



Packages - Basis

PL/SQL Package

- Ein Package ist ein Schema-Objekt, das Typen, Variablen und Unterprogramme zu einer logischen Einheit zusammenfasst (Bibliothek).
- Ein Package besteht aus 2 Teilen:
 - Einer Spezifikation
 - Einem Body (Implementierung der Spezifikation)
- Die Spezifikation stellt die Schnittstelle zum Paket dar. Hier werden die Datentypen, Variablen, Konstanten, Ausnahmen, Cursor und Unterprogramme deklariert, die von aussen zugegriffen werden können.
- Der Body definiert die Abfragen für die Cursor und den Code der Subprogramme.
- Die gesamte Paket wird einmal in den Speicher geladen.

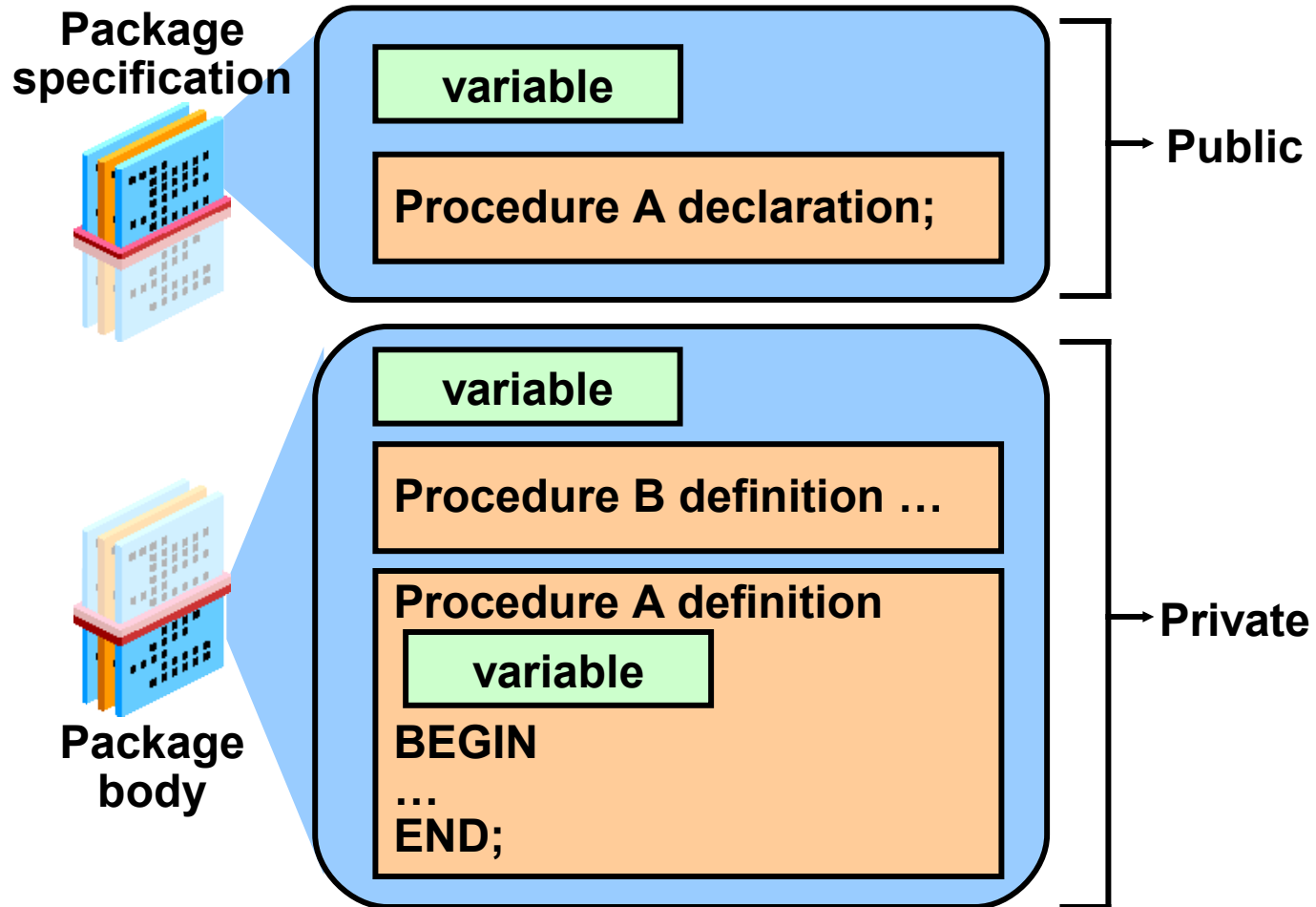
Vorteile von Packages 1

- Modularität und Kapselung
- Logisch abhängige Dinge werden gemeinsam verwaltet
- Verbessertes Design durch die Schnittstelle
- Information Hiding:
 - Nur die Deklarationen der Spezifikation sind extern sichtbar und nutzbar
 - Private Elemente im Body sind nicht von aussen zugreifbar
 - Die gesamte Codierung ist versteckt

Vorteile von Packages 2

- Persistenz von öffentlichen Variablen und Cursor
- Überladen (overloading) von Prozeduren und Funktionen ist möglich
- Performanz:
 - Das gesamte Package wird bei der ersten Referenzierung geladen
 - Es ist nur eine Kopie im Speicher für alle Nutzer
 - Die Abhängigkeitsverwaltung durch den Server wird vereinfacht

Komponenten eines PL/SQL Package



Erzeugen der Package Spezifikation

```
CREATE [OR REPLACE] PACKAGE package_name IS|AS  
    public type and variable declarations  
    subprogram specifications  
END [package_name];
```

- Variablen in der Spezifikation werden standardmäßig mit NULL initialisiert.
- Alle Konstrukte der Spezifikation sind sichtbar für Nutzer mit den entsprechenden Privilegien.

