

ADATBÁZIS-KEZELÉS

- SELECT I-II. -

Tarcsi Ádám, ade@inf.elte.hu

Adatok lekérdezése

2

- Input: 1 vagy több tábla
- Output: a megfogalmazott feltételeknek eleget tevő rekordokat tartalmazó „tábla”

Lekérdezések - SELECT

3

- **A jelölésekről**
- `< >` jelek közé írt értékeket mindig az aktuális értékekkel kell helyettesíteni
- `[]` jelek közé írt részek opcionálisak
- `{.. | ...}` a parancsba a felsoroltak közül az egyik lehetőséget kell írni

SELECT utasítás

4

```
SELECT [ALL/DISTINCT] {*/<mezőnév1>, ...,
    <mezőnévk>}
FROM <tablanév1> [<hivatkozásnév1>], ...,
    <tablanévnh> [<hivatkozásnévnh>]
[WHERE <feltétel1>]
GROUP BY <mezőnév1>, ..., <mezőnévm>
HAVING <feltétel2>
ORDER BY <mezőnév1> [ASC/DESC], ...,
    <mezőnévj> [ASC/DESC]
```

SELECT végrehajtási sorrend

5

1. FROM: melyik táblákból?
2. WHERE: melyik rekordok?
3. GROUP BY: mi szerint csoportosítva?
4. HAVING: melyik csoportok?
5. ORDER BY: mi szerint rendezve?
6. SELECT: melyik oszlopok?

Műveletek sorrendje:

- Több tábla esetén direkt szorzat képzése
- Feltételek kiértékelése (rekordok elhagyása)
- Rendezés
- Nem kért oszlopok elhagyása

SELECT

6

- `SELECT * FROM <táblanév>`
Eredménye: ugyanolyan tábla, mint a bemeneti
- `SELECT * FROM <táblanév1>, <táblanév2>`
Eredménye: a két tábla direktszorzata (oszlopainak száma a két bemeneti tábla oszlopszámainak összege)
- Ha több táblából kérdezünk le, meg kell adni, hogy a kívánt mező mely táblából való (minősítés):
`<táblanév>.<oszlopnév>`
`SELECT t1.a FROM t1, t2`
- Konstans érték megadás
`SELECT a, 1, 'hello' FROM t1`

SELECT – 2.

7

- Tábla hivatkozási (alias) név:

```
SELECT * FROM tDiak d, tTanar t
```

- Oszlop elnevezése:

```
SELECT Nev AS Diaknev FROM tDiak
```

- Mező elnevezés:

```
SELECT Nev, Szuldat AS "Születési  
dátum" FROM tDiak
```

- Különböző értékeket:

```
SELECT DISTINCT <oszlopnév> FROM  
tDiak
```

Példa: Határozzuk meg, hogy hány éves diákok vannak az adatbázisban!

SELECT példa

8

```
SELECT nev, upper(nev) as "Nagy  
betűs", ofo, ofo*2, dazon+ofo as  
"Összeg", 1, 'Hello'  
FROM tDiak;
```


Feltételek - WHERE

9

- A SELECT végrehajtásakor minden esetben először a direkt szorzat jön létre és csak utána kerülnek megvizsgálásra a feltételek.
- egyenlőség (egyenlőtlenség) mezők és/vagy konstansok között
- ÉS operátor
- AND, OR
- IN
- NOT (), !=, <>: nem egyenlő
- JOIN (összekapcsolás: hivatkozó tábla idegen kulcsa = hivatkozott tábla elsődleges kulcsa)
- IS NULL, IS NOT NULL

```
Pl.: SELECT Nev FROM tDiak
      WHERE Kor IS NULL
```

LIKE operátor

10

- LIKE operátor, értéktöredék keresésére
- CHAR, VARCHAR, DATE, BLOB esetén
- % - joker, tetszőleges számú karaktert helyettesít
_ (aláhúzásjel): 1 db tetszőleges karaktert helyettesít.
- Pl.: `SELECT * FROM tDiak WHERE Nev like '%Elek%'`
- Feladat: Keressük meg azokat a diákokat, akik nevének harmadik betűje 'b'.

IN operátor

11

- IN: értékek felsorolása ~ OR
in (érték1,érték2,érték3)

Példa:

```
SELECT * FROM tDiak  
WHERE Ofo in(1,2)
```

JOIN - összekapcsolás

12

- JOIN összekapcsolás WHERE feltételben:
- hivatkozó tábla idegen kulcsa = hivatkozott tábla elsődleges kulcsa

Mi a különbség a két lekérdezés között?

