



ADATBÁZIS–KEZELÉS HALADÓKNAK, AVAGY A HATÉKONY ADATBÁZIS–KEZELÉS

ESETFELVETÉS – MUNKAHELYZET

Ön egy autókereskedéssel foglalkozó vállalat egyik magyarországi telephelyén dolgozik ahol az autók adatainak nyilvántartását egy saját fejlesztésű adatbázis-kezelő programmal végzik.

Ez a szoftver idáig megfelelt az adatkezelési igényeknek, de a cég fejlődésének (több telephely megnyitása, új dolgozók felvétele, eladó autók számának növekedése) köszönhetően már nem felel meg az elvárásoknak. A program legnagyabb hátránya, hogy bár relációs adatmodellt használ, nem kezeli a szabványos SQL parancsokat, és a lekérdezési lehetőségek is korlátozottak.

A vállalat most vásárolt egy modern irodai programcsomagot, amely tartalmaz adatbáziskezelő rendszert is. Munkahelyi vezetőjétől következő feladatokat kapja:

- Oldja meg a jelenlegi adatok importálását a vállalat által megvásárolt irodai programcsomag részét képező adatbázis-kezelő szoftverbe!
- Ezt követően kihasználva a szoftver lehetőségeit készítsen látványos űrlapokat, jelentéseket valamint a vállalat igényeinek megfelelő hatékony lekérdezéseket! A lekérdezési igények a több telephelyből és a nagy mennyiségű adatból adódóan igen szerteágazóak, így a kereső lekérdezések mellet akciólekérdezéseket és SQL specifikus lekérdezéseket is meg kell valósítania!
- További feladata lesz még, hogy tartson belső képzést a telephelyek adminisztrátorainak számára, ahol ismerteti az új szoftver felületét és fő szolgáltatásait különös tekintettel az űrlapok, jelentések és lekérdezések készítésére!

SZAKMAI INFORMÁCIÓTARTALOM

FŐBB ADATMODELLEK

Az adatmodell egyértelműen meghatározza az adatbázis szerkezetét, magában foglalja az adatok típusát, kapcsolatát, a korlátozó feltételeket és az adatkezelési műveleteket.

A mai adatbázisokban négyféle logikai adatmodellt használunk: a hálós, a hierarchikus, az objektumorientált, illetve a relációs adatmodellt.

Hálós adatmodell: A hálós adatmodell szerkezetét gráffal adjuk meg. A gráfban a csúcspontok az egyedek, az élek pedig a kapcsolatok. Az egyedeket tulajdonságaikkal írjuk le, a kapcsolatokat mutatók segítségével adjuk meg.

Hierarchikus adatmodell: A hierarchikus adatmodell szerkezetét is gráffal adjuk meg, de a gráf egy fa. Az adatok alá-, és fölérendeltségi viszonyban állnak. Minden egyed csak egy ponton (egy szülő) kapcsolódhat.

Objektumorientált adatmodell: Az egyedeket objektumoknak nevezzük. Az objektum tulajdonságai az objektum állapotát (tulajdonságainak aktuális értékét) írják le. A tulajdonságok szerepe azonos a relációs adatmodellével, de lényeges különbség, hogy objektum orientált adatmodellben egy tulajdonság bármi lehet, akár egy másik objektum.

Relációs adatmodell: Az egyed megadása táblával (a táblázat oszlopai az egyed tulajdonságai, sorai az egyed konkrét értékei, a mezők a tulajdonságok konkrét értékei) történik.

A PC-k en a legelterjedtebb a relációs adatmodell. Ezt az adatmodellt használják legismertebb adatbázis-kezelő szoftverek (Access, MySQL, Paradox, Dbase, Oracle, OpenOffice.org Base)

RELÁCIÓS ADATBÁZIS-KEZELÉS ALAPJAI

A relációs adatmodellben az adatokat egymással logikai kapcsolatban álló táblákba rendszerezzük.

Egy tábla oszlopainak és sorainak a következő feltételeknek kell megfelelniük:

- • minden oszlopnak egyértelmű neve van,
- • minden sorban ugyanazok az oszlopok vannak,
- • az oszlopokban található adatok meghatározott értéket vehetnek fel,
- • az oszlopok soronként csak egy értéket vehetnek fel,
- a táblát a neve egyértelműen azonosítja.

A **tábla** a logikailag összetartozó adatokat foglalja össze. A tábla oszlopokból és sorokból áll, amelyeket mezőknek, illetve rekordoknak nevezünk.

	Vev	ők			2		
	Az	onos 👻	Cég	•	Utónév	•	Vezetéknév 🗸
6	÷	1	A cég		Anna		Bedecs
6	Ŧ	2	B vállalat		Péter		Budai
6	÷	3	C vállalat		Péter		Fischer

1 Rekord vagy sor

2 Mező vagy oszlop

1. ábra. Tábla felépítése

A **rekord** az adatbázis egy sora. Egy rekordban tároljuk az egymással összefüggő adatokat.

A mező az adatbázis egy oszlopa, amelyben az egyedek tulajdonságértékeit tároljuk.

Az elemi adatok a tábla celláiban szereplő értékek, amelyek az egyed konkrét tulajdonságai.

Az **egyed** az, amit le akarunk írni, amelynek az adatait tároljuk és gyűjtjük az adatbázisban. Az egyedet idegen szóval entitásnak nevezzük. Egyednek tekinthetünk például egy autó.

Az attribútum (vagyis tulajdonság) az egyed valamely jellemzője.

Az egyed az attribútumok összességével jellemezhető. Egy autó egy jellemzője lehet például a rendszáma.

Az egyedre vonatkozóan megadott tulajdonságok összességét **egyedtípusnak** nevezzük. Egy autó leírható például a típusával, rendszámával, gyártási évével, színével, hengerűrtartalmával együttesen.

Az egyedre vonatkozóan megadott konkrét tulajdonságokat **egyedelőfordulásnak** nevezzük. Egy egyed-előfordulás például egy Lada 2107, amelynek rendszáma ABC-123, 1987-ben gyártották, piros, és 1,200 cm³-es.

Elsődleges kulcs: a tábla rekordjainak egyértelmű azonosítója, értéke egyedi.

Idegen kulcs: olyan azonosító, amelynek segítségével egy másik tábla elsődleges kulcsára hivatkozhatunk. (Kapcsolatokban használjuk.)

Index: A táblákban való keresés és a sorba rendezés gyorsítására alkalmas eszköz.

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

 Egy-egy (1:1) kapcsolat: az egyik tábla egy eleméhez a másik tábla pontosan egy eleme kapcsolódik. Pl.: házasság férfi-nő (Magyarországon)

- Egy-több (1:N) kapcsolat: az egyik tábla egy eleméhez a másik tábla több eleme is tartozhat. Pl.: anya-gyerek
- Több-több (N:M) kapcsolat: bármely tábla elemeihez a másik tábla tetszőleges számú eleme tartozhat. Pl.: szerző-könyv

A mai modern relációs adatbázis-kezelő rendszerek (pl.: MS ACCES) egy fájlban különböző objektumokat tárolnak. Ezek közül a legfontosabbak:

Tábla: Az adatokat tartalmazza táblázatos formában. A tábla oszlopokból és sorokból áll, amelyeket mezőknek, illetve rekordoknak nevezünk.

Lekérdezés: Tábla adatainak kigyűjtésére szolgál, de alkalmas törlésre, módosításra és számítások elvégzésére is. Forrása tábla vagy lekérdezés lehet.

Űrlap: az adatok kényelmes és látványos formában való bevitelére és módosítására szolgál.

Jelentés: elsődleges feladata a látványos nyomtatás biztosítása, de alkalmas csoportosításokra és összegzésekre is. Forrása tábla vagy lekérdezés lehet.

