# A SELECT működése

A SELECT utasítás általános formája:

SELECT [ALL/DISTINCT] {*/<mező név$_1$>, …, <mező név$_k$>}
FROM <tábla név$_1$> [<hivatkozási név$_1$>], …, <tábla név$_h$> [<hivatkozási név$_h$>]
[WHERE <feltétel$_1$>
GROUP BY <mező név$_1$>, …, <mező név$_m$>
HAVING <feltétel$_2$>
ORDER BY <mező név$_1$> [ASC/DESC], …, <mező név$_j$> [ASC/DESC]]

műveletek sorrendje

1. FROM: melyik táblákból? (direkt szorzat képzés)
2. WHERE: melyik rekordok? (nem kívánt sorok elhagyása)
3. GROUP BY: mi szerint csoportosítva? (csoportosítás)
4. HAVING: melyik csoportok? (nem kívánt csoportok elhagyása)
5. ORDER BY: mi szerint rendezve? (rendezés)
6. SELECT: melyik oszlopok? (nem kívánt oszlopok elhagyása)

Vessézzük ki ezt a SELECT-et néhány példán keresztül!
Van két táblánk, az egyszerűség kedvéért ugyanazok, amiket az órán vettünk! Tanár és diák táblák az alábbi tartalommal:

Ttanar tábla:

| TAZON | NEV |
|---|---|
| 1 | Zsakó László |
| 2 | Szlávi Péter |
| 3 | Pap Gáborné |
| 4 | Illés Zoltán |

Tdiak tábla:

| DAZON | NEV | KOR | OFO |
|---|---|---|---|
| 1 | Zsíros B. Ödön | 16 | 1 |
| 2 | Feles Elek | 19 | 2 |
| 3 | Deb Ella | 21 | 1 |
| 4 | Ka Pál | 16 | 1 |
| 5 | Har Mónika | 28 | 2 |
| 6 | Elektrom Ágnes | 22 | 3 |

Maradjunk rögtön a két táblás lekérdezésnél!
**Legyen a feladat: Kiket tanít Zsakó László? Az eredményt jelenítsük meg név szerint rendezve!**
A feladat megoldása:

```
SELECT tdiak.nev
FROM tdiak, ttanar
WHERE tdiak.ofo = ttanar.tazon
     AND ttanar.nev LIKE 'Zsakó%'
ORDER BY tdiak.nev
```

Azonban tegyük föl, hogy ezt még nem tudjuk!
Elnézve a tábla szerkezetét, ehhet mindkét táblára szükségünk van. Induljunk ki most ebből!

***1. lépés***
Mert a SELECT utasításban elsőként a FROM záradék lép működésbe, ezért elkészül a tábláink direkt szorzata, azaz a:

```
SELECT *
FROM tdiak, ttanar
```

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |
| 2 | Feles Elek | 19 | 2 | 1 | Zsakó László |
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |
| 5 | Har Mónika | 28 | 2 | 1 | Zsakó László |
| 6 | Elektrom Ágnes | 22 | 3 | 1 | Zsakó László |
| 1 | Zsiros B. Ödön | 16 | 1 | 2 | Szlávi Péter |
| 2 | Feles Elek | 19 | 2 | 2 | Szlávi Péter |
| 3 | Deb Ella | 21 | 1 | 2 | Szlávi Péter |
| 4 | Ka Pál | 16 | 1 | 2 | Szlávi Péter |
| 5 | Har Mónika | 28 | 2 | 2 | Szlávi Péter |
| 6 | Elektrom Ágnes | 22 | 3 | 2 | Szlávi Péter |
| 1 | Zsiros B. Ödön | 16 | 1 | 3 | Pap Gáborné |
| 2 | Feles Elek | 19 | 2 | 3 | Pap Gáborné |
| 3 | Deb Ella | 21 | 1 | 3 | Pap Gáborné |
| 4 | Ka Pál | 16 | 1 | 3 | Pap Gáborné |
| 5 | Har Mónika | 28 | 2 | 3 | Pap Gáborné |
| 6 | Elektrom Ágnes | 22 | 3 | 3 | Pap Gáborné |
| 1 | Zsiros B. Ödön | 16 | 1 | 4 | Illés Zoltán |
| 2 | Feles Elek | 19 | 2 | 4 | Illés Zoltán |
| 3 | Deb Ella | 21 | 1 | 4 | Illés Zoltán |
| 4 | Ka Pál | 16 | 1 | 4 | Illés Zoltán |
| 5 | Har Mónika | 28 | 2 | 4 | Illés Zoltán |
| 6 | Elektrom Ágnes | 22 | 3 | 4 | Illés Zoltán |

