

1. FUNCIONES

Tienen la siguiente estructura general:

```

FUNCTION <nombrefuncion> [( <lista de parámetros> )]
IS
RETURN <tipo del valor devuelto>
    <declaraciones>;
BEGIN
    <instrucciones>;
    RETURN <expresión>
[EXCEPTION
    <excepciones>;]
END [<nombreprocedimiento>;]
    
```

Para crear una función lo haremos utilizando la orden **CREATE**
CREATE [OR REPLACE] FUNCTION <nombrefuncion>

```

CREATE OR REPLACE FUNCTION HOY
RETURN VARCHAR2
IS
DIA VARCHAR2(10);
BEGIN
DIA:=TO_CHAR (SYSDATE,'DAY');
RETURN DIA;
END;
    
```

Para ejecutar una función la llamaremos desde otro procedimiento o función.
 Podemos llamar al procedimiento que hemos creado, VER, para que nos muestre lo que devuelve la función.

```
EXECUTE VER (HOY);
```

Podemos borrar un procedimiento o una función con la orden **DROP**;

```

DROP PROCEDURE <nombreprocedure>;
DROP FUNCTION <nombrefuncion>;
    
```

2. SELECT, INSERT, UPDATE, DELETE en PL/SQL

PL/SQL permite acceder a los datos almacenados en las tablas de la base de datos Oracle y manipularlos.

Podemos realizar **INSERT, UPDATE o DELETE** de una o varias filas de una tabla.

Ejemplos:

Procedimiento para dar de alta un nuevo departamento en la tabla DEPART. Recibe el número de departamento, el nombre y la localidad e inserta una nueva fila con estos datos.

```
CREATE OR REPLACE PROCEDURE
    ALTA_DEPART (PNUM NUMBER, PNOMBRE VARCHAR2, PLOC VARCHAR2)
AS
BEGIN
    INSERT INTO DEPART VALUES (PNUM,PNOMBRE,PLOC);
END;
```

Procedimiento para subir una cantidad al salario de un empleado. El procedimiento recibe la cantidad y el apellido del empleado.

```
CREATE OR REPLACE PROCEDURE
    SUBE_SALAR (PCANTIDAD NUMBER, PAPELL VARCHAR2)
AS
BEGIN
    UPDATE EMPL
    SET SALARIO = SALARIO + PCANTIDAD
    WHERE APELLIDO = PAPELL;
END;
```

Procedimiento para dar de baja un departamento. El procedimiento recibe el número de departamento.

```
CREATE OR REPLACE PROCEDURE BORRA_DEPART (PNUM NUMBER)
AS
BEGIN
    DELETE DEPART
    WHERE DEPT_NO = PNUM;
END;
```

Probar los procedimientos creados con los siguientes datos:

1. ALTA_DEPART con el departamento 50 de nombre PUBLICIDAD en MADRID
2. SUBE_SALAR con 100 € al que se apellida SALA
3. BORRA_DEPART con el departamento 50 que acabamos de dar de alta.

TEMA 9: PL/SQL Funciones y Acceso a BD

¿Qué sucederá si queremos dar de alta un departamento que ya exista?

¿Y si le intentamos subir el salario a un empleado que no existe?

¿Y si queremos borrar un departamento con datos?

Cuando desde PL/SQL intentamos dar de alta un registro con clave duplicada o cuando queremos borrar un registro con datos relacionados en otra tabla, se produce un error. Y el procedimiento termina su ejecución con un error.

Para que esto no suceda deberíamos comprobar primero mediante las select necesarias si esto va a suceder o no.

SELECT

PL/SQL permite ejecutar cualquier consulta admitida por la base de datos. Pero cuando se ejecuta la consulta, el resultado no se muestra automáticamente en el terminal del usuario, sino que queda en un área de memoria denominada cursor a la que accederemos utilizando variables.

