Tipps & Tricks: Records bei Insert und Update und Bulk Binds

Bereich:	PL/SQL	Erstellung:	06/2002 HA
Versionsinfo:	9.2, 10.2, 11.1	Letzte Überarbeitung:	06/2009 EF

Records bei Insert, Update und Bulk Binds

Ab Version 9.2 ist es innerhalb von PL/SQL möglich, an eine Insert-Anweisung statt der einzelnen Felder einen kompletten Record zu übergeben. Dieser Record muss allerdings genau die Struktur der Tabelle widerspiegeln, auch wenn einzelne Felder leer bleiben dürfen. Eine Deklaration mit %ROWTYPE ist hier sehr empfehlenswert (aber nicht zwingend).
Gleiches gilt für Update-Anweisungen; hier wurde zu diesem Zweck das Schlüsselwort ROW in der SET-Klausel eingeführt.

Weder bei Insert noch bei Update dürfen einzelne Spalten angesprochen werden, und es können auch keine weiteren Variablen in der VALUES- bzw. SET-Klausel verwendet werden. Nicht belegte Felder werden bei beiden Befehlen auf NULL gesetzt. Objekte und Collections als Felder des Records sind erlaubt, nicht jedoch weitere Records. Auch im Zusammenhang mit EXECUTE IMMEDIATE sind Records nicht zulässig.

Beispiel:

```
DECLARE
-- explizite Deklaration des Records
   TYPE dept_type IS RECORD(
      deptno  scott.dept.deptno%TYPE,
       dname   scott.dept.dname%TYPE,
   location  VARCHAR2(13));
   dept_rec  dept_type;
   emp_rec   scott.emp%ROWTYPE;
BEGIN
   dept_rec.dname := 'Forschung';
   dept_rec.location := 'München';
   UPDATE scott.dept SET ROW = dept_rec WHERE deptno = 30;
   SELECT * FROM dept INTO dept_rec where deptno = 30
   INSERT INTO scott.emp_copy VALUES emp_rec;
END;
```

Returning-Klausel

Auch in der Returning-Klausel einer DML-Anweisung kann innerhalb von PL/SQL ein Record verwendet werden. Weitere Variablen sind dann nicht zulässig. Der Record braucht hier in der Struktur nicht der kompletten Tabelle zu entsprechen.

Beispiel:

```
DECLARE
  dept_rec   dept%ROWTYPE;
   v_string  VARCHAR2(2000);
BEGIN
   UPDATE  scott.dept SET deptno = 50 WHERE deptno = 40
```

```
   RETURNING dname, loc INTO v_rec;
   v_string := 'Abteilung '||v_rec.name||' in '||v_rec.loc||' wurde geändert';
   DBMS_OUTPUT.PUT_LINE(v_string);
END;
```

Bulk Binds

Ab Version 9.2 kann die Collection bei der BULK COLLECT INTO-Klausel als Datentyp auch einen Record haben. Dieser kann dann auch 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.

Beispiel:

```
DECLARE
   CURSOR c IS SELECT * FROM scott.emp;
   TYPE rectab IS TABLE OF scott.emp%ROWTYPE INDEX BY BINARY_INTEGER;
   v_rec  rectab;
BEGIN
   OPEN c;
   FETCH c BULK COLLECT INTO v_rec;
   CLOSE c;
  /*
      alternativ:
      SELECT * BULK COLLECT INTO v_rec FROM SCOTT.emp;
   */
   v_rec(1).empno := v_rec(1).empno + 1000;  -- hier zulässig
   FORALL i IN v_rec.FIRST..v_rec.LAST
      INSERT INTO DUMMY.emp VALUES v_rec(i);
   COMMIT;
END;
```