

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Licenciatura en Ingeniería en Sistemas de Información Sistemas

Cursado: Bases de Datos II

L4 – Implementación de Cursores en PL/SQL

Facilitador: Ing. Henry Lezcano

Integrantes:

Batista, Johel {8-914-587}

Pinilla, Miguel {8-975-2460}

Riley, Rolando {8-972-1033}

Villarreal, Andrés {8-970-1267}

Grupo: 1IF131

Fecha de Entrega: Lunes 10 de octubre de 2022.

Consigna del Problema Para Resolver

La Sociedad de Ingenieros y Arquitectos de Panamá, requiere que los Ingenieros en Sistemas de Información implementen el proceso de cálculo de salario quincenal para sus colaboradores, actualmente ellos cuentan con modelo físico que será proporcionado para esta tarea.

Colaboradores (id_codcolaborador number pk, nombre varchar2(25), apellido varchar2(25), cedula varchar2(12), sexo char, fecha_nacimiento date, fecha_ingreso date, status char, salario_mensual number(15,2))

salario_quincenal (id_salario number pk , id_codcolaborador number fk, fecha_pago date, salario_quincenal number(15,2), seguro_social number(15,2), seguro_educativo number(15,2), salario_neto number(15,2))

Las fechas de pago son los **15 y 30 de cada mes**. La información que mantienen en la base de datos incluye a colaboradores con estatus (**A= Activo, R=Retirado y V=Vacaciones**), las quincenas solo son pagadas a los colaboradores que están activos.

Se exige que en el proceso a implementar se utilice un cursor que busque en la base de datos la información de los colaboradores que se les hará el pago (por lo tanto, debe haber un procedimiento almacenado que ejecute la tarea de carga del colaborador preliminarmente) y que dentro del proceso a implementar solicitado cargue la tabla de los salarios quincenales con la información correspondiente a los cálculos del salario para cada colaborador.

Para calcular el seguro social, este es igual al **salario * 9.75%**. Para el cálculo del seguro educativo este es igual **salario * 1.25%**. Recuerde que este pago del salario quincenal dentro del proceso.

Por último, muestra una vista de cumplimiento de lo solicitado:

Codigo	nombre	apellido	salario mensual	salario quincenal	seguro social	seguro educativo	Salario Neto
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

Módelo Lógico-Relacional Normalizado

Colaboradores		
PK	id_codcolaborador	number
	nombre	varchar2(25)
	apellido	varchar2(25)
	cedula	varchar2(12)
	sexo	char
	fecha_nacimiento	date
	fecha_ingreso	date
	status	char
	salario_mensual	number (15,2)

Salario_quincenal		
PK	id_salario	number
	fecha_pago	date
	salario_quincenal	number(15,2)
	seguro_social	number(15,2)
	seguro_educativo	number(15,2)
	salario_neto	number(15,2)
FK	id_codcolaborador	number

Creación de Tablas y Secuencias

```
CREATE TABLE Colaboradores (  
    id_codcolaborador NUMBER,  
    nombre VARCHAR2(25) NOT NULL,  
    apellido VARCHAR2(25) NOT NULL,  
    cedula VARCHAR2(12) NOT NULL,  
    sexo CHAR,  
    fecha_nacimiento DATE,  
    fecha_ingreso DATE,  
    status CHAR DEFAULT 'A' NOT NULL ,  
    salario_mensual NUMBER(15,2) NOT NULL,  
    CONSTRAINT pk_Colaboradores PRIMARY KEY (id_codcolaborador),  
    CONSTRAINT ck_Colaboradores_Status  
    CHECK (status IN ('A', 'R', 'V')),  
    CONSTRAINT ck_Colaboradores_sexo  
    CHECK (sexo IN ('M', 'F')),  
    CONSTRAINT ck_Colaboradores_salario  
    CHECK (salario_mensual > 0)  
);
```

