AEA 4. Procediments

Amb l’espai de treball per a DML i DDL, crear una còpia de la taula **EMPLOYEES**

CREATE TABLE employees\_copy AS SELECT \* from employees;

Els procediments treballaran sobre **EMPLOYEES\_COPY** i es podrà comprovar el resultat comparant-lo amb la taula EMPLOYEES original.

1. Crear un procediment que incrementi el sou dels empleats en un 10%. Executar-lo des d’un bloc anònim i comprovar el resultat amb un SELECT, comparant-lo amb el salary de la taula EMPLOYEES.

El Update, igual que el delete, actualiza TODOS, salvo que se ponga un WHERE. Por eso no es necesario ni LOOP ni FOR

create or replace PROCEDURE aumentarSalario IS

BEGIN

UPDATE employees\_copy

SET salary = salary\*1.1;

END aumentarSalario;



1. Crear un procediment que modifiqui els salari dels **empleats\_copy** en funció del seu departament:

* Empleats del departament 80 🡪 salari 1000
* Empleats del departament 50 🡪 salari 2000
* Empleats de la resta de departaments 🡪 3000.

Comprovar el resultat amb un SELECT.

CREATE or REPLACE PROCEDURE modificarSalario IS

BEGIN

FOR r\_salarios IN (SELECT salary FROM employees\_copy) LOOP

UPDATE employees\_copy

SET salary = 3000;

UPDATE employees\_copy

SET salary = 1000

WHERE department\_id = 80;

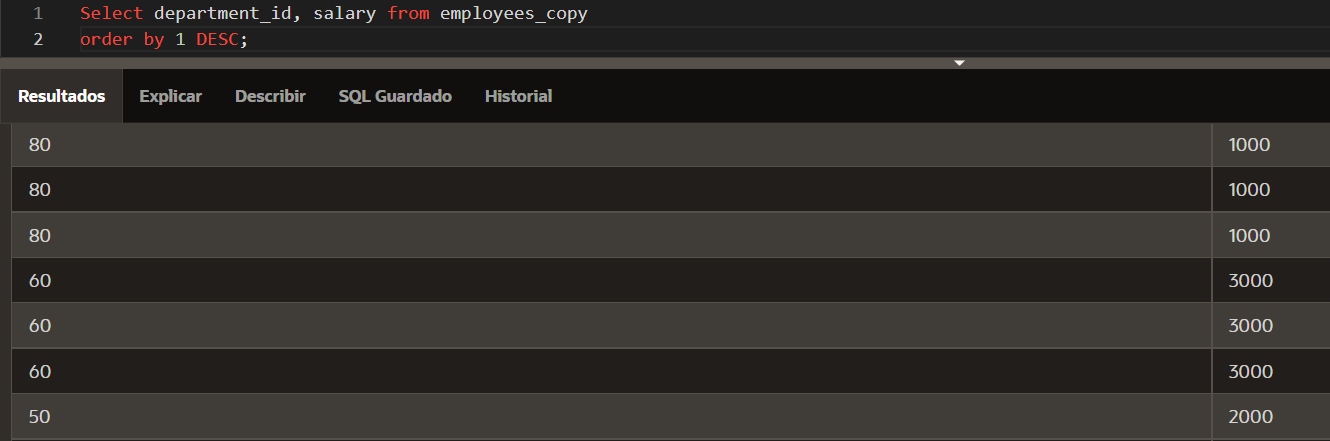
UPDATE employees\_copy

SET salary = 2000

WHERE department\_id = 50;

END LOOP;

END modificarSalario;



1. Crear el procediment **get\_location\_info**, que tingui 3 paràmetres:

* Un paràmetre d’entrada amb LOCATION\_ID
* Dos paràmetres de sortida amb la ciutat i la província corresponent al LOCATION\_ID.  
  Crea un bloc de codi que provi el procediment amb l’id 1800.