1.

```
SELECT d.nev t.nev
FROM tDiak d, tTanar t
```
2.

```
SELECT d.nev t.nev
FROM tDiak d, tTanar t
WHERE d.TAzon=t.TAzon
```

JOIN

13

- Join: egy lekérdezésen belül több tábla adatait kombináljuk össze
- Join feltétel: WHERE-ben szereplő részfeltétel, két tábla oszlopai közötti logikai kifejezés
- Equijoin: egyenlőség operátor
- Self join: egy tábla többször is szerepel a FROM részben

Join

14

- Ahogy tanultuk, a join alapértelmezésben direktszorzat képzést jelent, mely eredményét tipikusan join feltételekkel szűkítjük.
- Leggyakoribb példa az, amikor két tábla között egy-a-sokhoz (idegen kulcs) kapcsolat van, és a join feltétel az idegen kulcsban szereplő két oszlop között fogalmaz meg egyenlőséget

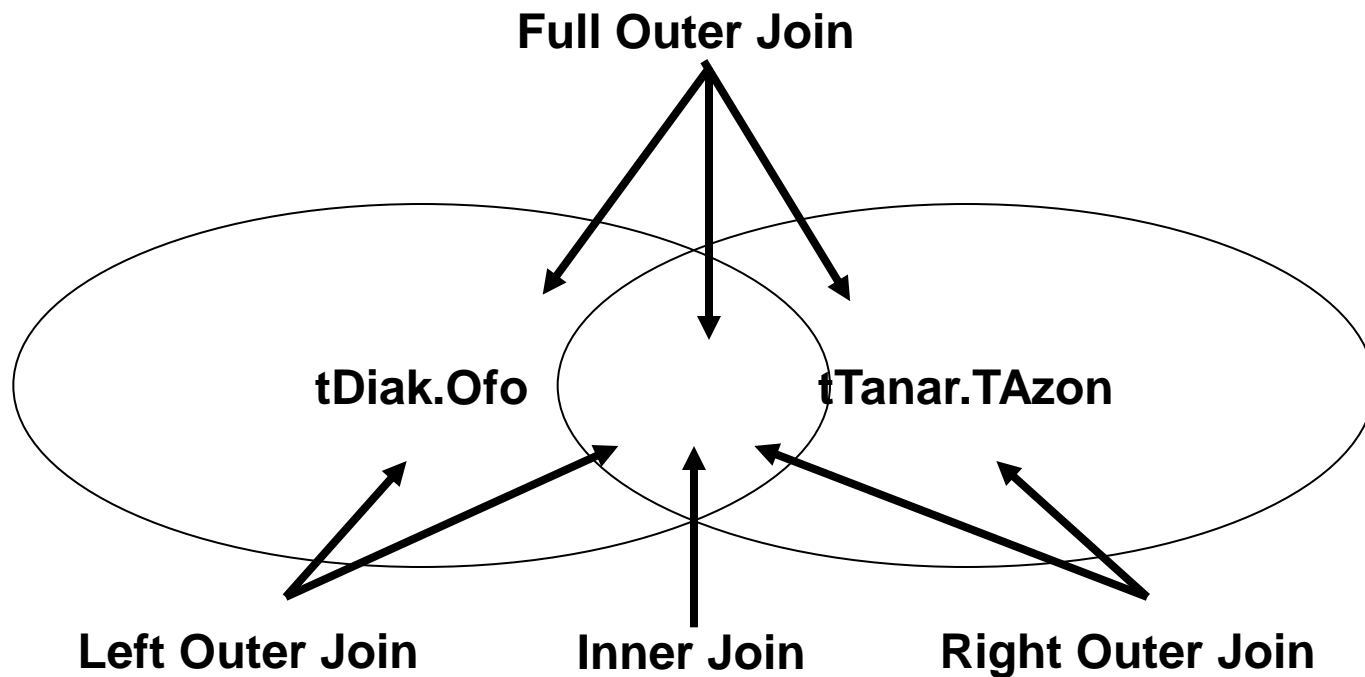
JOIN FROM részben

15

- `SELECT ... FROM t1 <join típus> JOIN t2 ON t1.a=t2.b WHERE ...`
- **INNER (belső, egyszerű) JOIN:** azok a sorok jelennek meg, melyek pontosan teljesítik a megadott (join) feltételt
- **OUTER (külső) JOIN:** visszaadott sorok halmaza bővebb.
Visszaadják az összes olyan sort, melyek kielégítik a join feltételt, plusz még az egyik vagy mindkét táblából azokat is, melyekhez a másik táblában egyetlen olyan rekord sem található, amellyel a join feltétel kielégíthető lenne.

