# ADATBÁZIS-KEZELÉS

Szabó Bálint

MÉDIAINFORMATIKAI KIADVÁNYOK

# ADATBÁZIS-KEZELÉS

Szabó Bálint




Eger, 2011

*Lektorálta:*

CleverBoard Interaktív Eszközöket és Megoldásokat Forgalmazó és Szolgáltató Kft.



A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Felelős kiadó: dr. Kis-Tóth Lajos
Készült: az Eszterházy Károly Főiskola nyomdájában, Egerben
Vezető: Kérészy László
Műszaki szerkesztő: Nagy Sándorné

Kurzusmegosztás elvén (OCW) alapuló informatikai curriculum és SCORM kompatibilis tananyag-fejlesztés Informatikus könyvtáros BA, MA lineáris képzésszerkezetben
TÁMOP-4.1.2-08/1/A-2009-0005

# Tartalom

**1. Bevezetés ..... 10**
1.1 Célkitűzés ..... 10
1.2 A kurzus tartalma ..... 10
1.3 Kompetenciák és követelmények ..... 11
1.4 Tanulási tanácsok, tudnivalók ..... 11
**2. Az adatbázis-kezelés alapfogalmai ..... 12**
2.1 Célkitűzés ..... 12
2.2 A lecke témakörei ..... 12
2.3 Adat, információ, kommunikáció ..... 12
2.3.1 Az adat ..... 12
2.3.2 Az információ ..... 13
2.3.3 Különbségek adat és információ között ..... 14
2.3.4 A kommunikáció ..... 16
2.4 Mit jelent az adatbázis-kezelés? ..... 16
2.5 Az adatbázisok felépítése ..... 17
2.5.1 Egyed és tulajdonság ..... 17
2.5.2 Tulajdonságtípus ..... 18
2.5.3 Egyedtípus ..... 19
2.5.4 Kapcsolatok ..... 19
2.5.5 A kapcsolatok tipizálása, kapcsolattípusok ..... 20
2.6 Az adatmodell ..... 20
2.6.1 A relációs adatmodell ..... 21
2.6.2 Speciális mezők a relációs adatmodellben ..... 23
2.6.3 Az adatbázis ..... 27
2.7 Összegzés ..... 28
2.8 Önellenőrző kérdések ..... 29
**3. Relációs adatbázisok tervezése ..... 30**
3.1 Célkitűzés ..... 30
3.2 A lecke témakörei ..... 30
3.3 Egyedtípusok tárolása a relációs adatmodellben ..... 30
3.4 A redundancia és következményei ..... 32
3.5 Anomáliák ..... 33
3.5.1 Bővítési anomália ..... 33
3.5.2 Törlési anomália ..... 33
3.5.3 Változtatási anomália ..... 33
3.6 Normalizálás ..... 34
3.6.1 Mezők viszonya a relációs adatbázis tábláiban ..... 34
3.6.2 Normálformák ..... 37
3.7 Adatbázis tervének ábrázolása ..... 41
3.8 Összefoglalás ..... 41
3.9 Önellenőrző kérdések ..... 42
**4. A Microsoft Access adatbázisok szerkezete, és kezelése ..... 43**

4.1 Célkitűzés ..... 43
4.2 A lecke témakörei ..... 43
4.3 adatbázis-kezelő rendszerek ..... 43
4.4 Relációs adatbázis-kezelő rendszerek funkciói ..... 44
4.5 Az MS Access kezelőfelülete ..... 45
4.5.1 Kezelőfelület ..... 45
4.5.2 Az Access indítása ..... 49
4.6 Access adatbázisok szerkezete ..... 49
4.6.1 Adatbázisok a fájlrendszerben ..... 49
4.6.2 Access adatbázisok objektumai ..... 49
4.7 Access adatbázis kezelése ..... 50
4.7.1 Adatbázis megnyitása ..... 50
4.7.2 Tallózás az Access adatbázis objektumai között ..... 50
4.7.3 Műveletek objektumokkal ..... 51
4.7.4 Adatbázis bezárása ..... 52
4.7.5 Adatbázis biztonsági mentése ..... 52
4.8 Összefoglalás ..... 52
4.9 Önellenőrző kérdések ..... 53

**5. Microsoft Access adatbázisok kialakítása ..... 54**

5.1 Célkitűzés ..... 54
5.2 A lecke témakörei ..... 54
5.3 Adatbázis létrehozása az Accessben ..... 54
5.4 Adatbázis állomány létrehozása ..... 54
5.5 Táblák létrehozása az adatbázisban ..... 55
5.5.1 A tábla tervező ablak felépítése ..... 56
5.5.2 Legfontosabb mezőtulajdonságok beállítása ..... 57
5.5.3 Adattípusok az Access tábláiban ..... 59
5.5.4 Adattípus kiválasztása ..... 61
5.5.5 További mezőtulajdonságok ..... 62
5.6 Elkészített táblák használata ..... 63
5.7 Összefoglalás ..... 63
5.8 Önellenőrző kérdések ..... 64

**6. Táblák közötti kapcsolatok kialakítása ..... 65**

6.1 Célkitűzés ..... 65
6.2 A lecke témakörei ..... 65
6.3 A kapcsolatok jelentősége ..... 65
6.4 Fogalmak ..... 65
6.5 A kapcsolatok ablak ..... 66
6.5.1 Kapcsolat létrehozása ..... 67
6.5.2 Hivatkozási integritás megőrzése ..... 68
6.6 Kapcsolatok mentése ..... 73
6.7 Összefoglalás ..... 74
6.8 Önellenőrző kérdések ..... 74

**7. Rekordok bevitele, rendezése, keresése, szűrése ..... 75**

7.1 Célkitűzés ..... 75
7.2 A lecke témakörei ..... 75
7.3 Kifejtés ..... 75
7.3.1 A gyakorló adatbázis szerkezete ..... 75
7.4 Táblák felhasználói nézete ..... 78
7.4.1 Rekordok tallózása ..... 79
7.4.2 Új rekord felvétele ..... 80
7.4.3 Rekord módosítása ..... 81
7.4.4 Rekordok törlése ..... 81
7.4.5 Rekordok sorba rendezése ..... 82
7.5 Rekordok felkutatása ..... 83
7.5.1 A keresés eredménye ..... 83
7.5.2 Egyezés ..... 83
7.5.3 Keresés ..... 84
7.5.4 Kis- és nagybetűk, formázott adat ..... 84
7.5.5 Helyettesítő, vagy dzsóker karakterek ..... 84
7.6 Rekordok kiválogatása ..... 85
7.6.1 Szűrés kijelölés alapján ..... 85
7.6.2 Eredményhalmaz kezelése, szűrő kikapcsolása, törlése ..... 86
7.6.3 Szűrés beírás alapján ..... 86
7.6.4 Irányított szűrés, rendezés ..... 86
7.7 Összefoglalás ..... 93
7.8 Önellenőrző kérdések ..... 94
**8. Lekérdezések készítése ..... 95**
8.1 Célkitűzés ..... 95
8.2 A lecke témakörei ..... 95
8.3 Lekérdezések jelentősége ..... 95
8.4 Rekordforrás, forráshalmaz, eredményhalmaz ..... 96
8.5 Lekérdezés létrehozása ..... 98
8.5.1 Rekordforrás megtervezése ..... 98
8.5.2 Új lekérdezés létrehozása, rekordforrás megadása ..... 98
8.5.3 Illesztési tulajdonságok ..... 99
8.5.4 Mezők elhelyezése a tervezőrácson ..... 99
8.5.5 Feltételek, rendezési irány megadása ..... 100
8.5.6 Mező, és Lekérdezés tulajdonságok ..... 100
8.5.7 Lekérdezés tervének mentése ..... 100
8.5.8 Eredményhalmaz megtekintése ..... 100
8.6 Illesztési tulajdonságok ..... 101
8.6.1 Szoros illesztés ..... 101
8.6.2 Laza illesztés ..... 102
8.6.3 Illesztési tulajdonságok beállítása ..... 103
8.6.4 Nem kapcsolódó rekordok kiválogatása ..... 104
8.7 Számított mezők ..... 105
8.7.1 Függvények használata ..... 106
8.8 Mező és lekérdezés tulajdonságok ..... 108

8.8.1 Lekérdezés tulajdonságok beállítása ..... 108
8.8.2 Fontosabb lekérdezés tulajdonságok: ..... 109
8.8.3 Mezőtulajdonságok beállítása ..... 110
8.9 Összefoglalás ..... 110
8.10 Önellenőrző kérdések ..... 111
**9. Speciális feladatok lekérdezésekkel ..... 112**
9.1 Célkitűzés ..... 112
9.2 A lecke témakörei ..... 112
9.3 Lekérdezések típusai ..... 112
9.3.1 Egyszerű lekérdezések ..... 113
9.4 Paraméteres lekérdezések ..... 115
9.4.1 Paraméteres lekérdezés kialakítása ..... 117
9.4.2 Paraméterek deklarálása ..... 117
9.5 Összesítő lekérdezések: ..... 117
9.5.1 Összesítő lekérdezés készítése lépésenként ..... 118
9.6 Kereszttáblás lekérdezések ..... 120
9.6.1 A kereszttáblás lekérdezés elkészítése ..... 121
9.7 Akció lekérdezések ..... 122
9.7.1 Akció lekérdezések elkészítésének lépései ..... 123
9.7.2 Törlő lekérdezés ..... 125
9.8 Összefoglalás ..... 125
9.9 Önellenőrző kérdések ..... 125
**10. Az Access adatbázisok kezelőfelületének kialakítása ..... 126**
10.1 Célkitűzés ..... 126
10.2 A lecke témakörei ..... 126
10.3 Az űrlapok felhasználási lehetőségei ..... 127
10.4 Az űrlapok megjelenése ..... 127
10.5 űrlapok és adatok ..... 128
10.6 Műveletek űrlapokkal ..... 128
10.7 Űrlapok használata ..... 129
10.7.1 Űrlap megnyitása ..... 129
10.7.2 Űrlap bezárása ..... 130
10.7.3 Az űrlapok tervező nézete ..... 130
10.8 Új űrlapok létrehozása ..... 132
10.8.1 Űrlap létrehozás varázslóval ..... 132
10.8.2 Az űrlap felületi megjelenésének szabályozása ..... 133
10.8.3 Címkék szövegének átírása ..... 133
10.8.4 Vezérlőelemek Betűformátuma ..... 133
10.8.5 Vezérlőelemek méretezése ..... 134
10.8.6 Vezérlőelemek mozgatása ..... 134
10.8.7 Vezérlőelemek típusának megváltoztatása ..... 134
10.8.8 Az űrlap mentése ..... 138
10.9 Az adatbázis menü űrlapja ..... 138
10.10 Összefoglalás ..... 139
10.11 Önellenőrző kérdések ..... 139

**11. Tárolt adatok nyomtatott megjelenítése ..... 140**
11.1 Célkitűzés ..... 140
11.2 A lecke témakörei ..... 140
11.3 A jelentések készítése varázsló segítségével ..... 140
11.4 A jelentések nézetei ..... 142
11.5 A jelentések szerkezete ..... 143
11.6 A jelentés elrendezésének szabályozása ..... 144
11.6.1 Csoportok kialakítása, és összegképzés ..... 144
11.7 A jelentés végső beállítása tervező nézetben ..... 148
11.7.1 Szakaszok és nyomtatott oldalak ..... 148
11.7.2 Futó összeg elhelyezése ..... 148
11.7.3 Új címkék lehelyezése ..... 149
11.8 Jelentés mentése ..... 149
11.9 Végső áttekintés és nyomtatás ..... 149
11.10 Jelentések elhelyezése a menü űrlapon ..... 150
11.11 Összefoglalás ..... 151
11.12 Önellenőrző kérdések ..... 151
**12. Összefoglalás ..... 152**
12.1 A kurzusban kitűzött célok összefoglalása ..... 152
12.2 A tananyagban tanultak részletes összefoglalása ..... 152
12.2.1 Az adatbázis-kezelés alapfogalmai ..... 152
12.2.2 Relációs adatbázisok tervezése ..... 152
12.2.3 A Microsoft Access adatbázisok szerkezete, és kezelése ..... 152
12.2.4 Microsoft Access adatbázisok kialakítása ..... 152
12.2.5 Táblák közötti kapcsolatok kialakítása ..... 152
12.2.6 Rekordok bevitele, rendezése, keresése, szűrése ..... 153
12.2.7 Lekérdezések készítése ..... 153
12.2.8 Speciális feladatok lekérdezésekkel ..... 153
12.2.9 Az Access adatbázisok kezelőfelületének kialakítása ..... 153
12.2.10 Tárolt adatok nyomtatott megjelenítése ..... 153
**13. Kiegészítések ..... 154**
13.1 Irodalomjegyzék ..... 154
13.1.1 Könyv ..... 154
13.1.2 Elektronikus dokumentumok / források ..... 154
**14. Ábrajegyzék ..... 155**
**15. Médiaelemek ..... 158**
**16. Tesztek ..... 160**
16.1 Próbateszt ..... 160
16.2 Záróteszt A. ..... 162
16.3 Záróteszt B. ..... 164
16.4 Záróteszt C. ..... 167

# 1. BEVEZETÉS

Mindannyian tudjuk, hogy napjainkban a tanulásnak nem csak azért van nagy szerepe, mert a szellemi tőke folyamatos felértékelődésének vagyunk tanúi, hanem azért is, mert életünk új és új hívások elé állít bennünket. A mai kor embere nem ritkán arra kényszerül, hogy munkáját számára addig még ismeretlen területen folytassa tovább. Ma már nem elégedhetünk meg a diákévek alatt szerzett ismeretekkel, tudásunkat folyamatosan bővítenünk, gyarapítanunk kell, sőt gyakran teljesen új ismeretanyagot kell elsajátítanunk.

Tudjuk, tapasztaljuk, hogy a nyomatatott és elektronikus sajtón, az Interneten keresztül szinte zúdulnak ránk az adatok, amelyek azonnali feldolgozására gyakran nem áll rendelkezésre a kellő idő, illetve szabad szellemi kapacitás. Az adatok, későbbi feldolgozás céljából történő, átgondolt, logikus szerkezetben való tárolására ezért soha nem volt olyan szükség, mint éppen korunkban. Tankönyvünkkel abban szeretnénk segítséget nyújtani, hogy Ön megismerkedhessen az informatika egyik legizgalmasabb területével, az adatbázis-kezeléssel. Ne csak megízlelje, de sajátítsa is el az ezzel kapcsolatos legalapvetőbb tudnivalókat.

## 1.1 CÉLKITŰZÉS

Könyvünk áttanulmányozása után tisztában lesz az adatbázis-tervezés és kezelés alapvető fogalmaival. Minden ismeret rendelkezésére áll majd ahhoz, hogy megértse az Ön által használt adatbázis működését. Képes lesz az adatokat megfelelően kezelni, kívánt szempontok szerint kiválogatni, rendszerezni, megfelelő formában megjeleníteni. Érteni fogja, a Microsoft Access relációs adatbázis-kezelő rendszer működését, sőt erősen bízunk abban, hogy meg is szereti e nagyszerűen használható alkalmazást. A könyv ismeretanyagát felhasználva nem válik Önből fejlesztő, de munkáját kiválóan értő felhasználó lehet, sőt elegendő tudással rendelkezik majd ahhoz, hogy az elsajátítottak alapján kisebb adatbázisokat készítsen, megkönnyítve ezzel kollégái és saját munkáját.

## 1.2 A KURZUS TARTALMA

1. Az adatbázis kezelés alapfogalmai
2. Relációs adatbázisok tervezése
3. A Microsoft Access adatbázisok szerkezete, és kezelése
4. Microsoft Access adatbázisok kialakítása
5. Kapcsolatok kialakítása az Access adatbázis táblái között
6. Táblák rekordjainak kezelése
7. Lekérdezések készítése
8. Speciális feladatok lekérdezésekkel
9. Az Access adatbázisok kezelőfelületének kialakítása
10. Tárolt adatok nyomtatott megjelenítése

## 1.3 KOMPETENCIÁK ÉS KÖVETELMÉNYEK

Az tananyag elsajátítása után ismerni fogja a relációs adatbázis-kezelés alapfogalmait. Képes lesz felismerni azokat, a munkájával kapcsolatos feladatokat, amelyek megoldása adatbázis-kezelő eszközökkel felgyorsítható, hatékonyabbá tehető.

Megismeri a Microsoft Office programcsomag adatbázis kezelő alkalmazását, a Microsoft Access-t és elsajátítja a program biztonságos használatát. Rendelkezik majd azokkal az elméleti és gyakorlati ismeretekkel, amelyek birtokában, hibátlan relációs adatbázisokat tud tervezni és építeni. Képes lesz az adatok tárolására, és hatékony visszakeresésére, a tárolt adatok közötti összefüggések elemzésére, az adatbázis adatainak képernyőn, vagy nyomtatott jelentések formájában történő megjelenítésére. Az adatbázis tervezése, kialakítása és a tárolt adatok kezelése mellett megismeri azokat a technikákat, amelyek birtokában hatékony, egyszerű, ugyanakkor tetszetős kezelőfelületet fejleszthet saját adatbázisaihoz.

## 1.4 TANULÁSI TANÁCSOK, TUDNIVALÓK

A könyv 10 leckéje felöleli mindazokat az ismereteket, amelyekre Önnek a témában szüksége van. A leckék nem csupán száraz tananyagot tartalmaznak, hanem megpróbálnak együtt gondolkodni az olvasóval. Problémákat, kérdéseket vetnek fel, példákat mutatnak be, feladatokat fogalmaznak meg. Minden lecke végén önellenőrző kérdések segítik az olvasót abban, hogy felmérje saját tudását.

A könyvhöz interneten elérhető elektronikus segédanyagok tartoznak, melyeket az Adatbázis-kezelés tantárgy kurzusában talál meg. Közöttük lelhetők fel azok az adatbázisok, amelyeket a szerző példaként mutat be, az olvasó pedig bátran használhat gyakorlásra.

Kérjük, alaposan olvasson el mindent, kövesse a könyv tanácsait, próbáljon válaszokat keresni a felvetett és az önmaga által megfogalmazott kérdésekre. Másolja gépére a kurzusban található állományokat, próbáljon ki minden példát, és kísérletezzen bátran.

# 2. AZ ADATBÁZIS-KEZELÉS ALAPFOGALMAI

## 2.1 CÉLKITŰZÉS

Tananyagunk első leckéje és második leckéje talán a legkeményebb anyag azok számára, akik most ismerkednek az adatbázis-kezelés tantárggyal. Itt kell ugyanis megtanulniuk azt szép számú alapfogalmat, amelyek valószínűleg nem szerepelnek eddigi informatikai tanulmányaikban, ugyanakkor elengedhetetlen feltételei a további leckék elsajátításának.

Az első lecke végére Ön már ismerni fogja az adatbázisok szerkezeti elmeit. Tudni fogja, hogyan épülnek fel az adatok tárolására alkalmas táblák. Megismeri a rekord, a mező és a mezőérték fogalmát. Tudni fogja, mit jelentenek és hogyan tárolhatók a táblák közötti kapcsolatok. Megismeri azokat a szabályokat, amelyek alkalmazásával a következő órákon valódi adatbázisokat tervezünk.

## 2.2 A LECKE TÉMAKÖREI

- Adat információ, kommunikáció
- Tulajdonság, egyed, tulajdonságtípus, egyedtípus
- Kapcsolat, kapcsolattípus
- Adatmodell, relációs adatmodell
- Tábla, rekord, mező, mezőérték
- Azonosító, összetett kulcs, ismétlődő értékű mező, többértékű mező, idegen kulcs
- Adatbázis

## 2.3 ADAT, INFORMÁCIÓ, KOMMUNIKÁCIÓ

### 2.3.1 Az adat

Talán nem meglepő, de az adatbázis-kezelésben az adat központi helyet tölt be. Az adatbázisokban adatokat tárolunk, rendszerezünk, rendezünk, válogatunk ki, megvizsgáljuk az adatok összefüggéseit, röviden adatokat kezelünk.

A fentiek nagy valószínűséggel senki számára nem jelentenek újdonságot. Érdemes azonban, a mondatot kissé továbbgondolva feltennünk az alábbi kérdéseket:

- Mit jelent pontosan az adat szó?
- Miért adatot tárolunk és miért nem információt?
- Miért jó, ha az adatainkat elektronikusan tároljuk?
- Mit kell elhelyeznünk egy adatbázisban?

Munkánk és vizsgálódásunk tárgya, az adat nem más, mint egy objektum valamilyen jellemzője, tulajdonsága. Az objektum bármi lehet, aminek a tulajdonságai érdekelhetnek bennünket. Lehet tárgy, élőlény, fogalom, lehet ember, állat, növény, egy kavics, egy tevé-

kenység, bármi, aminek tulajdonságai vannak. Az objektum tulajdonsága akkor válik adattá, ha azt olyan formában ábrázoljuk, hogy ember, vagy gép által értelmezhető legyen.




*1. ábra Objektumok, egyedek*

**Az adat egy objektum emberi vagy gépi feldolgozásra, értelmezésre alkalmas formában ábrázolt tulajdonsága.**

A következő mondat adatot tartalmaz.

A vonat 14:30 perckor indul.

A tulajdonság az indulási idő, az objektum a vonat. A tulajdonság értelmezhető, ebben az esetben írásos formában került ábrázolásra. El tudjuk olvasni, képesek vagyunk megérteni. Az informatikában általában objektumoknak nevezzük a világ különböző dolgait. Az adatbázis-kezelésben az **objektum helyett** az **egyed** szót használjuk.

### 2.3.2 Az információ

Az információ és az adat fogalma gyakran keveredik az emberek gondolataiban, pedig valójában a két dolog különbözik egymástól. Az adat egy egyed értelmezhetően ábrázolt tulajdonsága. Az információ pedig az adat jelentése. Az a jelentés, amelyet emberi, vagy

gépi intelligencia az adathoz társít. Az ember esetében az információ a tudatban keletkezik, és csak ott létezik.

**Az információ az adat jelentése, amely a befogadó számára új ismeretet hordozó adat értelmezése során keletkezik.**

**Adat:** A vonat 18:10 perckor indul.

**Információ:** Az információ az, amit a fenti adatból egy bizonyos befogadó megtud.

Jó, ha már most észrevesszük, hogy az első mondat az adat (18:10) mellett az egyedet (egy bizonyos vonat) is megnevezi.




*2. ábra Adat, és információ*

### 2.3.3 Különbségek adat és információ között

#### *Az információ nem ábrázolható*

Az információ tehát adatból jöhet létre, az értelmezés során. Az adatról tudjuk, hogy az egy ábrázolt tulajdonság, tehát szükségszerűen **ábrázolható**. Amikor egy adatot értelmezünk, úgymond dekódoljuk, hozzájutunk jelentéséhez az információhoz. **Az információ a tudatban keletkezik, és csak ott létezik.** Ebből következően **az információ nem tárolható el, az adat azonban igen**. Ha az információt ábrázolni, tárolni akarjuk, adattá kell alakítanunk, azaz kódolnunk kell.

Tegyük fel, hogy egy ismerősünk telefonon megkér bennünket, nézzük meg az Interneten, hogy mikor indul a vonata Szegedre. A megfelelő oldalon megnézzük a vonat

indulási idejét: 18:10. A weblapon látott számok egy objektum (vonat) egy tulajdonságát (indulási idő) azaz adatot ábrázolnak. A látottakat értelmezve hozzájutunk az adat jelentéshez. Az információ már ott van a tudatunkban. Ahhoz azonban, hogy barátunkkal is tudassuk a vonat indulását, a megszerzett információt ismét adattá kell kódolnunk. Írunk egy SMS-t: *A vonat délután fél háromkor indul.* Az információt írott szöveggé, adattá kódoltuk. Ebben a formában már továbbíthatjuk tudatlan cimboránknak.

### *Az adat objektív, az információ szubjektív*

Egy ábrázolt tulajdonság (leírt szöveg, megrajzolt ábra, egy hangfelvétel) fizikailag mindenki számára egyforma, azaz objektív. Az információ, a jelentés, amit az adathoz társítunk a tudatunk terméke, tehát szubjektív.

Ha két különböző ember megnézi ugyanannak a vonatnak az indulását, mindketten pontosan ugyanazt az adatot látják (18:10) Az értelmezés során azonban más-más jelentés alakulhat ki bennük. Egyikük azt gondolhatja, hogy sietnie kell, mert az indulásig még vásárolni szeretne, míg másikuk bosszankodhat, mert az indulásig hátralévő órát tétlen várakozással fogja tölteni.




*3. ábra Az adat objektív, az információ szubjektív*

### 2.3.4 A kommunikáció

Shannon és Weaver (1949) modellje szerint a kommunikáció során két kommunikáló fél, az „adó” és a „vevő” a kommunikációs csatornán keresztül információt cserél. Az adó információt ad át a vevőnek. Az eddig elmondottak alapján belátható, hogy ez nem valósítható meg közvetlenül, hiszen az információ fizikailag nem ábrázolható, így a kommunikációs csatornán sem továbbítható.

A kommunikáció során adónak fizikailag létező adattá kell kódolnia az információt. A vevő, miután átvette az adatot, dekódolja, értelmezi azt, így információhoz jut.

Az információ átadása során tehát két átalakítási folyamat is lejátszódik. A kódolás és dekódolás is a tudat működése, ezért eredmény szubjektív. Emiatt vevőben keletkező információ gyakran eltér az adó által birtokolt eredeti információtól.

Ha pl. tudjuk, hogy a vonat 18:10-kor indul, de barátunknak azt mondjuk „hat tízkor”, akkor a benne kialakuló információ már nem egyezik meg az eredetivel, hiszen nem fogja tudni, hajnali, vagy délutáni-e az időpont.

Az adó szeretné, ha a vevőben a megfelelő információ alakulna ki, azért az információ adattá kódolásakor igyekszik figyelembe venni a vevő ismereteit, pillanatnyi lelki állapotát és egy sor egyéb szempontot. Ezért úgy kódol, hogy a keletkezett adatok a lehető legegyszerűbben legyenek az eredeti információvá értelmezhetők. Ilyenkor mondjuk, hogy az adó érthetően kommunikál.

**Adatbázisokat azért hozunk létre, mert szeretnénk információinkat adattá kódolva úgy eljuttatni másokhoz, hogy azok akár térbeli, akár időbeli eltéréssel tudjanak információhoz jutni az adatok dekódolásával.**

## 2.4 MIT JELENT AZ ADATBÁZIS-KEZELÉS?

Az tudásközpontú világunk egyik fő jellemzője az információ, a tudás felértékelődése. E világban kiemelt szerepet kap a megszerzett tudás informatikai eszközökkel történő rögzítése, tárolása és a tárolt tudás hatékony újrafelhasználása. A tudás rögzítése azonban egyáltalán nem kézenfekvő dolog, hiszen tudjuk, hogy a rendelkezésünkre álló információ csak adattá kódolva válik átadhatóvá, tárolhatóvá, kezelhetővé. Éppen ebben rejlik az adatok tárolásának és későbbi információvá értelmezésének, azaz az adatbázis-kezelésnek a nehézsége, egyben jelentősége.

Az adatbázis-kezeléssel kapcsolatos feladatok két fontos csoportba sorolhatók:

- a tudás, az információ adattá kódolása és **tárolása**, illetve
- a tárolt adatok tér, és/vagy idő szerint eltérő pozícióban történő **újraértelmezése**.

A tudás tárolásakor:

- ismereteinket adattá kódoljuk,
- az adatokat informatikai eszközökkel,
- jól meghatározott szerkezetben,
- hatékonyan újraértelmezhető formában tároljuk.

Az adatok értelmezésekor:

- az adatbázisban rögzített adatokat kezeljük,
- velük műveleteket végzünk,
- illetve értelmezzük azokat,
- összefüggéseket igyekszünk feltárni,
- törekszünk megszerezni az adatok által hordozott információt.

**Az adatbázis-kezelés kétpólusú tevékenység. Feladatait egyfelől a tudás adattá kódolása, az adatok informatikai eszközökkel, jól meghatározott szerkezetben, hatékonyan értelmezhető formában történő tárolása, másfelől a tárolt adatok rendezése, kezelése, és újraértelmezése alkotja.**

## 2.5 AZ ADATBÁZISOK FELÉPÍTÉSE

Az adatbázis-kezelésben **munkánk hatékonyságát eredendően meghatározza az adatbázis szerkezete**. Ez dönti el ugyanis, hogy miről, mit, és milyen összefüggésekben tárolunk el. E szerkezet megértéséhez az alábbi kérdésekre kell válaszokat keresnünk:

- Hogyan épül fel egy adatbázis szerkezete?
- Milyen szerkezeti elemek alkotják?
- Mitől lehet ez a szerkezet jobb, vagy rosszabb?

A következőkben az adatbázist alkotó szerkezeti elemeket ismerjük meg. Arról beszélünk, miket is tárolunk el egy adatbázisban, mit jelent az egyed, egyedtípus, tulajdonság, tulajdonságtípus, kapcsolat és kapcsolattípus.

### 2.5.1 Egyed és tulajdonság

Az adat meghatározásában található „**objektum**” kifejezést, az adatbázis-kezelésben az **egyed** szóval helyettesítjük. Az adatbázisokban pedig adatokat, azaz **egyedek tulajdonságait** tároljuk.

**Az egyed egy konkrét (egyedi) objektum, amelynek tulajdonságait tároljuk. A tulajdonság az egyed valamely konkrét jellemzője.**

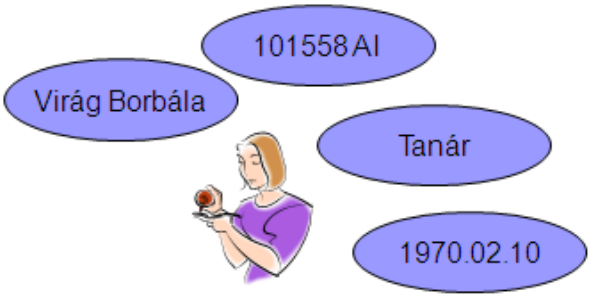
Ha az egyed például a Virág Borbála nevű személy, akkor tulajdonságai az alábbiak lehetnek:

Virág Borbála,

tanár,

1970.02.10,

101558 AI.



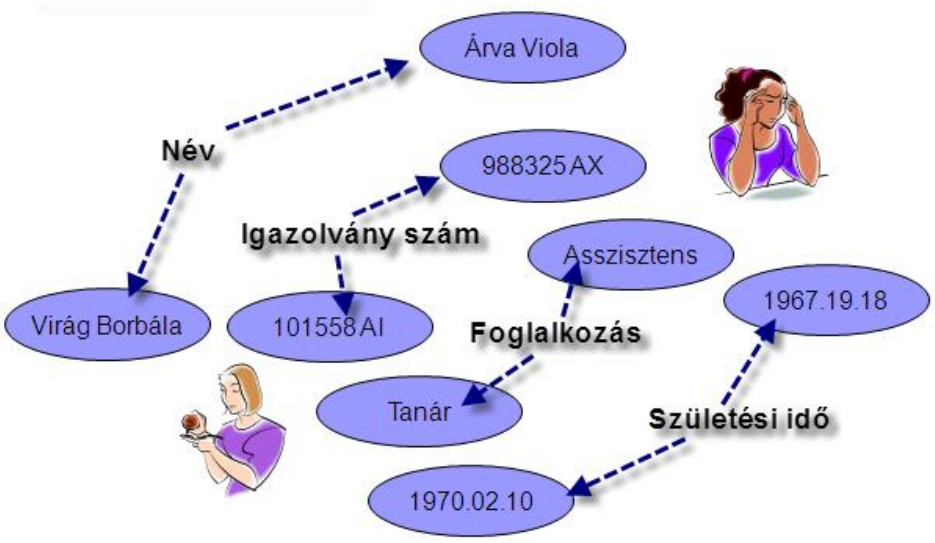

*4. ábra Egyed, és tulajdonságok*

### 2.5.2 Tulajdonságtípus

Amikor adatbázis készítünk, általában sok-sok egyed számos, különböző tulajdonságát tároljuk el. Az egyedek tulajdonságait csoportosíthatjuk, tipizálhatjuk aszerint, hogy milyen jellemzőt írnak le.

Az alábbi példában a „Virág Borbála”, és az „Árva Viola” tulajdonságok ugyanabba a tulajdonságcsoportba tartoznak, hiszen mindkettő név. A „tanár” és „asszisztens” tulajdonságok foglalkozásokat adnak meg, az 1970.02.10 és a z 1967.19.18 születési dátumok, a 101558 AI, 988325 AX tulajdonságok pedig személyi igazolványszámok.

| **Név** | **Foglalkozás** | **Születési idő** | **Személyi igazolványszám** |
|---|---|---|---|
| Virág Borbála | tanár | 1970.02.10 | 101558 AI |
| Árva Viola | asszisztens | 1967.19.18 | 988325 AX |




*5. ábra Tulajdonságtípusok*

**Az azonos jellemzőket meghatározó tulajdonságok halmazát tulajdonságtípusnak nevezzük.**

A tulajdonságtípusokat elnevezzük, ezzel a névvel utalunk arra, hogy milyen jellemzőt írnak le a tulajdonságtípusba tartozó tulajdonságok. A *név*, a *foglalkozás*, a *születési idő*, *személyi igazolványszám* mind tulajdonságtípusok.

### 2.5.3 Egyedtípus

Nem csak a tulajdonságok, de az egyedek is halmazokba sorolhatók, mégpedig aszerint, hogy milyen tulajdonságtípusok jellemzik őket.

Azok az egyedek, amelyek mindegyike rendelkezik névvel, foglalkozással, születési idővel, és személyigazolvány számmal, ugyanabba az egyedhalmazba, más néven **egyedtípusba** tartoznak.

**Az azonos tulajdonságtípusokkal jellemezhető egyedek halmazát egyedtípusnak nevezzük.**

Az adatbázis-kezelésben az egyedtípusokat is névvel látjuk el. A fenti példában mindkét egyed a *személyek* egyedtípusba tartozik.

### 2.5.4 Kapcsolatok

Egy adatbázisban általában több egyedtípust is tárolunk. Az adatbázis egyedtípusai nem függetlenek egymástól, közöttük általában valamilyen viszony, kapcsolat van. Az egyedtípusok közötti kapcsolatokból szintén nagyon hasznos és fontos információk nyerhetők.

**Az egyedtípusok közötti viszonyt kapcsolatnak nevezzük. Egy kapcsolat mindig két egyedtípus közötti viszonyt ír le.**

A közös adatbázisban tárolt *Személyek* és az *Autók* egyedtípus között kapcsolat lehet, hiszen egy személy tulajdonosa lehet egy autónak. Azaz a *Személyek* egyedtípus egy egyede tulajdonosi kapcsolatban lehet az *Autók* egyedtípus egy egyedével.

Ha egy autógyár adatbázisában külön egyedtípusban tároljuk a személygépkocsi *Motorok*-at, és egy külön egyedtípusban az autó *Karosszériák*-at, akkor a két egyedtípus egyedei között kapcsolat van. Hiszen minden motort beépítenek egy karosszériába.

Tegyük fel, hogy egy harmadik adatbázisban egy iskola adatait rögzítjük. A *Tanárok* egyedtípusban a tanár egyedek, az *Osztályok* egyedtípusban az iskola osztályainak, mint egyedeknek az adatai tárolódnak. A *Tanárok* és *Osztályok* között kapcsolat van, hiszen a tanárok tanítják valamilyen tárgyra az osztályokat.

Az adatbázisokban egyedtípusokba csoportosított egyedek, tulajdonságtípusokba sorolt tulajdonságait, valamint az adatbázis egyedtípusai közötti kapcsolatokat tároljuk.

### 2.5.5 A kapcsolatok tipizálása, kapcsolattípusok

Az egyedekhez és tulajdonságokhoz hasonlóan, a kapcsolatokat tipizálhatjuk, csoportosíthatjuk. Két egyedtípus (’A’ és ’B’) kapcsolatát aszerint sorolhatjuk típusokba, hogy az ’A’ egyedtípus egyedei, a ’B’ egyedtípus hány egyedével lehetnek kapcsolatban, és a ’B’ egyedtípus egyes egyedei az ’A’ egyedtípusban hány egyedhez kapcsolódhatnak.

A Motorok-Karosszériák egyedtípusok esetében egy motornak csupán egyetlen karosszériával van kapcsolata, és egy karosszéria is csak egy motorhoz kapcsolódik.

Az Személyek-Autók kapcsolatban egy személy, akár több autóhoz is kapcsolódhat tulajdonosként, de egy autónak csak egy tulajdonosa lehet.

Az Osztályok-Tanárok viszony esetén egy tanár több osztállyal is lehet tanári kapcsolatban, mint ahogyan egy osztálynak több tanára is lehet.

#### *Egy az egyhez kapcsolat (1:1)*

Az egy az egyhez kapcsolat esetén az 'A' egyedtípus bármelyik egyede a 'B' egyedtípus egyetlen egyedéhez kapcsolódhat, és a 'B' egyedtípus bármely egyede is csak egyedhez kapcsolódhat az 'A' egyedtípusban.

A kapcsolat jelölése: 1:1

A Motorok és Karosszériák között 1:1 kapcsolat van.

#### *Egy a többhöz (1:N) kapcsolat*

Az egy a többhöz kapcsolat esetén az 'A' egyedtípus bármely egyede a 'B' egyedtípus több egyedéhez is kapcsolódhat, azonban a 'B' egyedtípus egyedei az 'A' egyedtípusban csak egy egyedhez kapcsolódhatnak.

Jelölése: 1:N

A Személyek-Autók 1:N kapcsolat.

#### *Több a többhöz kapcsolat*

A több a többhöz kapcsolatban az 'A' egyedtípus egyedei a 'B' egyedtípus több egyedéhez kapcsolódhatnak, és ez fordítva is igaz. A 'B' egyedtípus egyedei az 'A' egyedtípus több egyedéhez kapcsolódhatnak.

Az Osztályok-Tanárok kapcsolat ilyen több a többhöz kapcsolat.

## 2.6 AZ ADATMODELL

Amikor adatbázist készítünk, feladatunk valójában nem más, minthogy az egyedeket megfelelő egyedtípusokba, tulajdonságaikat pedig tulajdonságtípusokba sorolva tároljuk. Ezt követően valamilyen módon rögzítenünk kell az egyedtípusok közötti kapcsolatokat is. Mindezt természetesen úgy kell elvégeznünk, hogy a keletkezett adathalmaz a lehető legkisebb helyet foglalja el, ugyanakkor gyorsan, egyszerűen és hatékonyan legyen kezelhető.

Az egyedek, egyedtípusok, tulajdonságok, tulajdonságtípusok, kapcsolatok, kapcsolattípusok tárolásának módját az **adatmodell** határozza meg. Különféle adatmodellek állnak rendelkezésre, és mindegyik más-más lehetőségeket kínál az adatok tárolására.

> **Az adatmodell egy szabályrendszer, amely egy adatszerkezet felépítésének elveit határozza meg. Pontosan rögzíti, hogyan tárolhatók az egyedek, egyedtípusok, tulajdonságok és tulajdonságtípusok, valamint az egyedtípusok közötti kapcsolatok.**

Az alkalmazott adatmodellen múlik az is, hogy milyen műveletek lesznek elvégezhetők az adatbázisban tárolt adatokkal.

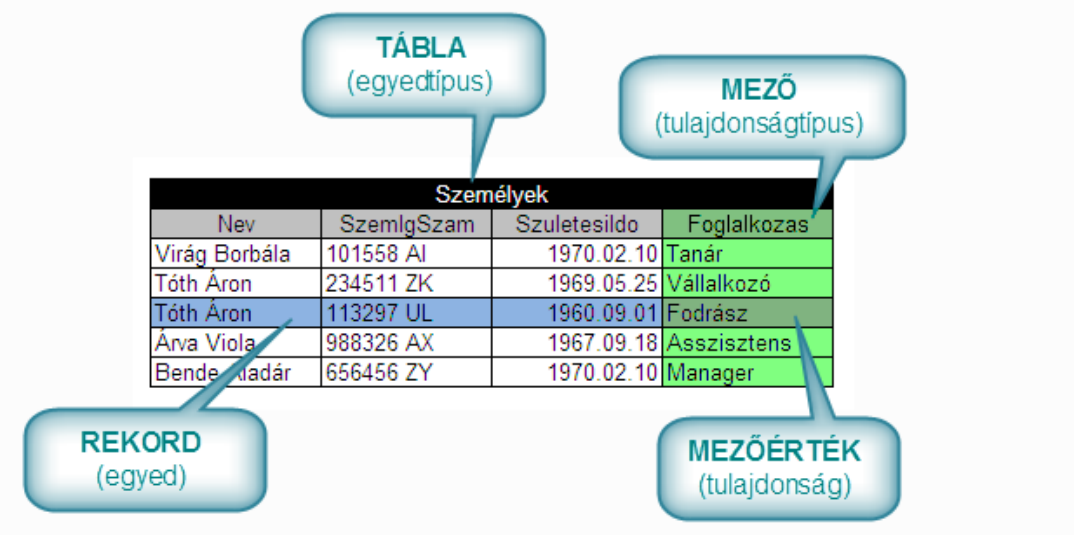
### 2.6.1 A relációs adatmodell

Az adatbázis-kezelésben többféle adatmodellt ismerünk. Ilyen a hierarchikus a hálós, az objektumorientált, és a relációs adatmodell. **Tananyagunkban a relációs adatmodellt mutatjuk be.** Ez a legegyszerűbb, legkönnyebben kezelhető, éppen ezért legelterjedtebb adatmodell. Tanulmányozásakor talán legfontosabb, hogy az egyedtípusok tárolására használt struktúra, a tábla szerkezetét megértsük!

#### *Tábla, mező, rekord, mezőérték*

A relációs adatmodellben egy egyedtípus minden adata egy egyszerű táblában tárolódik. A tábla mátrix, amelyet sorok és oszlopok alkotnak. Minden oszlop egy tulajdonságtípusnak felel meg. A sorokban az egyedek helyezkednek el, a sorok celláiban pedig a tulajdonságok kapnak helyet.

> A relációs adatmodellben az egyedtípus helyett a tábla megnevezést használjuk.
>
> A tulajdonságtípus fogalomnak a mező felel meg.
>
> Az egyedeket rekordoknak nevezzük.
>
> A tulajdonságot mezőértéknek hívjuk.




*6. ábra Tábla, mező, rekord, mezőérték*

### *A relációs adatmodell szabályai*

A relációs adatmodell tábláinak felépítését az alábbi szabályok határozzák meg:

- Egy táblán belül nem lehet két teljesen egyforma rekord. Azaz táblán belül minden rekordnak –valamiben- különböznie kell a többitől.
- Minden rekordban azonosnak kell lennie a mezők sorrendjének.
- Minden rekordban azonos számú mezőknek kell lennie.
- A táblában nem lehetnek többértékű mezők.

**Rekordok különbözősége**

Az, hogy a rekordoknak különbözniük kell, azt jelenti, hogy mindenképpen kell, hogy legyen olyan mező (esetleg mező kombináció), amelynek értéke minden rekordban egyedi.

| Személyek | | | |
|---|---|---|---|
| Nev | SzemIgSzam | Szuletesildo | Foglalkozas |
| Virág Borbála | 101558 AI | 1970.02.10 | Tanár |
| Tóth Áron | 234511 ZK | 1969.05.25 | Vállalkozó |
| Tóth Áron | 113297 UL | 1960.09.01 | Fodrász |
| Árva Viola | 988326 AX | 1967.09.18 | Asszisztens |
| Bende Aladár | 656456 ZY | 1970.02.10 | Manager |

*7. ábra Egyedi értékű mező*

**Mezők sorrendje**

Az hogy a mezők sorrendjének azonosnak kell lennie, azt jelenti, hogy nem tehetjük meg, hogy az egyik rekordban a *Nev* mező az első, a másikban pedig pl. a *SzemIgSzam*.

**Mezők száma**

Nagyon fontos szabály! A Személyek táblában tárolhatjuk az egyes egyedek (személyek) telefonszámait is. Van, akinek egy, van, akinek kettő telefonja van, de könnyen elképzelhető, hogy egy személy akár öt-hat telefonon is elérhető.

Ez a szabály nem engedi meg, hogy az egyes rekordokban több, míg másokban kevesebb telefonszámot tároló mezőt hozzunk létre.




*8. ábra Eltérő mezőszám*

**Többértékű mezők**

Mivel nem tehetjük meg, hogy az egyes rekordokban eltérő számú mezőt hozunk létre, a telefonszámokat egy mezőben tároltuk. Az olyan mezőket, amelyek az egyes rekordokban több értéket is tartalmazhatnak (Telefonszam) többértékű mezőknek nevezzük.




*9. ábra Többértékű mező*

A relációs adatmodell nem engedi meg a többértékű mezők használatát! Ez azt jelenti, hogy a személyek tábla sajnos az előbbi és az utóbbi formában sem használható.

A helyes kialakításról a tananyag későbbi, Relációs adatbázisok kialakítása leckéjében olvashatunk majd.

### 2.6.2 Speciális mezők a relációs adatmodellben

Bár a relációs adatmodell szabályai nem szólnak részletesen a speciális mezőkről, az azonosító, az összetett kulcs, a többértékű mező, az ismétlődő értékű mező, és az idegen kulcs fogalmának megismerése a későbbi tananyag megértésének fontos feltétele.

### *Azonosító, vagy kulcs*

A relációs adatmodell szerint készült táblákban kell, hogy legyen olyan mező, amelynek értéke minden rekordban más és más, két rekordban sohasem egyforma, így alkalmas arra, hogy a rekordokat egyértelműen megkülönböztesse egymástól. Az alábbi ábrán látható személyek táblában például a `SzemIgSzam` mező lehet azonosító, mert értékei biztosan különböznek minden személy rekordja esetén.




| Személyek | | | |
|---|---|---|---|
| Nev | SzemIgSzam | Szuletesildo | Foglalkozas |
| Virág Borbála | 101558 AI | 1970.02.10 | Tanár |
| Tóth Áron | 234511 ZK | 1969.05.25 | Vállalkozó |
| Tóth Áron | 113297 UL | 1960.09.01 | Fodrász |
| Árva Viola | 988326 AX | 1967.09.18 | Asszisztens |
| Bende Aladár | 656456 ZY | 1970.02.10 | Manager |

*10. ábra Azonosító, vagy kulcs*

**Az azonosító, vagy más néven kulcs, egy egyedtípus olyan tulajdonságtípusa, amelynek értékei egyértelműen megkülönböztetik egymástól az egyedtípus egyedeit.**

Adatbázisok készítésekor általában kerüljük az egyedtípus valódi tulajdonságtípusainak azonosítóként történő használatát.

Gyakoribb, hogy az egyedtípushoz hozzárendelünk egy új mezőt, amelyben minden rekord egy sorszámot kap. Ez a sorszám különbözteti majd meg az egyedtípus rekordjait. Az adatbázis kezelőjének kell gondoskodni arról, hogy ez a sorszám valóban különböző legyen az egyes rekordok esetén.

| Személyek | | | | |
|---|---|---|---|---|
| Sz_Az | Nev | SzemIgSzam | Szuletesildo | Foglalkozas |
| 1 | Virág Borbála | 101558 AI | 1970.02.10 | Tanár |
| 2 | Tóth Áron | 234511 ZK | 1969.05.25 | Vállalkozó |
| 3 | Tóth Áron | 113297 UL | 1960.09.01 | Fodrász |
| 4 | Árva Viola | 988326 AX | 1967.09.18 | Asszisztens |
| 5 | Bende Aladár | 656456 ZY | 1970.02.10 | Manager |

*11. ábra Sorszám az azonosítóban*

Előfordulhat, hogy egy táblában több olyan mező is van, amely alkalmas az azonosító szerepének betöltésére. Az ilyen mezőket alternáló kulcsoknak vagy kulcs jelölteknek nevezik.

Azt a mezőt, amit kiválasztunk, és kulcsként használjuk elsődleges kulcsnak is hívják.

### *Összetett kulcs*

Előfordulhat, hogy olyan táblát hozunk létre, amelyben egyetlen olyan mező sincsen, amit azonosítóként használhatnánk. Ilyen esetben kaphat szerepet az összetett kulcs.

Tegyük fel, hogy táblában mobil telefonok tulajdonságait szeretnénk tárolni!

Szükségünk van a telefon körzetszámára, a hívószámra, a tulajdonos nevére, valamint lakcímére.

| Telefonok | | | | |
|---|---|---|---|---|
| Körzet | Telefon | Nev | Város | Utca |
| 20 | 324 7889 | Virág Borbála | Eger | Kincses u. 3. |
| 30 | 324 7889 | Tóth Áron | Eger | Fodros u. 12. |
| 36 | 320 512 | Tóth Áron | Eger | Klopkovics u. 38. |
| 36 | 321 321 | Árva Viola | Eger | Szerencse tér 2. |
| 70 | 788 9130 | Árva Viola | Eger | Tűzkerék u. 52. |
| 30 | 788 9130 | Bende Aladár | Eger | Vasút tér 9. |

*12. ábra Összetett kulcs*

Jól látszik, hogy a Körzet mező nem lehet azonosító, hiszen a 30-as és 36-os érték több rekordban is előfordul. A Telefon mezőben is találunk ismétlődést, tehát ez sem alkalmas a rekordok azonosítására.

A Körzet és a Telefon mezők kombinációja azonban már minden rekordban egyedi értéket alkot. A két mező együtt lehet azonosító.

**Az összetett kulcs egy tábla kettő vagy több mezőjéből alkotott mezőkombináció, amelyet a rekordok azonosítására használunk.**

Az összetett kulcsot tartalmazó tábla rekordjaiban az összetett kulcs minden elemében kell, legyen érték!

### *Idegen kulcs*

Az adatbázisokban nem csupán az egyedtípusokat, hanem azok közti kapcsolatokat is el kell tárolnunk. A táblák közötti kapcsolatok tárolására használjuk az idegen kulcsokat.

**Idegen kulcsnak nevezzük egy tábla olyan mezőjét, amelynek értékei egy másik tábla rekordjait azonosítják.**

Tegyük fel, hogy feladatunk, hogy személygépkocsik és tulajdonosaik adatinak tárolását szeretnénk megvalósítani.

Az autók rendszámát, gyártmányát, típusát, és hengerűrtartalmát, a tulajdonosok nevét, személyi igazolványszámát, valamint lakcímét szeretnénk tárolni.

Az *Autók* táblában a *RendSzam* a *Személyek* táblában a *SzemIgSzam* lesz az azonosító.

Az *Autók* egyedtípusban úgy jelezzük a tulajdonost, hogy a táblában elhelyezünk a *Tulajdonos* mezőt, amelynek minden értéke a *Személyek* tábla azonosító mezőjéből származik, azaz azonosít egy személyt.

Az *Autók* tábla *Tulajdonos* mezője idegen kulcs.

Az idegen kulcs segítségével a táblák közötti kapcsolatokat tudjuk eltárolni a relációs adatmodellnek megfelelő adatbázisokban.

A kapcsolatok ilyen tárolása miatt a relációs adatmodellben két tábla között közvetlenül csak 1:1, vagy 1:N kapcsolat lehetséges.

Az N:M kapcsolat kialakításához kisebb "trükkre" lesz szükség. Erről a relációs adatbázisok kialakításba részben olvashat majd.

### *Többértékű mező*

Egyáltalán nem ritka jelenség, hogy egy egyedtípus valamelyik tulajdonságtípusa több értéket is felvehetne az egyes egyedek esetében.

Gondoljunk csak például arra, hogy személyek telefonszámait tároljuk. Egy személy több telefonszámmal is rendelkezhet, így kézenfekvő megoldásnak tűnik, hogy a Személyek tábla Telefon mezőjében rekordonként több számot is eltároljunk. Ilyen esetben többértékű mezőről beszélünk.

**A többértékű mező egy egyedtípus olyan tulajdonságtípusa, amely az egyes rekordokban több értéket is tartalmazhat.**




*13. ábra Többértékű mező*

A fenti tábla Telefon mezője többértékű mező. Egy rekordban több telefonszám is megtalálható...

Nagyon fontos tudnunk, hogy a relációs adatmodell nem engedi meg a többértékű mezők használatát.

A relációs adatmodellnek adatbázisok kialakításakor mindenképpen meg kell szűntetnünk a többértékű mezőket!

Az ezzel kapcsolatos technikákról a Relációs adatbázisok kialakítása szakaszban olvashat.

### *Ismétlődő értékű mező*

A többértékű mezővel szemben, az ismétlődő értékű mezőt nem tiltja a relációs adatmodell. A későbbiekben azonban látni fogjuk, hogy az ilyen mezők fölöslegesen megnövelik az adatbázis mértet, és általában kezelési nehézségeket is okoznak. A következő leckében megtanuljuk, hogy az adatbázis tábláinak megtervezésekor igyekeznünk kell elkerülni az ilyen mezők létrehozását.

**Egy mező akkor ismétlődő értékű, ha ugyanaz az a mezőérték több rekordban is előfordul.**




*14. ábra Ismétlődő értékű mezők*

### 2.6.3 Az adatbázis

Tananyagunk eddigi részében többször használtunk már az adatmodell és az adatbázis fogalmat, de csupán az adatmodell meghatározását láttuk. Előbbi, mint tudjuk egy adathalmaz tárolásának elvi szerkezetét adja meg. Az adatbázis az adatmodell szabályainak megfelelő tényleges adathalmaz. Ha úgy tetszik, az adatmodell a szabály, az adatbázis a szabály megvalósítása.

**Az adatbázis olyan ténylegesen tárolt adathalmaz, amely egy adatmodell szabályainak megfelelően alakítottak ki.**

**Az adatbázisban egyedtípusokba sorolt egyedek, tulajdonságtípusokba sorolt tulajdonságait, és az egyedtípusok egyedei közötti kapcsolatokat tároljuk.**

Minden adatbázis valamilyen adatmodell alkalmazása. Éppen ezért az adatbázisok jellemezőjeként szokás megemlíteni, hogy milyen adat modell alapján készültek.

Az általunk készített adatbázisok mind a relációs adatmodell alapján készülnek majd, tehát relációs adatbázisok lesznek.

A hálós, hierarchikus, illetve objektum orientált adatmodelleket ritkábban használják. Alkalmazásukkal hierarchikus adatbázisok, hálós adatbázisok, illetve objektum orientált adatbázisok készíthetők.

## 2.7 ÖSSZEGZÉS

Az adat egy egyed valamilyen értelmezhető módon ábrázolt tulajdonsága. Az információ az adat jelentése.

