

2016.06.15.

A nevet NYOMTATOTT, NAGYBETŰVEL írja, nem aláírást kérünk.

A rendelkezésre álló idő 90 perc.

Az elégséges szint 45% (31 pont), de feladatcsoportonként 20% elérése szükséges.

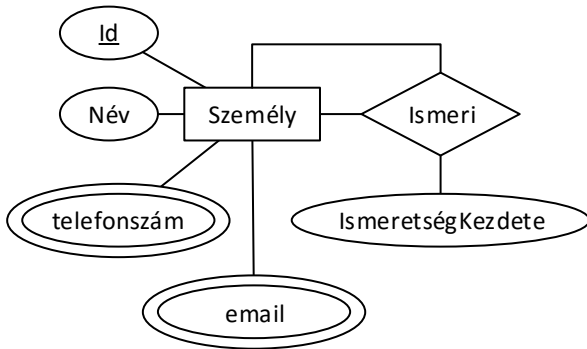
NÉV:

NEPTUN:

Adatbázisok		Hálózatok		Nyelvek	
1 [9]		3 [10]		6 [5]	
2 [16]		4 [4]		7 [6]	
		5 [8]		8 [12]	
Σ [25]		Σ [22]		Σ [23]	
Σ [70]				ZH+HF:	
Összesen:				Jegy:	

1. Feladat Adott az alábbi ábrán látható Entitás-Relációs diagram.

Írjon SQL utasításokat, amelyek az ennek megfelelő táblákat létrehozzák! **[9 pont]**



```
create table Személy (
    id int primary key,
    név nvarchar(100) not null,
);
create table telefonszám (
    szám nvarchar(20) not null,
    személyId int,
    foreign key (személyId) references Személy(Id)
);
create table Email (
    cím nvarchar(20) not null,
    személyId int,
    foreign key (személyId) references Személy(Id)
);
create table Ismeri(
    a int,
    b int,
    ismeretségKezdetek datetime,
    primary key(a,b),
    foreign key (a) references Személy(Id),
    foreign key (b) references Személy(Id)
);
```

2. Feladat Adott az alábbi ábrán látható adatbázis séma.

```

    erDiagram
        Hallgató ||--o{ Hallgató_Tárgy : "1:M"
        Hallgató_Tárgy ||--o{ Tárgy : "1:M"
        Tárgy ||--o{ Tárgy_Oktató : "1:M"
        Tárgy_Oktató ||--o{ Oktató : "1:M"
        Oktató ||--o{ Tanszék : "1:M"
  
```

The diagram illustrates the following entities and their attributes:

- Hallgató**: NEPTUN (PK), Név, TelefonSzám
- Hallgató_Tárgy**: HallgatóNEPTUN (FK), TárgyKÓD (FK), Félév, Jegy, Pontszám
- Tárgy**: KÓD (PK), Név, TárgyfelelősNEPTUN
- Tárgy_Oktató**: TárgyKÓD (FK), OktatóNEPTUN (FK)
- Oktató**: NEPTUN (PK), Név, TanszékId (FK), Fizetés
- Tanszék**: Id (PK), Név

The relationships are as follows:

- Hallgató** to **Hallgató_Tárgy**: 1:M relationship.
- Hallgató_Tárgy** to **Tárgy**: 1:M relationship.
- Tárgy** to **Tárgy_Oktató**: 1:M relationship.
- Tárgy_Oktató** to **Oktató**: 1:M relationship.
- Oktató** to **Tanszék**: 1:M relationship.

