

2. Kultúrtörténet

A 2018/19-es tanévben kultúrtörténeti versenyt szerveztek a középiskolásoknak. Minden fordulóban egy-egy jelentős művész kapcsán kellett feladatokat megoldaniuk a tanulóknak. A vetélkedés egész tanévben tartott, ahogy az egyik feladatsor beküldési határideje lejárt, szinte azonnal kitűzték a következő forduló feladatait. Az értékelésbe beleszámított a megoldási idő is, ezért a csapatok a feladatsor egy-egy megoldott feladatát azonnal beküldték. Minden csapat minden feladatsorra küldött megoldást.

Az adatbázis a versennyel kapcsolatos következő táblákat tartalmazza:

Táblák:

feladatsor (id, nevado, ag, kituzes, hatarido)

id	A feladatsor azonosítója (szám), ez az elsődleges kulcs
nevado	A feladatsor témáját adó személy neve (szöveg), nincs két azonos név A személy neve egy, két vagy három szóból áll, melyeket pontosan egy szóköz választ el egymástól
ag	A művészeti ág, amelyhez a személy tartozik (szöveg)
kituzes	A feladatsor kitűzési dátuma (dátum)
hatarido	A feladatsor feladatainak beadási határideje (dátum)

feladat (id, feladatsorid, pontszam)

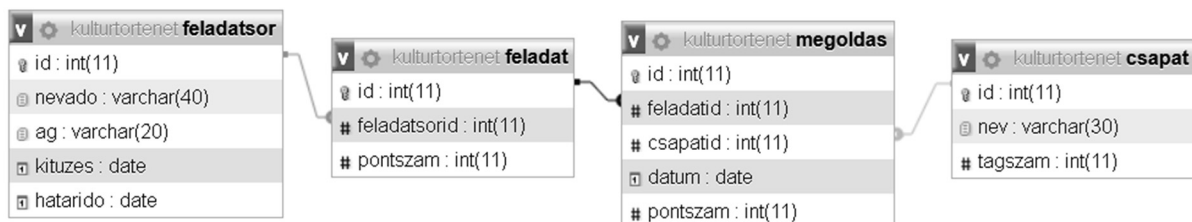
id	A feladatsor egy feladatának azonosítója (szám), ez az elsődleges kulcs
feladatsorid	A feladatsor azonosítója (szám), idegen kulcs
pontszam	A feladatban elérhető pontszám (szám)

megoldas (id, feladatid, csapatid, datum, pontszam)

id	A feladat egy megoldásának azonosítója (szám), ez az elsődleges kulcs
feladatid	A megoldott feladat azonosítója (szám), idegen kulcs
csapatid	A feladatot megoldó csapat azonosítója (szám), idegen kulcs
datum	A megoldás beadási dátuma (dátum)
pontszam	A megoldásra adott pontszám (szám)

csapat (id, nev, tagszam)

id	A versenyre nevező csapat azonosítója (szám), ez az elsődleges kulcs
nev	A csapat neve (szöveg), nincs két azonos nevű csapat
tagszam	A csapat létszáma (szám)



A következő feladatokat megoldó SQL-parancsokat rögzítse a feladatok végén zárójelben megadott nevű és .sql kiterjesztésű szöveges állományba! Például a 3. feladat megoldása a 3csakegy.sql nevű állományba kerüljön! A javítás során csak ezeknek az állományoknak a tartalmát értékeli. Ügyeljen arra, hogy a lekérdezésekben pontosan a kívánt mezők szerepeljenek, felesleges mezőt ne jelenítsen meg!

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Az adatbázist és a táblákat létrehozó, és az adatokat a táblába beszűrő SQL-parancsokat a *kulturforras.sql* állomány tartalmazza.

