

# Adatbázisok

# Adat-információ

Adat:

- **Oxford értelmező-szótár szerint:** Az **adat** egyfajta nyersanyag, amelynek feldolgozásával információ keletkezik, és amelyből az emberi tapasztalatok és képességek révén tudás lesz.
- **ISO szerint:** Az **adat** tények, fogalmak, eligazítások olyan **formai megjelenése**, amely **alkalmas** az emberi vagy az automatikus eszközök által történő **értelmezésre, vagy feldolgozásra**.

Információ:

**értelmezett adat**, ami bizonytalanságunkat csökkenti.

# Adatbázis fogalma

- Adatbázison köznapi értelemben adatok valamely célszerűen rendezett tárolását értjük.
- Az adatbázis esetén nem az adatok nagy számán van a hangsúly, hanem a célszerű rendezettségén.
- Az adathalmaz csak akkor válik adatbázissá, ha az valamilyen rend szerint épül fel, mely lehetővé teszi az adatok értelmes kezelését.
- Az adatbázisok mellé egy adatbáziskezelő rendszer (DBMS) is járul, mely az adatbázis vagy adatbázisok üzemeltetését biztosítja.

# Történelmi áttekintés

- Ókorban - kőtáblákra vagy papirusz tekercsekre írás.
- Az adatbázisok fejlettebb formái később a kartoték rendszerek lettek
- 50-es 60-as években az adatok tárolása még lyukszalagon, lyukkártyán történt, az adatok közvetlenül nem voltak elérhetők a számítógép számára.
- A mágneses háttértárolók elterjedésével az adatok tárolása egyszerűbbé, elérésük hatékonyabbá vált. Ezekben az időkben még nem léteztek univerzális módszerek illetve rendszerek, melyek segítségével az adatbázisokkal kapcsolatos problémák nagy része általánosan megoldható lett volna.

# Történelmi áttekintés

- A számítógépek fejlődésével együtt fejlődtek a programozói lehetőségek is. Az első számítógépeken csak a gépi kód (a bináris formában kiadott utasítások a mikroprocesszornak) állt rendelkezésre. Ezt első generációs programnyelvnek nevezzük.
- Ezt követték a második generációs (assembler) nyelvek, melyekben a gépi kód helyett úgynevezett mnemonikok és szimbólumok alkalmazhatók. Az első illetve második generációs programnyelvekben még nem készültek komoly adatbáziskezelő alkalmazások.
- Ezekre egyrészt a magas szintű nyelvek (3. generációs program nyelvek) COBOL, FORTRAN stb., másrészt a lemezes operációs rendszerek kialakulásáig kellett várni. Ekkor már komoly adatbázis alkalmazások születtek, melyek egyedi problémák megoldására voltak alkalmasak.
- Az adatbáziskezelés általános formában történő megfogalmazására jöttek létre az adatbázis kezelő rendszerek (DBMS) és a negyedik generációs nyelvek (4GL).
- Az objektum orientált programozási nyelvek térhódításával az adatbázis kezelő rendszerekkel kapcsolatos kutatások is az objektum orientált megközelítés irányába nyitott.

