



Békési Ferec

Adatbáziskezelés



A követelménymodul megnevezése:
Korszerű munkaszervezés

A követelménymodul száma: 1180-06 A tartalomelem azonosító száma és célcsoportja: SzT-007-20



ADATBÁZISKEZELÉS

ESETFELVETÉS – MUNKAHELYZET

Jelenleg egy ajtó- és ablakgyártó vállalkozás magyarországi telephelyén dolgozik. A vezetői döntést követően a most vásárolt, modern irodai programcsomag adatbáziskezelő szoftverének használatával az Ön feladatai a következők lesznek:

- a cég tevékenységének pontosabb és hatékonyabb működése érdekében a beszállítók és kereskedők adatainak, forgalmának nyilvántartásának megtervezése
- az alapanyagok és késztermékek naprakész értékesítési információinak számítógépes adatrögzítésének kivitelezése, a cég további munkatársainak bevonásával, közreműködésével

1. Feladat *Egy optimálisan használható adatbázis létrehozásához információkra lesz szüksége. Gyűjtse össze azokat a kérdéseket, amelyre adott válaszok a konkrét folyamat kidolgozásához hasznosak lehetnek!*

SZAKMAI INFORMÁCIÓTARTALOM

ADATBÁZISOK KÖRNYEZETÜNKBEN

A mindennapi életünk során szinte folyamatosan kapcsolatba kerülünk valamilyen adatbázissal. Személyi nyilvántartó kóddal, adószámmal, TAJ számmal, bankkártya számmal, stb. rendelkezünk. Az életünk részei ezek, amelyeket mind-mind egy működő adatbázis tartalmazza. Ezenkívül, vannak még további számunkra fontos feljegyzések, mint pl. kedvenc filmjeink és zenéink, ismerősök és üzleti partnerek telefonszámai. Ezek is olyan információkat, adatokat tartalmaznak, amelyek átgondolt, rendszerezett tárolás nélkül később elérhetetlenné válhatnak számunkra.

Az adathalmaz akkor alakul át adatbázissá, ha az valamilyen meghatározott szempontok szerint felépített és lehetővé teszi az adatok szisztematikus kezelését, elérhetőségét. Tehát az adatból, a feldolgozás során válik információ.

1. AZ ADAT FOGALMA

A számítástechnikában **adat**nak definiáljuk a számokkal leírható, a számítástechnika eszközeivel rögzíthető, feldolgozható és megjeleníthető objektumokat. Az adat egy általunk kiválasztott objektum (tetszőleges dolog, amire az adat vonatkozik), egy meghatározott változójának (tulajdonságának, jellemzőjének, karakterének) az értéke (karakterállapota, megvalósult formája).

Az adat tehát akkor tekinthető definiáltnak, ha meghatározzuk, hogy milyen objektum, melyik változója, milyen értéket vesz fel. Az adat fogalma jól elkülöníthető két másik, sokszor azonos értelemben használt fogalomtól, az ismerettől és az információtól: Az **ismeret** valamilyen objektummal kapcsolatos tapasztalataink, általánosításaink és fogalmaink összessége. Az **információ** olyan feldolgozott adat vagy ismeret, amely környezetünket és viselkedésünket befolyásolni képes.

objektum	tulajdonság
ország	valuta árfolyam
áru	mennyiség
autó	évjárat
könyv	író
ember	életkor

1. ábra. Az adat definiálhatósága

2. AZ ADABÁZIS ALAPVETŐ FOGALMAI

Az adatbázis tervezésének megkezdése előtt fontos alapfogalmakat kell megismernünk – adatbázis, kulcs, modell, tábla – melyek a tervezés módszertanának egyszerűbb megértését és alkalmazását teszik lehetővé.

Adatbázis: Adatok gyűjteménye, amelyeket egy adatbázis-kezelő rendszer (DBMS – **DataBase Management System**) kezel. Több felhasználó által is használható, többféle felhasználásra alkalmas, tárolt adatok és a köztük lévő kapcsolatok szisztematikus rendszere.

Kulcs: Egyértelműen azonosítja az egyedtípus bármely előfordulását, vagyis egy konkrét egyedet egyértelműen különböztet meg a többi egyedtől. Egyed az, amire az adat vonatkozik és amelynek a tulajdonságait nyilvántartjuk.

Tábla: Az adatokat táblákban (táblázatokban) tároljuk. A tábla oszlopait mezőnek, a tábla sorait rekordnak nevezzük. A különböző jellegű adathalmazokat külön táblában helyezünk el, pl.: termékek, szállítók, járművek, árak.

	MEZŐ			
	Azonosító	Név	Születési hely	Születési idő
	0001	Tóth Imre	Kiskunhalas	1941. szept. 16.
REKORD	0002	Kovács Katalin	Makó	1967. júl. 22.
	0003	Szép Ibolya	Kaposvár	1982. jan. 11.
	0004	Horváth Zoltán	Szolnok	1976. márc. 15.

2. ábra. Egy tábla részei

Mező: A táblázat oszlopai egy-egy az egyedre jellemző tulajdonságot tartalmaznak, ezeket az oszlopokat mezőknek nevezzük és mezőnével azonosítjuk.

Rekord: Egy táblázatba soronként egy-egy egyedet viszünk fel, amelyet rekordnak nevezünk és azonosítóval látjuk el.

3. AZ ADATBÁZIS SZERKEZETEK, ADATMODELLEK ALAPVETŐ ISMERETEI

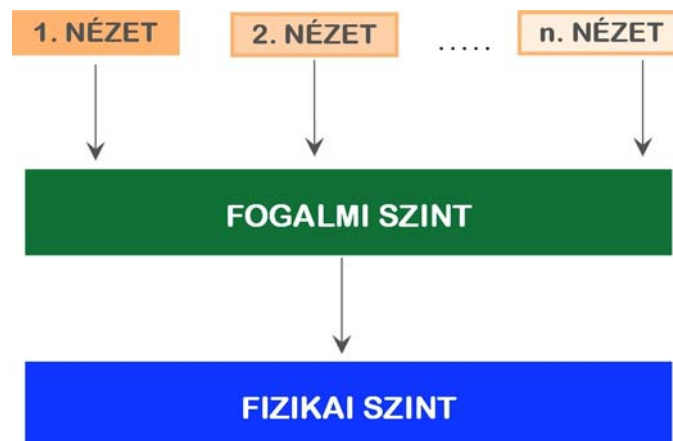
3.1. Adatmodell

Egy adatbázis kezelő rendszer mindig egy adatmodellre épül, amely tulajdonképpen a valós folyamatok absztrakt leképezése, egyedek, tulajdonságok és kapcsolatok halmaza. Egy eszközrendszer, amellyel leírható az adatok és a kapcsolatai, a tervezés és az elemzés szempontjából lényeges típusjegyekkel, illetve a közöttük lévő kapcsolatokkal (egyedtípus, tulajdonságtípus, kapcsolattípus).

Alacsony szintű adatmodell: hálós, hierarchikus, relációs.

Magas szintű adatmodell: E/R modell: Entity–Relationship, azaz Egyed–Kapcsolat modell.

- **Külső (logikai) vagy nézet szint:** Leírja az adatbázis megjelenését úgy, ahogy azt a felhasználók láthatják és ahogyan az adatok különbözően megjelennek az adathozzáférési jogoktól függően. Az egyes felhasználókat más–más adat és annak (logikai) összefüggései érdeklik. Talán természetes is, hogy határozottan eltérő irányokból szemlélnék, értékelnek egy ugyanazon jelenséget és a különböző ismereteik szerint dolgozzák fel azt.



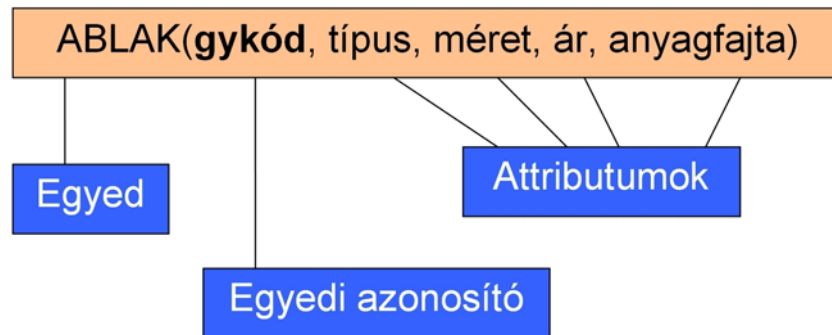
3. ábra. Az adatmodell három szintje

- **Koncepcionális (fogalmi) szint:** Ez a szint írja le a mindenki számára elérhető adatok megjelenését, formáját, az adatbázis összes objektumának szerkezetét és a közöttük lévő kapcsolatokat, valamint az objektumokhoz történő hozzáféréseket. Kettős funkciót lát el: 1. Adatbázissá szervezi az egyéni, külső szemléleteket. 2. Összekapcsolja a logikai és fizikai szintet. Ez alkotja a logikai vázat, modellt.
- **Belső (fizikai) szint:** A fizikai tárolás és elérés módjának leírása. Az adatok tényleges elhelyezkedése, tárolása, kapcsolódásai nem tartoznak a felhasználóra, ezért a felhasználó nem kezel indexeket, mutatókat, láncokat, hozzáférést. Amennyiben a körülmények azt teszik szükségessé, úgy az adatokat többszörösen is tárolni kell. Esetenként különböző gépeken és eltérő adatkezelőkkel feldolgozottan is. A fizikai kezelésmódot a rendszer biztosítja.

