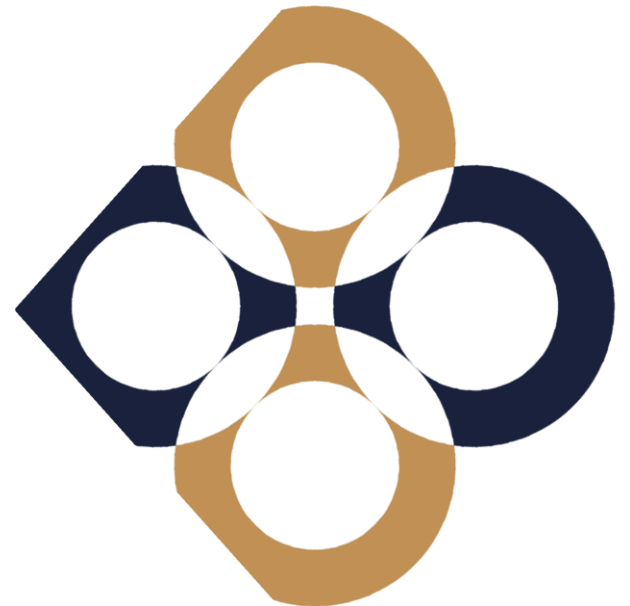


# Adatbázisok gyakorlat 02

*SQL alapismeretek II.*

*Összesítés. Csoportosítás. Rendezés*

*Táblák összekapcsolása*



# Összesítés (aggregálás)

Az összesítő függvények értékek egy halmazán végeznek számítást, és egyetlen értéket adnak vissza. Alapesetben a halmaz a tábla összes sorát jelenti

- A számítás egy kifejezés kiértékelését jelenti
- Az összesítő függvények (kivéve: COUNT(\*)) nem veszik figyelembe a NULL értékeket
- Fontosabb összesítő függvények és szerepük:
  - SUM(): egy adott kifejezés értékeit összegét adja vissza
  - AVG(): egy adott kifejezés értékeinek átlagát adja vissza
  - MIN() és MAX(): egy adott kifejezés értékei közül a legkisebbet, illetve legnagyobbban adja vissza
  - COUNT(): egy adott halmaz elemeinek számát adja vissza

# Összesítés - példa

Határozzuk meg a legdrágább és legolcsóbb dvd árát, a dvd-k átlagos árát, a dvd-k árainak összegét és a dvd-k darabszámát egyetlen lekérdezéssel!

```
SELECT MAX(nettoar) AS 'legdrágább',  
       MIN(nettoar) AS 'legolcsóbb',  
       AVG(nettoar) AS 'átlagár',  
       SUM(nettoar) AS 'összeg',  
       COUNT(*) AS 'db'  
FROM dvd
```

A COUNT(\*) a lekérdezés eredménysorainak számát adja vissza

# Csoportosítás

A GROUP BY részben felsorolt mezők vagy kifejezések szerint csoportokat képezhetünk, és a SELECT részben alkalmazott számításokat ezekre a csoportokra alkalmazhatjuk.

SELECT...	// oszlopok kiválasztása
FROM ...	// táblák kiválasztása
WHERE ...	// szűrőfeltétel megadása a sorokra
<b>GROUP BY ...</b>	<b>// csoportosítás</b>
HAVING ...	// szűrőfeltétel a csoportokra
ORDER BY...	// sorbarendezés

A csoportosítás során a SELECT részben gyakran összesítő függvényeket használunk

# Csoportosítás - példa

Készítsünk lekérdezést, amely megjeleníti, hogy melyik stílusú dvd-ből hány darab van összesen raktáron!

```
SELECT stilus AS 'Stílus',  
       SUM([raktár készlet]) AS 'db'  
FROM dvd  
GROUP BY stilus
```

- Ha egy oszlop neve több szóból áll, akkor tegyük szögletes zárójelek közé!
- A GROUP BY részben nem használhatjuk az AS kulcsszó után adott oszlopneveket (alias neveket)!

