

# Adatbázis-kezelő rendszerek I.



## ADATMODELLEK

# Adatmodellek- Bevezetés



# Adatmodellek



- Az adatmodellek fő kategóriái:
  - Koncepcionális (magas szintű, vagy objektum alapú) adatmodellek
  - Logikai (implementációs vagy rekord alapú) adatmodellek
  - Fizikai (alacsony szintű) adatmodellek
- Félig-strukturált adatmodell (XML, JSON)

# Adatmodellek



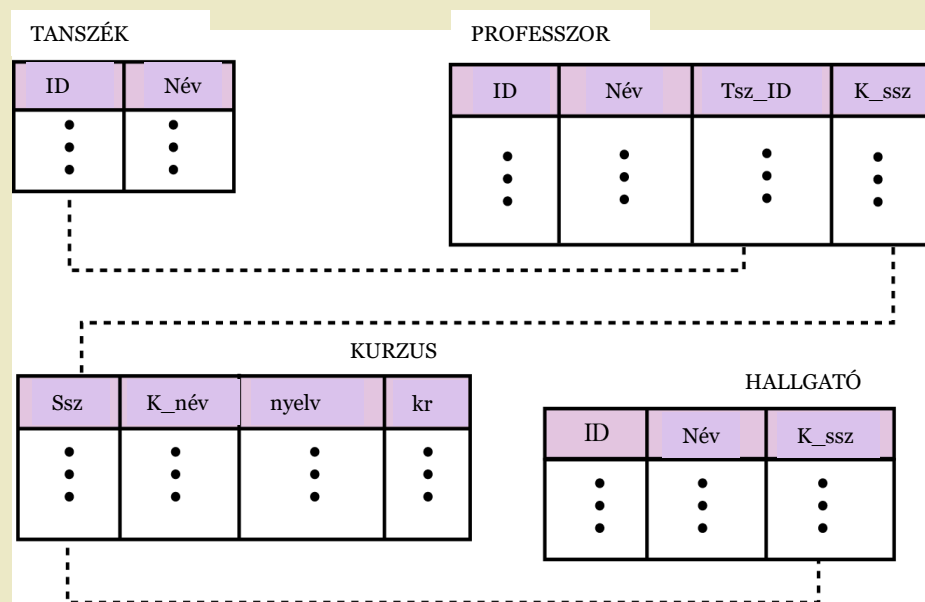
**JÓL DEFINIÁLT**

**More Details in: EN**

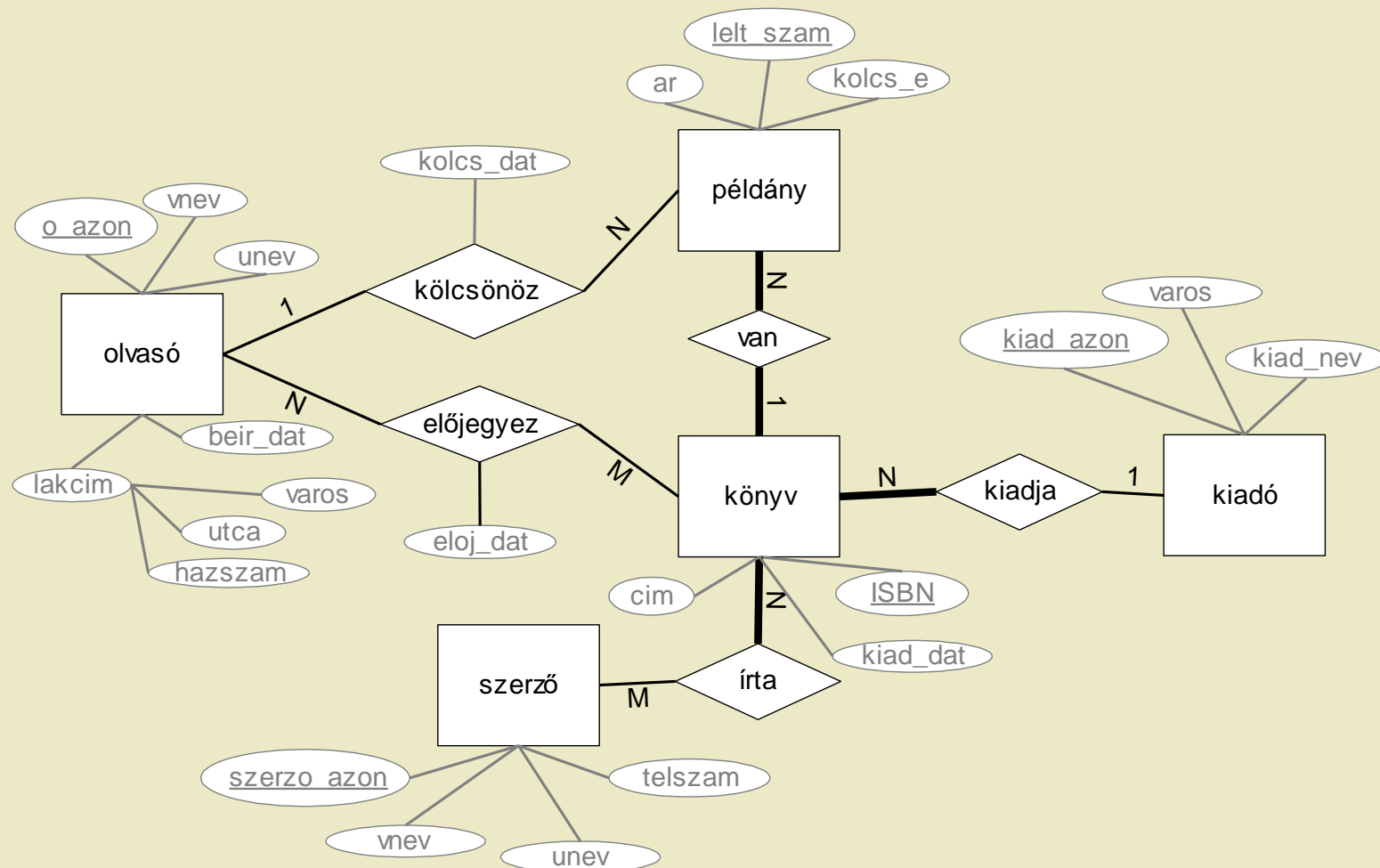
# A relációs modell



- Az első matematikai fogalmakon alapuló adatmodell
- Alapja a *reláció* matematikai fogalom.
- Az adatokat **táblázatokban** ábrázolja.
- A táblázatban az oszlopok egyedi névvel rendelkeznek.
- Nagyon egyszerű modell.
- Adatok lekérdezése magas szintű nyelvvvel, egyszerű, de hatékony.



# ER modell



# Az objektum-orientált modell



- Az objektum orientált modell a következő alapvető objektum-orientált fogalmakat tartalmazza:
  - **Objektum és objektum azonosító:** minden valós világbeli egyedet objektumként ír le, melyhez egyedi objektumazonosítót rendel.
  - **Attribútumok és metódusok:** minden objektumnak van egy állapota (az objektum attribútumértékeinek halmaza) és egy viselkedése (a metódusok – programkódok – halmaza, melyek az objektumok állapotán dolgoznak). Az állapot és a viselkedés az objektumba egységbe zárva található, csak külső üzenet által érhető el. Egy objektum egy másik objektum adatát kizárólag a másik objektum metódusának meghívása által érheti el (üzenetet küld az objektumnak).

