

Adatbázis rendszerek

Segédlet a 3. házi feladat elkészítéséhez

Dr. Molnár Bence, Dr. Koppányi Zoltán, Dr. Berényi Attila

11/30/2021

Kézirat

A segédlet azzal a céllal készült, hogy az Adatbázis rendszerek c. tárgy 3. házi feladatának elkészítését elősegítse. A segédlet a 2014. őszi félév követelményei alapján készült. Amennyiben valahol a segédlet hibás, vagy nehezen érthető, kérem jelezd a molnar.bence@emk.bme.hu email címen.



Ezen a területen olvashatsz olyan hibákról, és azok elhárításáról, melyeket könnyű elkövetni.



További érdekességek, megjegyzések, valamint külső referenciák.



Feladatok, melyeket el kell végezned!

Tartalom

Tudnivalók	3
Új adatbázis létrehozása	3
Táblák létrehozása	5
Új tábla létrehozása	5
Tábla mentése	6
Tábla nézetek	7
Adattípusok	7
Kulcs mező(k)	9
Mezők tulajdonságaink beállítása	10
Kapcsolatok kialakítása	13
Kapcsolatok létrehozása	14
Kapcsolatok szerkesztése	17
Kapcsolatok kialakítása Keresés varázsló segítségével	18
Adatok feltöltése és törlése	24
AutoNumber visszaállítása	28
Lekérdezések	29
Lekérdezés létrehozása	29
A lekérdezés tervező	31
Lekérdezés futtatása	33
Lekérdezés nézetek	34
Projekció	35
Szelekció	36
Unió	38
Descartes-szorzat	38
Természetes összekapcsolás	39

Théta összekapcsolás.....	41
Mező alias.....	43
Összesítő lekérdezések.....	44
Sorbarendeázés.....	49
Matematikai műveletek mezők között.....	49
Adat átalakítás	50
Idegen kulcsok előállítása Excel segítségével.....	50
Űrlapok és jelentések	51
Űrlapok.....	52
Jelentések.....	55
Adatbázis véglegesítése	56

Tudnivalók

A segédlet a hallgatók 3. házi feladatának elkészítéséhez nyújt segítséget. A 3. házi feladat szervesen az 1. és 2. házi feladaton alapszik. Cél az ott specifikált adatbázis megvalósítása Microsoft Access adatbázis-kezelő rendszer segítségével. A Microsoft Access a Microsoft Office programcsomag része. Az itt bemutatott példák 2013-es, angol verzióval készültek.



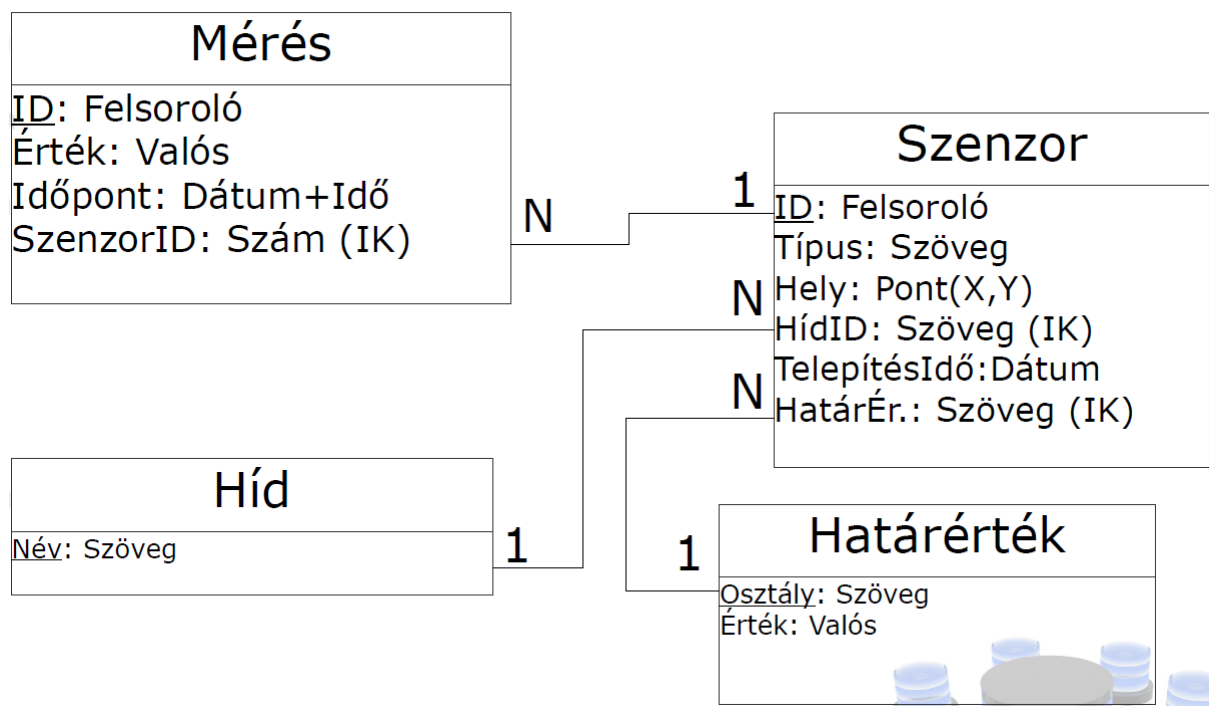
A feladattal kapcsolatban a következők az elvárásaink:

- Az 2. házi feladatból a táblák kialakítása
- Kapcsolatok kialakítása, legalább egy kapcsolat keresés varázslóval
- Beviteli maszk, érvényességi szabály és tizedes jegyek, kötelező paraméter beállítása, ott, ahol szükséges!
- min. 5 darab lekérdezés, melyből legalább 2 összesítő lekérdezés
- min. 3 darab űrlap
- min. 3 darab jelentés



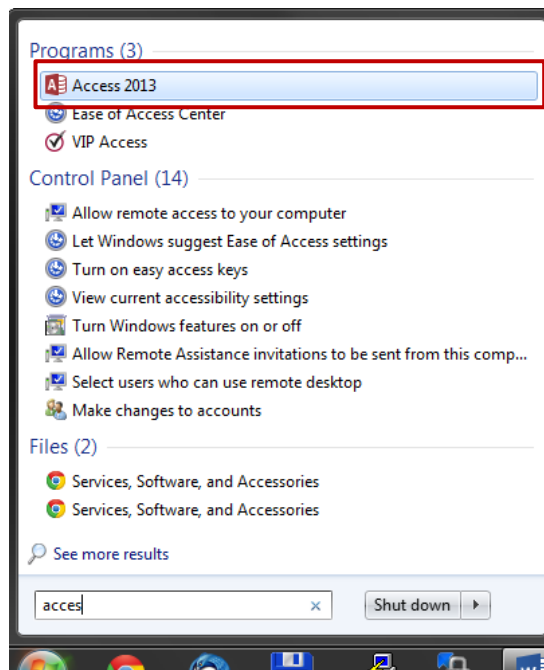
Sajnos a magyar területi beállítások használata esetén az Ms Access 2016 több helyen is lehetetlenné teszi a cél megvalósítását. Legtöbbször Dátum jellegű és tizedestört adatok importálása esetén. Ilyenkor egyetlen megoldás a területi beállítások megváltoztatása (vagy ilyen beállításokkal rendelkező gép használata).

Ebben a dokumentumban bemutatott példa az alábbi relációs séma diagramból indul ki.

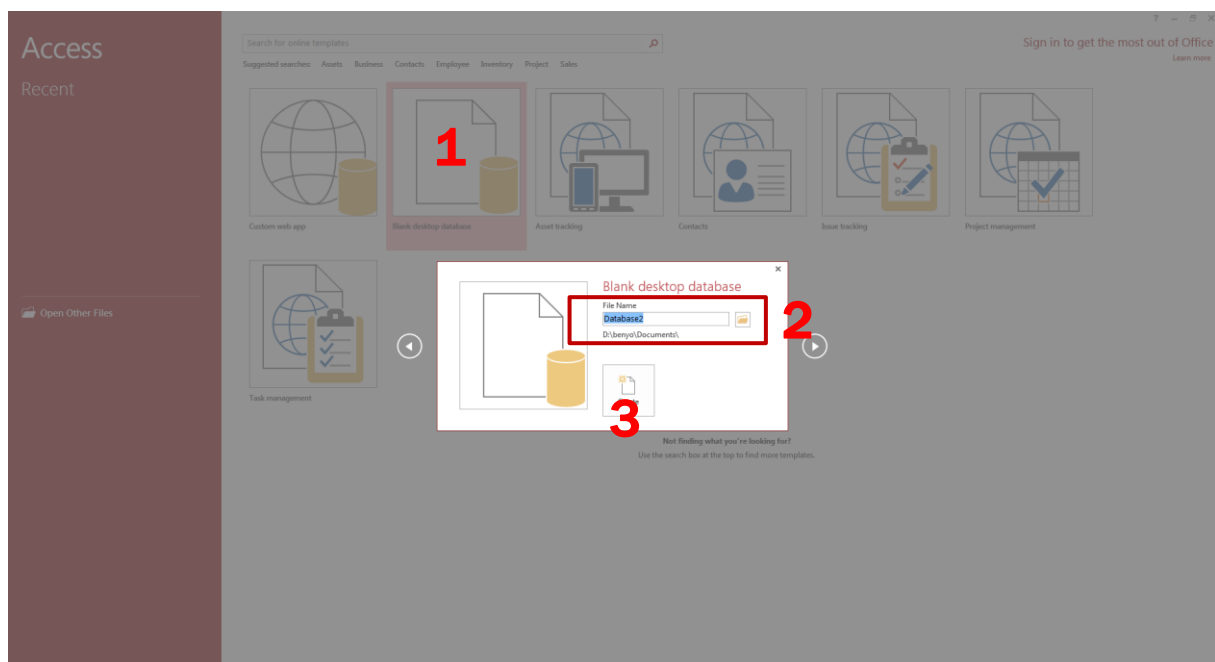


Új adatbázis létrehozása

Először nyissuk meg a Microsoft Access programot!

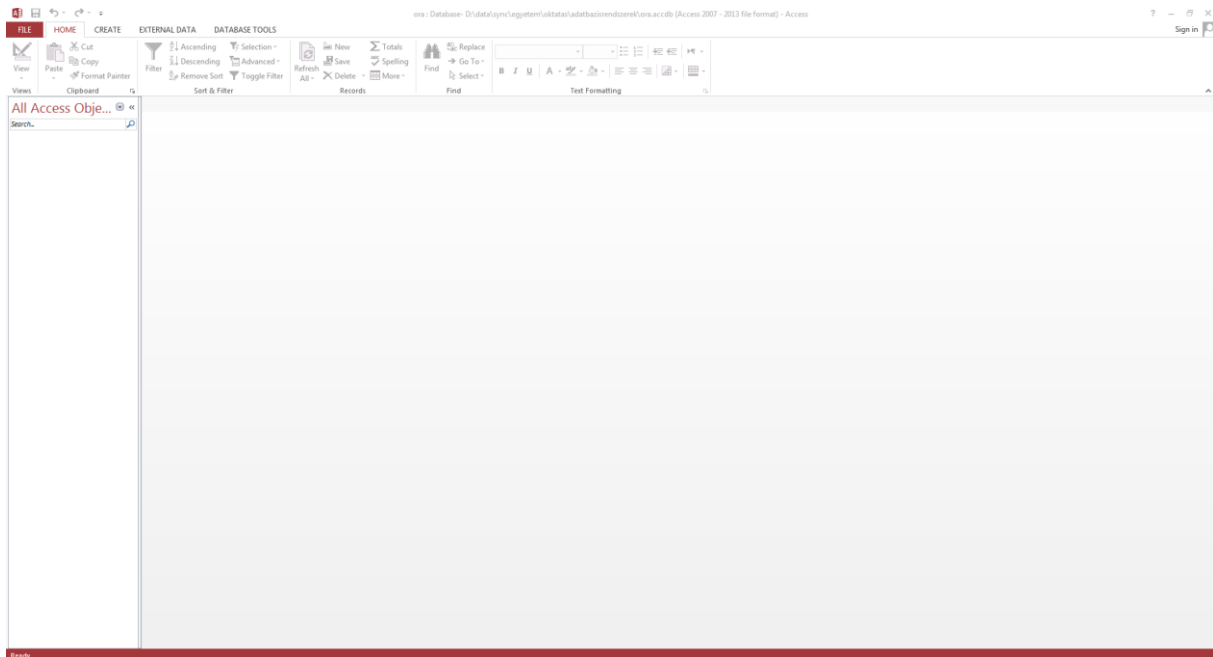


Megjelenik a Microsoft Access kezdőlapja!



1. Válasszuk az „Blank desktop database”-t a template fájlok közül.
2. A felpattanó ablakban adjuk meg az új adatbázis nevét és/vagy helyét!
3. Kattintsunk a „Create” gombra!

Ezek után megjelenik az alábbi képernyőkép.



Indítsuk el a Microsoft Access 2013-t és hozzuk létre az első adatbázisunkat!

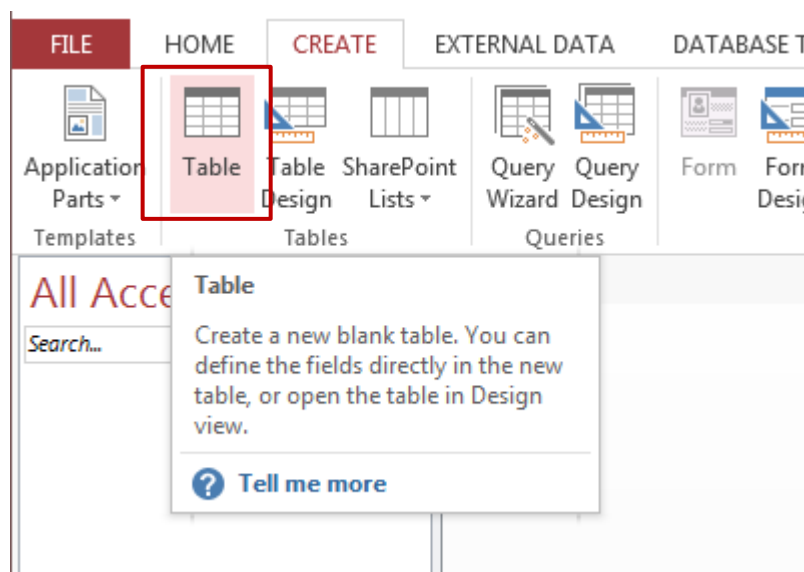


A feladat elkészíthető más verziójú Access-szel is, ekkor azonban az oktatói ellenőrzés során a futtatásról neked kell gondoskodni (pl. saját laptopon).

Táblák létrehozása

Új tábla létrehozása

Új táblát a „Create” fülre, majd a „Table” parancsot alkalmazva tudunk létrehozni.



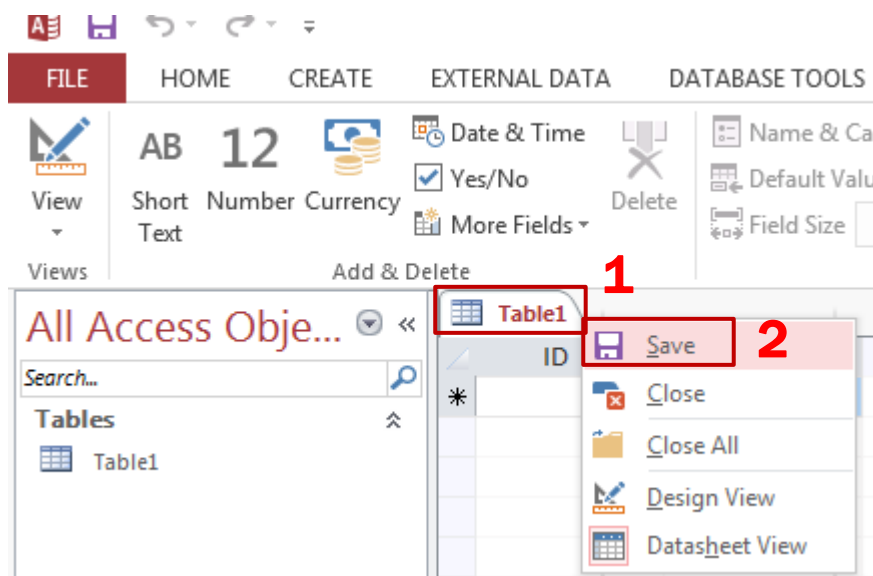


Fontos, hogy először azt a táblát hozzuk létre, ami semmilyen más táblára nem hivatkozik idegen kulccsal, mert majd sorban olyanokat, amik csak már létrehozott táblákra hivatkoznak idegen kulccsal.

Tábla mentése

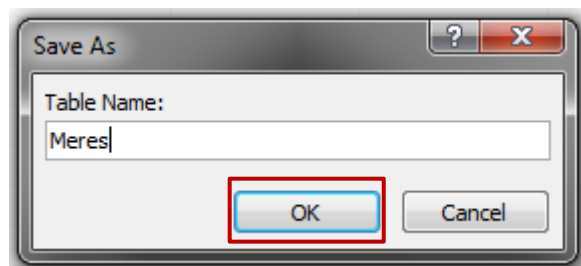
A táblák mentése a tábla fülre kattintás után érhető el. Ne felejtjük el minden változás után a táblát is menteni. Amennyiben olyan változtatást, vagy nézetet akarunk váltani, amihez szükséges a mentés, arról az Access figyelmeztetni fog minket.

Tehát a tábla mentéshez kattintsunk a tábla fülére **jobb** egérgombbal.



- A „Save”-re kattintva menthetjük a táblánkat.
- A „Close” a tábla nézetét bezárhatjuk.
- A „Close All” paranccsal az összes fül bezáródik.
- A „Design View” parancsra kattintva a tervező nézetre válthatunk át.
- A „Datasheet View” az adatlap nézetet adhatjuk meg.

Ha a még nem mentett táblát akarjuk menteni, megadhatjuk a tábla nevét. Tegyük ezt meg, és kattintsunk a „Save” opcióra. A táblaneveknél figyeljünk rá, hogy kerüljük az ékezetek és a szóközők használatát.

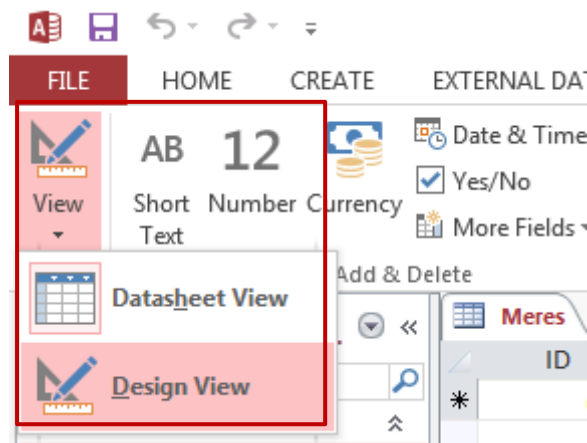


Adjuk meg a tábla nevét és kattintsunk az „OK” gombra.

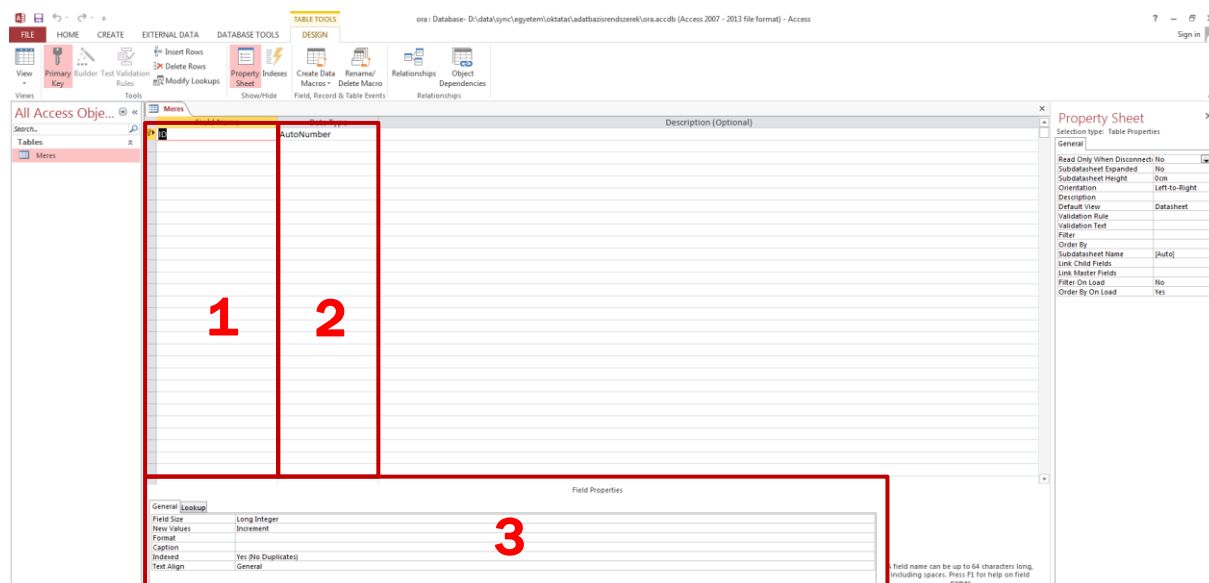
A táblát – amennyiben bezártuk – a bal oldali menüben a tábla nevére dupla kattintással tudjuk megnyitni.

Tábla nézetek

A táblának két nézete fontos, ezek a már korábban említett tervezési („**Design View**”) és adatlap („**Datasheet View**”) nézet. Már korábban láthattuk, hogy a tábla fülére jobb kattintással tudunk közöttük váltani. Másik lehetőség a szalag menükön („**Home**” fül) keresztül érhető el, a „**View**” opcióra kattintva.



Alap esetben az adatlap nézet jelenik meg. Váltunk át tervező nézetre („**Design View**”). Ekkor a következő képernyőképet kapjuk.



1. **Mező neve (Field Name):** Itt adhatjuk meg a mezők (attribútumok) neveit. A mezőneveknél (ahogy a táblaneveknél is) figyeljünk rá, hogy kerüljük az ékezetek és a szóközők használatát.
2. **Adattípus (Data Type):** Itt adhatjuk meg az adatok típusát.
3. **Mező tulajdonságok (Field Properties):** Itt adhatunk meg különböző megszorításokat a típusra nézve, illetve további más mező beállításokat.

Adattípusok

Az adat típusát egy legördülő menüből választhatjuk ki.

Field Name	Data Type
ID	AutoNumber
	Short Text
	Long Text
	Number
	Date/Time
	Currency
	AutoNumber
	Yes/No
	OLE Object
	Hyperlink
	Attachment
	Calculated
	Lookup Wizard...

