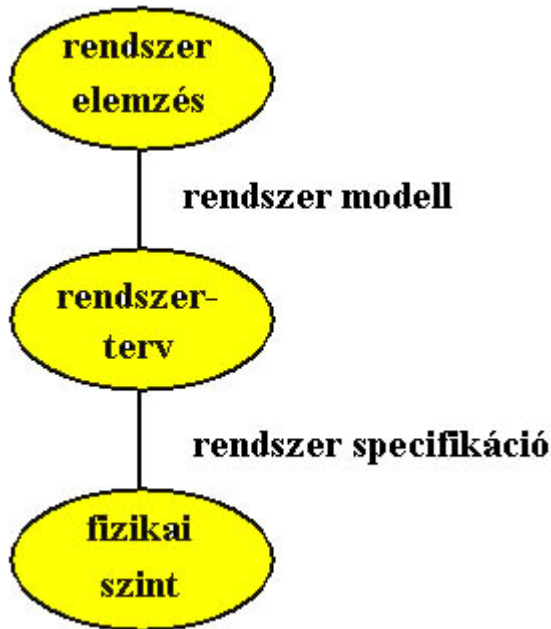


# **Adatbázis-kezelő rendszerek**

**II.**

dr. Siki Zoltán

# Adatbázis tervezés



Interjúk sorozata a leendő felhasználókkal, megrendelőkkel

Adatbázis logikai sémájának dokumentálása

Az adatbázis fizikai megvalósítása, rendszer kiválasztása, hatékonysági kérdések

Relációs adatbázisokban a logikai modell közvetlenül leképezhető  
A logikai tervezés során nem kell ismerni a később alkalmazott RDBMS-t illetve hardvert

# Adatbázis tervezés célja

Az adatbázis tervezés fő célja a redundancia mentes adattárolás kialakítása

Redundancia – ugyanannak a ténynek többszörös tárolása

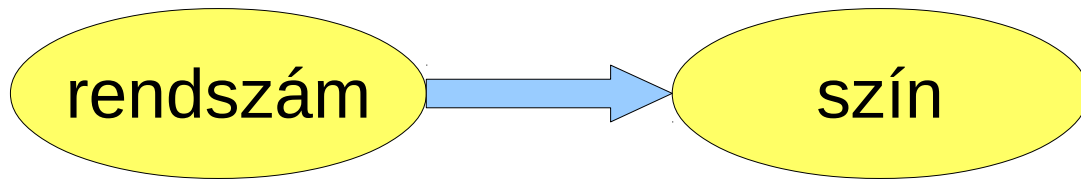
A redundanciát tartalmazó adatbázis a felhasználók hibájából könnyen inkonzisztenssé válhat

Inkonzisztencia – az adatbázis egymásnak ellentmondó tényeket tartalmaz

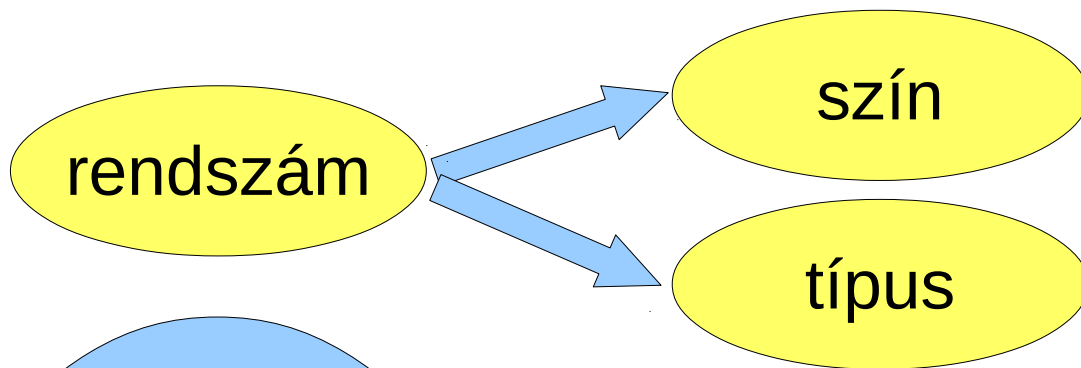
Könyvtári kartotékban redundancia:  
több szerző esetén több karton az ABC-re rendezett katalógusban  
Téma szerint katalógus megismétli a kartonokat

