More Package Concepts

1. 오버로드 (Overload)

1-1. 개요

- 패키지의 여러 서브 프로그램에서 동일한 이름 사용 가능
- 서브 프로그램의 형식 매개변수 개수, 순서, 데이터 유형 계열은 달라야 함
- 융통성을 더 많이 제공하므로 사용자 또는 응용 프로그램이 특정 데이터 유형 또는 형식 매개변수의 수에 제한을 받지 않음
- 제한사항
  - 로컬 또는 패키지 서브 프로그램만 오버로드 가능
  - 두 서브 프로그램에서 형식 매개변수의 이름 또는 매개변수 모드만 다른 경우 오버로드 불가능
  - 두 서브 프로그램에서 형식 매개변수의 데이터 유형만 다르고 그 데이터 유형이 동일한 계열에 속하는 경우 오버로드 불가능
  - 두 서브 프로그램에서 형식 매개변수의 서브 유형만 다르고 그 서브 유형이 동일한 계열의 유형을 기반으로 하는 경우 오버로드 불가능
  - 두 함수에서 반환 유형만 다른 경우 오버로드 불가능

1-2. 오버로드

```
CREATE OR REPLACE PACKAGE over_pack
IS
       PROCEDURE add_dept
         (v_deptno   IN dept.deptno%TYPE,
          v_name     IN dept.dname%TYPE DEFAULT 'unknown',
          v_loc      IN dept.loc%TYPE   DEFAULT 'unknown');
       PROCEDURE add_dept
         (v_name     IN dept.dname%TYPE DEFAULT 'unknown',
          v_loc      IN dept.loc%TYPE   DEFAULT 'unknown');
END over_pack;
```


1-2. 오버로드 (계속)

```
CREATE OR REPLACE PACKAGE BODY over_pack
IS
       PROCEDURE add_dept
         (v_deptno   IN dept.deptno%TYPE,
          v_name     IN dept.dname%TYPE DEFAULT 'unknown',
          v_loc      IN dept.loc%TYPE   DEFAULT 'unknown')
       IS
       BEGIN
         INSERT INTO dept
         VALUES (v_deptno, v_name, v_loc);
       END add_dept;
……
```

1-2. 오버로드 (계속)

```
……
        PROCEDURE add_dept
          (v_name     IN dept.dname%TYPE DEFAULT 'unknown',
           v_loc      IN dept.loc%TYPE   DEFAULT 'unknown')
        IS
        BEGIN
           INSERT INTO dept
           VALUES (99, v_name, v_loc);
        END add_dept;
END over_pack;
```


Oracle SQL Structured Query Language
Creating Packages

1-2. 오버로드 (계속)

```
SQL> EXECUTE over_pack.add_dept (77, '교육', '대구')
SQL> EXECUTE over_pack.add_dept ('인사', '광주')
```

2. 사용자 정의 패키지

2-1. 사용자 정의 패키지 생성

```
CREATE OR REPLACE PACKAGE taxes_pack
IS
       FUNCTION tax
         (p_value    IN NUMBER)
       RETURN NUMBER;
END taxes_pack;
/
```


2-1. 사용자 정의 패키지 생성 (계속)

```
CREATE OR REPLACE PACKAGE BODY taxes_pack
IS
       FUNCTION tax
         (p_value    IN NUMBER)
          RETURN NUMBER
       IS
          v_rate NUMBER := 0.08;
       BEGIN
          RETURN (p_value * v_rate);
       END tax;
END taxes_pack;
/
```

2-2. 사용자 정의 패키지 함수 호출

```
SELECT taxes_pack.tax(sal), sal, ename
FROM   emp;
```


3. 지속 상태 (Persistent State)

3-1. 패키지 커서

```
CREATE OR REPLACE PACKAGE pack_cur
IS
       CURSOR c1
       IS
         SELECT empno
         FROM   emp
         ORDER BY empno DESC;
       PROCEDURE proc1_3rows;
       PROCEDURE proc4_6rows;
END pack_cur;
/
```

3-1. 패키지 커서 (계속)

```
CREATE OR REPLACE PACKAGE BODY pack_cur
IS
       v_empno NUMBER;
       PROCEDURE proc1_3rows
       IS
       BEGIN
         open c1;
         LOOP
           FETCH c1 INTO v_empno;
           DBMS_OUTPUT.PUT_LINE ('Id : ' || (v_empno));
           EXIT WHEN c1%ROWCOUNT >= 3;
         END LOOP;
       END proc1_3rows;
……
```


3-1. 패키지 커서 (계속)

```
……
       PROCEDURE proc4_6rows
       IS
       BEGIN
         LOOP
           FETCH c1 INTO v_empno;
           DBMS_OUTPUT.PUT_LINE ('Id : ' || (v_empno));
           EXIT WHEN c1%ROWCOUNT >= 6;
         END LOOP;
       CLOSE c1;
       END proc4_6rows;
END pack_cur;
/
```

3-1. 패키지 커서 (계속)

```
SQL> SET SERVEROUTPUT ON
SQL> EXECUTE pack_cur.proc1_3rows
SQL> EXECUTE pack_cur.proc4_6rows
```


3-2. 패키지 PL/SQL 테이블 및 레코드

```
CREATE OR REPLACE PACKAGE emp_package
IS
       TYPE emp_table_type IS TABLE OF emp%ROWTYPE
         INDEX BY BINARY_INTEGER;
       PROCEDURE read_emp_table
         (emp_table  OUT emp_table_type);
END emp_package;
/
```

3-2. 패키지 PL/SQL 테이블 및 레코드 (계속)

```
CREATE OR REPLACE PACKAGE BODY emp_package
IS
       PROCEDURE read_emp_table
         (emp_table  OUT emp_table_type)
       IS
         i BINARY_INTEGER := 0;
       BEGIN
         FOR emp_record IN (SELECT * FROM emp)
         LOOP
           emp_table(i) := emp_record;
           i := i+1;
         END LOOP;
       END read_emp_table;
END emp_package;
/
```


Oracle SQL Structured Query Language
Creating Packages

3-2. 패키지 PL/SQL 테이블 및 레코드 (계속)

```
DECLARE
       emp_table    emp_package.emp_table_type;
BEGIN
       emp_package.read_emp_table (emp_table);
       DBMS_OUTPUT.PUT_LINE
           ('An example: ' || emp_table(4).ename);
END;
/
```

SQL 실무