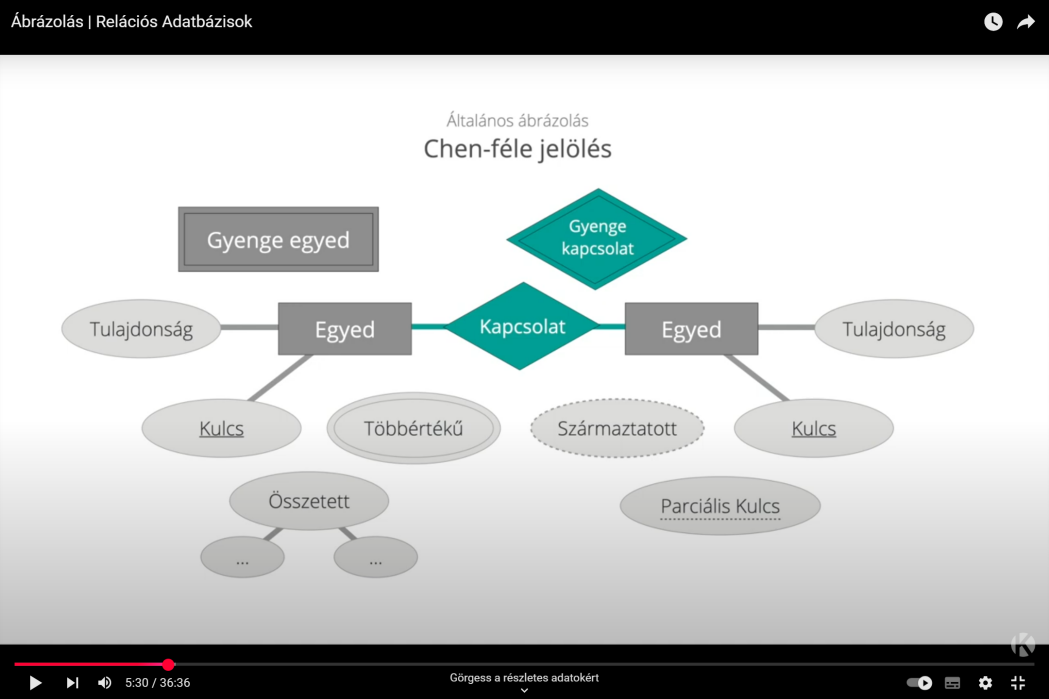
Relációs adatbázis



#### ****1. Egyed (Entitás)****

* Az **egyed** egy olyan objektum vagy dolog, amelyet az adatbázisban tárolunk.
* Téglalapokkal jelölik a diagramon.
* Az egyednek lehetnek **tulajdonságai** (attribútumai), amelyeket ellipszisek jelölnek.

#### ****2. Gyenge egyed****

* Egy speciális egyedtípus, amely **nem létezhet önállóan**, csak egy másik egyedhez kapcsolódva.
* Szürke színnel és dupla téglalappal jelölik.

#### ****3. Kapcsolatok****

* A kapcsolatok (relationship) az egyedek közötti összefüggéseket mutatják meg.
* **Rombusz alakú elemek** jelölik őket.
* Egy kapcsolaton belül több egyed is részt vehet.

#### ****4. Gyenge kapcsolat****

* Egy speciális kapcsolat, amely gyenge egyedeket köt össze az őket támogató egyeddel.
* **Kiemelt, zöld színű rombusszal** jelölik.

#### ****5. Attribútumok (Tulajdonságok)****

* Az egyedekhez tartozó információkat **ellipszisek** jelölik.
* Többféle attribútum létezik:
  + **Kulcs (Primary Key)**: Egyedi azonosító egy adott egyedhez.
  + **Összetett attribútum (Composite Attribute)**: Több részadatból áll.
  + **Többértékű attribútum (Multivalued Attribute)**: Egy entitáshoz több érték is tartozhat.
  + **Származtatott attribútum (Derived Attribute)**: Más attribútumokból számított érték.
  + **Parciális kulcs (Partial Key)**: Gyenge egyedek esetében az azonosítás segítésére szolgál.



#### ****1. Parciális kulcs****

* A **gyenge egyedek** azonosítására szolgáló kulcs.
* **Önállóan nem elégséges** az azonosításhoz, mert egy **erős egyed kulcsával együtt** kell használni.
* Az **erős egyed kulcsával kombinálva** biztosítja az egyed egyediségét az adatbázisban.

**Példa:** Képzeljünk el egy adatbázist, amely egy lakópark lakásait tárolja.

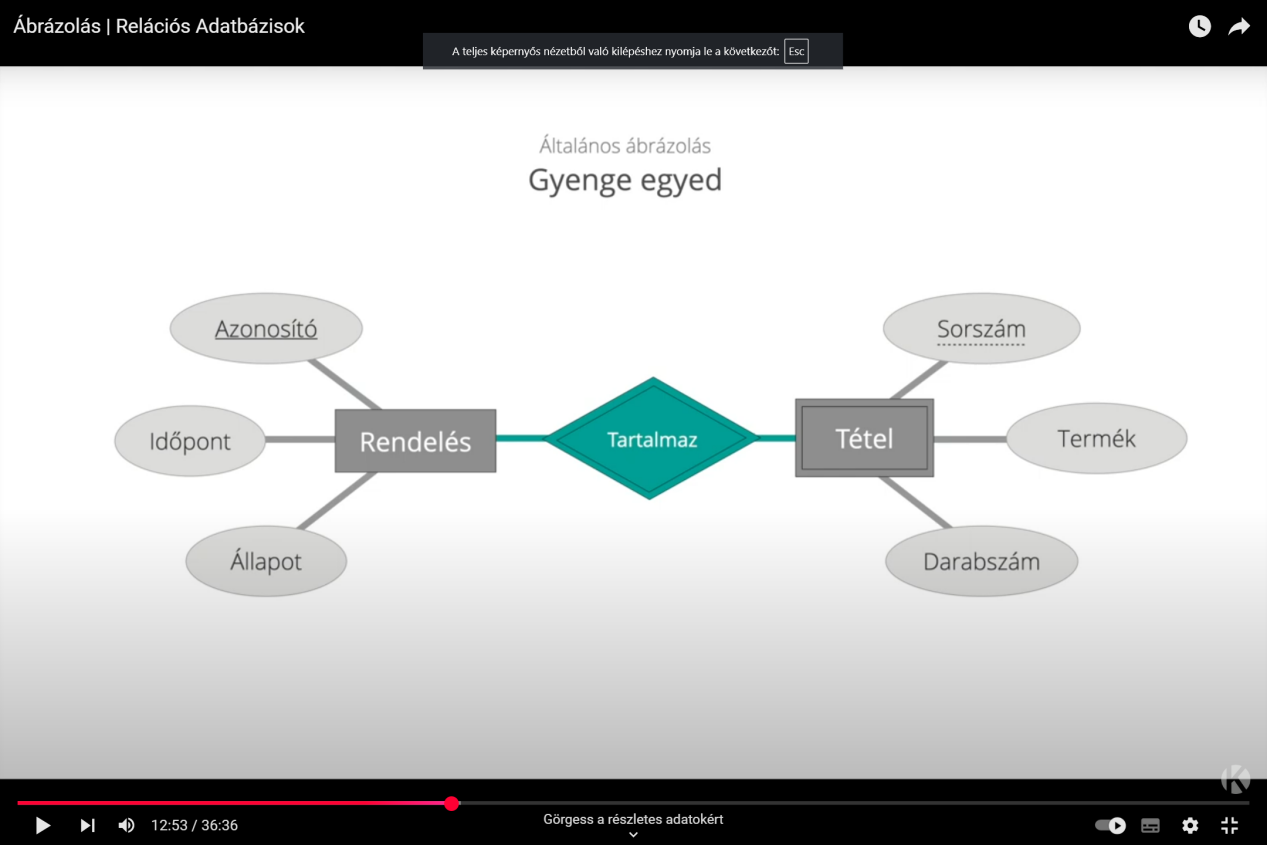
* **Lakás (Gyenge egyed):** Az épületen belüli lakásoknak lehet saját száma (pl. "2A", "3B"), de **ez nem egyedi**, mert több épületben is lehet ilyen lakás.
* **Épület (Erős egyed):** Minden épületnek van egy egyedi azonosítója (pl. "E001", "E002").
* **Parciális kulcs:** A lakások azonosításához szükséges az **épület azonosítója és a lakás száma együtt** (pl. "E001-2A", "E002-3B").

