

# ADATBÁZISKEZELÉS

## ADATBÁZIS

- *Az adatbázis adott (meghatározott) témakörre vagy célra vonatkozó adatok gyűjteménye.*  
- Pl. A megrendelések nyomon követése kereskedelemben. Könyvek nyilvántartása egy könyvtárban. Egészségügyben a beteg adatainak nyilvántartása, stb.
- *Az adatbázis az adatok és a köztük lévő összefüggések rendszere, amelyet egymás mellett tárolunk.*
- Nagyon fontos, hogy az adatbázisunk szerkezetét jól megtervezzük, mert a későbbiekben csak így tudunk hatékonyan dolgozni vele.

## ADAT

- *A környező világ egy kiválasztott részének olyan leíró egysége, amely táblázatos megjelenítésre, és számítógépes feldolgozásra alkalmas.* Az adat a tények, események, jelenségek stb. jellemzője = **ismérve** = ismertetője = **attribútuma**. Pl. Embernek a neve, autónak a rendszáma.
- Egy adatbázisban egy dologról általában több adatot tárolunk. Ezek jellemzik az adott dolgot, ezek a dolog tulajdonságai, attribútumai. Pl. Autó adatai (jellemzői): rendszáma, színe, típusa, forgalomba helyezése stb.

## AZ ADATBÁZIS-RENDSZERNEK AZ ALÁBBI KÖVETELMÉNYEKNEK KELL MEGFELELNI:

- biztosítsa nagy mennyiségű adat hatékony kezelését,
- egyszerre több felhasználó is használhassa,
- feleljen meg a megadott szabályoknak, őrizze meg az adatok integritását, (Integritás: annak a feltételnek a megléte, hogy az adat helyes, és semmiféle művelet, mint például adatátvitel, tárolás, visszaállítás nem károsította meg az eredeti adatot.)
- nyújtson adatvesztés elleni védelmet,
- tegye lehetővé az egyes felhasználók hozzáférési jogainak szabályozását,
- továbbfejleszthető legyen.

## A TÁBLA:

- *A tábla a logikailag összetartozó adatokat foglalja össze. A táblát el kell nevezni (beszélő név = utaljon a tartalmára)*
- A tábla oszlopokból és sorokból áll.
- *Az oszlopokat mezőknek nevezzük, fejlécükben található a mezőnév.*
- *A sorokat rekordoknak nevezünk.*
- *Az adatbázisok rendszerint több táblát tartalmaznak*  
Pl. *Iskolai nyilvántartás* nevű adatbázis táblái és a benne lévő mezőnevek.  
*Diákok* nevű tábla: vezetéknev, keresztnév, születési hely, születési idő, anyja neve, nem, irányítószám, város, utca- házszám, állampolgárság, stb.  
*Szülők* nevű tábla: szülő vezeték neve, szülő keresztnéve, szülő telefonja.  
*Dolgozók* nevű tábla: vezetéknev, keresztnév, születési hely, születési idő, anyja neve, nem, irányítószám, város, utca- házszám, magán telefon, beosztása belépés dátuma, kilépés dátuma.  
*Tanévek rendje* nevű tábla: tanév neve, 1. félév kezdete, 1. félév vége, 2. félév kezdete, 2. félév vége,  
*Osztályok tantárgyai* nevű tábla: tanév, osztály neve, tantárgyak, tanár.  
*Diákok osztályzatai* nevű tábla: dátum, tantárgy, osztályzat, osztályzat témája.  
Stb. **Egy adatbázisban nem lehet két azonos nevű tábla!**

