Empfohlene Namenskonventionen und Codierungsvorgaben für PL/SQL im Praktikum Datenbanksysteme

Fortsetzung von

“Empfohlene Namenskonventionen und Codierungsvorgaben für SQL im Praktikum Datenbanksysteme”

Autoren: Jens Lambert, Björn Salgert, Mareike Focken und Thomas C. Rakow

3. Namenskonventionen für PL/SQL

Grundsätzlich basieren die Namenskonventionen für PL/SQL auf denen von SQL (Teil 1)

Darüber hinaus gilt:

(6) Schreiben Sie Bezeichner von Kollektionen immer im Plural.

Beispiel: Kollektion: `hobbies`

(11) Bezeichner für Prozeduren beschreiben immer, was die Prozedur tut.

Beispiel: `calculate_salary`

(12) Bezeichner für Funktionen beschreiben immer das Ergebnis, das zurückgegeben wird (eine Funktion soll keine Seiteneffekte verursachen).

Beispiel: `employee_by_id`

Bezeichnung	Prefix	Suffix	Beispiel
PL/SQL			
Lokale Variable	`l_`	-	`l_name`
Constante	`co_`	-	`co_pi`
Cursor	`c_`	-	`c_employees`
Record	`r_`	-	`r_employee`
Table	`t_`	-	`t_employees`
Parameter	`p_`	-	`p_empno`
IN Parameter	`in_`		`in_salary`
OUT Parameter	`out_`		`out_salary`
IN-OUT Parameter	`io_`		`io_salary`
Exception	`e_`	-	`e_employee_exists`
Trigger	`<table>_`	`_trg`	`emp_salary_trg`

Ein Kooperationsvorhaben empfohlen durch die:
DIGITALE HOCHSCHULE NRW
Gefördert durch:
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

4. Codierungsvorgaben für PL/SQL

Grundsätzlich basieren die Codierungsvorgaben für PL/SQL auf denen von SQL (Teil 2)

Beispiel für PL/SQL:

```
PROCEDURE set_salary(in_employee_id IN employees.employee_id%TYPE) IS
    CURSOR c_employees(p_employee_id IN employees.employee_id%TYPE) IS
        SELECT last_name, first_name,
            salary
        FROM employees
        WHERE employee_id = p_employee_id
        ORDER BY last_name, first_name;

    r_employee c_employees%ROWTYPE;
    l_new_salary employees.salary%TYPE;
BEGIN

    OPEN c_employees(p_employee_id => in_employee_id);
    FETCH c_employees INTO r_employee;
    CLOSE c_employees;

    new_salary (
        in_employee_id => in_employee_id,
        out_salary => l_new_salary
    );

    -- Check whether salary has changed
    IF r_employee.salary <> l_new_salary THEN
        UPDATE employees
        SET salary = l_new_salary
        WHERE employee_id = in_employee_id;
    END IF;

END set_salary;
```


Ein Kooperationsvorhaben empfohlen durch die:
DIGITALE HOCHSCHULE NRW
Gefördert durch:
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen