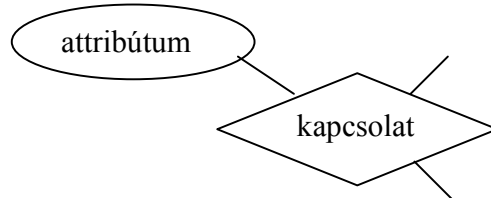
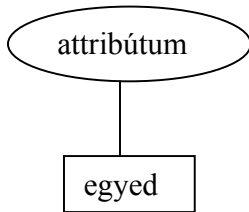
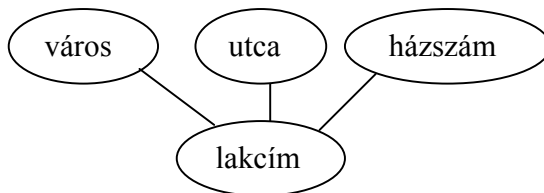


Adatmodellezés

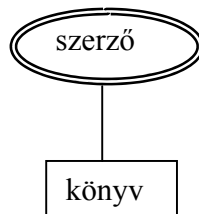
Az egyed-kapcsolat modell (EK-diagram) elemei (egyedek, attribútumok, kapcsolatok):



- a tulajdonságokat (attribútumokat) úgy választjuk meg, hogy azok egyértelműen meghatározzák az egyedeket
- egy attribútum lehet összetett is (állhat attribútumokból)

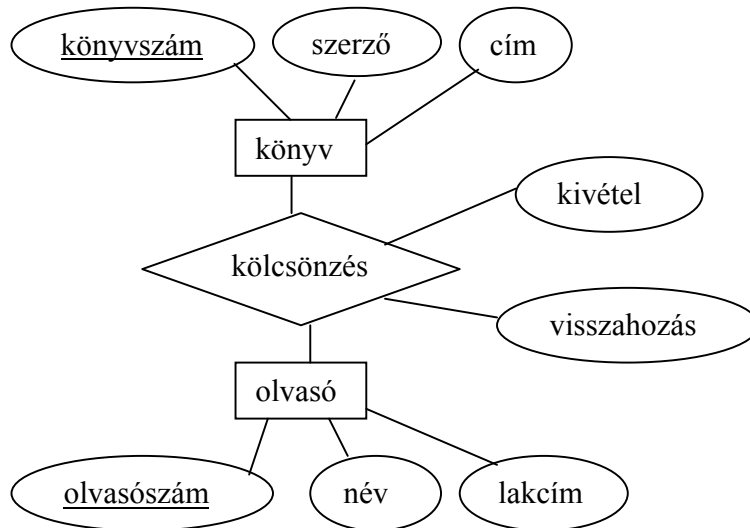


- egy attribútum lehet többértékű is: halmaz vagy lista az értéke



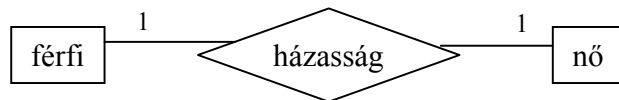
kulcs: egy vagy több attribútum együtt, amely egyértelműen meghatározza az egyedet (legyen minimális, azaz elhagyva valamit is belőle, a megmaradt attribútumok már ne azonosítsanak egyértelműen), a diagramon a kulcsot a megfelelő attribútumok aláhúzásával jelöljük

Példa: egy könyvtár könyveit és olvasóit modellezzük, nyilvántartást vezetünk a kölcsönzési és a visszahozási időpontokról is (az olvasókat és a könyveket egy általunk bevezetett könyvszám illetve olvasószám attribútummal azonosítjuk egyértelműen, azaz ezek lesznek a kulcsok)



kapcsolatok típusai (bejelölendők a diagramon):

- kettőnél több egyed közötti (nem jelöljük)
- két egyed közötti:
 - **1:1** kapcsolat (például férj-feleség kapcsolat Európában)



- **1:N** kapcsolat (például anya-gyerek kapcsolat)



- **N:M** kapcsolat (hasonlóan jelölendő)

Az EK-diagram konkrét adatmodellről független modellezést tesz lehetővé. Az általunk használt konkrét adatmodell a **relációs adatmodell**, de léteznek más modellek is (pl.:objektumorientált). A relációs adatmodell elemei a **táblák**, melyek **sorokból (rekordokból)** épülnek fel. Ez esetben a kulcs egy vagy több attribútum, mely a sort egyértelműen azonosítja és minimális. A kulcsban lévő attribútumokat **elsődleges attribútumoknak** nevezzük.

