

Database 3

الحزم البرمجية

Packages

المحاضرة الثانية

الحزم البرمجية Packages

- يمكن أن تحتوي الحزمة البرمجية على
 - مؤشرات و متحولات و استثناءات اسمية و إجراءات و توابع
- توصيف الحزمة (الواجهة)
 - يحتوي على أسماء أغراض الحزمة العامة
 - لا يحتوي على التعليمات التفصيلية لهذه الأغراض
- جسم الحزمة
 - يحتوي على التوصيفات التفصيلية لكل أغراض الحزمة
 - يحتوي على الأغراض التي لم يتم تعريفها ضمن توصيف الحزمة (الأغراض الخاصة)

الحزم البرمجية Packages

توصيف الحزمة

```
CREATE OR REPLACE PACKAGE package_name IS  
    object_type object_name (parameters) RETURN datatype;  
    object_type object_name (parameters) RETURN datatype;  
END package_name ;
```

جسم الحزمة

```
CREATE OR REPLACE BODY PACKAGE package_name IS  
    object_definition;  
    object_definition;  
END package_name ;
```

Defining Package Specification Syntax

```
1 CREATE [OR REPLACE] PACKAGE package_name
2     IS | AS
3     [variable_declaration ...]
4     [constant_declaration ...]
5     [exception_declaration ...]
6     [cursor_specification ...]
7     [PROCEDURE [Schema..] procedure_name
8         [ (parameter {IN,OUT,IN OUT} datatype [,parameter]) ]
9     ]
10    [FUNCTION [Schema..] function_name
11        [ (parameter {IN,OUT,IN OUT} datatype [,parameter]) ]
12        RETURN return_datatype
13    ]
14 END [package_name];
```

Creating Package Body Syntax

```
1 CREATE [OR REPLACE] PACKAGE BODY package_name
2     IS | AS
3     [private_variable_declaration ...]
4     [private_constant_declaration ...]
5     BEGIN
6         [initialization_statement]
7         [PROCEDURE [Schema..] procedure_name
8             [ (parameter [,parameter]) ]
9             IS | AS
10                variable declarations;
11                constant declarations;
12            BEGIN
13                statement(s);
14            EXCEPTION
15                WHEN ...
16            END
17        ]
18        [FUNCTION [Schema..] function_name
19            [ (parameter [,parameter]) ]
20            RETURN return_datatype
21            IS | AS
22                variable declarations;
23                constant declarations;
24            BEGIN
25                statement(s);
26            EXCEPTION
27                WHEN ...
28            END
29        ]
30        [EXCEPTION
31            WHEN built-in_exception_name_1 THEN
32                User defined statement (action) will be taken;
33        ]
34    END;
35    /
```

مثال – توصيف الحزمة

```
CREATE OR REPLACE PACKAGE employee_objects IS
    CURSOR list_employees RETURN emp%ROWTYPE;
    FUNCTION age (birth_date IN DATE) RETURN NUMBER;
    PROCEDURE add_employee(id INTEGER, name VARCHAR2, salary NUMBER);
END employee_objects;
/
```

مثال - جسم الحزمة

```
CREATE OR REPLACE PACKAGE BODY employee_objects IS
  CURSOR list_employees RETURN emp%ROWTYPE IS
    SELECT * FROM emp;
  FUNCTION age (birth_date IN DATE) RETURN NUMBER IS
    age NUMBER;
  BEGIN
    age := TRUNC(MONTHS_BETWEEN(CURRENT_DATE, birth_date)/12, 0);
    RETURN age;
  END age;
  PROCEDURE add_employee(id INTEGER, name VARCHAR2, salary NUMBER) IS
  BEGIN
    INSERT INTO emp (empno, ename, sal)
    VALUES (id, name, salary);
  END add_employee;
END employee_objects;
/
```

مثال – استدعاء أغراض الحزمة

```
DECLARE
    v_age NUMBER;
BEGIN
    FOR c IN employee_objects.list_employees LOOP
        v_age := employee_objects.age(c.birth_date);
        DBMS_OUTPUT.PUT_LINE(c.ename||' '||v_age);
    END LOOP;
    employee_objects.add_employee(7889,'Mill Gibson',1750);
END;
/
```

Example:

```
Select * from customers;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	3000.00
2	Khilan	25	Delhi	3000.00
3	kaushik	23	Kota	3000.00
4	Chaitali	25	Mumbai	7500.00
5	Hardik	27	Bhopal	9500.00
6	Komal	22	MP	5500.00

THE PACKAGE SPECIFICATION:

```
CREATE OR REPLACE PACKAGE c_package AS
  -- Adds a customer
  PROCEDURE addCustomer(c_id customers.id%type,
    c_name customers.name%type,
    c_age customers.age%type,
    c_addr customers.address%type,
    c_sal customers.salary%type);

  -- Removes a customer
  PROCEDURE delCustomer(c_id customers.id%TYPE);
  -- Lists all customers
  PROCEDURE listCustomer;

END c_package;
/
```

CREATING THE PACKAGE BODY:

```
CREATE OR REPLACE PACKAGE BODY c_package AS
  PROCEDURE addCustomer(c_id customers.id%type,
    c_name customers.name%type,
    c_age customers.age%type,
    c_addr customers.address%type,
    c_sal customers.salary%type)
  IS
  BEGIN
    INSERT INTO customers (id,name,age,address,salary)
      VALUES(c_id, c_name, c_age, c_addr, c_sal);
  END addCustomer;

  PROCEDURE delCustomer(c_id customers.id%type) IS
  BEGIN
    DELETE FROM customers
      WHERE id = c_id;
  END delCustomer;

  PROCEDURE listCustomer IS
  CURSOR c_customers is
    SELECT name FROM customers;
  TYPE c_list is TABLE OF customers.name%type;
  name_list c_list := c_list();
  counter integer :=0;
  BEGIN
    FOR n IN c_customers LOOP
      counter := counter +1;
      name_list.extend;
      name_list(counter) := n.name;
      dbms_output.put_line('Customer(' ||counter|| ')'||name_list(counter));
    END LOOP;
  END listCustomer;
END c_package;
/
```

Using The Package

```
DECLARE
    code customers.id%type:= 8;
BEGIN
    c_package.addcustomer(7, 'Rajnish', 25, 'Chennai', 3500);
    c_package.addcustomer(8, 'Subham', 32, 'Delhi', 7500);
    c_package.listcustomer;
    c_package.delcustomer(code);
    c_package.listcustomer;
END;
/
```

```
Customer(1): Ramesh
Customer(2): Khilan
Customer(3): kaushik
Customer(4): Chaitali
Customer(5): Hardik
Customer(6): Komal
Customer(7): Rajnish
Customer(8): Subham
Customer(1): Ramesh
Customer(2): Khilan
Customer(3): kaushik
Customer(4): Chaitali
Customer(5): Hardik
Customer(6): Komal
Customer(7): Rajnish
```