1. feladat: Készítsen tárolt eljárást *reszleglista* néven, amely egy ország kódját (pl. *US*) kapja bemenő paraméterként, majd válaszul kilistázza a felhasználói képernyőre (*Dbms Output*) az országban található részlegek nevét, városát és irányítószámát, a részleg neve szerinti sorrendben! A lista formázottan jelenjen meg és legyen fejléce!
Tesztelje a tárolt eljárást az *exec reszleglista('US');* utasítás lefuttatásával!

```
CREATE OR REPLACE PROCEDURE reszleglista (orszag_az IN varchar2) AS
  CURSOR dep_cur IS
    SELECT department_name, city, postal_code
    FROM departments INNER JOIN locations USING(location_id)
    WHERE country_id = orszag_az
    ORDER BY 1;
  dep_rec dep_cur%ROWTYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE(RPAD('Részleg', 25) || RPAD('Város', 30) ||
                       RPAD('Irányítószám', 15));
  DBMS_OUTPUT.PUT_LINE(RPAD('-', 70, '-'));
  FOR dep_rec IN dep_cur LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(dep_rec.department_name, 25) ||
                         RPAD(dep_rec.city, 30) ||
                         RPAD(dep_rec.postal_code, 15));
  END LOOP;
END reszleglista;
/

EXEC reszleglista('US');
```

2. feladat: a) Készítsen egy *dolgozo2* nevű táblát, amely minden adatot megkap az *employees* táblától!

b) Készítsen tárolt eljárást *fizetesemeles* néven, amely egy fizetéshatárt és egy összeget kap bemenő paraméterként. Az eljárás mindazoknak emelje meg a fizetését a második paraméterben kapott összeggel, akik
- a fizetéshatár alatt dolgoznak, és
- nem kapnak jutalékot!

Az eljárás listázza ki a felhasználói képernyőre (*Dbms Output*) az fizetésemelést kapott dolgozók vezetéknevét és új fizetését!
Tesztelje a tárolt eljárást az *exec fizetesemeles(5000, 30);* utasítás lefuttatásával!

c) A *dolgozo2* tábla megnyitásával ellenőrizze, hogy valóban megtörtént-e az adatok frissítése! Ezt követően adja ki a *ROLLBACK;* parancsot, majd győződjön meg róla, hogy a fizetések visszaálltak a tárolt eljárás lefuttatása előtti értékre!
Törölje a *dolgozo2* táblát!

```
CREATE TABLE dolgozo2
  AS SELECT * FROM employees;

CREATE OR REPLACE PROCEDURE fizetesemeles(hatar NUMBER, osszeg NUMBER) AS
  CURSOR dolg_cur IS
    SELECT last_name, salary
    FROM dolgozo2
    WHERE salary < hatar AND commission_pct IS NULL
    FOR UPDATE;
  vnev dolgozo2.last_name%TYPE;
  fiz  dolgozo2.salary%TYPE;
BEGIN
  OPEN dolg_cur;
```

```
  LOOP
    FETCH dolg_cur INTO vnev, fiz;
    EXIT WHEN dolg_cur%NOTFOUND;
    fiz := fiz + osszeg;
    UPDATE dolgozo2
      SET salary = fiz
      WHERE CURRENT OF dolg_cur;
    DBMS_OUTPUT.PUT_LINE(vnev || ': ' || fiz);
  END LOOP;
END fizetesemeles;
/

EXEC fizetesemeles(5000, 30);

/* ROLLBACK; */
/* DROP TABLE dolgozo2; */
```

3. feladat: Készítsen tárolt eljárást *orszagbeszur* néven, amely három szöveges bemenő paramétert vár: a beszúrandó ország kódját, a nevét és annak a régiónak a nevét, ahová az ország tartozik! Az eljárás
- írja ki a *Nem létező régió!* hibaüzenetet, ha a nem létező régió nevét adtuk meg;
- írja ki a *Már létező országkód vagy országnév!* hibaüzenetet, ha már létező értéket adunk meg;
- ha nincs hiba, akkor szúrja be a megfelelő rekordot a *countries* táblába, és írjon erről üzenetet!