JOIN-ok

16



Példák

17

- Részlet a táblák tartalmából...

tTanar

TAzon	Nev
1	Zsakó László
4	Illés Zoltán

tDiak

DAzon	Nev	Ofo
10	Zsíros B. Ödön	1
32	Deb Ella	1
99	Csin Csilla	

A példa kedvéért most megengedjük, hogy az Ofo mező NULL is lehessen (ezt hívjuk nem kötelező kapcsolatnak a két tábla között):

Left Outer Join

18

DAzon	tDiak.Nev	Ofo	TAzon	tTanar.Nev
10	Zsíros B. Ödön	1	1	Zsakó László
32	Deb Ella	1	1	Zsakó László
99	Csin Csilla			

```
SELECT *  
  FROM tDiak d, tTanar t  
 WHERE d.Ofo = t.TAzon(+);
```

vagy

```
SELECT *  
  FROM tDiak d LEFT OUTER JOIN tTanar t ON d.Ofo =  
  t.TAzon
```

Right outer join

```
SELECT *  
  FROM tDiak d, tTanar t  
 WHERE d.Ofo(+) = t.TAzon;
```

vagy

```
SELECT *  
  FROM tDiak d RIGHT OUTER JOIN tTanar t ON d.Ofo =  
  t.TAzon;
```

DAzon	tDiak.Nev	Ofo	TAzon	tTanar.Nev
10	Zsíros B. Ödön	1	1	Zsakó László
32	Deb Ella	1	1	Zsakó László
			4	Illés Zoltán

Full Outer Join

```
SELECT *  
FROM tDiak d FULL OUTER JOIN tTanar t ON  
d.Ofo = t.TAzon;
```

DAzon	Nev	Ofo	TAzon	Nev_1
10	Zsíros B. Ödön	1	1	Zsakó László
32	Deb Ella	1	1	Zsakó László
99	Csin Csilla			
			4	Illés Zoltán

Right outer join példa

21

□ Példa

Írjuk ki az összes tanár nevét, és hogy hány diáknak osztályfőnöke!

Azok a tanárok is jelenjenek meg, akik nem osztályfőnökök!

□ Megoldás

```
SELECT t.Nev, COUNT(d.DAzon) AS DB
FROM tDiak d RIGHT OUTER JOIN tTanar t ON d.Ofo = t.TAzon
GROUP BY t.Tazon, T.Nev
```

ORDER BY - rendezés

22

- `SELECT ...`
`FROM ...`
`ORDER BY mezőnév1 [ASC | DESC]`
`[, mezőnév2 [ASC | DESC]]`
- Az `ORDER BY` utasítás segítségével meghatározhatjuk, hogy az eredmény táblában milyen sorrendben jelenjenek meg a rekordok.
- Sorszámmal is hivatkozhatunk a `SELECT` záradékban szereplő kifejezésekre.
pl.: `ORDER BY 2, 5, 3 DESC`
- `ASC` – növekvő (alap), `DESC` – csökkenő
- Bal oldali: fő rendezés, jobbra: alrendezés
- Nem kötelező, hogy a kimenetben is szerepeljen a mezőnév.

Feladatok

23

- Adjuk meg azon diákok nevét, akiket az 1-es azonosítójú tanár tanít!
- Készítsünk egy olyan táblázatot, melynek első oszlopában a diák neve szerepel, második oszlopában pedig az adott diákot tanító tanár neve!
- Adjuk meg azon tanulók listáját, akiket Zsakó László tanít és elmúltak 18 évesek!

Aggregátum függvények

24

- **Számosság:** `SELECT COUNT(*) FROM <tábla név>`

Feladat: Adjuk meg, hogy hány diák adatát tároljuk az adatbázisban.

- **Összeg:** `SELECT SUM(<oszlop név>) FROM <tábla név>`

Feladat: Adjuk meg, hogy mennyi a diákok átlagéletkora!

