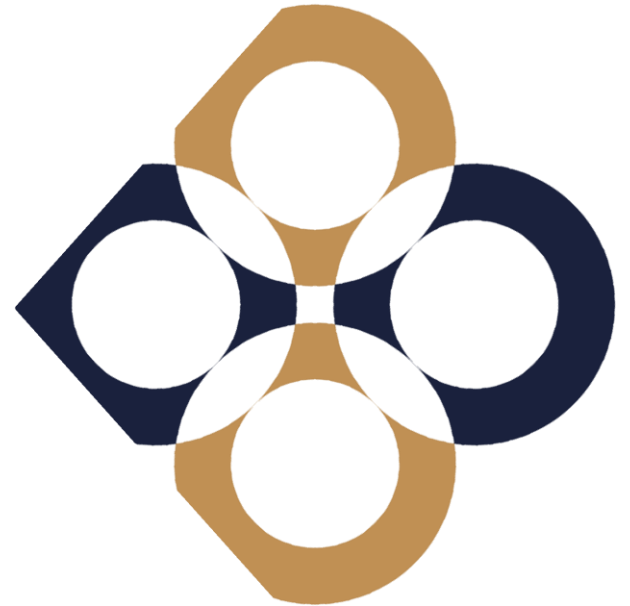


# Adatbázisok előadás I.

## Alapfogalmak



# Miről lesz szó?

1. Tudnivalók
2. Miről fogunk tanulni?
3. Alapfogalmak

# **1. Tudnivalók**

# Követelmények – részletesebb leírás a Moodle-ben

- Félév közben
  - Részvétel a gyakorlatokon (Maximálisan 25% hiányzás megengedett)
  - ZH-k és tesztek megírása, feladatok határidőre történő leadása

# A számonkérések tervezett üteme

- ☐ ZH1 – negyedik gyakorlat elején (SQL alapok + elméleti teszt)
- ☐ ZH2 – hetedik gyakorlat elején (haladó SQL + elméleti teszt)
- ☐ ZH3 - tizedik gyakorlat elején (dokumentum és gráf adatbázisok + elméleti teszt)
- ☐ Félévzáró ZH – tizenkettedik gyakorlaton (elmélet + gyakorlatból minden)
  
- ☐ Házi feladatok: 2 alkalommal, egy hetes határidővel (kiadás: 4. és 8. gyakorlat)
- ☐ Beadandó feladat (esettanulmány): határidő a félév vége, kiadás: 6. gyakorlat)

# Értékelés

- ☐ Félévközi tesztek és zh-k – 30%
- ☐ Házi feladatok – 10%
- ☐ Esettanulmány – 20%
- ☐ Félévzáró teszt és zh – 40%

Jeles: 90%-tól, Jó: 80%-tól, Közepes: 70%-tól, Elégséges: 60%-tól, Elégtelen: 60% alatt

A félév során max. 10% bónusz szerezhető

# Kapcsolat

- Email
  - [geza.molnar2@uni-corvinus.hu](mailto:geza.molnar2@uni-corvinus.hu)
  - A beazonosításhoz a hallgató adja meg a csoportját és NEPTUN-kódját is
- TEAMS

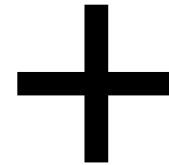
**Más csatornán (pl. Moodle) ne keressenek!**

## **2. Miről fogunk tanulni?**



# Tematika

- Alapfogalmak
- Relációs adatmodell
- Relációs algebra
- Adatbázisok tervezése
- Adattárházak, adattárházfejlesztés a gyakorlatban
- Nem relációs adatbázisok
- Adatbáziskezelő rendszerek a gyakorlatban



# Gyakorlatok I-VII.: SQL-nyelv



Az előadástól független tananyag

Fókusz: lekérdezések (SELECT)

Eszközök:

- ☐ Azure Data Studio
- ☐ Microsoft Fabric
- ☐ SQL Server Express
- ☐ Online SQL Editor

# Gyakorlatok VIII-XII.: NoSQL



Az előadással szinkronban

Fókusz: lekérdezések SQL-es szemszögből

Eszközök:

- ☐ MongoDB Atlas/Compass
- ☐ Neo4J Sandbox/Desktop
- ☐ Redis Cloud/CLI
- ☐ Cassandra Datastax Astra
- ☐ Microsoft Fabric

