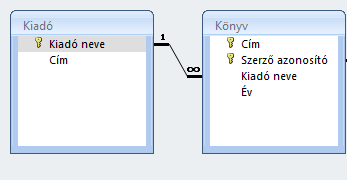
# Táblák közötti kapcsolatok

Azért, hogy megszüntessük az adatok redundanciáját (felesleges ismétlődését), az adatokat több táblába kell osztani, hogy egy adat csak egyszer legyen rögzítve. Ehhez azonban kell egy olyan mező, amelynek használatával az egyik tábla rekordjait össze tudjuk kapcsolni a másik tábla rekordjaival.

A közös mezőként használt mező a forrástáblában az elsődleges kulcs, a kapcsolt táblában az idegen kulcs. Az idegen kulcsra indexet kell létrehozni.



Pl.: A Kiadó táblában a Kiadó neve elsődleges kulcs, ugyanez a mező a Könyv táblában idegen kulcs.

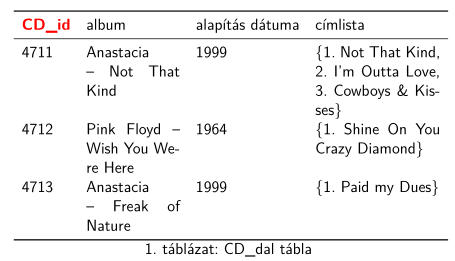
A táblák közötti kapcsolatok típusai:

* Egy az egyhez (1:1): Az első tábla minden rekordjához egyetlen rekord tartozik a másik táblában.
* Egy a többöz (1:N): Az első tábla rekordjaihoz több rekord is tartozhat a másik tábla rekordjai közül. Ez a leggyakoribb.
* Több a többhöz (N:N): Az első tábla minden rekordjához több rekord tartozhat a másik tábla rekordjai közül, és ez fordítva is igaz. Csak kapcsolótábla közbeiktatásával lehet megvalósítani.

# Adatbázisok normalizálása

A relációs adatbázisok tervezésének kialakult egy „normalizálás” elnevezésű módszere. Ennek célja, hogy az adatbázisok a lehető legkevesebb redundanciát (feleslegesen ismétlődő adatot) tartalmazzanak. Ez egy többlépcsős folyamat, amelyben az adatbázis különböző „normálformákba” kerül.

Példa: Zenei CD gyűjteményünket szeretnénk egy adatbázisban tárolni. Tárolni szeretnénk az albumok előadóját és címét, az előadó együttes alapításának évét (egyéni énekes esetén a zenei működésének kezdetét) és az albumok egyes számait sorrendben.



Pirossal kiemelve jelöljük az elsődleges kulcsokat. Példánkban ez a CD\_id mező, amely az egyes albumok egyedi azonosítója.

## Első normálforma (1NF)

Az első normálformában:

* az egyes mezőknek tovább nem oszthatóaknak (elemieknek) kell lenniük. Ha az album mező tartalmazza az előadót és az album nevét is, akkor nem tudunk csak az egyik szerint keresni. Az, hogy mi számít eleminek, az a feladattól is függ.
* Az egyes mezők nem tartalmazhatnak felsorolást (nem lehetnek többértékű mezők).

Az első normálformára hozott tábla:  


Megjegyzések:

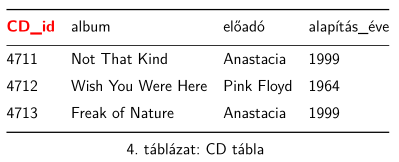
* Az album címét és az előadót külön mezőbe tettük, hogy elemiek legyenek az adatok.
* Az utolsó mezőben levő lista egyes elemeit külön-külön sorban kell szerepeltetni úgy, hogy a többi adat azonos marad.
* Ha az egyes dalok sorrendjét is rögzíteni szeretnénk, akkor egy új mezőt kell hozzáadni a táblához, amely a sorszámot (a sáv sorszámát) tartalmazza. Ebben az esetben a CD azonosítószáma (CD\_id) önmagában már nem lesz kulcs, hiszen több sorban azonos az értéke. A (CD\_id, sáv) mezőpárt célszerű elsődleges kulcsnak választani.

