ADMINISTRACIÓN DE BASE DE DATOS.

Tema Nº6:Programación PL/SQL II.

Indicador de logro Nº6:Elabora Cursores, utilizando la programación estructurada PL/SQL

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº6:**

Programación PS/SQL II.

**Subtema 6.1:**

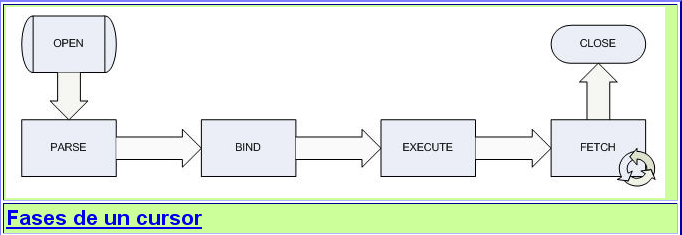
Implementa cursores explícitos e implícitos, y de actualización en bloques anónimos, utilizando la programación estructurada PL/SQL, basada en reglas de negocio.

**INTRODUCCIÓN:**

Los cursores se utilizan en PL/SQL para manejar las sentencias SELECT. Un cursor está formado por un conjunto de registros devueltos por una instrucción SQL del tipo SELECT. Desde un punto de vista interno a la base de datos Oracle, los cursores son segmentos de memoria utilizados para realizar operaciones con los registros devueltos tras ejecutar una sentencia SELECT.

Reciben el nombre de [cursores implícitos](http://www.paginasprodigy.com.mx/evaristopacheco/taller/cursores.html#cursorimplicito), cuando la sentencia SELECT regresa solo un registro. Para procesar instrucciones SELECT que devuelvan más de una fila, son necesarios [cursores explícitos](http://www.paginasprodigy.com.mx/evaristopacheco/taller/cursores.html#cursorexplicito) combinados con una estructura de bloque. Un cursor admite el uso de parámetros. Los parámetros deben declararse junto con el cursor.

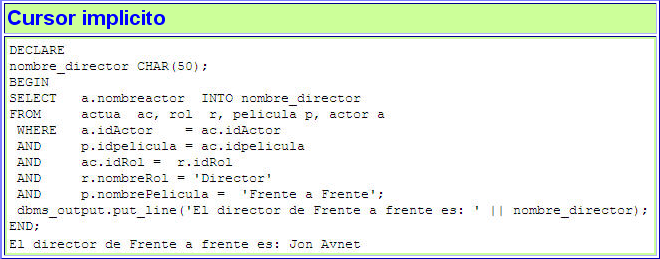
El siguiente diagrama representa como se procesa una instrucción SQL a través de un cursor.



**CURSORES IMPLÍCITOS**

Se utilizan cuando la sentencia **SELECT devuelve un solo registro**. En cada cursor implícito debe existir palabra reservada **INTO**. Las variables que reciben los datos devueltos por el cursor tienen que contienen el mismo tipo de dato que las columnas de la tabla.

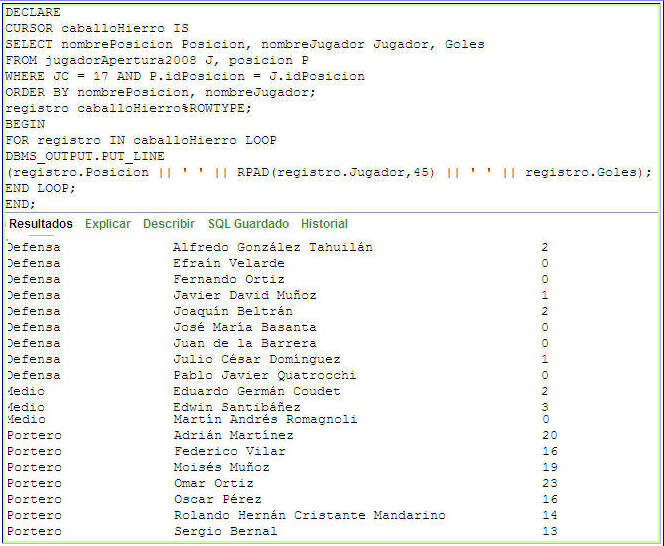
A continuación, se presenta un ejemplo en el que buscamos el nombre del director de la película **'Frente a Frente'**



**CURSORES EXPLÍCITOS:**

Se utilizan cuando **la sentencia SELECT puede devolver varios registros**. También se pueden utilizar en consultas que devuelvan un solo registro por razones de eficiencia con respecto a los cursores implícitos, eficiencia que mejorará especialmente si el cursor explícito se tiene que ejecutar varias veces dentro del bloque de código PL/SQL.