# Csoportosítás – fontos megjegyzés

Csoportosítás esetén a **SELECT** részben lévő oszlopoknak szerepelniük kell a **GROUP BY** felsorolásában, vagy egy összesítésben (mint az összesítő függvény paramétere)

Helyes lekérdezések

```
SELECT cim, stilus,  
       MIN(nettoar) AS  
       'MinÁr'  
FROM dvd  
GROUP BY cim, stilus
```

```
SELECT cim+'-'+stilus,  
       COUNT(*)  
FROM dvd  
GROUP BY cim+'-'+stilus
```

Helytelen lekérdezések

```
SELECT cim, stilus, szam,  
       MIN(nettoar) AS  
       'MinÁr'  
FROM dvd  
GROUP BY cim, stilus
```

```
SELECT cim, stilus,  
       COUNT(*)  
FROM dvd  
GROUP BY cim+stilus
```

# Csoportosítás – Szűrőfeltétel a csoportokra

A HAVING kulcsszó után megadhatunk egy vagy több szűrőfeltételt, amelyek a csoportokra vonatkoznak. Itt sem használhatjuk az AS kulcsszó után megadott oszlopneveket.

SELECT...	// oszlopok kiválasztása
FROM ...	// táblák kiválasztása
WHERE ...	// szűrőfeltétel megadása a sorokra
<b>GROUP BY ...</b>	<b>// csoportosítás</b>
<b>HAVING ...</b>	<b>// szűrőfeltétel a csoportokra</b>
ORDER BY...	// sorbarendezés

## Csoportosítás - szűrőfeltétel csoportra - példa

Készítsünk lekérdezést, amely megjeleníti, hogy melyik stílusú dvd-ből hány darab van összesen raktáron! A listában csak azok a stílusok szerepeljenek, ahol ez az összeg 20 felett van! Az 5000 Ft alatti dvd-ket hagyjuk ki a listából!

```
SELECT stilus AS 'Stílus',  
       SUM([raktár készlet]) AS 'db'  
FROM dvd  
WHERE nettoar >= 5000  
GROUP BY stilus  
HAVING SUM([raktár készlet]) > 20
```

A WHERE utáni szűrés a sorokra vonatkozik, a HAVING utáni pedig a csoportokra, ezek egymással kombinálva is használhatók



# Rendezés

SELECT...	// oszlopok kiválasztása
FROM ...	// táblák kiválasztása
WHERE ...	// szűrőfeltétel megadása a sorokra
GROUP BY ...	// csoportosítás
HAVING ...	// szűrőfeltétel a csoportokra
<b>ORDER BY...</b>	<b>// sorbarendezés</b>

A lekérdezés eredményét sorba rendezi a megadott szempontok alapján

- A rendezésnél megadhatjuk az egyes oszlopok nevét (az AS kulcsszó után megadott név is használható) vagy sorszámát
- Növekvő rendezés esetén az ASC, csökkenő rendezés esetén a DESC kulcsszót használhatjuk a rendezés irányának megadására
- Alapértelmezett a növekvő rendezés, ilyenkor az ASC kulcsszó elhagyható