Az adat tárolható, az információ nem. Az információ csak adattá kódolva tárolható.

Az egyed, egy adatbázisban tárolt egyedi objektum. Egy bizonyos személy, egy bizonyos könyv, egy konkrét árucikk, egy meghatározott autó, stb.

A tulajdonság egy egyedre jellemző adat:

*Borus Ádám, Családom, és egyéb állatfajták, 12500 Ft, TXA 231...*

A tulajdonságtípus azonos jellemzőt meghatározó tulajdonságok halmaza, amelyet egy megnevezéssel illetünk: *név, cím, ár, rendszám* stb.

Az egyedtípus olyan egyedek halmaza, amelyeket azonos tulajdonságtípusok jellemeznek: *hallgatók, könyvek, árucikkek, autók* stb.

Az egy adatbázisban tárolt egyedtípusok között kapcsolat lehet.

A kapcsolatokat aszerint csoportosítjuk, hogy az 'A' egyedtípus egyedei hány egyedhez kapcsolódhatnak a 'B' egyedtípusban.

A kapcsolatok három típusba (1:1, 1:N, N:M) sorolhatók.

Az adatbázisokban egyedtípusokba sorolt egyedek tulajdonságtípusokba sorolt tulajdonságait, és az egyedtípusok egyedei közötti kapcsolatokat tároljuk el.

Az egyedtípusok, ez egyedek, a tulajdonságtípusok, tulajdonságok és kapcsolatok tárolásának módját az adatmodell határozza meg. Az adatmodell egy adatszerkezet elvi leírása.

A relációs adatmodellben, táblákban tároljuk az egyedtípusok egyedeinek adatait!

A táblák oszlopai a mezők, amelyek a tulajdonságtípusoknak felelnek meg.

A sorok a rekordok. Minden rekord egy egyed tulajdonságait tárolja.

A relációs adatmodell szabályai:

- Táblán belül minden rekordnak különböznie kell a tábla többi rekordjától.
- Táblán belül, rekordonként meg kell egyeznie a mezők számának.
- Táblán belül, rekordonként meg kell egyeznie a mezők sorrendjének.
- Egyetlen táblában sem lehet többértékű mező

Az azonosító olyan mező, amely minden rekordban más értékű, így alkalmas a rekordok azonosítására.

Az összetett kulcs több mező kombinációjával kialakított azonosító.

Az idegen kulcs egy tábla olyan mezője, amelynek értékei egy másik tábla azonosítójából származnak, segítségével kapcsolhatók össze a különböző táblák rekordjai.

A többértékű mezőben több tulajdonság helyezkedhet el a rekordon belül.

A relációs adatmodellben nem engedi meg a többértékű mezők használatát!
Az adatmodell egy adatszerkezet elvi leírása, az adatbázis az adatmodell megvalósítása.

## 2.8 ÖNELLENŐRZŐ KÉRDÉSEK

1. Hogyan tárolható az információ?
2. Mi tárolunk egy adatbázisban?
3. Mi az a relációs adatmodell?
4. Mi kell tudni a többértékű mezőkről?
5. Hogyan nevezzük az olyan azonosítót, amely több mező kombinációjából áll?
6. Mit értünk kapcsolat alatt?
7. Milyen kapcsolattípusnak megfelelő kapcsolat lehet egy relációs adatbázis két táblája között?
8. Mi a különbség adatmodell és adatbázis között?

# 3. RELÁCIÓS ADATBÁZISOK TERVEZÉSE

## 3.1 CÉLKITŰZÉS

A relációs adatmodellnek megfelelő, relációs adatbázisokban egyedtípusok egyedeinek tulajdonságtípusokba tartozó tulajdonságait táblákban tároljuk a közöttük lévő kapcsolatokat pedig idegen kulcsokkal írjuk le.

A táblák egyszerű szerkezetűek, a relációs adatmodell szabályai miatt azonban kialakításukkor kisebb-nagyobb nehézségekbe ütközhetünk.

Ismerkedjünk meg ezekkel a problémákkal, és megoldásuk lehetséges módjaival is!

A következő oldalakon megtanulhatja, hogyan alakíthat ki a relációs adatmodellnek megfelelő, könnyen, és biztonságosan kezelhető adatbázisokat.

## 3.2 A LECKE TÉMAKÖREI

- Egyedtípusok a relációs adatmodellben
- A redundancia és következményei
- Kezelési anomáliák
- Funkcionális függés
- Normálformák
- Normalizálás
- Adatbázis tervének ábrázolása

## 3.3 EGYEDTÍPUSOK TÁROLÁSA A RELÁCIÓS ADATMODELLBEN

Egy adathalmaz tárolásakor első feladatunk, hogy kialakítsuk az egyedtípusok adatinak tárolására alkalmas, a relációs adatmodell szabályainak megfelelő adatszerkezeteket, a táblákat.

Egy táblán belül – a tárolt mezőértékek tekintetében – minden rekordnak különböznie kell a többitől.

Ugyanakkor a rekordok szerkezete azonos kell, legyen, azaz a mezők száma, és sorrendje nem különbözhet.

Leggyakrabban a mezőszám, és a többértékű mezők okozhatnak problémát.

Példaként tekintsünk egy korábbihoz hasonló feladatot:

Személyek adatait kell nyilvántartanunk: személyi igazolvány szám, név, állandó lakcím(irányítószám, város, utca), telefonos elérhetőség (telefonszám, vezetékes telefon-e, mikor hívható).

A problémát az okozza, hogy az egyes személyeknek különböző számú telefonjai lehetnek.

Ha nem relációs adatbázist készítenénk, megtehetnénk, hogy a telefonokat jellemző tulajdonságokat egy-egy többértékű mezőben tároljuk.

Az ilyen tábla alkalmas lenne ugyan az adatok tárolására, a relációs adatmodell követelményeinek azonban nem felelne meg.

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321<br>+36/70/788 9130<br>+36/20/315 5551 | Igen<br>Nem<br>Nem | 08:00-13:00<br>09:30-18:00<br>08:00-20:00 |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. | +36/36/321 321<br>+36/30/368 5552 | Igen<br>Nem | 08:00-13:00<br>08:00-20:00 |
| ... | ... | ... | ... | ... | ... | ... | ... |

*15. ábra Többértékű mezők*

Megoldás lehetne, hogy az egyes rekordokban több, másokban kevesebb telefon adatokat tároló mezőt hoznánk létre. Természetesen ez a tábla sem használható a relációs adatmodellben, ugyanis itt is megszegnénk a relációs adatmodell szabályait. Az egyes rekordokban eltérne a mezők száma!

| Személyek | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor | Szam2 | Vezetekes2 | Mikor2 | Szam3 | Vezetekes3 | Mikor3 |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 | | | | | | |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 | | | | | | |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 | | | | | | |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 | +36/70/788 9130 | Nem | 09:30-18:00 | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 | +36/30/368 5552 | Nem | 08:00-20:00 | | | |
| ... | ... | ... | ... | ... | ... | ... | ... | | | | | | |

*16. ábra Eltérő mezőszám*

Marad az alábbi megoldás! Azokat a személyeket, akiknek több telefonjuk is van, több rekordban is tároljuk, mindig más telefon megadásával. Így a relációs adatmodell minden szabályát betartottuk!

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

*17. ábra Relációs adatmodell követelményeit teljesítő tábla*

A következő oldalakon megtudhatjuk, hogy bár ez a megoldás megfelel a relációs adatmodell szabályainak, gyakorlati használhatósága azonban erősen megkérdőjelezhető...

## 3.4 A REDUNDANCIA ÉS KÖVETKEZMÉNYEI

Az előző feladat megoldása közben elért siker csak látszólagos. Azért, hogy betarthassuk a relációs adatmodell szabályait, nagy árat fizettünk.

A táblánkban számos adat ismétlődik.

Bende Aladár személyes adatait kétszer, Árva Violáét háromszor is tárolnunk kellett.

Árva Viola neve, lakcíme, személyi igazolvány száma, már a tábla negyedik rekordjából megtudható. Az ötödik, és hatodik rekordból már – ebben a tekintetben- nem nyerünk új információt!

Az adatok fölösleges ismétlése, a **redundancia** az adatbázis készítők réme, számos probléma okozója lehet!

**Az ismétlődő adatot, amelynek értelmezésével már nem jutunk új információhoz redundáns adatnak nevezzük. Az adatok fölösleges ismétlését redundanciának hívjuk.**

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321<br>+36/70/788 9130<br>+36/20/315 5551 | Igen<br>Nem<br>Nem | 08:00-13:00<br>09:30-18:00<br>08:00-20:00 |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. | +36/36/321 321<br>+36/30/368 5552 | Igen<br>Nem | 08:00-13:00<br>08:00-20:00 |
| ... | ... | ... | ... | ... | ... | ... | ... |

*15. ábra Többértékű mezők*

Megoldás lehetne, hogy az egyes rekordokban több, másokban kevesebb telefon adatokat tároló mezőt hoznánk létre. Természetesen ez a tábla sem használható a relációs adatmodellben, ugyanis itt is megszegnénk a relációs adatmodell szabályait. Az egyes rekordokban eltérne a mezők száma!

A korábban látott táblában tehát redundanciát találunk. Az ötödik, hatodik, és nyolcadik rekord első öt mezője redundáns adatokat tartalmaz. Az adatbázis-kezelésben a redundancia jelensége káros, hiszen **növeli az adatbázis méretét,** ráadásul további kezelési nehézségeket, úgynevezett **anomáliákat** okoz. Az anomáliák lehetetlenné, vagy nehézzé tesznek valamilyen egyszerű rekordkezelő műveletet. Új rekord felvételét nehezíti meg a bővítési anomália. A rekordok törlését a törlési anomália, a rekordokban tárolt mezőértékek átírását a módosítási anomália nehezíti meg.

## 3.5 ANOMÁLIÁK

### 3.5.1 Bővítési anomália

A 3.3 szakasz táblájában egyik mező sem lehet önállóan kulcs mező, hiszen mindegyikben találunk ismétlődést. A táblában összetett kulcs van, amelyet *SzemIgSzam* és a *Szam* mezők alkotnak. Ebből következik, hogy ha új rekordot veszünk fel a táblába, akkor az összetett kulcs egyik eleme sem maradhat üresen. Az új rekordban kell lennie személyi igazolvány számnak, és telefonszámnak egyaránt. A táblába nem vehetünk fel olyan személyt, akinek nincsen telefonja, hiszen ebben az esetben hiányozna az összetett kulcs egyik eleme. Ezt a problémát **bővítési anomáliának** nevezzük.

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

*18. ábra Összetett kulcs a Személyek táblában*

### 3.5.2 Törlési anomália

Hasonlóan problémába ütköznénk, ha törölni kellene az adatbázisból a *113297 UL* személyi igazolvány számú *Tóth Áront*, de az általa használt telefonszámot meg szeretnénk őrizni. A személy és a telefonjai tulajdonságai egy rekordot alkotnak, így csak együtt törölhetők. A jelenséget **törlési anomáliának** nevezzük.

### 3.5.3 Változtatási anomália

Ha *Árva Viola* a *Nagy térről* a *Kis térre* költözik, három rekordban is át kell írnunk a címét. Ez fölöslegesen sok munka, azon túl, hibalehetőséget is rejt. Ha a három rekord közül akár egyben is elfelejtjük megváltoztatni a címet, az adatbázisunk hibás adatokat fog tartalmazni. Az adatbázis **inkonzisztens** lesz, azaz ellentmondásokat tartalmaz, nem tükrözi pontosan a valóságot. A redundáns tábla rekordjainak változtatásakor fellépő kezelési nehézségeket nevezzük **változtatási anomáliának**.

> A rosszul megalkotott relációs adatbázisok táblái redundanciát tartalmazhatnak, ami növeli az adatbázis méretét, bővítési, változtatási, és törlési anomáliát okozhat.

Az adatbázisok kialakításakor arra kell törekedni, hogy a redundanciát minimálisra szorítsuk.

## 3.6 NORMALIZÁLÁS

Az adatbázisok kialakításakor egyik legfőbb feladatunk, a redundancia-mentes adatszerkezetet kialakítása. Ennek egyik módja a normalizálás.

A normalizálás során egy kezdeti állapotból több fázison keresztül átalakítjuk az adatbázist. Az átalakítás fázisait normálformáknak nevezzük. Megkülönböztetjük a nulladik, az első, második, harmadik, negyedik, ötödik, és hatodik normálformát. A normálformák jelölésére az **0NF, 1NF, 2NF, 3NF**…jelöléseket használjuk. Az adatbázis kialakítása mindig az alacsonyabbtól a magasabb normálformák felé halad. 0NF->1NF->2NF->3NF…

Minden normálforma kialakításának megvannak a maga szabályai, kialakításának előfeltételei, és a kialakításhoz szükséges műveletei. Ha egy tábla kielégíti az előfeltételeket, akkor elvégezhetjük vele a szükséges művelteket. Eredményként olyan táblát kapunk, amely teljesíti a normálforma szabályait.

A relációs adatmodell szerint elkészült táblákat legalább harmadik normálformába kell alakítani. Ez általában elegendő ahhoz, hogy hibamentesen kezelhető adatbázisokat kapjunk. Tananyagunkban az 1NF, 2NF, és 3NF kialakítását mutatjuk be.

### 3.6.1 Mezők viszonya a relációs adatbázis tábláiban

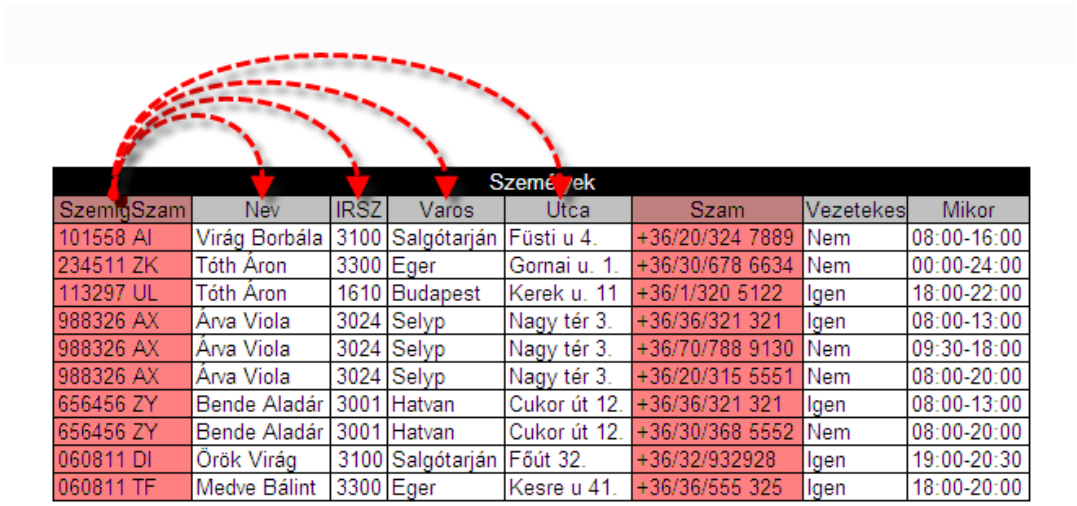
A normalizálás megértéséhez szükségünk van néhány további fogalom ismeretére, ezért most kisebb kitérőt teszünk. Megismerjük

- a funkcionális függés,
- a teljes funkcionális függés,
- a részleges funkcionális függés
- és a tranzitív függés jelentését.

#### *Funkcionális függés*

Funkcionális függésről akkor beszélünk, ha egy tábla valamelyik mezőjében lévő érték meghatározza egy másik mező értékét.

A *Személyek* táblában egyes „személy egyedek” többször is előfordulnak. Ha egy nevet megemlítünk nem biztos, hogy pontosan ki tudjuk választani a hozzá tartozó személyi igazolvány számot. Ha azonban egy személyi igazolvány számot vizsgálunk meg, biztosan meg tudjuk mondani a hozzá tartozó nevet. Ezért azt mondjuk, hogy a *Nev* mező funkcionálisan függ a *SzemIgSzam* mezőtől. A *SzemIgSzam* mező azonban nem függ a név mezőtől.



Személyek

| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
|---|---|---|---|---|---|---|---|
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

*19. ábra Funkcionális függések*

**A funkcionális függés két mező viszonya. „B” mező funkcionálisan függ „A” mezőtől, ha az „A” mező értéke egyértelműen meghatározza „B” mező értékét.**

### *Teljes funkcionális függés*

A funkcionális függés kiterjesztése a **teljes funkcionális függés**. Amikor egy adatbázist „normalizálunk”, arra törekszünk, hogy minden táblában teljes funkcionális függések legyenek.

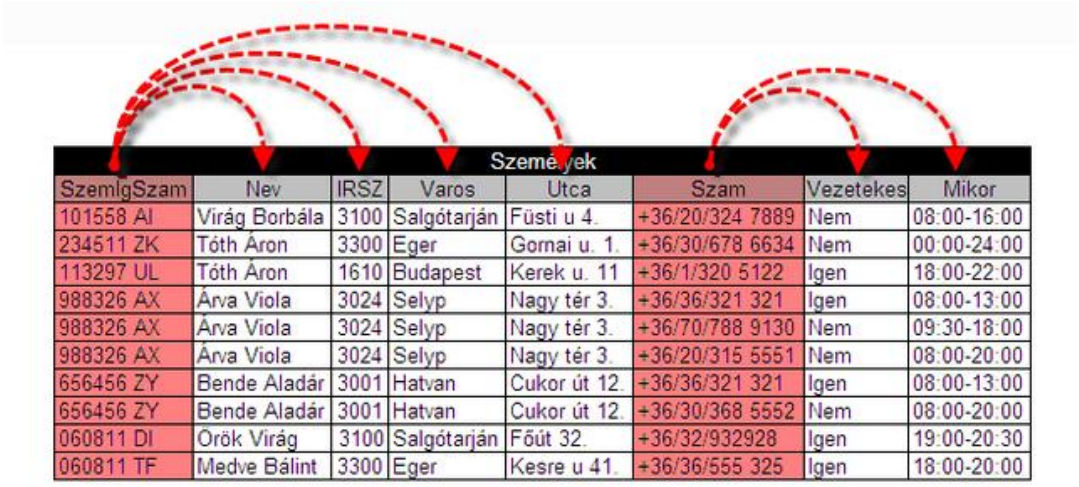
> A teljes funkcionális függésnek három feltétele van:
>
> egy tábla minden nem kulcs mezője függjön a kulcstól,
>
> minden, nem kulcs mező csak a kulcstól függjön,
>
> összetett kulcs esetén, minden nem kulcs mező függjön a kulcs minden elemétől!

### *Részleges funkcionális függés*

Részleges funkcionális függés a teljes funkcionális függés egyik akadálya. Akkor fordulhat elő egy táblában, ha abban van összetett kulcs és nem teljesül a teljes funkcionális függés alábbi feltétele.

„....összetett kulcs esetén minden nem kulcs mező függjön a kulcs minden elemétől!”

Az alábbi tábla részleges funkcionális függést tartalmaz.



Személyek

| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
|---|---|---|---|---|---|---|---|
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

*20. ábra Részleges funkcionális függés*

A *Nev*, az *IRSZ*, a *Varos*, és az *Utca* mezők a *SzemIgSzam* mezőtől függnek funkcionálisan.

A *Vezetekes* és a *Mikor* mezők nem a *SzemIgSzam*, hanem a *Szam* mezőtől függnek.

Egy személyi igazolványszám ismeretében pontosan meg tudjuk mondani, hogy hívják az illetőt, és hol lakik. Nem tudunk azonban biztos telefonszámot mondani, hiszem egy személyi igazolványszámhoz több telefonszám is tartozik.

Egy telefonszám ismeretében egyértelműen megmondható, hogy az vezetékes telefon-e és hogy mikor hívható. Az azonban még nem biztos, hogy egyértelmű nevet tudunk mondani, hiszen vannak személyek (4., 7. rekord, *Árva Viola, Bende Aladár*) akik ugyanazon a számon is elérhetők.

Részleges funkcionális függés csak akkor fordulhat elő egy táblában, ha abban összetett kulcs van.

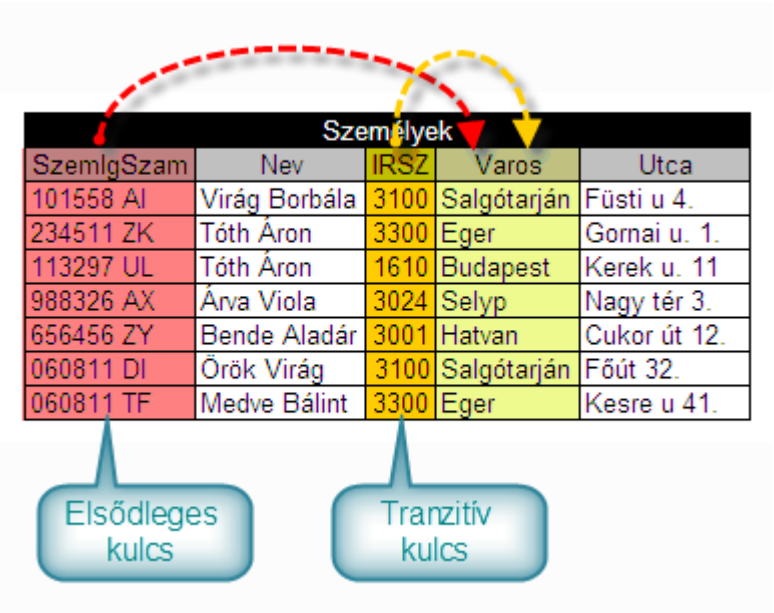
A normalizálás során a részleges funkcionális függést meg kell szüntetni!

### *Tranzitív függés*

Tranzitív függés esetén minden, nem kulcs mező függ a kulcstól, de van olyan mező, esetleg mezők, amely a kulcson kívül más mezőtől is függnek. A teljes funkcionális függés egyik feltétele, hogy „minden nem kulcs mező csak a kulcstól függjön”. Amennyiben ez a feltétel nem teljesül, **tranzitív függésről** beszélünk. Azt a mezőt, amelytől más mezők tranzitíven függnek, **tranzitív kulcsnak** hívjuk.

A normalizálás során a tranzitív függést meg kell szüntetni!

Az alábbi példában a *SzemIgSzam* mező a kulcs. A kulcstól függ minden mező, azonban a *Varos* a *SzemIgSzam* mezőn kívül az *IRSZ* mezőtől is függ. A *Varos* mező tehát tranzitív függésben van, az *IRSZ* mező a tranzitív kulcs.




*21. ábra Tranzitív függés*

### 3.6.2 Normálformák

Tudjuk már, mi a részleges funkcionális függés, mi a tranzitív függés. Most lássuk, hogyan szabadulhatunk meg tőlük, és hogyan érhetjük el, hogy adatbázisunkban csak **teljes funkcionális függések** legyenek!

#### *Első normálforma (1NF)*

Első normálformás adatbázist úgy alakítunk ki, hogy az egyedeinket egyedtípusokba, tulajdonságaikat tulajdonságtípusokba rendezzük, tárolásukra pedig olyan táblákat készítünk, amelyekre igazak a relációs adatmodell feltételei.

- Minden rekord különbözik
- Rekordonként megegyezik a mezők száma, és sorrendje
- Nincsenek többértékű mezők.

Azok táblák, amelyek nem elégítik ki a relációs adatmodell valamelyik feltételét, nulladik normálformában vannak (0NF)

Az alábbi a tábla azért nem megfelelő, mert a ***Szam***, a ***Vezetekes***, és a ***Mikor*** mezők többértékűek. A relációs adatmodellben nem lehet többértékű mező.

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| | | | | | +36/70/788 9130 | Nem | 09:30-18:00 |
| | | | | | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| | | | | | +36/30/368 5552 | Nem | 08:00-20:00 |

*22. ábra 0NF tábla*

A jó megoldást a következő változat nyújtja. Ebben a táblában a több telefonnal rendelkező személyeket többször is eltároltuk, annak megfelelően, hogy hány telefonnal rendelkeznek. Így elértük, hogy ne legyenek többértékű mezők.

| Személyek | | | | | | | |
|---|---|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

*23. ábra 1NF*

### *Második normálforma (2NF)*

A 2NF előfeltétele, hogy adatbázisunk minden táblája legalább 1NF-ben legyen! 2NF-ben vagyunk akkor, ha 1NF-ben vagyunk, és a táblákban megszűntetjük az esetleges részleges funkcionális függéseket.

A 2NF kialakításakor azt a táblát, amiben részleges funkcionális függés van, két új táblára bontjuk. Az egyik táblába az összetett kulcs egyik eleme kerül, a tőle függő összes mezővel együtt. A másik táblába a kulcs másik eleme kerül a tőle függő összes mezővel együtt. A két kapott táblában már nem lesz összetett kulcs, tehát nem lesz részleges funkcionális függés sem. Az új táblák azonosítói az eredeti összetett kulcs elemei lesznek.

### Kapcsolat az új táblák között

Az eredeti tábla „kettévágása” valójában azt eredményezte, hogy különválasztottunk két egyedtípust (*Személyek-Telefonok*) amelyek eddig egy táblában voltak. A művelet hatására csökken a redundancia. Ez mindenképpen hasznos, de észre kell vennünk egy súlyos problémát. A két új tábla között még nincs kapcsolat! Nem tudjuk, melyik személy melyik számon hívható, illetve, hogy egy bizonyos készüléken kik érhetők el.

Tudjuk, hogy a relációs adatmodellben idegen kulcsokkal ábrázoljuk a kapcsolatokat. Ha a Személyek táblában helyezzük el az Telefonok táblából származó idegen kulcsot, jelezve, hogy melyik személyt melyik számon lehet elérni, akkor az idegen kulcs többértékű lesz, hiszen egy személynek több telefonja lehet.

| Személyek | | | | | |
|---|---|---|---|---|---|
| SzemIgSzam | Nev | IRSZ | Varos | Utca | Telefon |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u. 4. | +36/20/324 7889 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gorkij u. 1. | +36/30/678 6634 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321<br>+36/70/788 9130<br>+36/20/315 5551 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321<br>+36/30/368 5552 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 |

ROSSZ MEGOLDÁS

*24. ábra Az idegen kulcs többértékű*

Ha a telefonok táblában helyezzük el az idegen kulcsot, megmutatva, hogy kik hívhatók az adott készüléken, a mező ismét többértékű lesz, mert egy számon több személy is elérhető.

A probléma megoldhatatlannak látszik. Ha jól megvizsgáljuk a táblákat, észrevehetjük, hogy közöttük N:M kapcsolat van, hiszen egy embernek több telefonja is lehet, de lehet olyan telefon, amin keresztül több személy is elérhető (pl. munkahelyi telefon: +36/36/321 321).

A kapcsolattípusokról tanulva említettük, hogy a relációs adatmodellben két tábla között közvetlenül nem lehet több-több (N:M) kapcsolat. Ennek éppen az az oka, hogy ilyenkor bárhová is tennénk az idegen kulcsot, az többértékű mező lenne. A probléma úgy oldható meg, hogy a keletkezett két tábla között még egy harmadik, úgynevezett kapcsolótáblát is létrehozunk, amiben mindkét tábla azonosítóját elhelyezzük idegen kulcsként. Így a két tábla nem közvetlenül, hanem egy kapcsolótáblán keresztül kapcsolódik egymáshoz.

A relációs adatmodellben „A” és „B” egymással több-több kapcsolatban lévő egyedtípusok táblái nincsenek közvetlen kapcsolatban. A kapcsolat egy harmadik („A_B”) tábla, a kapcsoló tábla közvetítésével valósul meg. A kapcsoló táblában lévő mezők

egyike, „A_Az” az „A” tábla rekordjait, még a másik mező, „B_Az” a „B” tábla rekordjait azonosítja.

A kapcsolótábla rekordjai elárulják, hogy az „A” tábla rekordjai mely rekordokhoz kapcsolódnak a „B” táblában és fordítva.

Megfigyelhetjük, hogy az „A” tábla 1:N kapcsolattal kapcsolódik az „A_B” kapcsoló táblához, és a „B” tábla is 1:N kapcsolattal kapcsolódik a kapcsoló táblához. A kapcsolótábla két 1:N kapcsolattá alakítja az N:M kapcsolatot.

### *Harmadik normálforma (3NF)*

A 3NF előfeltétele, hogy adatbázisunk minden táblája legalább 2NF-ben legyen! 3NF-ben vagyunk akkor, ha 2NF-ben vagyunk, és a táblákban megszűntetjük a tranzitív függéseket.

Minden tranzitív függést tartalmazó táblából két táblát csinálunk. Új táblába kerülnek a tranzitív függésben lévő mezők, azzal a tranzitív kulcs mezővel együtt, amelytől a kulcson kívül függnek. Az új táblában a tranzitív kulcs mező lesz az azonosító.

A 2NF kialakításakor létrejött Személyek tábla *Varos* mezője tranzitív függésben van. A *SzemIgSzam* kulcson kívül az *IRSZ* mezőtől is függ. A tranzitív függést úgy szüntetjük meg, hogy az *IRSZ* mezőt (tranzitív kulcs), és a *Varos* mezőt is új táblába, a *Varosok* táblába tesszük.

#### Kapcsolat a táblák között

A 3NF kialakítása közben ismét új táblákat hoztunk létre, amelyek között idegen kulcscsal kell biztosítanunk a kapcsolatot. Megfigyelhetjük, hogy a keletkezett *Városok* és *Személyek* táblák között egy-több, (1:N) kapcsolat van, hiszen egy városban több személy állandó lakhelye van, egy ember állandó lakhelye azonban csak egy városban lehet. Ebben a kapcsolatban a *Városok* tábla oldalát 1, a *Személyek* tábla oldalát több oldalnak nevezzük.

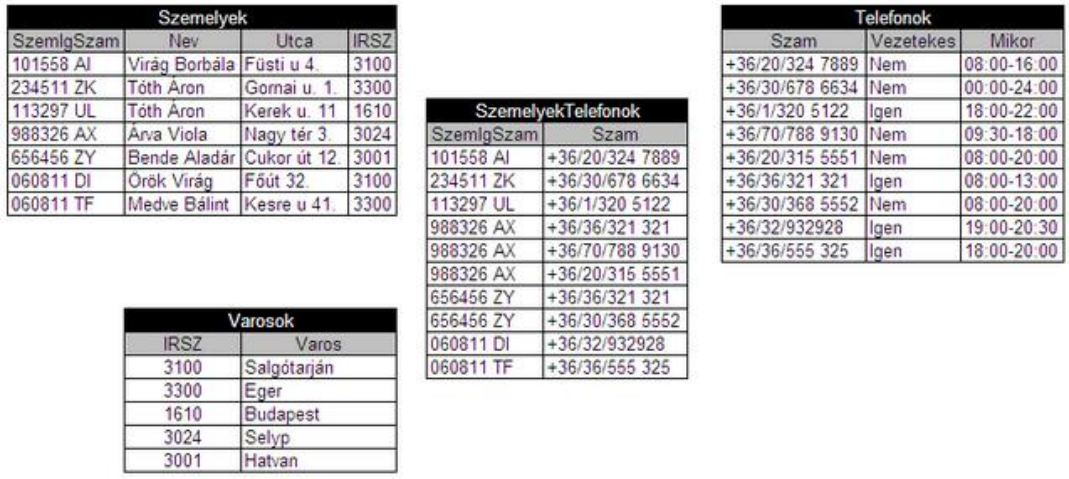
Ha az idegen kulcsot az 1 oldalra, a *Városok* táblába tennénk (jelezve, hogy kik laknak az adott városban), akkor az, többértékű mező lenne, hiszen egy városban többen is laknak. Ha azonban a több oldalra, a *Személyek* táblába tennénk az idegen kulcsot (megmutatva, hogy hol van az adott személy állandó lakhelye), akkor az nem lenne többértékű. Ebből a tapasztalatból kiindulva a következő szabályt alkothatjuk:

1:N kapcsolat esetén az idegen kulcsot mindig a több oldalon lévő táblában helyezzük el.

1:N kapcsolat esetén tehát a több oldalon, N:M kapcsolat esetén pedig kapcsoló táblában helyezzük el az idegen kulcsot. Ha két tábla között 1:1 kapcsolat van, akkor az idegen kulcs bármelyikbe kerülhet.

Az új tábla létrehozásának köszönhetően ismét csökkent a redundancia. Azoknak a városoknak a neveit, amelyekben több személy is lakik, most már csak egyszer kell tárolni.

Elértük, hogy adatbázisunk harmadik normálformába kerüljön.



| Szemelyek | | | |
|---|---|---|---|
| SzemIgSzam | Nev | Utca | IRSZ |
| 101558 AI | Virág Borbála | Füsti u 4. | 3100 |
| 234511 ZK | Tóth Áron | Gornai u. 1. | 3300 |
| 113297 UL | Tóth Áron | Kerek u. 11 | 1610 |
| 988326 AX | Árva Viola | Nagy tér 3. | 3024 |
| 656456 ZY | Bende Aladár | Cukor út 12. | 3001 |
| 060811 DI | Örök Virág | Főút 32. | 3100 |
| 060811 TF | Medve Bálint | Kesre u 41. | 3300 |

| SzemelyekTelefonok | |
|---|---|
| SzemIgSzam | Szam |
| 101558 AI | +36/20/324 7889 |
| 234511 ZK | +36/30/678 6634 |
| 113297 UL | +36/1/320 5122 |
| 988326 AX | +36/36/321 321 |
| 988326 AX | +36/70/788 9130 |
| 988326 AX | +36/20/315 5551 |
| 656456 ZY | +36/36/321 321 |
| 656456 ZY | +36/30/368 5552 |
| 060811 DI | +36/32/932928 |
| 060811 TF | +36/36/555 325 |

| Telefonok | | |
|---|---|---|
| Szam | Vezetekes | Mikor |
| +36/20/324 7889 | Nem | 08:00-16:00 |
| +36/30/678 6634 | Nem | 00:00-24:00 |
| +36/1/320 5122 | Igen | 18:00-22:00 |
| +36/70/788 9130 | Nem | 09:30-18:00 |
| +36/20/315 5551 | Nem | 08:00-20:00 |
| +36/36/321 321 | Igen | 08:00-13:00 |
| +36/30/368 5552 | Nem | 08:00-20:00 |
| +36/32/932928 | Igen | 19:00-20:30 |
| +36/36/555 325 | Igen | 18:00-20:00 |

| Varosok | |
|---|---|
| IRSZ | Varos |
| 3100 | Salgótarján |
| 3300 | Eger |
| 1610 | Budapest |
| 3024 | Selyp |
| 3001 | Hatvan |

*25. ábra 3NF*

## 3.7 ADATBÁZIS TERVÉNEK ÁBRÁZOLÁSA

Amikor egy adatbázist megtervezünk, általában nem állnak rendelkezésünkre valódi rekordok. A normalizálással kialakított adatbázis szerkezetet úgy ábrázoljuk, hogy elnevezzük mind a táblákat, mind pedig a táblák mezőit. A táblaneveket aláhúzzuk, a tábla mezőinek nevét pedig a táblanév alatt tűntetjük fel.

A mezők között megjelöljük az elsődleges kulcsokat. Ehhez használhatunk színjelölést, vagy rajzolhatunk az azonosító mezők nevei elé egy kulcsocskát.

Szintén megjelöljük az idegen kulcsokat. A jelöléshez használhatunk színt vagy aláhúzást.

Az egymással kapcsolatban lévő táblák a kapcsolatát a kapcsolódó mezőket összekötő vonallal jelezzük. A vonal két végén feltűntetjük a kapcsolat típusát.

## 3.8 ÖSSZEFOGLALÁS

A rosszul megtervezett reláció adatbázis tábláiban fölösleges ismétlődés, redundancia alakulhat ki. A redundáns adatok értelmezésével már nem juthatunk új információhoz!

A redundancia mentes adatbázis kialakításának egyik technikája a normalizálás.

A normalizálás során egy kiinduló állapotból több fázison keresztül átalakítjuk az adatbázis tábláit. Az átalakítás fázisait normálformáknak nevezzük.

A relációs adatbázis-kezelés egyik szabálya, hogy az adatbázisokat legalább harmadik (3NF) normálformára kell alakítani.

Első lépésként 1NF-be kell alakítani az adatbázist. Egy relációs adatbázis akkor van 1NF-ban, ha minden táblája kielégíti a relációs adatmodell követelményeit.

Ha az 1NF adatbázis minden táblájában megszűntetjük a részleges funkcionális függéseket, akkor 2NF adatbázist hoztunk létre.

Egy táblában úgy szüntetjük meg a részleges funkcionális függést, hogy a táblát két új táblára bontjuk. Az eredeti tábla összetett kulcsának elemeit a tőlük függő mezőkkel együtt

az új táblákba tesszük. A részleges funkcionális függés megszűntetésekor keletkező új táblák között több-több kapcsolat van.

A relációs adatmodellben az N:M kapcsolat nem ábrázolható közvetlenül. A problémát úgy oldjuk meg, hogy az N:M kapcsolatban lévő egyedtípusok táblái közé egy újabb, úgynevezett kapcsoló táblát iktatunk be. A kapcsoló tábla az összekapcsolt táblák azonosítóiból származó idegen kulcsokat tartalmaz. Ezek segítségével 1:N kapcsolatokkal kötődik mindkét táblához. Az N:M kapcsolatot két 1:N kapcsolattá alakítja.

A 2NF adatbázist harmadik normálformára alakíthatjuk, ha minden táblájában megszűntetjük a tranzitív függéseket. A tranzitív függéseket tartalmazó táblából két táblát alakítunk ki. Az eredeti tábláról leválasztjuk azt mezőt, amelyiktől más mezők tranzitíven függnek. Ez a mező lesz az új tábla kulcsa. Az új táblába tesszük a tranzitív függésben lévő mezőket is, a többi mező a régi táblában marad.

A régi tábla és az új tábla közötti kapcsolatot úgy valósítjuk meg, hogy az új tábla kulcsából idegen kulcsot hozunk létre a régi táblában.

## 3.9 ÖNELLENŐRZŐ KÉRDÉSEK

1. Miért kell megszűntetni a az adatbázisokban a redundanciát?
2. Igaz-e, hogy a normalizálás során meg kell szűntetni a funkcionális függéseket?
3. Hányadik normálformában van egy adatbázis, ha minden táblája teljesíti a relációs adatmodell feltételeit, és minden táblában egyszerű kulcs van?
4. Hogyan szűntetjük meg a részleges funkcionális függést?
5. Hogyan szűntetjük meg a tranzitív függéseket?
6. Hogyan tároljuk az egyes táblák közötti kapcsolatokat?

# 4. A MICROSOFT ACCESS ADATBÁZISOK SZERKEZETE, ÉS KEZELÉSE

## 4.1 CÉLKITŰZÉS

Miután szert tettünk némi jártasságra az adatszerkezetek kialakítása terén, kezdjük el a tanultak gyakorlati alkalmazását! Ebben a leckében megismeri a relációs adatbázis-kezelő alkalmazások legfontosabb funkcióit, a Microsoft Access kezelőfelületét és az Access adatbázisok szerkezetét. Megtanulja, hogyan lehet megnyitni az adatbázisokat, hogyan tallózhat az adatbázis objektumai között. Megismeri az egyes objektumok kezelésének alapvető fogásait, és megtanulja, hogyan készíthet biztonsági másolatot az adatbázisokról.

## 4.2 A LECKE TÉMAKÖREI

- Relációs adatbázis-kezelő rendszerek funkciói
- Az MS Access kezelő felülete
- Access adatbázisok szerkezete
- Tallózás az Access adatbázis objektumai között
- Objektumok kezelése
- Access adatbázis biztonsági mentése

## 4.3 ADATBÁZIS-KEZELŐ RENDSZEREK

Az adatbázis készítés legfontosabb fázisa az adatbázis helyes megtervezése. Miután szert tettünk némi jártasságra az adatszerkezetek kialakítása terén, hozzákezdhetünk az adatbázis gyakorlati kialakításához, majd az adatbázisban tárolt adatok kezeléséhez. Míg a tervezés nem igényel valódi számítógépes környezetet, az adatbázis elkészítése és kezelése csak megfelelő szoftverek segítségével végezhető el.

**Az adatbázisok kialakítására és kezelésére használható alkalmazásokat adatbázis kezelő rendszereknek nevezzük.**

Egy adatbázis-kezelő rendszerrel tetszőleges adattartalom tárolható, az adatok tárolási és kezelési módja azonban mindig valamilyen adatmodellhez kötődik. Egy bizonyos adatbázis-kezelő rendszer csak egyféle adatmodellnek megfelelő adatbázissal képes dolgozni. Ennek megfelelően léteznek hálós adatbázis-kezelő rendszerek, hierarchikus adatbázis kezelő rendszerek, objektumorientált rendszerek, és **relációs adatbázis-kezelő rendszerek**. Utóbbiakkal az általunk megismert relációs adatmodellnek megfelelően a táblákban tárolt egyedtípusokkal végezhetjük el a különféle adatbázis-kezelő műveletek.

A szoftverpiacon számtalan adatbázis-kezelő rendszer vásárolható meg, ezek zöme azonban a legelterjedtebb, relációs adatmodellhez kötődik. A tananyagunkban, az irodai gyakorlatban talán legelterjedtebb – elsősorban helyi adatbázisok kezelésére alkalmas – relációs adatbázis-kezelő rendszer, a Microsoft Access használatát ismerheti meg.

Ez a tananyag a Microsoft Access 2007 verzió használatát tárgyalja, de itt elsajátítható ismeretek kevés kiegészítéssel más verziókban is használhatók.

## 4.4 RELÁCIÓS ADATBÁZIS-KEZELŐ RENDSZEREK FUNKCIÓI

Amikor először dolgozunk adatbázis-kezelő rendszerrel nem csak a kezelés módja ismeretlen számunkra, de azt sem tudjuk pontosan, milyen szolgáltatások állnak majd rendelkezésünkre. Az adatbázis-kezeléssel kapcsolatos feladatok két nagy csoportba sorolhatók

- az adatbázis tervezése és **kialakítása**, illetve
- az elkészített adatbázisban tárolt adatok **használata**

Ezeket a feladatokat nagyon gyakran nem ugyanazok a személyek végzik, így az adatbázisokkal foglalkozó felhasználók is két kört alkotnak. Azokat, akik elsősorban az adatbázisok kialakításával foglakoznak, **fejlesztőknek**, a kész adatbázisokkal dolgozókat pedig egyszerűen **felhasználóknak** nevezzük. A modern adatbázis-kezelő rendszerek mindkét szerepkör számára biztosítják a megfelelő funkciókat, és eszközöket:

- **Adatbázis létrehozása:** Bármilyen furcsa is, az adatbázis-kezelők lényeges tulajdonsága, hogy képesek üres adatbázis kialakítására. Ez azért fontos, mert az üres adatbázis később tetszőleges adatok tárolására tehető alkalmassá.
- **Táblák kialakítása:** Az üres adatbázisban előzetes terveink alapján létrehozhatjuk a táblákat, amelyekben az egyedtípusok adatai tárolódhatnak majd.
- **Táblák kezelése:** Az elkészített táblák bármelyikében tárolhatók rekordok. Ezek később törölhetők, megváltoztathatók, különböző szempontok szerint rendezhetők, kereshetők, kiválogathatók. A rekordkezelő műveleteket külön-külön végezhetjük az egyes táblákban.
- **Műveletek kapcsolódó rekordokkal:** A normalizálás során általában sok-sok táblára bontjuk a relációs adatbázist. Ez redundancia mentes tárolást tesz lehetővé, de megnehezíti a felhasználó munkáját, hiszen nem az látja egyszerre, a különböző táblák kapcsolódó rekordjait. Az adatbázis-kezelő rendszerek rendelkeznek olyan eszközökkel, amelyek segítségével könnyűszerrel kezelhetők a legbonyolultabb táblakapcsolatok is.
- **Kezelőfelület:** Míg az adatbázis megtervezése, és kialakítása fejlesztői, addig a rekordok kezelése, a tárolt adatok feldolgozása jellemzően felhasználói munka. Az adatbázis-kezelő rendszerek biztosíthatnak olyan eszközöket, amelyek segítségével a fejlesztő egyszerűen használható, praktikus kezelőfelületet alakíthat ki az adatbázishoz. A kezelőfelület segítségével a felhasználó úgy is tud dolgozni az adatbázissal, hogy nem ismeri annak szerkezetét.
- **Nyomtatás:** Munka közben az adatbázis kezelőfelülete, és a tárolt adatok képernyőn jelennek meg, de gyakran van szükségünk azonban arra, hogy az adatokat nyomtatott jelentések formában is láthassuk. Az adatbázis-kezelők gyakran biztosítanak valamilyen lehetőséget az adatbázisban tárolt adatok nyomtatására.
- **Hálózati hozzáférés:** A hálózatok világában valamilyen szinten minden adatbázis-kezelő rendszernek támogatnia kell a hálózati felhasználást. A hálózaton használt

adatbázis egy szerveren helyezkedik el, de az adatokat a hálózat távoli gépeiről is lehet kezelni.

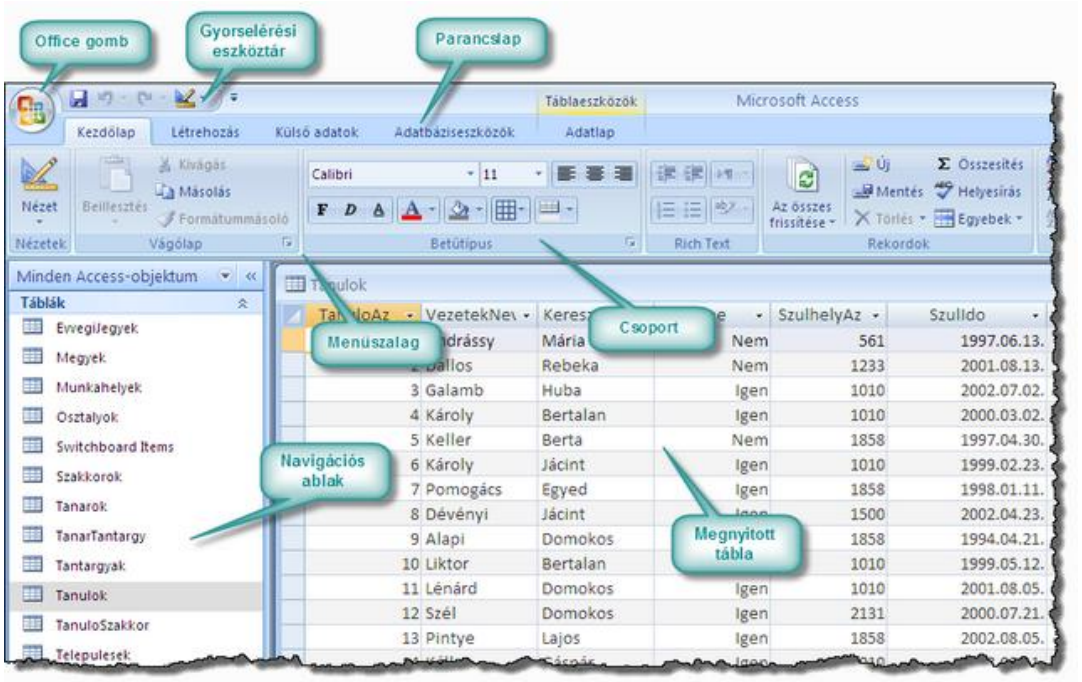
- **Programozhatóság:** Az adatbázis-kezelő rendszereket felkészítették arra, hogy az általános adatbázis-kezelő feladatokat el tudják végezni. Az ezektől eltérő, csak egy adott adatbázisra jellemző feladatokra, a fejlesztők egyedi programkódokat írhatnak, amelyeket a kérdéses adatbázisban tárolhatnak el. Az adatbázis specifikus programok megírása a fejlesztő feladata, futtatásukat azonban általában a felhasználóra végzi, sokszor anélkül, hogy erről tudomást szerezne. A beépített programok automatikusan, vagy felhasználói művelteknek köszönhetően futnak le.

## 4.5 AZ MS ACCESS KEZELŐFELÜLETE

Miután tudjuk, mit várhatunk egy relációs adatbázis-kezelő rendszertől, vizsgáljuk meg közelebbről a Microsoft Access-t. Elsőként a kezelőfelületet tekintjük át, majd az előző szakaszban felsorolt feladatok lebonyolítását biztosító eszközöket ismerjük meg.

### 4.5.1 Kezelőfelület

Az Office 2007 programcsomaggal a Microsoft szakított a korábbi verziókban megszokott, menüsorból, és eszköztárakból álló kezelőfelülettel. Az új, Microsoft Office Fluent felhasználói felülete vette át a menük, eszköztárak és a legtöbb munkaablak feladatát.




*26. ábra az MS Access 2007 felülete*

### *Menüszalag*

Az Office Access 2007-ben a korábban megszokott menük és eszköztárak legfontosabb helyettesítője a menüszalag, amely az Office Fluent felhasználói felület egyik alkotóeleme. A menüszalag adott funkciócsoportoknak megfelelő **lapokból** áll. A lapokon **csoportokba** rendezett **vezérlőelemekkel** – gombok, gyűjtemények, listák, lenyíló listák, párbeszédablak gombok – végezhetjük el az egyes feladatokat. A menüszalagon a lapok nevét tartalmazó **fülek** (Kezdőlap, Létrehozás, Külső adatok, Adatbáziseszközök, Bővítmények) segítségével választható ki a szükséges lap, majd pedig annak vezérlőelemei.

Munka közben a menüszalagon látható szokásos füleken kívül, **környezetfüggő eszközök** fülei is megjelenhetnek. Ezek csak adott feladatok végrehajtásakor válnak láthatóvá a felületen. A környezetfüggő eszközök az éppen kiválasztott feladat elvégzését segítik. Amikor rákattint egy objektumra, az arra vonatkozó környezetfüggő lapok csoportja jelölőszínnel megjelenik a szokásos lapok mellett.

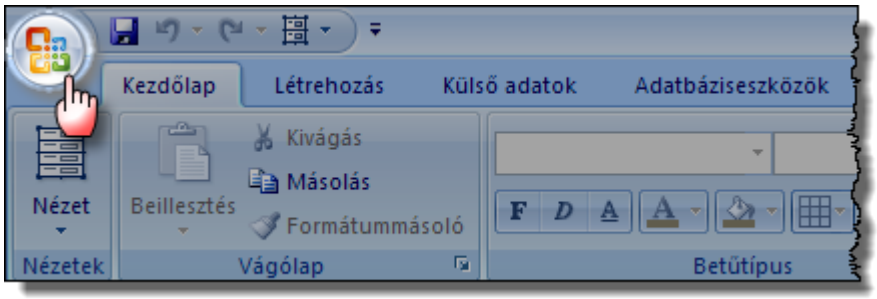
Az Access-ben egyes tevékenységek – például nyomtatási képek megtekintése – közben eltűnhetnek a szokásos lapok. Ilyenkor helyükön az úgynevezett **program lapok** jelennek meg, amelyek az adott feladathoz tartozó vezérlőelemeket tartalmazzák. Amikor a feladatot befejeztük, a program lapok eltűnnek, és ismét a szokásos menüszalag jelenik meg.



*27. ábra A menüszalag*

### *Microsoft Office gomb*

A menüszalag funkcióit egészíti ki a bal felső sarokban található Microsoft Office gomb, amely jórészt a korábbi verziók Fájl menüjének parancsait tartalmazza.

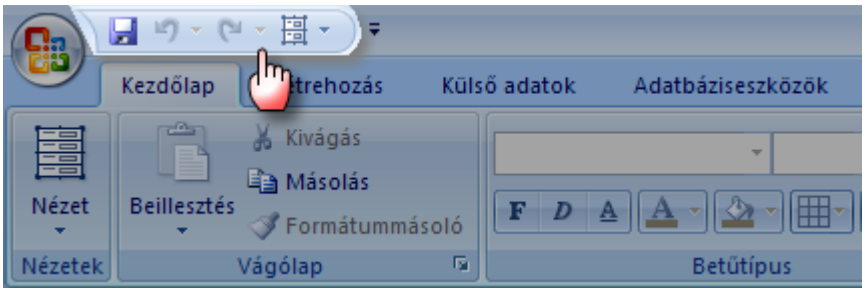



*1. kép Microsoft Office gomb*

### *Gyorselérési eszköztár*

Az Office gombtól jobbra megjelenő **gyorselérési eszköztár** leginkább a korábbi eszköztárakhoz hasonlít. Segítségével egyetlen kattintással érhetők el a leggyakoribb paran-

csok. Alaphelyzetben csak három gombot tartalmaz, de a felhasználó szabadon megváltoztathatja összetételét.



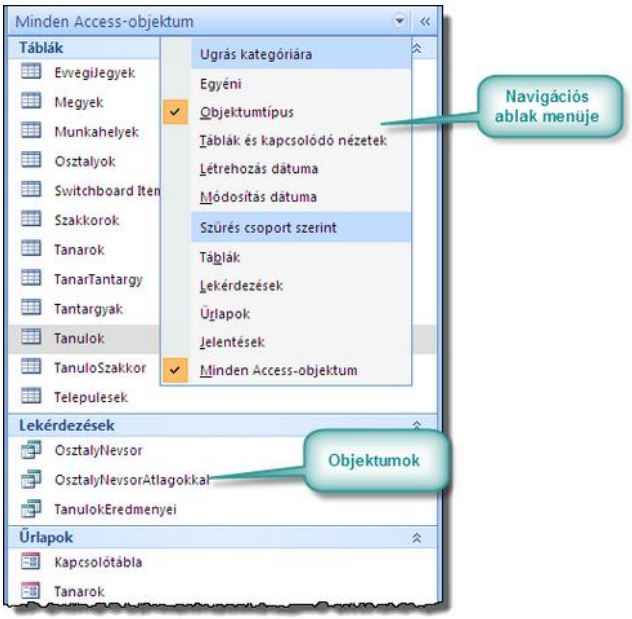

*2. kép Gyorselérési eszköztár*

### *Helyi menü*

A helyi menü a Windows alapú programok vezérlésére használt egyik legnépszerűbb eszköz. Használatához a jobb egér gombbal kell kattintanunk a kiválasztott elemen. A gyorsmenü azonnal láthatóvá válik a képernyőn, és megjeleníti az adott helyen használható parancsokat. Az Accessben szinte mindenhol használhatjuk. Számos feladat a gyorsmenü parancsaival végezhető el a legegyszerűbben.

