

# Kidolgozott példák

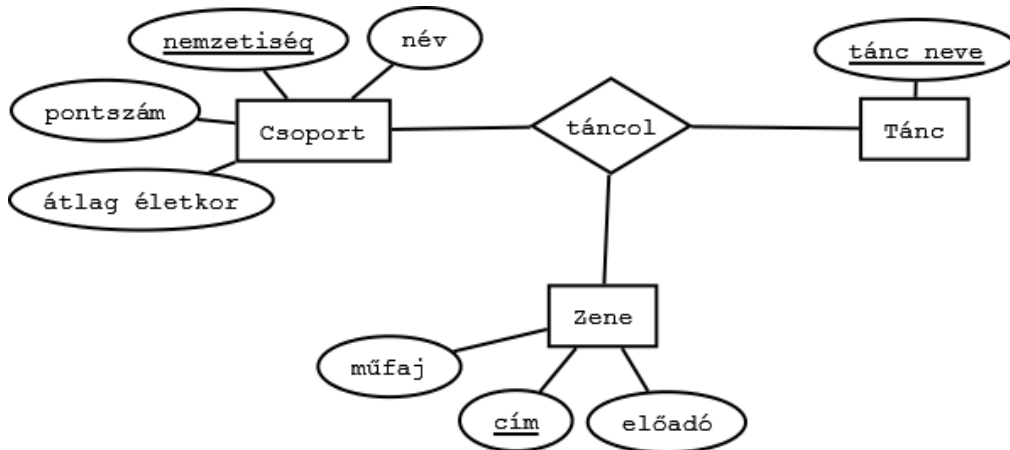
## E-K diagram

### 1.1. példa

Nemzetközi táncverseny:

Feladat: Olyan relációs adatbázist szeretnénk létrehozni, amely a Kék Osztriga Nemzetközi Táncversenyen résztvevő csoportok adatait tartalmazza. Tároljuk a csoport nevét, nemzetiségét, a csoport átlagéletkorát és a verseny folyamán elért pontszámot. Emellett tároljuk a bemutatott tánc nevét, valamint a zenére vonatkozó adatokat, azaz a zene műfaját, a szám címét és előadóját. Feltételezzük, hogy: 1) egyféle táncot több csoport is táncolhat, 2) egy csoport több táncot is előadhat, 3) a versenyen nincs kikötés a táncra vonatkozóan, így bármilyen táncot be lehet mutatni, 4) a zenét egyértelműen azonosítja a címe, azaz nincs két azonos című szám, 5) egy csoportban csak azonos nemzetiségűek táncolnak. A tánc műfaja, valamint a csoportok nemzetisége csak meghatározott értékeket vehet fel.

Megoldás:

