

Középszint, adatbázis-kezelés, művelet-lista

Az adatbázis-kezelés nehéz „műfaj”. Ehhez képest az érettségien nem díjazták túlzottan. Bár a megszerezhető húsz pont fele, a rutin-feladatok megoldásával, könnyen bezsebelhető.

Átnéztem az oktatás.hu-n elérhető érettségi vizsgák adatbázis-feladatait és a megoldásukhoz szükséges műveleteket egy táblázatban (jegyzőkönyv) gyűjtöttem össze, feladatok szerinti bontásban (név, dátum # sorszám). A listát néhány ponton kiegészítettem a szakmailag elengedhetetlen műveletekkel, majd a műveleteket csoportba foglaltam.

Tapasztalatom szerint, az érettségizők többsége a Microsoft Office programjaival vizsgázik, ezért a listában és a megjegyzésekben az Access elnevezéseit használom. De ez nem jelenti azt, hogy a feladatok is ezeket használják! Lássuk a műveleteket!

adatbázis-műveletek
új adatbázis létrehozása megadott fájlnevével

Minden feladatlap így kezdődik: „Készítsen új adatbázist...” És ezután egy név következik, újmagyarul. Például, cukraszda, szeleromu, repuloter. Szóval, ékezeteket nem kérnek. De nem csak az adatbázis-, de a belső objektumok nevében sem.

text fájl adatainak importálása új táblába
megadott kódlap beállítása
megfelelő kódlap kiválasztása
a mezőneveket tartalmazó első sor deklarációja
adattípus-csere: Rövid szövegből Dátum/Idő
adattípus-csere: Rövid szövegből megfelelő mezőméretű Szám
adattípus-csere: Számból Igen/Nem
elsődleges kulcsnak szánt mező indexelése (nem lehet azonos)
elsődleges kulcs mezőjének kiválasztása
tábla nevének megadása
tábla átnevezése a navigációs ablakban

A feladatok többségében megadják a text fájl elkészítéséhez használt kódrendszert, de ha hiányzik, akkor a megfelelőt, „próba szerencse” alapon, nekünk kell kiválasztanunk a varázsló listájából. Aprópó, lista. Az egyik feladat állományait, a leírás szerint, az ISO8859-2 szabvány szerinti kódolással hozták létre, de ilyet nem találunk a segédprogramban. Azért nem, mert ezt a kódrendszert az Access „közép-európai (ISO)”-nak nevezi.

A varázsló a dátumokat tartalmazó, valamint hiányzó adatokat és számokat vegyesen tartalmazó oszlopokhoz „Rövid szöveg” adattípust állít be. Viszont a csak nullákat (0) és egyeket (1), valamint a nullákat és mínusz egyeseket (-1) vegyesen tartalmazó oszlopokat „Hosszú egész” mezőméretű számnak látja. Ezeket az automatikus besorolásokat módosítanunk kell!

A dátumoknak „Dátum és időpont”-, az egész számoknak „Hosszú egész”-, a tizedestört számoknak „Dupla”-, a nulláknak és az egyeseknek, valamint a nulláknak és a mínusz egyeseknek „Igen/Nem” adattípust, illetve mezőméretet állítunk be!

a tábla módosítása tervező nézetben
adattípus-csere
elsődleges kulcs megadása
többmezős elsődleges kulcs megadása
mező létrehozása megadott néven
számláló adattípus beállítása elsődleges kulcs mező számára

Ha a varázslóban elmulasztottuk, akkor az adattípus-cserét a tábla tervező nézetében is el tudjuk végezni. Egy kivétellel! A dátumokat tartalmazó Rövid szövegnek beállított mező adatai az átállítás során törlődnek. Erre a program „a Microsoft Access által végzett konvertáláskor hibák történtek” szöveggel figyelmeztet. Ezt a problémát, tehát csak a text-fájl ismételt importálásával orvosolhatjuk. A többi esetben a csere nem jár adat-vesztéssel, annak ellenére sem, hogy az Access a műveletet követően a „néhány adat elveszett” szöveggel parázttatja a szegény felhasználót.

Az elsődleges kulcs mező beállítását tervező nézetben is el kell tudnunk végezni! Két feladat többmezős elsődleges kulcs beállítását kéri. A végrehajtás nem okozhat gondot! Először Ctrl billentyűvel jelöljük ki a mezőket, majd a szalagon kattintsunk az elsődleges kulcs vezérlőre.

kapcsolatok
kapcsolat létrehozása az integritás megőrzésével
kapcsolat törlése

Figyelem, figyelem! Több feladatlapon a kapcsolat ábrázolása szimbolikus! A vonal megvan, de nem a kapcsoló mezőket köti össze. Az idegen kulcs mezőt tehát nekünk kell „megtalálni”. Az esetek többségében ez nem okoz problémát, mert a neve tartalmazza az elsődleges tábla nevét és az „id” (identification) rövidítést. A kapcsolat egy-, illetve több oldalának jelzését (1, ∞) hiába keressük, mert azt egyik feladatlapon sem tüntették fel.

választó lekérdezés összeállítása a tervező-rácson
tábla, lekérdezés hozzáadása
tábla, lekérdezés törlése
összes mező hozzáadása
mező hozzáadása, pozícionálása, törlése
mező rejtése a lekérdezés eredményében
egy- és többmezős rendezés beállítása

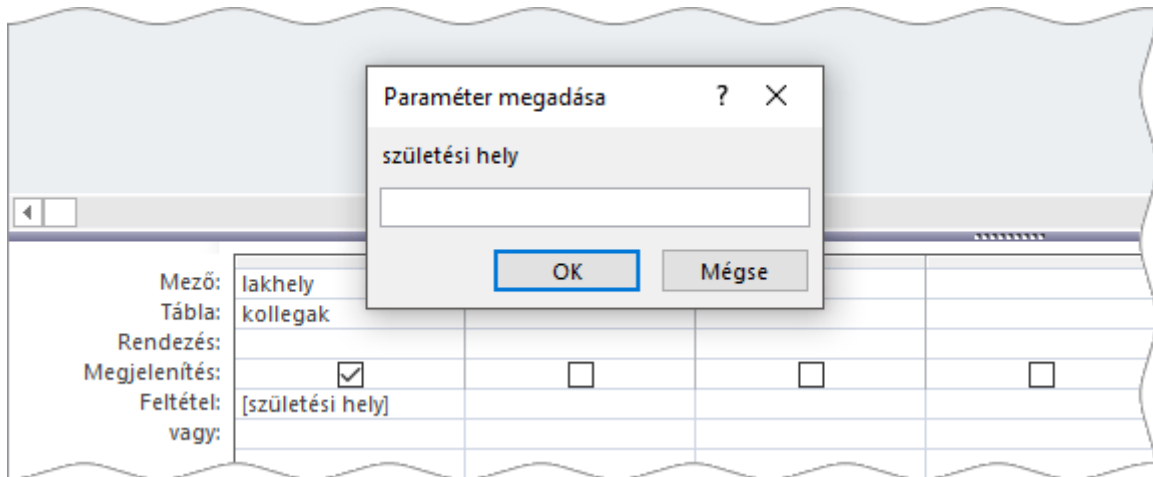
Az érettségi adatbázis-feladat nagyrésze, különböző „fajsúlyú”, lekérdezésekből áll, amelyeket tervező nézetben, a QBE (Query By Example) rács, vagy egyszerűen a tervező-rács, segítségével hozunk létre. Ez a grafikus felület teszi lehetővé, hogy ne SQL (Structured Query Language) nyelven kelljen megfogalmaznunk, milyen adatokra vagyunk kíváncsiak. A program nézet-váltáskor vagy a lekérdezés mentésekor „leolvassa” a rácsot és összeállítja a megjelenítési utasítást, amelyet SQL nézetben meg is jeleníthetünk.

egyszerű feltételek a lekérdezésben
szöveg, dátum, idő és logikai konstans szintaktikája
egyenlő reláció a feltételben
kisebb (<), nagyobb (>), kisebb egyenlő (<=) nagyobb egyenlő (>=)
értéktartomány (Between ... And ...)
logikai mező feltételei (Igaz/Hamis, Igen/Nem, Be/Ki, -1/0)
üres (Is Null), nem üres (Is Not Null/Not Is Null)
nem egyenlő reláció (<>, Not) a feltételben
helyettesítő karakterek (*, ?) szövegben, dátumban
mezőnév, mint feltétel

Az én gépemem (Win' 10, Office 2019) az Access, a lekérdezés feltételében, nem fogadja el a konkrét dátumot (#2020. 02. 02.#), „Érvénytelen dátumértéket tartalmazó kifejezést írt be”, kapom az üzenetet. Ha ez program-hiba, és valószínűleg az, akkor ne kísérletezzünk, én megtettem, semmi eredménye! Használjuk a DateSerial függvényt. Szintaktikája: DateSerial(évek száma; hónapok száma; napok száma). Például, DateSerial(2020;2;2). Vagy relációval: >=DateSerial(2020;2;2). Vagy periódust meghatározva: Between DateSerial(2020;2;2) And DateSerial(2020;3;2).

Az egyik feladatban a dátumot és időpontot is tartalmazó (2020. 02. 10. 13:48:52) mezőből kell kiválogatni egy teljes nap bejegyzéseit. Két megoldás között választhatunk: [1] Between #2020. 02. 10. 0:00:01# And #2020. 02. 10. 23:59:59# vagy [2] Year([mezőnév])=2020 And Month([mezőnév])=2 And Day([mezőnév])=10.

Figyelem, figyelem! A mezőnév a feltételben mindig szögletes zárójelek között áll! Ha a zárójelek megvannak, de elgépeljük a mezőnevet, akkor ezt a panelt jeleníti meg az Access:



A képen látható esetben a mezőnév biztos ékezetek nélküli, mi meg, természetesen, ékezetekkel írtuk. Mellékesen, ez az újmagyar, ékezetes karakterek nélküli, nyelv csak az érettségi vizsgán használatos, az ÉLET-ben, windozos környezetben, nem!

A mezőnévről jut eszembe! Mit jelent az, ha egy mezőnek egy másik mező a feltétele? Például, a „lakhely”, amely egy település-neveket tartalmazó mező, feltétele a „születési hely” mező, amely szintén település-neveket tartalmaz. Ebben az esetben a megjelenítési utasítás emberek között így hangzana: Mutasd, azoknak az embereknek az adatait, akik a szülőhelyükön laknak. Ember és az Access között meg így: Jelenítsd meg azokat a rekordokat, amelyekben a „lakhely” és a „születési hely” mezők bejegyzései azonosak!

feltétel-rendszerek a lekérdezésben
feltételek összekapcsolása (Or, And)
feltétel-lista VAGY viszonyban álló elemekből (In)
a rács egy sorában álló feltételek (ÉS viszony)
a rács különböző soraiban álló feltételek (VAGY viszony)

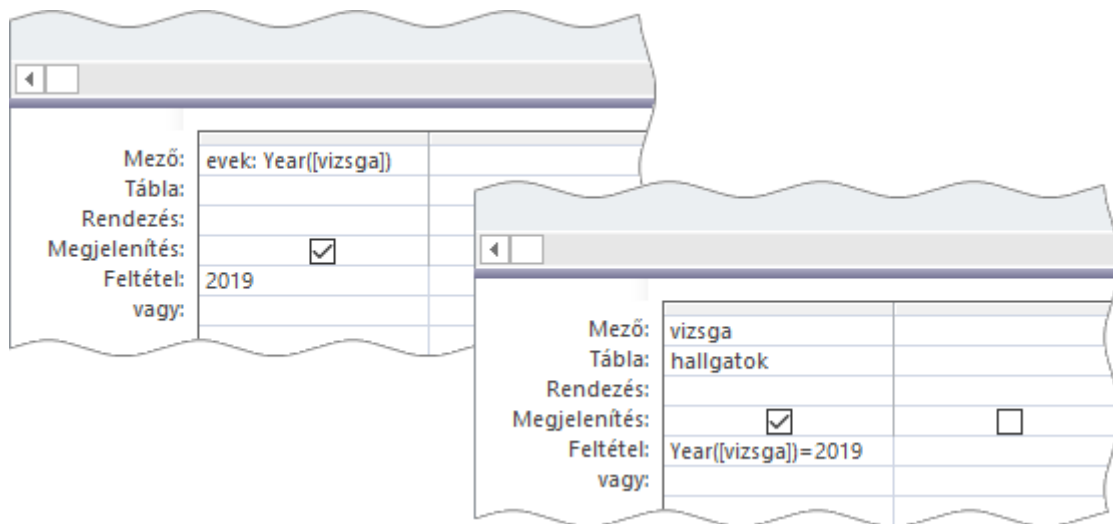
A feltételt tartalmazó mezőt, az egyszerűség kedvéért, nevezzük feltétel-mezőnek! Ha a lekérdezés csak egyetlen feltétel-mezőt tartalmaz, akkor a felsorolt „lehetőségek”, több feltétel esetén, különböző megoldásokat kínálnak. Több feltétel-mező esetén, ez a szabadság elvész, mindig az adott feladatnak megfelelő elrendezést, illetve írás-módot kell választanunk.