Tesztelje a tárolt eljárást az a következő értékekkel:
EXEC orszagbeszur('HU', 'Hungary', 'Európa');
EXEC orszagbeszur('UK', 'Hungary', 'Europe');
EXEC orszagbeszur('Hu', 'Hungary', 'Europe');
A tesztelés után dobja el az újonnan bevitt rekordot!

```
CREATE OR REPLACE PROCEDURE orszagbeszur(orszagkod VARCHAR2,
                                         orszagnev VARCHAR2,
                                         regio VARCHAR2) AS
  regio_az regions.region_id%TYPE;
  regio_nev regions.region_name%TYPE;
  talalat NUMBER(3);
  Letezo_Orszag EXCEPTION;
BEGIN
  SELECT region_id, region_name
    INTO regio_az, regio_nev
    FROM regions
    WHERE UPPER(region_name) = UPPER(regio);
  SELECT count(*)
    INTO talalat
    FROM countries
    WHERE UPPER(country_id) = UPPER(orszagkod)
      OR UPPER(country_name) = UPPER(orszagnev);
  IF talalat > 0 THEN
    RAISE Letezo_Orszag;
  END IF;
  INSERT INTO countries VALUES(UPPER(orszagkod),
                               INITCAP(orszagnev),
                               regio_az);
  DBMS_OUTPUT.PUT_LINE('A rekordot beszúrtuk!');
EXCEPTION
  WHEN No_Data_Found
    THEN DBMS_OUTPUT.PUT_LINE('Nem létező régió!');
  WHEN Letezo_Orszag
```

```
    THEN DBMS_OUTPUT.PUT_LINE('Már létezik az országkód vagy az
országnév!');
END orszagbeszur;
/

EXEC orszagbeszur('HU', 'Hungary', 'Európa');
EXEC orszagbeszur('UK', 'Hungary', 'Europe');
EXEC orszagbeszur('Hu', 'Hungary', 'Europe');

DELETE FROM countries WHERE country_id IN ('HU');
```

4. feladat: Készítsen tárolt függvényt *dolgozoszam* néven, amely bemenő paraméterként egy részleg nevét várja, válaszul pedig megadja a részlegen dolgozók számát!
Tesztelje a függvényt a egy utasításblokkal, amely bekér egy részlegnevet, majd megjeleníti a részleg létszámát a *dolgozoszam('&reszleg')* értékének kiírásával!

```
CREATE OR REPLACE FUNCTION dolgozoszam(reszlegnev VARCHAR2)
RETURN NUMBER AS
  talalat NUMBER(5);
BEGIN
  SELECT count(*)
    INTO talalat
    FROM employees INNER JOIN departments USING(department_id)
    WHERE UPPER(department_name) = UPPER(reszlegnev);
  RETURN talalat;
END dolgozoszam;
/

ACCEPT reszleg PROMPT 'Részleg neve:';
BEGIN
  DBMS_OUTPUT.PUT_LINE(dolgozoszam('&reszleg'));
END;
```

5. feladat: Készítsen függvényt *fizetesManagerFeletti*, amely
- *bemenő paraméterként* megkapja egy részleg nevét (*department_name*),
- *kimenő paraméterként* egy *VARCHAR2* és egy *NUMBER* típusú változót használ,
- *visszatérési értékként* szintén *NUMBER* típusú értékkel rendelkezik!