## A REKORD:

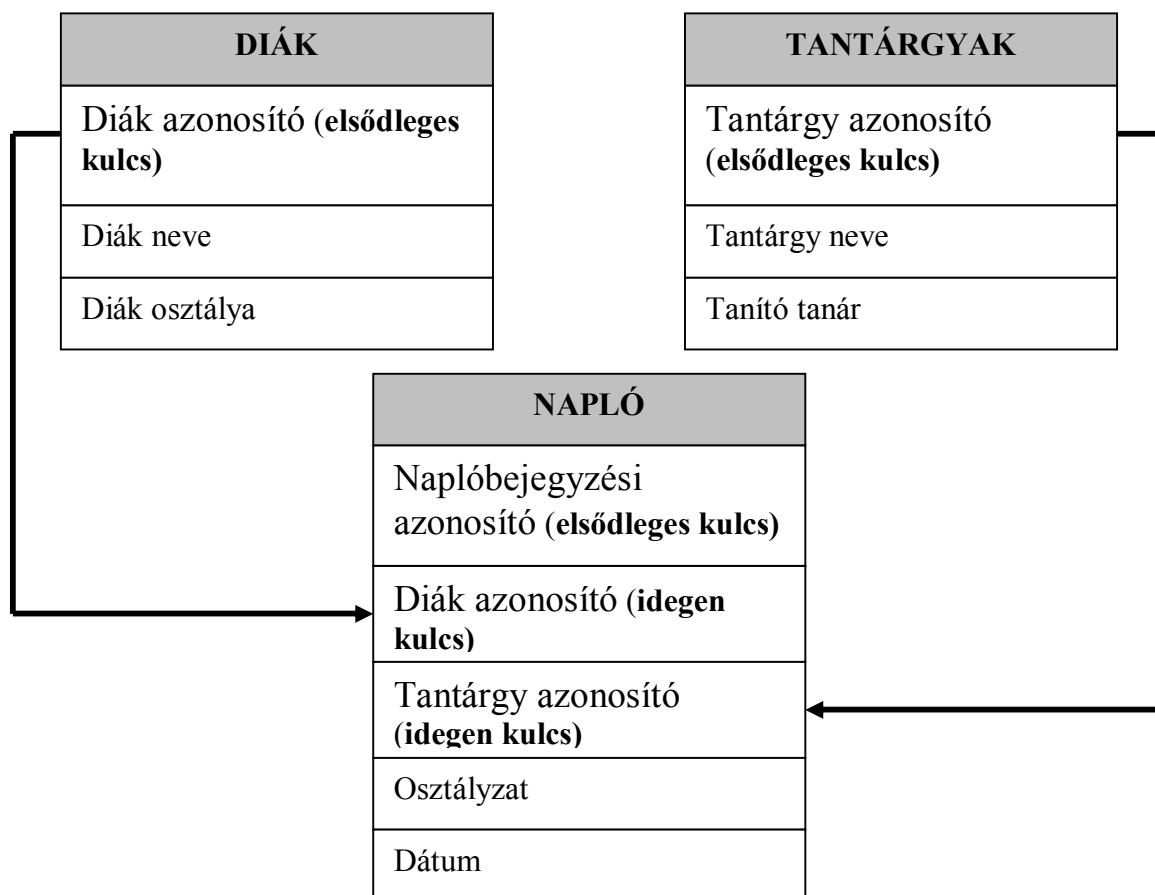
- ***A rekord a tábla egy sora. Egy rekordban tároljuk az egymással összefüggő adatokat, attribútumokat.***  
Pl. Egy emberről tárolt adatok: Vezetéknév, keresztnév, születési hely, születési idő, neme stb. Ezeket a tárolt adatokat, elemi adatoknak nevezzük.
- ***Egyednek (entitásnak) hívjuk azt a személyt, vagy dolgot, amiről az adatot tároljuk.***  
A rekordok az adott egyedről tartalmazzák, az egy sorban lévő elemi adatokat. Pl. egy 1000 rekordos táblának, 1000 sora illetve 1000 egyede (entitása) van.
- ***Egy táblába nem lehet minden adatában egyező két azonos sor, azaz rekord!***
- ***Egy táblába az egyed (entitás) csak egyszer szerepelhet!***  
***Ha mégis azonosak lennének két különböző egyed adatai, szükséges egy megkülönböztető adat. Ez lehet az egyedről egy olyan adat, ami minden kétséget kizáróan azonosítja (pl. személyi szám ember esetén), vagy lehet azonosító, vagy kód, amit az adatokat kezelő személy hozhat létre (ezt elsődleges kulcsnak nevezzük.)***