számított mező a lekérdezésben
a deklaráció szintaktikája (név:kifejezés)
aritmetikai műveletek a kifejezésben
összefűzés a kifejezésben (&)
Abs, Min, Max, Avg függvények a kifejezésben
Year, Month, Day, Hour függvények

A feladatok többségében a statisztikai függvények (Min, Max, Sum, Avg, Count) alkalmazása össze-sítő lekérdezéssel elkerülhető.

Az egyik feladatban két dátum között eltelt napok számát kell „lekérdeznünk”. Ez a számítás elég gyakori az adatbázis-kezelésben. Gondoljunk például egy munka elvégzéséhez szükséges napok számának megállapítására. Napok: [befejezes]-[kezdes]+1. Miért kell az a plusz egy? Képzeljünk el egy egynapos munkát. Reggel elkezdik, este befejezik. Tehát a két mező dátuma azonos. De mégse lehet nulla az eredmény, hiszen egy teljes napot dolgoztak az emberek!

A „feltétel-rendszerek” művelet-csoportban szándékosan nem említettem, hogy a feltétel kifejezés is lehet. Azért nem tettem, mert ez a lehetőség fogalmilag nem magyarázható. Nézzük miről van szó! Vegyünk példának egy dátumokat tartalmazó mezőt: „vizsga”. Tudni szeretném ki vizsgázott 2019-ben. Két megoldás közül választhatunk.



A „klasszikus”, tehát a fogalmilag helyes változat felül látható, az Access specifikus pedig alul. A két lekérdezés azonos eredményt ad.

a lekérdezés tulajdonságai	
egy mező szélső-értékeit tartalmazó sorok megjelenítése	
egyedi sorok megjelenítése	

Tehát a lekérdezés „Csúcsérték” és „Egyedi értékek” tulajdonságairól van szó. Fontos a pontos megnevezés, mert a „Csúcsérték” tulajdonságot a menüszalagon is beállíthatjuk, de ott „Visszatérés” a felirata a vezérlőjének. Az „Egyedi értékek”-et pedig könnyű összekeverni, a tulajdonságla-pon közvetlen alatta álló, „Egyedi rekordok” tulajdonsággal.

A lekérdezés tulajdonságlapját a menüszalag vezérlőjével vagy az Alt+Enter billentyű-paranccsal jeleníthetjük meg. Az esetek többségében, azonban nem a lekérdezés-, hanem az aktív mező tulajdonságait kapjuk. A lekérdezés tulajdonságainak megjelenítéséhez kattintsunk a tervező-rács fölé, a táblák területére!

A lekérdezés eredménye egy táblázat, az egyszerűség kedvéért, nevezzük eredmény-táblázat-nak. A két tulajdonság az eredmény-táblázat sorait befolyásolja. A Csúcsérték-vel csak azoknak a soroknak a megjelenítését írjuk elő, amely egy mező, rangsor szerinti, szélsőértékeit tartalmazzák. A lekérdezés rendező-mezőjének a szélső-értékeket tartalmazó mezőnek kell lennie. Ha a legnagyobbakra vagyunk kíváncsiak, akkor csökkenő, ha a legkisebbekre, akkor emelkedő rendezést kell beállítanunk. A Rövid szöveg adattípusú mező bejegyzéseinek rangsorán az ABC szerinti rendezettséget értjük. A megjelenítendő sorok számának, Csúcsérték szerinti, meghatározásánál az Access „nem tesz igazságot”. Tehát, ha a három legnagyobbat kérjük, és mondjuk, a harmadik legmagasabb pontszám tíz rekordban is szerepel, akkor az eredmény-táblázatban mind a tíz sor meg fog jelenni. A Csúcsérték nem csak egész szám, de százalék is lehet. A viszonyítási alap, vagyis a száz százalék, az eredmény-táblázat, szelektálás nélküli, sor-száma.