#### ****2. Gyenge kapcsolat****

* Akkor jön létre, ha egy **gyenge egyed kapcsolódik egy erős egyedhez**.
* Az ilyen kapcsolatot **gyenge kapcsolatnak nevezzük**.
* A gyenge kapcsolat biztosítja, hogy a gyenge egyed **nem létezhet önállóan**, csak az erős egyedhez kötve.

**Példa:** Gondoljunk egy adatbázisra, amely egy könyvtár könyveit tárolja.

* **Könyv (Erős egyed):** Egyedi azonosítóval rendelkezik (pl. ISBN szám).
* **Példány (Gyenge egyed):** A könyvtárban egy adott könyvből több példány is lehet, amelyeket példányazonosítóval különböztetünk meg (pl. "P001", "P002").
* **Gyenge kapcsolat:** Egy könyvpéldány csak egy adott könyvhöz tartozhat, önmagában nem létezhet.



### ****Gyenge egyed magyarázata – Rendelés és Tétel kapcsolat****

Az adatbázisok tervezésében az **egyedeket (entitásokat)** és azok **kapcsolatait** gyakran grafikus modellek segítségével ábrázoljuk. Az alábbi ábra a **gyenge egyed** fogalmát mutatja be egy **rendelési rendszer** példáján keresztül.

### ****1. Egyedek és attribútumaik****

Az ábrán két egyed látható:

**Rendelés (Erős egyed)**

* + Azonosító (egyedi kulcs)
  + Időpont
  + Állapot

**Tétel (Gyenge egyed)**

* + Sorszám
  + Termék
  + Darabszám

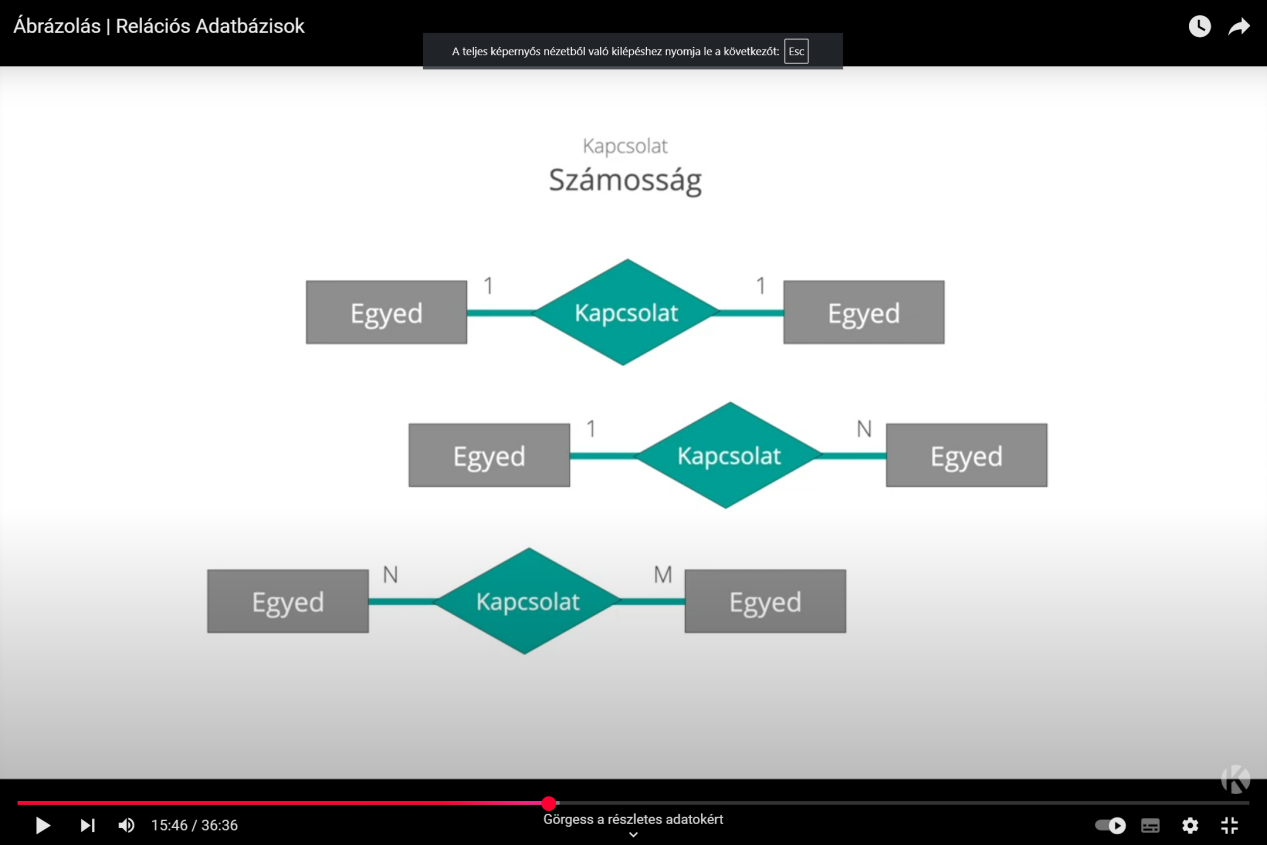
A **rendelés** egy **erős egyed**, mert minden rendelésnek van egyedi azonosítója.  
A **tétel** egy **gyenge egyed**, mert egyedül nem azonosítható egyértelműen. Egy rendelés több tételt is tartalmazhat, és önmagában egy tételsorszám nem biztosít egyediséget.

### ****2. Kapcsolat a két egyed között****

A **"Tartalmaz"** kapcsolat mutatja, hogy egy rendeléshez több tétel is tartozhat.  
Mivel a tétel egy **gyenge egyed**, ezért az azonosításához nem elég a saját sorszáma, hanem **a rendelés azonosítója is kell** hozzá.

### ****3. Miért gyenge egyed a "Tétel"?****

A "Tétel" nem létezhet önállóan az adatbázisban, mert csak egy rendeléshez kötve van értelme. Egy tételnek csak egy sorszáma van (pl. 1, 2, 3), de ha nincs hozzárendelve egy rendelési azonosítóhoz, akkor nem lehet tudni, hogy melyik rendeléshez tartozik.