General	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

Ezek közül a legfontosabbak:

- **AutoNumber:** A számláló (felsoroló) típus, mely esetén automatikusan, amikor egy új rekordot viszünk fel, akkor 1-től számozva a program automatikusan kitölti a mezőt. Ez a háttérben egy számszerű adattípus, az automatikus értékadás tulajdonságával felruházva. Amennyiben erre a mezőre hivatkozik egy idegen kulcs, akkor a hivatkozó tábla idegen kulcsának az adattípusa legyen **Number** (Szám) és a Field Size (Mezőméret) legyen **Long Integer** (Hosszú egész).
- **Number:** Szám típus.
- **Short Text:** Rövid szöveg típus (max 255 karakter).
- **Long Text:** Szöveges típus.
- **Date/Time:** Dátumot és/vagy időt adhatunk meg a mezőben.
- **Yes/No:** A logikai típus, mely Igen vagy Nem értéket vehet fel.
- **Lookup Wizard:** Később kerül ismertetésre.



A felsoroló (**AutoNumber**) típus folyamatosan számoz, azaz ha kitörlünk egy rekordot, akkor a törölt rekord azonosítója már soha nem lesz felhasználva a táblában. Ez nem jelent problémát, hiszen helyes tervezés esetén a felsoroló típust kulcsként használjuk, és ebben az értelemben mindegy, hogy mi a konkrét érték, csupán az az elvárás, hogy egyedi legyen. Ha mégis megkönnyítené az életünket, a számláló visszaállítása, későbbi fejezetekben annak módja is megtalálható.

Ezután elkezdhetjük felvinni a mezőket (attribútumokat) az 2. házi feladat relációs séma diagramja alapján.

Mérés	
ID: Felsoroló	
Érték: Valós	
Időpont: Dátum+Idő	
SzenzorID: Szám (IK)	

Mérés	
Field Name	Data Type
ID	AutoNumber
Ertek	Number
Idopont	Date/Time
SzenzorID	Number

A fenti ábrán jelölt rész mutatja az aktuális kulcs mezőt.



Figyeljünk oda, hogy az idegen kulcsok esetén a típus megegyezzen a hivatkozott mező típusával! Ha az elsődleges kulcs „AutoNumber” (Számláló), az idegen kulcs „Number” legyen és ezen belül a „Field Size” (Mezőhossz) „Long Integer” (Hosszú egész)!

Kulcs mező(k)

Ezután definiáljuk a kulcs mezőket. A fenti ábrán a kulcs jel (pirossal kiemelve) jelzi az épp aktuális kulcsot. Ezt meg tudjuk változtatni úgy,

1. hogy kijelöljük azon mező(ke)t, amelyeket kulccsá akarunk tenni (ehhez a kulcs alatti sávra kell kattintanunk),
2. majd a „Primary Key” parancsra kattintunk a szalagmenü File fülén.

The screenshot shows the Microsoft Access interface. The 'FILE' ribbon is active, and the 'Primary Key' button is highlighted with a red box and a red number '2'. The 'Mérés' table is open in Design view, and the 'ID' field is highlighted with a red box and a key icon. The 'Ertek' field is highlighted with a red box and a red number '1'.

Ekkor a kijelölt mezők mellett fog feltűnni a kis kulcs jel.

Mezők tulajdonságaink beállítása

A korábban már említett „Mezők tulajdonságai” résznél tudunk a mezőkön beállításokat eszközölni. Ahhoz, hogy a megfelelő mezőn ezt meg tudjuk tenni, először azt ki kell választani, ezt pedig úgy tehetjük meg, hogy a tervező nézetben egyszerűen rákattintunk.

General		Lookup
Field Size	Double	
Format		
Decimal Places	1	
Input Mask		
Caption		
Default Value	0	
Validation Rule	>0	
Validation Text		
Required	Yes	
Indexed	No	
Text Align	General	

Ha egy szám típusra kattintunk, akkor megjelennek a fenti beállítások. Ezek közül a következőket emelnénk ki:

- **Field size:** Itt a szám típusnál több lehetőségünk van, ezeket a legördülő menüből érhetünk el:
 - Integer: Egész számot szeretnénk tárolni.
 - Long integer: Hosszú egész, amennyiben a mező egy idegen kulcs és egy AutoNumber (számláló) adattípusú elsődleges kulcsra hivatkozik, akkor ezt a típust kell választanunk.
 - Double: Dupla pontos számok, melyek segítségével valós számokat is tudunk tárolni.
- **Decimal Places:** Itt adhatjuk meg a tizedes jegyek számát, ami a mérnöki gyakorlatban fontos lehet.
- **Default Value:** Alapértelmezett érték, amit a mező automatikusan értékül kap, amikor új rekordot viszünk fel.
- **Validation rule:** Érvényességi szabály, ennek segítségével megszorításokat tehetünk a mezőre vonatkozóan. Például a mintafeladatban beállított (>0) esetén azt adjuk meg, hogy a mezőben tárolt értékeknek pozitívnak kell lennie. Amennyiben a [...] gombra kattintunk, további megszorításokat adhatunk meg. További információk (angolul): <http://allenbrowne.com/ValidationRule.html> (2012)
- **Required:** megadhatjuk, hogy a mezőt kötelező-e kitölteni. Amennyiben igen, akkor nem maradhat a mező üresen, azaz nem vehet fel NULL értéket. A legtöbb mezőnket kötelező megadni, ha másként van, akkor jó eséllyel nem megfelelő a tervezett adatbázis séma!
- **Indexed:** Itt adhatjuk meg, hogy a mező értékek indexelve legyenek-e. Az indexek felgyorsítják a lekérdezéseket, a sűrűn keresett mezőket érdemes indexelni. Itt állítható be, hogy egy érték egyedi (No duplicates) legyen-e.

Még egy beállítás fontos, amit tovább részletezünk, ez pedig az „Input Mask” (beviteli maszk) megadása (tábla mentése után). Ezt akkor érdemes használni, amikor a mezőnek egy fix formátumot kell követnie. Ilyen lehet például egy telefonszám, ahol pl. három szám, kötőjel és


négy szám formátumúnak kell lennie (pl. 123-4567). A beviteli maszk segítségével nem csak segíthetjük az adatfelvételt, de biztosíthatjuk, hogy az adatok a megfelelő formátumban kerülnek felvitelre az adatbázisba. Az Access beviteli maszk varázslója csak szöveg és dátum típusú adatokat kezel.

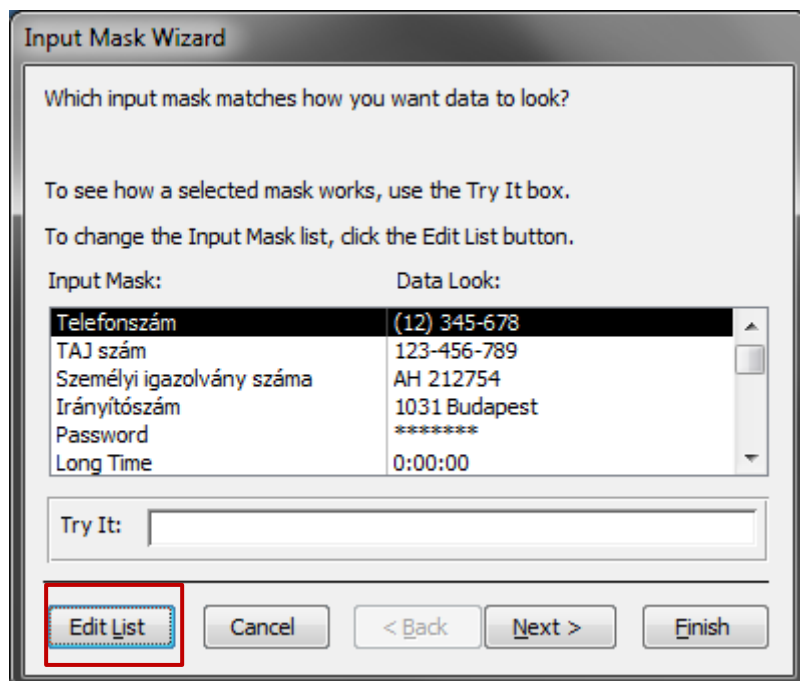
A példában esetünkben a *Hatarertek* tábla *Osztaly* mezőjében hozzunk létre beviteli maszkot! Tegyük fel, hogy itt olyan értékek szerepelhetnek, melyek két betűt, majd kötőjelet, végül számokat tartalmaznak (pl. AA-001). Ennek megfelelően az *Osztaly* mezőnk szöveg típusú.

Field Name	Data Type
Osztaly	Short Text
Ertek	Number

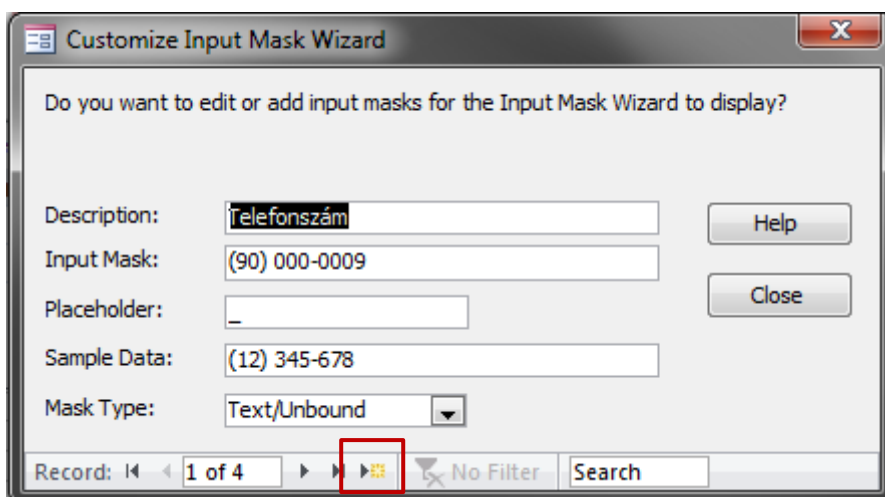
Field Properties

General	
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

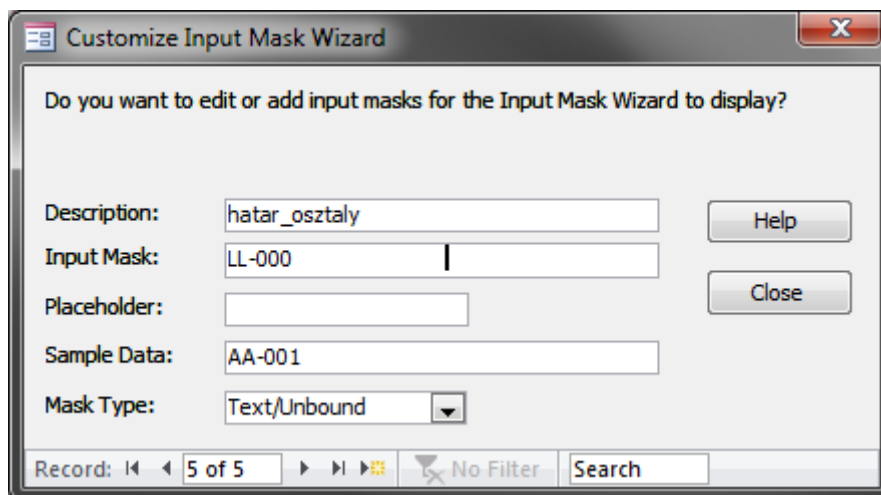
Kattintsunk a  gombra (lásd fenti ábra). Ekkor egy párbeszéd ablak nyílik meg (mentés szükséges).



Itt az „Edit List” gombra kattintva tudjuk szerkeszteni a maszklistánkat.

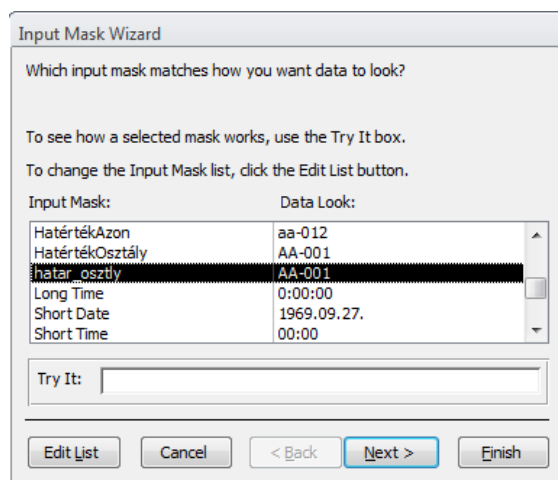


A fent piros négyzettel kijelölt gombbal tudunk új maszkot megadni. Ezt az alábbiaknak megfelelően töltjük ki.



A „**Description**” résznél nevezhetjük el a maszkunkat. Az „**Input Mask**” résznél definiálhatjuk a maszkunkat. Itt az „**L**” jelöl egy darab betűt, míg a „**0**” egy darab számot. Végül töltsük ki a „**Sample Data**” mezőt is, ahol egy minta adatot tudunk megadni, majd a 'Close' gombra kattintva zárjuk be a párbeszéd ablakot.

Ezután vegyük észre, hogy az „**Input Mask Wizard**” ablak listájába bekerült az általunk definiált új maszk. Ezt válasszuk ki, majd kattintsunk a „**Finish**” gombra. Ezzel definiáltuk a választott tábla mezőjén a beviteli maszkunkat.



A relációs séma diagram alapján hozzuk létre az összes táblát, valamint a mezőket, és azok típusait. Ahol indokolt, használjunk beviteli maszkot, érvényességi szabályt, és állítsuk be a tizedes jegyeket, valamint adjuk meg a kötelező mezőket. Ahol nem állítunk be ilyen tulajdonságokat, a bemutatás során szükséges lehet megindokolni!

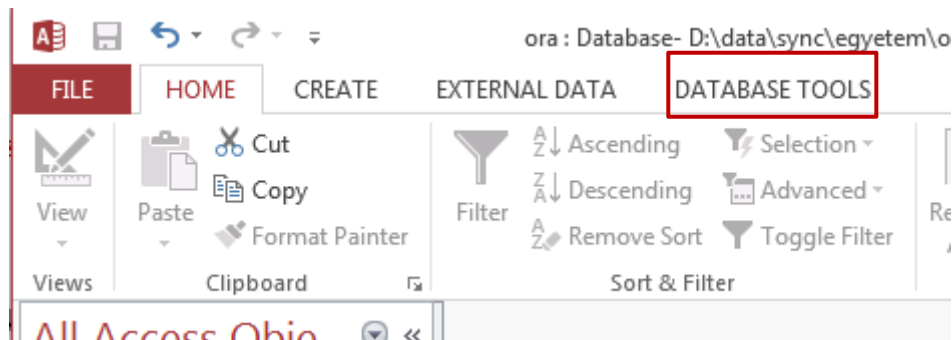
Kapcsolatok kialakítása



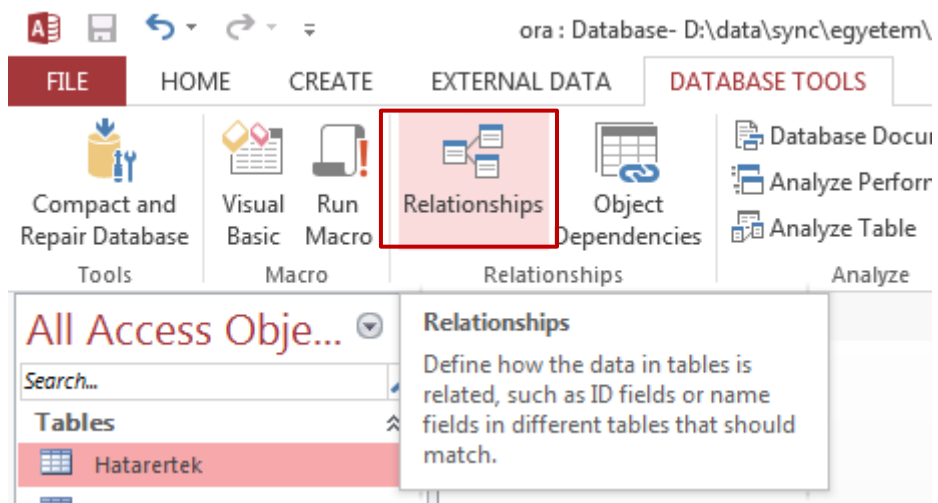
A kapcsolatok létrehozásához, szerkesztéséhez, törléséhez az összes tábla nézetét zárjuk be a korábban említett módon.

Kapcsolatok létrehozása

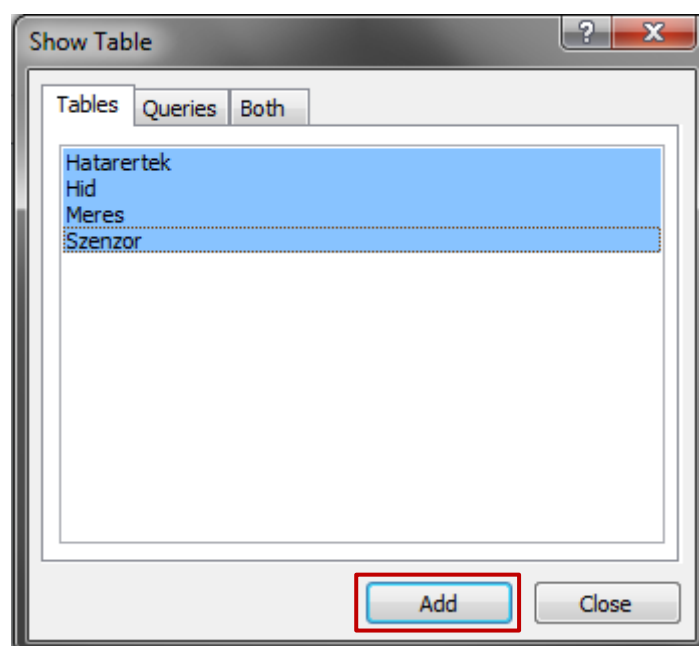
A kapcsolatok kialakítása az Access-ben grafikus módon történik. Ehhez kattintsunk a „Database Tools” fülre.



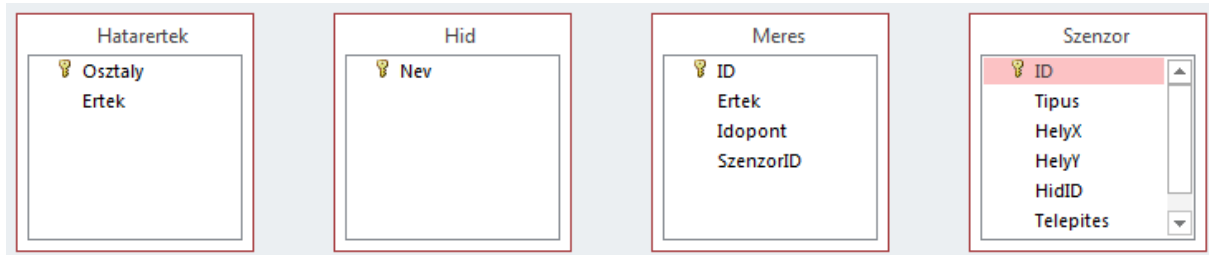
Ezután pedig a „Relationships” opcióra.



Ekkor az alábbi képernyőkép jelenik meg.



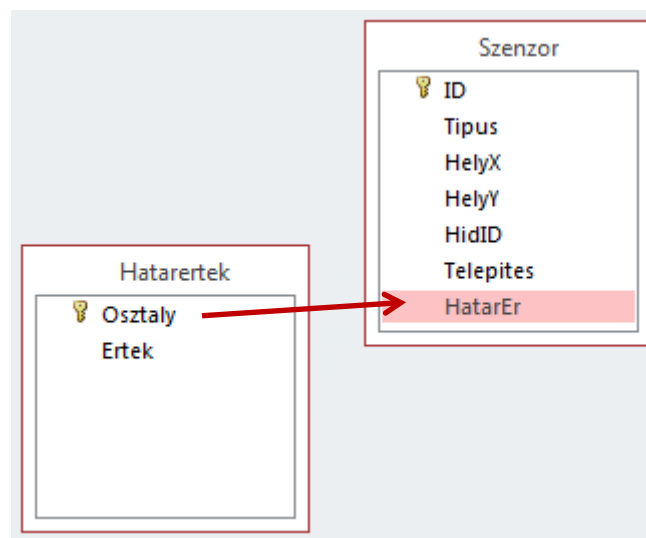
Itt egyenként adjuk hozzá az összes táblát (a nézeteket ne adjuk hozzá!) úgy, hogy jelöljük ki a táblákat (pl. SHIFT és kattintás segítségével, vagy egyenként a táblákra kattintva, majd az „Add” gombbal. Ezzel meg kell hogy jelenjenek a táblák, a „Close”-ra kattintva zárjuk be a párbeszédablakot.



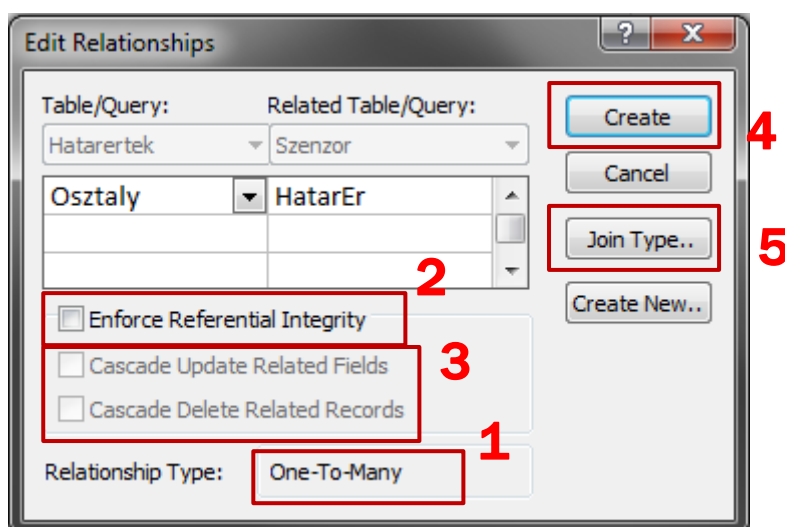
A kapcsolatok az 2. házi feladat alapján alakítsuk ki. Itt ténylegesen ki fog derülni, hogy a megoldásunk fizikailag megvalósítható-e. A kapcsolat kialakításához ragadjuk meg az egyik kapcsolódó mezőt (ha 1:N kapcsolatról van szó, akkor az elsődleges kulcsra) és húzzuk rá a másik kapcsolódó mezőre (drag-and-drop technika). Ezen kívül a táblák ablakait is a számunkra megfelelő elrendezésbe pakolhatjuk.



A drag-and-drop technikát az alábbi oldalon találhatsz bemutatót: http://hu.wikipedia.org/w/index.php?title=F%C3%A1jl:Drag_and_drop.gif&filetimestamp=20100219180442



Ha ezt megtettük megjelenik az „Edit Relationships” párbeszéd ablak.