A külső és a koncepcionális szint alkotja a koncepcionális/logikai adatmodellt, a belső szint a fizikai adatmodellt.

3.2. Adatmodellek elemei

- **Egyed (entitás):** A valós világnak az az eleme (tárgy, jelenség, elképzelés, személy, fogalom stb.), amely a modellezés tárgyát képezi.



4. ábra. Az egyed és jellemzői

- **Tulajdonság (attributum):** Az egyednek a modellezés szempontjából lényeges jellemzőinek halmaza.

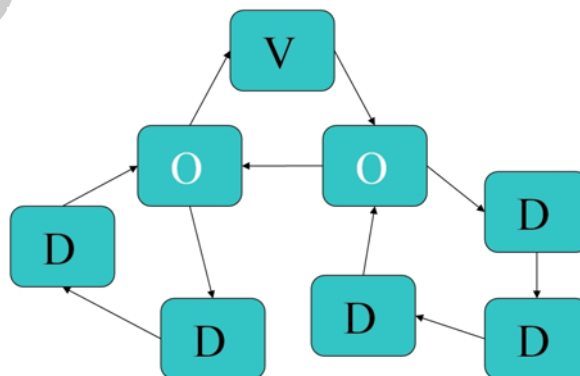
Egy adatbázis szerkezetének, struktúrájának leírásához különféle újabb és újabb adatmodellek alakultak ki.

3.3. Hierarchikus adatmodell

Az adatokat egy fastruktúra-szerű hierarchikus szerkezetben tárolja, mely egy fához hasonlítható leginkább. A fa mindegyik csomópontja egy rekordtípusnak felel meg. Előnye, hogy a hierarchikus szerkezet egyszerűen leírható, könnyen elkészíthető, mágnesszalagon is tárolható. Jobbára nagygépes környezetben fordult elő. Ma már ez az adatbázis szerkezet elavult, így napjainkban csak történeti jelentősége van.

3.4. Hálós adatmodell (1960 –1980)

A hierarchikus modell továbbfejlesztése. Ebben a modellben az adatok között tetszőleges kapcsolatrendszer alakítható ki. Egy adatnak több őse is lehet, egy "szülő" és több "gyermek" rekord pl., lehet egy vállalati osztály és a dolgozói. Hátránya: bonyolult kapcsolatok, nagy tárolóigény. Ez is nagygépes környezetben fordult elő, ma már szintén csak történeti jelentőséggel bír.



5. ábra. Hálós modell (Vállalat, Osztály, Dolgozó)

3.5. Relációs modell (1970 –)

A különböző jellegű, de mégis kapcsolatba hozható adathalmazokat önálló állományokban (táblákban) tároljuk, amely állományok (táblák) között egy azonos adatot tartalmazó mező tartja a kapcsolatot. A relációs modellre épülő adatbáziskezelőket RDBMS-nek (Relational DBMS) nevezzük. Szabványos leíró/lekérdező nyelvük az SQL.

3.5.1. Reláció: A relációt táblázatban szokás ábrázolni. A reláció egy eleme a táblázat egy sora.

- a reláció olyan táblázatnak tekinthető, melynek nem lehetnek azonos sorai (rekordjai)
- a sorok sorrendje tetszőleges
- a sorok és oszlopok találkozásában mindig csak egy érték van
- az oszlopok a reláció attribútumai

3.5.2. Funkcionális függőség

A P attribútum halmaz funkcionálisan meghatározza a Q attribútum halmazt, $P \rightarrow Q$, ha P értékei és Q értékei között P irányából egyértelmű hozzárendelés van, azaz egy reláción belül igaz, hogy ha P értékei megegyeznek a reláció két sorában, akkor Q értékei is megegyeznek ebben a két sorban. (P minden értékéhez Q-nak pontosan egy értéke tartozik.)

Funkcionális függőségek egy relációban

A kulcs funkcionálisan meghatározza a reláció többi attribútumát. Egy relációban nem csak a kulcs határozhat meg funkcionálisan egy attribútumot, ezek a nem kulcsjellegű funkcionális függőségek. A nem kulcsjellegű funkcionális függőségek egy relációban redundanciát okoznak, állomány-karbantartáskor pedig következetlenséget, anomáliát okozhatnak.

3.5.3. Normálformák

- A redundancia kiküszöbölésére vezették be a normálformákat (NF).
- Meghatározásukhoz az attribútumok csoportosítására van szükség.
- Elsődleges attribútumoknak nevezzük egy relációban azokat az attribútumokat, amelyek a reláció valamelyik kulcsának részei, a többi attribútumot a relációban másodlagos attribútumnak nevezzük.

Első normálforma, 1NF

- 1NF-ban van az adatállomány, ha minden tulajdonsága elemi, azaz minden előfordulásra csak egy értéket vesz fel
- Az 1NF-ban lévő adatállomány minden tulajdonsága (attribútuma) funkcionálisan függ a kulcstól.
- A reláció 1NF adatállomány.

Második normálforma, 2NF

- 2NF-ban van az adatállomány, ha 1NF-ban van, és nem tartalmaz részleges függést, azaz nincs olyan másodlagos attribútuma amely funkcionálisan függne a kulcs valódi részétől.
- Részleges függésről, csak összetett kulcs esetén beszélhetünk, ezért az 1NF-ben lévő és egyszerű kulccsal rendelkező adatállomány egyúttal 2NF-ben is van.

Harmadik normálforma, 3NF

- 3NF-ban van az adatállomány, ha 2NF-ban van, és nem tartalmaz tranzitív függést, azaz nincs olyan másodlagos attribútuma amely funkcionálisan függne egy másik másodlagos attribútumtól.
- Logikai adatmodellezésnél legalább a 3NF teljesedését megköveteljük.

Normálformák kialakítása, normalizálás

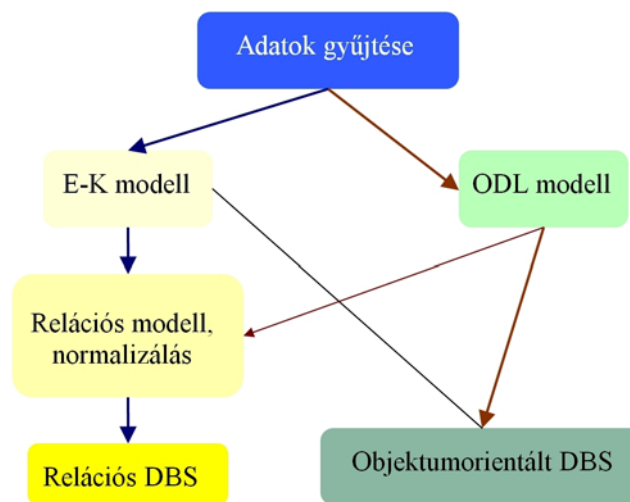
- Az optimális szerkezet kialakítása a normalizálatlan állomány szétbontásával történik, a nem kulcsjellegű funkcionális függőségek kiküszöbölésével.
- A műveletet dekomponálásnak nevezzük, ez olyan részekre bontást jelent, hogy a részekből az eredeti visszaállítható.
- A részek összekapcsolása az idegen (külső) kulcs segítségével történik

Idegen kulcsok

- A normalizált adatállomány részei között 1:N kapcsolat alakul ki, amelyet a funkcionális függőség határoz meg, ezért a modell egyed kapcsolati diagramon is ábrázolható.
- A kapcsolatokat azok az attribútumok biztosítják, amelyek a kapcsolódó adatállományok mindegyikében megtalálhatóak, és az egyikben kulcsként szerepelnek, a másikban nem (másodlagos attribútumok, vagy a kulcs valódi részei). Ezek az attribútumok az idegen kulcsok.

3.6. Objektumorientált modell

Az objektumorientált programozási nyelvek a C++ nyelvhez hasonló szabvány, eszközrendszer az adatstruktúrák leírására. Leíró nyelve az ODL (objektum definíciós nyelv), lekérdező nyelve az OQL (objektum lekérdező nyelv). Az objektumorientált modellre épülő adatbázis-kezelő rendszereket OODBMS-nek nevezzük (Object Oriented DBMS). Ezek fejlesztő nyelve általában C++ vagy Smalltalk.



