Creating and using packages, procedures and functions in Apex 5

Objectives

- **To Create a package**
- **To test the package procedures and functions**
- **To call the package procedures and functions from Oracle Apex**
- **To use DEBUG**

This tutorial is based on ORACLE documentation from the oracle technet site (http://technet.oracle.com)
See the documentation in the file: OracleHelpPackages.doc taken from the above site,
This document as been reworked for Apex.

Watch the video. Apex: Emp-Dept packaged procedures and functions.
Below is the supporting code.

1. Now inspect and run the code

scott.sql … to create necessary tables etc (see appendix A)

Create a Package
crPack.sql … to create package spec (I did this in the SQL workshop)
crPackBody.sql … (I used the object browser create package, pasted the code and compiled)

```
CREATE or replace PACKAGE emp_mgmt AS
  FUNCTION hire(ename VARCHAR2, job VARCHAR2, mgr NUMBER,
                sal NUMBER, comm NUMBER, deptno NUMBER)
    RETURN NUMBER;
  FUNCTION create_dept(dname VARCHAR2, loc VARCHAR2)
    RETURN NUMBER;
  PROCEDURE remove_emp(empno NUMBER);
  PROCEDURE remove_dept(deptno NUMBER);
  PROCEDURE increase_sal(empno NUMBER, sal_incr NUMBER);
     no_comm EXCEPTION;
    no_sal EXCEPTION;
END emp_mgmt;
```

```
CREATE OR REPLACE PACKAGE BODY emp_mgmt AS
  tot_emps  NUMBER;
  tot_depts NUMBER;

FUNCTION hire
  (ename VARCHAR2,
   job VARCHAR2,
   mgr NUMBER,
   sal NUMBER,
   comm NUMBER,
   deptno NUMBER)

RETURN NUMBER IS
  new_empno NUMBER(4);
```

```
BEGIN
  SELECT empseq.NEXTVAL
    INTO new_empno
    FROM DUAL;

  INSERT INTO emp
    VALUES (new_empno, ename, job, mgr, sysdate, sal, comm, deptno);
  RETURN(new_empno);
END;

FUNCTION create_dept(dname VARCHAR2, loc VARCHAR2)
  RETURN NUMBER IS
      new_deptno NUMBER(4);
  BEGIN
    SELECT deptseq.NEXTVAL
      INTO new_deptno
      FROM dual;
    INSERT INTO dept
      VALUES (new_deptno, dname, loc);
    RETURN(new_deptno);
  END;


PROCEDURE remove_emp(empno NUMBER) IS
  BEGIN
    DELETE FROM emp
    WHERE emp.empno = remove_emp.empno;
  END;

PROCEDURE remove_dept(deptno NUMBER) IS
  BEGIN
    DELETE FROM dept
    WHERE dept.deptno = remove_dept.deptno;
  END;


PROCEDURE increase_sal(empno NUMBER, sal_incr NUMBER) IS
  curr_sal NUMBER(7,2);
  BEGIN
    SELECT sal
    INTO curr_sal
    FROM emp
    WHERE emp.empno = increase_sal.empno;
    IF curr_sal IS NOT NULL THEN
      UPDATE emp
      SET sal = sal + sal_incr
      WHERE empno = increase_sal.empno;
    END IF;
  END;


END emp_mgmt;
```

2. Running/Testing/calling the package

Firstly test calling the functions and procedures for the package from the SQL commands.

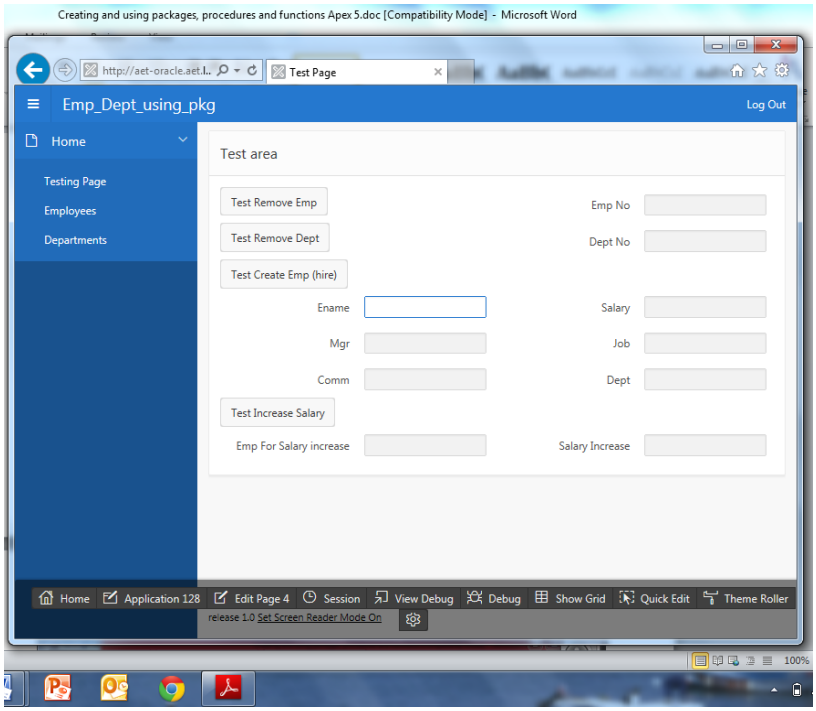
```
DECLARE
new_empno number;
BEGIN
new_empno := emp_mgmt.hire('joe', 'tester', '7499', '100', '10', '10');
END;

SELECT * from emp;

BEGIN
emp_mgmt.remove_emp('53');
END;

SELECT * from emp;

INSERT INTO dept VALUES(50,'test','leeds');

BEGIN
emp_mgmt.remove_dept(50);
END;

SELECT * FROM dept;
```

3. Call the package procedure and functions from within Apex

Now we know the procedures and functions work we are going to call it within an Oracle Apex Application.

