Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Licenciatura en Ingeniería de Sistemas de Información

Departamento de Sistemas de Información, Control y Evaluación de Recursos Informáticos

Sistemas de Bases de Datos II

Asignación #2 (Cursores)

Facilitador:

Ing. Henry Lezcano

Estudiante:

Reynaldo Rojas H.

Cédula:

8-950-792

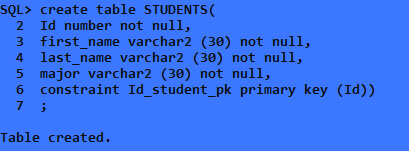
Grupo:

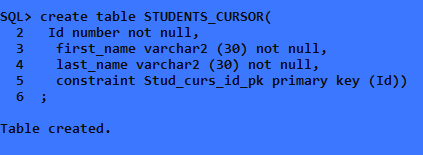
1IF131

II Semestre, 2020

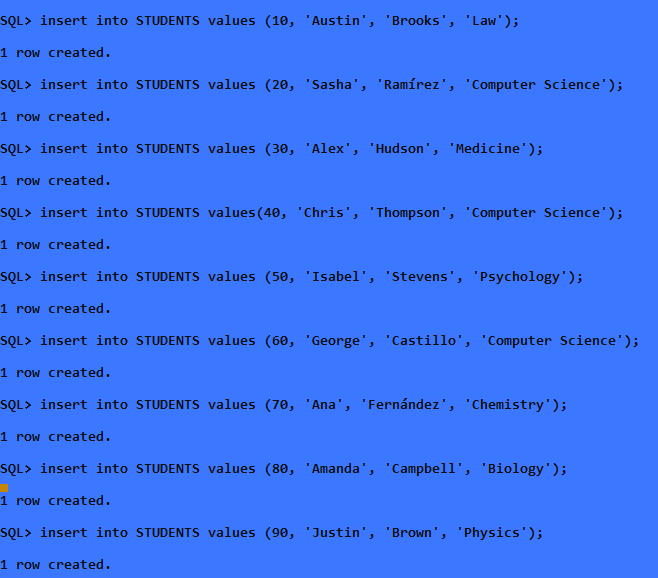
1. Implemente el bloque pl/sql de la ppt No.4, donde en el área de ejecución del  
   bloque, la información extraída por el cursor sea cargada en una relación o tabla  
   diseñada por usted, establezca controles para el proceso la inserción de la  
   información.

**CREACIÓN DE TABLAS**





**INSERCIÓN DE DATOS**



**CÓDIGO FUENTE**

DECLARE

V\_StudentID students.id%TYPE;

V\_FirstName students.first\_name%TYPE;

V\_LastName students.last\_name%TYPE;

V\_Major students.major%TYPE := 'Computer Science';

CURSOR c\_Students IS

select Id, first\_name, last\_name

from STUDENTS

where major = v\_major;

BEGIN

OPEN c\_Students;

LOOP

FETCH c\_Students INTO v\_StudentID, v\_FirstName, v\_LastName;

EXIT WHEN c\_Students%NOTFOUND;

INSERT INTO STUDENTS\_CURSOR values (v\_StudentID, v\_FirstName, v\_LastName);

DBMS\_OUTPUT.PUT\_LINE (v\_StudentID || ' ' || v\_FirstName || ' ' || v\_LastName);

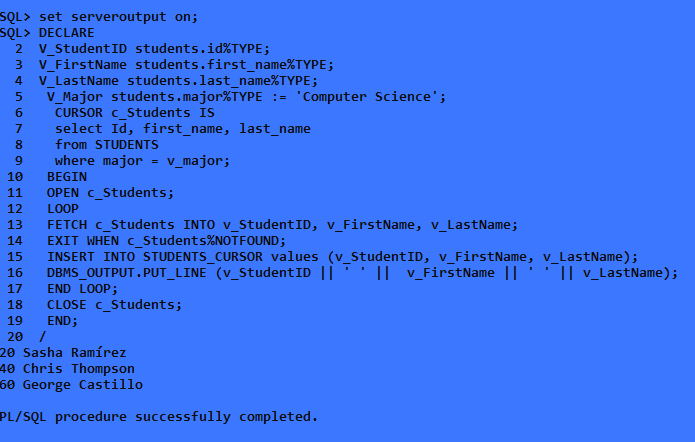
END LOOP;

CLOSE c\_Students;

END;