## Kapcsolódó tantárgyak/területek

Szoftverfejlesztés

Üzleti  
intelligencia

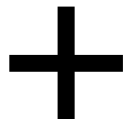
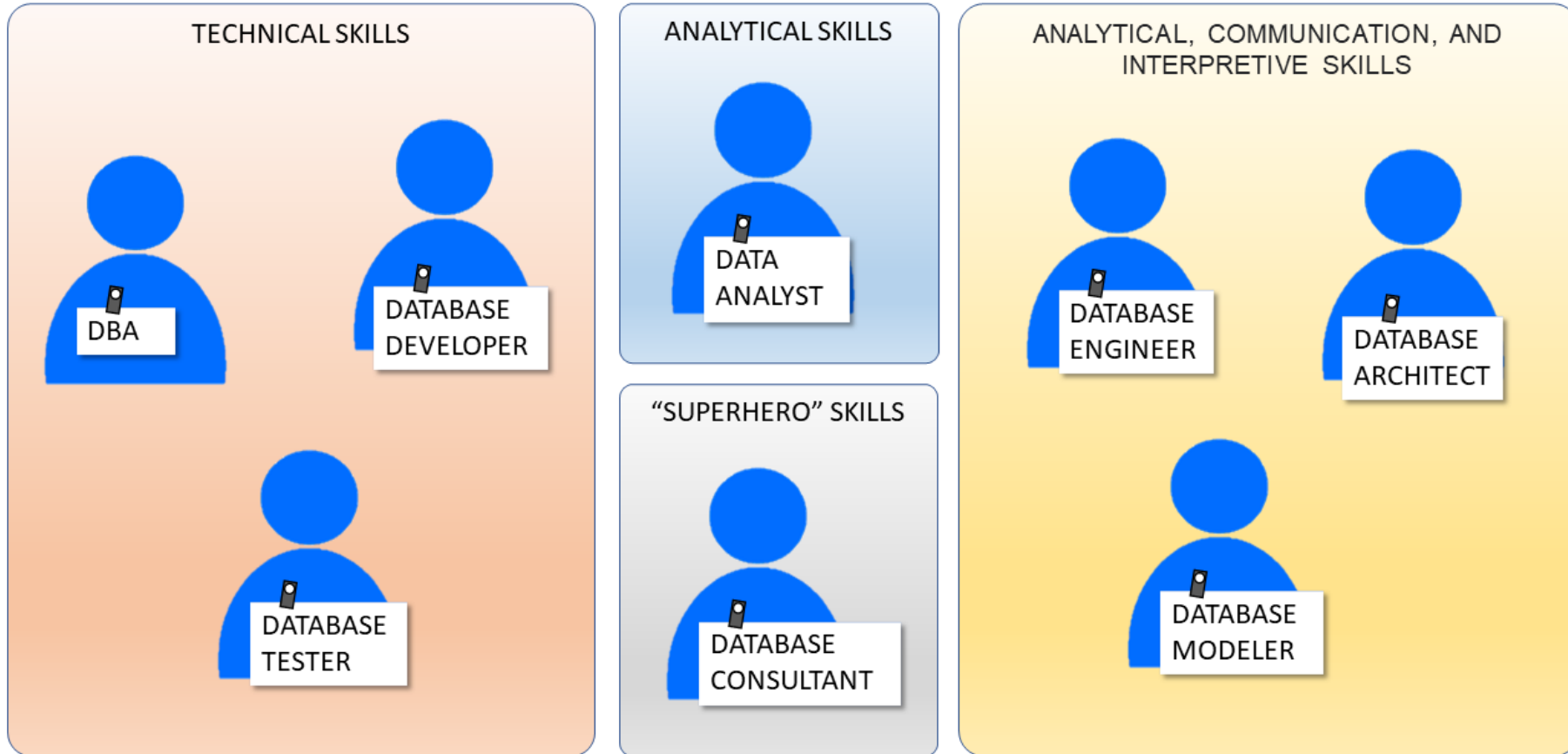
Adatelemzés,  
adattudomány

Információs  
rendszerek

Döntéstámogatás

IT Architektúrák

# Hol alkalmazható a megszerzett tudás?



A kapcsolódó területek pozíciói, pl: webfejlesztő, statisztikus, gazdaságinformatikus stb.

# **3. Bevezetés**

# Miért van szükségünk adatbázisokra?



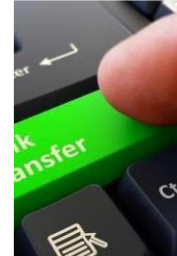
Nagy  
adatmennyiség



Sok  
felhasználó



Konziszten-  
cia



Adatok  
elemzése

Fontos: az adatok lekérdezhetők legyenek!

# A kezdetek (1960 előtt)

- Még nem voltak adatbázisok
- Az adatokat a felhasználói programok kezelték
  - Az adatokat egyszerű fájlokban (pl. szöveges fájl) tárolták
  - A tárolás sémája esetleges volt
  - Lekérdezni csak a programból lehetett
- Minden egyes adattárolási és lekérdezési feladatra külön programot kellett írni



