A black background with a black square

AI-generated content may be incorrect.

**UNIVERSITETI POLITEKNIK I TIRANËS**

**FAKULTETI I TEKNOLOGJISË DHE INFORMACIONIT**

**DEPARTAMENTI I INXHINIERISË INFORMATIKE**

**Punë Laboratori nr. 5**

**Lënda:** Bazat e tё dhёnave

**Grupi:** III-B

**Punoi:** Piro Gjikdhima **Pranoi**: MSc.Paola Shasivari

*1.Ndertoni funksionin qe gjen numrin total te punonjesve pasi merr titullin e punes.*

CREATE OR REPLACE FUNCTION *tot\_emp*(job\_title IN VARCHAR2) RETURN INT  
 IS  
 total\_emp INT;  
BEGIN  
 SELECT *COUNT*(*\**)  
 INTO total\_emp  
 FROM EMPLOYEES  
 WHERE JOB\_ID = job\_title;  
 RETURN total\_emp;  
END;  
BEGIN  
 DECLARE  
 result INT;  
 BEGIN  
 result := *tot\_emp*('AD\_VP');  
 DBMS\_OUTPUT.*PUT\_LINE*('Total employees: ' || result);  
 END;  
END;  
/

*2.Ndertoni nje funksion qe kthen n punonjesit me rrogen me te larte.*

CREATE OR REPLACE FUNCTION *top\_n\_employees*(n IN NUMBER)  
 RETURN SYS\_REFCURSOR  
 IS  
 emp\_cursor SYS\_REFCURSOR;  
BEGIN  
 OPEN emp\_cursor FOR  
 SELECT FIRST\_NAME, LAST\_NAME, SALARY  
 FROM EMPLOYEES  
 ORDER BY SALARY DESC  
 FETCH FIRST n ROWS ONLY;  
 RETURN emp\_cursor;  
END;

/

*3.Ndertoni nje funksion qe kthen punonjesin me rrogen me te larte ne baze te departamentit.*

CREATE OR REPLACE FUNCTION *top\_emp\_per\_dep*(dep\_ID IN INT) RETURN SYS\_REFCURSOR  
 IS  
 emp\_cursor SYS\_REFCURSOR;  
BEGIN  
 OPEN emp\_cursor FOR  
 SELECT FIRST\_NAME, LAST\_NAME, SALARY  
 FROM EMPLOYEES  
 WHERE DEPARTMENT\_ID = dep\_ID  
 FETCH FIRST 1 ROWS ONLY;  
 RETURN emp\_cursor;  
END;

*4.Ndertoni nje funksion qe kthen punonjesit qe kane te njejtin menaxher.*

CREATE OR REPLACE FUNCTION *same\_manager*(menag\_id IN INT) RETURN SYS\_REFCURSOR  
 IS  
 emp\_cursor SYS\_REFCURSOR;  
BEGIN  
 OPEN emp\_cursor FOR  
 SELECT FIRST\_NAME, LAST\_NAME, SALARY  
 FROM EMPLOYEES  
 WHERE MANAGER\_ID = menag\_id;  
 RETURN emp\_cursor;  
END;

*5.Ndertoni nje triger qe hedh exception nese shtohen produket me emra te dublikuara ne tabelen products.*

CREATE OR REPLACE TRIGGER prevent\_duplicate\_products  
 BEFORE INSERT OR UPDATE  
 ON PRODUCTS  
 FOR EACH ROW  
DECLARE  
 duplicate\_count INT;  
BEGIN  
 SELECT *COUNT*(*\**)  
 INTO duplicate\_count  
 FROM PRODUCTS  
 WHERE *LOWER*(PRODUCT\_NAME) = *LOWER*(:NEW.PRODUCT\_NAME);  
  
 IF duplicate\_count > 0 THEN  
 *RAISE\_APPLICATION\_ERROR*(-20001, 'Duplicate product name is not allowed.');  
 END IF;  
END;  
/

*6.Afishoni per punonjesit qe e kane salary me te madh se 3200 department\_name, d.department\_id, first\_name, last\_name, job\_id, salary.(duke perdorur kursorin)*

DECLARE  
 CURSOR cur\_for\_emp IS  
 SELECT D.DEPARTMENT\_NAME, D.DEPARTMENT\_ID, E.FIRST\_NAME, E.LAST\_NAME, E.JOB\_ID, E.SALARY  
 FROM EMPLOYEES E  
 JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID  
 WHERE E.SALARY > 3200;  
 emp\_record cur\_for\_emp%ROWTYPE;  
BEGIN  
 OPEN cur\_for\_emp;  
 LOOP  
 FETCH cur\_for\_emp INTO emp\_record;  
 EXIT WHEN cur\_for\_emp%NOTFOUND;  
 DBMS\_OUTPUT.*PUT\_LINE*('Department: ' || emp\_record.department\_name ||  
 ', Department ID: ' || emp\_record.department\_id ||  
 ', First Name: ' || emp\_record.first\_name ||  
 ', Last Name: ' || emp\_record.last\_name ||  
 ', Job ID: ' || emp\_record.job\_id ||  
 ', Salary: ' || emp\_record.salary);  
 END LOOP;  
 CLOSE cur\_for\_emp;  
END;  
/