1. Futtassa az SQL-szerveren a *kulturforras.sql* parancsfájlt, és a továbbiakban a **kulturortenet** adatbázisban dolgozzon! (A „Nincs kiválasztott adatbázis” üzenet nem befolyásolja az adatimportálás sikerességét.)
2. Lekérdezés segítségével írassa ki azon csapatok nevét, amelyek neve a # karakterrel kezdődik! (**2csapatok**)
3. A **feladatsor** táblát használva, lekérdezés segítségével jelenítse meg a feladatsor névadójának nevét, ha abban pontosan egy szóköz van! (**3csakegy**)
4. Készítsen lekérdezést, amely megadja, hogy ki a névadója a 2018. szilveszterkor aktív feladatsornak! (**4szilveszter**)
5. Készítsen lekérdezést, amely meghatározza a végeredményt! A csapatok neve és az általuk elért összpontszám jelenjen meg, utóbbi szerint csökkenő sorrendben! (**5eredmeny**)
6. Eredetileg úgy tervezték, hogy minden feladatsor 150 pontos lesz. Néhány esetben a kitűzés után kellett módosítani a feladatsoron, így ez nem valósult meg. Adja meg lekérdezéssel azokat a feladatsorokat, amelyek nem 150 pontosak! A feladatsor névadóját, a művészeti ágat és a pontszámot jelenítse meg! (**6nem150**)
7. Lekérdezés segítségével listázza ki azon csapatok nevét, amelyeknek volt maximális pontszámot érő feladata! Minden csapat neve egyszer jelenjen meg! (**7hibatlan**)
8. Bár a versenyzők lelkesek voltak és törekedtek minden feladatot megoldani, ennek ellenére előfordult, hogy nem minden feladatra adtak be megoldást. Készítsen lekérdezést, amely megadja, hogy a „#win” csapat mely feladatsorokból hány feladatot nem adott be! Jelenítse meg a feladatsor névadóját és a be nem adott feladatok számát! (**8#win**)
9. Készítsen lekérdezést, amely megadja, hogy az „irodalom” művészeti ághoz tartozó feladatsorok közül melyeket kellett ugyanabban a hónapban beadni, mint amikor kitűzték? Adja meg a feladatsorok névadóját! (**9ugyanabban**)

SQL-nyelven két dátum napokban kifejezett különbségét a **Datediff** függvénnyel lehet meghatározni. Az alábbi példák mutatják használatát:

```
Datediff('2020-12-02', '2020-12-01')
```

```
Datediff('2019-01-04', '2018-12-31')
```

Az első esetben 1, a másodikban 4 a visszaadott érték. A következő feladatok megoldásánál használhatja a **Datediff()** függvényt.

10. Lekérdezés segítségével adja meg, melyik feladatsor megoldására volt a legkevesebb idő! A feladatsor névadóját jelenítse meg! Ha több ilyen feladatsor van, elegendő az egyiket megadnia. (**10legrovidebb**)
11. Készítsen lekérdezést, amely megadja, hogy mely feladatokat tüzték ki az előző beadási határidejét követő napon! A feladatsor névadóját és a kitűzés idejét jelenítse meg! A feladat megoldása során kihasználhatja, hogy egyszerre csak egy feladatsor volt aktív. (**11rogtan**)

35 pont

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

3. Reklám

Egy termék hirdetésének hatékonyságát vizsgálták három, egymáshoz hasonló lélekszámú városban. Egy 30 napos időszak középső tíz napján zajlott reklámkampány a három város közül kettőben. Az egyik városban a helyi televízióban reklámozták a terméket, a másik városban utcai plakáton hirdették, mindkét városban az időszak 11-edik napjától a 20-adik napjáig. A harmadik városban nem volt reklámkampány, illetve az előbbi két városban sem volt az időszak első 10 és az utolsó 10 napján.

A kampány sikerességét a 30 nap során a három városból leadott rendelések alapján mérték. Ezek az adatok megtalálhatók a `rendel.txt` szöveges állományban, melynek néhány sora a következő:

```
13 PL 2
13 PL 1
13 TV 1
13 NR 1
13 TV 3
```

A rendelések időrendben vannak, minden sorban egy-egy rendelés szerepel. Egy soron belül az első szám a rendelés napja, a következő két betű azt a várost jelöli, ahol a rendelést leadták, míg a harmadik elem a termékből rendelt darabszám. Az adatokat a soron belül szóközök választják el egymástól. A fenti példában látható rendelések mind a 13. napon történtek. Az első két rendelést abban a városban adták le, ahol plakáton hirdettek (PL). A harmadik és ötödik rendelést abban a városban adták le, ahol a helyi televízióban hirdettek (TV). A negyedik rendelést abban a városban adták le, ahol nem volt reklámkampány (NR). Az első példában látható rendeléssel 2 terméket, az utolsó rendeléssel 3-at, míg a középső három rendelés mindegyikével 1–1 terméket rendeltek.