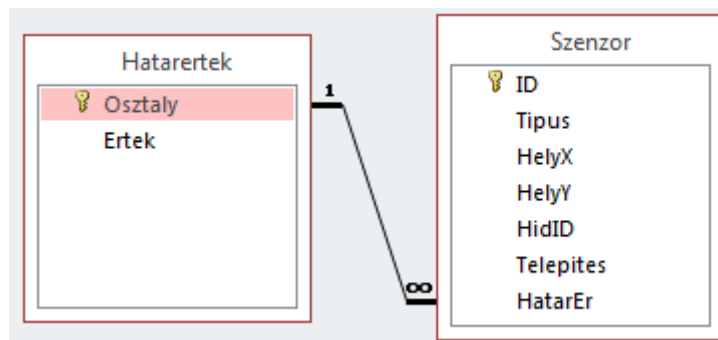
Itt a következő beállításokat tudjuk eszközölni:

1. Az Access számunkra megadja a **kapcsolat típusát**. Ezt úgy dönti el, hogyha az egyik kapcsolódó mező kulcs, akkor 1-több, ha mindkettő, akkor 1-1 kapcsolat lesz. Ha semelyik sem kulcs, akkor több-több kapcsolat kerül kialakításra. Ha az 2. házi feladatot helyesen oldottuk meg, akkor a relációs séma diagramon szereplő kapcsolati multiplicitás megfelelően tükrözi az itt megjelenőt.
2. A **hivatkozási integritást** legtöbb esetben be kell kapcsolnunk, amennyiben **nem tesszük meg kell indokolnunk!** Ez azt jelenti, hogy a hivatkozó táblában csak olyan értékek szerepelhetnek, amelyek a hivatkozott táblában is szerepelnek.
3. A **kaszádolt frissítés és törlés mezőket** is többnyire be kell kapcsolnunk (csak akkor érhető el, ha a hivatkozási integritást fentebb kikényszerítettük). Ez azt jelenti, hogy amennyiben a hivatkozott táblában egy adott mezőt egy konkrét rekordban töröljük, vagy megváltoztatjuk, akkor ez változást eredményez a hivatkozó táblában is, azaz a hivatkozott rekordban a mező is frissülni fog, vagy törlődni.
4. Végül a 'Create' gombra kattintva hozhatjuk létre a kapcsolatot.



Megemlítjük, hogy a „kapcsolódás” típusát is be tudjuk állítani ha szükséges. Amennyiben a „Join Type...” gombra kattintunk (5) akkor a megadott párbeszéd ablakban három típust adhatunk meg, az SQL-nek megfelelően ez a következő típusoknak felel meg: INNER JOIN, LEFT JOIN, OUTER JOIN. Az egyes típusok leírása megjelenik a párbeszéd ablakon, így ezt itt most nem részletezzük, valamint **később foglalkozunk vele**. Megjegyezzük, hogy a házi feladatban nagy valószínűséggel ezt nem kell alkalmazni, legfeljebb a lekérdezések esetén.

Az így kialakított kapcsolat a példa esetén a következő.



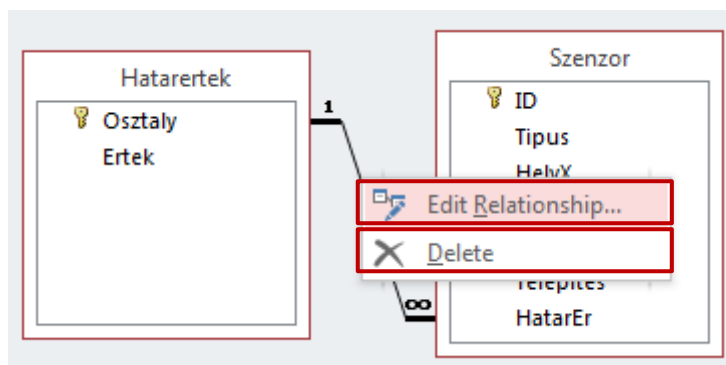
A végtelen jel itt megegyezik az 2. házi feladatban alkalmazott „n” jelöléssel.

Kapcsolatok szerkesztése



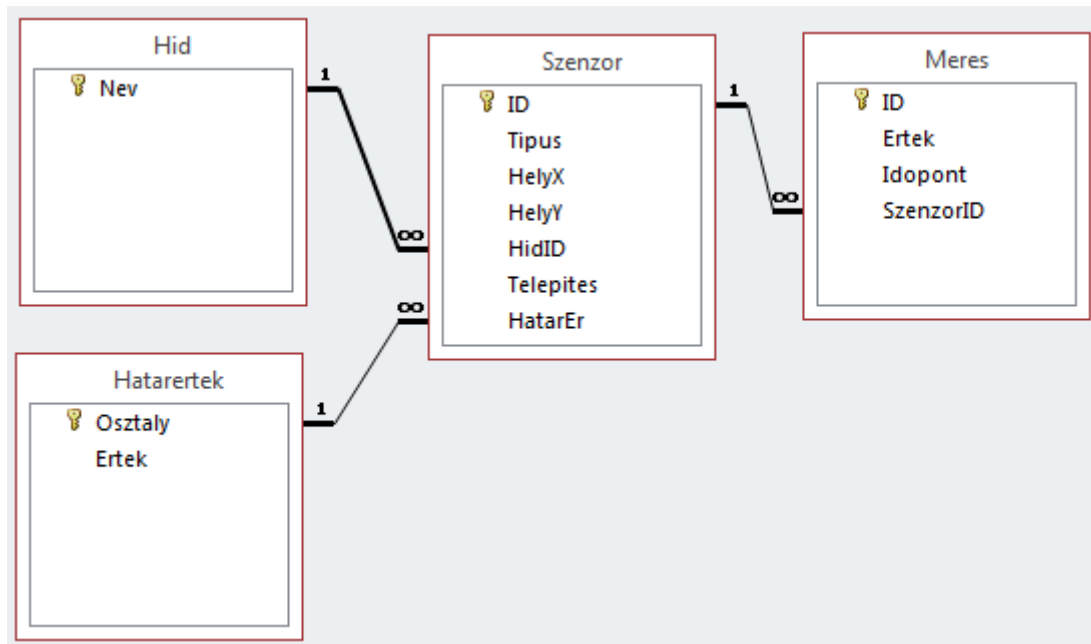
A kapcsolatok létrehozásához, szerkesztéséhez, törléséhez az összes tábla nézeteit zárjuk be a korábban említett módon.

A kapcsolatot a létrehozás után is tudjuk szerkeszteni úgy, hogy a kapcsolat felett jobb egérgombot kattintunk.



Itt kattintsunk az „Edit Relationship” gombra. A kapcsolat törlésére a „Delete” opcióra kattintsunk.

A példában a kialakított kapcsolatok az alábbi ábrán látható módon kerültek kialakításra.



A kapcsolatok kialakításakor fontos, hogy a hivatkozó és hivatkozott mezők típusaink egyeznie kell (Elsődleges kulcsnál AutoNumber, idegen kulcsnál „Number” és azon belül a „Field Size” „Long Integer” legyen). Amennyiben ez nem teljesül a kapcsolat nem jön létre!



Ha nem látszik a kapcsolatok esetén a multiplicitás (1:1, 1:∞, vagy ∞:∞), akkor nem állítottuk be megfelelően a hivatkozási integritásokat.



Minden kapcsolatot hozzunk létre a relációs séma diagram alapján az Ismertetet módon. Ügyeljünk oda a hivatkozási integritás beállítására.

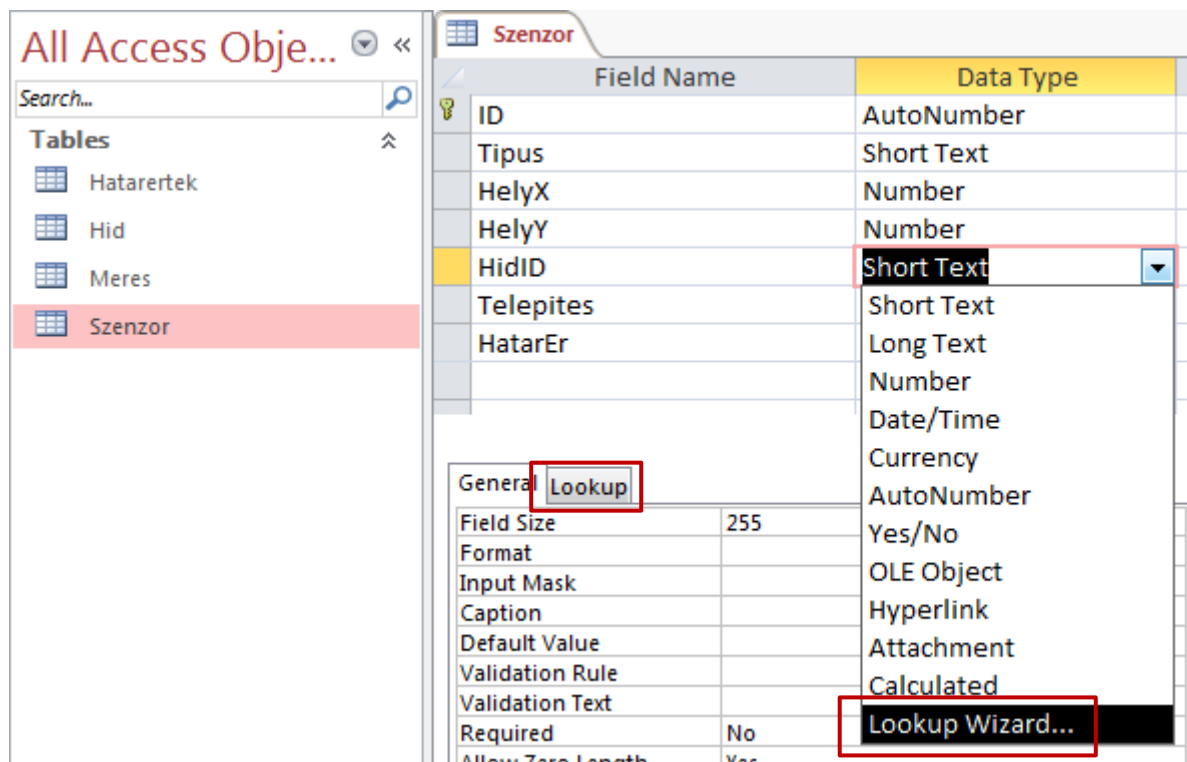
Kapcsolatok kialakítása Keresés varázsló segítségével

Bizonyos esetekben könnyebb és biztonságosabb feltöltést biztosíthatunk a „Keresés varázsló” segítségével. Ennek használatával az Access olyan tábla hivatkozást alakít ki, midőn a feltöltés során egy legördülő listából választhatjuk ki a megfelelő értéket. Ilyen kapcsolat kialakítását ún. *domain* táblákra történő hivatkozás esetén érdemes használni. A *domain* táblák olyan táblák, melyekben a rekordok a rendszer teljes működése során soha, vagy csak nagyon ritkán kerülnek megváltoztatásra. Tehát a tábla rekordjai rögzített értékeknek tekinthetők, melyeket a specifikáció során adunk meg. Ez származhat szabványból, speciális szakismeretből, stb. Példánk esetében ilyen táblának tekinthetők a *Hid* és a *Hataretek* táblák.




Azon mezők között, ahol ilyen kapcsolatot szeretnénk kialakítani először **törölnünk kell** a már meglévő kapcsolatot! Ennek módját [fentebb](#) találod.

Ezután a megadott hivatkozó tábla tervező nézetét nyissuk meg. Ennek módja [korábban](#) már ismertetésre került.

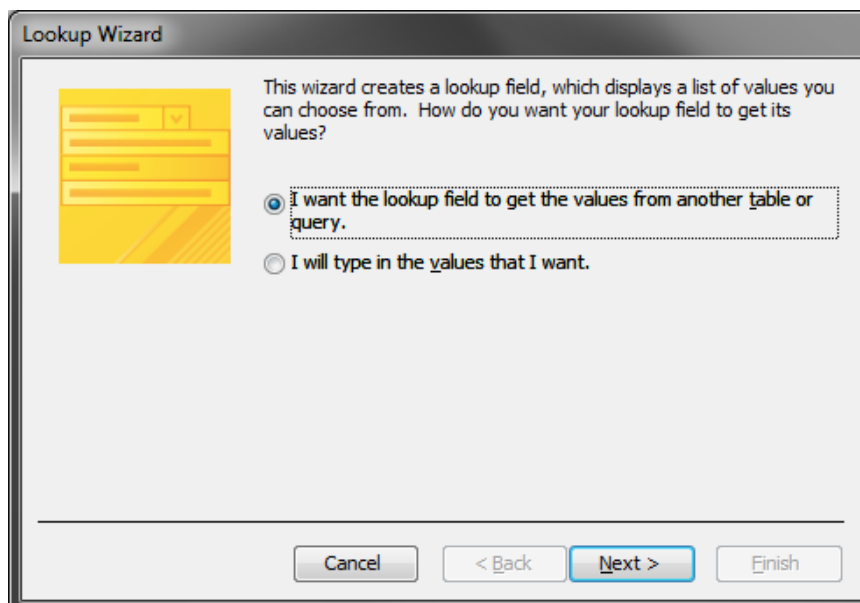
A példában ilyen kapcsolatot alakítunk ki a *Híd* és *Szenzor* táblák között. A kapcsolat megvalósítható üres táblák esetén is, de a létrehozás folyamatában segíthet, ha a hivatkozott tábla (esetünkben a *Híd* tábla) tartalmaz adatokat.



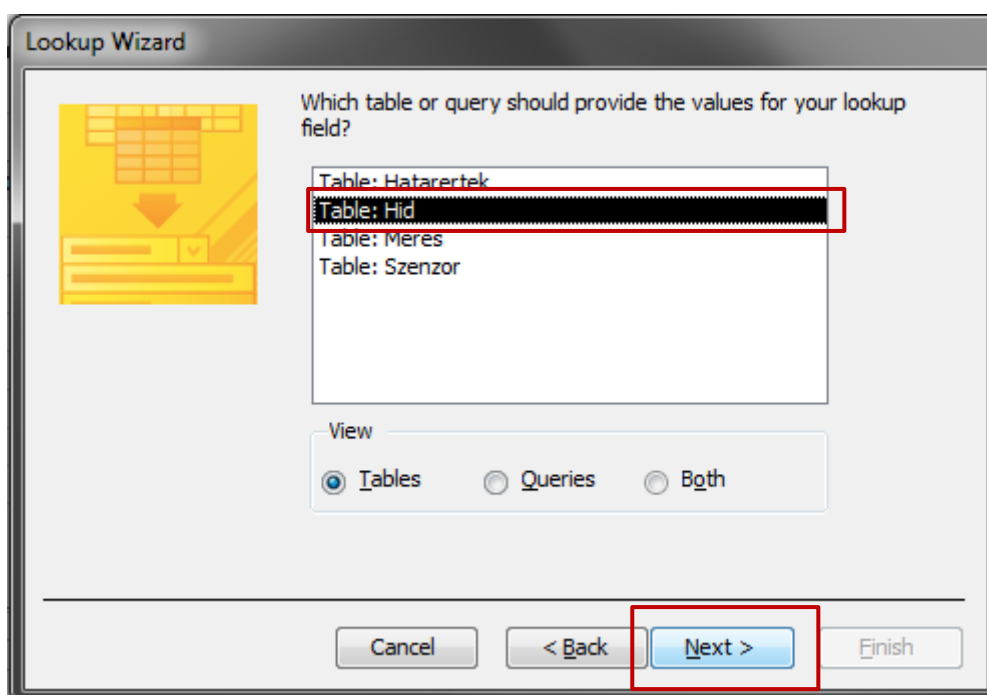
A tervezési nézeten a hivatkozó mező adattípusánál válasszuk ki a „**Lookup Wizard**” opciót.

- 
Fontos, hogy mielőtt a „Lookup Wizard” opciót választjuk, előtte már a mező típusát válasszuk ki. Ugyanis oda kell figyelünk, hogy a hivatkozó és hivatkozott mezők típusaink egyeznie kell. Amennyiben ez nem teljesül a kapcsolat nem jön létre.
- 
A „Lookup” mező módosítása az attribútum típus részletes beállítása melletti „Lookup” lapfülre kattintva tehető meg. Annak teljes törlése a „Display Control” mező „Text Box”-ra állításával valósítható meg.
- 
A kapcsolatok létrehozása előtt célszerű a hivatkozott táblákat adatokkal feltölteni!

Ezután elindul a varázsló.



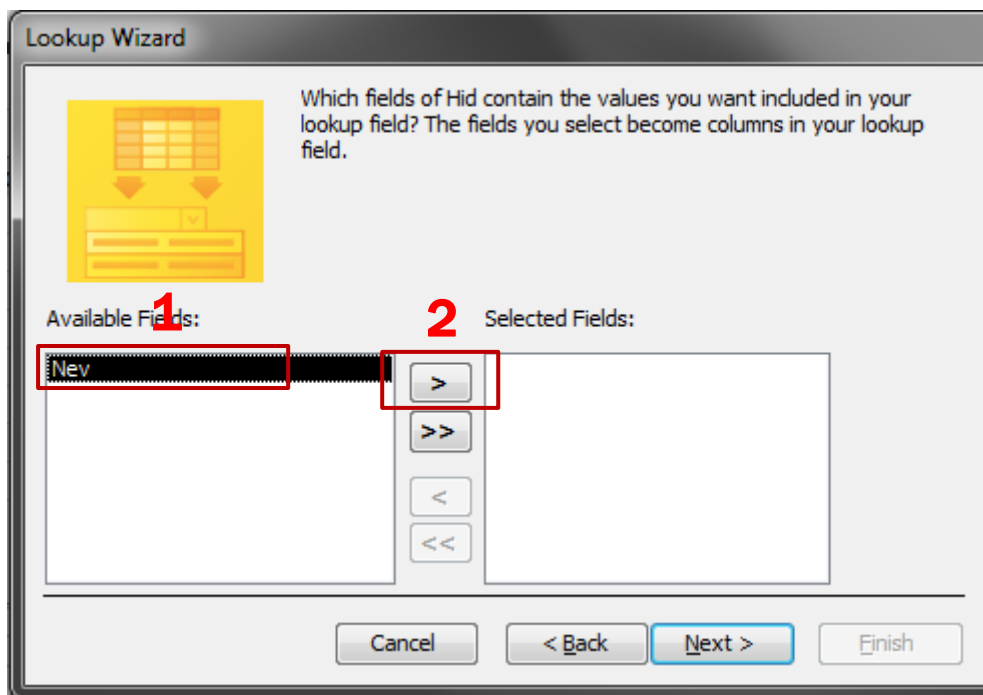
Mivel értékeket más táblából vennénk (táblából hivatkozunk), válasszuk az első opciót, majd kattintsunk a „Next” gombra.




Válasszuk ki, melyik táblával szeretnénk kapcsolatot létesíteni, majd kattintsunk a „Next” gombra.



Lehetőségünk van hivatkozást létrehozni lekérdezésre is. Ehhez a „Queries” rádió gombot is jelöljük meg.



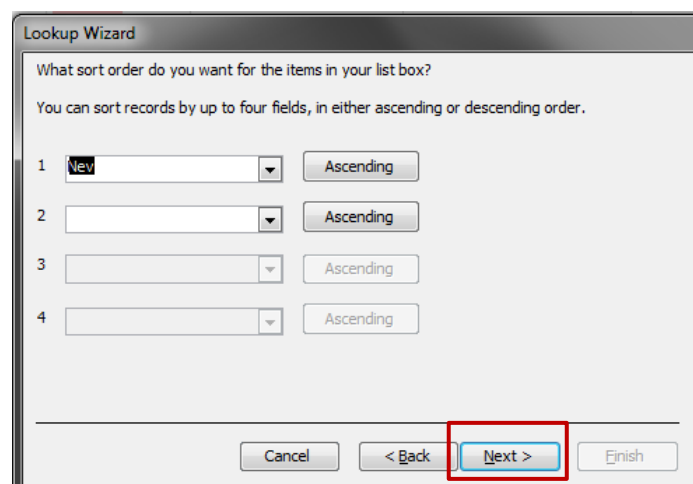
A következő lépésben azokat a mezőket tudjuk megadni, melyek a legördülő listában is szeretnénk szerepeltetni. Ennek módja a következő:

1. Válasszuk ki a mezőt (mezőket), majd
2. kattintsunk a  gombra.

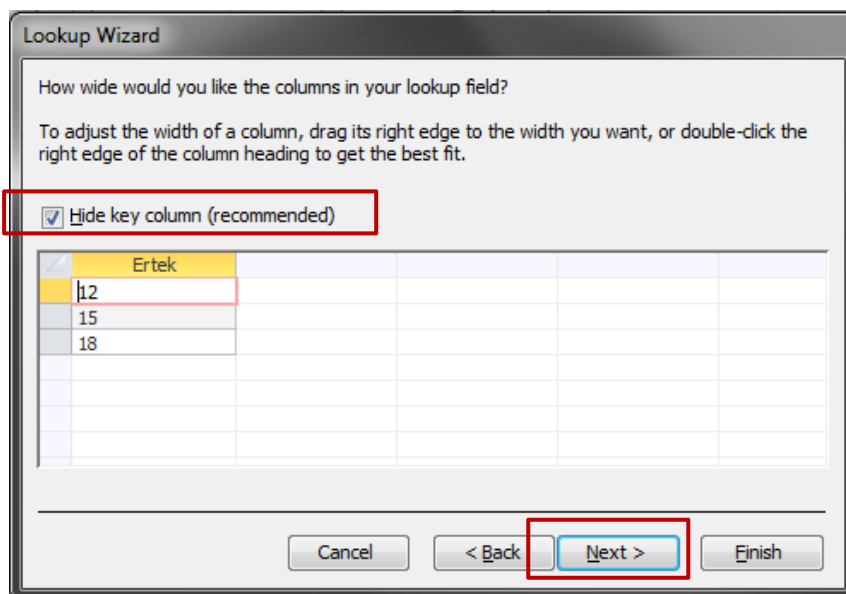
Ha kiválasztottunk minden mezőt, a „Next” gombra kattintva lépünk tovább.



A kulcs mezőt is adjuk hozzá.

