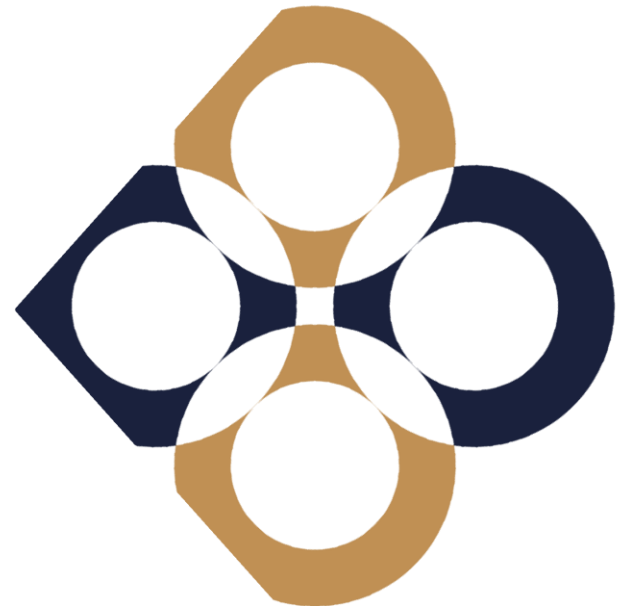


Adatbázisok gyakorlat 03

Halmazműveletek



Lekérdezések uniója

Két vagy több lekérdezés eredményseit fűzi össze egyetlen eredményhalmazzá.

Lekérdezések uniója SQL-
ben:

SELECT oszlopnevek
FROM ...

UNION (ALL)

SELECT oszlopnevek
FROM ...

A lekérdezéseknek azonos szerkezetűeknek kell lenniük (oszlopok száma, sorrendje, típusa azonos, illetve típus kompatibilis)

UNION esetén az eredményhalmazban a duplikált (mindegyik lekérdezésben előforduló) sorok csak egyszer szerepelnek, UNION ALL esetén pedig annyiszor, ahányszor előfordulnak

Lekérdezések metszete

Két vagy több lekérdezés eredményhalmazainak közös sorait adja meg

Lekérdezések metszete SQL-ben:

SELECT oszlopnevek
FROM ...

INTERSECT

SELECT oszlopnevek
FROM ...

A lekérdezéseknek azonos szerkezetűeknek kell lenniük (oszlopok száma, sorrendje, típusa azonos, illetve típus kompatibilis)

Lekérdezések különbsége

Két lekérdezés különbsége azokat a sorokat adja vissza, amelyek az első lekérdezés eredményhalmazában benne vannak, de a másodikban nincsenek

Lekérdezések különbsége SQL-ben:

SELECT oszlopnevek
FROM ...

EXCEPT
SELECT oszlopnevek
FROM ...

A lekérdezéseknek azonos szerkezetűeknek kell lenniük (oszlopok száma, sorrendje, típusa azonos, illetve típus kompatibilis)

Halmazműveletek és oszlopnevek

Amennyiben a halmazműveletekben résztvevő lekérdezések oszlopnevei nem egyformák, abban az esetben az eredményhalmaz oszlopainak nevei az első lekérdezés oszlopnevei lesznek

SELECT beosztas_id
AS 'Beosztás kód'
FROM Beosztasok
**UNION | INTERSECT |
EXCEPT**

Beosztasok.Beosztas_id
5
7
9

Oktatok.beosztás
1
3
9

UNIÓ

Beosztás kód
5
7
9
1
3

METSZET

Beosztás kód
9

KÜLÖNBSÉG

Beosztás kód
5
7

Halmazműveletek és rendezés

Amennyiben a halmazműveletek eredményének sorait rendezni szeretnénk, akkor a rendezést (ORDER BY) mindig a legutolsó lekérdezés végén kell definiálni

SELECT oszlop1, oszlop2...

FROM ...

