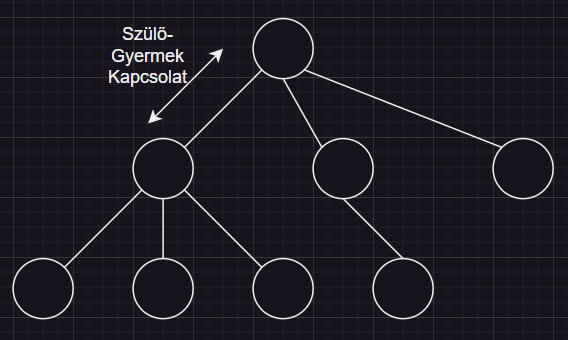
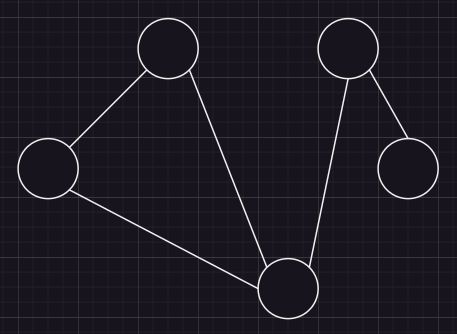
# adatbázis

Az adatok szervezett gyűjteménye, mely lehetővé teszi az adatok tetszőleges szempontok szerinti rendszerezését és visszakeresését