Mivel az adatbázis-kezelés alapjait egy másik modulfüzet tartalmazza az egyes objektumoknál csak a magasabb szintű, "haladó" műveletekre térek ki.

TÁBLÁK

Táblát létrehozhatunk új adatbázis létrehozásával, tábla meglévő adatbázisba történő beillesztésével, vagy külső adatforrás importálásával vagy csatolásával. Az információk importálása során másolatot készítünk az külső adatokról az aktuális adatbázis új táblájában. Ezzel szemben az adatok csatolásakor egy csatolt táblát hozunk létre az adatbázisban, amely élő kapcsolatban áll a máshol tárolt adatokkal Ha egy másik alkalmazás módosítja az eredeti forrás adatait, a változások megjelennek a csatolt táblában is.

Az importálás művelettel objektumokat vagy adatokat másolhatunk át egy másik adatbázisból vagy listából. Importálhatunk adatbázis fájlokból (Access, FoxPro, dBASE, Paradox), illetve Excel táblázatokból és szöveges dokumentumokból is. Ez utóbbi lehetőség akkor jöhet jól, ha a saját fejlesztésű nyilvántartó program adatait exportálhatjuk szövegfájlba.

Kezdőlap Létreh	ozás <mark>Külső adatok</mark> Adatbáziseszközök
Mentett importálások	> Szövegfáji ≫ XML-fáji → Egyebek → Excel Mentett exportálások
Importálás	<u>O</u> DBC adatbázis
Minden tábla	ODBC adatbázis importálása vagy hivatkozása (például SQL-kiszolgálóról)
Tábla1	HTML-dokumentum
🛄 Tábla1 : tábla	HTML-dokumentum importálása vagy hivatkozása
	Outlook mappa Outlook-mappa importálása vagy hivatkozása
	dB7 dB7 dB7 hivatkozása
	Paradox fájl Paradox fájl importálása vagy hivatkozása
	Lotus 1-2-3 fájl Lotus 1-2-3 fájl importálása

2. ábra. Importálási lehetőségek (MS ACCESS 2007)

Ha szövegfájlt szeretnénk importálni vagy csatolni, a fájl tartalmát úgy kell rendezni, hogy az importálást és csatolást végző varázslók fel tudják bontani a tartalmat rekordokra (sorokra), valamint az egyes rekordokat mezőkre (oszlopokra). Két ilyen típusú szövegfájl létezik: tagolt fájlok vagy rögzített szélességű fájlok.

J Szovegimp	ortáló varázsló							Szövegimp	ortáló varázsló						
Milyen hatá dokumentur Válassza I O Tabulá	roló választja el a me nra. ti a mezőket elválasz tegr <u>Pontx</u> sor tartalmazza a me	zőket? Vál tó határoló isvessző zőneveket	assza ki a n	negfelelő hatá Vessző	árolót; a r O Szó	nintán megláthatja jöz OEs Szövegjejölő:	a, milyen hatással van ez a gyéb:	Megadhatja szükséges in Mezőbeállít Mezőnév: Indexelt:	az importált mezők formációkat. isok nev Nem	nöz tartozó	információ	kat. Válassza l lattípus: Sz Ne importálja	ki a mezi töveg a mezőt	iket, majd a Mezőb (Gihagyja)	eállítások részen megadh
	1	_			1 1 1 1 1 1	and some the set of the set of sectors.	the second second second					-	1		
anulokod	nev	mat	ango1	2nyelv	tesi	csalaaszam	cesoszem	tanulokod	nev	mat	ango 1	2nve lv	tesi	loss ladszam	tesoszam
anulokod	nev Avon Mór	mat	angol 4. Kis	2nyelv német	F	4	1	tanulokod	nev Avon Mór	mat	angol 4. Kis	2nyelv német	tesi	csaladszam 4	tesoszam 1
anulokod	nev Avon Mór Bakt Ernő	mat alfa alfa	angol 4. Kis 4. Kis	2nyelv német spanyol	F	<u>csaladszam</u> 4 4	1	tanulokod	nev Avon Mór Bakt Ernő	mat alfa	angol 4. Kis 4. Kis	2nyelv német spanvol	tesi F	csaladszam 4 4	tesoszam 1 1
anulokod	nev Avon Mór Bakt Ernő Bal Margó Bakra Pál	mat alfa alfa alfa	angol 4. Kis 4. Kis 4. Kis 4. Kis	2nyelv német spanyol spanyol	r F L	<u>csaiadszam</u> 4 2 4	1 1 0	tanulokod 1 2 3	nev Avon Mór Bakt Ernő Bal Margó	mat alfa alfa alfa	angol 4. Kis 4. Kis 4. Kis	2nyelv német spanyol spanyol	tesi F F L	csaladszam 4 4 2	tesoszam 1 1 0

3. ábra. Szövegfájl importálása

Az importálás után mindenképp célszerű a tábla Tervező nézetében ellenőrizni a mezőneveket, adattípusokat és a mezőtulajdonságokat.

LEKÉRDEZÉSEK

A lekérdezés az adatokra vonatkozó utasításokból áll. A lekérdezés végrehajtásával ezeket az utasításokat hajtjuk végre. Az eredmények visszaadása mellett – amelyeket rendezhetünk, csoportosíthatunk, szűrhetünk – a lekérdezés adatokat hozhat létre, másolhat, törölhet vagy módosíthat.

1. Lekérdezések típusai

A következő fontosabb lekérdezéstípusokat különböztetjük meg:

- 1. . **Visszakereső lekérdezés**: adatok összegyűjtésére, kikeresésére és bemutatására, valamint űrlapok és jelentések adatokkal való ellátására használjuk. A lekérdezés eredménye egy adathalmaz, amely üres is lehet.
 - a) Választó lekérdezés: Választó lekérdezés minden olyan lekérdezés, amely adatokat gyűjt ki, illetve kér le megjelenítésre.
 - b) Kereszttáblás lekérdezés: kereszttáblás lekérdezésekkel összesíteni lehet az értékeket, majd két szempont szerint csoportosítani - az egyik szempont az adatlap oldalán függőlegesen, a másik pedig az adatlap tetején vízszintesen jelenik meg.
- 2. Akció lekérdezés: Műveleteket (akciókat) végeznek az adatokkal:
 - a) **Táblakészítő lekérdezés**: Új táblát, majd abban rekordokat (sorokat) meglévő táblából rekordmásolással létrehozó lekérdezés.
 - b) **Frissítő lekérdezés**: a megadott keresési feltételeknek megfelelő rekordkészletet változtatja meg.
 - c) **Törlő lekérdezés**: egy vagy több táblából törli a megadott feltételeknek megfelelő sorokat (rekordokat).
 - d) **Hozzáfűző lekérdezés**: lekérdezés eredményének rekordjait egy meglévő tábla végéhez illeszti.
- 3. **Paraméteres lekérdezés**: a fentiek közül bármelyik típusú lehet. Végrehajtásuk alkalmával egy értéket (paraméter) kérnek be a felhasználótól. Amikor megadjuk ezt az értéket, a lekérdezés mezőfeltételként alkalmazza. A lekérdezés tervétől függ, hogy melyik mezőre alkalmazza feltételként a megadott értéket. Ha nem adunk meg értéket, azt a lekérdezés üres karaktersorként értékeli.