Példa:

DOLGOZÓK

d. kód	d. neve	d. címe	fizetés
001	Kovács Pál	Szeged	25.000
002	Nagy Anett	Szeged	30.000
003	Kovács Pál	Pécs	28.000
004	Kis Géza	Pécs	25.000

Ebben a táblában a kulcs a **d.kód** mező, így az elsődleges attribútum. Az összes többi attribútum másodlagos.

Amennyiben csak a táblázat nevét, első sorát (az attribútumok neveit) valamint a kulcsokat adjuk meg, úgy **relációs adatbázissémáról** beszélünk.

Relációs adatbázisséma készítése EK-modellből

- **Egyedek átírása:** az EK-modell minden egyedéhez felírunk egy relációs adatbázissémát, melynek attribútumai az egyed attribútumai, kulcsa az egyed kulcsattribútumaiból áll.
- **Kapcsolatok átírása:**
 - 1:1 kapcsolat esetén kiválasztjuk a kapcsolatban résztvevő egyedek egyikét (bármelyiket) és annak relációs sémájába felvesszük új attribútumként a másik egyed kulcsattribútumait, valamint a kapcsolat attribútumait.
 - 1:N kapcsolat esetén az N oldali egyed relációs sémáját bővítjük úgy, mint 1:1 kapcsolat esetén
 - N:M kapcsolat esetén új relációs sémát veszünk fel, melynek attribútumai a kapcsolódó egyedek kulcsattribútumai valamint a kapcsolat saját attribútumai
 - kettőnél több egyed közötti kapcsolat esetén is úgy járunk el, mint N:M kapcsolat esetén

Példa: az előzőleg megadott könyvtári alkalmazás esetén a kölcsönzés N:M típusú kapcsolat, így az alábbi sémák keletkeznek.

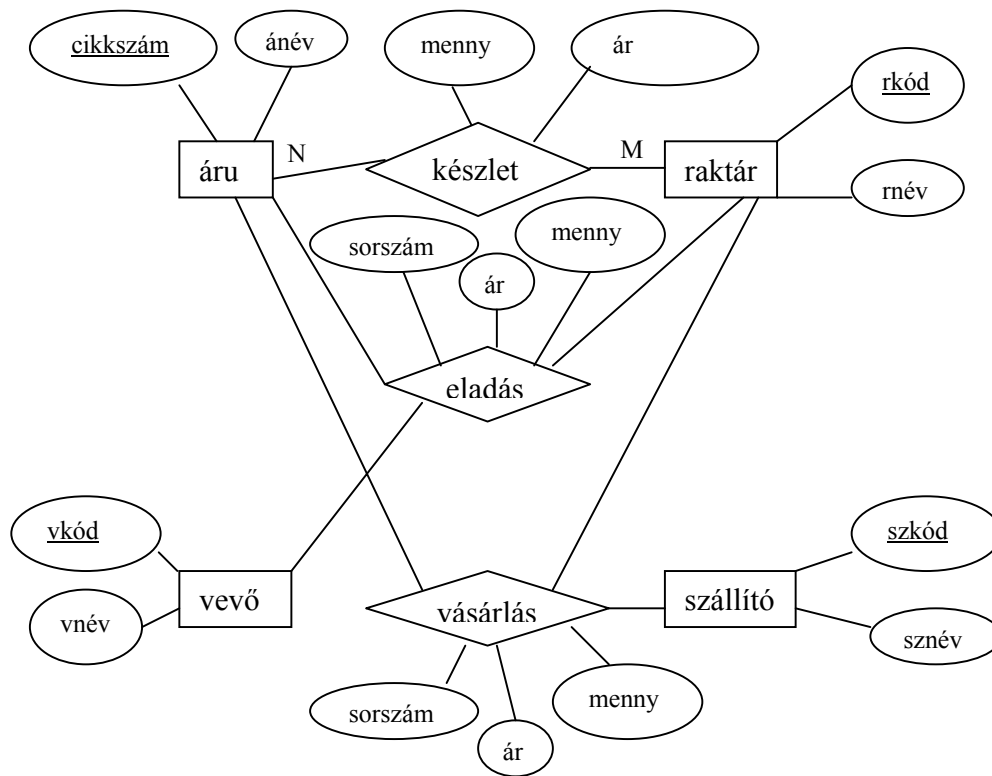
KÖNYV(könyvszám, szerző, cím)

