


UF6 · Actividad 11

Tarea Individual.

Trabajar con Procedimientos

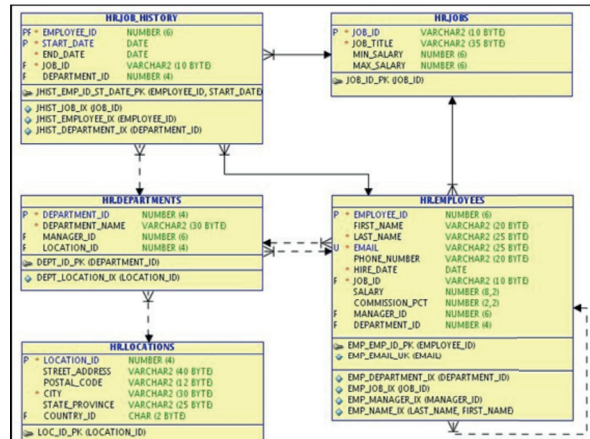
MIGUEL ÁNGEL LOZANO BERMEJO



MP_0484
Bases
de datos

Actividad 11. Tarea Individual. Trabajar con Procedimientos.

Seguimos trabajando con la base de datos creada en la unidad anterior.



1. Crea un procedimiento que reciba un número de empleado y una cadena correspondiente a su nuevo oficio (job_id) como parámetros. El procedimiento deberá localizar el empleado, modificar su oficio por el nuevo y visualizar los cambios realizados. Pero antes se deberán hacer las siguientes comprobaciones:
 - Se deberá comprobar si el id del oficio existe. Esta tarea la realizará una función aparte a la que se le pasará el oficio por parámetro y retornará un booleano. Si el oficio no existe, el procedimiento informará con un mensaje por consola.
 - Crea también un bloque anónimo donde ejecutar el procedimiento con los valores de los argumentos recogidos en variables de sustitución.

Primero buscamos los trabajadores, comprobamos sus ID y sus oficios.

```
SELECT employee_id, first_name, last_name, job_id FROM hr.employees;
```

Segundo buscamos el nombre de los empleos.

```
SELECT job_id, job_title FROM jobs;
```

Hoja de Trabajo Generador de Consultas

```
1 select employee_id, first_name, last_name, job_id FROM hr.employees;
```

Salida de Script: Resultado de la Consulta: Todas las Filas Recuperadas: 107 en 0,039 segundos

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID
1	100 Steven	King	AD_PRES
2	101 Neena	Kochhar	AD_VP
3	102 Lex	De Haan	AD_VP
4	103 Alexander	Hunold	IT_PROG
5	104 Bruce	Ernst	IT_PROG
6	105 David	Austin	IT_PROG
7	106 Valli	Pataballa	IT_PROG
8	107 Diana	Lorentz	IT_PROG
9	108 Nancy	Greenberg	FI_MGR
10	109 Daniel	Faviet	FI_ACCOUNT
11	110 John	Chen	FI_ACCOUNT
12	111 Ismael	Sciarra	FI_ACCOUNT
13	112 Jose Manuel	Uman	FI_ACCOUNT
14	113 Luis	Popp	FI_ACCOUNT
15	114 Den	Raphaely	PU_MAN
16	115 Alexander	Khoo	PU_CLERK
17	116 Shelli	Baida	PU_CLERK
18	117 Sigal	Tobias	PU_CLERK
19	118 Guy	Himuro	PU_CLERK
20	119 Karen	Colmenares	PU_CLERK
21	120 Matthew	Weiss	ST_MAN
22	121 Adam	Fripp	ST_MAN
23	122 Payan	Kaufling	ST_MAN
24	123 Shanta	Vollman	ST_MAN
25	124 Kevin	Mourgos	ST_MAN
26	125 Julia	Nayer	ST_CLERK

Hoja de Trabajo Generador de Consultas

```
1 SELECT job_id, job_title FROM jobs;
```