### *Navigációs ablak*

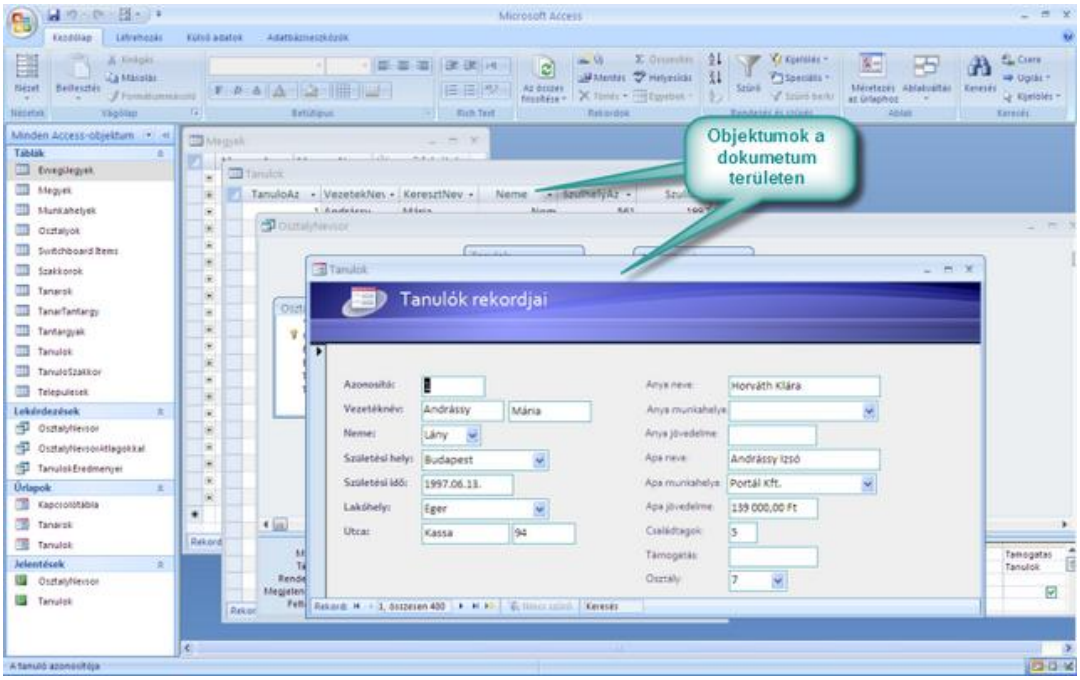
Egy adatbázis megnyitása után a képernyő bal oldalán megjelenik az úgynevezett Navigációs ablak, amelyben kiválaszthatjuk és kezelhetjük az adatbázis objektumait. (Lásd: 4.6.2 szakasz). A navigációs ablak tetején elhelyezkedő kis háromszöggel megjeleníthető az ablak kezelésére szolgáló menü.




*28. ábra A navigációs ablak*

### *Dokumentum terület*

Az Access felületének legnagyobb részét a dokumentum területe teszi ki. Itt jelennek majd meg azok az ablakok, amelyekben elvégezhetjük a különféle műveleteket.




*3. kép Dokumentum terület*

### 4.5.2 Az Access indítása

A program indításához meg kell keresnünk az alkalmazás ikonját, amelyet általában a **Start menü/Programok/Microsoft Office 2007/Microsoft Access 2007** helyen találunk. Az Access indításakor csak az Office gomb, és a Gyorsindítási eszköztár jelenik meg. A menüszalag akkor válik láthatóvá, amikor megnyitunk, vagy létrehozunk egy adatbázist.

## 4.6 ACCESS ADATBÁZISOK SZERKEZETE

### 4.6.1 Adatbázisok a fájlrendszerben

A különböző adatbázis-kezelők más-más módon tárolják az adatbázisokat a fájlrendszerben. Egyes rendszerek minden táblát, sőt az adatbázis egyéb komponenseit külön fájlokban helyeznek el. Az Access egy adatbázist, annak minden összetevőjével együtt, egyetlen ACCDB kiterjesztésű fájlban tárol.

### 4.6.2 Access adatbázisok objektumai

Bár egy Access adatbázis egyetlen fájlt alkot csupán, az állományon belül számos összetevő, úgynevezett objektum található. Ezek az objektumok biztosítják a 4.4 szakaszban leírt funkciókat.

Access adatbázisokat az alábbi objektumok alkothatják:

- **Táblák:** Minden tábla egy-egy egyedtípus egyedeinek tárolására, és kezelésre való. Egy táblába felvehetünk rekordokat, módosíthatjuk azok mezőértékeiket, töröl-

hetjük a fölösleges egyedeket. Lehetőségünk van a rekordok rendezésére, a rekordok közötti keresésére, a rekordok kiválogatására.

- **Lekérdezések:** Míg egy táblában csak a annak rekordjait kezelhetjük, a lekérdezések lehetővé teszik, hogy a több tábla kapcsolódó rekordjain, együttesen végezzünk el bizonyos műveleteket. Például: Ha egy adatbázisban van *Dolgozok* és *Lakohelyek* táblánk, akkor egy lekérdezés segítségével egyszerre jeleníthetők meg az egyes dolgozók nevei és lakóhelyei.
- **Űrlapok:** Az űrlapok segítségével alakítható ki az adatbázis kezelőfelülete. Űrlapokkal vezérelhetjük a különböző objektumok megnyitását, illetve a táblákban tárolt adatokat tetszetős, felhasználóbarát, jól áttekinthető felületen jeleníthetjük meg. Az űrlapokon elhelyezett vezérlőelemekkel (gombok, listák, szövegmezők…) egyszerűen kezelhetővé tehetjük adatbázisunkat.
- **Jelentések:** A jelentések olyan objektumok, amelyek segítségével nyomtatható formába alakíthatjuk az adatbázisban tárolt adatokat.
- **Oldalak:** Az űrlapokhoz hasonló objektum, amely lehetővé teszi egy lokális hálózaton lévő adatbázis, távoli gépről történő kezelését. Az oldalak weblapokként képesek megjeleníteni az adatbázisban tárolt rekordokat. Fontos tudnunk, hogy az Oldalak használatához az adatbázist olyan gépen kell elhelyeznünk, amelyen Microsoft webszerver, például IIS fut.
- **Makrók, modulok:** Ezek az objektumok teszik lehetővé, hogy az adatbázisban programkódokat írjunk és tároljunk el. A beépített kódokkal programozhatjuk a bonyolultabb, vagy gyakran végrehajtandó műveleteket. Az elkészült programokat űrlapjaink vezérlőelemeihez kapcsolva, a felhasználó interakcióval indíthatók el.

## 4.7 ACCESS ADATBÁZIS KEZELÉSE

### 4.7.1 Adatbázis megnyitása

Az Access adatbázisok megnyitása más Office alkalmazásokhoz hasonlóan az Office gomb **Megnyitás** parancsával elérhető párbeszédablakban lehetséges. Ha az adatbázis korábban már használtuk, akkor egyetlen kattintással megnyithatjuk az Office menüben, vagy a dokumentum terület jobb szélén lévő munka ablakban.

### 4.7.2 Tallózás az Access adatbázis objektumai között

Amikor megnyitunk egy adatbázist, megjelenik a navigációs ablak, és láthatóvá válnak a különböző objektumok. A navigációs ablak menüjével beállítható, hogy minden objektumot látni akarunk, vagy csak az egyes kategóriákat szeretnénk megjeleníteni. Az első esetben a **Minden Access-objektum** parancsra, egyébként pedig a kívánt objektumkategória nevére (például **Táblák**) kell kattintani.

### 4.7.3 Műveletek objektumokkal

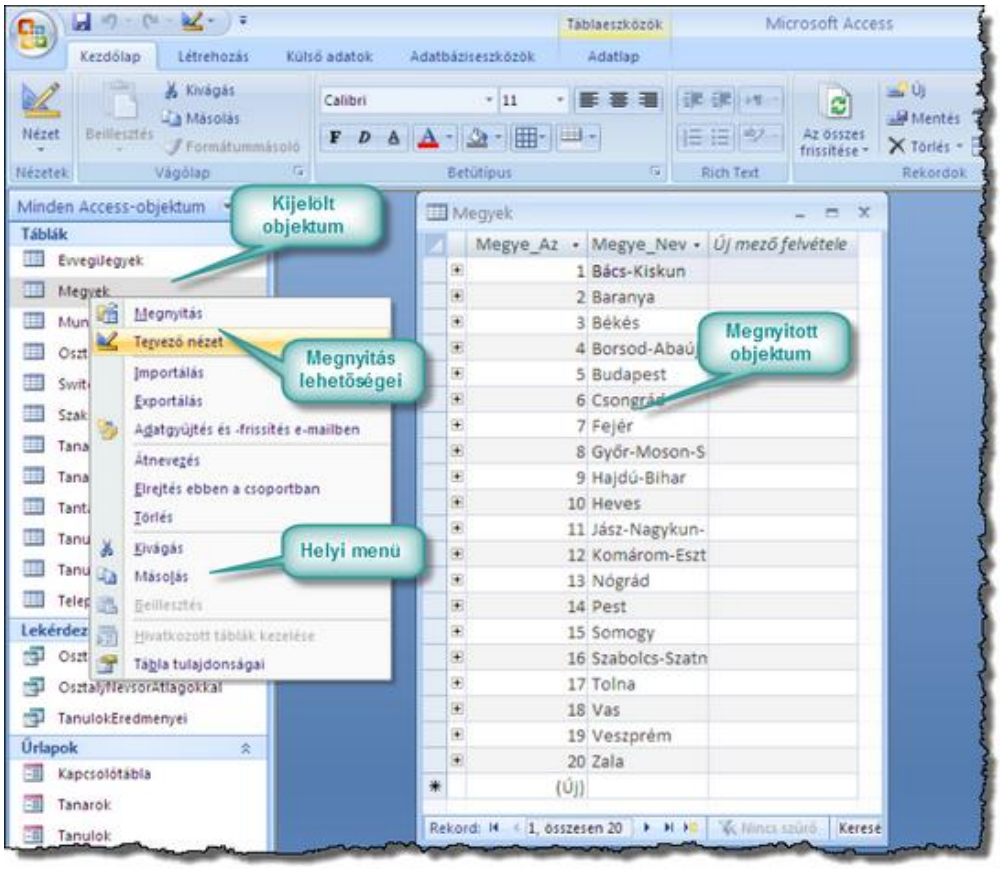
#### *Objektumok megnyitása különböző nézetekben*

Az Access adatbázisok objektumait különböző nézetekben tekinthetjük meg, aszerint, hogy éppen mit csinálunk velük. Más nézetben látunk például egy táblát, amikor a szerkezetét építjük föl, és más nézetben akkor, amikor már rekordokat rögzítünk benne.

Több másik nézet mellett, ez a kettő minden objektum esetében rendelkezésre áll.

Bár az Access más-más szavakat használ a nézetek elnevezésére a különböző objektumok esetén, a továbbiakban az egyszerűség kedvéért **tervező nézet** alatt a fejlesztéskor, **felhasználói nézet** alatt az alkalmazáskor használt nézetet értjük. Mindkét nézetnek kiemelt szerep jut, hiszen tervező nézetben tudjuk létrehozni, kialakítani az adott objektumot, felhasználói nézetben pedig valódi feladatára tudjuk majd használni.

Amikor meg akarunk nyitni egy objektumot (például egy táblát), egyszerűn keressük meg a navigációs ablakban, és a jobb gombbal rákattintva jelenítsük meg a helyi menüjét.




*29. ábra Megnyitás lehetőségei*

A helyi menü **Megnyitás** parancsa felhasználói nézetben, a **Tervező nézet** parancs természetesen tervező nézetben jeleníti meg az objektumot. A kiválasztott nézet ablaka a dokumentum területen jelenik meg.

Megnyitás után a menüszalag **Kezdőlap** parancslapján, a **Nézet** gomb közepére kattintva egyszerűen válthatunk a két nézet között. Ha befejeztük az objektummal végzett munkát, be kell zárnunk az adott nézetet, az objektumablak jobb felső sarkán lévő bezárás gombbal.

Az adatbázis objektumait nem csak megnyithatjuk. A helyi menü segítségével könnyedén átnevezhetjük, törölhetjük őket, vagy másolatot készíthetünk róluk.

A törléssel, átnevezéssel érdemes óvatosnak lenni, ugyanis a töröl objektum nem állítható vissza, az átnevezés pedig hibákat okozhat más objektumok működésében.

A másolást a helyi menü **Másolás** parancsával végezhetjük el.

A lemásolt objektum a vágólapra kerül, használható másolat csak akkor keletkezik, ha a lemásolt objektumot (ismét a helyi menüt használva) be is illesztjük.

### 4.7.4 Adatbázis bezárása

Az adatbázis bezárása, a megnyitáshoz hasonlóan az Office menüben végezhető el. A Bezárás parancsa az adatbázis bezáródik, de az Access tovább fut. A Kilépés az Access programból parancs a teljes programból kilép, de előbb természetesen bezárja az éppen használt adatbázis is.

### 4.7.5 Adatbázis biztonsági mentése

Az Access adatbázisok biztonsági mentése egyszerűen megoldható. Mivel egy adatbázis egyetlen fájlt alkot, le kell másolnunk az állományt, és a másolatot egy biztonságos meghajtóra kell áthelyeznünk. Fontos, hogy a biztonsági másolatot mindig bezárt adatbázisról készítsük el.

## 4.8 ÖSSZEFOGLALÁS

A Microsoft Access relációs adatbázis-kezelő rendszer, amely képes elvégezni a relációs adatbázis-kezelésben felmerülő feladatokat. Egy adatbázis minden összetevőjét egyetlen ACCDB kiterjesztésű fájlban tárolja, de az adatbázisokban táblákat, lekérdezéseket, űrlapokat, oldalakat, makrókat és modulokat képes tárolni. Ezek az objektumok teszik lehetővé a különböző feladatok elvégzését.

Az egyedtípusok tárolása és kezelése a táblákban valósul meg.

Az egyes táblák kapcsolódó rekordjait a lekérdezésekkel kezelhetjük.

Az űrlapok az adatbázis kezelőfelületét biztosítják.

A jelentésekkel nyomtatásban jeleníthetők meg az adatbázis rekordjai.

Az oldalak webes kezelési lehetőségeket biztosítanak.

A makrókkal és modulokkal adatbázis specifikus programok fejleszthetők.

Az Access 2007 vezérlése az Office 2007-ben bevezetett Microsoft Office Fluent felhasználói felületen történik. Ennek legfontosabb elemei a parancslapokból felépülő menüszalag, az Office Menü gomb, és a Gyorsindítási eszköztár.

Megnyitott adatbázis esetén megjelenik a navigációs ablak, amelyben kiválasztható a szükséges objektum. A kiválasztott objektum, tervező, illetve felhasználó nézetben nyitható meg. Előbbit az objektum létrehozásakor, illetve szerkezetének megváltoztatásakor, utóbbi az elkészült objektum használatakor választjuk.

Az objektumok megnyitása, átnevezése, törlése, és lemásolása legegyszerűbben a helyi menü parancsaival végezhető el.

A bezárt Access adatbázisról az ACCDB kiterjesztésű fájl duplikálásával készíthetünk biztonsági másolatot.

## 4.9 ÖNELLENŐRZŐ KÉRDÉSEK

1. Milyen funkciókra számíthat egy relációs adatbázis-kezelő rendszer használatakor?
2. Hogyan kell kezelni a menüszalagot?
3. Mik azok a környezetfüggő eszközök?
4. Hogyan tallózhat az adatbázis objektumai között?
5. Milyen formában használhatók az egyes objektumok?
6. Hogyan készíthet biztonsági másolatot az Access adatbázisokról?

# 5. MICROSOFT ACCESS ADATBÁZISOK KIALAKÍTÁSA

## 5.1 CÉLKITŰZÉS

Ebben a leckében megismerkedhet a tényleges adatbázis kialakítás lépéseinek sorrendjével, pontosan megismerheti az első és egyben legfontosabb mozzanatot, a táblakészítést. Megtanulhatja, hogyan hozhat létre új táblákat, hogyan alakíthatja ki azok mezőit, hogyan jelölheti az elsődleges kulcsot. Számos mezőtulajdonsággal ismerkedik majd meg, amelyek segítségével biztosíthatja az adatbázis konzisztenciáját, egyszerűbbé és gyorsabbá teheti az adatbázis későbbi kezelését.

## 5.2 A LECKE TÉMAKÖREI

- Új adatbázis létrehozása
- Táblák kialakítása
- Névadási szabályok
- Adattípusok
- Mezőtulajdonságok

## 5.3 ADATBÁZIS LÉTREHOZÁSA AZ ACCESSBEN

A tervezés alapvetően meghatározza egy adatbázis minőségét, használhatóságát, későbbi továbbfejleszthetőségét. Ezért bármilyen rendszerrel dolgozzunk is, az adatbázis fejlesztés első lépése az alapos tervezés legyen! A tényleges adatbázis kialakítás csak ezután következhet.

A Microsoft Access esetében az adatbázis elkészítését a következő lépésekben célszerű lebonyolítani:

- Adatbázis tervezése: Történhet normalizálással, de egyéb adatmodellezési technikákkal is. Tananyagunkban nem érintjük például az ER, és EER modelleket, amelyek segítségével érzékletes, jól áttekinthető, és alapos adatbázis tervek készíthetők.
- Üres adatbázis állomány létrehozása: Az adatbázist már valamilyen konkrét adatbázis-kezelő rendszerrel, esetünkben az MS Accessel készítjük el.
- Táblák kialakítása az adatbázis állományban
- Táblák közötti kapcsolatok létrehozása
- További fejlesztési feladatok: lekérdezések, űrlapok, programkódok elkészítés
- Adatok kezelése: Rekordok felvétele a táblákba, módosítások törlés, keresés, szűrés

## 5.4 ADATBÁZIS ÁLLOMÁNY LÉTREHOZÁSA

Az adatbázis tervezés lépéseit a 2. leckében megismertük. Lássuk, hogyan valósíthatjuk meg tervünket az Access eszközeinek segítségével!

Ha teljesen új adatbázist szeretnénk készíteni, elsőként magát az állományt kell létrehoznunk. Kövessük az alábbi lépéseket:

- Az **Office menü/Új parancsával** indítsuk el az új adatbázis létrehozását!
- A képernyő jobb oldalán megjelenő munkaablakban adjuk meg a fájl nevét és a **mappa** gombra kattintva azt, hogy melyik alkönyvtárba szeretnénk helyezni az állományt!
- A **Létrehozás** gombbal az üres adatbázist tartalmazó ACCDB fájl létrejön a fájlrendszerben.
- Az új adatbázis létrehozásakor azonnal megjelenik a képernyőn az adatbázis első, egyenlőre még teljesen üres táblája.

Az új adatbázis kialakításakor az Access azonnal létrehoz egy táblát, amelyet felhasználói nézetben nyit meg. Bár a nézet lehetővé teszi a tábla szerkezetének változtatását és a rekordok kezelését egyaránt, a kettős használatot célszerű elkerülni, és a felhasználói nézetet csak rekordok kezelésére használni. A tábla tervezését végezzük mindig tervező nézetben!

A képernyőn felhasználói nézetben megjelenített táblát inkább zárjuk be.

## 5.5 TÁBLÁK LÉTREHOZÁSA AZ ADATBÁZISBAN

Az adatbázis kialakítás következő lépése az egyedtípusok adatainak tárolására alkalmas táblák elkészítése. Ezt a feladatot, mint minden egyebet többféle módszerrel is elvégezhetjük.

A szoftver fejlesztői úgy alakították ki az Access eszközeit, hogy a felhasználók több lehetőség közül választhassák ki a számukra leghatékonyabb megoldást. Terjedelmi okokból nem térhetünk ki minden alternatívára, ezért azokat az eszközöket mutatjuk be, amelyek a legtöbb lehetőséget bocsátják a felhasználó rendelkezésére.

A táblák létrehozását a táblatervező ablakban fogjuk végezni. Amikor létre szeretnénk hozni egy új táblát, kövessük a következő lépéseket:

- A menüszalagon kattintsunk **Létrehozás** lapra!
- A Táblák csoportban kattintsunk a **Táblatervező** gombra!
- A képernyőn megjelenik egy új tábla tervező nézete, a menüszalagon pedig a környezetfüggő, **Táblaeszközök**-et tartalmazó **Tervezés** parancslap lesz látható.

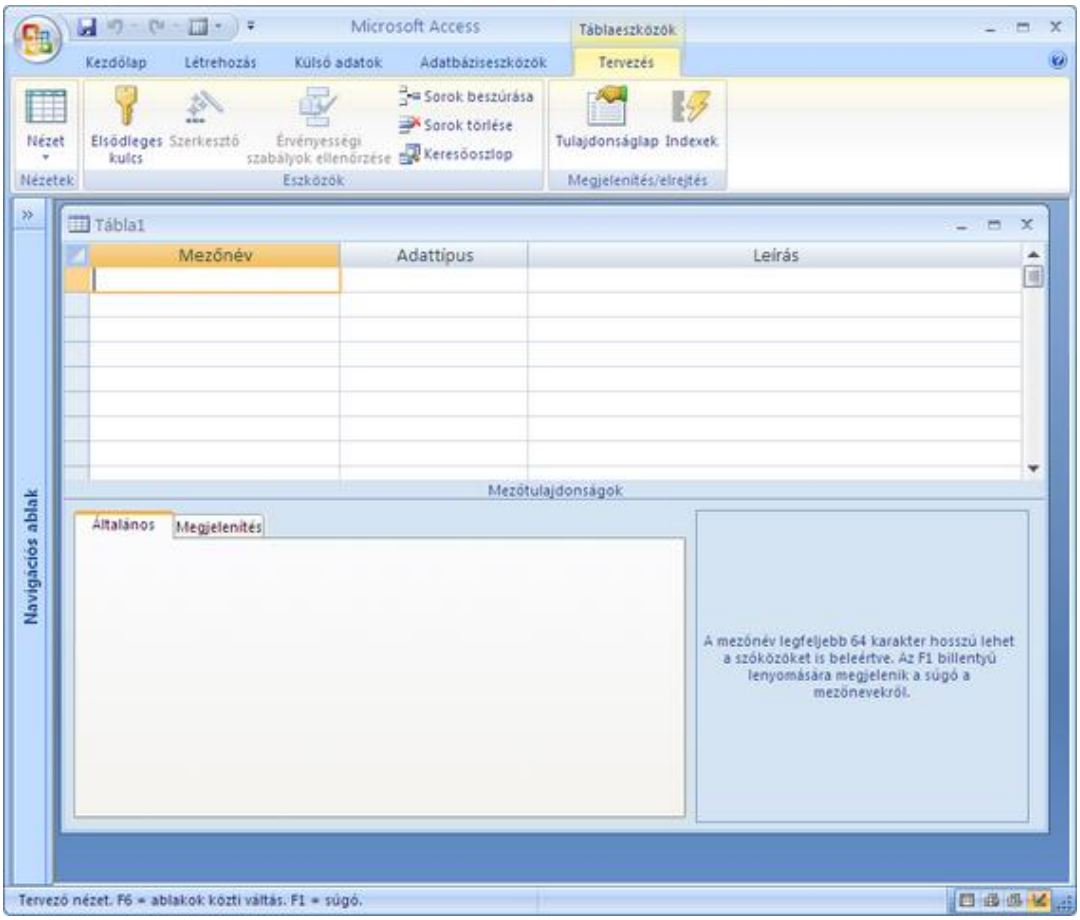



*30. ábra Tábla létrehozása*

### 5.5.1 A tábla tervező ablak felépítése

A tábla tervező ablak a felső, **mezők** részre, az alsó **mezőtulajdonságok** részre bomlik. A mezők rész egyes soraiban kell megadnunk a tábla mezőinek nevét, és az egyes mezőkben tárolni kívánt adat típusát. Minden mezőhöz meghatjuk a mező maximum 255 karakteres leírását, amely a mező kitöltésekor segíti a felhasználót.

A mezőtulajdonságok területén a kiválasztott mező jellemzői jelennek meg, illetve állíthatók be. A mezőket a soruk elején lévő kis négyzetre, a **mezőkijelölő** gombra kattintva választhatunk ki. Ha már vannak mezők a táblában, a mezőkijelölő gomb segítségével választhatjuk ki az azonosítót is.




*4. kép A táblatervező ablak*

A Tervező nézetben a tábla összes lehetséges jellemzője beállítható. Az itt elvégzett beállítások hatással lesznek más komponensek fejlesztésére is. Jelentősen megkönnyíthetik, vagy megnehezíthetik például a táblák adataival dolgozó űrlapok létrehozását. A legtöbb beállítás kitöltése belátásunkra van bízva. Kötelező azonban a mezőnév, és az adattípus kitöltése, az azonosító megjelölése pedig erősen ajánlott. Csak rendkívül indokolt esetben készítünk azonosító nélküli táblát.

### 5.5.2 Legfontosabb mezőtulajdonságok beállítása

#### *Mezőnév*

Az adatbázis objektumainak, valamint a táblák mezőinek nevét az Access névadási szabályai határozzák meg. Akár táblát, akár lekérdezést, vagy más objektumot hozunk létre, akár mezőt nevezünk el, tartsuk be az alábbi szabályokat:

- A nevek hossza 1–64 karakter lehet.
- A név tartalmazhat szóközt, de nem kezdődhet azzal.
- A mező nevében a tiltott jelektől eltekintve, szinte bármilyen nyomtatható karakterek lehetnek.
- Nem használható a névben a pont (.), felkiáltó jel (!), a szögletes zárójel ([]), valamint a nem nyomtatható karakterek (pl. sorvégjel)

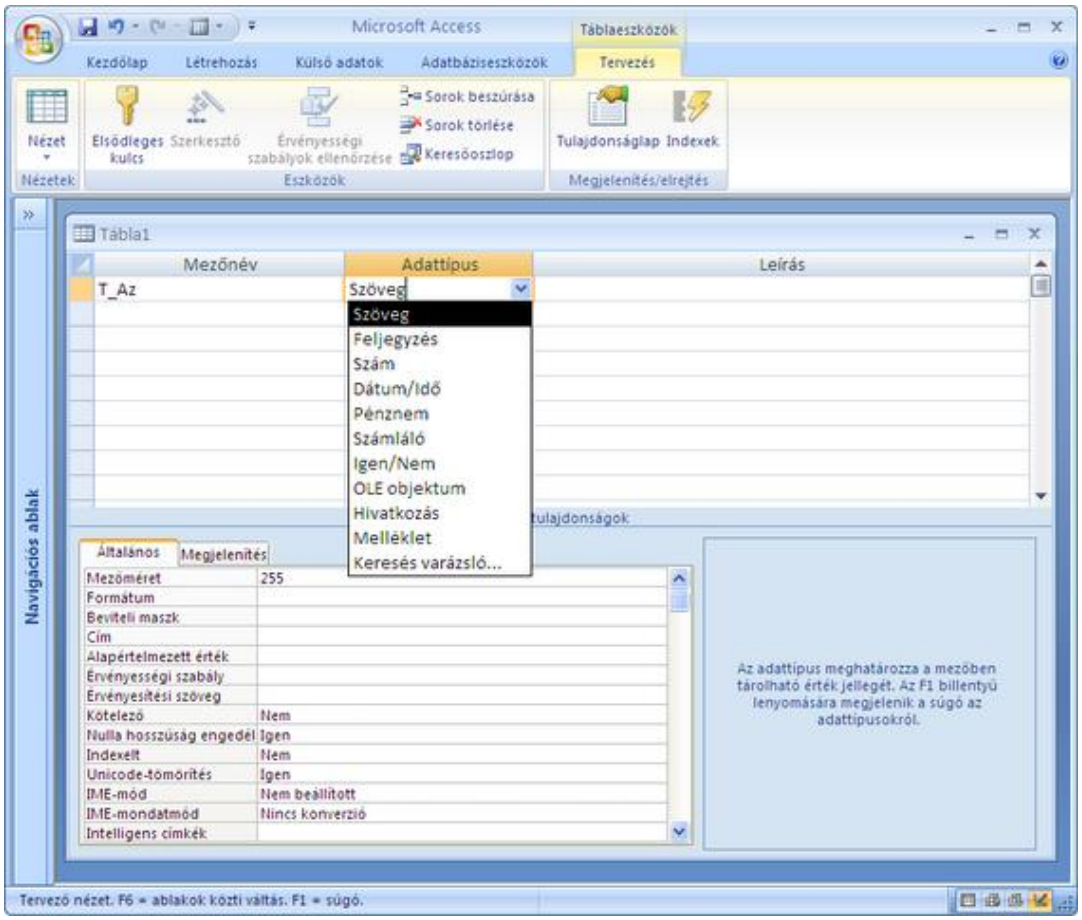
Más adatbázis kezelőkkel szemben az Access meglehetősen engedékeny a névadási szabályok dolgában. Célszerű ezeknél jóval szigorúbb megkötéseket használunk, mert így más rendszerek használatakor sem jövünk zavarba.

Lehetőség szerint

- használjunk beszédes neveket, amelyek emlékeztetnek a mezőben tárolt tulajdonságtípusra,
- legalább 3, de legfeljebb 16 karakteres neveket adjunk,
- csak az angol ABC betűit, aláhúzás jeleket, illetve számjegyeket használjunk a nevekben!

#### *Adattípus meghatározása*

Az adattípus a mezők egyik legfontosabb jellemzője. Ezzel szabályozzuk, hogy amilyen értékek tárolására lesz alkalmas a mező. Az **Adattípus** oszlopban választhatunk *Szöveg, Feljegyzés, Szám, Dátum/Idő, Pénznem, Számláló, Igen/Nem, OLE objektum*, *Hiperhivatkozás,* és *Melléklet* adattípusok közül. Az adattípusokról a következő oldalon bővebben olvashat!




*5. kép Az Access adattípusai*

### *Leírás*

A táblatervező ablak mezők rész harmadik oszlopában adható meg a mező leírása. Ez a szöveg jelenik majd meg az állapotsorban, amikor felhasználói nézetben a mezőre kattintunk. A leírással a felhasználót tájékoztatjuk arról, hogy mire használhatja a mezőt. A leírás megadása nem kötelező, maximális hossza 255 karakter lehet!

### *Azonosító megjelölése*

A tábla szerkezetének kialakításakor erősen ajánlott megjelölni az azonosítót.

- Kattintsunk a mező előtti, mezőkijelölő gombra!
- Ha összetett kulcsot akarunk készíteni, akkor előbb kattintsunk ez egyik elem mezőkijelölő gombján, majd a **Ctrl** billentyűt nyomva tartva, kattintsunk a kulcs többi eleme előtt!
- A mező(k) kijelölése után, kattintsunk a **Tervezés** parancslap **Elsődleges kulcs** gombjára!

### *Tábla mentése*

Az elkészített táblatervet menteni kell. A mentés elindítása megtörténhet az **Office menü/Mentés** parancsával, a gyorsindító eszköztár **Mentés** gombjával, vagy a **Ctrl+S** billentyű kombinációval, vagy a tábla bezárásával. (Ha egy táblaterven változtatunk, majd bezárjuk a tervezőablakot, az Access felajánlja a mentést.)

Mentéskor az ismert névadási szabályok betartásával kell megadni a tábla nevét.

## 5.5.3 Adattípusok az Access tábláiban

Az adattípus a mezők egyik legfontosabb, három jellemzőt is meghatározó tulajdonsága, így kicsit bővebben is foglalkozunk vele.

A mező adattípusa a következő jellemzőket befolyásolja:

- A mező **értékkészlete** dönti el, hogy milyen értékek lesznek tárolhatók a mezőben. (Szöveg mezőben bármilyen karakter tárolható, Szám mezőben csak számjegyek, matematikai előjelek.)
- Szintén az adattípuson múlik az, hogy milyen **műveleteket** lehet majd elvégezni a mezőben tárolt adattal. (Egy szám mezőtartalmát négyzetre lehet elemelni, egy szöveg mezőét nem.)
- Az adattípus határozza meg a mező **tárigényét** is. Egyes adattípusok értékei több, mások kevesebb bájton tárolhatók. Az informatikában fontos, hogy a lehető legkisebb tárterültet használjuk el, hiszen az a memória, illetve merevlemez takarékosabb használata mellett, az adatkezelés sebességét is befolyásolja.

Az alábbi táblázat az Access lehetséges adattípusainak leírását tartalmazza

| Adattípus | Mit tárolhatunk a mezőben | Tárigény |
|---|---|---|
| Szöveg | Legfeljebb 255 karakter hosszúságú, tetszőleges tartalmú szöveg.<br>Nem csak szöveges adatok, hanem olyan számadatok is tárolására is alkalmas, ha azokkal nem akarunk matematikai műveleteket végezni. Pl. irányítószám. | Karakterenként 2 bájt |
| Feljegyzés | Tetszőleges, akár 255 karakternél hosszabb szöveg tárolására alkalmas | Legfeljebb 2 gigabájt (2 bájt karakterenként), amelyből egyszerre 65 535 karakter jeleníthető meg. |
| Szám | Számításokban használni kívánt, egész, vagy tört számok tárolására alkalmas. Pontos beállítása a **Mezőméret** tulajdonságban adható meg:<br><br>**Bájt**: 0 és 255 közé eső számértékek tárolására.<br>**Egész**:-32 768 és +32 768 közé eső számértékek tárolására.<br>**Hosszú egész:** – -2 147 483 648 és +2 147 483 647 közé eső számértékek tárolására. | Bájt: 1 bájt<br>Egész: 2 bájt<br>Hosszú egész: 4 bájt<br>Egyszeres: 4 bájt<br>Dupla: 8 bájt<br>Decimális: 12 bájt |

| | | |
|---|---|---|
| | **Egyszeres**: – -3,4 x $10^{38}$ és +3,4 x $10^{38}$ közé eső lebegőpontos számérték, legfeljebb hét értékes számjeggyel.<br>**Dupla**:-1,797 x $10^{308}$ és +1,797 x $10^{308}$ közé eső lebegőpontos számérték, legfeljebb tizenöt értékes számjeggyel.<br>**Decimális**: -9,999... x $10^{27}$ to +9,999... x $10^{27}$ közé eső számérték. | |
| Dátum/idő | Dátum és időpont értékek tárolására használható. | 8 bájt. |
| Pénznem | Pénzügyi értékek (pénznem), vagy 4 tizedesnél nem nagyobb törtek tárolására. | 8 bájt. |
| Számláló | Valójában Hosszú egész típus, amelynek értékét az Access automatikusan képes kezelni.<br>Általában azonosítók adattípusául választjuk, mert egyedi, elsődleges kulcsként használható értékek generálhatók vele.. | 4 bájt |
| Igen/Nem | Ez az adattípus logikai adatok tárolására alkalmas. A logikai adatok mindössze két értéket vehetnek fel. (Igen/Nem, Igaz/Hamis, Férfi/Nő, Fej/Írás stb.…)<br>A tárolás számként történik, –1 felel meg az Igen, 0 a Nem értéknek. | 1 bit |
| OLE objektum | Az OLE az Object Linking and Embeding kifejezés rövidítése, objektum csatolást, és beágyazást jelent. Az ilyen mezőkben más programok által készített dokumentumok, képek, hangok vagy akár teljes Word dokumentumok is tárolhatók. | Legfeljebb 1 gigabájt. |
| Melléklet | Digitális képek és bármilyen bináris fájlok tárolására elsődlegesen használandó adattípus. | Tömörített mellékletek esetében 2 gigabájt. Tömörítetlen mellékletek esetében körülbelül 700 kilobájt, a tömöríthetőség mértékétől függően. |
| Hiperhivatkozás | Hiperhivatkozások tárolására alkalmas adattípus, amelynek tartalma négy részből állhat. Egy a képernyőn megjelenítendő szövegből, egy URL címből, a címhez kapcsolódó alcímből, illetve egy előugró tipp szövegre. Az egyes részeket # jellel választjuk el egymástól. | Legfeljebb 2 gigabájt (2 bájt karakterenként), amelyből egyszerre 65 535 karakter jeleníthető meg. |
| Keresés varázsló | Nem igazi adattípus, hanem a Keresés varázslót hívja elő. | |

### 5.5.4 Adattípus kiválasztása

Az adattípus kiválasztásakor az alábbi szempontokat érdemes megfontolni:

- Milyen értékeket szeretnénk eltárolni a mezőben? (Nem használhatunk Igen/Nem típust, ha dátumokat kell elhelyeznünk a mezőben.)
- Ha az értékek tárolására több típus is alkalmas lenne, döntsön az, hogy milyen műveleteket szeretnénk elvégezni a mezővel. (Könyvek kiadási éve például számként és szövegként is tárolható. A kiadási évet mégis szerencsésebb számként tárolni. Előfordulhat ugyanis, hogy két könyv kiadási évét kivonjuk egymásból, hogy megtudjuk hány év telt el a kiadások között. Szövegek nem vonhatók ki egymásból.)
- Ha a két előző szempont alapján még mindig nem tudunk dönteni, válasszuk az alkalmas típusok közül a lehető legkisebb tárigényűt!

#### *Azonosító típusa*

Azonosító adattípusa az **OLE objektum**-on, és **Melléklet**-en kívül bármi lehet (más kérdés, hogy **Igen/Nem** típusú azonosítót nem célszerű létrehozni, hiszen ez mindössze két rekord felvételét tenné lehetővé). A kulcsokat azonban szinte kivétel nélkül Számláló típusúra állítjuk. A számláló típusú mezőt ugyanis az Access automatikusan kezeli, azaz minden új rekord felvételekor maga gondoskodik kitöltéséről. A Számláló az adattípusú mező értékét felfoghatjuk úgy is, mint egy automatikus sorszámot, amelyet a mező akkor kap, amikor rekordját hozzáadjuk egy táblához. Ha például egy táblában van egy *ID* nevű **Számláló** típusú mező, akkor a tábla első rekordjában az *ID* mező értéke 1 lesz. A második rekord *ID* mezője a 2, a harmadik rekord *ID* mezője pedig a 3 értéket kapja.

> A Számláló típussal kapcsolatban már most meg kell jegyeznünk néhány fontos dolgot!
>
> Az ilyen mező **Hosszú egészként** tárolódik. (Lásd: Idegen kulcsok típusa)
>
> Egy táblában mindössze egy számláló típusú mező lehet.
>
> A mező kitöltése automatikus, a mező értékét a felhasználó nem befolyásolhatja.

#### *Idegen kulcsok típusa*

> Az idegen kulcsok típusát az határozza meg, hogy milyen elsődleges kulcshoz kapcsolódnak. Egy idegen kulcs típusa és mérete pontosan olyan kell, legyen, mint a kapcsolódó elsődleges kulcs tárolásának típusa és mérete.

Ha például egy elsődleges kulcs Számláló, (ne feledjük, hogy a Számláló típus Hosszú egészként tárolódik) akkor az idegen kulcs is Hosszú egész típusú Szám lesz. Ha az elsőleges kulcs 15 karakter hosszúságú szöveg, akkor az idegen kulcs is 15 karakter hosszúságú szöveg lesz!

### 5.5.5 További mezőtulajdonságok

A mezőtulajdonságok ablakrészben a mezők további jellemzői állíthatók be. Egy mező tulajdonságait úgy határozhatjuk meg, hogy a meződefiníciós részben rákattintunk a mező sorára, majd szintén egérkattintással a mezőtulajdonságok ablakrészre lépünk.

Az egyes tulajdonságok megadásakor használhatjuk a billentyűzetet, bizonyos esetekben legördülő listákat, illetve varázslót. Ez utóbbiak úgy indíthatók, hogy a mező neve mellett megjelenő ... gombra kattintunk.

Bizonyos mezőtulajdonságok egy adott típushoz kötődnek, míg mások minden típus esetén használhatók. Tananyagunkban csak röviden tekintjük át a mezőtulajdonságokat. Egy tulajdonság kiválasztása után az F1 billentyű lenyomásával bővebb információt kaphatunk az Access Súgójából.

#### *Mezőméret*

Szöveg és szám típusú mezők esetén használható tulajdonság. Szövegek esetén a szöveg maximális karakterszámát adhatjuk meg itt, számok esetén ezzel a tulajdonsággal adható meg a valódi adattípus.

#### *Formátum*

Szám, Dátum/Idő, Szöveg és Igen/Nem típusú adatok megjelenési módját szabályozza. Nincs hatása az adat tárolt változatára, csak annak megjelenését határozza meg.

#### *Tizedes helyek*

Lebegőpontos számok és Pénznem adattípus esetén a tizedes elválasztótól jobbra megjelenő számjegyek számát határozza meg. Csak akkor hatásos, ha a mező Formátum tulajdonságát Általános szám-tól eltérően állítjuk be.

#### *Bemeneti maszk*

Segítségével formázó karaktereket jelenthetünk meg a mezőben, illetve szabályozhatjuk, hogy a felhasználó milyen karaktereket gépelhet be.

#### *Cím*

A mezőkhöz olyan feliratot rendelhetünk ezzel a tulajdonsággal, amely a tábla felhasználói nézetben való megnyitásakor a mezőkijelölő gombon látható.

Ha üresen hagyjuk a gomb felirata a mező neve lesz.

#### *Alapértelmezett érték*

Új rekord felvételekor a mező automatikusan az itt megadott értéket kapja.

#### *Érvényességi szabály*

Segítségével szabályozható a mezőbe begépelhető adatok értékkészlete.

### *Érvényességi szöveg*

Olyan szöveg, ami az Érvényességi szabályt megsértő adat begépelésekor jelenik meg.

### *Kötelező*

Amennyiben Igen-re állítjuk, adatbevitelkor a mező nem hagyható üresen.

### *Nulla hosszúság engedélyezése*

Látszólag értelmetlen tulajdonság. Igen-re állításával azonban elérhető, hogy a felhasználó csak szándékosan hagyjon üresen egy mezőt.

### *Indexelt*

Rendezésre, keresésre gyakran használt mezők esetén van jelentősége.

## 5.6 ELKÉSZÍTETT TÁBLÁK HASZNÁLATA

Amikor egy táblát létrehoztunk és tervét mentettük, a tábla fizikailag rögtön alkalmas arra, hogy felhasználói nézetbe válthatva, rekordokat gépeljünk be. Célszerű azonban megvárni az adatbázis kialakítás következő lépését, a táblák közötti kapcsolatok létrehozását. A kapcsolatok létrehozása előtt begépelt rekordokban véletlenül elronthatjuk az idegen kulcsok értékeit. Ha előbb létrehozzuk a kapcsolatokat, és csak azután veszünk fel rekordokat a táblákba, akkor biztosak lehetünk abban, hogy az idegen kulcsok mindig helyes értékeket tartalmaznak majd.

A kapcsolatok kialakításáról a későbbi leckékben olvashat.

## 5.7 ÖSSZEFOGLALÁS

Az Access adatbázisok kialakítást mindig alapos tervezőmunkának kell megelőznie. Ha rendelkezünk a táblákat, azok mezőit és kapcsolatait is tartalmazó tervvel, akkor létrehozhatunk egy üres adatbázist, abban pedig kialakíthatjuk a táblákat.

A táblák többféleképpen is létrehozhatók, de a legprecízebb beállítási lehetőségeket a tervező nézet biztosítja. A tábla tervező nézetében a mező nevét, és adattípusát kötelező, az elsődleges kulcsot pedig ajánlott megadni. Ezeken kívül begépelhető a leírás, és megadhatók további mezőtulajdonságok is.

A mező- és táblanevek nevek megadásakor be kell tartani az Access névadási szabályait. Az adattípus kiválasztásakor pedig a tárolandó értékeket, a tervezett műveleteket, és a szükséges tárigényt kell figyelembe venni.

Az idegen kulcsok típusát mindig a kapcsolódó elsődleges kulcs típusa és mérete határozza meg. Általában Számláló típusú elsődleges kulcsot használunk. Az ilyen kulcshoz kapcsolódó idegen kulcsok típusa Hosszú egész.

A táblák, létrehozás után már alkalmasak rekordok tárolására, a rekordfelvétel előtt azonban célszerű kialakítani a táblák közötti kapcsolatot is.

## 5.8 ÖNELLENŐRZŐ KÉRDÉSEK

1. Miért érdemes Tervező nézetben kialakítani a táblákat?
2. Milyen beállításokat kötelező elvégezni a tábla létrehozásakor?
3. Tartalmazhat-e egy mező neve szóközt?
4. Lehet-e olyan táblát létrehozni, amiben nincs azonosító?
5. Hogyan jelzi, hogy melyik mező az azonosító
6. Milyen típusú legyen az azonosító?
7. Hány számláló mező lehet egy táblában?
8. Tárolhatók-e számok egy Szöveg típusú mezőben?
9. Tárolható-e szöveg a szám típusú mezőben?
10. Milyen típusú mezőben tárolná egy személyzeti adatbázisban dolgozók életrajzát?

# 6. TÁBLÁK KÖZÖTTI KAPCSOLATOK KIALAKÍTÁSA

## 6.1 CÉLKITŰZÉS

Az adatbázis tervezés során tapasztalhattuk, hogy a terv nem csupán arról szól, hogy milyen táblákat hozunk létre, azokban milyen rekordokat, és mezőket tárolunk el. Az adatbázis szerves részét képezik a táblák közötti kapcsolatok is.

A relációs adatmodellben a kapcsolatok leírására idegen kulcsokat használunk, amelyeket már tervezéskor elhelyezünk a kapcsolódó táblákban. Az Access-ben azonban, idegen kulcsok mellett grafikusan is jelezhetjük a kapcsolatokat, sőt szigorításokat vezethetünk be az idegen kulcsok értékeivel kapcsolatban. Ebben a leckében a kapcsolatok kialakításáról lesz szó. Megtudhatja, milyen előnyei vannak a kapcsolatok kialakításának, megismeri a kapcsolatszerkesztő ablak kezelését, megtanulhatja, hogyan biztosíthatja az idegen kulcsok helyességét az adatbázisban. Ha helyesen tervezi meg az adatbázist, megfelelően alakítja ki a táblákat, és pontosan beállítja a kapcsolatokat, biztos lehet benne, hogy adatbázisa hibátlanul működik majd.

## 6.2 A LECKE TÉMAKÖREI

- Kapcsolatok jelentősége
- Kapcsolat létrehozása
- Hivatkozási integritás megőrzése
- Kapcsolt mezők kaszkádolt frissítése
- Kapcsolt mezők kaszkádolt törlése

## 6.3 A KAPCSOLATOK JELENTŐSÉGE

Az Access-ben idegen kulcsok mellett grafikusan is jelezhetjük a kapcsolatokat, sőt a kapcsolatok ablakban többféle opcióval is szabályozhatjuk a kapcsolatok működését. A táblák létrehozása után mindig a kapcsolatokat alakítjuk ki, és csak azután tárolunk rekordokat a táblákban.

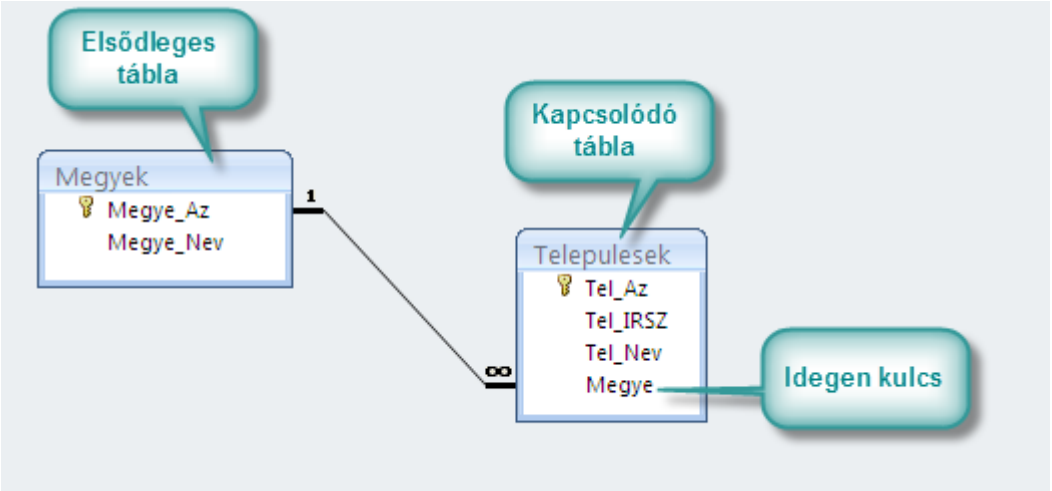
A kapcsolatok létrehozását három szempont indokolja:

- Az adatbázis szerkezete jól áttekinthető
- Egyszerűsödik a táblák kapcsolatára épülő további komponensek, Lekérdezések, Űrlapok, Jelentések létrehozása.
- Bekapcsolható a **Hivatkozási integritás megőrzése**, amellyel biztosítható az idegen kulcsok helyessége.

## 6.4 FOGALMAK

Két tábla kapcsolatát úgy írjuk le, hogy az egyik tábla azonosítójának értékeit a másik táblában elhelyezett idegen kulcsban tároljuk. Azt a táblát, amelyikben az elsődleges kulcs vesz részt a kapcsolatban, elsődleges táblának nevezzük. Azt, amelyikben az idegen kulcs

van, kapcsolódó táblának hívjuk. A kapcsoltban résztvevő mezőket másképpen kapcsolódó mezőknek nevezzük.




*31. ábra A Megyek-Telepulesek táblák kapcsolata*

## 6.5 A KAPCSOLATOK ABLAK

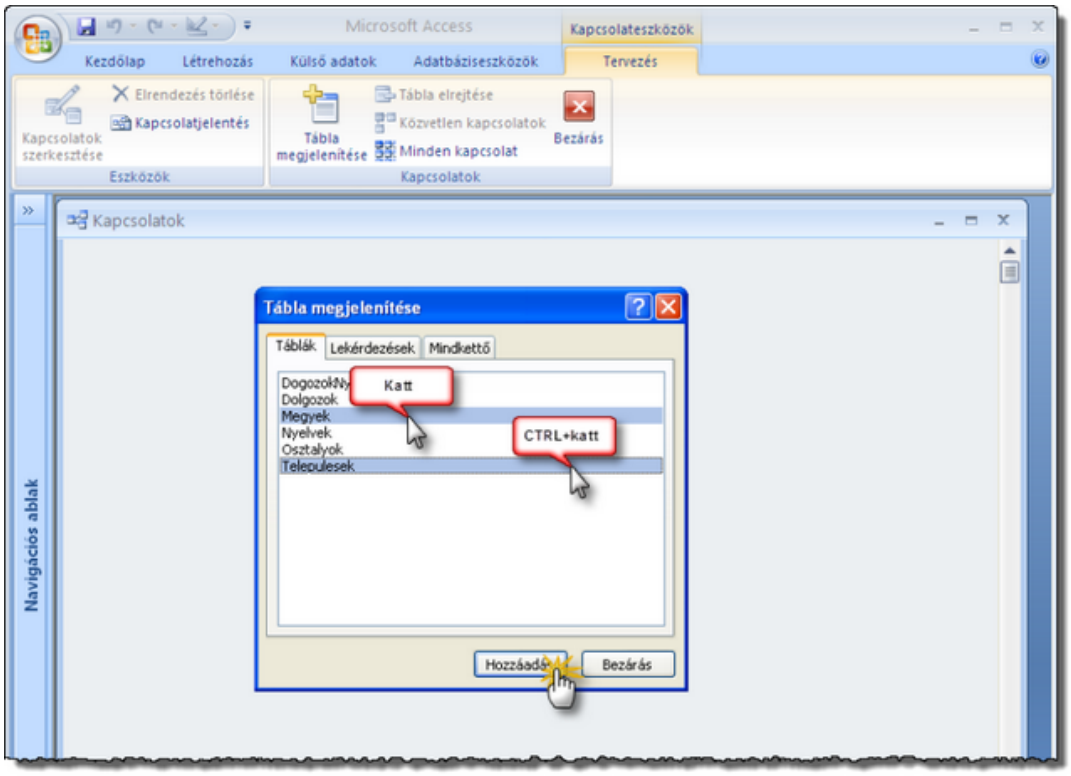
A kapcsolatok kialakítása a **Kapcsolatok** ablakban történik. Az ablak az **Adatbázis-eszközök** parancslapon, a **Megjelenítés/elrejtés** csoportban található **Kapcsolatok** gombbal nyitható meg.




*32. ábra Kapcsolatok ablak megjelenítése*

A **Kapcsolatok** ablak kezdetben üres. A kapcsolatok kialakítását az alábbi lépésekkel kezdhetjük el:

- Első lépésként, az automatikusan megjelenő, **Tábla megjelenítése** ablakban ki kell választanunk azokat a táblákat, amelyek között kapcsolat akarunk kialakítani. Az ablakból kiválaszthatjuk egyszerre az összes táblát, de haladhatunk tábla-páronként is.
- A kiválasztott táblákat a **Hozzáadás** gombbal helyezzük a **Kapcsolatok** ablakba.



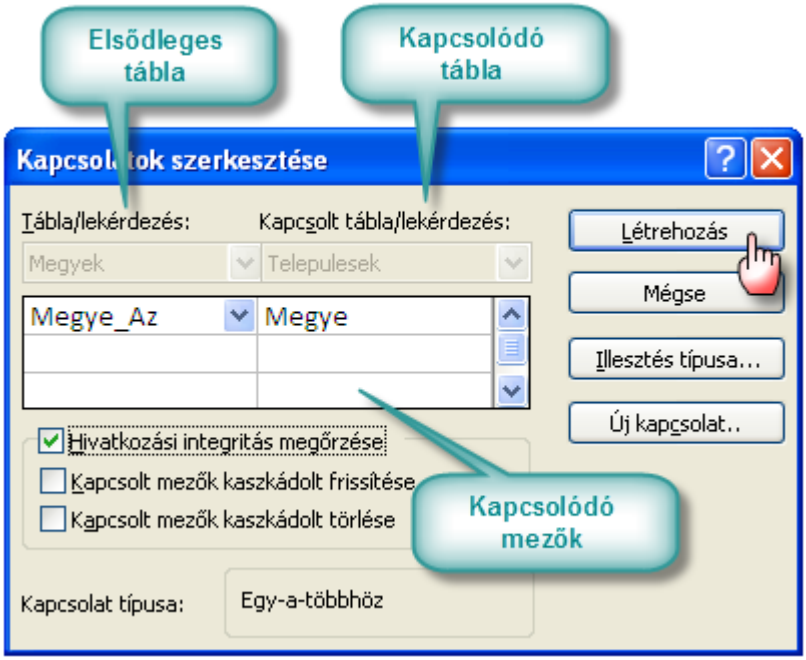

*33. ábra Táblák hozzáadása*

A **Kapcsolatok** ablakban mezőlisták jelképezik táblákat. A mezőlistákon olvasható a táblák, és a táblákban található mezők neve. A mezőlisták szabadon elrendezhetők, és méretezhetők.