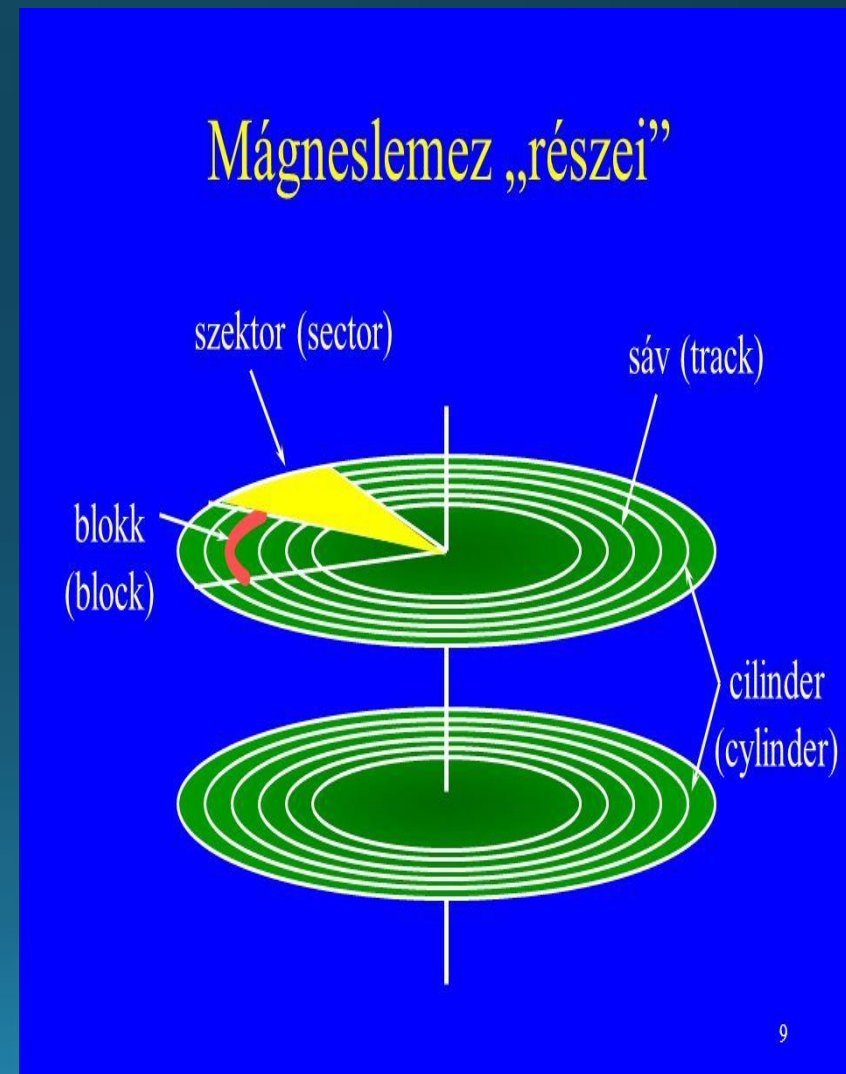
1. generációs nyelvek	Gépi kódú programozás
2. generációs nyelvek	Assembler nyelvek
3. generációs nyelvek	Imperatív nyelvek: Algol (1960), FORTRAN (1954), C (1973), Pascal (1970), Basic (1964)
4. generációs nyelvek)	SQL (1970), PL/SQL (1991), PL/pgSQL, MATLAB (1984
5. generációs nyelvek	Logikai programozási nyelvek: Prolog (1970), Lisp (1960)
Objektum orientált nyelvek	SmallTalk (1972), C++ (1981), Java (1994), Visual Basic
Script nyelvek	Perl (1987), Python (1991), PHP (1995), JavaScript (1995)

# Adatbáziskezelők szerepe, célja

- Adatbázis kezelő rendszer - DBMS Database Management System
- A DBMS-ek megváltoztatták a végfelhasználók adatnyerési lehetőségeit az **egyszerű lekérdezési nyelvek bevezetésével**. A lekérdező nyelvek lehetőséget nyújtanak a nem számítógépes szakemberek számára is **tetszőleges lekérdezés gyors végrehajtására**.
- A programozási eszközök mellett az operációs rendszerek illetve azoknak a háttértárakat kezelő része is komoly fejlődésen ment keresztül.
  - Nem volt már szükség a fizikai fájlstruktúra pontos ismeretére, ezt az operációs rendszer illetve az adatbáziskezelő rendszer elfedte a felhasználó és a programozó előtt is.
  - Az adatbázisokban gyakran előfordulnak olyan típusú adatok, melyeket az operációs rendszer vagy a harmadik generációs programnyelvek közvetlenül nem kezelnek, például dátum, időpont, pénzegység stb.

# Az adatbáziskezelők három alapvető feladata

- **Függetlenség az aktuális hardver konfigurációtól**
  - a fejlesztőnek ne kelljen törődnie a fizikai szintű adattárolással, ne kelljen közvetlenül lemez blokkokra, cilinderekre hivatkoznia.
- **Függetlenség az adatelérés módjától**
  - Az egyes operációs rendszerek a fájlokra többfajta adatelérési módot (szekvenciális, indexelt, véletlen) kínálnak az alkalmazások készítőinek.
- **Függetlenség az adatstruktúráktól**
  - Az adatbázisok szerkezetében beálló változások minél kevesebb módosítást okozzanak az alkalmazásokban.