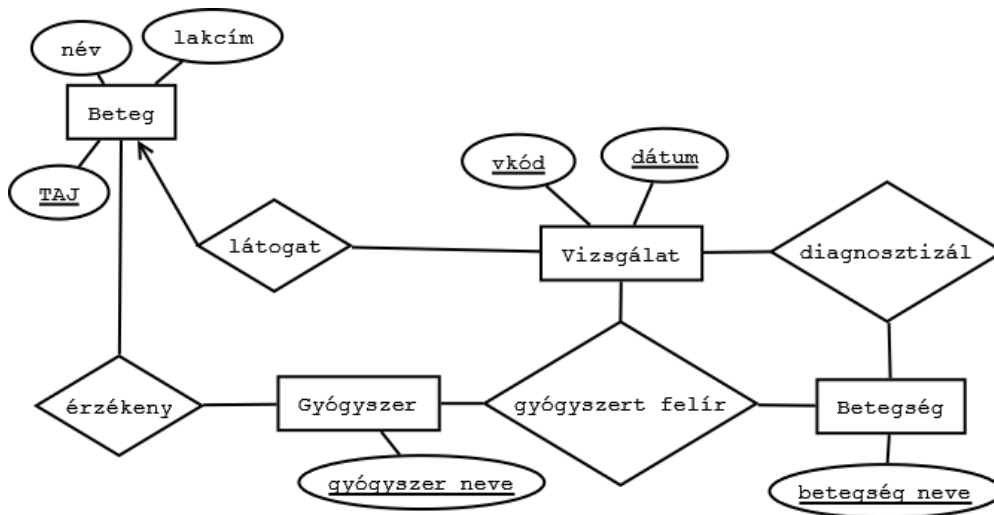


Vegyük fel először az egyedeket és azok attribútumait. A feladatban egyetlen kapcsolatot veszünk fel, ez lesz a *táncol*. A *táncol* kapcsolat összeköti a csoportot, a táncot és a zenét is a feladatban leírtak szerint. Érdekes megjegyezni, hogy a *Tánc* egyednek csak egyetlen attribútuma van. Most a megoldás úgy is helyes lenne, ha a *Tánc* egyed helyett a *tánc neve* attribútum a *táncol* kapcsolat attribútum lenne. Mi a különbség? Ha olyan adatbázist tervezünk, ahol az egyes táncok előre le van tárolva, akkor érdemes külön egyedként feltüntetni. Én ezt a megoldást gondolom jobbnak. Ha a *tánc neve* a *táncol* kapcsolat attribútuma lenne, akkor egy tánc csak akkor kerülne be az adatbázisba, ha azt valaki táncolta.

### 1.2. példa

Készítsünk EK-diagramot egy háziiorvosi betegnyilvántartó rendszerhez! Az orvosok elvárásainak megfelelően az adatbázisnak tartalmaznia kell a betegek személyi adatait, gyógyszerérzékenységüket, az egyes vizsgálatok időpontjait és a felírt gyógyszereket. Feltételezzük, hogy: 1) egy vizsgálaton több betegség is diagnosztizálható, 2) egy betegségre több gyógyszer is felírható, 3) egy-egy páciens több gyógyszerre is lehet érzékeny, 4) egy vizsgálatot a dátum és a vizsgálat kódja határoz meg egyértelműen.

Megoldás:

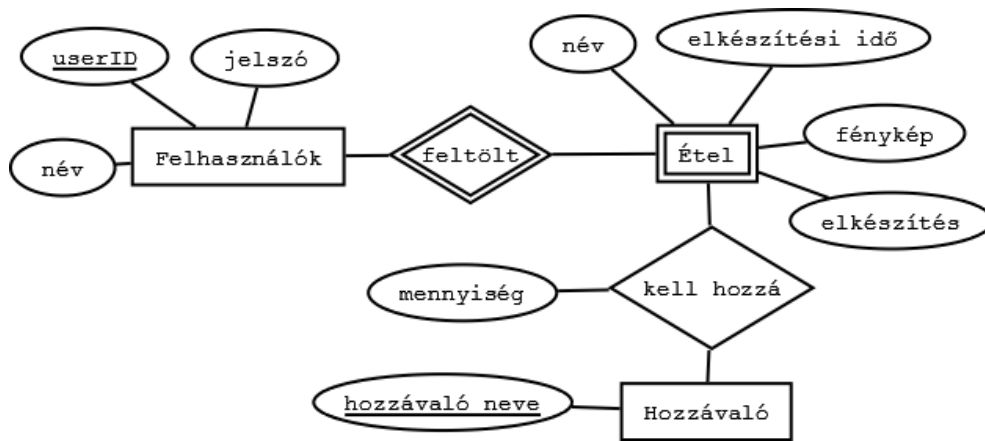


Vegyük fel az egyedeket. A szövegből ki kell szűrni, hogy milyen egyedek lesznek, és köztük milyen kapcsolat lesz. Biztos, hogy kell *Beteg*, *Gyógyszer*, *Vizsgálat* egyed. A *Betegség* nem ennyire nyilvánvaló, ez a szövegben megfogalmazott feltételekből derül ki. A *Beteg* és a *Gyógyszer* egyedek között az *érzékeny* kapcsolat egyértelmű. A *Vizsgálat* és a *Betegség* között is egyértelmű, hogy van egy kapcsolat. A diagnosztizál kapcsolathoz a *Beteget* nem kell hozzávenni, mivel egy vizsgálathoz egy beteg kapcsolódik, tehát elegendő annyit felvenni, hogy a *Vizsgálat* és a *Betegség* között van egy M:N kapcsolat. A *gyógyszert felír* kapcsolat három egyedet köt össze M:N kapcsolattal. Könnyű belátni, hogy egyik kettőt sem elég ezzel a kapcsolattal összekötni.