Írjon SQL lekérdezést, amely kilistázza, hogy az „Automatizálási Tanszék” egyes tárgyait az egyes félévekben hány hallgatója volt! Az eredményben a tárgy neve, a félév neve és a hallgatók száma szerepeljen. Egy tárgy akkor tartozik egy tanszékhez, ha a *tárgyfelelőse* az adott tanszéken dolgozik. [5 pont]

```
select t.Név, ht.Felev, count(ht.HallgatóNEPTUN)           // 1 pont
from Tárgy t
      left outer join Hallgató Tárgy ht on ht.TárgyKÓD = t.Kód // 1 pont
```

```

inner join Oktató o on o.NEPTUN = t.TárgyfelelősNEPTUN
inner join Tanszék tsz on tsz.Id = o.TanszékId           // 1 pont
where tsz.Név = "Automatizálási tanszék"                // 1 pont
group by t.KÓD, ht.Félév                                // 1 pont

```

Listázza ki azoknak a tanszékeknek a nevét, ahol legfeljebb 5-en dolgoznak! (Figyelem, az is előfordulhat, hogy egy tanszéknek egyetlen oktatója sincsen!) Az eredményben csak a tanszékek neve szerepeljen! **[6 pont]**

```

select tsz.Név                                           // 1 pont
from Tanszék tsz
    left outer join Oktató o on o.TanszékId = tsz.Id    // 2 pont
group by tsz.Id                                         // 1 pont
having count(o.*) <= 5                                  // 2 pont

```

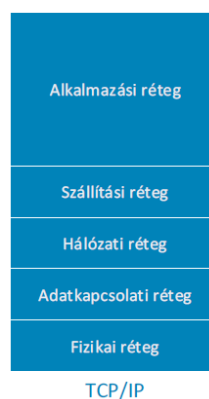
Listázza ki, hogy az egyes hallgatók milyen tanulmányi átlagot értek el a „2015/16”-os félévben! A tanulmányi átlag a félévben szerzett jegyek átlaga. **[5 pont]**

```

select h.Név, avg(ht.Jegy)                               // 2 pont
from Hallgató h
    inner join Hallgató_Tárgy ht on ht.HallgatóNEPTUN = h.NEPTUN // 1 pont
where ht.Félév = '2015/16'                               // 1 pont
group by h.NEPTUN                                         // 1 pont

```

3. Feladat Ismertesse az Internet működését leíró rétegzett architektúrát! Milyen rétegek vannak és mi az egyes rétegek feladata? **[10 pont]**



- Az alkalmazási réteg feladata, hogy az elosztott szoftver komponensek közötti kommunikációt biztosítsa, vagyis ennek a segítségével a különböző alkalmazások tudnak saját protokollokat kialakítani.
- A szállítási réteg feladata, hogy az alkalmazási rétegtől megkapott üzenetet elszállítsa a címzethez. A címzett ebben az esetben egy távoli csomóponton futó alkalmazás. Az egyes számítógépeken az alkalmazásokat az egyes futó folyamatok (processzek) azonosítják.
- A hálózati réteg feladata egy csomag eljuttatása a küldőtől a címzethez. Itt a küldő és a címzett is egy-egy hálózati végrendszert jelent.
- Az adatkapcsolati réteg feladata, hogy továbbítsa az adatot egy szomszédos csomópontnak a megfelelő adatkapcsolaton keresztül, tehát a megfelelő kommunikációs csatornán.
- A fizikai réteg feladata, hogy egy megkapott bitsorozat egyes bitjeit valóban átküldje a megfelelő fizikai összeköttetésen.

[5 x (1+1) pont]

4. Feladat Adott az alábbi ábrán látható 16 bites adat, amelyet 4 sorba és 4 oszlopba rendeztünk. Alkalmazzon rajta 2 dimenziós paritásellenőrzést: írja be a szükséges paritásbiteket! Írja le, hogyan kell képezni a paritásbiteket! **[4 pont]**

```

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

```

A **kétdimenziós paritásellenőrzés** esetén az adatbitek egy mátrixba rendezzük és minden egyes sorhoz, illetve oszlophoz paritásbitek számítunk. [2 pont] Akkor 0 a paritásbit, ha az adatban lévő 1-es bitek száma páros, egyébként páratlan. [1 pont]

