

A törlési anomália bizonyos értelemben a beszűrési anomália fordítottja. Tegyük fel, hogy egy témán egyetlen dolgozó dolgozott és azt sikeresen befejezte. Ezután elhagyta a céget. Szükségessé válik adatainak törlése a táblából. Ez azonban az általa elvégzett téma jellemzőinek törlését is jelenteni fogja. Pedig erre még később szükség lehet, például egy év végi zárójelentésben, ahol szerepeltetni kell az összes elvégzett munkát.

**Az anomáliákat mindig az okozta, hogy logikailag nem összetartozó adatokat szerveztünk egy struktúrába.**

Nagyon lényeges, hogy a tervezés során ezt a hibát ne kövessük el. Ezért **el kell választanunk egymástól a logikailag összetartozó, vagy mondhatjuk úgy hogy összefüggő adatokat.** S ebben lesz fontos szerepe a funkcionális függésnek, ugyanis ennek segítségével tudjuk kiválogatni hogy mik azok az attribútumok, amelyek ilyen csoportokat alkotnak. Ezeket aztán különválasztjuk, **önálló táblákat** alkotva. Így csökkenthetjük, illetve szüntethetjük meg a fenti problémákat.

### Normálformák

A relációk felbontásával különböző normálformájú relációkat kapunk.  
A legegyszerűbb az **első normálforma**.

Azt mondjuk, hogy **egy reláció első normálformában van**, ha a relációban minden érték elemi, vagyis a reláció nem tartalmaz adatcsoportot.

Ezt ugyanis már a relációs modellbeli táblázat alapvető tulajdonságai között is megköveteltük. Ha az alaphalmazok csak elemi adatokat tartalmaznak, akkor a megadott reláció rögtön első normálformájú is lesz. Csoportos adatok esetén azonban lényeges, hogy a keletkezett táblázatot első normálformájúvá alakítsuk. Például ha a bérszámfejtő rendszerünkben a dolgozók szakképzettségét is nyilvántartjuk, akkor kérdéses lehet, hogyan oldjuk meg azt a problémát, amikor **egy dolgozónak többféle szakképzettsége** van. Ha ezeket a többszörös értékeket hagyományos módon felírjuk egy táblázatba, akkor a kapott táblázat nem első normálformájú reláció. Itt a sorok megismétlésével járó technikát alkalmazhatjuk. Megjegyezzük, hogy a legújabb adatmodelleknél - ilyen az **objektumorientált** - ismét visszatértek a hagyományos módszerhez. Az objektumorientált modell összetett adatstruktúrák: **halmaz, lista, tömb, struktúra, multihalmaz** kínálja a csoportos adatok kezelésére.

Tekintsük a következő táblázatot:

Dolgozó

A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje	Szakképzettség
T234578	Kiss István	Eger	1968.12.11.	Közgazdász
T456734	Nagy József	Budapest	1972.01.30.	Programozó matematikus, közgazdász
T429877	Kovács János	Szeged	1967.05.12.	Programtervező matematikus, számítástechnik

				a tanár
--	--	--	--	---------

Ezt a táblázatot az alább módon alakíthatjuk át első normálformájú relációvá:

Dolgozó

A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje	Szakképzettség
T234578	Kiss István	Eger	1968.12.11.	Közgazdász
T456734	Nagy József	Budapest	1972.01.30.	Programozó matematikus
T456734	Nagy József	Budapest	1972.01.30.	Közgazdász
T429877	Kovács János	Szeged	1967.05.12.	Programtervező matematikus
T429877	Kovács János	Szeged	1967.05.12.	Számítástechnika tanár

Egy relációt **második normálformájúnak** szokás nevezni, ha első normálformájú, és egyetlen másodlagos attribútuma sem függ egyetlen kulcsának valódi részalmazától sem.

A második normálforma azt jelenti, hogy a relációban nincsenek a kulcs részeitől való nem triviális függőségek.

Ez a normálforma tulajdonképpen egy közbenső lépés a harmadik normálforma felé. Ezért jelentősége csupán annyi, hogy a normalizálási tevékenységet kisebb lépések beiktatásával segíti, vagyis ezáltal a normalizálást végző személynek először csak a kulcsokat és azok részalmazait kell megvizsgálni, hogy a nem kívánatos függőségeket eltávolítsa a relációból.

