

Premontrei Szent Norbert Gimnázium

Számítástechnika érettségi

szóbeli tételsor

ADATBÁZIS-KEZELÉS

ADATBÁZIS-KEZELÉS

Tartalomjegyzék

1) TÉTEL Az adatbázis-kezelés alapfogalmai	1
a) Az adatbázis fogalma, adatmodellek típusai	1
b) Adatbázis-kezelési funkciók (DDL, DML, DCL).....	3
c) A redundancia fogalma.....	3
2) TÉTEL Relációs adatbázis.....	5
a) A relációs adatbázis felépítése (egyed, egyedtípus, attribútum fogalma).....	5
b) Az egyedi, az elődleges és az idegen kulcs fogalma (elsődleges és másodlagos adatok).....	5
c) Adattáblák összekapcsolása, a kapcsolatok típusai, a hivatkozási integritás fogalma.....	6
3) TÉTEL Adatbázis-táblázatok szerkezete, tábla létrehozása	7
a) A mező jellemzői (mezőnév, mezőtípus, hossz).....	7
b) Az index fogalma. Rendezés, gyorskeresés.....	8
c) Műveletek, függvények. A számított mező fogalma.....	8
4) TÉTEL Adatbevitel, karbantartás	10
a) Ellenőrzött adatbevitel.....	10
b) Rekordok törlése (törlő lekérdezés).....	10
c) Csoportos adatmódosítás (módosító lekérdezés).....	11
d) Új táblát létrehozó és hozzáfűző lekérdezés.....	12
5) TÉTEL Űrlapok.....	13
a) Az űrlapok típusai. Az űrlap részei, az egyes részek funkciója.....	13
b) Az űrlapokon használatos vezérlőelemek.....	14
c) Adatbeviteli űrlapok tervezése és készítése.....	14
d) Segédűrlap fogalma, alkalmazása.....	15
6) TÉTEL Adatok megjelenítése I.: a kiválasztó lekérdezés	16
a) Kiválasztó lekérdezések szerkesztése, válogatási feltételek megadása.....	16
b) Összesítés, csoportosítás, aggregát műveletek.....	16
c) Keresztábrás lekérdezés.....	17
7) TÉTEL Adatok megjelenítése II.: a jelentés.....	18
a) A jelentések típusai, a jelentés szerkezete.....	18
b) A jelentéseken használatos vezérlőelemek és rendszerváltozók.....	18
c) Levélcímkék nyomtatása.....	18
d) Összefokozatos, sorszámozott lista készítése.....	19

1) TÉTEL Az adatbázis-kezelés alapfogalmai

- Az adatbázis fogalma, adatmodellek típusai.
- Adatbázis-kezelési funkciók (DDL, DML, DCL).
- A redundancia fogalma.

a) Az adatbázis fogalma, adatmodellek típusai.

ADAT Puszta tény, ismeret.

INFORMÁCIÓ A felhasználók számára értelmezhető adat.

AZ ADATMODELLEK ELEMELI

EGYED (entitás) Konkrét dolgok valamilyen nézőpontból tekintett, adott tulajdonságokkal rendelkező halmaza.

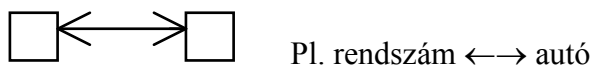
Pl. Egy személy nyilvántartási szempontból DOLGOZÓ a munkahelyén, ADÓZÓ az adóhivatal szemszögéből, BETEG egy kórházban stb.

TULAJDONSÁG (attributum) Az egyedek jellemzőinek halmaza.

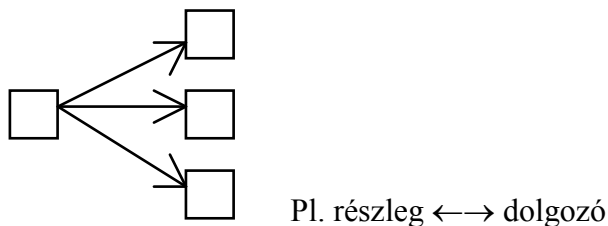
Pl. név, személyi szám, lakcím, adószám, munkabér, stb.

ADATKAPCSOLAT (reláció) Egyedek és tulajdonságok közti viszony. 3 féle lehet:

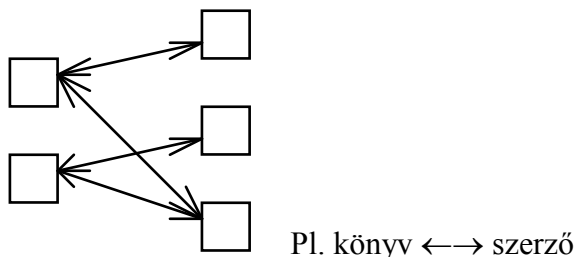
1:1 - Egy az egyhez típusú (One to One)



1:N - Egy a többhöz típusú (One to Many)



N:M - Több a többhöz típusú (Many to Many)



ADATMODELL Egyedek, tulajdonságaik és a köztük lévő kapcsolatok ábrázolása.

AZ ADATMODELLEK TÍPUSAI

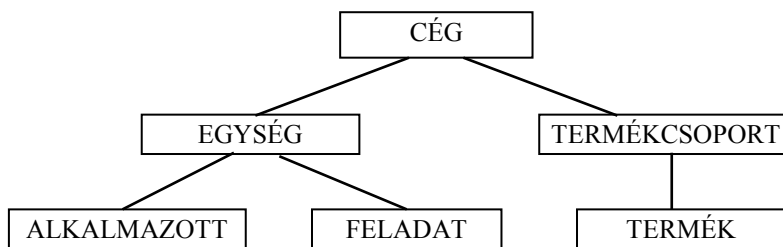
1. FORMÁLIS

Az adatmodellt ebben az esetben egy *gráf* segítségével adjuk meg, amelyben csomópontok jelentik az egyedeket, és élek ábrázolják a kapcsolatokat. Ilyen a hálós és hierarchikus modell.

