

PRÓBA ÉRETTSÉGI

**INFORMATIKAI  
ISMERETEK**

tantárgyból

2020.

Feladat	Szerző	Lektor
Hálózati ismeretek	<b>Kovacsics Tamás</b>	<i>Horváth Norbert</i>
Programozás	<b>Nits László</b>	<i>Kottra Richárd</i>
Web - Adatbázis-kezelés	<b>Bognár Pál</b>	<i>Nits László</i>

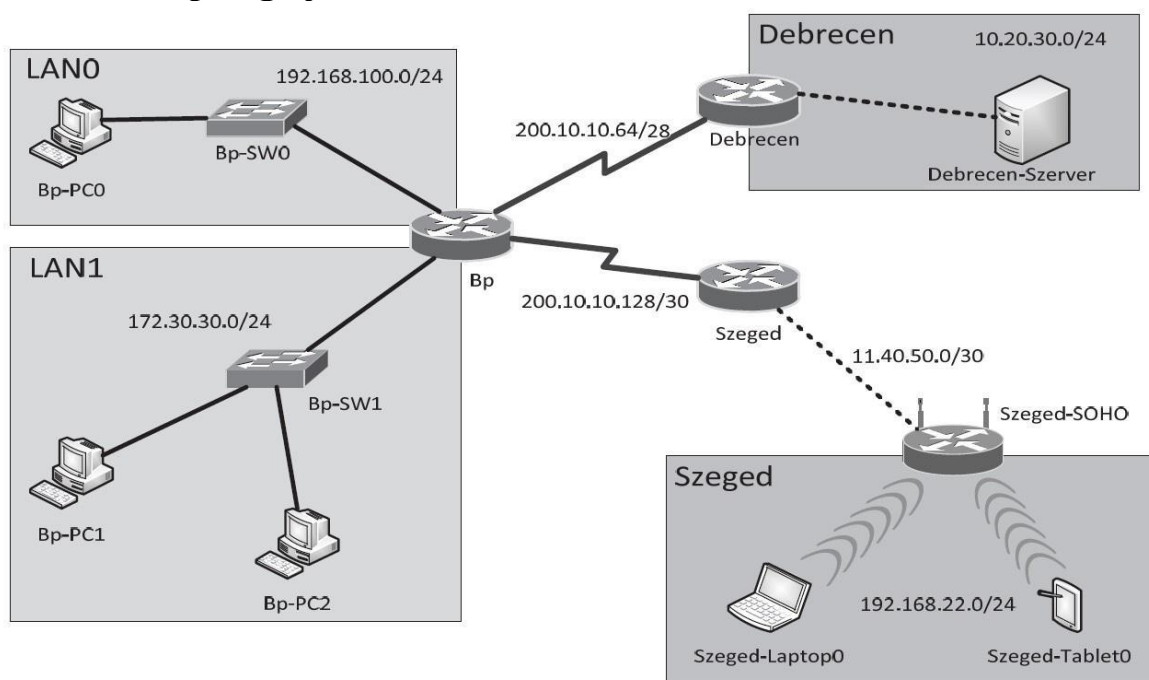
Feladatsor szerkesztő	<b>Kottra Richárd</b>
Programozás - Java megoldás	<b>Kiss Balázs</b> - People Come First egyesület

## 1. Hálózati ismeretek<sup>1</sup>

**40 pont**

Az UTP Bt. három különböző városban lévő telephelyén szándékozik hálózatot kiépíteni. Az Ön feladata, hogy a megadott tervek alapján szimulációs programmal elkészítse a vállalat teszhálózatát. Munkáját utpbt néven mentse az Ön által használt szimulációs program alapértelmezett formátumában!

### A hálózat topológiája



### Hálózati címzés

Eszköz	IP cím	Alhálózati maszk	Alapértelmezett átjáró
Bp	192.168.100.1	255.255.255.0	-
	172.30.30.1	255.255.255.0	-
	200.10.10.65	255.255.255.240	-
	200.10.10.129	255.255.255.252	-
Bp-SW1	172.30.30.3	255.255.255.0	172.30.30.1
Bp-PC0	192.168.100.11	255.255.255.0	192.168.100.1
Bp-PC1	DHCP-kliens		
Bp-PC2			
Debrecen	10.20.30.1	255.255.255.0	-
	200.10.10.66	255.255.255.240	
Debrecen-Szerver	10.20.30.99	255.255.255.0	10.20.30.1
Szeged	11.40.50.1	255.255.255.252	-
	200.10.10.130	255.255.255.252	-
Szeged-SOHO	192.168.22.1	255.255.255.0	-
	11.40.50.2	255.255.255.252	11.40.50.1
Szeged-Laptop0	DHCP-kliens		
Szeged-Tablet0			

<sup>1</sup> Forrás: Távközlés ismeretek középszintű érettségi vizsga 2018. október 19.