6. ábra. Az adatbázismodellezés folyamata

3.7. Egyed–kapcsolat modell

Egy adatbázis létrehozása során első lépésként rendszerint az E/R modell segítségével létrehozunk a problémakör adatstruktúrájának megfelelő leírást. Több úton haladva is létrehozhatjuk az E/R leírást, de célszerű az általánosan bevált módszert alkalmazni. Előbb a fontosabb, lényegesebb elemeket határozzuk meg, majd ezekre alapozva később finomítjuk, kibővítjük a modellt az egyéb, kevésbé fontos elemekkel.

Az E/R modell esetén a központi szerepet az egyedek játsszák, mivel körük csoportosulva léteznek a tulajdonságok és a kapcsolatok is. Az egyedek nélkül sem tulajdonság, sem kapcsolat nem létezik, ezért a tervezés során célszerű elsőként a problématerületen megjelenő egyedeket számba venni, nevet és jelentést adva nekik.

Az egyedek felrajzolása után sorra vehetjük az egyedek között fennálló kapcsolatokat, kijelölve a kapcsolatok jellegét is. Az egyedek és a kapcsolatok együtt alkotják az E/R modell gerincét, vázát. Alapvetően e váz határozza meg a későbbiekben létrehozandó DBMS adatmodell struktúráját is.

Az egyedek és kapcsolatok megadása után sorba vehetjük, hogy milyen információkra van szükség az egyes egyedekre és kapcsolatokra vonatkozóan. A tulajdonságok kijelölésénél ügyeljünk arra, hogy milyen értékeket vehetnek fel, megkülönböztetve a többértékű és összetett és származtatott tulajdonságokat.

A tervezés előbbiekben vázolt lépései:

- egyedek meghatározása
- kapcsolatok meghatározása
- tulajdonságok meghatározása

Nem mindig szükséges ezt szigorú sorrend szerint végre hajtani, hiszen a bonyolultabb rendszerek esetén is sokszor csak a későbbiek folyamán derülnek ki olyan szükséges módosítások, amelyek korábbi lépések eredményeire vannak hatással. Ezért az E/R modell tervezése során is, hasonlóan a szoftver termékek általános tervezési metodikájához, ciklikus, ismétlődő tevékenységek sorozatát szükséges megtennünk.

Egy magas szintű adatmodellből lehetséges egy alacsony szintű adatmodellre való áttérés, amelyet az ún. leképezési szabályok tesznek automatikussá. Így, a magas szintű adatmodellt (pl. E/R modell) **a leképezési szabályokkal**, egy másik, adatbázis-kezelő függő, alacsonyabb szintű (pl. relációs) modellre alakíthatunk át. Ez a leképezés az ún. CASE eszközök segítségével automatizálható.



7. ábra. Leképezés CASE eszközzel

Egy ilyen leképezés után kapott pl. relációs modell általában harmadik normálformában van, de esetleg a normalizálás további alkalmazásával finomíthatunk a kapott adatmodellünkön.

3.8. Számítógéppel támogatott rendszerfejlesztés (CASE - Computer Aided Software Engineering)

CASE eszközök: olyan programcsomag, amelynek moduljai egy rendszer fejlesztésének teljes folyamatát támogatják a problémadefiniálástól a kivitelezésen át a bevezetésig, az üzembe helyezésig és elkészítik a rendszer-dokumentációt. Az adatbázis megtervezéséhez nagyszámú feltételt és körülményt kell figyelembe venni. Az elemzés eredményét egy olyan adatmodellben kell tárolni, mely közvetlenül felhasználható az adatbázis létrehozásához. A nagy adatmennyiség mellett, egy ebből fakadó másik problémát jelent, hogy a tervezés rendszerint nem egy ember munkája, hanem csapatmunka. Emiatt különös gondot kell fordítani a dokumentációk szabványosságára, a feladatok kiosztására. A CASE eszközök jelentős segítséget nyújtanak a tervezésben, a csoportmunka összehangolásában. A CASE komponens eredménye legtöbb esetben közvetlenül felhasználható az űrlapok, jelentések és menük generálásához.

A CASE eszközök lehetőséget adnak egy rendszerfejlesztési (pl.: adatbázis) tevékenység elemzési és tervezési fázisaiban végzett munka hatékonyságának növelésére:

- a rendszer komponenseinek és működésének feltárásával,
- a fejlesztési információk rögzítésével és elemzésével,
- a fejlesztési projekt irányításával,
- valamint a fejlesztéshez szükséges információk és tevékenységek adminisztrálásával.

A 80-as évekre tehető a kialakulásuk, amikor az alkalmazások bonyolultsága, az ember-gép kommunikáció tervezésének előtérbe kerülése miatt a tervezői dokumentációk elkészítése, vagy egy projektben résztvevők munkájának összehangolása nagy adminisztrációs feladatot jelentett. Ekkor magától értődően, ezekre a feladatokra is számítógépes támogatást kezdtek használni: pl. folyamatábrák készítése grafikus editorral, képernyőtervek készítése fejlett adatbázis-kezelővel.

Ezeket a tevékenységeket azután egyetlen programrendszerbe ötvözték és ettől kezdve a részek közötti összefüggések is leírhatóvá és ellenőrizhetővé váltak. Az ilyen programcsomag szerkezeti struktúrájában is megtaláljuk a részfeladatokat, amelyek pl.: az ábraszerkesztő vagy az adatszótár-kezelő modul.

Az ábraszerkesztő modul önállóan is használható és a programcsomag által támogatott módszertan előírásainak megfelelő ábrák megszerkesztésére alkalmas. Általában készül hierarchikus rendszerábra, egyed-kapcsolati adatmodell, struktúra ábrák a programszerkezetekhez. Az adatszótár a rendszerfejlesztési munka adatait és azok összefüggéseit tartalmazza.

Az adatszótár táblái, az ábrák, ábraelemek, felhasználói követelmények adatait tartalmazza. Az ábraszerkesztő outputjait, vagyis a kész ábrák adatait – neveket, kódokat, kapcsolatokat – automatikusan tárolja és lehetőséget biztosít az ábrák és ábraelemek leírásainak kiegészítésére. Például egy E/K diagramm egyed típusaihoz attribútumok megadása, egyed-előfordulások számának becslése.

Elemzési feladatokhoz az adatszótár-kezelővel keresztreferencia táblázatokat készíthetünk: egyed/attribútum, attribútum/output, stb. Az adatszótár kezelő támogatja a relációs adatelemzést, ellenőrzi az ábrák kapcsolatait és teljességét. Az adatszótár adattartalmából elkészíthető a rendszerdokumentáció, automatikusan elkészíthető az adatbázis definíció egy választott adatbáziskezelő nyelvhez.

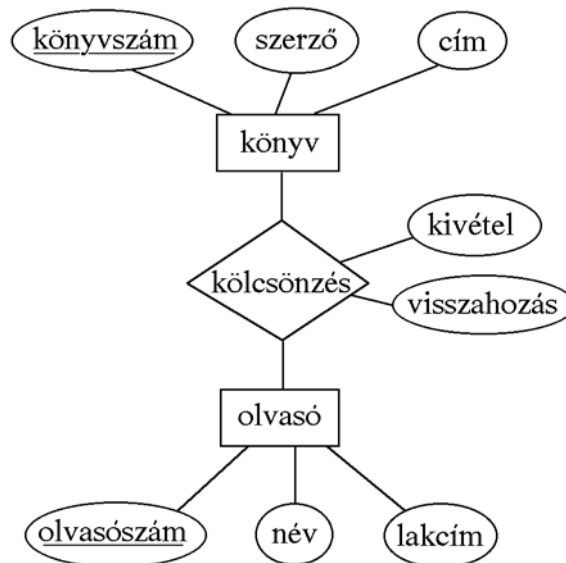
3.9. Entity-Relationship (E/R), azaz Egyed-Kapcsolat (E/K) diagram

Az egyedekkel, tulajdonságokkal és kapcsolatokkal valós világ jelenségeit leíró modellt egyed-kapcsolat modellnek, az ezt ábrázoló diagramot egyed-kapcsolat diagramnak nevezik. Rövidítve így is ismerhetjük: E/K modell és E/K diagram. Az angol elnevezése alapján – entity-relationship modell – az E/R modell és E/R diagram elnevezések is elterjedtek. A 8. ábra egy könyvtári rendszer E/R diagramja.

Az egyed-kapcsolat diagramoknak szabványos jelölésrendszerük van:

- téglalappal az egyedeket,
- ellipszissel az attribútumokat,
- rombuszal a kapcsolatokat,
- dupla téglalappal a gyenge egyedhalmazokat,
- dupla rombuszal a sok-egy kapcsolatot,
- aláhúzással az elsődleges (szuper)kulcshoz tartozó tulajdonságokat jelölik.

A tervezés kezdeti szakaszában vagy bonyolultabb E/R diagramok esetén az attribútumok ábrázolását el szokták hagyni.



8. ábra. E/R diagram¹

Az E/R diagramok a korszerű adatbázis-kezelő rendszerek kialakulásával, a modellezési igények megjelenésével jöttek létre. Feladatuk az, hogy áttekinthető grafikus eszközökkel megkönnyítsék az adatbázisok logikai tervezését.

