

Tipps & Tricks: CASE

Bereich:	PL/SQL	Erstellung:	07/2001 HA
Versionsinfo:	9.2 10.2 11.1	Letzte Überarbeitung:	05/2009 BK

CASE Fallunterscheidung in PL/SQL

Im Gegensatz zu anderen Programmiersprachen gab es bisher in PL/SQL abgesehen von IF...ELSE-Konstrukten keine Fallunterscheidungen. Diese Lücke wurde nun mit CASE geschlossen.

Syntax 1 (mit Selektor):

```
CASE variable
    WHEN wert1 | ausdruck1 THEN anweisung(en) 1;
    WHEN wert2 | ausdruck2 THEN anweisung(en) 2;
    ....
    [ ELSE anweisung(en) ; ]
END CASE ;
```

Syntax 2 (ohne Selektor):

```
CASE
    WHEN bedingung1 THEN anweisung(en) 1;
    WHEN bedingung2 THEN anweisung(en) 2;
    ....
    [ ELSE anweisung(en) ; ]
END CASE ;
```

Bei Syntax 1 sind auch Ausdrücke oder Funktionsaufrufe als Selektor erlaubt. Die einzigen nicht zulässigen Datentypen sind BLOB, BFILE, Record, Index-by-Tables, Nested Tables, Varrays und Objekttypen.

Es wird immer nur ein Zweig der Fallunterscheidung durchlaufen. Falls keine ELSE-Klausel angegeben wird, wird implizit ergänzt: ELSE RAISE CASE_NOT_FOUND;

CASE ist auch zulässig als Teil eines Ausdrucks in der Form (hier nur mit Selektor gezeigt):

```
wert :=
CASE variable
    WHEN wert1 | ausdruck1 THEN ergebnis1
    WHEN wert2 | ausdruck2 THEN ergebnis2
    ....
    [ ELSE ergebnis ; ]
END ;
```

Wird hier kein ELSE-Zweig angegeben, und keiner der WHEN-Zweige trifft zu, so wird NULL zurückgegeben.
Beispiel:

```
declare
gehaltserhoehung      EMP.SAL%TYPE;
    v_job              EMP.JOB%TYPE;
begin
```

```
SELECT job INTO v_job FROM emp WHERE empno = &empno ;
gehaltserhoehung := CASE v_job
    WHEN 'PRESIDENT' THEN 1
    WHEN 'SALESMAN' THEN 1.07
    WHEN 'MANAGER' THEN 1.05
    WHEN 'CLERK' THEN 1.1
    WHEN 'ANALYST' THEN 1.04
    ELSE 0
END ;
UPDATE emp SET sal = sal* gehaltserhoehung WHERE empno = &empno ;
end;
/
```