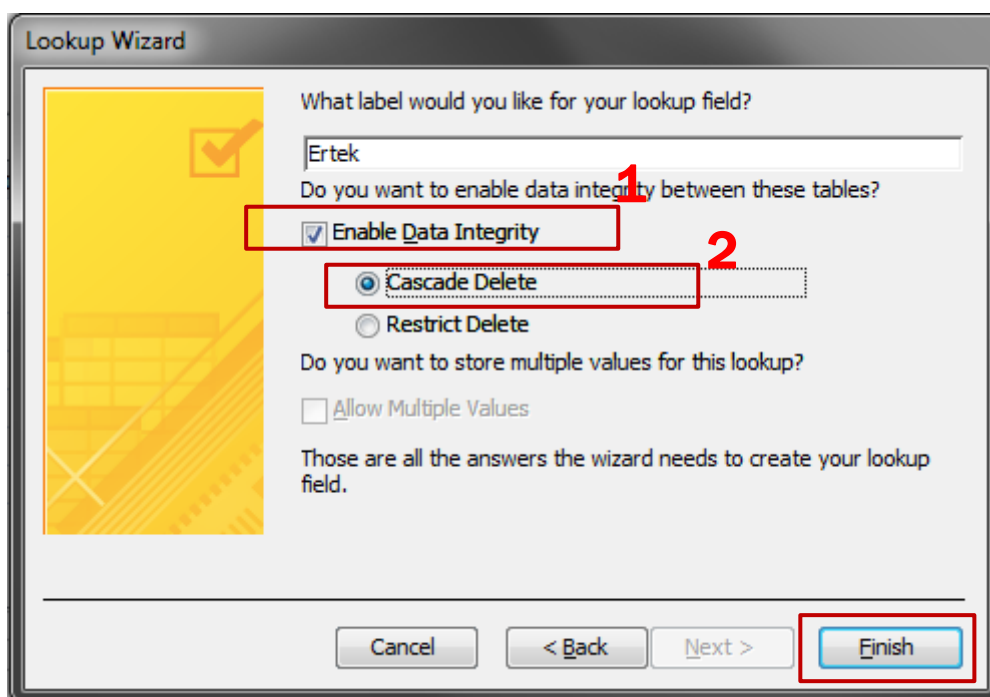


A következő lépésben az adatok rendezésének sorrendjét adhatjuk meg. Válasszunk ki tetszőleges rendezési sorrendet, majd kattintsunk a „Next” gombra.



A következő lépésben láthatjuk a legördülő listában megjelenő elemeket. Ha a hivatkozott táblába már töltöttünk fel adatokat, akkor azok listája itt megjelenik. Amennyiben van kulcs attribútum, akkor azt tüntessük el a „Hide key column” választó segítségével.

A fenti ábra egy másik kapcsolatot mutat, melyet a *Hatarertek* és *Szenzor* között hozzunk létre. Ebben az esetben azért váltottunk, hogy a fenti opciót megmutassuk, hiszen a *Híd* és *Szenzor* közötti kapcsolat esetén csak egy mező szerepel a legördülő listában. Ha végeztünk kattintsunk a „Next” gombra.

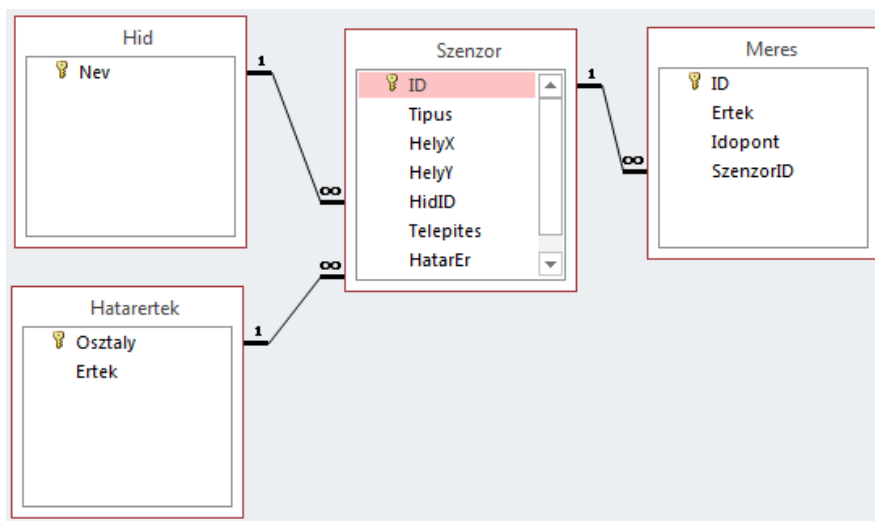


Végül állítsuk be a

1. hivatkozási integritást, és a
2. kaszkádolt törlést.

Ennek értelmét megtalálhatjuk [fentebb](#). A varázslót a „Finish” gombbal zárhatjuk be, ezzel a kapcsolat létrejön.

Amennyiben [megnyitjuk](#) a kapcsolatokat, láthatjuk, hogy a korábban törölt kapcsolatok ismét megjelentek.



Az adat feltöltésnél pedig nézzük meg, hogy a legördülő menü megjelent.

ID	Tipus	HelyX	HelyY	HidID	Telepites	Ha
*	(New)	0	0			
				Erzsébet híd		
				Lánchíd		
				Megyeri híd		

Természetesen ahhoz, hogy értékek jelenjenek meg, a hivatkozott táblát ki kell tölteni. Az adatfeltöltésről [lentebb](#) olvashatsz.



Vizsgáljuk meg kapcsolatainkat, és ha valamely táblát *domain* táblának értékelünk, akkor az ő kapcsolatait változtassuk meg a Keresés varázslónak megfelelően. A házi feladatnak legalább egy ilyen kapcsolatot tartalmaznia kell!



Amennyiben a kapcsolat multiplicitása nem leolvasható a diagramról, akkor az integritás beállítása elmaradt, ezt javítani szükséges!

General	Lookup
Display Control	List Box
Row Source Type	Table/Query
Row Source	SELECT cegek.nev, cegek.cid FROM cegek;
Bound Column	2
Column Count	1
Column Heads	No
Column Widths	
Allow Multiple Values	No
Allow Value List Edits	No
List Items Edit Form	
Show Only Row Source V	No

A Lookup mező (legördülő mezőben való megjelenítés) utólag is létrehozható, finomhangolható a „Lookup” fülön. A választó listát feltölthetjük SQL lekérdezés eredményeivel, ami különösen nagy segítség, ha az idegen kulcs által hivatkozott tábla egyéb mezőit szeretnénk megjeleníteni. Ebben az esetben fontos a lekérdezést úgy megfogalmazni, hogy a kiírandó mező szerepeljen először, és utána a hivatkozott elsődleges kulcs. A „Bound Column” értéke legyen 2, míg a „Column Count” legyen 1.

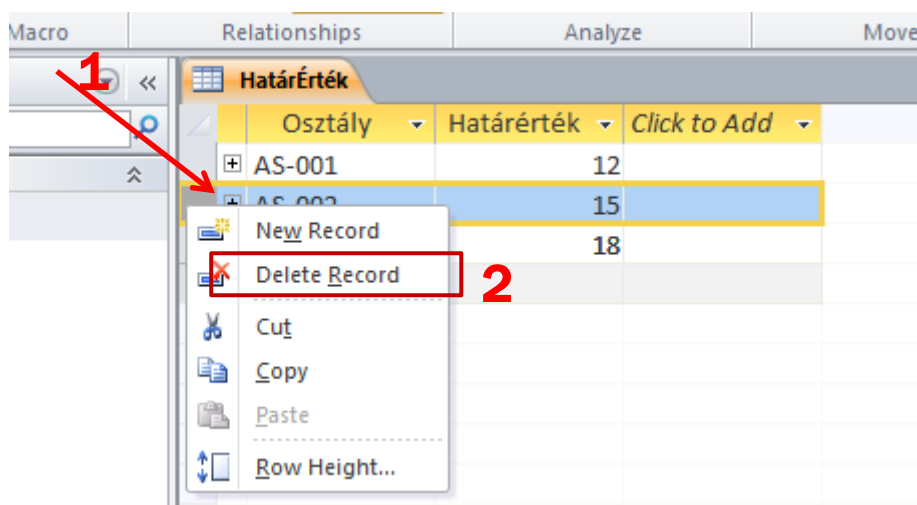
Adatok feltöltése és törlése




Az adatok feltöltése adatlap („Datasheet View”) nézetben történik. Itt egyszerűen fel lehet vinni az egyes adatokat. Ennek megnyitásáról [korábban](#) már volt szó.

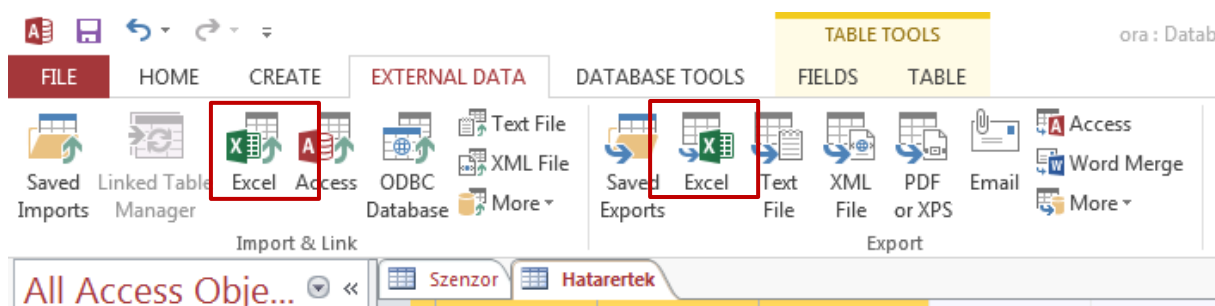
Szenzor		Hataretek	
	Osztaly	Ertek	Click to Add
+	AS-001	12	
*	-	0	


Adatokat úgy törölhetünk, hogy

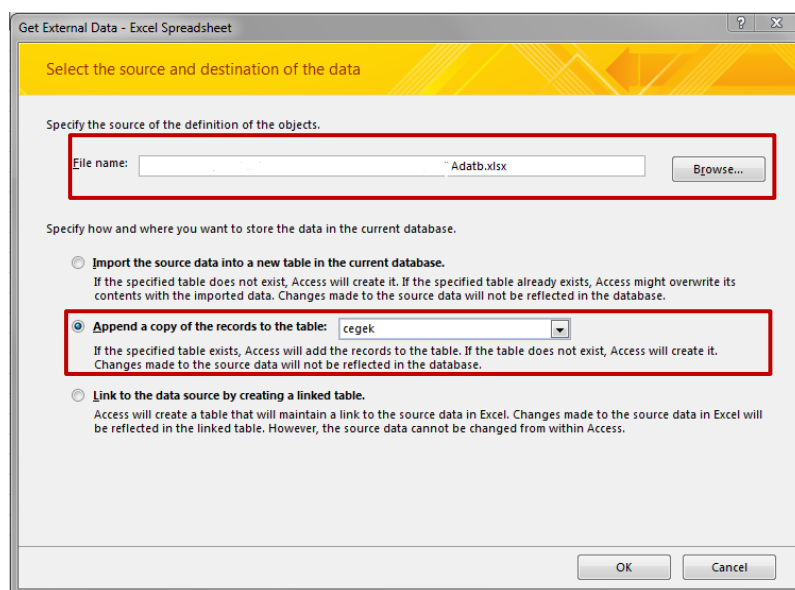
1. az adatlap nézet szélső sávjára bal gombbal kijelölünk egy vagy több sort
2. majd jobb gombbal a „Delete Record” parancsra kattintunk.



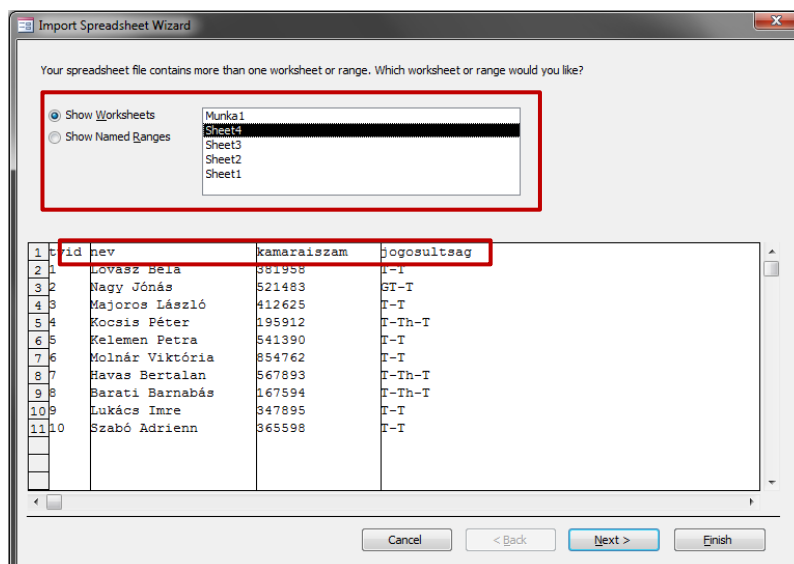
-  Ezen kívül lehetőségünk van adatok importálására és exportálására Excel fájlokból, ill. fájlokba. Ezeket az opciókat az „External Data” fülön érthetjük el. Az adatok átalakításhoz további tanácsok találhatóak a Lekérdezések fejezet végén is!
-  Amennyiben Excelből importálnánk adatot, és ott már létrehoztuk az elsődleges és idegen kulcsokat is a táblákon, akkor importálás előtt mindenképpen érdemes az adatokat ellenőrizni. Az idegen kulcsok létezését a hivatkozott tábla elsődleges kulcsai közt ellenőrizni kell, ezt a VLOOKUP Excel függvénnyel egyszerűen meg lehet tenni (Lásd Idegen kulcsok előállítás Excel segítségével). Ezen kívül az elsődleges kulcsok oszlopán az egyediséget is ellenőrizni kell, ismétlődő elsődleges kulcs esetén az import folyamata hibát fog okozni és komoly fejtörést okozhat utána a folytatás. (Lásd AutoNumber visszaállítása fejezet)
-  Sajnos a Windows 10 és magyar területi beállítások használata esetén az Ms Access 2016 több helyen is lehetetlenné teszi a cél megvalósítását. Legtöbbször Dátum jellegű adatok importálása esetén. Ilyenkor egyetlen megoldás a területi beállítások megváltoztatása (vagy ilyen beállításokkal rendelkező gép használata).



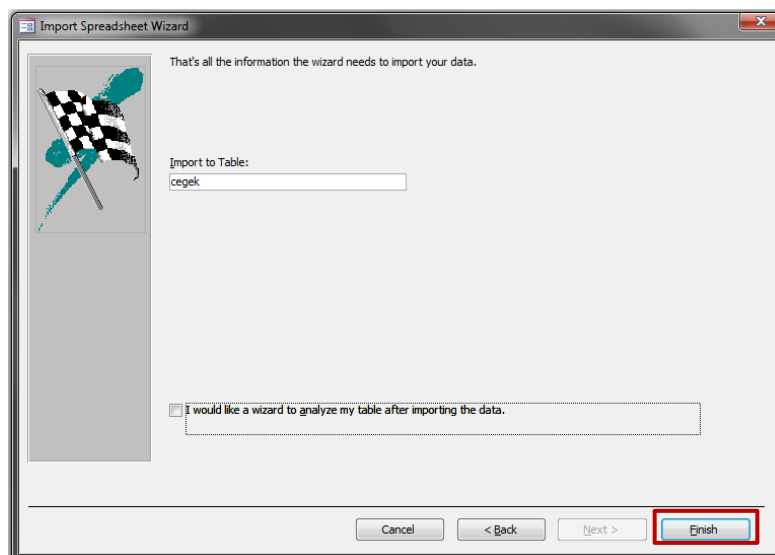
-  Töltsük fel adatokkal az adatbázist. A speciálisan „rögzített” értékeket tartalmazó táblákba (domain táblák) min. 3 rekord legyen, míg a használat során többet használt táblákba min. 5 rekord kerüljön. Az első csoportba például a példánkban a Híd és Hatarertek táblák tartoznak, míg az utóbbi csoportba a Szenzor és Meres tábla.



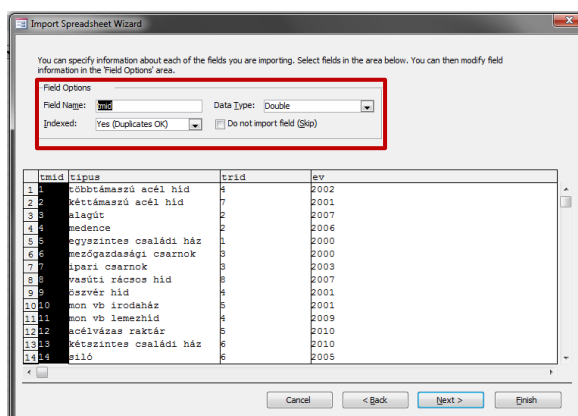
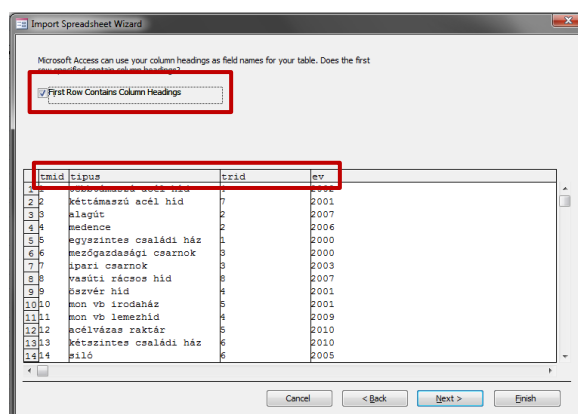
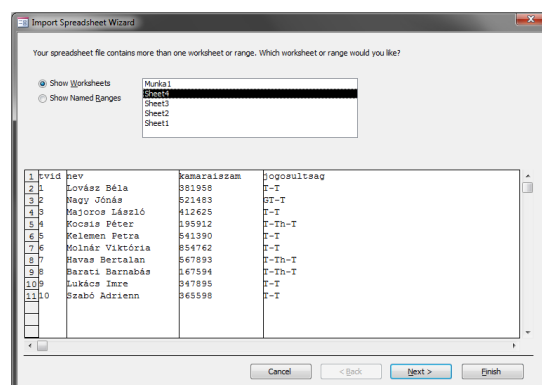
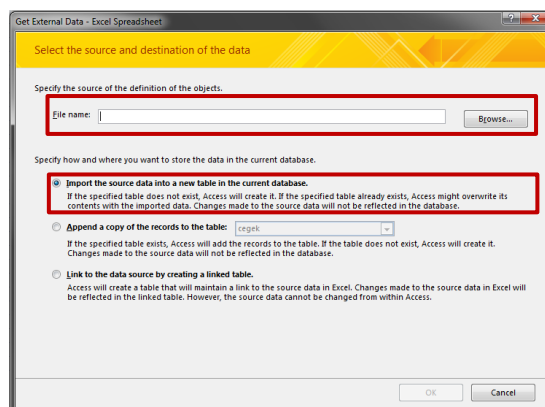
Az Excel fájl kitallózását követően adjuk meg, hogy a már meglévő táblát szeretnénk feltölteni adattal, válasszuk ki a megfelelő táblát is.

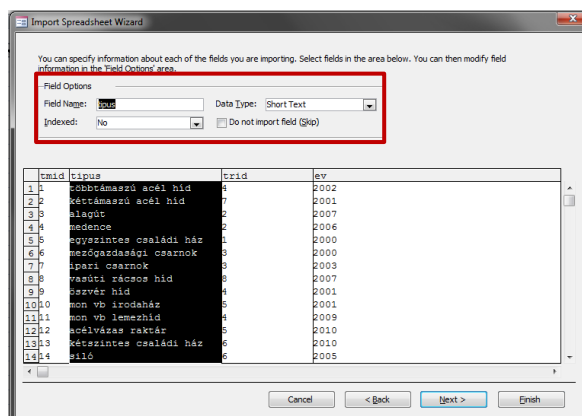


Figyeljünk oda, hogy az Excel táblában a mezőfejlécek teljesen megegyezzenek az Access tábla mezőneveivel!

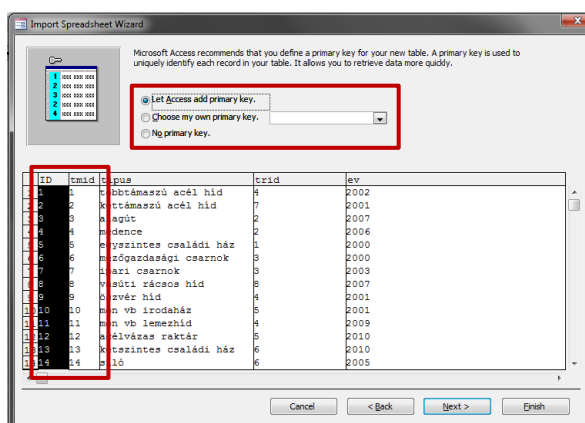


Ha a nyers adatokat az Access-ben szeretnénk átalakítani, arra is lehetőség van. Ennek az eljárásnak fő előnye, hogy a hivatkozott elsődleges kulcsok azonosítóit nem kézzel kell kikeresni, ezért a személyi hibát elkerülhetjük. Az előbb leírtakhoz hasonlóan töltjük be az Excel táblánkat, de most válasszuk az új tábla létrehozását. A létrehozott tábla csak egy köztes lépés lesz, ezt fogjuk adatforrásként felhasználni a megfelelően normalizált tábla struktúra feltöltésére lekérdezések segítségével.

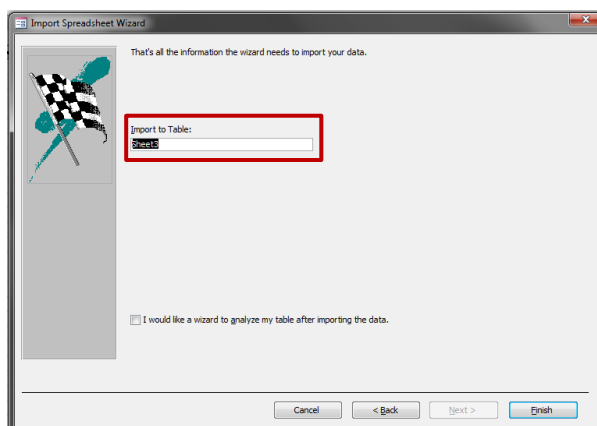




Minden oszlopra állítsuk be a megfelelő adattípust!



Válasszuk ki, hogy akarunk-e automatikusan létrehozni azonsító számot, vagy a forrásadatokban ez már megtalálható. Utóbbi esetben válasszuk ki a megfelelő mezőt elsődleges kulcsnak.