Hálós adatmodell Itt az egyedek mellérendeltségi viszonyban vannak, így hálót alkotnak.



Hierarchikus adatmodell Itt csak alá- és fölrendeltség van, tehát fastruktúrát alkot.



2. SZEMANTIKAI

Az adatkapcsolatokat ebben szövegesen írjuk le.

3. MATEMATIKAI

Az adatmodellt ebben az esetben matematikai értelemben vett relációk segítségével írjuk le. A relációs adatmodellt E.F CODD alkotta meg 1970-ben, amikor kitalálta, hogy tároljuk az adatokat, vagyis az egyes egyedek tulajdonságait táblákban. A táblázat egy sora tulajdonképpen az egyed egy konkrét előfordulásának attribútumait tartalmazza, és így a tábla egyes oszlopai relációt alkotnak. A különböző egyedek közötti kapcsolatot ugyanakkor fizikailag nem tároljuk, hanem kulcsmezőkkel illetőleg kapcsolómezőkkel teremthetjük meg.

Relációs adatmodell

Minden egyed egy táblázat, és a táblázatokat kapcsolómezőkkel köthetjük össze.

ADATBÁZIS (database)

Az adatmodell fizikai megvalósítása. Rendszerezett adatok valamely logikai séma (adatmodell) alapján történő tárolása olyan formában, hogy lehetővé tegye az adatok időszerűsítését, és egyszerű visszakérését.

b) Adatbázis-kezelési funkciók (DDL, DML, DCL).**DDL - Data Definition Language (Adatdefiniáló nyelv)**

Feladata az adatbázis struktúrájának, az adattábláknak a létrehozása, változtatása, és a köztük lévő kapcsolatoknak a definiálása.

CREATE DATABASE/TABLE/VIEW/INDEX név - Adatbázis/tábla/látvány/index létrehozása

DROP DATABASE/TABLE/VIEW/INDEX név - Adatbázis/tábla/látvány/index törlése

DML - Data Manipulation Language (Adatkezelő nyelv)

Feladata az adatok karbantartása (*időszerűsítése*: új adat felvitele, adat módosítása, törlése), *vizsgálata* (keresés, válogatás, rendezés) és különféle *jelentések* készítése.

INSERT INTO tábla - Új sor felvitele egy táblába

UPDATE tábla - Tábla módosítása

DELETE FROM tábla - Sorok törlése egy táblából

SELECT FROM táblák - Lekérdezés az adatbázisból

DCL - Data Control Language (Adatfelügyelő nyelv)

Feladata az adatvédelem, az adatbiztonság és az adatintegritás biztosítása. Ez magában foglalja a hozzáférési jogok szabályozását az illetéktelen adatelérés elkerülése érdekében, a konkurens hozzáférés biztosítását, azaz az adatmódosító műveletek szinkronizálását a holtpon (deadlock) megelőzésére, és az adatbázis időszakos mentését illetve a tranzakciók (komplex műveletsorok) naplózását, hogy egy esetleges rendszerösszeomlás után az adatbázis helyreállítható legyen.

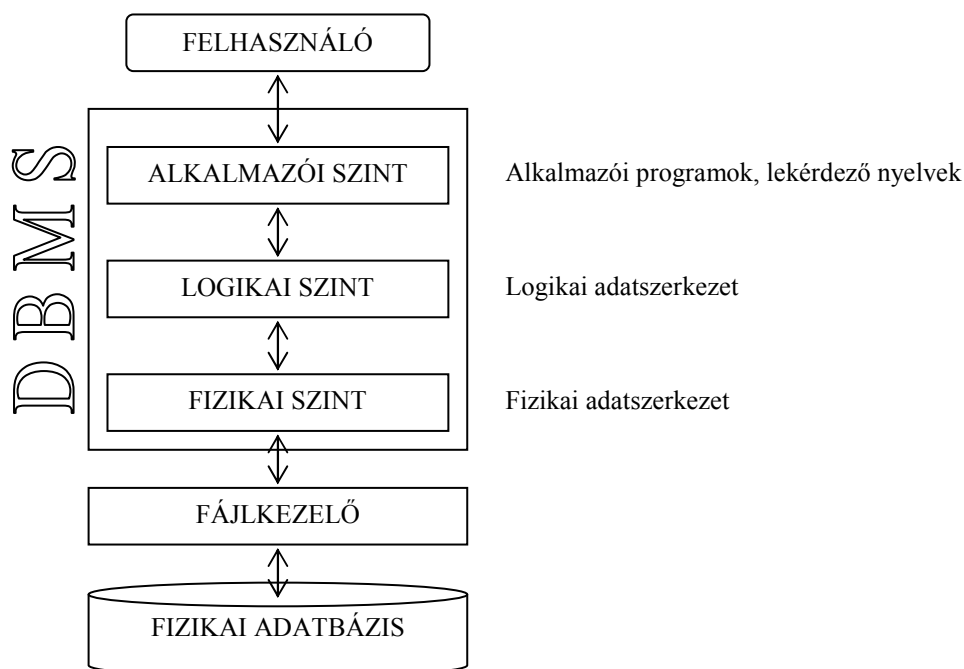
c) A redundancia fogalma.**Adatbázis-kezelő rendszer (Database Management System - DBMS)**

Integrált programcsomag, amely lehetővé teszi adatbázisok létrehozását, karbantartását, az adatok kezelését, és a releváns információ visszakeresését.