### 6.5.1 Kapcsolat létrehozása

Két tábla között úgy hozhatunk létre kapcsolatot, hogy mezőlistáikból egymásra húzzuk a kapcsolatban résztvevő mezőket.

Mindig az elsődleges tábla azonosítóját visszük a kapcsolódó tábla idegen kulcsára. Az egér elengedésekor megjelenik a **Kapcsolat szerkesztése** ablak, amelyben ellenőrizhetjük, sőt korrigálhatjuk is a kapcsolódó mezőket, valamint szabályozhatjuk a kapcsolat tulajdonságait. A tulajdonságok beállítása után a **Létrehozás** gombbal alakíthatjuk ki a kapcsolatot. Ilyenkor megjelenik a két tábla kapcsolatát jelképező vonal, amely a kapcsolódó mezőket köti össze.




*34. ábra Kapcsolatok szerkesztése*

### *Kapcsolat utólagos szerkesztése*

Ha utólag szeretnénk módosítani a kapcsolat tulajdonságain, kattintsunk duplán a kapcsolat vonalán! Célozzunk pontosan, az egérkurzor hegye legyen a vonal fölött!

A dupla kattintásra ismét látható lesz a **Kapcsolatok szerkesztése** ablak.

### 6.5.2 Hivatkozási integritás megőrzése

Az adatbázis-kezelő rendszerek az egyedtípusok adatainak és azok kapcsolatainak tárolásán kívül képesek ügyelni az adatbázis konzisztenciájára. Azaz képesek figyelni, sőt kizárni az egymásnak ellentmondó adatok keletkezését.

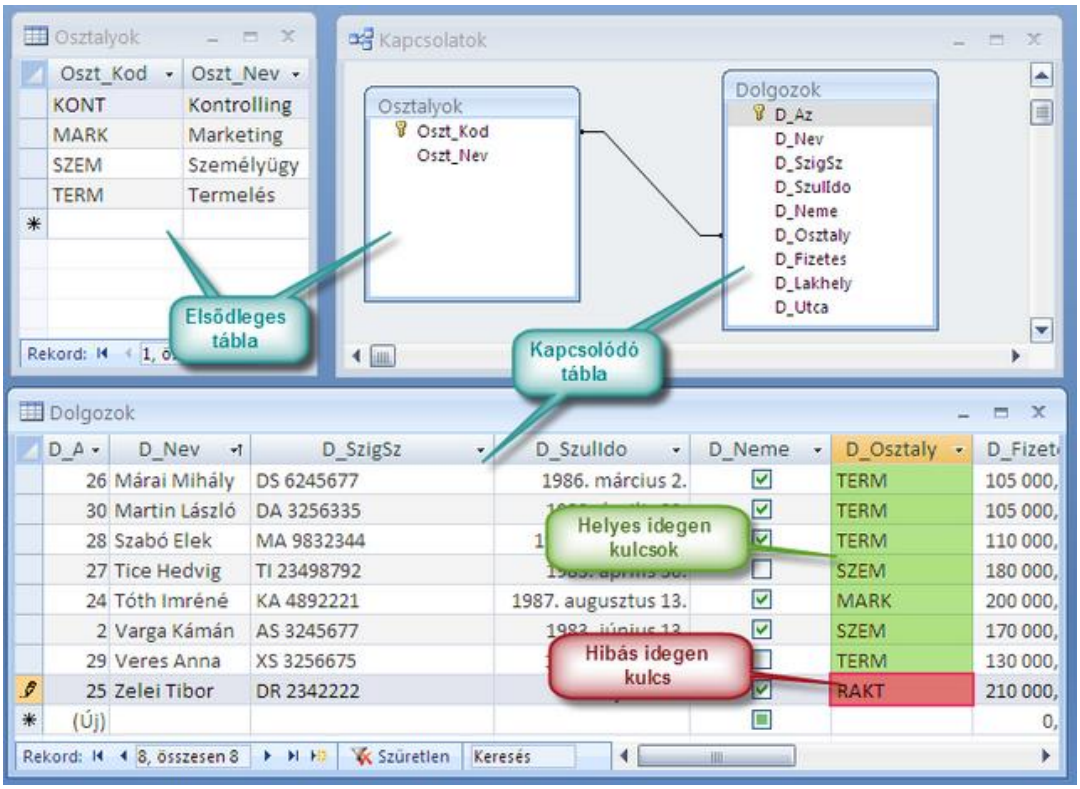
Ellentmondás keletkezhetne például akkor, ha egy idegen kulcsban tetszőleges értékeket tárolhatnánk. Ha egy idegen kulcs minden megkötés nélkül kitölthető, akkor elvileg semmi sem akadályozza meg a felhasználót abban, hogy olyan értéket tároljon benne, amely ez elsődleges táblában egyetlen rekordot sem azonosít.

A **Hivatkozási integritás megőrzésének** bekapcsolásával elérhetjük, hogy az Access, a hibás idegen kulcsok tárolásának megakadályozásával őrizze meg az adatbázis konzisztenciáját.

Az elsődleges tábla egyes rekordjait elsődleges rekordoknak nevezzük. A kapcsolódó táblában azt a rekordot, ami az idegen kulcsban egy elsődleges rekord azonosítóját tartalmazza, kapcsolódó rekordnak hívjuk. Egy elsődleges rekordhoz 1:1 kapcsolat esetén csak

egy, 1:N kapcsolat esetén több kapcsolódó rekord kapcsolódhat. Az elsődleges rekordot, és a kapcsolódó rekordot a kapcsolódó rekordban tárolt idegen kulcs értéke kapcsolja össze.

Az adatbázis inkonzisztenciáját az okozza, ha a kapcsolódó táblában hibás idegen kulcsok találhatók.




*35. ábra Helyes és hibás idegen kulcs*

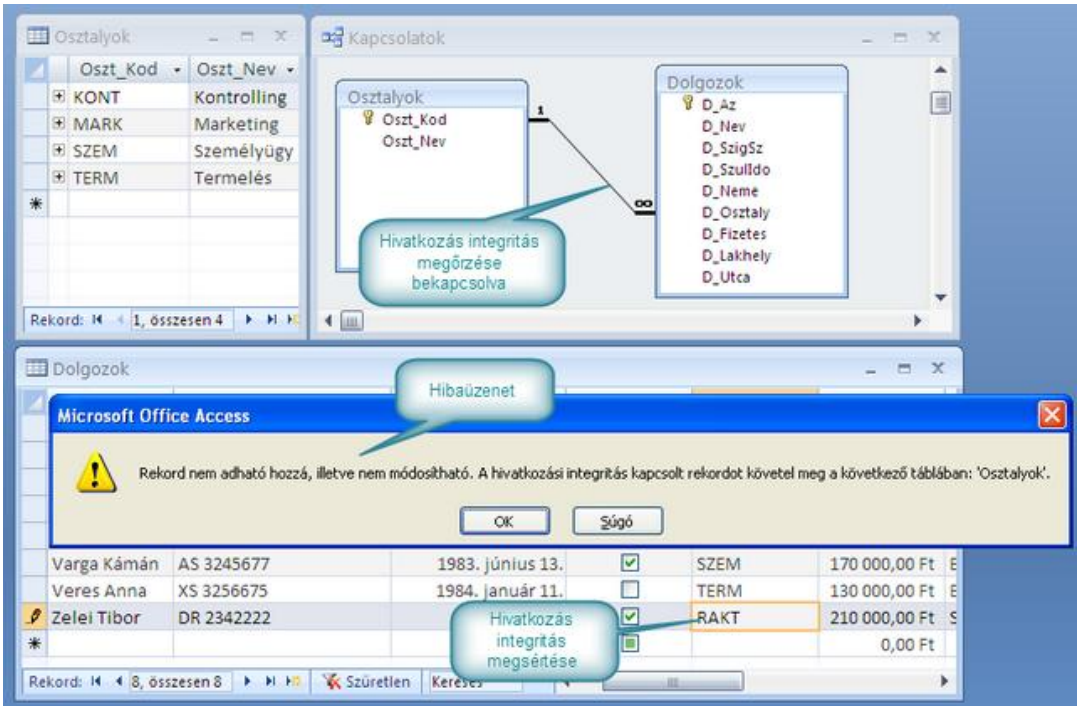
> A hivatkozási integritás azt jelenti, hogy egy táblakapcsolat kapcsolódó táblájában az idegen kulcs minden rekordban helyes értéket tartalmaz, azaz minden rekord kapcsolódik az elsődleges tábla valamelyik rekordjához.

A hivatkozási integritás megőrzése az adatbázis-kezelő rendszer szolgáltatása, amelyet a két tábla kapcsolatának kialakításakor kapcsolhatunk be. Ha bekapcsoljuk a szolgáltatást, akkor az adatbázis-kezelő rendszer ebben a kapcsolatban megakadályozza a hibás idegen kulcsok, egyben az inkonzisztencia kialakulását

### *Hivatkozási integritás megőrzésének bekapcsolása*

A Hivatkozási integritás megőrzését a kapcsolat létrehozásakor, vagy a kapcsolat tulajdonságainak megváltoztatásakor kapcsolhatjuk be. Ha két tábla kapcsolatában bekapcsoljuk a hivatkozási integritás megőrzését, majd a kapcsolódó tábla egy rekordjában hibás idegen kulcsot adunk meg, akkor az Access nem engedi menteni a rekordot, hanem hiba-

üzenetet jelenít meg. A hibaüzenet elolvasása után a felhasználó javíthatja a hibás idegen kulcsot, vagy visszavonhatja az új rekord felvételét, de semmiképpen sem tárolhatja a hibás értéket.




*36. ábra Hivatkozási integritás megsértése*

A Hivatkozási integritás megőrzésének bekapcsolása után kapcsolatok ablakban látható a kapcsolat típusa is (1:∞). Az Access 1:1 kapcsolatot jelez, ha az idegen kulcs egyedi indexeléssel rendelkezik. Más esetben 1:∞ kapcsolat jelenik meg.

> A hivatkozási integritás megőrzése nem kapcsolható be, ha a táblákban már vannak rekordok, és a kapcsolódó tábla eleve tartalmaz hibás idegen kulcsokat.
>
> Akkor sem lehet bekapcsolni a hivatkozási integritás megőrzését, ha az idegen kulcs adattípusa nem felel meg az elsődleges tábla azonosítójának.

### *Kapcsolt mezők kaszkádolt frissítése*

A hivatkozási integritás sérülhet, ha egy kapcsolódó táblába hibás idegen kulcsot tartalmazó rekordot veszünk fel, de az is előfordulhat azonban, hogy az elsődleges táblában már felvett rekordok értékeinek utólagos megváltoztatásakor keletkezik inkonzisztencia.

Ha egy táblakapcsolatban az elsődleges táblában utólag változtatjuk meg az azonosító értékét, akkor a kapcsolódó táblában a kapcsolódó rekordokban az eredeti értéket tartalmazó, idegen kulcsok maradnak, amelyek az elsődleges táblában már nem létező értéket tartalmaznak, tehát hibásak. Emiatt sérülne a hivatkozási integritás, és az adatbázis inkonzisz-

tens lenne. Ha bekapcsoljuk a hivatkozási integritás megőrzését, akkor az Accessnek arra is kell ügyelnie, hogy az utólag megváltoztatott elsődleges kulcsok miatti inkonzisztenciát megakadályozza.

Ez két úton lehetséges:

- Alapbeállítás szerint a program nem engedi, hogy utólag megváltoztassuk az elsődleges kulcs értékét,
- Ha mégsem akarunk lemondani az elsődleges kulcs utólagos megváltoztatásáról, el kell érnünk, hogy amikor megváltozik egy elsődleges kulcs értéke, akkor a kapcsolódó táblában automatikusan az új értékre változzanak a kapcsolódó rekordok idegen kulcsainak értékei is. Ezt, az utóbbi megoldást nevezzük kaszkádolt frissítésnek.

**A kapcsolt mezők kaszkádolt frissítése az jelenti, hogy az elsődleges kulcs értékének utólagos megváltoztatásakor az Access automatikusan az új értékre frissíti a kapcsolódó rekordok idegen kulcsainak értékeit is.**

A kapcsolt mezők kaszkádolt frissítését a kapcsolat kialakításakor, vagy tulajdonságainak utólagos megváltoztatásakor a **Kapcsolatok szerkesztése** ablakban kapcsolhatjuk be.

Ha nincs bekapcsolva a hivatkozási integritás megőrzése, akkor az Access nem ellenőrzi, hogy megváltozik-e egy elsődleges kulcs értéke. Ilyenkor a kaszkádolt frissítést be sem kapcsolhatjuk.

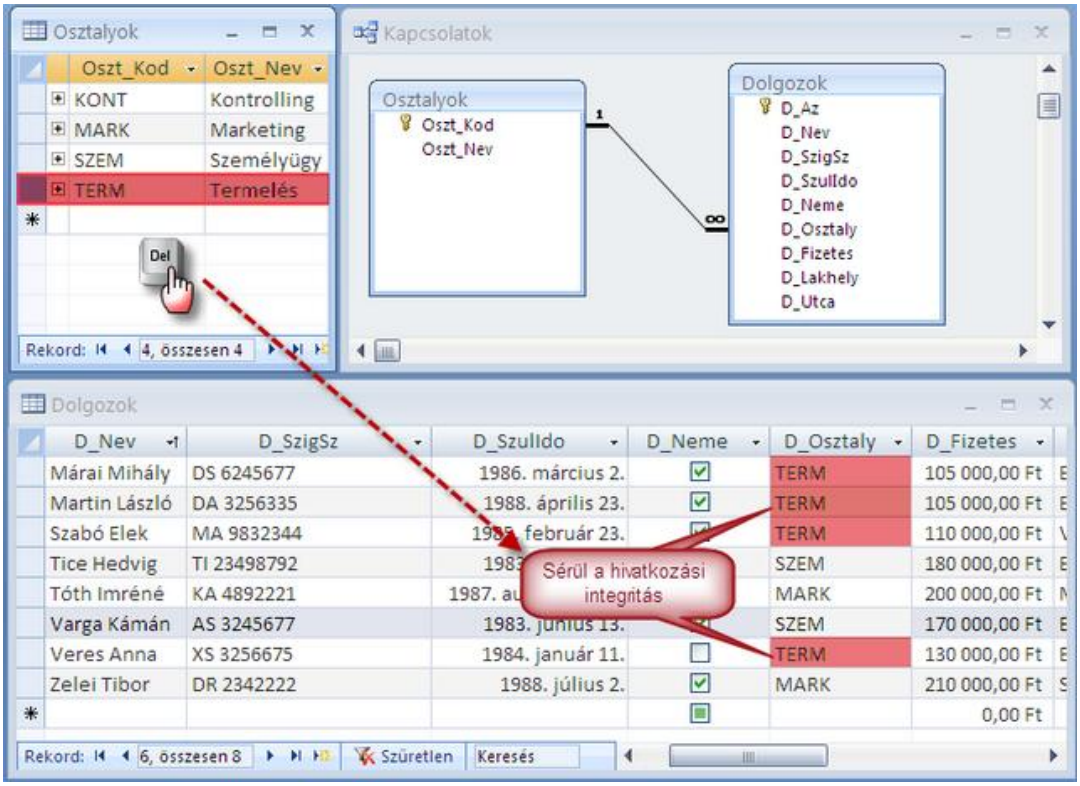
**Mikor nem kapcsoljuk be a kaszkádolt frissítést?**

A kaszkádolt frissítést soha sem kapcsoljuk be, ha az elsődleges tábla kulcsa számláló típusú. Az számláló típusú mező kezelése ugyanis automatikus, így utólag nem változtatható meg az értéke.

### *Kapcsolt mezők kaszkádolt törlése*

A hivatkozási integritás sérülhet, ezzel az adatbázis inkonzisztenssé válhat, ha hibás idegen kulcsot tárolunk, vagy az elsődleges táblában utólag változtatjuk meg a kulcs értékét. A fenti eseteken kívül okozhatja a hivatkozási integritás sérülését az is, ha az elsődleges táblából törlünk egy olyan rekordot, amihez a kapcsolódó táblában kapcsolódik rekord.

Ha egy kapcsolat elsődleges táblájából törlünk egy olyan rekordot, amihez a kapcsolódó táblában kapcsolódik rekord, akkor sérül a hivatkozás integritás, mert a kapcsolódó rekord „árván” marad, azaz olyan rekordra fog hivatkozni, amely már nem létezik az elsődleges táblában.

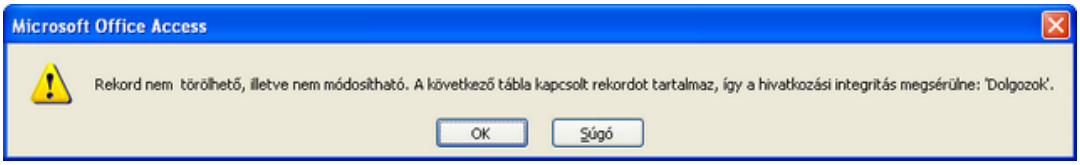



*37. ábra Hivatkozási integritás megsértése rekord törlésével*

**Hogyan reagál az Access az elsődleges rekordok törlésére?**

Ha bekapcsoljuk a hivatkozási integritás megőrzését, akkor az Access kétféle képen képes kizárni az elsődleges rekordok törlésekor keletkező inkonzisztenciát:

- Nem enged rekordokat törölni az elsődleges táblából, ha azokhoz rekord kapcsolódik a kapcsolódó táblában.



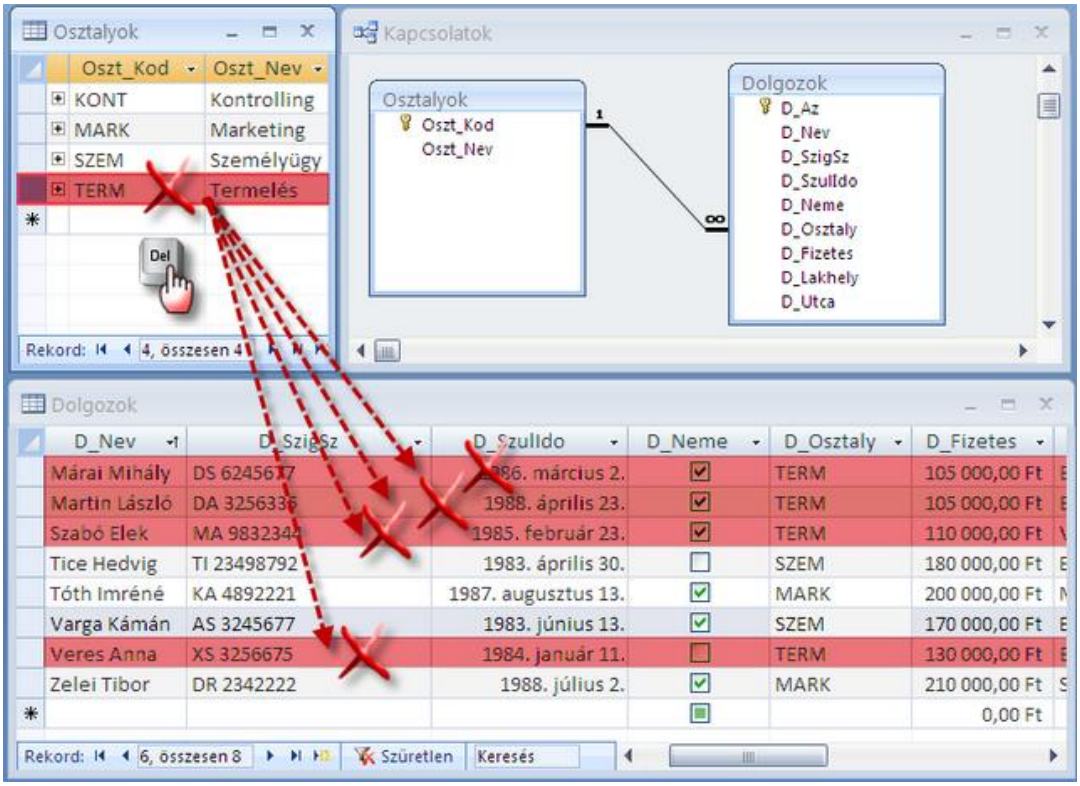

*6. kép Hibaüzenet törléskor*

- Ha az elsődleges táblából törlünk egy rekordot, akkor automatikusan törli a kapcsolódó tábla, kapcsolódó rekordjait is. Ezt a működést nevezzük kaszkádolt törlésnek.

**A kapcsolt mezők kaszkádolt törlése az jelenti, hogy ha törlünk egy rekordot az elsődleges táblából, akkor az Access automatikusan törli a kapcsolódó tábla kapcsolódó rekordjait is.**

### A kapcsolt mezők kaszkádolt törlésének bekapcsolása

A kapcsolt mezők kaszkádolt törlését a kapcsolat kialakításakor a kapcsolatok szerkesztése ablakban kapcsolhatjuk be. Ha nincs bekapcsolva a hivatkozási integritás megőrzése, akkor az Access nem ellenőrzi, hogy törlünk-e rekordot az elsődleges táblából. Ilyenkor a kaszkádolt törlést be sem kapcsolhatjuk.




*38. ábra Kaszkádolt törlés hatása*

A kaszkádolt törlés veszélyes művelet, mert 1:∞ kapcsolat esetén egyetlen rekord törlése az elsődleges táblában, a kapcsolódó tábla több rekordjának törlésével járhat. Bekapcsolását alaposan át kell gondolni.

## 6.6 KAPCSOLATOK MENTÉSE

A kapcsolatok ablakban elvégzett műveletek után mentenünk kell a beállításokat. A mentés elvégezhető az gyorsindítási eszköztár Mentés gombjával, a **Ctrl+S** billentyűkkel, illetve a Kapcsolatok ablak bezárásakor.

A mentett kapcsolatok bármikor megtekinthetők a Kapcsolatok ablak megnyitásával. Erre – főleg bonyolultabb adatbázisok esetén – gyakran van szükség, amikor több tábla kapcsolatára épülő lekérdezéseket, űrlapokat, vagy jelentéseket hozunk létre.

## 6.7 ÖSSZEFOGLALÁS

A táblák közötti kapcsolatok létrehozása és megjelenítése része az adatbázis dokumentációjának, megkönnyíti az adatbázis továbbfejlesztését, és segít a hivatkozási integritás megőrzésében.

Két tábla kapcsolatában elsődleges táblának hívjuk azt a táblát, amelynek azonosító értékei másik tábla idegen kulcsában tárolódnak. Az idegen kulcsot tároló táblát kapcsolódó táblának hívjuk. Az elsődleges tábla rekordjai az elsődleges rekordok. Egy elsődleges rekordhoz kötődő, kapcsoló táblabeli rekordot kapcsolódó rekordnak nevezünk.

Az elsődleges rekord, és a kapcsolódó rekord kapcsolatát a kapcsolódó rekordban tárolt idegen kulcs értéke tartja fenn. A kapcsolatban az elsődleges tábla azonosítója, és a kapcsolódó tábla idegen kulcs az úgynevezett kapcsolódó mezők.

A hivatkozás integritás azt jelenti, hogy egy kapcsolatban minden idegen kulcs helyes értéket tartalmaz, azaz a kapcsolódó tábla minden rekordja kapcsolódó rekord.

A Hivatkozási integritás megőrzése azt jelenti, hogy az Access nem engedi, hogy hibás idegen kulcsok keletkezzenek.

A kaszkádolt frissítéskor az egy elsődleges rekordban utólag megváltozó azonosító értékére frissül a kapcsolódó rekord(ok) idegen kulcsa is.

Kaszkádolt törléskor egy elsődleges rekord törlésekor automatikusan törlődnek a kapcsolódó tábla kapcsolódó rekordjai is.

## 6.8 ÖNELLENŐRZŐ KÉRDÉSEK

1. Miért nem célszerű a kapcsolatok kialakítása előtt rekordokkal feltölteni a táblákat?
2. Mit értünk elsődleges tábla alatt?
3. Elég-e ha két tábla kapcsolatát csak az idegen kulcsok tárolják?
4. Tegyük fel, hogy egy kapcsolatban bekapcsolta a hivatkozási integritás megőrzését, de nem kapcsolta be a kaszkádolt törlést. Törölhet-e rekordot az elsődleges táblából?
5. Tegyük fel, hogy egy kapcsolatban bekapcsolta a hivatkozási integritás megőrzését, de nem kapcsolta be a kaszkádolt törlést. Törölhet-e rekordot a kapcsolódó táblából?
6. Mi a kaszkádolt törlés veszélye?
7. Mikor nem célszerű bekapcsolni a kaszkádolt frissítést?
8. Mikor nem tudja bekapcsolni a hivatkozási integritás megőrzését?
9. Mi a teendő ilyenkor?
10. Megtekinthetők-e később a korábban létrehozott kapcsolatok?

# 7. REKORDOK BEVITELE, RENDEZÉSE, KERESÉSE, SZŰRÉSE

## 7.1 CÉLKITŰZÉS

A relációs adatbázisokban, táblákban tároljuk az egyes egyedtípusokat, a táblák rekordjaiban pedig az egyedtípus egyedeit. A rekorddokkal számos különböző művelet végezhető el. A legalapvetőbbek az új rekord felvétele, vagy bővítés, a törlés, illetve a változtatás. Ezen kívül lehetőségünk van a rekordok sorba rendezésére, a rekordok közötti keresésre, illetve a rekordok különböző szempontok szerinti kiválogatására, szűrésre.

Ebben a leckében megismerheti az Access adatbázisok tábláinak felhasználói nézetét, és a felhasználói nézetben használható eszközöket. Megtanulja, hogyan lehet új rekordokkal bővíteni egy táblát, tallózni a rekordok között, megváltoztatni a rekordokban tárolt mezőértékeket, illetve törölni a fölöslegessé vált rekordokat. A legelemibb rekordkezelő műveletek mellett, megismerheti a keresés, rendezés, és szűrés eszközöket, amelyek birtokában a tallózásnál hatékonyabban kutathatja fel a különböző szempontoknak megfelelő rekordokat, és tárhatja fel az egyedtípus rekordjai közötti összefüggéseket.

## 7.2 A LECKE TÉMAKÖREI

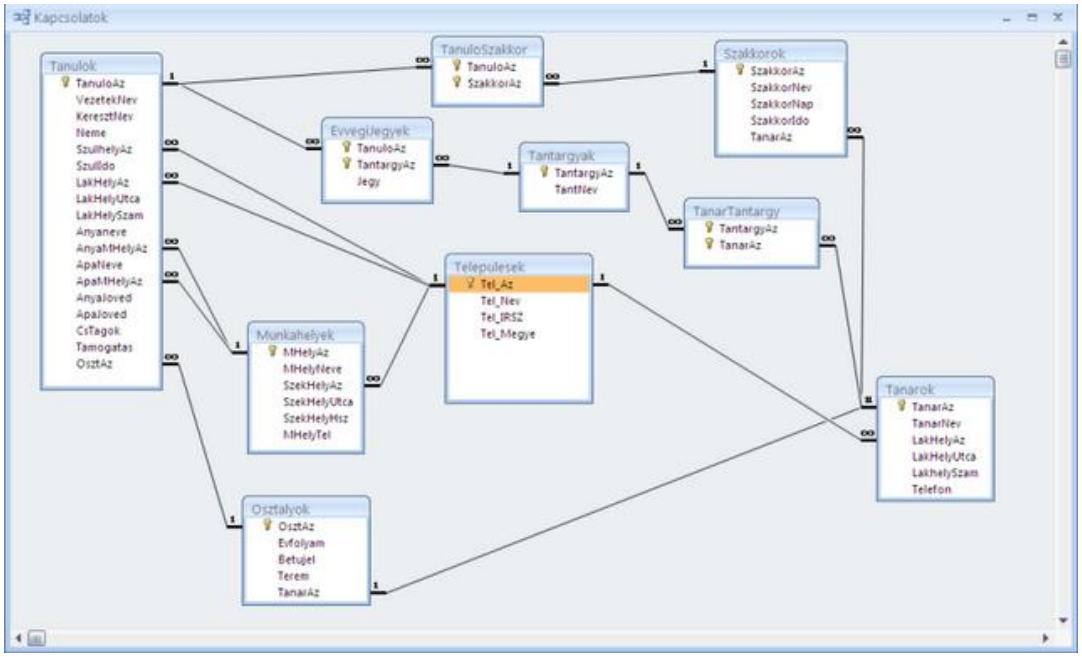
- Táblák felhasználó nézete
- Alapvető rekordkezelő műveltek
- Rekordok sorba rendezése
- Szövegek keresése és cseréje a rekordokban
- Rekordok kiválogatása szűrők segítségével

## 7.3 KIFEJTÉS

### 7.3.1 A gyakorló adatbázis szerkezete

A lecke elsajátítása során elengedhetetlen, hogy az olvasó ki is próbálja a tanultakat. Ehhez azonban jelentős számú rekorddal feltöltött táblákat tartalmazó adatbázisra lesz szüksége. Ebben és a további leckékben, egy elkészített, táblákkal, kapcsolatokkal rendelkező, rekordokkal feltöltött gyakorló adatbázist használhat, amely egy képzeletbeli iskola tanulóinak, tanárainak, osztályainak, szakköreinek adatait tartalmazza. Az adatbázist letöltheti az **1.4 Tanulási tanácsok, tudnivalók** szakaszban megadott helyről.

Mielőtt hozzákezdenénk a lecke anyagának feldolgozásához, vizsgáljuk és (a kapcsolatok ablak alapján) értsük meg a gyakorló adatbázis szerkezetét! Az adatbázis összesen 11 táblában tartalmaz rekordokat. 8 tábla valós egyedtípusokat tárol (***Tanulok, Szakkorok, Tanarok, Tantargyak, Osztalyok, Munkahelyek, Telepulesek, Megyek***), 3 tábla pedig más egyedtípusok közötti N:M kapcsolatokat ír le(***EvvegiJegyek, TanuloSzakkor, TanarTantargy***). Az adatbázis minden táblájában *Számláló* típusú azonosítók vannak, a táblák közötti kapcsolatok mindegyikében be van kapcsolva a *Hivatkozási integritás megőrzése*.




*39. ábra Az Iskola2007 adatbázis kapcsolatai*

### *Tanulok tábla*

Az iskola2007.accdb ***Tanulok*** táblájában az iskola 400 tanulójának rekordjai tárolódnak. A mezők elnevezése általában elárulja a tárolt tulajdonságtípust, ezért csak a legfontosabbakat emeljük ki.

- ***Neme***: a tanuló nemét leíró, Igen/Nem típusú mező. A *Nem* érték lány, az *Igen* fiú tanulót jelöl.
- ***SzulHelyAz***: A születési hely azonosítóját tartalmazó idegen kulcs. Az adatbázisban külön táblában (***Telepulesek***) tároljuk Magyarország településeinek nevét, és irányítószámát. Az adatbázisban egyébként minden további tábla települést leíró mezője, ehhez hasonló idegen kulcs, amely a ***Telepulesek*** tábla azonosítójának egy értékét tartalmazza.
- ***LakHelyAz***: A tanuló állandó lakóhelyét tárolja, mint a ***SzulHelyAz***.
- ***LakHelyUtca***: A tanuló állandó lakóhelyének címéből az utca nevét tárolja.
- ***LakHelySzam***: A tanuló állandó lakóhelyének címéből a házszámot tárolja.
- ***AnyaMHelyAz***, ***ApaMHelyAz***: A szülők munkahelyét leíró idegen kulcs. A szülők lehetséges munkahelyei a ***Munkahelyek*** táblában tárolódnak.
- ***ApaJoved***, ***AnyaJoved***: A szülők havi bruttó keresete. Ha a szülő nem rendelkezik jövedelemmel, a mező üres.
- ***CsTagok***: A családtagok száma a tanuló családjában.

- ***OsztAz***: Idegen kulcs, amely az mutatja meg, hogy a tanuló melyik osztályba jár. Az iskola 16 osztálya az osztályok táblában tárolódik.

### *Telepulesek*

Magyarország 2819 településének nevét, irányítószámát, és a település megyéjének azonosítóját tartalmazza. Utóbbi idegen kulcs, amelynek értékei a ***Megyek*** tábla azonosítójából valók.

### *Megyek*

A 19 megye, illetve Budapest neve, azonosítója.

### *Munkahelyek*

A szülők munkahelyeit külön táblában tároltuk, mert egy munkahelyen több szülő is dolgozhat.

- ***SzekHelyAz, SzekHelyUtca, SzekHelyHSz***: A táblában a tanuló lakóhelyéhez hasonlóan tároltuk el a munkahely székhelyének címét.
- ***MHelyTel***: A munkahely telefonszáma.

### *Osztalyok:*

A iskola 16 osztályának rekordjai.

- ***Evfolyam, Betujel***: Mindkét mező egy karakteres szövegmező. Az ***Evfolyam*** 1..8 közötti számjegyeket, a ***Betujel*** A, vagy B betűt tartalmaz.
- ***TanarAz***: Idegen kulcs, amely az osztályfőnök, ***Tanarok*** táblából származó azonosítóját tartalmazza.

### *Tanarok*

A tanárok azonosítóját, nevét, lakcímét (a tanulókéhoz hasonlóan, a település idegen kulccsal van megadva), és telefonszámát tartalmazza.

### *Szakkorok tábla*

Az iskolában tartott szakkörök azonosítóját, nevét, a szakkör napját (Hétfő..Péntek), a szakkör kezdési időpontját, illetve a szakkört tartó tanát azonosítóját tartalmazza.

### *Tantargyak*

Az iskolában tanított tantárgyak nevét, és azonosítóját tartalmazza.

### *TanuloSzakkor*

Kapcsolótábla. Azt tárolja, hogy melyik tanuló, melyik szakkörbe jár. Egy gyerek ***TanuloAz*** értéke többször is előfordulhat a táblában, más-más ***SzakkorAz***-kkel. Egy

szakkor **`SzakkorAz`** értéke is többször tárolódik a táblában, de mindig más **`TanuloAz`** értékekkel.

A táblából kiderül, hogy ki milyen szakkörbe jár, illetve, hogy melyik szakkörbe kik járnak.

### *TanarTantargy*

Kapcsolótábla, ami tanárok és tantárgyak kapcsolatát írja le. Megmutatja, melyik tanár, milyen tárgyakat tanít.

### *EvvegiJegyek*

Kapcsolótábla, amelyből kiderül, hogy melyik tanuló, milyen tantárgyakat tanult az aktuális tanévben, és hányas érdemjegyet szerzett a tárgyból.

## 7.4 TÁBLÁK FELHASZNÁLÓI NÉZETE

A táblákat tervező nézetben hozzuk létre, de a rekordok tárolását és kezelését már felhasználói nézetben végezzük. Ebben a nézetben kell megnyitni a táblát, ha rekordokat akarunk fölvenni, vagy törölni, ezt a nézetet használjuk a mezőértékek megváltoztatásakor, a rekordok közötti kereséskor, rendezéskor szűréskor.

> A táblák felhasználói nézetét az Accessben *Adattábla* nézetnek nevezik. A program felületén ezzel az elnevezéssel találkozhat.

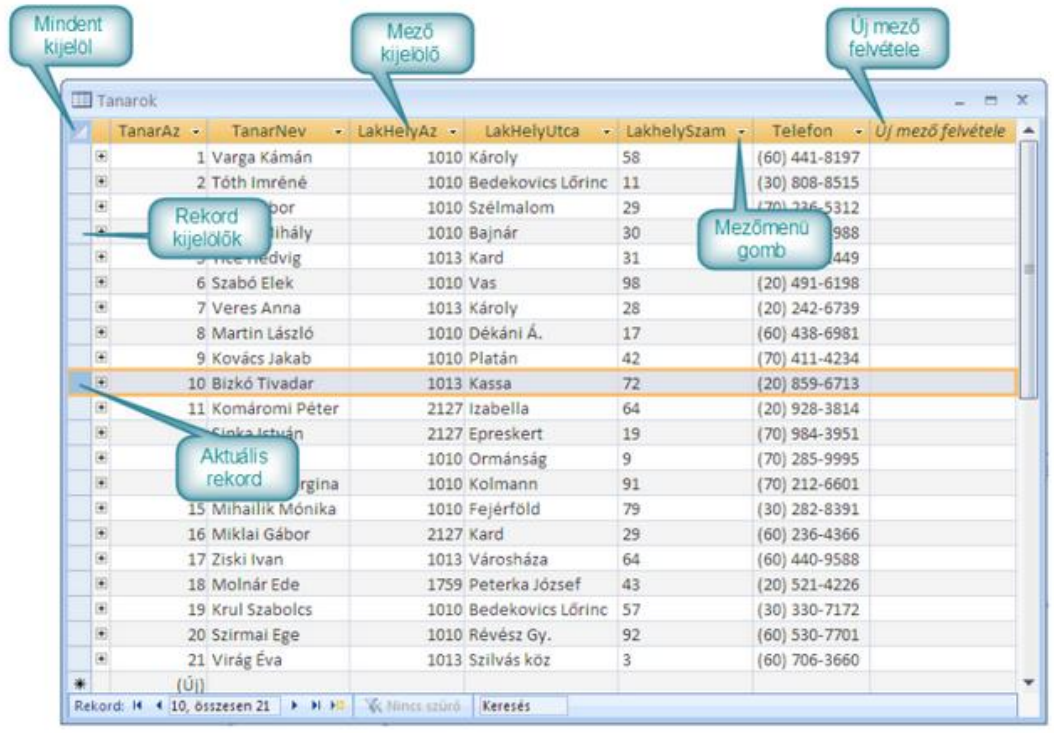
Egy tábla a **Navigációs ablak** segítségével nyitható meg Adattábla nézetben:

- A Navigációs ablakban jelenítsük meg a táblákat!
- Nézzük meg a megfelelő tábla helyi menüjét, majd
- válasszuk a **Megnyitás** parancsot!

> A tábla Adatlap nézetben történő megnyitása elvégezhető úgy is, hogy duplán kattintunk a tábla nevén.

Ilyenkor egyrészt megjelenik a tábla, különböző vezérlőelemeket tartalmazó ablaka, másrészt megváltozik a menüszalag. A menüszalagon a **Kezdőlap** parancslap, táblakezelő eszközei válnak használhatóvá, illetve jobb oldalán, a **Táblaeszközök** felirat alatt előtűnik az **Adatlap** parancslap.

A menüszalag egyes vezérlőiről később beszélünk, egyelőre koncentráljunk a tábla ablakára!




*40. ábra Táblák felhasználói nézete*

Az adatlap nézetben látható oszlopok fölött, mezőneveket tartalmazó, a mezők kijelölésére használható gombok látszanak, a sorok előtt pedig rekordkijelölő gombok helyezkednek el.

A mezőktől jobbra, az **Új mező felvétele** oszlop helyezkedik el, amelynek segítségével megváltoztatható a tábla, tervező nézetben kialakított szerkezete.

> Ebben a tananyagban nem tárgyaljuk, hogyan módosítható a tábla szerkezete az **Adatlap** nézetben. Új mezők felvételéhez használjuk a már megismert **Tervező** nézetet!

Az ablak alsó részén az úgynevezett navigációs gombok helyezkednek el. Ezek rekord kiválasztásakor, új rekord felvételekor, szűréskor illetve kereséskor használhatók.

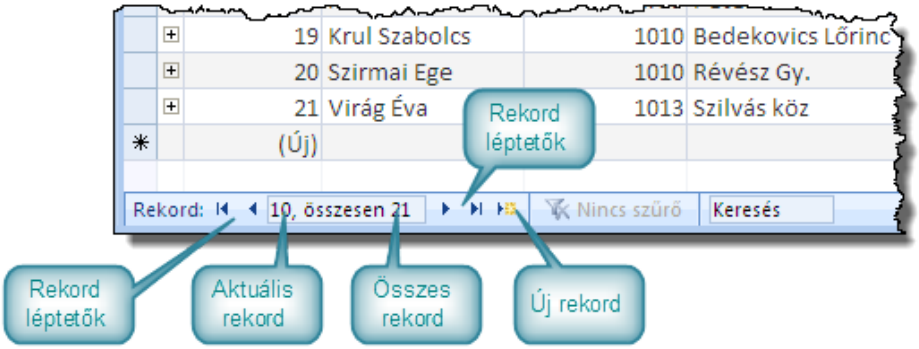
Az egyes rekordok sorainak elején pluszjellel ellátott gombok jelenhetnek meg. Azt jelzik, hogy a kapcsolatok ablak alapján a rekordhoz egy másik tábla valamelyik rekordja kapcsolódik. A gombokra kattintva meg kell adnunk a kapcsolódó mezőket, majd megnézhetjük a kapcsolódó rekordokat.

### 7.4.1 Rekordok tallózása

Bár egyszerre több rekord, és mező látszik a képernyőn, általában csak egy rekord, egyetlen mezője van kijelölt állapotban. A kijelöltet, aktuális rekordnak nevezzük. Az aktuális rekord rekordkijelölő gombját, és a kiválasztott mező mezőkijelölő gombját eltérő színnel jelzi a program. Bármelyik rekord, és mező kijelölhető egérkattintással, vagy a

billentyűzet kurzormozgató nyilaival. Az aktuális rekord kiválasztásához használhatjuk az ablak alján elhelyezkedő navigációs gombokat is.

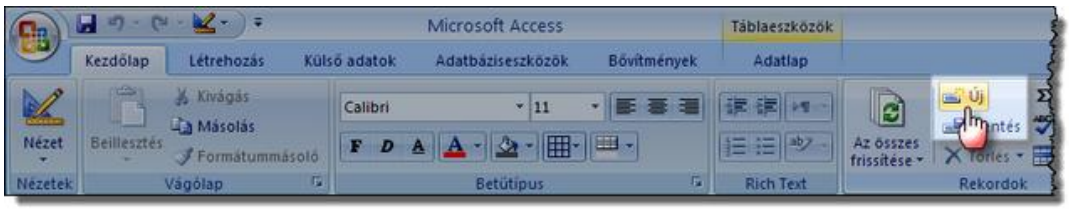
A navigációs gombok közötti felirat az aktuális rekord sorszámát, és a táblában található összes rekordok számát mutatja. Ha a számot átírjuk, az ENTER billentyű leütése után a kívánt sorszámú rekordra lépünk.




*41. ábra Rekordok tallózása*

### 7.4.2 Új rekord felvétele

Egy tábla bővítésekor a legalsó sorban vehetünk fel új rekordot. Ide egyszerű kattintással, vagy a menüszalag **Kezdőlap/Rekordok/Új gombjával**, vagy a navigációs gombok **Új (üres) rekord** gombjával juthatunk.




*42. ábra Új rekord fölvétele*

Új rekord felvételekor, a rekordkijelölő gombon megjelenő ceruza jelzi, hogy a rekord szerkesztés alatt áll. A begépelt adatok csak a szerkesztés befejezésekor kerülnek lemezre, addig csak a memóriában tárolódnak. A rekord szerkesztését a **Shift+Enter** billentyűkkel, a menüszalag **Kezdőlap/Rekordok/Mentés** gombjával, vagy egy másik rekord kiválasztásával fejezhetjük be.

Az adatok begépelése egyszerű, de mezők típusa, és tulajdonságai, valamint a tábla kapcsolatai befolyásolhatják, hogy az Access elfogadja-e a begépelt értékeket.

Ha például egy új tanár felvételekor a 10000 értéket írjuk be a ***LakHelyAz*** mezőbe, hibaüzenet kapunk, mert a ***Tanarok-Telepulesek*** tábla kapcsolatában bekapcsolt hivatkozási integritás megőrzése előírja, hogy a ***LakHelyAz*** mezőben helyes idegen kulcs értéknek, egy a ***Telepulesek*** táblában tárolt település azonosítójának kell lennie. 10000-es

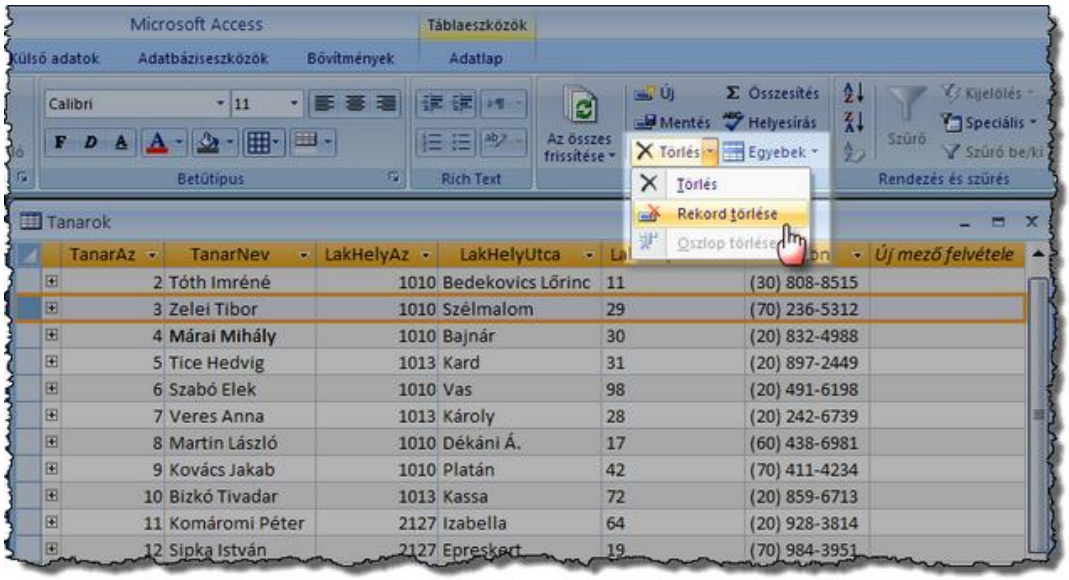
azonosítójú település pedig nincs a ***Telepulesek*** táblában. Ha a mezőbe az „Eger” szöveget írjuk szintén hibaüzenet jelenik meg, hiszen a ***LakHelyAz*** szám típusú mező.

### 7.4.3 Rekord módosítása

A rekordok módosításakor ki kell jelölnünk a megfelelő rekordot és mezőt, majd be kell gépelni az új értéket. A szerkesztett rekord mentése, és az új adat elfogadhatóságának értékelése **7.4.2 Új rekord felvétele** szakaszban leírtaknak történik.

### 7.4.4 Rekordok törlése

A fölöslegessé vált rekordok, kijelölés után a **Kezdőlap/Rekordok/Törlés** paranccsal, vagy a **Delete** billentyűvel törölhetők. Mindkét eszközzel érdemes vigyáznunk, mert ha elfelejtettük kijelölni a rekord sorát, akkor nem a rekordot, hanem a kiválasztott mező tartalmát fogjuk eltűntetni.




*43. ábra Rekord törlése*

A véletlenül törölt mezőtartalom az **Esc** billentyűvel visszaállítható.

Egyszerre több, egymást követő rekordot is törölhetünk. Ehhez csak annyit kell tenni, hogy kijelöléskor lenyomott bal gombbal végighúzzuk az egeret a törlendő rekordok kijelölő gombjain.

A rekordok végletörlése előtt meg kell erősíteni a szándékunkat.

> Ha a törlésekor megjelenő, megerősítő kérdésre **Igen** választ adunk, a kijelölt rekordok véglegesen törlődnek. A törlés nem vonható vissza!

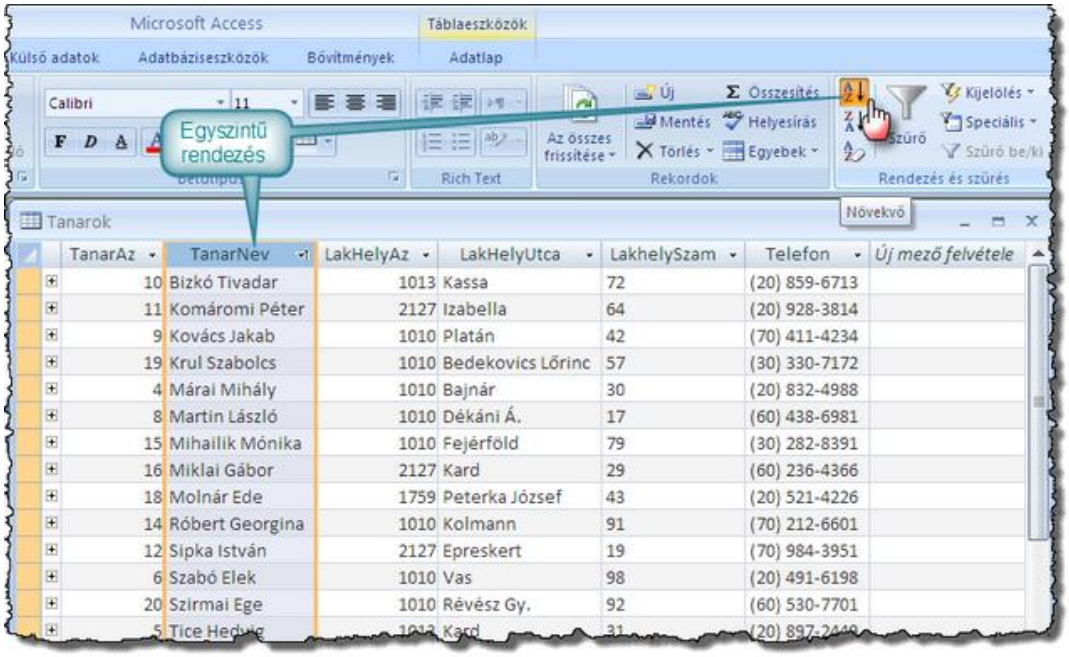
### 7.4.5 Rekordok sorba rendezése

Amikor másképp nem rendelkezünk, az Access az elsődleges kulcs alapján sorba rendezve jeleníti meg a rekordokat. A rendezéssel ezt a sorrendet változtathatjuk meg.

A rendezést elvégezhetjük egy, vagy több mező értékeinek figyelembevételével. Utóbbi esetben többszintű rendezésről beszélünk.

#### *Egyszintű rendezés*

Az egyszintű rendezés előtt a rendezés alapját adó mezőt ki kell jelölnünk. Ez történhet a mezőkijelölő gombbal, vagy úgy, hogy bármelyik rekordban rákattintunk a mezőre. A rekordokat növekvő, vagy csökkenő sorrendbe rendezhetjük a menüszalag **Kezdőlap/ Rendezés és szűrés /Növekvő** illetve **Csökkenő** gombjaival.




| TanarAz | TanarNev | LakHelyAz | LakHelyUtca | LakhelySzam | Telefon | Új mező felvétele |
|---|---|---|---|---|---|---|
| 10 | Bizkó Tivadar | 1013 | Kassa | 72 | (20) 859-6713 | |
| 11 | Komáromi Péter | 2127 | Izabella | 64 | (20) 928-3814 | |
| 9 | Kovács Jakab | 1010 | Platán | 42 | (70) 411-4234 | |
| 19 | Krul Szabolcs | 1010 | Bedekovics Lőrinc | 57 | (30) 330-7172 | |
| 4 | Márai Mihály | 1010 | Bajnár | 30 | (20) 832-4988 | |
| 8 | Martin László | 1010 | Dékáni Á. | 17 | (60) 438-6981 | |
| 15 | Mihailik Mónika | 1010 | Fejérföld | 79 | (30) 282-8391 | |
| 16 | Miklai Gábor | 2127 | Kard | 29 | (60) 236-4366 | |
| 18 | Molnár Ede | 1759 | Peterka József | 43 | (20) 521-4226 | |
| 14 | Róbert Georgina | 1010 | Kolmann | 91 | (70) 212-6601 | |
| 12 | Sipka István | 2127 | Epreskert | 19 | (70) 984-3951 | |
| 6 | Szabó Elek | 1010 | Vas | 98 | (20) 491-6198 | |
| 20 | Szirmai Ege | 1010 | Révész Gy. | 92 | (60) 530-7701 | |
| 5 | Tice Hedvig | 1013 | Kard | 31 | (20) 897-2449 | |

*44. ábra Rendezés egy mező szerint*

Ha például a ***Tanarok*** táblában a ***TanarNev*** mező kijelölése után a **Növekvő** gombra kattintunk, a tanárok rekordjai név szerint növekvő sorrendben, névsorban jelennek meg.

#### *Többszintű rendezés*

Többszintű rendezés esetén az alacsonyabb rendezési szint felől a magasabb felé kell haladni a rendezéssel.

Figyeljük meg, hogy a ***Tanulok*** táblában tároljuk a gyerekek vezeték- és keresztnevét is. Ha névsort akarunk felállítani, akkor előbb vezetéknevek, ezen belül pedig keresztnevek alapján kell elvégeznünk a rendezést. A vezetéknév a rendezés 1., maga-

sabb, a keresztnév a 2., alacsonyabb szintje. Előbb tehát ***KeresztNev***, majd pedig ***VezetekNev*** mezők alapján kell elvégeznünk a rendezést.

### *Rendezés mentése*

Amikor egy rendezés után bezárunk egy táblát a „terv" mentésére célzó kérdést kapunk. Igenlő válasz esetén a legközelebbi megnyitáskor az utolsóként beállított sorrendben jelennek meg a rekordok. Ellenkező esetben az eredeti rekordsorrendet kapjuk.

### *Rendezés törlése*

Az eredeti sorrend visszaállításához, azaz a rendezés törléséhez a menüszalag **Kezdőlap/Rendezés és szűrés/ Minden rendezés törlése** parancsát kell választanunk.

## 7.5 REKORDOK FELKUTATÁSA

Ha a táblában szeretnénk hamar megtalálni egy bizonyos szöveget tartalmazó rekordot, akkor a **Keresés** eszközt célszerű használnunk. Kereséskor egy megadott szöveget tartalmazó rekordot kutathatunk fel. A szöveg keresése történhet a tábla egyik kijelölt mezőjében, vagy az összes mezőben egyszerre.

Ha csak egy mező alapján keresünk, akkor a keresés elindítása előtt a mezőt ki kell jelölni, ha az összes mezőben végezzük a műveletet, akkor a kijelölésnek nincsen jelentősége. A keresés, a menüszalag **Kezdőlap/Keresés/Keresés** gombjával, vagy a **Ctrl+F** billentyűkkel indítható el. A keresett szöveget, és a keresés beállításait a **Keresés és Csere ablakban** adhatjuk meg. A **Keresett szöveg** szövegdobozt mindenképpen ki kell töltenünk. Itt gépeljük be azt a szöveget, amit keresünk. A keresést a **Következő** gombbal indítjuk el.

