



ELTE | IK  
INFORMATIKAI KAR

# Adatbázisok 2. előadás

Relációs adatmodell

# Mi az adatbázis?

---

- Összefüggő adatok praktikusan rendezett gyűjteménye és ezeknek valamelyen rendszerben való tárolása, amely a való világot valamelyen szempontból modellezi.
- Az adatbázis az alkalmazások egyik legfontosabb komponense. Nézzük meg a telefonunk alkalmazásait. Melyik használ adatbázist és melyik nem?
- Egy cégben a legértékesebb sokszor az adat, amivel rendelkezik.



# Adatbázis-kezelő rendszer (ABKR)

---

- Olyan szoftver, amely lehetőséget ad az adatok tárolására és kezelésére.
- Angolul: Database Management System (**DBMS**)
- Az általános célú DBMS feladatai:
  - Új adatbázisok létrehozása, ezek logikai szerkezetének, sémájának definiálása, adatdefiníciós nyelv (DDL, Data Definition Language),
  - Adatok lekérdezése, módosítása (DML, Data Manipulation Language)
  - Nagyméretű adatok hosszú időn keresztül történő tárolása, adatok biztonsága meghibásodásokkal, illetéktelen hozzáférőkkel szemben, hatékony adatbázishozzáférés,
  - Egyszerre több felhasználó egyidejű hozzáférésének biztosítása.

# Történelem

---

- Az első ABKR-ek az 1960-as években jelentek meg
  - IMS (1966) – hierarchikus adatmodell
  - IDS (1964) – hálós adatmodell
- Hátárnyok:
  - Nem támogattak magasabb szintű lekérdezőnyelveket.
  - A logikai és fizikai réteg nem volt szétválasztva.
  - Amikor az adatbázis szerkezete megváltozott, minden újra kellett írnia a programokat.

# Relációs adatmodell történelme

---

- Edgar Frank Codd 1969-ben javasolta a **relációs adatmodell** használatát.
- Egy projekt keretében létrehozta a System R-t, amely relációs megközelítést használt és implementálták az SQL egyik elődjét.
- Az egyszerűnek tűnő ötlet hamar elterjedt és nemsokra erre épültek a lengépszerűbb kereskedelmi ABKR-ek:
  - Oracle (1979)
  - IBM SQL/DS (1981)
  - IBM DB2 (1983)
  - Sybase (1984) – Microsoft SQL Server alapja



