AEA 4. Procediments i funcions

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 salari de la taula EMPLOYEES.

**PROCEDIMENT**

CREATE OR REPLACE PROCEDURE inc\_salary IS  
 v\_increment\_salari NUMBER(3,2) := 0.1;

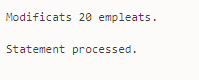
BEGIN  
UPDATE EMPLOYEES\_COPY  
SET SALARY = SALARY + SALARY\*v\_increment\_salari;   
DBMS\_OUTPUT.PUT\_LINE('Modificats ' || SQL%ROWCOUNT || ' empleats. ');

END;

**BLOC DE PROVA**

BEGIN  
 inc\_salary;

END;

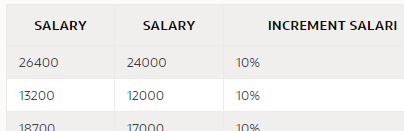


**RESULTAT**

SELECT EC.SALARY, E.SALARY, ((EC.SALARY-E.SALARY)\*100)/E.SALARY || '%'

AS "INCREMENT SALARI"

FROM EMPLOYEES e JOIN EMPLOYEES\_COPY ec USING (EMPLOYEE\_ID)



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.

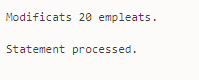
**PROCEDIMENT**

CREATE OR REPLACE PROCEDURE inc\_salary\_dept IS  
BEGIN  
 UPDATE EMPLOYEES\_COPY  
 SET SALARY = CASE DEPARTMENT\_ID WHEN 80 THEN 1000  
 WHEN 50 THEN 2000  
 ELSE 3000  
 END;  
 DBMS\_OUTPUT.PUT\_LINE('Modificats ' || SQL%ROWCOUNT || ' empleats. ');  
END;

**BLOC DE PROVA**

BEGIN  
 inc\_salary\_dept;

END;



**RESULTAT**

SELECT DEPARTMENT\_ID, SALARY

FROM EMPLOYEES\_COPY

ORDER BY DEPARTMENT\_ID

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.

**PROCEDIMENT**

CREATE OR REPLACE PROCEDURE get\_location\_info   
 (p\_location\_id IN LOCATIONS.LOCATION\_ID%TYPE,  
 p\_city OUT LOCATIONS.CITY%TYPE,  
 p\_province OUT LOCATIONS.STATE\_PROVINCE%TYPE)

IS  
BEGIN  
 -- Busca la ciutat i la provincia  
 SELECT CITY, STATE\_PROVINCE  
 INTO p\_city, p\_province  
 FROM LOCATIONS  
 WHERE LOCATION\_ID = p\_location\_id;

END;

**BLOC DE PROVA**

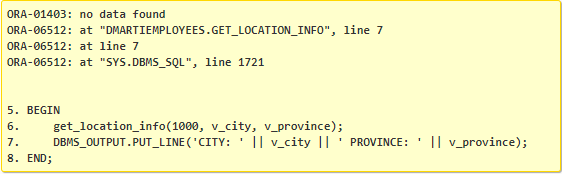
DECLARE  
 -- Variables locals per guardar el resultat  
 v\_city LOCATIONS.CITY%TYPE;  
 v\_province LOCATIONS.STATE\_PROVINCE%TYPE;  
BEGIN  
 get\_location\_info(1800, v\_city, v\_province);  
 DBMS\_OUTPUT.PUT\_LINE('CITY: ' || v\_city || ' PROVINCE: ' || v\_province);  
END;

**RESULTAT**



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

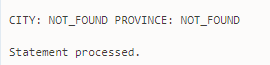
**Com no hi ha cap location amb aquest id, genera una excepció que no es captura.**



Es pot capturar l’excepció **NO\_DATA\_FOUND** al procediment i que retorni com a valor de ciutat i província NOT\_FOUND.

CREATE OR REPLACE PROCEDURE get\_location\_info   
 (p\_location\_id IN LOCATIONS.LOCATION\_ID%TYPE,  
 p\_city OUT LOCATIONS.CITY%TYPE,  
 p\_province OUT LOCATIONS.STATE\_PROVINCE%TYPE)  
IS  
BEGIN  
 -- Busca la ciutat i la provincia  
 SELECT CITY, STATE\_PROVINCE  
 INTO p\_city, p\_province  
 FROM LOCATIONS  
 WHERE LOCATION\_ID = p\_location\_id;  
EXCEPTION  
 -- Gestor d'excepcions NO\_DATA\_FOUND  
 -- Dona valor NOT FOUND als paràmetres de sortida  
 WHEN NO\_DATA\_FOUND THEN   
 p\_city := 'NOT\_FOUND';  
 p\_province := 'NOT\_FOUND';  
END;

Tornem a executar la prova amb id=1000:



**NO EXCEPCIÓ: 6**

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.