### 7.5.1 A keresés eredménye

A keresés eredménye mindig egyetlen rekord, amely a keresés után aktív rekorddá válik. Találat esetén a keresés tovább folytatható a **Következő** gombbal. Az eredmény mindig a keresett szöveget tartalmazó következő rekord lesz. Ha nincs találat, az Access üzenetablakban tájékoztatja a felhasználót.

### 7.5.2 Egyezés

Kereséskor az Access a keresett szöveget összehasonlítja az éppen vizsgált mező egy részével. Az **Egyezés** listában megadhatjuk, hogy melyik mezőrészlettel történjen az összehasonlítás.

- **Teljes mező:** akkor kapunk találatot, ha a keresett szöveg a mező teljes egészével egyezik
- **Mező elején:** akkor lesz találat, ha a mező a keresett szöveggel kezdődik.
- **Mező bármely részében:** akkor lesz találat, ha a vizsgált szöveg bárhol előfordul a mező tartalmában.

### 7.5.3 Keresés

Itt adhatjuk meg, hogy az aktuális rekordhoz képest milyen irányban haladjon a keresés.

- Fel: a tábla eleje felé
- Le: a tábla vége felé
- Mind: az aktuális rekordtól függetlenül, tábla elejétől a vége felé

### 7.5.4 Kis- és nagybetűk, formázott adat

A **Kis-és nagybetű különbözik** kiválasztó négyzettel jelezhetjük, különbséget karunk-e tenni a kicsi, és a nagybetűk között. **A Formázott adat keresése** kiválasztó négyzetet bejelölve a keresett szöveget, a megjelenítési szöveggel, különben pedig a tárolt adattal hasonlítja össze az Access. Ennek akkor van jelentősége, ha a mező formátum tulajdonságával megváltoztattuk a tárolt adat megjelenítését, például egy dátum mezőben kiíratjuk a hónapok nevét.

### 7.5.5 Helyettesítő, vagy dzsóker karakterek

Az **Egyezés** legördülő listánál már láthattuk, hogy hogyan lehet viszonylag egyszerűen rávenni az Access-t arra, hogy a mező különböző részeit vizsgálva keressen. Az adatbázis-kezelő azonban ennél jóval finomabb lehetőségeket is kínál a keresett adat megadására. A **Keresett szöveg** szövegdobozban úgynevezett helyettesítő karaktereket is elhelyezhetünk. A helyettesítő karakterek az jelzik a keresendő szövegben, hogy a helyükön más egyéb jelek állhatnak.

A dzsóker karakterekkel rugalmasan szabályozhatjuk egy keresési minta jelentését. A jelek külön-külön és együtt is alkalmazhatók, így sokféle minta létrehozására alkalmasak.

Az alábbi táblázat példái a ***Tanulok*** tábla ***ApaNeve*** mezője alapján történő keresésekre vonatkoznak

| Karakter | Leírás | Keresett szöveg | Keresési eredmény |
|---|---|---|---|
| * | Az egyik legjellemzőbb helyettesítő karakter. Helyén bármilyen, és bármennyi jel állhat. | * László | Az első olyan rekord, ahol a vizsgált mezőben tetszőleges karaktereket egy szóköz, majd a *László* szöveg követi. |
| ? | A *-hoz hasonlóan bármilyen karakter helyettesíthet, de egy **?** csak egy karaktert. | ????? László | Megtaláljuk például *Fehér László,* vagy *Andai László* rekordját, de *Lakatos László*-ét nem. |
| # | Egyetlen számjegyet helyettesítő karakter. | *# | Megtalálja *Pelle Marcell 2*-t, de nem találja meg *Pelle Marcell*-t. |
| [ ] | Egyetlen, karakterosztállyal megadott karaktert helyettesít. Az osz- | c[sz]* | Megtalálja *Czipó Menyhért*-et, *Csillik Menyhért*-et, de nem találja meg *Cifra Hubát.* |

| | | | |
|---|---|---|---|
| | tályt alkotó karaktereket egymás mellett soroljuk fel, vagy kötőjellel, intervallumként adjuk meg. | g[a-e]* | Megtaláljuk *Gerle Izsó*-t, *Gazsi András*-t, de nem találjuk meg *Gregor Ábrahám*-ot. |
| ! | A tagadás jele, amit csak karakterosztály elején használhatunk. Az osztály tagadására való. | g[!a-e]* | Az adat nem az *a-e* betűcsoportban van.<br><br>Nem találjuk *Gerle Izsó*-t, *Gazsi András*-t, de találjuk *Gregor Ábrahám*-ot. |

## 7.6 REKORDOK KIVÁLOGATÁSA

Bár a dzsóker karakterek használatával a keresés érdekessé válik, inkább szórakoztató, mint hasznos művelet. A gyakorlatban ritka az a feladat, amikor egyetlen, rekordot keresünk. Sokkal jellemzőbb az, hogy egy feltételnek megfelelő összes rekordot kigyűjtve, együtt kell azokat megjelenítenünk. Erre ugyan a keresés nem alkalmas, az Access rendelkezik a megfelelő eszközökkel, az úgynevezett szűrőkkel.

Ahhoz, hogy egy táblában szűrjünk, azaz rekordokat válogassunk ki, a következőket kell tennünk:

- Meg kell adnunk a kiválogatandó rekordok közös jellemzőjét, a szűrés feltételét, más néven a szűrő tartalmát!
- Be kell kapcsolni a szűrőt!
- Ezt követően, eredményként megjelennek azok a rekordok, amelyek megfelelnek a megadott feltételnek. Ezt a rekordhalmazt nevezzük a szűrés **eredményhalmazának**.

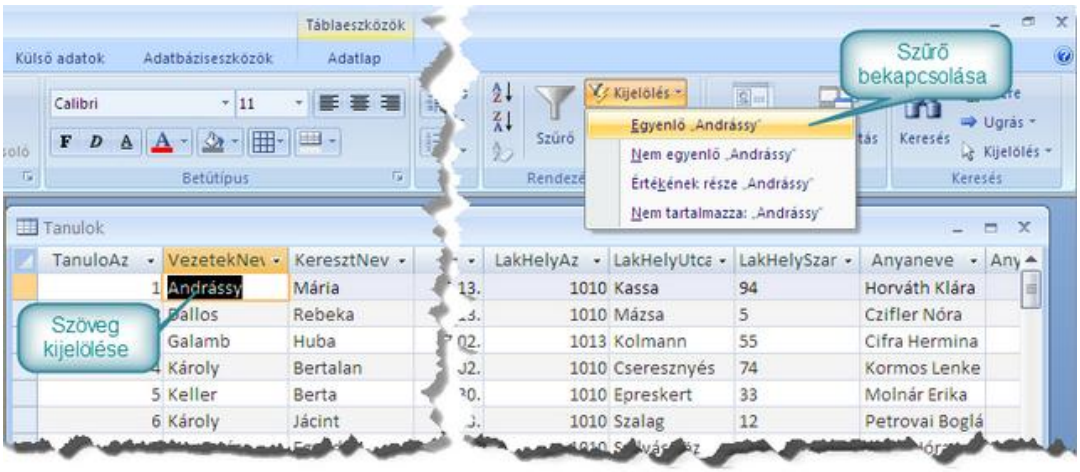
A szűrőben teljesen egyszerű, de egészen összetett feltételeket is megadhatunk. Az Access készítői számos lehetőséget építettek be a szűrőfeltételek megadására. Ennek megfelelően, bár a háttérben mindig ugyanaz történik, a szűrést többféleképpen is elvégezhetjük. Tananyagunkban a **kijelölés** alapján, a **beírással** történő, illetve az **irányított** szűrőt ismertetjük.

### 7.6.1 Szűrés kijelölés alapján

A kijelöléses szűrő kezelése talán a legegyszerűbb.

- Elsőként kiválasztjuk a mezőt, ami alapján a rekordokat szűrni szeretnénk.
- Keresünk egy rekordot, aminek ebben a mezőjében, megtalálható a feltételként használt szövegrészlet.
- Kijelöljük a szöveget, majd **Kezdőlap/Rendezés és szűrés/Kijelölés** gombra kattintunk.
- A megjelenő menüben pontosítjuk, hogy hol kell elhelyezkednie a kijelölt szövegnek, a kiválasztott mezőben.

– Ezt követően a szűrő bekapcsolódik, és a képernyőn megjelenik az eredményhalmaz.




*45. ábra Szűrés kijelöléssel*

### 7.6.2 Eredményhalmaz kezelése, szűrő kikapcsolása, törlése

A szűrő bekapcsolása után megjelenik az eredményhalmaz. Ezt a **Kezdőlap/Rendezés és szűrés/Szűrő be/ki** gomb, és a rekordléptető gomboktól jobbra található **Szűrt** gomb bekapcsolt állapota jelzi. A navigációs gombok közötti érték ilyenkor, az eredményhalmazban található, szűrt rekordok számát mutatja.

Az eredményhalmazban újabb szűrést is végezhetünk, azaz további mezőre adhatunk meg feltételt. A szűrőt **Kezdőlap/Rendezés és szűrés/Szűrő be/ki** gombbal, vagy a **Szűrt** gombbal kapcsolhatjuk ki. Ilyenkor újra megjelenik a tábla összes rekordja.

Amikor a szűrőt kikapcsoljuk, az Access „megjegyzi" a legutoljára beállított feltételeket, így a szűrést visszakapcsolhatjuk anélkül, hogy a szűrő tartalmát újra megadnánk.

### 7.6.3 Szűrés beírás alapján

Ha nem szeretnénk kijelölni a feltétel szövegét, használhatjuk a beírás alapján történő szűrést. Ilyenkor a megfelelő oszlop tetején, a mezőkijelölő gombon látható háromszögre kattintva jelenik meg a szűrésre alkalmas parancs, amely a mező típusától függően **Szövegszűrők**, **Számszűrők**, vagy **Dátumszűrők** lehet. A feltételt mindig a megjelenő almenü parancsaival pontosíthatjuk.

### 7.6.4 Irányított szűrés, rendezés

Ez eddig tárgyalt eszközökkel egyszerre csak egy mezőt szűrhettünk és a feltételt is csak korlátozottan szabályozhattuk. Az irányított szűrő jóval összetettebb feladatok megoldására alkalmas. Több mező bevonásával egészen összetett feltételeket adhatunk meg, egyszerre szűrhetünk és rendezhetünk, sőt olyan értékeket is felhasználhatunk a rekordok kiválogatásában, amelyek közvetlenül nem is tárolódnak az adatbázisban.

A kifinomult feltétel-szabályozásnak természetesen van némi ára. Meg kell tanulnunk egy olyan jelölésrendszer használatát, amellyel a képernyőn éppen nem látható adatra is megfogalmazható a kritérium.

### *Logikai kifejezések*

A logikai kifejezések olyan speciális jelsorozatok, amelyek segítségével feltételeket írhatunk le.

A számítástechnikában gyakran **kifejezésekkel** adjuk meg a gép által elvégezhető műveletet. A kifejezés tényezőkből és műveletei jelekből áll. A tényezőkkel végrehajtandó műveletet a műveleti jel adja meg. A kifejezések műveleti jeleit **operátoroknak**, a tényezőket **operandusoknak** is nevezik. A számítógép képes a kifejezésben leírt művelet elvégezésére és eredmény meghatározására.

**A kifejezés operandusok és operátorok sorozatával leírt, a számítógép által elvégezhető művelet. A művelet elvégzésével a számítógép képes annak eredményét kiszámítani.**

A **2 + 4** matematikai kifejezés, amelyben a 2 és a 4 az operandusok, és a + jel az operátor, a műveleti jel. A kifejezés eredménye **6.**

A **„tulajdonság" & „típus"** szöveges kifejezés. Operandusok, az idézőjelek között megadott szövegek. Az **&** operátor a szöveg összevonás jele. Hatására a számítógép összeilleszti a két operandust. A **„tulajdonság" & „típus"** kifejezés eredménye a **„tulajdonságtípus"** szó.

A logikai kifejezések olyan speciális kifejezések, amelyekkel **állítások** fogalmazhatók meg. Amikor a számítógép kiértékel egy logikai kifejezést, eredményként kétféle értéket kaphat. Az állítás lehet **igaz**, vagy **hamis.** A logikai kifejezések tehát igaz, vagy hamis eredményűek lehetnek. Műveleti jeleiket **összehasonlító operátoroknak** hívják.

**3 > 4** logikai kifejezés, amelyben a **3**, és a **4** az operandusok, a '>' jel az operátor, A kifejezés hatására a számítógép összehasonlítja a két tényezőt és eldönti, hogy az első valóban nagyobb-e, mint a második, azaz meghatározza, hogy az állítás igaz, vagy hamis.

Ebben az esetben a kifejezés eredménye hamis, hiszen 3 nem nagyobb, mint 4.

### *Összetett logikai kifejezés*

Eddig egyszerű logikai kifejezésekkel találkoztunk, mint például:

4>2 vagy 3=5

Az összetett logikai kifejezés több egyszerű logikai kifejezésből áll, de eredménye szintén igaz, vagy hamis. Két egyszerű logikai kifejezést úgynevezett **logikai operátorokkal** kapcsolhatunk össze, összetett logikai kifejezéssé. Ilyen logikai operátor az **AND** (és), illetve az **OR** (vagy).

**AND** operátorral kialakított összetett logikai kifejezés eredménye csak akkor **igaz**, ha mindkét egyszerű logikai kifejezés **igaz.**

**OR** operátor esetén, az összetett logikai kifejezés **igaz**, ha a benne lévő bármelyik egyszerű logikai kifejezés **igaz.**

| Példa | Magyarázat |
|---|---|
| 2=2 AND 4=2*2 | Igaz, mert mindkét egyszerű logikai kifejezés igaz. |
| 3>2 AND 3<1 | Hamis, mert a második egyszerű logikai kifejezés igaz. |
| 3=5 OR 4<5 | Igaz, mert a második logikai kifejezés igaz. |
| 3>4 OR 4<2 | Hamis, mert mindkét egyszerű logikai kifejezés hamis. |

### Mező logikai kifejezésekben

A **3>4** logikai kifejezésben két ismert értékkel, úgynevezett állandókkal fogalmaztuk meg az állítást. A szűrők feltételeit szabályozó logikai kifejezések egyik operandusa azonban mindig egy mező. Az ilyen logikai kifejezéseket úgy kell leírni, hogy első operandusként a megfelelő mező, szögletes zárójelek közé írt nevét adjuk meg.

A **[CsTagok]=6** logikai kifejezés akkor igaz, ha a vizsgált rekordban ***CsTagok*** mező értéke egyenlő 6-tal.

Az irányított szűrőben ehhez hasonlóan lehet majd megadni a feltételeket. A szűrő bekapcsolása után az Access rendre megvizsgálja a tábla rekordjaiban a ***CsTagok*** mező értékét, és csak azokat a rekordokat jeleníti meg az eredményhalmazban, amelyekben a kifejezés eredménye igaz.

Természetesen az összetett logikai kifejezések is tartalmazhatnak mezőneveket.

**[KeresztNev] = ”László” AND [SzulIdo] Like „2001*”**

Az így megadott feltétel akkor igaz, ha a ***KeresztNev*** mező tartalma **László**, ÉS a ***SzulIdo*** mező a **2001** karakterekkel kezdődik.

### Összehasonlító operátorok

A logikai kifejezésekben, többek között, a matematikában megismert relációs jeleket használjuk összehasonlító operátorként.

- > nagyobb mint,
- < kisebb mint,
- = egyenlő
- >= nagyobb, vagy egyenlő
- <= kisebb, vagy egyenlő
- <> nem egyenlő

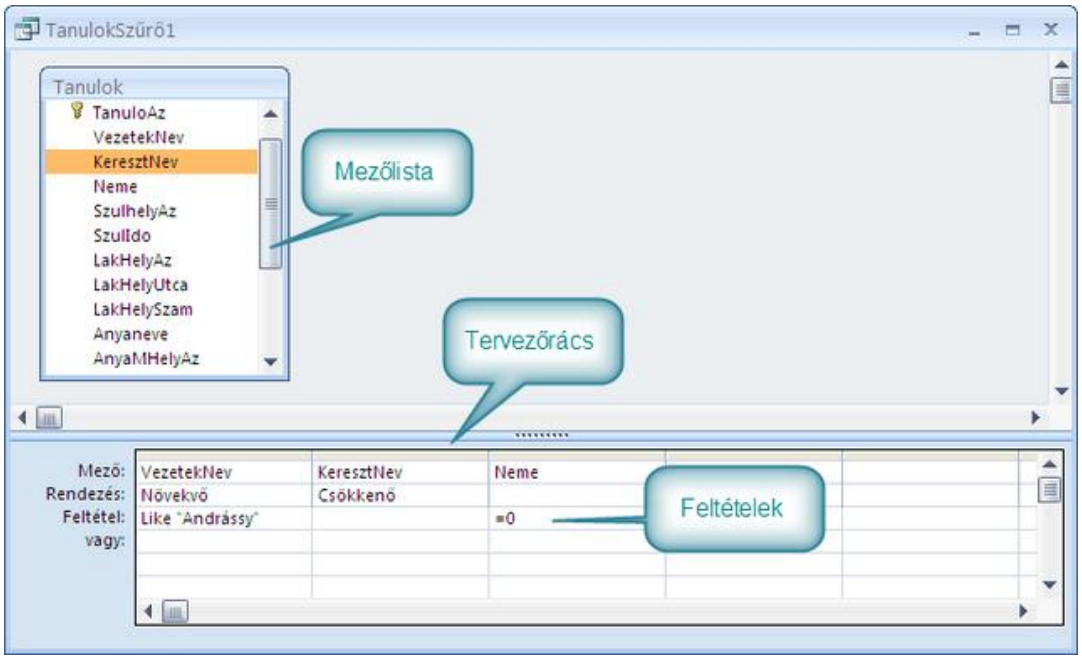
A matematikai relációs jelek mellett, az adatbázis-kezelésben további operátorokat is használhatunk:

| Operátor | Jelentés | Példa | Magyarázat |
|---|---|---|---|
| [*mező*] **Like** *szöveg* | Hasonló | [VezetekNev] Like „b*”<br><br>[SzulIdo] Like „1986*” | A like operátor segítségével a mező egy szöveggel hasonlítható össze. A szövegben használhatjuk a keresésnél tanult helyettesítő karaktereket is.<br>A Like-kal szűrve azokat a rekordokat kapjuk meg, amelyek kiválasztott mezője olyan, mint a megadott szöveg. |
| [*mező*] **Between** … **and** … | Között | [CsTagok] Between 4 and 6 | Egy intervallum két megadott végpontja között lévő értékek válogathatók ki. Itt például azok a rekordok, ahol tanuló családjában 4 és 6 közötti a családtagok száma. A két végpont része az intervallumnak. |
| [*mező*] In (*halmaz*) | Eleme | [OsztAz] In (2;4;6) | Az In jobb oldalán kerek zárójelek között, felsorolással megadott halmaz szerepel. Az elemeket pontosvesszővel választjuk el.<br><br>Az In operátorral azt vizsgálhatjuk meg, hogy a mezőértéke eleme-e a felsorolással megadott halmaznak.<br><br>A példában akkor kapunk vissza rekordokat, ha azok kiválasztott mezőjének értéke 2, 4, vagy 6. |

| | | | |
|---|---|---|---|
| [*mező*] Is Null | Üres | [AnyaJoved] Is Null | Az **Is** speciális operátor. Jobb oldalán csak a „null” érték állhat. A **null** a „semmit”, az üres mező-értéket jelzi. Az **Is Null** logikai kifejezés-sel olyan rekordokat szűrhetünk, amelyekben a kiválasztott mező üres. |

### *Az irányított szűrő szerkesztése*

Az irányított szűrő biztosítja az Accessben a legösszetettebb szűrési feltételek megadását. A feltételeket itt is különálló ablakban adhatjuk meg, amely a menüszalagon, a **Kezdőlap/Rendezés és szűrés/Speciális/Irányított szűrés rendezés...** paranccsal jeleníthető meg.




*46. ábra Irányított szűrő ablaka*

A szűrő ablaka egymás alatt elhelyezkedő területekből, a mezőlista ablakrészből és a tervezőrácsból épül fel. A mezőlista ablakrészben a tábla (Kapcsolatok szerkesztésekor megismert) mezőlistája található, amelyben megjelenik a tábla összes mezőjének neve. Innen kell a tervezőrács oszlopaiba tennünk azokat a mezőket, amelyekre valamilyen feltételt szeretnénk beállítani.

### Mezők elhelyezése a tervezőrácson

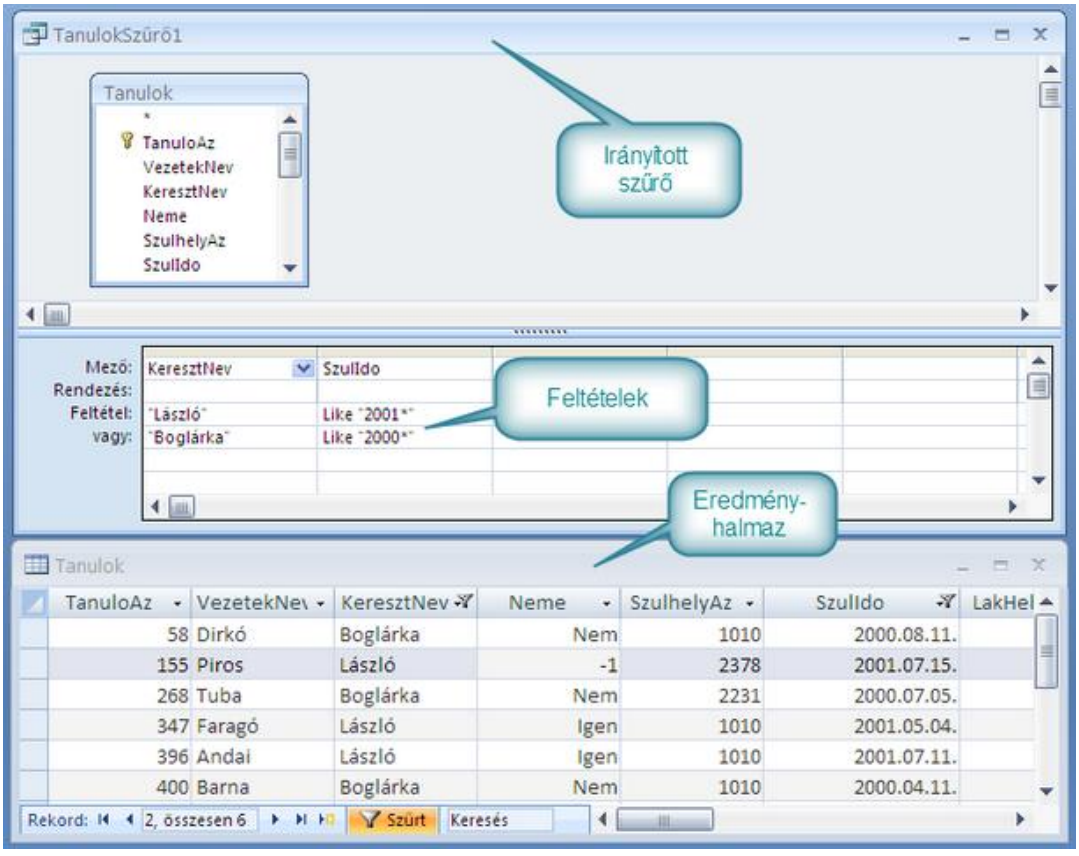
A mezőket a mezőlistáról a tervezőrácsra tehetjük dupla kattintással, vagy húzással. Kiválaszthatunk mezőt a rács, **Mező** sorának legördülő listájából, de ebben a sorban akár kézzel is begépelhetjük a mezőnevet.

A rács minden oszlopa fölött egy vékony oszlopkijelölő gomb található, amelyet az egérrel megragadva jobbra-balra elmozdíthatjuk az oszlopot. A kijelölt oszlopot a **Delete** billentyűvel törölhetjük, az **Insert** billentyűvel pedig új, üres oszlopot illeszthetünk elé.

A menüszalag **Kezdőlap/Rendezés és szűrés/Speciális/Rács törlése** parancsa mindent töröl a tervezőrácsról.

### Feltételek megadása

Az irányított szűrőben a tervezőrácson elhelyezett mezők oszlopaiban, a **Feltétel**, illetve a **Vagy** sorokban adhatjuk meg a logikai kifejezéseket. A kifejezések mindig az oszlopban feltűntetett mezőre fognak vonatkozni, így első operandusukat nem kell begépelni.




*47. ábra Feltételek az irányított szűrőben*

Ha a rácson több mező alatt is elhelyezünk feltételt, akkor az azonos sorban lévők között **AND** (és), a különböző sorok között pedig **OR** (vagy) kapcsolat lesz. Ezzel a módszerrel egyszerűen adhatunk meg több mezőre vonatkozó, összetett logikai kifejezéseket.

### Számított értékek

A szűrők használata közben előfordulhat, hogy olyan tulajdonságra akarunk feltételt megadni, amely közvetlenül nem tárolódik a táblában, de más mezőkből kiszámolható.

Például kíváncsiak lehetünk arra, hogy kik azok a tanulók, akiknek szülei együtt sem keresnek 200 000 Ft-ot. Az összesített kereset egyik mezőben sem tárolódik, de a két szülő jövedelme külön-külön rendelkezésre áll, az ***ApaJoved*** és az ***AnyaJoved*** mezőkben.

Az olyan értéket, mint a szülők összesített jövedelme, **számítható értékeknek** nevezzük.

Az adatbázis-kezelésben szabály, hogy számítható értéket külön mezőben nem tárolunk, mert az az adatok inkonzisztenciájához vezethetne.

Ha van [ApaJoved] illetve [AnyaJoved] mezőnk, akkor a kettő összegének tárolására nem készítünk a táblában pl. [OsszesKereset] mezőt, mert ha megváltozna valamelyik szülő jövedelme, akkor nem csak az egyes jövedelmet, hanem az összesített keresetet tartalmazó mezőt is meg kellene változtatnunk. Ez egyrészt fölösleges munka, másrészt, ha elmulasztjuk, akkor hibás érték marad az [OsszesKereset] mezőben.

Számítható értéket nem tárolunk az adatbázisban, de bizonyos műveletekben (pl. szűréskor) mégiscsak szükség lehet rájuk. Az Accessben az úgynevezett **számított mezőkkel** oldották meg a problémát. A számított mező ideiglenesen létrehozott mező, amely számított értéket tartalmaz. A számított mező nem tárolódik, csak addig létezik, amíg egy számítható értéket használunk. Az Access irányított szűrőjének tervezőrácsán létrehozhatunk számított mezőket, amelyeket azután feltételek megadására használhatunk.

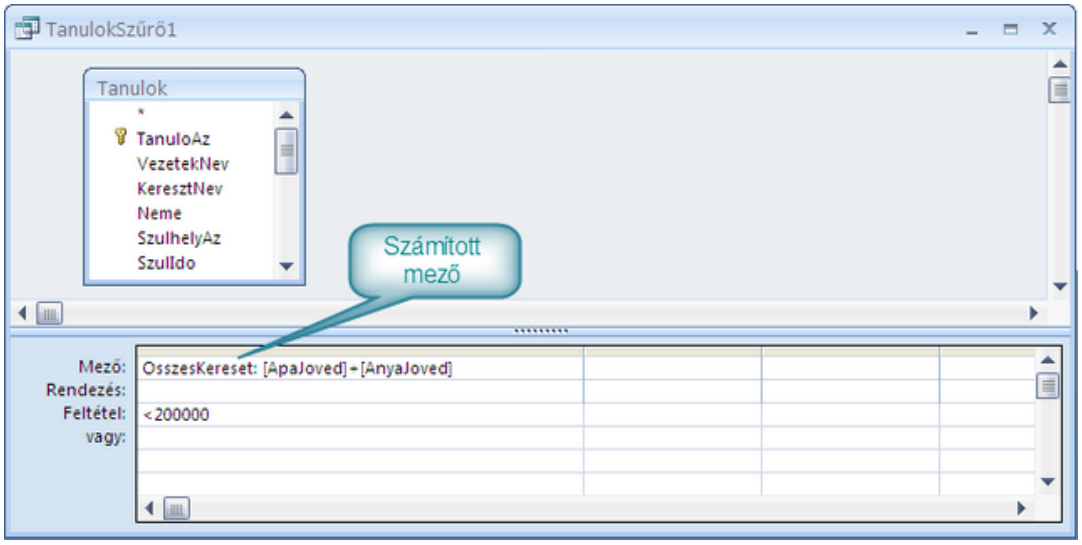
Számított mező létrehozásakor a tervezőrács egy üres oszlopának **Mező:** sorába egy általunk kitalált mezőnevet kell begépelni. Ez lesz a számított mező neve. A név nem egyezhet meg a tábla egyetlen valódi mezőjének nevével sem, és meg kell felelnie az Access névadási szabályainak. A név után kettőspontot kell tenni, majd be kell gépelni a mező értékét kiszámító kifejezést. A kifejezésben felhasználhatunk konstans értékeket, de más mezőkre is hivatkozhatunk.

A szülők összesített keresetét tartalmazó számított mező például így néz ki:

OsszesKereset: [ApaJoved] + [AnyaJoved]

Az így létrehozott számított mező alatt megadhatjuk a feltételt. Pl.:

<200000




*48. ábra Számított mező*

**Rendezés**

Az Access irányított szűrője a rekordok kiválogatásán kívül, azok sorba rendezésére is képes. A tervezőrács **Rendezés:** sorában a rácson elhelyezett bármely mező alatt beállíthatunk növekvő, vagy csökkenő irányú rendezést.

Ha több mezőt is felhasználunk a rendezésre, akkor az Access többszintű rendezést végez. Ilyenkor a rács első, rendezésre használt mezője szerint rendezi sorba a rekordokat. Ez lesz az első rendezési szint. Ha ezen a szinten egyes rekordokban azonos érték van, akkor a következő, második szint szerint folytatódik a rendezés. A rendezés tetszőleges szintet tartalmazhat, szintjeit a tervezőrácson elfoglalt hely határozza meg.

Ha a szinteket meg akarjuk változtatni, akkor az érintett mezőket egyszerűen el kell mozgatni a rácson.

## 7.7 ÖSSZEFOGLALÁS

Egy elkészített adatbázis tartalma, az első használatba vételtől kezdve dinamikusan változik. Tábláiban rekordokat helyezünk el, módosítjuk a tárolt értékeket, eltávolítjuk az időközben fölöslegessé vált rekordokat, törekszünk a pillanatnyi valóság lehető legpontosabb leképezésére. A tárolt adatokból információt próbálunk szerezni. A megfelelő adatok megjelenítéséhez, felkutatásához, kereséshez, rendezéshez, kiválogatáshoz gyakran kell az adatbázis-kezelő rendszer különböző eszközeit használnunk. Az Access, a táblák felhasználói (Adatlap) nézetében számos olyan eszközt biztosít, amelyek segítségével a rekordkezelő műveletek egyszerűen elvégezhetők.

Az Adatlap nézetben megnyitott tábla rekordjai között egyszerűen tallózhatunk a navigációs gombokkal, a kijelölt rekordot módosíthatjuk, szükség esetén törölhetjük. A menüszalag **Rekordok**, **Rendezés és szűrés**, valamint **Keresés** csoportjaiban lévő eszközökkel könnyedén változtathatjuk meg a rekordok sorrendjét, felkutathatunk keresett

szövegeket tartalmazó rekordokat, pontosan megadott feltételek szerint kiválogatott eredményhalmazokat készíthetünk.

## 7.8 ÖNELLENŐRZŐ KÉRDÉSEK

1. Honnan tudható meg egy tábla rekordjainak száma?
2. Mikor kerül lemezre egy új, vagy egy megváltoztatott rekord tartalma?
3. Lehet-e egyszerre több rekordot törölni?
4. Hogyan végezhet többszintű rendezést?
5. Rendezés után hogyan kaphatja vissza az eredeti rekordsorrendet?
6. Mire való a keresés?
7. Milyen értéket jelent kereséskor a ####.05* szöveg?
8. Hogyan keresne, ha teljes neveket tartalmazó mezőben szeretné megtalálni azt a nevet, amelyikben a keresztnév ly-t tartalmaz (pl. Ibolya)?
9. Mi az a logikai kifejezés?
10. Hogyan adhatja meg az irányított szűrőben a feltételeket?

# 8. LEKÉRDEZÉSEK KÉSZÍTÉSE

## 8.1 CÉLKITŰZÉS

A következő oldalakon a Microsoft Access adatbázisok egyik legérdekesebb, és legsokoldalúbban használható objektumaival a lekérdezésekkel ismerkedhet meg.

Segítségükkel az adatbázis felépítésével, rekordok tárolásával, módosításával, kiválogatásával, rendezésével kapcsolatos összes feladat elvégezhető. Leggyakrabban mégis azért használjuk őket, mert míg a táblákban használt eszközökkel kizárólag a kiválasztott tábla rekordhalmazában kereshettünk, rendezhettünk vagy szűrhettünk, addig a lekérdezések segítségével különböző táblák kapcsolódó rekordjainak együttes kezelésére nyílik lehetőség. A lecke olvasása során számos új fogalommal és művelettel találkozik majd. Törekedjen arra, hogy választ kapjon az alábbi kérdésekre:

Mi különbözteti meg a lekérdezést az irányított szűrőtől?
Mit jelent a forráshalmaz, és az eredményhalmaz kifejezés?
Hogyan kell összeállítani egy lekérdezés rekordforrását?
Mik azok a kapcsolódó rekordok?
Mi a különbség laza, és szoros illesztés között?
Hogyan készítünk és használunk számított mezőt a lekérdezésekben?
Hogyan használunk függvényeket?
Hogyan szabályozhatjuk a mezőértékek megjelenését az eredményhalmazban?
Mik azok a lekérdezés tulajdonságok?

## 8.2 A LECKE TÉMAKÖREI

- Rekordforrás, forráshalmaz, eredményhalmaz
- Új lekérdezés létrehozása
- Rekordforrás kialakítása
- Illesztési tulajdonságok beállítása
- Rekordok, és mezők kiválasztása
- Számított mezők és függvények
- Mezőtulajdonságok beállítása
- Lekérdezés tulajdonságok
- A lekérdezés eredményhalmaza

A leckében bemutatott példák az előző leckében használt **iskola2007.accdb** adatbázison alapulnak.

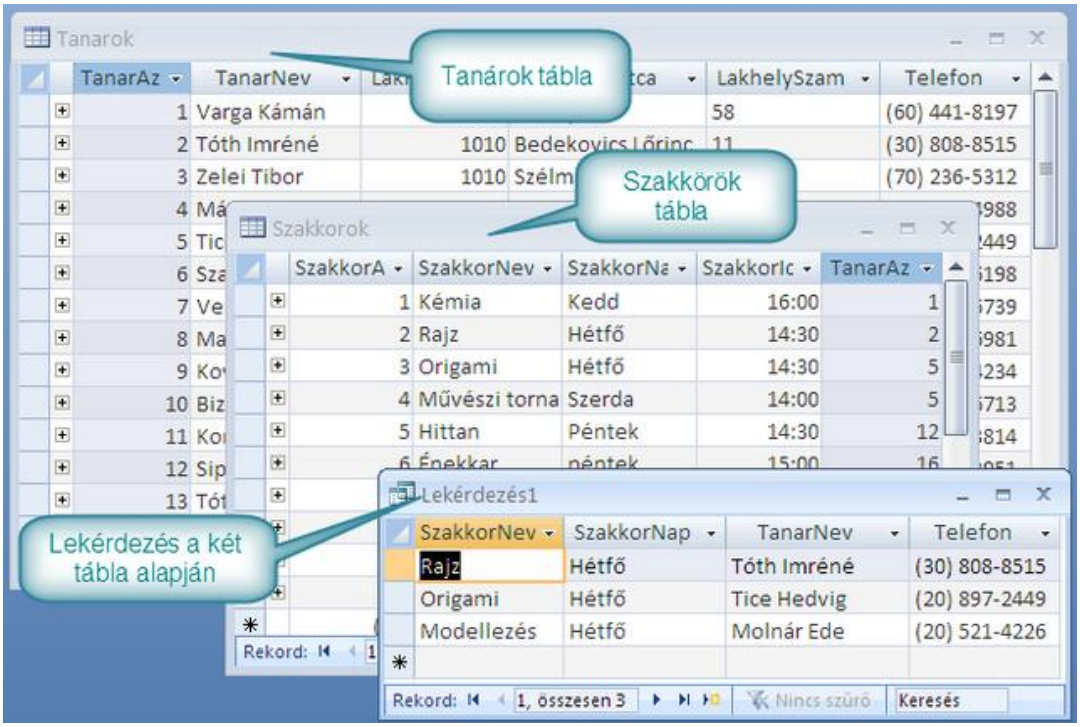
## 8.3 LEKÉRDEZÉSEK JELENTŐSÉGE

Az Access táblák rekordkezelő eszközei közül kiemelkedik az irányított szűrő, amely rekordok kiválogatására, rendezésre, sőt számított mezők készítésére is alkalmas. A szűrők rendkívül hasznos lehetőségeket kínálnak, de **mindig csak az éppen megnyitott tábla rekordjainak halmazán végeznek műveleteket.**

A legegyszerűbb lekérdezések funkciójukban és megtervezésük módjában is hasonlítanak az irányított szűrőkhöz, amennyiben **a lekérdezésekkel is rekordhalmazok dolgozhatók fel.** Lényeges különbség azonban, hogy ezek a **rekordhalmazok több tábla kapcsolódó rekordjaiból** is összeállíthatók.

Tegyük fel, hogy az iskola igazgatójának vasárnap délután az a remek ötlete támad, hogy iskolagyűlést szervez hétfő délutánra. Gyorsan értesíteni akarja az érintett tanárokat arról, hogy ebben az időben elmaradnak szakkörök. Meg kell tudnia, melyik szakkör tanárát, milyen telefonszámon hívhatja föl.

Az **iskola2007.accd** adatbázis ***Szakkorok*** táblájában csak azt tudhatjuk meg, hogy melyik szakköröket tartják hétfő délután, és mi a szakkört tartó tanár azonosítója. Ha meg akarjuk tudni a tanár nevét, és telefonszámát, akkor külön meg kell néznünk a ***Tanarok*** táblát is. Egy lekérdezés segítségével könnyedén megvizsgálhatjuk a két tábla (***Szakkorok-Tanarok***) kapcsolódó rekordjait, és ezeket mintegy összekapcsolva, együtt jeleníthetjük meg egy szakkör és a szakkört tartó tanár rekordját.



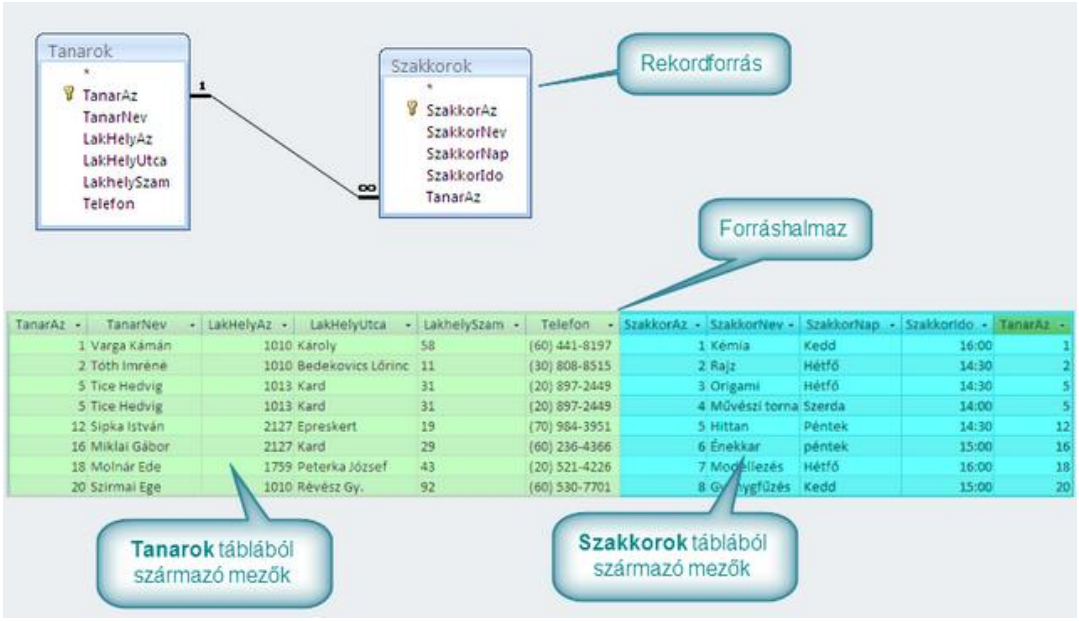

*49. ábra Táblák és lekérdezés*

## 8.4 REKORDFORRÁS, FORRÁSHALMAZ, EREDMÉNYHALMAZ

**Azt a rekordhalmazt, amit egy lekérdezés feldolgoz, a lekérdezés forráshalmazának nevezzük.**

A forráshalmaz– szerkezetét tekintve – olyan, mint egy tábla, azaz mezők és rekordok mátrixa alkotja. A mezők azonban nem csak egy, hanem egymással kapcsolatban álló, több táblából is származhatnak.

**Azokat az egymáshoz kapcsolódó táblákat, és kapcsolataikat, amelyek kapcsolódó rekordjaiból egy lekérdezés forráshalmaza származik, rekordforrásnak nevezzük.**




| TanarAz | TanarNev | LakHelyAz | LakHelyUtca | LakhelySzam | Telefon | SzakkorAz | SzakkorNev | SzakkorNap | SzakkorIdo | TanarAz |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Varga Kámán | 1010 | Károly | 58 | (60) 441-8197 | 1 | Kémia | Kedd | 16:00 | 1 |
| 2 | Tóth Imréné | 1010 | Bedekovics Lőrinc | 11 | (30) 808-8515 | 2 | Rajz | Hétfő | 14:30 | 2 |
| 5 | Tice Hedvig | 1013 | Kard | 31 | (20) 897-2449 | 3 | Origami | Hétfő | 14:30 | 5 |
| 5 | Tice Hedvig | 1013 | Kard | 31 | (20) 897-2449 | 4 | Művészi torna | Szerda | 14:00 | 5 |
| 12 | Sipka István | 2127 | Epreskert | 19 | (70) 984-3951 | 5 | Hittan | Péntek | 14:30 | 12 |
| 16 | Miklai Gábor | 2127 | Kard | 29 | (60) 236-4366 | 6 | Énekkar | péntek | 15:00 | 16 |
| 18 | Molnár Ede | 1759 | Peterka József | 43 | (20) 521-4226 | 7 | Modellezés | Hétfő | 16:00 | 18 |
| 20 | Szirmai Ege | 1010 | Révész Gy. | 92 | (60) 530-7701 | 8 | Gyógyfűzés | Kedd | 15:00 | 20 |

*50. ábra Rekordforrás, forráshalmaz*

Az irányított szűrőnek is volt forráshalmaza. Ez nem volt más, mint az a tábla, amiben a szűrést végeztük. A lekérdezés készítése alapvetően abban különbözik az irányított szűrő megtervezésétől, hogy itt nem adott a forráshalmaz. A lekérdezés kialakításakor nekünk kell megmondanunk, hogy mely táblák alkossák a rekordforrást, azaz honnan származzon a forráshalmaz.

A lekérdezés készítés legfontosabb lépése a **rekordforrás megadása**. Ha pontosan megadjuk, mely táblák kapcsolódó rekordjaival akarunk dolgozni, akkor az adatbázis-kezelő rendszer automatikusan kialakítja a forráshalmazt. A lekérdezés készítőjének csupán azt kell megmondania, melyik **mezőket**, és melyik **rekordokat** szeretné látni **a forráshalmazból.**

Ha megadtuk a rekordforrást, pontosan meghatároztuk, mely rekordokat, és mely mezőket szeretnénk megjeleníteni a forráshalmazból, akkor az Access automatikusan előállítja az eredményhalmazt.

**Az eredményhalmaz egy lekérdezés eredményeként keletkező rekordhalmaz, amely forráshalmaz kiválasztott rekordjaiból, és mezőiből áll.**

## 8.5 LEKÉRDEZÉS LÉTREHOZÁSA

A lekérdezések a forráshalmazzal végeznek műveleteket, a forráshalmaz összetétele pedig a rekordforrásban elhelyezett tábláktól, és azok kapcsolatától függ. Éppen ezért a rekordforrás tábláinak helyes kiválasztása a lekérdezés készítés szinte legfontosabb lépése. Miután képet alkottunk a forráshalmaz kialakításának módjáról, lássuk részleteiben, hogyan történik a lekérdezés készítése a gyakorlatban.

A lekérdezések elkészítésekor általában nem gondolkodunk azon, milyen lépések is szükségesek a feladat végrehajtásához. Most azonban, célszerű őket röviden összefoglalni:

- A rekordforrás megtervezése
- Létrehozzuk az új lekérdezést
- Kiválasztjuk a rekordforrás tábláit
- Meghatározzuk a táblák közötti illesztési tulajdonságokat
- A tervezőrácsra helyezzük a megfelelő mezőket
- Ha szükséges, számított mezőket hozunk létre
- Megadjuk a feltételeket, és rendezési szempontokat
- Szükség esetén lekérdezés tulajdonságokat állítunk be
- Mentjük a lekérdezés tervet
- Megtekintjük az eredményhalmazt

### 8.5.1 Rekordforrás megtervezése

A lekérdezések rekordforrásának meghatározása előtt célszerű megnézni a kapcsolatok ablakot. (menüszalag **Adatbáziseszközök/Kapcsolatok**) Innen minden információ megszerezhető, ami a helyes rekordforrás kialakításához szükséges. A lekérdezés készítésekor azokat a táblákat kell majd kiválasztanunk, amelyekben szerepelnek a leendő eredményhalmazban mezői. Ha az így kiválasztott táblák között nincs közvetlen kapcsolat, akkor azokra a táblákra is szükség lesz, amelyek a kapcsolatot biztosítják.

Ha például arra vagyunk kíváncsiak, hogy kik tartanak szakkört hétfőn, akkor elegendő a ***Szakkorok***, és ***Tanarok*** kiválasztása, mert a szükséges mezők ezekben a táblákban vannak, és a két tábla közvetlen kapcsolatban van.

Ha azonban azt akarjuk tudni, hogy melyik gyerek, melyik szakkörbe jár, akkor a ***Tanulok***, és ***Szakkorok*** táblán túl a ***TanuloSzakkor*** táblára is szükség lesz.

### 8.5.2 Új lekérdezés létrehozása, rekordforrás megadása

Új lekérdezést a menüszalag **Létrehozás** lapján az **Egyebek** csoport **Lekérdezéstervező** parancsával hozunk létre. A parancs hatására a háttérben megjelenik a lekérdezés tervező ablak, ami feltűnően hasonlít az irányított szűrő felületére. Az ablak alsó részében a **tervező rács**, felső részében a **mezőlista ablakrész** látszik. Itt fogjuk látni a rekordforrás tábláit és a közöttük lévő kapcsolatokat. A mezőlista rész egyenlőre természetesen üres, de a képernyőn látszik a **Tábla megjelenítése** ablak. Innen választhatjuk ki a rekordforrás tábláit. A szükséges táblákat rendje kijelöljük, majd a hozzáadás gombra kattintunk.

A **Tábla megjelenítés** ablak bezárása után a mezőlista ablakrészben megjelennek a rekordforrás táblái, és láthatóak a közöttük lévő kapcsolatok.

Ha véletlenül olyan táblát is beillesztettünk volna, amire nincs szükség, kattintsunk a tábla mezőlistájára, és a **Delete** billentyűvel töröljük a rekordforrásból.

Ha újabb táblával akarjuk bővíteni a rekordforrást, kattintsunk menüszalagon a **Lekérdező eszközök/Tervezés/Tábla** megjelenítése gombra.

A táblák mezőlistái a Kapcsolatok ablakban megadott módon kapcsolódnak. Célszerű úgy elrendezni őket, hogy minden tábla mezőlistája jól látszódjon, és a kapcsolatokat jelző vonalak ne keresztezzék egymást. Ennek a lekérdezés működése szempontjából semmi jelentősége nincs, de áttekinthetőbbé teszi a lekérdezés tervét.

### 8.5.3 Illesztési tulajdonságok

Az illesztési tulajdonságokról később, részletesen olvashat. Egyenlőre elég annyit tudnia, hogy az illesztési tulajdonságokkal szabályozhatjuk, hogy a rekordforrás rekordjai hogyan kapcsolódjanak össze. Kétféle illesztési tulajdonság létezik. **Laza**, és **szoros illesztés**. Ha másként nem rendelkezünk, akkor mindig szoros illesztés lesz a rekordforrás táblái között. Egyenlőre szoros illesztést fogunk használni.

### 8.5.4 Mezők elhelyezése a tervezőrácson

A rekordforrással meghatározott forráshalmaz nem jelenik meg a képernyőn. A lekérdezés készítőjének egyszerűen el kell képzelnie egy táblát (rekordok-mezők), amelyben a rekordforrás tábláinak kapcsolódó rekordjai, és a táblák összes mezője szerepel.

Ebből a virtuális táblából kell kiválasztanunk azokat a mezőket, amelyeket látni akarunk az eredményhalmazban. A megfelelő mezőlistából az irányított szűrőknél tanult módon helyezhetünk mezőt a rácsra:

- dupla kattintással a mezőn,
- mező tervezőrácsra húzásával,
- egy oszlop Mező sorának lenyíló listájából választva, vagy
- a mezőnevet egy oszlop Mező sorába gépelve.

Az eredményhalmazban csak a tervező rácson elhelyezett mezők fognak megjelenni.

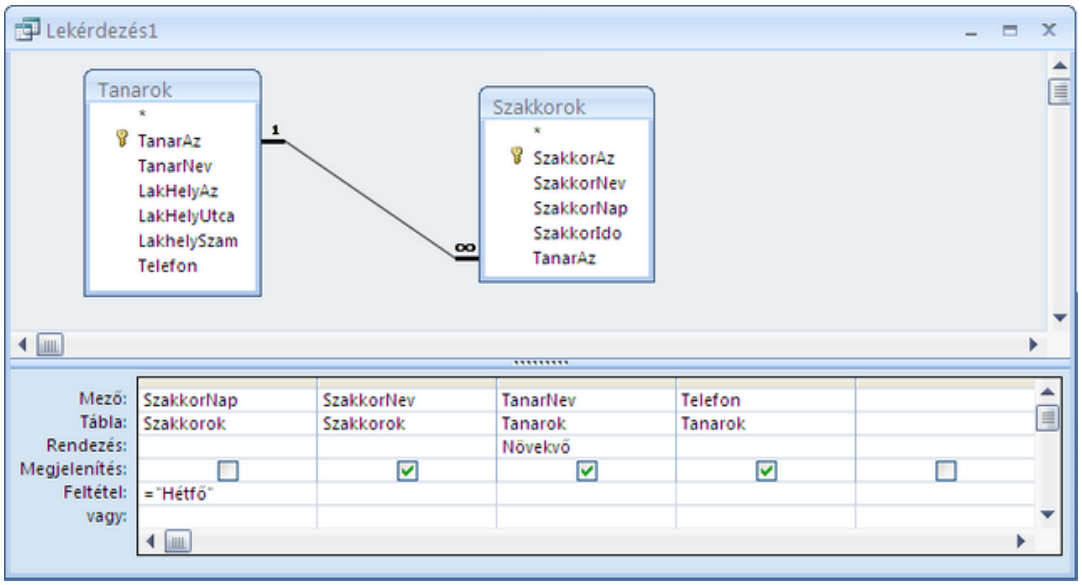
Szükség esetén az irányított szűrőkkel megegyező módon, a lekérdezésekben is létrehozhatunk számított mezőket. Ezek kialakításáról, és használatáról a tananyag további oldalain bővebben olvashat.

Az eredményhalmazban nem jelenik meg a forráshalmaz összes mezője, csak azok, amelyek a tervezőrácson szerepelnek. Ezek a mezők azonban, (így az esetleges számított mezők is), szerepelni fognak az eredményhalmaz mezői között.

A tervezőrács **Megjelenítés** kapcsolójával eltűntethetjük az eredményhalmazból azokat a mezőket, amelyeket csak feltétel, vagy rendezés megadására használunk a rácson, de az eredményhalmazban nem akarunk látni.

### 8.5.5 Feltételek, rendezési irány megadása

A feltételeket, és rendezési szempontokat a tervezőrácson elhelyezett mezők **Rendezés** és **Feltétel** soraiban adhatjuk meg az Irányított szűrőkkel teljesen megegyező módon.




*51. ábra Feltételek és rendezések*

### 8.5.6 Mező, és Lekérdezés tulajdonságok

A mező, lekérdezés tulajdonságokkal később foglakozunk. Egyelőre elég annyit tudnunk, hogy a lekérdezés eredményhalmazának tartalma, és formátuma az eddig tanultakon kívül, mező és lekérdezés tulajdonságokkal is szabályozható.

### 8.5.7 Lekérdezés tervének mentése

Az elkészült lekérdezést általában lemezre mentjük. Fontos tudni, hogy ilyenkor nem a lekérdezés forráshalmaza, vagy eredményhalmaza, hanem csupán a lekérdezés terve kerül lemezre. A forráshalmaz rekordjai csupán a rekordforrás megfelelő tábláiban tárolódnak.

A lekérdezés mentését a gyorsindító eszköztár **Mentés** parancsával, vagy a **Ctrl+S** billentyűkkel végezhetjük el.

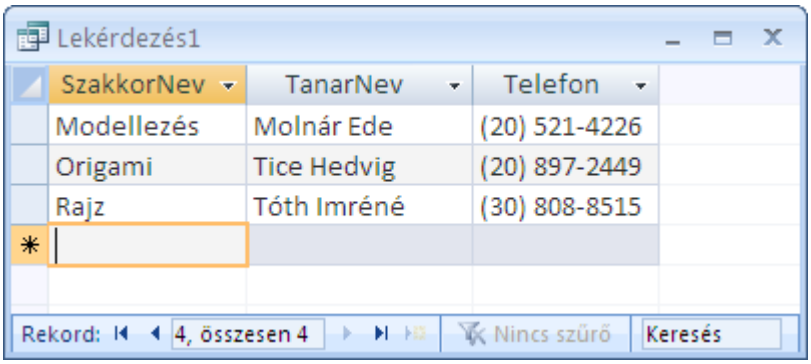
> A lekérdezés mentésekor nevet kell adni az objektumnak. A lekérdezések neveinek megadásakor be kell tartanunk az Access névadási szabályait, és figyelnünk kell arra is, hogy az új név nem egyezhet sem már létező lekérdezés, sem pedig létező tábla nevével.

