**1. Osztók**

**Feladat leírása**

(1) Hozzon létre egy „osztok” adatbázist. Az adatbázisban 2 tábla található:

-szamok { szam (1 és 1000000 közötti egész szám, elsődleges kulcs }

-szam\_osztoi { szam (egész szám, elsődleges kulcs), osztoja (egész szám, elsődleges kulcs), prim\_e (karakter: i/n) }

Pl. 12 esetében vagy 3 esetében:

A „szamok” táblában:

|  |
| --- |
| **szam** |
| 12 |
| 3 |

A „szam\_osztoi” táblában:

|  |  |  |
| --- | --- | --- |
| **szam** | **osztoja** | **prim\_e** |
| 12 | 1 | n |
| 12 | 2 | i |
| 12 | 3 | i |
| 12 | 4 | n |
| 12 | 6 | n |
| 12 | 12 | n |
| 3 | 1 | n |
| 3 | 3 | i |

Készítsen a teljes adatbázisról másolatot exportálással!

(2) Az adatbázishoz készítsen menüvezérelt grafikus alkalmazást, mely a következő információkat képes nyújtani jól átlátható formátumban, mindig csak egy kiválasztott kérdésre válaszolva:

a. Prímszám-e: a felhasználó által bekért számról kiírja, hogy prímszám-e.

b. Feltöltés: a felhasználó által megadott egész szám osztóit meghatározza. Ha még nincs az adatbázisban, akkor kiszámolja és eltárolja a megfelelő táblákban. Tárolásnál az egyes osztóknál azt is eltárolja, hogy az prímszám-e vagy sem. Ha a megadott szám már benne van az adatbázisban, akkor jelzi, hogy az a szám már feltöltésre került.

c. Kiíratás: a felhasználó által megadott szám osztóit kiírja vesszővel elválasztva. Ha az adott szám már benne van az adatbázisban, csak akkor történik a kiíratás. Ha még nincs az adatbázisban, akkor hibaüzenet, hogy az a szám még nincs feltöltve.

d. Prímszámok listája: a felhasználó által megadott alsó és felső határszámok közötti prímszámokat írja ki az adatbázisból, tehát csak a már feltöltött prímosztókat, és mindegyiket csak egyszer.

e. Statisztika: készítsen statisztikát, hogy a már feltöltött számok esetében melyik osztó hányszor szerepel az osztók között. A statisztika darabszám szerinti csökkenő sorrendben jelenjen meg.

f. Törlés: a felhasználó által megadott szám törlése az adatbázisból (a számot is és annak osztóit is).