A függvény visszatérési értéke adja meg az osztály azon dolgozóinak számát, akik nem *Manager* beosztásúak (a *job_title* nem *Manager*-re végződik), de többet keresnek, mint a cég bármely *Manager* beosztású dolgozója! A két kimenő paraméter a legkisebb fizetésű ilyen dolgozó nevét és fizetését adja vissza! A függvény mellékhatásként listázza a felhasználói képernyőre a feltételnek megfelelő dolgozók nevét és beosztását!

```
CREATE OR REPLACE FUNCTION fizetesMangerFelett(
  reszleg VARCHAR2, legkisebb_nev OUT VARCHAR2, legkisebb_fiz OUT NUMBER)
RETURN NUMBER AS
  CURSOR emp_cur IS
    SELECT last_name, salary, job_title
    FROM (employees INNER JOIN jobs USING(job_id))
      INNER JOIN departments USING(department_id)
    WHERE UPPER(department_name) = UPPER(reszleg)
      AND UPPER(job_title) NOT LIKE '%MANAGER'
      AND salary > ANY (
        SELECT salary
        FROM employees INNER JOIN jobs USING(job_id)
        WHERE UPPER(job_title) LIKE '%MANAGER'
      )
```

```
    ORDER BY salary;
  emp_rec emp_cur%ROWTYPE;
  talalat NUMBER(5);
BEGIN
  talalat := 0;
  legkisebb_nev := ''; legkisebb_fiz := 0;
  FOR emp_rec IN emp_cur LOOP
    talalat := talalat + 1;
    IF talalat = 1 THEN
      legkisebb_nev := emp_rec.last_name;
      legkisebb_fiz := emp_rec.salary;
    END IF;
    DBMS_OUTPUT.PUT_LINE(RPAD(emp_rec.last_name || ':', 20) ||
emp_rec.job_title);
  END LOOP;
  RETURN talalat;
END fizetesMangerFelett;
/
```

Vagy

```
CREATE OR REPLACE FUNCTION fizetesManagerFelett(
  reszlegnev VARCHAR2,
  legkisebb_nev OUT VARCHAR2,
  legkisebb_fiz OUT NUMBER)
RETURN NUMBER AS
  CURSOR emp_cur IS
    SELECT last_name, salary, job_title
    FROM (employees INNER JOIN jobs USING(job_id))
      INNER JOIN departments USING(department_id)
    WHERE UPPER(department_name) = UPPER(reszlegnev)
      AND UPPER(job_title) NOT LIKE '%MANAGER'
      AND salary > ANY (
        SELECT salary
        FROM employees INNER JOIN jobs USING(job_id)
        WHERE UPPER(job_title) LIKE '%MANAGER'
      );
  emp_rec emp_cur%ROWTYPE;
  fo NUMBER(5);
BEGIN
  fo := 0;
  legkisebb_nev := '';
  SELECT max(salary)
    INTO legkisebb_fiz
    FROM employees;
  FOR emp_rec IN emp_cur LOOP
    fo := fo + 1;
    IF emp_rec.salary < legkisebb_fiz THEN
      legkisebb_nev := emp_rec.last_name;
      legkisebb_fiz := emp_rec.salary;
    END IF;
  END LOOP;
  RETURN fo;
END fizetesManagerFelett;
/
```

A függvényt tesztelje egy olyan utasításblokkal, amely rendelkezik egy *nev* és egy *fiz* változóval (ezek fogadják a függvény kimenő paramétereit), valamint egy *fo* nevű változóval, amely megkapja a függvény visszatérési értékét! Hívja meg a függvényt (pl. a *Sales* osztályra) és írja

ki, hogy *Összesen: fo fő* (*fo* helyén a visszatérési értékkel). Ha a *fo* értéke nem 0, akkor írja ki a legkisebb megfelelő fizetésű dolgozó nevét és fizetését!

```
DECLARE
  nev employees.last_name%TYPE;
  fiz employees.salary%TYPE;
  fo  NUMBER(5);
BEGIN
  fo := fizetesMangerFelett('Sales', nev, fiz);
  DBMS_OUTPUT.PUT_LINE('Összesen: ' || fo || ' fő');
  IF fo > 0 THEN
    DBMS_OUTPUT.PUT_LINE('Közülük a legkisebb fizetésű: ' || nev);
    DBMS_OUTPUT.PUT_LINE('Az ő fizetése: ' || fiz);
  END IF;
END;
/
```