## A MEZŐ:

- ***mező az adatbázis egy oszlopa, amelyben az egyedek tulajdonságértékeit (elemi adatait - attribútumait) tároljuk.***
- A mező neve azonosítja az oszlopot.
- ***Egy táblába nem lehet két azonos nevű oszlop!***
- A tábla készítésekor a ***mező nevén kívül még meg kell adni a mező jellemzőit:***
  - milyen típusú adatokat kívánunk benne tárolni, (***Adattípusok:*** dátum, szöveg, szám, számláló, pénznem, igen/nem, feljegyzés stb.)
  - mekkora legyen a ***mező mérete*** – azaz a bevihető adat,
  - ***kötelező-e kitölteni,***
  - ***indexelt legyen-e,*** (Az indexelés során a háttérben egy rendezett másolat készül, ami gyorsítja a keresést, de növeli az adatbázis méretét.)
  - ***érvényességi szabály,*** Az adattípuson kívül, ezzel beállítható milyen tulajdonságnak feleljenek meg a bevitt adatok. Rossz érték megadásakor a megjelenő hibaüzenet szövege is megadható.  
Pl. érvényességi szabály magyarországi irányítószám esetén:  $> 999 \text{ AND } < 10000$   
A hibaüzenet szövege: A számnak nagyobbnak kell lennie, mint 999 és kisebb, mint 10000)

## AZ ELSŐDLEGES KULCS:

- ***Egy olyan kitüntetett helyzetben lévő mező, amely egyértelműen azonosítja a rekordot.***
- ***Az elsődleges kulcsot minden mezőnél kötelező kitölteni!***
- ***Az elsődleges kulcsot kötelező megadni (definiálni) a tábla készítésekor.***
- ***Elsődleges kulcs típusai:***
  - ***Egy mezőből álló elsődleges kulcs.*** Akkor használható, ha a mező értékei egyediek, tehát minden kétséget kizáróan azonosítja a rekordot. Pl. személyi szám, TAJ szám, stb. Az ilyen mező típusa ***szám***, vagy ***szöveg***. Ha ilyen mező nem áll rendelkezésre a ***számláló*** típusú mező automatikusan tölti fel sorszámmal az elsődleges kulcs mezőit.  
***Indexelni kell, és nem lehet azonos.***
  - ***Több mezőből álló elsődleges kulcs.*** Akkor használjuk, ha egyik mező sem képes önmagában egyértelműen azonosítani a rekordot, de kettő, vagy több mező már közösen már biztosan egyedi azonosítót ad. ***Indexelni kell, és lehet azonos.***
  - ***Idegen kulcs.*** Olyan azonosító, amelynek segítségével egy másik táblázat elsődleges kulcsára hivatkozhatunk. Lehetővé teszi a táblák közötti kapcsolatok létrehozását. Az idegen kulcsot ***indexelni kell, és lehet azonos.*** Az ilyen táblában szerepelhet saját elsődleges kulcs és a másik tábla kulcsa, idegen kulcsként.



1. ábra Az ábrán táblák és a kulcsaik láthatók, amelyek segítségével létrehozhatók a táblák közötti kapcsolat.

### AZ ADATMODELL:

- *Az adatmodell egyértelműen meghatározza az adatbázis szerkezetét,*
- magában foglalja:
  - az adatok típusát,
  - kapcsolatát,
  - a korlátozó feltételeket
  - az adatkezelési műveleteket.
- A mai adatbázisokban négyféle logikai adatmodellt használunk:
  - hierarchikus,
  - hálós,
  - objektum-orientált,
  - relációs adatmodellt.
- *Az Access a relációs adatmodellt használja.*

### A RELÁCIÓS ADATMODELL:

- *A relációs adatmodellben az adatokat egymással logikai kapcsolatban álló táblázatokba rendszerezük.*
- *A rekordtípusok között előre szervezett kapcsolat nincs. A kapcsolatokat a relációk révén hozzuk létre, a kapcsolat szerkesztése során.*
- A relációs adatmodellben a **táblák halmazok**, azok tartalma pedig a halmazok elemei.
- Egy táblázat 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 táblázatot a neve egyértelműen azonosítja.
- *A táblák összekapcsolásához szükség van közös mezőkre, kulcsokra. (1. ábra)  
Az indexelés pedig gyorsítja az adatok elérését.*

## A TÁBLÁK KÖZÖTTI KAPCSOLATOK:

- *A kapcsolat létrehozásakor az elsődleges tábla (fő tábla) elsődleges kulcsától jelölünk a másodlagos tábla idegen kulcsára.*
- *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.