CREATE OR REPLACE PROCEDURE get\_location\_info(p\_id NUMBER, p\_city OUT VARCHAR2, p\_province OUT VARCHAR2) IS

BEGIN

SELECT city

INTO p\_city

FROM locations

WHERE location\_id = p\_id;

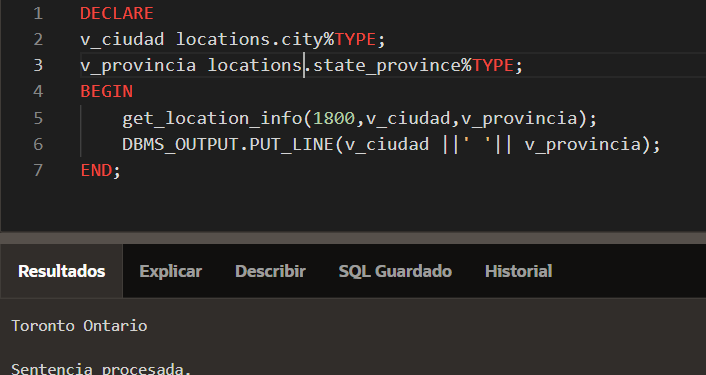
SELECT state\_province

INTO p\_province

FROM locations

WHERE location\_id = p\_id;

END get\_location\_info;



Provar el procediment amb el codi 1000. Que passa? Per què ? Com es pot evitar l’error.

No hay datos, no existe el location\_id 1000.

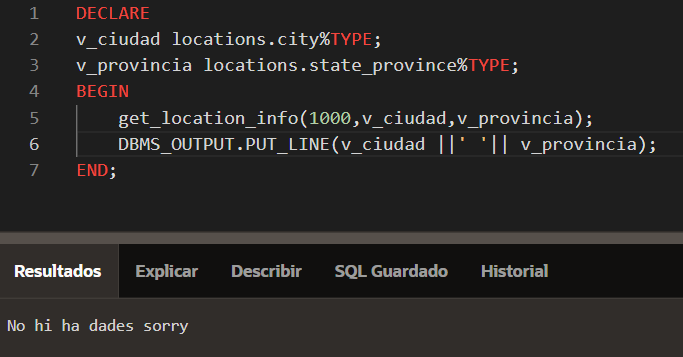
Solución:

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No hi ha dades sorry');

Al procedimiento



1. Fer un procediment **find\_names** que, donat l’id d’un país, retorni, amb dos paràmetres de sortida, el nom del país i el nom de la regió a la que pertany. El procediment ha de tractar l’excepció NO\_DATA\_FOUND per si l’id del país que s’envia no existeix.

Procediment:

CREATE OR REPLACE PROCEDURE find\_names(p\_id countries.country\_id%TYPE, p\_country\_name OUT countries.country%TYPE, p\_region\_name OUT regions.region\_name%TYPE) IS

BEGIN

SELECT country\_name, region\_name

INTO p\_country\_name, p\_region\_name

FROM countries JOIN regions USING(region\_id)

WHERE country\_id = p\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

p\_country\_name := ‘DESCONEGUT’;

p\_region\_name := ‘DESCONEGUT’;

END find\_names;

Fer un bloc de codi per provar el procediment amb els ids ‘CA’ i ‘PT’.

DECLARE

v\_region\_name regions.region\_name%TYPE;

v\_country\_name countries.country\_name%TYPE;

BEGIN

find\_names('CA',v\_region\_name,v\_country\_name);

END;



DECLARE

v\_country\_id countries.country\_id%type;

v\_region\_name regions.region\_name%TYPE;

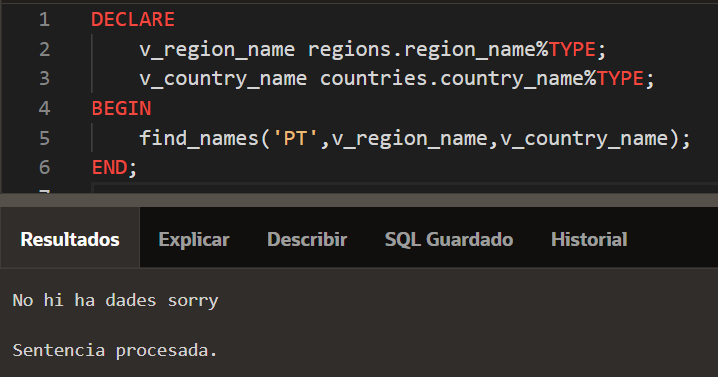
v\_country\_name countries.country\_name%TYPE;

BEGIN

v\_country\_id := :idPais

find\_names(v\_country\_id, v\_region\_name, v\_country\_name);

END;



1. Crear en procediment ***quadrat***, amb un paràmetre enter d’entrada-sortida. S’envia un número al procediment i retorna el quadrat del número.

CREATE or REPLACE PROCEDURE quadrat(p\_num IN OUT NUMBER) IS

BEGIN

p\_num := (p\_num) \* (p\_num);

END quadrat;

Fer un bloc de codi per provar el procediment amb els valor: 4, 7 i -20.

