

Bártfai Barnabás

Access 2019

ZSEBKÖNYV

Bártfai Barnabás

Access 2019

ZSEBKÖNYV

BBS-INFO Kiadó, 2019.

© Bártfai Barnabás, 2019.

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, WWW.BBS.HU 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.

Papírkönyv ISBN 978-615-5477-74-4

E-book ISBN 978-615-5477-75-1

Kiadja a BBS-INFO Kft.

Felelős kiadó: a BBS-INFO Kft. ügyvezetője

Nyomdai munkák: Biró Family Nyomda

Felelős vezető: Biró Krisztián

TARTALOMJEGYZÉK

1. Bevezető	11
2. Fogalmak	12
2.1. Az adatbázisok	12
2.2. Adatbázis szerkezetek	16
2.2.1. Hierarchikus adatbázis-szerkezet	16
2.2.2. Hálós adatbázis-szerkezet.....	16
2.2.3. Relációs adatbázis-szerkezet	17
2.3. Az Access adatbázisok részei	19
2.4. Hogyan épüljön fel egy adatbázis?.....	20
2.5. Adatbázisok tervezése.....	20
2.5.1. Általános tervezési elvek, adattípusok..	21
2.5.2. Indexelések	21
2.5.3. Elnevezések.....	23
2.5.4. Normalizálás.....	23
2.5.5. Kapcsolatok típusai.....	26
2.5.6. Mezőtulajdonságok.....	28
2.6. Adatformátumok	30
3. Alapfunkciók	35
3.1. A program indítása	35
3.1.1. Meglévő adatbázis megnyitása	36
3.1.2. Új, üres adatbázis létrehozása	38
3.1.3. Új adatbázist létrehozása sablonok alapján.....	39

3.2. Kilépés, bezárás.....	40
3.3. Alapvető kezelési feladatok.....	40
3.3.1. Szalagok	41
3.3.2. Helyi menük	42
3.3.3. Súlyó	42
3.3.4. Tevékenység visszaállítása, utolsó művelet ismétlése	43
3.3.5. Navigációs ablak	43
3.4. Ablakok és nézetek	47
3.5. Nyomtatás.....	48
4. Táblák.....	50
4.1. Táblák létrehozása	50
4.1.1. Tábla létrehozása adatok beírásával.....	51
4.1.2. Tábla létrehozása sablonból.....	53
4.1.3. Tábla létrehozása tervező nézetben.....	54
4.1.4. Tábla importálása.....	56
4.2. Meglévő táblák megtekintése, szerkesztése	57
4.3. Adatbevitel adattáblába	58
4.4. Rekordok kezelése	59
4.5. Táblák formázása, megjelenítésének módosítása	61
4.6. Táblák szerkezetének módosítása.....	63
4.6.1. Mezők mozgatása, másolása	64
4.6.2. Új mező beszúrása	65
4.6.3. Mező törlése.....	65
4.6.4. Mezőnév megváltoztatása	66
4.6.5. Adattípus megváltoztatása	66
4.6.6. Mezőtulajdonság megváltoztatása	66
4.7. Más táblából való válogatást lehetővé tevő mezők létrehozása	67
4.8. Táblák másolása, törlése és átnevezése.....	72
4.9. Keresés.....	73

4.10.	Csere.....	74
4.11.	Rendezés.....	74
4.12.	Szűrés.....	75
4.13.	Frissítés	78
5.	Kapcsolatok	79
5.1.	Kapcsolatok értelmezése és jelentősége	79
5.2.	A kapcsolatok kialakítása	81
5.3.	Közvetlen kapcsolatok	83
5.4.	Kapcsolatjellemzők megváltoztatása	84
5.5.	Kapcsolt táblák használata	85
6.	Lekérdezések.....	86
6.1.	Lekérdezés nézetek.....	87
6.2.	A lekérdezések típusai	86
6.2.1.	Választó lekérdezés.....	87
6.2.2.	Keresztábrás lekérdezés	88
6.2.3.	Táblakészítő lekérdezés.....	88
6.2.4.	Frissítő lekérdezés.....	88
6.2.5.	Hozzáfűző lekérdezés	89
6.2.6.	Törlő lekérdezés	89
6.2.7.	SQL lekérdezés.....	89
6.2.8.	Egyesítő lekérdezés.....	89
6.2.9.	Átadó lekérdezés.....	90
6.2.10.	Adatdefiniáló lekérdezés.....	90
6.3.	SQL parancsok	90
6.4.	Lekérdezések frissítése, futtatása.....	93
6.5.	Lekérdezések létrehozása	94
6.6.	Lekérdezések módosítása	97
6.6.1.	Táblák, lekérdezések hozzáadása	98
6.6.2.	Táblák törlése lekérdezésből.....	98
6.6.3.	Mező hozzáadása, módosítása, törlése .	99
6.6.4.	Mezősorrend megváltoztatása	100
6.6.5.	Mezők megjelenítésének tiltása.....	100