### 8.5.8 Eredményhalmaz megtekintése

Ha látni szeretnénk a lekérdezés eredményhalmazát, akkor tervező nézetből felhasználói nézetbe kell váltanunk. Ezt a menüszalag

**Lekérdezéseszközök/Tervezés/Eredmények/Nézet** gomb közepére kattintva tehetjük meg.

Ha a gomb alján látható nyilacskára kattintunk, a lekérdezés egyéb lehetséges nézeteit választhatjuk ki.




*7. kép Lekérdezés eredményhalmaza*

Az felhasználói nézet ablaka teljesen megegyezik a táblák felhasználói nézetével, sőt az eredményhalmazon a tábláknál használt összes művelet (keresés, rendezés, szűrés) elvégezhető.

> Fontos tudni, hogy a forráshalmazt, és az eredményhalmazt, az Access csak a memóriában tárolja, és mindig akkor hozza létre, amikor a lekérdezést felhasználó nézetben nézzük. Ha a felhasználói nézetben megváltoztatjuk valamelyik mező értékét, akkor a változtatás érvényesül a rekordforrás megfelelő táblájában, de a felhasználói nézet bezárásakor forráshalmaz, és az eredményhalmaz is törlődik a memóriából.

Az eredményhalmaz megtekintése után ismét a menüszalag **Nézet** gombjával térhetünk vissza a tervező nézetbe.

## 8.6 ILLESZTÉSI TULAJDONSÁGOK

A lekérdezés forráshalmaza a rekordforrás kapcsolódó tábláiban tárolt rekordok összeillesztésével alakul ki. Tudjuk, hogy a lekérdezés készítés legfontosabb lépése a rekordforrás helyes meghatározása, hiszen a lépéstől függ a forráshalmaz összetétele. Az eddigiekben megtudtuk, hogyan illeszti össze az Access a rekordforrás tábláinak rekordjait. Arra azonban még csak utaltunk, hogy az illesztés módja szabályozható az úgynevezett illesztési tulajdonságok beállításával.

### 8.6.1 Szoros illesztés

Két tábla kapcsolatából az Access, alaphelyzetben az kapcsolódó rekordokat illeszti össze. Mindkét tábla rekordjaiból csak az kapcsolódó rekordokat választja ki. Azokat összeilleszti a másik tábla kapcsolódó rekordjaival, majd az így kapott rekordokat elhelyezi a forráshalmazban. Ezt az illesztést nevezzük **szoros illesztésnek.**

Szoros illesztést használunk a rekordforrás táblái között, ha olyan forráshalmazt akarunk kialakítani, amely megmutatja, hogy egy tábla melyik rekordjai kapcsolódnak egy másik tábla valamelyik rekordjához. Az esetek többségében szoros illesztésre van szükségünk..

A ***Tanarok*** és ***Szakkorok*** táblák szoros illesztésű kapcsolatára van szükség például akkor, ha tudni szeretnénk, melyik tanárok tartanak szakkört. A ***Tanarok*** és ***Osztalyok*** táblák szoros illesztésű kapcsolatát alakítjuk ki olyankor, amikor arra vagyunk kíváncsiak, hogy melyik tanárok osztályfőnökök.

### 8.6.2 Laza illesztés

Előfordulhat, hogy egy tábla egyes rekordjai kapcsolódnak egy másik tábla rekordjaihoz, mások pedig nem. A szoros illesztés nem ad választ arra, hogy mely rekordok NEM kapcsolódnak a másik táblához. A ***Tanarok*** és ***Osztalyok*** táblák szoros illesztésű kapcsolata mutatja, hogy kik osztályfőnökök, de nem ad választ arra a kérdésre, hogy kik nem azok. Mit tehetünk, ha ennek éppen erre lennénk kíváncsiak:

Kik azok a tanárok, akik nem osztályfőnökök?

Ha azt akarjuk megtudni, hogy egy tábla mely rekordjai kapcsolódnak egy másik tábla rekordjaihoz, és melyek **nem**, akkor **laza illesztést** kell használnunk.

Amikor táblákat helyezünk egy lekérdezés rekordforrásába, akkor az Access alapértelmezés szerint szoros illesztést állít be a táblák között. A szoros illesztés külön beállítása ezért általában nem szükséges. A laza illesztés beállításához azonban egyrészt értenünk kell, hogyan is alakul ilyenkor a forráshalmaz, másrészt tudnunk kell, hol szabályozhatjuk a táblák közötti illesztési tulajdonságokat.

Az egyszerűség kedvéért nevezzük a kapcsolatban lévő táblák egyikét „A”, a másikat „B” táblának.

Szoros illesztéskor az „A” és a „B” táblából is csak a kapcsolódó rekordok kerülnek a forráshalmazba. A laza illesztéskor az „A” tábla minden rekordja belekerül a forráshalmazba, függetlenül attól, hogy kapcsolódik-e a „B” tábla valamelyik rekordjához, vagy sem. A „B” táblából azonban csak a kapcsolódó rekordok kerülnek a forráshalmazba, természetesen az „A” tábla kapcsolódó rekordjaival összeillesztve. Így azonban előfordulhat, hogy a forráshalmaz egyes „A” táblából származó rekordjaihoz egyetlen, a „B” táblából származó rekord sem kapcsolódik. A forráshalmazban ilyenkor NULL értékek fogják kitölteni „B” táblából származó mezőket.

A példában a ***Tanarok*** és ***Osztalyok*** tábla laza illesztésű kapcsolatából kialakított forráshalmazt látja.

| TanarAz | TanarNev | LakHelyAz | LakHelyUtca | LakhelySzam | Telefon | OsztAz | Evfolyam | Betujel | Terem | TanarAz |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Varga Kámán | 1010 | Károly | 58 | (60) 441-8197 | 7 | 4 | A | A.17. | 1 |
| 2 | Tóth Imréné | 1010 | Bedekovics Lőrinc | 11 | (30) 808-8515 | 5 | 3 | A | A.15. | 2 |
| 3 | Zelei Tibor | 1010 | Szélmalom | 29 | (70) 236-5312 | 16 | 8 | B | B.8. | 3 |
| 4 | Márai Mihály | 1010 | Bajnár | 30 | (20) 832-4988 | 4 | 2 | B | A.14. | 4 |
| 5 | Tice Hedvig | 1013 | Kard | 31 | (20) 897-2449 | 1 | 1 | A | A.11. | 5 |
| 6 | Szabó Elek | 1010 | Vas | 98 | (20) 491-6198 | 12 | 6 | B | B.4. | 6 |
| 7 | Veres Anna | 1013 | Károly | 28 | (20) 242-6739 | 15 | 8 | A | B.7. | 7 |
| 8 | Martin László | 1010 | Dékáni Á. | 17 | (60) 438-6981 | 14 | 7 | B | B.6. | 8 |
| 9 | Kovács Jakab | 1010 | Platán | 42 | (70) 411-4234 | 3 | 2 | A | A.13. | 9 |
| 10 | Bizkó Tivadar | 1013 | Kassa | 72 | (20) 859-6713 | 2 | 1 | B | A.12. | 10 |
| 11 | Komáromi Péter | 2127 | Izabella | 64 | (20) 928-3814 | 9 | 5 | A | B.1 | 11 |
| 12 | Sipka István | 2127 | Epreskert | 19 | (70) 984-3951 | 11 | 6 | A | B.3. | 12 |
| 13 | Tóth Attila | 1010 | Ormánság | 9 | (70) 285-9995 | 13 | 7 | A | B.5. | 13 |
| 14 | Róbert Georgina | 1010 | Kolmann | 91 | (70) 212-6601 | 6 | 3 | B | A.16. | 14 |
| 15 | Mihailik Mónika | 1010 | Fejérfold | 79 | (30) 282-8391 | 10 | 5 | B | B.2. | 15 |
| 16 | Miklai Gábor | 2127 | Kard | 29 | (60) 236-4366 | 8 | 4 | B | A.18. | 16 |
| 17 | Ziski Ivan | 1013 | Városháza | 64 | (60) 440-9588 | | | | | |
| 18 | Molnár Ede | 1759 | Peterka József | 43 | (20) 521-4226 | | | | | |
| 19 | Krul Szabolcs | 1010 | Bedekovics Lőrinc | 57 | (30) 330-7172 | | | | | |
| 20 | Szirmai Ege | 1010 | Révész Gy. | 92 | (60) 530-7701 | | | | | |
| 21 | Virág Éva | 1013 | Szilvás köz | 3 | (60) 706-3660 | | | | | |

*52. ábra Tanarok-Osztalyok laza illesztésű kapcsolatából kapott forráshalmaz*

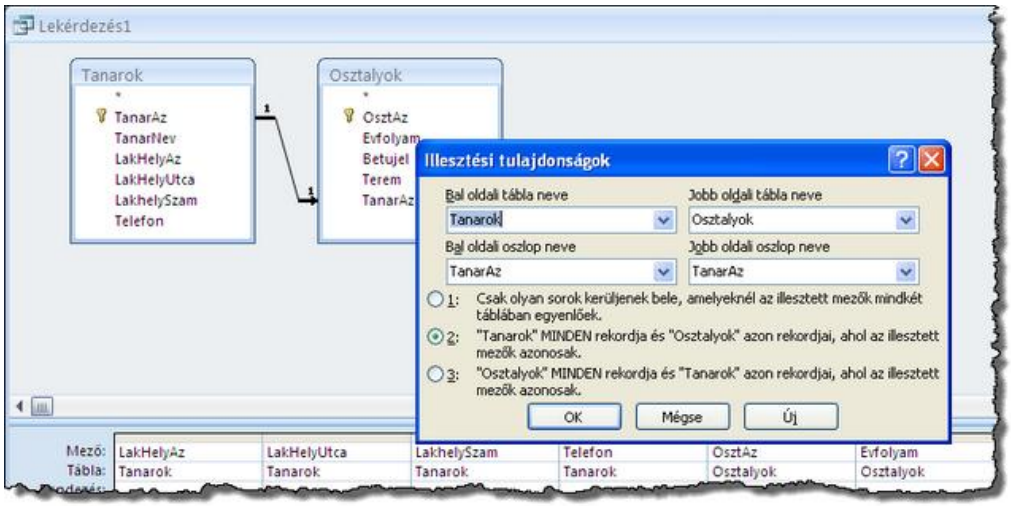
Laza illesztéssel olyan forráshalmazt kapunk, amelyben az „A” tábla összes rekordja benne van, de csak a kapcsolódó rekordokhoz illeszkedik rekord a „B” táblából. A nem kapcsolódó rekordokban NULL értékű mezők vannak. Ez alkalmat ad arra, hogy egy (**Is Null**) feltétellel éppen kiválogassuk az ilyen rekordokat, azaz megmutassuk, hogy az „A” tábla mely rekordjai **nem** kapcsolódnak a „B” táblához.

### 8.6.3 Illesztési tulajdonságok beállítása

A rekordforrás tábláinak illesztési tulajdonságait a kapcsolatot jelképező vonalon, dupla kattintással, vagy a kapcsolatvonal helyi menüjében az **Illesztési tulajdonságok** parancscsal megjelenített ablakban állíthatjuk be.

Az ablakban látható első opció szoros illesztést jelöl. A második két opcióval kapcsolhatjuk be a laza illesztést. Kétféle laza illesztés választható ki. Bal oldali laza illesztésről beszélünk, ha két tábla kapcsolatában a bal oldali táblából minden rekord, a jobb oldaliból pedig csak az illeszkedő rekordok kerülnek a forráshalmazba. Jobb oldali laza illesztésnél a jobb oldali táblából minden rekord, míg a bal oldaliból csak az illeszkedő rekordok kerülnek a rekordforrásba.

Az Access nem nevezi meg a bal, és jobb oldali laza illesztéseket, hanem szövegesen írja le az egyes opciók jelentését.

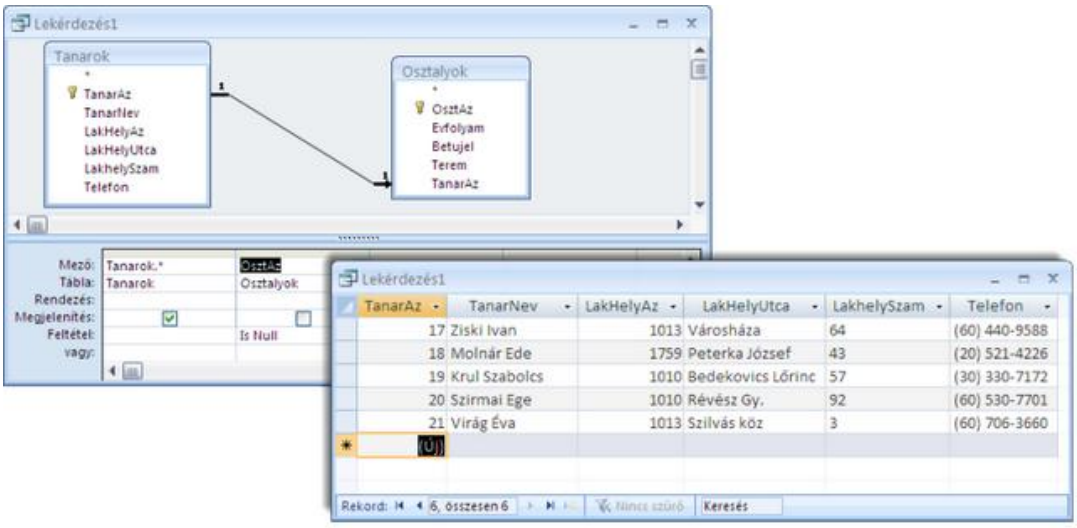



*8. kép Illesztési tulajdonságok ablak*

Az illesztés a lekérdezés tervezőablakában is felismerhető. Míg szoros illesztésnél egyszerű vonal, addig laza illesztésnél nyíl köti össze a kapcsolatban lévő táblákat. A nyíl mindig az összes rekordot adó táblából, a csak az kapcsolódó rekordokat adó tábla felé mutat.

### 8.6.4 Nem kapcsolódó rekordok kiválogatása

A laza illesztés bekapcsolása után a nyílhegy irányában lévő táblából származó mezőkben NULL értékek jelzik, ha nincs kapcsolódó rekord. Ha az ebből a táblából származó bármely mező alá az **Is Null** feltételt írjuk, az eredményhalmazban már csak a nem kapcsolódó rekordok jelennek meg.




*9. kép Lekérdezés terve és eredményhalmaza*

## 8.7 SZÁMÍTOTT MEZŐK

A lekérdezések készítése közben gyakori feladat, hogy olyan értéket akarunk megjeleníteni az eredményhalmazban, amely a forráshalmaz egyik mezőjében sem szerepel, de a forráshalmaz mezőinek felhasználásával kiszámítható. Ilyen esetben az irányított szűrőkben használható lehetőségnek megfelelően számított mezőket hozhatunk létre.

A számított mező olyan virtuális mező, amelynek értéke más mezők értékének felhasználásával állítható elő. Az ilyen mezőt úgy hozunk létre, hogy a tervezőrács egy üres oszlopának mező sorába beírjuk a mező nevét, majd egy kettőspont (:) után feltüntetjük a mező kiszámítására használható kifejezést.

Az alábbi példában a Tanulok táblából készült lekérdezésben számított mezőben határozzuk meg a Tanulók szüleinek összes keresetét.

A számított mező kialakításának módja pontosan megegyezik az irányított szűrőknél tanultakkal. A szűrőkkel ellentétben azonban a számított mező nem csak feltétel, vagy rendezési irány megadására alkalmasak, hanem meg is jelennek az eredményhalmazban.

A következő példában három számított mezőt láthat, a ***Tanulok*** táblából készült lekérdezésben

A szülők összes keresetét például így számoljuk ki:

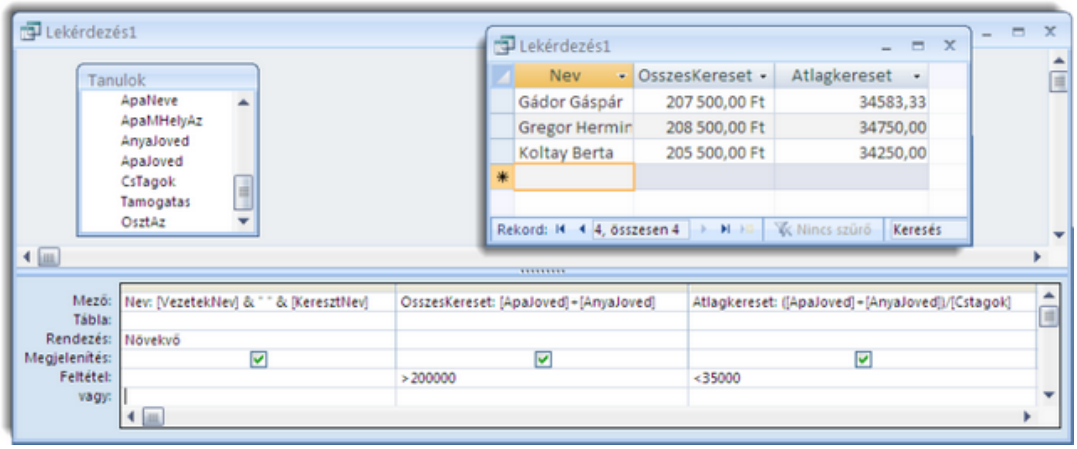
**OsszesKereset: [ApaJoved]+[AnyaJoved]**

A család egy főre jutó átlagjövedelmét a már létrehozott OsszesKereset mezőben használt képlet felhasználásával határozzuk meg:

**Atlagkereset: [ApaJoved]+[AnyaJoved]/[Cstagok]**

A teljes név megjelenítéséhez pedig összevonjuk a **[VezetekNev]**, és a **[KeresztNev]** mezőket, de közöttük még egy szóközt is elhelyezünk.

**Nev: [VezetekNev] & " " & [KereszNev]**




*53. ábra Számított mezők az eredményhalmazban*

### 8.7.1 Függvények használata

Számított mezők kialakítása, vagy akár speciális feltételek megadása közben szükségünk lehet olyan bonyolultabb számításokra, amelyek kifejezések formájában történő megfogalmazása bonyolult, vagy akár teljesen lehetetlen is.

Ezen a problémán igyekeznek segíteni a Access fejlesztői akkor, amikor különböző számítások megoldását függvények formájában beépítik egy rendszerbe. Számos függvénnyel találkozhatunk például az Excel-ben, de használhatunk függvényeket a Word-ben és természetesen az Accessben is.

**A függvények olyan névvel ellátott, beépített funkciók, amelyek kiszámítják, és visszaadják valamilyen műveletet eredményét. A függvény eredményét visszatérési értéknek is nevezzük.**

#### *Függvények neve, és meghívása*

Minden függvénynek van **neve**, ami utal a függvénnyel elvégezhető műveletre. A **Date()** nevű függvény neve például arra utal, hogy a függvény az aktuális dátumot adja eredményként. A függvények nevét mindig kerek zárójelpár követi.

A függvényeket kifejezésekben használhatjuk, pontosan úgy, mint egyéb tényezőket. A kifejezés kiértékelésekor az Access a függvény helyére annak eredményét helyettesíti, és azzal végzi el a kifejezés kiértékelését. A **Date()-[SzulIdo]** például ilyen kifejezés. A **Date()** függvény az aktuális dátumot adja vissza, a **[Szulido]** egy tanuló születési idejét tartalmazza. A kifejezés eredménye a két időpont között eltelt napok száma.

Amikor egy függvény nevét szerepeltetjük egy kifejezésben, azt mondjuk, hivatkozunk a függvényre, vagy meghívjuk a függvényt. Mindkét kifejezés ugyanazt jelenti: felszólítjuk

a rendszert, hogy végezze el a számítást, és a függvény visszatérési értékével számoljon a kifejezésben.

### *Függvények paraméterei*

A legtöbb függvény valamilyen adatot használ fel a számításhoz. Ezeket az átadott adatokat nevezzük a függvény paramétereinek. A paramétereket mindig a függvény neve után álló kerek zárójelek között adjuk át a függvénynek. A **Year** függvény például egy dátumból kiszámolja az évszámot, és azt adja vissza eredményként. A függvénynek ezért meghíváskor egy dátumot kell megadnunk! A következő függvényhívás például a tanuló születési évét adja eredményként: **YEAR([SzulIdo])**

Léteznek paraméter nélküli függvények, de olyanok is, amelyeknek több paraméterre is szükségük van a számításhoz. Ha a függvény több paramétert igényel, akkor azokat kerek zárójelpár között, egymás után, pontosvesszővel elválasztva adjuk meg.

A **DateSerial** függvény három számból (*év;hónap;nap*) egy dátumot hoz létre. A három szám a függvény három paramétere, amit így kell átadni:

**DateSerial(2008;04;20)** A függvény eredménye a 2008.04.20 dátum.

Amikor egy függvényt meghívunk a függvény összes szükséges paraméterét át kell adnunk, mégpedig a megfelelő sorrendben.

### *Függvények az Access súgójában*

MS Access 2007 Súgójában minden használható függvény leírása megtalálható. A Súgót az F1 billentyűvel jeleníthetjük meg. Ha a **Keresés** mezőbe beírjuk a függvény nevét, majd gombra kattintunk, megjelennek a megfelelő súgó témakörök.

### *Néhány fontosabb függvény*

Az Accessben nagyon sok függvényt használhatunk. Lássunk néhányat a fontosabbak közül:

| Függvény formátuma | Leírás | Példa hívásra és visszatérési értékre |
|---|---|---|
| **Date()** | Az aktuális dátumot adja | **Date()** =>pl.: 2008.04.20 |
| **Time()** | Az aktuális időpontot adja | **Time()** => pl.: 10:12:20 |
| **Now()** | Az aktuális dátumot, és időpontot adja | **Now()** => pl.: 2008.04.20 10:12:20 |
| **Year(***dátum***)** | Egy dátumból az évszámot adja | **Year( Date() )** => 2008 |
| **Month(***dátum***)** | Egy dátumból a hónapot adja | **Month( Date() )** => 4 |
| **Day(***dátum***)** | Egy dátumból a napot adja | **Day( Date() )** => 20 |
| **DateSerial(***év;hó;nap***)** | Az év, hónap, és nap számából dátumot készít | **DateSerial(2008;4;20)**=> 2008.04.20 |
| **Nz(***mező; érték_ha_null***)** | Visszatérési érték az első | **Nz([ApaJoved];0)** => az |

| | | |
|---|---|---|
| | paraméterben megadott mező értéke, ha az nem **Null**. Ha a mező **Null**, akkor a második paraméter a visszatérési érték. | [***ApaJoved***] értéke, ha az nem null.<br>Ha az [***ApaJoved***] Null, akkor a visszatérési érték **0**. |

### *Az NZ függvény*

Az NZ() függvényt gyakran kell alkalmazni mezőhivatkozást tartalmazó számításokban azért, mert egy Null értékű tényezőt tartalmazó kifejezés értéke mindenképpen Null lesz.

Az **OsszeKereset: [ApaJoved]+ [AnyaJoved]** számított mezőtől azt várjuk, hogy a két szülő összesített keresetét tartalmazza majd. Ez így is lesz, feltéve, hogy mindkét szülő rendelkezik jövedelemmel. Ha azonban valamelyikük jövedelme **Null** értékű, akkor az ***OsszesKereset*** mező akkor is **Null** lesz, ha a másik szülőnek van keresete.

Ez könnyen belátható a következők alapján:

A **Null** jelentése **semmi**.

**150000+100000=250000**

azonban

**semmi+100000=semmi**

Ezt a problémát tudjuk orvosolni az **Nz()** függvénnyel.

**OsszesKereset: Nz([ApaJoved];0)+ Nz([AnyaJoved;0)**

Ha így számoljuk ki az összes keresetet, akkor a **Null** jövedelemmel rendelkező szülő keresetének helyére 0 kerül a számításban. Így biztosan nem kapuk **Null** értékű összes jövedelmet.

## 8.8 MEZŐ ÉS LEKÉRDEZÉS TULAJDONSÁGOK

A lekérdezések eredményhalmazát elsősorban a rekordforrással a mezők kiválasztásával, a feltételek és rendezési irányok megadásával szabályozzuk. Az eredményhalmaz kialakításának utolsó lehetőségét az úgynevezett lekérdezés tulajdonságok adják. A lekérdezés tulajdonságok nevesített értékek, amelyek szabályozzák a lekérdezés működését. A számos lekérdezés tulajdonság közül Önnek az alábbiakat kell ismernie:

- Leírás
- Összes mező a kimentre
- Egyedi értékek
- Egyedi rekordok

### 8.8.1 Lekérdezés tulajdonságok beállítása

A lekérdezés tulajdonságok megtekintése és beállítása a lekérdezés tulajdonságlapján lehetséges, melyet a menüszalagon a

**Lekérdezőeszközök/Tervezés/Megjelenítés/elrendezés/Tulajdonságlap** parancsra kattintva, vagy a **Ctrl+Enter** billentyűkombinációt lenyomva jeleníthetünk meg. Miután a Tulajdonságlap megjelent, kattintsunk a mezőlista ablakrész egy üres területére!

### 8.8.2 Fontosabb lekérdezés tulajdonságok:

#### *Leírás*

Valójában nem befolyásolja az eredményhalmazt. A leírás csupán informatív szöveg, amit saját magunk számára tárolhatunk el a lekérdezésről. A leírás egyfajta emlékeztető, ami navigációs ablakban is megtekinthető a lekérdezés helyi menüjének **Tulajdonságok** parancsával.

#### *Összes mező a kimenetre*

Ezzel a tulajdonsággal elérhető, hogy a forráshalmaz minden mezője megjelenjen az eredményhalmazban, függetlenül attól, hogy a tervezőrácson milyen mezők vannak.

#### *Csúcsérték*

Az egyik leghasznosabb tulajdonság, amellyel azt szabályozzuk, hogy hány rekord jelenjen meg az eredményhalmaz elejéről.

Beállítható értékei:

- az „**Összes**”,
- valamilyen egész szám, vagy
- százalékos (%) érték.

A rekordok rendezésével együtt alkalmazva olyan kérdésekre kereshetünk választ, mint például:

- Melyik az a 10 tanuló, akiknél a szülők összes keresete a legmagasabb.
- Kik alkotják az iskola tanulói közül a legfiatalabb 5%-ot.
- Ki az első 5, legjobb egy főre jutó átlagkeresettel rendelkező gyerek az iskolában.

#### *Egyedi értékek*

Az egyedi értékek tulajdonság Igen/Nem értéket vehet fel. Ha bekapcsoljuk (Igenre állítjuk) akkor az eredményhalmazban csak egy példányban jelennek meg az egyébként ismétlődő rekordok.

Ha meg akarjuk mutatni, milyen női keresztnevek fordulnak elő az iskolában, akkor egy nagyon egyszerű lekérdezést készíthetünk, amelynek rekordforrását a ***Tanulok*** tábla adja, a tervezőrácson pedig mindössze a ***KeresztNev*** és a ***Neme*** mező áll. Utóbbi alatt **=0** feltételt adunk meg. Az eredményhalmazban azonban természetesen minden lány tanuló keresztneve megjelenik.

Ha azonban bekapcsoljuk az Egyedi értékek tulajdonságot, akkor az ismétlődő sorokat elrejti az Access, így minden női név csak egyszer jelenik meg!

#### *Egyedi rekordok*

Az egyedi rekordok hasonló hatású tulajdonság, mint az egyedi értékek, ezért **a két tulajdonság nem is állítható egyszerre igenre**! Az egyedi rekordokat akkor használjuk, ha rekordforrásban 1:N kapcsolatban álló táblák vannak, és csak az egy oldalon álló táblából teszünk mezőt a tervezőrácsra. A tulajdonság bekapcsolásával elérhető, hogy az 1:N kap-

csolat miatt többször is megjelenő rekordok csak egyszer szerepeljenek eredményhalmazban.

A ***Tanulok*** és ***TanuloSzakkor*** táblákból alkotott rekordforrás esetén a ***Tanulok*** táblának csak azok a rekordjai illeszkedő rekordok, amelyben a tanuló jár valamilyen szakkörbe. A tanuló azonosítója ilyenkor megtalálható a ***TanuloSzakkor*** táblában. Mivel egy gyerek több szakkörbe is járhat, elképzelhető, hogy rekordja többször is bekerül a forráshalmazba, persze mindig más ***TanuloSzakkor*** rekorddal összeillesztve.

Ha egy ilyen rekordforrásban a ***VezetekNev*** és a ***KeresztNev*** mezőket tesszük a tervezőrácsra, néhány gyerek neve többször is megjelenik. Ők több szakkörbe is járnak. Ha egy tanulót csak egyszer akarunk látni (nem vagyunk kíváncsiak arra, hogy hány szakkörbe jár, csak arra, hogy egyáltalán jár-e?), akkor kapcsoljuk be az **Egyedi rekordok** tulajdonságot!

Az egyedi értékek tulajdonság csak látszólag lenne helyes megoldás. Az ugyanis az eredményhalmazban rejtené el az ismétlődéseket, így ha több azonos nevű gyerek is ját szakkörbe, a nevük csak egyszer jelenne meg.

### 8.8.3 Mezőtulajdonságok beállítása

Lekérdezések készítésekor előfordulhat, hogy szabályozni szeretnénk az eredményhalmaz egy vagy több mezőjének jellemzőit. Erre a lekérdezés tulajdonságokhoz hasonlóan a **Tulajdonságlap**-ot használhatjuk. Amikor társzik a tulajdonságlap, kattintsunk a szabályozni kívánt mezőre. A rendelkezésre álló tulajdonságokról bővebb információt találunk az Access súgójában. Megjelenítéséhez kattintsunk a tulajdonságra, majd nyomjuk le az F1 billentyűt!

## 8.9 ÖSSZEFOGLALÁS

A lekérdezések a legsokoldalúbb objektumok az Access eszközei között. Alkalmazásukkal több tábla kapcsolódó rekordjainak együttes kezelésére nyílik lehetőség. A legegyszerűbb lekérdezések rekordokat és mezőket válogatnak ki egy táblából, vagy több tábla kapcsolatából álló rekordforrásból képzett forráshalmazból.

Lekérdezések készítésekor meghatározzuk a rekordforrás tábláit, beállítjuk a közöttük lévő illesztési tulajdonságokat, a tervezőrácsra helyezzük a szükséges mezőket, megadjuk a feltételeket és rendezési irányokat. Ha szükséges, számított mezőket készíthetünk, amelyek az irányított szűrőkkel szemben, meg is jelennek az eredményhalmazban.

A számított mezőkben használhatunk függvényeket. Segítségükkel bonyolult számítások végezhetők el. Az egyik legfontosabb, az **NZ(***mező***;***érték***)** függvény, amellyel mezők **Null** értékei kezelhetők.

A lekérdezés rekordforrását alkotó táblák között szoros és laza illesztésű kapcsolat lehet. Szoros illesztésű kapcsolat esetén mindkét táblából csak a kapcsolódó rekordok kerülnek a forráshalmazba. Laza illesztés esetén az egyik táblából minden rekord, a másik táblából csak a kapcsolódó rekordok.

A lekérdezések eredményhalmazának tartalmát és formátumát a mező és lekérdezés tulajdonságok is szabályozzák.

## 8.10 ÖNELLENŐRZŐ KÉRDÉSEK

1. Honnan tudhatjuk meg, hogy milyen táblákból állítsuk össze egy lekérdezés rekordforrását?
2. Mi az a forráshalmaz?
3. Mire való a lekérdezések tervezőrácsán a **Megjelenítés** sor?
4. Mire való a tervező rács **Tábla** sora?
5. Hogyan lehet egy eredményhalmazban megmutatni egy az 'A' és 'B' tábla kapcsolatán alapuló lekérdezésben az 'A' tábla minden (kapcsolódó, és nem kapcsolódó) rekordját?
6. Milyen értékek lesznek a 'B' mezőiben a nem kapcsolódó rekordok soraiban?
7. Mennyi lesz az **OsszKer:[ApaJoved]+[AnyaJoved]** mező értéke, azokban a rekordokban, ahol valamelyik szülőnek nincs keresete?
8. Hogyan tudna olyan, **EletKor** nevű számított mezőt csinálni, ami megmutatná, hogy hány éves egy tanuló?
9. Hogyan tudná egy lekérdezésben megjeleníteni az iskola 5 legjobb egy főre jutó átlagkeresetével rendelkező tanulójának nevét?
10. Mi tárolódik a lemezen, amikor mentünk egy lekérdezést? A forráshalmaz, vagy az eredményhalmaz?

# 9. SPECIÁLIS FELADATOK LEKÉRDEZÉSEKKEL

## 9.1 CÉLKITŰZÉS

Az eddig megszerzett ismeretek alapján beláthatjuk, hogy a gyakorlati alkalmazásban, lekérdezések az Access talán leghasznosabb komponensei. A lekérdezésekkel az adatbázisban tárolt adatokra, azok tartalmára, kapcsolataira, összefüggéseire vonatkozó legkülönbözőbb kérdéseket tudjuk megválaszolni. Ebben a leckében kitejesítjük a lekérdezésekre vonatkozó tudásunkat. Eddig úgynevezett választó lekérdezéseket készítettünk, amelyek egy rekordforrásból mezőket és rekordokat válogattak ki.

Az Accessben a rekordok kiválogatásán kívül, jó néhány egyéb feladatot végezhetünk el lekérdezésekkel. Ezekre a feladatokra további lekérdezés típusok állnak rendelkezésre.

Ebben a leckében áttekintjük a lekérdezés típusokat, majd részletesen megvizsgáljuk az egyes típusok alkalmazásának lehetőségeit.

Igyekezzen megjegyezni:

Hogyan csoportosítjuk a lekérdezéseket?

Milyen feladatokra alkalmazhatók az egyes lekérdezés típusok?

Hogyan készíthetünk statisztikai elemzéseket az adatbázisban tárolt adatok alapján?

Milyen technikák állnak rendelkezésre a rekordok tömeges feldolgozására?

## 9.2 A LECKE TÉMAKÖREI

- Lekérdezések típusai
- Paraméteres lekérdezések
- Összesítő lekérdezések
- Kereszttáblás lekérdezések
- Akció lekérdezések

## 9.3 LEKÉRDEZÉSEK TÍPUSAI

Az Accessben alapvetően az **egyszerű** és az **összetett** csoportokba soroljuk a lekérdezéseket. Az összetett lekérdezések azért alkotnak külön kategóriát, mert általában egy egyszerű lekérdezés által előállított rekordhalmazt dolgoznak fel.

A lekérdezések típusait az alábbi ábra mutatja be.




*54. ábra Lekérdezések típusai*

### 9.3.1 Egyszerű lekérdezések

Az egyszerű lekérdezések közé tartoznak a választó, a paraméteres, és az összesítő (összegző) lekérdezések.

#### *Választó lekérdezések:*

Rekordokat és mezőket válogatnak ki egy rekordforrás által meghatározott rekordhalmazból. A legelemibb, ugyanakkor **legfontosabb lekérdezés típus**, ugyanis minden további lekérdezés alapját képezi.

> Bármilyen további lekérdezést is készítünk, első lépésként egy választó lekérdezést kell kialakítanunk.

Az előző tananyagban megismert lekérdezések mindegyike választó lekérdezés volt. Bár az egyszerű jelzővel illetjük őket, láthattuk, hogy elkészítésük gyakran egészen komoly ismereteket kíván. Csupán azért nevezzük őket "egyszerű" lekérdezésnek, mert minden további lekérdezés alapját szolgáltatják.

#### *Paraméteres lekérdezés:*

Olyan lekérdezések, amelyek tervében az **eredményhalmazt befolyásoló változókat,** úgynevezett paramétereket használhatunk. A paraméteres lekérdezésekben a lekérdezés készítője változók felhasználásával hozhat létre számított mezőket, illetve adhat meg feltételeket. A változók értéke nem tárolódik a lekérdezésben. Az értékeket a lekérdezés használója közvetlenül a felhasználói nézet megtekintés előtt megjelenő párbeszédablakokban adhatja meg.

Paraméterek segítségével olyan lekérdezéseket készíthetünk, amelyek a felhasználó által megadott különböző értékektől függően más-más eredményhalmazt állítanak elő.

Bár önálló típusként emlegetjük őket, fontos tudni, hogy paramétereket bármely további lekérdezés típusban alkalmazhatunk.

### *Összesítő lekérdezések:*

Az összesítő lekérdezések, **statisztikák készítésekor** nyújtanak nagy segítséget. Egy választó lekérdezéssel előállított eredményhalmaz rekordjaiban, egy vagy több mező felhasználásával statisztikai műveletet végezhetünk el. Kiszámolhatjuk egy rekordhalmaz valamelyik mezőjében tárolt értékek összegét, átlagát, szórását, varianciáját, megkereshetjük a legkisebb, legnagyobb értéket, vagy egyszerűen meghatározhatjuk a mezőben tárolódó különböző értékek számát.

Lehetőség van arra is, hogy a számítást a választó lekérdezés eredményhalmazának összes rekordját felhasználva végezzük el, de arra is, hogy a rekordokat előbb részhalmazokba soroljuk, majd ezekben a **részhalmazokban** végezzük el a számítást.

### *Összetett lekérdezések*

Az összetett lekérdezések szintén három csoportba sorolhatók.

### *Kereszttáblás lekérdezések:*

Ezek a lekérdezések nagyon hasonlítanak az összesítő lekérdezésekhez. Funkciójukat tekintve nem is különböznek azoktól. Az összesítő lekérdezésekkel kialakított rekordhalmazt azonban jóval áttekinthetőbb, kimutatás formában jelenítik meg.

### *Akció lekérdezések:*

Ezek a lekérdezések egy választó lekérdezés által előállított eredményhalmaz rekordjaival **tömeges rekordkezelő** műveletek hajtanak végre. Az elvégzett rekordkezelő művelet alapján négyféle akció lekérdezést különítünk el.

#### A táblakészítő lekérdezés

Az eredményhalmaz rekordjai számára új táblát készít, majd a rekordokat bemásolja a táblába.

#### Hozzáfűző lekérdezés:

Lemásolja a választó lekérdezéssel létrehozott rekordhalmazt, majd a rekordokat hozzáfűzi egy meglévő tábla rekordjaihoz.

#### Frissítő lekérdezés:

Egy választó lekérdezéssel kiválogatott rekordhalmaz minden rekordjában megváltoztatja egy (esetleg több) mező értékét.

**Törlő lekérdezés:**

Nagy tömegű rekord törlését teszi lehetővé egy lépésben. A választó lekérdezéssel kiválogatott rekordokat törölhetjük az adatbázisból.

### *SQL specifikus lekérdezések:*

Az összetett lekérdezések harmadik csoportjába tartozó SQL specifikus lekérdezések abban különböznek a többi lekérdezéstől, hogy ezek nem hozhatók létre az eddig megismert lekérdezés tervező ablakban. Kialakításuk kizárólag SQL nyelven leírt, úgynevezett SQL mondatok segítségével történhet. Az SQL, a relációs adatbázis kezelő rendszerek által ismert adatkezelő nyelv, amelynek segítségével a programok irányíthatók.

Az SQL specifikus lekérdezések elkészítésével csak az SQL nyelv megismerése után foglalkozhatunk. Mivel tananyagunk nem terjed ki az SQL nyelv ismertetésére, csupán felsorolás szerűen említjük meg az ide tartozó altípusokat, és azok funkcióit.

### *Adatdefiníciós lekérdezés:*

Ezzel a lekérdezés fajtával táblák kialakítására nyílik lehetőség.

### *Egyesítő lekérdezés:*

Segítségével két választó lekérdezés eredményhalmazát egyetlen eredményhalmazzá egyesíthetjük.

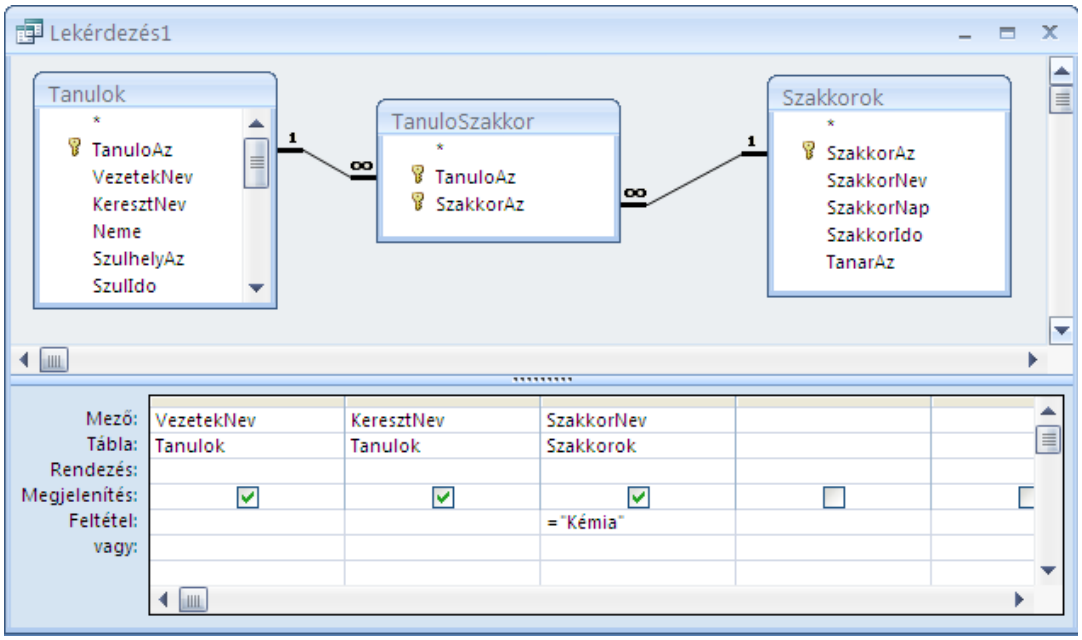
### *Átadó lekérdezés:*

Ezzel a lekérdezés típussal adatbázisok közötti kapcsolat valósítható meg. Az SQL-ben leírt adatkezelő utasítást az Access, hálózaton (pl. Internet) egy másik gép valamilyen adatbázis-kezelő rendszerének küldi el. A távoli adatbázis-kezelő végrajtja a feladatot, majd az eredmény visszaküldi az Accessnek.

Az átadó lekérdezésekkel tehát távoli adatbázisok adatai kezelhetők.

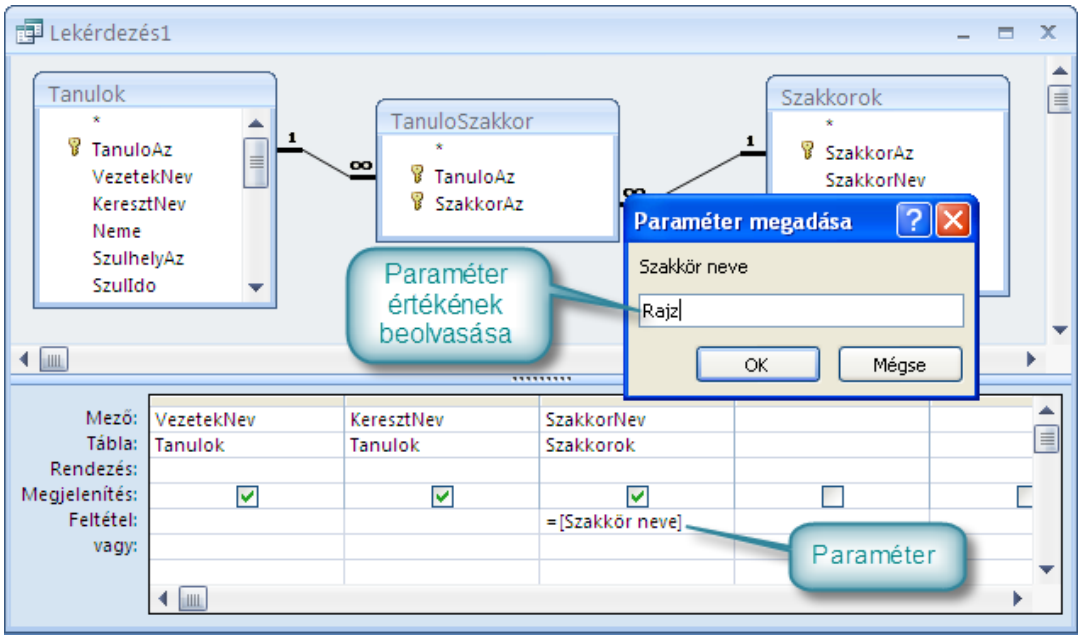
## 9.4 PARAMÉTERES LEKÉRDEZÉSEK

Az eddig megismert lekérdezések eredményhalmazát a felhasználó nem tudta befolyásolni. Az alábbi lekérdezés például mindig a „kémia” szakkörbe járó gyerekek adatait jeleníteti meg. Ha másik szakkör névsorát akarjuk a kiválogatni, akkor módosítanunk kell a lekérdezés tervét.




*55. ábra Kémia szakkörös gyerekek kiválogatása*

A paraméterek segítségével olyan lekérdezések állíthatók elő, amelyek eredményhalmazát a terv megváltoztatása nélkül is tudjuk befolyásolni. Ehhez nem kell mást tennünk, mint hogy a feltételben szereplő értéket egy paraméterrel helyettesítjük.




*56. ábra Paraméteres lekérdezés*

**A paraméter valójában egy olyan névvel rendelkező változó, amelynek értéke nem szerepel a lekérdezés tervében.**

A paraméter értékét csak akkor adjuk meg, amikor felhasználói nézetbe váltunk. A paraméterek nevét feltételben, vagy számított mezőben használhatjuk, pontosan úgy, mintha azok mezők lennének. A paraméterneveket a lekérdezés tervezésekor építjük be a lekérdezésbe.

A paraméterek nevének megválasztásakor be kell tartani az Access névadási szabályait! A nevek nem egyezhetnek meg a rekordforrásban szereplő táblák egyetlen mezők, sőt a lekérdezésben esetleg szereplő számított mezők neveivel sem!

### 9.4.1 Paraméteres lekérdezés kialakítása

A fenti példában a [***Szakkör neve***] paramétert feltételben használjuk. Amikor a lekérdezést végrehajtjuk, az Access megkérdezi a paraméter értékét. A begépelt adatot behelyettesíti a feltételbe, majd annak megfelelően válogatja ki a rekordokat.

A paraméter értéke nem tárolódik a lekérdezésben, ezért ismételt végrehajtásakor újra be kell gépelnünk.

### 9.4.2 Paraméterek deklarálása

Ha paramétereket szeretnénk használni, akkor azokat deklarálnunk kell. A deklarálás azt jelenti, hogy jelezzük a paraméter használat szándékát. A deklarálás egyik módja, hogy egyszerűen el kezdjük használni a paramétert, azaz hivatkozunk rá feltételben, vagy számításban. Az ilyen deklarálást **implicit**, „magától értetődő” deklarációnak nevezzük.

Az eddigi példákban implicit deklarációt alkalmaztunk.

A másik lehetőség az **explicit**, „határozott” deklaráció, amikor a paraméter használata előtt megadjuk annak **nevét** és **típusát**. Erre a menüszalag **Tervezés** lapján, a **Megjelenítés/elrejtés** csoport **Paraméterek** parancsával megnyitható ablakban van lehetőség.

Az implicit deklaráció egyszerűbben megvalósítható, de nem ad lehetőséget a paraméter típusának meghatározására. Az explicit deklaráció esetén, az Access csak a deklarációban megadott típusú értéket fogad el.

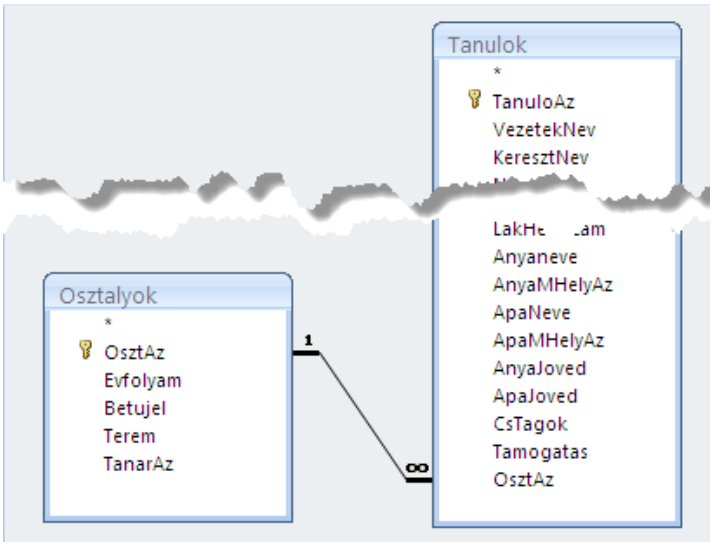
## 9.5 ÖSSZESÍTŐ LEKÉRDEZÉSEK:

Az összesítő lekérdezések segítségével egy rekordhalmaz részhalmazaiban, egy vagy több mező értékeivel statisztikai művelet végezhető el. Számolhatunk például osztálylétszámokat, tantárgyi tanulmányi átlagokat, évfolyam szintű összesített átlagot, osztályok átlagos egy főre jutó keresetét stb. … Az összesítő lekérdezésekkel rendkívül sok fárasztó számolástól kímélheti meg magát az adatbázis fejlesztője.

Az alábbi videó felvételen megnézheti, hogyan készítünk olyan összesítő lekérdezést, ami meghatározza az osztályok létszámát.

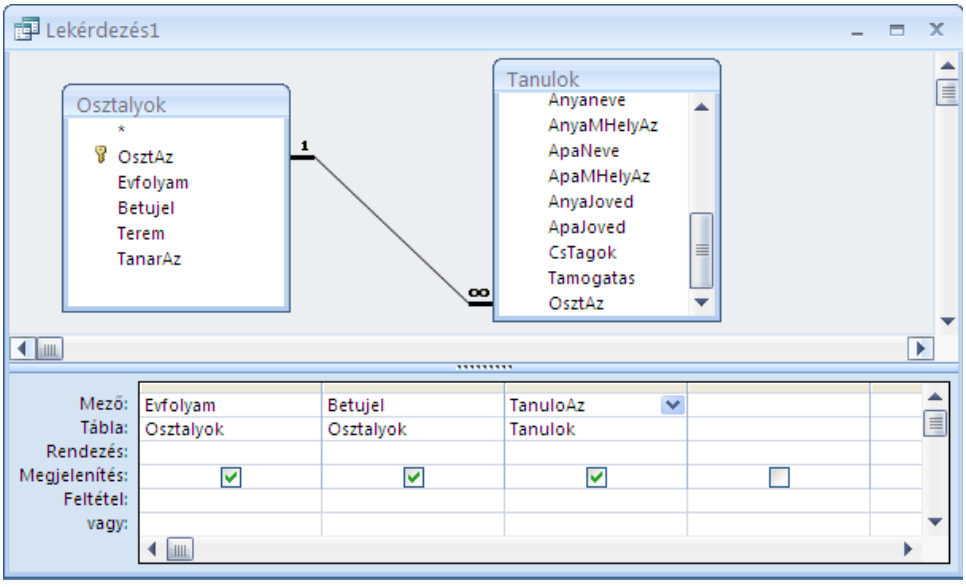
### 9.5.1 Összesítő lekérdezés készítése lépésenként

Az összesítő lekérdezések kialakításának első lépése a megfelelő választó lekérdezés elkészítése. Ennek eredményhalmazában minden rekordnak szerepelnie kell, amelyet be szeretnénk vonni a statisztikai műveletbe.



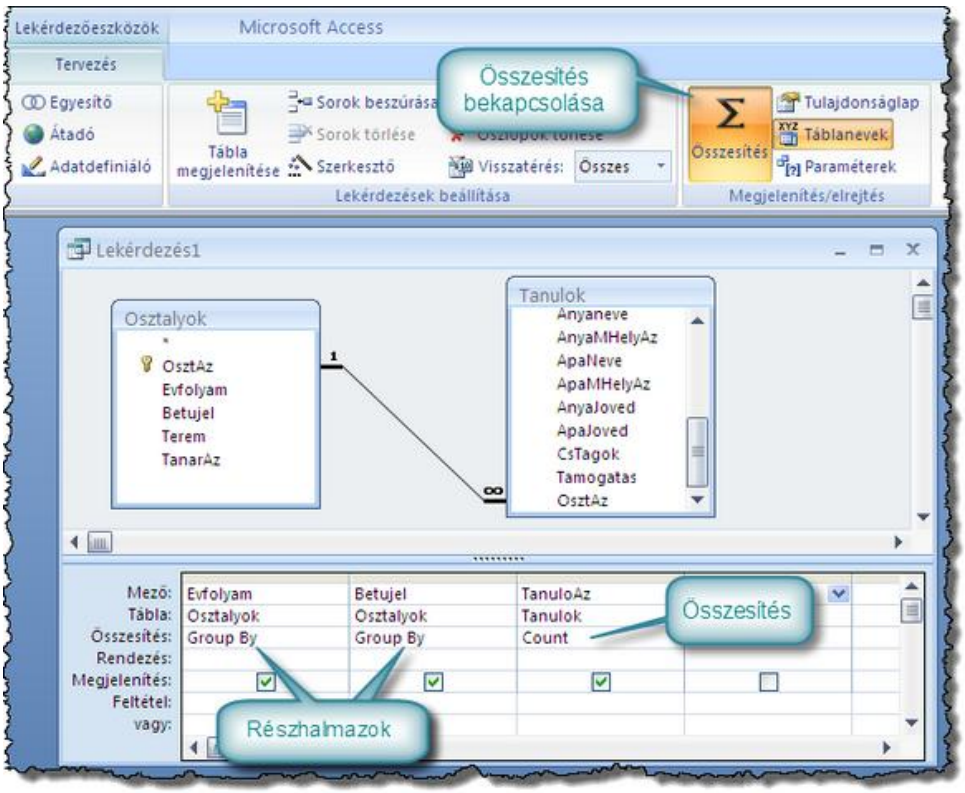

*57. ábra Lekérdezés rekordforrása*

A tervezőrácson azokat a mezőket kell elhelyezni, amelyekkel számolni akarunk, illetve azokat, amelyek értéke szerint részhalmazokat akarunk kialakítani.