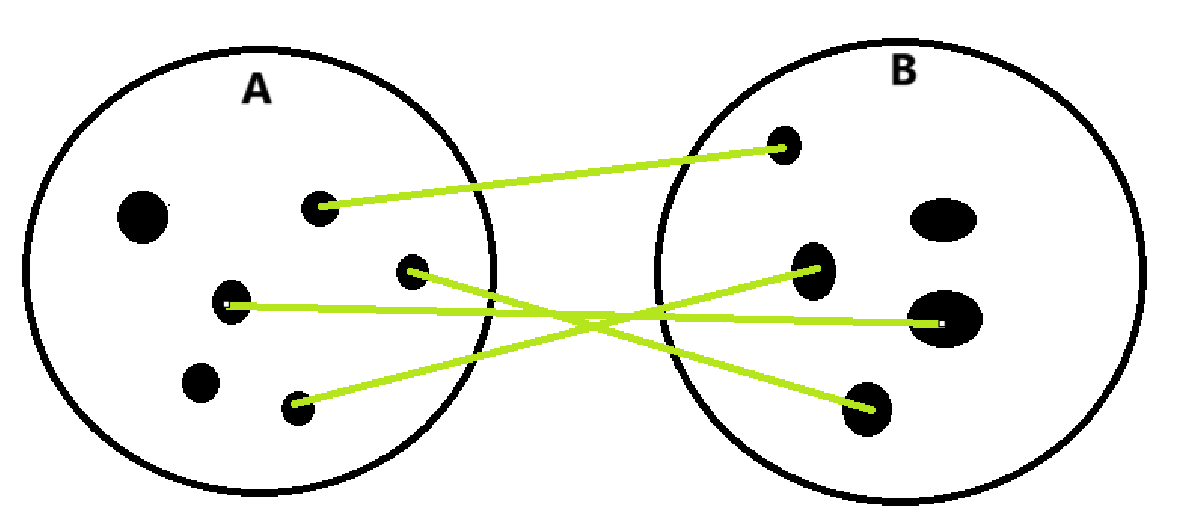
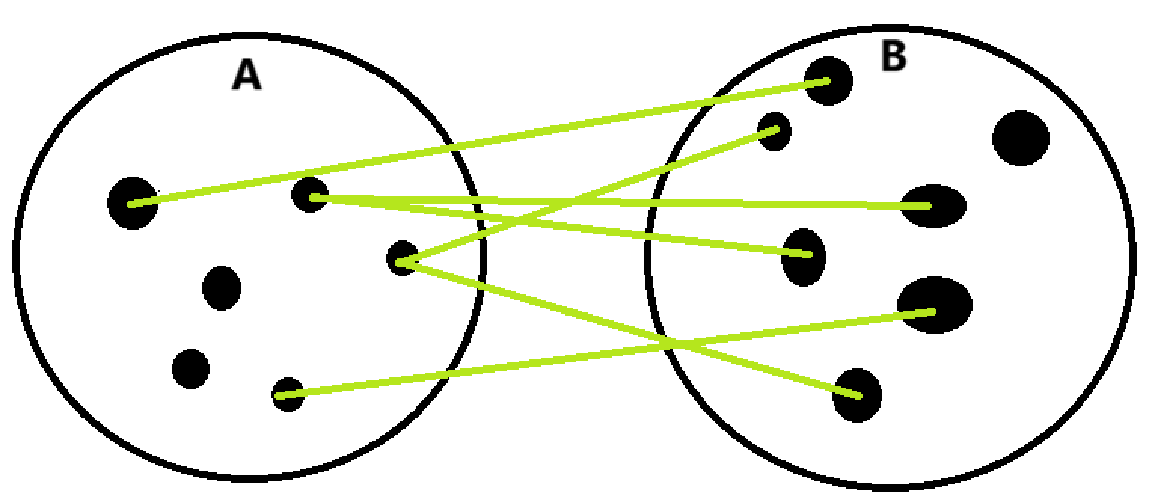
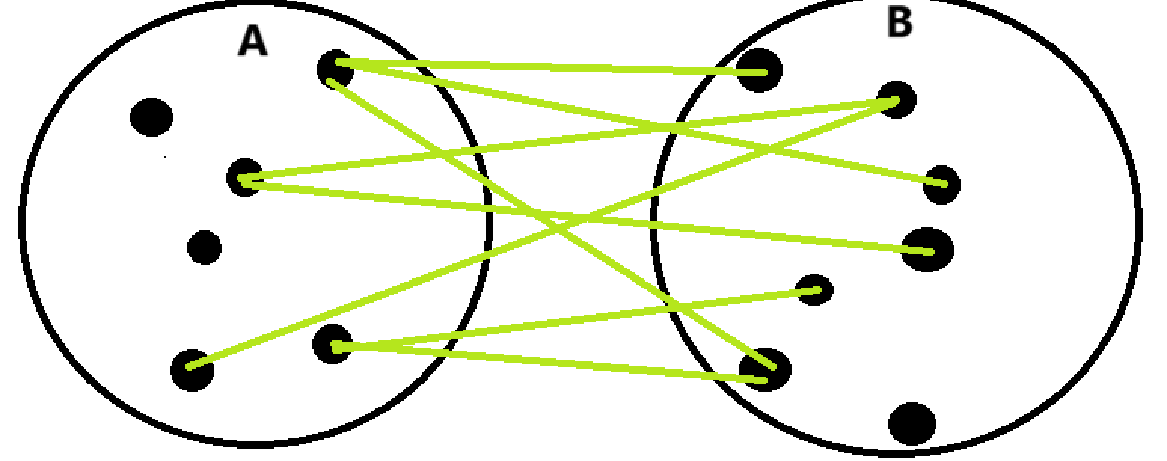
Adatmodellek: Egyértelműen meghatározza az adatbázis (avagy „db”) szerkezetét, magába foglalja az adatok típusát, kapcsolatát, a korlátozó feltételeket, és az adatkezelési műveleteket, 4 félét használunk

* Hierarchikus adatmodell:  
  Az adatok alá-felé rendeltségi viszonyban vannak  
  Szerkezetét gráffal adjuk meg, de a gráf egy fa  
  Jellemzői:
  + Minden egyed csak egy ponton kapcsolódhat (Szülő)
  + A hierarchiában alul lévő egyedtípusoknak csak egyetlen őse lehet (Szülő)
  + A hierarchiában felül lévő egyedtípushoz több, a hierarchiában lentebb lévő egyedtípusok kapcsolódhatnak
  + Ez a modell leginkább 1:1 vagy 1:N kapcsolat szemléltetésére használható
  + Mára ezt a modellt a relációs-adatmodell teljesen kiszorította
  + Egyetlen gyökérelemből elérhető az összes adat
  + Mai napig használt  
    