Az Egyedi értékek tulajdonság Igaz értékével az eredmény-táblázat ismétlődő sorainak megjelenítését tilthatjuk meg.

összesítő lekérdezés
egy- vagy több csoportosító mező létrehozása
egy- vagy több statisztikai mező deklarálása (Count, Sum, Avg...)
egy- vagy több feltétel-mező deklarálása
kifejezés-mező létrehozása

Az összesítő lekérdezésben állhatnak csoportosító (Group By), statisztikai (Sum, Count, Avg...), kifejezés (Expression) és feltétel (Where) mezők. Mindegyiket szelektálhatjuk feltétellel, csak a Where szócskával deklarált mezők nem fognak megjelenni az eredmény-táblázatban.

Az eredmény-táblázat sorait az Access a Group By mezők adatai alapján, növekvő sorrendben, automatikusan rendezi.

frissítő lekérdezés
módosító konstans megadása
módosító kifejezés deklarálása
a módosítandó adatok meghatározása feltétellel
lekérdezés futtatása

A módosító kifejezésben az adatokat tartalmazó mező nevét is szerepeltethetjük. Például, árleszállítás miatt az ár harminc százalékkal csökken: [ár]*0,7. A kis képlet két fontos tulajdonságot mutat: [1] az SQL nyelvben a pont a tizedes elválasztó, de ennek ellenére a tervező-rácson nekünk vesszőt kell használnunk, amit majd az Access pontra konvertál az SQL utasítás összeállításakor, [2] a program képleteiben a százalékjel nem használható.

tábla-készítő lekérdezés
létrehozása
lekérdezés futtatása

A tábla-készítő lekérdezés mentése egy utasítás rögzítése: ezekkel a mezőkkel és rekordokkal hozz majd létre egy új táblát, ha majd szólok! A tábla létrehozására felszólító utasítás a lekérdezés futtatása!

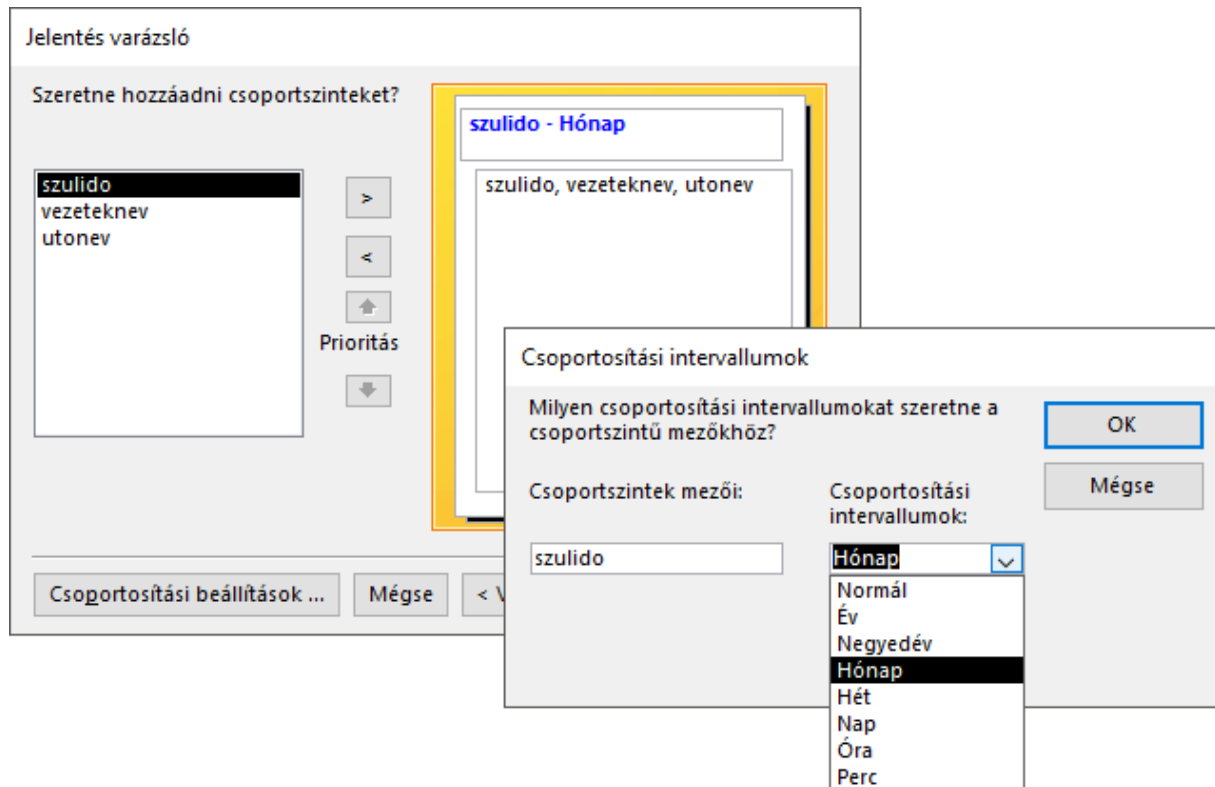
SQL utasítás (segéd-lekérdezés) a feltételben
az SQL utasítás másolása
az utasítás szintaktikája a feltételben (SQL)
SQL utasítás relációval és az In szócskával
segéd-lekérdezés függvény argumentumaként ((SQL))

