

# Adatbázis kezelés I.

## Módosítások

Rostagni Csaba

2023. április 24.

# Ezen az órán... I

## 1 Megszorítások

# Tartalom I

## 1 Megszorítások

- Elsődleges kulcs
- Idegen kulcs
- Egyedi értékek UNIQUE

# MySQL CONSTRAINTS (megszorítások)

MySQL-ben többféle megszorítás is létezik:

- **DEFAULT**: Alapértelmezett érték megadása
- **NOT NULL**: Adott mező értékei nem vehetik fel a NULL-t
- **UNIQUE**: Az adott mezőben nem szerepelhet kétszer ugyanaz
- **PRIMARY KEY**: Elsődleges kulcs
- **FOREIGN KEY**: Idegen kulcs
- **CHECK**: Adat egyszerű validálása

A megszorításokat hozzáadhatjuk a táblához már létrehozáskor (CREATE), vagy később is a tábla módosításával (ALTER). Utóbbi esetén a már beszúrt adatok okozhatnak problémát

# Tartalom

- 1 Megszorítások
  - Elsődleges kulcs
  - Idegen kulcs
  - Egyedi értékek UNIQUE

# Elsődleges kulcs

- Az elsődleges kulcs egyértelműen meghatároz egy rekordot (sort) a táblában.
- Az elsődleges kulcs állhat több mezőből, ekkor **összetett kulcsról** beszélünk.
- Egy táblának csak egy elsődleges kulcsa lehet!
- Kitöltése kötelező! Amennyiben nem adjuk meg, hogy legyen NOT NULL ezt az adatbázis kezelő hozzáteszi a háttérben.

<https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

# Elsődleges kulcs megadása

- Elsődleges kulcs megadás a mező tulajdonságként:

MySQL

```
CREATE TABLE `pelda` (  
  `id` INT NOT NULL PRIMARY KEY  
);
```

- Elsődleges kulcs megadás a mezők felsorolása után:

MySQL

```
CREATE TABLE `pelda` (  
  `id` INT NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

A fenti két utasítás ugyanazt eredményezi!

# Összetett (elsődleges) kulcs hibás példa

Az elsődleges kulcs állhat több mezőből, ekkor **összetett kulcs**ról beszélünk.

```
CREATE TABLE `pelda` (  
  `nev` VARCHAR(20) NOT NULL PRIMARY KEY,  
  `kor` INT(11) NOT NULL PRIMARY KEY,  
  `cim` VARCHAR(20)  
);
```

Szintaktikai hiba

Közvetlen tulajdonságként nem tudjuk megadni.

Többszörös elsődleges kulcs definiálás.

Hibakód: #1068



# Összetett (elsődleges) kulcs helyes példa

A mezők felsorolása után adhatjuk meg

MySQL

```
CREATE TABLE `pelda` (  
  `nev` VARCHAR(20) NOT NULL,  
  `kor` INT(11) NOT NULL,  
  `cim` VARCHAR(20),  
  PRIMARY KEY (`nev` , `kor`)  
);
```

# Tartalom

- 1 Megszorítások
  - Elsődleges kulcs
  - Idegen kulcs
  - Egyedi értékek UNIQUE

# Hivatkozási integritás

- A normalizálás során az adatok több táblába kerülhetnek.
- A hivatkozási integritás akkor áll fent, ha igaz, hogy ha az egyik táblából hivatkozunk egy másik táblára, akkor ez csak létező elemre történhessen.
- A hivatkozási integritást az idegen kulcsok biztosítják.
- A megszorítás következményei
  - Új rekord hozzáadása egy kapcsolt táblához csak akkor lehetséges, ha az elsődleges táblában már létezik egy ugyanolyan rekord.
  - Az elsődleges tábla elsődleges kulcsának értékét nem módosíthatjuk, amíg a kapcsolt táblában (másodlagos tábla) tartoznak hozzá adatok.
  - Nem lehet az elsődleges táblából olyan rekordot törölni, amelyhez tartoznak rekordok a másodlagos, kapcsolódó táblában

## Linkek:

- [http://www.nemesvamosiskola.hu/dokumentumok/m5/m5\\_elmelet.pdf](http://www.nemesvamosiskola.hu/dokumentumok/m5/m5_elmelet.pdf)
- [https://en.wikipedia.org/wiki/Referential\\_integrity](https://en.wikipedia.org/wiki/Referential_integrity)

# FOREIGN KEY

## Definition

Az olyan attribútum(ok)at, amelyek egy másik relációban alkotnak kulcsot idegen kulcsnak nevezzük.