* Hálós adatmodell
  + Hierarchikus adatmodell továbbfejlesztett változata
  + Szerkezetét gráffal adjuk meg
  + Bonyolultabb kapcsolatok szemléltetését is lehetővé teszi
  + A gráfban a csúcspontok az egyedek, az élek pedig a kapcsolatok
  + Egy-egy egyedtípusnak több őse is lehet, azaz itt egy gyereknek lehet több szülője is
  + Ezt használják a nagygépes db kezelők, pl. TurboIMAGE, IDMS, RDM Embedded, RDM Szerver  
    
* Objektumorientált adatmodell
  + OOP módszertanának egy része
  + Az egyedeket objektumoknak nevezzük
* Relációs adatmodell
  + Legfontosabb eleme a Matematikai reláció fogalma
  + A reláció két-dimenziós táblázatos adathalmaz
  + Ennek az elméletét E.F. Codd dolgozta ki a 60-as évek végén
  + A halmazelméletre épülnek alapjai
  + PC-n ezt a modellt használó programok:
    - **Access**
    - **MySQL**
    - dBase
    - Paradox
    - FoxPro
    - Clipper
    - **Oracle**
    - Ingres
    - OpenOffice Org Base
  + A relációs adatbázis-kezelés alapjai:
    - Az adatokat egymással logikai kapcsolatban álló táblákba rendszerezzük
    - Tábla: Logikailag összetartozó adatokat foglalja össze, sorokból és oszlopokból áll. Pl.:   
      Vevők:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Azonosító | Ceg | Vez.Nev | Ker.Nev | Rendszám |
| 1 | Mol Zrt. | Kiss | Béla | AA-123 |
| 2 | MVM | Nagy | Lajos | AA-222 |
| 3 | MÁV | Szép | Éva | AB-333 |

Autók

|  |  |  |
| --- | --- | --- |
| Rendszám | Típus | Szín |
| AA-123 | Opel | Ezüst |
| AA-222 | Fiat | Fekete |
| AB-333 | KIA | Piros |

* + - Oszlop: Mező (db egy sora, az egyedek tulajdonságértékeit tárolja)
    - Táblázat: A tábla
    - Felső szöveg: Mezőnév
    - Sor: Rekord (Adatbázis egy sora)
    - Elemi adat Az egyednek tulajdonsága
    - Attribútum: egyed jellemzője
    - Elsődleges kulcs: Olyan mező vagy mezőcsoport, mely egyértelműen azonosítja az adattábla rekordjait, értéke egyedi
    - Idegen kulcs: Olyan azonosító mellyel egy másik tábla elsődleges kulcsára vonatkozunk (csak kapcsolatokban)
      * Kapcsolatok: Egyik tábla elsődleges kulcsa (Primary Key, „PK”) és a másik tábla idegen kulcsa (Foreign Key, „FK”) között jön létre, a táblák közti kapcsolatok az egyedek egymáshoz való viszonyát írják le
      * Kapcsolat típusok:
        + 1:1 típusú kapcsolat  
            
          Az A tábla minden egyes rekordjához legfeljebb 1 rekord tartozhat a B táblában, vica-versa  
          Például házastársi kapcsolat (Férfi és Nő a tábla MO.-n)  
          Ritkán használatos kapcsolattípus
        + egy a többhöz (1:N) típusú kapcsolat  
            
          Az A tábla valamely rekordjához több rekord is tartozhat a B táblában, de a B tábla valamennyi rekordjához csak 1-1 rekord tartozhat az A táblában  
          A leggyakrabban használt kapcsolat típus  
          Például Szülőanya-Gyerek (Több), Osztály-Tanuló (Egy osztályba több tanuló járhat, de egy tanuló csak egy osztályba járhat), Szállító-Termékek, Gyerekorvos-Beteg
        + Több-a-Többhöz (N:M) típusú kapcsolat  
            