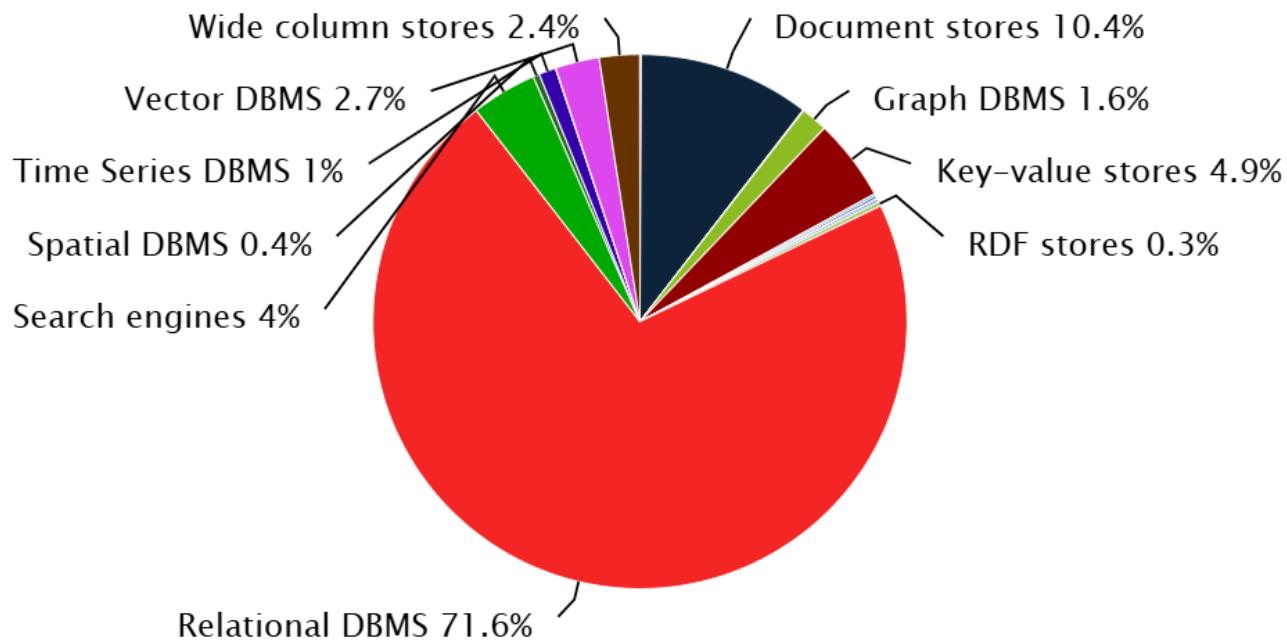
# Adatmodellek

---

- Az adatmodell az adatbázis logikai szerkezetének a leírása.
- Fontosabb adatmodellek:
  - **Relációs**
  - Kulcs-érték
  - Gráf
  - Dokumentum (JSON, XML)
  - Wide-column
  - Hierarchikus
  - Hálós
  - Szemantikus
  - Egyed-kapcsolat
  - Tömb (vektor, mátrix, tenzor)



# Adatmodellek népszerűsége



# ABKR-ek népszerűsége

Rank			DBMS	Database Model	Score		
Jul 2025	Jun 2025	Jul 2024			Jul 2025	Jun 2025	Jul 2024
1.	1.	1.	Oracle	Relational, Multi-model <a href="#">i</a>	1217.05	-13.33	-23.31
2.	2.	2.	MySQL	Relational, Multi-model <a href="#">i</a>	940.73	-12.85	-98.73
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model <a href="#">i</a>	771.14	-5.61	-36.51
4.	4.	4.	PostgreSQL <a href="#">+</a>	Relational, Multi-model <a href="#">i</a>	680.89	+0.23	+41.98
5.	5.	5.	MongoDB <a href="#">+</a>	Document, Multi-model <a href="#">i</a>	403.83	+0.99	-25.99
6.	6.	↑ 7.	Snowflake	Relational	176.17	+1.68	+39.64
7.	7.	↓ 6.	Redis	Key-value, Multi-model <a href="#">i</a>	149.72	-2.01	-7.05
8.	8.	↑ 9.	IBM Db2	Relational, Multi-model <a href="#">i</a>	127.51	+2.38	+3.11
9.	9.	↓ 8.	Elasticsearch	Multi-model <a href="#">i</a>	118.83	-2.45	-12.00
10.	10.	10.	SQLite	Relational	115.44	-1.60	+5.49



# Edgar Frank Codd 12 szabálya

---

## 1. Információs szabály

- minden adat táblákban, azaz sorok és oszlopok értékeiként jelenik meg.

## 2. Garantált lokalizálhatóság szabálya

- minden adat elérhető a tábla neve, sorazonosító (kulcs) és oszlopnév kombinációjával.

## 3. A NULL értékek egységes kezelése

- a hiányzó vagy nem értelmezhető adatot minden adattípus esetén egységesen kell kezelni.

## 4. A relációs modell alapján aktív online katalógust kell üzemben tartani

- a rendszernek támogatnia kell egy beépített katalógust, amelyet a felhasználók a lekérdező nyelv segítségével ugyanúgy le tudnak kérdezni, mint a közönséges táblákat.

## 5. A teljes körű „adatnyelv” szabálya

- minden tábla elérhető és kezelhető magas szintű relációs nyelven keresztül.

## 6. A nézetek frissítésének szabálya

- a rendszernek képesnek kell lennie az adatok összes nézetének frissítésére.

# Edgar Frank Codd 12 szabálya

---

## 7. Magas szintű beszúrás, frissítés és törlés

- A táblákban vagy nézetekben lehet adatot hozzáadni, módosítani, törölni magas szintű (nem rekordszintű) műveletekkel.

## 8. Fizikai szintű adatfüggetlenség

- Az adattárolás fizikai módjának változtatása nem befolyásolhatja a használatot vagy programokat.

## 9. Logikai szintű adatfüggetlenség

- A logikai változtatások (például új oszlop hozzáadása) nem zavarhatják az alkalmazásokat.

## 10. Jóság (integritás) függetlenség

- Az adatok jóságának (érvényességének) korlátait az adatfeldolgozási programuktól függetlenül kell tudni meghatározni, és azokat katalógusban kell nyilvántartani.

## 11. Elosztástól való függetlenség

- Az elosztott adattárolásnak átláthatatlannak kell lennie a felhasználók számára, mintha minden adat egy helyen lenne.

## 12. Megkerülhetetlenség szabálya

- Az alacsony szintű (rekord vagy mutató alapú) hozzáférés nem kerülheti meg a relációs nyelven kereszttüli integritási és biztonsági elveket.