4. ábra. Lekérdezések fő típusai (MS ACCESS 2007)

Lekérdezések létrehozása haladóknak

A haladó felhasználók a lekérdezések létrehozására két féle módszer közül választhatnak

 Tervező nézetben: QBE (Query By Example=lekérdezés példa lapján) tervezőrács kitöltésével. Ez az egyszerűbb és gyakoribb módszer. SQL- nézetben SQL utasítások segítségével is létrehozhatjuk a lekérdezést (Ez a legtöbb szakértelmet igénylő módszer, de egyben a legnagyobb rugalmasságot is biztosítja. Előfordulhat olyan lekérdezési igény, amely csak SQL utasítások segítségével valósítható meg.) Az SQL nyelv utasításait külön rész tartalmazza a tananyagban.

1.1. Választó lekérdezések

1.1.1 Számított mezők létrehozása lekérdezésben:

Az adatbázisokat érdemes úgy tervezni, hogy ne tároljunk olyan adatokat, amelyek a többiből kiszámíthatók. Ezeket az adatokat a számított mezőket tartalmazó lekérdezések segítségével hozhatjuk létre. Ehhez a megfelelő képletet a tervezőrács egy üres mezőjébe kell begépelnünk. A képletben használt mezőneveket szögletes zárójelek közé kell írnunk.

Amennyiben a lekérdezés több olyan rekordforrást tartalmaz, amelyekben azonos mezőnevek is vannak, a mezők egyértelmű megkülönböztetéséhez a megfelelő mezőt tartalmazó tábla vagy lekérdezés nevét is meg kell adnunk a képletben. A rekordforrás nevét szögletes zárójelek között, a mezőnév elé kell írnunk, és egy felkiáltójellel kell a mező nevétől elválasztanunk: [Táblanév]![Mezőnév] Például: [Autok]![Ertek]

A számított mezőknek alaphelyzetben a Kif1, Kif2 stb. neveket adja a program. Az alapértelmezett név helyett tetszőleges saját nevet is megadhatunk. A számított mező nevét a kifejezés elé kell beírnunk, és kettősponttal kell elválasztanunk a kifejezéstől.

Például autókereskedésünk a hétvégi születésnapja alkalmából 10% kedvezményt adva akciós árakkal dolgozik 3 napig. Ezt az akciós árat célszerű lekérdezéssel kiszámítanunk. (Persze használható frissítő lekérdezés is, de az nem szerencsés, mert az ünnepi hétvége után ismételt frissítő lekérdezéssel kellene visszaállítani az eredeti árakat.)

Mező: Tábla:	Rendszám autok		Tip: aut	us ok	Szín autol		Évjárat autok	Akciós ár: [Érték]*0,	9	Erték autok	~
Megjelenítés: Feltétel:				V			V				
vagy:								Számítot	mező		
🔀 Rendszám	Tipus	Szín		Évjárat •	Akciós ár 🔹	Érték •					
CUC-888	Volga	Kék		2003	1800000	2 000 000 Ft					
CSIN-95	Volga	Fehér		1912	510300	567 000 Ft					
DCS-343	Moszkvics	Fehér		1967	8010000	8 900 000 Ft					
DMC-123	Trabant	Fehér		1946	1080000	1 200 000 Ft	Lekérdezé	és eredmény	e		
GGG-333	Volga	Fehér		1995	108000	120 000 Ft		ie ereanienij			
HDJ-001	Volga	Piros		1963	72000	80 000 Ft					
HOO-001	Volga	Fehér		2001	126000	140 000 Ft					

5. ábra. Számított mező használata lekérdezésben

A Kifejezésszerkesztő segítségével a bonyolultabb képleteket kevesebb gépeléssel, kényelmesebben hozhatjuk létre. A képletben felhasználhatjuk a az ismert matematikai műveleteket, logikai operátorokat, illetve az adatbázis-kezelő előre definiált függvényeit. A Kifejezésszerkesztő megjelenítéséhez álljunk a lekérdezés tervezőrács valamely üres oszlopának Mező vagy Feltétel rovatába, attól függően, hogy új mezőt vagy feltételt szeretnénk létrehozni. Ezután a helyi menü Szerkesztés menüpontjával tudjuk megjeleníteni a szerkesztőablakot.

Akciós ár: [Érték]*0,9]	OK Mégse
+ - / * 8 = > < <> And Or Not Like ()	Visszavonás Beillesztés Súgó

6. ábra. Kifejezésszerkesztő

1.1.2. Összesítő lekérdezések:

Az összesítő lekérdezés a választó lekérdezés egyik fajtája, amely az adatok szűrésén kívül lehetőséget nyújt a rekordok különböző szempontok szerinti csoportosítására, és a csoportosított értékekhez kapcsolódó számítások – például összegzés vagy átlagolás – elvégzésére. Összesítő lekérdezés készítéséhez hozzunk létre egy választó lekérdezést a szokott módon, majd kapcsoljuk be az eszköztár Összesítés gombját. A csoportosítási módot, illetve az összesítő függvényt a megjelenő Összesítés sor mezőiben adhatjuk meg.

Az összesítő lekérdezésben az alábbi statisztikai függvények és beállítások használhatók.

Függvény neve	Jelentése
Sum	A csoport összegét számolja ki.
Avg	A csoport átlagát számolja ki.
Min	A csoport legkisebb elemét adja meg.
Max	A csoport legnagyobb elemét adja meg.
Count	A csoportba tartozó elemek számát adja meg.
StDev	A csoport szórását számolja ki.
Var	A csoport varianciáját számolja ki. (Variancia: a szórás négyzete, statisztikai adatoknál az adatkiugrás kimutatására szolgál.)
First	Az első rekord adott mezőben levő értékét adja meg.

Last	Az utolsó rekord adott mezőben levő értékét adja meg.
Beállítás	Jelentése
Group by	A rekordok csoportosítására szolgál. A csoportosított rekordok más mezőinek értékeivel szükség szerint különféle számításokat is végezhetünk.
Expression	Olyan számított mező létrehozására használjuk, amely összesítő – például Sum, Min vagy Max – függvényt tartalmaz a kifejezésben.
Where	Azoknál a mezőknél válasszuk, amelyeket csak szűrőfeltétel megadására használunk, így ezek a mezők nem képeznek csoportosítási alapot és a lekérdezés eredményében sem jelennek meg. Ezeknél a mezőknél a Megjelenítés jelölőnégyzetnek mindig kikapcsolt állapotban kell lennie, ezért azt a Where elem kiválasztásakor a program automatikusan kikapcsolja.

A lenti példában olyan lekérdezést készítettünk, amely megadja az autókereskedés számára a típusonkénti darabszámot, átlagárat és összértéket is.

			-		_			
Mező:	Tipus	~	Rendszám		Érték		Érték	
Tábla:	autok		autok		autok		autok	
Összesítés:	Group	Ву	Count		Avg		Sum	
Rendezés: Megjelenítés:		V				V		
Feitetel: vagy:		Te	rvezőrács					
		Tipus 🔹	CountOfRer -	AvgOf	Érték 🔹		SumOfÉrték	•
		Moszkvics	9	: 080 44	4,44 Ft		27 724 00	00,00 Ft
Lekérdezés	s	Polski Fiat	6	. 043 33	3,33 Ft		6 260 00	00,00 Ft
eredménye	•	Trabant	7	1092 85	7,14 Ft		14 650 00	00,00 Ft
		Volga	9	640 77	7,78 Ft		5 767 00	00,00 Ft
		Wartburg	4	. 290 30	0,00 Ft		5 161 20	00,00 Ft

7. ábra. Összesítő lekérdezés (MS ACCESS 2007)

1.1.3 A kereszttáblás lekérdezés segítségével olyan összesítést készíthetünk, amely kettő vagy több kiválasztott mező adatai alapján összesíti egy harmadik mező adatait.