A második, harmadik és Boyce-Codd normálformák esetén a normalizálási technikák megegyeznek.

**Gyakorlati szempontból a leglényegesebb normálforma a harmadik.**

Erre vonatkozóan az irodalomban több definíció található, amelyek egymással ekvivalensek.

**Definíció:** Egy  $R$  reláció **harmadik normálformában** van, ha

- 1) második normálformájú és
- 2) egyetlen másodlagos attribútuma sem függ tranzitíven kulcstól.

Egy  $A \rightarrow C$  funkcionális függést **tranzitív függésnek** nevezünk, ha létezik olyan  $B$  attribútum halmaz, hogy  $A \rightarrow B$  továbbá  $B \rightarrow C$ .

Vizsgáljuk meg, mit is jelent ez a definíció.

**Az hogy egy másodlagos attribútum nem függ tranzitíven egy kulcstól azt jelenti, hogy nem található olyan attribútumok, amelyekről funkcionálisan függ bármely másodlagos attribútum.**

Ugyanis ha ilyen lenne, akkor lenne tranzitív funkcionális függés, mivel a kulcstól minden más attribútum függ.

Ez alapján a harmadik normálforma másik definíciója a következő:

**Definíció:** Egy  $R$  reláció *harmadik normálformában* van, ha második normálformájú és a másodlagos attribútumok között nincsen funkcionális függés.

Végezetül lehet adni egy olyan definíciót is, amely nem igényli a második normálforma fogalmát. Azon szemléletmód esetén, amikor egyből a harmadik normálforma definíciójával indulnak, ezt szokás használni.

**Definíció:** Egy  $R$  reláció akkor és csak akkor van *harmadik normálformában*, amennyiben ha létezik az  $A \rightarrow B$  függőség valamely  $A, B$  attribútum halmazok esetén, akkor vagy az teljesül hogy az  $A$  az  $R$  szuperkulcsa, vagy a  $B$  része valamely kulcsnak.

Miért ekvivalens ez a meghatározás a korábbiakkal?

A definíció első része azt fejezi ki, hogy a relációban csak olyan függőségek lehetnek, amelynél a baloldal tartalmaz kulcsot. A definíció második része megengedi még az olyan jellegű függéseket, amelyek baloldalán tetszőleges attribútumok, jobboldalán pedig kulcshoz tartozó attribútumok vannak. A definíció két részét egybefogva láthatjuk, hogy a harmadik normálformánál a függések valamelyik oldalán mindenképpen szerepelni kell kulcshoz tartozó attribútumnak, tehát másodlagos attribútumok közötti függés nem fordulhat elő a relációban.

Definiáljuk a **Boyce-Codd normálformát**, amely a harmadik normálformának egy további általánosítása.

**Definíció:** Egy  $R$  reláció akkor és csak akkor van *Boyce-Codd normálformában*, amennyiben ha létezik az  $A \rightarrow B$  függőség valamely  $A, B$  attribútum halmazok esetén, akkor az  $A$  az  $R$  szuperkulcsa.

A szuperkulcs definíciója alapján a Boyce-Codd normálforma lényegében azt jelenti, hogy minden nem triviális funkcionális függőség bal oldalának tartalmaznia kell egy kulcsot.

Mint láttuk a második, harmadik és Boyce-Codd normálformák teljesülése mindig azon múlik, hogy a relációban vannak-e bizonyos jellegű funkcionális függőségek.

Ha szeretnénk elérni azt, hogy egy adott normálforma kritériuma teljesüljön, akkor **el kell tüntetnünk a számunkra nem kedvező függéseket a relációból.**

Ez egy **felbontással történhet, amelynek során a relációt két önálló relációra bontjuk.**

Tegyük fel hogy az  $A \rightarrow B$  függőség miatt sérül egy normálforma feltétele. A felbontásnál az egyik reláció ( $R_1$ ) a függőség bal és jobb oldalán szereplő attribútumokból tevődik össze, vagyis  $R_1 = A \cup B$ . A másik reláció ( $R_2$ ) a függőség baloldalán található attribútumokból és az összes a függőségben nem szereplő attribútumból tevődik össze, vagyis  $R_2 = A \cup \overline{A \cup B}$ .