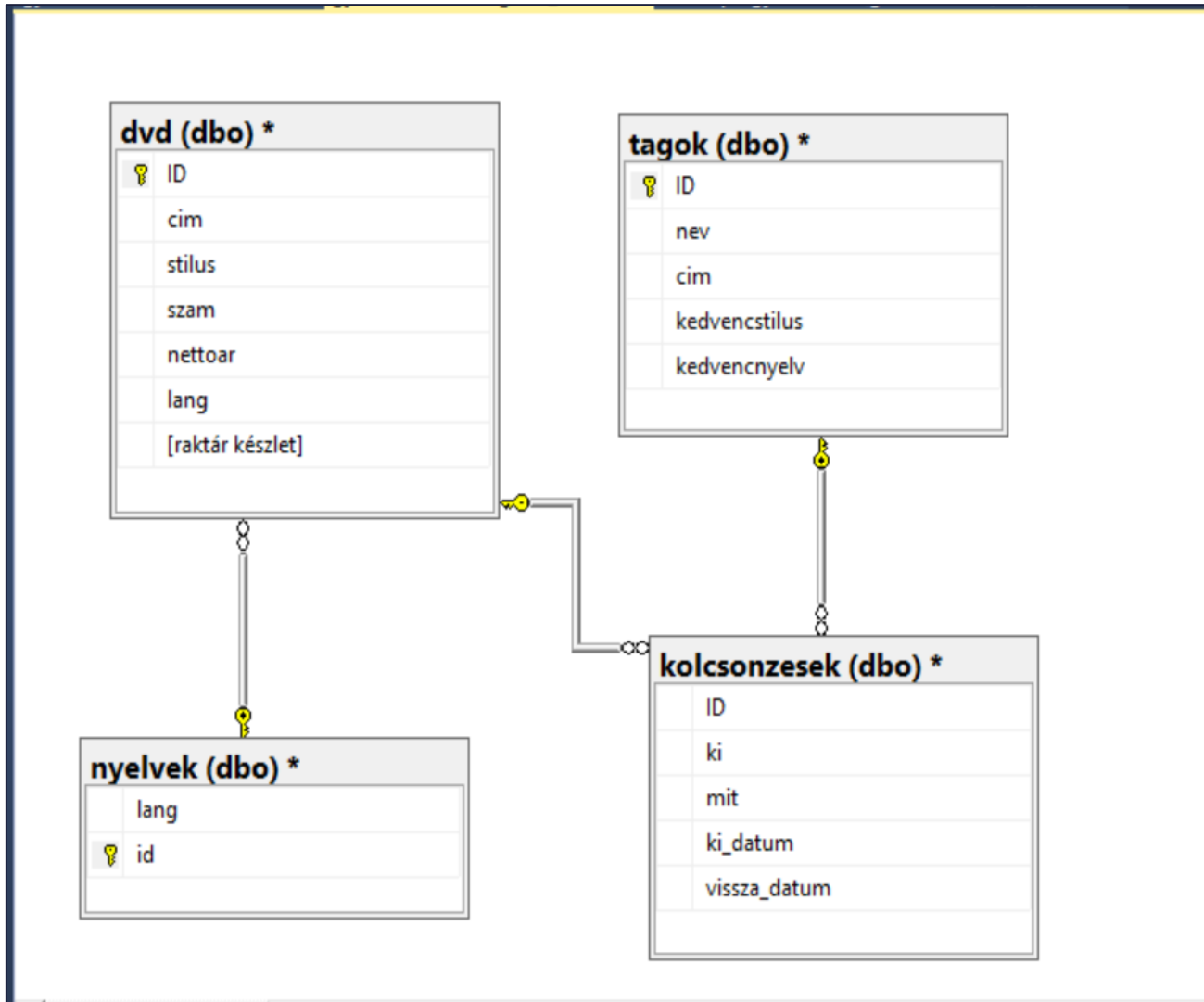
# Rendezés - példa

Rendezzük a dvd tábla sorait stílus szerint növekvő, azon belül nettó ár szerint csökkenő sorrendbe!

```
SELECT *  
FROM dvd  
ORDER BY stilus ASC, nettoar DESC
```

**Ha egy lekérdezésben csoportosítást is használunk, akkor az ORDER BY utáni felsorolásban lévő oszlopoknak benne kell lenniük a GROUP BY listában vagy egy összesítésben (mint fv. paraméter)**

# A gyakorlaton használt adatbázis - DVD



dvd tábla	
ID	A dvd azonosítója
cim	A dvd címe
stilus	A dvd stílusa
nettoar	A dvd nettó ára
lang	A dvd nyelve
raktár készlet	A raktárkészlet mennyisége az adott dvd-ből

kölcsönzések tábla	
ID	A kölcsönzés azonosítója
ki	A kölcsönző tag azonosító
mit	A kölcsönzött dvd azonosító
ki_datum	A kölcsönzés dátuma
vissza_datum	A visszahozatal dátuma

# Táblák összekapcsolása

- Akkor lehet rá szükség, ha a keresett információ egynél több táblában található meg
- **A kapcsolat alapja legtöbbször az egyik tábla idegen kulcsának és a másik tábla kulcsának azonos értéke**, ritkább esetben a két tábla egy-egy oszlopára vonatkozó összehasonlító feltétel
- Az összekapcsolás megvalósításának tipikus módja:  
**SELECT** oszlopnév\_lista  
**FROM** táblanév1 aliasnév1  
**(INNER) JOIN** táblanév2 aliasnév2  
**ON** aliasnév1.idegenkulcsérték = aliasnév2.kulcsérték

# Táblák összekapcsolásának tipikus módja – Példa

Listázzuk az órák azonosítóját, a sávok nevét és időpontját!