A kereszttáblás lekérdezések összesítik (az összesítő lekérdezéseknél használható valamelyik függvény segítségével) az értékeket, majd két tényhalmazba gyűjtik azokat- az egyik halmaz (a sorfejlécek halmaza) a függőleges oldal mentén, a másik (az oszlopfejlécek halmaza) pedig a tábla tetején helyezkedik el.

A lekérdezés típusának átállítása után a használni kívánt mezőket a szokott módon vehetjük fel a tervezőrácsba, majd az összesítő lekérdezésekhez hasonlóan az Összesítés sorban adjuk meg a csoportosítási szempontot vagy a használni kívánt összesítő függvényt.

Ez után a Kereszttábla sorban adhatjuk meg, hogy az egyes mezők az összesítés mely elemeként jelenjenek meg. Az Oszlopfejléc választása esetén az adatok mezőnevekként jelennek meg az oszlopok felett. A Sorfejléc választása esetén az adatok a sorok elején jelennek meg. Az Oszlop- és Sorfejléceknek mindig csoportosított adatoknak kell lenniük. Az Érték, az Oszlop- és Sorfejlécek metszéspontjainál, az általunk kiválasztott függvénnyel összegezve jelennek meg. Az Érték nem lehet csoportosított adat. Bonyolultabb összesítéseknél több Sorfejlécet is megadhatunk, de Oszlopfejléc és Érték mindig csak egy lehet.

A lenti példában az autók átlagos értékének meghatározása történik típus és szín szerint.

Mező:	Tipus	Sz	ín		É	rték	×	
Tábla:	autok	au	utok		a	utok		
Összesítés:	Group By	G	roup B	у	A	vg		
reszttábla:	Sorfejléc		Oszlopfejléc			Érték		
Rendezés:						1		
Feltétel:								
vagy:			1	- afraill-				
		kereszttabla)	,				
	NZ	Tipus 🗸	Fe	hér 👻	Ke	k 👻	Piros 🔹	
		Moszkvics	6 85	0 000 Ft	2 58	800 Ft	560 000 Ft	
		Polski Fiat	146	0 000 Ft	1 000	000 Ft	340 000 Ft	
		Trabant	2 20	0 000 Ft	1 250	000 Ft	2 250 000 Ft	
		and the second se		A LEAST AND A LOCAL OF	1111111111111111	100000000000000000000000000000000000000		
		Volga	50	6 750 Ft	1 600	000 Ft	180 000 Ft	

8. ábra. Kereszttáblás lekérdezés (MS ACCESS 2007))

1.2. Táblakészítő lekérdezés: Új táblát, majd abban rekordokat (sorokat) meglévő táblából rekordmásolással létrehozó lekérdezés. Alkalmas biztonsági másolat készítésére a tábláról, exportálás előtti adatkigyűjtésre, űrlapok, jelentések alapjainak elkészítésére.

Készítésékor a lekérdezés típusát át kell állítani az eszköztáron (4. ábra), majd meg kell adni a tábla nevét. Ezután a lekérdezést a választó lekérdezéshez hasonlóan készítjük el. A futtatás után létrejön az általunk megadott néven a tábla.

A következő példában olyan lekérdezést, készítettünk, amely Trabant nevű táblába másolja a Trabantok összes adatát.

	Tá	iblakészítő			? 🛛	
	1	Új tábla létrehozása			ОК	
•		Aktuális adatbázis Másik adatbázis: ájínév:		Tallózás	Mégse	
Mező: Tábla:	Rendszám autok	Tipus autok	Szín autok	Évjárat autok	Hengerürtartalom autok	Érték autok
Rendezés: Megjelenítés: Feltétel: vagy:		₹trabant	V			

9. ábra. Táblakészítő lekérdezés (MS ACCESS 2007)

1.3 Frissítő lekérdezés: a megadott keresési feltételeknek megfelelő rekordkészletet változtatja meg.

A lekérdezés típusának átállítása után a rekordok módosításának feltételeit a választó lekérdezéseknél használt módon határozhatjuk meg. Az egyes mezőkben elvégzendő változtatásokat a Módosítás sorban adhatjuk meg. Itt megadhatunk konkrét értékeket vagy tetszőleges képletet is.

Az alábbi példában az 1970 előtt gyártott autók értékét 20%-kal növeljük.

Mező: Érték	Évjárat	V
Tábla: autok	autok	
dosítás: [Érték]*1,2		
Feltétel:	<1970	
Feltétel: vagy:	<1970	

10. ábra. Frissítő lekérdezés (MS ACCESS 2007)

1.4 Törlő lekérdezés: egy vagy több táblából törli a megadott feltételeknek megfelelő sorokat (rekordokat).

A lekérdezés típusának átállítása után a rekordok törlésének feltételeit a választó lekérdezéseknél használt módon határozhatjuk meg. A lekérdezés bezárása után, minden megnyitáskor a lekérdezés újból végrehajtásra kerül.

A törlő lekérdezés lefuttatása után a törölt rekordok a Visszavonás parancs segítségével már nem állíthatók vissza.

Az alábbi példában az 1970 és 1975 között gyártott piros autók kerülnek törlésre.

Mező: Szín	Évjárat
Tábla: autok	autok
Törlés: Where	Where
eltétel: "piros" vagy:	>=1970 And <=1975

11. ábra. Törlő lekérdezés (MS ACCESS 2007)

1.5 Hozzáfűző lekérdezés: lekérdezés eredményének rekordjait egy meglévő tábla végéhez illeszti.

A lekérdezés típusának átállítása után meg kell adni, hogy mely táblába kerüljenek az adatok, majd a rekordok hozzáfűzésének feltételeit a választó lekérdezéseknél használt módon határozhatjuk meg.

A Hozzáfűzés sorban kiválaszthatjuk, hogy az egyes mezők tartalmát a céltábla melyik mezőjébe írja az adatbázis-kezelő a hozzáfűzés során. A táblához hozzáfűzött adatoknak a tábla mezőivel kompatibilis adattípusokat kell tartalmazniuk. (A legegyszerűbb, ha a két tábla azonos szerkezetű.)

Az alábbi példában a Wartburgok adatait fűzzük a Trabant táblához..

Mező:	Rendszám	Tipus	Szín	Évjárat	Hengerürtartalom	Érték
Tábla:	autok	autok	autok	autok	autok	autok
Rendezes: Hozzáfűzés:	Rendszám	Tipus	Szín	Évjárat	Hengerürtartalom	Érték
Feltétel:	-	"Wartburg"				
vagy:			1			

12. ábra. Hozzáfűző lekérdezés (MS ACCESS 2007)

1.6. Paraméteres lekérdezés:

Végrehajtásuk alkalmával egy értéket (paraméter) kérnek be a felhasználótól. Amikor megadjuk ezt az értéket, a lekérdezés mezőfeltételként alkalmazza.

A lekérdezés tervétől függ, hogy melyik mezőre alkalmazza feltételként a megadott értéket. Ha nem adunk meg értéket, azt a lekérdezés üres karaktersorként értékeli.

Ebben az esetben a felhasználó beavatkozásától is függ az eredmény. Paraméteres lekérdezés létrehozásához hozzunk létre egy lekérdezést a szokott módon, majd a tervezőrácsba írjuk be a szükséges képleteket és feltételeket. Azon adatoknál, amiket a felhasználótól szeretnénk bekérni, a feltétel helyett [] szögletes zárójelek között írjunk be egy üzenetet vagy a paraméter nevét. Az üzenetnek nem szabad megegyeznie a mező nevével.

A paraméteres lekérdezés egy párbeszéd panelen, a szögletes zárójelek között megadott szöveggel kéri be a szükséges adatokat.

Az alábbi példában az autó típusát kérjük be paraméterként a felhasználótól majd a típus felhasználó általi beírása után megjelennek az autók adatai.

