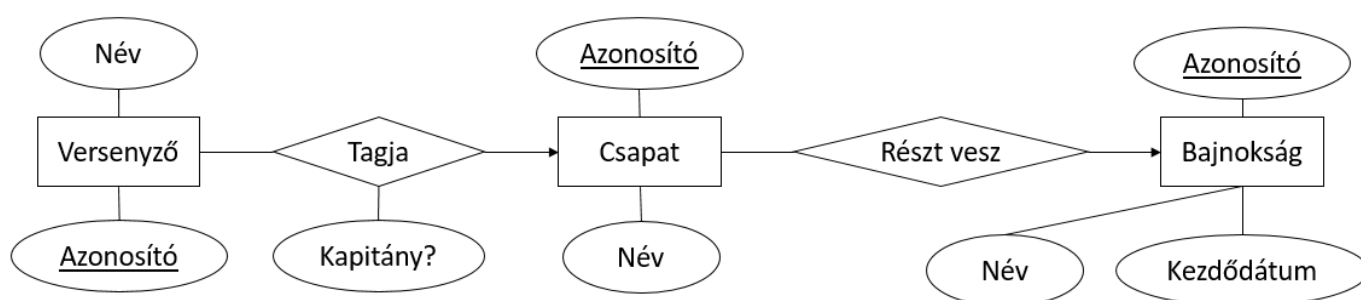


Informatika 2 vizsga 2018.05.29.	
A nevet NYOMTATOTT, NAGYBETŰVEL írja, nem aláírást kérünk. A rendelkezésre álló idő 90 perc. Az elégséges szint 45% (31 pont), de <u>feladatcsoportonként 20% elérése szükséges.</u>	
NÉV:	NEPTUN:

Adatbázisok		Hálózatok		Nyelvek	
1 [13]		4 [8]		7 [4]	
2 [15]		5 [6]		8 [4]	
		6 [10]		9 [10]	
Σ [28]		Σ [24]		Σ [18]	
Σ [70]				ZH+HF:	
Összesen:				Jegy:	

1. Feladat Adott az alábbi ábrán látható Entitás-Relációs diagram, amelyhez a következő kiegészítések tartoznak: (i) A versenyzők, csapatok, bajnokságok nevét kötelező megadni. (ii) Két azonos nevű csapat nem megengedett, de azonos nevű versenyzők és bajnokságok lehetnek. (iii) A tagja reláció rendelkezik egy „Kapitány?” attribútummal, amely vagy igaz, vagy hamis értékű lehet, NULL érték nem megengedett. Az alapértelmezett érték legyen a hamis. Írjon SQL utasításokat, amelyek a diagramnak és a kiegészítéseknek megfelelő táblákat létrehozzák! **[13 pont]**



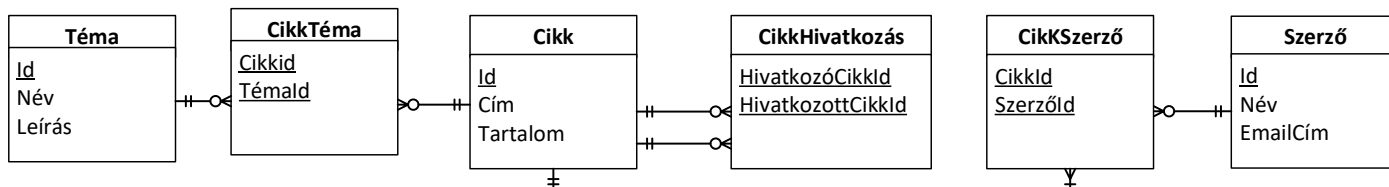
```

CREATE TABLE Bajnokság(
    Név NVARCHAR(100) NOT NULL,
    Id INT PRIMARY KEY,
    Kezdődátum Date;
);

CREATE TABLE Csapat(
    Név NVARCHAR(100) NOT NULL UNIQUE,
    Id INT PRIMARY KEY,
    BajnokságId INT,
    FOREIGN KEY (BajnokságId) REFERENCES Bajnokság(Id)
);

CREATE TABLE Versenyző (
    Név NVARCHAR(100) NOT NULL,
    Id INT PRIMARY KEY,
    CsapatId INT,
    Kapitany BIT DEFAULT 0 NOT NULL,
    FOREIGN KEY (CsapatId) REFERENCES Csapat(Id)
);
  
```

Adott az alábbi ábrán látható adatbázis séma, amellyel különböző témákban írt cikkeket és adataikat tároljuk.



