**UNIVERSIDAD TECNOLOGICA DE PANAMA**

**FACULTAD EN INGENIERIA DE SISTEMAS COMPUTACIONALES**

**LICENCIATURA EN INGENIERIA SISTEMAS DE INFORMACION**

**SISTEMAS DE BASE DE DATOS II**

**ASIGNACION NO.2 ‘CURSORES’**

**Nombre:** Milagros Aguilar **Cédula:** 3-740-771 **Grupo:** 1IF131

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.**

create table Students(

id varchar2(25) not null,

first\_name varchar2(20) not null,

last\_name varchar2(20) not null,

major varchar2(40) not null,

constraint pk\_Students\_id primary key (id)

);

create table Students2(

id varchar2(25) not null,

first\_name varchar2(20) not null,

last\_name varchar2(20) not null,

constraint pk\_Students2\_id primary key (id)

);

insert into Students values ('3-740-771', 'Ana', 'González', 'Computer Science');

insert into Students values ('8-950-5543', 'Hector', 'Rodriguez', 'Computer Science');

insert into Students values ('8-544-322', 'Inés', 'Gómez', 'Computer Science');

insert into Students values ('3-932-567', 'Iván', 'Atencio', 'Computer Science');

insert into Students values ('8-678-099', 'Jaime', 'López', 'Computer Science');

insert into Students values ('4-78-544', 'Amy', 'Aguilar', 'Chemistry');

insert into Students values ('9-432-112', 'Ernesto', 'Guerra', 'Physics');

insert into Students values ('8-903-778', 'Arnold', 'Lee', 'Chemistry');

insert into Students values ('5-87-555', 'Arturo', 'Johnson', 'Chemistry');

insert into Students values ('7-532-899', 'Rubén', 'Salinas', 'Physics');

DECLARE

/\* Variable de salida para almacenar los resultados de la Consulta \*/

v\_StudentID students.id%TYPE;

v\_FirstName students.first\_name%TYPE;

v\_LastName students.last\_name%TYPE;

/\* Valores de acoplamiento utilizado en la consulta \*/

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

/\* Declaración del Cursor\*/

CURSOR c\_Students IS

SELECT id, first\_name, last\_name

FROM Students

WHERE major = v\_Major;

BEGIN

/\* Identificar las filas en el conjunto activo y preparar el procesamiento ulterior de los datos \*/

OPEN c\_Students;

LOOP

/\* Recuperar cada fila del conjunto active y almacenarlos en las variables PL/SQL \*/

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

/\* SI no hay más filas que recuperar, salir del bucle \*/

EXIT WHEN c\_Students%NOTFOUND;

insert into Students2 (id, first\_name, last\_name) values (v\_StudentID, v\_FirstName, v\_LastName);

dbms\_output.put\_line(v\_StudentID);

dbms\_output.put\_line(v\_FirstName ||' '||v\_LastName );

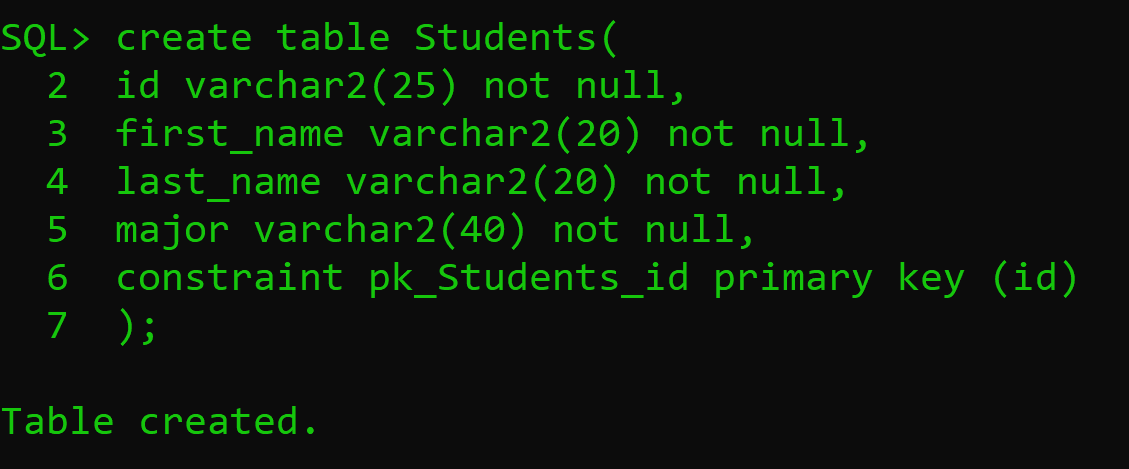
END LOOP;

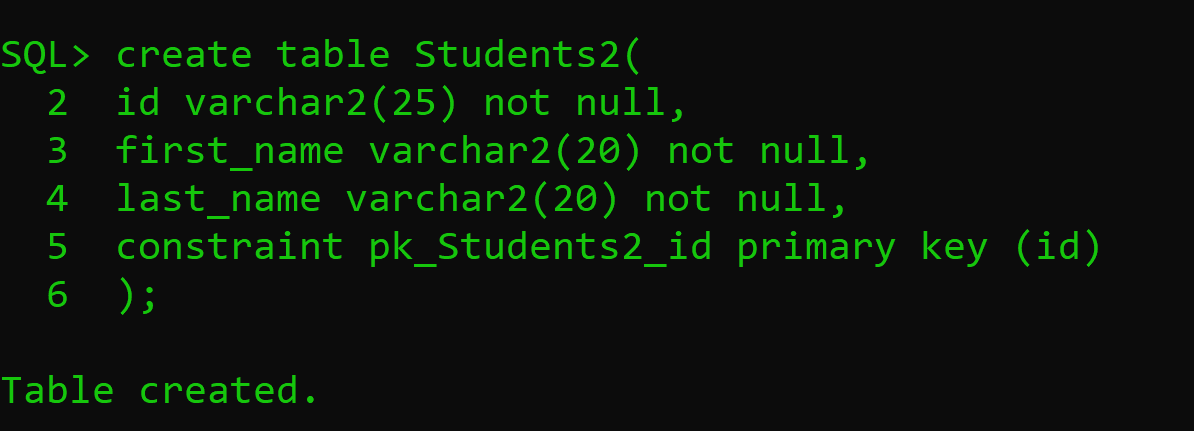
/\* Liberar los recursos usados para la consulta \*/

CLOSE c\_Students;

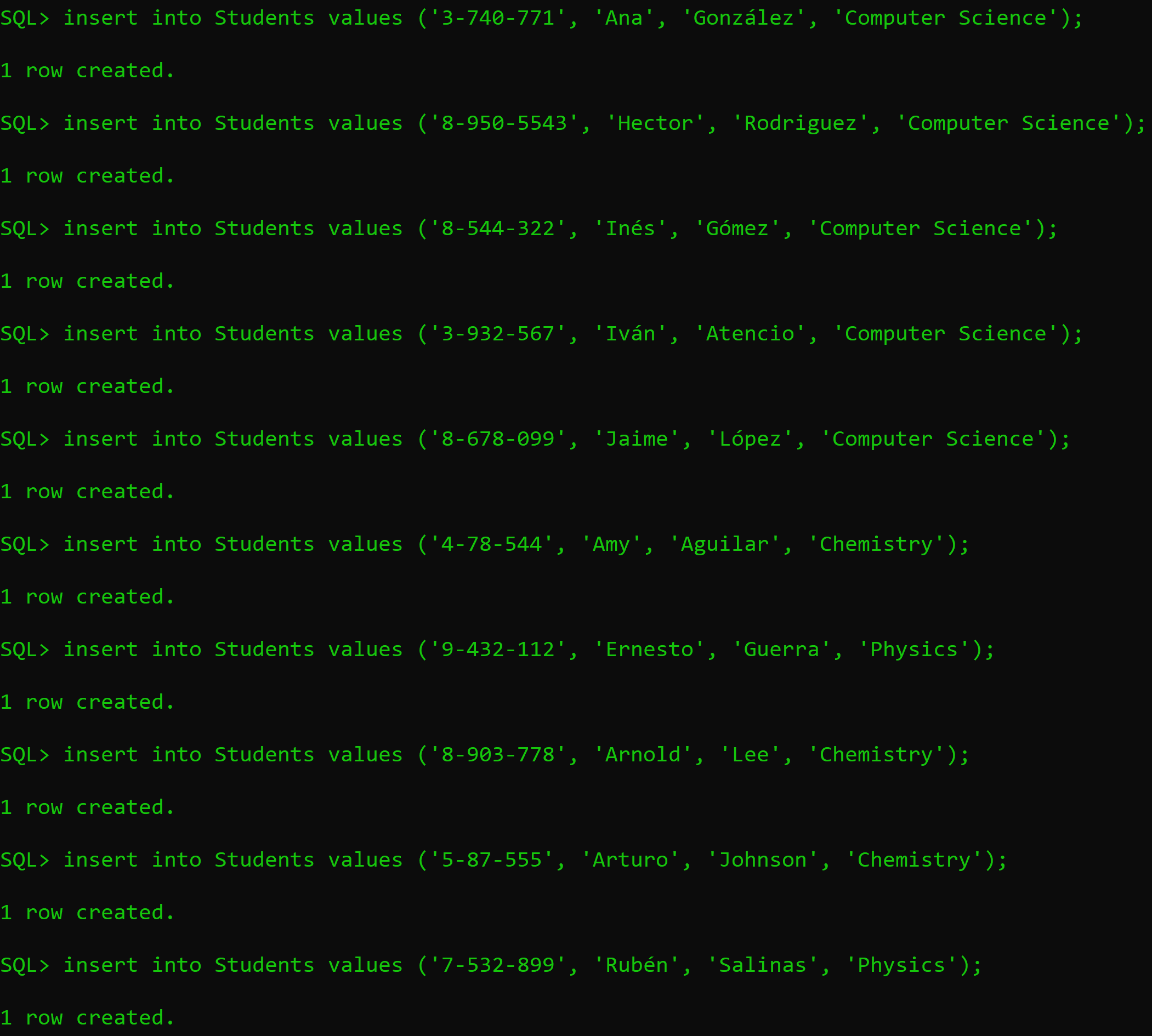
END;