		Paraméter me	gadása ?			
		Milyen típusú aut	ót keres? Mégse			
Mező:	Rendszám	Tipus	Szín	Évjárat	Hengerürtartalom	Érték
Mező: Tábla:	Rendszám autok	Tipus autok	Szín autok	Évjárat autok	Hengerürtartalom autok	Érték autok
Mező: Tábla: Rendezés: egjelenítés:	Rendszám autok	Tipus autok	Szín autok	Évjárat autok	Hengerürtartalom autok	Érték autok

13. ábra. Paraméteres lekérdezés (MS ACCESS 2007)

ŰRLAPOK

Az űrlap táblák vagy lekérdezések adatainak bevitelére, szerkesztésére és megjelenítésére használható adatbázis-objektum. Űrlapokkal szabályozhatjuk az adatokhoz való hozzáférést is, így meghatározhatjuk, hogy mely mezők vagy adatsorok jelenjenek meg.

Az űrlapok készülhetnek az adatbázis tábláiból vagy lekérdezéseiből. Attól függően, hogy milyen típusú űrlapra van szükségünk, választhatunk űrlaptípusok közül.

Az űrlapok létrehozásának legleterjedtebb módja – még haladó felhasználók számára is – a varázsló használata, míg az egyedi igények szerinti módosításokat utólagosan az Űrlap Tervező nézetében tehetjük meg.

Űrlap létrehozás MS ACCESS 2007-ben: Létrehozás lap/Űrlapok csoportban történhet.

Az űrlapon található beviteli mezők és más elemek elhelyezkedésének és formátumának módosításához nyissuk meg az űrlapot Tervező nézetben. Az egyes vezérlőelemeket az egérrel az elemre kattintva is kijelölhetjük. Több vezérlőelem együttes kijelöléséhez kattintsunk az első vezérlőelemre, majd a SHIFT gomb nyomva tartása mellett folytassuk a többi elem kijelölését. Ezután az eszköztár ikonjai segítéségével formázhatjuk az űrlapot.

Cn	19- 14	- + Adat	bázis1 : adatbáz	is (Access 2007) - M	Jelentés-elre	ndezési eszközök				_ = ×
9	Kezdőlap	Létrehozás	Külső adatok	Adatbáziseszközök	Formátum	Elrendezés Old	lalbeállítás			0
	Calibri		A - 🔢		24	Σ Összesítés *	HT	≡-		20000
Same Al Canada	FDA	11 -			Consultation	1 Részletek elrejtése		-		
Wezet	토 폰 크	1	Feiteteles	% 000 ,66 4,6	és rendezés		*	1 -	Emblema 🔂 🚄 * felvétele	formázás *
Nézetek		Betütípus	6	Formázás	Csoporto	osítás és összesítés	Rácsvon	alak	Vezérlők	Automatikus form

14. ábra. Űrlap tervező nézetének eszköztára (MS ACCESS 2007)

Látványossá és egyedivé tehetjük az űrlapot vezérlők elhelyezésével. Ezeket szintén a fenti eszköztárról választhatjuk ki. (Például a vállalkozás nevének elhelyezése történhet az űrlapon címke segítségével, vagy akár egyéni rekordléptető gombokat is elhelyezhetünk az űrlapon. Stb.)

JELENTÉSEK

A jelentés az adatok nyomtatott formában történő megjelenítésének hatékony módja, ugyanakkor segítségükkel csoportosításokat, rendezéseket és összegzéseket is végrehajthatunk.

Mivel a jelentés minden egyes elemének méretét és kinézetét magunk állíthatjuk be, az adatokat a kívánt formátumban jeleníthetjük meg.

A jelentések alapjául szolgáló adatok származhatnak táblából, illetve lekérdezésekből. Ha több táblából származó adatokat szeretnénk megjeleníteni, célszerű ezeket egy lekérdezésben összefoglalni, és erre alapozni a jelentést. Ha a jelentés alapjául szolgáló tábla vagy lekérdezés adatait módosítjuk, a módosítás megjelenik a jelentésben is.

Jelentéseket táblák vagy lekérdezések alapján hozhatunk létre. A jelentések készítésének legegyszerűbb módja az AutoJelentések vagy a Jelentés varázsló használata. Az AutoJelentés vagy a Jelentés varázsló funkcióval készített jelentések azonnal használhatók, de felhasználhatóak egyedi tervezésű jelentések kiindulási alapjaként is. Még a profi felhasználok is ez utóbbi módszert választják, azaz a kész jelentést az űrlaphoz hasonló módon Tervező nézetben utólagosan módosítják az egyedi igények szerint.

	🛃 47 - (°4 -) ∓ Ad	atbázis1 : adatbáz	tis (Access 2007) - M	Jelentés-eire	ndezési eszközök			_ = X
9	Kezdőlap Létrehozás	Külső adatok	Adatbáziseszközök	Formátum	Elrendezés Old	albeállítás		0
	Calibri F D A 11	· <u>A</u> ·	•		∑ Összesítés • ∰ Részletek elrejtése			
Nézet	E = = 🧭	Feltételes	🕎 % 000 👯 400	Csoportositás és rendezés		Rácsvonalak	Embléma 🔂 🚄 * Mezők	Automatikus formázás *
Nézetek	Betütip	us	Formázás	Csoporto	osítás és összesítés	Rácsvonalak	Vezérlők	Automatikus form

15. ábra. Jelentéstervező eszköztára (MS ACCESS 2007)

A Jelentés varázsló talán leghasznosabb funkciója a csoportosítási és összegzési beállítások lehetősége. A lenti példában célunk az, hogy kapjuk meg jelentésben az autók típusonkénti átlagárát és összértékét.

Jelentés varázsló		Összesítési beállítások	
Szeretne hozzáadni csoportszinteket? rendszen szin evjarat hengerurtart ár Prioritás *	tipus rendszam, szin, evjarat, hengerurtart, ár	Milyen összesítési értékeket szeretne számítani? Mező Össz Átl Min Max evjarat Image:	OK Mégse Megjelenítés ③ Iörzs és Összesítés ③ Csak Összesítés Az összegek százalékának számítása

16. ábra. Csoportosítás és összegzés jelentésben

SQL NYELV

Az SQL (Structured Query Language) szabványosított strukturált lekérdezőnyelv, mely relációs adatmodell alapján felépülő adatbázisok kezelésére képes.

(A szabványnak kötelező, ajánlott, és szabadon választott részei vannak, így például van némi eltérés például az ACCES és az ORACLE SQL között, a példákban az ACCES SQL-t használtam és csak a legfontosabb utasításokra tértem ki. Az ACCES SQL utasításai a lekrédezések SQL nézetében tekinthetőek meg és szerkeszthetőek.)

1. Az SQL nyelv utasításainak főbb csoportjai:

- DDL: adatdefiníciós nyelv (Data Definition Language), adatbázis és a szerkezeti elemek kialakítása:
 - CREATE, ALTER, DROP
- DML: adatmanipulációs nyelv (Data Manipulation Language), adatok karbantartása (bevitele, módosítása, törlése), lekérdezése:
 - INSERT, UPDATE, DELETE, SELECT

(A SELECT utasítást némely szakirodalom külön csoportba sorolja és DQL (Data Query Language) lekérdező nyelvnek is nevezi. A fenti parancsokat a továbbiakban részletezem ezért jelentésük nem szerepel itt.)