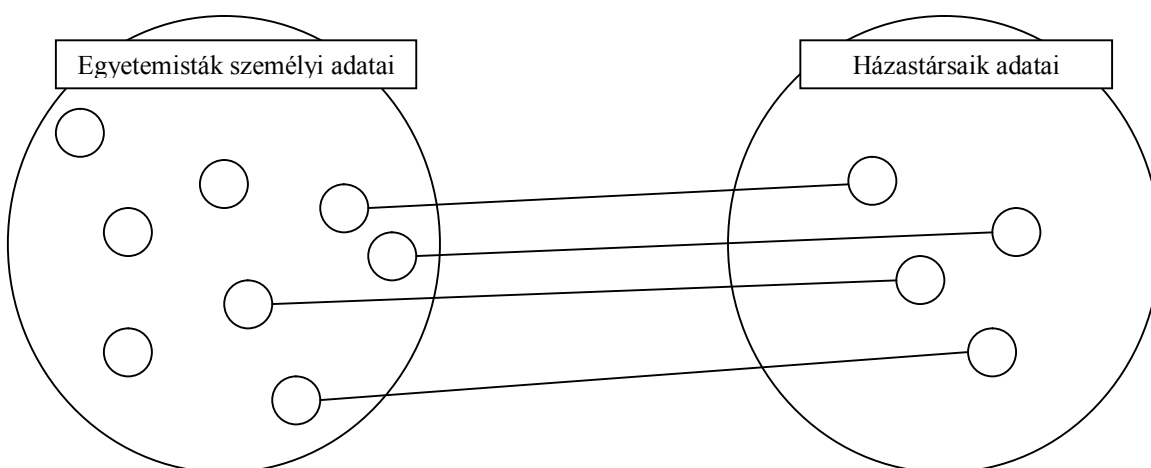
- *Akkor célszerű alkalmazni, ha:*

- Nem akarjuk, hogy az egyénhez tartozó minden adat nyilvános legyen.

- Ha az egyik táblában vannak olyan adatok, amik nem kerülnének mindenkinél kitöltésre, ha ott vennék fel ezeket az adatokat is.

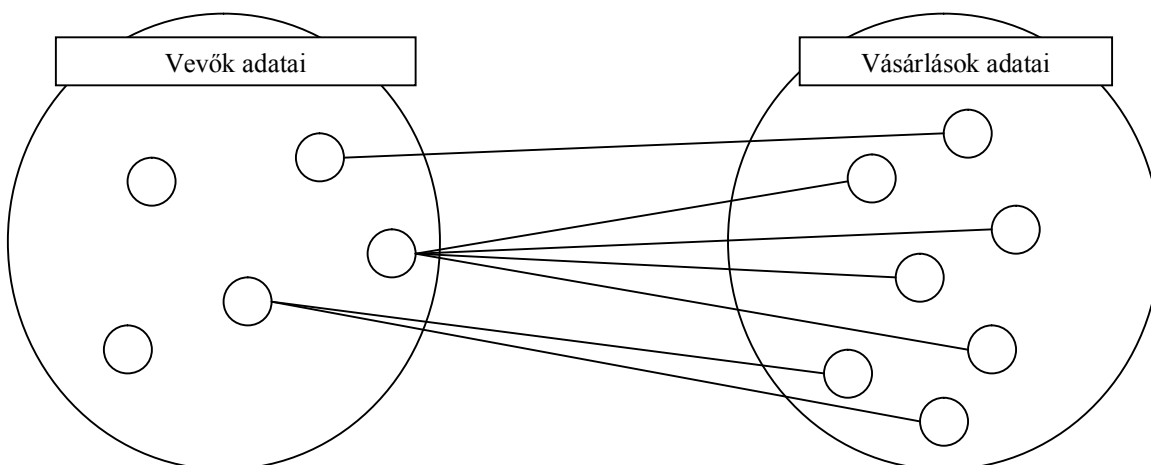
Pl. Egyetemisták házastársainak nyilvántartása esetén. Ilyenkor a házastárs adatai egy másik táblába kerülnek. De csak azoknak lesz rekordja a házastársak táblában, akinek van házastársa. Mivel ma Magyarországon egy személynek csak egy házastársa lehet, és sok egyetemistának nincs házastársa, ezért az adatbázis ilyen kialakítása a leg helytakarékosabb.