Un cursor explícito tiene que ser definido previamente como cualquier otra variable PL/SQL y debe serle asignado un nombre. **Veamos un ejemplo que muestra el nombre de los jugadores que participaron en todos los minutos del Torneo Apertura 2008, posición y goles anotados o recibidos**:



Los cursores explícitos admiten el uso de parámetros. Los parámetros deben declararse junto con el cursor. Por ejemplo: Considere a los jugadores de la apertura 2008 del Futbol Mexicano de primera división que jugaron todos los minutos y además anotaron al menos un gol.

**PROCESAMIENTO DE CURSORES:**

**Pasos a Seguir:**

1. Declarar el Cursor.
2. Apertura del Cursor.
3. Extracción de los resultados.
4. Cerrar el Cursor.

**Declaración un cursor:**

Sintaxis:

**DECLARE CURSOR** <Nombre\_Cursor> [{Parámetros}] **IS** [SENTENCIA\_SELECT];

Ejemplo:

**DECLARE CURSOR** C\_ALMACEN **IS** Select \* From PRODUCTOS**;**

**APERTURAR UN CURSOR:**

Sintaxis:

**OPEN** <Nombre del Cursor>**;**

Ejemplo:

**OPEN** C\_ALMACEN;

**EXTRAER DATOS DE UN CURSOR:**

Sintaxis:

**FETCH** <Nombre\_Cursor> **INTO** [Lista\_de\_Variables,];

Ejemplo:

**FETCH** C\_ALMACEN **INTO** Codigo, Descripcion, Precio

**CERRAR UN CURSOR:**

Sintaxis:

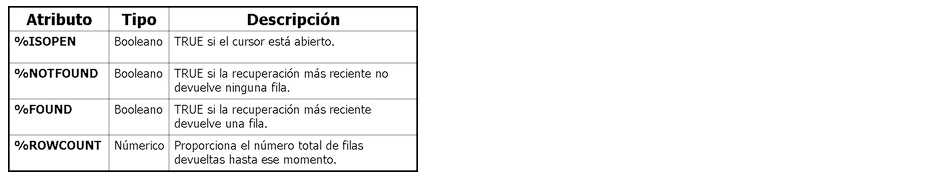
**CLOSE** <Nombre\_del\_Cursor>**;**

Ejemplo:

**CLOSE** C\_ALMACEN;

**ATRIBUTOS DE CURSORES:**

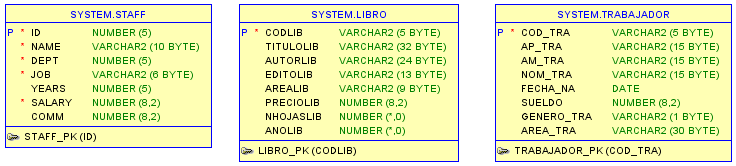
Estos proporcionan el estado del Cursor.



**EJEMPLOS DE CURSORES IMPLICITOS**

Los que solo devuelven un solo registro, en la consulta

Se tiene las siguientes tablas de datos:



**Ejemplo 1:**

Muestre los datos de un Libro determinado, ingresando mediante una variable de sustitución & el código del libro, por ejemplo ‘1000’, los datos de dicho libro deben ser mostrados.

**SOLUCION:** redacte este código desde el editor de Oracle PL/SQL.

**SET SERVEROUTPUT ON;**

**DECLARE**

Codigo LIBRO.CODLIB%TYPE; Titulo LIBRO.TITULOLIB%TYPE;

Autor LIBRO.AUTORLIB%TYPE; Editorial LIBRO.EDITOLIB%TYPE;

Area LIBRO.AREALIB%TYPE; Precio LIBRO.PRECIOLIB%TYPE;

CantidadHojas LIBRO.NHOJASLIB%TYPE; AnnoLibro LIBRO.ANOLIB%TYPE;

