
 salesianos <small>COLEGIO SAN JUAN BOSCO LA CUESTA</small>	Bases de datos Tema 8		 <small>UNIVERSIDAD DE CÁDIZ</small>	Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL			5

Tarea 1: Procedure/Function en PL/SQL

1. Crear una función PL/SQL llamada **Pagos_TipoTrabajo**, que devuelva la suma de los pagos que se han realizado a aquellos empleados cuyo tipo de trabajo sea pasado por parámetro. Éste parámetro debe ser solicitado al usuario. El tipo de trabajo siempre debe existir ya que aún no conocemos como controlar los errores.

Ejemplo: Tipo de trabajo ST_MAN, suma de los pagos es de 36400€

```



CREATE OR REPLACE FUNCTION Pagos_TipoTrabajo(job1 IN
EMPLOYEES.JOB_ID%TYPE) RETURN NUMBER
IS
    v_sal NUMBER;
BEGIN
    SELECT SUM(salary) INTO v_sal
    FROM EMPLOYEES
    WHERE JOB_ID = job1;

    RETURN v_sal;
END;
/

-----

SET SERVEROUTPUT ON
ACCEPT job CHAR PROMPT 'Introduzca el tipo de trabajo.'
DECLARE
    v_job VARCHAR(20) := '&job';
    v_sal NUMBER := 0;
BEGIN
    v_sal := pagos_tipotrabajo(v_job);
    dbms_output.put_line('EL salario total del trabajo ' ||
v_job || ' es ' || v_sal);
END;
/

```

 salesianos <small>COLEGIO SAN JUAN BOSCO LA CUESTA</small>	Bases de datos Tema 8		 <small>UNIVERSIDAD DE CÁDIZ</small>	Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL			5

2. Crear un procedimiento PL/SQL llamado **sp_Formateo_Cuenta**, al cual se le pase un único parámetro que indica un número de cuenta de 22 dígitos y devuelva dicho número formateado de la siguiente manera ESXX-XXXX-XXXX-XX-XXXXXXXXXX en otro de sus parámetros.

Ejemplo: 1320900000290350000083 → ES13-2090-0000-29-0350000083

```



CREATE OR REPLACE PROCEDURE sp_Formateo_Cuenta( num1 IN
VARCHAR2, num2 OUT VARCHAR2)
AS
BEGIN
    num2 := 'ES' || SUBSTR(num1,1,4) || '-' ||
SUBSTR(num1,5,4) || '-' || SUBSTR(num1,9,4) || '-' ||
SUBSTR(num1,13,2) || '-' || SUBSTR(num1,15);

END;
/

-----

SET SERVEROUTPUT ON
DECLARE
    v_num VARCHAR2(22) := '1320900000290350000083';
    v_fnum VARCHAR2(30);
BEGIN
    sp_formateo_cuenta(v_num, v_fnum);
    dbms_output.put_line(v_num || ' -> ' || v_fnum);
END;
/

```

	Bases de datos Tema 8		Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL		5

3. Crear un procedimiento PL/SQL que realice un duplicado de una tabla de la cual su nombre será pasado como uno de los parámetros, este procedimiento se llamará **Copy_Table**. El procedimiento recibirá dos parámetros, el primero parámetro se llamará **TablaOrigen** que hace referencia a la tabla a copiar y como segundo parámetro será **TablaDestino** que indica el nombre de la tabla a crear. Mostrar los datos de la tabla destino desde el programa principal. El valor por defecto para la **TablaDestino** es *Copy*.

Ejemplo: Copy_Table('Jobs', 'Ejemplo'); → Se creará la tabla Ejemplo con la estructura y datos de la tabla Jobs

```

CREATE OR REPLACE PROCEDURE Copy_Table(
    TablaOrigen IN VARCHAR2,
    TablaDestino IN VARCHAR2 DEFAULT "Copy"
)
AS
    v_Columnas VARCHAR2(4000);
    v_ExisteTabla NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_ExisteTabla
    FROM all_tables
    WHERE table_name = UPPER(TablaDestino);



    IF v_ExisteTabla = 0 THEN
        EXECUTE IMMEDIATE 'CREATE TABLE ' || TablaDestino ||
        ' AS SELECT * FROM ' || TablaOrigen;
    END IF;

    SELECT LISTAGG(column_name, ', ') WITHIN GROUP (ORDER BY
column_id)
    INTO v_Columnas
    FROM all_tab_columns
    WHERE table_name = UPPER(TablaOrigen);

    EXECUTE IMMEDIATE 'INSERT INTO ' || TablaDestino || ' ('
|| v_Columnas || ')
                        SELECT ' || v_Columnas || '
                        FROM ' || TablaOrigen;

    EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001, 'Fallo al copiar los
datos: ' || SQLERRM);
END;
/

```



 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 8		 EDUCATIVA CONSEJO REGULADOR	Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL			5

```

SET SERVEROUTPUT ON
DECLARE

BEGIN
    copy_table('Jobs', 'Ejemplo');
END;
/
SELECT * FROM Ejemplo;

```

	Bases de datos Tema 8		Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL		5

4. Diseñar un procedimiento PL/SQL llamado **sp_conver_temp**, que convierta una temperatura en la escala de Fahrenheit a Celsius y viceversa. Dicho procedimiento recibe los siguientes parámetros:
- **temperatura** del tipo de entrada y que indica la temperatura a convertir.
 - **escala** del tipo de entrada y que indica en qué escala se encuentra el parámetro temperatura. Este parámetro debe ser controlado y solo podrá recibir o una 'C' o una 'F' para indicar Celsius o Fahrenheit.
 - **newTemp** de tipo salida, y será el valor obtenido tras la conversión.
 - **NewEscala** de tipo salida, será la escala a la que fue convertida la temperatura pasada como parámetro.

