Talleres

Práctica

- **Ejecute el procedimiento PL/SQL**

```
BEGIN
      DBMS_OUTPUT.PUT_LINE('HOLA MUNDO');
END;
```

Práctrica

- **Ejecute el procedimiento PL/SQL**

```
BEGIN
      DBMS_OUTPUT.PUT('HOLA ');
      DBMS_OUTPUT.PUT_LINE('MUNDO');
END;
```

Práctica

- **Ejecute el procedimiento PL/SQL**

```
BEGIN
      DBMS_OUTPUT.PUT('HOLA ');
      DBMS_OUTPUT.PUT_LINE('MUNDO');

      DBMS_OUTPUT.PUT_LINE('');

      DBMS_OUTPUT.PUT_LINE(sysdate);

END;
```

Práctica

Consulte:

- **user_tab_columns**
- **cat**

Punto 1

- **Desarrolle un procedimiento PL/SQL que genere PL/SQL**

```
CREATE OR REPLACE FUNCTION inDepartments (p_department_id IN
departmenst.department_id%TYPE,
  p_department_name IN departmenst.department_name%TYPE,
  p_manager_id IN departmenst.manager_id%TYPE,
  p_location_id IN departmenst.location_id%TYPE)
RETURN NUMBER
IS
  nuRetorno NUMBER := 0;
BEGIN
  INSERT INTO departments
  VALUES (p_department_id,
          p_department_name,
          p_manager_id,
          p_location_id);
  nuRetorno := SQL%ROWCOUNT;
  RETURN nuRetorno;
END inDepartments;
```

Punto 1

Tenga en cuenta:

- **user_tab_columns**
- **Pregunte la tabla deseada**
- **Un parámetro por cada columna**
- **Es una función para insertar datos**
- **La retroalimentación es la cantidad de filas insertadas**

Punto 1

- **Posible solución (Parte 1)**

```
ACCEPT tabla PROMPT 'Ingresa la tabla : '
SET SERVEROUTPUT ON
SET FEEDBACK OFF
SET VERIFY OFF
SPOOL resultado.sql
DECLARE
    CURSOR cu_col
    IS
        SELECT t.COLUMN_NAME
        FROM user_tab_columns t
        WHERE t.table_name=Upper('&tabla');
    ctrl BOOLEAN := FALSE;
```

Punto 1

- **Posible solución (Parte 2)**

```
BEGIN
    -- Encabezado de la funcion
    DBMS_OUTPUT.PUT('CREATE OR REPLACE FUNCTION in'||Initcap('&tabla ('));
    -- Parametros para la funcion
    FOR re_col IN cu_col
    LOOP
        IF ctrl THEN
            DBMS_OUTPUT.PUT_LINE(', ');
            DBMS_OUTPUT.PUT(Chr(09));     -- Caracter tabulador
        END IF;
        DBMS_OUTPUT.PUT('p_' || Lower(re_col.COLUMN_NAME) || ' IN &tabla' || '.'
|| Lower(re_col.COLUMN_NAME) || '%TYPE');
        ctrl := TRUE;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(')');
    DBMS_OUTPUT.PUT_LINE('RETURN NUMBER');
    DBMS_OUTPUT.PUT_LINE('IS');
    DBMS_OUTPUT.PUT_LINE(Chr(09) || 'nuRetorno NUMBER := 0;');
```

Punto 1

- **Posible solución (Parte 3)**

```
    DBMS_OUTPUT.PUT_LINE('BEGIN');
    DBMS_OUTPUT.PUT_LINE(Chr(09) || 'INSERT INTO &tabla');
    DBMS_OUTPUT.PUT(Chr(09) || 'VALUES (');
    ctrl := FALSE;
    FOR re_col IN cu_col
    LOOP
        IF ctrl THEN
            DBMS_OUTPUT.PUT_LINE(', ');
            DBMS_OUTPUT.PUT(Chr(09) || Chr(09));
        END IF;
        DBMS_OUTPUT.PUT('p_' || Lower(re_col.COLUMN_NAME));
        ctrl := TRUE;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(');');
    DBMS_OUTPUT.PUT_LINE(Chr(09) || 'nuRetorno := '||'SQL%ROWCOUNT;');
    DBMS_OUTPUT.PUT_LINE(Chr(09) || 'RETURN nuRetorno;');
    DBMS_OUTPUT.PUT_LINE('END in'||Initcap('&tabla')||';');
END;
/
SPOOL OFF
QUIT
```