Az adatbázis-kezelés koncepciónak (Data Management Concept) három óriási előnye van a fájlkezelő rendszerekkel szemben:

- redundanciamentesség
- adatintegritás megőrzése
- a program és adat függetlenség

Az adatbázis-kezelő rendszerekben az adatkezelésnek különböző szintjei vannak:



Redundancia

A redundancia fölösleges adatismétlést jelent. Ha például a hallgatók nevét, címét és telefonszámát külön tároljuk a tanulmányi eredményeknél és az ösztöndíjknál is. Ez többszörösen foglalja a tárolóterületet, nehézkessé válik az adatbázis kezelése, lassabb lesz az adatelérés, és nehezebb a visszakeresés. A legnagyobb probléma azonban az adatok frissítésekor jelentkezik. Ha valamely adat - mondjuk egy hallgató címe - megváltozik, akkor azt mindenhol következetesen ki kell javítani, ami nem csak sok felesleges munkát jelent, hanem hibalehetőségeket is rejt magában, és így veszélyezteti az adatintegritást. Az egyetlen megoldás, ha minden adatot csak egy helyen tárolunk pontosan ott, ahová tartozik. Ezt megfelelő adatmodell felállításával lehet elérni. Ha harmadik normálformára hozott, kulcsmezőkkel ellátott adattáblákat hozunk létre, akkor más táblákban elég az adat egyedi azonosítójára hivatkozni ahelyett, hogy újra letárolnánk azt. A változásokat így csak a saját táblájában kell végrehajtani.

Adatintegritás

Az adatintegritás az adatok érvényességét, jóságát jelenti. Ez magában foglalja az adatok hitelességét, megbízhatóságát, pontosságát, időszerűségét és ellentmondásmentességét. Ha hibás adat kerül az adatbázisba, vagyis megsértjük az adatintegritást (például 2 azonos kulcs, elírt érték, szám helyett szöveg stb.) belső inkonzisztenciát okozunk, és ez téves információt eredményezhet.

Az adatintegritás ellenőrzött adatbevitellel, a hivatkozási integritás megőrzésének automatikus figyelésével (ez azt jelenti, hogy csak létező kulcsra hivatkozunk), és a javítások konzekvens végigvitelével biztosítható. (Ld. 4.a)

Adatfüggetlenség

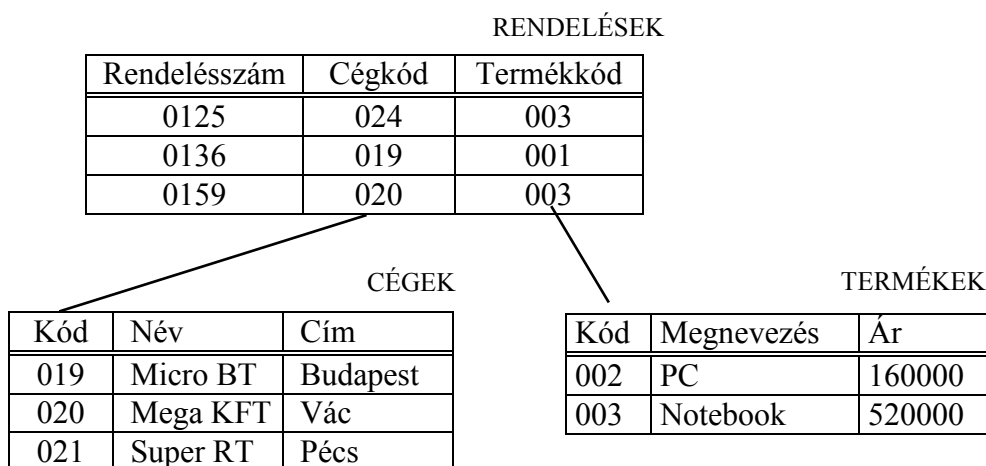
A logikai és fizikai adatfüggetlenség esetén a logikai vagy fizikai adatszerkezet megváltoztatása nincs hatással a felhasználói programokra és megfordítva. Egy új adatmező felvétele miatt például nem kell megváltoztatni a programokat, de egy program módosítása sem vonja maga után az adatok megváltoztatását.

2) TÉTEL Relációs adatbázis

- A relációs adatbázis felépítése (egyed, egyedtípus, attribútum fogalma).
- Az egyedi, az elődleges és az idegen kulcs fogalma (elsődleges és másodlagos adatok).
- Adattáblák összekapcsolása, a kapcsolatok típusai, a hivatkozási integritás fogalma.

a) A relációs adatbázis felépítése (egyed, egyedtípus, attribútum fogalma).

Példa a relációs adatmodellre



EGYED = Tábla (pl. RENDELÉSEK, CÉGEK, TERMÉKEK)

TULAJDONSÁG = Oszlop (pl. a CÉGEK egyed attribútumai a Kód, a Név és a Cím)

EGYEDTÍPUS = Azonos tulajdonságokkal rendelkező egyedek (táblák) halmaza.

Megjegyzés: A tábla sorait rekordoknak, a rekordokhoz tartozó tulajdonságokat mezőknek is nevezik.

b) Az egyedi, az elődleges és az idegen kulcs fogalma (elsődleges és másodlagos adatok).

EGYEDI KULCS = Egy (esetleg több) tulajdonság, amely egyértelműen azonosít egy rekordot, azaz az egyedhalmaz egy elemét, a táblázat egy sorát. Például a rendelésszám alapján azonosítható egy rendelés, vagy név+születési dátum+anyja neve alapján egy személy.

ELSŐDLEGES KULCS = A tábla egy sorának egyedi azonosítója (pl. Rendelészám)

IDEGEN KULCS = Kapcsolómező, amely egy másik tábla elsődleges kulcsára hivatkozik (pl. a RENDELÉSEK táblában a Cégekód a CÉGEK tábla Kód mezőjére mutat)

c) Adattáblák összekapcsolása, a kapcsolatok típusai, a hivatkozási integritás fogalma.