6.6.6.	Mezőtulajdonságok megváltoztatása ..	100
6.6.7.	Egyéb mezőműveletek	101
6.6.8.	Lekérdezés típusának megváltoztatása	101
6.7.	Összetett lekérdezések	102
6.8.	Adatok rendezése lekérdezés segítségével	102
6.9.	Feltételek meghatározása	102
6.9.1.	Összehasonlító operátorok	103
6.9.2.	Logikai operátorok	103
6.9.3.	Aritmetikai operátorok	104
6.9.4.	Egyéb operátorok	104
6.9.5.	Több mezőtől függő feltételmegadás ..	104
6.9.6.	Számított kifejezések a lekérdezésekben	105
6.10.	Összesítések és összegzések	106
6.10.1.	Feltételek viselkedése az összesítésekben	108
6.11.	Egyéb lekérdezések készítése	109
6.11.1.	Tábla adatainak módosítása frissítő lekérdezéssel	109
6.11.2.	Rekordok törlése törlő lekérdezéssel	109
6.11.3.	Új táblák létrehozása táblakészítő lekérdezéssel	110
6.11.4.	Új rekordok létrehozása hozzáfűző lekérdezéssel	110
6.11.5.	Keresztábrás lekérdezés készítése	111
6.11.6.	Azonos elemek keresése lekérdezéssel	111
6.11.7.	Nem egyező elemek keresése lekérdezéssel	112
6.12.	Felhasználói paraméterek használata lekérdezésekben	112

6.13. Lekérdezés eredményének megtekintése.....	113
6.14. Lekérdezések törlése, átnevezése	114
7. Űrlapok.....	116
7.1. Űrlap nézetek	117
7.2. Adatbevitel táblára űrlap segítségével.....	117
7.3. Az űrlapok részei	118
7.4. A segédűrlap	118
7.5. Űrlapok készítése.....	118
7.5.1. Űrlapok készítése automatikusan.....	118
7.5.2. Űrlapok készítése varázslóval.....	120
7.5.3. Űrlapok készítése tervezéssel.....	122
7.6. Az űrlapokon használható elemek	123
7.7. Meglévő űrlapok szerkesztése	124
7.7.1. Új elem űrlapra helyezése.....	125
7.7.2. Meglévő elem elhelyezkedésének módosítása	127
7.7.3. Űrlapstílusok	128
7.7.4. Elemek betűtípusai, méretei, igazításai.....	129
7.7.5. Elem tulajdonságainak megváltoztatása.....	129
7.7.6. Objektumok beszúrása űrlapra	132
7.7.7. Bejárési sorrend	133
7.8. Diagramok készítése.....	134
8. Jelentések.....	137
8.1. Jelentés készítése	137
8.1.1. Jelentés automatikus készítése	137
8.1.2. Jelentés készítése varázsló segítségével.....	137
8.1.3. Jelentés készítése egyedi tervezéssel ...	141
8.2. Jelentés módosítása.....	141

8.3. Jelentés megtekintése	146
8.4. Jelentés nyomtatása	146
9. Makrók és modulok.....	147
9.1. Új makró készítése	147
9.2. Meglévő makró módosítása	149
9.3. Makró futtatása	149
9.4. Makró törlése és átnevezése	150
9.5. Makró eseményhez kapcsolása	150
9.6. Modulok.....	151
10. Egyéb lehetőségek.....	152
10.1. Küldés és exportálás.....	152
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. Megosztott adatok kezelése SharePointtal	159
11. Testreszabás	161
11.1. Beállítások.....	161
11.2. A gyorselérési eszköztár és a szalagok módosítása.....	161
11.2.1. A gyorselérési eszköztár módosítása.....	165
11.2.2. A szalagok módosítása	168

1. Bevezető

E könyvecske segítségével megismerhetjük az Office 2019 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

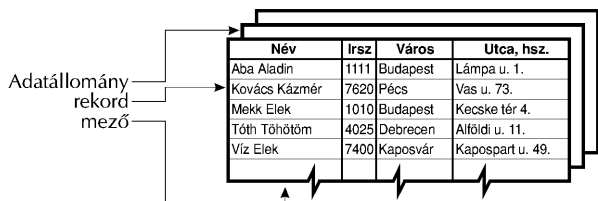
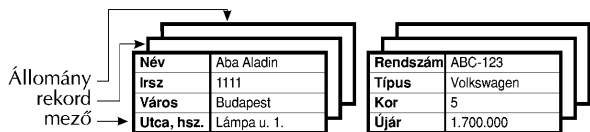
Az adatok feldolgozásának egyik legelterjedtebb módja az adatbázis-kezelés. Mivel az adatokat nem tárolhatjuk egyenként, külön-külön (hiszen akkor azok kezelése szinte lehetetlen volna), ezért szükséges őket valamilyen módon egy jól szervezett állományban elhelyezni. Az **adatállomány** tehát egy olyan 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 egy-azon 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