Tehát: ha van olyan egyetemista, akinek van házastársa, annak lesz rekordja a másik táblába is, de csak egy.



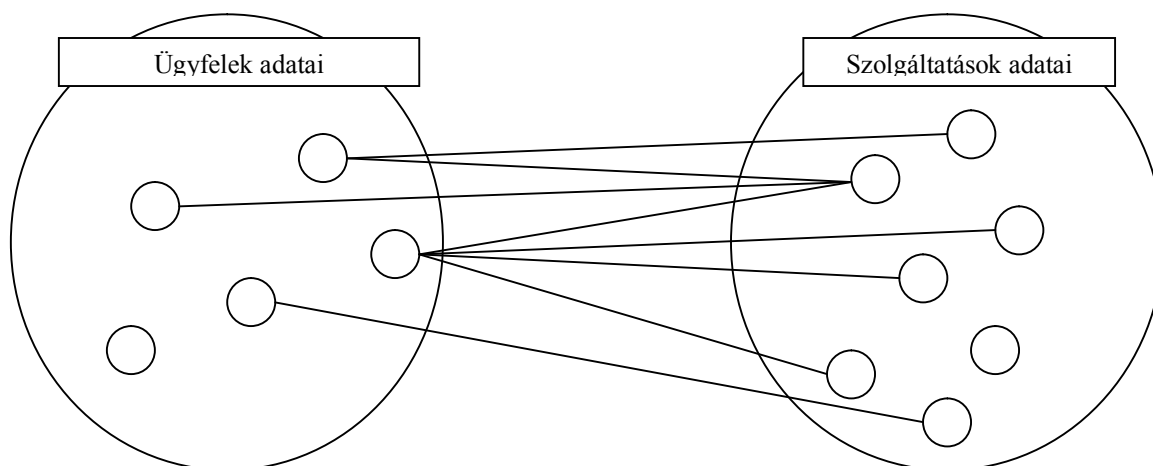
2. ábra 1:1-hez kapcsolat.

- **Egy-több (1:N) kapcsolat:** az egyik tábla egy eleméhez a másik tábla több eleme is tartozhat.
- Pl. Egy vevő többször is vásárolhat ezért az egyik tábla eleméhez a másik tábla több eleme is tartozhat.



3. ábra 1:N-hez kapcsolat.

- **Több-több (N:M) kapcsolat:** bármely tábla elemeihez a másik tábla tetszőleges számú eleme tartozhat.  
Több a többhöz (n:m) kapcsolat esetén egy adott egyed egy vagy több előfordulása kapcsolatban állhat egy vagy több másik egyed előfordulásával. Ilyenkor a kapcsolatot egy harmadik (illesztő) tábla beiktatásával, hálós szerkezettel képezzük le. Az illesztő táblába kell felvenni mindkét tábla elsődleges kulcs mezőjét.  
Ezzel a kapcsolattípussal írhatjuk le például egy cég ügyfeleinek és az ügyfelek számára nyújtott szolgáltatásainak kapcsolatát.



4. ábra N:M-hez kapcsolat.