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

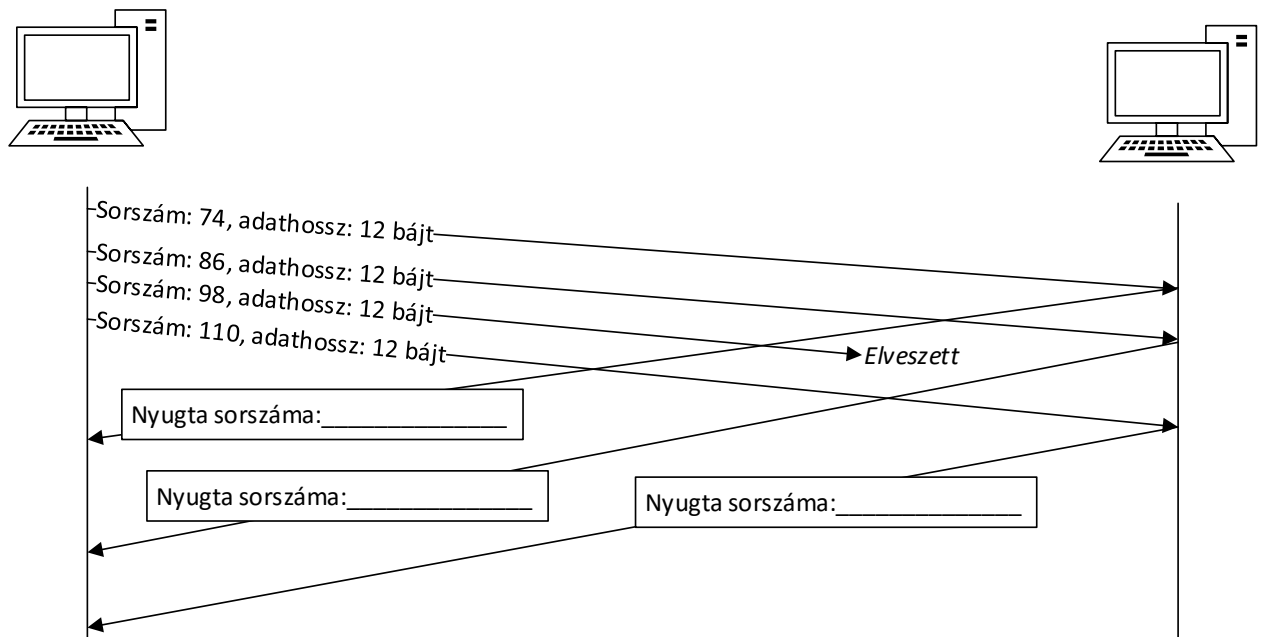
[1 pont]

5. Feladat Az alábbi kérdések a TCP szállítási rétegbeli protokollra vonatkoznak:

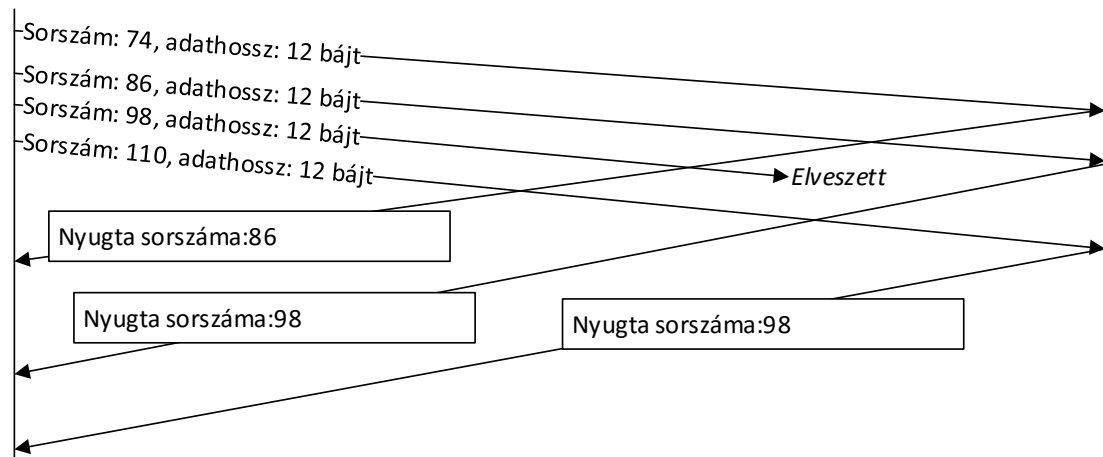
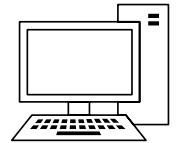
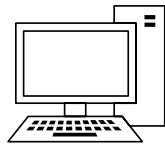
Hogyan képezzük a csomagok sorszámát? [2 pont]