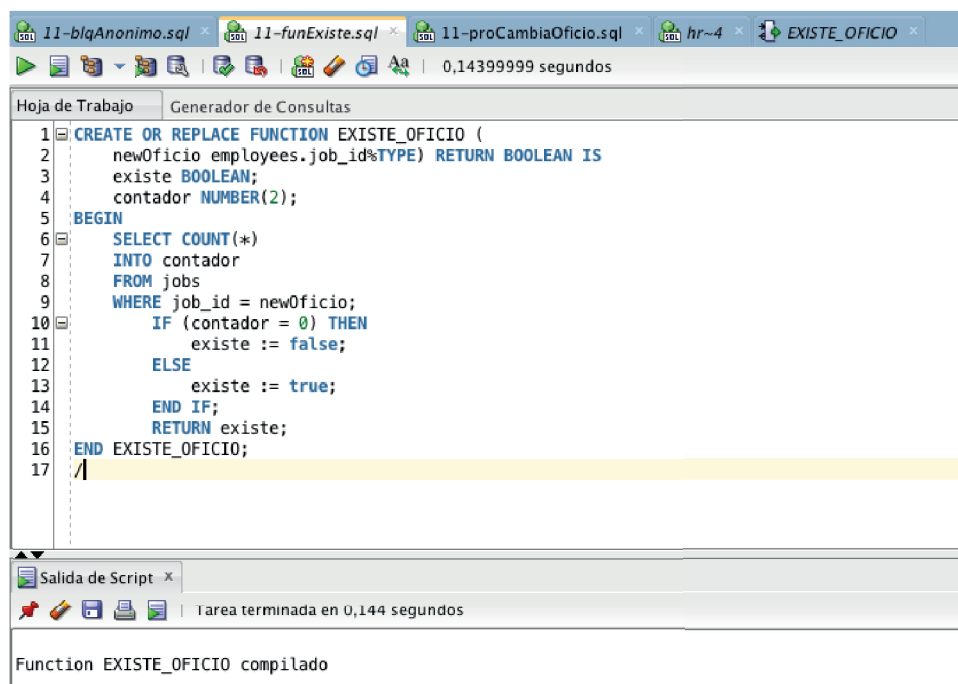
Salida de Script: Resultado de la Consulta: Todas las Filas Recuperadas: 20 en 0,009 segundos

JOB_ID	JOB_TITLE
AD_PRES	President
AD_VP	Administration Vice President
AD_ASST	Administration Assistant
FI_MGR	Finance Manager
FI_ACCOUNT	Accountant
AC_MGR	Accounting Manager
AC_ACCOUNT	Public Accountant
SA_MAN	Sales Manager
SA_REP	Sales Representative
PU_MAN	Purchasing Manager
PU_CLERK	Purchasing Clerk
ST_MAN	Stock Manager
ST_CLERK	Stock Clerk
SH_CLERK	Shipping Clerk
IT_PROG	Programmer
MK_MAN	Marketing Manager
MK_REP	Marketing Representative
HR_REP	Human Resources Representative
PR_REP	Public Relations Representative
FOR_ORACLE	Formador

Crear la FUNCIÓN

Creemos la *Función EXISTE_OFICIO*, en la cual comprobaremos si existe el oficio que vamos a asignar a los empleados. Nuestra función recibe un parámetro de entrada *newOficio* que es el que hemos introducido por consola desde el bloque anónimo. Declaramos la variable *existe* como booleano y *contador* como número. En la sección *BEGIN*, introduciremos en *contador* *true* o *false*, si dentro de *jobs* hay un oficio o más que se llame igual al oficio que buscamos, el contador será distinto de cero por lo que la función devolverá *TRUE*, si no encuentra coincidencia, contador será igual a cero, con lo que la función devolverá *FALSE*.

```
CREATE OR REPLACE FUNCTION EXISTE_OFICIO (  
    newOficio employees.job_id%TYPE) RETURN BOOLEAN IS  
    existe BOOLEAN;  
    contador NUMBER(2);  
BEGIN  
    SELECT COUNT(*)  
    INTO contador  
    FROM jobs  
    WHERE job_id = newOficio;  
    IF (contador = 0) THEN  
        existe := false;  
    ELSE  
        existe := true;  
    END IF;  
    RETURN existe;  
END EXISTE_OFICIO;  
/
```