### 1.3. példa

Online szakácskönyvhöz készíts adatbázis tervet! A szakácskönyvben el kell tárolni az ételek nevét, elkészítésének szöveges leírását, az elkészítési időt, fényképet, valamint a hozzávalókat. A szakácskönyv olyan szempontból interaktív, hogy regisztrált felhasználók is tölthetnek fel receptet. Rajzolja le a szakácskönyv E-K diagramját!

Megoldás:



Egyértelmű, hogy kell *Felhasználó* és *Étel* egyed. Az *Étel* egyedet azonban jelen esetben érdemes gyenge egyedként megjelölni, mivel több felhasználó is feltöltheti ugyanazt az ételt, esetleg más hozzávalókkal, vagy elkészítési javaslattal. A *Hozzávalókat* itt is érdemes külön egyedként felvenni, mivel a hozzávalókat előre el lehet tárolni az adatbázisba, és esetleg a felhasználó csak kiválogatja, hogy mi kell az étel elkészítéséhez.

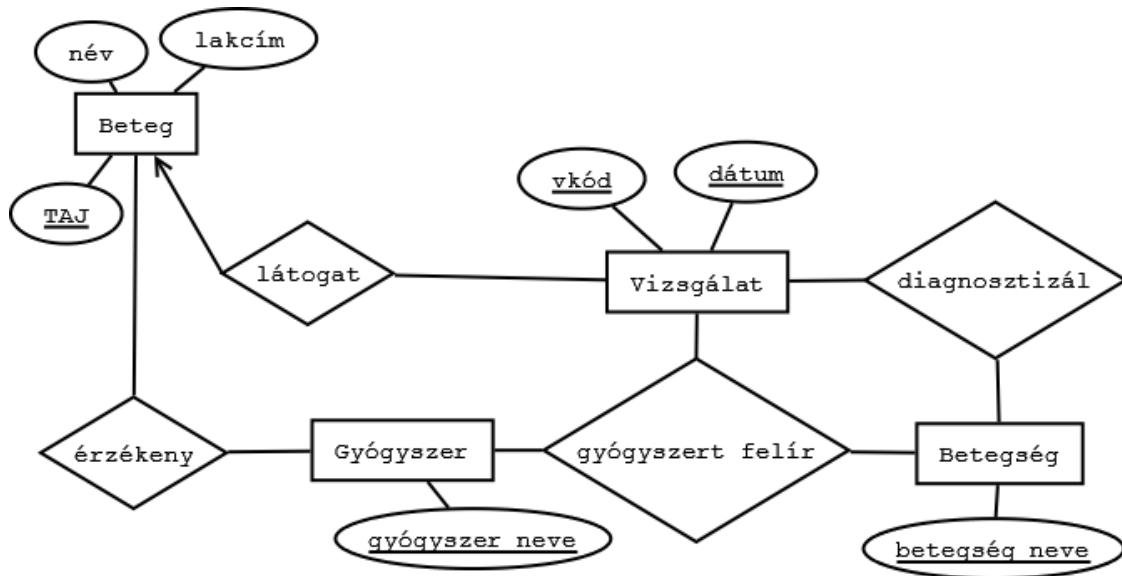
Az is egy jó megoldás lehet, hogy ha az *Étel* nem gyenge egyed, ekkor akár a {név, elkészítés} vagy {név, fénykép} halmaz lehet kulcs. Ha az *Étel* nem gyenge egyed, akkor a feltölt kapcsolat sem meghatározó kapcsolat.

Fontos, hogy ha szerepel egy gyenge egyed az E-K diagramban, akkor biztos, hogy (legalább) egy meghatározó kapcsolat kapcsolódni fog hozzá!

## E-K diagram leképezése relációsémává

### 2.1. példa

Vegyük első példaként az előbbi háziorvosi rendelős feladatot.



Megoldás:

Első lépésként az egyedeket képezzük le. Felvesszük a relációsémákat az egyed nevével, és felsoroljuk az attribútumokat. Ezekben a sémákban a kulcs meg fog egyezni az egyed kulcsával.

BETEG(taj, név, lakcím)

GYÓGYSZER(gyógyszer neve)

BETEGSÉG(betegség neve)

VIZSGÁLAT(vkód, dátum)

Ezután a kapcsolatokat képezzük le sémákká úgy, hogy az attribútumok halmazát a kapcsolódó egyedek kulcsait vesszük a halmazba. (Dőlt betűvel azt jelölöm, hogy egy másik séma külső kulcsa.) Ezt követően ki kell jelölni a kulcsokat a sémákban.