Hiszen amíg nem mondjuk meg az idegen kulcs által leírt kapcsolattal nem törődik!
Keletkezett tehát 4*6, azaz 24 sor. (4 sor volt a Ttanar és 6 sor a tDiak táblában, ezeket kombinálta, minden lehetséges módon)
Kiemelve jelölöm a helyes sorokat. Tehát, ha azt akarom, hogy a kapcsolatot is figyelembe vegye, már pedig ez mindig szükséges nálunk, azt a WHERE záradékban külön meg kell mondanom. Erre később visszatérek.

***2. lépés.***
WHERE záradék. Itt adom meg azon feltételeket, amelyeket minden olyan sornak teljesítenie kell, amely az eredményben bent szeretne maradni.
A feltételeknek teljes körűen teljesülniük kell. Ha több feltétel van tehát, akkor mindnek teljesülnie kell. Az ellenőrzés soronként fut le, amelyik sor megszegi, kiesik.
Jelen esetben két feltételünk van. A tanár neve: Zsakó László és mivel a tábláink között kapcsolat van, az OFO és a TAZON értékeknek soron belül meg kell egyezniük.
Ennek három sor (azaz rekord) tesz eleget:

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |

A SELECT utasításunk tehát most így néz ki:

```
SELECT *
FROM tdiak, ttanar
WHERE tdiak.ofo=ttanar.tazon
  AND ttanar.nev LIKE 'Zsakó%'
```

***3. lépés.***

Még rendeznünk kell diák neve szerint a sorokat. Ezt az ORDER BY záradékkal tehetjük meg.

```
SELECT *
FROM tdiak, ttanar
WHERE tdiak.ofo = ttanar.tazon
     AND ttanar.nev LIKE 'Zsakó%'
ORDER BY tdiak.nev
```

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |

***4. lépés.***

Végezetül pedig megmondjuk, hogy csak a név oszlopra vagyunk kiváncsiak. Azaz más már nem maradt hátra, eldobjuk a nem fontos oszlopokat. Az oszlopkiválasztás fut le utoljára.

```
SELECT tdiak.nev
FROM tdiak, ttanar
WHERE tdiak.ofo = ttanar.tazon
     AND ttanar.nev LIKE 'Zsakó%'
ORDER BY tdiak.nev
```

| NEV |
|---|
| Deb Ella |
| Ka Pál |
| Zsiros B. Ödön |

**Következő feladat:**

Adjuk meg azon osztályfőnökök nevét ABC sorrendben és tanítványainak átlagéletkorát, akik legalább 2 diáknak osztályfőnökei!

Ebben a feladatban a SELECT utasítás minden záradékát használni fogjuk!

```
SELECT t.Nev, AVG(d.Kor) AS Atlag
FROM tDiak d, tTanar t
WHERE d.Ofo=t.TAzon
GROUP BY t.Nev,t.TAzon
HAVING COUNT(d.DAzon)>=2
ORDER BY t.Nev
```

Az utasítás első három és az utolsó sorának, tehát a FROM, WHERE, ORDER BY zaradékok és az oszlopkiválasztás lefutása már ismert, az előző feladatéhoz hasonló.