2. Feladat Kik azok a szerzők, akik már legalább 5 cikket írtak? [6 pont]

```

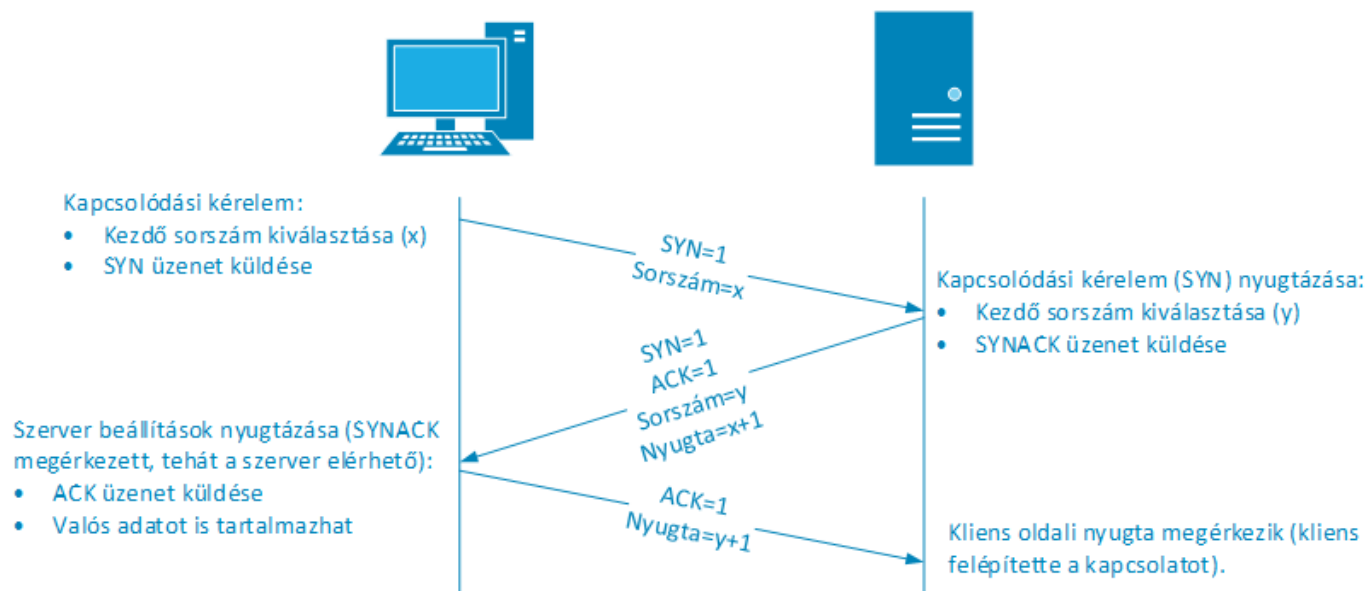
SELECT sz.Id, sz.Név -- 1 pont
FROM Szerző sz -- 1 pont
    INNER JOIN CikkSzerző csz ON sz.Id = csz.SzerzőId -- 1 pont
    INNER JOIN Cikk c ON csz.CikkId = c.Id -- 1 pont
GROUP BY sz.Id -- 1 pont
HAVING COUNT(c.Id) > 5 -- 1 pont
    
```

Kik azok a szerzők, akiknek legalább 5 „orvostudomány” témájú cikkét hivatkozták? [9 pont]

```

SELECT sz.Id, sz.Név -- 1 pont
FROM Szerző sz -- 1 pont
    INNER JOIN CikkSzerző csz on sz.Id = csz.SzerzőId -- 1 pont
    INNER JOIN CikkHivatkozás ch ON ch.HivatkozottCikkId = csz.CikkId -- 1 pont
    INNER JOIN CikkTéma ct ON ct.CikkId = csz.CikkId -- 1 pont
    INNER JOIN Téma t ON t.Id = ct.TémaId -- 1 pont
WHERE t.Név = 'Orvostudomány' -- 1 pont
GROUP BY sz.Id -- 1 pont
HAVING COUNT(csz.HivatkozottCikkId) > 5 -- 1 pont
    
```

3. Feladat Ismertesse a TCP kapcsolat felépítésének forgatókönyvét! (Üzenetek típusa, sorrendje, jelzőbitek, sorszámok.) [8 pont]



4. Feladat Adott az alább látható 16 bites adat. Alkalmazzon rajta 2 dimenziós paritásellenőrzést: számítsa ki és írja le a szükséges paritásbiteket! Magyarázza el, hogyan kell képezni a paritásbiteket! Tegyük fel, hogy az átvitel során a 7. bit elromlik (1 helyett, 0 érkezik meg). Mutassa meg és magyarázza el, hogyan tudja ezt a hibát kezelni a 2 dimenziós paritásellenőrzés! [6 pont]

0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0

- $$\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \end{array} \quad [1 \text{ pont}]$$

$$\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \end{array}$$

Verzió	Fejrész hossz	Szolgáltatás típus	Datagram hossza	
16 bites azonosító			Jelző- bitek	Darabolási eltolás
TTL	Felső réteg protokollja		Fejrész ellenőrző összege	
32 bites forrás IP cím				
32 bites cél IP cím				
Opciók				
Változó hosszúságú adatok (pl. TCP, UDP szegmens)				