ÉRZÉKENY(taj, gyógyszer neve)

GYÓGYSZERT FELÍR(gyógyszer neve, betegség neve, vkód, dátum)

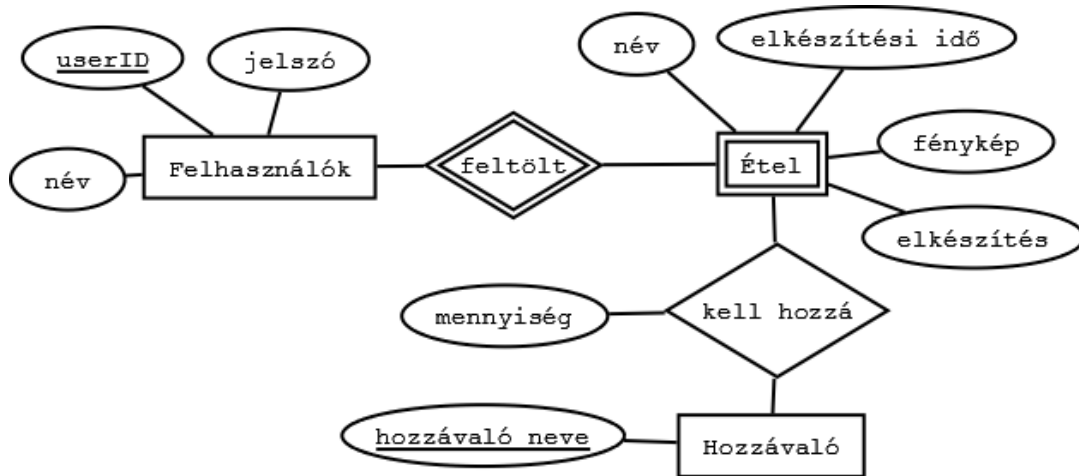
LÁTOGAT(taj, vkód, dátum)

DIAGNOSZTIZÁL(betegség neve, vkód, dátum)

Nincs olyan kapcsolatból leképezett relációséma, amelyben bejelölt kulcs megegyezik egy egyedből leképezett relációséma, ezért nem tudunk összeolvasztani két relációsémát.

## 2.2. példa

Legyen ez a példa is az első feladatban látottak közül. Képezzük le relációsémákká az online szakácskönyv E-K modelljét!



### Megoldás:

Ahogy eddig is, az egyedek leképezzük relációsémákká!

Az Étel gyenge egyed, ezért ehhez hozzá kell venni a meghatározó kapcsolattal kapcsolódó egyed kulcsát.

FELHASZNÁLÓK(név, userID, jelszó)

ÉTEL(fénykép, elkészítés, név, elkészítési idő, userID)

HOZZÁVALÓ(hozzávaló neve)

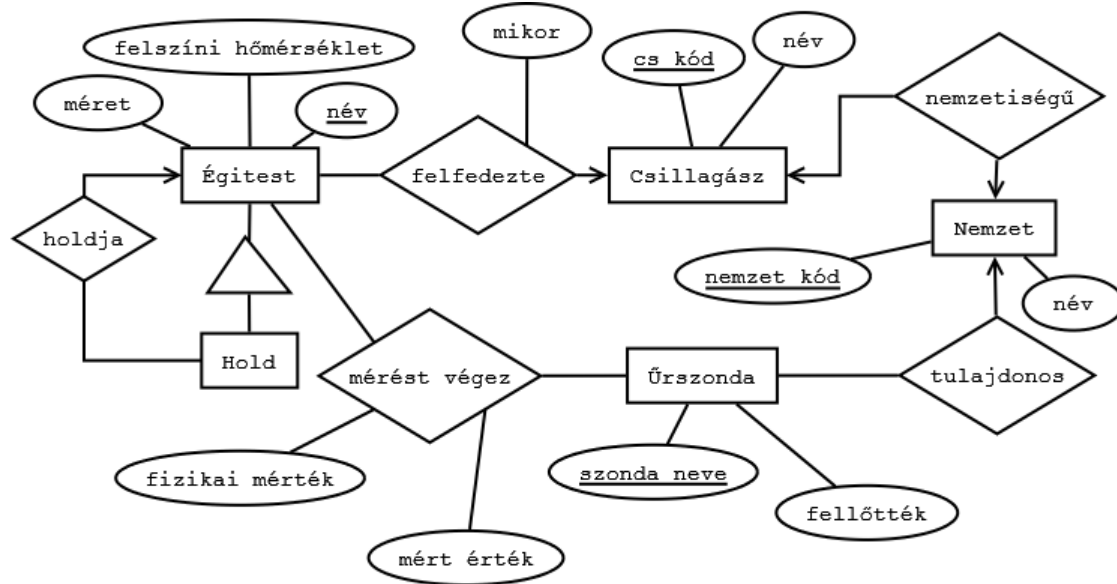
Most csak a *kell hozzá* kapcsolatot kell leképezni, mivel a *feltölt* kapcsolatot lényegében leképeztük azzal, hogy hozzávettük az *ÉTEL* sémához a *userID* attribútumot külső kulcsként.