Egy-egy napon általában több rendelés érkezett, és tudjuk, hogy minden nap rendeltek valamelyik városból, de nem feltétlenül rendeltek minden nap mindhárom városból. Az összes rendelések száma 1000-nél kevesebb, és minden rendelésben a darabszám 10-nél kisebb.

Készítsen programot, amely a három városból származó rendelések adatait felhasználva az alábbi feladatokat oldja meg! A program forráskódját mentse `reklam` néven! A program megírásakor a bemeneti állományban található, vagy a felhasználó által megadott adatok helyességét, érvényességét nem kell ellenőriznie, feltételezheti, hogy azok a leírtaknak megfelelnek.

A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: `2. feladat`) és utaljon a megjelenítendő tartalomra! Ha a felhasználótól kér be adatot, jelenítse meg a képernyőn, hogy milyen értéket vár! Az ékezetmentes kiírás is elfogadott a képernyőn történő megjelenítés és a szöveges állományba való írás során.

1. Olvassa be és tárolja el a további feldolgozáshoz a `rendel.txt` állomány tartalmát!
2. Állapítsa meg, hogy hány rendelés történt a teljes időszakban, és írja a képernyőre a rendelések számát!
3. Kérje be a felhasználótól egy nap számát, és adja meg, hogy hány rendelés történt az adott napon!
4. Számolja meg, hogy hány nap nem volt rendelés a reklámban nem érintett városból, és írja ki a napok számát! Ha egy ilyen nap sem volt, akkor írja ki „Minden nap volt rendelés a reklámban nem érintett városból” szöveget!

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

5. Állapítsa meg, hogy mennyi volt az egy rendelésben szereplő legnagyobb darabszám, és melyik volt az a nap, amikor az első ilyen számú rendelést leadták! Az eredményt a lenti minta szerint írja ki!
6. Készítsen függvényt `osszes` néven, amely megadja, hogy mennyi volt egy adott városból egy adott napon a rendelt termékek száma! A függvény bemenete a három város egyikére utaló kétbetűs szöveg és a nap sorszáma legyen. Amennyiben szükséges, akkor további paramétert is felvehet a rendelések adatainak elérése érdekében. A függvény visszaadott értéke a rendelt darabszámok összege legyen! A függvényt például a következő módon lehessen meghívni: `osszes("PL", 7)`. A függvényt a későbbiekben felhasználhatja a további feladatok megoldásakor.
7. Számítsa ki, hogy a kampány utáni első napon, azaz a 21-edik napon melyik városból mennyit rendeltek a termékből! Az eredményt a lenti mintának megfelelő formában írja ki!
8. Összesítse városonként, hogy hány rendelés érkezett az első 10, a 11-20-adik valamint a záró 10 napon! Az eredményt (a fejléct is beleértve) táblázatos formában, tabulátorokkal tagoltan jelenítse meg a képernyőn, illetve írja azonos formátumban a `kampany.txt` szöveges állományba!

Minta a felhasználóval történő kommunikációhoz:

```
2. feladat:  
A rendelések száma: 971  
3. feladat:  
Kérem, adjon meg egy napot: 9  
A rendelések száma az adott napon: 27  
4. feladat:  
3 nap nem volt a reklámban nem érintett városból rendelés  
5. feladat:  
A legnagyobb darabszám: 9, a rendelés napja: 22  
7. feladat:  
A rendelt termékek darabszáma a 21. napon PL: 43 TV: 36 NR: 18  
8. feladat:  
Napok    1..10   11..20   21..30  
PL        98     159     106  
TV        97     143     100  
NR        91     86      91
```

A feladat a következő oldalon folytatódik.