# 1960-as évek eleje

Egyre  
problémásabb  
adatkezelés

Nem mindegy,  
hogyan  
tároljuk az  
adatokat

Szükség van  
valamilyen  
logikára

Megjelentek  
az első  
adatmodellek

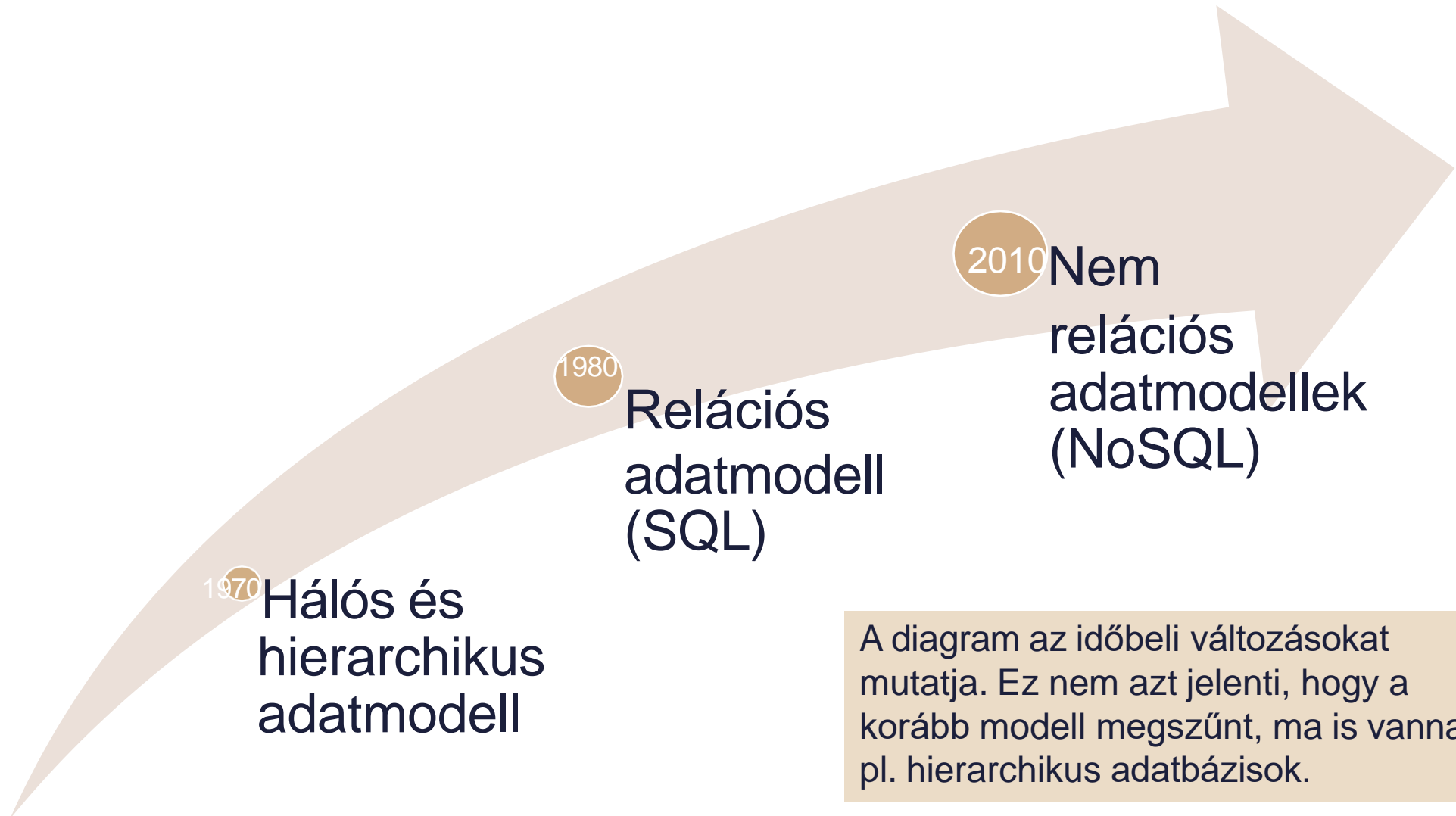
# Az 1960-as évek

- Az első adatbázisok és adatmodellek megjelenése
  - IDS (Integrated Data Store)
    - Az egyik legelső adatbáziskezelő
    - Charles Bachman fejlesztette ki
  - CODASYL
    - A hálós adatmodell megfogalmazása
    - COBOL programozók fejlesztése
  - IMS (Information Management System)
    - IBM fejlesztése
    - Hierarchikus adatmodellt használt

# Az 1960-as évek után

- Sorra jelentek meg az új adatmodellek, és ezekre épülve az új adatbáziskezelő rendszerek
- Minden adatmodellnek voltak közös elemei
  - Olyan dolgok, amelyekről adatokat tárolunk (egyedek)
  - Az adott dolgok jellemzői (tulajdonságok)
  - Az adott dolgok közötti összefüggések (kapcsolatok)
- **A domináns adatmodell a relációs lett**
- Napjainkban megjelentek és kezdenek elterjedni a nem relációs (NoSQL) adatbázisok