ORAK tábla
ora_id
tanar
nap
sav
targy
terem

sav: Idegen kulcs

SAVOK tábla
Sav_id
Sav
Időpont

Sav\_id: Elsődleges kulcs

```
SELECT o.ora_id, s.Sav, s.Időpont FROM  
Orak o  
    INNER JOIN Savok S  
    ON o.sav = s.Sav_id
```

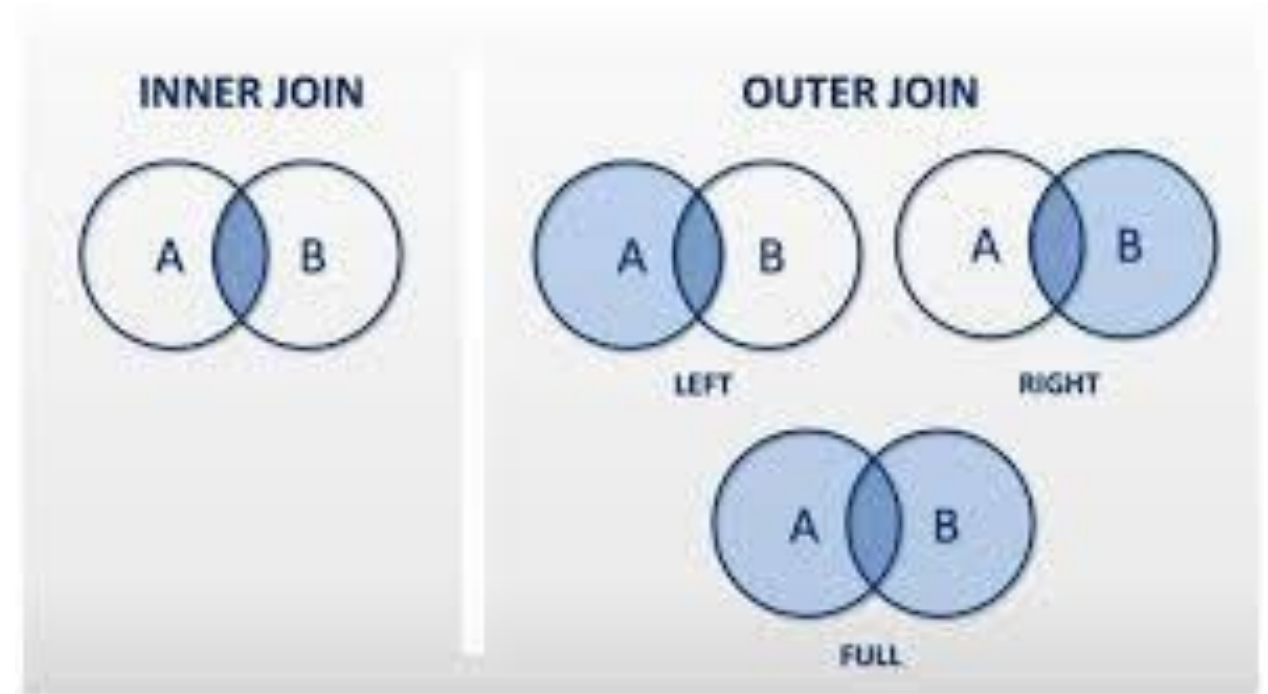
# Táblák összekapcsolása - Megjegyzések

- Az aliasnevek (másodlagos táblanevek, rövid nevek) használata javítja a lekérdezés olvashatóságát\*
- Egy tábla akár saját magával is összekapcsolható, ilyenkor az aliasnevek használata kötelező
- Mivel a két táblának lehetnek azonos nevű mezői is, ezért a SELECT utáni oszlopnév listában használjunk minősített oszlopneveket (tábla aliasnév.oszlopnév) vagy aliasnevek hiányában táblanév.oszlopnév formában
- Az INNER kulcsszó használata nem kötelező
- Az ON kulcsszó után összetett feltétel is szerepelhet
- **Az eredménylistából INNER JOIN esetén kimaradnak azok a sorok, ahol az idegenkulcs = kulcs feltétel valamelyik oldalán NULL érték található, pl: a diák azonosítója nincs megadva a munka táblában**

\*Egy tábla neve a legáltalánosabb esetben [Szervernév].[Adatbázisnév].[Séma név].[Táblanév] formátumú, nagyon hosszú is lehet

# JOIN TÍPUSOK\*

- (INNER) JOIN:  
Az A tábla idegen kulcsa megegyezik a B tábla kulcsával
- LEFT (OUTER) JOIN:  
Az INNER JOIN eredményéhez hozzá veszi az A tábla minden további sorát is
- RIGHT (OUTER) JOIN:  
Az INNER JOIN eredményéhez hozzá veszi az B tábla minden további sorát is
- FULL (OUTER) JOIN:  
Az INNER JOIN eredményéhez hozzá veszi az A és B tábla minden további sorait is