## ****1. Mi az a számosság?****

A **számosság** azt mutatja meg, hogy egy adott egyedhez **hány másik egyed kapcsolódhat** egy adott kapcsolatban.  
Az egyedek közötti kapcsolatokat mindig **1 (egy) vagy N (több)** lehetőséggel jelöljük.

Például:

* Ha egy diák **csak egy iskolába** járhat, de egy iskola **több diákot is fogadhat**, akkor ez egy **1:N kapcsolat**.
* Ha egy autóhoz **több alkatrész tartozhat**, és egy alkatrész **több autóban is megtalálható**, akkor ez egy **N:M kapcsolat**.

## ****Kapcsolatok típusai az ábrán****

Az ábrán háromféle kapcsolat látható:

### ****a) 1:1 kapcsolat (egy az egyhez)****

Egy **egyedhez csak egy másik egyed tartozhat**.  
Példa:

* Egy személynek **csak egy személyi igazolványa** van, és egy személyi igazolvány **csak egy személyhez tartozik**.

### ****b) 1:N kapcsolat (egy a többhöz)****

Egy **egyedhez több másik egyed is kapcsolódhat**, de a másik irányban csak egy kapcsolat lehetséges.  
Példa:

* Egy tanár **több diákot** taníthat, de egy diák **csak egy tanárhoz tartozik** egy adott tantárgy esetén.
* Egy ügyfél **több rendelést** adhat le, de egy rendelés **csak egy ügyfélhez tartozik**.

### ****N:M kapcsolat (több a többhöz)****

Egy **egyedhez több másik egyed is kapcsolódhat**, és ez fordítva is igaz.  
Példa:

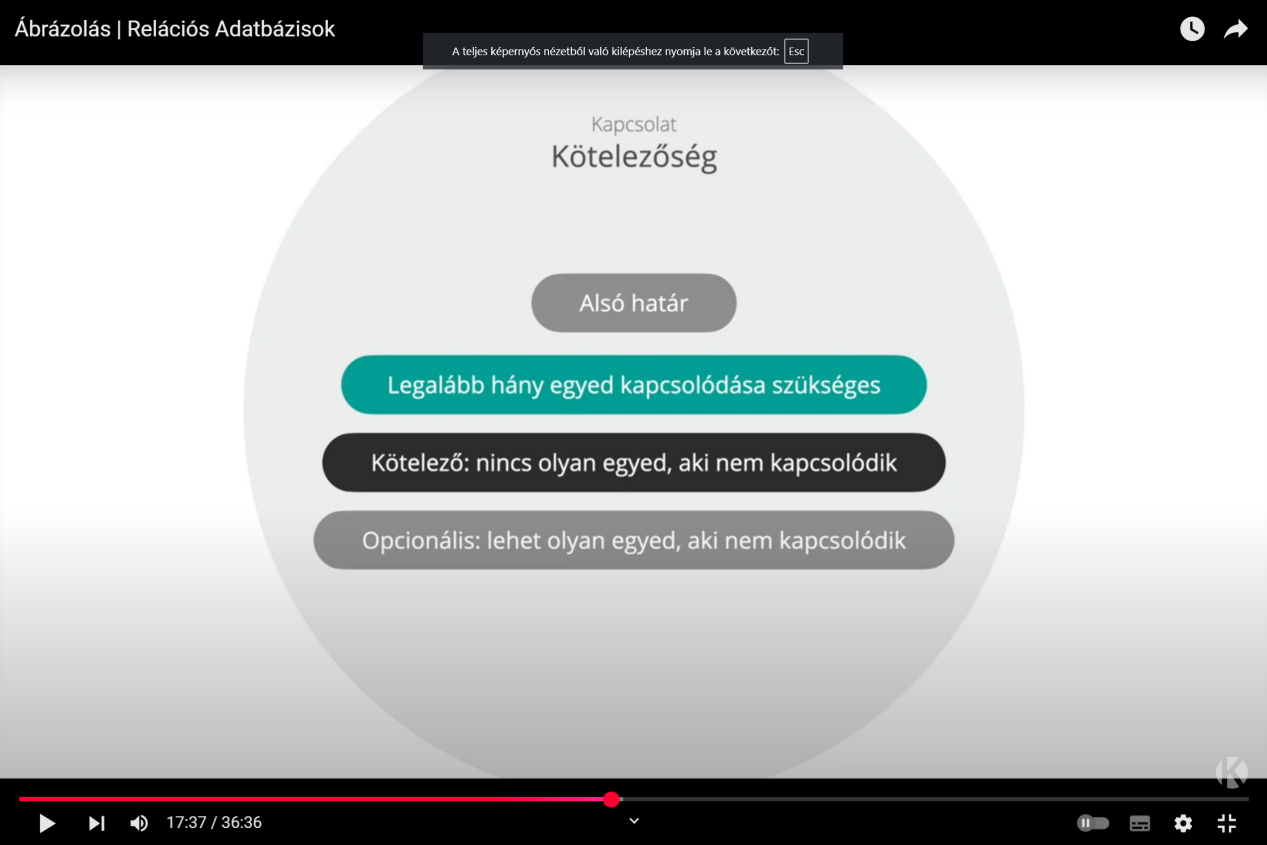
* Egy diák **több tantárgyat** tanulhat, és egy tantárgyat **több diák is tanulhat**.
* Egy könyvet **több olvasó kikölcsönözhet**, és egy olvasó **több könyvet is kikölcsönözhet** a könyvtárban.

További példák:

**1:1** – Egy embernek **egy** születési anyakönyvi kivonata van.

**1:N** – Egy tanár **több** osztályt taníthat.

**N:M** – Egy diák **több** sportágban is versenyezhet, és egy sportágban **több** diák is részt vehet.



Az adatbázisok tervezésénél **a kapcsolatok nemcsak a számosságot határozzák meg**, hanem azt is, hogy egy egyednek **feltétlenül kell-e kapcsolódnia egy másik egyedhez** vagy sem. **Ezt nevezzük kötelezőségnek.**

A kötelezőség segít abban, hogy **meghatározzuk az alsó határt**, vagyis **legalább hány egyed kapcsolódása szükséges** egy adott kapcsolaton belül.

## ****Kötelezőség fogalma és jelentősége****

A kötelezőség azt határozza meg, hogy egy egyednek **mindenképpen kell-e kapcsolódnia egy másik egyedhez**, vagy lehet önállóan is létező.

**Mire használjuk?**

* Adatbázisok tervezésekor meghatározzuk, hogy egy egyed mindig kapcsolódik-e egy másikhoz, vagy előfordulhat-e, hogy nem kapcsolódik semmihez.
* Ez segít az **adathibák kiszűrésében** és az **üzleti szabályok betartásában**.

