

En este laboratorio trabajaremos cursores y retorno de tablas desde funciones.

Cursor

Un cursor es un apuntador al resultado de la consulta, es una forma eficiente de manejar el resultado de una consulta que retorna un gran conjunto de filas.

El cursor sigue el proceso: declarar – abrir – recorrer – cerrar.

- Se declara en una variable
`CURSOR <nombre> IS <query>;`
- Abrir
`OPEN <nombre>;`
- Recorrer
`LOOP`
`FETCH <nombre> INTO <variable>;`
`EXIT WHEN <nombre>%NOTFOUND;`
`-- Codigo que usa los datos del cursor`
`END LOOP;`
- Cerrar
`CLOSE <nombre>;`

Ejemplo:

```
CREATE OR REPLACE FUNCTION empleados
RETURN VARCHAR AS
    rEmp empleado%ROWTYPE;
    CURSOR cursEmp IS SELECT empleado.*
                        FROM empleado NATURAL JOIN departamento
                        WHERE nombreDpto = ' Ventas'
                        ORDER BY nombre;
    correos VARCHAR(500) := '';
BEGIN
    OPEN cursEmp;
    LOOP
        FETCH cursEmp INTO rEmp;
        EXIT WHEN cursEmp%NOTFOUND;
        correos := correos || rEmp.email || ' ; ' ;
    END LOOP;
    CLOSE cursEmp;
    RETURN correos;
END;
```

El proceso se puede realizar con pasos implícitos usando el FOR. El FOR abre el cursor, hace el fetch en cada iteración y lo cierra al terminar de recorrer los resultados.

```

CREATE OR REPLACE FUNCTION empleadosFOR
RETURN VARCHAR AS
    rEmp empleado%ROWTYPE;
    CURSOR cursEmp IS SELECT empleado.*
                        FROM empleado NATURAL JOIN departamento
                        WHERE nombreDpto = 'Ventas' ORDER BY nombre;
    correos VARCHAR(500) := '';
BEGIN
    FOR rEmp IN cursEmp LOOP
        correos := correos || rEmp.email || ' ; ' ;
    END LOOP;
    RETURN correos;
END;

```

Declaracion de Registros

```
TYPE <nombreTipo> IS RECORD ( <listaDeAtributos> );
```

Ejemplo:

```

DECLARE
    TYPE persona IS RECORD
        ( nombre  VARCHAR2(10),
          cedula  VARCHAR2(9),
          edad    NUMBER(3) );
    personaRecord persona;
BEGIN
    personaRecord.nombre:='Sofia';
    personaRecord.edad:=35;

    SELECT nombres, apellidos,null INTO personaRecord
    FROM   persona where cedula = 480156054;
    ...

```

Retorno de tablas

Para retornar tablas desde una función se debe:

- Crear un tipo de objeto para los registros de la tabla
- Crear el tipo para la tabla
- Escribir una función que retorna el tipo de la tabla, en el cuerpo de esa función capturar los datos y finalmente, la función retorna la tabla.

Ejemplo 1: capturando los datos con BULK COLLECT

```

CREATE OR REPLACE TYPE t_record AS OBJECT ( nom VARCHAR2(30), apel
VARCHAR2(30), edad NUMBER(3) );

```

```

CREATE OR REPLACE TYPE t_table AS TABLE OF t_record;

CREATE OR REPLACE FUNCTION tablaNombres
RETURN t_table AS
    v_ret t_table;
BEGIN
    SELECT t_record(nombre,apellido,null)
    BULK COLLECT INTO v_ret
    FROM empleado WHERE ROWNUM <= 10;
    RETURN v_ret;
END;

```

Ejemplo 2: asignando los registros a la tabla

```

CREATE OR REPLACE TYPE t_record1 AS OBJECT
    (apellido VARCHAR2(20), numero NUMBER(10));
CREATE OR REPLACE TYPE t_table1 AS TABLE OF t_record1;

CREATE OR REPLACE FUNCTION tablaNombres1
RETURN t_table1 AS
    v_ret t_table1;
BEGIN
    v_ret := t_table1(); -- constructor
    FOR rec_ape IN (SELECT apellido
                    FROM empleado WHERE salario > 1000000)
    LOOP
        v_ret.extend;
        v_ret(v_ret.count) := t_record1(rec_ape.apellido,
                                         v_ret.count+1);
    END LOOP;
    RETURN v_ret;
END;

```

Ejercicios

Usando el esquema de base de datos creado y modificado en los laboratorios anteriores, escriba los comandos para dar respuesta a los requerimientos especificados a continuación.

- 1) Se requiere una función que retorne reciba como parámetro el nombre de una ciudad y un año, y retorne una tabla con los siguientes datos de las matrículas de ese año: número de matrícula, fecha de expedición, placa, y descripción del tipo de carro.
- 2) Escriba una función que reciba como parámetro un año y retorne una tabla. Cada registro de la tabla tiene el nombre de una ciudad, y la información de cuantos partes han recibido los carros

matriculados en esa ciudad en los 4 años anteriores al dado, presentando la información con la siguientes estructura:

Ciudad	Partes	Partes-1	Partes-2	Partes-3

* En este caso el atributo “partes” se refiere a la cantidad de partes dados para el año que se recibió como parámetro. “Partes-1” para el año anterior, y así sucesivamente.

- a) Use ésta función en una consulta que retorne el total de partes registrados en cada uno de los años.
- 3) Se va a ofrecer un descuento a los conductores con menos infracciones registradas. Se requiere una función que retorne en una tabla los datos de los primeros diez conductores (con menos infracciones), y el valor del descuento, calculado de esta forma:
 - Para el primer conductor el 8% del valor de su ultimo parte
 - Para el segundo y tercer conductor 6% del valor de su ultimo parte
 - Para los puestos en 4º y 5º, 5%
 - Y para los restantes 2%

Si varios conductores tienen la misma cantidad de infracciones, se da prioridad a quienes tienen mas tiempo desde el último parte que recibieron.
- 4) Tomando como base la lógica de la función anterior, cree un procedimiento que use un **CURSOR FOR UPDATE** para actualizar el valor de la multa en las infracciones del último parte de los conductores, aplicando el descuento concedido según las condiciones del ejercicio anterior.

Al finalizar la sesión, cada estudiante debe enviar el script a mcpabon@javerianacali.edu.co.