**Creación de tablas:**

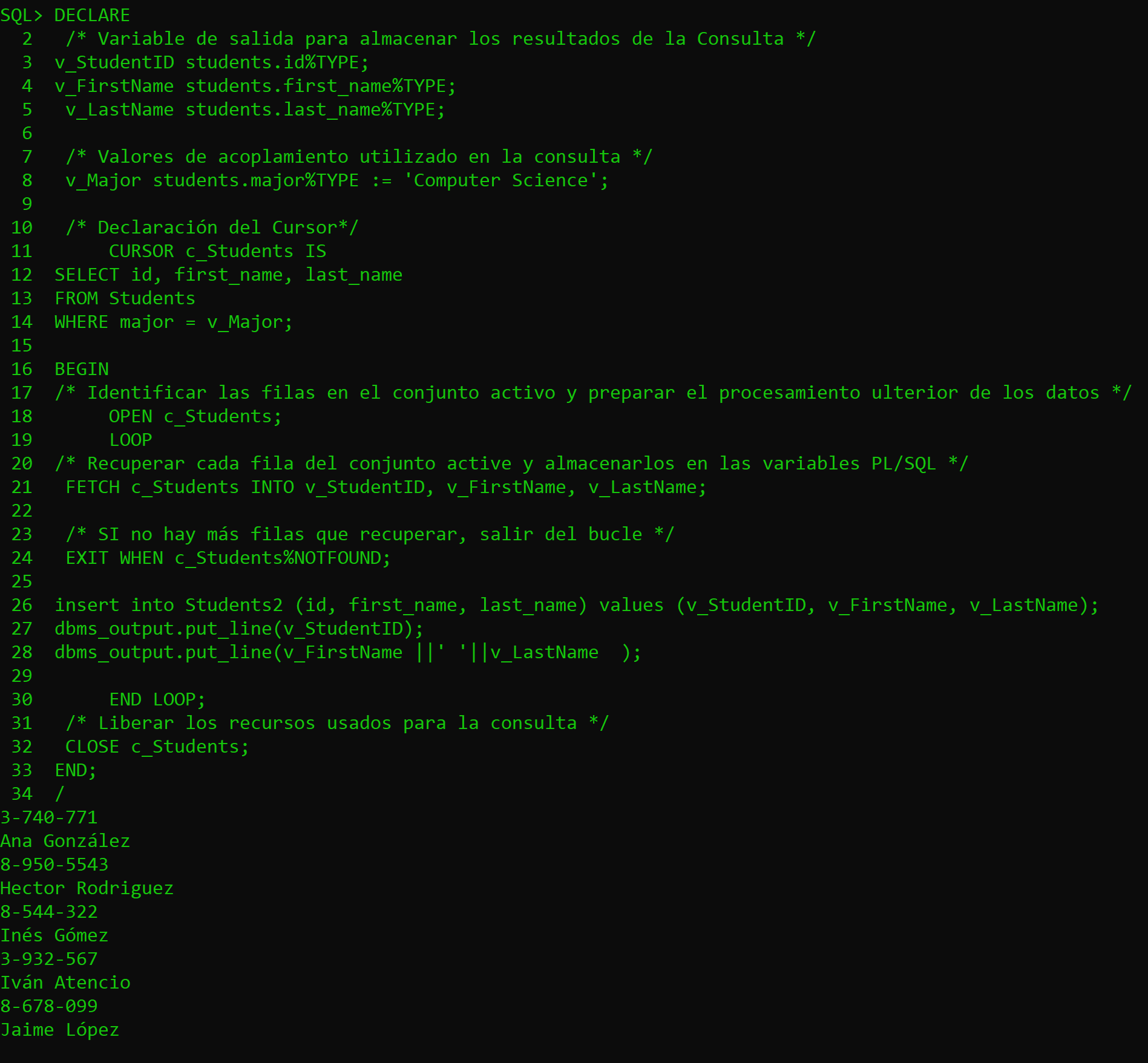




**Inserción de datos:**



**Bloque PL/SQL en ejecución:**



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.**

create table Rooms(

room\_id varchar2(20) not null,

building varchar2(5) not null,

constraint pk\_Rooms\_room\_id primary key (room\_id)

);

create table Classes(

room\_id varchar2(20) not null,

department varchar2(5),

course varchar2(25),

constraint fk\_Classes\_room\_id foreign key (room\_id)

references Rooms (room\_id),

constraint pk\_Classes\_room\_id primary key (room\_id)

);

insert into Rooms values('A-1', 'A');

insert into Rooms values('B-1', 'B');

insert into Rooms values('C-1', 'C');

insert into Rooms values('A-2', 'A');

insert into Rooms values('B-2', 'B');

insert into Rooms values('C-2', 'C');

insert into Classes values('A-1', 'CS', '100');

insert into Classes values('B-1', 'HIS', '101');

insert into Classes values('C-1', 'PHY', '102');

insert into Classes values('A-2', 'CHE', '103');

insert into Classes values('B-2', 'MAT', '104');

insert into Classes values('C-2', 'ENG', '105');

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

-- Asignar las variables de Acoplamiento antes de abril el cursor

v\_Department := 'HIS';

v\_Course := 101;

-- Abril el Cursor

OPEN c\_Building;

FETCH c\_Building into v\_Building;

-- Reasignar las variables de acoplamiento – No tienen efecto alguno, ya que el cursor esta abierto

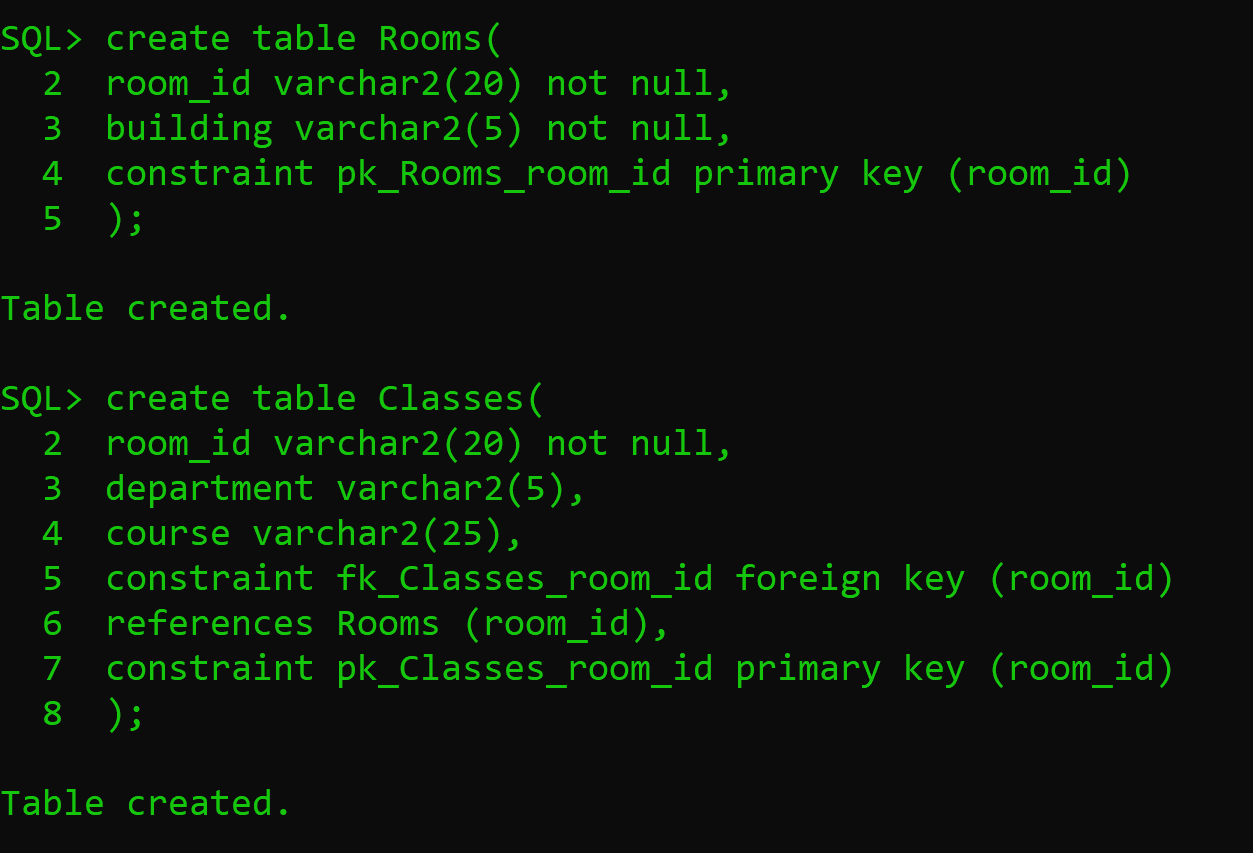
v\_Department := 'XXX';

v\_Course := -1;

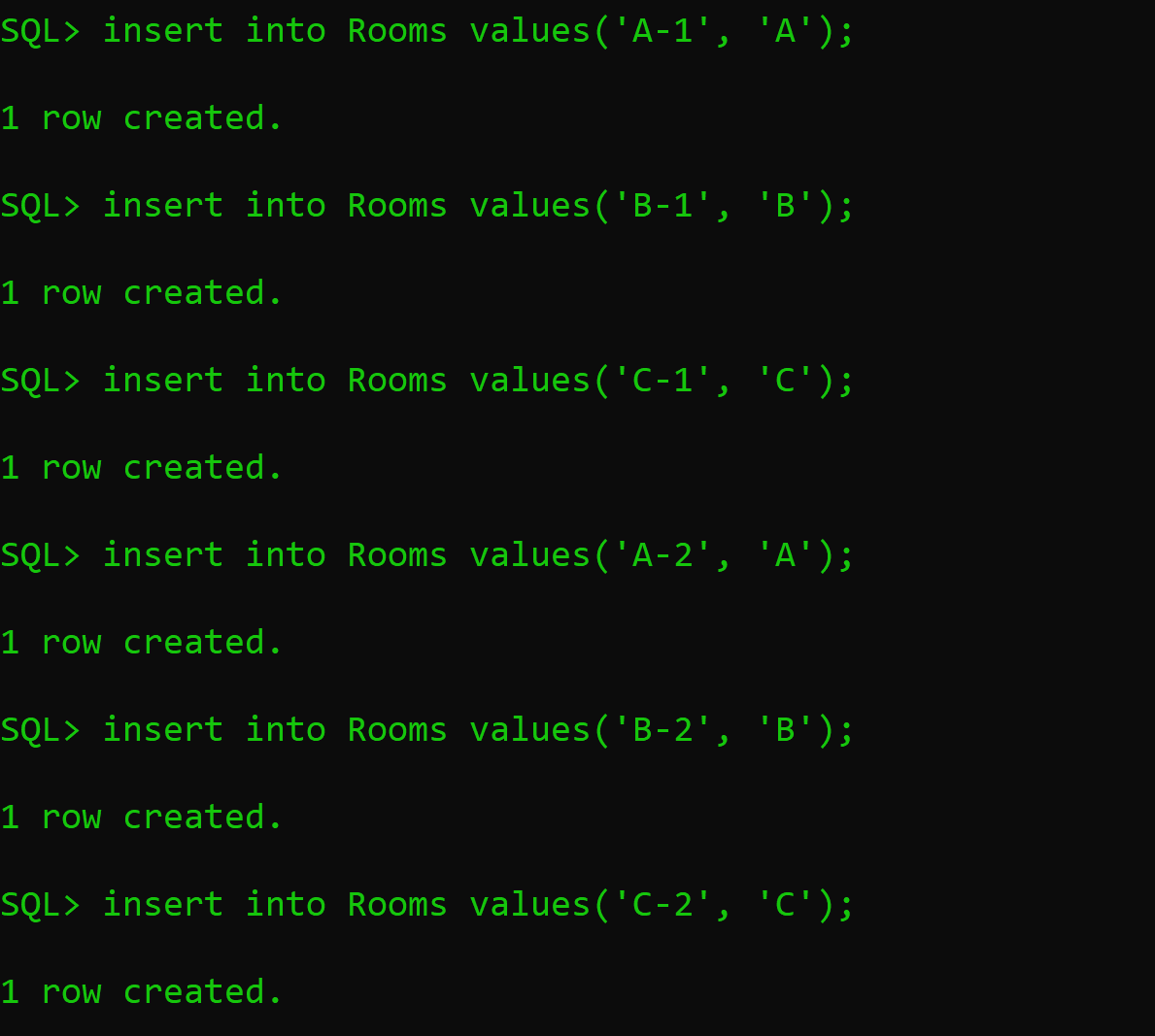
dbms\_output.put\_line(v\_Building);

