

Az adatbáziskezelés elemei

**Pázmány Péter Katolikus Egyetem
Jog- és Államtudományi Kar**

**Összeállította: Groma Sarolt
2003**

Az adatbázis-kezelés elemei

Alapfogalmak

(Adatbázis I. A következő félévben még egy alkalommal az adatbázisokkal fogunk foglalkozni.)

Adatbázisokkal a mindennapi életünkben nap mint nap és várhatóan egyre gyakrabban találkozunk: akár a MÁV menetrendet vagy a MATÁV telefonkönyvet nézzük, akár bankügyleteinket intézzük az interneten keresztül. De az egyetemi életben sem nélkülözhetjük ezeket: a Neptun rendszer, a könyvtári adatbázis, a diákfórumok mögött mind-mind adatbázis húzódik. Speciálisan a jogáshallgatóknak szólna a törvényhez, joganyagokhoz, hivatkozásokhoz is most már adatbáziskezelésen keresztül vezet az út, így nemcsak hasznos, de szükséges is, hogy az adatbáziskezelés elemeivel minden hallgató tisztában legyen. Ahhoz, hogy eltájékozódjunk a témában, néhány alapvető fogalommal tisztában kell lennünk:

Adat: "jelentéssel" bíró mennyiség, karaktersorozat

(a 120 önmagában nem adat, csak ha tudjuk, hogy ez a raktárban lévő biciklik száma)

Adatbázis: adatoknak olyan rendszerezett formában történő tárolása, amely magában foglalja az adatok szerkezetének és egymáshoz való kapcsolatának leírását is.

Táblázat: Tulajdonságokkal leírt egyedek rendezett halmaza

Sor: A táblázat egy egyede (rekord)

Oszlop: Az egyedek egy tulajdonsága (mező)

Adattípusok:

szöveges (tetszőleges karaktersorozat)

dátum (csak valós dátumértéket elfogadó - nemzeti dátumformátumok megadása)

numerikus (számok)

logikai (igen-nem, kétértékű)

Azonosító: a sor (rekord) azonosítását biztosító adat (mező, tétel)

tulajdonságai:

- minden rekordnak van
- különböző rekordoknak különböző

Objektumok:

- Táblák

Az adatok rendezett tárolására szolgáló objektumok. Az adatbázisban előforduló minden adat valamilyen táblában szerepel. Jól szervezett adatbázisban a nem azonosító adatok csak 1-szer szerepelnek.

- Lekérdezések
A lekérdezések lényegében kis programok, melyek kapcsolatok és logikai kifejezések alapján rendezett, szűrt, válogatott adatokat képesek szolgáltatni, a szerkezeti tulajdonságok megtartásával.
- Űrlapok
A felhasználói, adatkezelői hozzáférési felület az adatbázis adataihoz (megfelelően megfogalmazva és áttekinthetőséget, könnyen kezelhetőséget biztosítva).
- Jelentések
A jelentések lényegében kis programok, melyek kapcsolatok és logikai kifejezések alapján rendezett, szűrt, válogatott adatokat képesek szolgáltatni, áttekinthető, nyomtatható formában, a szerkezeti tulajdonságok elvesztésével.
- Külső elemek
Bizonyos adatbáziskezelők be tudnak illeszteni az adatbázisba „külső elemeket (file-okat)”, mint például kép, hang stb., ezek azonban nem „aktív” részei az adatbázisnak (pl. nem rendezhető, gyűjthető stb. tulajdonság”).

Adatbázis feladatok:

Adattárolás
Rendezés
Fókuszálás
Szűrés
Gyűjtés
Matematikai műveletek elvégzése

Indexelés:

adott tulajdonság szerinti gyorskeresés (fókuszálás) előkészítése

Kulcs:

Egyedi kulcs
Elsődleges kulcs: tételazonosító, nem változtatható
Számláló (automatikus elsődleges kulcs "sorszám")

Adatbázis-műveletek:

Keresés/olvasás (fókuszálás) adott rekordra (tételre) rákeresés
Felvétel: új tétel megadása a tulajdonságainak megadásával (űrlap)
Módosítás (adott tulajdonságok megváltoztatása, javítása, módosítása)
Törlés

Logikai feltételek:

és kapcsolat (egy egyeden 2 tulajdonság vizsgálata: igaz, ha mindkét tulajdonság igaz, egyébként hamis, logikai függvény)
vagy kapcsolat (egy egyeden 2 tulajdonság vizsgálata : igaz, ha legalább az egyik tulajdonság igaz, hamis, ha mindkét tulajdonság hamis, logikai függvény)

logikai **nem** függvény (egy egyeden az igaz tulajdonságoknál hamis, a hamis tulajdonságokra igaz)

Bármilyen válogatás, szűrés a fent említett logikai függvények alkalmas kombinációjából előállítható.