# Relációs adatmodell I.

---

- A relációs adatmodellben az adatokat kétdimenziós táblákban (**relációkban**) tároljuk.
- A reláció fejrészében találhatók az **attribútumok**.
- minden attribútumhoz hozzáartozik egy értékkészlet.
- A reláció neve és a reláció-attribútumok halmaza együtt alkotják a **relációsémát**.
- A reláció attribútumainak sorrendje felcserélhető (nem változtatja meg a relációt).

## Sorozatok (cím, megjelenés, évadok)

cím	megjelenés	évadok
Lost	2004	6
Game of Thrones	2011	8
The Leftovers	2014	3

# Relációs adatmodell II.

---

- Egy-egy reláció soroknak egy halmaza.
- Halmaz: tehát nem számít a sorrend, valamint egy elem csak egyszer szerepelhet.
- A reláció sorainak halmazát előfordulásnak nevezzük.
- Általánosan:
  - R - relációnév
  - $A_i$  - attribútumok ( $i \in [1..n]$ )
  - $R(A_1, \dots, A_n)$  – relációséma
  - $Dom(A_i)$  – lehetséges értékek halmaza (tartomány, domén)
- r – relációelőfordulás  
 $r \subseteq Dom(A_1) \times \dots \times Dom(A_n)$



# Relációs adatmodell: kulcs

---

- Az attribútumok egy halmaza egy **kulcsot** alkot egy relációra nézve, ha a reláció bármely előfordulásában nincs két olyan sor, amelyek a kulcs összes attribútumának értékein megegyeznének.
- Megjegyzés: egy kulcs nem feltétlenül egy attribútumból áll.
- A kulcsot aláhúzás jelöli:

**Sorozatok** (cím, megjelenés, évadok)

cím	megjelenés	évadok
Lost	2004	6
Game of Thrones	2011	8
The Leftovers	2014	3



# Relációs adatmodell: kulcs

---

- A gyakorlatban az ABKR-ek, ha nincs meghatározva kulcs, akkor automatikusan új oszlopot készítenek, amelyet elsődleges kulcsként használnak.
  - PostgreSQL/Oracle: **SEQUENCE**
  - MySQL: **AUTO\_INCREMENT**

**Sorozatok** (**id**, cím, megjelenés, évadok)

<b>id</b>	<b>cím</b>	<b>megjelenés</b>	<b>évadok</b>
1	Lost	2004	6
2	Game of Thrones	2011	8
3	The Leftovers	2014	3



# Relációs algebra

---

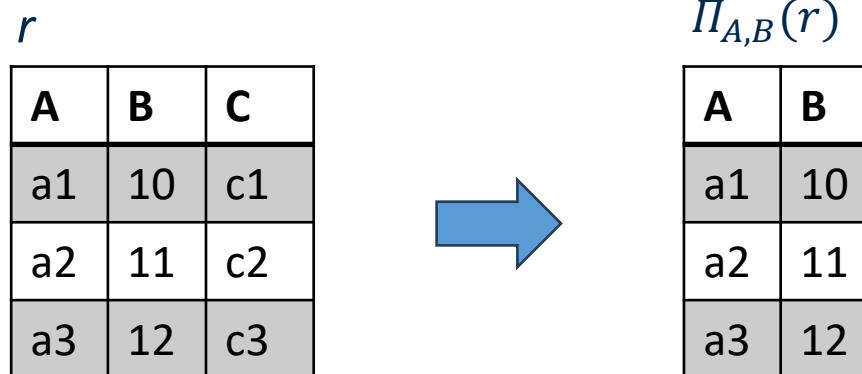
- A relációs algebra egy adatmanipulációs nyelv, amelynek az operandusai a relációkat reprezentáló változók és konstansok.
- minden művelet végeredménye reláció, amelyen további műveletek adhatók meg.
- Alapműveletek:
  - Projekció (vetítés)
  - Szelekció (kiválasztás)
  - Unió
  - Metszet - származtatott
  - Különbség
  - Descartes-szorzat
  - Théta-összekapcsolás és természetes összekapcsolás - származtatott
  - Átnevezés



# Projekció (vetítés)

---

- Adott relációt vetít le az alsó indexben szereplő attribútumokra.
- $\Pi_X(r) := \{t \mid \exists t' \in r, t'[X] = t\}$ , ahol X a reláció attribútumainak a részhalmaza.
- $|\Pi_X(r)| \leq |r|$



```
SELECT A, B
FROM R;
```