**BEGIN**

SELECT CODLIB, TITULOLIB, AUTORLIB, EDITOLIB, AREALIB,

PRECIOLIB, NHOJASLIB, ANOLIB

INTO Codigo, Titulo, Autor, Editorial, Area,

Precio, CantidadHojas, AnnoLibro

FROM LIBRO WHERE CODLIB = **&Codigo**;

DBMS\_OUTPUT.PUT\_LINE('DATOS DEL LIBRO : ' || **Codigo**);

DBMS\_OUTPUT.PUT\_LINE('============================================');

DBMS\_OUTPUT.PUT\_LINE('Código del Libro : ' || **Codigo**);

DBMS\_OUTPUT.PUT\_LINE('Título del Libro : ' || **Titulo**);

DBMS\_OUTPUT.PUT\_LINE('Autor del Libro : ' || **Autor**);

DBMS\_OUTPUT.PUT\_LINE('Editorial del Libro : ' || **Editorial**);

DBMS\_OUTPUT.PUT\_LINE('Area del Libro : ' || **Area**);

DBMS\_OUTPUT.PUT\_LINE('Precio del Libro : ' || **Precio**);

DBMS\_OUTPUT.PUT\_LINE('Cantidad de Hojas : ' || **CantidadHojas**);

DBMS\_OUTPUT.PUT\_LINE('Año del Libro : ' || **AnnoLibro**);

DBMS\_OUTPUT.PUT\_LINE('============================================');

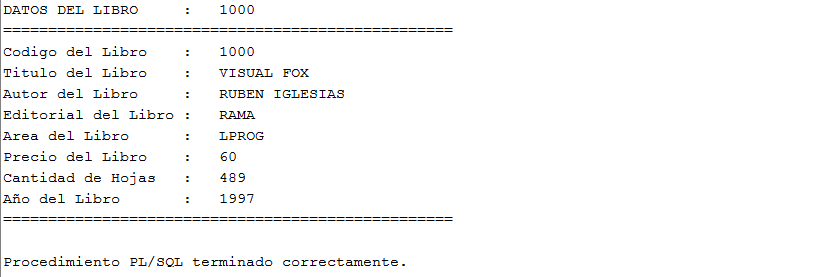
**EXCEPTION**

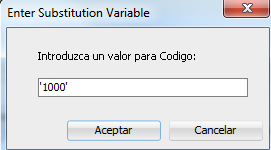
WHEN **NO\_DATA\_FOUND** THEN

DBMS\_OUTPUT.put\_line ('Libro No Existe!!!!');

**END;**

**Resultado de la ejecución:**





**Ejemplo 2:**

Se desea vender todos los libros de la librería, clasificados por área del libro **(ANALISIS, ARTE, PROYECTOS, LPROG, LIDEREZGO y CAD)**, genere un código PL/SQL que realice dicha acción, con totalizaciones incluidas. Redacte el código desde el editor de Oracle PL/SQL.

**SOLUCIÓN:** redacte este código desde el editor de Oracle PL/SQL

**SET SERVEROUTPUT ON;**

**DECLARE**

TOTAL\_ANALISIS NUMBER:=0;

TOTAL\_ARTE NUMBER:=0;

TOTAL\_PROYECTOS NUMBER:=0;

TOTAL\_LPROG NUMBER:=0;

TOTAL\_LIDERAZGO NUMBER:=0;

TOTAL\_CAD NUMBER:=0;

TOTAL\_IMPORTE NUMBER:=0; IGV NUMBER:=0; TOTAL\_GENERAL NUMBER:=0;

**BEGIN**

SELECT SUM (PRECIOLIB) INTO TOTAL\_ANALISIS

FROM LIBRO WHERE AREALIB='ANALISIS';

SELECT SUM (PRECIOLIB) INTO TOTAL\_ARTE

FROM LIBRO WHERE AREALIB='ARTE';

SELECT SUM (PRECIOLIB) INTO TOTAL\_PROYECTOS

FROM LIBRO WHERE AREALIB='PROYECTOS';

SELECT SUM (PRECIOLIB) INTO TOTAL\_LPROG

FROM LIBRO WHERE AREALIB='LPROG';

SELECT SUM (PRECIOLIB) INTO TOTAL\_LIDERAZGO

FROM LIBRO WHERE AREALIB='LIDERAZGO';

