

Informatika 2 dolgozat

2016.05.26.

A csoport

A nevet nyomtatott, nagybetűvel írja, nem aláírást kérünk.

A rendelkezésre álló idő 50 perc

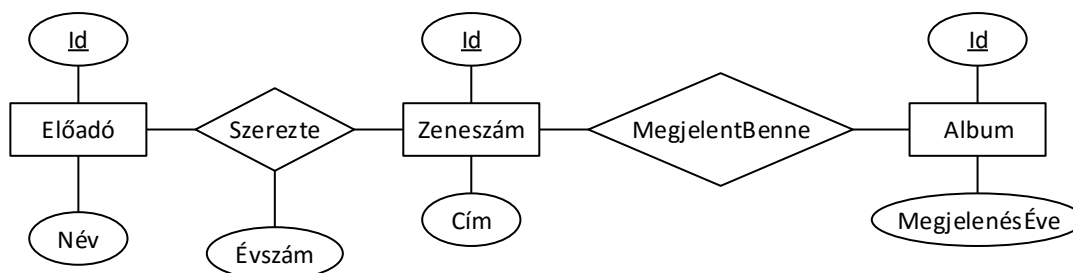
Elégséges szint: 45% (18 pont).

NÉV:

NEPTUN:

Feladat	Pontszám
1 [9]	
2 [5]	
3 [3]	
4 [6]	
5 [3]	
6 [3]	
7 [3]	
8 [8]	
Σ [40]	

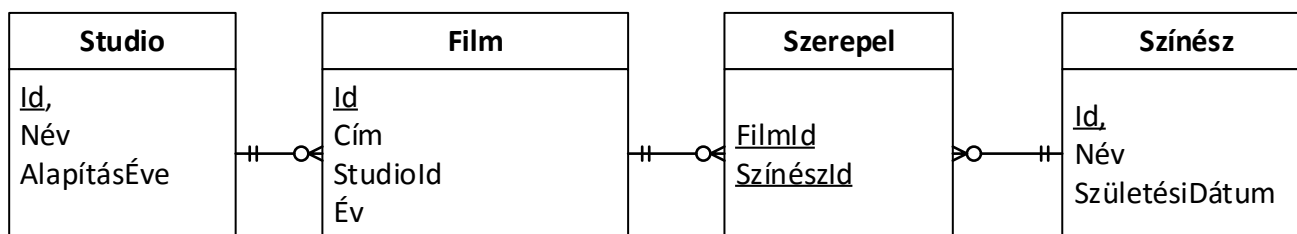
1. Feladat Adott az alábbi ábrán látható Entitás-Relációs diagram. Írjon SQL utasításokat, amelyek az ennek megfelelő táblákat létrehozzák! [9 pont]



```

create table Előadó( // 1 pont
    Id int primary key,
    Nev nvarchar(100)
);
create table Zeneszam( // 1 pont
    Id int primary key,
    Cim nvarchar(100)
);
create table Album( // 1 pont
    Id int primary key,
    MegjelenesEve int
);
create table Szerezte( // 1 pont
    EloadoId int,
    ZeneszamId int,
    Evszam int, // 1 pont
    primary key (EloadoId, ZeneszamId),
    foreign key (EloadoId) references Eloado(Id), // 1 pont
    foreign key (ZeneszamId) references Zeneszam(Id)
);
create table MEgjelentBenne( // 1 pont
    AlbumId int,
    ZeneszamId int,
    primary key (AlbumId, ZeneszamId), // 1 pont
    foreign key (AlbumId) references Album(Id), // 1pont
    foreign key (ZeneszamId) references Zeneszam(Id)
);
  
```

Adott az alábbi ábrán látható adatbázis séma.



2. Feladat Írjon SQL lekérdezést, amely megjeleníti, hogy az egyes filmstúdiók mely színészeket foglalkoztatták már a filmjeikben. A lekérdezés eredményében a stúdió és a színész neve szerepeljen, más ne. Egy stúdió-színész pár csak egyszer szerepeljen az eredményben. [5 pont]

```

select distinct s.név, sz.név
from Stúdió s
      inner join Film f on f.StudioId = s.Id
      inner join szerepel sze on sze.FilmId = f.Id
      inner join szinesz sz on sz.Id = sze.SzineszId
  
```

1 pont
1 pont
1 pont
1 pont
1 pont

3. Feladat Írjon SQL lekérdezést, amely megjeleníti, hogy az egyes színészek összesen hány filmben szerepeltek! [3 pont]

```

select sz.nev, count(filmId)
from szinesz sz
      left outer join szerepel sze on sze.SzineszId = sz.Id
group by sz.id
  