Si se pasa como parámetro una letra diferente a las especificadas deberá mostrar un mensaje indicando que la escala no es válida y como parámetros de salida serán *newtemp* será **0** y *newEscala* será igual a **C**.

Ejemplos de llamadas desde el programa principal:

sp_conver_temp(30, 'C', nT, nE) → nT = 86 , nE = F



sp_conver_temp(1, 'F', nT, nE) → nT = -17,2222 , nE = C

sp_conver_temp(30, 'X', nT, nE) → nT = 0 , nE = C

```

CREATE OR REPLACE PROCEDURE sp_conver_temp(temp IN NUMBER,
escala IN VARCHAR2, newTemp OUT NUMBER, newEscala OUT
VARCHAR2 )
AS
BEGIN
    IF escala = 'C' THEN
        newTemp := (temp * 9 / 5) + 32 ;
        newEscala := 'F';
    ELSIF escala = 'F' THEN
        newTemp := (temp - 32) * 5 / 9;
        newEscala := 'C';
    ELSE
        newTemp := 0;
        newEscala := 'C';
    END IF;
END;
/



```

 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 8		 UNIVERSIDAD DE CÁDIZ	Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL			5

```

SET SERVEROUTPUT ON
DECLARE
  a NUMBER := 32;
  b VARCHAR2(20) := 'F';
  c NUMBER := 0;
  d VARCHAR2(20) := '';
BEGIN
  sp_conver_temp(a,b,c,d);
  dbms_output.put_line(a);
  dbms_output.put_line(b);
  dbms_output.put_line(c);
  dbms_output.put_line(d);
END;
/

```

	Bases de datos Tema 8		Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL		5

5. Crear una función llamada **GetEmail** que devuelva un email con el siguiente formato: dia-mes-minutos-segundos@gmail.com

La función anterior será llamada desde un procedimiento que debemos crear denominado **sp_create_empleadosv2** que creará una tabla llamada **EmpleadosV2** con la siguiente estructura.

emp_id NUMBER y **emp_email** VARCHAR2(50)

El procedimiento insertará en dicha tabla un registro por cada uno de los empleados que existen en la tabla **employees** de la base de datos HR, es decir, si existen 20 empleados entonces se insertarán 20 registros en la nueva tabla. Para generar el contenido del campo **emp_email** usaremos la función anteriormente creada. Pista usar la función **EXTRACT**.

Ejemplo de un registro: **emp_id** = 1, **emp_email** = 24-4-40-20@gmail.com

```



CREATE OR REPLACE FUNCTION GetEmail
RETURN VARCHAR2 IS
    v_email VARCHAR2(50);
BEGIN
    v_email := TO_CHAR(SYSDATE, 'DD-MM-MI-SS') ||
    '@gmail.com';
    RETURN v_email;
END;
/

CREATE OR REPLACE PROCEDURE sp_create_empleadosv2
AS
    counter NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO counter
    FROM all_tables
    WHERE table_name = 'EMPLEADOSV2';

    IF counter = 0 THEN
        EXECUTE IMMEDIATE 'CREATE TABLE EmpleadosV2 (
            emp_id NUMBER,
            emp_email VARCHAR2(50)
        )';
    END IF;

    FOR emp IN (SELECT employee_id FROM employees) LOOP
        INSERT INTO EmpleadosV2 (emp_id, emp_email)
        VALUES (emp.employee_id, GetEmail());
    END LOOP;
END;
/

```

 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 8		 UNIVERSIDAD EAFEC	Práctica
	Oracle PL/SQL – Procedure/Function en PL/SQL			5

```

DECLARE
BEGIN
    sp_create_empleadosv2;
END;
/
SELECT * FROM empleadosv2;

```

	EMP_ID	EMP_EMAIL	
1	100	04-06-38-26@gmail.com	
2	101	04-06-38-26@gmail.com	
3	102	04-06-38-26@gmail.com	
4	103	04-06-38-26@gmail.com	
5	104	04-06-38-26@gmail.com	
6	105	04-06-38-26@gmail.com	
7	106	04-06-38-26@gmail.com	
8	107	04-06-38-26@gmail.com	
9	108	04-06-38-26@gmail.com	
10	109	04-06-38-26@gmail.com	
11	110	04-06-38-26@gmail.com	
12	111	04-06-38-26@gmail.com	
13	112	04-06-38-26@gmail.com	
14	113	04-06-38-26@gmail.com	
15	114	04-06-38-26@gmail.com	
16	115	04-06-38-26@gmail.com	
17	116	04-06-38-26@gmail.com	
18	117	04-06-38-26@gmail.com	
19	118	04-06-38-26@gmail.com	
20	119	04-06-38-26@gmail.com	
21	120	04-06-38-26@gmail.com	
22	121	04-06-38-26@gmail.com	
23	122	04-06-38-26@gmail.com	
24	123	04-06-38-26@gmail.com	
25	124	04-06-38-26@gmail.com	