*10. kép Mezők a tervezőrácson*

A menüszalag **Tervezés** lapján, a **Megjelenítés/elrejtés** csoport **Összesítés** parancsával Át kell váltanunk összesítő lekérdezésre.



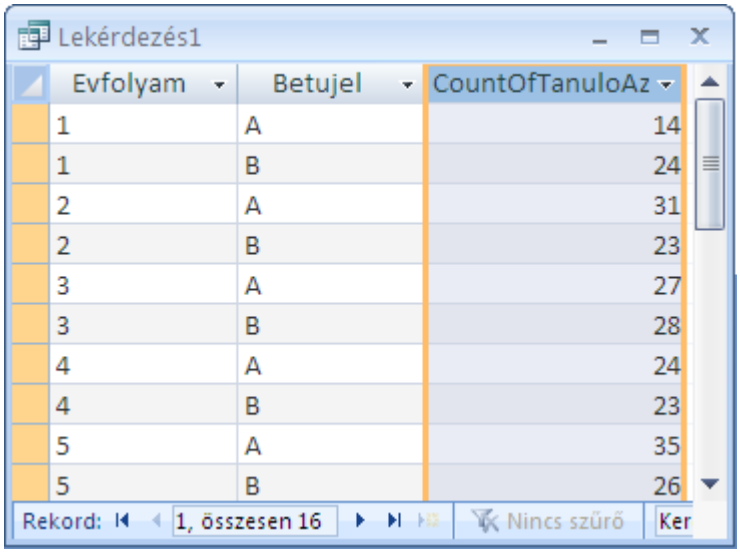

*58. ábra Összesítő lekérdezés terve*

Ezt követően a tervezőrács Összesítés sorában kiválasztott összesítő függvénnyel meg kell adnunk, hogy az eredményhalmaz melyik mezőjével, milyen statisztikai műveltet szeretnénk elvégezni.

Az összesítő lekérdezésekben az alábbi összesítő függvények segítségével tudjuk elvégezni a statisztikai műveleteket:

| Összesítő függvény | Művelet |
|---|---|
| Avg | Átlagszámítás |
| Sum | Összegzés |
| Count | Megszámlálás |
| Min, Max | Minimum, maximum érték |
| First, Last | Első, utolsó érték |
| Stdev | Szórás |
| Var | Variancia |

Ha a műveletet a teljes rekordhalmaz részhalmazaiban lévő rekordokkal szeretnénk elvégezni, akkor a részhalmazok kialakításra mezők alatti Összesítés sorban a **Group by** opciót kell elhelyeznünk.



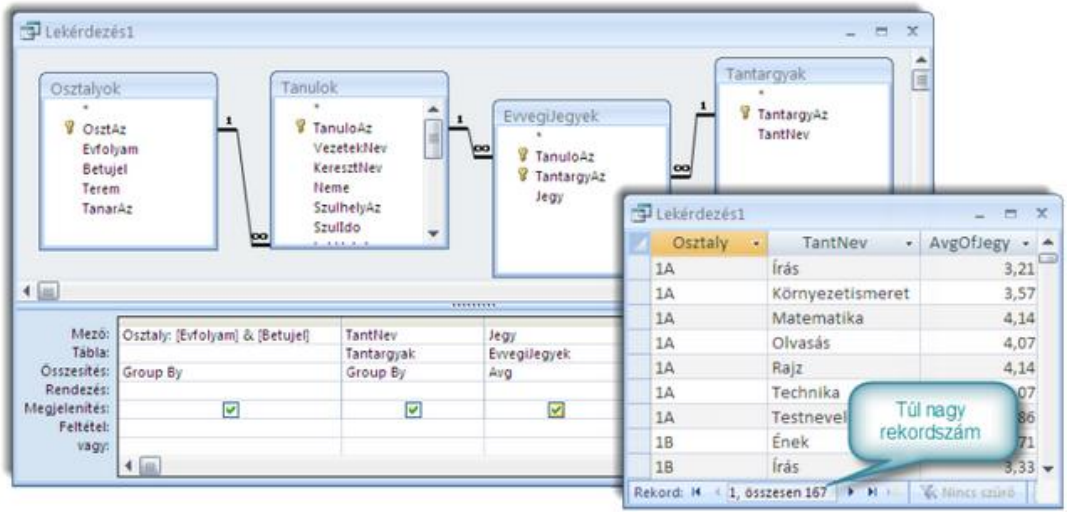

*11. kép Eredményhalmaz*

## 9.6 KERESZTTÁBLÁS LEKÉRDEZÉSEK

Az összesítő lekérdezésekkel roppant hasznos számításokat végezhetünk el percek alatt. Eredményhalmazuk azonban gyakran sok rekordot tartalmaz, ezért nehezen áttekinthető.

A kereszttáblás lekérdezésekkel, összesítő lekérdezések eredményhalmazának áttekinthető, kimutatásszerű változatát készíthetjük el. Használatukra akkor van lehetőség, ha az összesítő lekérdezés rekordjait legalább két mező szerint soroltuk részhalmazokba. Kereszttáblás lekérdezéssel elérhetjük, hogy az összesítő lekérdezés eredményhalmazában szereplő számítási értékek ne különálló rekordokban, hanem a részhalmaz képező mezők értékei szerint, sorokba és oszlopokba rendezve jelenjenek meg.

Ha például a tanulók osztályainak, és tantárgyak nevének megfelelően részhalmazokba sorolt érdemjegyeket átlagát, az osztályonkénti, tantárgyi átlagokat határozzuk meg, akkor az átlagok nagyon sok rekordban jelennek meg, így nehezen áttekinthetőek.




*59. ábra Összesítő lekérdezés*

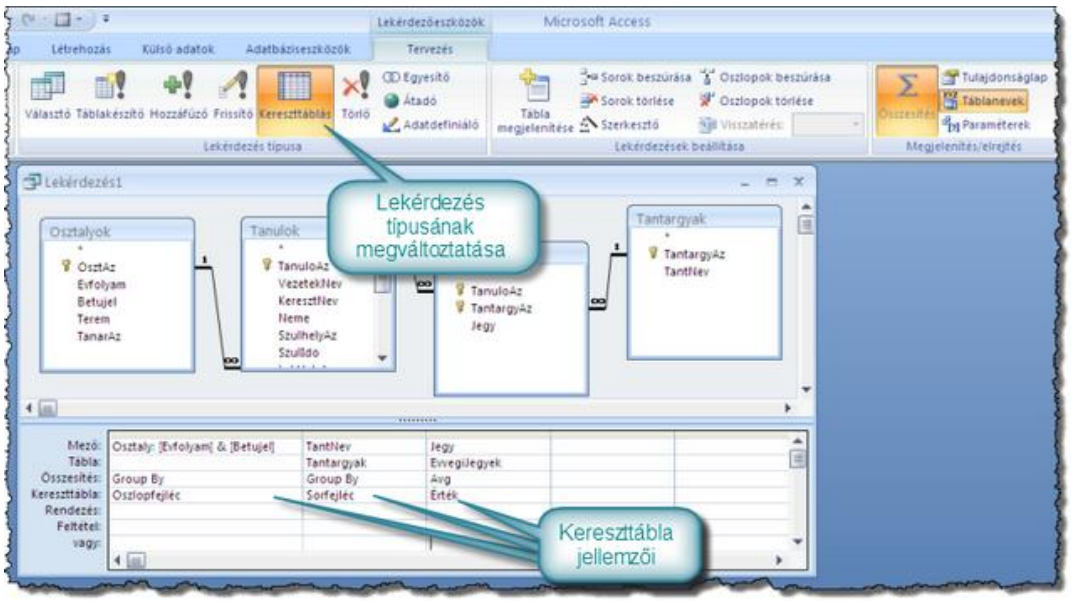
Kereszttáblás lekérdezéssel elérhetjük egy olyan tábla kialakítását, amely oszlopainak tetején (oszlopfejléc) osztályok nevei, a sorok elején (sorfejléc) pedig tartárgyak nevei látszanak, az átlagok pedig a megfelelő sor és oszlop pozícióban jelennek meg.

### 9.6.1 A kereszttáblás lekérdezés elkészítése

Kereszttáblás lekérdezés elkészítéséhez elsőként olyan összesítő lekérdezést kell létrehoznunk, amelyben legalább két részhalmazok képzésére használt mező, és egy számításra használt mező szerepel.

Ezt követően meg kell változtatni a lekérdezés típusát, azaz jelezni kell, hogy a lekérdezést kereszttáblás lekérdezéssé akarjuk alakítani. Ezt a menüszalag **Lekérdezés típusa** csoportjában, a **Kereszttáblás** gombbal tehetjük meg.

A tervező rácson megjelenő **Kereszttábla** sorban meg kell adnunk, melyik mező szeretnénk sorfejlécként, melyiket oszlopfejlécként használni, és melyik mező jelenjen meg a sorok és oszlopok metszésében értékként.




*60. ábra Kereszttáblás lekérdezés terve*

Az eredményhalmaz megtekintésekor még beállíthatjuk az oszlopszélességeket, így akár egyetlen pillantással áttekinthetjük a számítás eredményeit.

Lekérdezés1

| TantNev | 1A | 1B | 2A | 2B | 3A | 3B | 4A | 4B | 5A | 5B | 6A | 6B | 7A | 7B | 8A | 8B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Biológia | | | | | | | | | | | 3,48 | 3,68 | 3,57 | 3,79 | 4,04 | 3,64 |
| Ének | 3,57 | 3,71 | 3,65 | 3,83 | 3,85 | 4,11 | 3,92 | 3,78 | 3,31 | 3,42 | 3,33 | 3,77 | 3,62 | 4,00 | 3,89 | 4,00 |
| Fizika | | | | | | | | | | | | | 3,76 | 3,45 | 3,67 | 3,79 |
| Földrajz | | | | | | | | | | | 3,38 | 3,64 | 4,00 | 3,52 | 3,63 | 3,48 |
| Idegen nyelv | | | | | | | | | 3,60 | 3,96 | 3,81 | 3,95 | 3,81 | 3,38 | 3,30 | 3,44 |
| Írás | 3,21 | 3,33 | 3,42 | 3,78 | 3,63 | 3,61 | 3,67 | 3,91 | | | | | | | | |
| Kémia | | | | | | | | | | | | | 3,48 | 3,34 | 4,00 | 3,46 |
| Környezetismeret | 3,57 | 3,50 | 3,74 | 3,30 | 3,63 | 3,57 | 3,96 | 3,65 | 4,03 | 3,77 | | | 5,00 | | | |
| Magyar irodalom | | | | | | | | | 3,51 | 3,96 | 3,62 | 3,77 | 3,33 | 3,10 | 4,00 | 3,60 |
| Magyar nyelv | | | | | | | | | 3,69 | 3,54 | 4,05 | 3,59 | 3,62 | 3,48 | 4,11 | 3,64 |
| Matematika | 4,14 | 3,50 | 3,45 | 3,96 | 4,04 | 3,79 | 3,38 | 3,87 | 4,03 | 3,88 | 3,71 | 3,59 | 3,52 | 3,79 | 4,00 | 3,84 |
| Olvasás | 4,07 | 3,42 | 3,58 | 3,83 | 3,93 | 4,32 | 4,04 | 3,65 | | | | | | | | |
| Rajz | 4,14 | 3,63 | 3,55 | 3,43 | 3,67 | 3,71 | 4,00 | 3,65 | 3,43 | 3,88 | 3,81 | 3,95 | 3,95 | 3,97 | 3,37 | 3,76 |
| Számítástechnika | | | | | | | | | 3,49 | 3,62 | 4,00 | 3,27 | 3,86 | 3,31 | 3,52 | 3,60 |
| Technika | 4,07 | 3,71 | 3,97 | 3,61 | 3,81 | 3,50 | 3,83 | 3,83 | 3,97 | 3,92 | 3,81 | 4,00 | 3,57 | 3,66 | 3,63 | 3,84 |
| Testnevelés | 3,86 | 3,71 | 3,52 | 3,78 | 3,44 | 3,71 | 4,04 | 4,00 | 3,49 | 3,62 | 3,38 | 4,09 | 3,71 | 3,41 | 2,96 | 3,84 |
| Történelem | | | | | | | | | 3,74 | 3,92 | 3,62 | 4,14 | 3,57 | 3,90 | 3,07 | 3,76 |

Rekord: 17, összesen 17 | Nincs szűrő | Keresés

*12. kép Kereszttáblás lekérdezés eredményhalmaza*

## 9.7 AKCIÓ LEKÉRDEZÉSEK

Mint tudjuk az akció lekérdezések valamilyen műveletet, akciót végeznek el egy választó lekérdezés eredményhalmazának minden egyes rekordjával. Kiválóan alkalmazhat-

juk őket nagy mennyiségű rekord gyors, automatikus feldolgozására. Hasznosságuk mellett azonban feltétlenül említést érdemel veszélyes mivoltuk is.

> Mivel egyszerre nagyon sok rekordot is feldolgozhatnak, hibás alkalmazásuk igen nagy károkat okozhat. Használatuk előtt ezért ajánlott az érintett táblákról, vagy akár az egész adatbázisról biztonsági másolatot készíteni.

### 9.7.1 Akció lekérdezések elkészítésének lépései

Az akció lekérdezéseken belül négy további altípus létezik, de ezek elkészítésnek lépései némileg hasonlóak. Minden akció lekérdezést az alábbi négy lépésben készítünk el.

#### *Választó lekérdezés készítése:*

A legelső és egyben legmeghatározóbb lépés, az akció lekérdezés által feldolgozott rekordokat szolgáltató választó lekérdezésnek az elkészítése.

#### *Lekérdezés típusának beállítása:*

Miután rendelkezésünkre állnak a megfelelő rekordok, meg kell mondanunk, milyen akciót szeretnék veleül elvégezni. Ezt a lekérdezés típusának beállításával határozzuk meg, a menüszalag **Lekérdezés típusa** csoportjában. Ebben a lépésben adjuk meg, hogy **Táblakészítő, Hozzáfűző, Frissítő,** vagy **Törlő** lekérdezést szeretnénk csinálni.

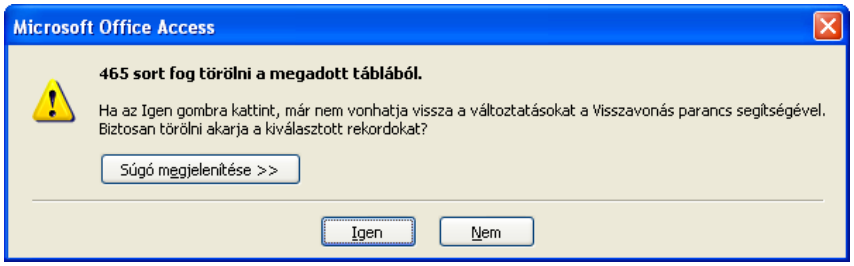
#### *Kiegészítő adatok megadása:*

Minden akció lekérdezésnek szüksége van speciális adatokra. A **Táblakészítő** lekérdezés esetén például azt kell megadni, hogy mi legyen az új tábla neve. Ezeket az adatokat állítjuk be a harmadik lépésben.

#### *Lekérdezés futtatása:*

Az elkészült akció lekérdezésben meghatározott művelet csak kerül végrehajtásra, ha erre felszólítjuk az adatbázis-kezelő rendszert. A lekérdezés futtatása ezt jelenti.

> Az adatbázisok megnyitásakor az adatbázisban lévő makrókat és modulokat az Access „veszélyforrásként” értékeli, és letiltja. Erről természetesen értesítést kapunk és a tiltást felül is bírálhatjuk. Ha azonban ezt elmulasztjuk, előfordulhat, hogy az akció lekérdezések nem fognak működni. Ilyenkor futtatásuk hatástalan.




*13. kép Biztonsági riasztás törlő lekérdezés futtatásakor*

### *Táblakészítő lekérdezés*

A táblakészítő lekérdezéssel egy választó lekérdezés eredményhalmazában megjelenő rekordok egy új, táblába másolódnak. Az új táblát a lekérdezés hozza létre.

Az alábbi példa azt mutatja be, hogyan hozunk létre új táblát a tanulók tábla 2001-ben született tanulónak rekordjai számára.

Az első lépésben egy választó lekérdezés segítségével kiválogatjuk a Tanulok táblából az 2001-ben született gyerekek rekordjait.

A második lépésben megváltoztatjuk a lekérdezés típusát **Táblakészítő** lekérdezésre.

A harmadik lépésben megadjuk az új tábla nevét: ***UjTabla***

A negyedik lépésben lefuttatjuk a lekérdezést, azaz végrehajtjuk az akciót.

Ha **Táblakészítő** lekérdezésben egy létező tábla nevét adjuk meg új táblaként, akkor figyelmeztetést kapunk. Ha ennek ellenére továbbhaladunk, akkor az Access előbb törli a létező táblát, majd ismét létrehozza azt.

### *Hozzáfűző lekérdezés*

A hozzáfűző lekérdezés szintén egy választó lekérdezés eredményhalmazával dolgozik. A kiválogatott rekordokat egy már meglévő tábla végéhez illeszti hozzá, azaz rekordokkal bővít egy már meglévő táblát.

Példánkban az imént létrehozott **UjTabla** táblához illesztjük hozzá a ***Tanulok*** tábla 2002-ben született tanulóinak adatait.

Hozzáfűző lekérdezés esetén ügyelnünk kell arra, hogy a választó lekérdezéssel kiválogatott mezők típusa megfeleljen a célzott tábla mezői típusának. Azaz a választó lekérdezéssel a célzott tábla szerkezetének megfelelő mezőszerkezetet kell kialakítanunk.

Ha egy hozzáfűző lekérdezést egymás után többször is lefuttatunk, akkor az Access minden alkalommal, ismételten elvégzi a tábla bővítését.

### *Frissítő lekérdezés*

A frissítő lekérdezéssel gyorsan és egyszerűen írhatjuk át egy mező értékét egy rekordhalmaz minden rekordjában.

Az imént használt ***UjTabla*** táblában találunk egy ***Tamogatas*** nevű mezőt, amely a tanuló eddig kapott tankönyvtámogatását tartalmazza. Tegyük fel, hogy 5000 Ft-os összeget kap minden olyan gyerek, aki eddig még nem részesült támogatásban, és a család egy főre jutó átlagkeresete kisebb, mint a 2010-ben törvényes minimálbér fele, azaz 73500/2 Ft A támogatást be kell jegyezni az ***UjTabla*** táblába!

A Táblakészítő és Hozzáfűző lekérdezések hatása a feldolgozott rekordokat biztosító táblákban semmit sem változtattak. Vagy egy új táblát hoztak létre, vagy egy meglévő, másik táblához illesztettek hozzá rekordokat. A frissítő lekérdezés abban a táblában változtatja meg a rekordokat, amelyikből azokat kiválogattuk!

### 9.7.2 Törlő lekérdezés

Az utolsó akciólekérdezéssel egyszerűen törölhetünk olyan rekordokat, amelyekre már nincs szükségünk az adatbázisban. A rekordokat egy választó lekérdezéssel kell kiválogatnunk. Bár a lekérdezés rekordforrásban több tábla is lehet, a törlés ezek közül csak az egyikre vonatkozhat.

A választó lekérdezést úgy kell elkészíteni, hogy a törlendő rekordokat tartalmazó tábla mezőlistájából az összes mezőt szimbolizáló *-ot kell a rácsra kerüljön. Amikor megváltoztatjuk a lekérdezés típusát **Törlő** lekérdezésre, a tervező rácson megjelenő **Törlés** sorban, a * mező alatt a **From**, a többi mező alatt a **Where** értékeket kell kiválasztani.

Ha a rekordok törlésekor használt tábla egy kapcsolat elsődleges táblája, és a kapcsolatban be van kapcsolva a kaszkádolt törlés, akkor a törlő lekérdezés futtatásakor a kapcsolódó táblából is törlődhetnek rekordok.

## 9.8 ÖSSZEFOGLALÁS

A legelemibb, mégis **legfontosabb**, minden további lekérdezés alapját képező **lekérdezés típus** a választó lekérdezés, rekordok és mezők kiválasztására szolgál.

A paraméteres lekérdezésekben változók teszik lehetővé, hogy a felhasználó befolyásolja az eredményhalmazt.

Az összegző és kereszttáblás lekérdezésekkel statisztikai műveletek, az akció lekérdezésekkel pedig tömeges rekordkezelés végezhető el.

## 9.9 ÖNELLENŐRZŐ KÉRDÉSEK

1. Miért mondhatjuk, hogy a választó lekérdezések ismerete a legfontosabb?
2. Mit ért a paraméterek explicit deklarálása alatt?
3. Mi történik, ha egy lekérdezésben számított mezőt készít, és hibásan gépeli be egy mező nevét, például [***ApaJoved***] helyett [***ApaJövedeme***] mezőnevet használ? **Osszker: [ApaJövedelem] + [AnyaJoved]**
4. Mit tehet ilyenkor?
5. Mikor alakíthat egy összesítő lekérdezést kereszttáblás lekérdezéssé?
6. Melyik lekérdezés csoportba tartoznak, és mire használhatók az átadó lekérdezések?
7. Milyen lépésekben hozhat létre akció lekérdezéseket?
8. Mikor nem tudja futtatni az akció lekérdezéseket?
9. Milyen biztonsági lépéseket tehet, mielőtt akció lekérdezést futtat?

# 10. AZ ACCESS ADATBÁZISOK KEZELŐFELÜLETÉNEK KIALAKÍTÁSA

## 10.1 CÉLKITŰZÉS

Ha lekérdezésekről azt mondtuk, hogy az Microsoft Access legizgalmasabb objektumai, akkor az űrlapokról nyugodtan kijelenthetjük, hogy az Access adatbázis leglátványosabb elemei. Segítségükkel kezelő felületet alakíthatunk ki az adatbázishoz, „eltakarhatjuk" az adatbázis tábláit, lekérdezéseit, kapcsolatait a felhasználók elől. Az űrlapok alkalmazásával biztosíthatjuk, hogy azok a felhasználók is hatékonyan tudjanak dolgozni az adatbázissal, akik nem rendelkeznek kellő ismeretekkel az idegen kulcsokkal összekapcsolt táblák, kezeléséhez, vagy a lekérdezések készítéséhez.

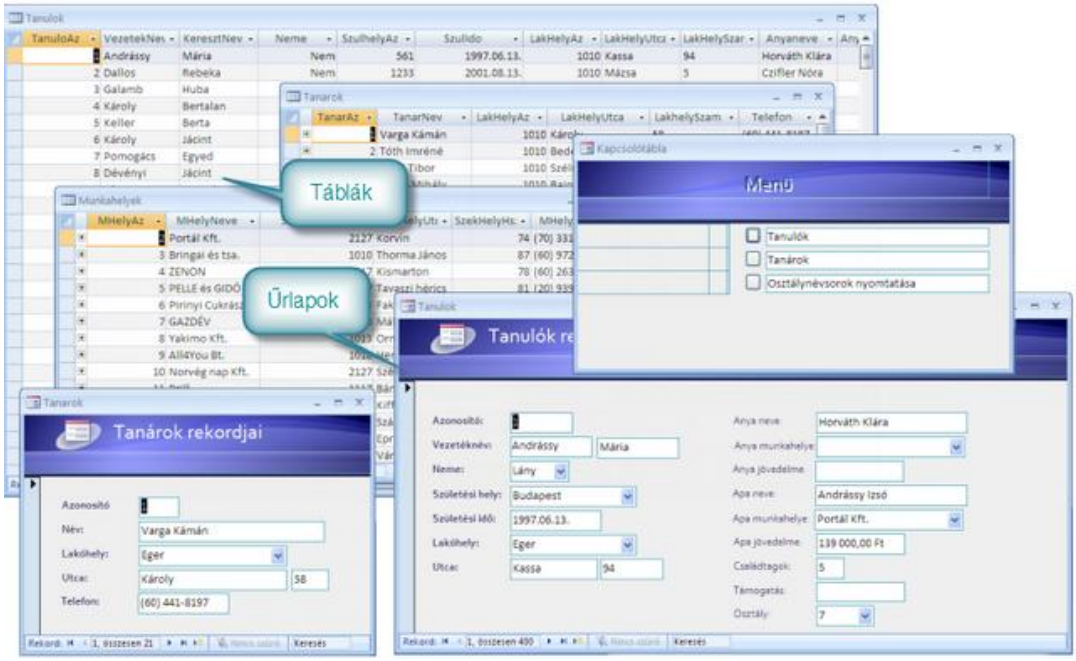
Az űrlapokkal arcot adhatunk adatbázisunknak, megformálhatjuk a kezelőfelület küllemét, kényelmesebbe tehetjük, egyben fel is gyorsíthatjuk az adat kezelést.

Az űrlapok rendkívül sokoldalú objektumok. Használhatjuk őket tábláink, lekérdezéseink rekordjainak megjelenítésére, a felhasználóval való interakciók lebonyolítására, az adatbázis alkalmazássá fejlesztésére. Ha röviden kellene megfogalmaznunk funkciójukat, azt mondhatnánk, hogy az űrlapok grafikus felületet nyújtanak az adatbázis kezelésére.

Ebben a leckében megtanulhatja, hogyan jelenítheti meg, és teheti egyszerűen kezelhetővé a táblák, vagy lekérdezések eredményhalmazának rekordjait, hogyan érheti el, hogy az elkészült adatbázis felhasználói egyszerű, ergonomikus, és nem utolsó sorban szép felületeken kezelhessék az adatbázist.

## 10.2 A LECKE TÉMAKÖREI

- Az űrlapok felhasználási lehetőségei
- Űrlapok megjelenése
- Űrlapok készítése
- Vezérlőelemek az űrlapokon
- Speciális mezőértékek megjelenítése
- Az adatbázis menüablakának elkészítése




*61. ábra Táblák és űrlapok*

## 10.3 AZ ŰRLAPOK FELHASZNÁLÁSI LEHETŐSÉGEI

1. Az űrlapok elsődleges feladata az adatkezelés. Egy űrlapon táblákban tárolt rekordok, vagy lekérdezések eredményhalmazaiban lévő rekordok mezőinek értékei jeleníthetők meg. Az adtok megtekintésén, túl azok módosítására, törlésére, beszúrására is van lehetőségünk, ugyanúgy mintha táblákat használnánk. Az adatkezelés azonban jól áttekinthető, biztonságosan kezelhető grafikus felületen végezhetjük el.

3. Előre programozott módon reagálhatunk a felhasználó tevékenységére. Elfogadhatjuk, vagy elvethetjük a begépelt adatokat, kérdéseket tehetünk fel a felhasználónak, az általa megadott adatokkal számításokat végezhetünk, vagy akár programkódokkal is feldolgozhatjuk azokat.

4. Bár elsősorban adatmegjelenítésre használhatók, kialakíthatunk olyan űrlapokat is, amelyek egyfajta menüablakként, más űrlapok megjelenítésében segítenek. Ha menü űrlapot készítünk, a felhasználónak még a navigációs ablak kezelését sem kell ismernie, mert az adatok kezelését biztosító űrlapokat menü űrlapról érheti el.

## 10.4 AZ ŰRLAPOK MEGJELENÉSE

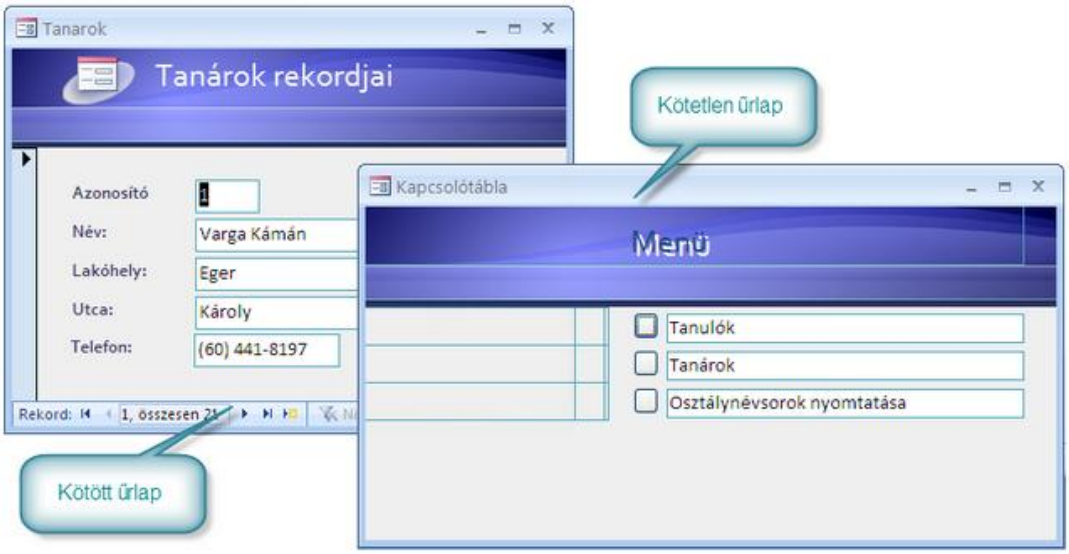
A táblákhoz és lekérdezésekhez hasonlóan, az űrlapok is tervező és felhasználói nézettel rendelkeznek. Felhasználói nézetük azonban – a sokoldalú felhasználhatóságnak megfelelően – többféle is lehet. az űrlapok beállíthatók úgy, hogy felhasználói nézetben egyszerre csak egy rekord mezőit jelenítsék meg, vagy akár úgy is, hogy több rekord, egymás alatt folyamatosan helyezkedjen el. Az első estben **egyszeres**, a második estben pedig **folyamatos** űrlapról beszélünk.

Ha szükséges az űrlapok, a táblák és lekérdezések esetében használt **adatlap** formában is megjelenhetnek. Szerkezetüket felépíthetjük úgy, hogy azon egyetlen rekordforrás adatai legyenek láthatók, de **segédűrlapok** segítségével akár több, kapcsolatban lévő tábla űrlapjait is egymásba ágyazhatjuk.

## 10.5 ŰRLAPOK ÉS ADATOK

Az űrlapok küllemén kívül, aszerint is osztályozhatjuk őket, hogy kapcsolódnak-e az adatbázis valamelyik táblájához, lekérdezéséhez, azaz rendelkeznek-e *rekordforrással*. Azt az űrlapot, amely egy rekordforráshoz kapcsolódik, és annak adatait teszi kezelhetővé, **kötött űrlapnak** nevezzük, az adatforráshoz nem kapcsolódó űrlapokat pedig **kötetlen űrlapoknak** hívjuk.

A kötött űrlapok segítségével eltakarjuk a felhasználó elől a táblákat, és kezelésüket az űrlap felületén tesszük lehetővé. A kötetlen űrlapokkal elsősorban a többi űrlap közötti navigációt valósíthatunk meg.




*62. ábra Kötött és kötetlen űrlap*

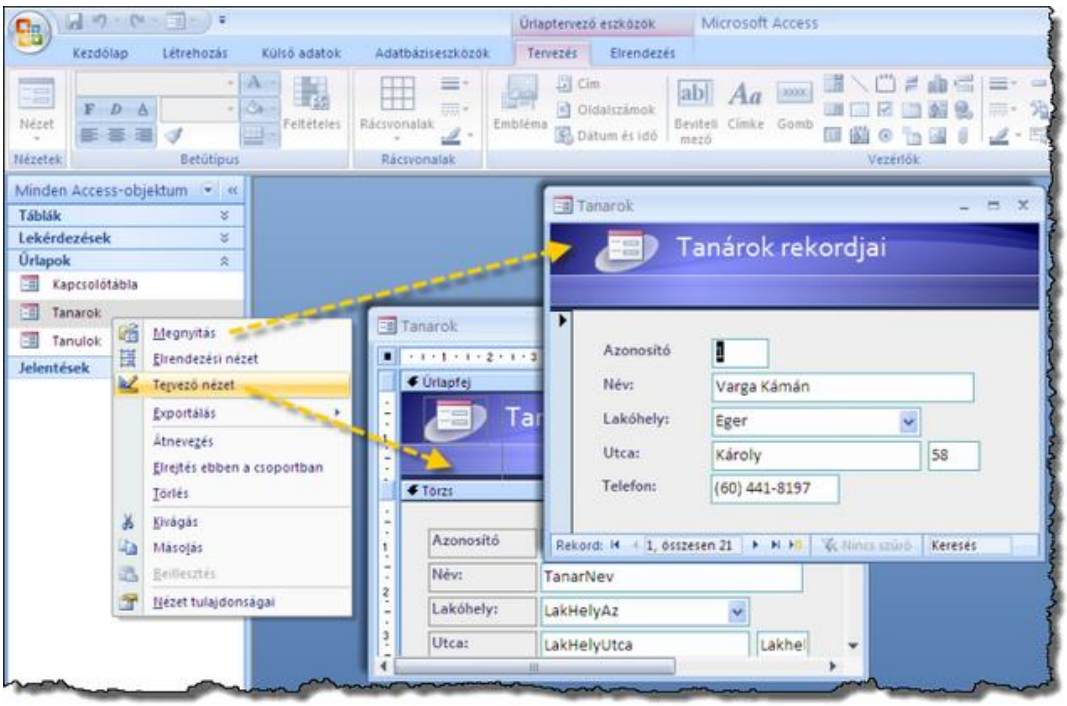
## 10.6 MŰVELETEK ŰRLAPOKKAL

Mielőtt saját űrlapjainkat elkészítenénk, ejtsünk néhány szót az űrlapokkal elvégezhető műveletekről. Az űrlapok kezelésekor a navigációs ablakban érdemes láthatóvá tenni az **Űrlap** objektumcsoportot. A többi objektumhoz hasonlóan itt tudjuk megnyitni a már létrehozott űrlapokat, mégpedig háromféle különféle nézetben

- Tervező nézet: egy új űrlap létrehozásakor, illetve egy már létező űrlap megváltoztatásakor használjuk.
- Űrlap nézet: a megtervezett űrlapot, a felhasználónak szánt formában mutatja meg. ez az űrlapok felhasználói nézete.

– Elrendezés nézet: ebben a nézetben megváltoztathatja az űrlapok szerkezetét, miközben az adataikat megtekinti. Ez a nézet valójában az előző kettő funkcióit együtt valósítja meg.

Az egyes nézeteket űrlap helyi menüjének **Tervező nézet**, **Megnyitás**, és **Elrendezés nézet** parancsival választhatjuk ki. A dupla kattintás a **Megnyitás** parancsnak felel meg.




*63. ábra Űrlap megnyitása*

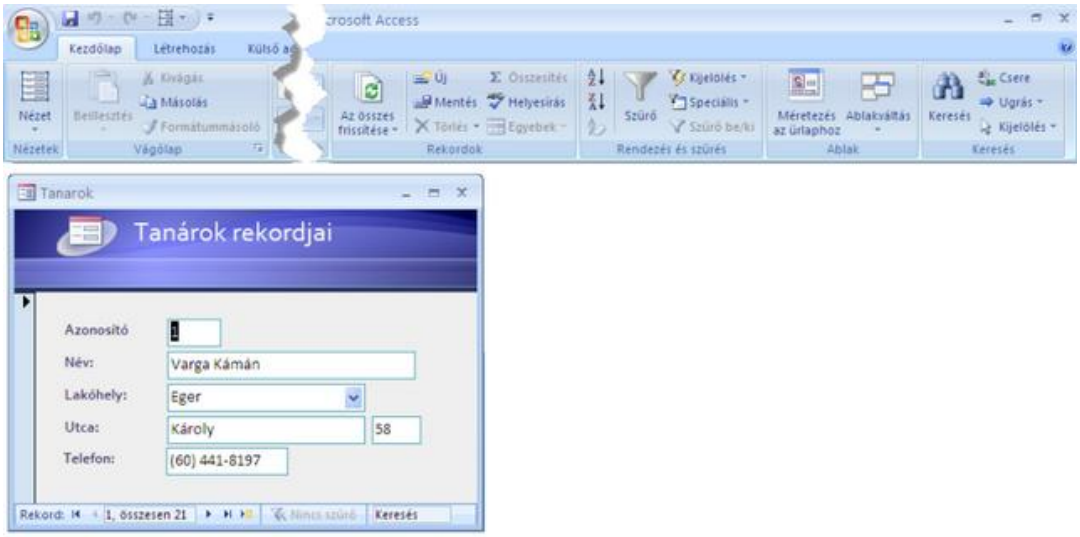
## 10.7 ŰRLAPOK HASZNÁLATA

Ha az űrlapot használni akarjuk, el akarjuk vele végezni mindazokat a műveleteket, amelyekre azt tervezték, az űrlapot **Űrlap** nézetben kell megnyitnunk.

### 10.7.1 Űrlap megnyitása

Az **űrlap nézetben az űrlap használható**, kötött űrlap esetén a rekordokkal elvégezhetjük a szokásos műveletek. A mezők értékei különböző vezérlőelemekben, általában úgynevezett szövegmezőkben jelennek meg.

A menüszalag **Kezdőlap** parancslapján a táblák kezelésekor látott eszközök helyezkednek el, az űrlap alján pedig megjelennek a **navigációs gombok**. Ezekkel az eszközökkel éppen dolgozhatunk a rekordokon, mint a táblák felhasználói nézetében. Amikor tallózunk a rekordok között, az űrlapon található feliratok változatlanok maradnak, míg a szövegmezők és egyéb vezérlőelemek tartalma a kiválasztott rekordnak megfelelően változik. A felületen mindig a kiválasztott rekord mezőértékei jelenik meg.




*64. ábra Tábla űrlap nézete*

### 10.7.2 Űrlap bezárása

Az űrlap nézetben lévő űrlapot a megnyitáshoz hasonlóan többféleképpen bezárásához kattintsunk az ablak bezárás gombjára, vagy ha pedig az ablak teljes méretű, akkor a menüszalag jobb szélén látható bezárás gombra!

### 10.7.3 Az űrlapok tervező nézete

A **tervező nézetben határozzuk meg az űrlap formátumát, és adatkapcsolatát**. Itt adhatjuk meg az űrlap rekordforrását, itt helyezhetjük a felületre a mezőértékeket megjelenítő vezérlőelemeket, és itt állíthatjuk be azok formátumát.

Az űrlap tervező nézetű megnyitásakor a menüszalag jobb oldalán a két parancslapon kiválasztható **Űrlaptervező eszközök** jelennek meg. A **Tervezés** parancslap gombjai vezérlőelemek elhelyezését és formázását, az **Elrendezés** lap pedig azok igazítását, elrendezést teszi lehetővé. A tervező nézet nagy előnye, hogy a WYSIWYG szerkesztésnek megfelelően azonnal láthatóak a beállított formátumok.

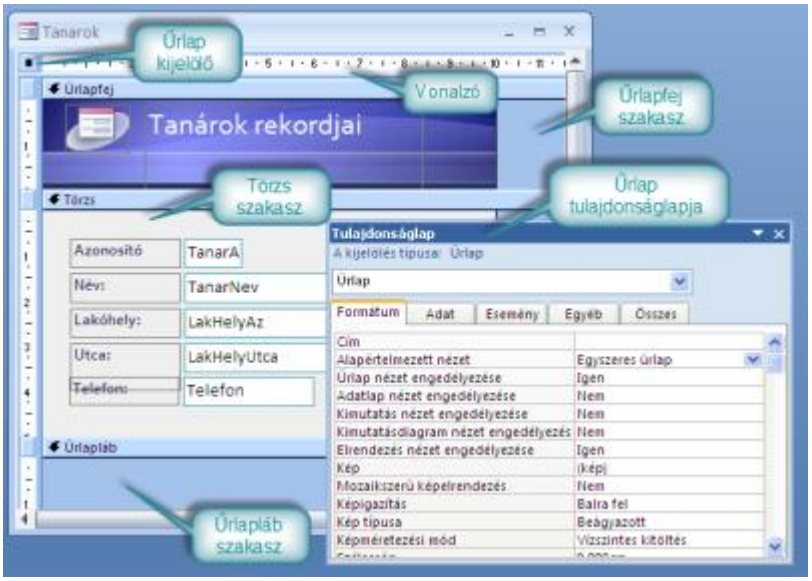


*14. kép Menüszalag az űrlaptervezés eszközeivel*

Az ablak felső és bal oldali szegélyén az úgynevezett **vonalzók** helyezkednek el, amelyek a vezérlőelemek pozícionálásában, méretezésében és kijelölésében játszanak szerepet.

A vonalzók találkozásánál található az **űrlapkijelölő gomb** amelynek, szerepét neve is elárulja. Az űrlap különböző tulajdonságainak beállítása előtt, az űrlap kiválasztására fogjuk használni.

Az űrlaptervező ablak munkaterületén az űrlap felülete látszik, alapértelmezés szerint általában fehér színnel jelölve. Ezen a felületen, helyezhetők el a különböző vezérlőelemek. Az űrlap függőlegesen több részre, az úgynevezett **szakaszokra** tagolódik. A legfontosabb szakasz a **Törzs**. Itt szoktuk elhelyezni a rekordok mezőinek megjelenítésre szolgáló vezérlőelemeket. Az **űrlapfej**, és **-láb** a Word élőfej, élőláb szövegterületeihez hasonlítanak. Ide olyan feliratokat, egyéb vezérlőelemeket helyezünk, amelyek tartalma független az éppen kiválasztott rekordtó. A törzs alatt és fölött helyezkedhetnek el az **oldalfej** és **oldalláb** szakaszok, amelyeknek akkor van jelentőségük, ha űrlapunk több képernyőoldalas. Ilyenkor az egyes oldalak alján és tetején megjelenítendő vezérlőelemek helyezhető itt el.




*65. ábra Tervező nézet*

A szakaszok közül a törzsnek mindig látszania kell, a másik két szakaszpár azonban ki-, vagy bekapcsolható az űrlap helyi menüjében. Minden szakasz fölött hosszú, szakaszkijelölő gomb helyezkedik el, amelyeknek a szakasz kiválasztásban, és méretezésében van jelentőségük.

Az űrlaptervező ablak fontos része a kötött űrlap rekordforrásának mezőneveit tartalmazó a **Mezőlista**, amelynek a kötött vezérlőelemek létrehozásában lesz szerepe. A mezőlista a **Tervezés** parancslap **Eszközök** csoportjának **Mezők felvétele** gombjával, vagy az **Alt+F8** billentyűkkel jeleníthető meg.

A menüszalag **Tervezés** parancslapján,a **Vezérlők** csoportban találjuk meg azokat a gombokat, amelyekkel az adatok megjelenítésére használható vezérlőelemeket az űrlap felületére helyezhetjük.

Az űrlaptervező egyik legfontosabb alkotóeleme a **Tulajdonságlap**, amelyen az éppen kijelölt objektum – űrlap, szakasz, vezérlőelem – jellemzői állíthatók be. Ha éppen nem látszik a képernyőn, akkor a **Tervezés** parancslap **Eszközök** csoportjának **Tulajdonságlap** gombjával, vagy az **F4** billentyűvel kapcsolhatjuk be.

A tulajdonságlap különböző kartonjain a kijelölt objektum minden tulajdonsága megtalálható, és szerkeszthető.

### *A tulajdonságlap használata*

A **Tulajdonságlap** mindig az éppen kijelölt objektum tulajdonságait mutatja. Ezeket, nagy számuk miatt, egy öt különböző lapon állíthatjuk be.

- **Formátum:** megjelenését befolyásoló tulajdonságok.
- **Adat:** a vezérlőelemben megjelenő adatra vonatkozó tulajdonságok.
- **Esemény**: a vezérlő programkóddal, vagy makróval való összekapcsolását lehetővé tévő tulajdonságok.
- **Egyéb**: vegyes tulajdonságok.
- **Összes**: az objektum minden tulajdonsága.

A legtöbb tulajdonság megadására egyszerű szövegdobozok állnak rendelkezésre, amelyekbe az űrlap tervezője begépelheti a szükséges adatot. Más tulajdonságok lehetséges értékeit legördülő listákról választhatjuk ki. A néhány tulajdonság értékének beállításában pedig segítségül hívhatjuk az Access beépített szerkesztőit. Az ilyen módon megadható tulajdonságok mellett három pontot (…) tartalmazó nyomógomb jelenik meg, amelyre kattintva szerkesztő elindítható.

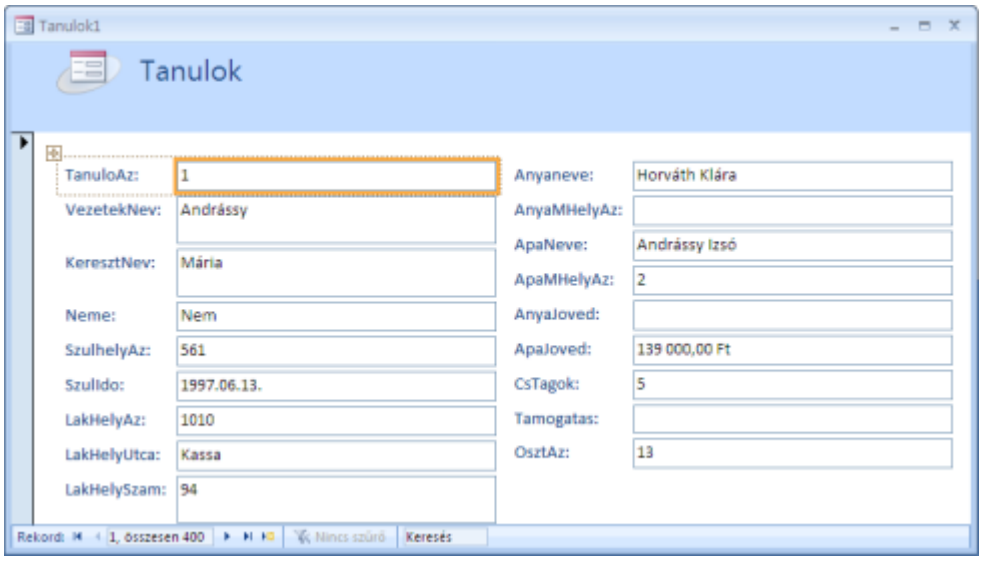
## 10.8 ÚJ ŰRLAPOK LÉTREHOZÁSA

Miután némi előismeretre tettünk szert az űrlapokkal kapcsolatban, már nem kell attól félnünk, hogy az új űrlap tervezésekor elénk táruló, meglehetősen összetett kép elriaszt bennünket. Vizsgáljuk meg tehát, hogyan készíthető el a legegyszerűbben egy űrlap.

### 10.8.1 Űrlap létrehozás varázslóval

Bár eddig mellőztük az Access által biztosított varázslókat, az űrlap gyors létrehozásakor ezek nyújtják a legegyszerűbb megoldást. Ha gyorsan és kényelmesen szeretnénk elkészíteni egy tábla, vagy lekérdezés rekordjait megjelenítő űrlapot, végezzük el a következő lépéseket:

- Kattintsunk a navigációs ablakban a táblára, vagy lekérdezésre. Ne nyissuk meg az, csak jelöljük ki az objektumot!
- Ezt követően, a menüszalag **Létrehozás** lapján, az **Űrlapok** csoportban kattintsunk az **Űrlap** gombra.
- Az Access azonnal létrehoz egy **kötött űrlapot**, amely a kiválasztott tábla, vagy lekérdezés rekordjainak megjelenítésére alkalmas.
- A varázsló beállítja az űrlap **rekordforrását**, elhelyezi rajta a mezőértékeket megjelenítő, **kötött vezérlőelemeket** és megformázza az űrlapot.
- A létrehozott objektum **Elrendezési nézetben** jelenik meg.
- Akár ebben a nézetben, akár **Tervező** nézetre váltva folytathatjuk az űrlap formázását.
- Ha elkészültünk, szokásos módon lemezre menthetjük az űrlapot.




*15. kép Varázslóval létrehozott űrlap*

Az űrlap létrehozásakor használhatjuk a menüszalagon a **Létrehozás** lap, **Űrlapok** csoportjának **Osztott űrlap**, vagy **Több elem** gombját is. Előbbi olyan űrlapot hoz létre, a melynek felső részén egy kiválasztott rekord, az alsó részben pedig a rekordok adattábla nézete látszik. A **Több elem** gomb folyamatos űrlapot állít elő.

### 10.8.2 Az űrlap felületi megjelenésének szabályozása

Az elkészített űrlap dizájnját az **Access 2007** téma alapján formázza meg a varázsló. Ha más megjelenést szeretnénk, váltsunk tervező nézetre, és válasszunk új témát, a menüszalag **Elrendezés** lapjának **Automatikus formázás** csoportjában.

### 10.8.3 Címkék szövegének átírása

A mezőértékeket megjelenítő vezérlőelemek előtt a mezők neveit tartalmazó címkék jelennek meg. Ezek szövegét tetszés szerint megváltoztathatjuk úgy, hogy a betűk fölött duplán kattintunk, majd átírjuk a szöveget. A varázslóval létrehozott űrlap fejlécében szintén címkeként jelenik meg a az űrlap rekordforrását alkotó tábla neve. A fenti módon ezt a szöveget is átírhatjuk.

### 10.8.4 Vezérlőelemek Betűformátuma

Ha valamelyik vezérlőelem betűtípusán, betűméretén, igazításán, színén, vagy más tulajdonságán változtatni akarunk, kattintással jelöljük ki az elemet, és a **Tervezés** parancslap **Betűtípus** csoportjában állítsuk be az új formátumot.

### 10.8.5 Vezérlőelemek méretezése

A vezérlőelemek méretezését a kijelölt elemet körülvevő narancsszínű keret húzásával lehet elvégezni. Ha az űrlapot varázslóval hoztuk, méretezéskor létre tapasztalhatjuk, hogy egy elem változtatásakor más elemek mérete is módosul. Ennek az az oka, hogy a varázsló az egy oszlopot alkotó vezérlőelemeket egyfajta csoportban, **Halmozott elrendezésben** helyezi el az űrlapon. A közös halmazba tartozó elemek együtt méretezhetők, és csak együtt kijelölve pozícionálhatók. A halmazból való eltávolításhoz jelöljük ki az elemet, majd a menüszalag **Elrendezés** lap **Elrendezés beállítása** csoportjában kattintsunk az **Eltávolítás** parancsra! Ezt követően az elem önállóan méretezhető.

> A halmaz teljes megszűntetéséhez jelöljük ki annak minden elemét, és így kattintsunk az **Eltávolítás** parancsra!

### 10.8.6 Vezérlőelemek mozgatása

Amikor egy vezérlőelemet el szeretnénk mozdítani az úrlapon, egyszerűen jelöljük ki, majd a befoglaló szegélyt megfogva húzzuk a megfelelő helyre az elemet.

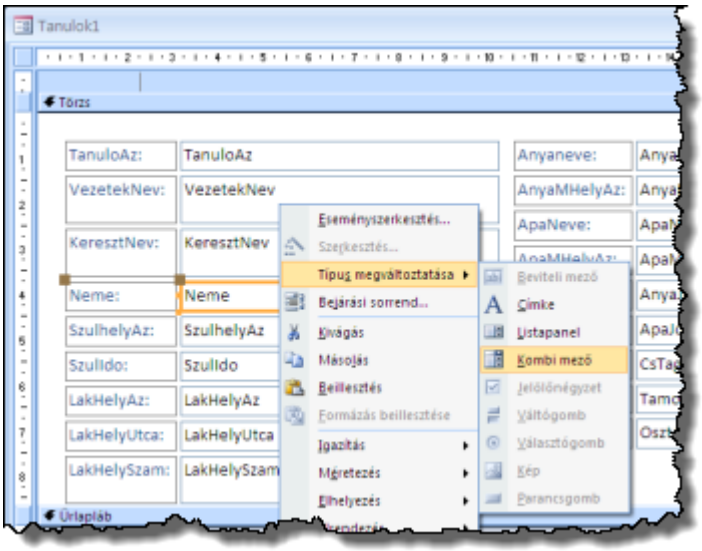
Tapasztalni fogjuk, hogy áthelyezéskor a vezérlőelem a hozzá tartozó címkével együtt mozog. Ha csak az elemet akarjuk mozgatni, fogjuk meg, és húzzuk el a kijelölt vezérlő bal felső sarkában lévő négyzetet! Ugyanezt megtehetjük a címkével is. Ha a címkére egyáltalán nincs szükségünk, kijelöljük ki, és nyomjuk le a **Delete** billentyűt.

> A halmozott vezérlők csak együtt mozgathatók! Ha egy halmozott vezérlőt önállóan akarunk mozgatni, távolítsuk el a halmazból!

### 10.8.7 Vezérlőelemek típusának megváltoztatása

A varázslóval létrehozott űrlapokon az Access minden mezőértéket szövegmezőben jelenít meg. Ez megfelelő azoknál a mezőknél, amelyek valós tartalmát akarjuk megmutatni a felhasználónak. Ilyen például a **Tanulok** táblában a ***VezetekNev***, ***KeresztNev***, ***SzulIdo*** stb. mező. Az **Igen/Nem** típusú mezők (például ***Neme***) esetében azonban a felhasználó nem biztos, hogy ismeri az értékek jelentését. Még nagyobb lehet a bizonytalanság akkor, amikor egy idegen kulcs értéke jelenik meg a képernyőn. Még az adatbázis készítője sem tudja fejből, melyik települést azonosíthatja egy tanuló ***SzulHely*** mezőjében az **561**-es érték. Ilyen esetekben a vezérlőelem típusának megváltoztatásával elérhetjük, hogy a mezőérték sokkal világosabb formában jelenjen meg.

Egy vezérlőelem típusát az elem helyi menüjében, a Típus megváltoztatása paranccsal, majd a megjelenő almenüben az új típus kiválasztásával végezhetjük el. Egy szövegmezőt például kombi mezőre cserélhetünk a **helyi menü/Típus megváltoztatása/Kombi mező** paranccsal.




*16. kép Vezérlőelem típusának megváltoztatása*

### *Idegen kulcsok értékeinek megjelenítése*

A vezérlőelem típusának megváltoztatásához meg kell értenünk, hogyan kapcsolódik az kötött űrlap egy kötött vezérlőeleme a rekordforrás valamelyik mezőjéhez.

Vegyünk példaként, egy a ***Tanulok*** tábla alapján készült űrlapot, amelyen többek között szerepel a tanuló születési helyének megjelenítésére szolgáló szövegmező! Az űrlap rekordforrása a ***Tanulok*** tábla, az említett szövegmező pedig, a tábla ***SzulHely*** mezőjéhez kapcsolódik. Amikor az űrlapot **Űrlap** nézetben nézzük, a szövegmezőben mindig a kiválasztott rekord ***SzulHely*** mezőjének értéke látszik, ami nem más, mint egy szám. Egy rekord kiválasztásakor, az *Access űrlapkezelő rendszere* kimásolja a ***Szulhely*** mező tartalmát a táblából, és megjeleníti azt a szövegmezőben. Ha a szövegmező tartalmát átírjuk, akkor visszamásolja az új értéket a táblába.

