

Merkblatt Datenbankprogrammierung (Oracle)

Funktion erstellen und aufrufen

```
DROP FUNCTION VerdoppelungFunction;

CREATE OR REPLACE FUNCTION VerdoppelungFunction(zahl int) RETURN Integer
IS
    rueckgabe INTEGER DEFAULT 0;          -- Variable definieren und initialisieren
BEGIN
    rueckgabe := 2*zahl;                  -- Eingabewert mit 2 Multiplizieren
    RETURN rueckgabe;
END;
/
```

Aufruf der gespeicherten Funktion: `SELECT VerdoppelungFunction(2) FROM dual;`

Prozedur erstellen und aufrufen

```
DROP PROCEDURE VerdoppelungProcedure;

CREATE OR REPLACE PROCEDURE VerdoppelungProcedure (zahl IN int)
AS
    doppelteZahl INTEGER;
BEGIN
    SELECT VerdoppelungFunction(zahl) INTO doppelteZahl FROM dual;
    DBMS_OUTPUT.put_line ('Das Doppelte von'||zahl||' ist'|| doppelteZahl);
END;
/
```

Konsolenausgabe einschalten: `SET SERVEROUTPUT on;`

Aufruf der gespeicherten Prozedur: `CALL VerdoppelungProcedure(2);`

Trigger

```
DROP Trigger NeuerArtikelTrigger;

CREATE OR REPLACE Trigger NeuerArtikelTrigger
BEFORE INSERT ON Artikel
FOR EACH ROW
DECLARE
    PreisNegativ EXCEPTION;
BEGIN
    IF (:new.Preis<0)
    THEN
        RAISE PreisNegativ;
    END IF;
EXCEPTION
    WHEN PreisNegativ
    THEN raise_application_error (-20501, 'Artikel '||:new.Artikelnummer||': Preis<0');
END;
/
```

Impliziter Aufruf des Triggers: `INSERT INTO Artikel VALUES (...);`

Variablen

Deklaration und Initialisierung: `DECLARE var1 INTEGER :=0;`
Wertzuweisung: `var1 := 0;`
Wertzuweisung durch Select: `SELECT COUNT(*) INTO var1 FROM Kunde;`

Fallunterscheidungen

```
DECLARE rueckgabe CHAR(80);
IF anrede ='F' THEN
    rueckgabe := 'Sehr geehrte Frau ' || name;
ELSEIF anrede='H' THEN
    SET rueckgabe = 'Sehr geehrter Herr ' || name;
ELSE SET rueckgabe = 'Sehr geehrte Damen und Herren';
END IF;

CASE anrede
    WHEN 'F' THEN rueckgabe := 'Sehr geehrte Frau ';
    WHEN 'H' THEN rueckgabe := 'Sehr geehrter Herr ';
    ELSE rueckgabe:= 'Sehr geehrte Damen und Herren';
END CASE;
```

Schleifen

WHILE-Schleife: `WHILE i<10 LOOP`
 `i := i+1;`
 `END LOOP;`

REPEAT-Schleife: `LOOP`
 `i := i+1;`
 `EXIT WHEN i=10;`
 `END LOOP;`

FOR-Schleife: `FOR i IN 1..10 LOOP`
 `END LOOP;`

Cursor

```
CREATE OR REPLACE FUNCTION Kundenliste RETURN VARCHAR2
IS
    kundenname      VARCHAR(20);
    liste           VARCHAR(2000) := "";
    CURSOR kundencursor IS SELECT Nachname FROM Kunde;
BEGIN
    FOR k in kundencursor LOOP
        kundenname:=TRIM(k.nachname);
        liste := liste || kundenname;
    END LOOP;
    RETURN liste;
END;
/
```