OLVASÓ(olvasószám, név, lakcím)

KÖLCSÖNZÉS(könyvszám, olvasószám, kivétel, visszahozás)

Ha valamely kulcsattribútum egy másik séma elsődleges kulcsára hivatkozik, akkor azt **külső kulcsnak** nevezzük, és dőlt írással jelöljük.

Példa: az alábbi EK-diagramm azt modellezi, hogy áruk érkeznek szállítóktól, raktárakban tárolják őket, majd a vevők megveszik őket.



Az ehhez a diagrammhoz tartozó relációs adatbázissémák:

áru (cikkszám, ánév)

raktár (rkód, rnév)

szállító (szkód, sznév)

vevő (vkód, vnév)

készlet (cikkszám, rkód, menny, ár) → N:M kapcsolat átírása

eladás (cikkszám, rkód, vkód, sorszám, menny, ár)

vásárlás (cikkszám, rkód, szkód, sorszám, menny, ár)

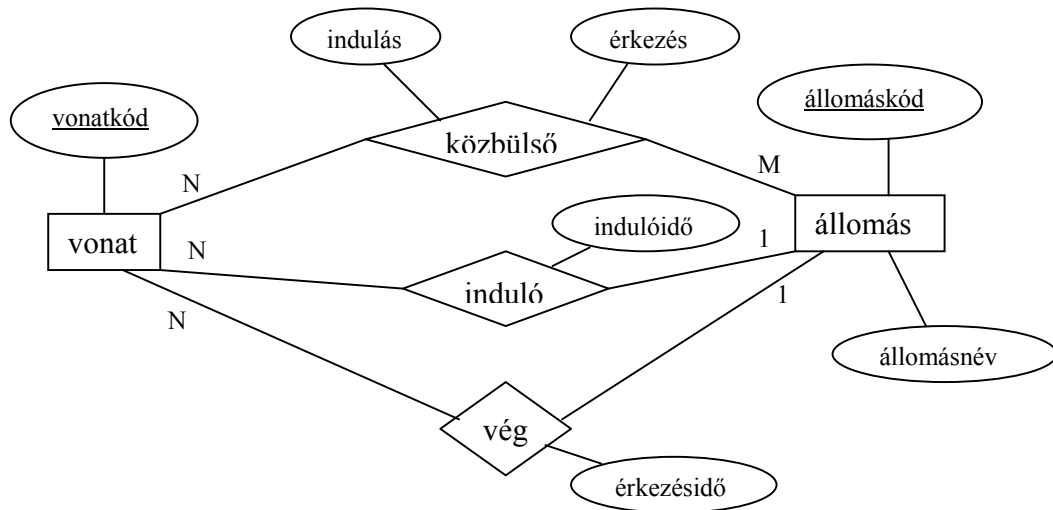
az egyedekre vonatkozó átírási szabályok alapján

} Bekerült a sorszám is a kulcsba,
csak így egyértelmű az azonosítás!

Milyen feltételek mellett lehet ez az átírás?

Milyen más lehetséges átírások lehetnek még, ha máshogy értelmezzük az attribútumokat?