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

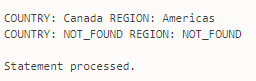
**PROCEDIMENT**

CREATE OR REPLACE PROCEDURE find\_names  
 (p\_country\_id IN COUNTRIES.COUNTRY\_ID%TYPE,  
 p\_country\_name OUT COUNTRIES.COUNTRY\_NAME%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\_country\_id;  
EXCEPTION  
 -- Tracta excepció NO\_DATA\_FOUND  
 WHEN NO\_DATA\_FOUND THEN   
 p\_country\_name := 'NOT\_FOUND';  
 p\_region\_name := 'NOT\_FOUND';  
END;

**BLOC DE PROVA**

DECLARE  
 -- Variables locals per guardar el resultat  
 v\_country\_name COUNTRIES.COUNTRY\_NAME%TYPE;  
 v\_region\_name REGIONS.REGION\_NAME%TYPE;  
BEGIN  
 -- Prova amb Canadà  
 find\_names('CA', v\_country\_name, v\_region\_name);  
 DBMS\_OUTPUT.PUT\_LINE('COUNTRY: ' || v\_country\_name ||   
 ' REGION: ' || v\_region\_name);  
 -- Prova amb un codi que no existeix  
 find\_names('PT', v\_country\_name, v\_region\_name);  
 DBMS\_OUTPUT.PUT\_LINE('COUNTRY: ' || v\_country\_name ||   
 ' REGION: ' || v\_region\_name);  
END;

**RESULTAT**



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

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

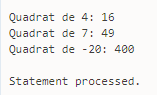
**PROCEDIMENT**

CREATE OR REPLACE PROCEDURE quadrat (p\_numero IN OUT NUMBER) IS  
BEGIN  
 p\_numero := p\_numero \* p\_numero;   
END;

BLOC DE PROVA

DECLARE  
 -- Variable local per guardar el resultat  
 v\_numero NUMBER(5);  
BEGIN  
 -- Prova amb 4  
 v\_numero := 4;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de 4: ' || v\_numero);  
 -- Prova amb 7  
 v\_numero := 7;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de 7: ' || v\_numero);   
 -- Prova amb -20  
 v\_numero := -20;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de -20: ' || v\_numero);   
END;

**RESULTAT**



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.

**FUNCIÓ**

CREATE OR REPLACE FUNCTION quadrat (p\_numero IN NUMBER) RETURN NUMBER IS  
BEGIN  
 RETURN p\_numero \* p\_numero;

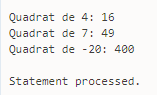
END;

**BLOC DE PROVA**

DECLARE  
 -- Variable local per guardar el resultat  
 v\_numero NUMBER(5);  
BEGIN  
 -- Prova amb 4  
 v\_numero := 4;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de 4: ' || v\_numero);  
 -- Prova amb 7  
 v\_numero := 7;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de 7: ' || v\_numero);   
 -- Prova amb -20  
 v\_numero := -20;  
 quadrat(v\_numero);  
 DBMS\_OUTPUT.PUT\_LINE('Quadrat de -20: ' || v\_numero);

END;

**RESULTAT**



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.

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

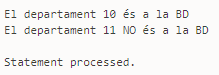
**FUNCIÓ**

CREATE OR REPLACE FUNCTION existeix\_dept   
(p\_depart\_id IN DEPARTMENTS.DEPARTMENT\_ID%TYPE) RETURN BOOLEAN IS  
v\_depart\_id DEPARTMENTS.DEPARTMENT\_ID%TYPE;  
BEGIN  
 -- Busca el departament  
 SELECT DEPARTMENT\_ID  
 INTO v\_depart\_id  
 FROM DEPARTMENTS   
 WHERE DEPARTMENT\_ID = p\_depart\_id;  
  
 -- Si troba el departament retorna TRUE.  
 -- Si el departament no hi és, genera una excepció  
 RETURN TRUE;  
EXCEPTION

-- Si no troba dades retornem FALSE  
 WHEN NO\_DATA\_FOUND THEN   
 RETURN FALSE;

END;

**NOTA**: Encara que no ens interessi el valor que retorna el SELECT, si no li posem INTO genera un error de compilació.



**BLOC DE PROVA**

DECLARE  
 -- Variable local per guardar l'id de departament  
 v\_depart\_id DEPARTMENTS.DEPARTMENT\_ID%TYPE;  
BEGIN  
 -- Prova amb 10  
 v\_depart\_id := 10;   
 IF existeix\_dept(v\_depart\_id) THEN   
 DBMS\_OUTPUT.PUT\_LINE('El departament ' || v\_depart\_id || ' és a la BD');  
 ELSE  
 DBMS\_OUTPUT.PUT\_LINE('El departament ' || v\_depart\_id || ' NO és a la BD');  
 END IF;  
  
 -- Prova amb 11  
 v\_depart\_id := 11;   
 IF existeix\_dept(v\_depart\_id) THEN   
 DBMS\_OUTPUT.PUT\_LINE('El departament ' || v\_depart\_id || ' és a la BD');  
 ELSE  
 DBMS\_OUTPUT.PUT\_LINE('El departament ' || v\_depart\_id || ' NO és a la BD');  
 END IF;  
END;

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\_location\_id IN DEPARTMENTS.LOCATION\_ID%TYPE) RETURN NUMBER IS

v\_num\_departments NUMBER DEFAULT 0;

BEGIN

-- Busca el nombre de departament de la location

SELECT COUNT(DEPARTMENT\_ID)

INTO v\_num\_departments

FROM DEPARTMENTS

WHERE LOCATION\_ID = p\_location\_id;

-- Retorna el nombre de departaments del LOCATION\_ID.

RETURN v\_num\_departments;

EXCEPTION

-- Si no hi ha cap departament en auqesta location, retorna 0

WHEN NO\_DATA\_FOUND THEN

RETURN 0;

END;