Az E/R diagramok a hálós adatbázis koncepciónak köszönhetően létrejöttek és az adatbázis létrehozásának vizuális segítésére szolgáltak. A programozó a diagramok szemlélésével könnyebben tudta a kérdéses adatbázis-kezelő szoftver DDL-jében (Data Definition Language – adatdefiníciós nyelv) programozni a gazda és tag rekordokat. Az idő múlásával azonban a relációs adatbázis kezelők fokozatosan kiszorították a hálózati modellt és az E/R diagramokat relációk létrehozására kezdték használni.

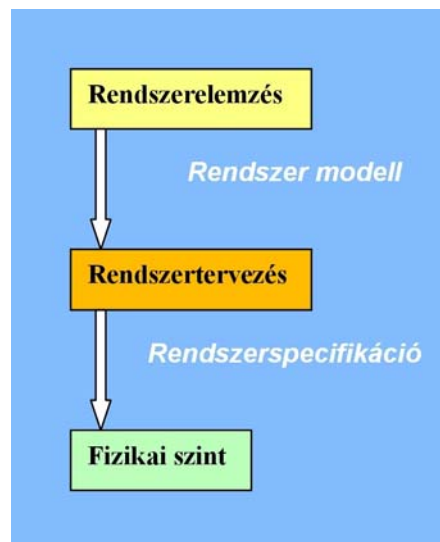
Ez elvileg lehetséges, hiszen a relációs adatbázis-kezelők kezdetben a programozást megkönnyítő interfészek voltak a hálózati modellű adatbázis-kezelő rendszereken. Ez a megoldás azonban nem volt tökéletes. Részben azért, mert az E/R nem támogatta közvetlenül a relációs adatmodellt, részben pedig azért mivel a finomabb részletekben (elnevezések, megírási szabályok, részdiagramok összekapcsolása, stb.) számtalan szokás alakult ki.

Ezek egyrészt megnehezítették a diagramok portabilitását, másrészt, és ez a fontosabb, megnehezítették az E/R módszer alkalmazását a logikai adatbázis séma automatizálásra.²

¹ Forrás: dr. Katona Endre: Adatbázisok

4. AZ ADABÁZIS TERVEZÉS LÉPÉSEI

Egy adatbázis tervezése során először az adatbázisunkat meghatározó rendszer felépítését, működését kell megismernünk és elemzésekkel meghatároznunk a feldolgozandó adatok körét, valamint az adatbázissal szemben támasztott igényeket.



9. ábra. Az adatbázis tervezés folyamata

Ezt követően megvizsgáljuk a logikailag összetartozó adatok közötti kapcsolatokat. Így hozzuk létre a tervezett rendszer specifikációját vagy logikai modelljét. Ezután a felhasználható szoftver és hardver körülmények figyelembe vételével fizikai szinten is létrehozunk az adatbázisunkat.

Az adatbázis tervezés folyamatának fő lépései:

- Egyed-kapcsolat modell felírása.
- Relációs adatbázis séma felírása.
- Relációsémák normalizálása.

Kapcsolat foka az egyedek között fennálló viszonyt hordozza.

A kapcsolat fokát a két egyed előfordulásai közötti összefüggés számossága határozza meg, ami alapvetően háromféle lehet:

- **Egy az egyhez** kapcsolatban az összefüggések az egyedek előfordulásai között mindkét irányból egyértelműek (1:1).
- **Egy a többhöz** kapcsolatban az összefüggések csak az egyik irányból egyértelműek (1:N).

² Forrás: Sárközy Ferenc: Térinformatika

- **Több a többhöz** kapcsolatban az összefüggések egyik irányból sem egyértelműek (N:M).

Áttekinthető szerkezet létrehozása.

A kapcsolati fokok között kitüntetett szerepe van az 1:N kapcsolatnak, amit elemi hierarchiának is nevezünk. Elsődleges szempont az adatmodell olyan átalakítása, hogy az csak 1:N kapcsolatot tartalmazzon. Ebből a célból az 1:1 kapcsolatokat összevonjuk, az N:M kapcsolatokat feloldjuk egy olyan kapcsoló egyedtípus közbeiktatásával, amely már 1:N fokú kapcsolatot jelent a vizsgált egyedtípusok között.

5. RELÁCIÓS ADATBÁZISKEZELŐ – MS ACCESS

A Microsoft Access egy a Windows operációs rendszerhez fejlesztett relációs adatbáziskezelőrendszer, amellyel az adatok:

- egyszerűen tárolhatóak,
- kereshetőek és megjeleníthetőek,
- rendezhetőek és lekérdezhetőek.

A relációs adatbáziskezelők szabványos programnyelve az SQL. Ezt használja a Microsoft Access is. Az MS Access használatával egyetlen adatbázisfájlba menthetjük el a számítógépen rögzített adatainkat. Az ilyen adatbázisfájl nemcsak az adatokat tartalmazó táblákat, hanem az ezek megjelenítéséhez, rendezéséhez és lekérdezéséhez szükséges objektumokat is tartalmazza.

Az ilyen, modern adatbáziskezelő rendszer leggyakoribb objektumai:

- a táblák,
- a lekérdezések,
- az űrlapok és a jelentések.

Amikor megnyitunk egy adatbázisfájlt, nem az adatok jelennek meg rögtön, hanem az a munkaterület, ahol ezek közül kiválaszthatjuk a munkánkhoz éppen szükséges objektumot és funkciót. Ezen a munkaterületen lehetőségünk van új adatbázisobjektumok létrehozására, valamint a már korábban eltárolt objektumok megnyitását követően, bővíthetjük, rendezhetjük az adatbázisba felvett tartalmakat, adatokat.



10. ábra. Az MS ACCESS munkaterület részei

6. AZ MS ACCESS PROGRAM HASZNÁLATA

Start menü / Programok/ Microsoft Office/ Microsoft Access

6.1. Új adatbázis létrehozása

Fájl menü / Új parancs vagy Eszközsor / Új parancs

- 6.1.1. Megjelenik a munkaterület ablak
- 6.1.2. Üres adatbázis választása
- 6.1.3. A létrehozandó adatbázis nevének és helyének megadása az Új adatbázis párbeszédpanelen, majd ezután a Létrehozás gomb lenyomása
- 6.1.4. Az Adatbázis ablak megnyílik
- 6.1.5. Az új adatbázist mentenünk kell, az adatokkal való feltöltés előtt

6.2. A korábban elkészített, lementett és tárolt adatbázis megnyitása

Megnyitás parancs vagy Eszközsor / Megnyitás parancs

Az adatokat ADATTÁBLÁban rögzítjük és tároljuk, ez látható a XXX ábrán.

6.3. Adattábla létrehozása

- 6.3.1. Az Objektumok listán a Táblák ikonra kattintunk
- 6.3.2. Tábla létrehozása Tervező nézetben sorra duplát kattintunk
- 6.3.3. Az új tábla minden mezőjének kitöltése



11. ábra. Adattábla létrehozása

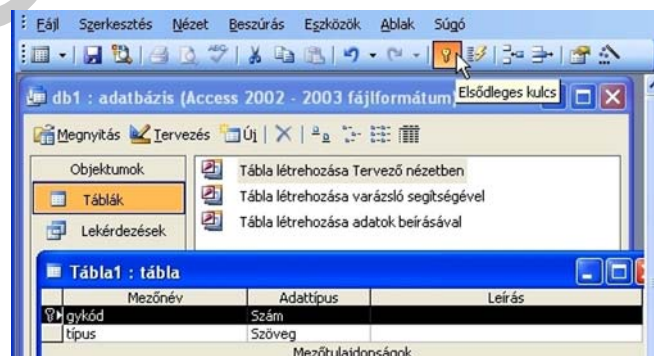
6.4. Mezők kitöltése

- 6.4.1. A Mezőnév oszlopra kattintunk és beírjuk a mezőnevet. (pl.: terméknev)
- 6.4.2. Az Adattípus oszlopban megtartjuk az alapértelmezett (Szöveg) adattípust vagy kattintunk az Adattípus oszlopba, a nyílra és kiválasszuk ki a megfelelő adattípust (Szöveg, Szám, Logikai, stb.)
- 6.4.3. Nem kötelező, de adhatunk ismertető leírást is az utolsó (Leírás) oszlopban. Amikor a mezőbe adatokat viszünk be, ez a leírás az állapotsorban fog megjelenni.

6.5. Az Elsődleges kulcs megadása

Elsődleges kulcs: Minden táblának rendelkeznie kell minimum egy olyan mezővel, amely egyedileg azonosítja a táblában tárolt rekordokat. Ezt a mezőt a tábla elsődleges kulcsának nevezzük, amelyik nem lehet üres, azaz nem lehet Null értéke. Az elsődleges kulcs egyedi, nem ismétlődhet az adatbázisban.

Azt a rekordot kijelöljük, amely az aktuális táblában biztosan nem lehet azonos és az eszközsoron az Elsődleges kulcs ikonra kattintunk. Ekkor a rekordunk mellett egy kis kulcs jelenik meg.