SELECT SUM (PRECIOLIB) INTO TOTAL\_CAD

FROM LIBRO WHERE AREALIB='CAD';

TOTAL\_IMPORTE:=TOTAL\_ANALISIS + TOTAL\_ARTE + TOTAL\_PROYECTOS +

TOTAL\_LPROG + TOTAL\_LIDERAZGO + TOTAL\_CAD;

IGV:= TOTAL\_IMPORTE \* 0.18;

TOTAL\_GENERAL:= TOTAL\_IMPORTE + IGV;

DBMS\_OUTPUT.PUT\_LINE('VENTA DE LA LIBRERIA POR AREA DEL LIBRO');

DBMS\_OUTPUT.PUT\_LINE('=================================================');

DBMS\_OUTPUT.PUT\_LINE('TOTAL LIBROS DE ANALISIS : ' || TOTAL\_ANALISIS);

DBMS\_OUTPUT.PUT\_LINE('TOTAL DE LIBROS DE ARTE : ' || TOTAL\_ARTE);

DBMS\_OUTPUT.PUT\_LINE('TOTAL DE LIBRO DE PROYECTOS : ' || TOTAL\_PROYECTOS);

DBMS\_OUTPUT.PUT\_LINE('TOTAL DE LIBRO DE LPROG : ' || TOTAL\_LPROG);

DBMS\_OUTPUT.PUT\_LINE('TOTAL LIBRO DE LIDERAZGO : ' || TOTAL\_LIDERAZGO);

DBMS\_OUTPUT.PUT\_LINE('TOTAL LIBRO DE CAD : ' || TOTAL\_CAD);

DBMS\_OUTPUT.PUT\_LINE('==================================================');

DBMS\_OUTPUT.PUT\_LINE('TOTAL IMPORTE : ' || TOTAL\_IMPORTE);

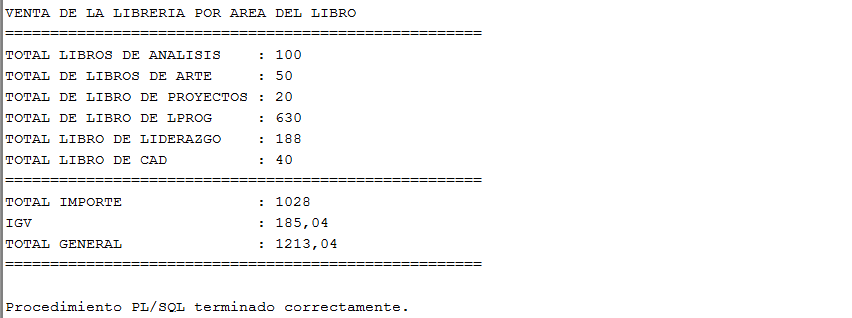
DBMS\_OUTPUT.PUT\_LINE('IGV : ' || IGV);

DBMS\_OUTPUT.PUT\_LINE('TOTAL GENERAL : ' || TOTAL\_GENERAL);

DBMS\_OUTPUT.PUT\_LINE('==================================================');

**END**;

**Resultado de la ejecución:**



Aquí desarrollarás el contenido del subtema 1.1.

Ejemplos:

Aquí colocarás los ejemplos que facilitarán la comprensión del subtema

**EJEMPLOS DE CURSORES EXPLICITO**

Los que devuelven varios registros, en la consulta

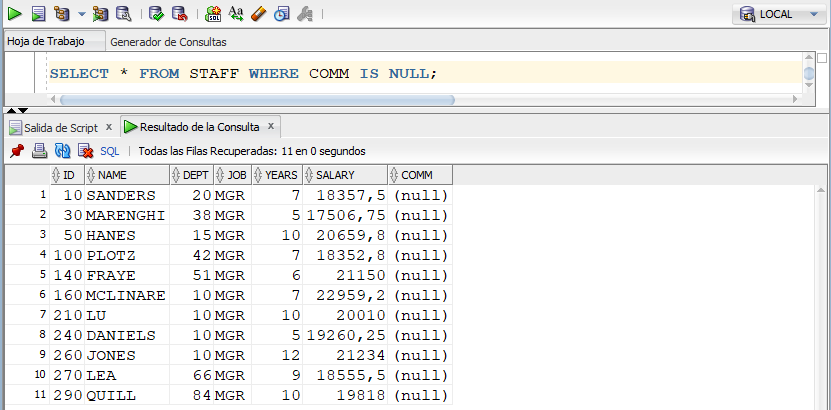
**Ejemplo 1:**

La tabla STAFF posee elementos nulos en la columna COMM (comisiones), crear un cursor de actualización el cual modifique estos elementos nulos por cero.