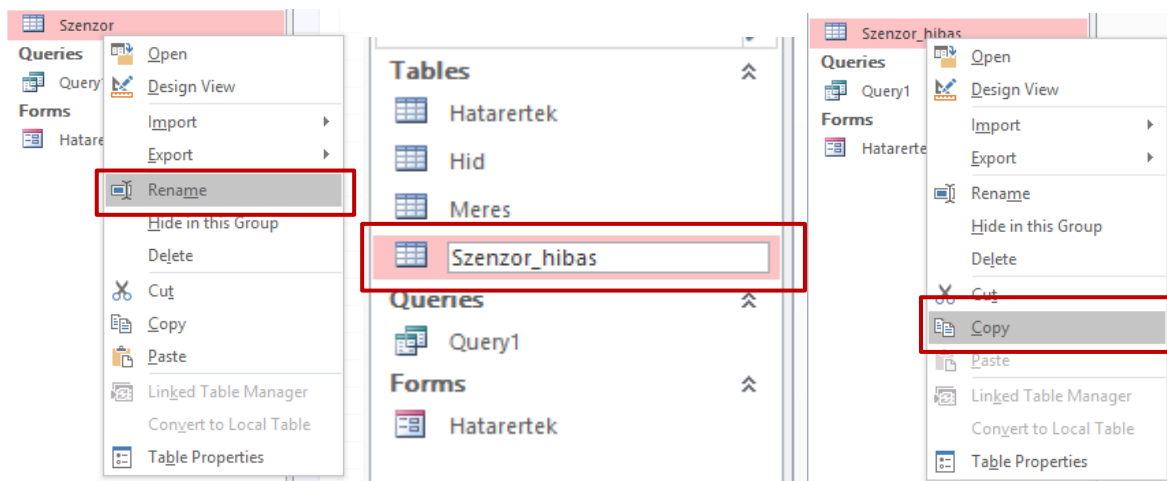
Adjuk meg a tábla nevét. Érdemes arra figyelni, hogy beleírjuk a névbe, hogy ez az ideiglenes tábla, melyet az adatok átszervezéséhez használunk. Az adat átalakítás után ne felejtsük el az ideiglenes táblát törölni. Az adat átalakítást a lekérdezések menüpont végén tárgyaljuk.

AutoNumber visszaállítása

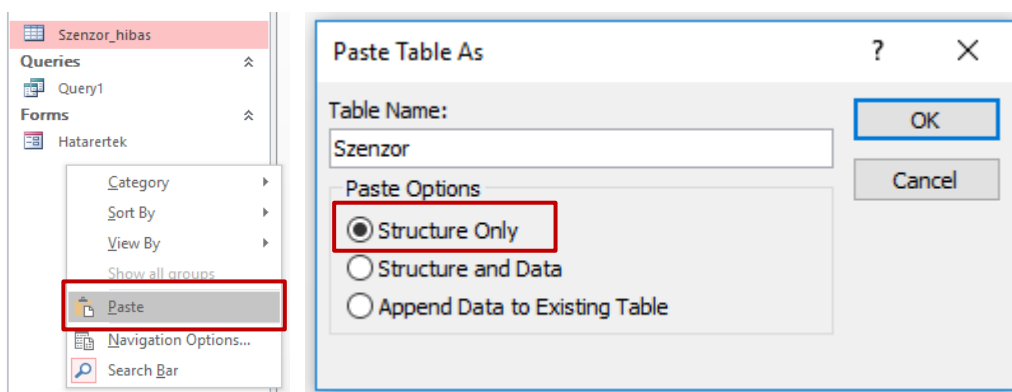
Adatfeltöltéskor ha valamilyen okból hibás adatot vittünk fel és azt a sort töröltük és az elsődleges kulcs adattípusa AutoNumber, akkor a törölt érték elsődleges kulcsa többé nem használható fel. Ekkor az ezen táblára hivatkozó tábláknál hivatkozási hiba léphet fel. Sajnos az Access csak

kerülőúton ad lehetőséget az adott kulcsérték újra használására. Egy új táblát kell létrehoznunk azonos struktúrával, melynek legegyszerűbb módja a következő:

Az elcsúszott azonosítókat tartalmazó tábla nevére jobb egérklikkel kattintva válasszuk a „Rename” menüelemre kattintva nevezzük át a hibás táblát. Majd hasonlóan a táblára jobb egérgombbal kattintva másoljuk a táblát a vágólapra.



Ezt követően a táblák alá, az üres területre kattintva jobb egérgombbal, szűrjük be kattintva a vágólapra másolt táblát. A megjelenő ablakon állítsuk be a tábla eredeti nevét, majd válasszuk a „Structure Only” opciót, ami csak a tábla szerkezetet másolja, a hozzá tartozó adatokat, így az AutoNumber értékét se.



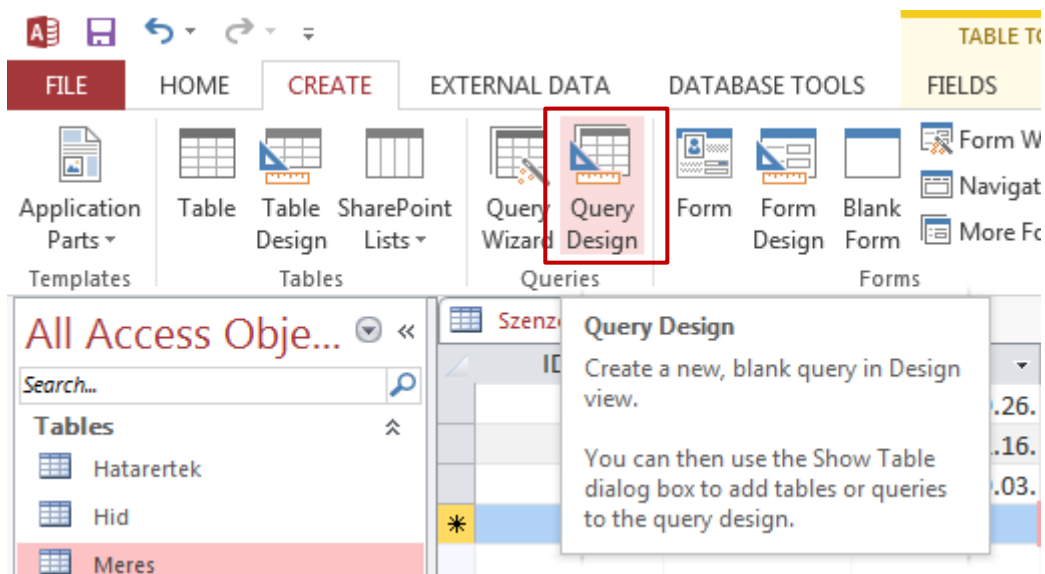
Figyeljünk oda, hogy az így létrehozott új táblán nincsenek beállítva a kapcsolatok, se az adatintegritási beállítások, ezeket tegyük meg!

Lekérdezések

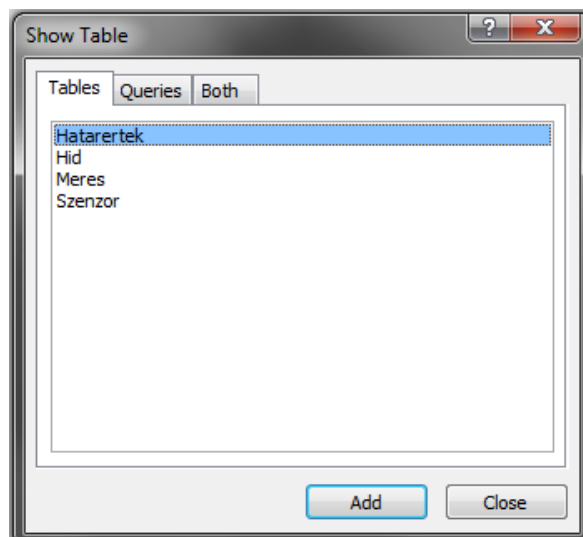
Lekérdezés létrehozása

A következőekben lekérdezéseket fogunk végrehajtani a korábban létrehozott adatbázison. A lekérdezéseket grafikus felületen adhatjuk meg, így nincs szükség az SQL nyelv ismeretére. Ennek ellenére a lekérdezések SQL nyelven is megfogalmazhatóak.

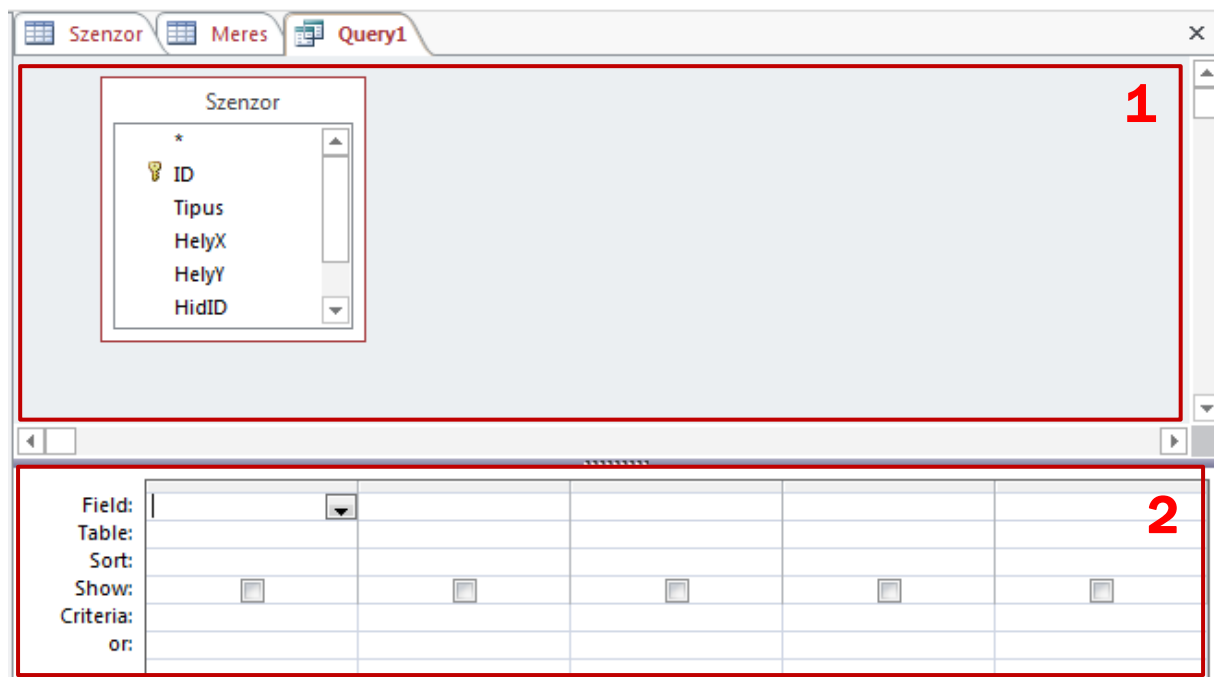
Új lekérdezés létrehozásához kattintsunk a „Create” fül „Query Design” opciójára.



Ekkor megjelenik egy, a kapcsolatoknál megismert ablakhoz hasonló felület.



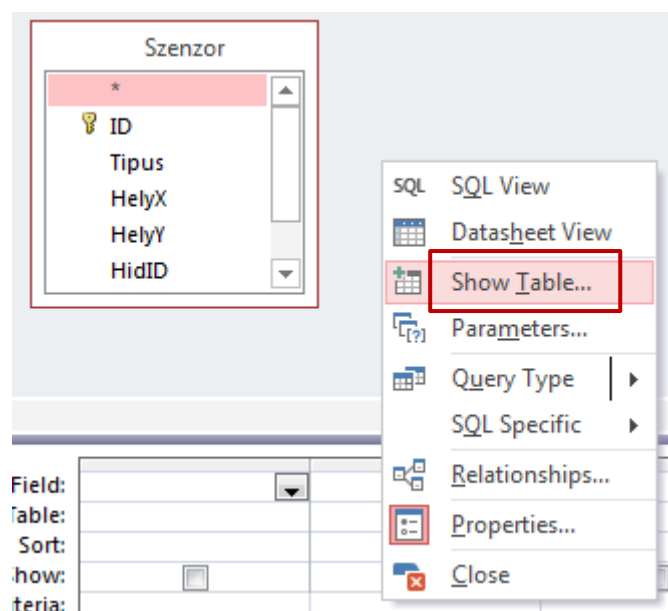
Itt a [korábban](#) ismertetett módon adjuk hozzá azon táblákat, amelyeken a lekérdezéseket végre akarjuk hajtani. A kapott felület a következő módon néz ki.



A felület a következő elemekből áll:

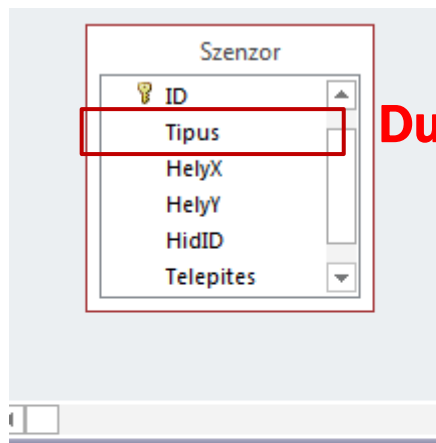
1. **Táblák és kapcsolataik:** A táblák kapcsolatai a korábban kialakítottaknak felelnek meg. A lekérdezéshez átalakíthatjuk a kapcsolatokat a [korábban](#) ismertetett módon.
2. **Lekérdezés tervező:** Itt adhatjuk meg a lekérdezéseinket.

Új táblát a felületünkhöz a Táblák és kapcsolataik részben jobb egérgattintással, majd a „**Show Table...**” parancs segítségével érhetjük el.



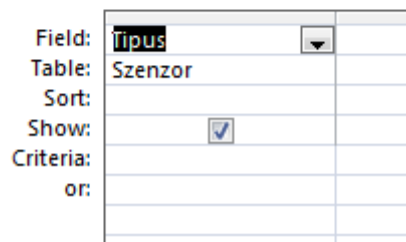
A lekérdezés tervező

Egy új oszlop felvételét a Lekérdezés tervezőben a mezőre (attribútumra) való dupla kattintással tehetjük meg.



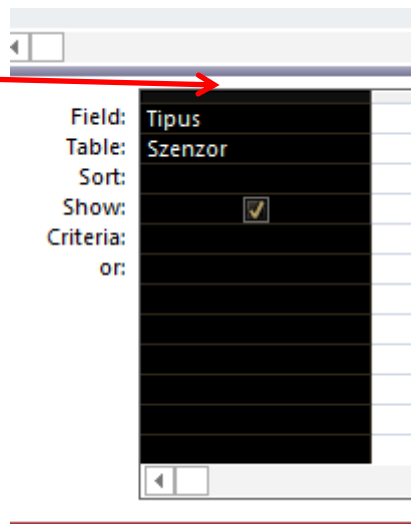
Dupla kattintás

Megjelenik



Egy ilyen mezőt törölhetünk is, úgy hogy kijelöljük majd a „DEL” gombot megnyomjuk.

Ide
kattintani,
majd DEL



A lekérdezés tervező elemei a következők.

Field:	ID	SzenzorID			
Table:	Szenzor	Mérések			
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:					
or:					

- **Field:** Mező (attribútum) neve, melyen az adott oszlopban hivatkozunk.
- **Table:** A tábla, ahol a mező (attribútum) szerepel.

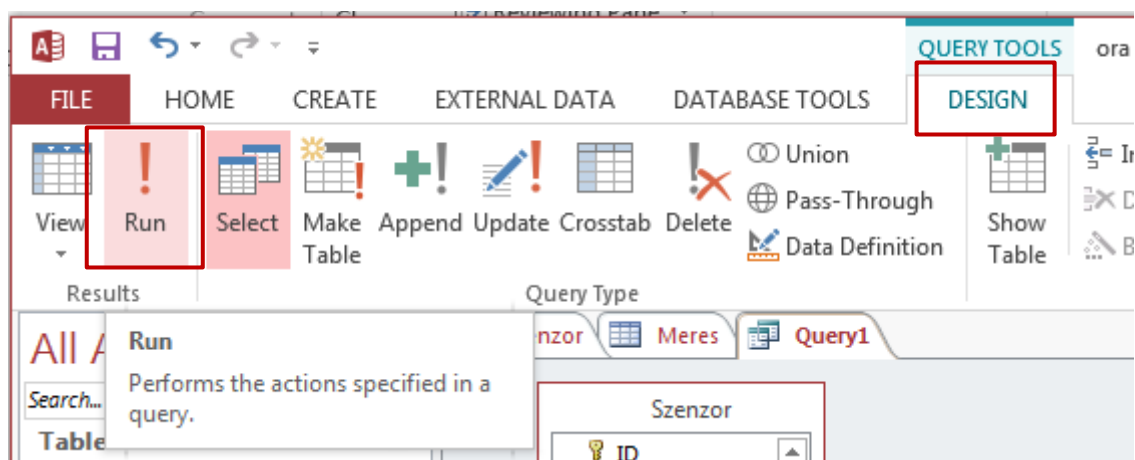
- **Sort:** Rendezési módot adhatunk meg.
- **Show:** Az oszlop a lekérdezés után megjelenjen-e. Gyakran hasznos lehet, ugyanis lehetnek olyan lekérdezések, melyek során az adott mezőt használjuk egy feltételben, de nem akarjuk azt megjeleníteni.
- **Criteria:** Feltételeket adhatunk meg.



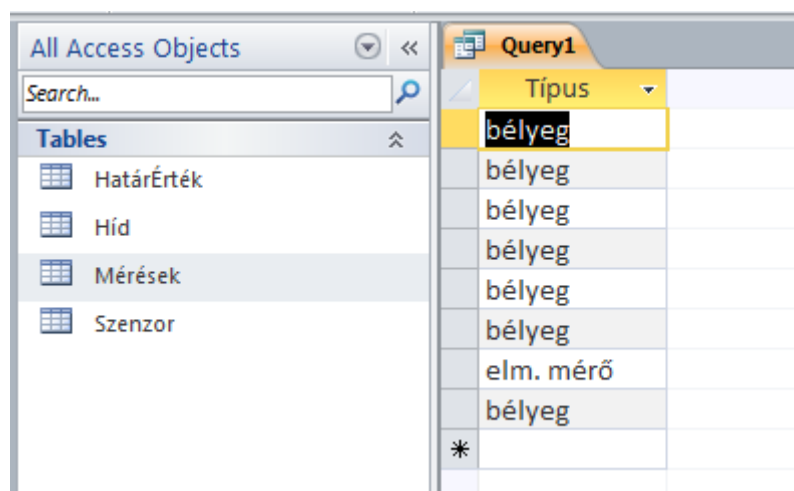
A később bemutatott példák segítik a lekérdezés tervező működésének megértését.

Lekérdezés futtatása

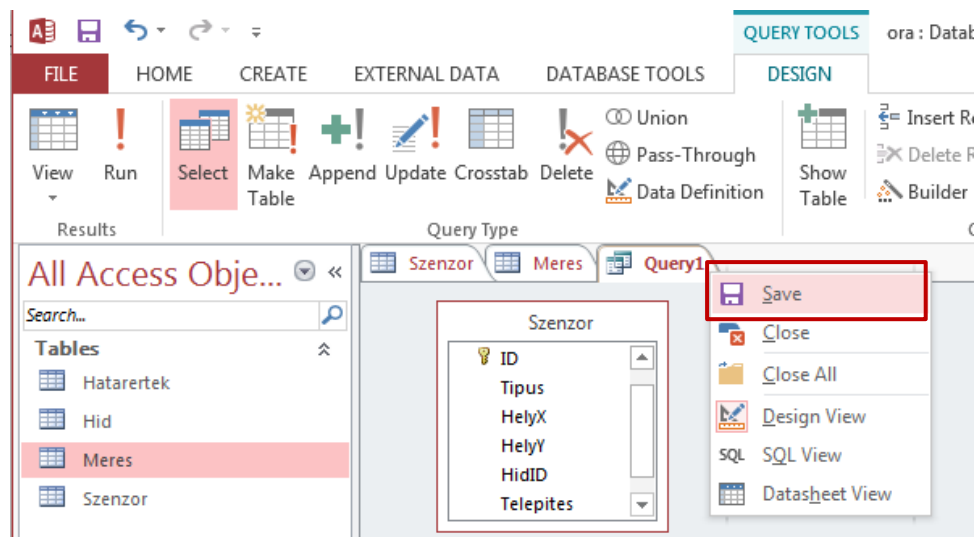
Miután lekérdezéseinket létrehoztuk, futtatni kell azokat. Ezt a „Design” fül futtatás opciójával tehetjük meg.



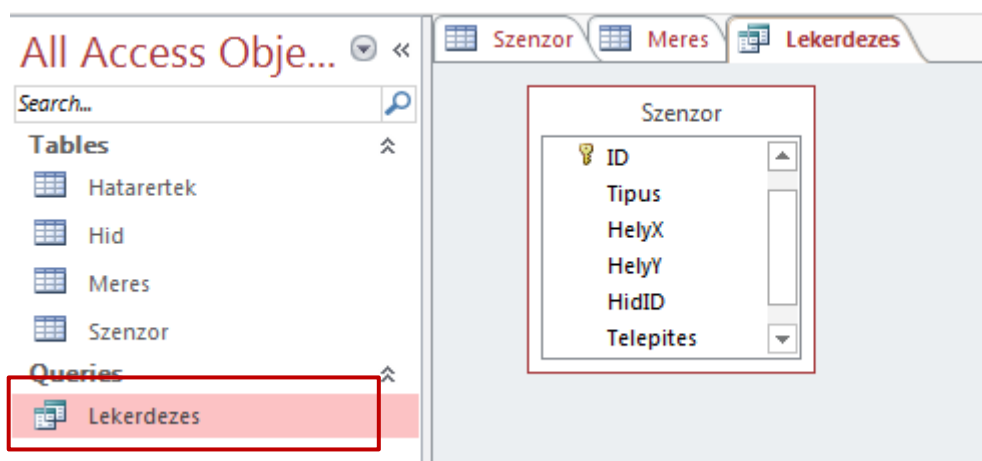
Ezután megjelenik a lekérdezésünk eredménye, ha nem vétettünk valamilyen hibát.



Lekérdezésünket el is menthetjük a fülre történő jobb egérgomb kattintással, majd a „Save” opció választásával.



A mentés után a bal oldali menüben megjelenik a lekérdezésünk.

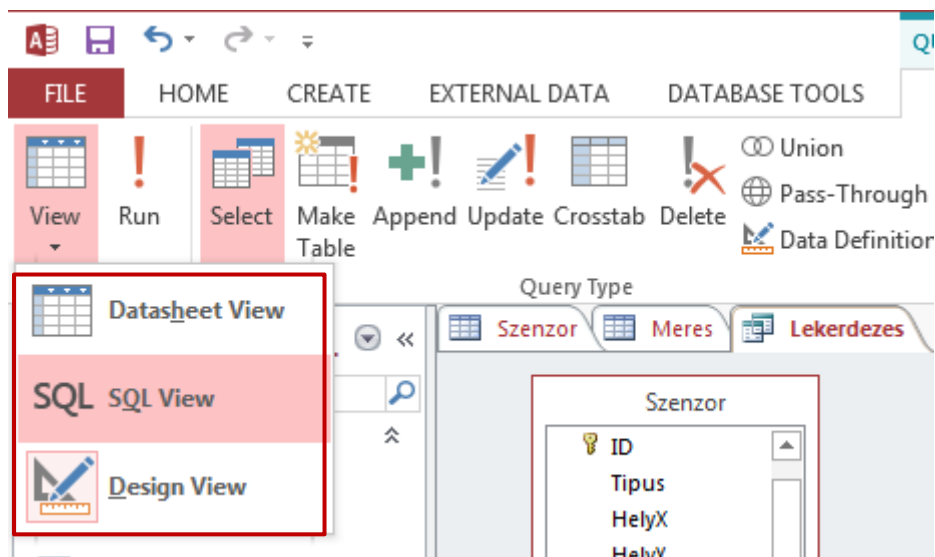


A bal oldali menüben történő dupla bal egér gomb kattintással is tudjuk futtatni lekérdezésünket.

