

Bártfai Barnabás

Access 2010 zsebkönyv

Bártfai Barnabás

Access 2010

ZSEBKÖNYV

BBS-INFO Kiadó, 2010.

© Bártfai Barnabás, 2010.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a szerző írásbeli hozzájárulásával történhet.

A betűtípus elnevezések, a Microsoft, a Windows, a Windows logo, az Office és az Access bejegyzett védjegyek.

A Microsoft Szoftver Információ telefonszáma: (06-1) 267-46-36

A könyv nagyobb mennyiségben megrendelhető a kiadónál:

BBS-INFO Kiadó, 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére, mint minden könyvben, ebben is előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző, sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

ISBN 978-963-9425-70-5

Kiadja a BBS-INFO Kft.

1630 Budapest, Pf. 21.

Felelős kiadó: a BBS-INFO Kft. ügyvezetője

Nyomdai munkák: Biró Family Nyomda

Felelős vezető: Biró Krisztián

TARTALOMJEGYZÉK

1. Bevezető	11
2. Fogalmak	13
2.1. Az adatbázisok	13
2.2. Adatbázis szerkezetek	17
2.2.1. Hierarchikus adatbázis-szerkezet	17
2.2.2. Hálós adatbázis-szerkezet.....	17
2.2.3. Relációs adatbázis-szerkezet	18
2.3. Az Access adatbázisok részei	20
2.4. Hogyan épüljön fel egy adatbázis?.....	21
2.5. Adatbázisok tervezése.....	21
2.5.1. Általános tervezési elvek, adattípusok..	22
2.5.2. Indexelések	22
2.5.3. Elnevezések.....	24
2.5.4. Normalizálás.....	24
2.5.5. Kapcsolatok típusai.....	27
2.5.6. Mezőtulajdonságok.....	29
2.6. Adatformátumok	33
3. Alapfunkciók	36
3.1. A program indítása.....	36
3.1.1. Meglévő adatbázis megnyitása	37
3.1.2. Új, üres adatbázis létrehozása	37
3.1.3. Új adatbázist létrehozása sablonok alapján.....	38

3.2. Kilépés, bezárás.....	39
3.3. Alapvető kezelési feladatok.....	40
3.3.1. Szalagok	40
3.3.2. Helyi menük	41
3.3.3. Súlyó	41
3.3.4. Tevékenység visszaállítása, utolsó művelet ismétlése	42
3.3.5. Navigációs ablak	42
3.4. Ablakok és nézetek	46
3.5. Nyomtatás.....	47
4. Táblák.....	49
4.1. Táblák létrehozása	49
4.1.1. Tábla létrehozása adatok beírásával.....	50
4.1.2. Tábla létrehozása sablonból.....	52
4.1.3. Tábla létrehozása tervező nézetben.....	53
4.1.4. Tábla importálása.....	55
4.2. Meglévő táblák megtekintése, szerkesztése	56
4.3. Adatbevitel adattáblába	57
4.4. Rekordok kezelése	58
4.5. Táblák formázása, megjelenítésének módosítása	60
4.6. Táblák szerkezetének módosítása.....	62
4.6.1. Mezők mozgatása, másolása	63
4.6.2. Új mező beszúrása	63
4.6.3. Mező törlése.....	64
4.6.4. Mezőnév megváltoztatása	64
4.6.5. Adattípus megváltoztatása	65
4.6.6. Mezőtulajdonság megváltoztatása	65
4.7. Más táblából való válogatást lehetővé tevő mezők létrehozása	65
4.8. Táblák másolása, törlése és átnevezése.....	69
4.9. Keresés.....	70

4.10.	Csere.....	71
4.11.	Rendezés.....	72
4.12.	Szűrés.....	72
4.13.	Frissítés	75
5.	Kapcsolatok	76
5.1.	Kapcsolatok értelmezése és jelentősége	76
5.2.	A kapcsolatok kialakítása	78
5.3.	Közvetlen kapcsolatok	80
5.4.	Kapcsolatjellemzők megváltoztatása	81
5.5.	Kapcsolt táblák használata	82
6.	Lekérdezések.....	83
6.1.	Lekérdezés nézetek.....	84
6.2.	A lekérdezések típusai	84
6.2.1.	Választó lekérdezés.....	84
6.2.2.	Keresztábrás lekérdezés	85
6.2.3.	Táblakészítő lekérdezés.....	85
6.2.4.	Frissítő lekérdezés.....	85
6.2.5.	Hozzáfűző lekérdezés	86
6.2.6.	Törlő lekérdezés	86
6.2.7.	SQL lekérdezés.....	86
6.2.8.	Egyesítő lekérdezés.....	86
6.2.9.	Átadó lekérdezés.....	87
6.2.10.	Adatdefiniáló lekérdezés	87
6.3.	SQL parancsok	87
6.4.	Lekérdezések frissítése, futtatása.....	90
6.5.	Lekérdezések létrehozása	91
6.6.	Lekérdezések módosítása	94
6.6.1.	Táblák, lekérdezések hozzáadása	95
6.6.2.	Táblák törlése lekérdezésből.....	95
6.6.3.	Mező hozzáadása, módosítása, törlése .	96
6.6.4.	Mezősorrend megváltoztatása	97
6.6.5.	Mezők megjelenítésének tiltása.....	97

6.6.6.	Mezőtulajdonságok megváltoztatása	97
6.6.7.	Egyéb mezőműveletek	98
6.6.8.	Lekérdezés típusának megváltoztatása	98
6.7.	Összetett lekérdezések	99
6.8.	Adatok rendezése lekérdezés segítségével	99
6.9.	Feltételek meghatározása	99
6.9.1.	Összehasonlító operátorok	100
6.9.2.	Logikai operátorok	100
6.9.3.	Aritmetikai operátorok	101
6.9.4.	Egyéb operátorok	101
6.9.5.	Több mezőtől függő feltételmegadás ..	101
6.9.6.	Számított kifejezések a lekérdezésekben	102
6.10.	Összesítések és összegzések	103
6.10.1.	Feltételek viselkedése az összesítésekben	105
6.11.	Egyéb lekérdezések készítése	106
6.11.1.	Tábla adatainak módosítása frissítő lekérdezéssel	106
6.11.2.	Rekordok törlése törlő lekérdezéssel	106
6.11.3.	Új táblák létrehozása táblakészítő lekérdezéssel	107
6.11.4.	Új rekordok létrehozása hozzáfűző lekérdezéssel	107
6.11.5.	Keresztábrás lekérdezés készítése ..	108
6.11.6.	Azonos elemek keresése lekérdezéssel	108
6.11.7.	Nem egyező elemek keresése lekérdezéssel	109
6.12.	Felhasználói paraméterek használata lekérdezésekben	109

6.13.	Lekérdezés eredményének megtekintése	110
6.14.	Lekérdezések törlése, átnevezése	111
7.	Űrlapok.....	113
7.1.	Űrlap nézetek	114
7.2.	Adatbevitel táblára űrlap segítségével.....	114
7.3.	Az űrlapok részei	115
7.4.	A segédűrlap	115
7.5.	Űrlapok készítése.....	115
7.5.1.	Űrlapok készítése automatikusan.....	115
7.5.2.	Űrlapok készítése varázslóval.....	117
7.5.3.	Űrlapok készítése tervezéssel.....	120
7.6.	Az űrlapokon használható elemek	121
7.7.	Meglévő űrlapok szerkesztése	122
7.7.1.	Új elem űrlapra helyezése.....	123
7.7.2.	Meglévő elem elhelyezkedésének módosítása	124
7.7.3.	Űrlapstílusok	126
7.7.4.	Elemek betűtípusai, méretei, igazításai.....	127
7.7.5.	Elem tulajdonságainak megváltoztatása.....	127
7.7.6.	Objektumok beszúrása űrlapra	131
7.7.7.	Bejárasi sorrend	131
7.8.	Diagramok készítése.....	132
8.	Jelentések.....	135
8.1.	Jelentés készítése	135
8.1.1.	Jelentés automatikus készítése	135
8.1.2.	Jelentés készítése varázsló segítségével.....	135
8.1.3.	Jelentés készítése egyedi tervezéssel ...	140
8.2.	Jelentés módosítása.....	140

8.3. Jelentés megtekintése	145
8.4. Jelentés nyomtatása	145
9. Makrók és modulok.....	146
9.1. Új makró készítése	146
9.2. Meglévő makró módosítása	148
9.3. Makró futtatása	148
9.4. Makró törlése és átnevezése	149
9.5. Makró eseményhez kapcsolása	149
9.6. Modulok.....	150
10. Egyéb lehetőségek.....	151
10.1. Küldés és exportálás	151
10.2. Importálás	153
10.3. Adatbázis tömörítése és helyreállítása	155
10.4. Biztonsági mentés.....	156
10.5. Régi adatbázis konvertálása.....	156
10.6. Adatlap	157
10.7. Titkosítás és korlátozás jelszóval.....	158
10.8. Adatgyűjtés E-mailben	159
10.9. Megosztott adatok kezelése SharePointtal.....	159
11. Testreszabás	161
11.1. Beállítások	161
11.2. A gyorselérési eszköztár és a szalagok módosítása	166
11.2.1. A gyorselérési eszköztár módosítása.....	166
11.2.2. A szalagok módosítása	169

1. Bevezető

E könyvecske segítségével megismerhetjük az Office 2010 adatbázis-kezelő programjának, az Access-nek a kezelését. A könyvet bátran ajánljuk akár kezdőknek is, de hasznos lehet azok számára is, akik ECDL vagy egyéb vizsgára készülnek, vagy akik a program további lehetőségeivel kívánnak megismerkedni.