          A tábla valamely rekordjához több rekord is tartozhat a B táblában, és a B tábla valamely rekordjához is több rekord tartozhat az A táblában  
          A több a többhöz kapcsolat két darab 1:N kapcsolattal és egy kapcsolótáblával működik  
          Például Számítógép-Diák (Egy számítógépet több használhat, egy diák több gépet is használhat), Nagyszülő-Unoka, Szerző-Könyv
    - Összetett kulcs: A kulcs nem egy mezőből áll
    - Reláció fokszáma: a táblában az oszlopok száma
    - Reláció számossága: a sorok száma
    - Tábla sorai, oszlopai felt.:
      * Egyértelmű, egyedi oszlopnév
      * Minden sorban mindig ugyanazok az oszlopok vannak, ugyanabban a sorrendben
      * Az oszlopokban található adatok meghatározott értéket vehetnek fel
      * A táblát a neve egyértelműen azonosítja
      * Minden táblához tartozik egy azonosító
      * Sorok és oszlopokban található metszetpontok atomiak
      * Táblázat minden sorában ugyanannyi adat található
      * Nem lehet 2 vagy több azonos sor
      * Sorok, oszlopok sorrendje nem lényeges

Autók

|  |  |  |
| --- | --- | --- |
| Rendszam | Tipus | Szin |
| ABC-123 | Audi | Piros |
| B-25 | Skoda | Kék |
| AAA-222 | Audi | Piros |

Tulajdonos

|  |  |  |  |
| --- | --- | --- | --- |
| Azonosito | Vez.nev | Ker.nev | Rendszam |
| 1 | Kis | Béla | AAA-222 |
| 2 | Nagy | Éva | B-25 |
| 3 | Kovács | János | ABC-123 |

XAMPP indítás

1. XAMPP Control Panel megnyitása
2. Apache, Start – Megvárni, míg zöld  
   tűzfalon átengedni ha kell
3. MySQL, Start – Megvárni, míg zöld, addig nem megyünk tovább  
   A panel bezárható, tálcán marad
4. phpmyadmin megnyitása (MySQL Admin gomb)  
   VAGY böngészőbe beírjuk, hogy „localhost/phpmyadmin/”

XAMPP bezárás

1. Böngésző bezárása
2. Control Panel megnyitás (tálcáról)
3. MySQL, Stop, ha kész, akkor
4. Apache, Stop, ha kész, akkor
5. Quit gomb leállítja a XAMPP Control Panelt

## SQL

(Structured Query Language, Szabványosított strukturált lekérdezőnyelv)

Az utasításainak fő csoportjai:

1. **DDL** (Data Definition Language)  
   Adatdefiníciós nyelv  
   Ide tartozik: Az adatbázis és szerkezeti elemek kialakítása, módosítása, törlése  
   Pl.: CREATE, ALTER, DROP
2. **DML** (Data Manipulation Language)  
   Adatmódosító nyelv  
   Adatok karbantartására (bevitel, módosítás, törlés) és lekérdezésére használandó   
   Pl.: INSERT, UPDATE, DELETE(, SELECT, bizonyos esetekben)
3. DQL (Data Query Language)  
   Adatlekérdező nyelv  
   Adatok lekérésére használandó  
   Pl.: SELECT
4. DCL (Data Control Language)  
   Adatvezérlő nyelv  
   - Tranzakció kezelés, pl: COMMIT, ROLLBACK, SAVEPOINT  
   - Adatvédelem, felhasználói hozzáférés szabályozás, Pl.: GRANT, REVOKE

Intellisense van, „intelligens segédlet a kurzor pozíciójánál”  
Van Breadcrumb („Kiszolgáló: xxx.x.x.x >> Adatbázis <adatbázis neve>”), az elérési úton akármelyik szintet el lehet rajta érni