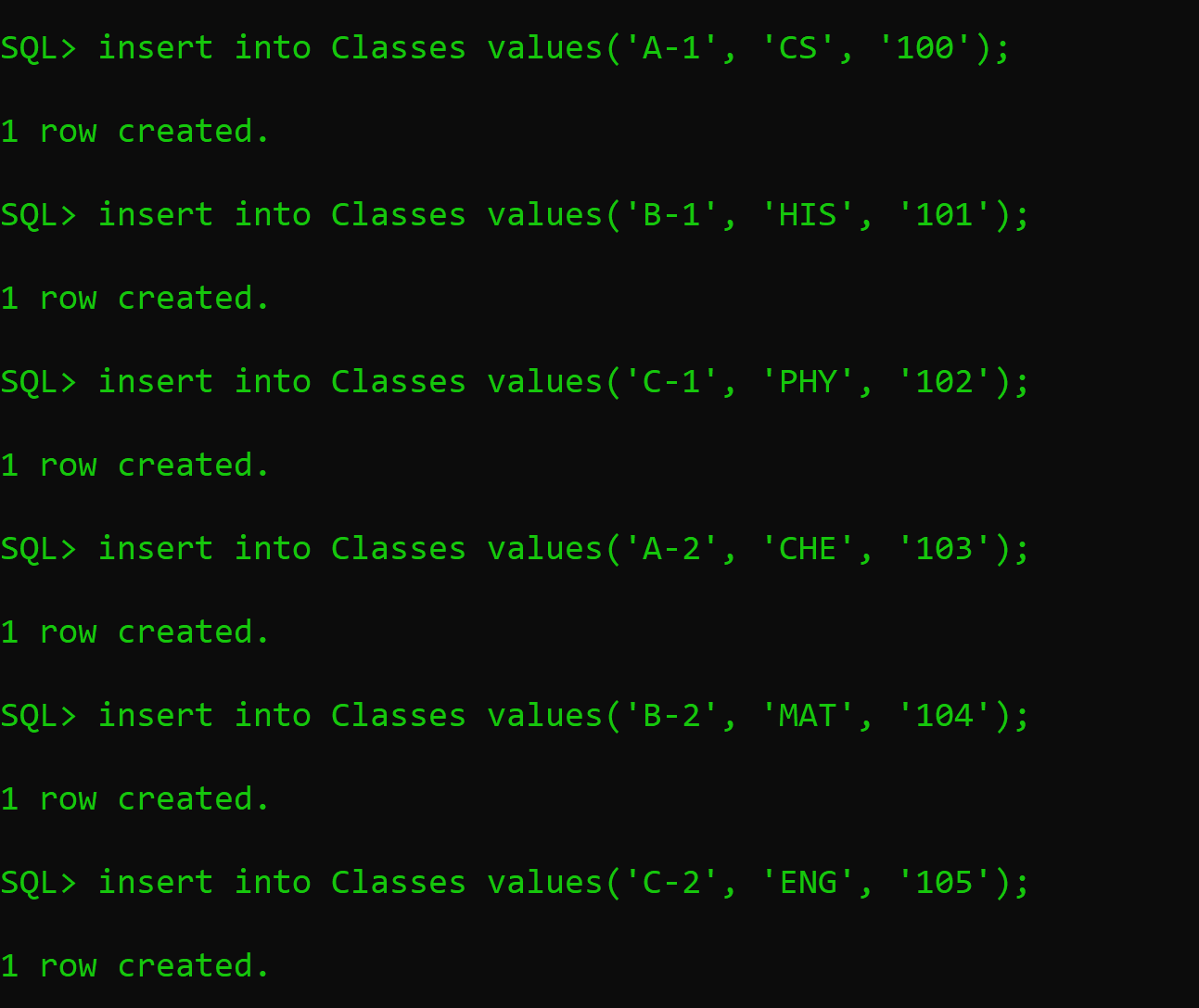
END;

**Creación de tablas:**

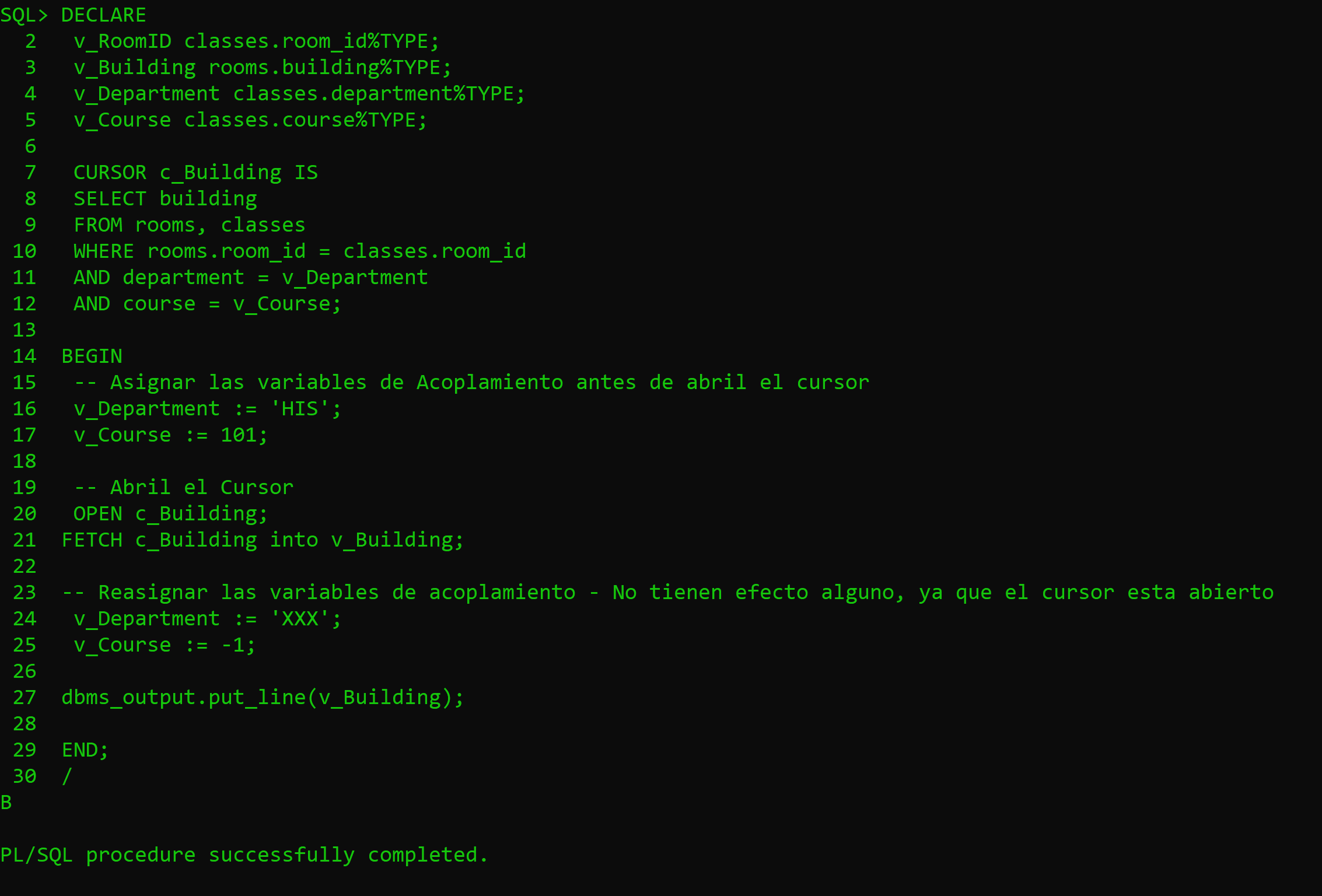


**Inserción de datos:**





**Bloque PL/SQL en ejecución:**



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

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

-- Asignar las variables de Acoplamiento antes de abril el cursor

v\_Department := 'HIS';

v\_Course := 101;

-- Abril el Cursor

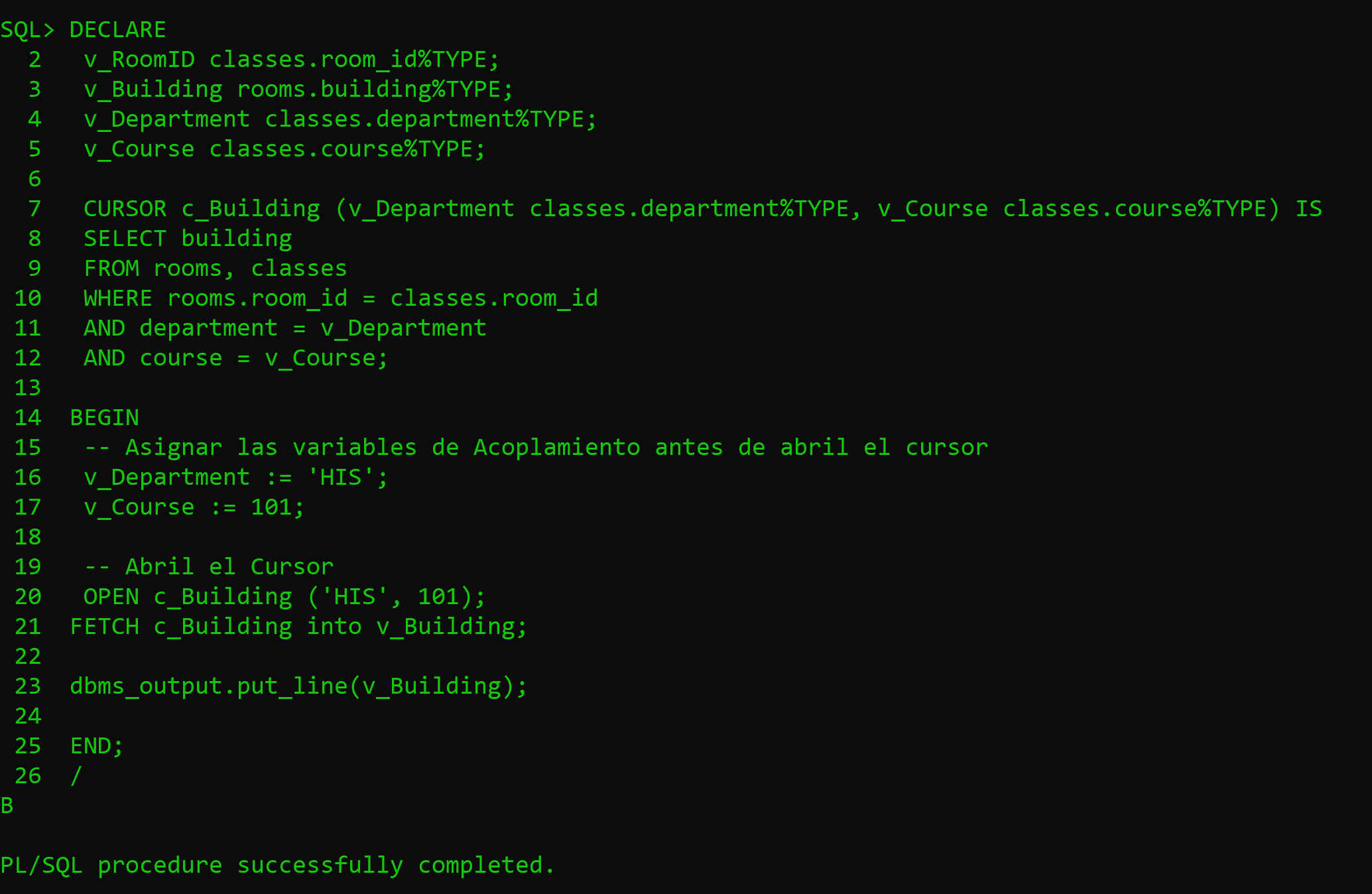
OPEN c\_Building ('HIS', 101);

FETCH c\_Building into v\_Building;

dbms\_output.put\_line(v\_Building);

END;

**Bloque PL/SQL en ejecución:**



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.**

create table temp\_table(

num\_col varchar2(50),

char\_col char(50)

);

DECLARE

-- Registro para almacenar la información acerca de una clase.

v\_RoomData rooms%ROWTYPE;

BEGIN

-- Extraer la información sobre la clase ID -1

SELECT \* INTO v\_RoomData

FROM rooms

WHERE room\_id = -1;

/\* La siguiente orden no se ejecutará nunca, ya que el control pasa inmediatamente al gestor de excepciones \*/

IF SQL%NOTFOUND THEN

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

END IF;

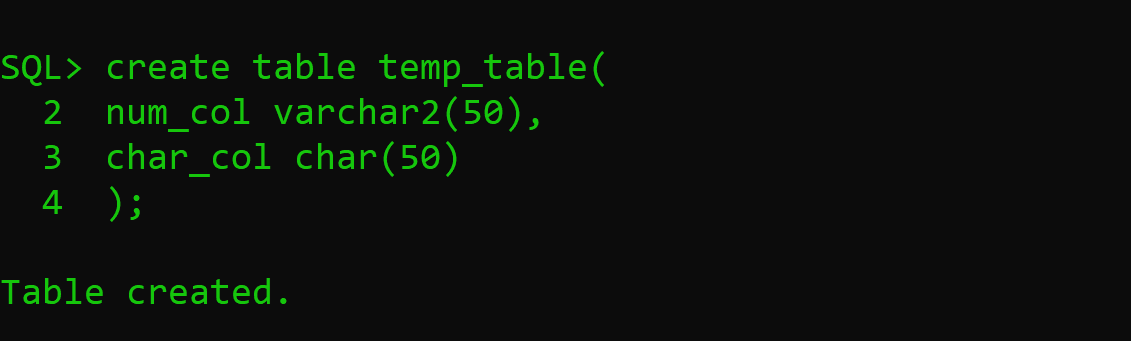
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