## ****Példák a kötelezőségre****

**Kötelező kapcsolat (Legalább 1 egyedhez kell kapcsolódnia)**  
Példa:

* **Egy rendelésnek mindig kell tartalmaznia legalább egy tételt.**
  + Egy rendelés nem lehet üres, tehát **minden rendelés kapcsolódik legalább egy tételhez**.
  + **Alsó határ: 1**

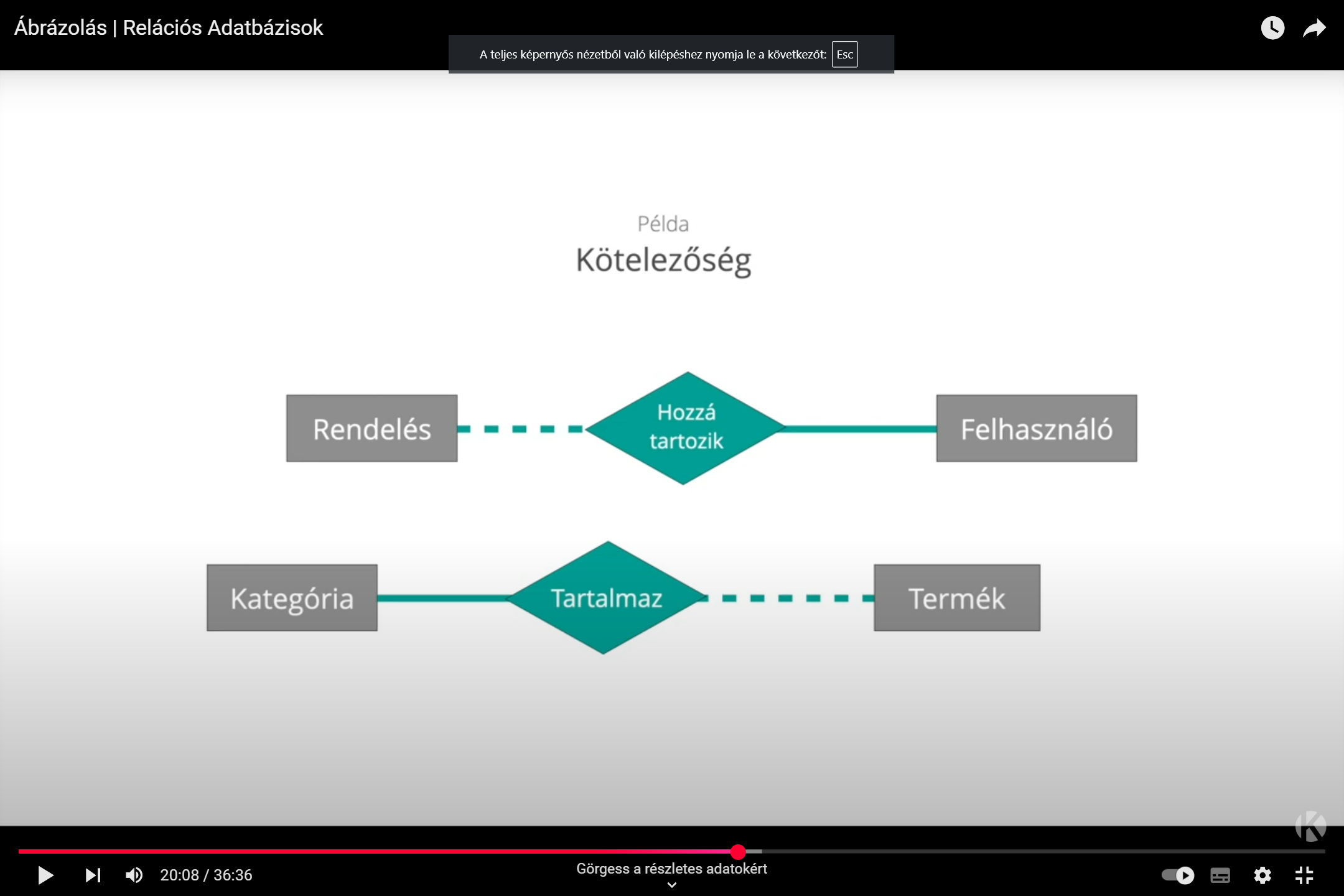
**Opcionális kapcsolat (Lehet kapcsolódás, de nem kötelező)**  
Példa:

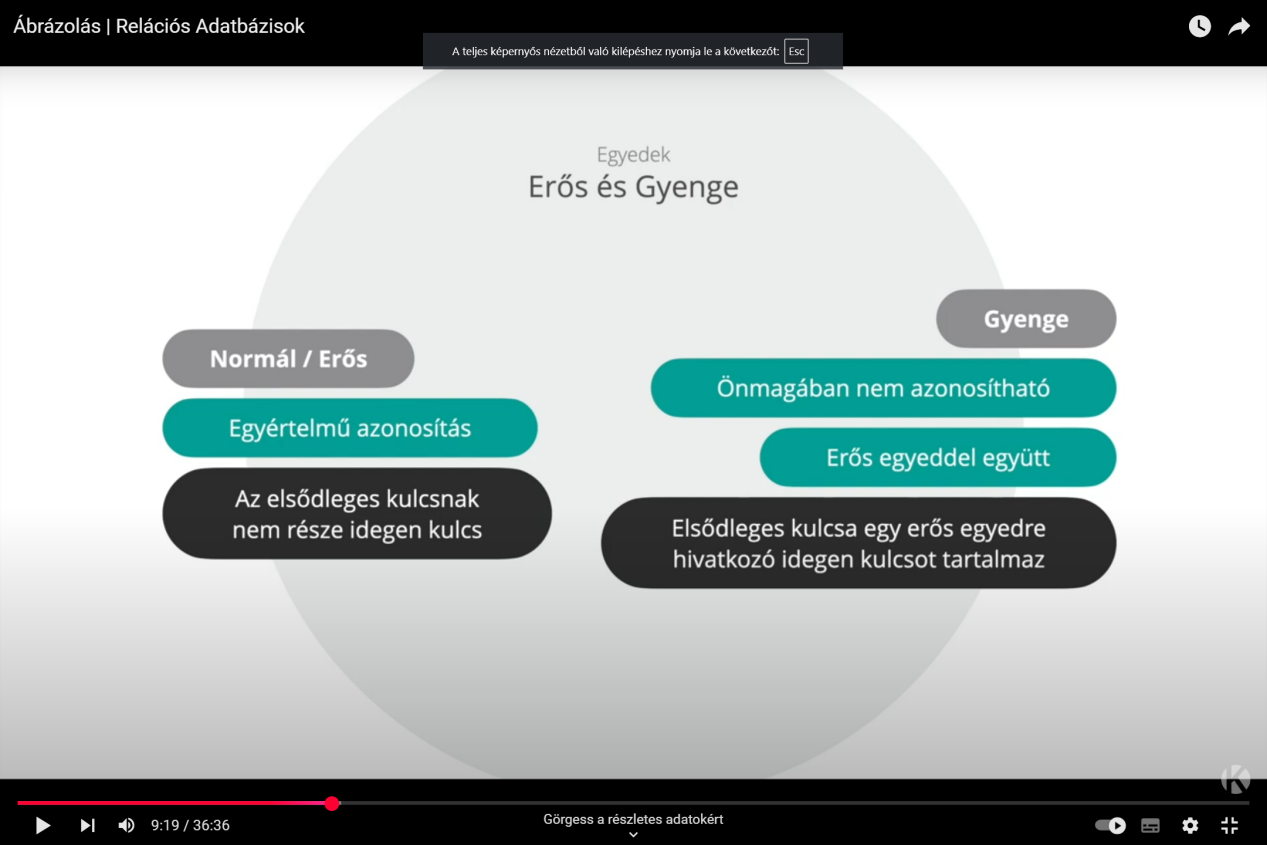
* **Egy ügyfél lehet, hogy még nem adott le rendelést.**
  + Az adatbázisban lehetnek olyan ügyfelek, akik még nem vásároltak semmit, így **nem kötelező, hogy minden ügyfélhez tartozzon rendelés**.
  + **Alsó határ: 0**