# Az adatmodellek fejlődésének mérföldkövei



## **4. Alapfogalmak**

# Alapfogalmak

- Adatmodell
- Adatbázis
- Adatbáziskezelő rendszer
- Adatbázis rendszer

# Egyedek (entitások)

A valós világ olyan dolgai, amelyek minden más dologtól megkülönböztethetők

Pl: autó, személy, termék

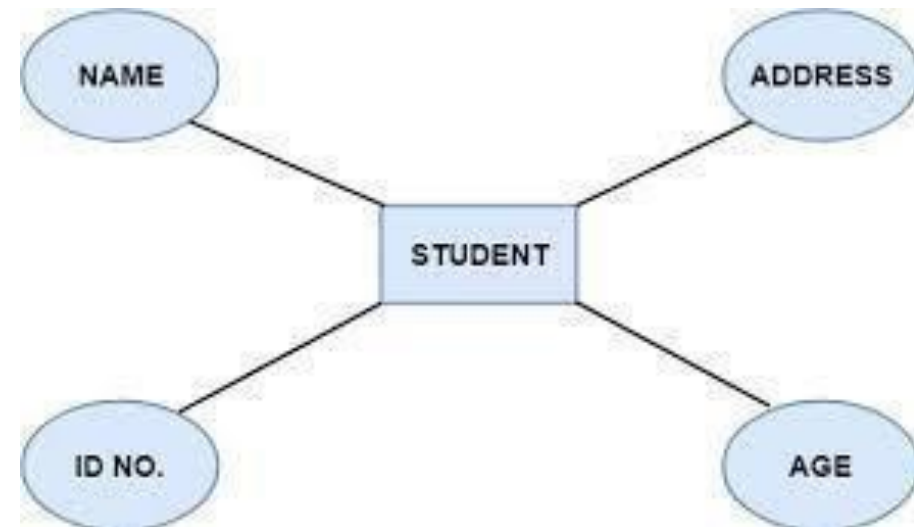
# Tulajdonságok

Az egyedeket tulajdonságaikkal írjuk le, azaz a tulajdonságok az egyedek belső szerkezetét jelentik.

Pl:

A tanuló egyed tulajdonságai lehetnek:

- Azonosító
- Név
- Életkor
- Cím stb.



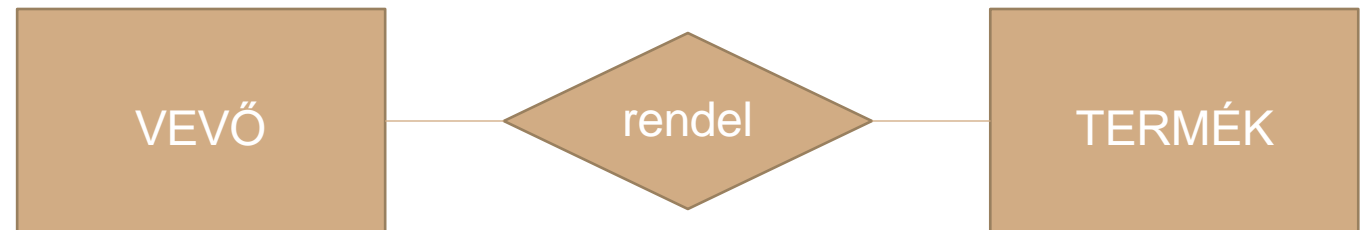


# Kapcsolatok

Az egyedek közötti viszonyt kapcsolatnak nevezzük.  
Másképpen fogalmazva a kapcsolat az egyed külső szerkezete

Példák:

VEVŐ-TERMÉK  
TERMÉK-ELADÁS  
TANULÓ-ISKOLA

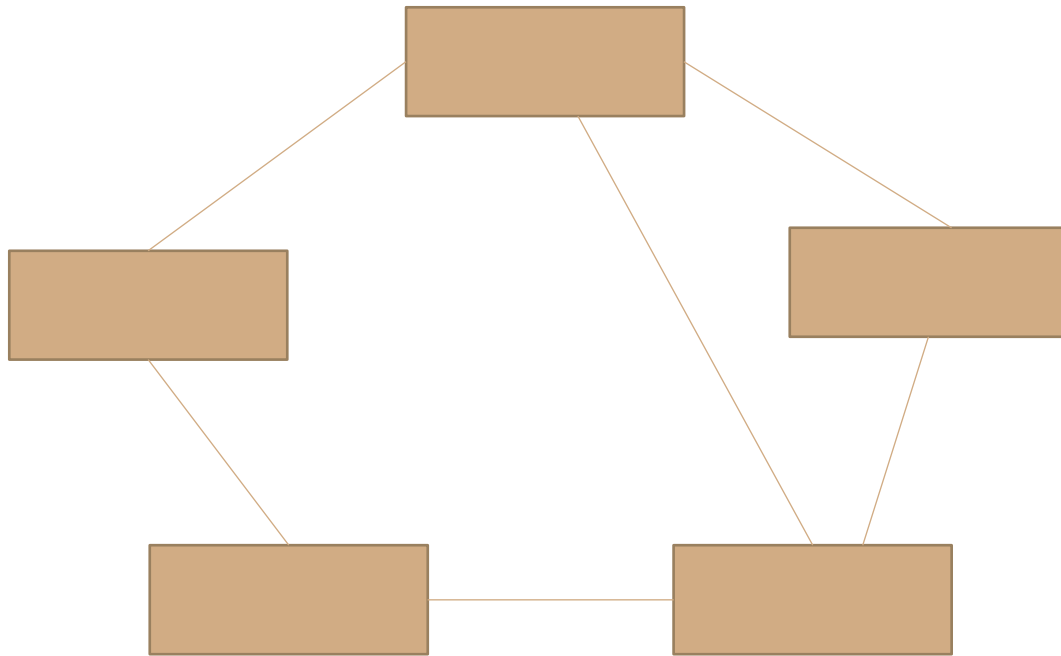


Adatmodellnek nevezzük az egyedek, tulajdonságok és kapcsolatok halmazát

Tipikus adatmodellek:

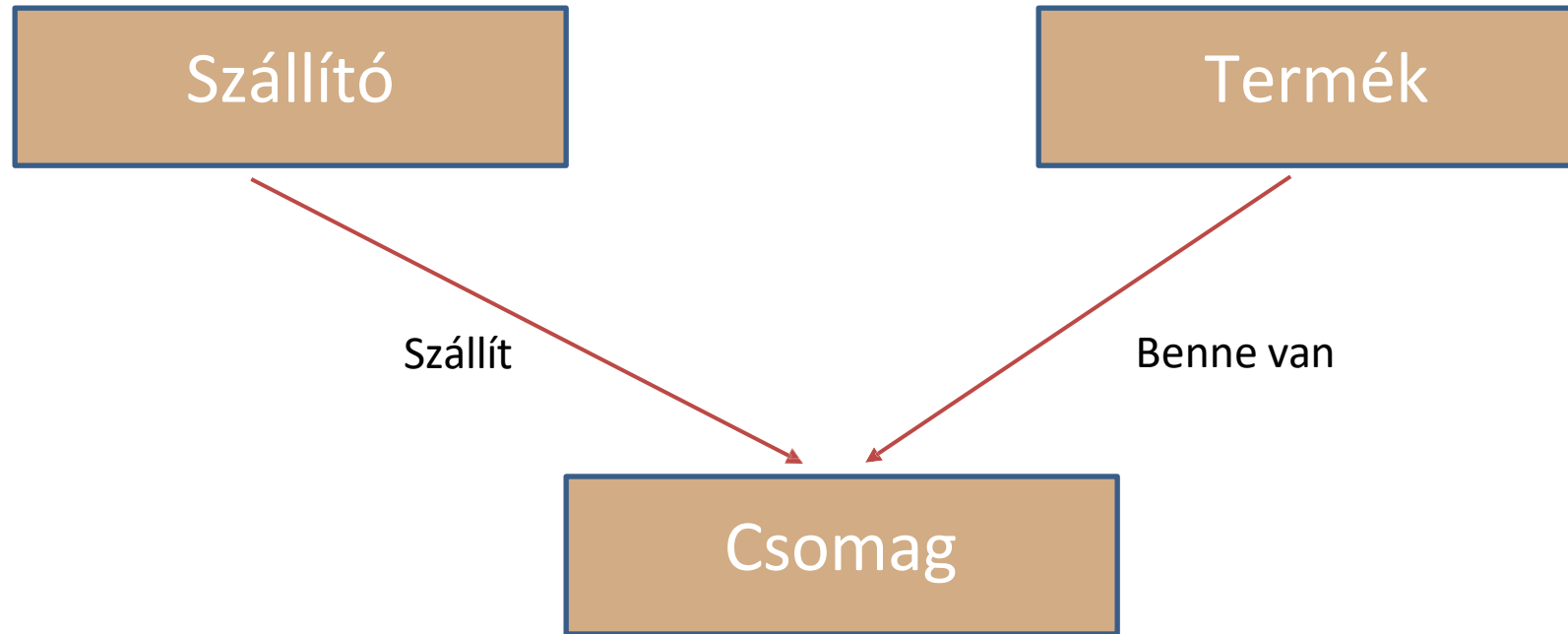
- ☐ Hálós adatmodell (gráf)
- ☐ Hierarchikus adatmodell (fa)
- ☐ **Relációs adatmodell (táblák)**
- ☐ Nem relációs adatmodellek