A ***SzulHely*** mező történetesen idegen kulcs, a ***Telepulesek*** tábla azonosítójának értékeit tartalmazza. Amikor a mezőben az **561-es** érték látszik, az valójában Budapestet jelenti, hiszen a ***Telepulesek*** tábla **561**-es azonosítójú rekordjában a ***Tel_Nev*** mezőben **Budapest** érték van.

Hogyan lehetne elérni, hogy az űrlapon ne **561** jelenjen meg a születési helyet megjelenítő vezérlőelemben, hanem a **Budapest** szöveg? Arra kellene rávenni az Accesst, hogy amikor az űrlapon megjeleníti egy tanuló rekordját, ne az 561-et jelenítse meg az űrlapon, hanem ***Telepulesek*** táblában nézze meg, hogy melyik települést azonosítja az 561, és a képernyőn a település nevét, azaz a **Budapest** szöveget nézze meg. Ha pedig szövegmező tartalmát Veszprémre változtatjuk, akkor nézze meg a ***Telepules*** táblában **Veszprém** azonosítóját, és azt írja vissza a Tanulok tábla ***SzulHely*** mezőjébe.

A megoldás a **kombi mezőben** rejlik. A kombi mező olyan speciális vezérlőelem, amely egy szövegmezőből és egy többsoros listából áll. Bár nem mindig látszik, a lista

több oszlopot, akár egy teljes táblát is tartalmaz(hat). A rekordforrás kötődő mezőjéhez a legördülő lista egy oszlopa kötődik, nem pedig a szövegmező. Amikor a listából kiválasztunk egy sort, a kötött oszlop értéke kerül a kötődő mezőbe.

Új rekord kiválasztásakor a kötődő mező értékét a lista kötött oszlopában keresi meg az Access. Amikor megtalálja a megfelelő sort, annak egy celláját jeleníti meg a szövegmezőben.

Ha a tanulók születési helyének azonosítóját tartalmazó SzulHely mezőt kombi mezővel akarjuk kezelni, meg kell adni, hogy a kombi mező listája a ***Telepulesek*** tábla rekordjaiból épüljön föl. Be kell állítani, hogy a település azonosítóját tartalmazó oszlop legyen a kötött oszlop, de a kombi mezőben a listából kiválasztott település neve jelenjen meg.

Lássuk a lépéseket!

- A ***SzulHely*** mezőhöz kötött vezérlőelem típusát állítsuk szövegmezőről kombi mezőre!
- Jelöljük ki a kombi mezőt, majd nyissuk meg a tulajdonságlapját! (F4)
- Kattintsunk az **Adat** fülön, és a **Sorforrás** tulajdonságnál válasszuk ki a ***Telepulesek*** táblát! Ezzel elérjük, hogy a lista tartalma a teljes ***Telepulesek*** tábla legyen!
- Most némi képzelőerőre van szükségünk… föl kell idéznünk a ***Telepulesek*** tábla felépítését! A tábla négy mezőből áll. 1. mezője (oszlopa) tartalmazza az azonosítót, a 2. mező a település nevét, a 3. az irányítószámot, a 4. a megye azonosítóját.
- Figyeljük meg, hogy a kombi mező tulajdonságlapján, a **Kötött oszlop** tulajdonság értéke 1. Ez azt jelenti, hogy a lista 1. oszlopa kötődik majd a rekordforrás kötődő mezőjéhez. Ez éppen megfelel nekünk, hiszen azt akarjuk, hogy egy település sorának kiválasztásakor annak azonosítója kerüljön a kötődő mezőbe.
- Váltsunk most a kombi mező tulajdonságlapján a **Formátum** fülre! Az **Oszlopszám** tulajdonság értéke 1. Ha ezt így hagyjuk, a kombi mezőnk nem fog jól működni, hiszen a lista 4 oszlopos. És az első két oszlopra mindenképpen szükségük van. Állítsuk az **Oszlopszám** tulajdonság értékét 2-re!
- Váltsunk Űrlap nézetre, és nézzük meg az eredményt! FÉL SIKER! A lista tartalmazza ugyan a települések neveit, de amikor kiválasztunk egy sort, akkor az azonosító jelenik meg a kombi mezőben.
- Az utolsó trükk következik. Menjünk tervező nézetbe, és kattintsunk a kombi mező tulajdonságlapján az **Oszlopszélességek** tulajdonságra! Gépeljük be **0;2** szöveget! Ezzel elérjük, hogy a lista első oszlopa **0 cm** széles lesz, azaz nem is látszik, a település neveket tartalmazó 2. oszlop viszont **2 cm** szélességben jelenik meg.
- Próbáljuk ki az eredményt!

### *Igen/Nem mezők megjelenítése kombi mezőben*

A kombi mezők kiválóan alkalmasak **Igen/Nem** típusú mezők értékeinek megjelenítésére is. A **Tanulok** táblában ilyen például a gyerekek nemét tároló mező. Ezt például szerencsés lenne úgy megjeleníteni, hogy az űrlapon **Igen** és **Nem** értékek helyett a **Fiú**, illet-

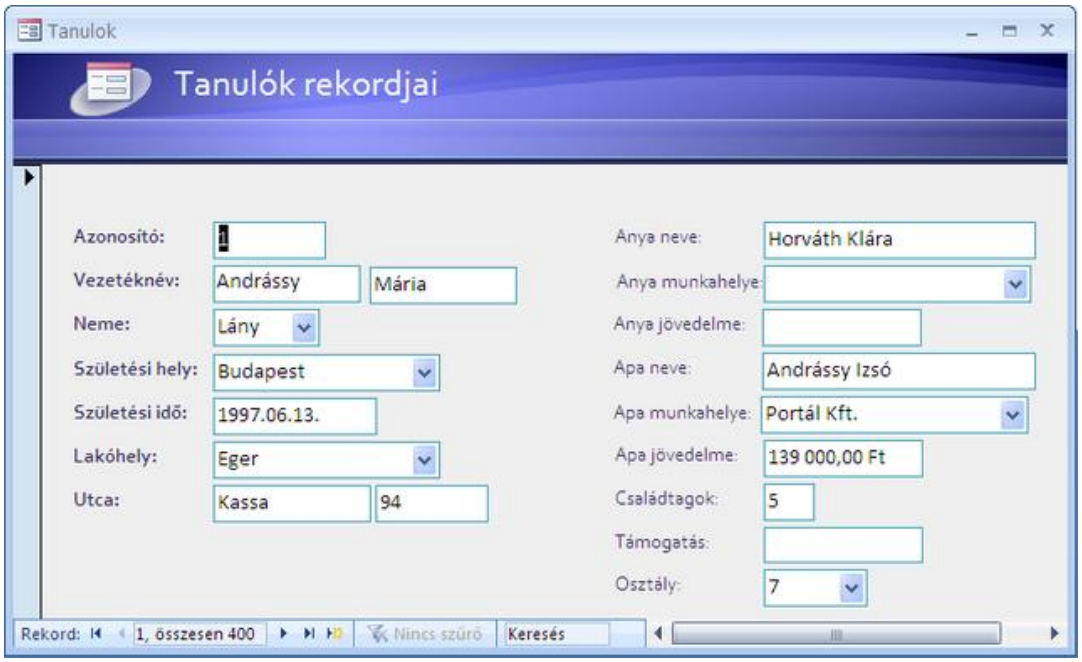
ve **Lány** szavak jelenjenek meg. Ehhez olyan, két soros, két oszlopos kombi mezőt kell készíteni, amelynek első oszlopa **0**,és **-1** értéket, második oszlopa pedig a **Lány** és **Fiú** szöveget tartalmazza.

> Talán emlékszünk arra, hogy az **Igen** értékként **-1**-et, a **Nem** érték esetében pedig **0**-t tárol.

A lépések a következők:

- A ***Neme*** mezőhöz kötött vezérlőelem típusát állítsuk szövegmezőről kombi mezőre!
- Jelöljük ki a kombi mezőt, majd nyissuk meg a tulajdonságlapját! (F4)
- Kattintsunk az **Adat** fülön! Az előbbi példában a ***Telepulesek*** tábla adta a kombi mező listájának sorait. Most jóval egyszerűbb, mindössze két sorból és két oszlopból álló listára van szükségünk.
- Amikor a lista tartalma előre megadható, a kombi mező tulajdonságlapjának **Adat** fülén, a **Sorforrás típusa** tulajdonságot állítsuk **Lista**-ra!
- Kattintsunk a **Sorforrás** tulajdonságra, és gépeljük be a következő szöveget: **0;Lány;-1;Fiú**
  Ezzel megadtuk, hogy a lista egyes celláinak tartalmát.
- A Formátum fülön állítsuk **2**-re az **Oszlopszámot**, és **0;2**-re az **Oszlopszélességek**-et!
- Ellenőrizzük az eredményt! Figyeljük meg, hogy minden tanulónál helyesen jelenik meg a nem!

Az eddigi megszerzett tudás birtokában már elkészíthetjük az alábbi űrlapot.




*17. kép A Tanulók rekordjai űrlap*

### 10.8.8 Az űrlap mentése

Amikor elkészültünk az űrlapunkkal, semmiképpen se feledkezzünk meg a mentésről! Kattintsunk a gyorsindító eszköztár mentés gombjára, vagy üssük le a **Ctrl+S** billentyűket! Adjuk meg az űrlap nevét, és kattintsunk az **Ok** gombra!

## 10.9 AZ ADATBÁZIS MENÜ ŰRLAPJA

Ha már el tudjuk készíteni az adatbázis tábláit kezelő űrlapokat, nincs más hátra, minthogy elkészítsük az a speciális űrlapot, amely az adatbázis menüablakaként fog szolgálni, majd beállítsuk, hogy ez az űrlap automatikusan megjelenjen az adatbázis megnyitásakor.

A menü űrlap elkészítése valamivel bonyolultabb, némi programozói tudást is igénylő feladat lenne, de szerencsére ezt a munkát is megkönnyíti egy speciális varázsló.

Lássuk, hogyan készül el az adatbázisunk menüje!

- Kattintsuk a menüszalag **Adatbáziseszközök** lap **Adatbáziseszközök** csoportjában **Kapcsolótábla-kezelő** parancsra!
- Válaszoljunk **Igen**-nel az előugró ablakban!
- A megjelenő **Kapcsolótábla-kezelő** nevű ablakban kattintsunk a szerkesztés gombra!
- A **Kapcsolótábla neve** szövegmezőbe írjuk be a „**Menü**” szöveget!
- Kattintsunk az **Új** gombra!
- A **Szöveg**-hez írjuk be a szöveget, a **Parancs** listában válasszuk az **Űrlap Szerkesztés módban** sort!
- Az **Űrlap** sorban adjuk meg, melyik űrlapot szeretnénk megnyitni a paranccsal!
- Kattintsunk az **OK** gombra, majd zárjuk be a szerkesztő ablakát!
- Az űrlapok között megjelenik a ***Kapcsolótábla*** űrlap a táblák között pedig a ***Switchboard Items***.
- Az űrlapot megnyithatjuk, és megváltoztathatjuk a formátumát. Arra azonban vigyázzunk, hogy ne töröljünk semmit, mert különben a menü nem fog helyesen működni.
- Ha mégis elrontanánk valamit, töröljük a ***Switchboard Items*** táblát és a ***Kapcsolótábla*** űrlapot, majd készítsünk új kapcsolótáblát!

Ha szeretnénk elérni, hogy a kapcsolótábla űrlap automatikusan megjelenjen, amikor az adatbázis megnyitjuk, kövessük a következő lépéseket:

- Kattintsunk az **Office menü** gombra,majd az **Access beállításai** parancsra!
- A bal oldali listában válasszuk az Aktuális adatbázis parancsot!
- A jobb oldali ablakrészben keressük meg a megjelenítendő űrlap tulajdonságot, és állítsuk be a Kapcsolótábla űrlap nevét!
- Válasszuk az OK gombot, zárjuk be, majd nyissuk meg az adatbázist!
- A képernyőn megjelenik a kapcsolótábla űrlap, amivel elindíthatjuk az adatbázis további űrlapjait.

Adatbázisunkat szabályos alkalmazássá fejlesztettük! Ha később újabb parancsokat szeretnénk elhelyezni a menü űrlapon, ismételjük meg a fent leírt lépéseket!

## 10.10 ÖSSZEFOGLALÁS

Az űrlapok az adatbázis kezelőfelületének kialakítására használható objektumok. Az adatbázis fejlesztője kötött űrlapokat készíthet a táblákban tárolt rekordok megjelenítésre és kezelésre, kötetlen űrlapokat pedig vezérlési funkciókra például menük kialakítására használhatja.

Az űrlapok elkészítésnek legegyszerűbb módja, az űrlap készítő varázsló használata, amelyet a rekordforrásként szolgáló tábla, vagy lekérdezés kijelölése után, a menüszalag **Létrehozás** lap **Űrlapok** csoportjában az **Űrlap** paranccsal kezdeményezhetünk. Az Access létrehozza az űrlapot, és elhelyezi rajta a rekordforrás mezőértékeinek megjelenítéséhez szükséges vezérlőelemeket.

A kialakított űrlap **Elrendezési nézetben** jelenik meg, amelyben az űrlap úgy szerkeszthető, hogy közben a rekordforrás adatait is megjeleníthetjük.

**Tervező** nézetbe váltva megváltoztathatjuk az űrlap, illetve az azon elhelyezett vezérlők összes tulajdonságát. Az űrlap dizájnjának megváltoztatáshoz különböző témákat választhatunk a menüszalag **Automatikus formázás** gombjával. Egyéb formázáshoz használhatjuk az **Űrlap-elrendezési eszközök** **Formátum** és **Elrendezés** lapján elhelyezkedő parancsokat, illetve a kiválasztott objektum tulajdonságlapját.

A mezők értékeinek megjelenítéshez általában megfelelő a szövegmező vezérlőelem, de egyes esetekben, például idegen kulcsok, vagy Igen/Nem típusú mezők esetén célszerű megváltoztatni a vezérlőelem típusát. Ilyenkor tehet jó szolgálatot a kombi szövegmező, amelyben nem a szövegmező, hanem a lista valamelyik oszlopa kötődik a rekordforrás mezőjéhez. A lista sorait egy tábla (vagy lekérdezés), vagy pedig egy a Sorforrásba begépelt, pontosvesszőkkel tagolt szöveg szolgáltathatja.

Az adatbázis űrlapjainak elkészítése után az menüszalag **Adatbáziseszközök** lap **Adatbáziseszközök** csoportjában a **Kapcsolótábla-kezelő** paranccsal menü űrlapot hozhatunk létre. A menü űrlapok más űrlapok közötti tallózásával megkönnyítik a kialakított kezelőfelület használatát.

## 10.11 ÖNELLENŐRZŐ KÉRDÉSEK

1. Milyen célokra használhatók az űrlapok?
2. Mi a különbség egyszeres és folyamatos űrlap között?
3. Hogyan épül fel egy űrlap szerkezete?
4. Milyen két nagy csoportra oszthatók az űrlapon elhelyezett vezérlőelemek?
5. Hogyan változtathatja meg egy vezérlőelem típusát?
6. Melyik vezérlőelem a legalkalmasabb az idegen kulcsok megjelenítésére?
7. Hogyan méretezheti a vezérlőelemeket?
8. Hogyan tudja néhány kattintással teljesen megváltoztatni egy űrlap megjelenését?
9. Hogyan készíthet menü űrlapot az adatbázishoz.

# 11. TÁROLT ADATOK NYOMTATOTT MEGJELENÍTÉSE

## 11.1 CÉLKITŰZÉS

A jelentések adatok nyomtatott megjelenítése használhatók. Az űrlapokhoz hasonlóan általában kötöttek[1], azaz különböző vezérlőelemek segítségével, egy rekordforrás (tábla, lekérdezés) rekordjainak mezőértékeit jelenítik meg a nyomaton.

Elkészítésük közben számtalan lehetőség kínálkozik a rekordok csoportosítására, és rendezésére, a mezőket megjelenítő vezérlőelemek elhelyezésére, formázására, számított értékek képzésre és megjelenítésére. Készítésük és használatunk szinte külön könyvet érdemelne, ezért tananyagunkban a legfontosabb funkciókat tekintjük csak át.

Ebben a leckében egy példán keresztül azt tanulhatja meg, hogyan egyszerűsítheti le a jelentés elkészítését az Access jelentéskészítő varázslójával, és hogyan alakíthatja át saját elképzeléseinek megfelelően az objektumot. A lecke végére képes lesz egy lekérdezés, vagy tábla rekordjait jelentésben megjeleníteni, ismerni fogja a jelentések nézeteit, képes lesz formátumuk szabályozására, megtanulja, hogyan lehet csoportosítani, és rendezni a rekordforrás rekordjait, tudni fogja, hogyan lehet az egyes rekordcsoportokhoz fejlécet, és láblécet kapcsolni, azokban összegzések eredményeit elhelyezni.

Megtanulja, hogyan lehet a jelentést kinyomtatni, vagy különböző külső állományokba, például PDF dokumentumba exportálni.

## 11.2 A LECKE TÉMAKÖREI

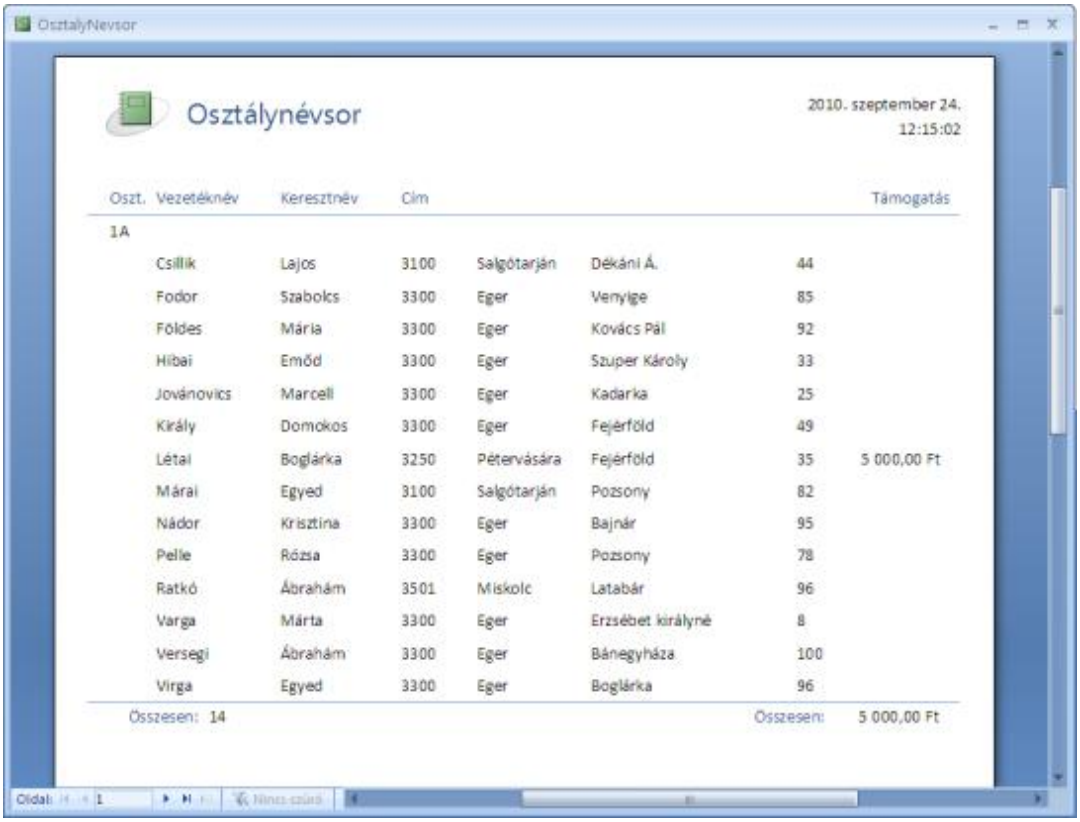
- Rekordforrás
- szakaszok
- csoportok fejlécek
- számított értékek

## 11.3 A JELENTÉSEK KÉSZÍTÉSE VARÁZSLÓ SEGÍTSÉGÉVEL

A jelentések elkészítésekor rengeteg fáradságtól kíméljük meg magunkat, ha az első változatot varázslóval készítjük el, majd testre szabjuk az objektumot.

Leckénkben olyan jelentést készítünk majd, amely a példa adatbázisban tárolt iskola egyes osztályainak névsorát, a tanulók lakcímét, és az iskolától eddig kapott tankönyvtámogatást jeleníti meg. A jelentésben minden osztály külön oldalon jelenik majd meg, és osztályonként látható lesz az osztálylétszám, az osztályban kiosztott összes tankönyvtámogatás.

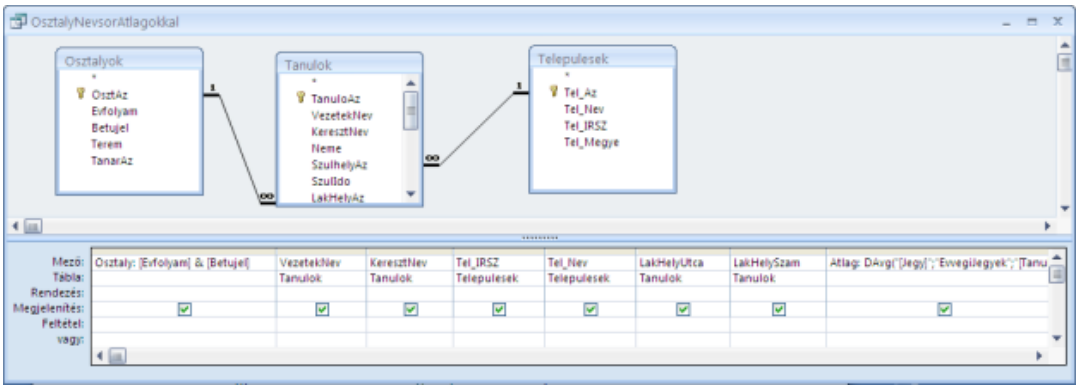
[1] Készíthetünk ugyan kötetlen jelentést is, ennek azonban kevés értelme van, hiszen az adattartalomtól független szöveget bármilyen szövegszerkesztővel elkészíthetjük.




*18. kép Jelentés nyomtatási képe*

Jelentésünk rekordforrásaként egy olyan lekérdezést[2] használunk majd, amely az Osztalyok-Tanulok-Telepulesek táblák kapcsolatából származó rekordokat válogatja ki. Megjeleníti a tanulók osztályát, vezeték és keresztnevét, lakcímét, és a kapott támogatást.

[2] A lekérdezés megtalálható a példa adatbázisban.




*66. ábra A rekordforrás*

A jelentés elkészítéséhez, a navigációs ablakban kattintsunk az ***OsztalyNevSor*** lekérdezés nevére, de ne nyissuk meg azt!

Kattintsunk a menüszalag **Létrehozás** lap **Jelentések** csoportjában a **Jelentés** parancsra.

A varázsló azonnal elkészíti a jelentést, amely **Elrendezési nézet**-ben jelenik meg a képernyőn.




| Osztaly | VezetekNev | KeresztNev | Tel_IRSZ | Tel_Nev | La |
|---|---|---|---|---|---|
| 1A | Virga | Egyed | 3300 | Eger | Bo |
| 1A | Létai | Boglárka | 3250 | Pétervására | Fe |
| 1A | Nádor | Krisztina | 3300 | Eger | Ba |
| 1A | Csillik | Lajos | 3100 | Salgótarján | Dé |
| 1A | Pelle | Rózsa | 3300 | Eger | Po |
| 1A | Jovánovics | Marcell | 3300 | Eger | Ka |
| 1A | Hibai | Emőd | 3300 | Eger | Sz |
| 1A | Fodor | Szabolcs | 3300 | Eger | Ve |
| 1A | Versegi | Ábrahám | 3300 | Eger | Bá |
| 1A | Varga | Márta | 3300 | Eger | Er |
| 1A | Földes | Mária | 3300 | Eger | Ko |
| 1A | Márai | Egyed | 3100 | Salgótarján | Po |
| 1A | Ratkó | Ábrahám | 3501 | Miskolc | La |
| 1A | Király | Domokos | 3300 | Eger | Fe |

*67. ábra A varázsló eredménye*

## 11.4 A JELENTÉSEK NÉZETEI

Az űrlapokhoz hasonlóan a jelentések is többféle nézetben jeleníthetők meg. A nézeteket a menüszalag bal szélén megjelenő Nézet gombon látható nyilacskával választhatjuk ki.

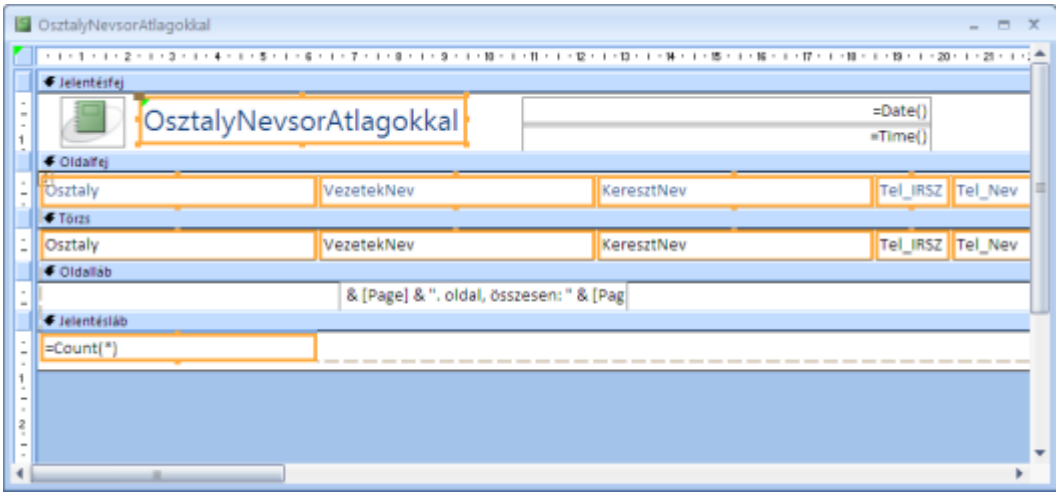
- **Elrendezési nézet:** ebben a nézetben úgy tudjuk megváltoztatni a jelentés vezérlő elemeinek elhelyezkedését, méretét, igazítását, azaz elrendezését, hogy közben láthatók a jelentés vezérlőelemei, és azok adattartalma is.
- **Tervezés nézet:** Ebben a nézetben a jelentés, az azon elhelyezett vezérlőelemek összes tulajdonsága beállítható, de csak a vezérlőelemek látszanak, adattartalom nélkül.
- **Jelentés nézet:** A jelentés áttekintő nézete, amelyben nem lehet szerkeszteni az objektum elrendezését és formátumait, ellenben lehetőségünk nyílik a rekordok áttekintésére, és esetleges szűrésére is. Ha szűrést végzünk, akkor csak a kiválogatott rekordok jelennek meg a jelentés nyomtatásakor.
- **Nyomtatási kép nézet:** Ez a nézet abban a formában mutatja meg a jelentést, ahogyan az majd nyomtatásban is megjelenik. Nyomtatási kép nézetben megváltozik a menüszalag összetétele. A megszokott lapok eltűnnek, és csak a nyomtatáshoz kapcsolódó parancsokat tartalmazó **Nyomtatási kép** parancslap látszik. Ha vissza akarunk térni az előző nézetbe, kattintsunk a **Nyomtatási kép bezárása** gomba!

## 11.5 A JELENTÉSEK SZERKEZETE

A jelentések, az űrlapokhoz hasonlóan vezérlőelemek segítségével jelenítik meg a rekordforrás mezőinek értékeit. A **vezérlőelemek a jelentés különböző szakaszaiban helyezkednek el**.

- **Törzs:** A jelentés legfontosabb szakasza a **törzs**, amelyben a rekordforrás egy rekordja tehető láthatóvá. Amikor a jelentést kinyomtatjuk, a törzs annyiszor jelenik meg, ahány rekord található a rekordforrásban.
- **Jelentésfej, jelentésláb:** A törzsön kívül különböző fej- és lábléceket használhatunk. A jelentésfej-jelentésláb szakaszok tartalma egyszer jelenik meg a jelentés első oldalának tetején, illetve utolsó oldalának végén.
- **Oldalfej, oldalláb:** Többoldalas jelentés esetén az egyes oldalak tetején és alján megjelenő szakaszok.
- **Csoport fejlécek, láblécek:** A rekordforrás rekordjait folyamatosan, vagy különböző szempontok szerint csoportosítva jeleníthetjük meg. A csoportfej és csoportláb, egy rekordcsoport fölött illetve alatt megjelenő szakaszok. Általában a csoport rekordjain elvégzett összegző számítások megjelenítésére használjuk őket. A tanulók rekordjait például osztályonként fogjuk csoportosítani, és minden osztály csoportjának láblécében megmutatjuk az osztálylétszámot (azaz a csoport rekordjainak számát), illetve az osztályban kiosztott összes tankönyvtámogatást. A csoport fejlécek és láblécek csak akkor jelennek meg, ha a rekordcsoportokat már létrehozzuk.

A szakaszok a jelentés tervező nézetében tekinthetők meg a legegyszerűbben.




*19. kép Lekérdezés tervező nézetben*

## 11.6 A JELENTÉS ELRENDEZÉSÉNEK SZABÁLYOZÁSA

A jelentés rekordcsoportjainak kialakítására, szakaszok, és a vezérlőelemek elrendezésére, illetve összegzések beállítására leginkább az **Elrendezési nézetet** használhatjuk. Ebben a nézetben az ablak két részre bomlik.

- A felső részben szabályozhatjuk az elrendezést. Szürke szaggatott vonalak mutatják a jelentés margónak határait. Látjuk a vezérlőelemeket, (amelyek az esetek többségében egyszerű szövegmezők).
- A **Csoportosítás, rendezés, és összeg** részben a csoportok kialakítására és rendezés beállítására használható gombokat találjuk.

Elsőként a csoportosítást és összegképzést végezzük el, majd beállítjuk a vezérlőelemek elrendezését.

### 11.6.1 Csoportok kialakítása, és összegképzés

Az ablak felső részét görgetve jól látszik, hogy a rekordok egyenlőre folyamatosan, mindenféle rendezés és csoportosítás nélkül jelennek meg.

Figyeljük meg, hogy minden gyerek rekordjában látható az ***Osztaly*** mező értéke.

#### *Csoportok kialakítása*

Ha azt szeretnénk, hogy a rekordforrás valamelyik mezője alapján rekordcsoportok alakuljanak ki, kattintsunk a **Csoport hozzáadása** gombra! A megjelenő mezőlistában válasszuk ki a csoportképző mezőt!

Ez példánkban ***Osztaly*** mező lesz. A rekordok sorrendje azonnal megváltozik. Az azonos osztályba járó tanulók rekordjai egymást követően jelennek meg, egy csoportot

alkotnak. A csoportosítás alapjául szolgáló ***Osztaly*** mező értéke már csak a csoport fölött jelenik meg.

Csoportosításkor az ablak alsó részében megjelenik a csoport kezelésre szolgáló **csoportkezelő sáv**. A csoportosításra használt mező vezérlőeleme a törzsből átkerül az automatikusan kialakuló csoportfejlécbe.



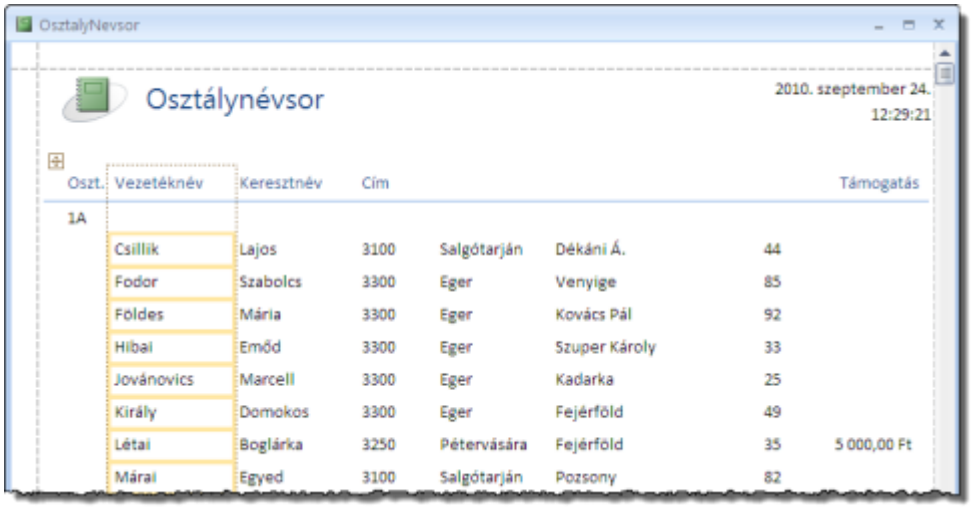

*68. ábra Csoportok kialakítása*

Rossz választás esetén a csoportkezelő sáv jobb oldalán lévő törlés gombbal törölhetjük a csoportot.

### *Rendezés*

Egy csoporton belül megváltoztathatjuk a rekordok sorrendjét a is. Ehhez kattintsunk a csoport alatt a **Rendezés hozzáadása** gombra, majd válasszuk ki a rendezésre használt mezőt!

Példánkban névsorba akarjuk állítani az osztályok tanulónak rekordjait. Először a ***VezetekNev*** mező alapján rendezzünk, majd ismételjük meg a műveletet a ***KeresztNev*** mezővel. A rekordok most már osztályonként csoportosítva, a nevek szerint rendezve jelennek meg.




*69. ábra Csoportosított és rendezett rekordok*

### *Összegzések beállítása*

A következő lépésben összegzéseket készítünk és jelenítünk meg a jelentésben.

Az összegzés mindig valamelyik csoport, egy kiválasztott mezője alapján számított összeg, átlag, darabszám, minimum, maximum érték, szórás, vagy variancia számítás. Ha a jelentést varázslóval hozzuk létre, előfordulhat, hogy a szám tartalmú mezők alapján az Access automatikusan létrehoz, a teljes jelentésre vonatkozó összegzést, amely a jelentéslábban jelenik meg.

Az osztálynévsorokat megjelenítő jelentés végén találni fogunk egy ilyen automatikusan kialakított összegzést, amely az iskolában kiosztott összes támogatást mutatja meg.




*70. ábra Összegzések a jelentésben*

Ha a kialakított csoportokon belül akarunk egy mezőt összegezni, kattintsunk a csoport **csoportkezelő sávjára**, majd az mezőre!

Ezután gördítsük le a menüszalag **Formátum** lapjának **Csoportosítás és összesítés** csoportjában, az **Összesítés** listát, és válasszuk ki a megfelelő műveletet!
A csoport alatt létrejön a csoportláb, és benne megjelenik az összegzett mezőértéket tartalmazó szövegmező.

Összegzést kell kialakítanunk például akkor, ha azt szeretnénk elérni, hogy minden osztály csoportja alatt megjelenjen az osztálylétszám, illetve a gyereknek kiosztott összes támogatás. Ehhez kattintsunk az **Osztaly** csoport **csoportkezelő sávjára,** majd bármelyik rekordban a ***Tamogatas*** mezőre! A menüszalag **Összesítés** listájában kattintsunk az összegzés parancsra!

Ezt követően ismételjük meg a műveletet a ***VezetekNev*** mezővel is, de úgy, hogy az **Összesítés** listában a **Rekordok számlálása** parancsot választjuk

### *Elrendezés, méretezés*

A jelentés szerkezetének beállítása után célszerű megfelelőképpen elrendezni a vezérlőelemeket.

Példánkban jól látszik, hogy a vezérlőelemek túlságosan hosszúak, így a jelentés túlnyúlik a lap szélességén! Állítsuk be a vezérlőelemek szélességét!

A jelentés készítő varázsló, úgynevezett **táblázatos elrendezésben** helyezi el a mezőértékeket megjelenítő vezérlőelemeket, és hozzájuk tartozó feliratokat, címkéket. A táblázatos elrendezés azt jelenti, hogy a címkék az oldalfejben, a vezérlőelemek az oldalfej és oldalláb közötti szakaszokban, úgy helyezkednek el, mintha egy táblázat celláit alkotnák. Ez az elrendezés azért kedvező, mert lehetőséget nyújt a táblázat több szakaszon átívelő oszlopainak együttes formázására.

Ha például az ***Osztaly*** mező oszlopát túl szélesnek találjuk, az oszlop bármelyik kijelölt cellájának szegélyével elvégezhetjük a teljes oszlop szélességének beállítását! Ezzel a technikával olyan széles jelentést alakíthatunk, amely biztosan elfér a laptükörben.

Mivel az Elrendezési nézetben látszik a mezők tartalma is, pontosan fel tudjuk mérni a szükséges oszlopszélességeket. Amikor egy oszlop szélességét szabályozzuk, megváltozik a teljes jelentés szélessége!

Ha szükséges, az egész táblázatot elmozgathatjuk a jelentésben. Ennek akkor van jelentősége, amikor a laptükör szélessége nagyobb, mint a méretezés utáni táblázat. A pozicionálást a táblázat bal felső sarkában megjelenő, négyágú nyilat tartalmazó gombbal végezhetjük el. Ha gombot megfogjuk, a táblázat egyben mozgatható.

### *Címkék átírása*

A jelentés vezérlőelemeihez tartozó címkék alapvetően a felhasználó tájékoztatására, a jelentésben megjelenő adatok magyarázatára szolgálnak. Alapértelmezésben azonban, a kapcsolt mező nevét tartalmazzák, ami nem mindig felel meg feliratként. Az **Elrendezési nézetben** bármelyik címke szövegét átírhatjuk. Nem csak a mezők címkéit, hanem az egész jelentés címét is átírhatjuk. Kattintsunk duplán a címke szövegén és írjuk be az új szöveget!

Példánkban az alábbiak szerint célszerű átírni a címkék szövegét:

| | | |
|---|---|---|
| **Osztaly** | => | **Osztály** |
| **VezetekNev** | => | **Vezetéknév** |
| **KeresztNev** | => | **Keresztnév** |
| **Tel_Irsz** | => | **Cím** |
| **Tel_Nev** | => | *töröljük a szöveget* |
| **LakHelyUtca** | => | *töröljük a szöveget* |
| **LakHelySzam** | => | *töröljük a szöveget* |
| **Tamogatas** | => | **Támogatás** |
| **OsztalyNevsor** | **=>** | **Az osztályok névsora** |

## 11.7 A JELENTÉS VÉGSŐ BEÁLLÍTÁSA TERVEZŐ NÉZETBEN

Mint láttunk az Access 2007 jelentéstervezőjében szinte mindent megvalósíthatunk az **Elrendezési nézetben**. **A Tervező néz**etre csak a jelentés „finomhangolásakor" van szükségünk. A tervező nézet természetesen rengeteg lehetőséget biztosít, hiszen itt a jelentés minden összetevőjét, és formátumát szabályozhatjuk. Most csak három feladatot tekintünk át. Megtanuljuk, hogyan lehet szabályozni **a szakaszok oldalankénti elhelyezked**ését, hogyan tudjuk sorszámozni, úgynevezett **futó összeggel** ellátni a jelentéstörzset, és hogyan lehet további címkék elhelyezésével érthetőbbé tenni a jelentést.

### 11.7.1 Szakaszok és nyomtatott oldalak

Elrendezési nézetben szaggatott vonalak jelezték a laptükör szélességét, de a magasság nem jelent meg, ezért nem láttuk, és nem is szabályoztuk, hogy mi az, ami új oldalra kerül majd a jelentésben. Gyakran előfordul, hogy a csoportokat tartalmazó jelentés egyes szakaszai után mindenképpen új oldalt akarunk kezdeni.

Ha biztosítani akarunk lenni, hogy egy szakasz után új oldal kezdődjön a nyomtatásban, kövessük a következő lépéseket:

- Váltsunk **Tervező nézetre**,
- jelöljük ki a szakaszt, a fölötte található szakasz kijelölő gombbal,
- jelenítsük meg a tulajdonságlapot (F4),
- váltsunk az **Formátum** fülre, keressük meg a **Kötelező új oldal** tulajdonságot, és
- állítsuk be a **Szakasz után** értéket.

Példánkban az **Osztaly lábléc** szakasszal végezzük el a műveletet!

### 11.7.2 Futó összeg elhelyezése

Szintén nem ritka, hogy szeretnénk sorszámozni a jelentésben megjelenő rekordokat.

- Ehhez a megfelelő szakaszban helyezzünk el egy szövegmezőt!

  - **Tervező** nézetben válasszuk a **Tervezés** parancslapot,
  - a **Vezérlők** csoportban kattintsunk a **Beviteli mező** gombra, majd megfelelő szakaszban (általában törzsben) oda, ahol a sorszámot el szeretnénk helyezni.
- Írjuk át, vagy ha nincs rá szükség, töröljük a szövegmezőhöz tartozó címkét!
- Méretezzük és pozícionáljuk a szövegmezőt.
- Jelenítsük meg a Tulajdonságlapot (**F4**), és váltsunk az Adat fülre!
- A **Mező vagy kifejezés** sorba írjuk az **=1** szöveget, **a Futó össz**eg sorban pedig válasszuk a **Csoportonként** opciót!

A futó összeg ezzel elkészült. Nyomtatáskor, a beviteli mezőben, csoportonként újrakezdődő, automatikus sorszámozás jelenik meg.

Példánkban futó összeggel sorszámozhatjuk az osztályok tanulóit.

### 11.7.3 Új címkék lehelyezése

Ha a jelentést további címkékkel akarjuk kiegészíteni,

- **Tervező** nézetben válasszuk a **Tervezés** parancslapot,
- a **Vezérlők** csoportban kattintsunk a **Címke**gombra, majd megfelelő szakaszban oda, ahol a címkét el szeretnénk helyezni.
- Írjuk be a címke szövegét, majd pozícionáljuk megfelelő helyre a címkét!
- Ha szeretnénk elérni, hogy az új címke méretei, és pozíciója igazodjon a jelentés korábban elhelyezett vezérlőihez, jelöljük ki a címkét, majd a Shift billentyűt nyomva tartva a referenciaként szolgáló vezérlőelemet, és menüszalag **Elrendezés** lapjának **Vezérlők igazítása** illetve **Méret** csoportjában, a elhelyezkedő parancsot

Példánkban helyezzünk új az **Osztály létszáma** és az **Összes támogatás** szövegű címkéket az összegzések előtt!

## 11.8 JELENTÉS MENTÉSE

Ha nem szeretnénk, hogy a verejtékes munkával létrehozott jelentés egy pillanat alatt az enyészeté legyen, mentsük munkánkat a szokásos módon!

## 11.9 VÉGSŐ ÁTTEKINTÉS ÉS NYOMTATÁS

A jelentéskészítés célja, hogy az adatok nyomtatásban jelenjenek meg. A nyomtatás közvetlenül az **Office menüből** is elvégezhető, de szerencsésebb erre a célra a jelentés Nyomtatási kép nézetét használni. Ebben a nézetben pontosan úgy jelenik meg a dokumentum, ahogyan papírra kerül majd, Ha valamit elrontottunk a tervezésben, itt minden bizonnyal észrevesszük majd. A **Nyomtatási kép** nézetben megjelenő parancslapon a nyomtatás szokásos tulajdonságait szabályozhatjuk, illetve elindíthatjuk a nyomtatást, de az **Adatok** csoport parancsaival különböző formátumú fájlokba is exportálhatjuk a jelentést.

Ezek között figyelmet érdemel a **Pdf agy XPS** gomb, amivel a népszerű **PDF** állományformátumú dokumentumok hozhatók létre, illetve **Egyebek lista** **Snapshot Viewer** parancsa, amellyel az Access **Snapshot** formátumában exportálhatunk.

Utóbbi kiválóan alkalmas arra, hogy jelentéseinket, az eredeti megjelenést tökéletesen megőrizve e-mailben, vagy más csatornákon továbbítsuk kollégáinknak, vagy üzletfeleinknek.

## 11.10 JELENTÉSEK ELHELYEZÉSE A MENÜ ŰRLAPON

Az űrlapok kapcsán szó esett már az úgynevezett **Kapcsolótáblákról**, az adatbázis űrlapjai közötti navigációt segítő menü űrlapról. A kapcsolótáblák nem csak űrlapok, de jelentések **Jelentés nézetben** történő megtekintésére is kiválóan alkalmasak!

Ha már van kapcsolótábla űrlapunk, és szeretnénk fölvenni egy jelentést a Kapcsolótábla űrlapba, járjunk el a következő lépések szerint!

- Kattintsuk a menüszalag **Adatbáziseszközök** lap **Adatbáziseszközök** csoportjában **Kapcsolótábla-kezelő** parancsra!
- A megjelenő **Kapcsolótábla-kezelő** nevű ablakban kattintsunk a szerkesztés gombra!
- Kattintsunk az **Új** gombra!
- A **Szöveg**-hez írjuk be a megfelelő szöveget (Osztálynévsorok nyomtatása), a **Parancs** listában válasszuk az **Jelentés megnyitása** sort!
- Az **Jelentés** sorban adjuk meg, melyik jelentést szeretnénk megnyitni a paranccsal!
- Kattintsunk az **OK** gombra, majd zárjuk be a szerkesztő ablakát!
- Nyissuk meg a Kapcsolótábla űrlapot, amin megjelenik az új parancs!




*71. ábra Kapcsolótábla-kezelő*

## 11.11 ÖSSZEFOGLALÁS

A jelentések megismerésével szinte teljessé vált az Access adatbázisok kezelésével kapcsolatos ismeretanyagunk. Megtanultuk, hogyan készíthetünk gyorsan és egyszerűen tetszetős, és az adatbázis adatit hűen visszaadó nyomatokat.

A jelentéseket egy előre elkészített rekordforrás (tábla, vagy lekérdezés) alapján, varázslóval alakítottuk ki, de az Elrendezési, és Tervező nézetben tovább finomítottuk tartalmát és megjelenését egyaránt. Megtanultuk, hogy Elrendezési nézetben csoportosíthatók és rendezhetők a rekordok. Összegzéseket építhetünk a jelentésbe, és a táblázatos elrendezésnek köszönhetően egyszerűen megváltoztathatjuk a vezérlőelemekből álló oszlopok és sorok tulajdonságai.

Tervező nézetben áttekintettük a jelentés szakaszait, oldaltöréseket, futó összegeket, és új címkéket készítettük. Megtanultuk, hogyan lehet kinyomtatni, illetve külső állományokba menteni a jelentés nyomtatási képét.

## 11.12 ÖNELLENŐRZŐ KÉRDÉSEK

1. Csak nyomtatásra használhatjuk a jelentéseket?
2. Mit értünk egy jelentés rekordforrása alatt?
3. Hogyan hozunk létre jelentést az Access jelentés varázslójával?
4. Hogyan csoportosíthatja a jelentés rekordforrásának rekordjait?
5. Hogyan hozhat létre összegzést egy csoport mezője alapján?
6. Megváltoztathatja-e a varázslóval létrehozott jelentés címkéinek szövegét?
7. Mi az táblázatos elrendezés?
8. Mi az a futóösszeg?
9. Hogyan tudja elérni, hogy egy jelentés valamelyik szakaszának kinyomtatása oldaltörést eredményezzen?
10. Hogyan tudja egy eljuttatni egy jelentés nyomtatási képét olyan partnereknek, akik nem rendelkeznek hozzáféréssel az adatbázishoz?

# 12. ÖSSZEFOGLALÁS

## 12.1 A KURZUSBAN KITŰZÖTT CÉLOK ÖSSZEFOGLALÁSA

Tananyagunk 2-11 leckéiben Ön megismerte az adatbázis-tervezés és -kezelés alapvető fogalmait. Megérti a relációs adatbázisok szerkezetét, képes kisebb adatbázisok megtervezésére. Ismeri a Microsoft Access relációs adatbázis-kezelő rendszer működését, meg tudja valósítani a megtervezett adatbázisokat, tudja megfelelően kezelni az adatokat, képes azokat a kívánt szempontok szerint kiválogatni, rendszerezni, képernyőn, vagy nyomtatott formában megjeleníteni.

## 12.2 A TANANYAGBAN TANULTAK RÉSZLETES ÖSSZEFOGLALÁSA

### 12.2.1 Az adatbázis-kezelés alapfogalmai

Tankönyvünk első leckéjében olyan, a későbbi leckék tanulása során nélkülözhetetlen alapfogalmakkal ismerkedtünk meg, mint az adat, információ, egyed, egyedtípus, tulajdonság, tulajdonságtípus, kapcsolat, kapcsolattípus. A lecke bemutatta a relációs adatmodellt, ismertette az adatmodell szabályait, tárgyalta az azonosító, összetett kulcs, idegen kulcs, többértékű, és ismétlődő értékű mezők fogalmát.

### 12.2.2 Relációs adatbázisok tervezése

Ebben a leckében a redundancia mentes relációs adatbázisok kialakításának problémáival, és egy lehetséges módszerével ismerkedtünk meg. Megtanultunk a mik a funkcionális függések, és milyen kapcsolat van a 3NF és a teljes funkcionális függés között. Láttuk, hogyan lehet táblák felosztásával megszűntetni a részleges és tranzitív függéseket, megtanultuk hogyan tárolhatjuk a normalizálás során keletkező táblák közötti kapcsolatokat.

### 12.2.3 A Microsoft Access adatbázisok szerkezete, és kezelése

Tananyagunk harmadik leckéje a Microsoft Access relációs adatbázis-kezelő rendszer felületét és működését ismertette. Megismertük az accdb adatbázisokat alkotó objektumok jelentőségét, betekintést nyertünk kezelésük alapjaiba.

### 12.2.4 Microsoft Access adatbázisok kialakítása

A „Microsoft Access adatbázisok kialakítása” című leckében megtanultuk, hogyan valósíthatók meg az Access felületén azok az adatbázis tervek, amelyek készítéséről a 2. lecke szól. Elsajátítottuk a táblák kialakításával kapcsolatos tudnivalókat, megismerkedtünk a mezők adattípusának jelentőségével, az idegen kulcsok típusának meghatározásával, az elsődleges kulcsok jelölésének módjával.

### 12.2.5 Táblák közötti kapcsolatok kialakítása

Ez a lecke az Access adatbázisok fontos, bár sokszor elhanyagolt lehetőségét, kapcsolatok kialakítását mutatta be. Megvilágítottuk a kapcsolatok jelentőségét, megismerkedtünk, a kapcsolatkialakítás eszközével. Megismertük az egyik legfontosabb kapcsolattulajdonsá-

got, a hivatkozási integritás megőrzését. Láttuk, hogyan biztosítja az Access a helyes idegen kulcsokat, megtanultuk, mit jelent a kapcsolt mezők kaszkádolt törlése, és frissítése.

### 12.2.6 Rekordok bevitele, rendezése, keresése, szűrése

Az adatbázisok tervezése és létrehozása után, ebben a leckében a tényleges adatkezeléssel kezdtünk foglalkozni. Megtanultuk, hogyan rögzíthetjük, módosíthatjuk, illetve törölhetjük egy tábla rekordjait. Elsajátítottuk a keresés, rendezés, és szűrés technikáit.

### 12.2.7 Lekérdezések készítése

A lekérdezések készítéséről szóló lecke bemutatta a különböző táblákban tárolt, de egymással kapcsolódó rekordok kezelésének lehetőségét. Megismerkedtünk a lekérdezések kialakításának módszerével, a rekordforrás, a forráshalmaz, az illesztési tulajdonságok, és az eredményhalmaz fogalmával. Megtanultuk, hogyan szabályozhatjuk az eredményhalmaz tartalmát, mezők, feltételek, és lekérdezés tulajdonságok segítségével.

### 12.2.8 Speciális feladatok lekérdezésekkel

Ez a lecke bemutatta, hogyan válnak a lekérdezések az adatkezelés legfontosabb objektumaivá. Megtanultuk, hogyan készíthetünk statisztikákat, és hogyan végezhetünk tömeges rekordkezelő műveleteket a lekérdezések segítségével.

### 12.2.9 Az Access adatbázisok kezelőfelületének kialakítása

Utolsó előtti leckénk az Access adatbázisok kezelőfelületének kialakítását mutatta be. A leckében megtanultuk, hogyan alakíthatjuk ki a táblák, és lekérdezések eredményhalmazainak megjelenítésére alkalmas űrlapokat, hogyan tehetjük kényelmessé az igen/nem típusú mezők, és az idegen kulcsok kezelését. Láttuk, hogyan rejthetők el a felhasználó elől az adatbázis tényleges objektumai, hogyan biztosítható úgynevezett kapcsolótábla űrlapokkal, a kezelőfelület elemei közötti navigáció.

### 12.2.10 Tárolt adatok nyomtatott megjelenítése

Tananyagunk utolsó leckéje az adatbázisban tárolt adatok nyomtatott listák formájában történő megjelenítését tárgyalta. Megismertük a tetszetős nyomtatási képek kialakításának gyors lehetőségét. Megtanultuk, hogyan módosíthatjuk a varázslóval készült jelentések megjelenését. Megismertük a rekordcsoportok kialakításának, rendezésének technikáját, megtanultuk a rész-, és futóösszegek kialakításának módját.

# 13. KIEGÉSZÍTÉSEK

## 13.1 IRODALOMJEGYZÉK

### 13.1.1 Könyv

HALASSY, Béla: *Az adatbázis-tervezés alapjai és titkai*. Budapest, IDG Magyarországi Lapkiadó Kft, 1995.

A. A. AVLICINO, JOYCE COX - CURTIS FRYE, JOAN PREPPERNAU: *Microsoft Office 2007 lépésről lépésre*. Budapest, Szak Kiadó, 2007.

### 13.1.2 Elektronikus dokumentumok / források

MICROSFT: *A Microsoft Access 2007 hivatalos honlapja*. [elektronikus dokumentum] [2010.június 3.]
<URL: http://office.microsoft.com/hu-hu/access/FX100487571038.aspx >

MICROSOFT: *A Microsoft Office Fluent felhasználói felület*. MICROSOFT, [elektronikus dokumentum] [2010.június3.]
<URL: http://office.microsoft.com/hu-hu/help/HA101679411038.aspx >

MICROSOFT: *Access 2003-parancsok az Access 2007 alkalmazásban*. MICROSOFT, [elektronikus dokumentum] [2010.június3.]
<URL: http://office.microsoft.com/hu-hu/access/HA101666061038.aspx>