KAPCSOLATOK TÍPUSAI

Egy az egyhez (One To One)

Ez kölcsönösen egyértelmű megfeleltetés, ami azt jelenti, hogy az egyik tábla egy rekordjához a másik táblából csak egy rekord tartozhat, vagy esetleg egy se. Ez megfordítva ugyanúgy igaz.

Például modjuk egy dolgozó legfeljebb egy egységnek lehet a főnöke, és egy egységnek is csak egy főnöke lehet. Persze van olyan alkalmazott, aki egyik egységnek sem főnöke.

Egy a többhöz (One To Many)

Csak az egyik irányban egyértelmű a hozzárendelés, a másik oldalról nézve egy elemhez több is tartozhat a másik táblából. Például egy munkás egyértelmű, hogy melyik egységnél dolgozik, egy egységnél viszont többen is dolgoznak.

Több a többhöz (Many To Many)

Egyik irányban sem egyértelmű a hozzárendelés, tehát egy elemhez több is tartozhat a másik táblából, ugyanakkor a másik tábla egy eleme tartozhat többhöz is. Például egy filmnek több szereplője van, egy színész pedig általában több filmben is játszik.

HIVATKOZÁSI INTEGRITÁS

A hivatkozási integritás azt jelenti, hogy csak olyan kulcsra hivatkozunk, ami létezik a másik táblában.

HIVATKOZÁSI INTEGRITÁS MEGŐRZÉSE

Két tábla összekapcsolásakor bekapcsolhatjuk a hivatkozási integritás megőrzése opciót, aminek hatására automatikusan figyelni fogja a rendszer, hogy ne vihessünk be rossz hivatkozást, vagy ne módosíthassunk illetve törölhessünk ki olyan kulcsot, amire egy másik táblában hivatkoznak. Amennyiben olyan műveletet kezdeményezünk, ami megsértené a hivatkozási integritást, a rendszer figyelmeztetést ad erről, és nem hagyja elvégezni.

További lehetőség a hivatkozási integritás biztosítására, ha egy kulcs módosításának vagy törlésének következményeit konzekvensen kijavítjuk a többi táblában is, ahol hivatkoztak rá.

KASZKÁDOLT FRISSÍTÉS

Ha megváltoztatunk egy kulcsot, akkor mindenhol javítja, ahol hivatkoznak rá.

KASZKÁDOLT TÖRLÉS

Ha kitörlünk egy kulcsot, akkor minden olyan rekordot töröl, ami hivatkozott rá.

3) TÉTEL Adatbázis-táblázatok szerkezete, tábla létrehozása

- a) A mező jellemzői (mezőnév, mezőtípus, hossz).
 b) Az index fogalma. Rendezés, gyorskeresés.
 c) Műveletek, függvények. A számított mező fogalma.

a) A mező jellemzői (mezőnév, mezőtípus, hossz).

Adattípus (DataType)	Értéktartomány	Tizedes pontosság	Tárolási méret
Szöveg (Text)	Maximum 255 karakter		Maximum 255 bájt Nem foglal helyet az üres szövegmezőknek
Feljegyzés (Memo)	Maximum 65535 karakter		Maximum 65535 bájt
Szám (Number)			
Bájt (Byte)	0 - 255	Nincs	1 bájt
Egész (Integer)	-32 768 +32 767 ($-2^{15}; 2^{15}$)	Nincs	2 bájt
Hosszú egész (Long Integer)	-2 147 483 648 - +2 147 483 647 ($-2^{31}; 2^{31}$)	Nincs	4 bájt
Egyszeres (Single)	-3,402823E38 - -1,401298E-45, +1,401298E-45 - +3,402823E38	7 értékes jegy	4 bájt
Dupla (Double)	-1,79769313486231E308 - -4,94065645841247E-324, +1,79769313486231E308 - +4,94065645841247E-324	15 értékes jegy	8 bájt
Pénznem (Currency)	A tizedesjeltől balra 15 számjegy állhat.	A tizedesjeltől jobbra 4 számjegy állhat.	8 bájt
Számláló (Counter)	= Hosszú egész		4 bájt
Dátum/Idő (Date)	100.01.01. - 9999.12.31		8 bájt
Igen/Nem (Yes/No)	Igen/Nem, Igaz/Hamis vagy Be/Ki		1 bit
OLE objektum	Access táblához csatolt, vagy abba beágyazott objektum (pl Excel adatlap, Word dokumentum, grafika, hang, ...).		Maximum 1 gigabájt (a rendelkezésre álló szabad lemez hely korlátozza).
Hiperhivatkozás	A hiperhivatkozás-cím három részből állhat: 1. megjelenített szöveg 2. szöveg cím - egy fájl elérési útja (UNC elérési út) vagy oldal (URL). 3. alcím - egy hely a fájl vagy oldalon		A Hiperhivatkozás típusú adat mindhárom része maximum 2048 karaktert tartalmazhat (külön-külön).

b) Az index fogalma. Rendezés, gyorskeresés.

INDEX = A rekordok logikai rendezése valamilyen szempont szerint. Csak a rendezési mezőt vagy kifejezést tárolja és a rekordok sorszámát, amely alapján fizikailag eléri a rekordot. Az indexek alapján a rekordok gyorsan sorba rendezhetők, és egy másik index aktivizálásával könnyen át lehet állni egy másik szempont szerinti sorrendre.

GYORSKERESÉS = A keresést az indextáblában végzi, amely sokkal gyorsabb, mint ha sorban kellene végignézni a rekordokat. Egy n db rekordból álló táblában a hagyományos kereséssel átlagosan n/2 lépés kell a keresett adat megtalálásához, míg az indexállományban történő logaritmikus keresés esetén ez legfeljebb csak $\log_2 n$. Egy 1024 soros adattáblánál például ez maximum 10 hasonlítást jelent az átlagos 512-vel szemben.

c) Műveletek, függvények. A számított mező fogalma.