# Hálós adatmodell



Olyan gráf, ahol a  
csomópontok az egyedek, az  
élek a kapcsolatok

# Hálós adatmodell – példa (séma)



# Hálós adatmodell – példa (adatokkal)

## Szállító

Kod	Név	Cím
001	XY Kft	BP
002	ZZ Rt	Vác

## Termék

Kód	Név	EgysÁr
T01	Tej	250
T02	Tea	500
T03	Kóla	350

## Csomag

DB	Ár
25	50000
33	16500

Szülő	Gyerek

Szülő	Gyerek

# Hálós adatmodell – előnyök és hátrányok



Egyszerű elv

Többféle  
kapcsolat  
kezelése

Az adatok  
könnyen  
elérhetők

Adatintegritás,  
adatfüggetlenség



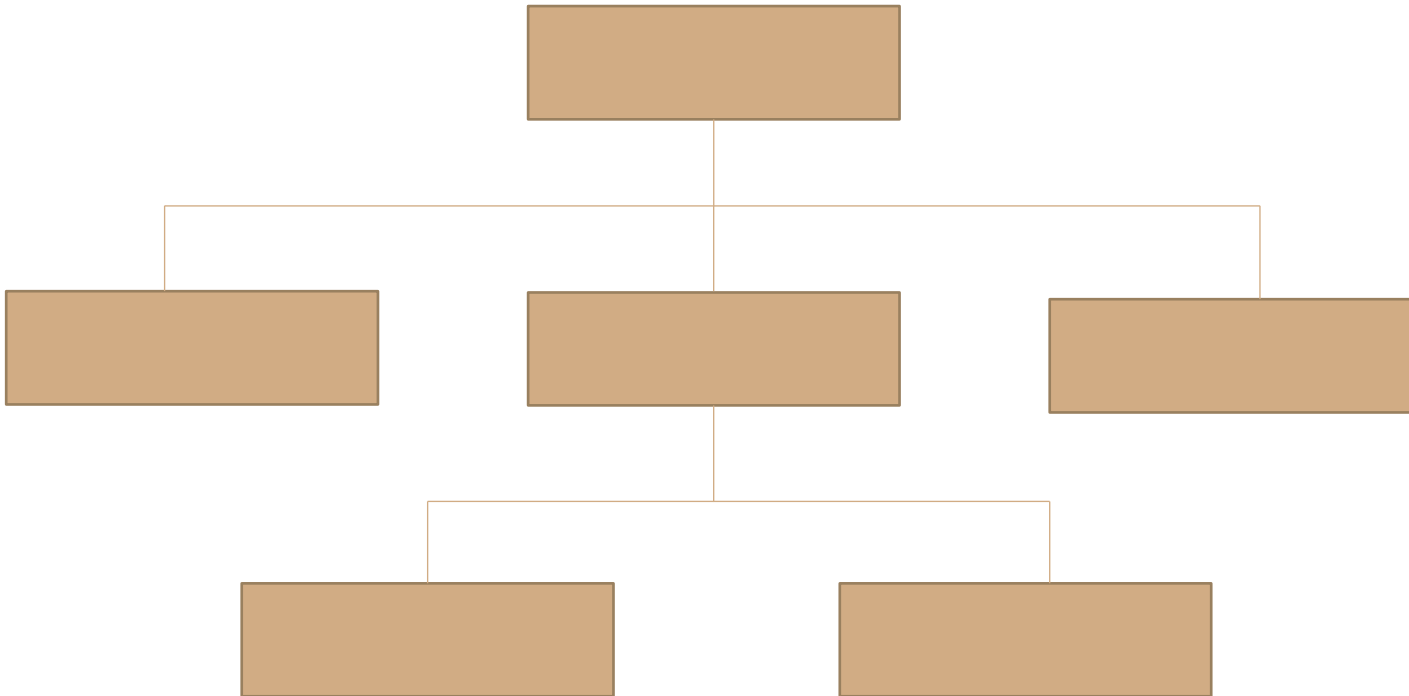
A mutatók  
kezelése

Komplex  
lekérdezések

Adatok  
törlése,  
módosítása

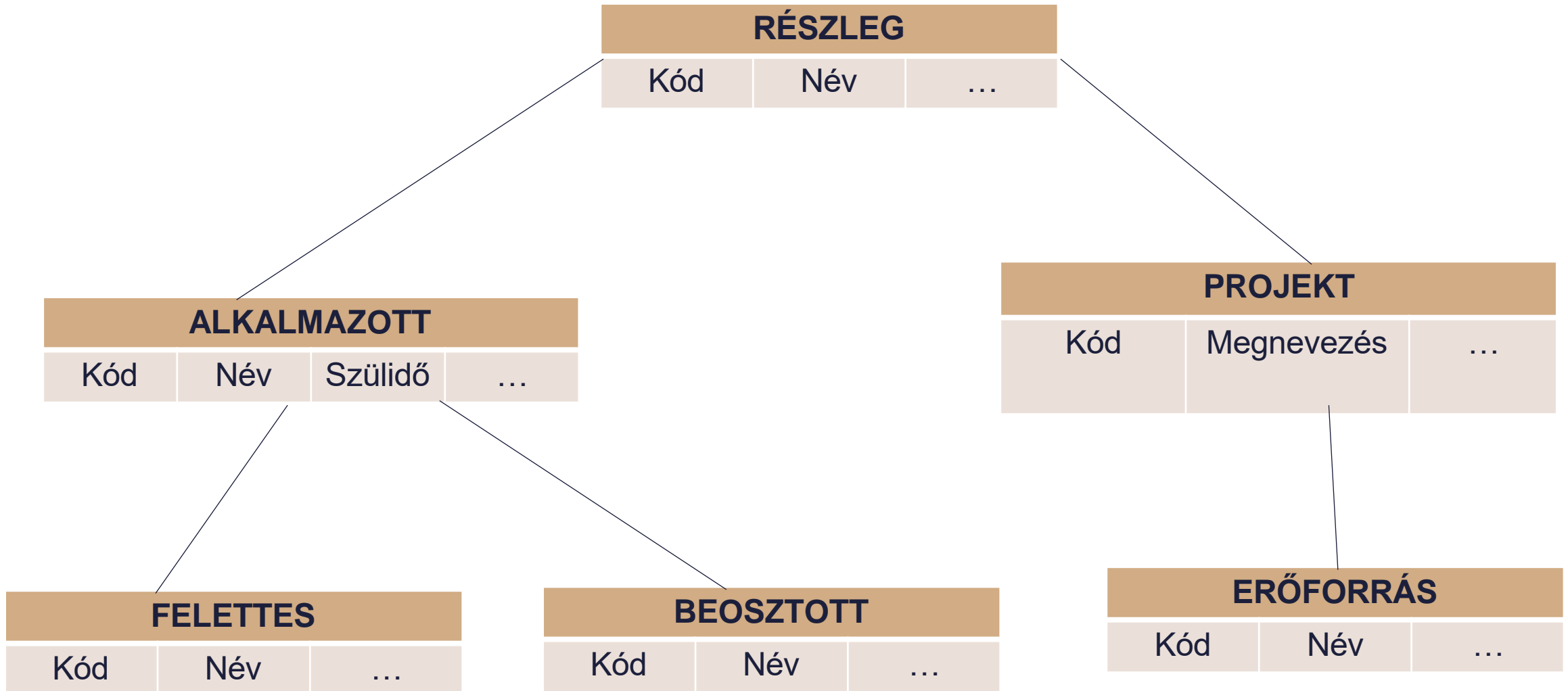
Séma  
módosítása  
nehéz

# Hierarchikus adatmodell



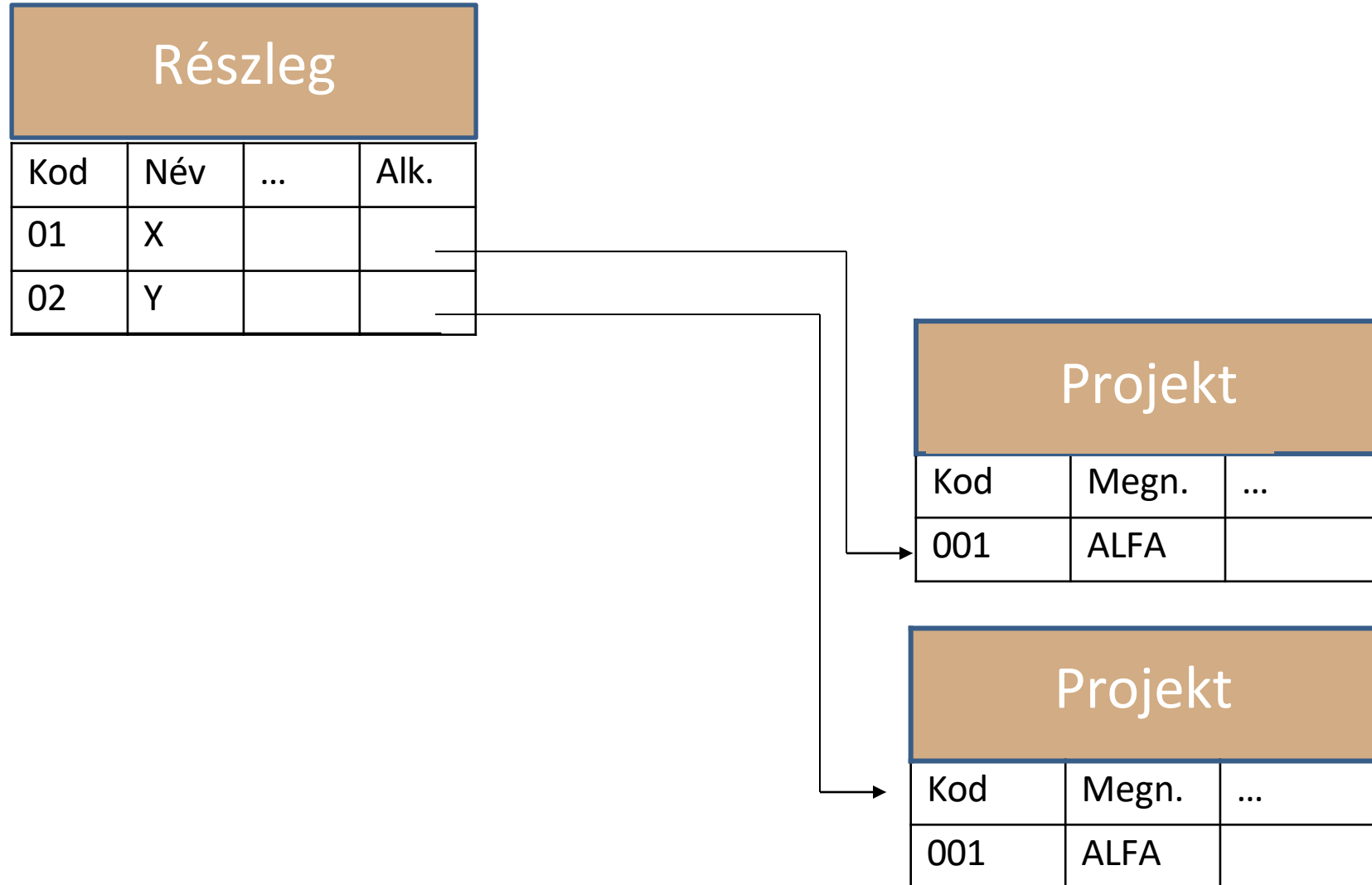
Olyan fa struktúra,  
ahol a csomópontok  
az egyedek, az élek  
a kapcsolatok

# Hierarchikus adatmodell – példa (séma)





# Hierarchikus adatmodell – példa (adatokkal) - részlet



# Hierarchikus adatmodell – előnyök és hátrányok



Egyszerű elv

Adatok  
megosztása

Az adatok  
gyorsan  
elérhetők

Természetes  
hierarchiák  
megvalósítása



Duplikációk

A fizikai  
adattárolás  
programfüggő

Adatok  
törlése,  
módosítása

Séma  
módosítása  
nehéz

# Edgar Codd (~1970)

- Felmérte a hálós és hierarchikus modellek hátrányait
- Új alapelveket javasolt (12 pontban)
- Ötleteinek lényege
  - Egyszerű adatszerkezetek használata
  - Adatok elérése magasszintű programnyelven
  - Fizikai adattárolás függetlensége