Nézzük most meg egy példán keresztül, hogyan történik a felbontás:

**Példa**

Tekintsük a következő relációt:

**Dolgozótéma**(A dolgozó törzsszáma, A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje, A vezető törzsszáma, Téma azonosítója, Témavezető)

Már korábban említettük, hogy a reláció elsődleges kulcsa A dolgozó törzsszáma, Téma azonosítója attribútumokból áll. Látható, hogy a relációban fennáll a

**A dolgozó törzsszáma** → { **A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje, A vezető törzsszáma** }

funkcionális függés. E miatt még a 2. normálforma feltétele sem teljesül a relációra. A felbontási szabály szerint az egyik reláció

**a Dolgozó**(A dolgozó törzsszáma, A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje, A vezető törzsszáma)

reláció lesz.

A felbontás második táblája pedig a

**Téma**(A dolgozó törzsszáma, Téma azonosítója, Témavezető)

reláció lesz.

A **Dolgozó** reláció kulcsa a **A dolgozó törzsszáma**, a **Téma** relációé pedig a **Téma azonosítója**.

Mivel ezek egyszerű kulcsok, és az ezektől való függőségeken kívül egyik relációban sincsenek más függőségek, ezért mindkét reláció harmadik normálformájú, tehát további felbontásra nincs szükség.

A harmadik normálforma mindig megfelel a gyakorlatban előforduló követelményeknek. Azaz egy relációt mindig fel tudunk bontani harmadik normálformájú sémákra úgy, hogy közben nem veszítünk semmilyen információt.

### **Relációalgebrai műveletek**

A relációs modell egyik legnagyobb előnye, hogy különböző műveleteket definiálhatunk, amelyek az **egyes relációkból újakat** szolgáltatnak.

Ez biztosít számunkra egy olyan rendszert, amely segítségével egységes módon tudunk a már **meglévő adatainkból új információt nyerni**, úgy, hogy nem lépünk ki a modell keretei közül.

Ezért a relációalgebrai műveletek fontos részét képezik a modellnek. Most ezeket ismertetjük.

**Definíció:** Legyen  $R$  és  $S$  két  $n$ -edfokú reláció. A két reláció **egyesítésén** ( $R \cup S$ ) azok halmazelméleti egyesítésével kapott relációt értjük, azaz azt a relációt, amelynek előfordulásai vagy  $R$ -nek, vagy  $S$ -nek előfordulásai.

### **Példa**

Tekintsük az alábbi Dolgozó1 és Dolgozó2 relációkat:

Dolgozó1

A dolgozó törzsszáma	A dolgozó Neve	A dolgozó születési helye	A dolgozó születési ideje
T234578	Kiss István	Eger	1968. 12. 11.
T456734	Nagy József	Budapest	1972. 01. 30.

Dolgozó2

A dolgozó törzsszáma	A dolgozó Neve	A dolgozó születési helye	A dolgozó születési ideje
T456734	Nagy József	Budapest	1972. 01. 30.
T429877	Kovács János	Szeged	1967. 05. 12.

Ezek egyesítése a következő Dolgozó reláció (Dolgozó = Dolgozó1  $\cup$  Dolgozó2)

Dolgozó

A dolgozó törzsszáma	A dolgozó Neve	A dolgozó születési helye	A dolgozó születési ideje
T234578	Kiss István	Eger	1968. 12. 11.
T456734	Nagy József	Budapest	1972. 01. 30.
T429877	Kovács János	Szeged	1967. 05. 12.

Egyesítés esetén meg kell követelni, hogy a két reláció attribútum halmazai megegyezzenek, vagyis hogy a megfelelő oszlopok azonos típusúak legyenek. Csak ekkor várhatjuk ugyanis, hogy a keletkezett reláció is megfelelő struktúrájú legyen.

**Definíció:** Legyen  $R$  és  $S$  két  $n$ -edfokú reláció. A két reláció **metszetén** ( $R \cap S$ ) azok halmazelméleti metszetével kapott relációt értjük, azaz azt a relációt, amelynek előfordulásai  $R$ -nek, és  $S$ -nek is előfordulásai.