Az objektum belső részei, vagyis az adattagok (állapotváltozók) és a metódusok kódjai kívülről nem láthatóak.

Egy attribútum tulajdonképpen egy objektumváltozó, melynek értelmezési tartománya tetszőleges osztály lehet: felhasználó által definiált, vagy primitív. Az objektumok értékeit **objektumváltozók** tárolják. A rekord alapú modellektől eltérően ezek önmagukban is objektumok. Így az objektumok tetszőleges mértékben egymásba ágyazhatóak.

# Az objektum-orientált modell (folyt.)



- Az objektum orientált modell a következő alapvető objektum-orientált fogalmakat tartalmazza:
  - **Osztály:** az azonos attribútumokkal és metódusokkal rendelkező objektumok gyűjteménye. Egy objektum pontosan egy osztályhoz tartozik, azon osztály egy konkrét előfordulása. Az osztály hasonlóan az absztrakt adattípushoz az objektumok típusdefiníciójaként tekinthető. Az osztály lehet primitív (interfész) is (nincs attribútuma), mint pl. egész, sztring, boolean.
  - **Osztályok hierarchiája és öröklés:** egy új osztály származtatása (gyerekosztály) egy létező osztályból (ősosztály). A gyerekosztály örökli a ősosztály összes attribútumát és metódusát, továbbá rendelkezhet új kiegészítő attribútumokkal és metódusokkal is. Egyszerű öröklődés vs. többszörös öröklődés.



# Példa



- Tekintsük a bankszámlát, mint objektumot.
  - Objektumváltozó: a *bankszámla száma* és a rajta található *összeg*.
  - Metódus: *kamatfizetés*
- A kamatrátá megváltoztatása számos adatmodellben az alkalmazás programkódjának változását eredményezheti. (nem feltétlenül!)
- Objektumorientált modellben ez a kamatfizetési metódus módosítását jelenti.