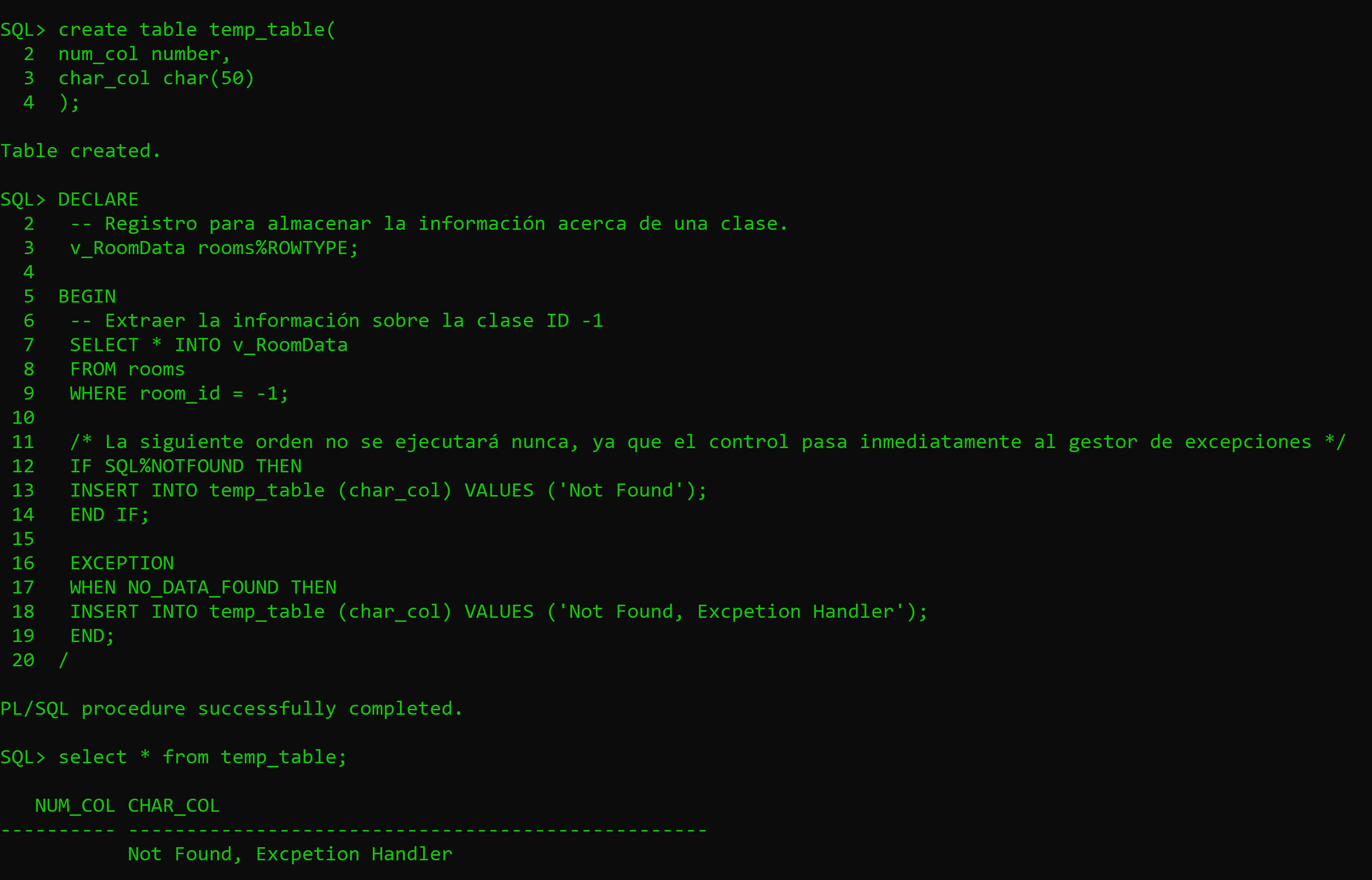
INSERT INTO temp\_table (char\_col) VALUES ('Not Found, Excpetion Handler');

END;

**Creación de tabla:**



**Bloque PL/SQL en ejecución:**



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 basado reglas establecidas.**

insert into Students values ('8-55-422', 'Aldo', 'Martinez', 'History');

insert into Students values ('8-543-776', 'Esteban', 'Pardo', 'History');

**No. 19 – Bucle Simple**

create table registered\_students(

id varchar2(25) not null,

department varchar2(25),

course varchar2(25),

constraint fk\_reg\_students\_id foreign key (id)

references Students(id)

);

DECLARE

/\* Declaración de variables para almacenar información acerca de los estudiantes que cursan la especialidad de Historia \*/

v\_StudentID students.id%TYPE;

v\_FirstName students.first\_name%TYPE;

v\_LastName students.last\_name%TYPE;

-- Cursor para recuperar la informacion sobre los estudiantes de Historia

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major = 'History';

BEGIN

-- Abre el cursor e inicializa el conjunto activo

OPEN c\_HistoryStudents;

LOOP

-- Recupera la información del siguiente estudiante

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

-- Salida del bucle cuando no hay más filas por recuperar

EXIT WHEN c\_HistoryStudents%NOTFOUND;

/\* Procesa las filas recuperadas. En este caso matricula a cada estudiante en Historia 301, insertándolo en la tabla registered\_students.

Registra también el nombre y el apellido en la tabla temp\_table \*/

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;

-- Libera los recursos utilizados por el curso

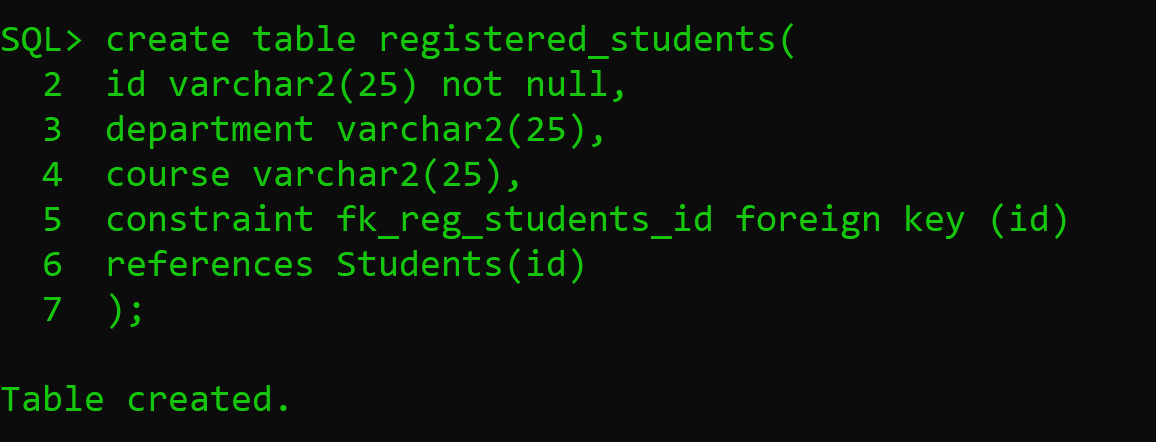
CLOSE c\_HistoryStudents;

-- Confirmamos el trabajo

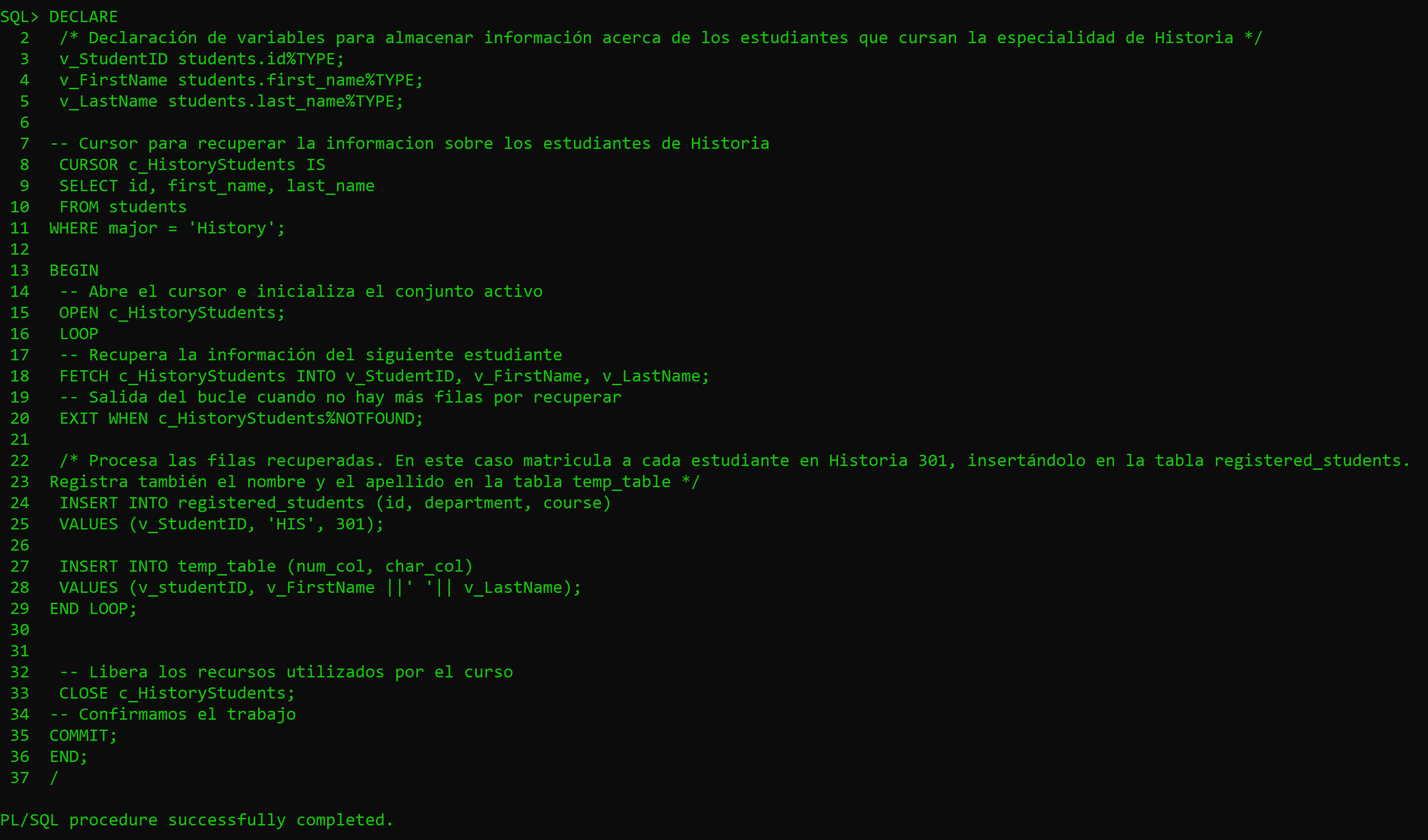
COMMIT;