DCL: adatvezérlő nyelv (Data Control Language),

 tranzakció kezelése: COMMIT (Véglegesítés: rögzíti a tranzakció folyamán végrehajtott adatmódosításokat az adatbázisban), ROLLBACK Visszagörgetés: visszaállítja a tranzakció megkezdésekor (a legutolsó COMMIT-nál) rögzített állapotot az adatbázisban, így a tranzakcióhoz tartozó utasítások egyikének sem érvényesül a hatása. SAVEPOINT (mentési_pont;

Lehetővé teszi, hogy a tranzakció közben olyan pontot képezzünk, amely pontig részlegesen visszagörgethetjük a tranzakciót)

• adatvédelem, felhasználói hozzáférés szabályozása:

 GRANT (Bizonyos jogokat biztosít meglévő felhasználó vagy csoport számára), REVOKE (Bizonyos jogokat von vissza meglévő felhasználótól vagy csoporttól.)

1.1 DDL (adatdefiníciós) parancsok:

1.1.1 Tábla létrehozása: CREATE TABLE tábla_név (mezőnév1 típus, [DEFAULT kifejezés][, CONSTRAINT megszorítás_név megszorítás_típus], mezőnév2 típus [DEFAULT kifejezés] [, CONSTRAINT megszorítás_név megszorítás_típus],...);

<u>Alapértelmezett érték:</u> Alapértelmezett értéket veszi fel a mező, ha nem viszünk be más értéket. Egyébként az üresen hagyott mező NULL értéket vesz fel. Megadása: DEFAULT alapértelmezett_érték

<u>Megszorítások:</u>

Oszloponként kell megadni a CONSTRAINT kulcsszó után. Maga a CONSTRAINT kulcsszó minden mezőnél újra kell, hogy szerepeljen.

- Elsődleges kulcs (ha egy oszlopos a kulcs), CONSTRAINT megszorítás_név PRIMARY KEY
- Nem lehet üres CONSTRAINT megszorítás_név NOT NULL

Például az alábbi utasítás létrehoz egy autoadat nevű táblát rendszam, tipus, szin, evjarat, hengerurtart és ár nevű mezőkkel, a megfelelő adattípusokkal, legfeljebb 7 karakteres rendszam elsődleges kulccsal:

CREATE TABLE autoadat(rendszam TEXT (7) CONSTRAINT kulcs PRIMARY KEY, tipus TEXT, szin TEXT, evjarat INTEGER, hengerurtart INTEGER, ár INTEGER);

	Mezőnév	Adattípus
81	rendszam	Szöveg
	tipus	Szöveg
	szin	Szöveg
	evjarat	Szám
	hengerurtart	Szám
	ár	Szám

17. ábra. A fenti CREATE TABLE parancs eredményeként létrejött tábla Tervező nézetben

Főbb adattípusok ACCESS SQL-ben:

Adattípus	Leírás
BIT	Yes és No értékeket, valamint mindig két lehetséges

	érték egyikét tartalmazó mezők.
DATETIME	A 100. és 9999. évek közé eső dátum- vagy időpontérték.
REAL	Egyszeres pontosságú lebegőpontos szám
FLOAT	Dupla pontosságú lebegőpontos szám
INTEGER	Nagyobb egész szám
TEXT	Nulla és 2,14 GB közötti terjedelmű.
CHARACTER	Nulla és 255 karakter közötti terjedelmű.

1.1.2 Tábla törlése: DROP TABLE táblanév;

1.1.3 Tábla szerkezetének módosítása: ALTER TABLE

Mező hozzáadása: ALTER TABLE táblanév ADD mezőnév

Például a tulajdonos nevű mező hozzáadása az autoadat táblához: *ALTER TABLE autoadat ADD tulajdonos TEXT;*

Mező törlése: ALTER TABLE táblanév DROP mezőnév

Például a szín nevű mező törlése az autoadat táblából: ALTER TABLE autoadat DROP szin;

<u>Mező adattípusának megváltoztatása</u>: ALTER TABLE táblanév ALTER COLUMN mezőnév új_adattípus

Például ha az ár nem csak egész szám lehet, akkor az adattípus módosítása: ALTER TABLE autoadat ALTER COLUMN ár REAL;

1.2 DML (adatmanipuláló) parancsok:

1.2.1 Rekordok bevitele táblába: INSERT INTO táblanév [(oszlopnév-lista)] VALUES (értéklista);

Ha minden oszlopba viszünk be adatot, akkor nem kell az oszlopneveket felsorolni. Ha valamelyik értéket nem akarjuk megadni, akkor két vessző között üresen kell hagyni, vagy meg kell adni az oszlop neveket.

Például egy rekord bevitele az autoadat táblába: INSERT INTO autoadat (rendszam, tipus, Szin, evjarat, Hengerurtart, ár) VALUES ("ABC-123", "LADA", "piros", 1989,1200, 80000);

1.2.2 Mezők értékeinek módosítása: UPDATE táblanév SET oszlopnév=kifejezés [,oszlopnév=kifejezés] [WHERE feltétel];

Például az autoadat táblában növeljük az 1970 előtti autók árát 20 %-al: UPDATE autoadat SET ár = [ár]*1.2 WHERE (((evjarat)<1970));

1.2.3 A tábla rekordjainak törlése: DELETE FROM tábla_név WHERE feltétel;

Például az autoadat táblában töröljük az 1975 előtti autók adatait: DELETE FROM autoadat WHERE (((evjarat)<1975));

1.2.4 Lekérdezés végrehajtása: SELECT

Általános formája a következő: SELECT [DISTINCT] mezőnevek megadása FROM táblanév(-ek) [WHERE ...] feltételek (sorok kiválasztása) [GROUP BY...]csoportosítás [HAVING...] csoportok közötti válogatás [ORDER BY...] eredménysorok rendezése;

DISTINCT: csak az egymástól különböző sorokat adja vissza

Például a típusok lekérdezése az autoadat táblából, úgy, hogy egy típus csak egyszer látszik: SELECT DISTINCT tipus FROM autoadat;

WHERE alparancs: a szelekciót valósítja meg. Általános alakja: WHERE feltétel

A parancs hatására az eredménytáblában a SELECT után felsorolt oszlopokba azoknak a soroknak az értékei kerülnek, amelyekre teljesül a feltétel. A feltételre vonatkozó fontosabb szabályokat az alábbi táblázattal foglalhatjuk össze:

A feltétel típusa:	A feltétel alakja:	Az operátor:
Egyszerű összehasonlítás relációs operátorok	oszlopnév operátor kifejezés	=; != ;< ;> ;<>; >=; <=
Összehasonlítás egy halmaz elemeivel	oszlopnév operátor halmazdef.	BETWEEN szám1 AND szám2 (kif1 és kif2 közé esés (zárt intervallum)); IN (lista) (Adott értékek listáján egyezés keresése);;LIKE karakterminta (Karakterminta – egyezés keresése);
Összehasonlítás NULL értékkel	oszlopnév IS NULL	IS NULL (üres–e)
Logikai kereséséi feltétel	feltétel1 operátor feltétel2	AND (és); OR; (vagy); NOT(nem)

Az ORDER BY alparancs az eredménytábla rendezésére szolgál

Például az autoadat táblából lekérdezzük a fehér Volgák összes adatát ár szerint növekvő sorrendben: *SELECT * FROM autoadat WHERE ((([tipus])="volga") AND (([szin])="fehér")) ORDER BY ár;*

.A rendezés alapértelmezésben növekvő, DESC hatására csökkenő lesz a sorrend.

Például az autoadat táblából lekérdezzük az 1 000 000-nál olcsóbb Opelek rendszámait és árait ár szerint csökkenő sorrendben: *SELECT rendszam, ár FROM autoadat WHERE (([tipus])="opel") AND (([ár]<100000)) ORDER BY ár DESC;*

A GROUP BY alparancs alakja: GROUP BY oszlopnév [,oszlopnév] [HAVING feltétel]

A parancs hatására a megadott oszlop azonos értékei szerint csoportosítja a rekordokat. Amennyiben szerepel a HAVING, az azt jelenti, hogy a GROUP BY által kialakított eredménytáblából kiválasztja azon sorokat, amelyek eleget tesznek a HAVING utáni feltételnek.

