

Tipps & Tricks: Associative Arrays

Bereich:	PL/SQL	Erstellung:	10/2002 HA
Versionsinfo:	9.2, 10.2, 11.1	Letzte Überarbeitung:	05/2009 HA

Associative Arrays

Index By-Tabellen, auch PL/SQL-Tabellen genannt, sind der einfachste **Collection**-Typ unter Oracle. Mit Version 9.2 wurden sie erneut umbenannt, diesmal zu Associative Arrays. Der Anlass dafür war, dass weitere Datentypen als Key des Arrays zulässig sind neben `BINARY_INTEGER`. Die Syntax wurde entsprechend erweitert.

Syntax:

```
TYPE typename IS TABLE OF datatype_der_elemente
    INDEX BY BINARY_INTEGER | PLS_INTEGER | VARCHAR2(groesse);
```

Neben `VARCHAR2` sind auch seine Subtypen zulässig, also `STRING` und `LONG`.

Umgang mit Associative Arrays

Im Umgang mit solchen Arrays hat sich prinzipiell nichts geändert, Collections-Methoden (`EXISTS`, `COUNT`, `FIRST`, `LAST`, `PRIOR`, `NEXT`) sind nach wie vor anwendbar.

Die Sortierung der Einträge - also der Rückgabewert z.B. bei `FIRST` und `LAST` - hängt bei `VARCHAR2`-Keys von der Einstellung von `NLS_COMP` in der aktuellen Session ab: steht dieser Parameter auf `BINARY` (was üblich ist), so werden die Einträge entsprechend dem Binärwert der Keys sortiert; wird `NLS_COMP` umgestellt auf `ANSI`, so orientiert sich die Reihenfolge an `NLS_SORT`, und das heisst in der Regel alphabetische Sortierung unter (weitgehender) Vernachlässigung von Groß-/Kleinschreibung. Als Wert für den Key ist prinzipiell alles zulässig, was sich in `VARCHAR2` umwandeln lässt.

Beispiel:

```
CREATE OR REPLACE PROCEDURE ASSOCIATIVE
IS
    TYPE v_type IS TABLE OF VARCHAR2(2000)
        INDEX BY VARCHAR2(20);
    v_array v_type;
    v_index VARCHAR2(2000);
BEGIN
    v_array('atest') := 'aaaa';
    v_array('xtest') := 'xxxx';
    v_array('Atest') := 'AAAA';
    v_array('btest') := 'bbbb';
    v_array('Ctest') := 'CCCC';
    v_array('ltest') := '1111';
    v_array('') := 'Null';
    DBMS_OUTPUT.PUT_LINE('Anzahl: ' || v_array.COUNT);
    IF v_array.EXISTS('btest') THEN
```

```
        DBMS_OUTPUT.PUT_LINE('gefunden!');
    END IF;
    v_index := v_array.FIRST;
    DBMS_OUTPUT.PUT_LINE('1. Wert: ' || v_array(v_index));
    LOOP
        v_index := v_array.NEXT(v_index);
        EXIT WHEN v_index IS NULL;
        DBMS_OUTPUT.PUT_LINE('nächster Wert: ' || v_array(v_index));
    END LOOP;
END;
/
SET SERVEROUTPUT ON
EXEC ASSOCIATIVE
ALTER SESSION SET NLS_COMP = 'ANSI';
EXEC ASSOCIATIVE
```

Ausgabe:

```
Anzahl: 7
gefunden!
1. Wert: Null
nächster Wert: 1111
nächster Wert: AAAA
nächster Wert: CCCC
nächster Wert: aaaa
nächster Wert: bbbb
nächster Wert: xxxx
```

```
Anzahl: 7
gefunden!
1. Wert: Null
nächster Wert: aaaa
nächster Wert: AAAA
nächster Wert: bbbb
nächster Wert: CCCC
nächster Wert: xxxx
nächster Wert: 1111
```