KELL HOZZÁ(hozzávaló neve, étel.név, userID, mennyiség)

Muszáj felvenni a *userID*-t is, hiszen az lényegében azzal lehet azonosítani az ételt. Gyenge egyednél nem tudunk kulcsot kijelölni az E-K diagramban (de leképezés után igen). Természetesen a kapcsolat attribútumát is ide vesszük fel.

### 2.3. példa

A következő E-K diagram egy olyan adatbázist modellez, amely a égitesteket tartja számon.



#### Megoldás:

Ez egy eléggé összetett E-K diagram. Mint eddig is, most is az egyedek leképezésével kell kezdeni.

A specializáló kapcsolat valójában most nem fog túl sok szerepet játszani, de természetesen erre is ki fogok térni. Lényegében a speciális egyed örökli az ős attribútumait.

ÉGITEST(név, méret, felszíni hőmérséklet)

HOLD(név, méret, felszíni hőmérséklet)

CSILLAGÁSZ(cs kód, név)

NEMZET(nemzet kód, név)

ŰRSZONDA(szonda neve, fellötték)

Most következnek a kapcsolatok leképezése. A specializáló kapcsolattal nem foglalkozunk, az már lényegében benne van az eddigi sémákban.

HOLDJA( hold.név, égitest.név )

FELFEDEZTE(égitest.név, cs\_kód, mikor)

NEMZETISÉGŰ(nemzet\_kód, cs\_kód)

TULAJDONOS(szonda\_neve, nemzet\_kód)

MÉRÉST VÉGEZ(szonda\_neve, égitest.név, mért érték, fizikai mérték)

Most meg kell nézni, hogy melyik sémát lehet beolvasztani egy másikba. A FELFEDEZTE séma kulcsa megegyezik az ÉGISTEST séma kulcsával, ezért töröljük a FELFEDEZTE sémát, és vegyük az ÉGITEST séma attribútumaihoz a cs\_kód és mikor attribútumokat. A kibővített séma kulcsa nem változik.

ÉGITEST(név, méret, felszíni hőmérséklet, cs\_kód, mikor)

A HOLDJA relációséma esetében hasonló a helyzet, töröljük a HOLDJA sémát, és vegyük a HOLD séma attribútumaihoz az égitest.név attribútumot.

HOLD(név, méret, felszíni hőmérséklet, égitest.név)

A NEMZETISÉGŰ séma szintén törölhető, ha a CSILLAGÁSZ sémához hozzávesszük a nemzet\_kód-ot.

CSILLAGÁSZ(cs\_kód, név, nemzet\_kód)

A TULAJDONOS séma hasonlóképpen beolvasható az ŰRSZONDA sémába, így az ŰRSZONDA séma bővílni fog a nemzet\_kód attribútummal.

ŰRSZONDA(szonda\_neve, fellötték, nemzet\_kód)

A MÉRÉST VÉGEZ séma nem olvasható be másik sémába.

Összefoglalva a következő sémákba lehet alakítani a fenti E-K diagramot:

ÉGITEST(név, méret, felszíni hőmérséklet, cs\_kód)

HOLD(név, méret, felszíni hőmérséklet, égitest.név)

CSILLAGÁSZ(cs\_kód, név, nemzet\_kód)

NEMZET(nemzet\_kód, név)

ŰRSZONDA(szonda\_neve, fellötték, tulajdonos\_nemzet\_kód)

MÉRÉST VÉGEZ(szonda\_neve, égitest.név, mért érték, fizikai mérték)

## **Relációséma normalizálása:**

### 3.1. példa

Hozzuk 1NF, 2NF, 3NF alakra az alábbi relációsémát!