# Az objektum-orientált modell (folyt.)

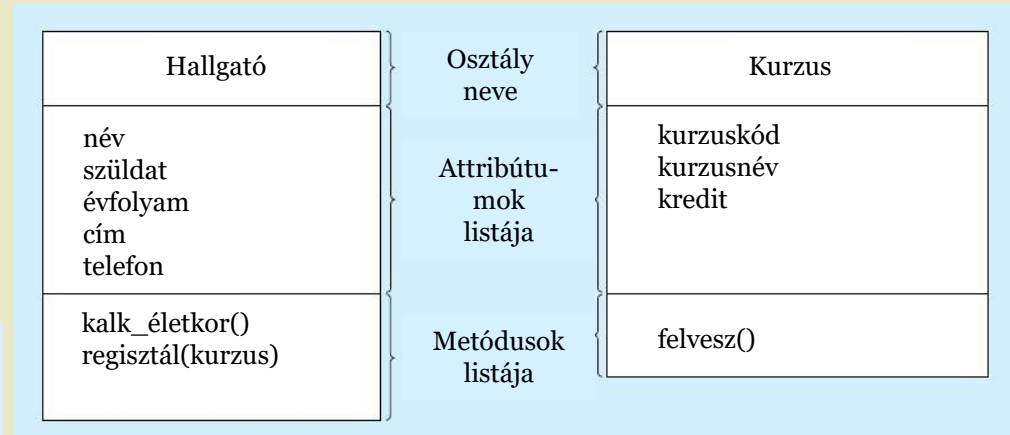
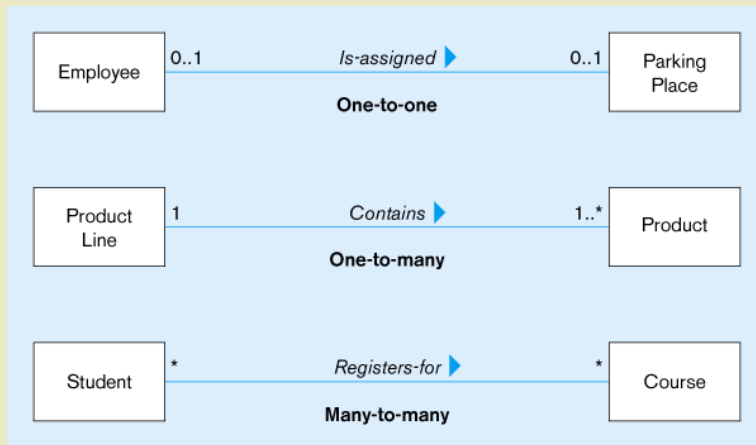


- Az ER modelltől különbözően minden objektumnak van egyedi azonosítója, amely független az objektum értékeitől.
  - Két objektum, amely ugyanazt az értéket tartalmazza különböző.
  - Az objektumok elkülönítése az objektumazonosítókön keresztül történik, fizikai szinten.

# Objektum-orientált modellezés



- **UML Osztály diagramm:** az objektum-orientált modell statikus struktúráját mutatja:
  - objektum osztályok,
  - osztályok belső struktúrája,
  - asszociációk/kapcsolatok



# Objektum-orientált modellezés



- **UML Objektum diagramm:** egy adott állapotot, az osztálydiagrammal kompatibilis objektumokat mutatja.

Mary Jones: Hallgató

név = Mary Jones  
születés = 1978.04.15.  
évfolyam = junior  
...

MIS385: Kurzus

kurzuskód = MIS385  
kurzusnév = Adatbázis  
kredit = 3

# Az objektum-relációs modell – Motiváció



- Relációs adatmodell: letisztult és egyszerű
  - Nagyszerűen használható az adminisztratív adatokhoz
  - nem annyira jó másféle adatokhoz (pl: multimédia, hálózatok, CAD)
- Objektum-orientált adatmodellek(80-as évek): bonyolult, de hatékony ötletekkel teli
  - Komplex adattípusok
  - Objektum szintű azonosítás/hivatkozás
  - Egységbe zártság (a tulajdonságok az adatokkal mozognak)
  - Öröklődés
- Ötlet: építsünk DBMS-t OO alapokon

# Az objektum-relációs adatmodell



- Objektum orientáltsággal egészíti ki a relációs adatmodellt és hozzáadott *adattípusokkal* (pl. sor típus) építkezik.
- Megőrzi a relációs alapokat, főképpen az adathozzáférést illetően, mialatt kiterjeszti modellező erejét.
- A rekordok attribútumaiként megenged komplex típusokat is, beleértve a nem atomi értékeket is, mint pl. *beágyazott relációk*.
- *Öröklődés*: típus és táblahierarchia
- A meglévő relációs nyelvekkel biztosítja a kompatibilitást.
- Az ORDBMS támogatása: SQL3