```
[CONSTRAINT [symbol]] FOREIGN KEY  
[index_name] (col_name, ...)  
REFERENCES tbl_name (col_name,...)  
[ON DELETE reference_option]  
[ON UPDATE reference_option]
```

reference\_option:

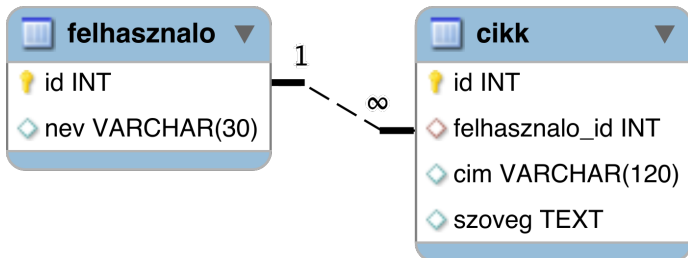
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

<https://dev.mysql.com/doc/refman/8.0/en/create-table-foreign-keys.html>

# Idegenkulcs opciói

- RESTRICT** Nem engedi a törlést/módosítást (ez az alapértelmezett)
- NO ACTION** MySQL-ben megegyezik a RESTRICT hatásával (SQL szabvány)
- CASCADE** A szülő táblában történő törlést/módosítást végrehajtja a gyerek táblán is
- SET NULL** A szülő táblában történő törlés/módosítás esetén a gyerek táblában NULL-ra állítja az értéket. Nem lehet NOT NULL az érték, mert az gondot okozna!
- SET DEFAULT** InnoDB és az NDB is figyelmen kívül hagyja (van, de minek)

# Idegenkulcs példa (blog és szerző) I.



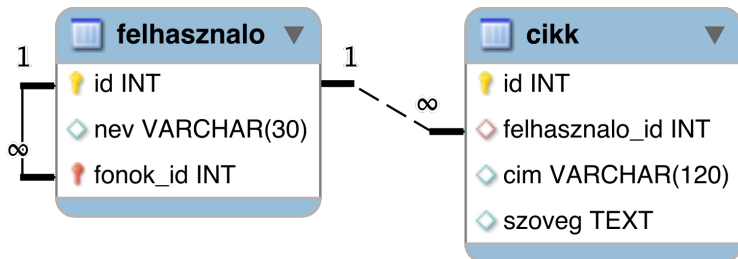
## Idegenkulcs példa (blog és szerző) II.

MySQL

```
ALTER TABLE `cikk`  
  ADD constraint `fk_cikk_felhasznalo_id`  
    FOREIGN KEY (`felhasznalo_id`)  
    REFERENCES `felhasznalo` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE;
```

- A felhasználó törlése esetén a bejegyzés is törlésre kerül

# Idegenkulcs példa (szerző és főnöke) I.





## Idegenkulcs példa (szerző és főnöke) II.

MySQL

```
ALTER TABLE `felhasznalo`  
  ADD COLUMN `fonok_id` INT,  
  ADD CONSTRAINT `fk_felhasznalo_fonok_id`  
    FOREIGN KEY (`fonok_id`)  
    REFERENCES `felhasznalo` (`id`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT;
```

- A főnök törlését nem engedni, amíg van hozzá tartozó beosztott dolgozó

# Idegenkulcs törlése

- Az idegen kulcs törlésénél jól jön, ha tudjuk hogy milyen nevet adtunk neki!

**MS SQL, ORACLE**

```
ALTER TABLE felhasználó  
DROP CONSTRAINT fk_fonok_id;
```

**MySQL**

```
ALTER TABLE felhasználó  
DROP FOREIGN KEY fk_fonok_id;
```

# Idegen kulcs által okozott hibák

**Error Code: 1451** Cannot delete or update a parent row: a foreign key constraint fails.

(A megadott sor(ok) nem törölhető(ek) idegen kulcs miatt)

**Error Code: 1217** Cannot delete or update a parent row: a foreign key constraint fails

(A tábla nem törölhető idegen kulcs miatt)

## Megoldás:

- Ellenőrzés kikapcsolása: **SET** foreign\_key\_checks = 0;
- Ellenőrzés visszakapcsolása: **SET** foreign\_key\_checks = 1;

# Tartalom

- 1 Megszorítások
  - Elsődleges kulcs
  - Idegen kulcs
  - Egyedi értékek UNIQUE

# Egyedi mező (UNIQUE)

MySQL

```
ALTER TABLE `felhasznalo`  
  ADD UNIQUE(`email`);
```

- Az adott oszlopban ugyanaz az érték kétszer nem szerepelhet!
- Több mező megadásakor csak akkor számít azonosnak két sor, ha mindegyik felsorolt mező megegyezik.