A feladat érdekessége a csoportosítás. Amire azért van szükség, mert valamit tanáronként szeretnénk megkapni, ilyenkor mindenféleképpen csoportosítani kell. Illetve, ha egy összegzőfüggvény (mint amilyen az átlagot kiszámító AVG is) használata mellett egyéb, de az egyes csoportokban azonos értéket szeretnénk kiírni. Jelen esetben a tanár nevét. Ilyenkor persze ezen mező szerint (is) csoportosítani kell.
Na de mi is az a csoportosítás és hogyan is működik?
Csoportosítás előtt a SELECT-ünk itt tart:

```
SELECT *
FROM tDiak d, tTanar t
WHERE d.Ofo=t.Tazon
```

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |
| 2 | Feles Elek | 19 | 2 | 2 | Szlávi Péter |
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |
| 5 | Har Mónika | 28 | 2 | 2 | Szlávi Péter |
| 6 | Elektrom Ágnes | 22 | 3 | 3 | Pap Gáborné |

Látható, hogy azok a sorok maradtak bent, amiket a WHERE feltétel bent hagyott. Ezeket fogjuk tanáronként csoportosítani, tehát lesz két csoportunk, mert két tanár maradt bent. Azaz a tanár neve szerint külön-külön annyi csoportot alkotunk, ahány tanárunk van. Ez persze csak a háttérben történik meg, az egyes csoportokat így kiirattatni nem tudjuk.

```
SELECT *
FROM tDiak d, tTanar t
WHERE d.Ofo=t.TAzon
GROUP BY t.Nev,t.TAzon
```

Zsakó László csoportja

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |

Szlávi Péter csoportja

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 2 | Feles Elek | 19 | 2 | 2 | Szlávi Péter |
| 5 | Har Mónika | 28 | 2 | 2 | Szlávi Péter |

Pap Gáborné csoportja:

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 6 | Elektrom Ágnes | 22 | 3 | 3 | Pap Gáborné |

Innentől a következő, a HAVING záradék ezekben a csoportokban dolgozik. A HAVING-ben a WHERE feltételhez hasonlóan feltételeket adhatunk meg. Azonban itt már csoportfeltételeket. Azaz olyanokat, amely egy adott csoportra összességében, teljeskörűen vonatkozik. Például megszámolhatjuk hány sor található az egyes csoportokban és arra adhatunk egy feltételt. Jelen esetben azt, hogy legalább ketten legyenek az egyes csoportokban.

```
SELECT *
```

```
FROM tDiak d, tTanar t
WHERE d.Ofo=t.TAzon
GROUP BY t.Nev,t.TAzon
HAVING COUNT(d.DAzon)>=3
```

Tehát csak Zsakó László és Szlávi Péter csoportja maradt bent. Ezek után újra összeilleszti a csoportokban lévő sorokból a nagy táblát:

| DAZON | NEV | KOR | OFO | TAZON | NEV |
|---|---|---|---|---|---|
| 2 | Feles Elek | 19 | 2 | 2 | Szlávi Péter |
| 5 | Har Mónika | 28 | 2 | 2 | Szlávi Péter |
| 1 | Zsiros B. Ödön | 16 | 1 | 1 | Zsakó László |
| 3 | Deb Ella | 21 | 1 | 1 | Zsakó László |
| 4 | Ka Pál | 16 | 1 | 1 | Zsakó László |

Ez után fut le az ORDER BY, majd az oszlopkiválasztás. Csak olyan oszlopot választhatok ki, ami szerint csoportosítottam, vagy pedig egy vagy több összesítő függvényt. A végeredmény pedig:

| NEV | ATLAG |
|---|---|
| Szlávi Péter | 23.5 |
| Zsakó László | 1.76666666666666666666666666666666666667E01 |