## Beállítások

1. A szimulációs programban válassza ki a feladat megoldásához szükséges eszközöket a következő információk alapján:
  - a. A forgalomirányítók rendelkezzenek minimum két Ethernet interfésszel, amelyek legalább 1 Gb/s sebességűek, valamint két szinkron soros interfésszel!
  - b. A kapcsolók legalább 24 portosak legyenek!
2. A kiválasztott eszközöket kösse össze a topológiai ábrának megfelelően!
3. A hálózati eszközökön (kapcsolók, forgalomirányítók) a konfigurációban megjelenő eszköznév a topológiai ábrának megfelelő név legyen (kivéve a vezeték nélküli forgalomirányítón)!
4. Állítsa be a forgalomirányítók és az Bp-SW1 kapcsoló, illetve a Szeged-SOHO vezeték nélküli forgalomirányító IP-címeit a táblázatnak és a topológiai ábrának megfelelően!
5. A Bp forgalomirányítónál és az Bp-SW1 kapcsolónál biztosítani kell a távoli – telnet protokollon keresztüli – elérést! A távoli eléréshez használt jelszó **BPvty123** legyen!
6. A Bp forgalomirányítónál és az Bp-SW1 kapcsolónál a privilegizált módot védő jelszó a **BPena123** legyen!
7. Állítsa be a Debrecen-Szerver és a Bp-PC0 állomás számára az IP konfigurációt a táblázat alapján! A PC-n a DNS szerver címe Debrecen-Szerver IP-címe legyen!
8. A Bp forgalomirányító a LAN1 számára DHCP szerverszolgáltatást biztosít. A DHCP szervernél a következő beállításokat kell elvégeznie:
  - a. A címkiosztást a 100. IP-címmel kezdje!
  - b. A DNS szerver címe a Debrecen-Szerver címe legyen!
  - c. Az alapértelmezett átjárót állítsa be megfelelően!
9. Állítsa be a LAN1 hálózatban lévő munkaállomások számára, hogy IP-adataikat automatikusan kapják a DHCP szervertől!
10. A Szeged-SOHO nevű vezeték nélküli forgalomirányítón végezze el a következő beállításokat:
  - a. DHCP szerverként 192.168.22.50-nel kezdődően, 192.168.22.150-nel bezárólag osszon IP-címeket a klienseknek! A DNS szerver címe legyen a Debrecen-Szerver IP-címe!
  - b. A vezeték nélküli hálózaton az SSID SzegedNet legyen!
  - c. A vezeték nélküli hitelesítés WPA2/PSK, a titkosítás AES segítségével történjen! A kulcs **Paprika777** legyen!
11. Csatlakoztassa a vezeték nélküli klienseket a vezeték nélküli hálózathoz! Állítsa be, hogy a kliensek az IP-adataikat automatikusan kapják a DHCP szervertől!
12. A cégnél dinamikus forgalomirányítást kívánnak alkalmazni. Állítsa be a RIP forgalomirányító protokoll 2-es verzióját mindhárom forgalomirányítón a következők szerint:
  - a. Mindhárom forgalomirányítón hirdesse az összes közvetlenül csatlakoztatott hálózatot!
  - b. Mindhárom forgalomirányítón állítsa be, hogy a használt Ethernet típusú interfészeken ne történhessen meg a forgalomirányítási csomagok hirdetése, azaz ezek az interfészek legyenek passzívok!
  - c. Mindhárom forgalomirányítón tiltsa le az automatikus útvonalösszevonást a RIP protokoll esetén!

13. A későbbiek során a Debrecen forgalomirányítón beállított tűzfal miatt kívülről nem lesznek elérhetőek a privát címek, ezért statikus NAT beállításával kell biztosítani a szerver elérhetőségét. A Debrecen forgalomirányítón állítson be statikus NAT szolgáltatást, amellyel biztosítja, hogy a Debrecen-Szerver a budapesti és a szegedi hálózathoz a **200.10.10.67** IP-címmel legyen elérhető!
14. Minden hálózati eszközön mentse el a konfigurációt, hogy azok újraindítás után is a megadott beállításokkal működjenek!