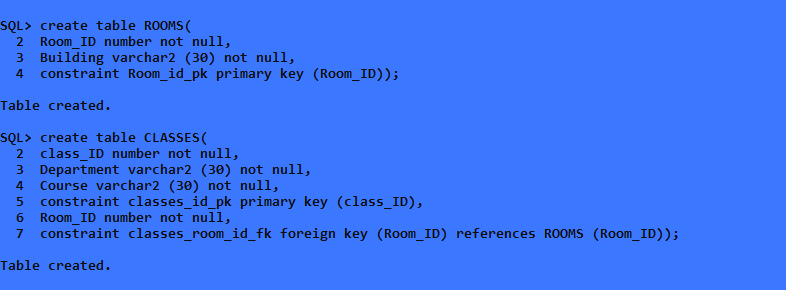
/

**EJECUCIÓN DEL CÓDIGO**



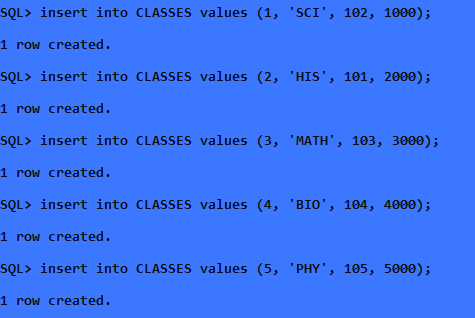
1. Complete el bloque pl/sql de la ppt No.8 de manera que podamos observar que  
   información está extrayendo el cursor basado en variables de acoplamiento.

**CREACIÓN DE TABLAS**



**INSERCIÓN DE DATOS**





**CÓDIGO FUENTE**

DECLARE

V\_RoomID classes.room\_id%TYPE;

V\_Building rooms.building%TYPE;

V\_Department classes.department%TYPE;

V\_Course classes.course%TYPE;

CURSOR c\_Building IS

SELECT building

FROM rooms, classes

WHERE rooms.room\_id = classes.room\_id

AND department = V\_Department

AND course = V\_Course;

BEGIN

V\_Department := 'HIS';

V\_Course := 101;

OPEN c\_Building;

FETCH c\_Building INTO V\_Building;

v\_Department := 'XXX';

V\_Course := -1;

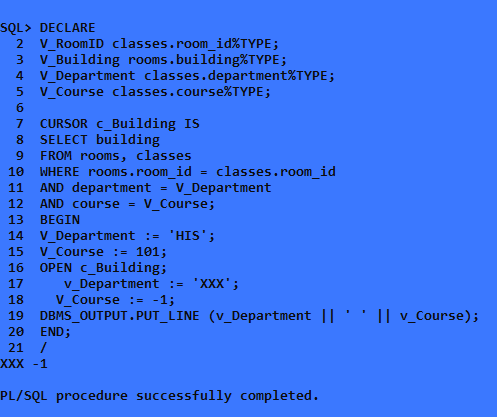
CLOSE c\_Building;

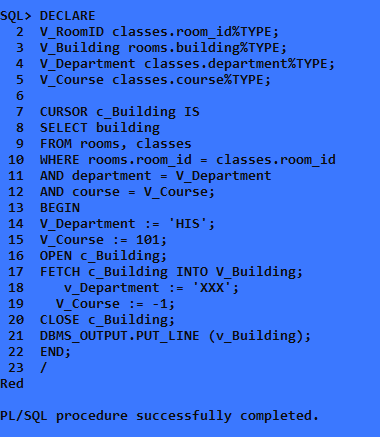
DBMS\_OUTPUT.PUT\_LINE (v\_Building);

END;

/

**EJECUCIÓN DEL CÓDIGO**





1. Modifique el bloque anterior aplicando el concepto de cursores parametrizados.

**CÓDIGO FUENTE**

DECLARE

V\_RoomID classes.room\_id%TYPE;

V\_Building rooms.building%TYPE;

V\_Department classes.department%TYPE;

V\_Course classes.course%TYPE;

CURSOR c\_Building (V\_Department classes.department%TYPE, V\_Course classes.course%TYPE) IS

SELECT building

FROM rooms, classes

WHERE rooms.room\_id = classes.room\_id

AND department = V\_Department

AND course = V\_Course;

BEGIN

OPEN c\_Building ('HIS', 101);

FETCH c\_Building INTO V\_Building;

v\_Department := 'XXX';

V\_Course := -1;