## A HIVATKOZÁSI INTEGRITÁS:

- *Az összekapcsolt tábláknak a működését a hivatkozási integritás határozza meg.*
- *Ha a táblák közötti kapcsolatok létrehozásakor aktiváljuk ezt a funkciót, akkor csak meghatározott szabályok alapján tudunk adatokat törölni és módosítani.*
- **A HIVATKOZÁSI INTEGRITÁS az alábbi korlátozást eredményezheti:**
  - Új rekord hozzáadása egy kapcsolt táblához csak akkor lehetséges, ha az elsődleges táblában már létezik egy ugyanolyan rekord.
  - Az elsődleges tábla elsődleges kulcsának értékét nem módosíthatjuk, amíg a kapcsolt táblában (másodlagos tábla) tartoznak hozzá adatok.
  - Nem lehet az elsődleges táblából olyan rekordot törölni, amelyhez tartoznak rekordok a másodlagos, kapcsolódó táblában.
- **Kapcsolt mezők KASZKÁDOLT FRISSÍTÉSE:**  
Ha azt szeretnénk, hogy az Access módosítsa az összes kapcsolt táblában is a rekordokat, amikor az elsődleges táblában módosítjuk az elsődleges kulcsot, akkor be kell pipálni a kaszkádolt mezők frissítése pontot a kapcsolat szerkesztésekor.
- **Kapcsolt mezők KASZKÁDOLT TÖRLÉSE:**  
Ha azt szeretnénk, hogy az Access törölje az összes kapcsolt táblában is a rekordokat, amikor az elsődleges táblában töröljük a rekordot, akkor be kell pipálni a kaszkádolt mezők törlése pontot a kapcsolat szerkesztésekor.

## AZ ADATBÁZIS TERVEZÉSÉNEK LÉPÉSEI:

1. **Követelmények elemzés.** A megoldandó feladatot, az adatbázis célját, az alkalmazás rendeltetését határozzuk meg.
2. **Egyedek, táblák meghatározása.** Az összegyűjtött adatokat rendszerezük, és egy információrendszerbe szervezzük.
3. **Attribútumok, mezők, meghatározása.** Ebben a lépésben tervezzük meg a táblákat és a táblákat felépítő mezőket, vagyis konkrétan definiáljuk az egyedtípusokat. Az adatbázis készítésénél cél, hogy a táblák ne tartalmazzanak üres adatokat. A táblák és kapcsolatok kialakításánál ezt is figyelembe kell venni.
4. **Az Azonosítók meghatározása.** A táblák közötti kapcsolatok kialakításához követelmény a táblákban tárolt információ egyértelmű azonosítása.
5. **A Kapcsolatok meghatározása.** Ebben a lépésben a táblák rekordjait kapcsoljuk össze egymással, a táblák kialakítása során azonosított elsődleges kulcsmezők segítségével. A kapcsolat a rendszer szempontjából fontos két egyed összetartozását fejezi ki.
6. **Ellenőrzés.** A táblák, a mezők és a szükséges kapcsolatok megtervezése után nézzük át a tervet, nem maradt-e benne hiba. Alapos ellenőrzést követően könnyebb az adatbázis tervét most megváltoztatni, mint amikor a táblákat már feltöltöttük adatokkal.  
Az Access segítségével hozzuk létre a táblákat, határozzuk meg közöttük a kapcsolatot, vagyis alakítsuk ki az adatbázis külső szerkezetét. Ezt követően próbaképpen írjunk be néhány rekordot minden táblába és vizsgáljuk meg, hogy megválaszolható-e a kívánt kérdések az adatbázis alapján.
7. **Adatbevitel és további objektumok létrehozása.** Ha elvégeztük a szükséges javításokat és ellenőrzésünk szerint az adatbázis terve hibátlan és a táblaszerkezet megfelel a céloknak, továbbmehetünk, és bevihetjük az adatokat a már létező táblákba.  
**Kialakíthatjuk a többi objektumot: létrehozhatunk például beviteli űrlapokat, lekérdezéseket, jelentéseket is.**

## ADATBÁZIS ANOMÁLIÁK:

- **Az anomáliák egy nem megfelelő modellből eredő problémák, ellentmondások. Egy relációs adatbázisban a következő anomáliák léphetnek fel:**
  - **Bővítési anomália:** ha egy rekord felvételekor a már korábban tárolásra került információkat is újra be kell vinni.
  - **Törlési anomália:** amikor az elem megszüntetésekor a nem hozzá tartozó információk is elvesznek.
  - **Módosítási anomália:** amikor az elemi adat módosulásakor az adatbázisban az elemi adat összes előfordulási helyén el kell végezni a módosítást.