## ****Kötelezőség és számosság összekapcsolása****

Egy kapcsolat lehet **1:N vagy N:M**, de közben azt is meg kell határozni, hogy **kötelező-e a kapcsolódás vagy opcionális**.

| **Példa** | **Számosság** | **Kötelezőség** |
| --- | --- | --- |
| Egy ügyfél több rendelést is leadhat | 1:N | **Opcionális** (Nem minden ügyfél rendel) |
| Egy rendeléshez tartoznia kell legalább egy tételnek | 1:N | **Kötelező** (Nincs üres rendelés) |
| Egy diák több tantárgyat is felvehet | N:M | **Opcionális** (Nem minden diák vesz fel minden tantárgyat) |
| Egy munkavállalónak kötelezően tartoznia kell egy osztályhoz | 1:N | **Kötelező** (Minden dolgozónak van osztálya) |





### 1. ****Normál/Erős egyed****:

* **Erős egyed (strong entity)** olyan entitás, amely rendelkezik egy **egyedi azonosítóval (elsődleges kulccsal)**, ami önállóan elegendő az azonosításhoz. Az erős egyedek nem függnek más entitásoktól az azonosításukhoz, tehát saját egyedi kulcsuk van, amely alapján beazonosíthatók.
  + **Példa:** Egy „Diák” tábla, ahol minden diák egyedi azonosítójával (pl. diák ID) rendelkezik, és egyedülállóan beazonosítható.

### 2. ****Egyértelmű azonosítás****:

* Az egyértelmű azonosítás azt jelenti, hogy az entitás azonosítója egyértelműen meghatározható, és nincs két olyan egyed, amely ugyanazzal az azonosítóval rendelkezne. Az **elsődleges kulcs** biztosítja az egyértelmű azonosítást.
  + **Példa:** A diákok tábla esetében a diák ID lenne az elsődleges kulcs, amely minden diákot egyedileg azonosít.

### 3. ****Az elsődleges kulcsnak nem része idegen kulcs****:

* Az **elsődleges kulcs** az a mező vagy mezők csoportja, amely biztosítja az egyértelmű azonosítást az entitás számára. **Idegen kulcs** akkor szerepelne, ha egy másik tábla azonosítójára hivatkozik. Az elsődleges kulcsnak nem lehet része idegen kulcs, mert az elsődleges kulcs az entitás saját egyedi azonosítóját képviseli.
  + **Példa:** A „Diák” táblában a „diák\_id” elsődleges kulcs, és nem lehet idegen kulcs.

### 4. ****Gyenge entitás****:

* A **gyenge entitás** olyan entitás, amely nem rendelkezik saját elsődleges kulccsal. A gyenge entitásokat csak más entitásokkal együtt lehet azonosítani, tehát **támogatásra szorulnak** egy erős entitás által. Az ilyen entitások **idegen kulcsot** tartalmaznak, amely a szülő erős entitás elsődleges kulcsára hivatkozik.
  + **Példa:** Egy „Tétel” tábla, amely egy „Rendelés” tábla részét képezi, és nem rendelkezik saját elsődleges kulccsal, hanem a rendelés ID-jával együtt azonosítható.

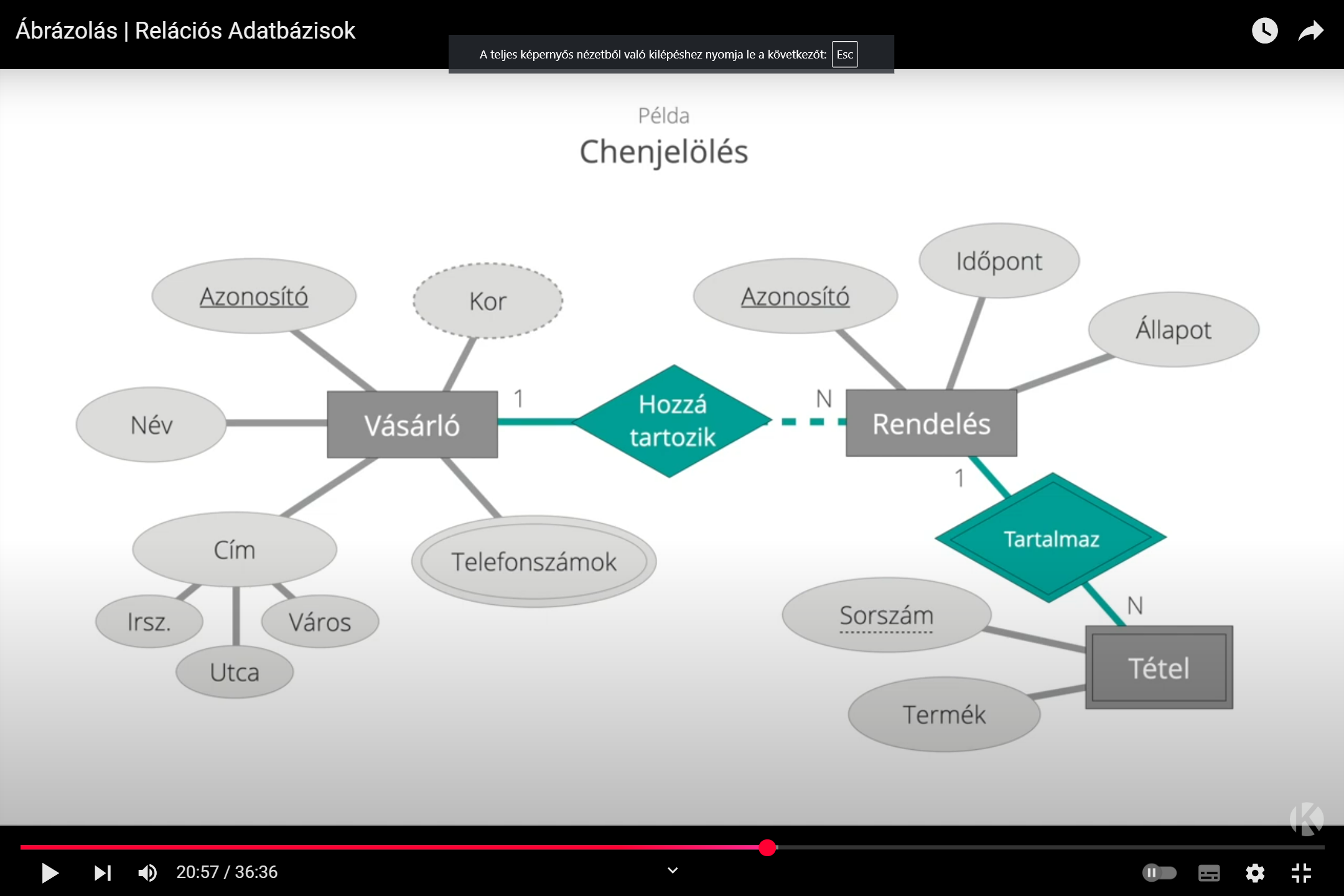
### 5. ****Önmagában nem azonosítható****:

