

Tipps & Tricks: FORALL

Bereich:	PL/SQL	Erstellung:	01/2002 HA
Versionsinfo:	9.2, 10.2, 11.	Letzte Überarbeitung:	06/2009 EF

FORALL

Um anschliessend an einen [Bulk Collect](#) beliebig viele Werte an einen DML-Befehl zu übergeben, der dann durch ein einziges Execute ausgeführt wird, brauchen Sie die FORALL-Klausel. Dabei handelt es sich NICHT um eine Schleife, auch wenn ein Index beteiligt ist.

FORALL bezieht sich auf genau einen DML-Befehl, der unmittelbar dahinter steht und den Index in der FORALL-Klausel referenzieren muss. Die Collection darf allerdings bei der Standard-Syntax keine Lücken im Index enthalten (Diese Einschränkung lässt sich durch Verwendung der Klausel [IN INDICES OF](#) umgehen).

Ab Version 9.2 können auch auf Records basierende Collections in einem FORALL DML-Befehl verwendet werden. Allerdings darf dann nur der komplette Record angesprochen werden, nicht aber einzelne Felder daraus. Daher ist ein sinnvolles FORALL mit Update nur über Umwege machbar. Einzelheiten dazu finden Sie [hier](#). Diese Beschränkung ist in Version 11g endlich aufgehoben, siehe [Bulk DML mit Record-Arrays in 11g](#).

Sie können die Klausel SAVE EXCEPTIONS nutzen (ab Version 9i), damit der DML-Befehl für alle Indexwerte durchgeführt wird, auch wenn er für einen bestimmten Indexwert fehlschlägt. Einzelheiten zu SAVE EXCEPTIONS finden Sie [hier](#).

Beispiele:

```
DECLARE
  TYPE ma_nummern IS TABLE OF emp.empno%TYPE;
  v_empno ma_nummern;
  TYPE ma_gehalt IS TABLE OF emp.sal%TYPE;
  v_sal ma_gehalt;
BEGIN
  SELECT empno, sal
  BULK COLLECT INTO v_empno, v_sal
  FROM emp;
  ...
  -- Berechnungen und Änderungen an v_sal
  ...
  FORALL i IN emp_tab.FIRST .. emp_tab.LAST SAVE EXCEPTIONS
    UPDATE emp
      SET sal = v_sal(i)
      WHERE empno = v_empno(i);
END;
/

-- Folgendes Beispiel ist ab Version 9.2 lauffähig:
CREATE TABLE emp_copy AS SELECT * FROM emp WHERE 1=2;
DECLARE
  CURSOR c IS SELECT * FROM scott.emp;
  TYPE emp_tab_type IS TABLE OF emp%ROWTYPE INDEX BY BINARY_INTEGER;
  emp_tab emp_tab_type;
```

```
BEGIN
  OPEN c;
  FETCH c BULK COLLECT INTO emp_tab;
  CLOSE c;

  emp_tab(1).empno := emp_tab(1).empno + 1000;  -- hier zulässig

  FORALL i IN emp_tab.FIRST..emp_tab.LAST
    INSERT INTO emp_copy VALUES emp_tab(i);
  COMMIT;
END;
/
```