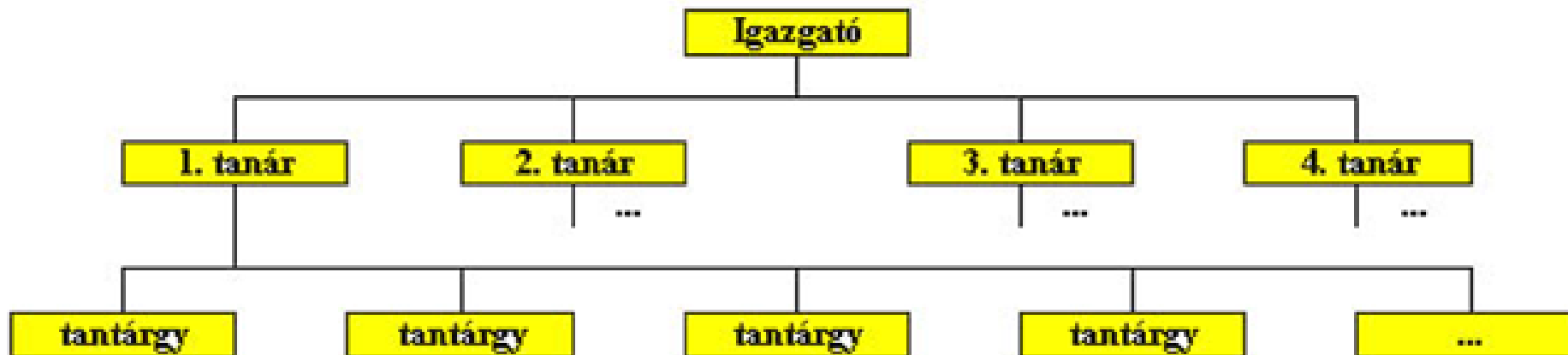
# Különböző adatbázis modellek

- Az adatbáziskezelők fejlődése során többfajta logikai modell alakult ki, melyek főként az adatok közötti kapcsolatok tárolásában térnek el egymástól.
- Hierarchikus,
- Hálós,
- **Relációs,**
- Objektum relációs és az objektum orientált modell.

Ezek közül manapság a Windows illetve UNIX operációs rendszerekben döntően a relációs modellre épülő adatbáziskezelőket használnak.

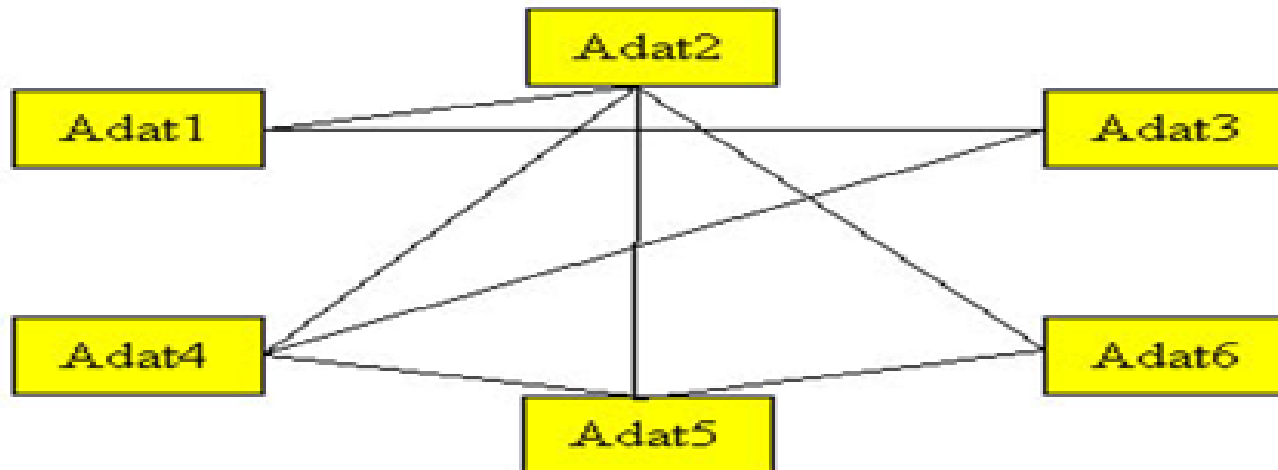
# Hierarchikus adatbázis modell

- A hierarchikus modell az 1960-s évek végén alakult ki és az 1970.s évek végéig használták. **IBM IMS**
- Szerkezetét gráffal adjuk meg:
  - Az adatbázis több egymástól független fából állhat. A fa csomópontjaiban és leveleiben helyezkednek el az adatok. A közöttük levő kapcsolat, szülő gyermek kapcsolatnak felel meg.