Az OUTER szó használata nem kötelező

\* A CROSS JOIN-nal és a táblák összekapcsolásának egyéb módjaival (pl. OUTER APPLY) nem foglalkozunk

# JOIN TÍPUSOK- Példa

MUNKA tábla		
MHelyID	DiakAz	Allas
1	1	Pénztáros
2	NULL	Takarító

DIAK tábla		
DiakAz	Nev	Szulido
1	Kiss Béla	1998.01.01
3	Tóth Ottó	1997.12.23

SELECT m.MhelyId, m.DiakAz AS 'DiakAZ in m',  
FROM munka m INNER JOIN diak d ON m.DiakAz =

d.DiakAz AS 'DiakAz in d', d.Nev  
d.DiakAz

SELECT m.MhelyId, m.DiakAz AS 'DiakAZ in m', FROM  
munka m LEFT JOIN diak d ON m.DiakAz =

d.DiakAz AS 'DiakAz in d', d.Nev  
d.DiakAz

SELECT m.MhelyId, m.DiakAz AS 'DiakAZ in m', FROM  
munka m RIGHT JOIN diak d ON m.DiakAz =

d.DiakAz AS 'DiakAz in d', d.Nev  
d.DiakAz

SELECT m.MhelyId, m.DiakAz AS 'DiakAZ in m',  
FROM munka m FULL JOIN diak d ON m.DiakAz =