DECLARE

v\_num NUMBER;

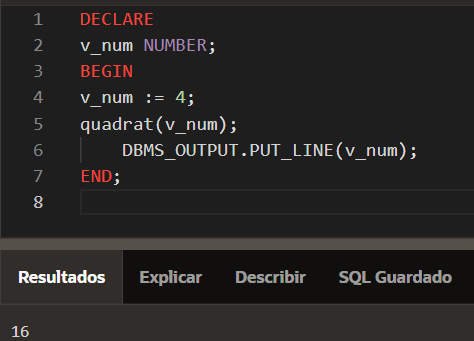
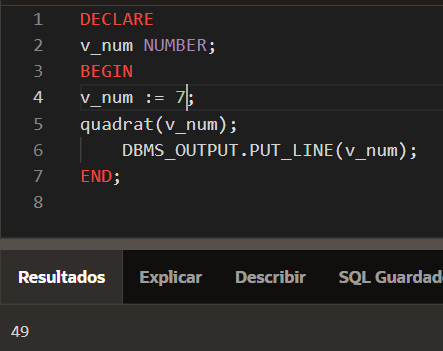
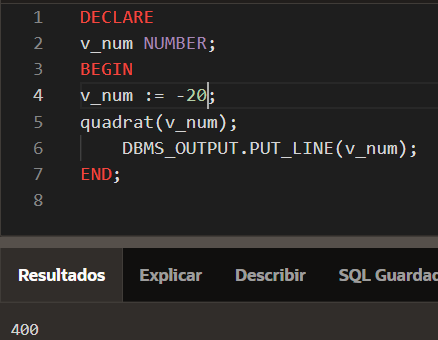
BEGIN

v\_num := 4;

quadrat(v\_num);

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

END;

1. Modificar l’exercici 5 convertint el procediment en una funció que retorna el quadrat del número que rep com a paràmetre.

Fer un bloc de codi per provar el procediment amb els valor: 4, 7 i -20.

1. Crear una funció **existeix\_dept**, que rebi com a paràmetre l’id d’un departament i retorni si el departament és a la taula de departaments o no.

CREATE OR REPLACE FUNCTION existeix\_dept(p\_id NUMBER)

RETURN BOOLEAN IS

v\_dep\_id departments.department\_id%TYPE;

BEGIN

SELECT department\_id

INTO v\_dep\_id

FROM departments

WHERE department\_id = p\_id;

RETURN true;

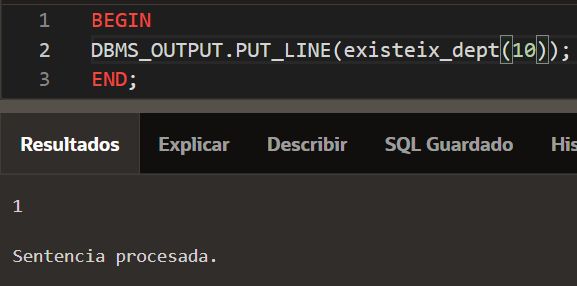
EXCEPTION

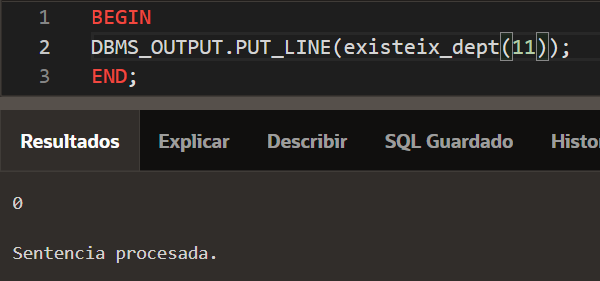
WHEN NO\_DATA\_FOUND THEN

RETURN false;

END existeix\_dept;

Fer un bloc de prova amb els ids de departament 10 i 11.





1. Fer la funció **num\_departaments\_location**, que donat un LOCATION\_ID, retorni el nombre de departaments que tenen aquest LOCATION\_ID. Si el location\_id no té cap departament, que retorni 0.

Fer un bloc de prova amb els LOCATION\_ID 1700, 1800 i 1900.

CREATE OR REPLACE FUNCTION num\_departaments\_location(p\_id NUMBER)

RETURN NUMBER IS

v\_count NUMBER;

BEGIN

select count(department\_id)

INTO v\_count

FROM departments

where location\_id = p\_id;

return v\_count;

EXCEPTION WHEN no\_data\_found THEN

return 0;

END num\_departaments\_location;