Történeti áttekintés

Sokan azt hiszik, hogy az adatbáziskezelés újkeletű fogalom, pedig tágabb értelemben véve az alapjai egészen az ókorig nyúlnak vissza. Már akkor is szükség volt bizonyos adatok (népszámlálásból származó adatok, terménymennyiségek, adók) tárolására, rendszerezésére és kiértékelésére. Ekkor még az adatok tárolására fadarabokat, agyag- majd kőtáblákat vagy éppen papiruszt használtak. Azt hiszem, nyilvánvaló ezen eszközök használatának nehézsége, megbízhatóságuk korlátai.

Előrelépést jelentett ehhez képest a papír, majd a könyvek megjelenése, hiszen kisebb helyigény mellett nagyobb mennyiségű adatot tudtak így tárolni. Azonban nem csak az adattárolás technikája fejlődött, hanem az egész világ, így a tárolandó adatok mennyisége is megsokszorozódott, amivel egy öngerjesztő folyamat indult be. Ezzel párhuzamosan nem csak az adatok mennyiségében történt hatalmas mértékű növekedés, hanem minőségi változás következett be az adattárolás rendszerében és az adatok lekérdezésének igényében is. Alapvető hátránya volt ezeknek a rendszereknek, hogy az adathozzáférés elég nehézkes volt, általában csak egyféle rendszerben lehetett adatokat lekérdezni (többnyire abc rendben történt a tárolás vagy valamilyen egyéb szempont szerint, ami meglehetősen rugalmatlanságot adott a rendszernek), valamint könnyen elveszhettek az adatok, ami a megbízhatóságot jelentősen csökkentette. Például a könyvtárakban külön katalógusokat kellett készíteni a cím szerinti, a szerző szerinti és a téma szerinti keresésekhez, holott mindhárom ugyanazokat az adatokat tartalmazta - csak más sorrendben.

A következő nagyobb állomást az adatbáziskezelésben a számítógépek megjelenése jelentette. A hőskorban még csak nagy számítógépekkel találkozhattunk, amelyekhez többnyire lyukkártyaolvasó (és -író), majd mágnesszalagos egységek csatlakoztak. Mágnesszalagokon már viszonylag nagymennyiségű adatot lehetett tárolni, azonban alapvető hátrányuk a soros hozzáférés volt, ami gyakorlatilag azt jelentette, hogy egy információ eléréséhez végig kellett olvasni az összes előző adatot a szalagon. Ez jelentős adathozzáférési időt jelentett, ami a rendszer lassúságát vonta maga után.

A diszkek megjelenésével megoldódott ez a probléma is, ugyanis itt már közvetlen hozzáféréssel tudták az információt lekérdezni, vagyis minden adat gyakorlatilag azonos időn belül elérhető volt. Kezdetben a szervezést a filekezelők végezték, amelyek az azonos struktúrájú adatokat file-okba szervezték és ezeket tárolták a diszkeken. Az adatok lekérdezéséhez a felhasználónak alkalmazásokat kellett írnia, saját magának kellett a file-okkal bajlódnia, az adatokat kinyerni belőlük.

Az operációs rendszerek megjelenésével könnyebbé vált a file-okban tárolt adatokhoz való hozzáférés, de még mindig a felhasználónak kellett megírnia a programokat, alkalmazásokat, amelyekkel hozzáférhetett a file-okban tárolt adatokhoz.

A PC-k megjelenésével és elterjedésével azonban ez a filozófia tarthatatlanná vált, hiszen a felhasználók nagy száma miatt már nem lehetett elvárni, hogy mindenki számítástechnikai szakemberré váljon csupán egy részfeladat elvégzése miatt. Ezért olyan környezetre volt szükség, amelyben nem számítástechnikai szakemberek is képesek használni az adatkezelő rendszereket.