Példa: Az alábbi EK-diagramm vonatok induló-, közbűlső- és végállomásait modellezi (egy vonatnak csak egy indulóállomása van, de erről az állomásról több vonat is indulhat, ezért ez 1:N kapcsolat). Értelmezzük a többi kapcsolatot is!



Az ehhez a diagrammhoz tartozó relációs adatbázissémák az egyedekre és az N:M kapcsolatokra vonatkozó átírási szabályok alkalmazása után:

vonat (vonatkód)

állomás (állomáskód, állomásnév)

közűlsőállomás (állomáskód, vonatkód, indulás, érkezés)

Az indulóállomásra vonatkozó 1:N kapcsolat esetén a **vonat** sémát bővítenünk kell:

vonat (vonatkód, állomáskód, indulóidő)

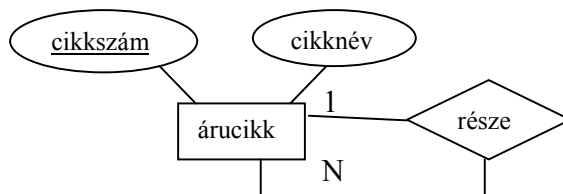
Majd tovább bővítjük a végállomásra vonatkozó 1:N kapcsolat miatt (az *állomáskód* mező kétszer szerepel a sémában, ezért meg kell különböztetnünk őket névlegesen).

vonat (vonatkód, indulóállomáskód, indulóidő, végállomáskód, érkezésidő)

Megjegyzés: A „közűlsőállomás” sémánál a vonatkód+indulás vagy a vonatkód+érkezés is funkcionálhat kulcsként.

Előfordulhat, hogy egy egyed ugyanabban a kapcsolatban többször vesz részt:

Példa: egy számítástechnikai kereskedés árul számítógép-konfigurációkat, de árul külön alkatrészeket is, melyek azonban konfigurációknak is lehetnek elemei



Hasonló kapcsolat áll fent egy vállalat dolgozói és a főnökök között. A diagrammhoz tartozó relációs adatbázisséma:

árucikk (cikkszám, cikknév)

Az 1:N kapcsolat átírása után (megkülönböztetve a két *cikkszám* mezőt):

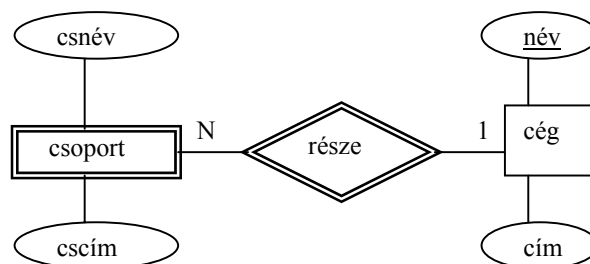
árucikk (cikkszám, cikknév, minekrész_cikkszám)

Előfordulhat, hogy egy egyed az attribútumai nem határozzák meg egyértelműen csak a kapcsolatai révén lesz egyértelműen meghatározott → gyenge entitás (jele: kettős téglalap), az ilyen egyedet meghatározó kapcsolat neve meghatározó kapcsolat (jele: kettős rombusz).

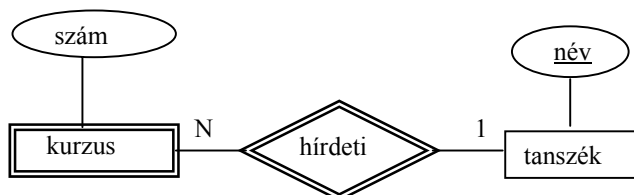
Előfordulhat:

- Amikor kettőnél több egyedes kapcsolatokat átírunk kétegyedes kapcsolatokra
- Redundancia csökkentése során (adatok ismétlésének csökkentése)

Példa: A csoportnév nem kulcs (sok cégnél lehet ugyanolyan nevű csoport), de ha a kapcsolaton keresztül a céget is be vesszük az azonosításba, akkor egyértelmű lesz, hogy melyik csoportról beszélünk (ugyanaz a helyzet egyetemek és karaik között).