```

1 pont
1 pont
1 pont

4. Feladat Írjon SQL lekérdezést, amely megjeleníti a „20th Century Fox” nevű stúdióban gyártott 2013 utáni filmekben szereplő összes színészt! Az eredmény csak a színészek neveit tartalmazza, mindegyiket pontosan egyszer! [6 pont]

```

select distinct sz.név
from Stúdió s
      inner join Film f on f.StudioId = s.Id
      inner join szerepel sze on sze.FilmId = f.Id
      inner join szinesz sz on sz.Id = sze.SzineszId
where
  s.Név = '20th Century Fox' and
  f.Ev > 2013
  
```

1 pont
1 pont
1 pont
1 pont
1 pont
1 pont

5. Feladat Írjon SQL utasítást, amely frissíti a „20th Century Fox” nevű stúdió alapítási évét 1950-re! [3 pont]

```

update Studio
set AlapitasEve=1950
where nev = '20th Century Fox'
  
```

6. Feladat Mire való a HTTP protokoll? Ismertesse a működését! [3 pont]

- A HTTP egy információátviteli (és kollaborációs) protokoll.
- Kliens-szerver/kérés-válasz elven működik. A kliens elküld egy műveletet egy szervernek megcímezve egy adott erőforrást a szerveren belül. A szerver végrehajtja a műveletet, majd visszaküld egy választ a kliensnek. Tipikusan böngészők és webszerverek között a dokumentumok, egyéb tartalmak átvitelére használjuk.
- Állapotmentes protokoll, két különböző kérés-válasz kommunikáció között alapértelmezetten nem tárolunk információt.

7. Feladat Mi a szállítási réteg feladata? Milyen szolgáltatást nyújt a felsőbb rétegnek? Milyen alsóbb rétegre épül és annak milyen szolgáltatását veszi igénybe? [3 pont]

- A szállítási réteg feladata, hogy a különböző végrendszeren futó alkalmazások folyamatai között logikai összeköttetést biztosítson.
- A szállítási réteg elérésének felülete a socket.
- Gondoskodik az adatátvitelről a feladó folyamat és a címzett folyamat között.
- Az átvitelhez felhasználja az alsóbb hálózati réteg szolgáltatását. A hálózati réteg biztosítja, hogy egy hálózati csomóponttól eljuttatja az üzenetet egy másik hálózati csomópontba. A csomópontokon belül a folyamatok azonosítása már a szállítási réteg feladata.

8. Feladat Ismertesse a socketkezelés kliens- és szerveroldali függvényeit! Írja le, hogy melyiket használjuk kliens és melyiket szerveroldalon (esetleg mindkettőn)! Fejtse ki, hogy melyik függvény mire szolgál! [8 pont]

1. Új socket létrehozása (`socket`). A létrehozás során meg kell adni, hogy milyen szállítási rétegbeli protokollt szeretnénk használni (TCP vagy UDP). **kliens, szerver - 1 pont**
2. A socket hozzárendelése egy hálózati címhez és portszámhoz (`bind`). Itt a programozó feladata, hogy egy még nem használt portszámot megadjon. Ha a portszám foglalt, akkor nem jön létre a socket, hanem hibát kapunk. **szerver - 1 pont**
3. Harmadik lépésben a szerver socketet passzív állapotba helyezzük, felkészítve arra, hogy később beérkező kapcsolatokat fogadjon (`listen`). **szerver – 1 pont**
4. Várakozás kliensek kapcsolódására (`accept`). Amennyiben egy kliens csatlakozik, akkor létrehozunk egy új socketet, amelyen keresztül az újonnan csatlakozott klienssel tudunk kommunikálni. Erre azért van szükség, hogy az eredeti socketet fenntartsuk további csatlakozni akaró kliensek számára. **szerver – 1 pont**
5. Adatok fogadása (`recv`). **kliens, szerver – 1 pont**
6. Adatok küldése (`send`) a socketen keresztül. **kliens, szerver – 1 pont**
7. A socket lezárása (`closesocket`), ha már nincsen rá szükség. Erre azért van szükség, hogy az operációs rendszer erőforrásait ne foglaljuk feleslegesen. **kliens, szerver – 1 pont**
8. A socket segítségével kapcsolódás egy szerverhez (`connect`). A paraméternél meg kell adni, a szerver IP címét és portszámát. **kliens – 1 pont**