- Az elküldendő üzenet leírható egy bájtsorozatként. A szegmensek a bájtsorozat egyes részeit küldik el. Egy szegmens sorszáma az első bájt sorozatbeli indexe.
- A számozás azonban nem 0-tól, hanem egy véletlen számtól indul.

Az alábbi ábrán két számítógép TCP segítségével kommunikál. A küldő 4 szegmenst küld, amiből egy elveszik. A vevő nyugtákat küld vissza. Írja be a nyugta csomagokba a nyugták számát. Válaszát indokolja! [6 pont]



Megoldás:



- 86: a következőnek várt adat kezdősorszáma. Mivel a 74. bájtól 12 bájt érkezett, ez 86 lesz. [2 pont]
- 98: a következőnek várt adat kezdősorszáma (86+12) [2 pont]
- 98: a nyugták kumuláltak. Egy nyugta azt jelzi, hogy a következőnek várt bájtig minden megérkezett. Mivel a 98-110 bájtok hiányoznak, ezért csak a 98-ig lehet nyugtázni. [2 pont]

6. Feladat Mire szolgál a munkamenet (session)? [1 pont]

A HTTP állapotmentes protokoll. A HTTP kérésék egymástól függetlenek. A munkamenettel (session) a HTTP kérések között kapcsolat teremthető. Protokoll szinten a kapcsolat cookie-k segítségével valósul meg.

PHP-ben hogyan oldható meg a munkamenet kezelése? Milyen változók és függvények használhatók a munkamenet kezelésénél? Hogyan lehet változókat tárolni és kiolvasni a munkamenetből? [4 pont]

Az alábbi PHP programrészlet illusztrálja a munkamenet kezelésnél használható változókat és függvényeket:

```
<?php
// Start the session
session_start(); //munkamenet elindítása a fájl elején [1 pont]
?>
<!DOCTYPE html>
<html> <body>
...
<?php
// Set session variable - $_SESSION is an associative array
$_SESSION["favcolor"] = "green"; // változó eltárolása a munkamenetben [2 pont]
?>
...
<?php
// Echo session variable that was set on previous page
echo "Kedvenc szín: " . $_SESSION["favcolor"] . "<br>";
print_r($_SESSION);
?>
...
<?php
// remove all session variables
session_unset(); // változók eltávolítása a munkamenetből - ha már nem kellene

// destroy the session
session_destroy(); // munkamenet megszüntetése [1 pont]
//csak akkor kell, ha már nincsen rá szükség többet

?>
```

...
</body> </html>

7. Feladat Írjon reguláris kifejezést, amely ellenőrzi, hogy a kérdéses szöveg megfelel-e egy éééé-hh-nn formában megadott dátumnak. Az év vagy 19-cel vagy 20-szal kezdődik. Példa helyesen írt dátumra: 2015-05-12. A hónapok értéke 01 és 12, a napok értéke pedig 01 és 31 közötti érték lehet. [6 pont]