12. ábra. Az elsődleges kulcs megadása

6.6. A tábla elnevezése, mentése

Amikor megadtuk az elsődleges kulcsot és elkészült minden mező, a táblánkat mentjük el. Ekkor, itt adjuk meg a tábla nevét:



13. ábra. Tábla mentése és elnevezése

6.7. A tábla feltöltése

A fentiek szerint létrehozott táblákat az Adatlap nézetben tölthetjük fel adatokkal.



14. ábra. Az adatok feltöltése

7. MŰVELETEK TÁBLÁKKAL ÉS ADATOKKAL

A táblákkal végzett műveletek legtöbbször – pl. az oszlopok szélesség beállítása vagy az oszlopok és sorok cseréje, a rekordok rendezése – az excel táblázatkezelőben alkalmazott módon történhet.

7.1. Mozgás az adattáblán belül

A léptetőgombokkal a legegyszerűbb a feladathoz szükséges rekordra lépni, amelyek az ablak alsó szélén találhatóak. Az adattáblán a kurzormozgató nyilakat és a görgetősávot is használhatjuk.

gykód	típus	méret	ár
101	AJTÓ	0	
1200	ABLAK	0	
0		0	

15. ábra. Rekord kiválasztása

7.2. Új rekord felvételének lehetősége többféle módon

- Új rekord beírása gomb
- Beszúrás menü / Új rekord vagy CTRL+ +
- Adattábla alsó sora

7.3. Rekordok törlése /Nem lehet visszavonni a törlést!/ 7.3.1. A rekord kijelölésekor az egeret a rekordkijelölő oszlopon húzzuk, mint például az excelben is tesszük.

7.3.2. A kijelölt rekordot a Delete gombbal vagy a Szerkesztés menü / Rekord törlése parancsával töröljük.

7.4. Adatmódosítás

Az adatok módosítása egyszerűen az adatok átírásával történik. Amikor módosítjuk az adatokat, azok azonnal bekerülnek az adatbázisfájlba.

7.5. Az adatok rendezése

A rendezés gyakori feladat az adatokkal való tevékenységeink során. Ezzel a funkcióval az adattáblában, az általunk megadott feltételeknek megfelelő rekordokat kereshetünk meg egyszerűen.

Ennek használatával pl. az ár mező csökkenő rendezésével rögtön megkapjuk a legdrágább terméket.

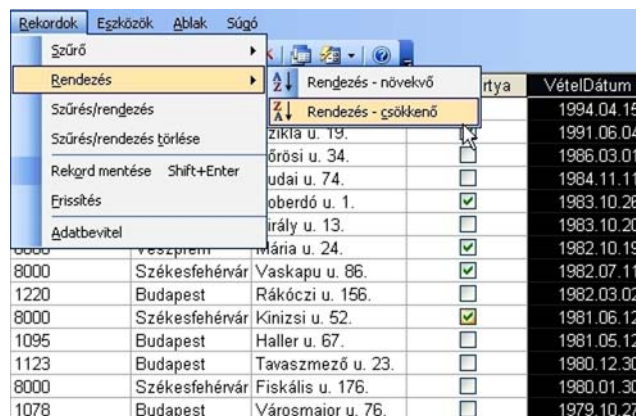
típus	méret	ár
AJTÓ	210x170	41800
ERKÉLY AJTÓ	190x170	39250
ABLAK	145x114	29550
ABLAK	120x150	19550
ABLAK	120x120	16500
		0

16. ábra. Csökkenő rendezés

7.5.1. Sorba rendezés

Minden mező esetén megadhatjuk, hogy az adott tulajdonság szerint növekvően vagy csökkenően legyenek rendezve a rekordok.

- Rekordok menü / Rendezés
- Eszközsor / Rendezés ikon



17. ábra. Rendezés

Amennyiben az ajtókat szeretnénk leválogatni, akkor elég rendezni a termék táblát a típus mező szerint. Az ilyen rendezést minden táblázatkezelő támogatja és általában több rendezési szempontot is megadhatunk. Lehet például az első rendezési szempontunk az anyag, a második az ár. Ezzel a táblázatkezelőnk az azonos anyagú termékeket az áruk szerint is rendezni fogja.

7.6. Az adatok szűrése

Az adatok szűrése azt jelenti, hogy megadunk egy szűrő (logikai) feltételt és az adatbáziskezelőnk csak azokat a rekordokat jeleníti meg, amelyek ennek a feltételnek eleget tesznek. A szűrés legegyszerűbb módja, a szűrés kijelöléssel. Ekkor az egyes feltételeket elegendő csupán kiválasztani. Amennyiben csak az ablakokat szükséges kiemelnünk, akkor az alább felsoroltakat kell követnünk:

- 1. Kijelöljük az ablak terméktípust.
- 2. Az eszközsoron a Szűrés kijelöléssel ikonra vagy a Rekordok menü / Szűrés kijelöléssel parancsra kattintunk
- 3. Megjelenik a ablakok listája



18. ábra. Az adatok szűrése

7.7. Lekérdezések

Adott tulajdonságú adatok listázásának másik módja a lekérdezés. A lekérdezések segítségével az adatbázisból **megjeleníthetjük, módosíthatjuk, törölhetjük** az adott feltételeknek megfelelő adatokat.

A lekérdezések elmenthetőek és bármikor újra felhasználhatjuk őket. Egy táblához több lekérdezés is készíthető, de egy lekérdezés több tábla adatait is megjelenítheti. Leggyakrabban alkalmazott lekérdezések:

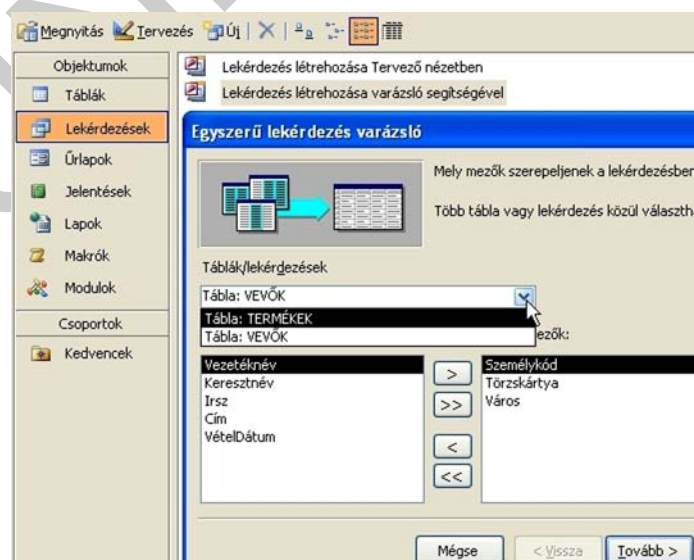
- **Választó lekérdezés** – a megadott feltételeknek megfelelő rekordokat mutatja meg
- **Törlő lekérdezés** – a feltételeknek megfelelő rekordokat törli
- **Frissítő lekérdezés** – a feltételeknek megfelelő adatok módosítja

Két módon is lehetséges lekérdezéseket elkészíteni. Az egyik a QBE-rács (Query By Example), amikor a lekérdezési paramétereket vizuálisan állíthatjuk össze. A másik az SQL nyelv (Structured Query Language), amellyel matematikai, logikai úton adjuk meg a lekérdezési feltételeket. Fontos tudni, hogy mindkét esetben ugyanaz a végeredmény, mivel a QBE-rácson megadot paraméterek is az SQL nyelv utasításait állítja elő.

7.7.1. Lekérdezés készítése

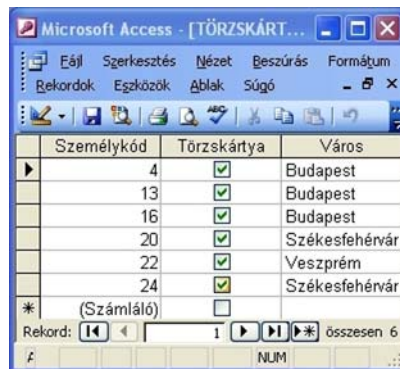
A kereskedelmi részleg vezetője szeretné megtudni, mennyi a törzskártyával vásárlók száma. ABLAKGYÁR adatbázisunk VEVŐK táblájából tudjuk elkészíteni a Törzskártya lekérdezést!