UNION | INTERSECT | EXCEPT

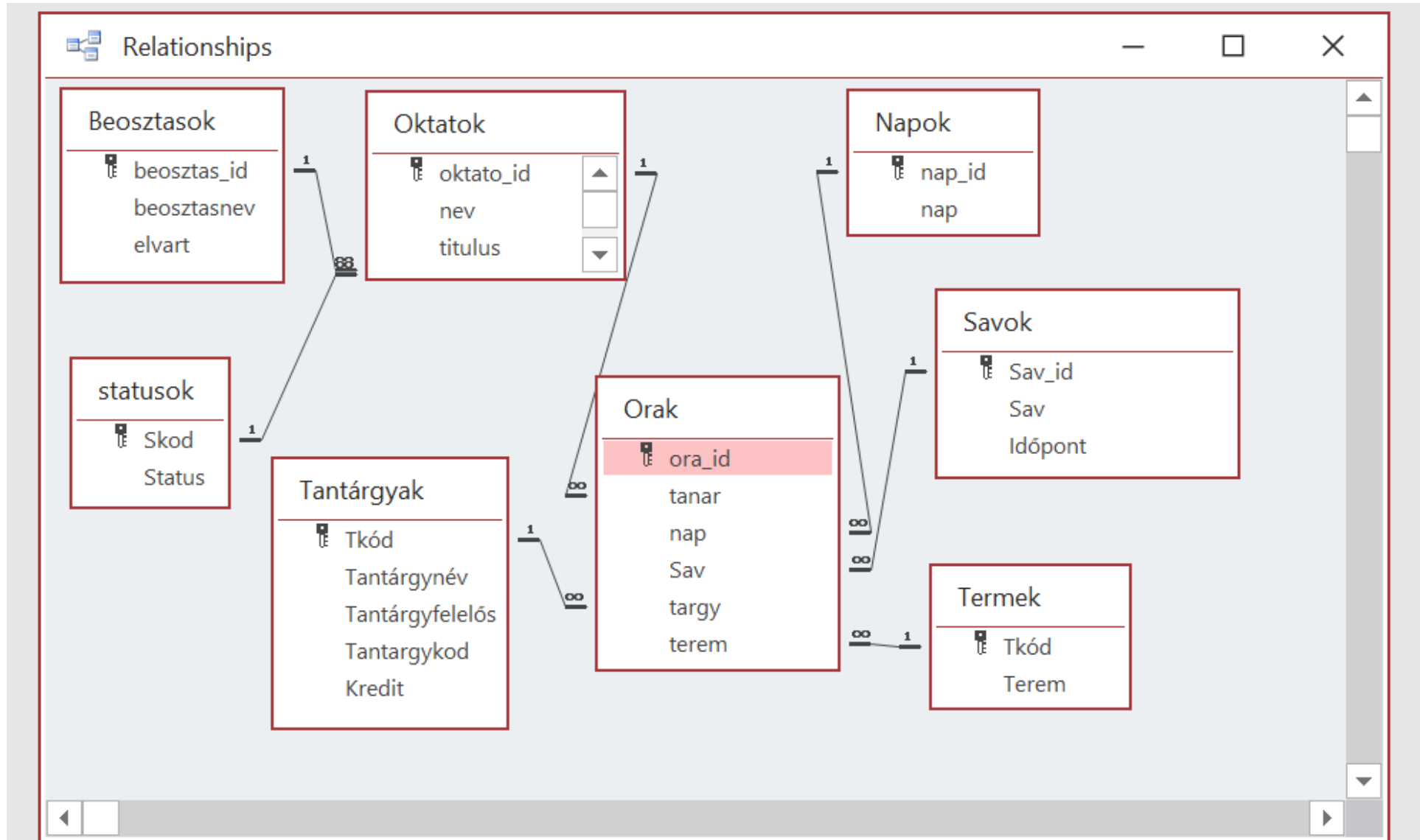
SELECT oszlop1, oszlop2...

FROM ...

ORDER BY oszlop2

A rendezésnél alias nevek is használhatók

A gyakorlaton használt tanulmányi adatbázis



Példák – tanulmányi adatbázis

1. Listázzuk azon órák adatait, amelyek az 1-es vagy 2-es sávban vannak!

```
SELECT *  
FROM orak  
WHERE Sav = 1  
  
UNION  
  
SELECT *  
FROM orak  
WHERE Sav = 2
```

2. Listázzuk azon oktatók adatait, akik beosztása óraadó vagy van órájuk a 219-es teremben! Az eredménylista tartalmazza az esetlegesen duplikált sorokat is!

```
SELECT o.*  
FROM Oktatok o JOIN Beosztasok b ON  
    o.beosztas = b.beosztas_id  
WHERE b.beosztasnev = 'Óraadó'  
  
UNION ALL  
  
SELECT o.*  
FROM Oktatok o JOIN Orak ora ON  
    o.oktato_id = ora.tanar  
    JOIN Termek t ON ora.terem = t.Tkód WHERE  
    t.terem = '219'
```


Példák – tanulmányi adatbázis

3. Vannak-e olyan tanárok, akiknek nincs órájuk? Adjuk meg az illető tanárok azonosítóját és nevét! A listát rendezzük név szerint csökkenő sorrendbe!

```
SELECT oktato_id,  
       nev AS 'Név'  
FROM Oktatok
```

EXCEPT

```
SELECT DISTINCT o.oktato_id, o.nev  
FROM Orak ora JOIN oktatok o ON  
      ora.tanar = o.oktato_id  
ORDER BY Név DESC
```

4. Listázzuk azon tantárgy(ak) azonosítóit, amely(ek)ből nincs egyetlen óra sem! Az eredménylista oszlopát nevezzük el „Nem tanított tantárgyak”-nak!

```
SELECT Tkód  
       AS 'Nem tanított tantárgyak' FROM  
Tantárgyak
```

EXCEPT

```
SELECT DISTINCT targy  
FROM orak
```

Megjegyzések

- A halmazműveletek helyett többnyire van más megoldási lehetőség is (Pl: összetett WHERE –feltétel, JOIN)
- Több halmazművelet esetén a kiértékelési sorrend:
 - Zárójelben lévő műveletek
 - Metszet
 - Különbség és Unió
 - Azonos precedencia esetén: balról jobbra



**Köszönöm
a figyelmet!**