Látható, hogy a tendencia a nagy mennyiségű adat tárolása és hatékony elérése felé mutat. Ennek tükrében jöttek létre a különféle szoftvercsomagok és kialakultak az adatbáziskezelő rendszerek. A felhasználót már nem érdekli az operációs rendszer, nem kell külön program ahhoz, hogy hozzáférjünk az adatokhoz. Ma már ezek a rendszerek platformfüggetlenek, lényegében ugyanúgy működnek UNIX-os környezetben, VMS alatt, és PC-ken. Az adatok nagy mennyisége miatt ezek a rendszerek már fel vannak készítve a hálózaton keresztüli adatelérésre is, nagy mennyiségű, tartósan tárolt adatot képesek hatékonyan és biztonságosan kezelni, akár a hálózaton keresztül is. Ezen feladatok nagy részét egy filekezelő rendszer is el tudta látni, de az nem tudott mozogni a file-ok között, és csak azokat

az adatokat tudta elérni, amelyek az alkalmazói programban szerepeltek, vagyis nem lehetett olyan adatot megkapni, amire nem készítettem fel előre a programomat.

A történet ezzel még nem ért véget, a fejlődés ma is tovább tart. A tendencia a felhasználó által még hatékonyabban és még kényelmesebben használható rendszerek felé mutat, amelyekben megvalósulhat az igazi platformfüggetlenség, az egységesség, a még nagyobb biztonság és hatékonyság.

Az adatbázis fogalma

Környezetünk leírásához meg kell azt figyelnünk, **adatokat** kell gyűjtenünk az objektumairól, összegeznünk kell azok tulajdonságait, valamint meg kell figyelnünk a köztük levő kapcsolatokat. Ezek után juthatunk a rendelkezésre álló, tárolt adatokból új ismeretekhez, **információhoz**. Az adatoknak egy jól struktúrált halmazát **adatbázisnak** nevezzük, amelyből a későbbiekben információt akarunk majd nyerni. Az információnyerés céljából pedig olyan szoftvereket használunk, amelyek lehetővé teszik az adatbázisok kezelését (lekérdezés, módosítás, törlés, karbantartás) és amelyeket együtt **adatbáziskezelő rendszernek** nevezünk. Az olyan rendszereket, amelyek a hardveren tárolt adatokból szoftverek és egyéb alkalmazói programok segítségével lehetővé teszik a felhasználó számára az információ nyerését, **információs rendszereknek** nevezzük. Látható tehát, hogy egy információs rendszerhez szükség van adatokra, hardver eszközökre, szoftverekre és a felhasználóra. Szoftveren itt olyan programcsomagot értünk, amely gyorsítja a munka elvégzését, például szövegszerkesztők, táblázatkezelők, adatbáziskezelő rendszerek. Az alkalmazói programok pedig egy-egy szakember által valamilyen programozási nyelven, egy konkrét feladatra írt programok.

Adatbáziskezelő rendszerek létrehozása

Adatbáziskezelő rendszert alapvetően kétféleképpen szoktak létrehozni. Az egyik módja, hogy egy programozó csoport valamelyik - többnyire harmadik generációs - programozási nyelvben (pl. C, Cobol) megírja a rendszert. A másik módszerben pedig egy már meglévő rendszerben fejlesztik ki az új rendszert. Ehhez többnyire negyedik generációs nyelveket használnak. Ilyen típusú rendszerek a CASE (számítógép által támogatott szoftver fejlesztő) rendszerek. A legtöbb adatbáziskezelő rendszer rendelkezik ilyen CASE eszközökkel. Továbbá vannak alkalmazásfejlesztő rendszerek is, például: QBE, POWER BUILDER, UNIFACE. A legismertebb adatbáziskezelő rendszerek manapság az ORACLE, INGRES, SYBASE, INFORMIX. Ezek többnyire nagygépes környezetben működnek. Természetesen vannak PC-re íródott adatbáziskezelő programrendszerek is, mint például: DBASE, CLIPPER, FOXPRO.

Az adatbázis létrehozását mindig az adabázistervezés előzi meg. Gondosan fel kell mérni a feladat igényeit, meg kell fogalmazni a problémákat. Egy adatbázist manapság körülbelül egy-két hónap alatt fejlesztenek ki. Ennek körülbelül 70-75%-a a tervezésre megy el, 10-15%-a a programozásra és a maradék a tesztelésre. Alapos tervezés nélkül ugyanis a rendszer átláthatatlan lesz, nagyon nehéz lesz a későbbiekben módosítani.

Az adatbázistervezés, az adatok jellegének vizsgálata és a köztük levő kapcsolatok alapos elemzése után következhet az **adatmodell** létrehozása.