La sintaxis es la siguiente:

```
SELECT <columna/s> INTO <variable/s>
FROM <tabla/s>
WHERE ...
```

Por ejemplo:

```
SELECT EMP_NO, SALAR INTO V_EMP_NO, V_SALAR
FROM EMPL
WHERE APELLIDO = 'SALA';
```

V_EMP_NO Y V_SALAR serán dos variables que se han tenido que declarar antes y que deben ser del mismo tipo que las columnas de la tabla.

```
DECLARE
  V_EMP_NO NUMBER;
  V_SALAR NUMBER;
BEGIN
  SELECT EMP_NO, SALARIO INTO V_EMP_NO, V_SALAR
  FROM EMPL
  WHERE APELLIDO = 'SALA';
  DBMS_OUTPUT.PUT_LINE ('SU NÚMERO ES EL '||V_EMP_NO||' Y GANA '||V_SALAR||'€');
END;
```

Una select de este tipo puede devolver 0 filas, 1 fila o varias filas. En el caso de que devuelva 0 filas o varias filas se provoca un error y el procedimiento interrumpe su ejecución.

TEMA 9: PL/SQL Funciones y Acceso a BD

Estos errores se pueden capturar en la zona de EXCEPTION de tratamiento de errores. En el siguiente capítulo veremos más sobre tratamiento de errores, pero por ahora vamos a capturar los siguientes:

```
DECLARE
V_EMP_NO NUMBER;
V_SALAR NUMBER;
BEGIN
SELECT EMP_NO, SALARIO INTO V_EMP_NO, V_SALAR
FROM EMPL
WHERE APELLIDO <> 'SALA';
DBMS_OUTPUT.PUT_LINE ('SU NÚMERO ES EL '||V_EMP_NO||' Y GANA '||V_SALAR||'€');
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE ('LA CONSULTA NO HA DEVUELTO NINGUNA FILA');
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE ('LA CONSULTA DEVUELVE VARIAS FILAS');
END;
```

Vamos ahora a completar los procedimientos anteriores.

En ALTA_DEPART primero comprobamos si ya existe el departamento que queremos dar de alta.

Podemos “contar” cuantos departamentos hay con ese número. Si hay 0 lo damos de alta y si hay más de 0 damos un mensaje de error.

Una SELECT COUNT(*) siempre devuelve un valor, si no hay nada devuelve 0

```
-- VERSIÓN 1
CREATE OR REPLACE PROCEDURE
ALTA_DEPART (PNUM NUMBER, PNOMBRE VARCHAR2, PLOC VARCHAR2)
AS
CUANTOS NUMBER;
BEGIN
SELECT COUNT(*) INTO CUANTOS
FROM DEPART
WHERE DEPT_NO = PNUM;
IF CUANTOS = 0 THEN
INSERT INTO DEPART VALUES (PNUM,PNOMBRE,PLOC);
ELSE
DBMS_OUTPUT.PUT_LINE ('DEPARTAMENTO YA EXISTE');
END IF;
END;
```

TEMA 9: PL/SQL Funciones y Acceso a BD

En esta otra versión hacemos la select que comprueba si existe el departamento. Si esta select no devuelve ninguna fila, se produce un error que es capturado en la zona EXCEPTION con WHEN NO_DATA_FOUND, y es ahí donde daremos de alta el nuevo departamento.

```
-- VERSIÓN 2
CREATE OR REPLACE PROCEDURE
    ALTA_DEPART (PNUM NUMBER, PNOMBRE VARCHAR2, PLOC VARCHAR2)
AS
EXISTE NUMBER;
BEGIN
    SELECT DEPT_NO INTO EXISTE
        FROM DEPART
        WHERE DEPT_NO = PNUM;
    DBMS_OUTPUT.PUT_LINE ('DEPARTAMENTO YA EXISTE');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        INSERT INTO DEPART VALUES (PNUM,PNOMBRE,PLOC);
END;
```

Vamos ahora a modificar el procedimiento que borra departamentos para comprobar primero que no tiene empleados-

```
CREATE OR REPLACE PROCEDURE BORRA_DEPART (PNUM NUMBER)
AS
CUANTOS NUMBER;
BEGIN
SELECT COUNT(*) INTO CUANTOS
    FROM EMPL
    WHERE DEPT_NO = PNUM;
IF CUANTOS = 0 THEN
    DELETE DEPART
        WHERE DEPT_NO = PNUM;
ELSE
    DBMS_OUTPUT.PUT_LINE ('DEPARTAMENTO TIENE EMPLEADOS, NO SE PUEDE BORRAR');
END IF;
END;
```