--SECUENCIA PARA ID DE COLABORADOR

```
CREATE SEQUENCE col_pk_seq  
    MINVALUE 1  
    START WITH 1  
    INCREMENT BY 1  
    CACHE 20;
```

```
CREATE TABLE Salario_quincenal (  
    id_salario NUMBER,  
    id_codcolaborador NUMBER NOT NULL,  
    fecha_pago DATE,  
    salario_quincenal NUMBER(15,2) NOT NULL,  
    seguro_social NUMBER(15,2),  
    seguro_educativo NUMBER(15,2),  
    salario_neto NUMBER(15,2),  
    CONSTRAINT pk_Salario_quincenal PRIMARY KEY (id_salario),  
    CONSTRAINT fk_Colaboradores  
    FOREIGN KEY (id_codcolaborador)  
    REFERENCES Colaboradores(id_codcolaborador)  
);
```

--SECUENCIA DE ID SALARIO

```
CREATE SEQUENCE sal_pk_seq  
    MINVALUE 1  
    START WITH 1  
    INCREMENT BY 1  
    CACHE 20;
```

Implementación de Procesos

```
CREATE OR REPLACE PROCEDURE AgregarColaborador (  
    p_nombre Colaboradores.nombre%TYPE,  
    p_apellido Colaboradores.apellido%TYPE,  
    p_cedula Colaboradores.cedula%TYPE,  
    p_sexo Colaboradores.sexo%TYPE,  
    p_fecha_nac Colaboradores.fecha_nacimiento%TYPE,  
    p_status Colaboradores.status%TYPE,  
    p_salario_mensual Colaboradores.salario_mensual%TYPE,  
    p_resultado OUT VARCHAR2  
    ) AS  
  
BEGIN  
  
    p_resultado := 'Insercion Exitosa';  
    INSERT INTO Colaboradores  
    VALUES (col_pk_seq.nextval, p_nombre, p_apellido, p_cedula,  
    p_sexo, TO_DATE(p_fecha_nac, 'DD/MM/YYYY'),  
    TO_DATE(SYSDATE, 'DD/MM/YYYY HH24:MI:SS'),  
    p_status, p_salario_mensual);  
  
    COMMIT;  
  
EXCEPTION  
  
    WHEN DUP_VAL_ON_INDEX THEN  
        p_resultado := 'Error en la Insercion';  
    WHEN OTHERS THEN  
        p_resultado := 'Error en la Insercion';  
  
END;
```

--LLAMADA A PROCEDIMIENTO DE INSERCIÓN

DECLARE

```
v_nombre Colaboradores.nombre%TYPE := '&Nombre';
v_apellido Colaboradores.apellido%TYPE := '&Apellido';
v_cedula Colaboradores.cedula%TYPE := '&Cedula';
v_sexo Colaboradores.sexo%TYPE := '&Sexo';
v_fecha_nac Colaboradores.fecha_nacimiento%TYPE := '&Fecha_nac';
v_status Colaboradores.status%TYPE := '&Status';
v_salario_mensual Colaboradores.salario_mensual%TYPE := '&SalarioMen';
v_resultado VARCHAR2(100);
```

BEGIN

```
AgregarColaborador(v_nombre, v_apellido, v_cedula, v_sexo,
v_fecha_nac, v_status, v_salario_mensual, v_resultado);
DBMS_OUTPUT.PUT_LINE(v_resultado);
```

END;

```
new 4: v_cedula Colaboradores.cedula%TYPE := '8-967-1022';
Enter value for sexo: M
old 5: v_sexo Colaboradores.sexo%TYPE := '&Sexo';
new 5: v_sexo Colaboradores.sexo%TYPE := 'M';
Enter value for fecha_nac: 11/JAN/2001
old 6: v_fecha_nac Colaboradores.fecha_nacimiento%TYPE := '&Fecha_nac';
new 6: v_fecha_nac Colaboradores.fecha_nacimiento%TYPE := '11/JAN/2001';
Enter value for status: A
old 7: v_status Colaboradores.status%TYPE := '&Status';
new 7: v_status Colaboradores.status%TYPE := 'A';
Enter value for salariomen: 5650
old 8: v_salario_mensual Colaboradores.salario_mensual%TYPE := '&SalarioMen';
new 8: v_salario_mensual Colaboradores.salario_mensual%TYPE := '5650';

PL/SQL procedure successfully completed.
```