TÍPUSMŰVELETEK (a kiértékelés sorrendjében)

ARITMETIKAI MŰVELETEK

1.	^	hatványozás	pl.	2^3	8
2.	-	negatív előjel	pl.	$-2*3$	-6
3.	*/	szorzás, osztás	pl.	$8/2*4$	16
4.	\	egész osztás	pl.	$19\backslash 5$	3
5.	mod	maradék	pl.	$19 \text{ mod } 5$	4
6.	+,-	összeadás, kivonás	pl.	$1+4*2^3$	33

KARAKTERLÁNC MŰVELETEK

7.	& +	2 sztring összefűzése	pl.	'Kis'+ ' '+ 'Bea'	'Kis Bea'
----	-----	-----------------------	-----	-------------------	-----------

RELÁCIÓS MŰVELETEK

8.	<, >	kisebb, nagyobb
8.	=, <>	egyenlő, nem egyenlő
8.	<=, >=	kisebb egyenlő, nagyobb egyenlő

LOGIKAI MŰVELETEK

9.	NOT	logikai tagadás	pl.	NOT yes	no
10.	AND	logikai ÉS	pl.	$x < 2 \text{ AND } x > 3$	no
11.	OR	logikai VAGY	pl.	$x < 2 \text{ OR } x > 1$	yes

A kiértékelési sorrendet a prioritási számok (1-11) jelzik.
Az egyforma prioritású műveletek balról jobbra kerülnek végrehajtásra.

FÜGGVÉNYEK

NUMERIKUS FÜGGVÉNYEK

Függvény	Feladata	Példa	Eredmény
ABS(x)	x abszolút értéke	ABS(2-8)	5
INT(x)	x egész része	INT(3.8), INT(-3.8)	3, -4
FIX(x)	a tizedesrész levágása	FIX(-3.8)	-3
SGN(x)	x előjele (-1:x<0; 0:x=0; 1:x>0)	SGN(-3)	-1
SQR(x)	négyzetgyök x	SQR(25)	5
EXP(x)	e az x-ediken	EXP(1)	2.718282
LOG(x)	x természetes logaritmusa (lnx)	LOG(EXP(3))	3
IIF(felt,kif1,kif2)	ha a feltétel igaz kif1, különben kif2	IIF(1>0,'poz','neg')	'poz'
RAND()	véletlenszám generálása: rand∈[0,1)	RAND()	0.47
PI()	A Ludolf féle szám	PI()	3.141593
SIN(x)	sinx	SIN(PI()/6)	0.5
COS(x)	cosx	COS(PI()/6)	SQR(3)/2
TAN(x)	tan(x)	TAN(PI()/4)	1

KARAKTERKEZELŐ FÜGGVÉNYEK

Függvény	Feladata	Példa	Eredmény
LEFT(s,n)	s-ből balról n karakter	LEFT('fakanál',2)	'fa'
RIGHT(s,n)	s-ből jobbról n karakter	RIGHT('fakanál',5)	'kanál'
MID(s,n[m])	s-ből n-től m db karakter	MID('almafa',3,2)	'ma'
LEN(s)	az s sztring hossza	LEN('almafa')	6
LCASE(s)	s csupa kisbetűvel	LCASE('FoxPro')	'foxpro'
UCASE(s)	s csupa nagybetűvel	UCASE('FoxPro')	'FOXPRO'
TRIM(s)	s a végén levő szóközök nélkül	TRIM('Bag')	'Bag'
InStr([n],[s,r])	n-től s-ben hol kezdődik r	InStr('kabala','al')	4

DÁTUMKEZELŐ FÜGGVÉNYEK

Függvény	Feladata	Példa	Eredmény
DATE() Now()	rendszerdátum	DATE()	1997.02.10
YEAR(d)	év	YEAR(DATE())	1997
MONTH(d)	hónap	MONTH(NOW())	2
DAY(d)	nap	DAY(DATE())	10
WEEKDAY(d)	a hét hányadik napja	WEEKDAY(DATE())	2 (hétfő)
TIME()	a rendszeridő	TIME()	15:31

KONVERZIÓS FÜGGVÉNYEK

Függvény	Feladata	Példa	Eredmény
STR(x)	számot szöveggé alakít	STR(1.2)	'1.2'
VAL(s)	szöveget számmá alakít	VAL('23.9')	23.9
CHR(x)	ASCII kódot karakterré alakít	CHR(65)	'A'
ASC(s)	s első karakterének ASCII kódja	ASC('A')	65

4) TÉTEL Adatbevitel, karbantartás

- a) Ellenőrzött adatbevitel.
- b) Rekordok törlése (törlő lekérdezés).
- c) Csoportos adatmódosítás (módosító lekérdezés).
- d) Új táblát létrehozó és hozzáfűző lekérdezés.

a) Ellenőrzött adatbevitel.

Ellenőrzött adatbevitel segítségével elérhetjük, hogy minimálisra csökkentsük az adatintegritás megsértésének kockázatát. Ezt a megfelelően megválasztott adattípusok és mezőtulajdonságok, valamint a kapcsolatok létrehozásánál megadott hivatkozási integritásra vonatkozó beállítások teszik lehetővé.

