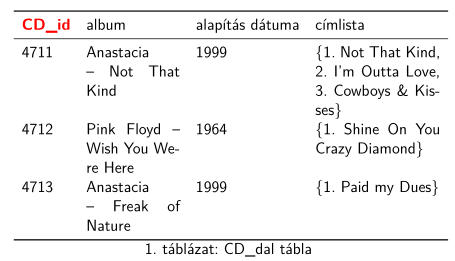
# Adatbázisok normalizálása

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, amelynek során 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 (primary key), amelyek egyértelműen azonosítják a rekordokat. Példánkban ez a CD\_id mező, amely az egyes albumok egyedi azonosítója.

Több tábla esetén az elsődleges kulcshoz kapcsolódó mezőt a másik táblában idegen kulcsnak nevezik. (foreign key)

## 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. Persze, 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:

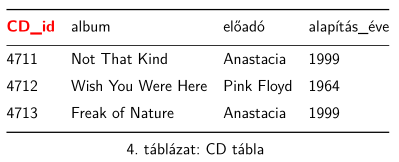
* Az album címét és az előadót külön mezőbe tettük, hogy elemiek legyenek az adatok.
* Az utolsó mezőben levő lista egyes elemeit külön-külön sorban kell szerepeltetni úgy, hogy a többi adat azonos marad.
* Ha az egyes dalok sorrendjét is rögzíteni szeretnénk, akkor egy új mezőt kell hozzáadni a táblához, 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.

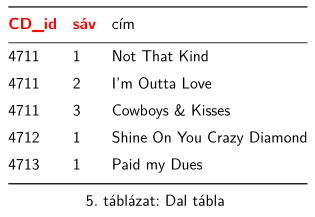
## Második normálforma (2NF)

Kritériumai:

* legyen az adatbázis első normálformulában (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)

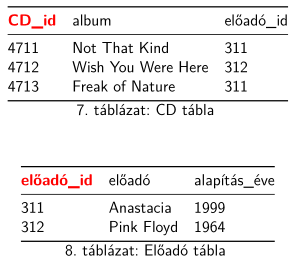
Kritériumai:

* az adatbázis második normálformátumban 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.

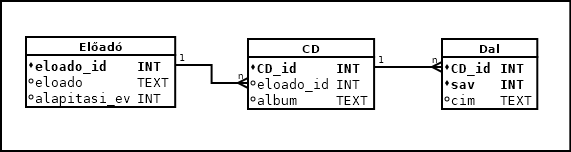
A Dal táblában nincs tranzitív függőség, de a CD táblában az alapítás éve az előadó mezőtől függ (az pedig a kulcstól), tehát itt tranzitív függőség van.

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/>

# 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!

A táblák között általában egy a többhoz kapcsolat van. Egy az egyhez kapcsolat esetén egy táblába össze lehet vonni az adatokat. Több a többhöz kapcsolat nem valósítható meg közvetlenül, kapcsolótáblát kell alkalmazni.

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!