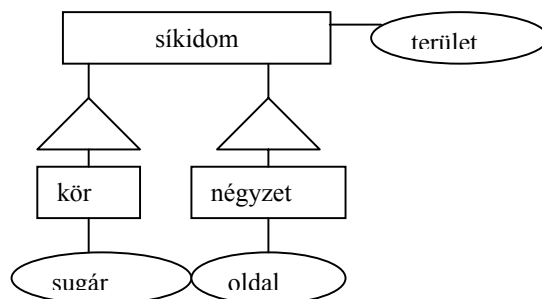


Feladat: Tervezzünk EK diagrammot az alábbi problémára. Egyedek: kurzusok, tanszékek. Egy kurzust egy tanszék hirdet meg, és azt egy számmal azonosítja. Különböző tanszékek adhatják ugyanazt a számot a kurzusoknak, de egy tanszék tárgyai mind különböző számot kapnak.



Specializáló kapcsolat (ha bizonyos altípusokat külön szeretnénk modellezni).

Példa.:



Adatbázisséma készítése

Gyenge entitások leképezése: a gyenge entitás relációssémáját bővíteni kell a meghatározó kapcsolat(ok)ban szereplő egyed(ek) kulcsával.

CSOPORT(csnév, cscím, név)

CÉG(név, cím)

Összetett attribútumok leképezése: A sémát úgy vesszük fel, hogy abban elemi attribútumok legyenek csak.

Többértékű attribútumok leképezése:

1. Új tábla felvétele. A KÖNYV(könyvszám, szerző, cím) sémát helyettesítjük a KÖNYV(könyvszám, cím) és SZERZŐ(könyvszám, szerző) sémákkal.
2. Ha a sorrend is fontos, akkor a SZERZŐ táblát egy sorszám mezővel kell bővíteni → SZERZŐ(könyvszám, sorszám, szerző).

Specializáló kapcsolatok leképezése: több megoldás is lehetséges, mindegyiknek van előnye és hátránya is.

Normalizálás

- **1. normálforma:** a cellákban csak egyszerű adatok lehetnek (nincsenek értékhalmozok, érték n-esek)
- **2. normálforma:** minden másodlagos attribútum teljesen függ a kulcstól, azaz a kulcsból bármely attribútumot elhagyva már nem áll fenn függőség
- **3. normálforma:** nincs kulcstól való tranzitív függés

Példa: hozzuk az alábbi adatbázissémát 1NF-re, 2NF-re, 3NF-re

számla (számlaszám, dátum, vevőkód, vevőnév, vevőcím, sorok)

- a *sorok* mező a számlán szereplő sorokat jelzi
- egy sor felépítése a következő: (árukód, árunév, egységár, mennyiség)
- egy sort az árukód egyértelműen azonosít, azaz ugyanazon a számlán nem szerepelhet két ugyanolyan árukódot tartalmazó sor

1 NF: A *sorok* cella nem egyszerű, hiszen egy sor 4 bejegyzést tartalmaz.

megoldás:

- válasszuk le az összetett cellát az eredeti sémából
- vegyünk fel egy új sémát, melynek mezői a leválasztott cella mezői valamint az eredeti séma kulcsa ('új séma kulcsa' = 'eredeti séma kulcsa' + 'leválasztott cella kulcsa')

számla (számlaszám, dátum, vevőkód, vevőnév, vevőcím)

sorok (számlaszám, árukód, árunév, egységár, mennyiség)

2 NF: A **számla** sémában a kulcs egyelemű, így igaz, hogy a kulcsból bármely attribútumot elhagyva már nem áll fenn függőség (hiszen az üres halmaz nem határozhat meg semmit), tehát a **számla** séma már 2 NF-ben van. A **sorok** séma azonban nincs 2NF-ben, mert az *árukód* egyedül is meghatározza, az *árunév* mezőt (az *egységár* mezőt nem, mert az függ attól, mikor adták ki a számlát, tehát a *számlaszám* mezőtől is).