A következő részben a relációs algebrának megfeleltetjük lekérdezéseinket a lekérdezés tervezőben.

Lekérdezés nézetek

A táblákhoz hasonlóan a lekérdezések is rendelkeznek nézetekkel. Ezeket ugyanúgy érhetjük el, mint a táblák [esetén](#).



Lehetőségek:

- **Datasheet View:** adatlap nézet, ahol a lekérdezés eredményét tekinthetjük meg.
- **SQL View:** SQL nézet, ahol a grafikus nézetben előállított lekérdezés SQL utasításait tekinthetjük meg, illetve lehetőségünk van itt lekérdezést megfogalmazni SQL nyelven.
- **Design View:** tervező nézet, ahol lekérdezéseinket tudjuk megfogalmazni, illetve később módosítani.

i Előfordulhat, hogy olyan lekérdezést kell végrehajtatnunk, melyet tervező nézetben, vagyis grafikus felületen nehezen, vagy egyáltalán nem tudunk megfogalmazni. Ekkor SQL nézetben vagyunk kénytelenek definiálni a lekérdezést.

i A következőekben az egyes relációs algebrai műveleteket képezzük le az Access grafikus lekérdező felületére. Ehhez azonban szükséges a relációs algebra alapvető ismerete, mely már bemutatásra került az előadások folyamán.

Projekció

A projekció valamely tábla bizonyos attribútumait adja vissza. Nézzünk egy példát.

$$\pi_{\text{Tipus,Telepites}}(\text{Szenzor})$$

Ezt a következő módon adhatjuk meg tervező nézetben.

Field:	Tipus	Telepites	
Table:	Szenzor	Szenzor	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			
or:			

SQL utasítás formájában a lekérdezés a következő módon néz ki.

```
SELECT Szenzor.Tipus, Szenzor.Telepites
FROM Szenzor;
```

Az SQL utasításban a SELECT záradékban adjuk meg, milyen oszlopokat (attribútumokat) kívánunk lekérdezni, a FROM záradékban pedig hogy melyik táblából (itt több tábla is szerepelhet). Mind azt később látni fogjuk a FROM záradékban kapcsolhatjuk össze a táblákat (természetes összekapcsolás). Az SQL szintaxisa megengedi, hogy az összes oszlopot (egy táblán belül) egyszerűen megadjunk a „*” jel használatával:

```
SELECT *
FROM Szenzor;
```

Ez alkalmazható abban az esetben is, ha a FROM záradékban több tábla található.

```
SELECT Szenzor.*
FROM Szenzor, Meres;
```



Az SQL (Structured Query Language) egy szabványosított lekérdező nyelv relációs adatbázis-kezelő programokhoz. Maga a lekérdező nyelv tekinthető egyfajta deklaratív nyelvnek, ami nagyjából azt takarja, hogy azt adjuk meg, hogy mit csináljon a program, és nem azt hogy hogyan (nem algoritmusokkal adjuk meg a lekérdezéseket). Ez teszi a nyelvet gyorsan tanulhatóvá és ugyanez adja korlátait is. Az SQL az előadásokon már bemutatásra került; az Access által megvalósított SQL nyelvről bővebben: <http://office.microsoft.com/hu-hu/access-help/bevezetes-az-access-sql-nyelv-hasznalataba-HA010341468.aspx> (2012). Az SQL nézetre váltás megtalálható [korábban](#).

Szelekció

A szelekció egy tábla (reláció) bizonyos rekordjait (előfordulását) adja vissza, amit a mezőkre (attribútumokra) való feltétellel érhetünk el. Az Accessben a lekérdezés esetén általában valamilyen projekciót is alkalmazunk mellé. Végezzük el a következő lekérdezést.

$$\sigma_{TelepitesIdje > 2012.10.13}(\pi_{Tipus, Telepites}(Szenzor))$$

Ezt a lekérdezést a következő módon fogalmazhatjuk meg.

Field:	Tipus	Telepites	
Table:	Szenzor	Szenzor	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		> #2012.10.13. #	
or:			

A feltételben a „#” jelek segítségével jelezzük, hogy Dátum típust adtunk meg. A szám esetén egyszerűen használhatjuk magát a számot (pl. >10), szöveg esetén idézőjeleket kell használnunk (pl. =”valami ami szöveg”), logikai érték esetén az 1 igaz, míg a 0 hamisat jelent. (pl. = 1).

SQL utasítás formájában a lekérdezés a következő módon néz ki.

```
SELECT Szenzor.Tipus, Szenzor.Telepites
FROM Szenzor
WHERE Szenzor.Telepites > #10/13/2012#;
```

A szelekciót az SQL utasításban a WHERE záradékot követően adhatjuk meg.

Amennyiben feltételt is tartalmaz a szelekció (ÉS, VAGY), azokat is kezelhetjük az Access grafikus felületén, ehhez tekintsük a következő kifejezést.

$$\sigma_{(Telepites > 2010.11.12 \text{ AND } Telepites < 2010.11.16) \text{ OR } (Telepites = 2010.11.20)}(\pi_{Tipus, Telepites}(Szenzor))$$

Ezt a következő módon hozhatjuk létre Accessben.

Field:	Tipus	Telepites	Telepites
Table:	Szenzor	Szenzor	Szenzor
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		>#2012.10.02.#	<#2012.11.16.#
or:		#2012.08.18.#	

A következő beállításokat kell eszközölni:

1. A VAGY feltételt ugyanabban az oszlopban adhatjuk meg, akár többet is.
2. Az ÉS feltételhez egy új oszlopot adjunk hozzá a lekérdezéshez, ekkor azonban
3. az oszlop megjelenítését kapcsoljuk ki. Ezt a „checkbox” kikapcsolásával oldhatjuk meg.

Az ÉS és a VAGY kezeléséhez használhatjuk az AND és OR operátorokat is, ezzel egyszerűsítve a lekérdezést.

Field:	Tipus	Telepites	
Table:	Szenzor	Szenzor	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		(>#2012.10.02.# And <#2012.11.16.#) Or #2012.08.18.#	
or:			

A feltételeket természetesen SQL-ben is kezelhetjük.

```
SELECT Szenzor.Tipus, Szenzor.Telepites
FROM Szenzor
WHERE (Szenzor.Telepites > #10/2/2012# AND
Szenzor.Telepites < #11/16/2012#) OR
Szenzor.Telepites = #8/18/2012#;
```

Unió

Az Access a hagyományos halmazműveletek közül az uniót támogatja. A többi halmazműveleti lekérdezést (különbség, metszet) megoldható összekapcsolásokkal és a WHERE záradékkal, így ezeket nem támogatja, habár a szabványos SQL nyelvben ezen műveletek megtalálhatóak.

Az unió műveletet az SQL tervezőben készítjük el, ezért itt SQL utasítást mutatunk be. Az SQL nézet előhívása megtalálható [korábban](#).

Végezzük el a következő műveletet.

$$\pi_{ID}(Szenzor) \cup \pi_{SzenzorID}(Meres)$$

Az unió műveletet a UNION utasítással hozhatunk létre.

```
SELECT ID FROM Szenzor
UNION ALL
SELECT SzenzorID FROM Meres;
```



Az itt bemutatott halmazműveletek csak olyan relációk (allekérdezések, táblák) között alkalmazható, amelyek oszlopainak (attribútumainak) típusai megegyeznek.



Az ALL parancs ebben az esetben azt jelenti, hogy az ismétlődő sorokat a lekérdezés nem távolítja el (halmaz vs. multihalmaz). További részletek:
<http://office.microsoft.com/hu-hu/access-help/tobb-valaszto-lekerdez-es-eredmenyenek-egyesitese-egyesito-lekerdezessel-HA010341541.aspx?CTT=5&origin=HA010341468> (2012)

Descartes-szorzat

Descartes-szorzatot két reláció (tábla, vagy allekérdezés) között tudunk definiálni. Alapértelmezésben, az SQL nyelvben amennyiben nem teremtünk kapcsolatot két tábla között, hanem csak a FROM záradékban felsoroljuk azokat, akkor Descartes-szorzást végzünk. Ennek megfelelően végezzük el a következő műveletet.

$$\pi_{ID}(Szenzor) \times \pi_{SzenzorID}(Meres)$$

Ezt a lekérdezés tervezőben a következő módon adhatjuk meg.

Field:	ID	SzenzorID
Table:	Szenzor	Meres
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		
or:		

SQL lekérdezés formájában pedig a következő formában.

```
SELECT Szenzor.ID, Meresek.SzenzorID
FROM Szenzor, Meres;
```

Természetes összekapcsolás

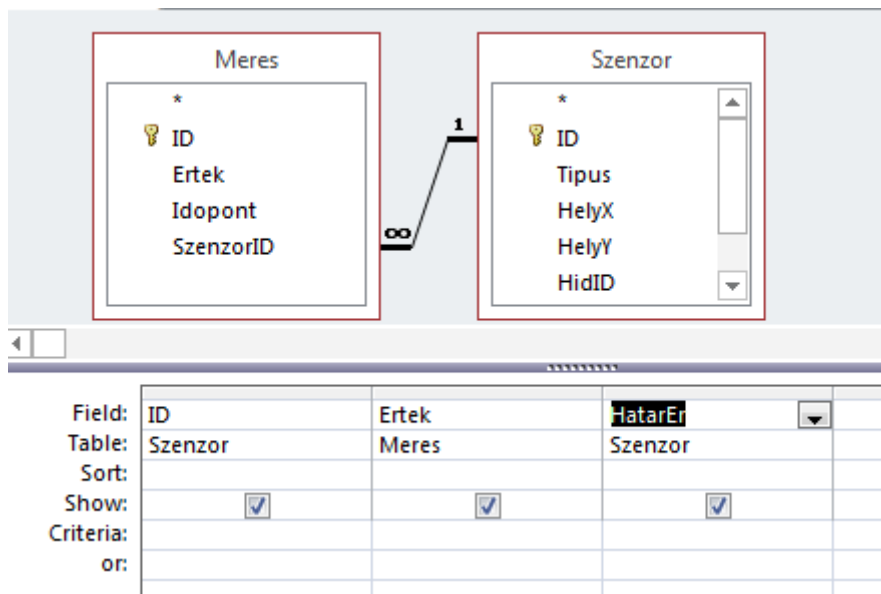
A természetes összekapcsolással táblákat köthetünk össze. Az SQL nyelvben a kapcsolódó mezőket is meg kell adni, így a természetes összekapcsolás relációs algebrában definiált módon nem alkalmazható, azonban itt ezt most szétválasztjuk a théta összekapcsolástól (amelyet a következő fejezetben tárgyalunk). Tehát itt most olyan kapcsolatokat nézzünk, melyek egy vagy több mező(kö)n (attribútum(ok)on) keresztül kapcsolódnak.

A kapcsolatokat a lekérdezés [tervező nézetében](#) hozhatunk létre, a [korábban](#) már bemutatott módon. Nézzünk egy példát.

$$\pi_{SzenzorID, Ertek, HatarEr}(Szenzor \bowtie_{SzenzorID=ID} Meres)$$

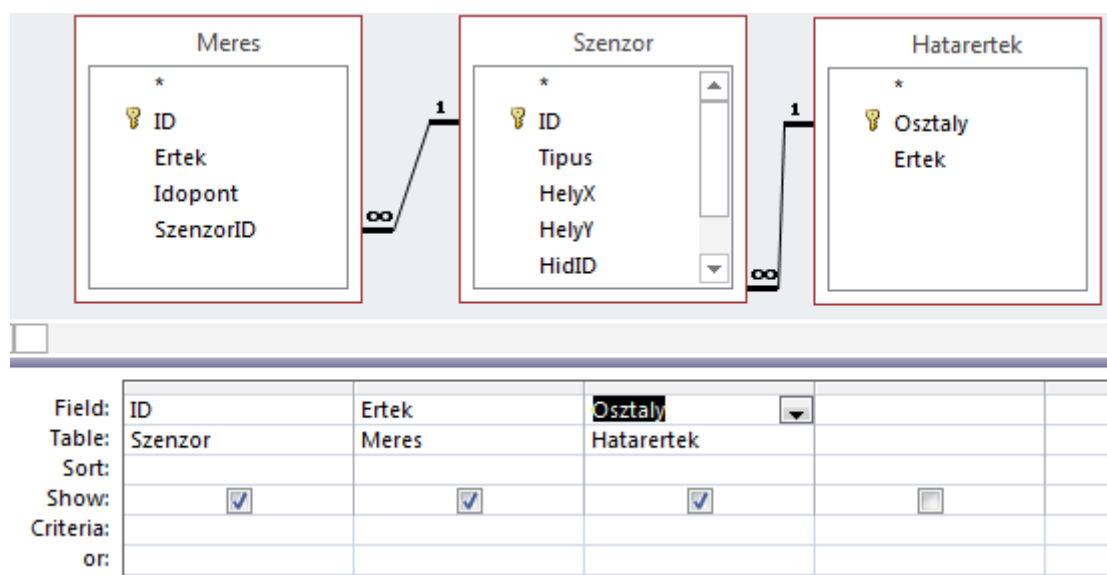
E fenti kifejezés nem természetes összekapcsolás, mivel a kapcsolódó mezőket meg kellett adnunk. De mivel nem teszünk további megkötéseket a lekérdezéssel kapcsolatban, ezért itt most a természetes összekapcsolás alatt tárgyaljuk.

A fenti lekérdezést a következő módon hozhatjuk létre.



Természetesen ilyen módon több táblát is összekapcsolhatunk.

$\pi_{SzenzorID, Ertek, HatarEr}(Szenzor \bowtie_{SzenzorID=ID} Meres \bowtie_{HatarEr=Osztaly} HatarErtek)$



Amikor a kapcsolatokat a lekérdező nézetben átalakítjuk, az nem lesz hatással a séma kialakításakor létrehozott kapcsolatainkra, csak a lekérdezésben jelentkeznek.

SQL nyelven is meg tudjuk fogalmazni lekérdezésünket az INNER JOIN parancs segítségével. Ezt a FROM záradékban adhatjuk meg a következő módon.

```
tabla_1 INNER JOIN tabla_2 ON kapcsoló_mezo_1 = kapcsoló_mezo_2
```

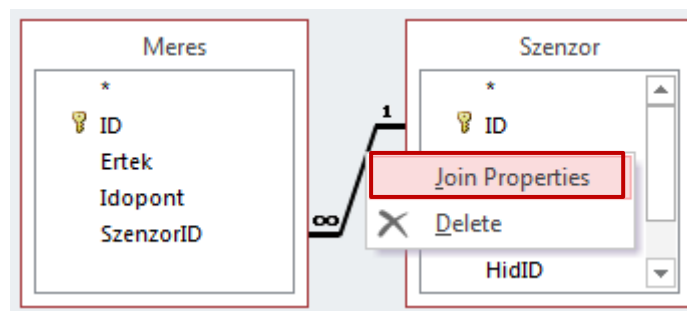
Tehát szintaktikailag az INNER JOIN a kapcsolódó táblák közé kerül, míg az ON részben adhatjuk meg a kapcsolódó mezőket. Ezek után a fenti 3 táblás összekapcsolás SQL kódja a következő.

```
SELECT Szenzor.ID, Meres.Ertek, Hatarertek.Osztaly
FROM (Meres INNER JOIN Szenzor ON Meres.SzenzorID =
Szenzor.ID) INNER JOIN Hatarertek ON Szenzor.HatarEr =
Hatarertek.Osztaly;
```

Meg kell említeni, hogy az INNER JOIN helyett alkalmazhatjuk még a LEFT JOIN, és RIGHT JOIN parancsokat is. Ezek a következőket jelentik:

- **INNER JOIN:** A jobb és bal oldali reláció (tábla, vagy allekérdezés) csak azon a rekordjai szerepelnek a lekérdezésben, melyek illeszkednek az ON részben megadott feltétel(ek)nek.
- **LEFT JOIN:** A bal oldali reláció (tábla, vagy allekérdezés) minden rekordja szerepel a lekérdezésben, viszont a jobb oldali reláció (tábla vagy allekérdezés) csak azon rekordjai, melyek illeszkednek az ON részben megadott feltétel(ek)nek.
- **RIGHT JOIN:** A jobb oldali reláció (tábla, vagy allekérdezés) minden rekordja szerepel a lekérdezésben, viszont a bal oldali reláció (tábla vagy allekérdezés) csak azon rekordjai, melyek illeszkednek az ON részben megadott feltétel(ek)nek.

Ezeket a grafikus felületen is beállíthatjuk amennyiben a kapcsolatra jobb gombbal kattintunk és a „Join Properties” opciót választjuk.



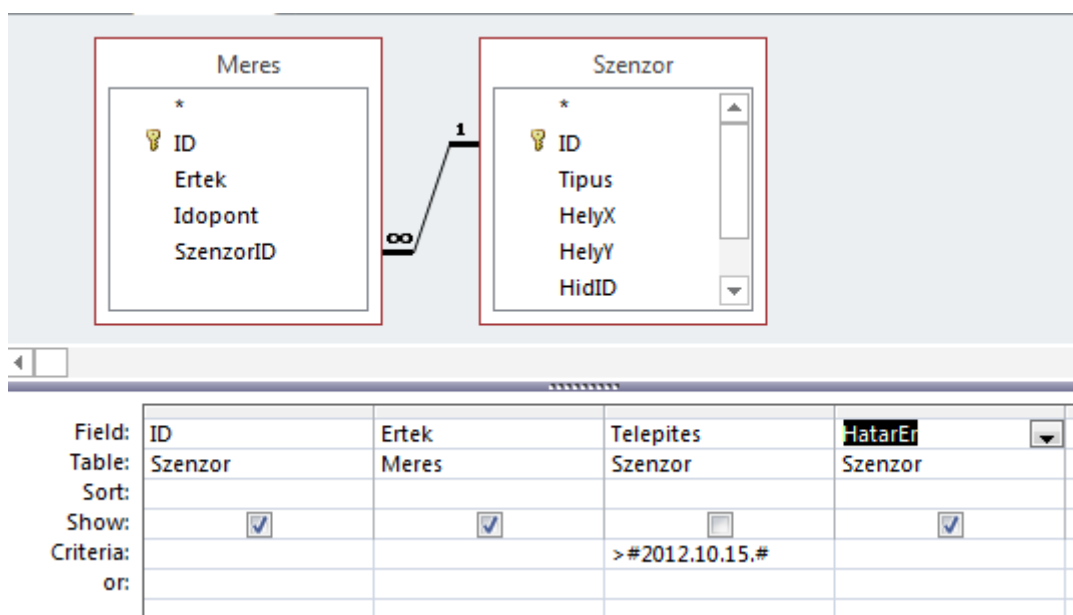
Théta összekapcsolás

A théta összekapcsolást többféle módon képezhetjük le:

- Az INNER JOIN parancs ON részében megadhatjuk a feltételeket, vagy
- a kapcsolat létrehozása után a WHERE feltételben megadhatjuk a feltételeket.

A grafikus felületen a második opciót tudjuk alkalmazni. Példaként tekintsük a következő lekérdezést.

$$\pi_{Szenzor.ID, Ertek, HatarEr}(Szenzor \bowtie_{SzenzorID=ID \text{ AND } Telepites > '2010.11.15'} Meres)$$



Vegyük észre, hogy a *Telepites* nem látható, mivel a fenti kifejezés külső projekciójában ez a mező (attribútum) nincs felsorolva.

SQL utasítás formájában ez a következő.

```
SELECT Szenzor.ID, Meres.Ertek, Szenzor.HatarEr
FROM Szenzor INNER JOIN Meres ON Szenzor.ID = Meres.SzenzorID
WHERE Szenzor.Telepites>#10/15/2012#;
```

Vagyis a kapcsolaton kívüli feltételeket a WHERE záradékban adhatjuk meg.

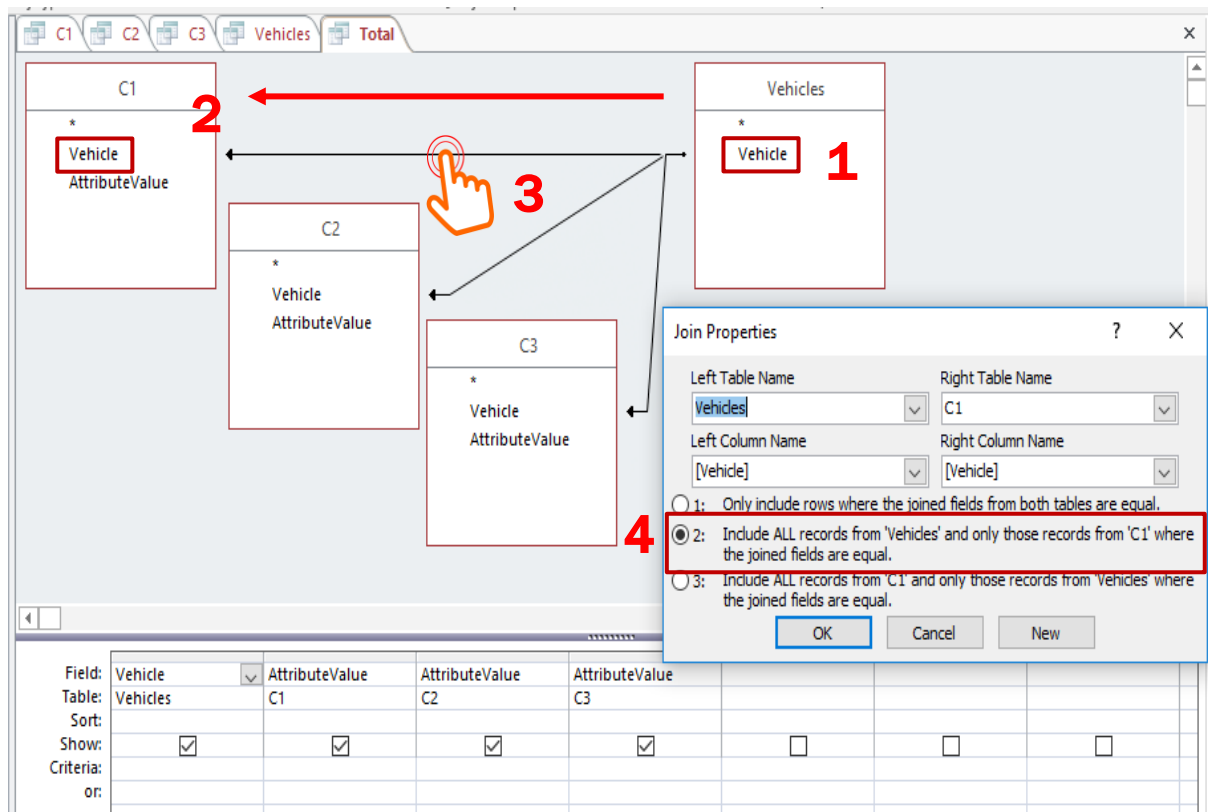
Ahogy már korábban említettük, az INNER JOIN ON ágában is meg tudjuk adni a feltételünket, mely egy természetesebb leképezése a relációs algebrának.

```
SELECT Szenzor.ID, Meres.Ertek, Szenzor.HatarEr
FROM Szenzor INNER JOIN Meres ON (Szenzor.ID = Meres.SzenzorID
AND Szenzor.Telepites>#10/15/2012#);
```



Amennyiben az ON ágban adjuk meg a feltételt, nem használhatjuk az Access lekérdezés tervezőjét, mivel az nem tudja ábrázolni az ilyen módon megadott SQL utasítást. Ebből is látszik, hogy az SQL, mint nyelv, több lehetőséget biztosít számunkra, mint az Access grafikus nézete.