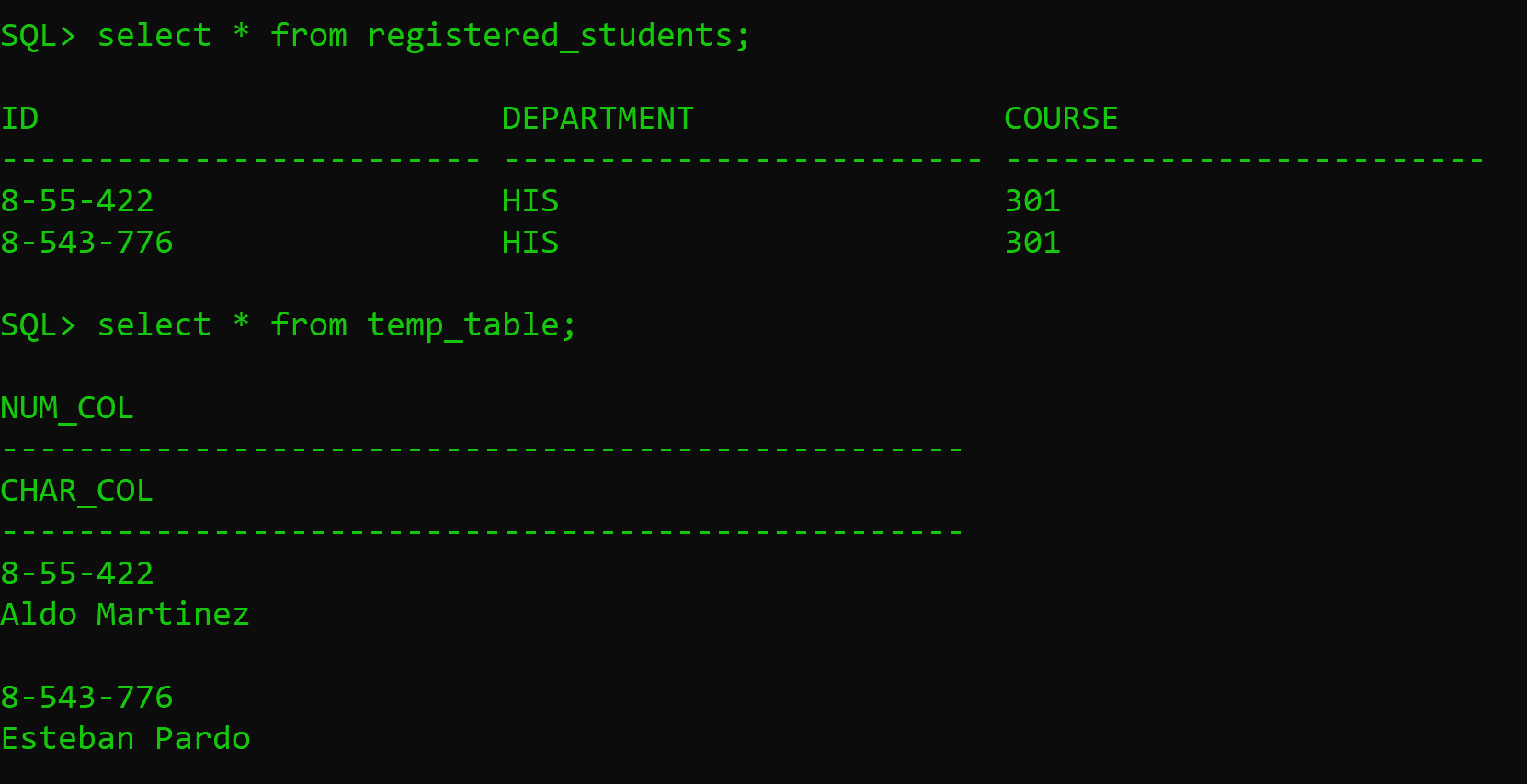
END;

**Creación de la tabla:**



**Bloque PL/SQL en ejecución:**





**No. 20 – Bucle Simple**

DECLARE

/\* Declaración de variables para almacenar información acerca de los estudiantes que cursan la especialidad de Historia \*/

v\_StudentID students.id%TYPE;

v\_FirstName students.first\_name%TYPE;

v\_LastName students.last\_name%TYPE;

-- Cursor para recuperar la información sobre los estudiantes de Historia

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major = 'History';

BEGIN

-- Abre el cursor e inicializa el conjunto activo

OPEN c\_HistoryStudents;

LOOP

-- Recupera la información del siguiente estudiante

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

/\* Procesa las filas recuperadas. En este caso matricula a cada estudiante en Historia 301, insertándolo en la tabla registered\_students.

Registra también el nombre y el apellido en la tabla temp\_table \*/

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);

-- Salida del bucle cuando no hay más filas por recuperar

EXIT WHEN c\_HistoryStudents%NOTFOUND;

END LOOP;

-- Libera los recursos utilizados por el curso

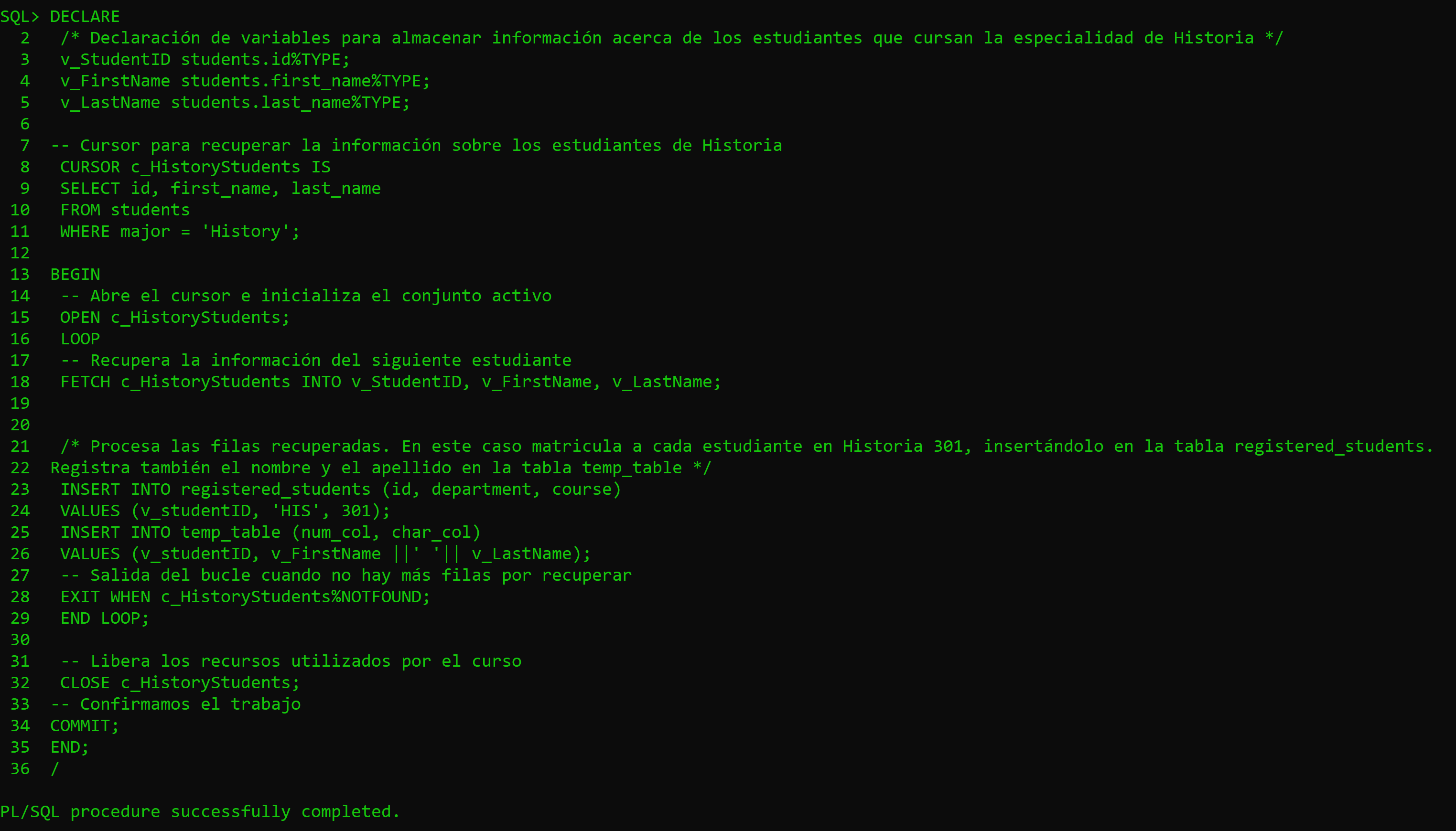
CLOSE c\_HistoryStudents;

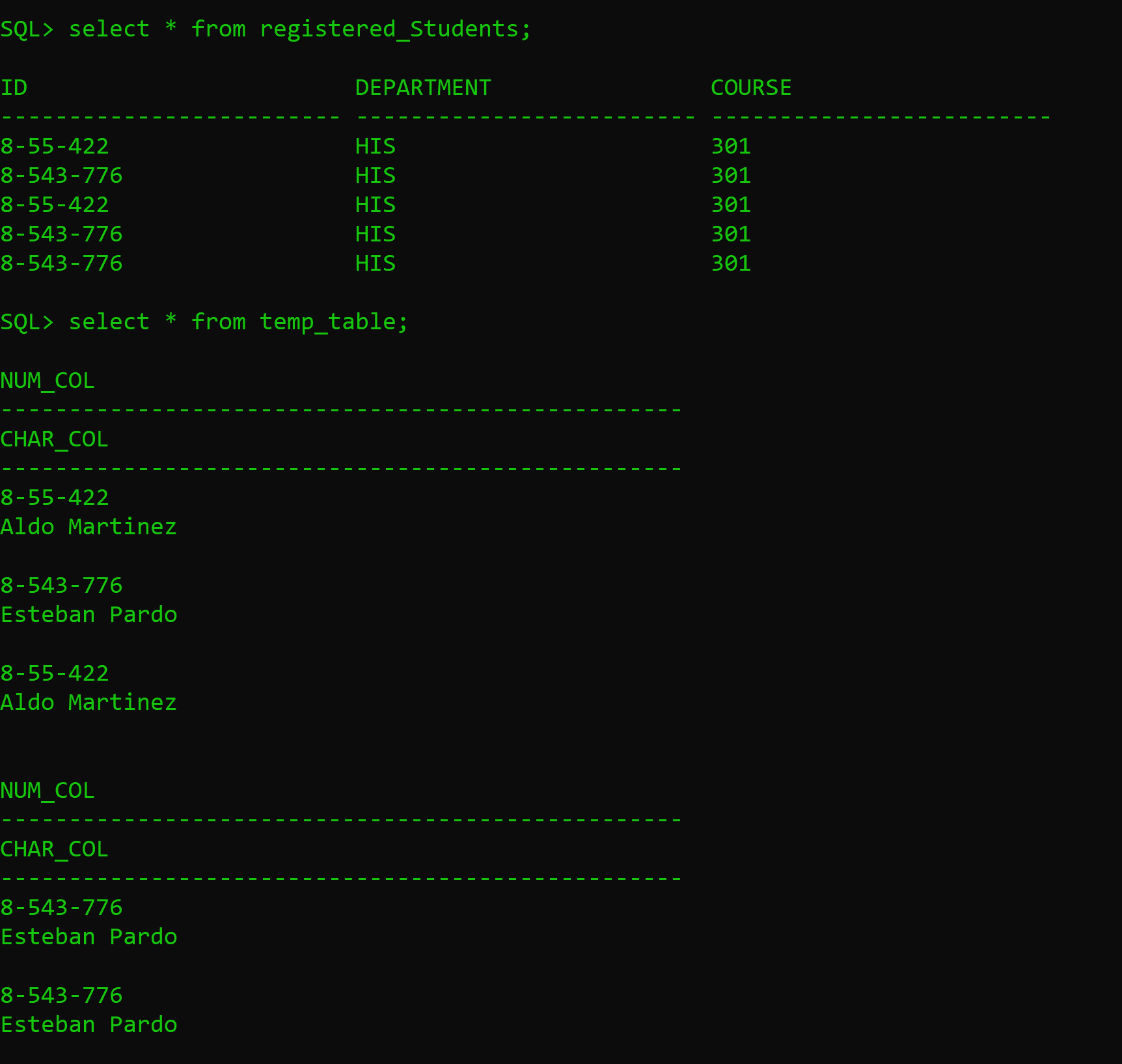
-- Confirmamos el trabajo

COMMIT;

END;

**Bloque PL/SQL en ejecución:**





**No. 21 – Bucle While**

DECLARE

-- Cursor para recuperar la información sobre los estudiantes de Historia

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major = 'History';

-- Declaración el registro para almacenar información extraída

v\_StudentData c\_HistoryStudents%ROWTYPE;

BEGIN

-- Abre el cursor e inicializa el conjunto activo

OPEN c\_HistoryStudents;

-- Recupera la información del siguiente estudiante

FETCH c\_HistoryStudents INTO v\_StudentData;

-- El bucle continua mientras haya mas filas que extraer

WHILE c\_HistoryStudents%FOUND LOOP

/\* Procesa las filas recuperadas. En este caso matricula a cada estudiante en Historia 301, insertándolo en la tabla

registered\_students. Registra también el nombre y el apellido en la tabla temp\_table \*/

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);

-- Recuperar la fila siguiente. La condición %FOUND se comprobara antes de que el bucle continúe

FETCH c\_HistoryStudents INTO v\_StudentData;

END LOOP;