megoldás:

- válasszuk le azokat a cellákat az eredeti sémából, melyeket a kulcs része egyértelműen meghatároz (jelen esetben az *árunév* mezőt)
- vegyünk fel egy új sémát, melynek mezői a leválasztott mezők és az őket meghatározó kulcsrészlet (jelen esetben az *árukód* mező) ('új séma kulcsa' = 'leválasztott mezőket meghatározó kulcsrészlet')

számla (számlaszám, dátum, vevőkód, vevőnév, vevőcím)

sorok (számlaszám, árukód, egységár, mennyiség)

áruk (árukód, árunév)

3 NF: A **sorok** illetve az **áruk** sémában nincs tranzitív (többlépéses) függés, így azok már 3NF-ben vannak. A **számla** sémában a következő tranzitív függés van: a *számlaszám* meghatározza a *vevőkód* mezőt, az pedig a *vevőnév* illetve a *vevőcím* mezőket.

számlaszám → *vevőkód* → *vevőnév*, *vevőcím*

megoldás:

- az eredeti sémában csak a középső (csatoló) mezőt hagyjuk meg (jelenleg: *vevőkód*)
- vegyünk fel egy új sémát, melynek kulcsa a csatolómező lesz, mezői pedig a tranzitív függés második felét alkotó mezők

számla (számlaszám, dátum, vevőkód)

vevők (vevőkód, vevőnév, vevőcím)

sorok (számlaszám, árukód, egységár, mennyiség)

árúk (árukód, árunév)

Példa: hozzuk az alábbi videokölcsönzői adatbázissémát 1NF-re, 2NF-re, 3NF-re

kazetták (kazetta száma, film címe, film száma, rendelés száma, rendelés dátuma, sorok)

- egy kazettán csak egy film van, de egy film több példányban is meglehet
- a rendelés dátuma azt jelzi, hogy a kölcsönző mikor vásárolta a filmet
- a *sorok* mező azt tartalmazza, hogy mikor ki kölcsönözte ki a kazettát
- egy sor felépítése a következő: (kölcsönzés száma, tag sorszáma, tag neve, kölcsönzés dátuma, visszahozás dátuma)
- egy sort a kölcsönzés száma egyértelműen azonosít, a kölcsönzés száma úgy jön létre, hogy a tag sorszámahoz "hozzáragasztják" a kölcsönzés dátumát

1 NF: felbontjuk a *sorok* cellát, mert az összetett

kazetták (kazetta száma, film címe, film száma, rendelés száma, rendelés dátuma)

sorok (kazetta száma, kölcsönzés száma, tag sorszáma, tag neve, kölcsönzés dátuma, visszahozás dátuma)

2 NF: A **sorok** sémát felbontjuk, mert ott a *kölcsönzés száma* (mely a kulcsnak csak egy része) maga is egyértelműen azonosítja a *tag sorszáma*, *tag neve*, *kölcsönzés dátuma* illetve *visszahozás dátuma* mezőket.

kazetták (kazetta száma, film címe, film száma, rendelés száma, rendelés dátuma)

sorok (kazetta száma, kölcsönzés száma)

kölcsönzések (kölcsönzés száma, tag sorszáma, tag neve, kölcsönzés dátuma, visszahozás dátuma)

3 NF: Tranzitív függések felbontása. A **kazetták** séma tranzitív függései:

- *kazetta száma* → *film száma* → *film címe*
- *kazetta száma* → *rendelés száma* → *rendelés dátuma*

A **kölcsönzések** séma tranzitív függése:

- *kölcsönzés száma* → *tag sorszáma* → *tag neve*

sorok (kazetta száma, kölcsönzés száma)

kölcsönzések (kölcsönzés száma, tag sorszáma, kölcsönzés dátuma, visszahozás dátuma)

tagok (tag sorszáma, tag neve)

kazetták (kazetta száma, film száma, rendelés száma)

filmek (film száma, film címe)

rendelések (rendelés száma, rendelés dátuma)