Mezőtulajdonságok (Field Properties)

Mezőméret (Field Size)	Szöveg esetében a szövegmező maximális hosszát jelenti, a szám típusnál pedig az altípust határozza meg.
Formátum (Format)	A különféle adattípusokhoz előre megadott vagy egyéni formátumok a bevitt adatok megjelenését szabályozzák.
Beviteli maszk (InputMask)	Az adatbevitelt szabályozza, azaz egy maszk segítségével meghatározza, hogy egy mezőbe mit írhatunk bele.
Alapértelmezett érték (Default Value)	A sűrűn előforduló értékek beírásától megkíméli a felhasználót.
Érvényességi szabály (Validation Rule)	Az adat beírása után ellenőrzi, hogy a bevitt érték megfelel-e az itt megadott feltételnek.
Érvényességi szöveg (Validation Text)	Ha rossz adatot vittünk be, ez jelenik meg. Ha ez nincs kitöltve, az érvényességi szabályra utal a hibaüzenetben.
Indexelt (Indexed)	Indexelni lehet úgy, hogy előfordulhat 2 azonos érték, de olyan is van, hogy ez nem megengedett (pl. kulcs esetében).

b) Rekordok törlése (törlő lekérdezés).

AKCIÓ LEKÉRDEZÉS = A törlő lekérdezést, a frissítő és a hozzáfűző lekérdezésekkel együtt úgynevezett akció lekérdezéseknek nevezzük, mivel nem egy virtuális táblát generálnak, hanem módosításokat hajtanak végre az adatbázisban.

TÖRLŐ LEKÉRDEZÉS = A törlő lekérdezés segítségével egy megadott feltételnek megfelelő sorokat, azaz rekordokat törölhetjük ki a kiválasztott adattáblából.

Ezt elvégezzük az SQL DELETE utasítással, vagy a lekérdezés tervező nézetében.

Az SQL DELETE utasítás szintaxisa:

```
DELETE FROM tábla
[WHERE feltétel]
```

Pl. A budapesti vevők törlése: `SELECT FROM Vevő WHERE Cím="Budapest"`

Tervező nézetben ugyanez a következőképpen néz ki:



c) Csoportos adatmódosítás (módosító lekérdezés).

FRISSÍTŐ LEKÉRDEZÉS = A frissítő, vagy módosító lekérdezővel megváltoztathatjuk a megadott feltételnek megfelelő rekordok tulajdonságait, vagyis a kiválasztott sorok egyes mezőihöz új értéket rendelhetünk.

Ezt elvégezhetjük az SQL UPDATE utasítással, vagy a lekérdező tervező nézetében.

Az SQL UPDATE utasítás szintaxisa:

```
UPDATE {tábla}
    SET {oszlop= kifejezés | NULL}
    [WHERE feltétel]
```

Pl. A 11. kódú cég 20%-os áremelése: UPDATE Cikk SET Ár=Ár*1.2 WHERE Gykód=11

Tervező nézetben ugyanez a következőképpen néz ki:



d) Új táblát létrehozó és hozzáfűző lekérdezés.

HOZZÁFŰZŐ LEKÉRDEZÉS = A hozzáfűző lekérdezéssel vihetünk fel egy adattáblába új rekordot, azaz a táblát új sorral illetve sorokkal bővíthetjük. Az adatokat beírhatjuk konstansként, vagy egy másik táblából is vehetjük egy subselect segítségével. Ezt elvégezhetjük az SQL INSERT utasítással, vagy a lekérdezés tervező nézetében.

Az SQL INSERT utasítás szintaxisa:

```
INSERT INTO tábla [oszloplista]
VALUES ({konstans | NULL}) | subselect
```

1.pl Új vevő felvétele: INSERT INTO VEVO VALUES (6,"DATE","Debrecen")

2.pl Az ELADO tábla bővítése a GYOR-i fiók adataival:

```
INSERT INTO ELADO SELECT * FROM GYOR
```

Tervező nézetben ugyanez a következőképpen néz ki:



5) TÉTEL Űrlapok

- Az űrlapok típusai. Az űrlap részei, az egyes részek funkciója.
- Az űrlapokon használatos vezérlőelemek.
- Adatbeviteli űrlapok tervezése és készítése.
- Segédűrlap fogalma, alkalmazása

a) Az űrlapok típusai. Az űrlap részei, az egyes részek funkciója.

AZ ŰRLAPOK OSZTÁLYOZÁSA

Típus szerint

- Űrlapszerű
- Diagram
- Kimutatás

Az űrlap szerkezete szerint

- Oszlopos
- Táblázatos
- Adatlap
- Sorkizárt

AZ ŰRLAP RÉSZEI

Egy űrlap szakaszai

Új szakaszok hozzáadásával megnövelhető az űrlapok hatékonysága. A legtöbb űrlap csak egy törzsszakasszal rendelkezik, de tartalmazhat további szakaszokat is: űrlapfejet, oldalfejét, oldallábat és űrlaplábat.

Tervező nézet

The diagram illustrates the structure of a form in design view. It shows a vertical stack of sections, each with a label and a corresponding control element:

- Űrlapfej**: The top section, containing the text "Northwind üzletház".
- Oldalfej**: The second section, which is currently empty.
- Törzs szakasz**: The main body of the form, containing two rows of labels and input fields: "Terméknév:" and "Egységár:" on the left, and "Terméknév" and "Egységár" on the right.
- Oldalláb**: The section below the main body, containing the text "=[Oldal]".
- Űrlapláb**: The bottom section, which is currently empty.

On the left side of the diagram, there is a legend with labels and arrows pointing to the corresponding sections in the form:

- Űrlapfej
- Oldalfej
- Törzs szakasz
- Oldalláb
- Űrlapláb

Megjegyzés: Az Űrlapfej és Űrlapláb, valamint az Oldalfej és Oldalláb a Nézet menüben kapcsolható be vagy ki.

b) Az űrlapokon használatos vezérlőelemek.

Objektumok kitöltése		Vezérlőelem Varázsló
Felirat		Beviteli mező
Vezérlőelem-csoport		Váltógomb
Választógomb		Jelölő négyzet
Kombi panel		Listapanel
Parancsgomb		Kép
Kötetlen objektumkeret		Kötött objektumkeret
Oldaltörés		Karton vezérlőelem
Segédűrlap/segédjelentés		Vonal
Téglalap		További vezérlők

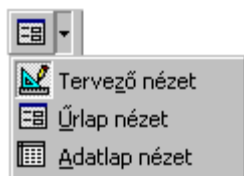
c) Adatbeviteli űrlapok tervezése és készítése.

