORE STATEMENT Trigger zuerst den Durchschnitt aller Berufsgruppen einlesen und

beispielsweise in einem Package oder in einer Tabelle zwischenspeichern. Der Row Trigger kann dann auf diese Werte zugreifen, ohne die zu ändernde Tabelle lesen zu müssen. Das ist machbar, aber aufwendig.

Ab Version 11g kann die Lösung viel einfacher aussehen: Compound Trigger haben Abschnitte für bisher unterschiedliche Triggertypen und bieten die Möglichkeit, Werte für die Dauer der Durchführung des DML-Befehls zu speichern. Obiges Beispiel könnte mit einem Compound Trigger so aussehen:

```
CREATE OR REPLACE TRIGGER check_sal
  FOR UPDATE OF sal ON SCOTT.EMP
COMPOUND TRIGGER
  TYPE t_number IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
  TYPE t_varchar IS TABLE OF VARCHAR2(30) INDEX BY PLS_INTEGER;
  TYPE t_avgsal IS TABLE OF NUMBER INDEX BY VARCHAR2(30);

  v_avgsal t_number;
  v_job    t_varchar;
  v_avgJob t_avgsal;
  v_index  VARCHAR2(30);

  BEFORE STATEMENT IS
  BEGIN
    SELECT AVG(sal), job
      BULK COLLECT INTO v_avgsal, v_job
    FROM EMP e
    GROUP BY e.job;

    FOR i IN 1..v_avgsal.COUNT
    LOOP
      v_avgJob(v_job(i)) := v_avgsal(i);
    END LOOP;
  END BEFORE STATEMENT;

  AFTER EACH ROW IS
  BEGIN
    IF v_avgJob.EXISTS(:NEW.job)
    THEN
      IF :NEW.sal > 1.5 * v_avgJob(:NEW.job)
      THEN
        RAISE_APPLICATION_ERROR(-20000,
          'Verdienst ist zu weit über Durchschnitt');
      END IF;
    END IF;
  END AFTER EACH ROW;
END;
/
```

Erlaubter Update:

```
UPDATE emp
  SET sal = 1500
 WHERE empno = 7934
1 row updated.
```

Verbotener Update:

```
UPDATE emp
  SET sal = 1800
  WHERE empno = 7934
ORA-20000: Zu weit über Durchschnitt
ORA-06512: in "SCOTT.CHECK_SAL", Zeile 29
ORA-04088: Fehler bei der Ausführung von Trigger 'SCOTT.CHECK_SAL'
```