- Protokoll verziószáma: az IP jelenleg leggyakrabban használt változata a 4-es. **[1 pont]**
- Fejrész hossza: erre azért van szükség, mert a fejrész hossza nem állandó és tudni kell, hogy honnan kezdődik a valós adata. **[1 pont]**
- Datagram hossza (a teljes datagram bájtainak a száma): erre is szükség van, hiszen az adatrész is változó hosszúságú lehet. **[1 pont]**
- A szolgáltatás típusa mezővel lehet jelezni, hogy milyen jellegű adatot szállít a csomag. Például megadható, hogy a csomag egy valós idejű (real-time) datagram. **[1 pont]**
- Jelzőbitek: különböző beállítások jelzésére **[1 pont]**
- Az IP datagramok rendelkeznek saját 16 bites azonosítóval. A darabolási eltolás egy 13 bites szám, azért szükséges, mert szükség lehet egy nagyobb szállítási rétegbeli szegmens feldarabolására. **[1 pont]**
- TTL (Time to Live) érték: a TTL egy egész szám. Minden útválasztó 1-gyel csökkenti a TTL értékét, amikor továbbítja a datagramot. Amikor a TTL értéke eléri a 0-t, akkor a datagram nem kerül továbbításra, az azt éppen kezelő útválasztó eldobja. **[1 pont]**
- A felső rétegbeli protokoll az a szállítási rétegbeli protokoll, aminek továbbítani kell az üzenetet, tehát ez tipikusan TCP vagy UDP lehet. **[1 pont]**
- Forrás és cél IP cím: értelemszerűen azonosítani kell a feladót és a címzettet. **[2 pont]**
- Opcionális részek lehet például az időbélyeg **[1 pont]**
- Adat **[1 pont]**

7. Feladat Írjon egy reguláris kifejezést, amely ellenőrzi egy dátum helyességét. A dátum 1900-01-01 és 2099-12-31 közötti érték lehet. Az év 1900 és 2099 közötti, a hónapok 01 és 12 közötti, a napok pedig 01 és 31 közötti értéket vehetnek fel. Az évet, a hónapot és napot a ‘-’ jelen kívül még elválaszthatja szóköz és per jel (/). **[4 pont]**

$^{(19|20)[0-9][0-9](-|/)(0[1-9]|1[012])(-|/)(0[1-9]|12)[0-9]|3[01])\$}$

- $^{\$}$ – 1 pont
- Minden év – 1 pont
- Hónap, nap – 1 pont
- Elválasztás – 1 pont

8. Feladat Mire szolgál a munkamenet (session)? PHP-ben hogyan oldható meg a munkamenet kezelése? Milyen változók és függvények használhatók a munkamenet kezelésénél? **[4 pont]**

A http állapotmentes protokoll. A http kérések egymástól függetlenek. A munkamenettel (session) a http kérések között kapcsolat teremthető. Protokoll szinten a kapcsolat cookie-k segítségével valósul meg. Az alábbi PHP programrészlet illusztrálja a munkamenet kezelésnél használható változókat és függvényeket: **[1 pont]**

```
<?php
// Start the session
session_start();           // 1 pont
?>
<!DOCTYPE html>
<html> <body>
...
<?php
// Set session variable - $_SESSION is an associative array
$_SESSION["favcolor"] = "green";    // 1 pont
?>
...
<?php
// Echo session variable that was set on previous page
echo "Kedvenc szín: " . $_SESSION["favcolor"] . "<br>";
print_r($_SESSION);
?>
...
<?php
// destroy the session
session_destroy();    // 1 pont
?>
...
</body> </html>
```

9. Feladat Egy nyelv mondatai a és b karakterekből állnak. A mondatok első karaktere a , az utolsó pedig b , továbbá a mondatban található karakterek száma páros. Adja meg a nyelvnek a leírását reguláris kifejezéssel, automatával és nyelvtannal, továbbá elemezze az „abab” szöveget az elkészített automatával. **[10 pont]**

Regexp **[2 pont]**:

$$r = a((a + b)(a + b))^*b$$

Automata: **[3 pont]**

	a	b
→S	A	

A	B	B,C
B	A	A
<u>C</u>		

Nyelvtan: [3 pont]

$S \rightarrow aA$
 $A \rightarrow aB / bB / b$
 $B \rightarrow aA / bA$

Elemzés: [2 pont]

Feldolgozandó szöveg	Az automata állapota
<i>abab</i>	S
<i>bab</i>	A
<u><i>ab</i></u>	B,C
<u><i>b</i></u>	A
ϵ	B, <u>C</u> - Elfogadva