Továbbra is fontosnak tartottuk, hogy ne azt mutassuk meg, hogy egy adott programfunkció mire való, hanem azt, hogy egy adott feladatot miként tudunk megoldani.

A könyvben nem tárgyaljuk az alapismereti részeket, feltételezzük, hogy a Tisztelt Olvasó már tisztában van a Windows kezelésével, és az alapvető számítástechnikai fogalmakkal. Ha ez mégsem így lenne, úgy ajánljuk az utolsó oldalon lévő könyvajánlóban szereplő, illetve a kiadó weblapján lévő kezdő, illetve Windows-ról szóló könyveket.

2. Fogalmak

A Microsoft Access egy Windows alatt futó relációs adatbázis-kezelő alkalmazás. Segítségével létrehozhatunk munkánkhoz kapcsolódó adatnyilvántartásokat, adatnyilvántartó rendszereket.

2.1. Az adatbázisok

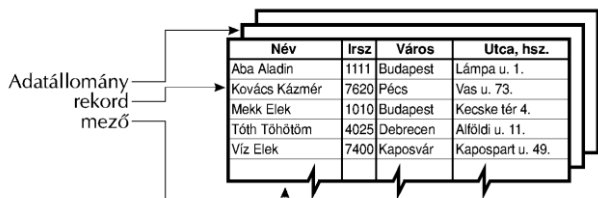
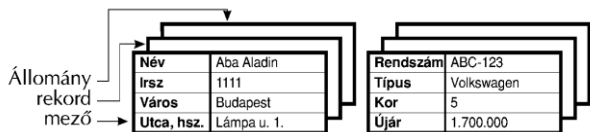
Az adatok feldolgozásának egyik legelterjedtebb módja az adatbázis-kezelés. Mivel az adatokat nem tárolhatjuk egyenként, külön-külön (hiszen akkor azok kezelése szinte lehetetlen volna), ezért szükséges őket valamilyen módon egy jól szervezett állományban elhelyezni. Az **adatállomány** tehát egy olyan összefüggő adathalmaz, melyben minden olyan adat megtalálható, amire egy bizonyos cél megvalósítása érdekében szükséges lehet. Mivel azonban több okból is szükségünk lehet különböző adatokra, így a számítógépünkön többféle egymástól független adatállományt is tárolhatunk. Néha azonban előfordulhat, hogy ezen adatok között kapcsolatot kell találnunk, s ilyenkor

nélkülözhetetlen azok ismételt tárolása. Az egymással kapcsolatba hozható adatállományok összességét nevezzük **adatbázisnak**. A fizikai megvalósulás kis mértékben eltér az elvtől. Az Access ugyanis – modernebb program lévén – egy állományban (fájlban) egy teljes adatbázist tárol, így az állomány helyett inkább a tábla kifejezést használjuk. Access esetében az adatbázis-fájl részét képezik továbbá az adatbevitelt segítő űrlapok, az adatok különféle szempontok szerinti kiválogatását segítő lekérdezések, a megjelenítést és nyomtatást áttekinthetővé tevő jelentések, az automatizálásért felelős makrók, és még jónéhány egyéb elem is.

Mivel az önálló táblák igen sokféle adat tárolását szolgálják, szükséges valamilyen szerkezet felépítése az adatállományon belül. E szerkezet kezelését az adatbázis-kezelő program végzi, így a felhasználónak illetve a különböző alkalmazásoknak csupán a szerkezetét kell ismerniük, az adatbázis fizikai tárolásának módját nem. A felépített illetve felépítendő adatbázis szerkezetét még az adatok felvitele előtt meg kell határozni, így ilyenkor kell eldönteni azt, hogy milyen adatokat szeretnénk tárolni az adatállományokban és melyiket melyik táblában.

Az adatállományok szerkezetére vonatkozóan azonban néhány kifejezés megismerése nélkülözhetetlen.

Az adatállományban **rekordnak** nevezzük az egy egységet leíró különböző jellemzőket. (Pl. egy rekord az adatállományban egy ember neve, a hozzá tartozó irányítószámmal, városnévvel, illetve utca, házzszámmal.)