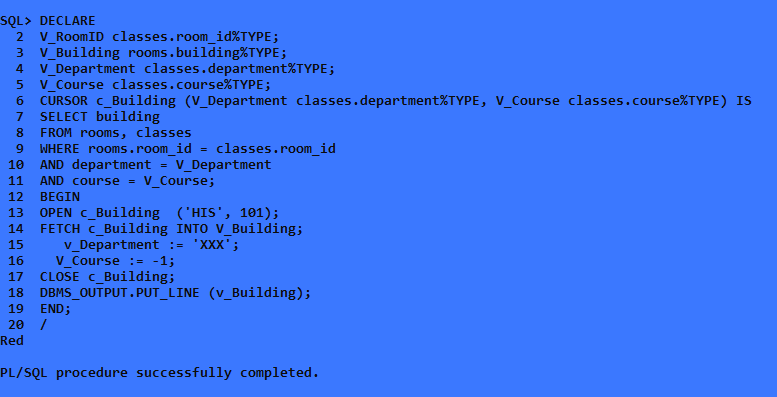
CLOSE c\_Building;

DBMS\_OUTPUT.PUT\_LINE (v\_Building);

END;

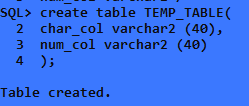
/

**EJECUCIÓN DEL CÓDIGO**



1. Implemente el bloque pl/sql de la **ppt No.17** donde se valida el uso de los atributos  
   para los cursores implícitos y la cláusula SELECT.

**CREACIÓN DE TABLAS**



**CÓDIGO FUENTE**

DECLARE

V\_RoomData rooms%ROWTYPE;

BEGIN

SELECT \* INTO V\_RoomData

FROM rooms

WHERE room\_id = -1;

IF SQL%NOTFOUND THEN

INSERT INTO temp\_table (char\_col) VALUES ('Not Found’);

END IF;

EXCEPTION

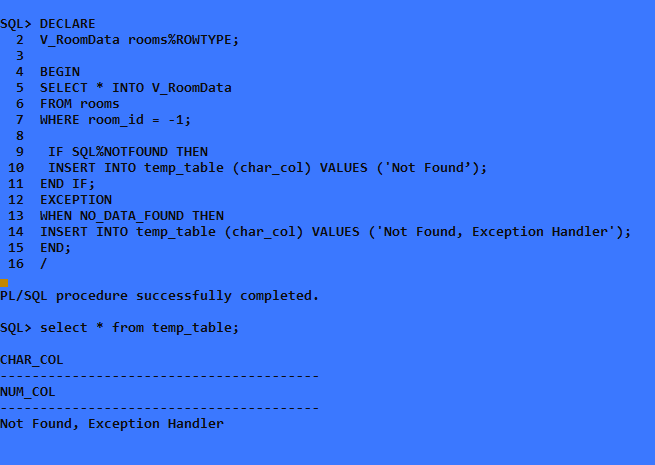
WHEN NO\_DATA\_FOUND THEN

INSERT INTO temp\_table (char\_col) VALUES ('Not Found, Exception Handler');

END;

/

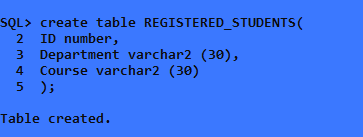
**EJECUCIÓN DEL CÓDIGO**



El resultado fue Not Found, Exception Handler porque no le ingresé ningún dato a la tabla y no encontró nada.

1. Implementos lo bloques pl/sql que se detalla en las **ppt No.19, 20, 21, 22** que  
   aplicación el ciclo de repetición integrado a los cursores basados reglas  
   establecidas.

**CREACIÓN DE TABLAS**



**CÓDIGO FUENTE PPT19**

DECLARE

V\_StudentID students.id%TYPE;

V\_FirstName students.first\_name%TYPE;

V\_LastName students.last\_name%TYPE;

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major= 'History';

BEGIN

OPEN c\_HistoryStudents;

LOOP

FETCH c\_HistoryStudents INTO v\_StudentID, v\_FirstName, v\_LastName;

EXIT WHEN c\_HistoryStudents%NOTFOUND;

INSERT INTO registered\_students (ID, department, course)

VALUES (v\_StudentID, 'HIS', 301);

INSERT INTO temp\_table (num\_col, char\_col)

VALUES (V\_StudentID, V\_FirstName || ' ' || v\_LastName);