HALLGATÓ(aha, név, város, irányítószám, utca, házszám, szak, kar)

Megoldás:

Jelöljük be a sémában a kulcsattribútumok halmazát!

HALLGATÓ(aha, név, város, irányítószám, utca, házszám, szak, kar)

**1NF:** a HALLGATÓ séma 1NF-ben van, mivel nincs benne összetett attribútum.

**2NF:** a HALLGATÓ séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulcstól.

**3NF:** a HALLGATÓ séma nincs 3NF-ben, mivel az *irányítószámtól* függ a város (bontsuk fel a sémát az *irányítószám* mentén)

HALLGATÓ(eha, név, *irányítószám*, utca, házszám, szak, kar)

VÁROS(irányítószám, város)

A HALLGATÓ séma még mindig nincs 3NF-ben, mivel a *szak* függ a *kar*tól (bontsuk fel a sémát a *szak* mentén)

HALLGATÓ(eha, név, *irányítószám*, utca, házszám, *szak*)

SZAK(szak, kar)

VÁROS(irányítószám, város)

A HALLGATÓ séma most már 3NF-ben van, mert nem tudunk több függést kijelölni.

### 3.2. példa

Hozzuk 1NF, 2NF, 3NF normálformára a KÖLCSÖNZÉS relációsémát!

KÖLCSÖNZÉS(olvasójegy, név, lakcím, leltári szám, isbn, könyv címe, kölcsönzés dátuma, visszahozta), ahol LAKCÍM összetett attribútum

LAKCÍM(város, irányítószám, utca, házszám)

Megoldás:

Jelöljük be a sémában a kulcsattribútumok halmazát!

KÖLCSÖNZÉS(olvasójegy, név, lakcím, leltári szám, isbn, könyv címe, kölcsönzés dátuma, visszahozta)

LAKCÍM(város, irányítószám, utca, házszám)

**1NF:** A KÖLCSÖNZÉS relációséma nincs 1NF-ben, mert tartalmaz összetett attribútumot. Ahhoz, hogy 1NF-re hozzuk, az összetett attribútumo(ka)t fel kell bontani.

KÖLCSÖNZÉS(olvasójegy, név, város, irányítószám, utca, házszám, leltári szám, isbn, könyv címe, kölcsönzés dátuma, visszahozta)

A KÖLCSÖNZÉS séma most már 1NF-ben van, mert nem tartalmaz összetett attribútumot.

**2NF:** a KÖLCSÖNZÉS séma nincs 2NF-ben, mert a {*név*, város, irányítószám, utca, házszám} attribútumhalmaz csak az *olvasójegy*től függ, vagyis *olvasójegy* → {*név*, város, irányítószám, utca, házszám}. Bontsuk fel a sémát az *olvasójegy* mentén!

KÖLCSÖNZÉS(olvasójegy, leltári szám, isbn, könyv címe, kölcsönzés dátuma, visszahozta)

OLVASÓ(olvasójegy, név, város, irányítószám, utca, házszám)



A KÖLCSÖNZÉS séma még mindig nincs 2NF-ben, mivel a {könyv címe, isbn} halmaz csak a *leltári számtól* függ, vagyis *leltári szám* → {*könyv címe, isbn*}. Bontsuk fel a sémát a *leltári szám* mentén.

KÖLCSÖNZÉS(olvasójegy, leltári szám, kölcsönzés dátuma, visszahozta)  
OLVASÓ(olvasójegy, név, város, irányítószám, utca, házszám)  
KÖNYV(leltári szám, isbn, könyv címe)

Most már minden séma 2NF-ben van, mivel a sémákban minden másodlagos attribútum teljesen függ bármely kulcstól.

**3NF:** a KÖLCSÖNZÉS séma 3NF-ben van, az OLVASÓ séma nincs 3NF-ben, mivel *irányítószám* → *város*. Bontsuk fel az OLVASÓ sémát az *irányítószám* mentén.

KÖLCSÖNZÉS(olvasójegy, leltári szám, kölcsönzés dátuma, visszahozta)  
OLVASÓ(olvasójegy, név, irányítószám, utca, házszám)  
IRÁNYÍTÓSZÁM(irányítószám, város)  
KÖNYV(leltári szám, isbn, könyv címe)

Az OLVASÓ séma most már 3NF-ben van, azonban a KÖNYV még nincs, mivel a *könyv címe* függ az *isbn számtól*. Bontsuk fel a KÖNYV sémát az *isbn szám* mentén.