## ADATBÁZIS NORMALIZÁLÁSA:

- A normalizálás folyamata során az adatbázisból kiküszöböljük a különféle anomáliákat, a redundanciát (adattöbbszörözést), így csökken az adatbázisfájl mérete, és az adatbázis tartalma logikailag áttekinthetőbb lesz.

## MS ACCESS OBJEKTUMOK:

- **Táblák:** Ezek az objektumok tartalmazzák az adatbázis adatokat.
- **Lekérdezések:**
  - Ezek segítségével gyűjthetjük ki a táblák és más lekérdezések adatait az általunk meghatározott szempontok szerint.
  - Készíthetünk számított mezőket.
  - A gyűjtött rekordokon műveleteket is végrehajthatunk.
- **Űrlapok:** Az adatok bevitelére és módosítására szolgálnak.
- **Jelentések:** A táblákból és lekérdezésekből származó adatokból nyomtatható kimutatásokat készíthetünk.

## LEKÉRDEZÉS:

- A lekérdezés egy olyan feltételrendszer, amely alapján egy vagy több táblában tárolt rekordot válogathatunk ki. A feltételrendszert mi magunk alakítjuk ki, és azt el is menthetjük.

## A LEKÉRDEZÉS TÍPUSAI

- Választó lekérdezés: ez soha nem változtat a rekordokon és táblákon, de **tartalmazhat számított értéket, összesítést, rendezést, feltételt, megjelenítést. Ezeket a QBE-rács mezőrészében kell beállítani.** A QBE rács részei: mező, tábla, rendezés, megjelenítés, feltétel, vagy.
- Összesítő lekérdezés: a választó lekérdezés egy fajtája. Az adatok különböző szempontok szerinti csoportosítására és az ehhez kapcsolódó számítások elvégzésére használjuk.
- Keresztábrás lekérdezés: olyan összesítés, amely két kiválasztott mező adatai alapján összesíti egy harmadik mező adatait. (Pl. egy adattábla adatokat tartalmaz az iskolába járó diákok testvéreinek számáról, és hogy ki melyik osztályba jár. Egy lekérdezésben megadhatjuk, hogy osztályonként hány olyan gyerek van, akinek 1, 2, 3, stb. testvére van.) Varázslóval célszerű létrehozni.
- Töröl lekérdezés (akció típusú lekérdezés): a feltételnek megfelelő rekordokat törli a táblából. (Nem vonható vissza a futtatás után! Le kell futtatni az életbeléptetéshez!)
- Frissítő lekérdezés (akció típusú lekérdezés): a feltételnek megfelelő rekordokat módosítja. (Nem vonható vissza a futtatás után! Le kell futtatni az életbeléptetéshez!)
- Táblakészítő lekérdezés (akció típusú lekérdezés): a feltételeknek megfelelő rekordokból készít egy új táblát. (Nem vonható vissza a futtatás után! Le kell futtatni az életbeléptetéshez!)
- Hozzáfüző lekérdezés (akció típusú lekérdezés): a feltételnek megfelelő rekordokat hozzáfüzi egy meglévő táblához. (Nem vonható vissza a futtatás után! Le kell futtatni az életbeléptetéshez!)

## A LEKÉRDEZÉSEK ÖSSZEKAPCSOLÁSA:

- A lekérdezések összekapcsolhatók. Vannak olyan adatok, amelyek nem gyűjthetők ki a táblából egyetlen lekérdezéssel. Ilyenkor segédlekérdezést kell készíteni. A főlekérdezésben fel kell használni a táblát és a segédlekérdezést is. Ilyenkor szükséges a tábla és a lekérdezés összekapcsolása.

## HELYETTESÍTŐ KARAKTEREK:

- \* (csillag) akárhány karakter lehet. Pl. Like "A\*" – Olyan szavak listázása, aminek az első betűje A betű.
- ? (kérdőjel) pontosan egy karakter lehet. Pl. Like "?a\*" – Olyan szavak listázása, aminek a második betűje a betű.