#### Példa

Tekintsük ismét az előző példában megismert Dolgozó1 és Dolgozó2 relációkat:

Ezek metszete a következő Dolgozó reláció (Dolgozó = Dolgozó1  $\cap$  Dolgozó2)

Dolgozó

A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje
T456734	Nagy József	Budapest	1972. 01. 30.

**Definíció:** Legyen  $R$  és  $S$  két  $n$ -edfokú reláció. A két reláció *különbségén* ( $R \setminus S$ ) azok halmazelméleti különbségével kapott relációt értjük, azaz azt a relációt, amelynek előfordulásai  $R$ -nek igen, de  $S$ -nek nem előfordulásai.

### 6. Példa

Ismét a Dolgozó1 és Dolgozó2 relációkat használjuk. Ezek különbsége a következő Dolgozó reláció (Dolgozó = Dolgozó1 - Dolgozó2)

A dolgozó törzsszám a	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje
T234578	Kiss István	Eger	1968. 12. 11.

**Definíció:** Legyen  $R$  egy  $n$ -edfokú,  $S$  pedig egy  $m$ -edfokú reláció. A két reláció *Descartes-szorzatán* ( $R \times S$ ) azt az  $n+m$ -edfokú relációt értjük, amelynek előfordulásait úgy alakítjuk ki, hogy  $R$  minden előfordulásához egyenként hozzávesszük  $S$  minden előfordulását. Az így kapott párok fogják alkotni a szorzat reláció elemeit.

Láthatjuk, hogy a Descartes-szorzat esetén meg kell különböztetnünk egymástól a két reláció attribútumainak neveit. Ha ezek eredendően különbözőek, akkor nincs semmi tennivalónk. Azonban ez nem mindig teljesül, nincs arra vonatkozóan semmi megkötés, hogy olyan relációkat nem szorozhatunk össze, amelyek azonos attribútum nevekkel rendelkeznek. A megkülönböztetés céljából ilyenkor az attribútum neve elé szoktuk írni a reláció nevét is, attól egy ponttal elválasztva.

### Példa

Tekintsük most az alábbi Dolgozó és Dolgozó téma relációkat:

A dolgozó törzsszám a	A dolgozó neve	A dolgozó születési helye
T234578	Kiss István	Eger
T456734	Nagy József	Budapest
T429877	Kovács János	Szeged

A dolgozó törzsszáma	Téma azonosítója	Témavezető neve
T429877	T25	Nagy Ádám
T456734	T26	Mekk Elek

Ezek Descartes szorzata az alábbi módon néz ki:

DTéma

Dolgozó.A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	Téma.A dolgozó törzsszáma	Téma azonosítója	Témavezető neve
T234578	Kiss István	Eger	T429877	T25	Nagy Ádám
T234578	Kiss István	Eger	T456734	T26	Mekk Elek
T456734	Nagy József	Budapest	T429877	T25	Nagy Ádám
T456734	Nagy József	Budapest	T456734	T26	Mekk Elek
T429877	Kovács János	Szeged	T429877	T25	Nagy Ádám
T429877	Kovács János	Szeged	T456734	T26	Mekk Elek

A Descartes-szorzat látszólag nem túl hasznos művelet gyakorlati szempontból, mivel nagyon sok felesleges információ is keletkezik általa. Ugyanis minden sort minden sorral összepárosít, és ezeknek az új soroknak jó része szükségtelen. De ez fogja képezni az alapját a legfontosabb reláció algebrai műveleteknek.

**Definíció:** Legyen  $R$  egy  $n$ -edfokú reláció. Az  $R$  **projekcióján** egy olyan  $\pi(R)$  relációt értünk, amelyet úgy kapunk, hogy  $R$ -ből elhagyunk bizonyos attribútumokat, a maradékot pedig valamilyen módon átrendezzük. Jelöljük a keletkezett attribútumokat  $A_1, \dots, A_n$ -nel. Ekkor új relációra szokásos a  $\pi_{A_1, \dots, A_n}(R)$  jelölést használni.