## 2. Programozás

**40 pont**

### Iskola<sup>2</sup>

A `nevek.txt` állományban rögzítettük egy középiskola tanulóinak néhány adatát. Feltételezheti, hogy nincs két azonos nevű tanuló egy osztályban. Az állomány tartalma soronként:

- iskola kezdésének éve (2004-2007)
- az osztály betűjele (a-e)
- a diák neve (ékezetek nélkül).

Az adattagok pontosvesszővel vannak elválasztva. Példa (részlet) a `nevek.txt` állományra:

```
2004;d;Vavrek Kristof  
2006;e;Hidas Reka  
2006;d;Kun Michael
```

### Megoldásában:

- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!*

Oldja meg a következő feladatokat:

1. Készítsen `Iskola` néven **konzol típusú projektet**, melyben megoldja a következő feladatokat!
2. Olvassa be és tárolja el egy célszerűen megválasztott adatszerkezetben (pl.: vektor, lista) a `nevek.txt` állományban lévő adatokat!
3. Írja ki a képernyőre, hogy hány tanuló jár az iskolába!
4. Írja ki a képernyőre a minta szerint, hogy ki rendelkezik a leghosszabb névvel! A szóközöket ne számolja a név hosszához! Holtverseny esetén minden, leghosszabb névvel rendelkező tanuló neve jelenjen meg! Jelenítse meg a kiírásban a leghosszabb név/nevek hosszát is a minta szerint!
5. Az iskolai rendszergazdának egyedi azonosítókat kell készítenie a számítógép-hálózat használatához. Az azonosítókat a következő módon alakítja ki: első karaktere az évfolyam utolsó számjegye (pl.: 2006 esetén 6), következő karakter az osztály betűjele, majd a vezetéknév első három karaktere, végül első keresztnév első három karaktere következik. Az azonosítóban mindenütt kisbetűk szerepelnek. Feltételezhetjük, hogy a vezetéknév és az első keresztnév legalább 3 karakteres. Készítsen jellemzőt vagy függvényt, melyben meghatározza a rendelkezésre álló adatokból a tanuló azonosítóját! Az azonosítókat ne tárolja! Az elkészített jellemzőt/függvényt felhasználva írja ki az adatszerkezetben tárolt első és utolsó tanuló azonosítóját a minta szerint!

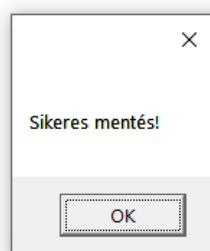
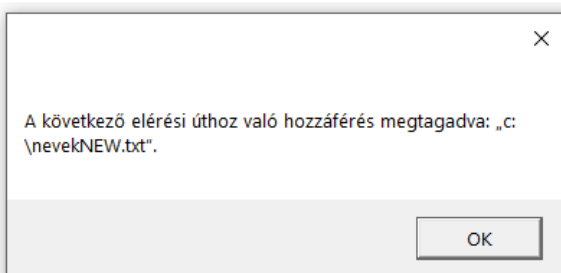
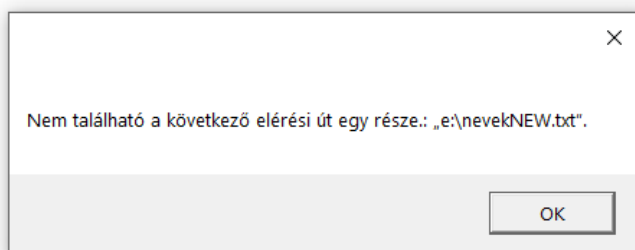
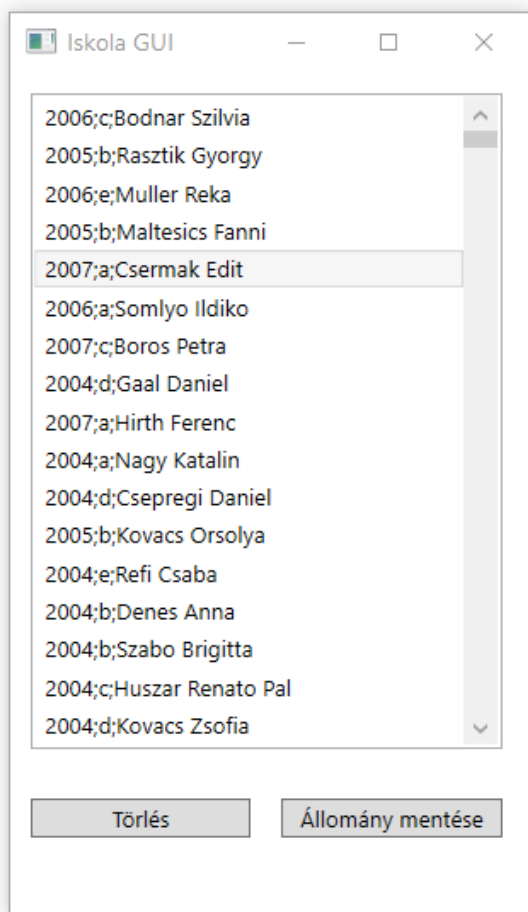
<sup>2</sup> Forrás: <https://docplayer.hu/4402962-Informatika-emelt-szintu-probaerettsegi.html>