Consulta para saber que comisiones son nulas obtendremos:

**SELECT \* FROM STAFF WHERE COMM IS NULL;**

**Resultado:**



Ahora procedemos a crear nuestro cursor de **actualización**, cambiar esos **nulos** por **ceros**, desde el editor de Oracle PL/SQL.

**SET SERVEROUTPUT ON;**

**DECLARE**

**CURSOR** ASIGNAR\_CEROS **IS** **SELECT ID, COMM FROM STAFF WHERE COMM IS NULL**;

Codigo STAFF.ID%TYPE;

Comision STAFF.COMM%TYPE;

**BEGIN**

**OPEN** ASIGNAR\_CEROS;

**FETCH** ASIGNAR\_CEROS **INTO** Codigo, Comision;

**WHiLE** ASIGNAR\_CEROS%Found

**LOOP**

**IF** (Comision IS NULL) THEN

**UPDATE** STAFF SET COMM = 0 **WHERE** ID=Codigo;

**COMMIT**;

**END IF;**

**FETCH** ASIGNAR\_CEROS **INTO** Codigo, Comision;

END LOOP;

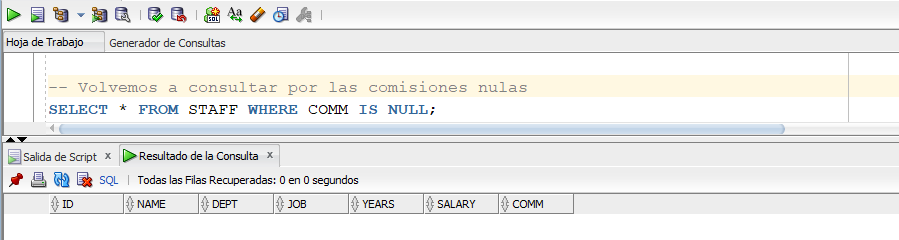
CLOSE ASIGNAR\_CEROS;

**END;**

**Su ejecución devuelve:**



**Después de ejecutar el código, proceda a consultar nuevamente por las comisiones nulas.**



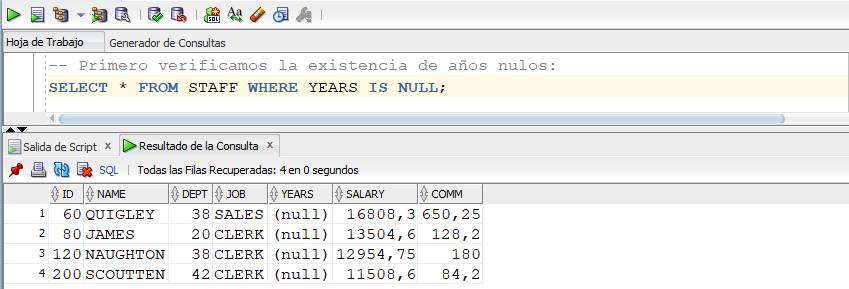
Como se observa ya no existen comisiones nulas.

**Ejemplo 2:**

En la tabla STAFF, existen años nulos en la columna YEARS, se pide crear una copia de seguridad de aquellas filas, pasar esos registros a la tabla BACKUP\_STAFF.

**Primero verificamos la existencia de años nulos:**

**SELECT \* FROM STAFF WHERE YEARS IS NULL;**



**Segundo, Creamos la tabla BACKUP\_STAFF que tenga la misma estructura de la tabla STAFF.**

**CREATE TABLE** BACKUP\_STAFF(

**ID** Numeric (5, 0) PRIMARY KEY NOT NULL ,

**NAME** Varchar2 (10) NOT NULL ,

**DEPT** Numeric (5, 0) NOT NULL ,

**JOB** Varchar2 (6) NOT NULL ,

**YEARS** Numeric (5, 0) NULL ,

**SALARY** Numeric (8, 2) NOT NULL ,

**COMM** Numeric (8, 2) NULL

);