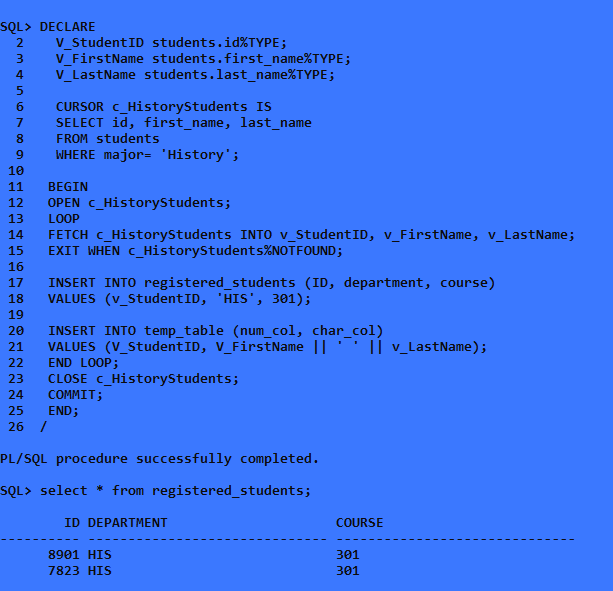
END LOOP;

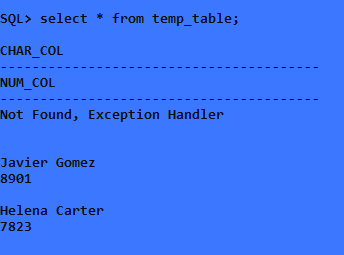
CLOSE c\_HistoryStudents;

COMMIT;

END;

**EJECUCIÓN DE CÓDIGO PPT19**





**CÓDIGO FUENTE PPT20**

DECLARE

V\_StudentID students.id%TYPE;

V\_FirstName students.first\_name%TYPE;

V\_LastName students.last\_name%TYPE;

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major= 'History';

BEGIN

OPEN c\_HistoryStudents;

LOOP

FETCH c\_HistoryStudents INTO v\_StudentID, v\_FirstName, v\_LastName;

INSERT INTO registered\_students (ID, department, course)

VALUES (v\_StudentID, 'HIS', 301);

INSERT INTO temp\_table (num\_col, char\_col)

VALUES (V\_StudentID, V\_FirstName || ' ' || v\_LastName);

EXIT WHEN c\_HistoryStudents%NOTFOUND;

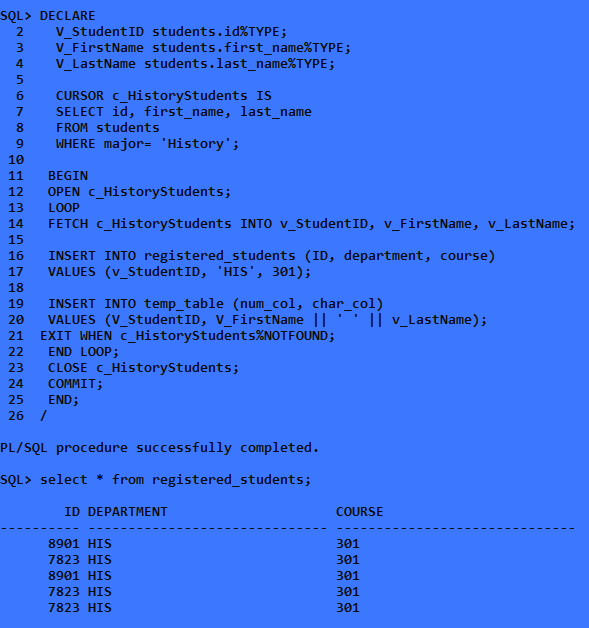
END LOOP;

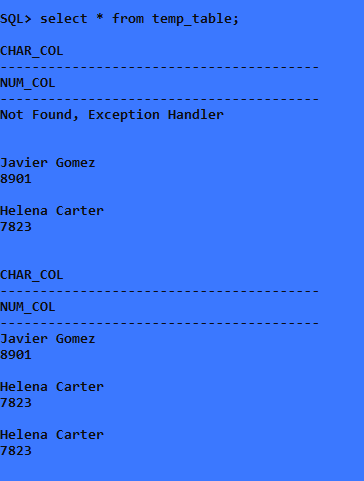
CLOSE c\_HistoryStudents;

COMMIT;

END;

**EJECUCIÓN DE CÓDIGO PPT20**





**CÓDIGO FUENTE PPT21**

DECLARE

CURSOR c\_HistoryStudents IS

SELECT ID, first\_name, last\_name

FROM students

WHERE major = 'History';

v\_StudentData c\_HistoryStudents%ROWTYPE;

