

## Tipps & Tricks: FORALL in 10g

Bereich:	PL/SQL	Erstellung:	09/2005 HA
Versionsinfo:	10.2, 11.1	Letzte Überarbeitung:	06/2009 EF

### Syntax-Erweiterung zu FORALL (ab Version 10g)

Nach der Erweiterung der Möglichkeiten in 9i durch Einführung der [SAVE EXCEPTIONS-KLAUSEL](#) wurde Bulk DML in 10g noch komfortabler einsetzbar:

- Führt es bisher unweigerlich zu einem Fehler, wenn die verwendete [COLLECTION](#) Lücken enthielt, so wurde nun die Syntax dahingehend erweitert, dass es möglich ist, auf vorhandene Indizes einzuschränken.
- Auch die Möglichkeit wurde eingeführt, den DML-Befehl auf ganz bestimmte vorgegebene Indizes einzuschränken.

#### Einschränkung auf vorhandene Indizes

Fehler durch Lücken können vermieden werden durch die neue Syntax

```
FORALL <index> IN INDICES OF [ BETWEEN <x> AND <y> ] [ SAVE EXCEPTIONS ]
```

, wobei zusätzlich durch die optionale Klausel BETWEEN Untergrenze AND Obergrenze auf einen bestimmten Bereich eingegrenzt werden kann. Auch die Ergänzung SAVE EXCEPTIONS ist weiterhin möglich.

Beispiel für 9i:

```
DECLARE
  TYPE v_tab_type IS TABLE OF emp%ROWTYPE;
  v_tab v_tab_type;
BEGIN
  SELECT * BULK COLLECT INTO v_tab FROM EMP;
  v_tab.DELETE(3);
  FORALL I IN v_tab.FIRST..v_tab.LAST SAVE EXCEPTIONS
    INSERT INTO emp_forall VALUES v_tab(i);
  DBMS_OUTPUT.PUT_LINE('Fertig');
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(sqlerrm);
END;
```

In dieser 9i-Version werden zwar 13 Datensätze eingefügt, aber hinterher wird der Ausführungsteil verlassen; die Ausgabe von "Fertig" findet nicht mehr statt; stattdessen wird die Fehlermeldung ausgegeben.

Hier nun die 10g-Variante, die bis zum Ende durchläuft:

```
DECLARE
  TYPE v_tab_type IS TABLE OF emp%ROWTYPE;
  v_tab v_tab_type;
BEGIN
```

```

SELECT * BULK COLLECT INTO v_tab FROM EMP;
v_tab.DELETE(3);
FORALL I IN INDICES OF v_tab SAVE EXCEPTIONS
    INSERT INTO emp_forall VALUES v_tab(i);
DBMS_OUTPUT.PUT_LINE('Fertig');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
END;
/

```

### Einschränkung auf vorgegebene Indizes

In dieser Variante benötigt man mindestens zwei Collections, eine für den DML-Befehl und eine, die die zu verarbeitenden Indizes aufnimmt; letztere muss vom Datentyp PLS\_INTEGER oder BINARY\_INTEGER sein. Die angegebenen Indizes müssen vorhanden sein, sonst kommt es zu einem Laufzeitfehler. Auch hier ist die Ergänzung von SAVE EXCEPTIONS zulässig.

```
FORALL <index> IN VALUES OF <collection> [SAVE EXCEPTIONS]
```

Beispiel:

```

DECLARE
    TYPE v_tab_type IS TABLE OF emp%ROWTYPE;
    v_tab v_tab_type;
    TYPE index_type IS TABLE OF PLS_INTEGER INDEX BY PLS_INTEGER;
    v_index index_type;
BEGIN
    SELECT * BULK COLLECT INTO v_tab FROM EMP;
    FOR i IN 1..v_tab.COUNT LOOP
        IF MOD(i, 2) = 0 THEN
            v_index(i) := i;
        END IF;
    END LOOP;
    FORALL I IN VALUES OF v_index SAVE EXCEPTIONS
        INSERT INTO emp_forall
            VALUES v_tab(i);
    DBMS_OUTPUT.PUT_LINE('Fertig');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
END;
/

```

In diesem Beispiel wird nur jeder zweite Datensatz aus emp in emp\_forall eingetragen; die Lücken in der Collection v\_index stören nicht.