## OPERÁTOROK ÉS KIFEJEZÉSEK (a kifejezés-szerkesztő segíthet a feltételek elkészítésben):

- <= kisebb vagy egyenlő
- >= nagyobb vagy egyenlő
- = egyenlő
- < kisebb
- > nagyobb
- < > nem egyenlő
- IS NULL - üres mező
- IS NOT NULL - nem üres mező
- Between - a két érték között pl. Between 1 and 3
- In – a megadott értéket keresi pl. In (1;3;5)
- Like – a megadott értéket keresi pl. Like "A\*"

- AND - és, - amikor a megadott feltételek közül mindnek együttesen érvényesülnie kell. Az egy sorban elhelyezkedő feltételek.
- OR - vagy, - amikor a megadott feltételek külön, külön is érvényesülhetnek. A különböző sorokban elhelyezkedő feltételek vagy feltételek. Megengedő vagy. Egyik, vagy másik, vagy mind a kettő.
- NOT – nem – tagadás
- XOR – kizáró vagy. Vagy egyik, vagy másik.

### **SZÁMÍTOTT MEZŐ LÉTREHOZÁSA A LEKÉRDEZÉSBEN:**

- A számított mező, olyan mező, ami a táblában szereplő mezők adatai alapján kiszámítható.
- Általában akkor szokás számított mezőt létrehozni, ha több függvény is szerepel a kifejezésben.
- A számított mező nem szerepel a táblában, azt a lekérdezésben hozzuk létre a QBE rács egy tetszőleges oszlopának első sorában.
- A számított mezőnek nevet adunk, majd kettőspontot írunk. Ezt követi a képlet, amit szeretnénk kiszámítani. A képletben az adatforrásként kiválasztott tábla vagy lekérdezés mezőnevei szögletes [] zárójelbe kerülnek.
- A számított mezőben használhatóak a szokásos műveleti jelek (\* / % + - ) és a kerek zárójel a műveleti sorrend megadására. Pl. Fizetendő ár:[Egységár]\*(1+[Áfa]  
Több tábla, és több azonos mezőnév esetén a tábla nevét is meg kell adni szögletes zárójelben. A táblát és a mezőnevet pont választja el. Pl. [Termékek].[Egységár]
- A számított mezőhöz ugyan úgy adhatunk meg feltételt, vagy rendezési irányt, mint bármely más mezőhöz.

### **ÖSSZESÍTÉS, CSOPORTOSÍTÁS (összesítő lekérdezés):**

- Az összesítő függvények kiválasztására szolgáló sor alapértelmezettként nem látható. Jobb kattintás a QBE rácson és ott a  $\Sigma$  kiválasztására kell kattintani a megjelenítéshez.
- Az összesítéssel egy számított értéket kaphatunk meg. Pl. Legkisebb mezőérték (min), a legnagyobb mezőértéket (max), mezők átlagértékét (avg), összegét (sum) stb.

### **ÖSSZESÍTŐ FÜGGVÉNYEK:**

- Sum (a mező értékeinek összege)
- Expression (a számított mező értékeinek összege)
- Avg (a mező értékeinek átlaga)
- Min (a mező legkisebb értéke)
- Max (a mező legnagyobb értéke)
- Count (hány darab nem üres értéket tartalmaz a mező)
- StDev (a mező értékeinek szórása) Szórás megmutatja milyen mértékű az eltérés az átlagtól. Statisztikai mérőszám.
- Var (a mező értékeinek szórásnégyzete)
- First (az első rekord értéke a mezőben)
- Last (az utolsó rekord értéke a mezőben)
- Group by (Csoportok meghatározása, amelyeken a számítást végre szeretnénk hajtani. Az azonos értékű rekordokból csak egyet jelenít meg)
- Where ( a feltétel mezőben megadott értéknek megfelelő rekordok mezőjét jeleníti meg). Pl. ha a 2010 év termékeinek értékesített mennyiségét kell megjeleníteni. Az év mezőben a Where kifejezést és a feltétel sorba a 2010-et írjuk.