

## Solutii Tutoriat 5

- a) Să se scrie o procedură care primește numele unui angajat ca parametru și furnizează numele celor mai bine-plătiți 2 colegi din departamentul său

```
--declararea procedurii
CREATE OR REPLACE PROCEDURE p_salaries
  (v_name IN EMPLOYEES.last_name%TYPE, v_col1 OUT
  EMPLOYEES.last_name%TYPE, v_col2 OUT EMPLOYEES.last_name%TYPE) IS
--sectiunea de declarare a variabilelor din cadrul procedurii
  CURSOR c_salaries IS  SELECT last_name
  FROM (
    SELECT last_name, salary
    FROM employees
    WHERE department_id IN (
      SELECT department_id FROM employees WHERE last_name
= v_name
    )
    ORDER BY salary DESC
  )
  WHERE ROWNUM <= 2;
BEGIN
  v_col1 := 'Default';
  OPEN c_salaries;
  LOOP
    IF v_col1 = 'Default' THEN
      FETCH c_salaries INTO v_col1;
    ELSE
      FETCH c_salaries INTO v_col2;
    END IF;
    EXIT WHEN c_salaries%NOTFOUND;
  END LOOP;
  CLOSE c_salaries;
END p_salaries;

--blocul de apelare a procedurii
DECLARE
  v_1 EMPLOYEES.last_name%TYPE;
  v_2 EMPLOYEES.last_name%TYPE;
```

```

BEGIN
    p_salaries('Weiss', v_1, v_2);
    DBMS_OUTPUT.PUT_LINE(v_1);
    DBMS_OUTPUT.PUT_LINE(v_2);
END;

```

- b) Să se scrie o cerere SQL care să ne dea informații despre angajații care au ca id numere prime. Se pot folosi funcții auxiliare

```

CREATE OR REPLACE function is_prime
(v_number NUMBER) RETURN NUMBER IS
v_divizor NUMBER(10) := 2;
BEGIN

    LOOP
        IF MOD(v_number, v_divizor) = 0 THEN RETURN 0;
        END IF;
        EXIT WHEN v_divizor > v_number / 2;
        v_divizor := v_divizor + 1;
    END LOOP;
    RETURN 1;
END is_prime;

SELECT *
FROM employees
WHERE is_prime(employee_id) = 1;

```

- c) Să se creeze o procedură care furnizează numărul de angajați care lucrează într-o anumită țară al cărei nume este dat ca parametru

```

CREATE OR REPLACE PROCEDURE p_countries
(v_countryname IN COUNTRIES.country_name%TYPE, v_numberemployees OUT
NUMBER) IS

BEGIN
    SELECT count(employee_id)
    INTO v_numberemployees
    FROM employees e JOIN departments d ON (e.department_id = d.department_id) JOIN
locations l ON (l.location_id = d.location_id) JOIN countries c ON (c.country_id =
l.country_id)
    GROUP BY country_name
    HAVING country_name = v_countryname;
EXCEPTION

```

```

        WHEN NO_DATA_FOUND THEN v_numberemployees := 0;
    END;

    DECLARE
    v_number NUMBER;
    BEGIN
        p_countries('Switzerland', v_number);
        DBMS_OUTPUT.PUT_LINE(v_number);
    END;

```

- d) Să se creeze o funcție care primește codul unei regiuni și returnează numărul de țări din acea regiune în care nu lucrează niciun angajat.

```

CREATE OR REPLACE function f_countries
(v_regionid COUNTRIES.region_id%TYPE) RETURN NUMBER IS

    CURSOR c_countries(region COUNTRIES.region_id%TYPE) IS SELECT country_name
    FROM countries
    WHERE region_id = region;

    v_counter NUMBER(10) := 0;
    v_countryname COUNTRIES.country_name%TYPE;
    v_employees NUMBER(10);
    BEGIN
        OPEN c_countries(v_regionid);
        LOOP
            FETCH c_countries INTO v_countryname;
            p_countries(v_countryname, v_employees);
            IF v_employees = 0 THEN v_counter := v_counter + 1;
            END IF;

            EXIT WHEN c_countries%NOTFOUND;
        END LOOP;
        CLOSE c_countries;
        RETURN v_counter;
    END;

    BEGIN
        DBMS_OUTPUT.PUT_LINE(f_countries(1));
    END;

```