Az SQL utasítás szerepeltetése a feltételben, a segéd-lekérdezés kezelésének legegyszerűbb módja. Néhány szabályt azonban be kell tartanunk. Az utasítás mindig álljon zárójelben! Ha függvénybe helyezzük, akkor a függvény zárójeleit is ki kell tennünk! Például, Year((SQL)). Ha a segéd-lekérdezés egynél több adatot ad vissza, akkor az SQL utasítást az In szócska előzi meg: In(SQL). Természetesen ezt a kifejezést is tagadhatjuk! Nem ezekre vagyok kíváncsi: Not In(SQL).

Egy kivételtől eltekintve, az érettségi feladatok úrlap-készítést nem tartalmaznak. Tehát, akkor már csak a jelentés van hátra. A jelentéseket a vizsgán varázslóval hozzuk létre!

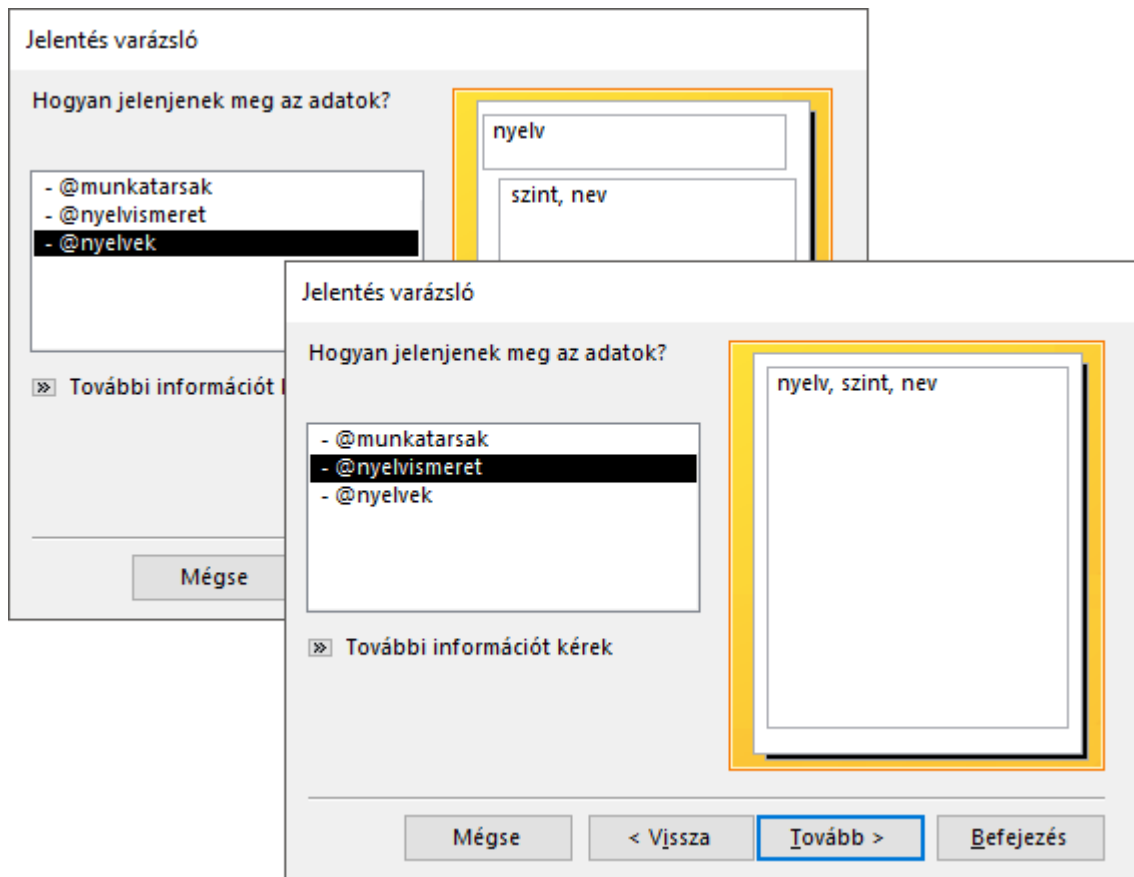
jelentés készítése varázslóval
tábla vagy lekérdezés alapján
egy- vagy több csoportosító mezővel
csoportosító dátum-egység beállítása
rendezés egy- vagy több mező alapján
fekvő tájolás beállítása

Ha a jelentés lekérdezésen alapul, akkor a lekérdezést is nekünk kell létrehozni. A lekérdezés tartalmazza, sorrendben, a jelentés mezőit, csoportosítás és rendezés nélkül, az esetleges feltételekkel! A csoportosítást és a rendezést a varázslóban állítsuk be! Ha a csoportosító mező Dátum/Idő adat-típusú, akkor az Access automatikusan hónapos csoportokat képez. Ezt a *Csoportosítási beállítások...* feliratú vezérlőre kattintva módosíthatjuk.



A lista *Normál* tételét választva a rekordok dátum szerint lesznek csoportosítva. Például, 2020.02.14-i csoportban három, a 2020.02.15-i csoportban öt vásárlás lesz felsorolva.