-- Libera los recursos utilizados por el curso

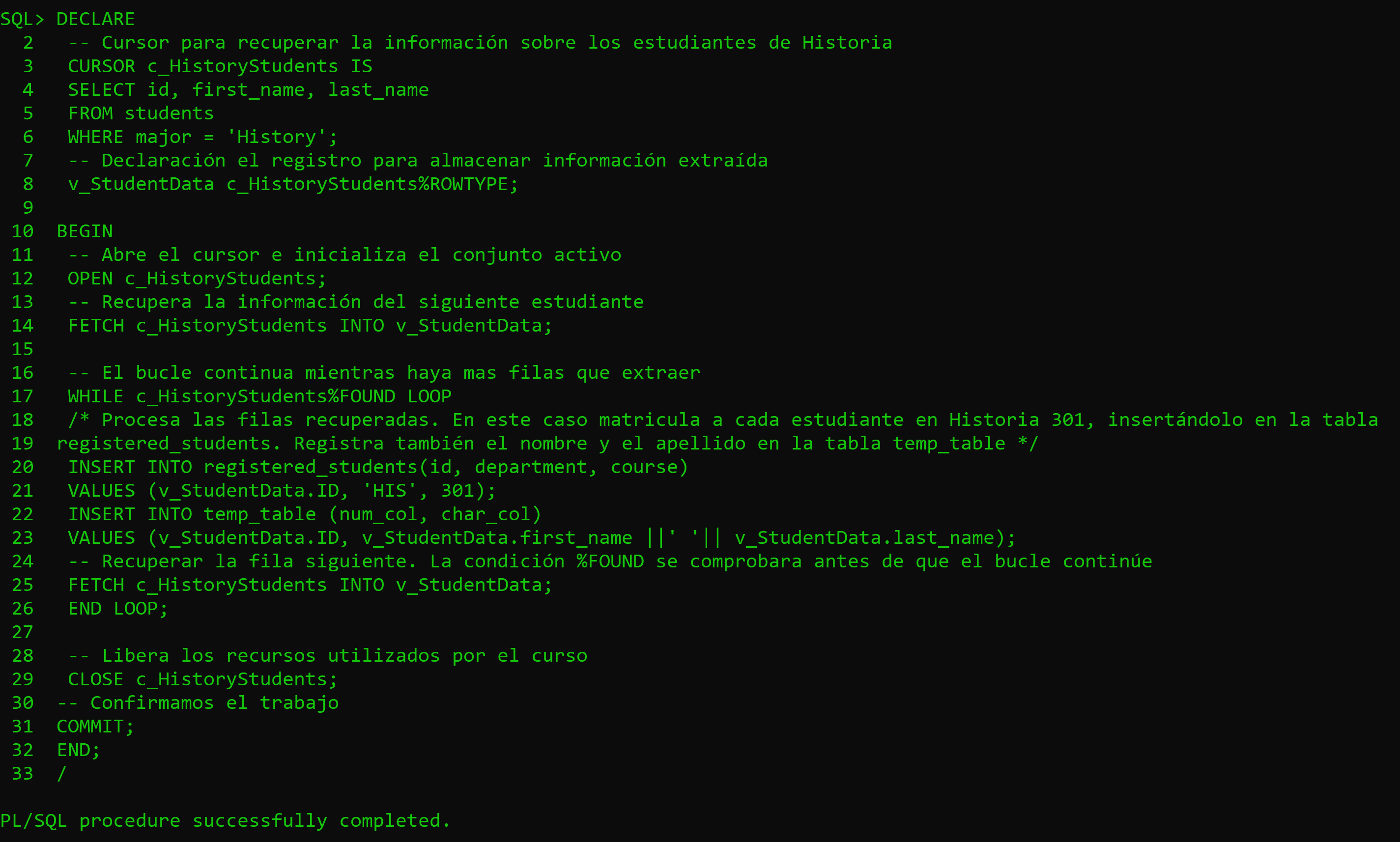
CLOSE c\_HistoryStudents;

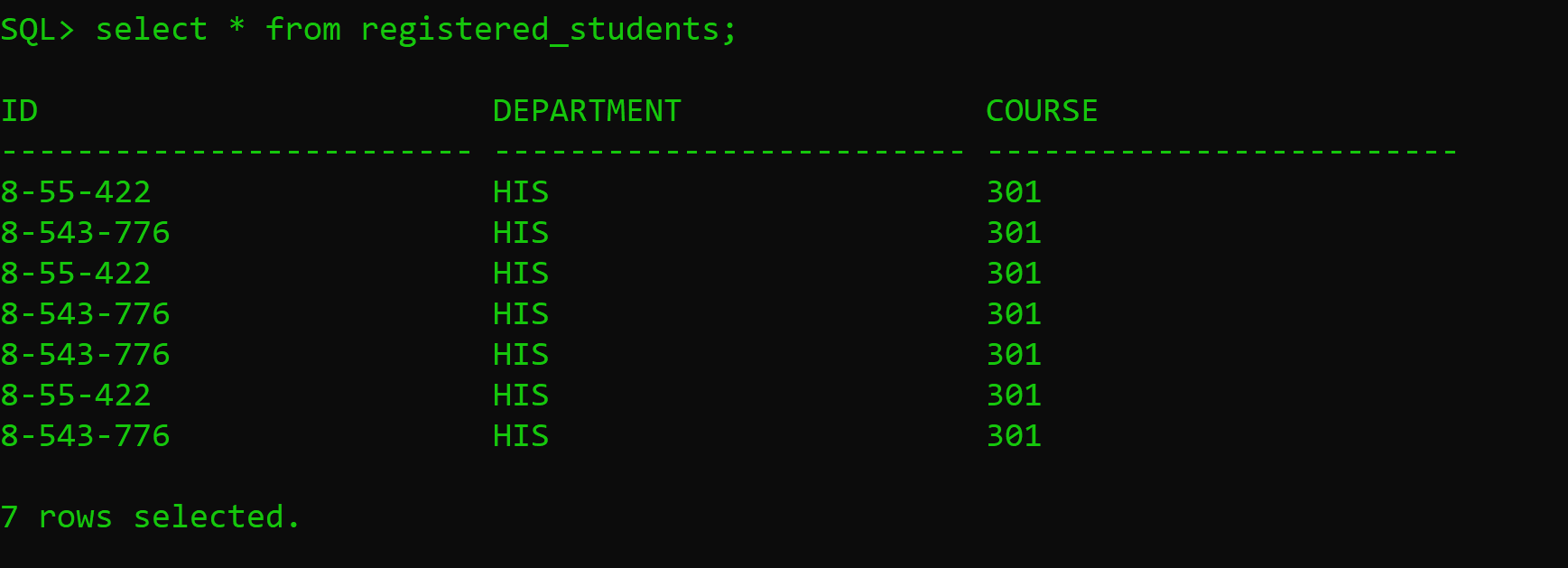
-- Confirmamos el trabajo

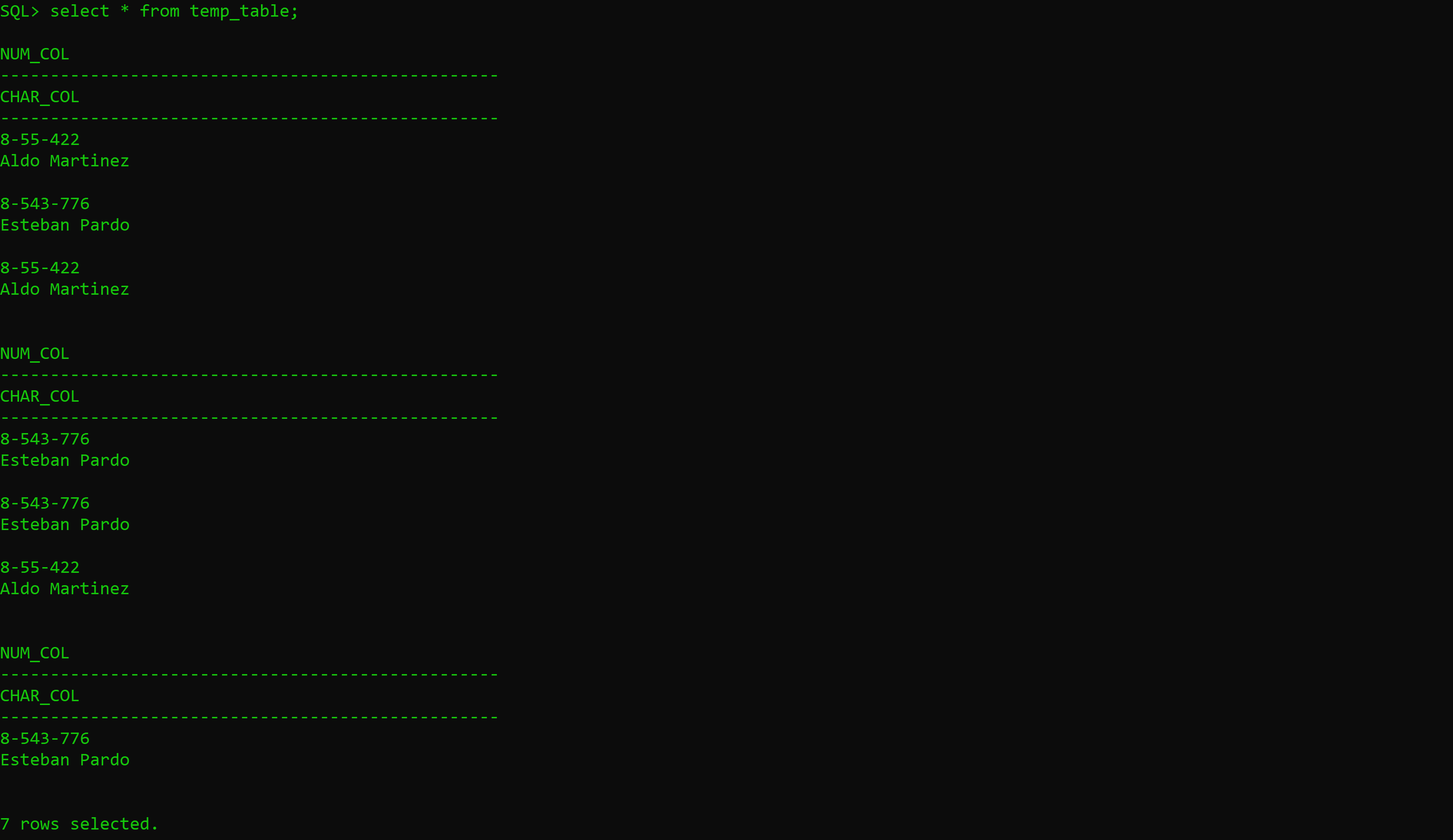
COMMIT;

END;

**Bloque PL/SQL en ejecución:**







**No. 22 – Bucle For**

DECLARE

-- Cursor para recuperar la información sobre los estudiantes de Historia

CURSOR c\_HistoryStudents IS

SELECT id, first\_name, last\_name

FROM students

WHERE major = 'History';

BEGIN

/\* Inicio del bucle. Aquí se ejecuta una orden OPEN

implícita sobre c\_HistoryStudents \*/

FOR v\_StudentData IN c\_HistoryStudents LOOP

-- Aquí se ejecuta una orden FETCH implícita

/\* Procesa las filas recuperadas. En este caso matricula a cada estudiante en Historia 301, insertándolo en la tabla

registered\_students. Registra también el nombre y el apellido en la tabla temp\_table \*/

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);

-- Antes de continuar con el bucle, aquí se hace una comprobacion implícita de c\_HistoryStudents %NOTFOUND.