Annak sincs akadálya ha két olyan tábla közt szeretnénk összekapcsolást kialakítani, melyek közt a struktúrában („**Relationships**”) nincs definiált kapcsolat. Ekkor a lekérdezés létrehozásakor egyszerűen kiválasztjuk a két kérdéses táblát, majd az egyik tábla megfelelő mezőjére kattintva, az egeret nem felengedve (**Drag&Drop** jelleggel) áthúzzuk azt a másik tábla megfelelő mezőjére. Ekkor az összekapcsolás alapja e két mező egyenlősége lesz. Ha szeretnénk a lógó sorokat (egyik vagy másik táblában nem szereplő elemeket), akkor az összekötő vonalra duplán kattintva ez is megvalósítható, a megfelelő változatra való kattintással.



Mező alias

Előfordulhat, hogy egy mezőt máshogy akarunk elnevezni, mint ahogy az egyébként megjelenne. Ezt alias létrehozásával tehetjük meg. Ennek formátuma a következő.

Field:	ID	Ertek	Hatarertekem: HatarEr
Table:	Szenzor	Meres	Szenzor
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

Vagyis az új mező nevet „:” használatával adhatjuk meg a következő szintaxissal: új_mező_név: mező_név. Ennek SQL nyelven megadott módja a következő (aláhúzással jelezve).

```
SELECT Szenzor.ID, Meres.Ertek, Szenzor.HatarEr AS
Hatarertekem
FROM Szenzor INNER JOIN Meres ON Szenzor.ID = Meres.SzenzorID;
```

Azaz erre a célra az AS kulcsszó használható. A lekérdezésben ekkor az oszlop neve a megadott alias nevét veszi fel.

Alias		
ID	Ertek	Hatarertek
1	10	15
1	16	15
1	19	15
1	10	15



Az AS kulcsszóval az SQL nyelvben nem csak mezőket nevezhetünk át, hanem táblákra és allekérdezésekre is hivatkozhatunk. Így az aliasok segítségével az esetleges névütközéseket elkerülhetjük. További információkat az Access-ben alkalmazható aliasokkal kapcsolatban: <http://office.microsoft.com/hu-hu/access-help/talalhivatkozasi-nevenek-alias-letrehozasa-lekerdezesben-adp-HP003083994.aspx> (2012), valamint az SQL nyelvben alkalmazott aliasokkal kapcsolatban: <http://www.rendszerinformatika.com/SQL/sql-alias-alnev.html> (2012).

Összesítő lekérdezések

Az összesítő lekérdezésekkel valamely mező(k) (attribútum(ok)) alapján vonhatjuk össze rekordjainkat. Bemutatása egy egyszerű példán keresztül szemléletes; nézzük meg például a *Meres* táblánk rekordjait.

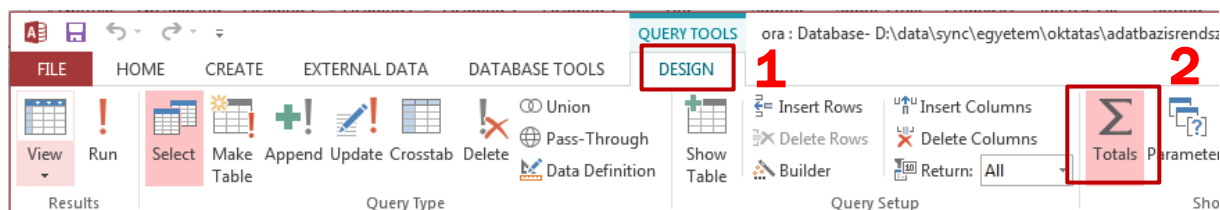
Meres					
ID	Ertek	Idopont	SzenzorID	Click to Add	
1	10	2013.09.26.	1		
2	16	2012.11.16.	1		
3	15	2013.09.03.	2		
4	14	2014.04.01.	25		
5	22	2014.08.20.	26		
6	19	2013.05.16.	1		
7	10	2014.10.27.	26		
8	8	2014.05.11.	6		
9	22	2012.07.29.	23		
10	16	2012.07.12.	3		
11	10	2013.07.26.	12		
12	20	2014.05.12.	12		
13	10	2013.04.13.	14		
14	13	2012.09.29.	8		
15	23	2014.01.17.	22		
16	10	2014.02.18.	1		
17	9	2012.07.24.	4		
18	7	2014.10.25.	26		
19	12	2012.08.22.	22		
20	23	2013.04.07.	21		
21	10	2012.11.24.	5		
22	12	2012.11.04.	9		

Vegyük észre, hogy a *SzenzorID* többször is előfordul, mivel elvileg folyamatosan ebbe a táblába kerülnek be a szenzoroktól érkező mérési értékek. Amennyiben kíváncsiak vagyunk arra, hogy milyen szenzoroktól érkeztek eddig mérések akkor a korábbiaknak megfelelően lehetőségünk van az összes *SzenzorID* kiíratására. Ám ez nem célszerű, mivel egy adott *SzenzorID*-hez több sorunk is tartozik, így egy adott azonosító többször megjelenik.

Field:	SzenzorID	
Table:	Meres	
Sort:		
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		
or:		

SzenzorID
1
1
2
25
26
1
26
6
23
3
12
12

A célunk tehát ezeket összevonni, hogy minden ilyen szenzor esetén csak egyet lássunk. Ehhez alkalmazhatunk egy egyszerű összesítő lekérdezést. Ehhez a „Design” fül „Totals” opcióját kell választanunk.



Vegyük észre, hogy a lekérdezés tervező kiegészült egy újabb sorral.

Field:	SzenzorID	
Table:	Meres	
Total:	Group By	
Sort:		
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		
or:		

Ha most futtatjuk a lekérdezést, akkor minden érték csak egyszer fog szerepelni.

SzenzorID
1
2
3
4
5
6
7
8

Az összesítő lekérdezésnek nem is ez az egyik legértékesebb lehetősége, hanem hogy az összegzett rekordok bizonyos attribútumain műveleteket végezhetünk. Más szavakkal, az egy csoportba tartozó rekordok bizonyos attribútumain műveleteket hajthatunk végre. Ezt is egy példán keresztül mutatjuk be. A következőket kiegészítve adjuk meg az adott szenzorok által mért átlagértékeket. Ehhez az összesítő lekérdezésünket módosítsuk, adjuk hozzá az érték táblát, és „Total” sornál görgessük le a választható opciókat.

Field:	SzenzorID	Ertek
Table:	Meres	Meres
Total:	Group By	Avg
Sort:		
Show:	<input checked="" type="checkbox"/>	
Criteria:		
or:		

Válasszuk az „Avg” opciót és végezzük el a lekérdezést.

SzenzorID	AvgOfErtek
1	14
2	16
3	16
4	10,5
5	10
6	13,2
7	11,5
8	12
9	16
10	16,2

Mi történt? A „**Group By**” opcióval megadott mező alapján (ami a *SzenzorID*) történt a rekordok csoportosítása, így azon rekordok, ahol a *SzenzorID* mező megegyezik, ott az *Érték* attribútumban lévő adatokból átlagot számolt a szoftver.

Összesítő opciók:

- **Group By:** Ezen mezőn keresztül történik a rekordok csoportosítása.
- **Sum:** A megadott attribútumon összeadja a csoportban szereplő értékeket.
- **Avg:** A megadott attribútumon átlagolja a csoportban szereplő értékeket.
- **Min:** A megadott attribútumon a legkisebb értékkel tér vissza.
- **Max:** A megadott attribútumon a legnagyobb értékkel tér vissza.
- **First:** A megadott attribútumon az első értékkel tér vissza.
- **Count:** A megadott attribútumon az összevont attribútumok számával tér vissza. Erről az esetről mutatunk egy példát.

A következőekben hozzunk létre lekérdezést arra, hogy hány darab mérés történt az egyes szenzorokon! Ehhez használjuk a „**Count**” opciót.

Field:	SzenzorID	Ertek
Table:	Meres	Meres
Total:	Group By	Count
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		
or:		

Az eredmény természetesen a helyes megoldás.

SzenzorID	CountOfErtek
1	9
2	4
3	1
4	4
5	1
6	5
7	6
8	4
9	4
10	5

Az összesítő lekérdezésekben továbbra is használhatóak ugyanazon műveletek, melyeket korábban már ismertettünk. Kérdezzük le például azon szenzorokat ahol legalább 3 mérés történt. Ezt a következő módon hozhatjuk létre.

Field:	SzenzorID	Ertek
Table:	Meres	Meres
Total:	Group By	Count
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		>=5
or:		

Az eredmény pedig a következő.

SzenzorID	CountOfErtek
1	9
6	5
7	6
10	5
18	6
19	6
26	7



Az összesítő lekérdezés esetén láthatjuk, hogy ahol nem „Group By” opciót alkalmaztunk, ott az oszlop neve egy elég nehezen értelmezhető, a program által generált név lett. Ezeket célszerű mező alias segítségével megváltoztatni.



Amennyiben az összes megjelenített mező szerint csoportosítunk (és a szerkezet 3. NF követelményeit teljesíti), akkor a csoportosítás művelete felesleges és lassítja a lekérdezést. Ezért felül – a szalagmenüben – kapcsoljuk ki a csoportosítást a „Totals” gomb segítségével.

Természetesen ugyanezen lekérdezéseket SQL nyelven is megfogalmazhatjuk; ehhez a GROUP BY záradékot használhatjuk. Itt adhatjuk meg mely mezőn történjen az összevonás. A mezőn elvégzendő műveletet a SELECT záradékban specifikálhatjuk. A példa kedvéért nézzük meg a fenti lekérdezés SQL utasítását.

```
SELECT Meres.SzenzorID, Count(Meres.Ertek) AS CountOfErtek
FROM Meres
GROUP BY Meres.SzenzorID
HAVING ((Count(Meres.Ertek))>=5);
```



További információk az Access-ben alkalmazható összesítő lekérdezésekhez: <http://office.microsoft.com/hu-hu/access-help/az-adatok-osszegzese-lekerdezes-hasznalataval-HA010096310.aspx>, a PostgreSQL-ben alkalmazható lekérdezésekhez pedig: <http://www.postgresql.org/docs/8.0/static/sql-select.html>

Sorbarendezés

A lekérdezésben megadott rekordjainkat sorba rendezhetjük. Ezt értelemszerűen a lekérdezés tervező „Sort” sorában adhatjuk meg.

Field:	SzenzorID	Ertek
Table:	Meres	Meres
Total:	Group By	Count
Sort:	<div style="border: 1px solid red; padding: 2px;"> Ascending Descending (not sorted) </div>	
Show:		<input checked="" type="checkbox"/>
Criteria:		>=5
or:		

A példánkban a szenzorokat azonosítójuk alapján tudjuk növekvő vagy csökkenő sorrendbe tenni. Amennyiben több mezőn alkalmazzuk a sorbarendeризést, úgy ez balról jobbra fog megtörténni. Azaz először a bal oldali első mező kerül sorbarendezésre, majd az első mezőben azonos rekordok bal oldali második attribútumai alapján történik a rendezés, és így tovább.

A sorba rendezés természetesen alkalmazható összegző lekérdezés nélkül is.



A házi feladatban min. 5 darab lekérdezést hozunk létre, melyből legalább 2 összesítő lekérdezés. Alkalmazzunk sorbarendeризést ott ahol ezt szükségesnek érezzük, valamint adjunk meg mező aliasokat is a mezők neveinek jobb értelmezése céljából.

Matematikai műveletek mezők között

Lehetőség van több mező között matematikai műveletek végrehajtására is, így újabb információk vezethetőek le az adatokból. A műveletek végrehajthatóak a projekciónál és a szűrési feltételek esetén is. A tervező nézetben a mezők és köztük végrehajtandó műveletek begépelésével hozható létre az új érték.

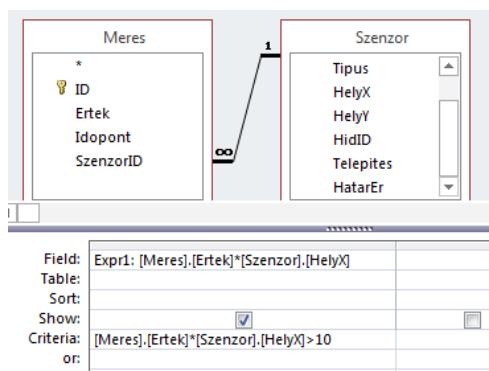
Field:	[Meres].[Ertek]*[Szenzor].[HelyX]
Table:	
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	[Meres].[Ertek]*[Szenzor].[HelyX]>10
or:	



A példa két egymáshoz logikailag nem kapcsolódó mező szorzatát képezi, mely nem követendő példa, csak logikailag értelmes műveleteket valósítandóak meg!

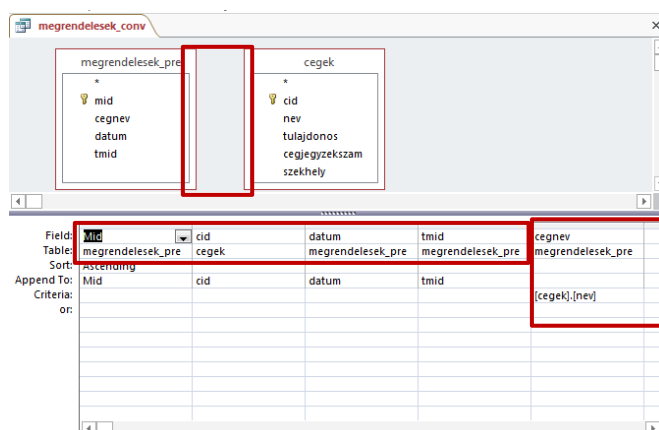


A futtatást követően a tervező nézetbe visszalépve a program elé írja a következőt: Expr1 (magyarul: Kif1).

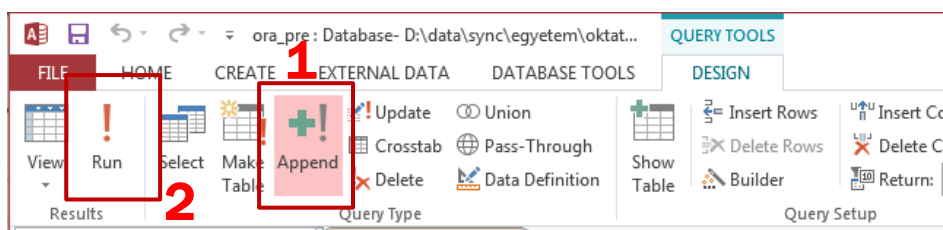


Adat átalakítás

Ha egy táblát másik táblák adatai alapján szeretnénk feltölteni adatokkal, akkor arra is van lehetőség. Ez a feladat jellemzően a nyers adatok átalakítása során merül fel. Ekkor töltjük be ideiglenes táblába a nyers adatainkat. Ezt követően az ideiglenes táblák adataiból lekérdezéssel lehet előállítani a normalizált táblák tartalmát. Tehát a kulcslépés a lekérdezés helyes megfogalmazása, mely rendszerint összekapcsolásokat tartalmaz. Azonban az ideiglenes táblák közt nem tudjuk létrehozni a kapcsolatokat a „Relationships” fülön, ezért a lekérdezéseknél Descartes szorzatot és szűrési feltételt szükséges megadnunk a mező azonosságok ellenőrzésére.



A kívánt eredmény elérése után a menüsorban kiválasztható, hogy az eredményt ne lekérdezésként értelmezze az Access, hanem egy tábla adataihoz fűzze („Append”) hozzá, majd futtasuk („Run”) a lekérdezést.



Idegen kulcsok előállítása Excel segítségével

Az idegen kulcsok értékét kikereshetjük Access segítségével (általában jobb és biztosabb megoldás), de Excel segítségével is megoldható a VLOOKUP (FKERES) függvény segítségével.

Jelen esetben a második táblában szereplő szöveges adatot (cégnév) szeretnénk kikeresni az első táblából, majd a hozzá tartozó elsődleges kulcs értékét beírni a második táblába. Így

végeredményben lecserélhetjük a szöveges értéket az első táblában a helyes idegen kulcs értékre.

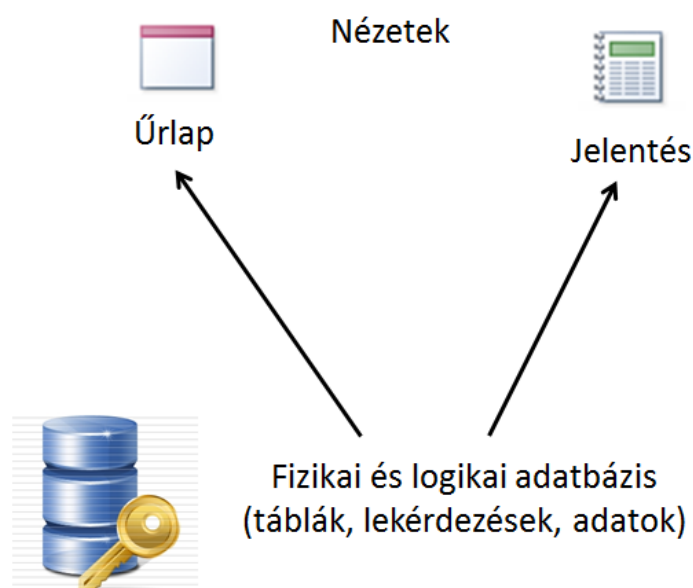
nev	tulajdonos	cegjegyzekszar	szezhely	cid	mid	ceg	datum	tmid	cid
Kiss és Tsa Kft	Kiss Elemér	1212123456	Budapest	1	1	Schneider Kft	#####	1	12
Scávok Kft	Kovács Károly	4587354196	Szombath	2	2	Kiss és Tsa Kft	#####	1	1
Szabó & Szabó Rt	Szabó Ilona	4259751365	Zalaegers	3	3	Berényi Kft	#####	9	7
Hátas Bt	Osváth Géza	9543675189	Miskolc	4	4	Kovács Kft	#####	7	13
Víg Rt	Róbert Károly	3497851354	Budapest	5	5	Víg Rt	#####	1	5
Zsoszó Bt	Rózsa Emília	5487964351	Budapest	6	6	Novák Kft	#####	6	9
Berényi Kft	Berényi Miklós	6423589751	Budapest	7	7	Rézműves Bt	#####	20	16
Bartha Kft	Bartha Zolt	6435128756	Debrecen	8	8	Kocsor Kft	#####	1	19
Novák Kft	Novák László	3964504398	Pécs	9	9	Kardos Kft	#####	13	22
Bokros Kft	Bokros Gizella	8426167743	Debrecen	10	10	Scávok Kft	#####	2	2
Nagy Kft	Nagy Tóbiás	4448190548	Budapest	11	11	Nemes Kft	#####	3	24
Schneider Kft	Schneider Róbert	2852063583	Budapest	12	12	Kertész Kft	#####	9	15
Kovács Kft	Kovács Melinda	5276624293	Győr	13	13	Pongi kft	#####	21	21
Bokros2 Kft	Bokros Ádám	3341762471	Veszprém	14	14	Sághy Kft	#####	9	20
Kertész Kft	Kertész Vilmos	1523507642	Budapest	15	15	Schneider Kft	#####	9	12
Rézműves Bt	Dr. Balogh Nóra	2205933248	Szombath	16	16	Bartha Kft	#####	3	8
Szentmihályi Kft	Szentmihályi Zsolt	5348126637	Eger	17	17	Vida Kft	#####	9	18
Vida Kft	Vida Rudolf	3340843537	Budapest	18	18	Szabó & Szabó Rt	#####	4	3
Kocsor Kft	Kocsor Zolt	6354092450	Budapest	19	19	Bodolai Kft	#####	23	25
Sághy Kft	Sághy Tamás	8330163534	Budapest	20	20	Berényi Kft	#####	9	7
Pongi kft	Dr. Pongrácz Péter	5230556723	Esztergom	21	21	Berényi Kft	#####	9	7
Kardos Kft	Kardos Zoltán	6422856138	Székesfeh	22	22	Nemes Kft	#####	11	24
Szabó Bt	Szabó Szilárd	5311355582	Budapest	23	23	Kovács Kft	#####	2	13
Nemes Kft	Nemes Viktor	5210521953	Budapest	24	24	Kiss és Tsa Kft	#####	11	1
Bodolai Kft	Bodolai András	5411257145	Szolnok	25	25	Bodolai Kft	#####	19	25
					26	Bokros Kft	#####	10	10

nev	tulajdonos	cegjegyzekszar	szezhely	cid	mid	ceg	datum	tmid	cid
Kiss és Tsa Kft	Kiss Elemér	1212123456	Budapest	1	1	Schneider Kft	#####		=VLOOKUP(H21;\$A\$21:\$E\$45;5;FALSE)
Scávok Kft	Kovács Károly	4587354196	Szombath	2	2	Kiss és Tsa Kft	#####	1	1
Szabó & Szabó Rt	Szabó Ilona	4259751365	Zalaegers	3	3	Berényi Kft	#####	9	7
Hátas Bt	Osváth Géza	9543675189	Miskolc	4	4	Kovács Kft	#####	7	13
Víg Rt	Róbert Károly	3497851354	Budapest	5	5	Víg Rt	#####	1	5
Zsoszó Bt	Rózsa Emília	5487964351	Budapest	6	6	Novák Kft	#####	6	9
Berényi Kft	Berényi Miklós	6423589751	Budapest	7	7	Rézműves Bt	#####	20	16
Bartha Kft	Bartha Zolt	6435128756	Debrecen	8	8	Kocsor Kft	#####	1	19
Novák Kft	Novák László	3964504398	Pécs	9	9	Kardos Kft	#####	13	22
Bokros Kft	Bokros Gizella	8426167743	Debrecen	10	10	Scávok Kft	#####	2	2
Nagy Kft	Nagy Tóbiás	4448190548	Budapest	11	11	Nemes Kft	#####	3	24
Schneider Kft	Schneider Róbert	2852063583	Budapest	12	12	Kertész Kft	#####	9	15
Kovács Kft	Kovács Melinda	5276624293	Győr	13	13	Pongi kft	#####	21	21
Bokros2 Kft	Bokros Ádám	3341762471	Veszprém	14	14	Sághy Kft	#####	9	20
Kertész Kft	Kertész Vilmos	1523507642	Budapest	15	15	Schneider Kft	#####	9	12
Rézműves Bt	Dr. Balogh Nóra	2205933248	Szombath	16	16	Bartha Kft	#####	3	8
Szentmihályi Kft	Szentmihályi Zsolt	5348126637	Eger	17	17	Vida Kft	#####	9	18
Vida Kft	Vida Rudolf	3340843537	Budapest	18	18	Szabó & Szabó Rt	#####	4	3
Kocsor Kft	Kocsor Zolt	6354092450	Budapest	19	19	Bodolai Kft	#####	23	25
Sághy Kft	Sághy Tamás	8330163534	Budapest	20	20	Berényi Kft	#####	9	7
Pongi kft	Dr. Pongrácz Péter	5230556723	Esztergom	21	21	Berényi Kft	#####	9	7
Kardos Kft	Kardos Zoltán	6422856138	Székesfeh	22	22	Nemes Kft	#####	11	24
Szabó Bt	Szabó Szilárd	5311355582	Budapest	23	23	Kovács Kft	#####	2	13
Nemes Kft	Nemes Viktor	5210521953	Budapest	24	24	Kiss és Tsa Kft	#####	11	1
Bodolai Kft	Bodolai András	5411257145	Szolnok	25	25	Bodolai Kft	#####	19	25
					26	Bokros Kft	#####	10	10
					27	Berényi Kft	#####	15	7

A VLOOKUP első paramétere a keresendő érték, majd megadjuk, hogy melyik táblázatban keresünk. Fontos, hogy a kereső táblázatban csak az első oszlopban keresi az értéket és a tőle jobbra lévő oszlopokból adhatunk vissza értéket. A harmadik paraméter a visszaadandó érték oszlopszáma (egyesítől számozva). Az utolsó paraméter értéke legyen FALSE (HAMIS).