BEGIN

OPEN c\_HistoryStudents;

FETCH c\_HistoryStudents INTO v\_StudentData;

WHILE c\_HistoryStudents%FOUND LOOP

INSERT INTO registered\_students (ID, department, course)

VALUES (v\_StudentData.ID, 'HIS', 301);

INSERT INTO temp\_table (num\_col, char\_col)

VALUES (v\_StudentData.ID, V\_StudentData.first\_name || ' ' || v\_StudentData.last\_name);

FETCH c\_HistoryStudents INTO V\_StudentData;

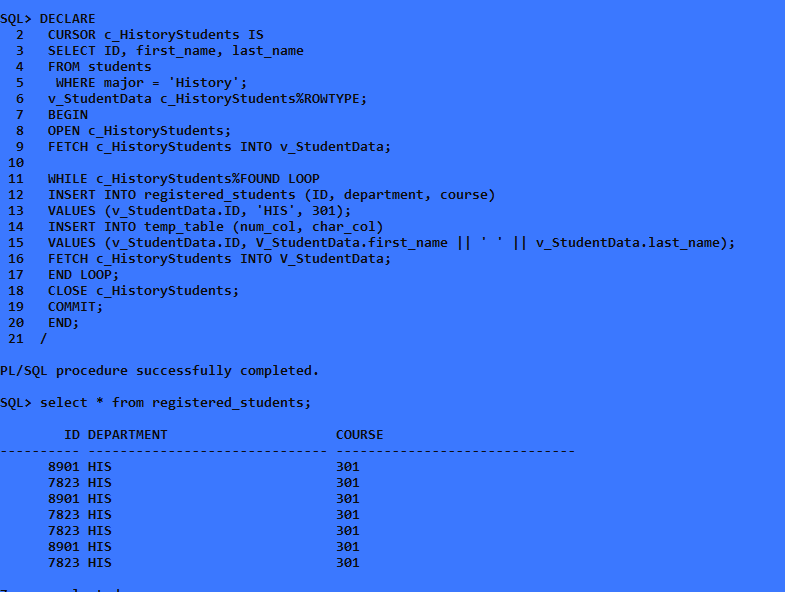
END LOOP;

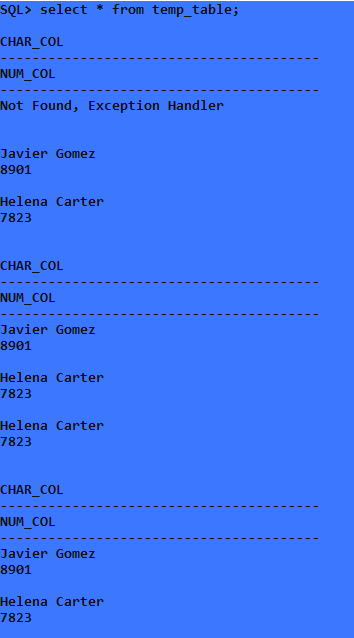
CLOSE c\_HistoryStudents;

COMMIT;

END;

**EJECUCIÓN DE CÓDIGO PPT21**





**CÓDIGO FUENTE PPT22**

DECLARE

CURSOR c\_HistoryStudent IS

SELECT id, first\_name, last\_name

FROM students

WHERE major = 'History';

BEGIN

FOR v\_StudentData IN c\_HistoryStudent LOOP

INSERT INTO registered\_students (ID, department, course)

VALUES (v\_StudentData.ID, 'HIS', 301);

INSERT INTO temp\_table (num\_col, char\_col)

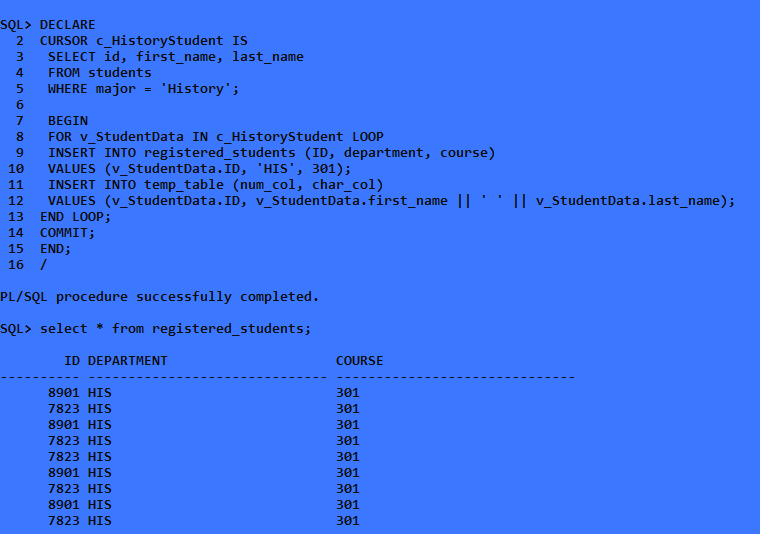
VALUES (v\_StudentData.ID, v\_StudentData.first\_name || ' ' || v\_StudentData.last\_name);