END LOOP;

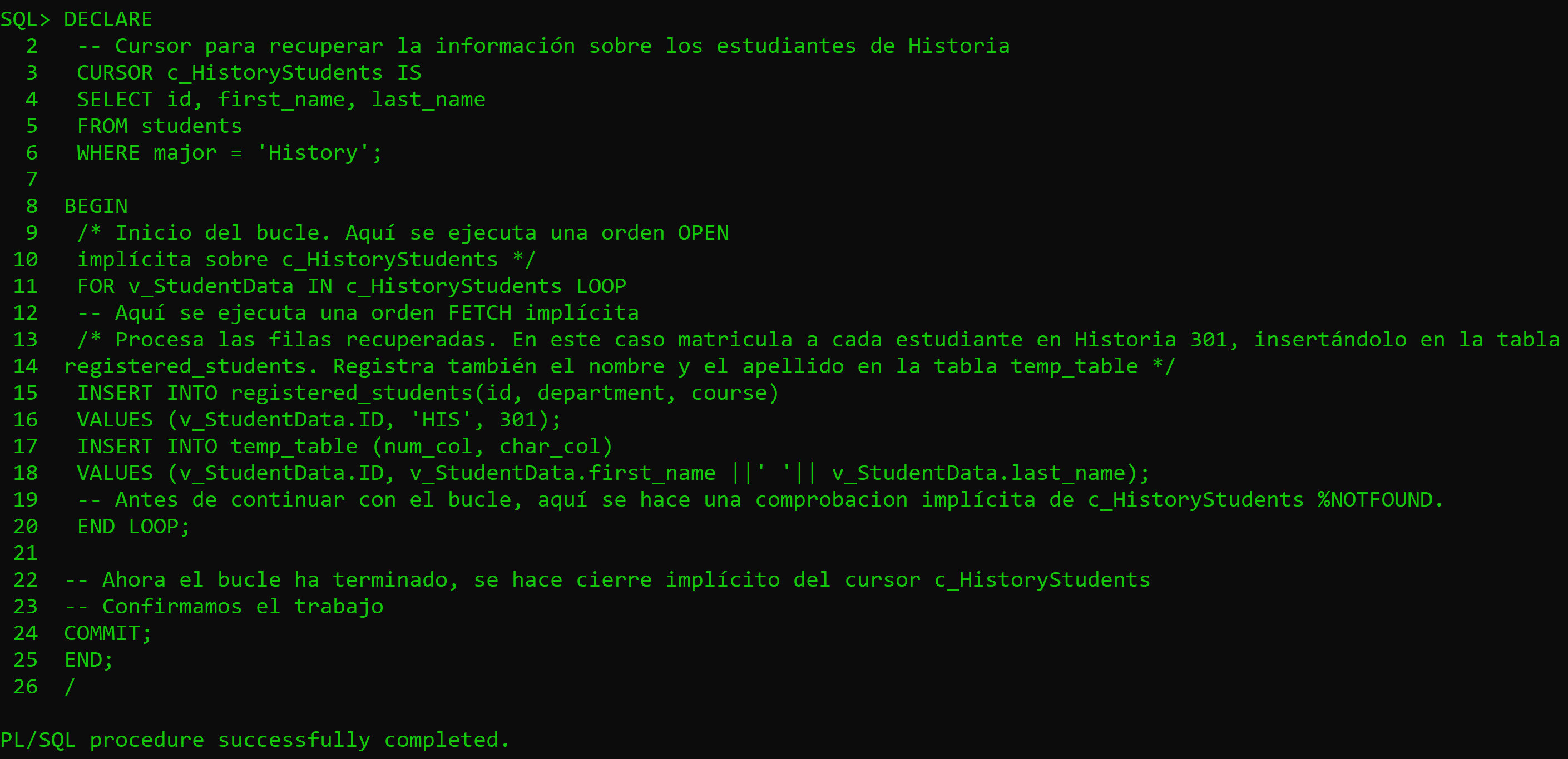
-- Ahora el bucle ha terminado, se hace cierre implícito del cursor c\_HistoryStudents

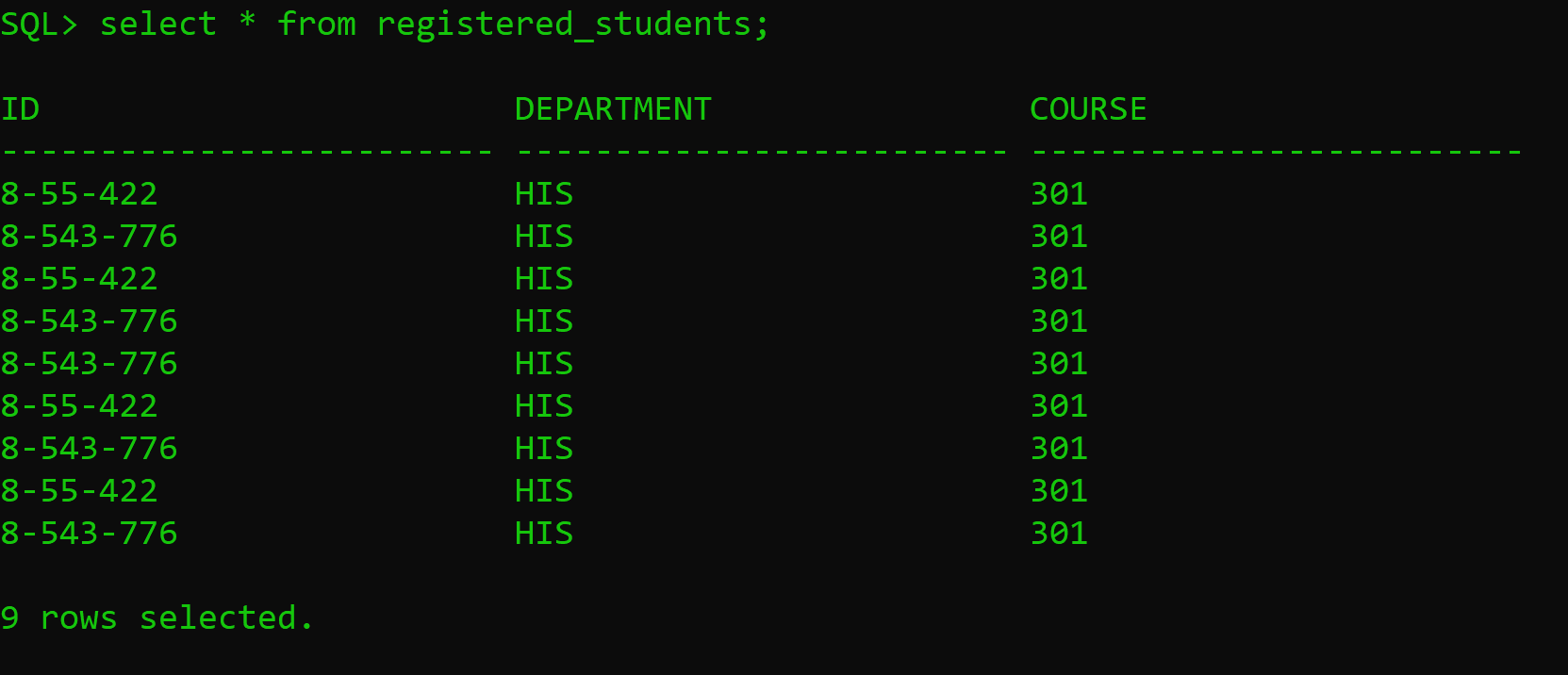
-- Confirmamos el trabajo

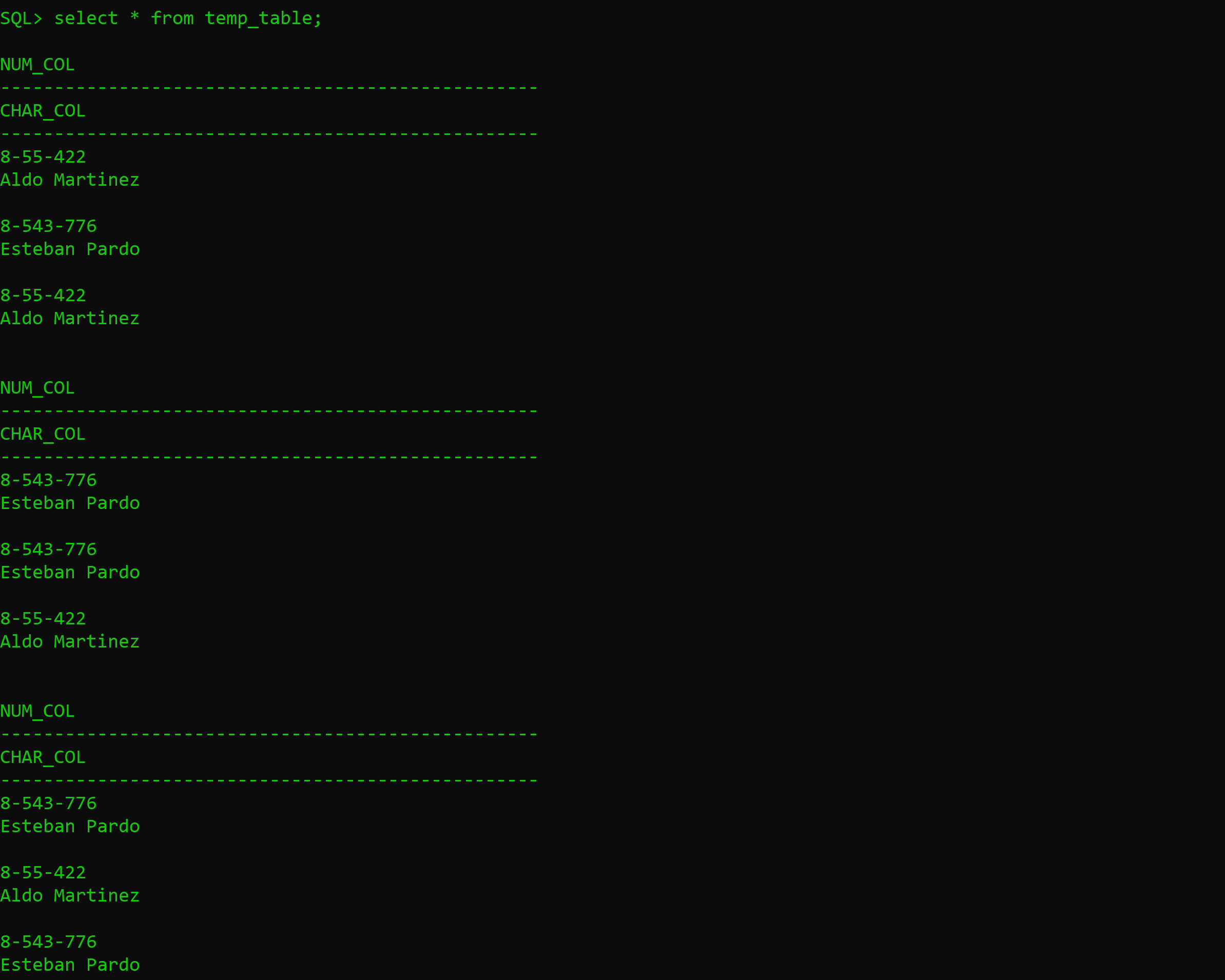
COMMIT;

END;

**Bloque PL/SQL en ejecución:**







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.**

create table Jugador(

id varchar2(10),

nombre varchar2(20),

apellido varchar2(20),

equipo varchar2(20),

provincia varchar2(20),

constraint pk\_Jugador\_id primary key(id)

);

insert into Jugador values ('8-456-332', 'Cristian', 'Quintero', 'Tauro', 'Panamá');

insert into Jugador values ('8-875-443', 'Guido', 'Rouse', 'San Francisco', 'Chorrera');

insert into Jugador values ('8-367-889', 'Saúl', 'Espinosa', 'Plaza Amador', 'Panamá');

insert into Jugador values ('4-665-211', 'Edward', 'Chacón', 'Atlético Chiriquí', 'Chiriquí');

DECLARE

/\* Variables de salida para almacenar los resultados de la Consulta \*/

v\_JugadorID jugador.id%TYPE;

v\_nombre jugador.nombre%TYPE;

v\_apellido jugador.apellido%TYPE;

v\_equipo jugador.equipo%TYPE;

v\_provincia jugador.provincia%TYPE;

/\* Declaración del Cursor \*/

CURSOR c\_JugadorPanama IS

select id, nombre, apellido, equipo

from Jugador

where provincia='Panamá';

BEGIN

/\* Identificar las filas en el conjunto activo y preparar el procesamiento ulterior de los datos \*/

OPEN c\_JugadorPanama;

/\* Recuperar cada fila del conjunto active y almacenarlos en las variables PL/SQL \*/

LOOP

FETCH c\_JugadorPanama INTO v\_JugadorID, v\_nombre, v\_apellido, v\_equipo;

/\* SI no hay más filas que recuperar, salir del bucle \*/

EXIT WHEN c\_JugadorPanama%NOTFOUND;

UPDATE Jugador

SET equipo = 'Árabe Unido'

WHERE provincia = 'Colón';

-- Si la anterior orden UPDATE no se aplica a ninguna fila, inserta una nueva fila en la tabla Jugador

IF SQL%NOTFOUND THEN

insert into Jugador (id, nombre, apellido, equipo, provincia)

values ('3-567-24', 'Azael', 'Brown', 'Árabe Unido', 'Colón');