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

- ^\$: 1 pont
- -: 1 pont
- éééé: 2 pont
- hh: 1 pont
- nn: 1 pont

8. Feladat Ismertesse a veremautomata matematikai leírását és működését! Készítsen veremautomatát az $a^i b a^i$ ($i > 0$) szövegek elemzéséhez. Adja meg a veremautomata működését az *aabaa* szöveg esetén! [12 pont]

A veremautomata (angol terminológiában pushdown automaton) formális leírására egy hetes szolgál: $P(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, ahol:

- Q az automata lehetséges állapotainak véges halmaza,
- Σ az elemzendő jelsorozatok karakterkészlete,
- Γ a verem karakterkészlete,
- δ az automata mozgási szabályainak halmaza,
- q_0 a kezdő állapot,
- Z_0 a verem kezdeti tartalma és
- F az elfogadó állapotok halmaza.

[3 pont]

A veremautomata mozgását az automata állapota, a bemenetről olvasott és a verem tetején található szimbólum alkotta hármasság határozza meg. A mozgás hatására az automata új (lehet az eredetivel megegyező is) állapotba kerül, és egy jelsorozatot ír a verem tetejére. A mozgási szabályok az alábbi formájúak: $\delta(q, a, b) = (q', \alpha)$, ahol:

- $q \in Q$ – az automata aktuális állapota,
- $a \in \Sigma$ – az inputról beolvasott karakter,
- $b \in \Gamma$ – a verem tetején lévő elem, amely a kiolvasáskor eltűnik a veremből,
- $q' \in Q$ – ebbe az állapotba kerül az automata,
- $\alpha \in \Gamma^*$ – ezt a karaktersorozatot írja a verembe az automata. (Lehet nulla hosszú is. A karaktersorozat legbaloldalibb szimbóluma kerül a verem tetejére.)

[2 pont]

A veremautomatának lehet egy másik mozgási lehetősége is, ez az ε mozgás. Ebben az esetben a mozgást az automata aktuális állapota és a verem tetején lévő szimbólum szabja meg. Az ε mozgás az alábbi módon írható le: $\delta(q, \varepsilon, b) = (q', \alpha)$.

Egy szöveg akkor kerül mondatként elfogadásra, ha az automata beolvasta a szöveget, és elfogadó állapotba került, vagy további ε mozgásokkal elfogadó állapotba kerül. A veremautomatáknál az elfogadásnak nem feltétele a mozgásképtelenség. Az is elfogadható, ha az ε szabályok ciklikusságot idéznek elő, de közben elfogadó állapotban is áthalad az automata. Az elfogadó állapotban megkövetelhetjük, hogy a verem üres legyen.

A $a^i b a^i$ ($i > 0$) mondatokat felismerő veremautomata esetén:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, \#\}$
- Kezdő állapot: q_0
- A verem tartalma indulásnál: $Z_0 = \#$
- Elfogadó állapot: $F = \{q_3\}$

- Mozgási szabályok:
 - $\delta(q_0, a, \#) = (q_1, a\#)$
 - $\delta(q_1, a, a) = (q_1, aa)$
 - $\delta(q_1, b, a) = (q_2, a)$
 - $\delta(q_2, a, a) = (q_2, \varepsilon)$
 - $\delta(q_2, \varepsilon, \#) = (q_3, \#)$

[5 pont]

Az automata működése:

<i>bemenet</i>	<i>állapot</i>	<i>verem</i>
<u>aabaa</u>	q_0	<u>#</u>
<u>abaa</u>	q_1	<u>a#</u>
<u>baa</u>	q_1	<u>aa#</u>
<u>aa</u>	q_2	<u>aa#</u>
<u>a</u>	q_2	<u>a#</u>
<u>ε</u>	q_2	<u>#</u>
<u>ε</u>	q_3 Elfogadva	<u>#</u>

[2 pont]