Egy új űrlap készítése legegyszerűbben az Űrlaptervező Varázslóval végezhető el. Az így generált űrlapot azonban rendszerint érdemes módosítani, amelyet a Tervező nézetben tehetünk meg.

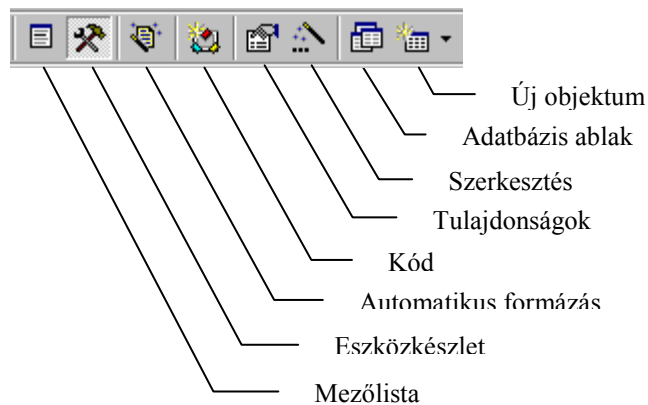
AZ ŰRLAPTERVEZŐ VARÁZSLÓ LÉPÉSEI:

1. Ki kell választani az adattáblát vagy lekérdezést, amelynek mezőit fogjuk felhasználni az űrlapban. Lekérdezésre akkor van szükség, ha több tábla adataival akarunk egyszerre dolgozni.
2. Ki kell választani azokat a mezőket, amelyek szerepelni fognak az űrlapon.
3. Meg kell határozni az űrlap szerkezetét.
4. El kell dönteni, hogy milyen legyen az űrlap megjelenési formátuma.
5. Végül meg kell adni az űrlap címét.

Az űrlap nézetei:



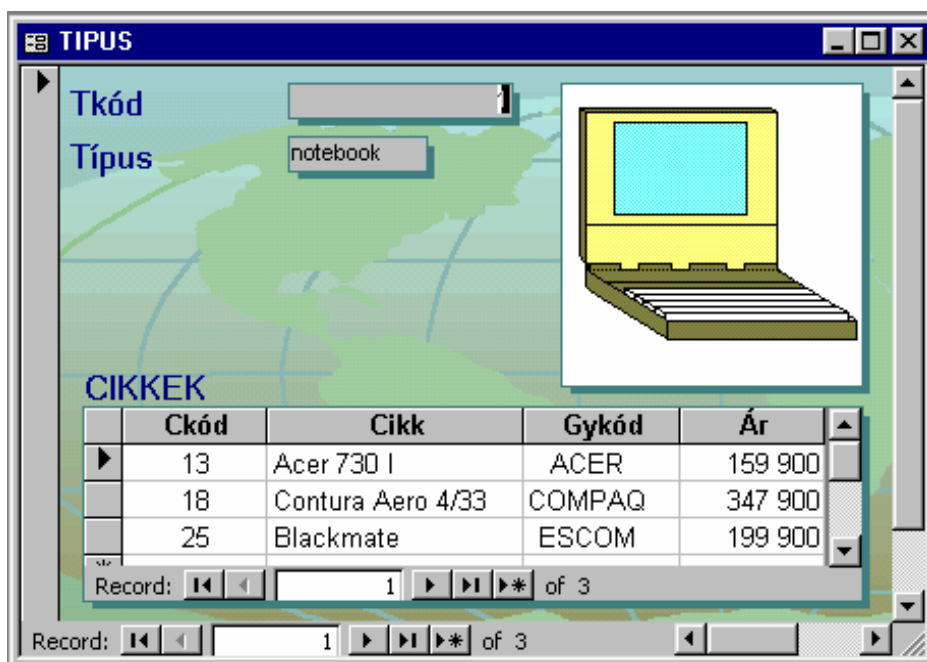
A tervezésnél használt fontosabb ikonok:



d) Segédűrlap fogalma, alkalmazása

SEGÉDŰRLAP = Űrlap, amelyet önálló vezérlőelemként egy másik űrlapba illesztnek. A fő űrlap valamelyik mezőjéhez tartozó adatokat gyűjti össze egy külön táblába.

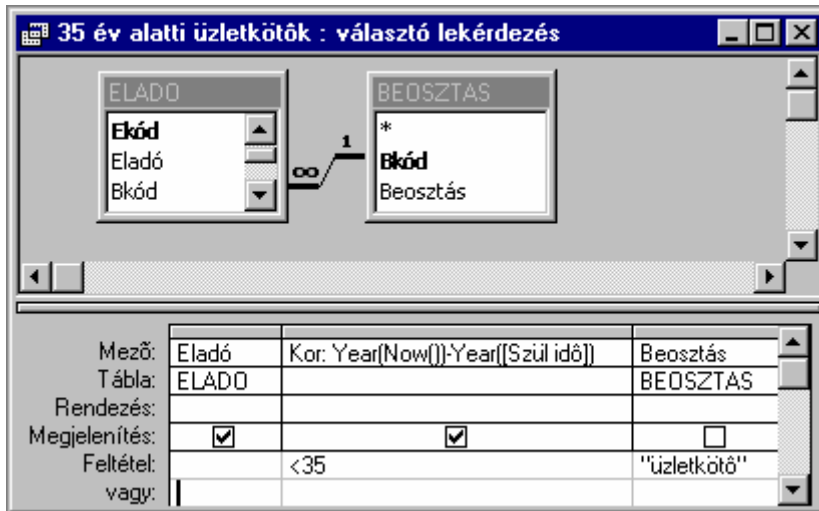
Az alábbi példa a TIPUS űrlapba illesztett CIKKEK segédűrlap működését mutatja. A két űrlap a Tkód alapján össze van kapcsolva, és így mindig az adott típushoz tartozó cikkek jelennek meg a segédűrlapon. A képen látható, hogy jelen esetben csak a notebook-ok látszanak.