KÖLCSÖNZÉS(olvasójegy, leltári szám, kölcsönzés dátuma, visszahozta)  
OLVASÓ(olvasójegy, név, irányítószám, utca, házszám)  
IRÁNYÍTÓSZÁM(irányítószám, város)  
KÖNYV(leltári szám, isbn)  
ISBN(isbn, könyv címe)

Most már az összes relációséma 3NF-ben van.

### 3.3. példa

Hozzuk 1NF, 2NF, 3NF normálformára az ÁRAM relációsémát!

ÁRAM( ünév, vóra\_ száma, szavatosság, mérés\_ kezd, mérés\_ vége, ü cím ),  
ahol az *ü cím* attribútum összetett attribútum (város, utca, házszám, irsz).

Megoldás:

Jelöljük be a kulcsattribútumok halmazát!

ÁRAM( ünév, vóra\_ száma, szavatosság, mérés\_ kezd, mérés\_ vége, ü cím )

Az ÁRAM relációséma nincs 1NF-ben, mert az *ü cím* összetett attribútum.

**1NF:**

ÁRAM( ünév, vóra\_ száma, szavatosság, mérés\_ kezd, mérés\_ vége, város,  
utca, házszám, irsz )

Az ÁRAM relációséma most már 1NF-ben van.

**2NF:** az ÁRAM relációséma nincs 2NF-ben, mert az ügyfél neve (ünev), a szavatosság, a város, utca, házszám, irsz a vóra\_számtól függ. Bontsuk fel a sémát a vóra\_száma attribútum mentén!

ÁRAM(vóra\_száma, mérés\_kezd, mérés\_vége)  
ÜGYFÉL(vóra\_száma, név, város, utca, házszám, irsz)

Így már mindkét relációséma 2NF-ben van.

**3NF:** Az ÜGYFÉL séma nincs 3NF-ben a {vóra\_száma} → {irsz} → {város} tranzitív függés miatt. Bontsuk fel a sémát az irányítószám mentén!

ÁRAM(vóra\_száma, mérés\_kezd, mérés\_vége)  
ÜGYFÉL(vóra\_száma, ünev, utca, házszám, irsz)  
IRÁNYÍTÓSZÁM(irsz, város)

Az összes relációséma 3NF-ben van, készen vagyunk.

## **SQL lekérdezések:**

### 4.1. példa

Egy kikötő adatbázisához az alábbi relációsémák tartoznak:

HAJÓ( szám, név, tulajdonos\_szemszám, típus\_kód )  
TÍPUS( kód, név, hossz, vitorlák\_száma, motor)  
TULAJDONOS( szemszám, név, születési\_dátum, nem )  
ÉRKEZETT( hajó\_szám, dátum, dokk\_szám )  
TÁVOZOTT( hajó\_szám, dátum, dokk\_szám )  
DOKK( szám, méret, napi\_ár )  
FIZET( sorozatszám, hajó\_szám, dokk\_szám, érték, erk\_dátum, táv\_dátum )

a) Hozza létre a HAJÓ táblát, ahol a

- szám egy 10 elemű betűkből, számokból és '-' jelből álló karaktersorozat,
  - név egy legfeljebb 20 karaktert tartalmazó karaktersorozat,
  - tulajdonos\_szem.szám egy 8 hosszú karaktersorozat,
  - típus\_kód egy 4 hosszú egész szám!
- Ügyeljen az integritásellenőrzésre!

```
CREATE TABLE hajo(szam CHAR(10) PRIMARY KEY NOT NULL, név VARCHAR(20) NOT NULL, tulajdonos_szem.szam CHAR(8), típus_kod INT(4), FOREIGN KEY(tulajdonos_szem.szam) REFERENCES tulajdonos(szemszám) );
```

b) A H-123456AB számú hajó 2009. július 3-án érkezett a 23-as dokkba. Regisztrálja ezt a hajót az adatbázisba!

```
INSERT INTO erkezett (hajó_szam, datum, dokk_szam) VALUES ('H-123456AB',
```

'2009-07-03', 23);

c) Törölje az egy évnél régebbi fizetési bejegyzéseket!

```
DELETE FROM fizetes WHERE tav_datum < '2008-09-29';
```