Adatmodellezés

Az adatmodellezés tehát az adatállományok tervezésének korszerű módszere. Az adatmodellezéssel az a célunk, hogy egy információs rendszer adatait és az adatok között fennálló kapcsolatokat következetesen ábrázolva, elősegítsük a számítógépes információ-

feldolgozást. Az adathalmaz és az adathalmaz elemei között fennálló kapcsolatok struktúrált leírását adatmodellnek nevezzük.

*Az adatmodellek -mint ahogy a nevükből is következik-, általában nem írják le teljesen és hűen a modellezett rendszerben található adatokat és azok kapcsolatait, hanem azoknak csak egy részét. Egy **kielégítő modellnek** teljesítenie kell bizonyos feltételeket: **átfogónak** kell lennie, azaz minden lehetséges adatot és minden lehetséges kapcsolatot tudnia kell ábrázolni és kezelni; le kell tudnia írni a **valóság általános, lényeges és tartós összefüggéseit**; **redundanciamentesnek** kell lennie, azaz minden adatot lehetőleg csak egyszer tartalmazzon; **következetesnek** kell lennie; valamint **az alkalmazott hardverrel és szoftverrel összhangban levőnek** kell lennie*

Az adatmodell nemcsak felsorolása a fileban tárolt mezőknek, hanem az adatok jelentését is tartalmazza. Így az adatmodellben

- világosan megkülönböztethetők az ábrázolt dolgok és a dolgok tulajdonságai;
- pontosan meghatározott, hogy melyek az ábrázolandó dolgok olyan tulajdonságai, amelyekkel a dolgok egyértelműen azonosíthatók;
- valamint az adatmodell megadja az ábrázolt dolgok közötti kapcsolatokat.

Az adatmodell a valóság egy szeletének a leképezését jelenti. A valós világhoz képest az adatmodellek tartalmazznak bizonyos megszorításokat és egyszerűsítéseket, sőt még a modellező személyétől függő vonásokat is. Egy adatmodell úgy viszonyul a valósághoz, mint a térkép az általa ábrázolt tájhoz: a térképrajzolónak egyszerűsíteni kell, ki kell jelölnie a térkép széleit, és el kell döntenie, hogy mit is akar egyáltalán a térképen feltüntetni. Ugyanígy, az adatmodell kialakításakor a rendszerszervezőnek is el kell döntenie, hogy melyek a modelljében az ábrázolandó információelemek, és ki kell választania a dolgok között fennálló lényeges kapcsolatok közül azokat, amelyeket be akar a modellbe építeni. A lényeges információelemek és a modellben ábrázolandó kapcsolatok kiválasztása párhuzamosan történik.

Egy adatmodell a következő elemeket tartalmazza:

- egyed,
- tulajdonság,
- kapcsolat.

Vagyis:

- **Egyedek** a valóság olyan dolgai, amelyek más dolgoktól megkülönböztethetők, azaz az egyed a valós világban létező, fogalmi vagy fizikai léttel rendelkező dolog, amelyet tulajdonságokkal akarunk leírni. (pl. dolgozó, tanuló, vevő, autó, stb.) Ugyanaz a dolog többféle egyednek is tekinthető (pl. egy vevő lehet egyben dolgozó is), de a képviselt konkrét elemek halmazát egyedhalmaznak nevezzük (a vevő, mint egyedhalmaz az összes vevőt tartalmazza).
- **Tulajdonságnak** nevezzük a valóság dolgainak azon jellemzőit, amelyekkel az egyedeket leírjuk. (Pl. autónál tulajdonság a típus, szín, rendszám, stb. A típus értékei lehetnek: BMW, Opel, stb.) Az egyedek a tulajdonságértékeikkel azonosíthatók.
- **Kapcsolatnak** nevezzük az egyedek közötti viszonyok fogalmi tükörképeit. Objektumok közötti viszony pl. az, hogy a munkás a kft. dolgozója. Ennek egy konkrét előfordulása pl. Kovács János a Zabhegyező kft. dolgozója.

Még két fogalmat érdemes az érdeklődőbbek számára megemlíteni:

- **Séma (típus):** a konkrétan létező egyedek, tulajdonságok, kapcsolatok absztrakciója. Az egyedek bizonyos tulajdonságtípusaik alapján halmazokba rendeződnek. Egy-egy halmaz neve az egyed típusa, a halmazba tartozó egyedek az előfordulások, megfelelő tulajdonság előfordulásokkal. Ez a szint tartalmazza az egyed típusok tartalmi leírását, de az

adatfüggetlenség érdekében nem tartalmazza az adatok helyét és a hozzáférési módot. Ugyancsak ez a szint írja le, hogy milyen kapcsolatok vannak az egyedek (rekordok) között. A kapcsolatok leírásának módja az adatmodelltől függ.