#### Példa

Vegyük az előző példában Descartes-szorzattal keletkezett DTéma relációt. Használjuk az következő rövidítéseket: A=Dolgozó.A dolgozó törzsszáma, B=A dolgozó neve, C=Téma.A dolgozó törzsszáma, D=Téma azonosítója, E= Témavezető neve.

A  $\pi_{A,C,B,D,E}(DTéma)$  reláció a következő:

$$\pi_{A,C,B,D,E}(DTéma)$$

Dolgozó.A dolgozó törzsszáma	Téma.A dolgozó törzsszáma	A dolgozó neve	Téma azonosítója	Témavezető neve
T234578	T429877	Kiss István	T25	Nagy Ádám
T234578	T456734	Kiss István	T26	Mekk Elek
T456734	T429877	Nagy József	T25	Nagy Ádám
T456734	T456734	Nagy József	T26	Mekk Elek
T429877	T429877	Kovács János	T25	Nagy Ádám
T429877	T456734	Kovács János	T26	Mekk Elek

A projekció gyakorlati haszna abban rejlik, hogy egyrészt segítségével egy táblázatból kiemelhetjük a lényeges információt, másrészt azt tetszőleges elrendezésű táblázattá alakíthatjuk.

**Definíció:** Legyen  $R$  egy  $n$ -edfokú reláció. Az  $R$  **szelekcióján** egy olyan  $\sigma_F(R)$  relációt értünk, amelyben  $R$ -ből azokat az előfordulásokat hagyjuk meg, amelyek eleget tesznek az  $F$  logikai kifejezéssel megadott formulának.

A definícióban szereplő  $F$  formula a következő objektumokból állhat:

- **operanduszok**, amelyek konstansokat, attribútum neveket vagy sorszámokat tartalmazhatnak
- operanduszokon értelmezett **aritmetikai operátorok** ( $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $=$ ,  $<>$ ).
- operanduszokon értelmezett **logikai operátorok** ( $\wedge$ , (és),  $\vee$ , (vagy),  $\neg$ , (nem))

**Példa**

Ismét a projekció műveletnél keletkezett  $\pi_{A,C,B,D,E}(DTema)$  relációt tekintjük, az ott használt rövidítésekkel. Legyen  $F$  a  $D = \text{"Nagy Ádám"}$  logikai kifejezés. Ekkor a  $\sigma_F(\pi_{A,C,B,D,E}(DTema))$  reláció a következő:

$$\sigma_F(\pi_{A,C,B,D,E}(DTema))$$

Dolgozó.A dolgozó törzsszám a	Téma.A dolgozó törzsszáma	A dolgozó neve	Téma azonos ítója	Témavezető neve
T234578	T429877	Kiss István	T25	Nagy Ádám
T456734	T429877	Nagy József	T25	Nagy Ádám
T429877	T429877	Kovács János	T25	Nagy Ádám

A szelekció ugyancsak fontos művelet abból a szempontból, hogy egy relációból ki tudjuk válogatni a lényeges információt. **A lényeges különbség a projekcióhoz képest az, hogy ennél a táblázat soraira történik a válogatás.** A Descartes-szorzat után keletkezett nagyméretű táblákat kell gyakran szűrni. A **szűrés** mindig szelekcióval történik, a megfelelő logikai feltétel megadásával.



## 1 Az SQL nyelv AB-2 (SQL)

**SQL (Structured Query Language = Strukturált Lekérdező Nyelv).**

A lekérdezési funkciók mellett a nyelv több olyan elemmel is rendelkezik, amelyek más adatkezelési funkciók végrehajtására is alkalmasak. A nyelv legújabb szabványos változatai pedig már egészen kiterjedt adatbázis-kezelési műveletek megvalósítására is használhatók.

Az SQL szabványos adatbázis-kezelő nyelvvé vált.

Az SQL lekérdező része tulajdonképpen ekvivalens a **relációs algebrával**, ezért az egyik legalkalmasabb eszköz a relációs adatmodell kezelésére, segítségével nyelvi eszközökkel is megfogalmazhatjuk a matematikailag jól megalapozott műveleteket.

Lehetőség van programokba történő beépítésére, ami egy nagyon hatékony, gyors lekérdezési lehetőséget biztosít.

Az SQL-nek több verziója van.