Beispiel: Package comm_pkg

```
CREATE OR REPLACE PACKAGE comm_pkg IS
  v_std_comm NUMBER := 0.10;  --initialized to 0.10
  PROCEDURE reset_comm(p_new_comm NUMBER);
END comm_pkg;
/
```

- V_STD_COMM ist eine öffentliche Variable, initialisiert mit 0.10.
- RESET_COMM ist eine öffentliche Prozedur, deren Implementierung im Body definiert wird.

Erzeugung des Package Body

```
CREATE [OR REPLACE] PACKAGE BODY package_name IS|AS
    private type and variable declarations
    subprogram bodies
    [BEGIN initialization statements]
END [package_name];
```

- Im Body definierte Bezeichner sind privat und nicht öffentlich sichtbar.
- Alle privaten Elemente müssen definiert werden, bevor sie genutzt werden können.
- Öffentliche Elemente (Spezifikation) sind hier nutzbar.

Beispiel des Package Body: comm_pkg

```
CREATE OR REPLACE PACKAGE BODY comm_pkg IS
  FUNCTION validate(p_comm NUMBER) RETURN BOOLEAN IS
    v_max_comm      employees.commission_pct%type;
  BEGIN
    SELECT MAX(commission_pct) INTO v_max_comm
    FROM   employees;
    RETURN (p_comm BETWEEN 0.0 AND v_max_comm);
  END validate;

  PROCEDURE reset_comm (p_new_comm NUMBER) IS BEGIN
    IF validate(p_new_comm) THEN
      v_std_comm := p_new_comm; -- reset public var
    ELSE  RAISE_APPLICATION_ERROR(
          -20210, 'Bad Commission');
    END IF;
  END reset_comm;
END comm_pkg;
```

Nutzen des Package: Beispiel

```
-- Invoke a function within the same packages:  
CREATE OR REPLACE PACKAGE BODY comm_pkg IS ...  
  PROCEDURE reset_comm(p_new_comm NUMBER) IS  
  BEGIN  
    IF validate(p_new_comm) THEN  
      v_std_comm := p_new_comm;  
    ELSE ...  
    END IF;  
  END reset_comm;  
END comm_pkg;
```

```
-- Invoke a package procedure from SQL*Plus:  
EXECUTE comm_pkg.reset_comm(0.15)
```

```
-- Invoke a package procedure in a different schema:  
EXECUTE scott.comm_pkg.reset_comm(0.15)
```

Packages ohne Body

```
CREATE OR REPLACE PACKAGE global_consts IS
    c_mile_2_kilo      CONSTANT  NUMBER  :=  1.6093;
    c_kilo_2_mile      CONSTANT  NUMBER  :=  0.6214;
    c_yard_2_meter     CONSTANT  NUMBER  :=  0.9144;
    c_meter_2_yard     CONSTANT  NUMBER  :=  1.0936;
END global_consts;
```

```
BEGIN  DBMS_OUTPUT.PUT_LINE('20 miles = ' ||
    20 * global_consts.c_mile_2_kilo || ' km');
END;
```

```
CREATE FUNCTION mtr2yrd(p_m NUMBER) RETURN NUMBER IS
BEGIN
    RETURN (p_m * global_consts.c_meter_2_yard);
END mtr2yrd;
/
EXECUTE DBMS_OUTPUT.PUT_LINE(mtr2yrd(1))
```

Removing Packages

```
-- Remove the package specification and body  
DROP PACKAGE package_name;
```

```
-- Remove the package body only  
DROP PACKAGE BODY package_name;
```

Packages im Data Dictionary

```
-- View the package specification.  
SELECT text  
FROM    user_source  
WHERE   name = 'COMM_PKG' AND type = 'PACKAGE';
```

```
-- View the package body.  
SELECT text  
FROM    user_source  
WHERE   name = 'COMM_PKG' AND type = 'PACKAGE BODY';
```

Richtlinien

- Packages sollten für möglichst allgemeine Nutzung entwickelt werden.
- Zuerst die Spezifikation, dann die Implementierung.
- Die Spezifikation enthält nur die Teile, die auch öffentlich sichtbar sein sollen.
- Elemente, die innerhalb eine Session oder Transaktion beibehalten werden sollen, gehören in den Deklarationsteil des Body.
- Die Spezifikation sollte so wenig Elemente, wie möglich enthalten.