END IF;

dbms\_output.put\_line(v\_JugadorID);

dbms\_output.put\_line(v\_nombre ||' '||v\_apellido);

dbms\_output.put\_line(v\_equipo);

END LOOP;

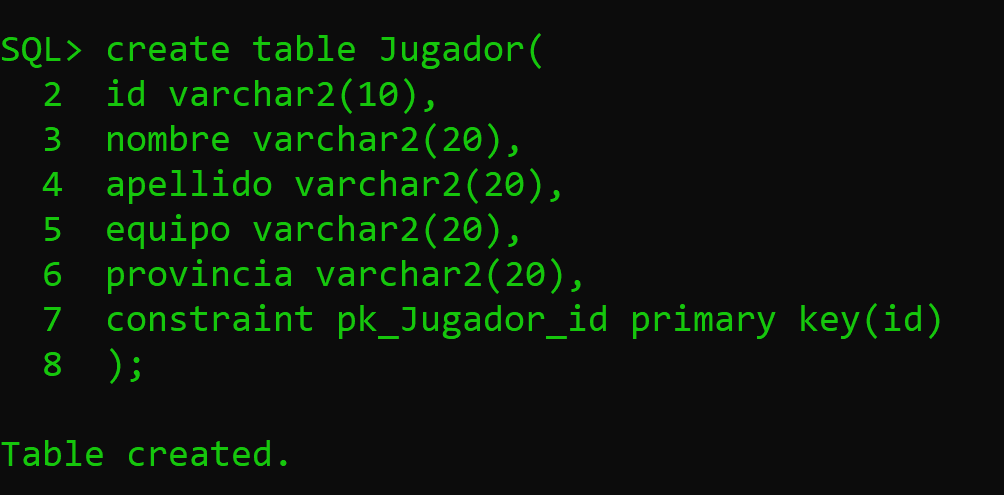
/\* Liberar los recursos usados para la consulta \*/

CLOSE c\_JugadorPanama;

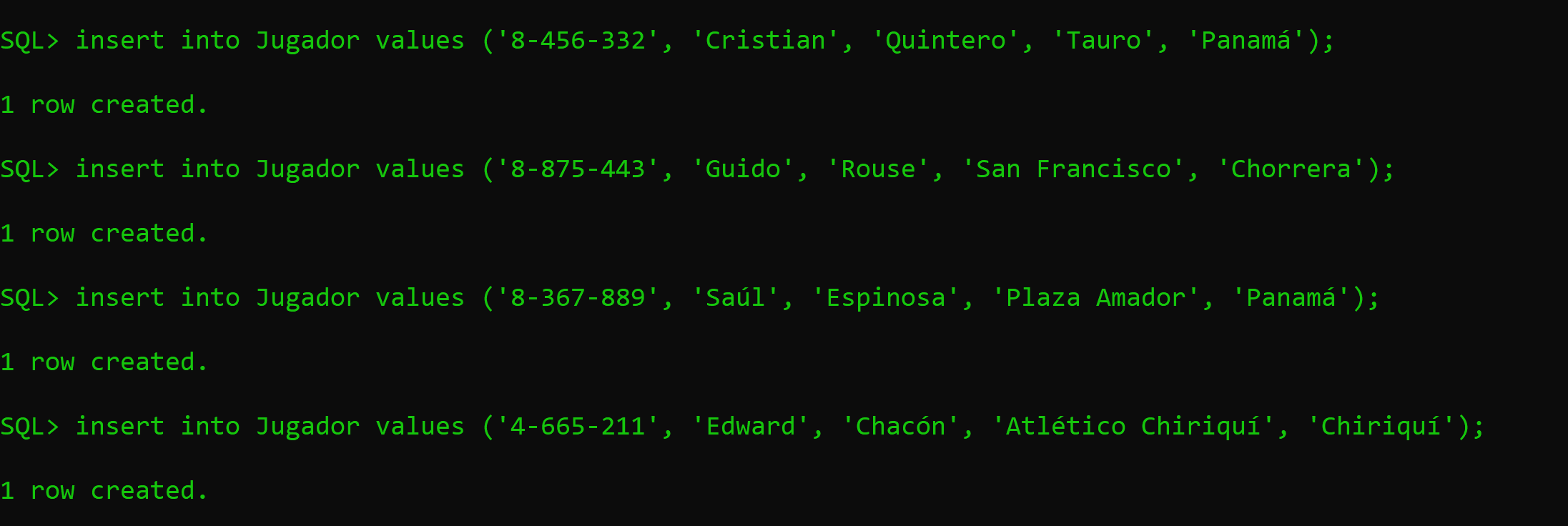
COMMIT;

END;

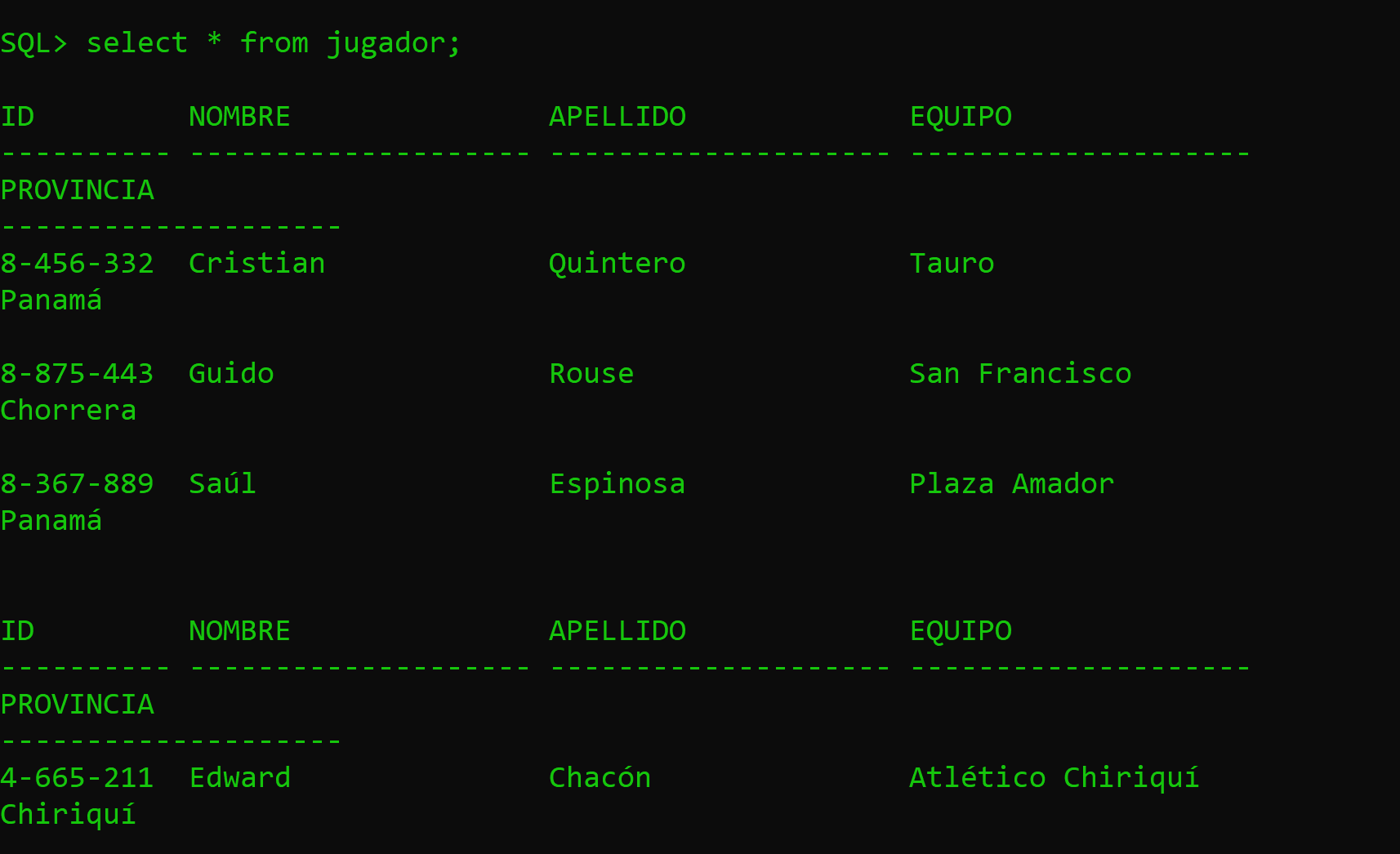
**Creación de tabla:**



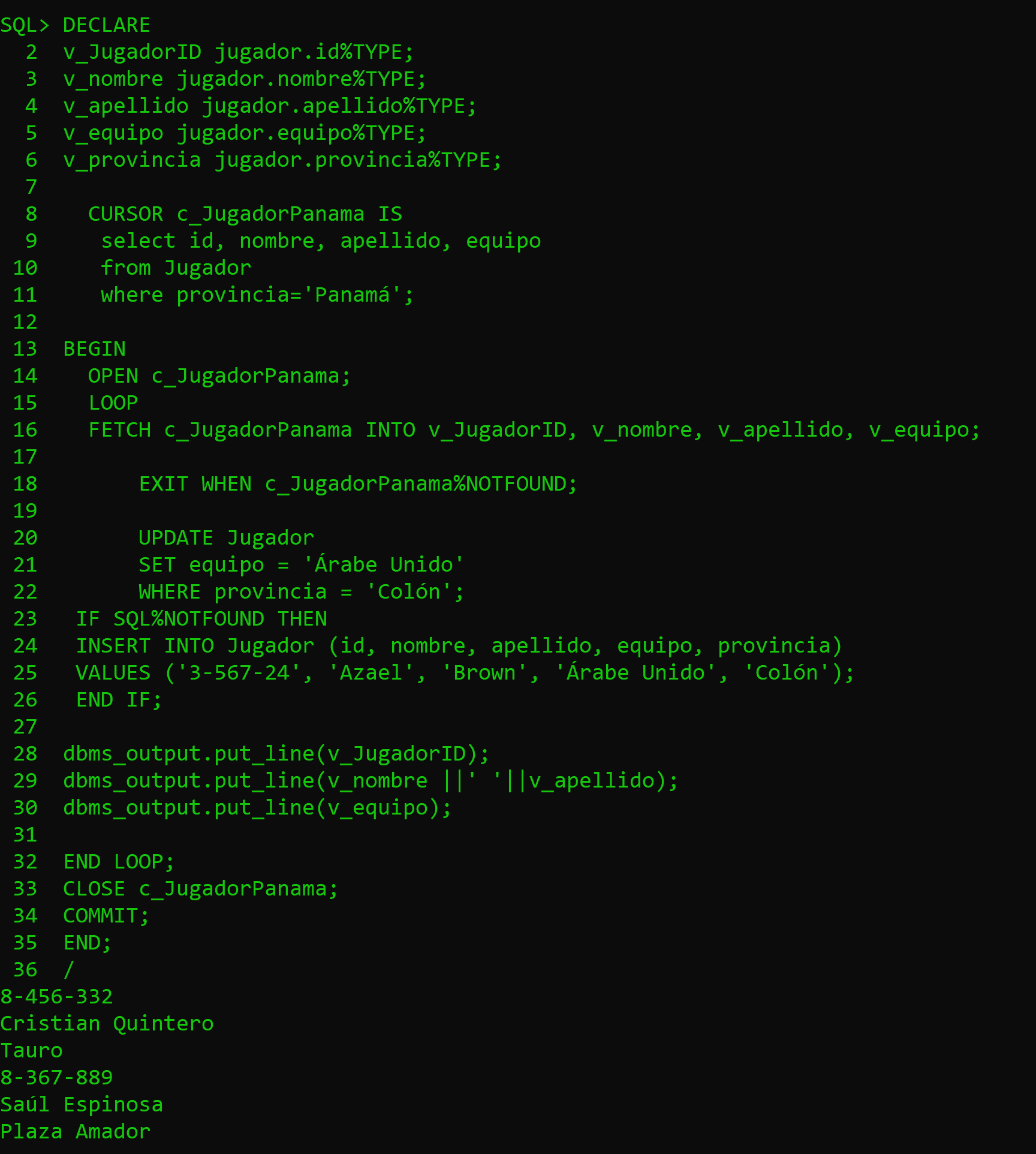
**Inserción de datos:**



**Primera tabla:**



**Bloque PL/SQL en ejecución:**



**Tabla actualizada:**