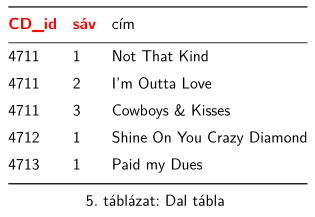
## Második normálforma (2NF)

Kritériumai:

* legyen az adatbázis első normálformában (1NF), és
* minden nemkulcs mező teljes függőségben álljon a kulcstól (vagyis a kulcs mezők egyértelműen meghatározzák a többi mezőt).

Példánkban az album címe csak az egyik kulcsmezőtől (a CD\_id mezőtől) függ, a másiktól (a sávtól) nem. Ezt úgy lehet megoldani, hogy a táblát két külön táblára bontjuk:





## Harmadik normálforma (3NF)

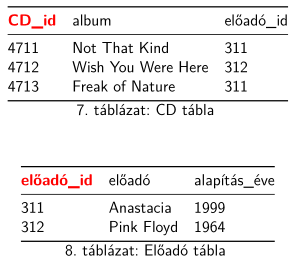
Kritériumai:

* az adatbázis második normálformában van (2NF), és
* ne legyen az adatbázisban tranzitív függőség.

Tranzitív függőség esetén egy táblában egy tulajdonság nem közvetlenül az elsődleges kulcstól, hanem olyan tulajdonságtól függ, amelyet a kulcs határoz meg.

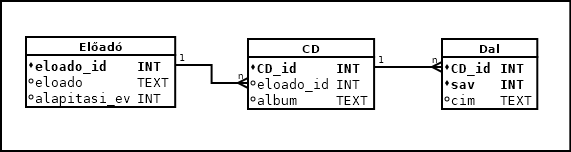
A Dal táblában nincs tranzitív függőség, de a CD táblában az alapítás éve az előadó mezőtől függ (az pedig a kulcstól), tehát itt tranzitív függőség van.

A megoldás a CD tábla újbóli kettéosztása:



Megjegyzés: Az előadó\_id bevezetése helyett az előadó nevét is használhatnánk elsődleges kulcsként az Előadó táblában, de általában célszerűbb egy azonosító számot alkalmazni. Ez kevesebb helyet foglal, mint egy szöveg típusú mező, és gyorsabban lehet vele dolgozni.

A táblák közötti kapcsolatokat szemlélteti a következő ábra:



Az ábra az ingyenes DIA programmal készült: <http://dia-installer.de/>

Használhatod a [draw.io](https://www.draw.io/) programot is. Az alakzatokat az Entity Relation csoportban találod. Sőt, a +/Advanced/SQL parancsot választva a táblákat létrehozó SQL parancsokból is elkészítheted a táblák dobozait. (<https://desk.draw.io/support/solutions/articles/16000082007-how-to-create-an-entity-relation-diagram-> )

# Megjegyzések

Az adatbázis tervezése előtt fel kell mérni az igényeket:

* beszélni kell a felhasználókkal,
* megnézni az eddig használt űrlapokat, számlákat,
* megnézni az eddig használt megoldásokat.

A többi adatból kiszámítható mezőket ne vegyünk fel!

Az adatokat védeni kell a következő szabályok betartásával:

* Az elsődleges kulcs soha nem lehet üres (NULL). Több mezőből álló kulcs esetén egyik sem.
* Minden idegen kulcs kapcsolódjon egy elsődleges kulcshoz a másik táblában. Erre figyelni kell az adatbevitelnél, illetve az elsődleges kulcs módosításánál, törlésénél is!
* Ellenőrizzük a bevitt adatokat!