6. Kérjen be egy azonosítót és állapítsa meg, hogy ki tartozhat hozzá! A tanuló adatait írja a képernyőre! Ha nem talál megfelelő diákot, akkor a „Nincs megfelelő tanuló.” mondatot jelenítse meg a minta szerint!
7. A forráskódjába a meglévő osztálydefiníció fölé illessze be a `java.txt` vagy a `csharp.txt` forrásállományból a `JelszóGeneráló` osztályt definiáló kódrészletet, majd hozzon létre belőle egy osztálypéldányt (objektumot)! Az osztálypéldány konstruktorát a véletlenszámok előállítására alkalmas beépített objektummal paraméterezze! Válaszon ki véletlenszerűen egy tanulót, majd a `JelszóGeneráló` osztály `Jelszó()` metódusának hívásával írjon ki egy 8 karakter hosszú jelszót a minta szerint!
8. Készítsen **grafikus alkalmazást** a következő feladatok megoldására, melynek projektjét `IskolaGUI` néven mentse el!
9. Az alkalmazás grafikus felhasználói felületét alakítsa ki a minta szerint! Az ablak címsorában a „Iskola GUI” szöveget jelenítse meg!
10. Az ablakon található listába a program induláskor töltse be a `nevekGUI.txt` állomány sorait! A lista elemei a forrásfájl egy-egy sora (továbbiakban tanulója) legyen!
11. Oldja meg, hogy a kijelölt tanuló a „Törlés” parancsgomb lenyomása után törlésre kerüljön a listából! Ha a listában nincs kijelölt tanuló, akkor törléskor a „Nem jelölt ki tanulót!” szöveg jelenjen meg egy felugró ablakban!
12. Ha az „Állomány mentése” parancsgombra kattintunk, akkor történjen meg a listából a tanulók mentése a `nevekNEW.txt` állományba, melynek szerkezete a forrásállomány szerinti legyen! Ha a mentés sikeres volt, akkor a „Sikeres mentés!” felirat jelenjen meg egy felugró ablakban! Ha az állomány mentése sikertelen, akkor a hibaüzenet (a hibához tartozó beépített üzenet/message) jelenjen meg egy felugró ablakban! Lehetséges hibaokkokhoz tartozó beépített üzeneteket a minták között talál!

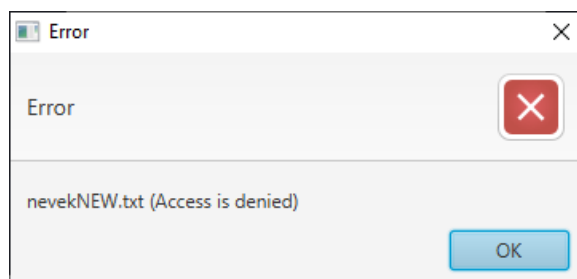
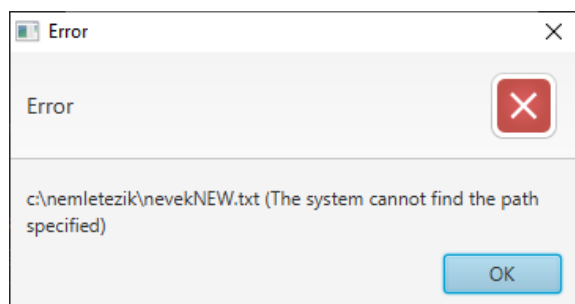
**Minták a konzolos alkalmazás futására:**