- **Előfordulás:** konkrétan létező egyed, tulajdonságérték, kapcsolat.

Adatfüggetlenség

Mivel ugyanazon adatokat több felhasználó és több program is használhatja, az adatokat és a programokat - amennyire lehetséges - függetleníteni kell egymástól: ha például valamelyik programban a rekordokat kibővítjük egy mezővel, akkor ne kelljen a többi programot megváltoztatni. Az adatok fizikai leírását külön kell választani a programok által látott adatszerkezettől. Ez az adatfüggetlenség. Az adatfüggetlenségnek két fokozata van:

- **fizikai adatfüggetlenségről** beszélünk, ha a programok vagy felhasználók információkérései gyakorlatilag függetlenek az adatok tárolási és elérési módjától;
- **logikai adatfüggetlenségről** beszélünk, ha az adatbázis logikai szerkezetében létrehozott változások az adatbázis felhasználó programokat nem befolyásolják.

A tényleges adatfüggetlenség e kettő együttes megvalósulása.

Az adatbáziskezelő rendszer részei

- **Kapcsolattartó:** ez lehetővé teszi, hogy más alkalmazások kapcsolatba tudjanak lépni az adatbázissal.
- **Feladat átkonvertáló:** az adatbázis kezelő rendszer nyelvén kapott utasításokat elemzi, értékeli, és ha végrehajthatóak, meghatározza a végrehajtás várható leghatékonyabb módját.
- **Elérés meghatározó:** az objektumok elérésére meghatározott módszert átfordítja az operációs rendszer számára érthető parancsokká. Elindítja, majd értékeli ezen parancsok végrehajtását.

Az adatleíró nyelv (Data Definition Language)

Séma: az adatmodellnek és a leírásnak további számítógépes feldolgozásra is alkalmas tartalmi és formai megfogalmazása. Az adatleíró nyelv a séma megfogalmazására szolgáló nyelv.

Egy adatbázishoz egy séma tartozik, amely pontosan meghatározza az adatszerkezetet, a tárolási struktúrát, valamint az egyes adatelemek között fennálló logikai kapcsolatokat. Az alsémák mutatják meg, hogy az egyes alkalmazói programok az adatbázist hogyan látják. Egy adatbázis sémához akárhány alséma illeszthető. Más-más alsémákban lehet az adatoknak a típusa, neve, csoportosítása.

Az adatleíró nyelv három szinten írja le az adatokat:

- a **felhasználó szintjén**, amely a tényleges adatbázis valamilyen alsémáját, az adatmodell egy külső szintjét jelenti.
- az adatok **logikai** kapcsolatának **szintjén**, mely a teljes logikai séma, a koncepciók modell leírását tartalmazza.
- az adatok **fizikai** tárolásának meghatározása **szintjén**, mely az egyes adatok fizikai elérhetőségét, tárolási módját és a köztük fennálló fizikai kapcsolatokat (mutatók, indexek stb.) írja le. A tárolási módot és a fizikai kapcsolatokat rendszerint egy önálló nyelv segítségével valósítjuk meg.

Az adatkezelő nyelv (Data Manipulation Language)

A feldolgozás céljaira szükség van egy adatkezelő nyelvre. Ennek segítségével a felhasználó az adatleíró nyelvvel definiált adatbázison műveleteket végezhet.

A műveletek vonatkozhatnak az adatbázis adatainak:

- előállítására,
- módosítására,
- lekérdezésére.

Az adatkezelő nyelv által az adatbázis megnyitható, zárható, kulcs vagy cím alapján kereshetünk benne, beolvashatunk, beszúrhatunk, törölhetünk, módosíthatunk rekordokat.

Az adatbáziskezelő rendszer

Az adatbáziskezelő rendszer tehát egy szoftver, amely biztosítja az adatbázisban tárolt adatok létrehozását, kezelését, valamint leírja és kezeli az adatok közötti komplex kapcsolatokat. Az adatbáziskezelő rendszernek támogatnia kell valamilyen adatmodellt, hogy a valóságot le tudja képezni egy számunkra megfogható objektumra. Két részből épül fel:

- **jelölésrendszer:** hogy valamilyen formában a valóságot ábrázolni tudjuk,
- **műveletek halmaza:** amely lehetővé teszi, hogy az előbbieket kezelni tudjuk.