# AL-SELECT használata, működése

Két esetben használunk mindenféleképpen al-selektet. Tagadás esetén illetve, ha olyan feltételünk van, amit egy soron belül nem tudunk megvizsgálni. Hiszen a WHERE feltétel ellenőrzése soronként fut le, más sorra, más sornak vagy táblának egy konkrét értékére nem hivatkozhatunk. Miképpen olyan értékre sem, melyet egy másik tábla adataiból számoltunk ki. Ilyenkor ezt a más értéket egy al-select segítségével számítjuk ki és ezt az al-select által adott értéket hasonlítjuk össze. Érdemes úgy tekinteni az alselectre, mint egy konkrét értékre, számra, stb-re.
Alselectes feladatokat a legegyszerűbben úgy lehet elkészíteni, ha előbb elkészítjük a belső selecttet, majd ezt zárójelbe téve e köréépítjük a külsőt!
Nézzük meg ezt egy egyszerű példán keresztül!

**Feladat**: Ki a legidősebb diák?
A feladat megoldásához egy táblára, a diák táblára van szükségünk.

| DAZON | NEV | KOR | OFO |
|---|---|---|---|
| 1 | Zsiros B. Ödön | 16 | 1 |
| 2 | Feles Elek | 19 | 2 |
| 3 | Deb Ella | 21 | 1 |
| 4 | Ka Pál | 16 | 1 |
| 5 | Har Mónika | 28 | 2 |
| 6 | Elektrom Ágnes | 22 | 3 |

A fentiek alapján könnyen átgondolhatjuk, hogy ez egy lefutásban nem oldható meg, hiszen nem tudjuk, hogy hány éves a legidősebb diák, más sorok tartalmára pedig nem hivatkozhatunk. A tábla tartalma alapján ez egy ellenőrzéssorozattal nem oldható meg. Ha jobban tetszik úgy is mondhatjuk, hogy egy számlálós (FOR) ciklussal nem tudjuk ezt a feladatot megoldani.
Első lépésben tehát ki kell számolnunk, hány éves a legidősebb diák, azaz:

```
SELECT max(kor)
FROM tdiak
```

| MAX(KOR) |
|---|
| 28 |

Majd azt kell megnéznünk, hogy ki az, vagy kik azok, akik szintén ennyi idősek (azaz ebben a pillanatban 28 évesek).
Hogy tovább egyszerűsítsük, írjuk meg a feladatot úgy, mintha tudnánk, hogy a legidősebb diákunk 28 éves. Azaz a feladat most úgy szól, ki 28 éves?

```
SELECT nev
FROM tdiak
WHERE kor = 28
```

| NEV |
|---|
| Har Mónika |

Ez azonban nyilvánvalóan nem megoldása az eredeti feladatnak, hiszen csak az adattábla egy pillanati állapotát tükrözi, ez azóta megváltozhatott (új, még idősebb diák jöhetett, Har Mónika kiléphetett az iskolából, vagy időközben akár a születésnapját is ünnepelhette).

Tehát a belső szelektet be kell építenünk a külsőbe, azaz:

```
SELECT nev
```

```
FROM tdiak
WHERE kor = (SELECT max(kor)
             FROM tdiak)
```

| NEV |
| --- |
| Har Mónika |

Figyelni kell arra, hogy relációsjeleket (=, <, <=, >, >=, <>, !=) csak akkor használhatunk, ha a belső select csak egy értéket adott vissza. Ezt összesítő függvény használatával, vagy csoportosítással elérhetjük. Ha előfordulhat, hogy több értéket ad vissza, akkor meg kell mondani, hogy a kapott lista:

- Bármely elemével lehet egyenlő (előfordulásvizsgálat): IN
- Egyik elemével sem lehet egyenlő: NOT IN
- Minden eleme teljesíti a feltételt: ALL (például nagyobb nála)
- Nem minden eleme teljesíti a feltételt, azaz legalább van egy akivel nincs relációban: NOT ALL
- Van egy elem, ami kielégíti a feltételt: ANY
- Vagy semelyikkel sincs a megadott relációban (egyetlen esetben sem igaz): NOT ANY.