d.DiakAz AS 'DiakAz in d', d.Nev  
d.DiakAz



# JOIN TÍPUSOK- Példa - Eredmény sorok

MUNKA tábla		
MHelyID	DiakAz	Allas
1	1	Pénztáros
2	NULL	Takarító

**INNER JOIN**

DIAK tábla		
DiakAz	Nev	Szulido
1	Kiss Béla	1998.01.01
3	Tóth Ottó	1997.12.23

Mhelyld DiakAZ in m DiakAz in d  
1 1 1

Nev  
**Kiss Béla**

Mhelyld DiakAZ in m DiakAz in d  
1 1 1  
2 NULL NULL

Nev  
**Kiss Béla**  
NULL

Mhelyld DiakAZ in m DiakAz in d  
1 1 1  
NULL NULL 3

Nev  
**Kiss Béla**  
Tóth Ottó

**RIGHT JOIN**

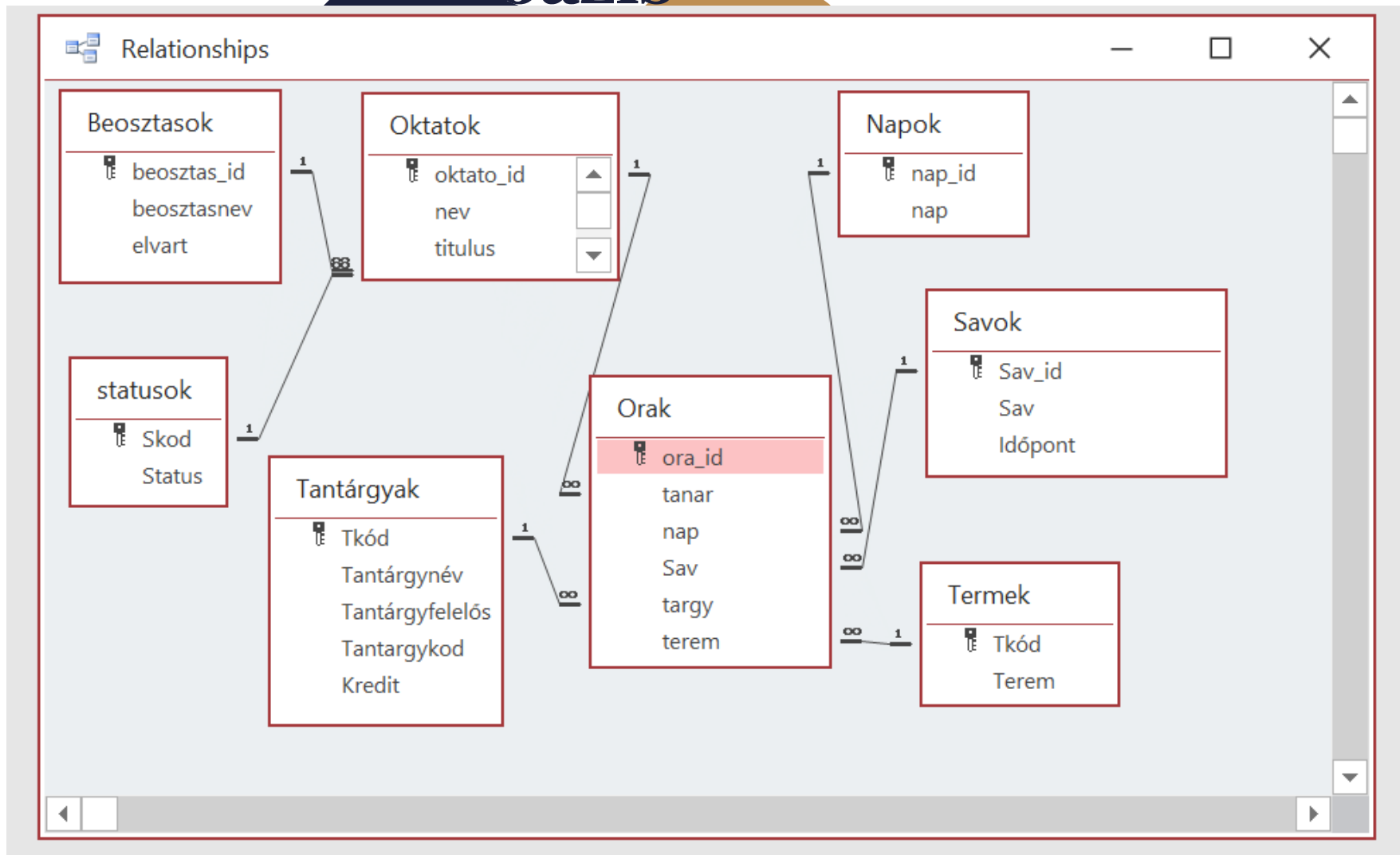
**LEFT JOIN**

**FULL JOIN**

Mhelyld DiakAZ in m  
1 1  
2 NULL  
NULL NULL

DiakAz in d Nev  
1 **Kiss Béla**  
NULL NULL  
3 Tóth Ottó

# A gyakorlaton használt tanulmányi adatbázis



# Példák – tanulmányi adatbázis

1. Készítsünk listát arról, hogy melyik órán ki a tanár! Adjuk meg az órák azonosítóját és a tanár nevét és titulusát!

```
SELECT o.ora_id, okt.nev, okt.titulus FROM Orak  
o JOIN Oktatok okt  
ON o.tanar = okt.oktato_id
```

2. Készítsünk listát az egyes oktatók beosztásáról! A listában szerepeljenek azok a beosztások is, amelyekhez nem tartozik rekord az Oktatok táblában! Jelenítsük meg az oktatók nevét és a beosztásuk nevét!

```
SELECT o.nev, b.beosztasnev  
FROM Oktatok o RIGHT JOIN Beosztasok b  
ON o.beosztas = b.beosztas_id
```

3. Mely órák kezdődnek a 8.00-s sávban? Csak az órák azonosítóit jelenítsük meg!

```
SELECT o.ora_id  
FROM Orak o JOIN Savok s ON  
o.Sav = s.Sav_id  
WHERE s.időpont = '8.00'
```

# Példák – tanulmányi adatbázis

4. Listázzuk a tanórák azonosítóját, a tanár és a tantárgy, valamint a terem nevét is! Szűrjünk csak a keddi és szerdai napokra!

```
SELECT o.ora_id, okt.nev, tt.Tantárgynév, t.Terem
FROM Orak o JOIN Oktatok okt ON o.tanar = okt.oktato_id JOIN
      Tantárgyak tt ON o.targy = tt.Tkód JOIN Termek t ON
      o.terem = t.Tkód
      JOIN Napok n ON o.nap = n.nap_Id WHERE
n.nap IN ('Kedd', 'Szerda')
```

5. Készítsünk lekérdezést, amely megjeleníti azon tanár párosításokat, akiknek ugyanaz a státuszuk és beosztásuk! Jelenítsük meg a tanárok neveit, valamint a státusz és beosztás megnevezéseket!

```
SELECT o1.nev, o2.nev, b.beosztasnev, s.status FROM
Oktatok o1
      JOIN Oktatok o2 ON o1.oktato_id <> o2.oktato_id AND
      o1.beosztás = o2.beosztás
      AND o1.status = o2.status
      JOIN Beosztasok b on o1.beosztás = b.beosztas_id JOIN Statusok
s on o1.status = s.skod
```



**Köszönöm  
a figyelmet!**