3. feladat: Az iskolába 650 tanuló jár.
4. feladat: A leghosszabb (25 karakter) nevű tanuló(k):  
Karatson Patricia Mercedesz
5. feladat: Azonosítók  
Első: Bodnar Szilvia - 6cbodsz  
Utolsó: Krizsan Vivien Evelin - 6ckriviv
6. feladat: Kérek egy azonosítót [pl.: 4dvavkri]: 4atolrek  
2004 a Tolcsvay-Nagy Reka Luca
7. feladat: Jelszó generálása  
Szabo Mihaly - hn4xyutl
3. feladat: Az iskolába 650 tanuló jár.
4. feladat: A leghosszabb (25 karakter) nevű tanuló(k):  
Karatson Patricia Mercedesz
5. feladat: Azonosítók  
Első: Bodnar Szilvia - 6cbodsz  
Utolsó: Krizsan Vivien Evelin - 6ckriviv
6. feladat: Kérek egy azonosítót [pl.: 4dvavkri]: 5cmogab  
Nincs megfelelő tanuló.
7. feladat: Jelszó generálása  
Tompos Zsolia - smr94dud

### Minták a grafikus alkalmazás futására:



### Java hibaüzenetek:



### 3. Weboldalak kódolása és adatbázis-kezelés 20+20pont

#### Formula One<sup>3</sup>

A következő feladatban egy weboldalt kell készítenie a Forma-1 történetével kapcsolatban a feladatleírás és a minta szerint, valamint a 2019-es szezon adataiból álló adatbázisban kell műveleteket végrehajtania.

A két feladatrész egymástól függetlenül, tetszőleges sorrendben megoldható. Az első feladatrészben a forrásként kiadott weboldalon kell módosításokat végeznie a leírás és a minta alapján! Ahol a feladat másként nem kéri, a formázási beállításokat a `site.css` stílusállományban végezze el, az új szelektorokat az állomány végén helyezze el!

Nyissa meg az `index.html` állományt és szerkessze annak tartalmát az alábbiak szerint:

1. A weboldal karakterkódolása `utf-8`, a weboldal nyelve magyar, a böngésző címsorában megjelenő cím „Formula One” legyen!
2. A weboldal fejrészében helyezzen el hivatkozást a `site.css` stíluslapra, valamint a `main.js` állományra! Ügyeljen arra, hogy a `jquery.min.js` állomány hamarabb kerüljön betöltésre, mint a `main.js`!
3. A weboldalon készítsen egy újabb menüpontot az alábbi leírás és a minta alapján:
  - a. Az új menüpont a „Kezdőlap” és a „Leg-ek” menüpontok között helyezkedjen el! A neve „Történet” legyen és kattintásra az oldalon belül a `history` azonosítójú szekcióra ugorjon!
  - b. A `history` azonosítójú szekciót formázza meg az alábbi három osztályazonosítóval: `section-100`, `bg-image` és `p-2`!
  - c. A `tortenete.txt` állomány tartalmát másolja be a `history` azonosítójú szekcióba!
  - d. Alakítsa ki a `history` azonosítójú szekción belül a bekezdéseket és a címsorokat! A „Története” felirat második szintű címsor, az egyes alcímek („A kezdetek”, „A Forma-1 születése”, stb.) harmadik szintű címsorok legyenek!
  - e. A „Története” második szintű címsort formázza meg a `text-center` és `py-3` osztályazonosítók segítségével!
4. A „Leg-ek” (`records`) szekcióban a képeken látható versenyzők nevét („Rubens Barrichello”, „Michael Schumacher”, „Juan Manuel Fangio”), valamint a „Ferrari” csapatnevet a minta szerint alakítsa félkövérré! Ehhez hozzon létre a `site.css` stílusállományban egy saját szelektort, majd alkalmazza a kiemeléshez!
5. Az alábbi lépések segítségével alakítsa ki a legfiatalabb és legidősebb versenyző, valamint a legtöbb Grand Prix kereteket! (Az alábbi lépéseket háromszor kell végrehajtania, a három „leg”-hez kapcsolódóan. Mindhárom keretnek a helye egy-egy megjegyzéssel jelölve van az `index.html` dokumentumban.) *A tartalomhoz tartozó szövegeket és fájlneveket megtalálja a `legek.txt` állományban.*
  - a. Hozzon létre keretet (`div`) és formázza meg az alábbi osztályazonosítókkal: `bg-black`, `col-sm-12`, `col-md-4`, `p-3` és `text-center`!
  - b. Az előző pontban létrehozott kereten belül hozzon létre egy harmadik szintű címsort, amit formázzon meg a következő osztályazonosítókkal: `bg-dark`, `p-2` és `w-100`!
  - c. Az így létrehozott címsorba írja vagy másolja be a címet („Legfiatalabb versenyző”, „Legidősebb versenyző”, „Legtöbb Grand Prix”)!