Punto 2

- **Desarrolle un procedimiento PL/SQL que genere PL/SQL**

```
CREATE OR REPLACE PACKAGE padDepartments
IS
      FUNCTION getDepartment_Id (p_ IN departments.%TYPE)
      RETURN departments.department_id%TYPE;
      FUNCTION getDepartment_Name (p_ IN departments.%TYPE)
      RETURN departments.department_name%TYPE;
      FUNCTION getManager_Id (p_ IN departments.%TYPE)
      RETURN departments.manager_id%TYPE;
      FUNCTION getLocation_Id (p_ IN departments.%TYPE)
      RETURN departments.location_id%TYPE;
END padDepartments;
```

Punto 2

Tenga en cuenta:

- **user_tab_columns**
- **user_cons_columns**
- **user_constraints**
- **Pregunte la tabla deseada**
- **Una función por cada columna**
- **Son funciones para consultar datos**
- **La retroalimentación es el dato mismo o null**
- **Solo el especification**

Punto 2

- **Posible solución (Parte 1)**

```
ACCEPT tabla PROMPT 'Ingresa la tabla : '
SET SERVEROUTPUT ON
SET FEEDBACK OFF
SET VERIFY OFF
SPOOL resultado.sql
DECLARE
    CURSOR cu_col
    IS
        SELECT t.COLUMN_NAME, t.DATA_TYPE
        FROM user_tab_columns t
        WHERE t.table_name=Upper('&tabla');
    CURSOR cu_pk
    IS
    SELECT col.column_name
    FROM   user_cons_columns col, user_constraints con
    WHERE  col.constraint_name = con.constraint_name
      AND  con.constraint_type = 'P'
      AND  con.table_name = Upper('&tabla');
    ctrl BOOLEAN := FALSE;
    v_pk user_cons_columns.column_name%TYPE;
```

Punto 2

- **Posible solución (Parte 2)**

```
BEGIN
    -- Encabezado del Package
    DBMS_OUTPUT.PUT_LINE('CREATE OR REPLACE PACKAGE pad'||Initcap('&tabla'));
    DBMS_OUTPUT.PUT_LINE('IS');

    -- Parametros para la funcion
    FOR re_col IN cu_col
    LOOP
        DBMS_OUTPUT.PUT(Chr(09));     -- Caracter tabulador
        DBMS_OUTPUT.PUT('FUNCTION get' || Initcap(re_col.COLUMN_NAME) || ' (');
        OPEN cu_pk;
        FETCH cu_pk INTO v_pk;
        CLOSE cu_pk;
        DBMS_OUTPUT.PUT_LINE('p_'|| InitCap(v_pk) ||' IN &tabla' || '.' || Lower(v_pk) ||
'%TYPE)');
        DBMS_OUTPUT.PUT(Chr(09));     -- Caracter tabulador
        DBMS_OUTPUT.PUT_LINE('RETURN &tabla' || '.'|| Lower(re_col.COLUMN_NAME) ||
'%TYPE;');
        DBMS_OUTPUT.PUT_LINE(' ');
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('END pad'||Initcap('&tabla')||';');
END;
/
SPOOL OFF
QUIT
```

Punto 3

- **Desarrolle un procedimiento PL/SQL que genere PL/SQL, en este caso el PACKAGE BODY del SPECIFICATION**

```
CREATE OR REPLACE PACKAGE padDepartments
IS
      FUNCTION getDepartment_Id (p_ IN departments.%TYPE)
      RETURN departments.department_id%TYPE;
      FUNCTION getDepartment_Name (p_ IN departments.%TYPE)
      RETURN departments.department_name%TYPE;
      FUNCTION getManager_Id (p_ IN departments.%TYPE)
      RETURN departments.manager_id%TYPE;
      FUNCTION getLocation_Id (p_ IN departments.%TYPE)
      RETURN departments.location_id%TYPE;
END padDepartments;
```