**Feladat 001:** Készítsen függvényt, amely a paraméterben megkapott számról eldönti, hogy prím-e. A függvényt csak PL/SQL-ből szeretnénk hívni.

**Feladat 002:** Készítsen tárolt eljárást, amely paraméterben megkapja az intervallum kezdetét, az intervallum végét (zárt intervallum) és a lépésközt. Az eljárás ellenőrzi a paraméterek értékét, és amennyiben azok helyesek, az egyes értékekre az intervallumból (lépésköz figyelembevételével) meghívják a Feladat 001-ben megvalósított függvényt. Az eljárás a lefutását követően kimenő paraméterben átadja a hívó félnek a lefolytatott vizsgálatok számát, valamint, hogy hány prímet talált az adott intervallumon.

**Feladat 003:** Készítsen tárolt függvényt, amely a paraméterben megkapott sorszámú Fibonacci számot adja vissza a hívás helyére. A megoldás során használjon rekurziót.

**Feladat 004:** A feladat megoldásának első lépéseként az alábbi táblát hoztuk létre az adatbázisban.

**CREATE TABLE fibonacci\_szamok(**

**sorszam INT NOT NULL,**

**ertek INT NOT NULL,**

**CONSTRAINT PK\_fibonacci\_szamok PRIMARY KEY(sorszam)**

**);**

Készítsen tárolt eljárást, amely a paraméterben megkapja, hogy az első hány darab Fibonacci számot szeretnénk a fenti táblában eltárolni. A sorszámozás 0-tól kezdődik. A törlési művelet csak arra használható, hogy a felesleges számokat töröljük a táblából, azaz ha már létezik az adott sorszámú elem akkor frissíteni kell azértékét, ha nem, akkor be kell szúrni. Az eljárás a futás végén kimenő paraméterekben közli a beszúrt, a frissített és a törlésre került rekordok számát. Biztosítsa, hogy a számok ilyen módon való generálására azoknak a felhasználóknak is legyen lehetőségük, akik nem rendelkeznek semmilyen jobba a táblán.

**Feladat 005:** Készítsen tárolt függvényt, amely a paraméterben megkapott sorszámú Fibonacci számot adja vissza a hívás helyére. A megoldás során NE használjon rekurziót.

**Feladat 006:** Készítsen tárolt függvényt, amely a paraméterében megkapott szám faktoriálisát állítja elő. A megoldás során használjon rekurziót.

**Feladat 007:** Készítsen tárolt függvényt, amely a paraméterében megkapott szám faktoriálisát állítja elő. A megoldás során ne használjon rekurziót.

**Feladat 008:** Készítsen függvényt,amely egy szóról eldönti, hogy palindrom-e. Ennél a feladatnál palindrom szónak tekintjük azt az szót, amely visszafelé olvasva is ugyanazt jelenti (pl.: pap, kék, stb.). A függvényt úgy készítse el, hogy lekérdezésből is hívható legyen.

**Feladat 009:** A következő feladat megoldásához az alábbi táblát készítettük az adatbázisban.

**CREATE TABLE szavak(**

**id INT NOT NULL,**

**szo VARCHAR2(250) NOT NULL,**

**CONSTRAINT PK\_szavak PRIMARY KEY(id)**

**);**

Készítsen tárolt eljárást, amely paraméteben megkaphatja, hogy a fenti táblából hány szót szeretnénk vizsgálni arra vonatkozóan, hogy palindrom-e (opcionálisan). Ha a paraméterben nem közlünk értéket, akkor a teljes táblát vizsgálni fogjuk. Az eljárás kimenő paraméterben megmondja, hogy a szavak táblában a vizsgálat alapján hány palindrom szó van. Az eljárás végrehajtása végén tudnunk kell azt is, hogy hány vizsgálat történt.

**Feladat 010:**  A következő feladat megoldásához az alábbi táblát készítettük az adatbázisban.

**CREATE TABLE palindrom\_szavak(**

**id INT NOT NULL,**

**szo VARCHAR2(250) NOT NULL,**

**CONSTRAINT PK\_palindrom\_szavak PRIMARY KEY(id)**

**);**

Készítsen függvényt, amelyet meghívva előállítjuk a fenti tábla számára a következő lehetséges id értéket: ha a tábla üres, akkor az id értéke 0, ha nem üres, akkor a maximális id értéket léptetjük a következő id-re. Ügyfeljen arra, hogy a függvény mindig a következő páros értéket adja vissza a hívás helyére, függetlenül attól, hogy a maximális id értéke páros vagy páratlan.

**Feladat 011:** Készítsen tárolt lejárást amely a szavak táblából a paldindrom\_szavak táblába másolja a palindromokat. A művelet kezdetén a tárolt eljárás kiüríti a palindrom\_szavak táblát. Az eljárás futásának végén a hívás helyére kimenő paraméterben adja át, hogy a palindrom táblába hány rekordot szúrt be.

**Feladat 012:** Készítsen függvény, amely egy paraméterben megkapott szövegről eldönti, hogy lehet-e használható Neptun kód. A függvényt úgy készítse el, hogy csak PL/SQL-ből szeretnénk hívni.

**Feladat 013:** Készítsen PL/SQL függvényt, amely a paraméterében megkapott TAJ számról eldönti, hogy helyes-e.

**Feladat 014:** Készítsen PL/SQL függvényt, amely a paraméterben megkapott adóazonosító jelről eldönti, hogy helyes-e. A függvény az ellenőrzéshez a születési dátumot is bekéri.

**Feladat 015:** Készítsen PL/SQL függvényt, amely a paraméterben megkapott rendszármól eldönti, hogy helyes-e!