# A háromféle megközelítés különbségei



Kritérium	RDBMS	ODBMS	ORDBMS
<i>Szabványok</i>	SQL2	ODMG-2.0	SQL3 (folyamatban)
<i>Objektumorientáltság támogatása</i>	Nem támogatja	Hatékony támogatottság	Limitált támogatás; főként új adattípusok
<i>Használat</i>	Könnyű	A programozóknak OK; néhány SQL hozzáférés a végfelhasználóknak	Néhány kiterjesztéstől eltekintve könnyű
<i>Összetett kapcsolatok támogatása</i>	Nem támogatja az absztrakt adattípusokat	Támogatja az adattípusok széles körét és az adatok komplex belső kapcsolatrendszerét	Támogatja az absztrakt adattípusokat és komplex kapcsolatokat
<i>Teljesítmény</i>	Nagyon jó	Meglehetősen lassú	Elvárhatóan jó

# A háromféle megközelítés különbségei (folyt.)



Kritérium	RDBMS	ODBMS	ORDBMS
<i>Letisztultság</i>	Meglehetősen régi, ezáltal letisztult	Pár éves, relatíve letisztult	Folyamatosan fejlődik
<i>SQL használata</i>	Széleskörű SQL támogatottság	Az OQL hasonló az SQL-hez, de kiegészül objektum orientált elvekkkel és komplex adattípusokkal	SQL3 szabvány, benne OO jellemzők
<i>Előnyök</i>	SQL-el való kapcsolata, relatív egyszerű kérdések és optimalizáció, ezért gyors	Összetett alkalmazásokhoz használható, kód újrahasznosítása, kevesebb kódolás	Komplex alkalmazások lekérdezésének és használatának lehetősége
<i>Hátrányok</i>	Komplex alkalmazásokat nem tud kezelni	Alacsony sebesség az összetett optimalizációból fakadóan, nem képes nagyméretű rendszereket kezelni	Alacsonyabb teljesítmény web alkalmazásokban
<i>Támogatottsága</i>	Nagy támogatottság, de számos gyártó elmozdul az ORDBMS irányába	Jelenleg hiány az elterjedt RDBMS miatt	Valószínűleg szép jövő előtt áll, a fejlesztők ebbe az irányba mozdulnak



# Adatmodellek



**FÉLIG STRUKTÚRÁLT**

# XML: Extensible Markup Language



- A WWW Consortium (W3C) definiálta
- Kezdetben dokumentumok kezeléséhez fejlesztve, nem adatbázis-kezelő nyelv
- Az új tag-ek létrehozásának és a beágyazott tag struktúrák létrehozásának leheletősége jó lehetőséget teremt az **adatok cseréjére**.
- Az adatcsere egyik meghatározó formátumává vált.
- Eszközök széles választéka az XML dokum./adatok kereséséhez, elemzéséhez, lekérdezéséhez

```
<?xml version="1.0" ?>
<!-- Bookstore with no DTD -->
- <Bookstore>
- <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
  <Title>A First Course in Database Systems</Title>
  - <Authors>
  - <Author>
    <First_Name>Jeffrey</First_Name>
    <Last_Name>Ullman</Last_Name>
  </Author>
  - <Author>
    <First_Name>Jennifer</First_Name>
    <Last_Name>Widom</Last_Name>
  </Author>
  </Authors>
</Book>
- <Book ISBN="ISBN-0-13-815504-6" Price="100">
  <Remark>Buy this book bundled with "A First Course" -- a great deal! </Remark>
  <Title>Database Systems: The Complete Book</Title>
```

# JavaScript Object Notation (JSON)



- Széles körben terjedő formátum az adatok felsorolás jellegű file-ban történő tárolásához
- Könnyen olvasható, hasznos az alkalmazások közti adatcserék megvalósításához.
- Megfelelő háttér féligstrukturált adatok tárolására
- Parser számos nyelvhez
  - nem csak JavaScript-hez használható

```
{ "Books":  
  [  
    { "ISBN":"ISBN-0-13-713526-2",  
      "Price":85,  
      "Edition":3,  
      "Title":"A First Course in Database Systems",  
      "Authors":[ { "First_Name":"Jeffrey", "Last_Name":"Ullman"},  
                   { "First_Name":"Jennifer", "Last_Name":"Widom"} ] }  
    ,  
    { "ISBN":"ISBN-0-13-815504-6",  
      "Price":100,  
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",  
      "Title":"Database Systems:The Complete Book",  
      "Authors":[ { "First_Name":"Hector", "Last_Name":"Garcia-Molina"},  
                   { "First_Name":"Jeffrey", "Last_Name":"Ullman"},  
                   { "First_Name":"Jennifer", "Last_Name":"Widom"} ] }  
  ]  
}
```