Az adatbáziskezelő rendszernek biztosítania kell:

- különféle igények hatékony kielégítését,
- az adatbázis következetességét, vagyis, hogy csak valós adatokat tároljunk,
- az adatbázisbeli valamennyi adat elérését, (ezt egy filekezelő rendszer nem teszi meg)
- adatfüggetlenséget, (fizikai és logikai függetlenséget)
- adatok közti komplex kapcsolatok kezelését és ábrázolását,
- redundancia mentességet és annak ellenőrzését (ne szerepeljen az adatbázisban többször is ugyanaz az adat),
- az adatok védelmét, [mind fizikai, mind emberi (véletlen vagy szándékos rongálás) értelemben]
- helyreállíthatóságot,
- több felhasználós rendszerekben az egyidejű hozzáférést (tranzakciókezelés),
- osztott adatbázisokban az adatok szétosztását, megtalálását,
- az adatforgalom optimalizálását.

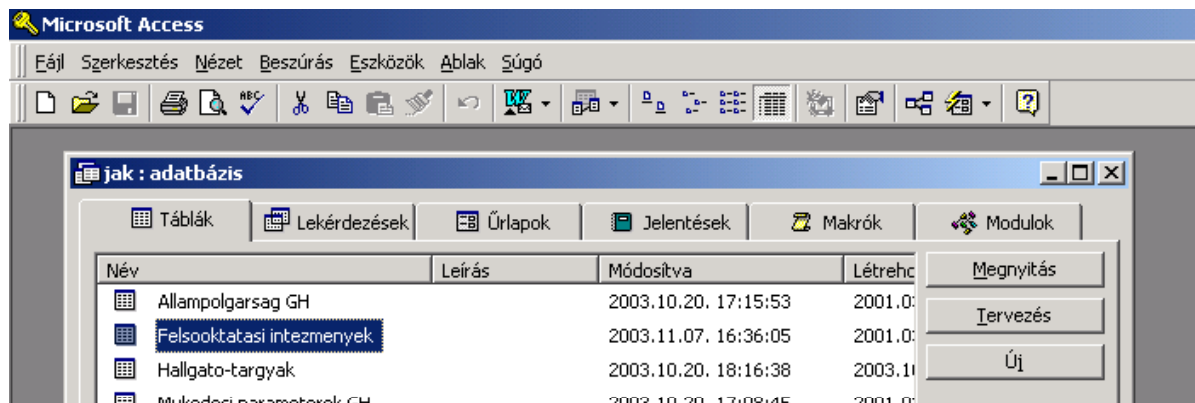
Amikor létrehozunk egy adatbázist, bizonyos szabályoknak kell eleget tenni. Így érhetjük el, hogy az adatmodell mindig a valóságot tükrözze. Ha tényleges változás következik be, akkor az adatbáziskezelő rendszer szerkezetét kell módosítani.

Egy felhasználóközel adatbáziskezelő:

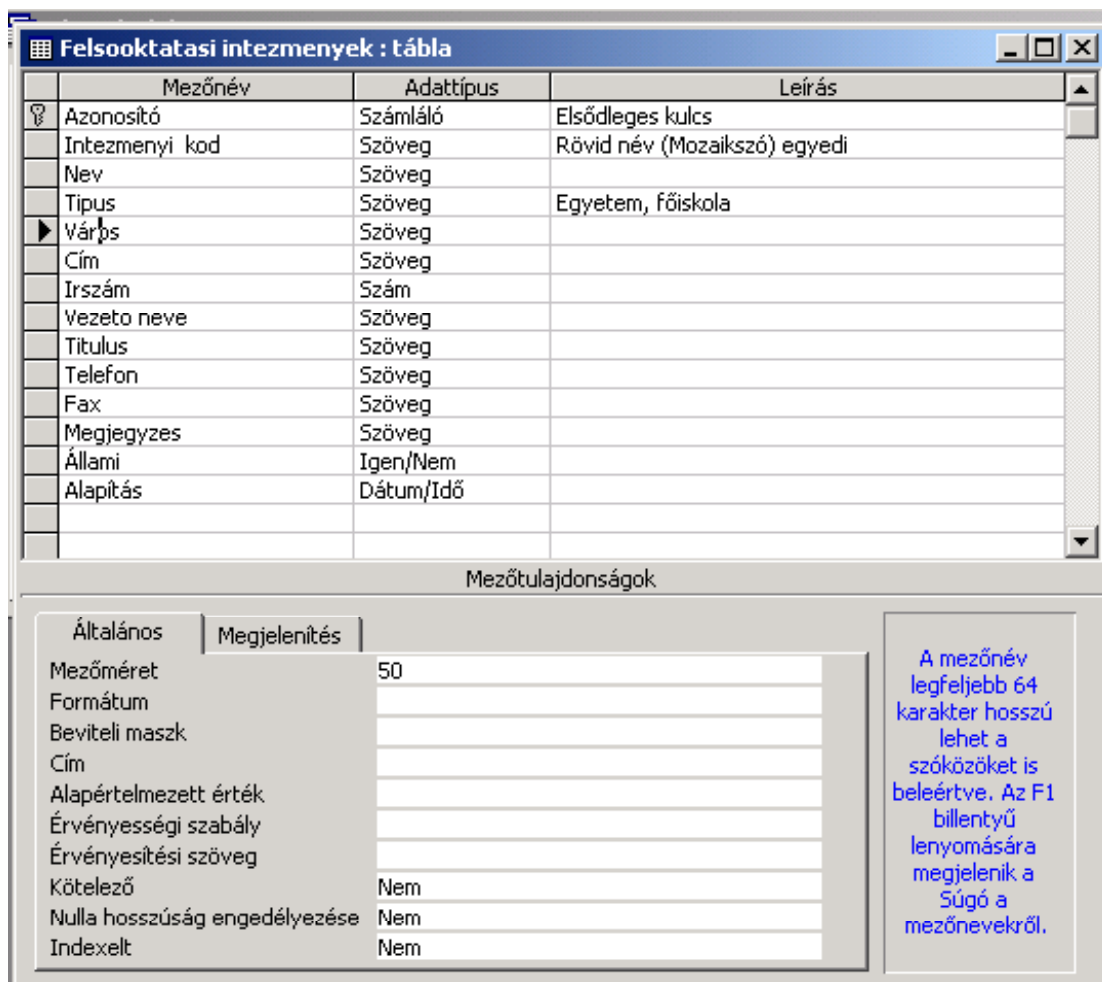
MsAccess (az MsOffice része)

Objektumok:

táblák (legtöbbször elsődleges kulcs:számláló)
lekérdezések (táblák összekapcsolása, szűrések)
űrlapok (felhasználóbarát) adatlap és űrlap nézet
jelentések (tetszetős elrendezés -> MsWordnek átadható: adatbázis-
tulajdonságok elvesztése (keresés, rendezés, szűrés)



Itt látható az MsAccess "objektumgyűjteménye". A makrók és a modulok az adatbázis programkörnyezetét tartalmazzák.



A fenti ábra egy tábla szerkezetét mutatja. Jól láthatók az általánosan használt adattípusok.

A mezőknek minden esetben le kell kötni egy hosszat (esetünkben a Város 50 karakter), amit a várható leghosszabb előfordulásra kell kalkulálni (ha pl. átlagba kalkulálunk, akkor a hosszabb városneveket nem fogjuk tudni teljes hosszban felvenni az adatbázisba).

Azokat a városneveket, melyek sokkal rövidebbek is, 50 karakter területen tárol az adatbázis: különben a keresetőség, "lapozás" elve csorbulna: ha fix hosszú rekordjaink vannak, akkor az 1543-ik rekordhoz pozícionálni még egy gyerek is könnyedén tudna: $1542 * \text{rekordméret} + 1$ karakterre. Ha viszont egyik rekord rövidebb lenne, a másik hosszabb, akkor bizony kotorászni kellene az adathalmazban.

Jól látható a táblában elhelyezkedő Számláló mező, mely az adatbázis adta automatikus elsődleges-kulcs lehetőség. Ehhez a felhasználó semmilyen direkt módon nem férhet hozzá (elolvashatja)

Vannak mezőtípusok, melyeknek nem kell hosszat választanunk: a dátum, logikai mező pl. Itt az adat természete adja a hosszat: a dátum 8 hosszú (ééééhhnn), a logikai mező pedig 1bit= igen, nem, 2 állapot.

Lehet "külső adatot" is hozzárendelni egy táblához (fénykép, feljegyzés), de ez nem igazi adatbázis tulajdonság: nem rendezhető, nem szűrhető eszerint az adathalmaz.