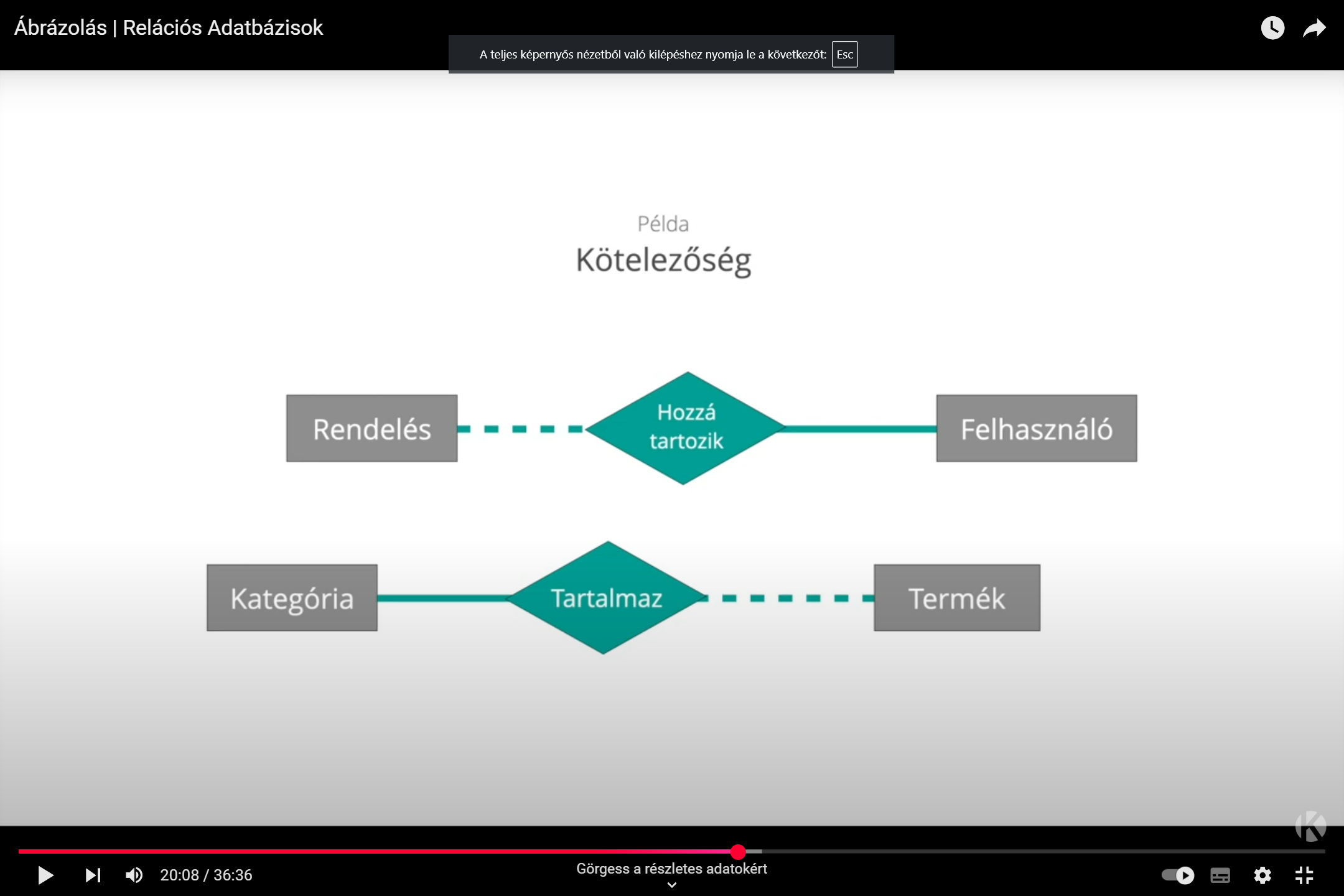
* A gyenge entitás „önmagában nem azonosítható”, mert nincs egyedülálló azonosítója. Az ilyen entitásokat csak egy másik entitás (erős entitás) segítségével lehet azonosítani. Ezért az erős entitás elsődleges kulcsát idegen kulcsként tartalmazzák a gyenge entitásban.
  + **Példa:** Egy „Rendelés tétel” tábla, amely csak akkor azonosítható, ha kapcsolódik egy rendeléshez.

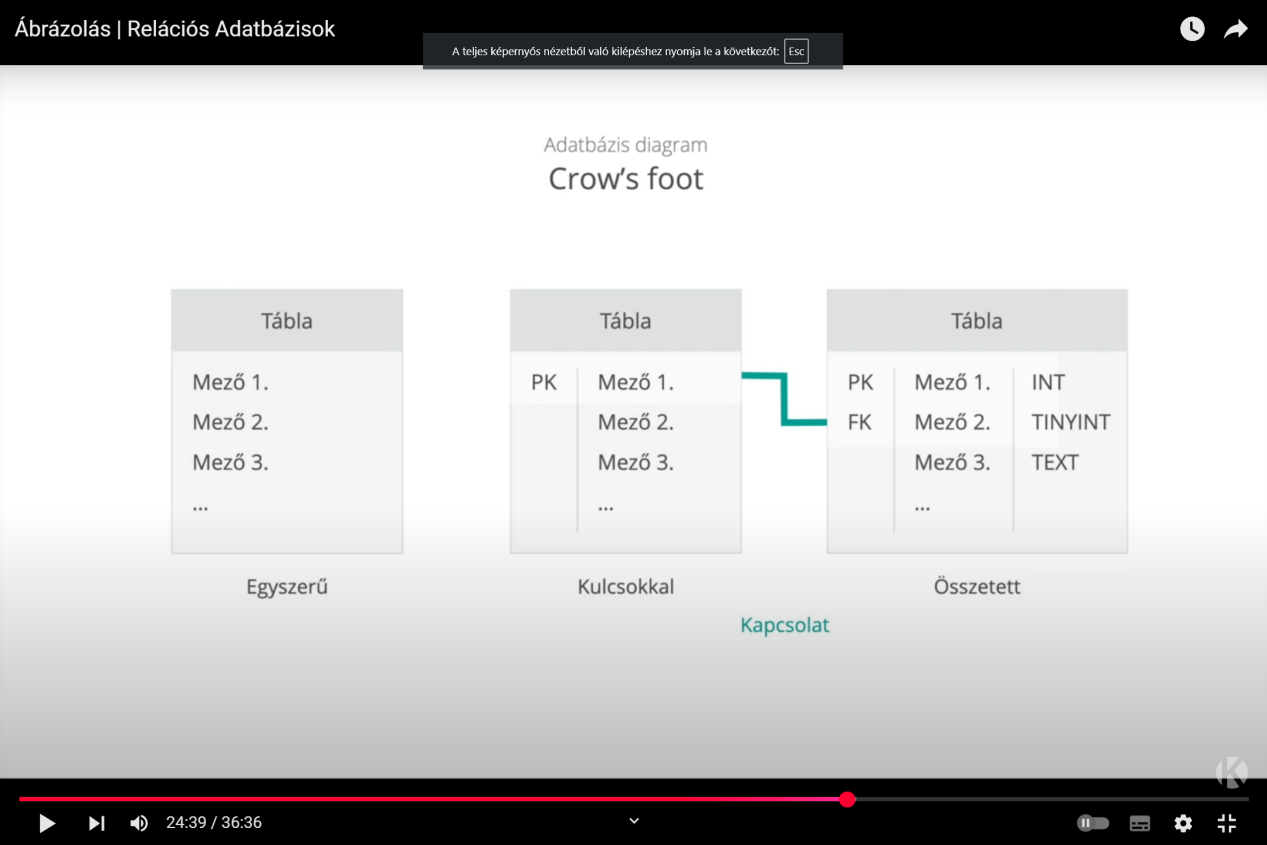
### 6. ****Erős egyeddel együtt****:

* Egy **gyenge entitás** az erős entitás segítségével van azonosítva. Ezért az erős entitás elsődleges kulcsa egy idegen kulcsként jelenik meg a gyenge entitásban, és együtt alkotják az egyedi azonosítót.
  + **Példa:** A „Tétel” tábla az „Rendelés” táblához tartozik, így az „Rendelés ID” és a „Tétel ID” együtt biztosítja a tétel egyedi azonosítását.