Ejecutamos el código de creación de la tabla BACKUP\_STAFF

**Resultado**:



**Tercero, Creamos el cursor explícito de Inserción de datos, en editor de ORACLE PL/SQL.**

**SET SERVEROUTPUT ON;**

**DECLARE**

CURSOR RESPALDO IS

SELECT ID, NAME, DEPT, JOB, YEARS, SALARY, COMM FROM STAFF

WHERE YEARS IS NULL;

Codigo STAFF.ID%TYPE;

Nombre STAFF.NAME%TYPE;

Departamento STAFF.DEPT%TYPE;

Trabajo STAFF.JOB%TYPE;

Annos STAFF.YEARS%TYPE;

Salario STAFF.SALARY%TYPE;

Comision STAFF.COMM%TYPE;

**BEGIN**

OPEN RESPALDO;

FETCH RESPALDO INTO Codigo, Nombre, Departamento, Trabajo,

Annos, Salario, Comision;

WHiLE RESPALDO%Found

LOOP

IF (Annos IS NULL) THEN

INSERT INTO BACKUP\_STAFF

VALUES (Codigo, Nombre, Departamento, Trabajo,

Annos, Salario, Comision);

COMMIT;

END IF;

FETCH RESPALDO INTO Codigo, Nombre, Departamento, Trabajo,

Annos, Salario, Comision;

END LOOP;

CLOSE RESPALDO;

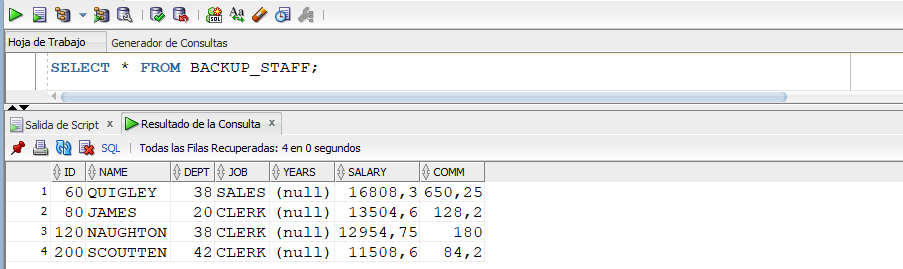
**END;**

**POR FAVOR EJECUTE EL CODIGO.**



Después de la ejecución del código, verificamos en la tabla **BACKUP\_STAFF** la copia de los registros.

**SELECT \* FROM BACKUP\_STAFF;**



**¡Éxito!**

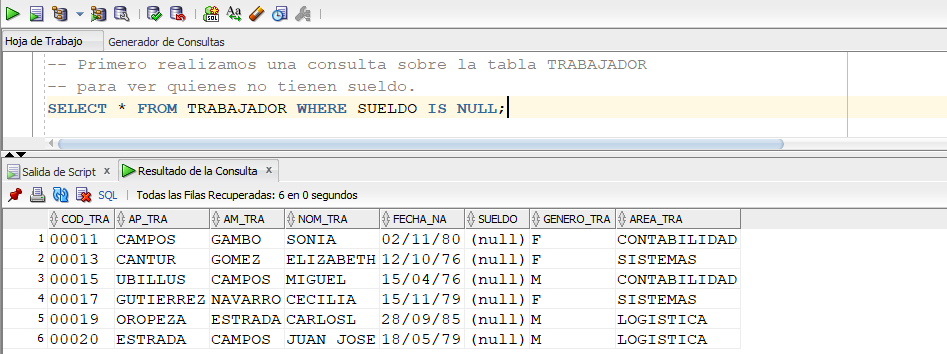
**Ejemplo 3:**

En la tabla **TRABAJADOR**, existe personal sin sueldo (**null**), crear un cursor que asigne 500 soles a aquellas personas que no tienen sueldo.

Primero realizamos una consulta sobre la tabla **TRABAJADOR** para ver quienes no tienen sueldo.

**SELECT \* FROM TRABAJADOR WHERE SUELDO IS NULL;**

Obtendremos los siguientes Registros:



Como ya sabemos quiénes son, procedemos a crear el cursor de actualización de sueldos, en la tabla **TRABAJADOR**, desde el editor de Oracle PL/SQL.