- 1. Válasszuk ki az Objektumok listán a Lekérdezések elemet!
- 2. Kattintsunk duplán a Lekérdezés létrehozása varázsló segítségével sorral!
- 3. Válasszuk ki a VEVŐK táblát!
- 4. Adjuk meg a lekérdezéshez szükséges mezőket! /19. ábra./



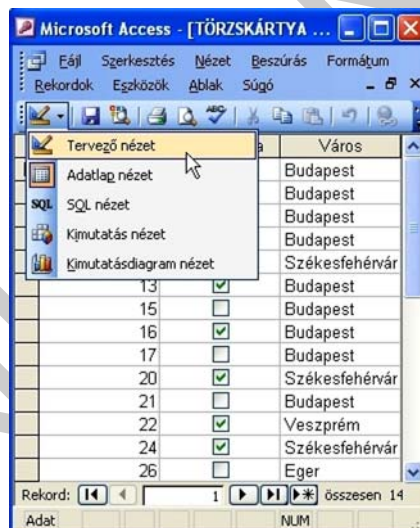
19. ábra. Lekérdezés készítése

- 5. Kattintsunk a Tovább gombra!
- 6. Adjuk meg a TÖRZSKÁRTYA lekérdezés nevet!
- 7. Elkészült a TÖRZSKÁRTYA lekérdezésünk.



20. ábra. Törzskártya lekérdezés

- 8. Váltunk Tervező nézetre!



21. ábra. Tervezői nézet

- 9. A Törzskártya mezőben megadjuk a lekerdezés logikai /igen/ feltételét!
- 10. Futtassuk le a lekérdezést!

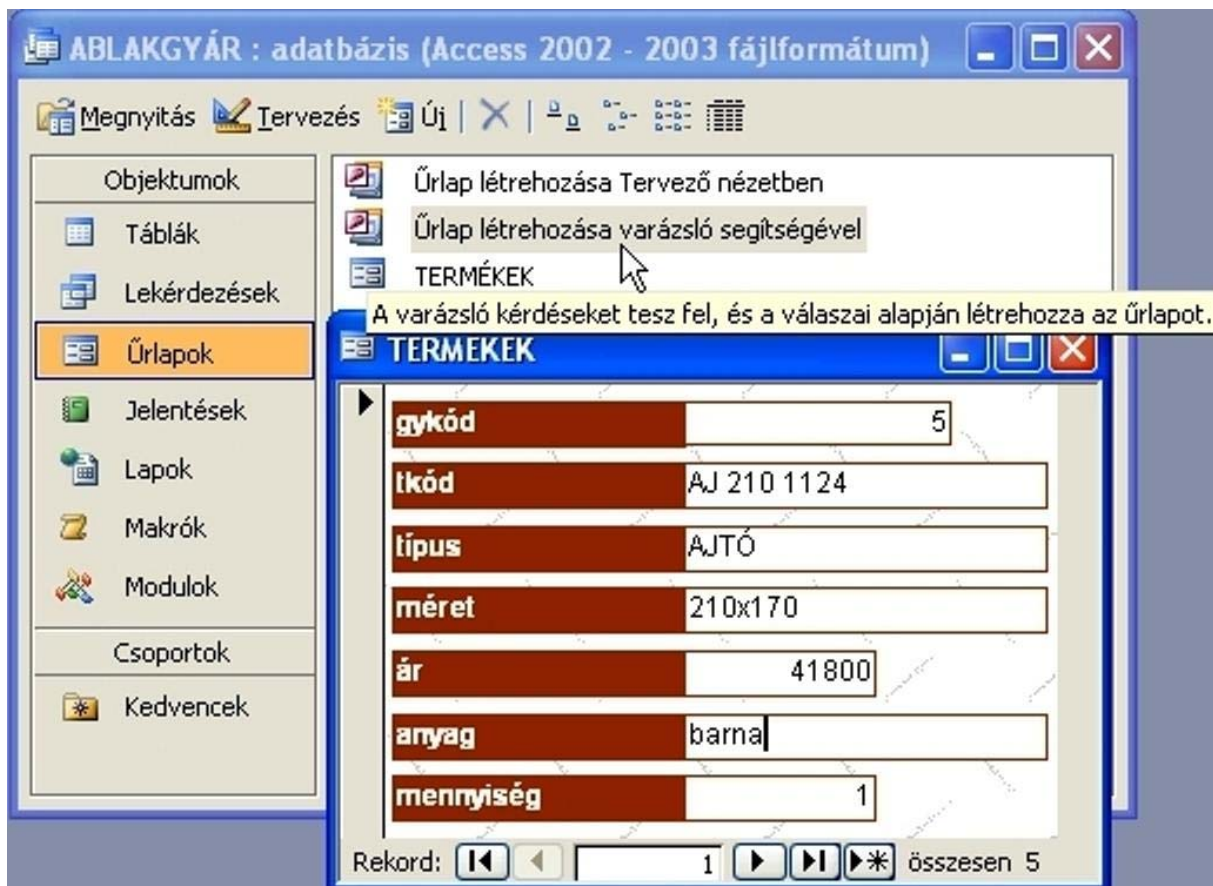


22. ábra. Lekérdezés futtatása

A lefuttatott lekérdezésben már csak a Törzskártyával rendelkező vevőink látszanak.

7.8. Űrlap létrehozása

Az Űrlap, az adatok bevitelét és módosítását megkönnyítő, látványos objektum. Készülhet adattáblából vagy akár lekérdezésből is. Egy igazán jól használható űrlap megtervezése időgényes folyamat, ezért az adatbáziskezelők varázslókkal segítik a gyors és automatikus elkészítésüket. Ezért többnyire az a gyakorlat, hogy az űrlapot a varázslóval készítjük, majd utólag módosítjuk, ha szükséges. Elkészítése során a varázsló kérdéseket tesz fel és a válaszaink alapján létrehozza az űrlapot. A varázslóval nagyon hasonló lépések szerint tudunk űrlapokat készíteni, mint az előzőleg ismertetett lekérdezéseket.



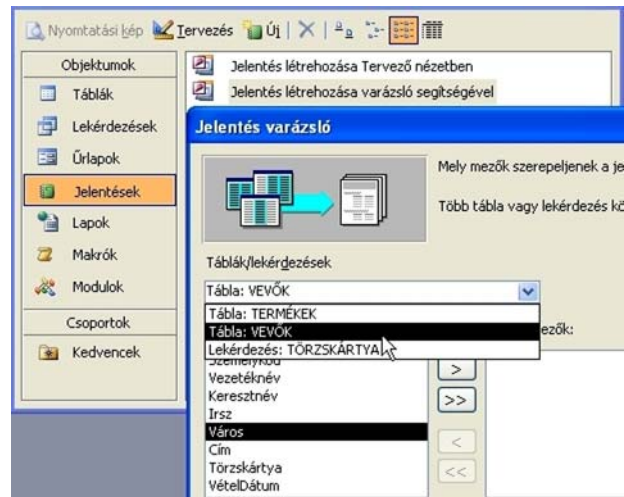
23. ábra. Úrlap létrehozása

7.9. Jelentés készítése

A jelentés elsősorban az adatok nyomtatható formában történő megjelenítését teszik lehetővé. A jelentés segítségével adatainkat csoportosíthatjuk, rendezhetjük és leggyakrabban kinyomtatjuk, de elküldhetjük e-mail mellékletként is. A jelentéseket az űrlapokhoz hasonlóan varázslóval készítjük, formázhatjuk és utólag módosítjuk. A jelentések alapjául szolgáló adatok származhatnak táblából, illetve lekérdezésekből. Amennyiben több táblából származó adatokat szeretnénk megjeleníteni, célszerű ezeket egy lekérdezésben összefoglalni és erre alapozni a jelentést.

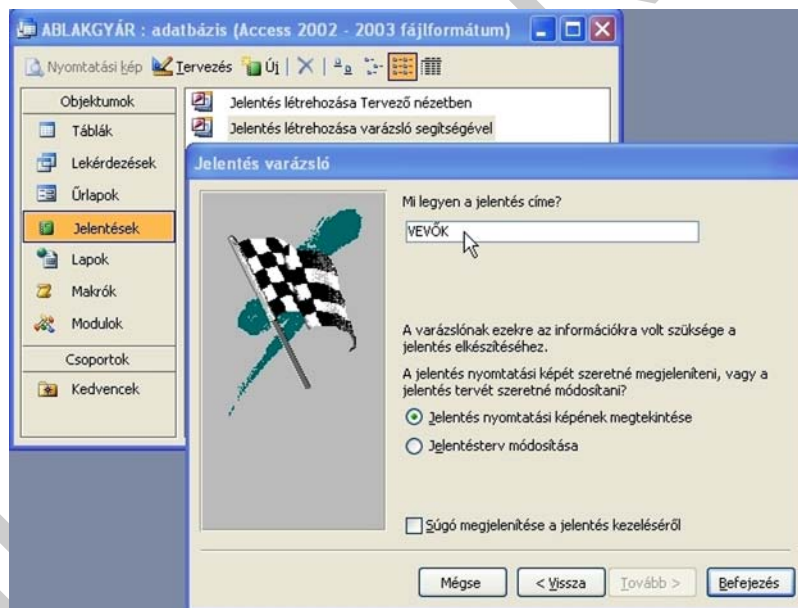
A kereskedelmi részleg megbeszélésén szeretné megmutatni a munkatársainak, hogy melyik településekről és kik a törzskártyával vásárlók. Ezt az adatbázisból egy VEVŐK jelentés elkészítésével egyszerűen meg lehet oldani.