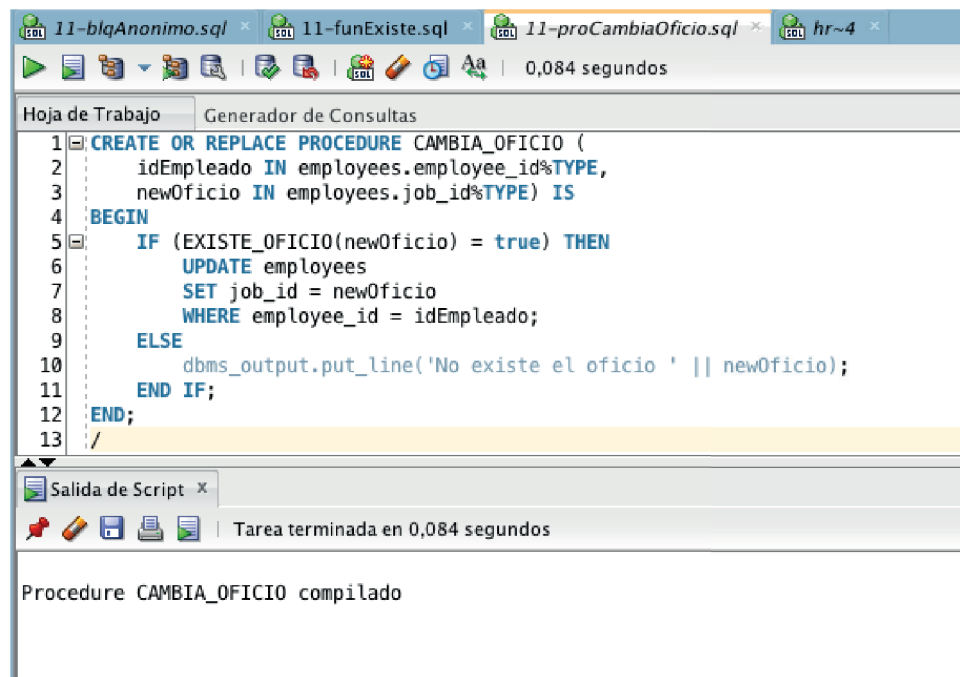


Podemos observar que la función *EXISTE_OFICIO* se ha compilado satisfactoriamente.

Crear el PROCEDIMIENTO

Crearemos el *Procedimiento CAMBIA_OFICIO*, en el cual realizaremos el cambio de oficio de los empleados, como vemos este procedimiento recibe dos parámetros, la id del empleado y el nuevo oficio que se la va a asignar. En la sección *BEGIN* podemos observar como pasamos un parametro a la función *EXISTE_OFICIO* que será devuelto con un booleano *true* o *false*. Si es *TRUE* fijamos el *job_id* con el valor del nuevo oficio en la tabla *employees* donde la id del empleado sea igual a la id que le hemos pasado en la cabecera, en caso de haber recibido un *FALSE* nos saldrá un mensaje diciendo que ese oficio no existe.

```
CREATE OR REPLACE PROCEDURE CAMBIA_OFICIO (  
    idEmpleado IN employees.employee_id%TYPE,  
    newOficio IN employees.job_id%TYPE) IS  
BEGIN  
    IF (EXISTE_OFICIO(newOficio) = true) THEN  
        UPDATE employees  
        SET job_id = newOficio  
        WHERE employee_id = idEmpleado;  
    ELSE  
        dbms_output.put_line('No existe el oficio ' || newOficio);  
    END IF;  
END;  
/
```



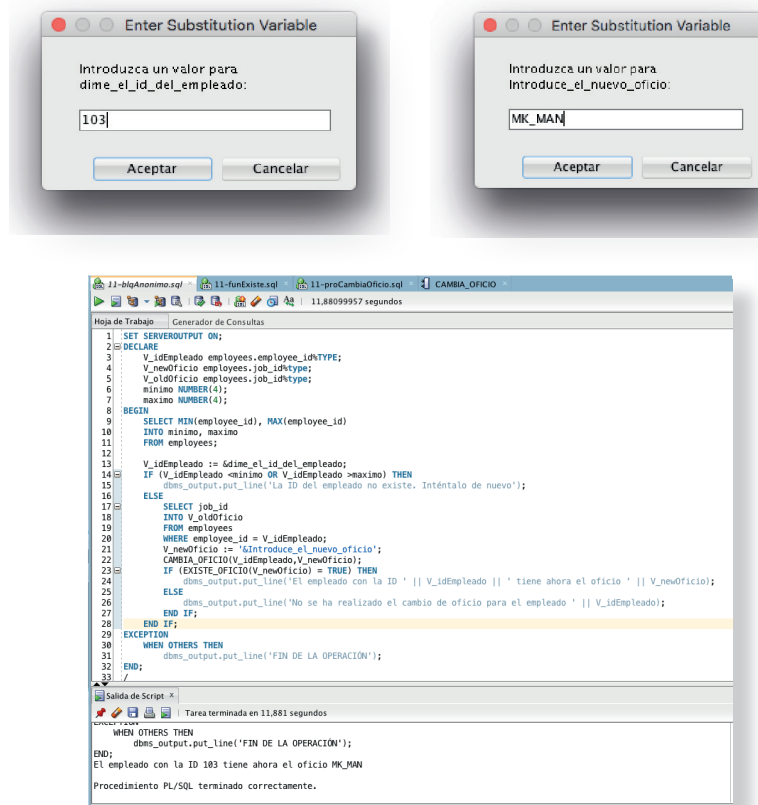
Podemos observar que el procedimiento *CAMBIA_OFICIO* se ha compilado satisfactoriamente.