<sup>3</sup> Forrás: Ergast Developer API <http://ergast.com/mrd/>



- d. A cím alá szúrja be az `img` könyvtárban található képeket (`max_verstappen.jpg`, `louis_chiron.jpg`, `monza_info.jpg`)! A képeket formázza meg a `w-100` osztályazonosítóval! Amennyiben a képek nem jelennek meg, vagy fölé viszik az egérkurzort, akkor a versenyző/pálya neve jelenjen meg („Max Verstappen”, „Louis Chiron”, „Monza”)!
  - e. A képaláírásoknak hozzon létre a képek alatt egy-egy bekezdést, és másolja bele a képek alatti szövegeket a minta szerint!
6. Végezze el az alábbi műveleteket az `index.html` dokumentum „Átlagsebesség számítás” (`avgspeed`) szekciójában!
- a. Az űrlap alatti JavaScript blokkban hívja meg a `main.js` fájlban definiált `showCircuit` nevű függvényt!
  - b. A pályákat tartalmazó (`circuit` azonosítójú) lenyíló listát bővítse ki a monzai pályával! A megjelenő szöveg „Monza” legyen, az opció értéke pedig „ITA”!
  - c. A köridő rögzítésére szolgáló beviteli (`laptime` azonosítójú) mezőt módosítsa úgy, hogy csak numerikus értéket (számot) lehessen rögzíteni benne!
  - d. A `main.js` állományban készítse el a `calculate` függvény törzsét! A függvénynek a pálya hosszából és a köridőből kell átlagsebességet számolnia: a kiválasztott pálya hosszát kell elosztani a köridővel (a köridőt – mivel másodpercben van megadva – el kell osztani 3600-zal). Az egyes pályák hossza km-ben:
    - Hungaroring (HUN): 4,381 km
    - Monaco (MON): 3,337 km
    - Spa-Francorchamps (BEL): 7,004 km
    - Monza (ITA): 5,793 kmA kapott eredményt – mértékegységgel (km/h) együtt – jelenítse meg az átlagsebesség (`averagespeed` azonosítójú) mezőben!
7. A láblécben (`footer`) található linkeket alakítsa hivatkozássá! A két linkhez tartozó szöveg legyen „F1 történet” és „Wikipedia”!

A következő beállításokat a `site.css` stíluslapon végezze!

8. Hozzon létre új elemszelektort a bekezdések formázásához! A bekezdések legyenek sorkizártak!
9. A weboldal háttérszíne legyen fekete (`#000`), a weboldalon megjelenő szöveg színe legyen fehér (`#fff`)!
10. Egészítse ki a láblécben található hivatkozásokra vonatkozó szelektort úgy, hogy a hivatkozások legyenek aláhúzva!

**Minták a következő oldalakon:**

## FORMULA ONE

Kezdőlap

Történet

"Leg"-ek

Átlagsebességek

## Érdekességek a Forma-1 világából

### Története

#### A kezdetek:

A Forma-1 gyökerei egészen a 20. század első éveig vezethetők vissza, az autóversenyzés bölcsőjébe, Franciaországba. Az első nagydíjat, grand prix-t 1906-ban rendezték, a Renault-t vezető győztes pedig nem volt más, mint a magyar Szisz Ferenc.

#### A Forma-1 születése:

Már a 30-as években felmerült a grand prix-k világbajnokságba való szervezése, de a második világháború közbeszólt, így az európai autóversenyzés csak a 40-es évek végén éledt újra. Az első bajnoki idényt pedig 1950-ben rendezték, a legelső futam a silverstone-i Brit Nagydíj volt.

#### A Forma-1 aranykora:

Az F1-be 1977-ben belépő Renault hozta a sportág következő nagy újítását. A francia gyártó turbómotorjai kezdetben ugyan siralmasan megbízhatatlanok voltak, de a technológia elterjedt és egy évtizedig uralta a száguldó cirkuszt, az erőforrások az egészen elképesztő, 1000 lóerő feletti teljesítményt is hozták. Manapság talán az 1980-as évek tekintenek legtöbbször aranykorként, amikor Nelson Piquet, Alain Prost, Nigel Mansell, Ayrton Senna küzdött a Brabham, a McLaren, a Lotus és a Williams autókban ülve.