SET SERVEROUTPUT ON;

DECLARE

CURSOR ASIGNAR\_500 IS SELECT COD\_TRA, SUELDO

FROM TRABAJADOR WHERE SUELDO IS NULL;

Codigo TRABAJADOR.COD\_TRA%TYPE;

Paga TRABAJADOR.SUELDO%TYPE;

BEGIN

OPEN ASIGNAR\_500;

FETCH ASIGNAR\_500 INTO Codigo, Paga;

WHiLE ASIGNAR\_500%Found

LOOP

IF (**Paga IS NULL**) THEN

UPDATE TRABAJADOR SET SUELDO=500 WHERE COD\_TRA=Codigo;

COMMIT;

END IF;

FETCH ASIGNAR\_500 INTO Codigo, Paga;

END LOOP;

CLOSE ASIGNAR\_500;

DBMS\_OUTPUT.PUT\_LINE ('SUELDOS ACTUALIZADOS');

END;

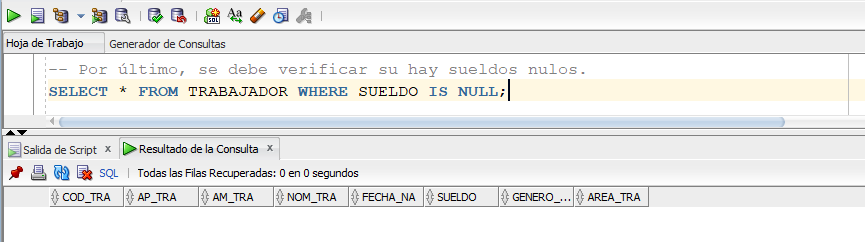
.

En el momento que ejecute el código, obtendrá el siguiente mensaje:



Por último, se debe verificar su hay sueldos nulos.

**SELECT \* FROM TRABAJADOR WHERE SUELDO IS NULL;**



**¡Éxito!** Ya no hay personal sin sueldo.

**CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA**

La codificación PL/SQL de Oracle es tan amplia, que ofrece a los estudiantes una alta gama de alternativas para la resolución de problemas, no difíciles de entender, con la prioridad de manejar los datos con suma responsabilidad por el programador Oracle.

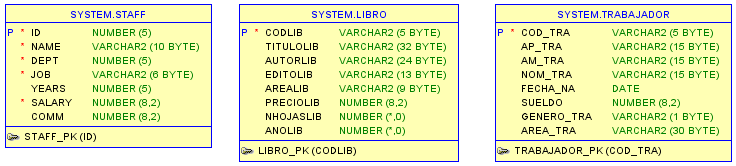
Si una base de datos es demasiado grande, se debe hacer un rediseño de base de datos centralizada, para que esta sea una base de datos distribuida ya que se pueden almacenar los datos en localidades donde son utilizados con mayor frecuencia, de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reduce el tráfico en la red.

De tal manera podemos manejar estos datos con los métodos de cursores, según reglas de negocio.

**ACTIVIDAD VIRTUAL**

Revisar y analizar el material presentado en el Tema 06.

SE TIENE LA SIGUIENTES TABLAS: **STAFF**, **LIBRO** y **TRABAJADOR**.



**LA ESTRUCTURA Y DATOS DE LA TABLAS: STAFF, LIBRO, TRABAJADOR SE ENCUENTRAN**

**EN EL ARCHIVO SCRIPT DE BASE DE DATOS.**

**Semana06\_2\_PL\_SQ.sql,**

**FAVOR DE DESCARGAR DICHO ARCHIVO.**

De acuerdo al material presentado Semana 06, Responda a las siguientes Preguntas:

**RESOLVER:**

Crear los siguientes programas anónimos PL/SQL:

1. Mediante un cursor implícito, se desea mostrar una lista de trabajadores clasificados por sexo, se desea mostrar el total de sueldos tanto de varones como de mujeres y el total de ambos. Si el total de ambos supera los 16000 entonces se debe mostrar el mensaje **LA PLANILLA NO ES PAGABLE**, en caso contrario **LA PLANILLA SERÁ PAGABLE**.
2. Mediante un cursor explicito, se desea crear una copia de seguridad de los datos de la tabla **STAFF**, solo aquellos que en cuyo trabajo (**JOB**) sea ‘**MGR’**. Estos daos deberán ser pasados a una nueva tabla, con la misma estructura de la tabla **STAFF**.