Shortcutok:

1. Ctrl+Enter - Indítás

Parancsok

1. „SHOW DATABASE;” – Az adatbázisok listázása a szerveren
2. „CREATE DATABASE …;”

Adatbázis létrehozása:

„CREATE DATABASE <db név>;” (<> ->Behelyettesítendő)

Teljes szintaktika

„CREATE DATABASE <db név>;”  
„ [DEFAULT] CHARACTER SET [=] <Karakter készlet név>;”, a karakter készlet  
„ [DEFAULT] COLLATE [=] <rendezés név>;” rendezési szempont

[] -> elhagyható  
DEFAULT -> alapértelmezett érték/tulajdonság megadása   
- szerver  
 - Adatbázis  
 - Tábla  
 - Mező (Oszlop)  
Szinteken

Magyar ékezetes karakterkészlet: utf8 (UTF-8), latin2 (Nem-cirill közép-kelet európai karakterek)  
Az ABCbe rendezés sorrendje (a, b, c … A, B, C)  
A magyar rendezés: utf8\_hungarian\_ci, latin2\_hungarian\_ci

Példa:  
„CREATE DATABASE probadb”  
„ DEFAULT CHARACTER SET = utf8”  
„ DEFAULT COLLATE = utf8\_hungarian\_ci;”

„CREATE DATABASE tesztecske

DEFAULT CHARACTER SET utf8

DEFAULT COLLATE utf8\_hungarian\_ci;”

Táblák listázása

„SHOW TABLES;”

Adatbázis törlése

„DROP DATABASE <név>”, Feltétele kilépni az adatbázisból

Tábla létrehozása

„CREATE TABLE <név> (

<nev1> <típus> {PRIMARY} {KEY},

<nev2> <típus>

);”

Alapértelmezett érték megadása

„ALTER TABLE <tábla> ALTER <mezőnév> SET DEFAULT <érték>”

Adatok feltöltése táblába

„INSERT INTO <tábla> VALUES

(<mezoertek1>, <mezoertek2>, …), … ;”

Pl.: „INSERT INTO tanulok VALUES („Bolyai János”, „11E”, 18);”

Vagy

„INSERT INTO <tábla>

(<oszlop1>, <oszlop2>, …)

VALUES

(<érték1>, <érték2>, …), … ;”

Pl:”

INSERT INTO tanulok („osztaly”, „nev”) VALUES

(„11E”, „Új Diák”);”

Adatmódosítás táblába

„UPDATE <tábla>

SET <oszlop> = <kifejezés>, …

WHERE <feltétel>;”

pl.: „

UPDATE tanulok

SET kor = 15

WHERE nev = ’Kis Béla’;” („Kis Béla”-nak 15-re állítjuk az életkorát)

„

UPDATE tanulok

SET osztaly = „12E”

WHERE osztaly = „11E”;” („11E” osztályt „12E”re állítjuk)

„UPDATE tanulok

SET kor += 1;” (Mindenkinek +1 év a kor mezőbe)

Adattörlése táblából

„DELETE FROM <tábla> {WHERE <feltétel>}”

pl: „DELETE FROM tanulok

WHERE osztaly LIKE „12%”;” (Minden 12.-est kitöröl)

„DELETE FROM <tábla>” == „TRUNCATE TABLE <tábla>”

SELECT utasítás

„SELECT [ALL || DISTINCT] <oszlopnevek>

FROM <tábla>

[WHERE <feltétel>] - Feltétel

[GROUP BY …] - Csoportosítás

[HAVING …] - Utófeltétel

[ORDER BY … [ASC || DESC]] - Rendezés;”

SUM(…) – Összeadás

AVG(…) – Átlag

MAX(…), MIN(…) – Maximum – Minimum

COUNT(…)

Hány tanuló jár az iskolába? „SELECT COUNT(nev)…”

„SELECT COUNT(nev) AS „Létszám” …”

„SELECT darab \* 100 AS „ujnev” ”