# Funkcionális függőség

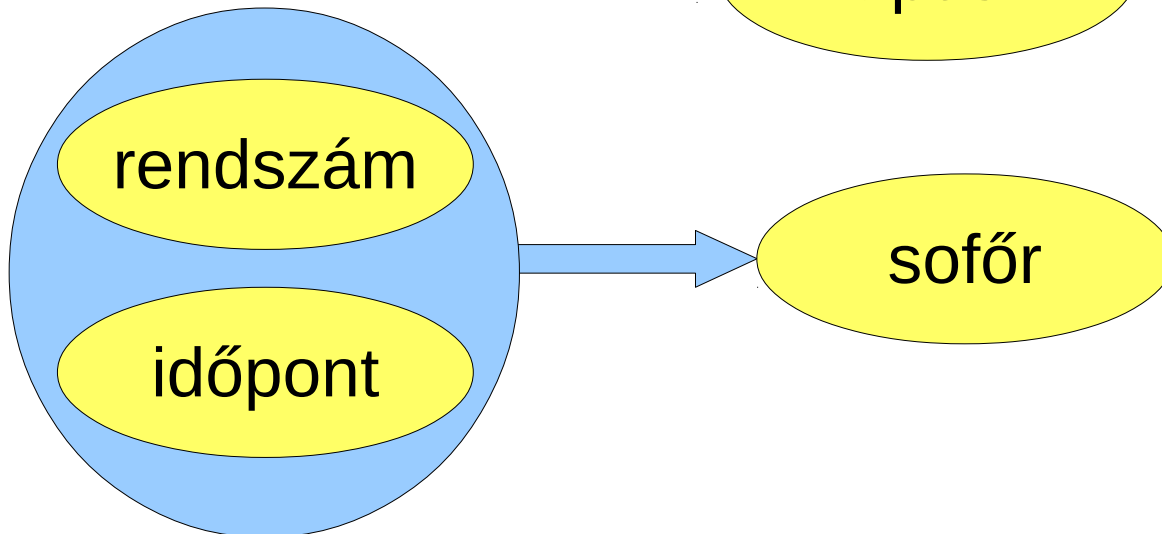
Az egyik adat értékéből a másik adat egyértelműen következik



rendszer → szín



rendszer → szín, típus



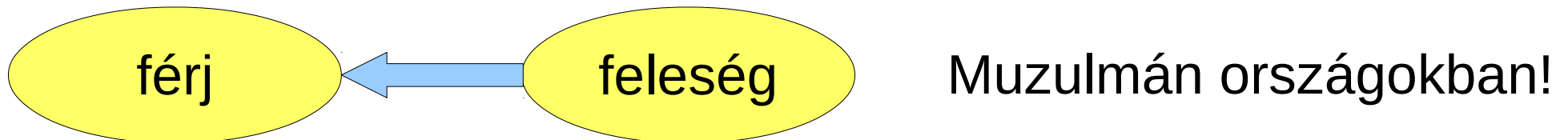
Rendszer, időpont → sofőr

# Funkcionális függőség folyt.

Kölcsönös funkcionális függőség



A földrajzi helytől, a pillanatnyi törvényektől függhet, hogy funkcionális függőség van-e az adatok között



# Többértékű függőség

Az egyik adat értékéből a másik adat több értéke következik



A funkcionális függőségek a másik irányban többértékű függőségek, kivéve a kölcsönös funkcionális függőségek

A többértékű függőségek csak akkor lényegesek a tervezés szempontjából, ha nincs funkcionális kapcsolat az adatok között

# Reláció kulcs

A reláció kulcs a reláció egy sorát azonosítja egyértelműen

- az attribútumok egy olyan csoportja, melyek csak egy sort azonosítanak (egyértelműség)
- a kulcsban szereplő attribútumok egyetlen részhalmaza sem alkot kulcsot (minimális halmaz)
- a kulcsban szereplő attribútumok értéke nem lehet definiálatlan (NULL)

A NULL érték speciális érték, mely azt jelenti, hogy nem ismerjük az értéket, nem azonos a 0 (nulla) értékkel

Egy relációnak több kulcsa is lehet

# Külső (idegen) kulcs

Két reláció közötti kapcsolat, az egyik reláció kulcsát alkotó attribútumok megjelennek egy másik táblában, ahol nem alkotnak kulcsot

Neptun	Név	email
kulcs		

Neptun	Tárgy	oszt
külső		
kulcs		





# Redundancia

Ugyanannak a ténynek többszörös tárolása az adatbázisban

Tanár	Tantárgy	Össz_óraszám	Tanított_órák
Kiss Péter	Adatbázis kezelés	64	12
Nagy Andrea	Matematika	32	8
Szabó Miklós	Adatbázis kezelés	64	4
Kovács Rita	Matematika	32	5
	Angol	48	

a tantárgyak összes óraszámát  
annyiszor tároljuk, ahány tanár  
tanítja az adott tantárgyat

Ha egy tantárgy összóraszámja megváltozik több helyen kell módosítani a relációban. Valahányszor egy új tanár kerül be a relációba ugyanannak a tantárgynak az előző soraiból kell elővenni az összóraszám adatot.

Az utolsó sorban szereplő tantárgy (angol) esetén még nem került kitöltésre a tanár személye.

Új tanárnak a listára történő felvételekor ezt az esetet másként kell kezelni. Ilyenkor csak két üres értéket (tanár, tanított órák) kell átírni.

Levezetett adatok tárolása is redundanciához vezet  
pl. darab, egységár és összérték (darab \* egységár) tárolása