Create a ‘test’ application:
- With a classic report to list all employees
- With a classic report department to list all departments
- With a test page to call the functions and procedures.

The test page can look like this – you can ‘build’ it up as you test each procedure.



Edit the page, Under processes add a process for each procedure. Under the PL/SQL you will not add the actual values, you will use the values that have been inputted onto the screen (the names of the text fields). For example:

```
DECLARE
new_empno number;
begin

new_empno := emp_mgmt.hire(:p4_ename, : p4_job, : p4_mgr, : p4_sal, :
p4_comm, : p4_deptno);

end;
```

Where : **p4**_ename etc is the field attribute on the page. (in example below it is P4_)
Add messages so you know it is your code being run. Ensure you choose the condition for the code to run condition of which button is pressed.

Then test from apex. This code is run when the page is submitted.

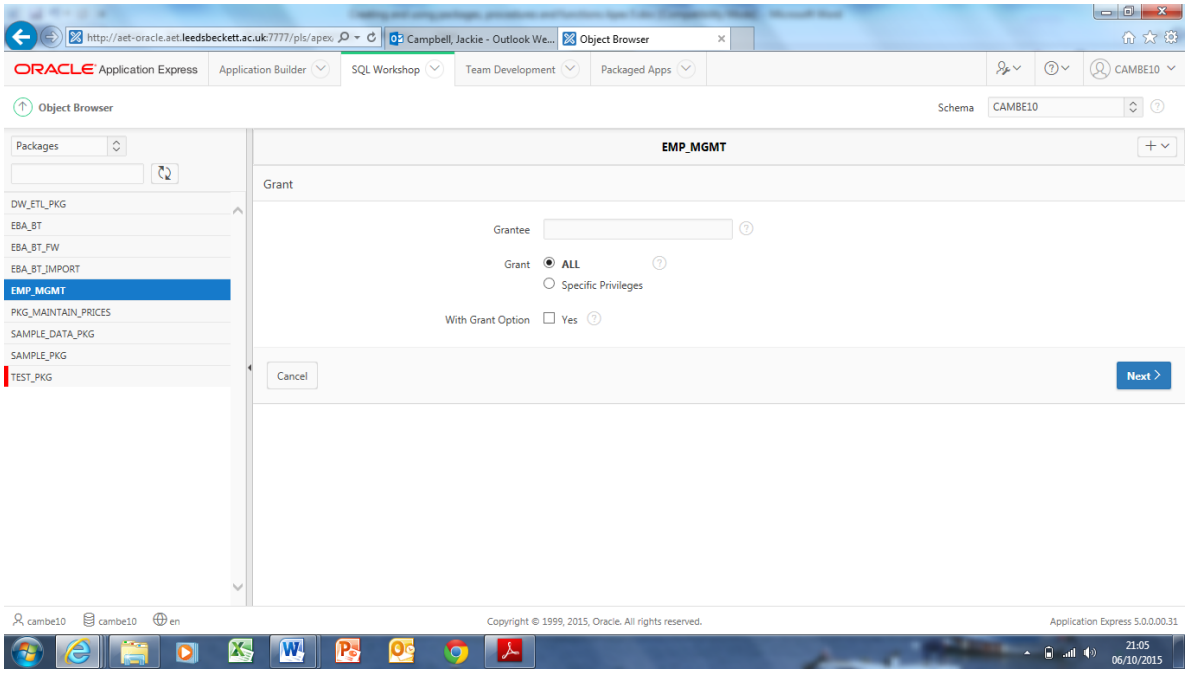
4. Outstanding issues

Need exception handling in the program units. Try entering employee without all fields. Can result in an error which is not gracefully handled.

5. Providing the package to others

Opensource!
You can grant other people access to a package you have created by:

Click on Grants, Click on Grants again – and you can give access to your colleagues! See if it works!



6. ok heres an important bit.
-- in apex it does, insert, update, create for you
-- so why have a package ...?

-- because in the package you can (and should) add the extra business specific, or constraints or security code.
-- so for example when you create an employee you also 1) check they are over 18, 2) assign them a manager 3) assign them a holiday allocation 4) check the have emergency contact details
-- and/or anything else.

… now to creating an apex application and calling this code.

Appendix A
Code from lecture: scott.sql

```
drop sequence empseq;
create sequence empseq;
drop sequence deptseq;
create sequence deptseq start with 50 increment by 10;
drop table emp cascade constraints;
create table emp as select empno, ename, job, mgr, sal, comm, deptno from scott.emp;
drop table dept cascade constraints;
create table dept as select * from scott.dept;
alter table emp add constraint emp_pk primary key (empno);
alter table dept add constraint dept_pk primary key(deptno);
alter table emp add constraint emp_dept_fk foreign key (deptno) references dept on delete cascade;
```