Felsőoktatási intézmények

Intézményi kód	ELTETOFK
Név	ELTE Tanító- és Óvóképző Főiskolai Kar
Típus	A
Cím /város	Budapest
Cím /utca	Kiss János altb. u. 40.
Cím /ir. szám	1126
Vezető neve	
Titulus	
Telefon	224-0800
Fax	202-3859
Megjegyzés	

Rekord: 30 összesen 183

Az adatkezeléshez az adatokat áttekinthető, felhasználóbarát űrlapokba rendezzük és segítjük az adatkarbantartást.

Az adatokból tetszőleges csoportosításban és rendezésben nyomtatható, sőt a Wordnek "átadható" jelentéseket készíthetünk, de ekkor megint elvész az "adatbázis-tulajdonság", a jelentés már sima szöveges (ha szépen formázott is) állomány, melyben keresni, szűrni stb. már nem egyszerű.

Végül nézzünk egy jelentésoldalt:

<i>Felsőoktatási intézmények</i>		
<i>Név</i>	<i>Város</i>	<i>Felület</i>
A. Poni Kárpási Budapesti F. Iskola	Budapest	200-8 712
Adventista Felsőokt. F. Iskola	Pécel	(20)45 2-145, 45 2-13
Általános Váltóiskola F. Iskola	Budapest	485-8 900
Ápca Vilmos Katolikus F. Iskola	Sármalk	(20)94 1-35 F., (20)94 2
Bauzany Dániel F. Iskola	Sárospatak	(34)32 5-311, (34)32 12
BGF. Kereskedelmi, Vándorlási és Idegenforgalmi F. Iskola	Budapest	974-8 200
BGF. Kereskedelmi F. Iskola Kari	Budapest	487-7 300, 487-7 302
BGF. Péterfy István Széchenyi F. Iskola Kari	Budapest	221-9 722, 922-3431
BKJC. Általános és Felsőokt. F. Iskola Kari	Budapest	925-2 122
BKJC. Gaudéus szülői dományi Kari	Budapest	213-3 076
BKJC. Károlyi szülői dományi Kari	Budapest	217-8 725
BKJC. Felsőokt. szülői dományi Kari	Budapest	217-5 457
BKF. Sándor Dorottya Gáspár F. Iskola Kari	Budapest	924-9 325, 210-1450
BKF. Károlyi Károly Vilmos F. Iskola Kari	Budapest	210-1 442, 210-1415
BKF. Károlyi Károly Gaudéus F. Iskola Kari	Budapest	210-1 441
BKF. Neumann János Informatika F. Iskola Kari	Budapest	925-4 810
BKF. Rajkó Sándor Károlyi F. Iskola Kari	Budapest	925-7 517, 250-0222
BKF. Gyula F. Iskola Kari		485-9 321
BKF. Gyula F. Iskola Kari		485-9 321
BKF. Gaudéus szülői dományi Kari		485-9 421
BKF. Gáspár F. Iskola Kari		485-9 341
BKF. Károlyi Károly F. Iskola Kari		485-9 351
BKF. Felsőokt. szülői dományi Kari		485-9 361
BKF. Gyula F. Iskola Kari		485-9 371
BKF. Vilmos F. Iskola Kari		485-9 381
Buda pesti Gaudéus F. Iskola	Budapest	925-3 491
Buda pesti Károlyi szülői dományi és Általános F. Iskola	Budapest	213-8 255, 210-1 250
Buda pesti Általános és Gaudéus szülői dományi F. Iskola	Budapest	485-1 111
Buda pesti Általános F. Iskola	Budapest	250-0 222
DE. Általános és Felsőokt. szülői dományi Kari	Debrecen	
DE. Felsőokt. szülői dományi Kari	Debrecen	

JVS november 7.

Oldal: 1/6

Sok minden tanulnivalónk van még az adatbázisokról: erre a jövő félévben kerül sor.

Tartalomjegyzék

Az adatbázis-kezelés elemei	79
Alapfogalmak	79
Történeti áttekintés	81
Az adatbázis fogalma	82
Adatbáziskezelő rendszerek létrehozása	82
Adatmodellezés	82
Adatfüggetlenség	84
Az adatbáziskezelő rendszer részei	84
Az adatleíró nyelv (Data Definition Language)	84
Az adatkezelő nyelv (Data Manipulation Language)	85
Az adatbáziskezelő rendszer	85
MsAccess (az MsOffice része)	85