Itt be lehet vetni azt a trükköt, hogy a mai dátum évszámából kivonunk egyet.

d) Növelje meg 10 százalékkal azoknak a dokkoknak a napi árát, melyek mérete meghaladja a 8 m-t!

```
UPDATE dokk SET napi_ar * 1.1 WHERE meret > 8;
```

e) Gyűjtse ki az adatbázisból azokat a hajótulajdonosokat (név, születési dátum), akik elmúltak 40 évesek! Rendezze őket életkoruk szerinti növekvő sorrendbe!

```
SELECT nev, szuletesi_datum FROM tulajdonos WHERE szuletesi_datum < '1969-09-29';
```

f) Gyűjtse ki azon hajók nevét, tulajdonosát, érkezésük idejét, akik 2008. áprilisában kötöttek ki.

```
SELECT hajo.nev AS hajo_nev, tulajdonos.nev AS tulaj_nev, erkezett.datum  
FROM hajo, tulajdonos, erkezett  
WHERE tulajdonos.szemszam = hajo.tulajdonos_szemszam AND  
      hajo.szam = erkezett.hajo_szam AND  
      erkezett.datum > '2008-03-31' AND erkezett.datum < '2008-05-01';
```

A FROM után képezzük a tulajdonos, hajo és erkezett táblák Descartes-szorzatát. A WHERE feltételben ebből kiszűrjük azokat a sorokat, amelyeknél a a hajo táblában tárolt szám megegyezik az erkezett tabla külső kulcsával és a hajo tábla tulajdonos\_szemszam értéke megegyezik a tulajdonos tábla szemszam ertekevel. A maradék sorok összetartoznak, már csak a kérdéses időpontban kikötött hajókat kell kiszűrni. Ezt megtehetjük a kikötés dátuma alapján. Végül kiválasztjuk a hajo.nev, tulajdonos.nev, erkezett.datum oszlopokat. Az AS itt csak arra való, hogy az eredményben az oszlop neve ne hajo.nev legyen, hanem hajo\_nev.

g) Egy Kalóz típusú vitorlás érkezik a kikötőbe. Ellenőrizze egy SQL lekérdezéssel, hogy van-e számára üres dokk!

Ez egy eléggé összetett, kicsit bonyolult feladat. Több részre kell osztani, úgy könnyebb megoldani:

- ki kell gyűjteni az üres dokkokat
- meg kell nézni, hogy a Kalóz típusú hajónak mekkora a hossza
- meg kell nézni, hogy melyek azok a dokkok, amelyek üresek és elfér benne egy Kalóz típusú vitorlás

Az alábbi lekérdezés azokat a dokkokat gyűjti ki, ahová érkezett hajó, de már távozott is

```
SELECT erk.dokk_szam FROM érkezett AS erk, tavozott AS tav WHERE
erk.dokk_szam = tav.dokk_szam GROUP BY erk.dokk_szam HAVING
MAX(erk.datum) < MAX(tav.datum);
```

Ezeket a dokkokat ki kell venni abból a halmazból, ahová eddig érkezett hajó, így megkapjuk azokat a dokkokat, amelyekben még vannak hajók. Az előző lekérdezést allekérdezésként építjük be.

```
SELECT DISTINCT dokk_szam FROM érkezett WHERE dokk_szam NOT IN
(SELECT erk.dokk_szam FROM érkezett AS erk, tavozott AS tav WHERE
erk.dokk_szam = tav.dokk_szam GROUP BY erk.dokk_szam HAVING
MAX(erk.datum) < MAX(tav.datum))
```

Ezeket a dokkokat ki kell venni a dokkok halmazából, és azt is ellenőrizni kell, hogy melyek azok a dokkok, ahol elfér egy Kalóz típusú hajó. (Az áttekinthetőség kedvéért tagoltam a lekérdezések egyes részeit.)

```
SELECT dokk.szam
FROM dokk, tipus
WHERE tipus.nev = 'Kalóz' AND dokk.meret >= tipus.hossz AND dokk.szam NOT
IN (
    SELECT DISTINCT dokk_szam
    FROM érkezett
    WHERE dokk_szam NOT IN (
        SELECT erk.dokk_szam
        FROM érkezett AS erk, tavozott AS tav
        WHERE erk.dokk_szam = tav.dokk_szam
        GROUP BY erk.dokk_szam
        HAVING MAX(erk.datum) < MAX(tav.datum)
    )
);
```