- **További függvények:** `MIN, MAX, AVG`
- **Számított értékek:** `SELECT ofo/2 FROM tdiak`
`SELECT SUM(szuldat+jegy) FROM tDiak`

GROUP BY

25

```
SELECT ...  
  FROM ...  
    [WHERE ...]  
  GROUP BY mező1, [mező 2]  
    [ORDER BY ...]
```

- **Csoportosítás, csoport képzés.** Megszámolhatjuk, hogy hány sor van egy csoportban, vagy kiválaszthatjuk a csoport egy kívánt tagját (pl.: a legidősebbet a csoportban)
- **Oszlopfüggvényeket (SUM, AVG, COUNT, stb.) az egyes csoportokban külön-külön hajtja végre**
- **A GROUP BY-ban megadott mezőnek szerepelnie kell a SELECT utáni felsorolásban.**

GROUP BY – 2.

26

- A WHERE feltételnek megfelelő sorokat csoportosítja úgy, hogy a megadott csoportosítási kifejezéseken (GROUP BY x,y...) belül az egyező értékkel bírók, illetve azok teljes sorai kerülnek 1-1 csoportba.
- Üres csoportok nem kerülnek be!
- Feladat: Adjuk meg, hogy az egyes tanárok hány diáknak osztályfőnökei! (Azonosítóval)

```
SELECT Ofo, COUNT(DAzon) AS Db
FROM tDiak
GROUP BY Ofo;
```

GROUP BY feladatok

27

- Adjuk meg az egyes tanárok (névvel megadva) által oktatott diákok átlagéletkorát!

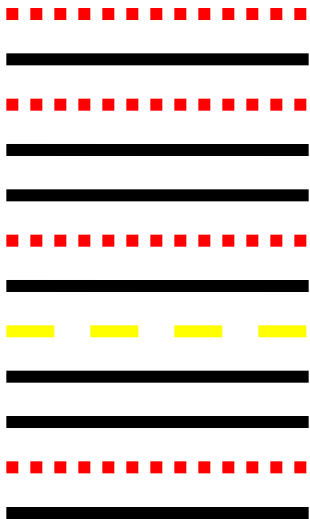
Megjegyzés: Az eredmény táblába nem kerülnek bele azok a tanárok, akik nem tanítanak senkit sem.

HAVING

28

- Csoportokra vonatkozó feltétel megadás. csak a GROUP BY utasítással együtt használjuk.

Minden



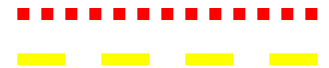
WHERE



GROUP BY



HAVING



HAVING feladatok

29

- Adjuk meg azon tanárok nevét, akik több mint 2 diáknak az osztályfőnökei!
- Adjuk meg azon tanárok nevét és tanítványainak átlagéletkorát, akik legalább 3 diáknak az osztályfőnökei!

Gyakorlás

30




- Hozzuk létre a Tanítás táblát, majd töltsük fel az alábbiak szerint:

Tanár	Diák	Tantárgy
Zsakó László	Zsiros B. Ödön	Fizika
Zsakó László	Feles Elek	Fizika
Zsakó László	Deb Ella	Fizika
Zsakó László	Ka Pál	Fizika
Zsakó László	Har Mónika	Fizika
Zsakó László	Elektrom Ágnes	Fizika
Zsakó László	Feles Elek	Matematika
Zsakó László	Deb Ella	Matematika
Szlávi Péter	Elektrom Ágnes	Matematika
Szlávi Péter	Feles Elek	Informatika
Szlávi Péter	Har Mónika	Matematika

Segítségül

31

- Hozzuk létre a tTanitas táblát, majd töltsük fel az alábbiak szerint:

Tanar 	Diak 	Tantárgy 
1	10	Fizika
1	26	Fizika
1	32	Fizika
1	41	Fizika
1	57	Fizika
1	69	Fizika
1	26	Matematika
1	32	Matematika
2	69	Matematika
2	26	Informatika
2	57	Matematika

Feladatok – 1.

32

1. Hányan tanulnak fizikát?
2. Add meg azokat a tanárokat, akiknek az osztályában van kiskorú!
3. Add meg azokat a tanárokat, akik tanítanak kiskorút!
4. Add meg azokat névsor szerint rendezve, akiket Szlávi Péter tanít!
5. Adjuk meg, hogy az egyes tantárgyakat hány diák tanulja!
6. Add meg azon tanulók átlagéletkorát, akiknek Zsakó László az osztályfőnöke és tanulnak fizikát! (Egy ember fizikát csak egy tanárnál tanul.)