#### A '90-es évek:

A Forma-1-ben tulajdonképpen csak Mansell '92-es révbe érése, Prost '93-as visszavonulása és Senna '94-es halála zárta le végleg a 80-as éveket, ezután emelkedtek fel a korszak új sztárjai.

#### Napjaink:

A Ferrari és Schumacher totálisan uralta a 2000-es évtized elejét, öt éven keresztül minden bajnoki címet begyűjtött. A Renault és Fernando Alonso 2005-2006-ban törte meg a sorozatot, a főszerep ezután már a következő generációé, napjaink sztárjaié lett.

### FORMULA ONE

- Kezdőlap
- Történet
- "Leg"-ek
- Átlagsebességek

#### Legtöbb nagydíj



1. Rubens Barrichello (326)
2. Kimi Räikkönen (315)
3. Fernando Alonso (314)
4. Jenson Button (309)
5. Michael Schumacher (308)
6. Felipe Massa (272)

#### Legtöbb megnyert futam



1. Michael Schumacher (91)
2. Lewis Hamilton (84)
3. Sebastian Vettel (53)
4. Alain Prost (51)
5. Ayrton Senna (41)
6. Fernando Alonso (32)

#### Bajnoki címek




1. Michael Schumacher (7)
2. Lewis Hamilton (6)
3. Juan Manuel Fangio (5)
4. Alain Prost (4)
- Sebastian Vettel (4)

#### Csapat bajnoki címek




1. Ferrari (16)
2. Williams (9)
3. McLaren (8)
4. Lotus (7)
5. Mercedes (6)
6. Red Bull (4)

#### Legfiatalabb versenyző



Max Verstappen 2015-ben 17 éves és 166 napos korában indult az Osztrák nagydíjon.

#### Legidősebb versenyző



Louis Chiron 58 évesen és 288 naposan állt rajthoz 1958-ban Monacóban.

#### Legtöbb Grand Prix



1950 óta a legtöbb versenyt az olaszországi Monzában rendezték.

### FORMULA ONE

- Kezdőlap
- Történet
- "Leg"-ek
- Átlagsebességek

## Átlagsebesség számítás

CANADIAN

FRENCH

AUSTRIAN

BELGIAN

ITALIAN

SINGAPORE

MEXICAN

BRAZILIAN

ABU DHABI

Pálya

1 kör ideje (másodpercben)

pl.: 105.954

Átlagsebesség

### F1 Circuits - Season 2019

Források: F1 történet, Wikipedia

## Átlagsebesség számítás

### HUNGARIAN GRAND PRIX



Hungaroring

Sector 1

Sector 2

Sector 3

Number of laps  
**70**

Circuit length  
**4.381 km**

Pálya  
Hungaroring

1 kör ideje (másodpercben)  
108,333

Átlagsebesség  
145.58444795214754 km/h



A második feladatrészen a Forma-1-es szezon 2019-es eredményeit tartalmazó adatbázissal kell dolgoznia! Az adatbázis a következő táblákat tartalmazza:

**pilotak**

id	Egész szám, a pilóta rekord azonosítója (PK)
rajtszam	Egész szám, a pilóta rajtszáma
vezeteknev	Szöveg, a pilóta vezetékeve
keresztnev	Szöveg, a pilóta keresztnéve
szulesidatum	Dátum, a pilóta születési dátuma
rovidites	Szöveg, rövidítés a pilóta azonosítására
csapatnev	Szöveg, annak a csapatnak (vagy csapatoknak) a neve, ahol a 2019-es szezonban a pilóta versenyzett

**nagydijak**

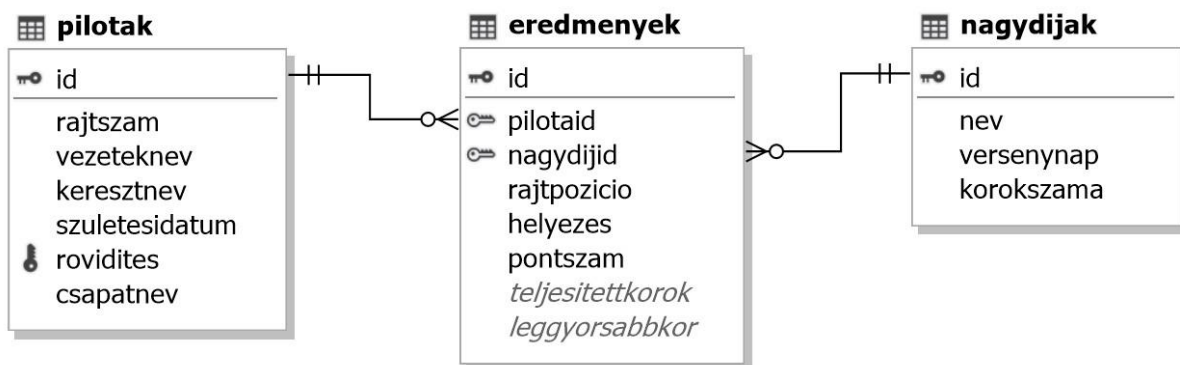
id	Egész szám, a nagydíj azonosítója (PK)
nev	Szöveg, a nagydíj neve angolul
versenynap	Dátum, a verseny megrendezésének napja
korokszama	Egész, a verseny során teljesítendő körök száma