- 1. Válasszuk ki az Objektumok listán a Jelentések elemet!
- 2. Kattintsunk duplán a Jelentés létrehozása varázsló segítségével sorra!
- 3. Válasszuk ki a VEVŐK táblát!



24. ábra. Vevők tábla

- 4. Adjuk meg a jelentéshez szükséges mezőket!
- 5. Kattintsunk a Tovább gombra!
- 6. Adjuk meg a VEVŐK jelentés nevet!



25. ábra. Elnevezés

- 7. Kattintsunk a Tovább gombra!
- 8. Elkészült a VEVŐK jelentésünk.

VEVŐK

<i>Törzskártya</i>	<i>Város</i>	<i>VételDátum</i>	<i>Személykód</i>
<i>Igen</i>			
	Budapest	1982.03.02.	16
	Budapest	1983.10.26.	4
	Budapest	1991.06.04.	13
	Székesfehérvár	1981.06.12.	24
	Székesfehérvár	1982.07.11.	20
	Veszprém	1982.10.19.	22
<i>Nem</i>			
	Budapest	1979.10.27.	15
	Budapest	1980.12.30.	3
	Budapest	1981.05.12.	1
	Budapest	1984.11.11.	21
	Budapest	1986.03.01.	5
	Budapest	1994.04.15.	17
	Eger	1983.10.20.	26
	Székesfehérvár	1980.01.30.	11

26. ábra. Vevők jelentése

A jelentés alapjául szolgáló tábla vagy lekérdezés adatainak módosításakor azok automatikusan frissülnek és megjelennek a jelentésben is. Azonban a tábla szerkezetének módosítása esetén – ha például egy mezőt törölünk – a jelentésbe is át kell vezetnünk a változtatást.

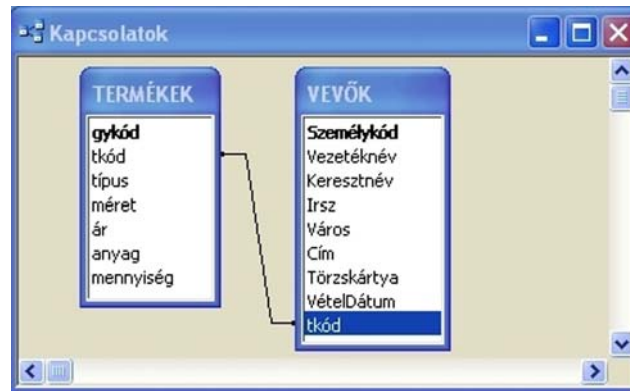
8. KAPCSOLAT A TÁBLÁK KÖZÖTT

A kapcsolat két tábla között a kapcsolómezőn keresztül valósul meg. Kapcsolómezőnek nevezzük azt a mezőt, amelyik mindkét táblában megtalálható, mindkettőnek ugyanaz a tulajdonsága, mindkettő azonos tartalmú. Általában a táblákban ugyanazzal a mezőnével nevezzük el őket. A legtöbb esetben, az egyik táblában ez a mező úgynevezett elsődleges kulcs, amely mint egyedi azonosító szolgál. A másik táblában, mint kapcsoló mező szerepel az adott mező.

A kapcsolatot úgy hozzuk létre két tábla között, hogy az egyik tábla kulcsmezőjét áthúzzuk annak a táblának a kapcsolómezőjére, amellyel létre szeretnénk hozni a kapcsolatot.

A Kapcsolatok ablakban beállítható a hivatkozási integritás tulajdonság, melynek hatására a kapcsolódó adatokat nem lehet véletlenszerűen törölni vagy módosítani. Ahhoz, hogy a hivatkozási integritás beállítható legyen, teljesülnie kell a következő feltételek mindegyikének:

- Az elsődleges tábla megfelelő mezője egy elsődleges kulcs.
- A kapcsolódó mezőknek ugyanaz az adattípusuk. Kivételt képez a Számláló és a Szám típusú mező, ahol a számláló kapcsolatba hozható egy hosszú egész típusú szám mezővel.
- Mindegyik tábla ugyanabban az adatbázisban található.



27. ábra. Táblák közötti kapcsolat

A táblák közti kapcsolatok az egyedek egymáshoz való viszonyát írják le. Az egyedek közti kapcsolatot háromféleképpen írhatjuk le.

- 1. egy-egy (1:1) kapcsolat: az egyik tábla egy eleméhez a másik tábla pontosan egy eleme kapcsolódik
- 2. egy-több (1:N) kapcsolat: az egyik tábla egy eleméhez a másik tábla több eleme is tartozhat
- 3. több-több (N:M) kapcsolat: bármely tábla elemeihez a másik tábla tetszőleges számú eleme tartozhat
- Az adatbáziskezelő a táblákat, kapcsolómezőkön keresztül vonallal köti össze.

9. ANGOL, MAGYAR BETŰSZAVAK ÉS KIFEJEZÉSEK

- **ABR** – Adatbázisrendszer
- **CASE** – Computer Aided Software Engineering
- **CDD** – Common Data Dictionary
- **DDL** – Data Definition Language (adatdefiníciós nyelv)
- **DML** – Data Manipulation Language (adatmanipulációs nyelv)
- **DCL** – Data Control Language (adatvezérlő nyelv)
- **DBS** – Data Base System
- **DB** – Data Base (adatbázis)
- **DBA** – Data Base Administrator (adatbázis adminisztrátor)
- **DBMS** – Data Base Management System (adatbázis-kezelő rendszer)
- **ER** – Entity Relationship
- **ODL** – Object Definition Language (objektumdefiníciós nyelv)
- **OODBMS** – Object Oriented DBMS
- **OQL** – Object Query Language
- **PCR** – Parent Child Relationship
- **QBE** – Query By Example (lekérdezés példa alapján)
- **QL** – Query Language

ADATBÁZISKEZELÉS

- **RDBMS** – Relational Data Base Management System (relációs adatbázis-kezelő rendszer)
- **SQL** – Structured Query Language (struktúrált lekérdező nyelv)
- **VPCR** – Virtual Parent Child Relationship

Összefoglalás A fent leírtak bizonyára sok segítséget adhatnak egy fokozatosan változó vállalkozás vagy intézmény tevékenységeinek és az ezekhez kapcsolódó információk, adatok naprakész feldolgozásához, tárolásához. Fontos, hogy az adatok tárolását, feldolgozhatóságát gondosan tervezzük meg, majd azt követően fogjunk hozzá az elképzelések megvalósításához. Amennyiben eszerint járunk el, sokkal nagyobb a siker lehetősége, bár nem egyszer így is előfordulhat, hogy újra és újra át kell terveznünk az adatbázis rendszerünket. *Az adatbázisban az egyed tulajdonságait egy táblában tároljuk. A tábla soraiba (rekordjaiba) kerülnek az egyedtípusok tényleges előfordulásai, a tábla oszlopaiba (mezőibe) az attributumok. Minden adatot csak egy táblában tároljunk, hogy később csak egy helyen kelljen frissítenünk azokat. Egy táblában csak egy adott témára vonatkozó információ legyen, így az egyes témákra vonatkozó adatokat egymástól függetlenül lehet törölni vagy megtartani.* Mindemellett fokozott figyelemmel legyünk a biztonságos adatmentésre és adatvédelemre is !

TANULÁSIRÁNYÍTÓ

Végezze el az alábbi tevékenységeket!

A feladatok megoldása során használja a adatbázis-kezelő program súgóját, valamint a szakmai információtartalmat!

1.feladat

Készítse el az Ön által használt adatbázis-kezelő programmal, az alábbi táblát! Az így elkészített táblát mentse el TERMÉKADATOK néven !

gykód	tkód	típus	méret	ár	anyag	mennyiség
5	AJ 210 1702	AJTÓ	210x170	41800	barna	1
3	EJ 190 1702	ERKÉLY AJTÓ	190x170	39250	barna	4
1	AB 145 1141	ABLAK	145x114	29550	fehér	3
2	AB 120 1502	ABLAK	120x150	19550	barna	6
4	AB 120 1201	ABLAK	120x120	16500	fehér	5

Egészítse ki a táblát további 5–10 rekorddal az alábbi feladatok megvalósításához megfelelően !

1.1. Készítsen űrlapot a tábla adatainak bevitelére!

1.2. Készítsen az előzőekben ismertett, az Ön által meghatározott feltételekkel választó lekérdezést!

1.3. Készítsen jelentést, amellyel a termékek típusonkénti mennyiségét, méretét és árát tudja nyomtatható formára elkészíteni!

2.feladat

Olvassa el és válaszolja meg az alábbi angol nyelvű szakmai tesztet, amely négy adatbázis-kezeléssel kapcsolatos kérdést tartalmaz ! Értelmezze a válaszokat, majd írja le magyarul, a helyes válasz betűjével együtt !

Each question has one correct answer. Answer each of the four questions by writing your answer into the answer row of each question.

Question 1

Data is stored on the Customers of a company. What will be held in a field in the database ?