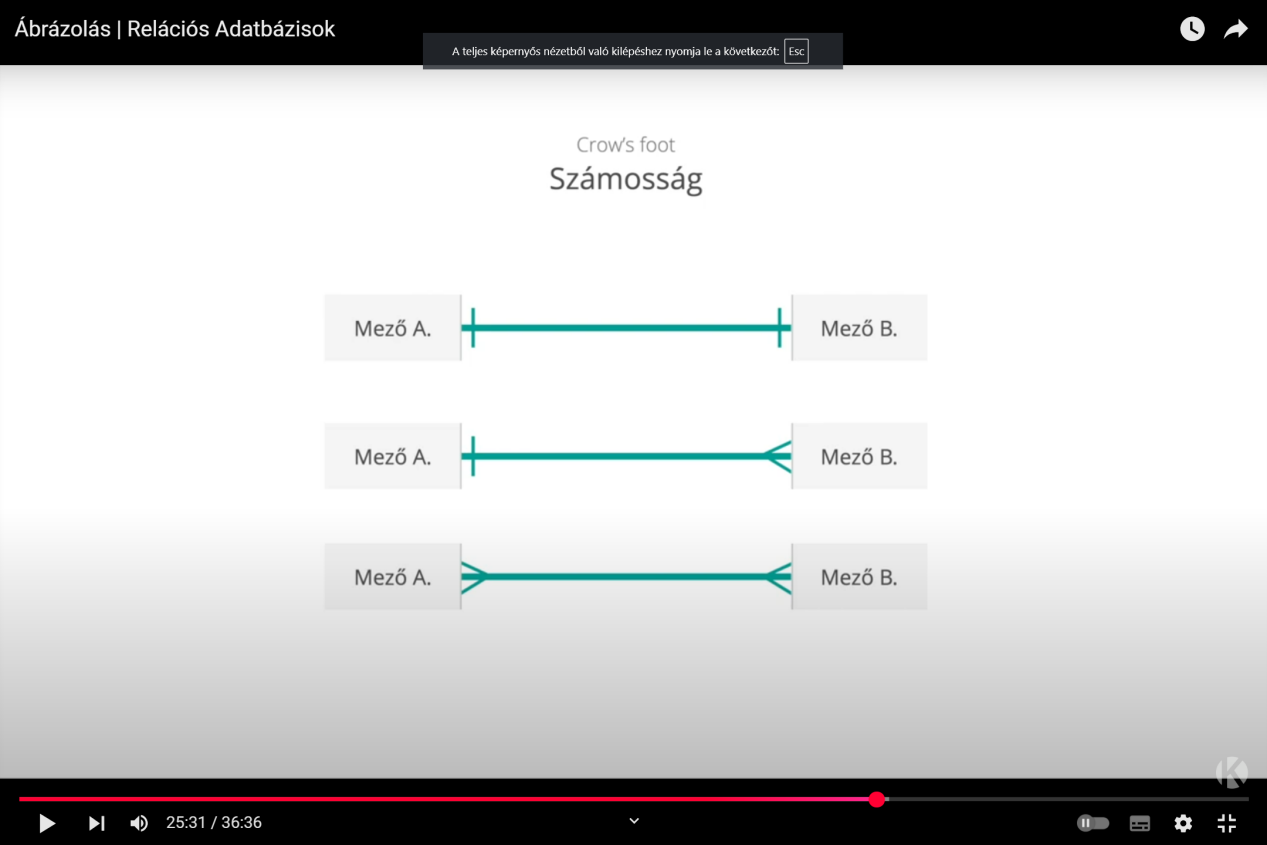
### 7. ****Elsődleges kulcs, amely egy erős egyedre hivatkozó idegen kulcsot tartalmaz****:

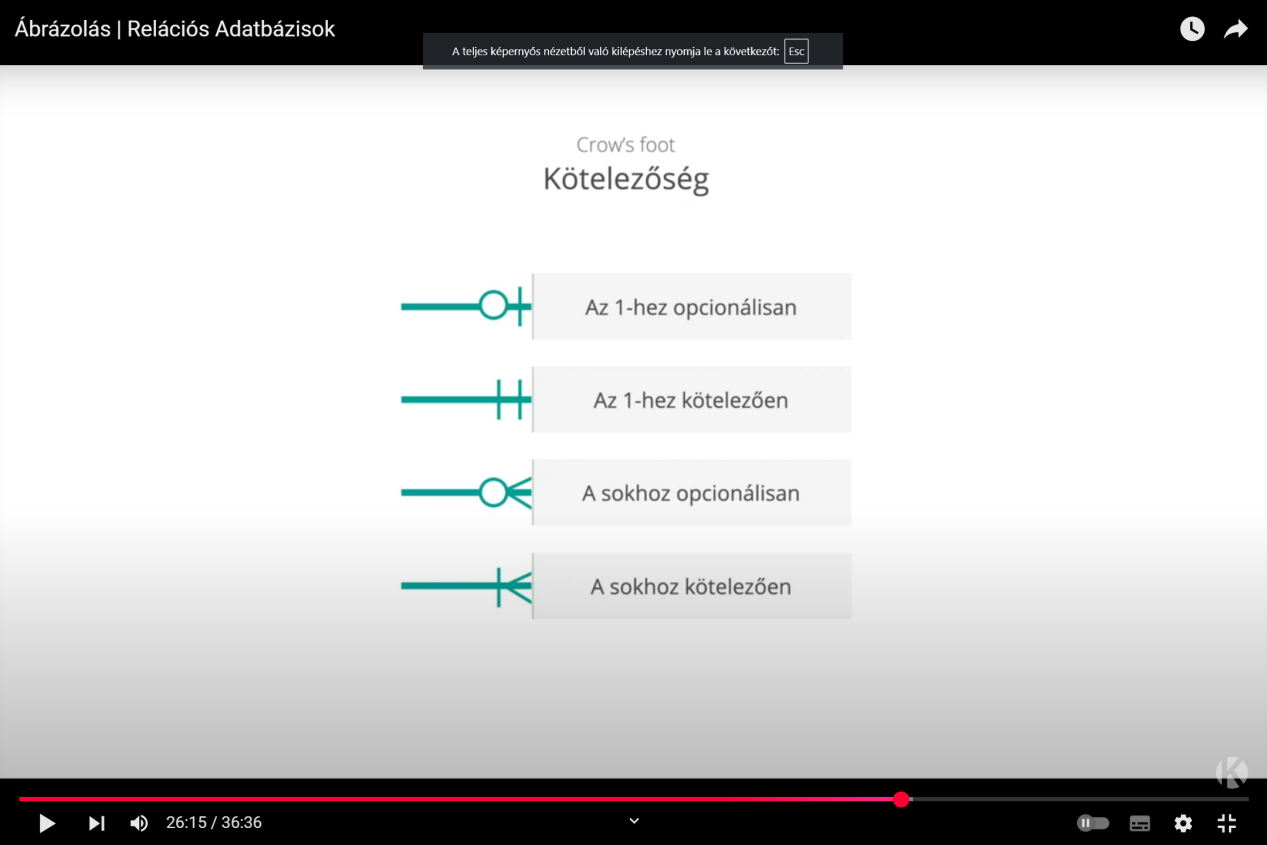
* Ez azt jelenti, hogy a gyenge entitás az erős entitás elsődleges kulcsát tartalmazza idegen kulcsként, és ennek a kombinációnak az összessége alkotja a gyenge entitás egyedi azonosítóját. A gyenge entitás az idegen kulcsot és egyéb mezőket tartalmaz.
  + **Példa:** A „Rendelés tétel” táblában az „Rendelés ID” idegen kulcs, amely az erős „Rendelés” tábla elsődleges kulcsára hivatkozik, és a tétel azonosítását az „Rendelés ID” és a „Tétel ID” együttesen biztosítja.