--PROCEDIMIENTO INSERT TABLA SALARIOS

--Este procedimiento es invocado por el bloque anónimo con el cursor

```
CREATE OR REPLACE PROCEDURE AgregarSalarioQ (  
    p_id_codcolaborador Colaboradores.id_codcolaborador%TYPE,  
    p_salario_quincenal Salario_quincenal.salario_quincenal%TYPE,  
    p_seguro_social Salario_quincenal.seguro_social%TYPE,  
    p_seguro_educativo Salario_quincenal.seguro_educativo%TYPE,  
    p_salario_neto Salario_quincenal.salario_neto%TYPE,  
    p_resultado OUT VARCHAR2  
    ) AS  
  
BEGIN  
    p_resultado := 'Insercion Exitosa';  
    --sentencia input  
    INSERT INTO Salario_quincenal VALUES (sal_pk_seq.nextval,  
    p_id_codcolaborador, TO_DATE(SYSDATE, 'DD/MM/YYYY'), p_salario_quincenal,  
    p_seguro_social, p_seguro_educativo, p_salario_neto);  
  
EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
        P_resultado := 'Error en la insercion';  
    WHEN OTHERS THEN  
        p_resultado := 'Error al Ejecutar el Procedimiento';  
  
END;
```


-- Cursor Solicitado

-- BLOQUE CALCULO DE SALARIO CURSOS

DECLARE

CURSOR c_SalarioCal IS

SELECT id_codcolaborador, salario_mensual

FROM Colaboradores

WHERE status = 'A';

--para almacenar el salario quincenal del colaborador

v_sal_quin Salario_quincenal.salario_quincenal%TYPE;

v_resultado VARCHAR2(100);

BEGIN

FOR i IN c_SalarioCal LOOP

v_sal_quin := (i.salario_mensual/2);

AgregarSalarioQ(i.id_codcolaborador,

(v_sal_quin),

((v_sal_quin)*0.0975),

((v_sal_quin)*0.0125),

((v_sal_quin)-((v_sal_quin)*0.0975)-((v_sal_quin)*0.0125)),
v_resultado);

DBMS_OUTPUT.PUT_LINE(v_resultado);

END LOOP;

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

v_resultado := 'Error en la insercion';

DBMS_OUTPUT.PUT_LINE(v_resultado);

WHEN OTHERS THEN

v_resultado := 'Error al Ejecutar el Procedimiento';

```
DBMS_OUTPUT.PUT_LINE(v_resultado);
```

```
END;
```

```
16      ((v_sal_quin)*0.0975),
17      ((v_sal_quin)*0.0125),
18      ((v_sal_quin)-((v_sal_quin)*0.0975)-((v_sal_quin)*0.0125)), v_res
19      DBMS_OUTPUT.PUT_LINE(v_resultado);
20  END LOOP;
21
22  EXCEPTION
23      WHEN DUP_VAL_ON_INDEX THEN
24          v_resultado := 'Error en la insercion';
25          DBMS_OUTPUT.PUT_LINE(v_resultado);
26      WHEN OTHERS THEN
27          v_resultado := 'Error al Ejecutar el Procedimiento';
28          DBMS_OUTPUT.PUT_LINE(v_resultado);
29
30  END;
31  /
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

--VISTA FINAL

```
CREATE VIEW ResumenSalarios AS
```

```
    SELECT c.id_codcolaborador, c.nombre, c.apellido, c.salario_mensual,
           sq.salario_quincenal, sq.seguro_social, sq.seguro_educativo,
           sq.salario_netto FROM Colaboradores c JOIN Salario_quincenal sq
           ON c.id_codcolaborador = sq.id_codcolaborador;
```

```
SQL> select * from ResumenSalarios;
```

ID_CODCOLABORADOR	NOMBRE	APELLIDO	SALARIO_MENSUAL	SALARIO_QUINCENAL	SEGURO_SOCIAL	SEGURO_EDUCATIVO	SALARIO_NETO
1	MIGUEL	PINILLA	3000	1500	146.25	18.75	1335
2	JUAN	RIVERA	5650	2825	275.44	35.31	2514.25