Űrlapok és jelentések

Az űrlapok és jelentések célja, hogy a fizikai tárolást leképezzük különböző nézetekre.



Ennek megfelelően az Űrlap az adatok felvitelét segítő eszköz, míg a Jelentés az adatbázisban tárolt adatok megjelenítésére szolgál. Az adatbázis tervezője, alkotója ezeket a nézeteket biztosítja az adatbázis (közönséges) felhasználójának az interakcióra.

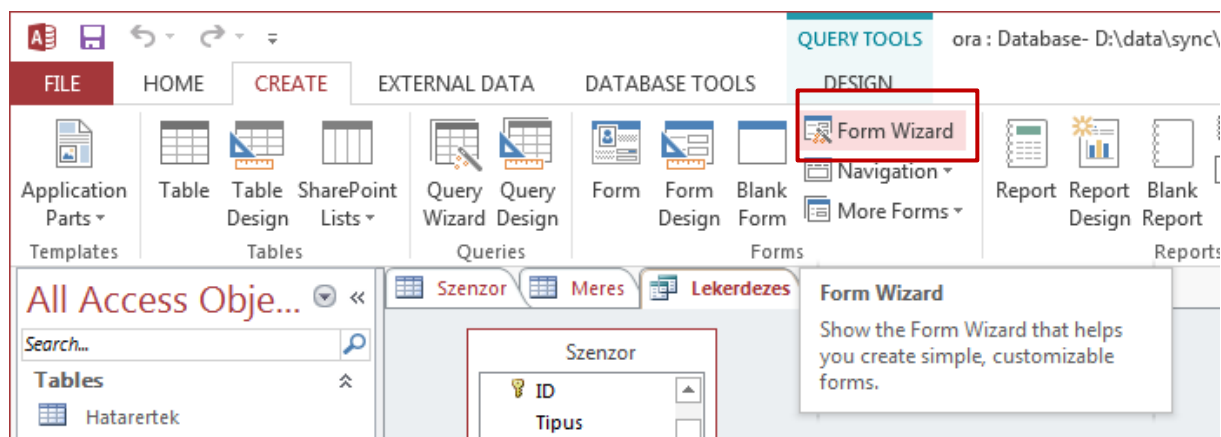


Mielőtt létrehoznánk az Űrlapokat és Jelentéseket, mentjük el tábláinkat, lekérdezéseinket a korábban már ismertetett módon.

Űrlapok


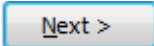
Mivel az űrlapok **adatok felvitelére** szolgálnak, ezért célszerű táblák feltöltésére alkalmazni, akkor is olyan táblák esetén, melyek valószínűleg a rendszer működése közben gyakran változnak (azaz nem ún. *domain* táblák, *domain* táblákról lásd [fentebb](#)).

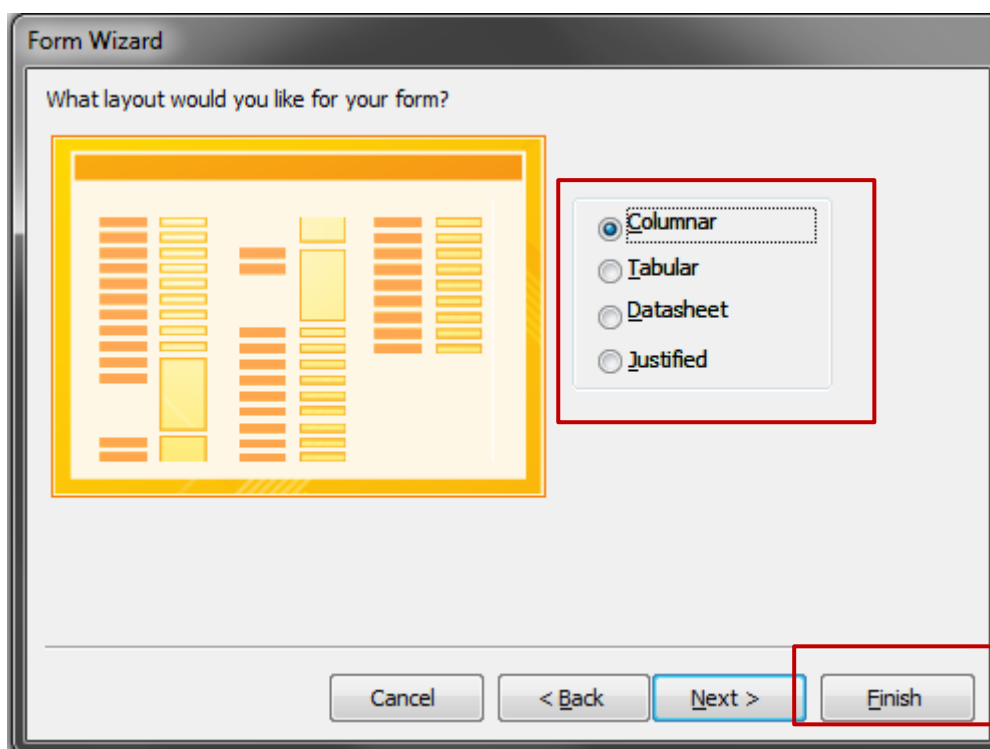
Több lehetőségünk van űrlapok létrehozására, mi a legegyszerűbben, varázsló segítségével fogjuk létrehozni azokat. Kattintsunk a „Create” fül „Form Wizard” opciójára.



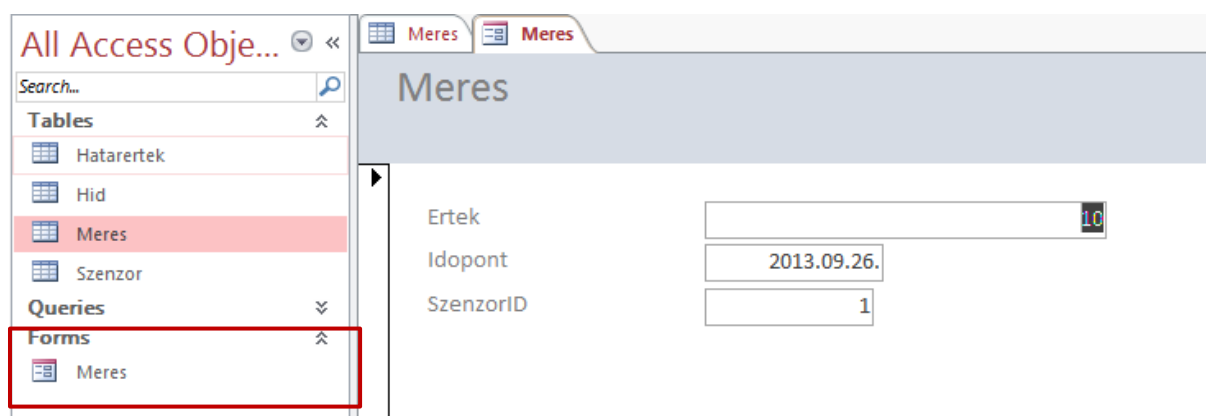
Megjelenik a varázsló.

A következő beállításokat tegyük meg:

1. Válasszuk ki a táblát, amelyen az űrlapot létre kívánjuk hozni,
2. majd válasszuk ki azon mezőket (attribútumokat), melyek felvitelét elérhetővé kívánjuk tenni. A felsoroló típusú ID-kat ebben az esetben nincs értelme megadni.
3. Ezután a  gomb segítségével hozzáadhatjuk a kiválasztott mezőket a jobb oldali listához.
4. Majd a  gombbal tovább léphetünk.



A következő lépésben az űrlap típusát választhatjuk ki, az egyes típusokat itt most nem részletezzük. Végül kattintsunk a „Finish” gombra, ezzel létrejön az új űrlap, melyet ezután a bal oldali sávban is elérhetünk.



Az űrlap nézeteinek elérése és megnyitása hasonló a tábla nézeteinek [eléréséhez](#).

Az űrlapok segítségével az adatbázisunkból egy független alkalmazást is varázsolhatunk, saját menürendszerrel. Ennek módja a Jelentések és űrlapok órai diasorában kerül bemutatásra.



További Űrlap beállítások megtalálhatóak az alábbi weboldalon:
<http://office.microsoft.com/en-us/access-help/examples-of-microsoft-access-data-entry-forms-switchboards-and-dialog-boxes-HA001034548.aspx> (2012)

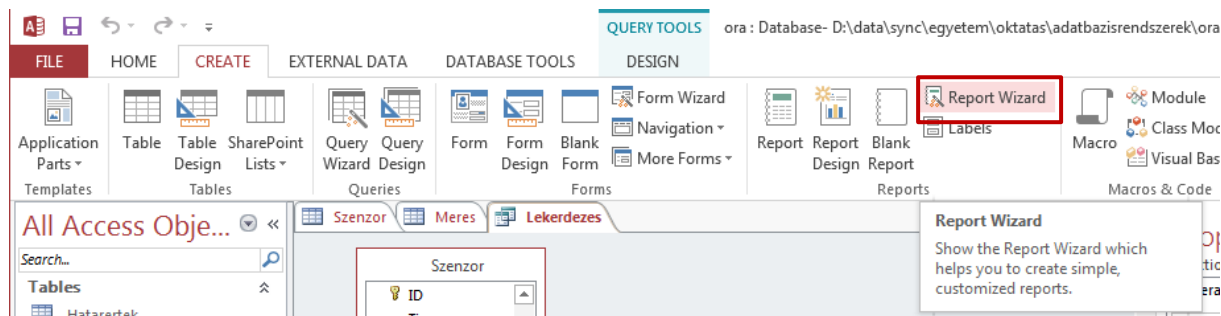


Hozzunk létre legalább 3 darab űrlapot.

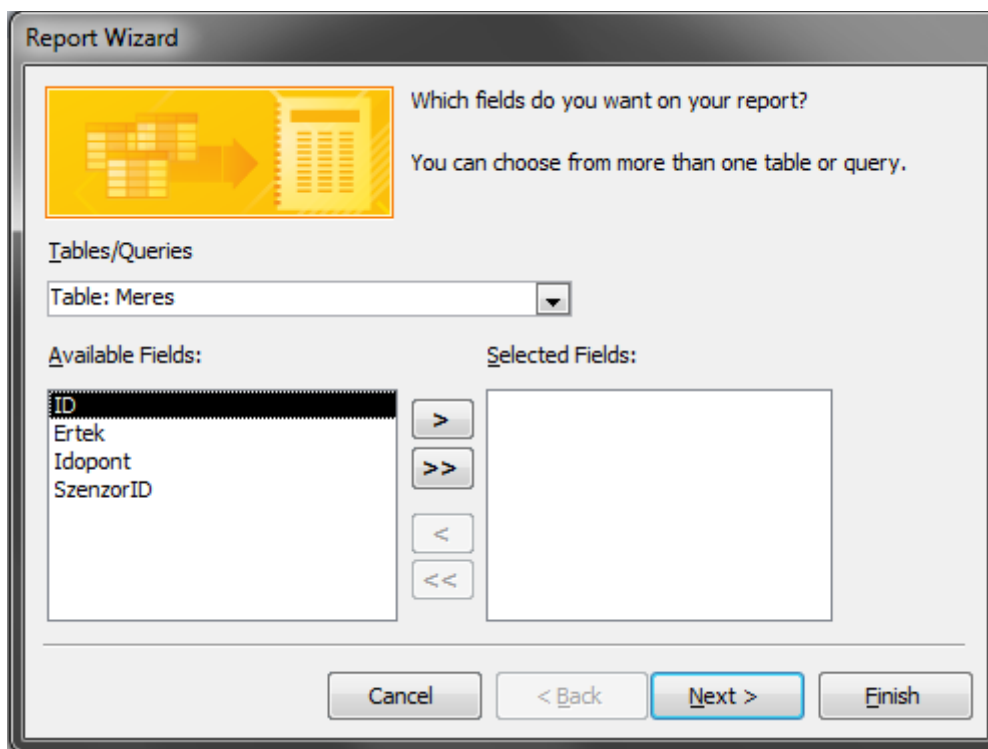
Jelentések

A jelentések szerepe, hogy táblákból vagy lekérdezésekből olyan nézetet hozzunk létre, melyek azután **publikálhatóak, nyomtathatóak**. Jelentések létrehozása többféle módon végrehajtható, mi a legegyszerűbb módon varázsló segítségével fogjuk elkészíteni.

Válasszuk a „Create” fület majd a „Report Wizard” opciót.



A varázslóban az űrlaphoz **hasonló** ablak jelenik meg. A beállításokat ennek megfelelően hajthatjuk végre.



A további beállításokat az Olvasóra bizzuk, a feladat szempontjából a lényeg, hogy nézzük át, milyen eszközöket biztosít a varázsló.

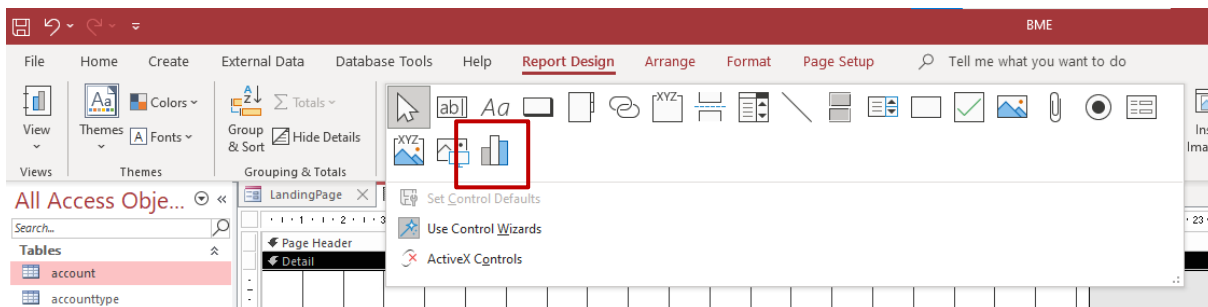


További Jelentéssel kapcsolatos beállítások megtalálhatóak az alábbi weboldalon: <http://office.microsoft.com/en-us/access-help/create-a-simple-report-HA001230739.aspx> (2012)

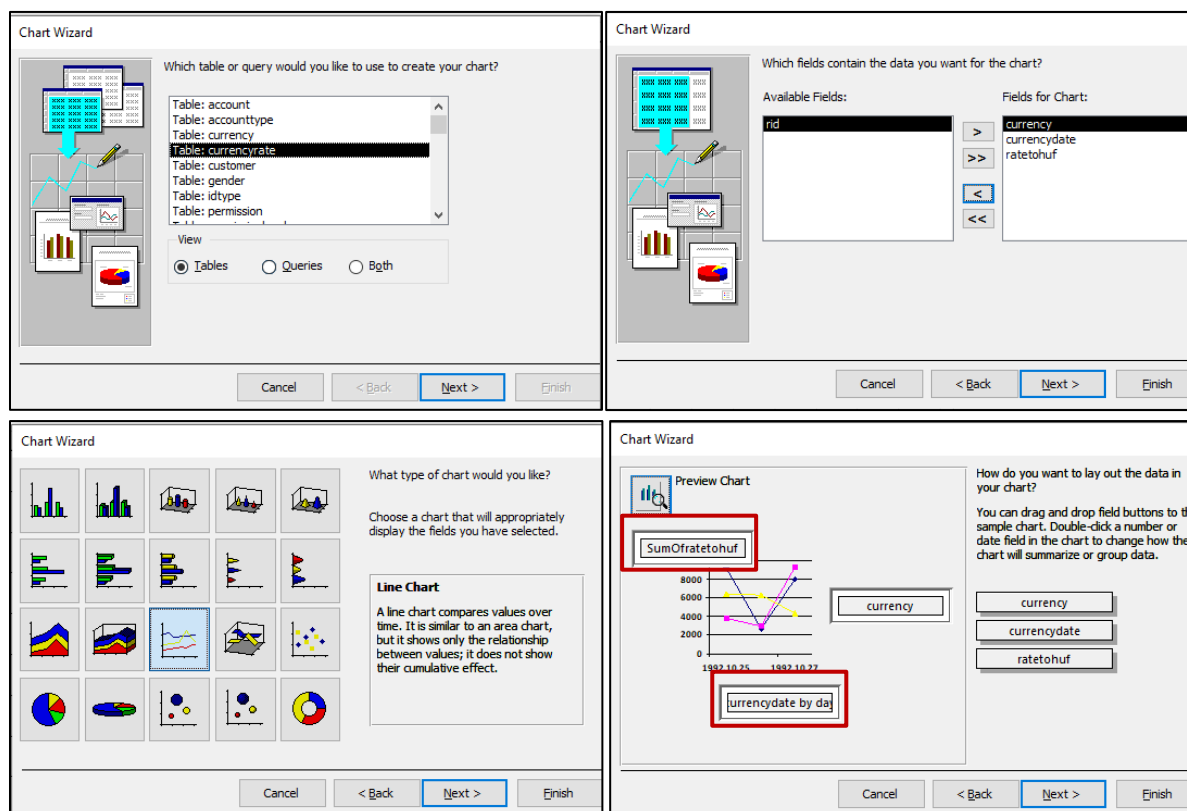


Hozzunk létre legalább 3 darab jelentést.

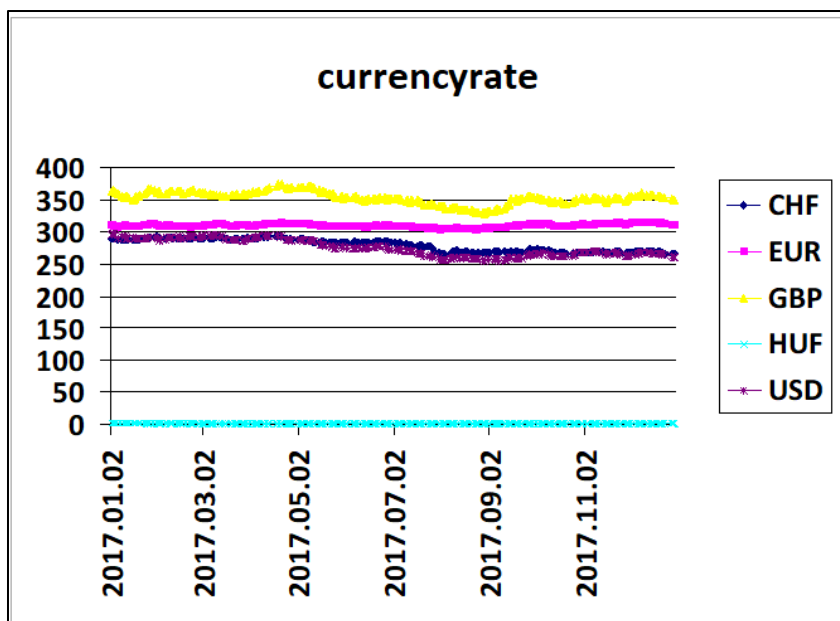
Jelentésekbe grafikonokat, diagrammokat is szúrhatunk. Ehhez használjunk egy Üres jelentést (Blank Report), és szerkesztő nézetben (Design view) szúrjunk be egy grafikont.



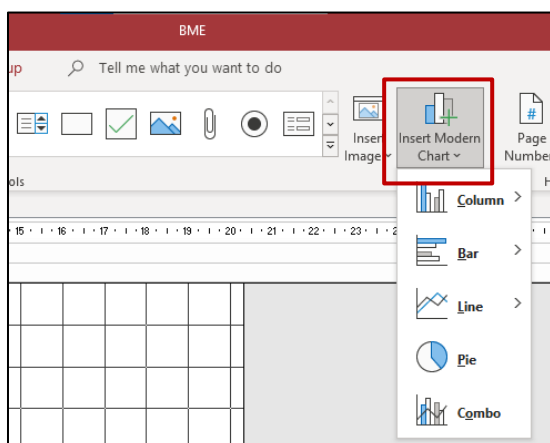
Válasszuk ki az adatforrást (tábla, vagy lekérdezés)



Választhatunk más megjelenítési formát is, figyeljünk arra, hogy több esetben (pl. dátumok tömbösítése) megköveteli a rendszer az összesítést. Ebben az esetben a következő beállítások használhatóak: Series: currency; Axis: currencydate by day; Data: SumOfRatetohuf



Újabb Accessben van lehetőség Modern grafikon beillesztésére is:



Adatbázis véglegesítése

Amennyiben az adatbázist elkészítettük, adatainkat véglegesítettük, érdemes áttekinteni az adatbázist, hogy ténylegesen csak a szükséges elemek szerepeljenek benne. Az adatok importálása során például keletkezhetnek ideiglenes táblák, amikre nincs többé szükségünk. A következő elemek átnézését ajánljuk:

1. Ideiglenes, átalakítást/számítást segítő attribútumok
2. Ideiglenes táblák
3. Felesleges lekérdezések

Végül az Access maga is biztosít egy lehetőséget, hogy kicsit tömörítsük az állományt. Ehhez válasszuk a „Database Tools” lapfület a szalagmenüben, és ott a „Compact and Repair Database” pontot. Kérjük használjátok ezt a lehetőséget a feladat beadás előtt!