



Adatbázisrendszerek

2. gyakorlat

Adatok lekérdezése

```
SELECT [{ALL|DISTINCT}] mezőkifejezés [álnév] [, mezőkifejezés [álnév]]...  
FROM táblakifejezés [álnév]  
[WHERE feltétel]  
[GROUP BY csoportosítómező [, csoportosítómező]...]  
[HAVING feltétel]  
[ORDER BY mezőkifejezés [, mezőkifejezés]...]
```

A **SELECT** utasítás az adatok egy halmazát válogatja ki egy táblázatba az adatbázisból.

Opcionálisan megadható:

- **WHERE:** Az utána álló feltételnek megfelelő sorok leválogatása.
- **GROUP BY:** Az utána álló mezőkifejezések alapján csoportosítja az adatokat.
- **HAVING:** A feltételnek megfelelő sorok leválogatása a csoportosítás után.
- **ORDER BY:** Sorok rendezése a megadott mezők alapján.

A *null* érték

- Az a mező, amelynek értéke nem került megadásra a *null* értéket veszi fel.
- A *null* nem nulla és nem üres sztring!
- Vizsgálata:
 - kifejezés `IS [NOT] NULL`
- Példa: Kérdezzük le azon könyvek címét, amelyeknek nincs megadva az oldalszáma de meg van adva az ára!
 - `SELECT CIM FROM KONYV
WHERE OLDALSZAM IS NULL AND AR IS NOT NULL;`

A *null* érték

- Ha két érték összehasonlításánál vagy egy műveletnél az egyik operandus null értékű, az eredmény is null lesz.
 - pl.: $a > \text{null}$, $a < > \text{null}$, $\text{null} * 2$, $a - \text{null}$ stb.
- Egy null értéket tartalmazó logikai kifejezés eredménye csak abban az esetben lesz null, ha a kifejezés értéke függ attól, hogy a null érték helyén igaz, vagy hamis érték szerepelne.
 - pl.: $1 = 0 \text{ AND } \text{null}$ (*hamis*), $1 = 0 \text{ OR } \text{null}$ (*null*)
- A szelekciós feltétel null értéke esetén az érintett sorok nem kerülnek leválogatásra.

Mintaillesztés

- Mintaillesztést a LIKE operátor segítségével végezhető.
 - kifejezés LIKE minta
- A mintában szerepelhet aláhúzás jel (_), mely egy darab tetszőleges karaktert helyettesít vagy százalék jel (%), amelyre tetszőleges karaktersorozat illeszkedik.
- Példa: Kérdezzük le azon könyvek címét és árát, amelyek címe a „Fény” karaktersorozattal kezdődik!
 - ```
SELECT CIM, AR FROM KONYV
WHERE CIM LIKE 'Fény%';
```

# Táblák összekapcsolása: cross join

```
SELECT ... FROM táblakifejezés { , | CROSS JOIN }
táblakifejezés
```

Kereszt összekapcsolás létrehozása a táblakifejezések vesszővel elválasztott felsorolásával, vagy a **CROSS JOIN** kulcsszóval lehetséges.

Az eredményként kapott táblában a két tábla sorainak összes lehetséges kombinációja pontosan egyszer jelenik meg, azaz Descartes-szorzatot képez.

- Egy **WHERE** feltétel megadásával lehet elérni, hogy csak a szükséges sorok jelenjenek meg.

# Táblák összekapcsolása: inner join

```
SELECT ...
FROM táblakifejezés [INNER] JOIN táblakifejezés
ON feltétel
```

Belső összekapcsolás létrehozásához a táblakifejezések között az **INNER JOIN** vagy **JOIN** kulcsszavakat kell használni. Az összekapcsolás feltételét az **ON** kulcsszó után kell megadni.

Az eredményként kapott táblában a két tábla azon sorainak kombinációi szerepelnek, amelyek esetében igaz az összekapcsolási feltétel.

- Ugyan azt az eredményt adja, mint egy megfelelő WHERE feltételt tartalmazó kereszt összekapcsolás.
- Itt is van lehetőség WHERE feltétel megadására.