END LOOP;

COMMIT;

END;

**EJECUCIÓN DE CÓDIGO PPT22**

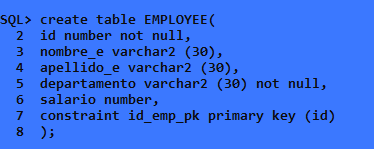


Aquí se muestra todos los resultados de los inserts de los loops de las ppt 19, 20, 21 y 22.

1. Diseñe bloque pl/sql que extraiga información de una relación o tabla de base de  
   datos por medio de un cursor y esta se actualizada en el área de ejecución del  
   proceso estableciendo los controles para el proceso de actualización.

**En este problema cree una tabla llamada Employee donde le inserte valores de id, nombre, apellido, departamento y salario. El programa lo que hace es que actualiza 100 dólares de bono a los empleados del departamento de Marketing.**

**CREACIÓN DE TABLAS**



**CÓDIGO FUENTE**

DECLARE

V\_ID employee.id%TYPE;

V\_nombre employee.nombre\_e%TYPE;

V\_apellido employee.apellido\_e%TYPE;

V\_departamento employee.departamento%TYPE := 'Marketing';

V\_salario employee.salario%TYPE;

CURSOR emp\_sal IS

SELECT id, nombre\_e, apellido\_e, salario

FROM Employee

where departamento = V\_departamento;

BEGIN

OPEN emp\_sal;

LOOP

FETCH emp\_sal INTO V\_ID, V\_nombre, V\_apellido, V\_salario;

EXIT WHEN emp\_sal%NOTFOUND;

UPDATE employee

SET salario = salario+100

WHERE departamento = 'Marketing';

END LOOP;

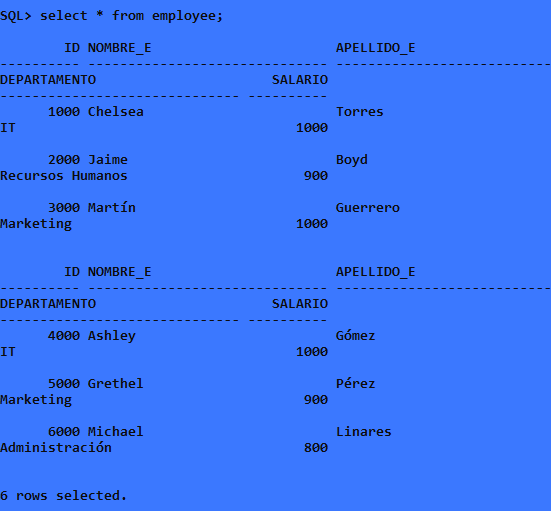
CLOSE emp\_sal;

COMMIT;

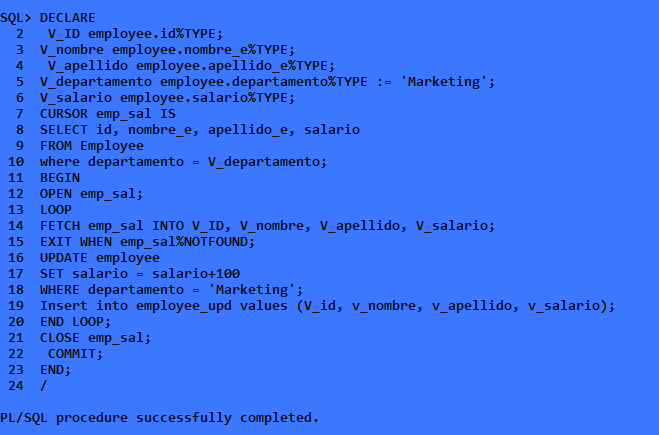
END;

/

**TABLA NORMAL**



**EJECUCIÓN DEL CÓDIGO**



**TABLA ACTUALIZADA**