➔ Megalkotta a relációs adatmodellt

# Relációs adatmodell


## Relációs adatmodell esetén

- Az egyedek táblák (relációk)
- A tulajdonságok a táblák oszlopai (attribútumok)
- A kapcsolatok indirekt formában vannak jelen

# Relációs adatmodell - példa

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P



student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

# Mi történt 1980 és 2010 között?

- A relációs adatmodell egyeduralkodó lett
  - Megjelentek újabb adatmodellek (pl. objektumorientált), de tömegesen nem terjedtek el
  - Elterjedtek viszont az adattárházak
  - Tömegessé váltak az internetes, adatbázisokra épülő alkalmazások
  - A kezelendő adatmennyiség sokszorosára nőtt
- ➔ A relációs adatbázisok egyre több kihívással néztek szembe

## 2010 után

Nem relációs (NoSQL) adatmodellek megjelenése

- A legtöbb esetben egy régi ötlet újragondolását jelentik
- Nincs szabványos lekérdező nyelvük
- Részletesebben ld. 9-13. előadásokon

# Napjainkban + Közeljövőben

Relációs és nem relációs adatbázisok együttélése

- A közöttük lévő határok elmosódnak (pl. NewSQL)
- Hibrid rendszerek
- Többmodelles adatbázis rendszerek
- Felhős adatbázis rendszerek (DBaaS)
- Adattavak (Data Lake)



# Alapfogalmak - Adatbázis

Adatbázisnak nevezzük az egyedeknek és kapcsolataiknak valamilyen adatmodell szerinti elrendezését

# Alapfogalmak – Adatbáziskezelő rendszer

Programok olyan gyűjteménye ami lehetővé teszi a felhasználók számára adatbázisok készítését és fenntartását.

## Fontosabb feladatai:

- Adatok logikai és fizikai tárolása, karbantartása
- Adatok megjelenítése (lekérdezése)
- Egyidejű hozzáférések kezelése
- Jogosultságok kezelése
- Adatok mentése és helyreállítása
- Adatintegritás biztosítása
- Programozói és kommunikációs interfész biztosítása



ORACLE®

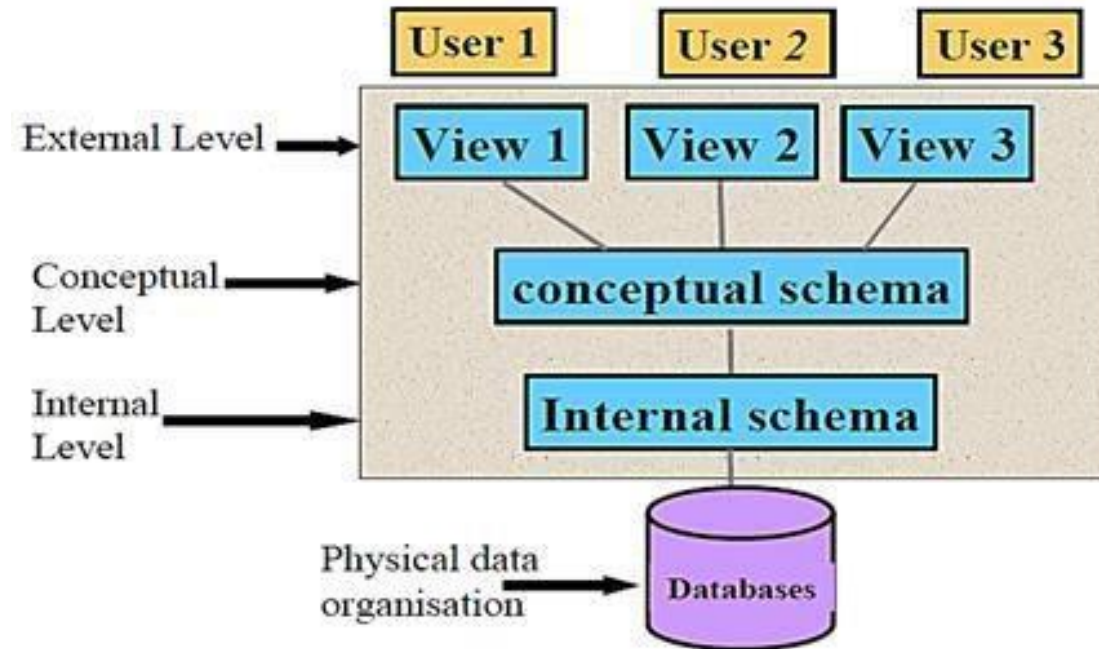


# ANSI/X3/SPARC modell

Külső szint

Konceptcionális  
szint

Belső szint



Az adatbáziskezelő  
rendszert három szintre  
osztja fel

Adatfüggetlenség:

Egy adott szinten történő változás nem érinti a felett lévő szinteket.

Pl: ha megváltoztatjuk a fizikai tárolás módját, az nem érinti a felhasználók lekérdezéseit

# Alapfogalmak – Adatbázis rendszer

Az adatbáziskezelő rendszer és az adatbázisok együttese

DBMS + DATABASE



**Köszönöm  
a figyelmet!**