Általában a különböző ABKR implementációk a saját képükre formált SQL megvalósítással rendelkeznek.

Van azonban a nyelvnek két fő szabványos változata. Ezek a **ANSI (American National Standards Institute = Amerikai Nemzeti Szabványügyi Intézet)** által definiált SQL,

illetve az 1992-ben elfogadott módosított szabvány, az

**SQL-92 vagy SQL2.**

Később ezeket az **ISO (International Standards Organization = Nemzetközi Szabványügyi Szervezet)** is elfogadta.

A legújabb verzió az **SQL3.**

Minden SQL parancs egy kulcsszóval kezdődik, amelyet paraméterek, esetleg további kulcsszavak követhetnek. A paraméterek általában azok az objektumok, melyekre a parancsok vonatkoznak. Ilyen lehet például egy tábla vagy attribútum neve, esetleg egy konstans, vagy valamilyen azonosító.

A parancsok egyszerűek és összetettek is lehetnek, ami azt jelenti, hogy például egy parancson belül másik parancs is előfordulhat.

Az SQL parancsokat két fő csoportra osztjuk, ezek a

**lekérdező parancsok,**

illetve az **adatdefiníciós parancsok.**

### Lekérdezések az SQL-ben

#### Egyszerű lekérdezések

A legalapvetőbb SQL parancs egy teljes tábla, illetve a tábla valamely oszlopainak lekérdezésére szolgál. **Ez tulajdonképpen a projekció relációalgebrai művelet megvalósítása.** A parancs általános szintaxisa a következő:

```
SELECT [ALL|DISTINCT] * | <oszlopnévlista> FROM  
<táblanév>
```

A parancs első részében a \* jel azt jelenti, hogy a lekérdezés a teljes táblára vonatkozik.

Az <oszlopnévlista> segítségével adhatjuk meg azt, hogy a lekérdezés eredményeképpen melyik oszlopokat kívánjuk megjeleníteni.

Lehetőség van különböző módosításokra, például a listában szereplő elemeket átnevezhetjük, esetleg kifejezéseket képezhetünk belőlük.

Emellett alkalmazhatunk különböző függvényeket is, amelyek egy része az aggregáló függvények, másik része pedig olyan függvények, amelyeket a gazdanyelv biztosít az SQL környezet számára.

*Az <oszlopnévlista> szintaxisa a következő:*

```
<oszlopkifejezés>[, <oszlopkifejezés>]...
```

Amennyiben az oszlopkifejezésben aggregáló függvényt használunk, akkor az eredménytáblában nem a tábla egyes előfordulásainak adatai jelennek meg, hanem azoknak a megfelelő aggregáltját fogja képezni a parancs. A tábla soraiból álló halmazra végrehajtódik a megfelelő függvény.

Az <oszlopkifejezés> a következő aggregáló függvényeket tartalmazhatja:

- **COUNT:** Megadja a tábla sorainak számát.
- **SUM:** Megadja a paraméterében szereplő oszlop adatainak az összegét az összes rekordra. Csak numerikus attribútumra alkalmazható.
- **AVG:** Megadja a paraméterében szereplő oszlop adatainak az átlagát az összes rekordra. Csak numerikus attribútumra alkalmazható.
- **MIN:** Megadja a paraméterében szereplő oszlop adatainak a minimumát az összes rekordra. Csak numerikus attribútumra alkalmazható.
- **MAX:** Megadja a paraméterében szereplő oszlop adatainak a maximumát az összes rekordra. Csak numerikus attribútumra alkalmazható.

Lehetőség van oszlopok átnevezésére is. Ennek szintaxisa a következő:

```
<oszlopkifejezés> AS <oszlopnév>
```

Az eredménytáblában az <oszlopnév> paraméterben megadott oszlopnév szerepel.

A parancsban használható **ALL** és **DISTINCT** kulcsszavaknak speciális jelentése van. Előfordulhat, hogy egy lekérdezés végrehajtása után a keletkezett elemek között ugyanaz a rekord többször előfordul.

Ilyenkor az **ALL** opció alkalmazásakor, illetve alapértelmezés szerint az azonos előfordulások többszörösen fognak megjelenni az eredményben.