A.	All the data for the application
B.	The data for one customer
C.	An individual data item
D.	The data for all customers

Answer:

Question 2

Which of these describes the purpose of a key field ?

A.	It uniquely identifies a record
B.	It uniquely identifies a table
C.	It uniquely identifies a field
D.	It uniquely identifies a database

Answer:

Question 3

Which data type would you choose to store a value that can only be either True or False ?

A.	Text
B.	Date
C.	Numeric
D.	Yes/No

Answer:

Question 4

Which of these is necessary if two tables are to be related ?

A.	The tables must have the same primary key
B.	The tables must have a field that contains the same data
C.	The tables must be indexed
D.	The tables must have the same length

Answer:

MEGOLDÁSOK

1.feladat



28. ábra.

TERMEKEK

<i>mennyiség</i>	<i>típus</i>	<i>ár</i>	<i>méret</i>
1	AJTÓ	41800	210x170
4	ERKÉLY AJTÓ	39250	190x170
3	ABLAK	29550	145x114
6	ABLAK	19550	120x150
5	ABLAK	16900	120x120

29. ábra.

2.feladat

1. D – Answer: A vevőkre vonatkozó adatok
2. A – Answer: Egyedileg azonosít rekordot.
3. D – Answer: Igen/Nem
4. A – Answer: A tábláknak azonos elsődleges kulcsuknak kell lennie.

ÖNELLENŐRZŐ FELADATOK

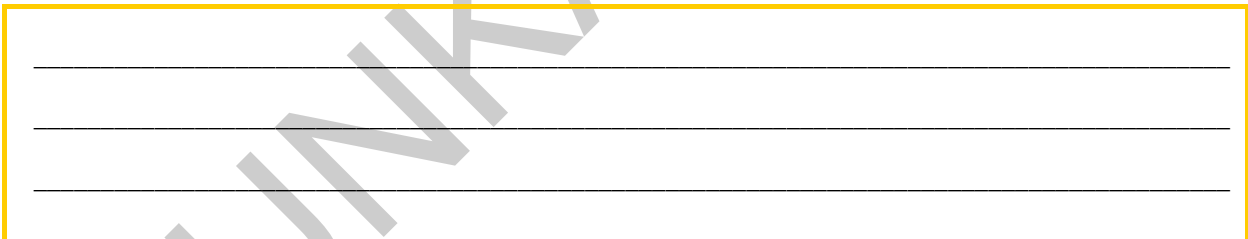
1. feladat

Határozza meg, mikor lesz az adatból információ?



2. feladat

Sorolja fel, melyek a modern adatbáziskezelő rendszer leggyakoribb objektumait ?



3. feladat

Írja le vagy ábra segítségével mutassa be egy tábla részeit!



4. feladat

Röviden írja le vagy ábra segítségével ismertesse az adatbázis tervezés folyamatát!

5. feladat

Soroja fel és jellemezze táblák közötti kapcsolatokat!



MUNKAANYAG

MEGOLDÁSOK

1. feladat

Az adathalmaz akkor alakul át adatbázissá, ha az valamilyen meghatározott szempontok szerint felépített és lehetővé teszi az adatok szisztematikus kezelését, elérhetőségét. **Tehát az adatból, a feldolgozás során válik információ.**

2. feladat

- táblák
- lekérdezések
- űrlapok
- jelentések

3. feladat

Mező: A táblázat oszlopai egy-egy az egyedre jellemző tulajdonságot tartalmaznak, ezeket az oszlopokat mezőknek nevezzük és mezőnévvel azonosítjuk.

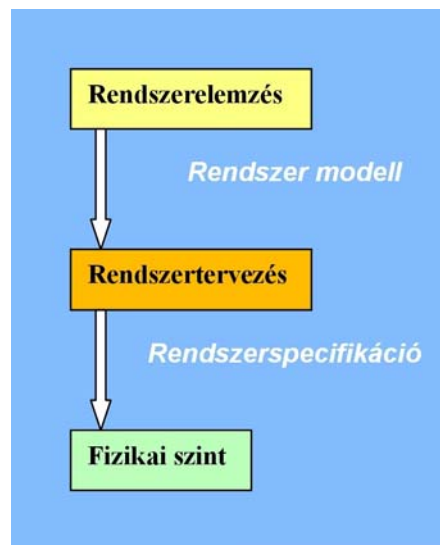
Rekord: Egy táblázatba soronként egy-egy egyedet viszünk fel, amelyet rekordnak nevezünk és azonosítóval látjuk el.

	MEZŐ			
	Azonosító	Név	Születési hely	Születési idő
	0001	Tóth Imre	Kiskunhalas	1941. szept. 16.
REKORD	0002	Kovács Katalin	Makó	1967. júl. 22.
	0003	Szép Ibolya	Kaposvár	1982. jan. 11.
	0004	Horváth Zoltán	Szolnok	1976. márc. 15.

30. ábra.

4. feladat

Egy adatbázis tervezése során először az adatbázisunkat meghatározó rendszer felépítését, működését kell megismernünk és elemzésekkel meghatároznunk a feldolgozandó adatok körét, valamint az adatbázissal szemben támasztott igényeket. Ezt követően megvizsgáljuk a logikailag összetartozó adatok közötti kapcsolatokat. Így hozzuk létre a tervezett rendszer specifikációját vagy logikai modelljét. Ezután a felhasználható szoftver és hardver körülmények figyelembe vételével fizikai szinten is létrehozuk az adatbázisunkat.



31. ábra.

5. feladat

A táblák közti kapcsolatok az egyedek egymáshoz való viszonyát írják le. Az egyedek közti kapcsolatot háromféleképpen írhatjuk le.

1. egy-egy (1:1) kapcsolat: az egyik tábla egy eleméhez a másik tábla pontosan egy eleme kapcsolódik
2. egy-több (1:N) kapcsolat: az egyik tábla egy eleméhez a másik tábla több eleme is tartozhat
3. több-több (N:M) kapcsolat: bármely tábla elemeihez a másik tábla tetszőleges számú eleme tartozhat

IRODALOMJEGYZÉK**FELHASZNÁLT IRODALOM**

Dr. Katona Endre: Adatbázisok, Előadási jegyzet, 2004

Jeffrey D. Ullman – Jennifer Widom: Adatbázisrendszerek, Panem kiadó, 2008

Illés Zoltán: Számítástechnikai szoftverek – Biztonsági másolatok kezelése, adatvédelem, NSZFI, 2010

<http://www.kurt.hu> (2010-08-03) – adatmentés, adatbiztonság

[http://hu.wikipedia.org/wiki/Adatvédelem_\(jog\)](http://hu.wikipedia.org/wiki/Adatvédelem_(jog)) (2010-08-03)

AJÁNLOTT IRODALOM

Békési – Geda – Holovács – Perge : Adatbázis kezelés, Líceum kiadó, Főiskolai jegyzet (Eger)

Bódy Bence: Az SQL példákon keresztül, Jedlik Oktatási Stúdió

Joe Celko: SQL fejtörők, Kiskapu kiadó

A(z) 1180-06 modul 007-es szakmai tankönyvi tartalomeleme felhasználható az alábbi szakképesítésekhez:

A szakképesítés OKJ azonosító száma:	A szakképesítés megnevezése
55 481 01 0000 00 00	Általános rendszergazda
55 481 02 0000 00 00	Informatikai statisztikus és gazdasági tervező
55 481 03 0000 00 00	Telekommunikációs asszisztens
55 481 04 0000 00 00	Web-programozó
55 810 01 0010 55 01	Energetikai mérnökasszisztens
55 810 01 0010 55 02	Építettkörnyezetmérnök-asszisztens
55 810 01 0010 55 03	Faipari terméktervező
55 810 01 0010 55 04	Faipari termelés-szervező
55 810 01 0010 55 05	Gépipari mérnökasszisztens
55 810 01 0010 55 06	Hálózati informatikus
55 810 01 0010 55 07	Kohómérnök asszisztens
55 810 01 0010 55 08	Könnyűipari mérnökasszisztens
55 810 01 0010 55 09	Mechatronikai mérnökasszisztens
55 810 01 0010 55 10	Műszaki informatikai mérnökasszisztens
55 810 01 0010 55 11	Vegyész mérnökasszisztens
55 810 01 0010 55 12	Vegyipari gépészmérnök-asszisztens
55 810 01 0010 55 13	Villamosmérnök-asszisztens

A szakmai tankönyvi tartalomelem feldolgozásához ajánlott óraszám:

50 óra

MUNKANYAG

A kiadvány az Új Magyarország Fejlesztési Terv
TÁMOP 2.2.1 08/1-2008-0002 „A képzés minőségének és tartalmának
fejlesztése” keretében készült.

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

Kiadja a Nemzeti Szakképzési és Felnőttképzési Intézet
1085 Budapest, Baross u. 52.

Telefon: (1) 210-1065, Fax: (1) 210-1063

Felelős kiadó:
Nagy László főigazgató