Crear el BLOQUE ANONIMO

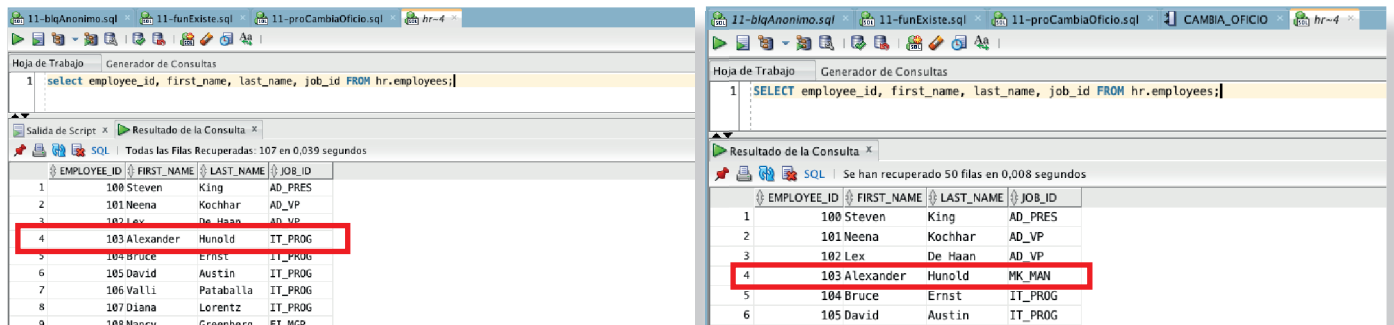
Creamos el *Procedimiento CAMBIA_OFICIO*, en el cual realizaremos el cambio de oficio de los empleados, como vemos este procedimiento recibe dos parámetros, la id del empleado y el nuevo oficio que se la va a asignar. En la sección *BEGIN* podemos observar como pasamos un parametro a la función *EXISTE_OFICIO* que será devuelto con un booleano true o false. Si es TRUE fijamos el *job_id* con el valor del nuevo oficio en la tabla *employees* donde la id del empleado sea igual a la id que le hemos pasado en la cabecera, en caso de haber recibido un FALSE nos saldrá un mensaje diciendo que ese oficio no existe.

```
SET SERVEROUTPUT ON;
DECLARE
    V_idEmpleado employees.employee_id%TYPE;
    V_newOficio employees.job_id%type;
    V_oldOficio employees.job_id%type;
    minimo NUMBER(4);
    maximo NUMBER(4);
BEGIN
    SELECT MIN(employee_id), MAX(employee_id)
    INTO minimo, maximo
    FROM employees;

    V_idEmpleado := &dime_el_id_del_empleado;
    IF (V_idEmpleado < minimo OR V_idEmpleado > maximo) THEN
        dbms_output.put_line('La ID del empleado no existe.
Inténtalo de nuevo');
    ELSE
        SELECT job_id
        INTO V_oldOficio
        FROM employees
        WHERE employee_id = V_idEmpleado;
        V_newOficio := '&Introduce_el_nuevo_oficio';
        CAMBIA_OFICIO(V_idEmpleado,V_newOficio);
        IF (EXISTE_OFICIO(V_newOficio) = TRUE) THEN
            dbms_output.put_line('El empleado con la ID ' || V_
idEmpleado || ' tiene ahora el oficio ' || V_newOficio);
        ELSE
            dbms_output.put_line('No se ha realizado el cambio de
oficio para el empleado ' || V_idEmpleado);
        END IF;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('FIN DE LA OPERACIÓN');
END;
/
```



Podemos observar, en la imagen anterior y en la segunda imagen de abajo, como se ha realizado el cambio de oficio del empleado con ID 103, ahora su oficio es MK_MAN (Marketing Manager).



En el momento que introducimos un oficio que no existe, nos sale un mensaje de advertencia diciendo que no existe.

