

SQL

Táblák összekapcsolása
lekérdezőkor

Aliasok használata

Allekérdezések

Nézettáblák

A SELECT UTASÍTÁS ÁLTALÁNOS ALAKJA (ISM.)

SELECT [**DISTINCT**] megjelenítendő oszlopok

FROM táblá(k) direkt szorzata)

[**WHERE** feltétel]

[**GROUP BY** csoportosítási szempont]

[**HAVING** csoportok szűrése]

[**ORDER BY** rendezési szempont];

TERMÉSZETES ÖSSZEKAPCSOLÁS SQL-BEN

ID	Név	Fizetés	Szül. dát.	OsztályId
123	Kiss Mária	120500	1955-12-11	3
124	Kovács Gyula	134000	1967-09-09	11

Dolgozó

ID	Név
3	Hulladékgazdálkodási
4	Pénzügyi
11	Marketing

Osztály

```
SELECT Dolgozó.Név, Fizetés, Osztály.Név  
FROM Dolgozó, Osztály  
WHERE Dolgozó.OsztályId = Osztály.ID;
```

vagy

```
SELECT Dolgozó.Név, Fizetés, Osztály.Név  
FROM Dolgozó INNER JOIN Osztály ON  
Dolgozó.OsztályId = Osztály.ID;
```

TERM. ÖSSZEKAPCSOLÁS SQL-BEN

- ▶ Mindkét jelölésnél képződik a táblák direkt szorzata, és projekcióval kiválasztódnak az általunk megjeleníteni kívánt oszlopok
- ▶ Ahol névütközéstől kell tartanunk a direkt szorzatban, a táblák nevét is kiírjuk az oszlopnév elé (pl. Dolgozó.Név)
- ▶ az első módszernél a WHERE feltételben **kell** megadni azt, hogy a külső kulcsok mentén egyező sorok maradjanak meg az eredménytáblában

Példa. Határozzuk meg az előző táblák esetén, hogy hány fő dolgozik az egyes osztályokon.

```
SELECT Osztály.Név, COUNT(Dolgozó.Név)
FROM Dolgozó, Osztály
WHERE Dolgozó.OsztályId = Osztály.ID
GROUP BY Osztály.Név;
```

TERM. ÖSSZEKAPCSOLÁS SQL-BEN

- ▶ A kapott eredmény a következő lenne:

Osztály.Név	COUNT(Dolgozó.Név)
Hulladékgazdálkodási	1
Marketing	1

- ▶ Vegyük észre, hogy a Pénzügyi osztálynál **nem jelenik meg**, hogy 0 fő dolgozik ott
- ▶ Ennek oka: Egy dolgozó sincs az osztályon => a direkt szorzat egyik sorában sem lesz egyezés a külső kulcs mentén '4' érték esetén => nem fogja tartalmazni az eredménytábla a pénzügyi osztályt => nem fog összesítéskor szerepelni

KÜLSŐ ÖSSZEKAPCSOLÁS

- ▶ A látott probléma kiküszöbölésére létezik a külső összekapcsolás: a “lógó” sorok is megmaradhatnak az összekapcsolt táblában
- ▶ LEFT OUTER JOIN – baloldali külső összekapcsolás; a bal oldali tábla lógó sorai maradnak meg, a nem baloldali táblából származó oszlopok az ilyen sorokban NULL értéket kapnak
- ▶ RIGHT OUTER JOIN – jobb oldalról ua.
- ▶ FULL OUTER JOIN – mindkét tábla nem párosított sorai megőrződnek
- ▶ **Észrevétel.** Természetesen – bármilyen összekapcsolásról is legyen szó – kettőnél több tábla is összekapcsolható.

KÜLSŐ ÖSSZEKAPCSOLÁS

- ▶ Így a korábbi probléma – az összes (akár nulla létszámú) osztályhoz tartozó dolgozói létszámok kiírása – már meg is oldható:

```
SELECT Osztály.Név, COUNT(Dolgozó.Név)
FROM Dolgozó RIGHT OUTER JOIN Osztály
ON Dolgozó.OsztályId = Osztály.Id
GROUP BY Osztály.Név;
```

Osztály.Név	COUNT(Dolgozó.Név)
Hulladékgazdálkodási	1
Marketing	1
Pénzügyi	0

ALIASOK

- ▶ A SELECT utasítás után adandó oszloplista nem csak oszlopneveket, hanem kifejezéseket is tartalmazhat, amiket el lehet nevezni – ez az aliasok egyik szerepe; szintaxis: kifejezés AS alias_név

- ▶ Példa. Ország népsűrűsége.

```
SELECT név, lakosság/terület AS népsűrűség FROM ország;
```

- ▶ Alias használandó akkor is, ha egy SQL utasításon belül **adott táblára többszörösen hivatkozunk**, például azonos nevű dolgozók lekérése:

```
SELECT d1.Név FROM  
Dolgozó AS d1, Dolgozó AS d2  
WHERE d1.Név = d2.Név AND d1.id < d2.id;
```

ALLEKÉRDEZÉSEK

- ▶ Korábban láttunk már példát rájuk – tulajdonképpen arról van szó, hogy egy adott SQL utasításon belül elhelyezünk egy SELECT-tet.

