

# MEGSZORÍTÁSI SZABÁLYOK

Készítette: Takács Gábor

Az adattípusok eleve meghatároznak egy értékalmazt, amit az adott tulajdonság felvehet. Ennek kapcsán fontos, hogy pontos tudással rendelkezünk az adattípusok révén definiált értékalmazról.

Pl.:

- "A" eset: ha egy tulajdonság adattípusa tinyint, akkor az az érték 0-255 között vehet fel csak értéket.
- "B" eset: ha egy karaktorsor tulajdonsága char(6), akkor az csak az „[ASCII printable characters](#)” táblából vehet fel értéket. Egy neptunkód esetében ez megfelelő, mivel nincsen benne ékezet, de egy név esetében már ez kevés lenne. (Ez esetben nvarchar-t lenne célszerű használni)

A Check constraint az egy előre definiált érvényességi szabály, mely a meglévő adattípust szűkíti tovább.

Azaz "A" esetben ha érdemjegynek a tinyint adattípust adom, akkor ez önmagában még nem elegendő, mivel 0-s, 6-os, 7-es, stb. jegyet érdemjegyet nem is mer a magyar oktatási rendszer, így a felvehető értékalmazt is tovább kell szűkíteni.

## Példa megszorítási szabályokra egész számok esetében:

1. Feltétel: 1-5 közé eső érdemjegy rögzíthető csak a rendszerben  
tulajdonég neve: [jegy] | adattípusa: tinyint  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás 1: [jegy] BETWEEN 1 AND 5  
megoldás 2: [jegy]>=1 AND [jegy]<=5
2. Feltétel: Kreditértéke maximum 16 lehet  
tulajdonég neve: [kredit] | adattípusa: tinyint  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás 1: [kredit] < 16  
megoldás 2: [kredit] <= 15  
Vegyük észre, hogy itt csak felső értékhatárt vizsgáltunk. Alsó értékhatárt nem volt szükséges, mivel lehetnek nulla kredit értékű tárgyak Pl: testnevelés, nyelvi órák, stb., az adattípus pedig mínuszba nem megy át.

## Példa megszorítási szabályokra karakterláncok esetében:

Karakterláncok esetében karakterenként kell megadni, hogy mit vehet fel az adott érték.

Jelölések:

- [] között jelöljük az egy karakter által felvehető értéket, vagy értékalmazt
- '' között jelöljük a függvényeket.

3. Feltétel: Neptun kód, karakterenként csak nagybetűt vagy számot vehessen fel.  
tulajdonég neve: [neptun] | adattípusa: char (6)  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás 1: [neptun] LIKE '[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'

4. Feltétel: A karakterként tárolt magyar irányítószám csak számokat vehessen fel.  
tulajdonég neve: [iranyitoszam]  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás 1: [iranyitoszam] LIKE '[1-9][0-9][0-9][0-9]'  
Vegyük észre, hogy az első szám esetében nem engedtem meg a nulla érték felvételét, mivel a legkisebb magyar irányítószám az 1011, így az első helyiértéken a nulla érték nem megengedhető.

5. Feltétel: A karakterként tárolt magyar irányítószám csak számokat vehessen fel, de konvertáljuk át számmá a karaktersort, hogy megvizsgálhassuk matematikailag is, hogy az adott irányítószám nagyobb-e vagy kisebb, mint ami lehet.  
tulajdonég neve: [iranyitoszam]  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás: [iranyitoszam] LIKE '[1-9][0-9][0-9][0-9]' AND CAST([iranyitoszam] AS smallint)>=1011

Vegyük észre, hogy a feltételben egy CAST típuskonverziót is alkalmaztunk, tehát a vizsgálat idejére tudtunk adni neki egy ideiglenes szám adattípust. Ugyan ezt a CONVERT típuskonverzióval is meg tudtuk volna tenni. A két függvényről [ITT](#) olvashatsz bővebben.

6. Feltétel: A vizsga típusát helytakarékosági okokból csak egy betűvel jelöljük. Ha v-t mentünk el, akkor vizsgás tárgyról van szó. Ha f-t mentünk el, akkor félévközi teljesítésével abszolválható a tárgy. Ha s-t írunk, akkor szigorlatozni kell a tárgyból. Írjuk olyan megszorítási szabályt, ami csak ezt a három karaktert engedi felvenni.

tulajdonég neve: [vizsgatipus]  
Feltételnek megfelelő megszorítási szabály(ok):  
megoldás 1: [vizsgatipus] LIKE '[fvs]'

Vegyük észre, hogy itt csak egy karaktert kell vizsgálnunk, ami három értéket vehet fel, de egy időben, a három közül is csak egyet. Vegyük észre azt is, hogy itt a szögletes zárójelben nem használunk se szóközt, se kötőjelet. Ha kötőjelet használnánk, akkor a kötőjellel jelölt két szélsőérték között bármit felvehetne a karakter.

Példák:

- megszorítási szabály: [vizsgatipus] LIKE '[f-sv]'  
Jelentése: f és s közötti karaktereket veheti fel és a v-t is. Ebbe f, és s is beletartozik
- megszorítási szabály: [vizsgatipus] LIKE '[fs-v]'  
Jelentése: felveheti f-et, és s és v között bármit. Ebbe s, és v is beletartozik

### Dátum típusú megszorítási szabályok

Dátumok megszorítási szabályainak esetében a leggyakrabban az alábbi két függvényt vesszük igénybe:

- [datediff](#)
- [dateadd](#)

A függvények használatáról a CAST-hez hasonlóan érünk el információt a súgóban, ahol példákat is találunk. Az adott függvény súgó leírását a fenti függvények nevére kattintva lehet elérni.