De az automatikus csoportosítás, illetve annak felajánlása akkor is előfordulhat, ha a jelentés adatai több táblából származnak. Az alábbi képen egy ilyen helyzetet látunk.



A segédprogram automatikus csoportosítás ablakának bal oldalán álló, a jelentés által érintett, táblák nevére kattintva tudjuk a csoportosítást módosítani, illetve elvetni. Felül az automatikusan csoportosított-, alul a csoportosítás nélküli elrendezést látjuk. Az automatikus csoportosítás elfogadása vagy elvetése, nincs befolyással a jelentés-készítés további lépéseire. Magyarul, az automatikus csoportosítás ablaka után a varázsló, minden esetben, megjeleníti az egyéni csoportosítás ablakát is.

jelentés módosítása tervező nézetben
vezérlők méretezése, pozícionálása
címkék szövegének módosítása

Visszatérve oda, ahol elkezdjük, az adatbázis-kezelés nehéz „műfaj”. Időt kell szánni rá! Gyakoroljunk érettségi feladatokon, és nézegessük a megoldásukat! A lekérdezések tervező nézete, számos esetben, érthetetlen részleteket tartalmaz. Ennek oka, hogy a feladatok készítői a lekérdezéseket SQL nyelven állítják össze, és ennek elemeit próbálja az Access a tervező rácson megjeleníteni. De ez nem mindig eredményez „értelmes” elrendezést, mert ahogy azt már említettem, ezt a felületet pont az ellenkező célra fejlesztették ki: SQL utasítás készítésére, a tervező-rácsra helyezett elemekből.



margitfalvi.arpad@gmail.com