**eredmenyek**

id	Egész szám, a rekord azonosítója (PK)
pilotaid	Egész szám, a pilóta azonosítója (FK)
nagydijid	Egész szám, a nagydíj azonosítója (FK)
rajtpozicio	Egész szám, a pilóta hányadik rajtrácsról indult a nagydíjon
helyezés	Egész szám, a pilóta helyezése a nagydíjon
pontszam	Egész szám, a pilóta hány bajnoki pontot kapott ezen a nagydíjon
teljesitettkorok	Egész szám, a pilóta által teljesített körök száma a nagydíjon
leggyorsabbkor	Szöveg, a pilóta leggyorsabb körének ideje a nagydíjon

Az elsődleges kulcsok PK-val, az idegenkulcsok FK-val lettek jelölve.

Az adattáblák közti kapcsolatokat az alábbi ábra mutatja:



A feladatok megoldására elkészített SQL parancsokat a `megoldasok.sql` állományba illessze be a feladatok végén zárójelben jelölt sor alá! **A javítás során csak ennek az állománynak a tartalma lesz értékelve!** Ügyeljen arra, hogy a lekérdezésben pontosan a kívánt mezők és mezőnevek szerepeljenek, és felesleges mezőt ne jelenítsen meg!

1. Hozzon létre a lokális SQL serveren `formula1` néven adatbázist! Az adatbázis karakterkódolását állítsa be UTF-8-ra! (1. feladat)
2. Az `adatok.sql` állomány tartalmazza a táblákat létrehozó és az adatokat a táblába beszűrő SQL parancsokat! Futtassa az `adatok.sql` parancsfájlt a `formula1` adatbázisban!
3. A magyar nagydíj („Hungarian Grand Prix”) versenyköreinek száma hibásan került az adatbázisba, javítsa ki a helyes értékre! A helyes érték: 70. (3. feladat)
4. Listázza ki az idényben részt vevő pilóták vezetéknevét, rajtszámát, csapatnevét és hogy a 2019-es idényben hány éves volt (2019 mínusz a születési év)! A számított mező címkéje legyen „életkor”. Az eredményt rendezze életkor szerint csökkenő sorrendbe! (4. feladat)

vezeteknev	rajtszam	csapatnev	eletkor
Räikkönen	7	Alfa Romeo	40
Kubica	88	Williams	35
Hamilton	44	Mercedes	34
Grosjean	8	Haas F1 Team	33
Vettel	5	Ferrari	32
Hülkenberg	27	Renault	32
...			

5. Listázza ki az egyes nagydíjak első helyezettjeit! A listában jelenítse meg a nagydíj nevét, a verseny dátumát és a győztes pilóta vezeték- és keresztnévét! A listát rendezze a verseny napja szerint növekvő sorrendbe! (5. feladat)

nev	vezeteknev	keresztnev	versenynap
Australian Grand Prix	Bottas	Valtteri	2019.03.17 0:00:00
Bahrain Grand Prix	Leclerc	Charles	2019.03.31 0:00:00
Chinese Grand Prix	Gasly	Pierre	2019.04.14 0:00:00
Azerbaijan Grand Prix	Leclerc	Charles	2019.04.28 0:00:00
Spanish Grand Prix	Hamilton	Lewis	2019.05.12 0:00:00
Monaco Grand Prix	Gasly	Pierre	2019.05.26 0:00:00
...			

6. Listázza ki a 2019-es bajnoki szezon első három helyezettjét! A listában szerepeljen a pilóta teljes neve, a csapatának a neve, valamint a versenyek során elért összpontszáma! (6. feladat)

nev	csapatnev	osszpontszam
Lewis Hamilton	Mercedes	413
Valtteri Bottas	Mercedes	326
Max Verstappen	Red Bull	278