6) TÉTEL Adatok megjelenítése I.: a kiválasztó lekérdezés

- a) Kiválasztó lekérdezések szerkesztése, válogatási feltételek megadása.
- b) Összesítés, csoportosítás, aggregát műveletek.
- c) Keresztábrlás lekérdezés.

a) Kiválasztó lekérdezések szerkesztése, válogatási feltételek megadása.



Eladó	Kor
Kiss Tamás	20
Kis Ágnes	23
Kántor Pál	31
Németh István	29

Az azonos sorba írt feltételek között *és kapcsolat*, a különböző sorokba írt feltételek között *vagy kapcsolat* van.

b) Összesítés, csoportosítás, aggregát műveletek.



	Fiók	Férfi	Összbér
►	Budapest	5	605000
	Győr	4	305000
	Pécs	3	330000

c) Keresztábrás lekérdezés.



	Fiók	Férfi	Nő
	Budapest	5	2
▶	Győr	4	1
	Pécs	3	1

7) TÉTEL Adatok megjelenítése II.: a jelentés

- A jelentések típusai, a jelentés szerkezete.
- A jelentéseken használatos vezérlőelemek és rendszerváltozók.
- Levélcímkék nyomtatása.
- Összefokozatos, sorszámozott lista készítése.

a) A jelentések típusai, a jelentés szerkezete.

A JELENTÉSEK TÍPUSAI

- Listaformátum
- Címkeformátum
- Diagram

A JELENTÉSEK SZERKEZETE LEHET

- Oszlopos
- Táblázatos

A JELENTÉS RÉSZEI

Jelentésfej	A jelentés legelején jelenik meg az első lapon
Oldalfej	Minden oldal tetején megjelenik. (általában az oszlopfejléc)
Csoport fejléc	Minden új csoport elején megjelenik
Törzs	Magukat az adatsorokat definiálja. Egy adott csoporthoz tartozó adatok ebben a részben jelennek meg.
Csoport lábléc	Minden csoport végén jelenik meg. (általában csoportösszesítő)
Oldalláb	Minden oldal alján megjelenik. (általában oldalszám, dátum)
Jelentésláb	A jelentés legvégén jelenik meg az utolsó lapon. (pl. összesítő)

b) A jelentéseken használatos vezérlőelemek és rendszerváltozók.

A jelentéseken használt vezérlőelemek lényegében megegyeznek az űrlapokon használtakkal.

c) Levélcímkék nyomtatása.

A címkeformátumú jelentés azt jelenti, hogy az egyes adatsorokban lévő adatok egy-egy címke kerületébe kerülnek, amelyeknek megadható a mérete, és hogy hány oszlopban helyezkednek el a papíron. Ez használható például öntapadós címkék, az úgynevezett etikettek nyomtatásához.

A CÍMKE VARÁZSLÓ LÉPÉSEI:

- Címkeméret és oszlopszám beállítása
- Betűtípus és betűstílus magadása
- Milyen mezők legyenek a levélcímkén
- Mi szerint legyen rendezve
- A jelentés neve

Nagy Pál Géza <u>Budapest</u> Fő u. 1.	Kiss Ágnes <u>Budapest</u> Kis krt. 4.
Szabó Zsuzsa <u>Gödöllő</u> Sarló u. 2.	Kovács Bálint <u>Vác</u> Pipa u. 3.

d) Összefokozatos, sorszámozott lista készítése.

ÖSSZEGFOKOZATOS LISTA

Az összefokozatos lista azt jelenti, hogy az adatokat valamilyen szempont szerint – általában valamelyik mező szerint – csoportosítva jelenítjük meg. A csoportosítás lehet többszintű, vagyis egy csoporton belül is tovább csoportosíthatjuk az adatokat. Minden csoportnak lehet fejléce, ahová általában a csoport megnevezése kerül, és lehet lábléce, ami pedig összesítéseket tartalmazhat az adott csoportra vonatkozóan.

A JELENTÉSVARÁZSLÓ LÉPÉSEI:

1. Adattábla vagy lekérdezés kiválasztása
2. Milyen mezők szerepeljenek a jelentésben
3. Mi szerint legyenek az adatok csoportosítva
4. Összesítési beállítások (Sum, Avg, Min, Max)
5. Mi szerint legyen rendezve
6. Milyen elrendezésben jelenjenek meg a sorok?
7. Milyen legyen a jelentés stílusa
8. A jelentés neve

SORSZÁMOZOTT LISTÁK

Ha egy lista sorait szeretnénk beszámoztatni, akkor a törzsben meghatározott mezők elé egy beviteli mezőt helyezünk el, mezőkifejezésnek írjuk be, hogy „=1”, és a Tulajdonságok közül a **Futó összeget** állítsuk be attól függően, hogy *Folyamatosan* vagy *Csoportonként* akarjuk a sorokat beszámozni.

Példa egy sorszámozott, összefokozatos listára

Bérlista fiókonként				
Fiók		Név	Beosztás	Bér
Budapest				
	1	Kertész Béla	könyvelő	120 000
	2	Kiss Ágnes	titkárnő	40 000

	7	Szabó Zsuzsa	fiókvezető	150000

				795 000
Győr				
	1	Andor Iván	fiókvezető	140000
	2	Bakó Lajos	üzletkötő	60 000

	5	Németh István	üzletkötő	50000

				350000
...				
	<i>Végösszeg:</i>			1 515 000