Mező alatt az adatbázis összes elemének egyazon jellemző adatát értjük. (Mező például a név.)

A kétdimenziós adatállományokat ezekből kifolyólag lehetséges táblázatos formában is ábrázolni, ahol a sorok jelképezik a rekordokat, az oszlopok pedig a mezőket. (Innen ered az állomány helyett használt tábla elnevezés.) Egy rekord a táblában egy sornak felel meg, egy mező pedig egy oszlopnak.

Amennyiben egy adatállománnyal folyamatosan dolgozunk, a rekordok számát szaporítjuk, esetleg csökkentjük. Ezzel az adatállomány na-

gyobbá illetve kisebbé válik. A mezők számának megváltoztatása nem jellemző feladat. A szerkezetet még az adatok bevétele előtt célszerű kialakítani, hiszen egy utólagos módosítás az adatbázis használó programok megváltoztatásának szükségességét vonná maga után. Mindemellett egy adatállományban túl sok mező elhelyezése sem célszerű, ez ugyanis bonyolultabbá teheti az adatállományt.

Amikor létrehozunk egy táblát, tulajdonképpen azt határozzuk meg, hogy abban az adatok mely jellemzői kerüljenek tárolásra. Fontos tehát, hogy az összes szükséges jellemző tárolása megtörténjen, de felesleges és többszörös adatokat ne tároljunk. Az adatok többszörös tárolása ugyanis **redundanciához** vezet, amely azon kívül, hogy megnöveli az adatállomány méretét, ellentmondásokat is eredményezhet. Ennek megfelelően tehát egy adatállományban azon adatokat, amelyek a tárolt jellemzők felhasználásával egyértelműen számítható soha se tároljuk el. (Például ne tároljunk egy személyről személyi számot, születési időt és kort, hiszen a személyi szám elegendő mind a születési idő, mind pedig a kor meghatározásához.)

2.2. Adatbázis szerkezetek

Egy-egy önálló tábla sok esetben nem tartalmaz elegendő adatot bizonyos információk meghatározásához, az összes adat egy táblán történő tárolása pedig bonyolult vagy megvalósíthatatlan. Ilyenkor válhat szükségessé a táblák együttes kezelése, amely valamilyen adatbázis szerkezetben realizálódik.

2.2.1. Hierarchikus adatbázis-szerkezet

Ebben a szerkezetben a fa-struktúrájához hasonló módon valósul meg az adatok tárolása. Az adatok között ún. szülő-gyermek kapcsolatot hoznak létre oly módon, hogy az adatoknak tetszőleges számú leszármazottja, de csak egy őse lehet. Ezt a szerkezetet személyi számítógépek esetén csak igen ritkán alkalmazzák, főként nagygépes környezetben fordul elő.

2.2.2. Hálós adatbázis-szerkezet

A hálós adatbázis nagyban hasonlít a hierarchikus szerkezethez, de itt nem csak több leszármazottja, hanem több őse is lehet az adatnak. A többnyire szintén nagygépes környezetben használt szerkezet hátránya, hogy a használathoz bonyolult láncolólisták szükségesek, így nagy a tárigényük.

2.2.3. Relációs adatbázis-szerkezet

Manapság legelterjedtebben a relációs adatbázis-szerkezetet alkalmazzák, hiszen ezen szerkezet könnyen illeszthető a korábban illusztrált táblázatos leírásra. A relációs adatbázisban az önálló táblák többnyire csak azon adatokat tartalmazzák, melyeknek közül van egymáshoz, s használatuk is többnyire egyidőben történik. A különböző jellegű, de mégis kapcsolatba hozható adathalmazokat önálló táblákon tároljuk, amely táblák között egy azonos adatot tartalmazó mező tartja a kapcsolatot. Ezt nevezzük **kapcsolómezőnek**. Mivel ezen kapcsolómezőnek a kapcsolat irányában egyértelmű azonosítást kell megvalósítani, célszerű, ha a kapcsoló táblának ez az azonosítómezője is. Az azonosítómező a rekordok egyértelmű azonosítását szolgálja, így ennek adattartalma ilyen esetben nem ismétlődhet. (A gyakorlatban azonban nem minden adatbázisban teljesül az a feltétel, hogy az adatbázis bármely mezője is egyértelműen azonosít, így ilyenkor egyértelmű kapcsolat nem hozható létre.)

Ha például az ábrán látható táblák felhasználásával a budapesti autótulajdonosok gépkocsi típusait szeretnénk megtudni, akkor ez a következőképpen valósítható meg:

Megnézzük, hogy az első táblán mely rekordok esetében szerepel a város mezőben Budapest. Az