# Hálós adatbázis modell

- A hálós adatmodell esetén az egyes azonos vagy különböző összetételű adategységek (rekordok) között a kapcsolat egy gráffal írható le. A gráf csomópontok és ezeket összekötő élek rendszere, melyben tetszőleges két csomópont között akkor van adatkapcsolat, ha őket él köti össze egymással. Egy csomópontból tetszőleges számú él indulhat ki, de egy él csak két csomópontot köthet össze.



# Relációs adatbázis modell

- A relációs az egyik legáttekinthetőbb és a 80-as évektől kezdve a legelterjedtebb adatmodell.
- Kidolgozása E. F. Codd (1923-2003) nevéhez fűződik, 1970-ben jelent meg alapvető műve a "A Relational Model Data Large Shared Data Banks".
- A relációs modellben az adatokat táblázatok soraiban képezzük le.
- A legfontosabb eltérés az előzőekben bemutatott két modellhez képest az, hogy itt nincsenek előre definiált kapcsolatok az egyes adategységek között, hanem a kapcsolatok létrehozásához szükséges adatokat tároljuk többszörösen.
- Ezzel egy sokkal rugalmasabb és általánosabb szerkezetet kapunk.

# Relációs adatmodell

- Az egyedet táblázattal adjuk meg, a táblázat oszlopai a tulajdonságok, sorai pedig az egyed előfordulási (értékei).  
A táblázat egy-egy sorát a tulajdonságok konkrét értékei adják.
- A táblázat maga az egyedhalmaz.

Cikknév	Gyártó
Csavarkészlet	10001
Kalapács	10002
Vízszintező	10002

relációs kapcsolat

Gyártó száma	Gyártó	Telephely
10001	Nagy Bt.	Debrecen
10002	Meier Gmbh	Mannheim

# Objektum-relációs adatbázis modell

- Az objektum relációs adatmodell a relációs adatmodell bővítésével állt elő.
- Egyrészt az objektum orientált megközelítésben használt osztály, objektum, öröklődés fogalmakat alkalmazza az relációs adatbázis táblákra és a lekérdező nyelvet is ez irányba bővíti.
- Másrészt pedig támogatja az adatmodell bővítését saját adattípusokkal és azokat kezelő beépített függvényekkel.

# Relációs adatbázisok -részletesebben

- A relációs adatszerkezet egyszerűen értelmezhető a felhasználók és az alkalmazás készítőik számára is, így ez lehet közöttük a kommunikáció eszköze.
- A logikai adatmodell relációi egy relációs adatbáziskezelő rendszerbe módosítások nélkül átvihetők.
- A relációs modellben az adatbázistervezés a normál formák bevezetésével egzakt módon elvégezhető

# Néhány relációs adatbázis-kezelő

- Oracle
- MS SQL Server
- IBM DB2
- PostgreSQL
- MySQL
- SQLite



# Relációs adatbázisok, alapfogalmak

- A reláció nem más mint egy táblázat, a táblázat soraiban tárolt adatokkal együtt.
  - Az oszlopok névvel rendelkeznek, melyeknek a reláción belül egyedieknek kell lenniük
  - A reláció soraiban tároljuk a logikailag összetartozó adatokat.

## Fogalmak megfeleltetése:

Tábla= Egyed,  
Oszlop= Tulajdonság,  
Sor=Egyedtípus (Rekord),  
Mező (Cella) = Attribútum

	Oszlop			
Sor				
			Mező	

# Adatbázistervezés

- Az adatbázistervezés egy folyamat, mely több lépésből tevődik össze.
  - Először az adatbázisban leképezendő rendszert elemzésnek vetjük alá és meghatározzuk a tárolandó adatok körét, azok egymás közötti kapcsolatait és az adatbázissal szemben felmerülő igényeket
  - Ezután következik a rendszer tervezés, melynek eredménye az adatbázis logikai modellje.
  - Végül fizikai szinten képezzük le a logikai adatbázis modellt az alkalmazott szoftver és hardver függvényében.