A létrejövő csoportokra különféle beépített, úgynevezett csoportfüggvények használhatók: AVG(), MIN(), MAX(), COUNT(), SUM(), stb. (Lásd az összesítő lekérdezéseknél megadott táblázatot.

Például az autoadat táblából lekérdezzük a típusonkénti darabszámot és az átlagárat: SELECT Tipus, Count(Rendszam) AS darabszám, Avg(Ár) AS átlagár FROM autoadat GROUP BY Tipus;

	Betűtípus		Rich Text
Adatlap nézet	osszesitessql		
Kimutatás pázat	Tipus 👻	darabszám 👻	átlagár 👻
<u>Aimutatas nezet</u>	LADA	1	80000
_ 	Moszkvics	8	3671350
Kimutatás <u>d</u> iagram nézet	Opel	3	1466666,66666667
	Polski Fiat	6	1043333,33333333
SQL SQL néz <u>e</u> t	Trabant	6	2148333,33333333
	Volga	9	660266,666666667
Tervező nézet	Wartburg	3	575066,666666667
SELECT Tipus, Count(Re FROM autoadat GROUP BY Tipus;	ndszam) AS dara	abszám, Avg(Ár) AS átlagár

18. ábra. A fenti csoportosító és összesítő SQL parancs eredménye Adatlap illetve SQL nézetben

Például a fenti lekérdezésből csak azon típusoknál íratjuk ki az átlagárat, amelyekből öt darabnál több autónk van: SELECT Tipus, Count(Rendszam) AS darabszám, Avg(Ár) AS átlagár FROM autoadat GROUP BY Tipus HAVING COUNT(rendszam)>5;

Számított mező SQL-ben:

Például akcíós ár néven új számított mező létrehozása az eredeti árból 10% kedvezményt adva: *SELECT Rendszam, Tipus, Szin, Evjarat, [ár]*0.9 AS [Akciós ár], ár FROM autoadat;*

<u>Egymásba ágyazott SELECT:</u> az SQL nyelvben megengedett, hogy egy SELECT utasításban további SELECT-ek (allekérdezések) is előforduljanak. A belső SELECT önmagában is kiértékelhető és belülről kifelé haladva lesz feldolgozva. A kiértékelés menete: a belső SELECT kiértékelődik és egy, vagy több sort vagy oszlopértéket átad a külső SELECT-nek, a külső SELECT ezen értékek alapján összeállítja az eredményt.

Például az autoadat táblából lekérdezzük az átlagár feletti autók összes adatát: SELECT * FROM autoadat WHERE ár > (SELECT AVG(ár) FROM autoadat);

ANGOL SZAKMAI SZÓJEGYZÉK A ADATBÁZIS-KEZELÉS TÉMAKÖRÉBŐL

Az alábbi táblázat azokat az angol szavakat és magyar megfelelőiket tartalmazza, amelyekkel adatbázis-kezelés során leggyakrabban találkozhatunk:

Angol	Magyar	Angol	Magyar
Action Queries	Akció Lekérdezések	Multiple Table Queries	Többtáblás lekérdezések
Append Queries	Hozzáfűző lekérdezések	Normalization	Normalizálás
AutoNumber	Számláló	Primary Key	Elsődleges kulcs
AutoReports	Autojelentés	Query	Lekérdezés
Columns	Oszlopok	Query Design View	Lekérdezés Tervező nézet
Criteria Row	Feltétel sor	Referential Integrity	Hivatkozási integritás
Data	Adat	Relational Database	Relációs adatbázis
Data Types	Adattípusok	Relationships	Kapcsolatok
Database	Adatbázis	Report	Jelentés
Design Grid	Tervezőrács	Report Design View	Jelentést Tervező nézet
Design View	Tervező nézet	Report Sections	Jelentés szakaszok
Expression	Kifejezés	Report Wizard	Jelentés varázsló
Field	Mező	Run	Futtatás
Field Properties	Mező tulajdonságai	Select Query	Választó lekérdezés
Filters	Szűrők	Sort	Rendezés
Form Wizard	Űrlap varázsló	Table	Tábla
Input Mask	Beviteli maszk	Update Queries	Frissítő lekérdezések

Összefoglalás

- A régi adatbázis-kezelő szoftverből történő importálás során tisztában kell lennie relációs adatbázis-kezelés alapjaival.
- A vállalat adatkezelési igényeinek kielégítése érdekében a modern adatbázis-kezelő szoftver lekérdezési, valamint űrlap- és jelentéskészítési funkcióit célszerű használnia.

 A különböző típusú lekérdezések létrehozásakor a tervezőrács használata mellett speciális esetekben az SQL nyelv használatára is szüksége lesz, így a főbb utasításokat szintén használnia kell.

TANULÁSIRÁNYÍTÓ

Végezze el az alábbi tevékenységeket!

A feladatok megoldása során használhatja a programok súgóját és a szakmai információtartalmat is.

1. Készítse el az Ön által használt adatbázis-kezelő programmal, a példaként bemutatott autoadat táblát az alábbi minta alapján!

-	udioudut
	Mezőnév
8	rendszam
	tipus
	szin
	evjarat
	hengerurtart
	ár

Töltse fel legalább húsz rekorddal a táblát!

Hajtsa végre a szakmai információtartalomban példaként leírt lekérdezéseket (akciós ár kiszámítása, összesítő lekérdezés, kereszttáblás lekérdezés, táblakészítő lekérdezés, frissítő lekérdezés, törlő lekérdezés, hozzáfűző lekérdezés, paraméteres lekérdezés) és SQL parancsokat az Ön által létrehozott táblán!

Készítsen űrlapot a tábla adatainak bevitelére! Az űrlapon helyezzen el rekordléptető és nyomtatásra szolgáló nyomógombokat!

Készítse el a szakmai információtartalom résznél leírt jelentést, amellyel az autók típusonkénti átlagárát és összértékét határozza el!

2. Olvassa el az alábbi angol nyelvű szakmai szöveget, amely négy adatbázis-kezeléssel kapcsolatos meghatározást tartalmaz! Értelmezze a szöveget, majd írja le a fogalmat és a meghatározás lényegét magyarul!

A. An action query not only finds designated records, but performs some sort of action on those records.

B. A database is a collection of records pertaining to a specific topic. Examples of databases include: customer lists, order records, telephone books and product listings.

C. The design grid is a graphical tool that allows you to create a query by placing fields from the upper half of the Query window into the design grid in the lower half of the window.

D. The primary key is the unique identifier for each record, such as an employee number, a product number, or a customer number.



3. Keresse meg Interneten az Office irodai programcsomag adatbázis-kezelő szoftverének jelenleg elérhető legfrissebb verzióját angol nyelvű kipróbálható változatban! Töltse le, majd telepítse a programot! Írja ide a telepített program nevét és verziószámát!

Az angol nyelvű felületet vesse össze az Ön által használt magyar nyelvű szoftver felületével, és ezek alapján írjon le legalább nyolc egymásnak megfeleltethető angol illetve magyar nyelvű parancsot (menüt). Például: Save as= mentés másként stb.

ADATBÁZIS-KEZELÉS HALADÓKNAK

ÖNELLENŐRZŐ FELADATOK

1. feladat

Sorolja fel a legfontosabb logikai adatmodelleket:

2. feladat

Írja le az ACCES adatbázis-kezelő rendszer fő objektumait és azok jellemzőit:

3. feladat

Csoportosítsa az alábbi lekérdezéseket! Írja betűjelzéseiket a megfelelő vonalra!