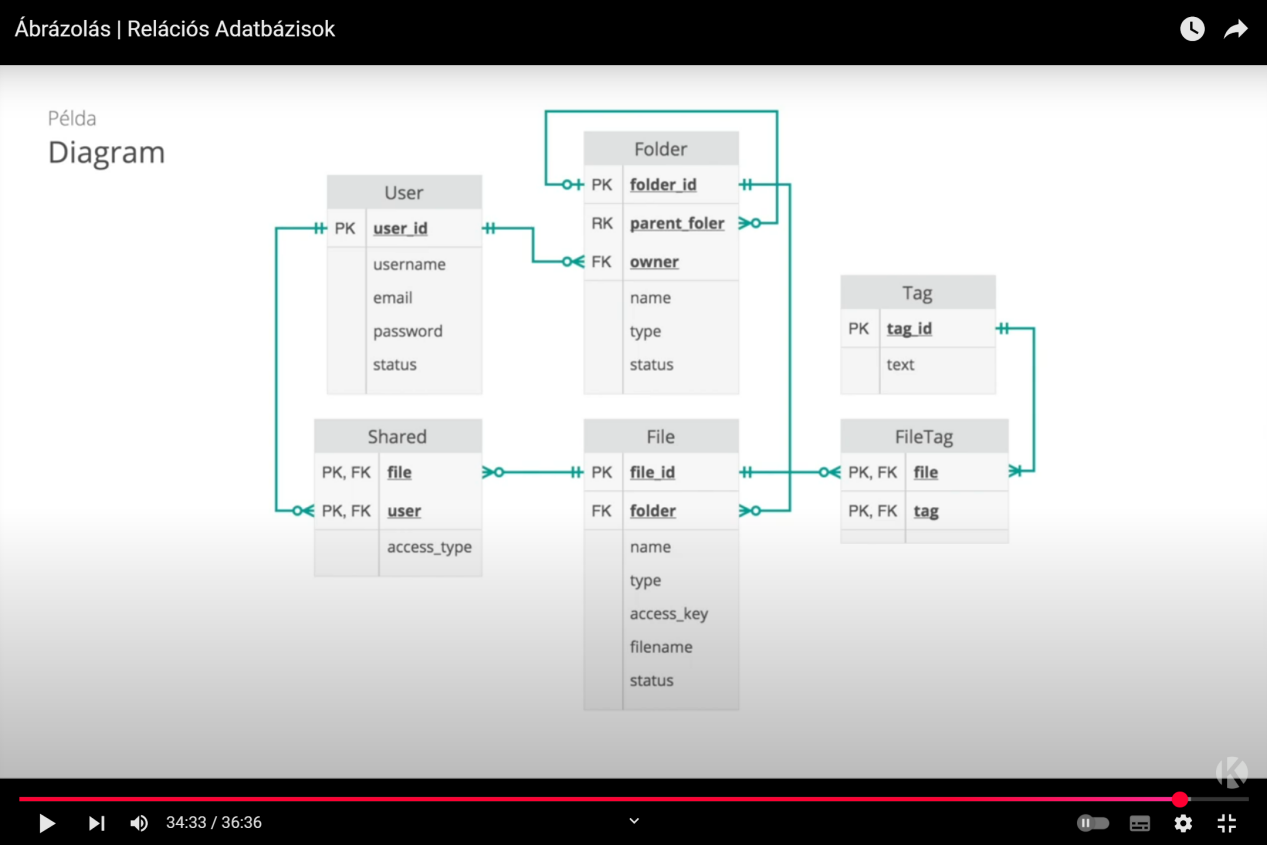




A **Crow’s Foot (Varjúláb) jelölés** egy **grafikus ábrázolási mód**, amelyet **kapcsolati adatbázisok** (ERD – **Entity-Relationship Diagramok**) tervezésére használnak. A neve a kapcsolatokat jelző **"varjúláb"** szimbólumból származik, amely egy ágas vonalhoz hasonlít.







### ****Adatbázis diagram magyarázata (Crow’s Foot jelöléssel)****

Ez az **ERD (Entity-Relationship Diagram)** egy fájlkezelő rendszer adatbázisának **tervezési modelljét** mutatja. Az egyedeket (táblákat) és azok kapcsolatait a **Crow’s Foot** jelöléssel ábrázolja.

## ****2. Kapcsolatok és jelentésük****

A **kapcsolatokat** a Crow’s Foot jelölések mutatják:

**User – Folder kapcsolat (1:N)**

* Egy **felhasználó több mappát birtokolhat**, de egy mappa **csak egy felhasználóhoz tartozhat**.

**Folder – Folder kapcsolat (1:N, önkapcsolat)**

* Egy mappa **lehet egy másik mappa alá rendelve** (például egy főmappának lehetnek almappái).

**Folder – File kapcsolat (1:N)**

* Egy **mappában több fájl lehet**, de egy fájl **csak egy adott mappában** található.

**Shared kapcsolat (M:N kapcsolat)**

* Egy fájl **több felhasználóval is megosztható**, és **egy felhasználónak több megosztott fájlja is lehet**.

**File – Tag kapcsolat (M:N kapcsolat)**

* Egy fájlhoz **több címke tartozhat**, és egy címke **több fájlhoz** kapcsolódhat.

A kapcsolótábla mindig **legalább két külső kulcsot (FK - Foreign Key) tartalmaz**, amelyek a kapcsolódó táblák elsődleges kulcsaira mutatnak.

**"Shared" tábla esetében:**

* **file (PK, FK)** → a fájl azonosítója (kapcsolat a File táblához).
* **user (PK, FK)** → a felhasználó azonosítója (kapcsolat a User táblához).
* **access\_type** → Meghatározza a hozzáférés típusát (pl. olvasás, írás).

**"FileTag" tábla esetében:**

* **file (PK, FK)** → a fájl azonosítója (kapcsolat a File táblához).
* **tag (PK, FK)** → a címke azonosítója (kapcsolat a Tag táblához).

## ****Mire jó a kapcsolótábla?****

**M:N kapcsolatok kezelése** – Az adatbázisok nem támogatnak közvetlenül sok-sok kapcsolódást, ezért **kapcsolótáblát kell létrehozni**.  
**Rugalmasság növelése** – Később új mezőket adhatunk hozzá (pl. a fájlmegosztásnál a hozzáférési jogokat).  
**Redundancia elkerülése** – Nem kell duplikálni az adatokat egyetlen táblában.