Példa. Allekérdezéssel előállított halmaz IN után:

```
SELECT Név
FROM Dolgozó
WHERE Dolgozó.OsztályId IN (
    SELECT Osztály.Id
    FROM Osztály
    WHERE Osztály.Id > 5
);
```

ALLEKÉRDEZÉSEK

- ▶ Allekérdezést persze használhatunk a FROM parancs után is:

Példa. Adott az Employee(id, name, salary, address) reláció. Állítsunk ebből elő egy olyan táblát, amiben magyar oszlopnevek vannak a következő adatokkal: név, azonosító, éves bér. Csak a \$4000-nál többet kereső “Smith” családnevű dolgozók jelenjenek meg.

```
SELECT altabla.nev, altabla.id, altabla.eves_ber
FROM
(
    SELECT name AS nev, id, salary*12 AS
    éves_ber, address
    FROM Employee
    WHERE salary > 4000
) AS altabla
WHERE altabla.nev LIKE '% Smith';
```

- ▶ Sőt, még akár INSERT-en belül is elhelyezhető alkérdés: INSERT INTO tábla SELECT ...

ALLEKÉRDEZÉSEK

- ▶ Halmazműveleteknél – UNION, INTERSECT, EXCEPT – jellemzően két alkérdés áll az utasítás oldalán:

```
(SELECT nev FROM oktatok)
```

```
UNION
```

```
(SELECT nev FROM hallgatok);
```

- ▶ Ilyenkor a UNION-nál – mint azt már tudjuk – a többszörösen szereplő értékek csak egyszer jelennek meg. Ha azt akarjuk, hogy a duplikált sorok többször megjelenjenek, UNION ALL használandó
- ▶ A táblák kompatibilitására ügyelni kell!

ALLEKÉRDEZÉSEK

- ▶ Nem minden esetben tudunk a korábbiakban látottakhoz hasonló megoldást adni a problémáinkra

Példa.

Növeljük meg azon dolgozók fizetését 10%-al, akik az átlagfizetésnél kevesebbet keresnek:

```
UPDATE Dolgozo AS d1
SET d1.fizetes = 1.1 * d1.fizetes
WHERE (SELECT AVG(d2.fizetes)
       FROM Dolgozo AS d2
       ) > d1.fizetes;
```

A MySQL nem engedi meg, hogy módosítsunk egy táblát úgy, hogy a WHERE feltételben ugyanezen táblára vonatkozó alkérdés van! **Szintén nem jó**, ha INSERT után próbálkozunk hasonlóan.

NÉZETTÁBLÁK

- ▶ Nem valódi táblák, feladatuk az adatbázisban lévő adatok bemutatása egy adott transzformáció végrehajtása után
- ▶ Tulajdonképpen megadunk egy “formulát”, ami a már meglévő adatok alapján létrehoz egy nézettáblát
- ▶ Létrehozása:

```
CREATE VIEW táblanév [(oszloplista)] AS SELECT ...;
```

- ▶ Törlése:

```
DROP VIEW táblanév;
```

- ▶ A nézettáblákon is végrehajthatók ugyanazon műveletek, mint a tárolt táblákon, azaz pl. az értékek aktualizálhatók – és ilyenkor a forrástáblába is bekerülnek a módosítások
- ▶ Nyilván a forrástábla módosítása is automatikusan látszik a nézettáblában

NÉZETTÁBLÁK

- ▶ Van néhány eset, amikor azonban a nézettábla módosítása nem lehetséges:

Példa. Raktár(cikkszám, név, egységár, mennyiség)

```
CREATE VIEW Keszlet (aru, ertek) AS
```

```
SELECT nev, egysegar*mennyiseg FROM raktar;
```



- ▶ Nem lehet módosítani a nézettáblát, ha
 - ▶ nem lehet egyértelmű a végrehajtás (pl. fent), vagy ha
 - ▶ DISTINCT opciót,
 - ▶ valamilyen összekapcsolást,
 - ▶ GROUP BY alparancsot tartalmaz a definíció.