- a. Törlő lekérdezés
- b Választó lekérdezés
- c. Hozzáfűző lekérdezés
- d. Kereszttáblás lekérdezés
- e. Táblakészítő lekérdezés
- f. Frissítő lekérdezés

Visszakereső lekérdezések:

Akció lekérdezések:

4. feladat

Írjon le legalább négy összesítő függvényt jelentésével együtt!

5. feladat

Írja le, hogy mi a paraméteres lekérdezés lényege! Mire használható?

6. feladat

SQL nyelv segítségével oldja meg a következő feladatokat! Írja le a vonalra az SQL utasításokat!

a. Készítsen egy táblát, melynek neve BOOK, mezői pedig a következők (<u>könv_ID</u>, szerzo, cim, kategoria, kiado, kiadev, ar) állítson be elsődleges kulcsot!

b. Töltse fel a táblát egy rekorddal (mindegyik mezőnek adjon értéket)! (Pl.:F13, Fekete István, Vuk, regény, , Kossuth, 1985, 300)

c. Adjon hozzá egy oldalszám nevű mezőt a táblához!

d. Az összes könyv árát csökkentse 15 %-al!

e. Petőfi Sándor könyveinek kategóriáját állítsa vers-re!

f. Törölje a táblából a 1961 és 1966 közötti könyveket!

g. Kérdezze le a Arany János könyveit összes adatukkal kiadási év szerint csökkenő sorrendben!

h. Kérdezze le az 1970 előtt kiadott könyvek szerzőjét és címét!

i. Kérdezze le a könyvek átlagárát!

j. Kérdezze le könyvek kiadónkénti számát!

Oldja meg a fenti feladatokat adatbázis-kezelő programban tervező nézetben is! Készítsen a táblához tetszőleges űrlapokat és jelentéseket is!

MEGOLDÁSOK

1. feladat

Logikai adatmodellek: hierarchikus, hálós, relációs, objektumorientált

2. feladat

Tábla: Az adatokat tartalmazza táblázatos formában. A tábla oszlopokból és sorokból áll, amelyeket mezőknek, illetve rekordoknak nevezünk.

Lekérdezés: Tábla adatainak kigyűjtésére szolgál, de alkalmas törlésre, módosításra és számítások elvégzésére is. Forrása tábla vagy lekérdezés lehet.

Űrlap: az adatok kényelmes és látványos formában való bevitelére és módosítására szolgál.

Jelentés: elsődleges feladata a látványos nyomtatás biztosítása, de alkalmas csoportosításokra és összegzésekre is. Forrása tábla vagy lekérdezés lehet.

3. feladat

Visszakereső lekérdezések: b; d

Akció lekérdezések: a; c; e; f

4. feladat

SUM: összegzés; AVG: átlag; COUNT: számolás; MIN: minimum; MAX: maximum

5. feladat 💧

Paraméteres lekérdezés: Végrehajtása alkalmával egy értéket (paraméter) kér be a felhasználótól. Amikor megadjuk ezt az értéket, a lekérdezés mezőfeltételként alkalmazza. Bármelyik olyan lekérdezésnél használható, ahol feltétel lehet megadni.

6. feladat (ACCES SQL)

- a. CREATE TABLE book (könyv_ID TEXT CONSTRAINT kulcs PRIMARY KEY, szerzo TEXT, cím TEXT, kategoria TEXT, kiado TEXT, kiadev INTEGER, ar INTEGER);
- b. INSERT INTO book VALUES ("F13", "Fekete István", "Vuk", "regény", "Kossuth", 1985, 300);
- c. ALTER TABLE book ADD oldalszam INTEGER;
- d. UPDATE book SET ár=ár*0.85;
- e. UPDATE book SET kategoria="vers" WHERE szerzo="Petőfi Sándor";

- f. DELETE * FROM book WHERE kiadev BETWEEN 1961 AND 1965;
- g. SELECT * FROM book WHERE szerzo="Arany János" ORDER BY kiadev DESC;
- h. SELECT szerzo, cim FROM book WHERE kiadev<1970;
- i. SELECT AVG (ar) FROM book;
- j. SELECT kiado, COUNT(könyvID) FROM book GROUP BY kiado;

IRODALOMJEGYZÉK

FELHASZNÁLT IRODALOM

Bártfai Barnabás: Adatbázis-kezelés, BBS-Info, 2004.

Czenky Márta: Access programozás: 2003-as és 2007-es verzió, ComputerBooks, 2007.

Demeter M. Ibolya: Egyszerűen Access 2000 for Windows Panem, 2000.

Farkas Csaba: Az Office 2007 használata, Jedlik Oktatási Stúdió, 2007.

Perry, Greg: Microsoft Office 2007: minden egyben, Kiskapu, 2007.

AJÁNLOTT IRODALOM

Devecz Ferenc – Jónás Katalin – Juhász Tibor – Kévés Rita – Reményi Zoltán – Siegler Gábor – Takács Barnabás: Irány az ECDL, a középszintű érettségi!, Nemzeti Tankönyvkiadó, 2004.

Holczer József – Farkas Csaba – Takács Attila: Informatikai feladatgyűjtemény, Jedlik Oktatási Stúdió, 2007.

sdt.sulinet.hu (folyamatosan)

A(z) 1142–06 modul 014-es szakmai tankönyvi tartalomeleme felhasználható az alábbi szakképesítésekhez:

A szakképesítés OKJ azonosító száma:	A szakképesítés megnevezése
54 482 01 0010 54 01	Adatbázistervező
54 482 01 0010 54 02	Adatelemző
54 481 01 1000 00 00	CAD-CAM informatikus
54 481 01 0100 31 01	Számítógépes műszaki rajzoló
54 481 02 0010 54 01	Infokommunikációs alkalmazásfejlesztő
54 481 02 0010 54 02	Információrendszer-elemző és -tervező
54 481 02 0010 54 03	Internetes alkalmazásfejlesztő
54 481 02 0010 54 04	Szoftverfejlesztő
54 481 03 0010 54 01	Informatikai hálózattelepítő és -üzemeltető
54 481 03 0010 54 02	Informatikai műszerész
54 481 03 0010 54 03	IT biztonság technikus
54 481 03 0010 54 04	IT kereskedő
54 481 03 0010 54 05	Számítógéprendszer-karbantartó
54 481 03 0010 54 06	Szórakoztatótechnikai műszerész
54 481 03 0010 54 07	Webmester
54 481 03 0100 52 01	Számítástechnikai szoftverüzemeltető
54 481 04 0010 54 01	Gazdasági informatikus
54 481 04 0010 54 02	Infostruktúra menedzser
54 481 04 0010 54 03	Ipari informatikai technikus
54 481 04 0010 54 04	Műszaki informatikus
54 481 04 0010 54 05	Távközlési informatikus
54 481 04 0010 54 06	Telekommunikációs informatikus
54 481 04 0010 54 07	Térinformatikus
54 482 02 0010 54 01	IT mentor
54 482 02 0010 54 02	Közösségi informatikai szolgáltató
54 482 02 0010 54 03	Oktatási kommunikációtechnikus
54 213 04 0010 54 01	Designer
54 213 04 0010 54 02	E-játék fejlesztő
54 213 04 0010 54 03	E-learning tananyagfejlesztő
54 213 04 0010 54 04	Multimédiafejlesztő
54 213 04 0010 54 05	Tartalommenedzser
33 523 01 1000 00 00	Számítógép-szerelő, -karbantartó

A szakmai tankönyvi tartalomelem feldolgozásához ajánlott óraszám:

15 óra

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