# Szelekció (kiválasztás)

- Kiválasztja az argumentumban szereplő reláció azon sorait, amelyek eleget tesznek az alsó indexben szereplő feltételnek.
- $\sigma_F(r) := \{t \mid t \in r \wedge F(t) = \text{IGAZ}\}$ , ahol  $F$  felépítése a következő:
  - atomi feltétel:  $A_i \Theta A_j$  vagy  $A_i \Theta c$ ,  
ahol  $c$  konstans és  $\Theta \in \{=, <, >, \leq, \geq, \neq\}$
  - ha  $F_1$  és  $F_2$  feltételek, akkor  $\neg F_1, F_1 \wedge F_2, F_1 \vee F_2$  is feltételek

```
SELECT *
FROM R
WHERE B >= 11;
```

r	A	B	C
	a1	10	c1
	a2	11	c2
	a3	12	c3

$\sigma_{B \geq 11}(r)$	A	B	C
	a2	11	c2
	a3	12	c3



# Szelekció (kiválasztás) - folytatás

---

- $|\sigma_F(r)| \leq |r|$
- A feltételben függvénynek nem használhatók.
- Az összetett feltételek átírhatók elemi feltételeket használó kifejezésekkel a következő szabályok segítségével:
  - $\sigma_{F_1 \wedge F_2}(r) \cong \sigma_{F_1}(\sigma_{F_2}(r)) \cong \sigma_{F_2}(\sigma_{F_1}(r))$
  - $\sigma_{F_1 \vee F_2}(r) \cong \sigma_{F_1}(r) \cup \sigma_{F_2}(r)$
- A De Morgan azonosság segítségével a negáció beljebb vihető:
  - $\neg(F_1 \wedge F_2) \cong (\neg F_1) \vee (\neg F_2)$
  - $\neg(F_1 \vee F_2) \cong (\neg F_1) \wedge (\neg F_2)$
- Elemi feltétel tagadása helyett használhatjuk a fordított összehasonlítást:  $\neg(A < B)$  helyett  $(A \geq B)$



# Unió

---

- $r, s$  és  $r \cup s$  azonos sémájú relációk.
- $r \cup s := \{t \mid t \in r \vee t \in s\}$
- $|r \cup s| \leq |r| + |s|$ , ahol  $|.|$  egy reláció sorainak a számát adja meg.
- Mivel a reláció egy sorhalmaz, azért azonos sor csak egyszer szerepel.

***r***

A	B	C
a1	10	c1
a2	11	c2
a3	12	c3

***s***

A	B	C
a3	12	c3
a4	14	c4

***r ∪ s***

A	B	C
a1	10	c1
a2	11	c2
a3	12	c3
a4	14	c4

```
SELECT * FROM R
UNION
SELECT * FROM S;
```



# Különbség

---

- $r, s$  és  $r - s$  azonos sémájú relációk.
- $r - s := \{t \mid t \in r \wedge t \notin s\}$
- $|r - s| \leq |r|$

***r***

A	B	C
a1	10	c1
a2	11	c2
a3	12	c3

***s***

A	B	C
a3	12	c3
a4	14	c4

***r - s***

A	B	C
a1	10	c1
a2	11	c2

```
SELECT * FROM R
EXCEPT
SELECT * FROM S;
```



# Metszet

---

- $r, s$  és  $r - s$  azonos sémájú relációk.
- $r \cap s := \{t \mid t \in r \wedge t \in s\}$
- $|r \cap s| \leq |r|$  és  $|r \cap s| \leq |s|$

<b>r</b>	A	B	C
A	a1	10	c1
A	a2	11	c2
A	a3	12	c3

<b>s</b>	A	B	C
A	a3	12	c3
A	a4	14	c4

<b><math>r \cap s</math></b>	A	B	C
A	a3	12	c3

```
SELECT * FROM R
INTERSECT
SELECT * FROM S;
```

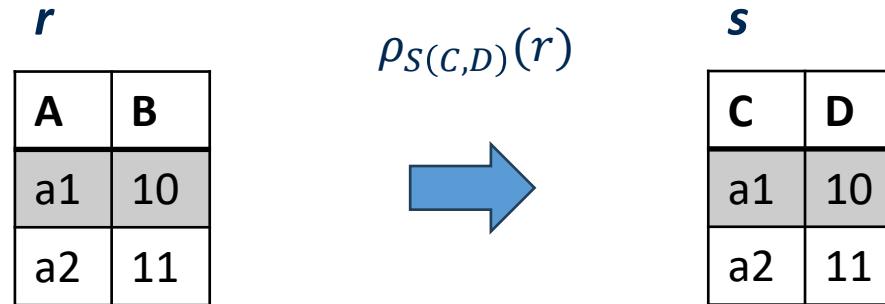
A metszet **származtatható** korábbi műveletekből:  
 $r \cap s = r - (r - s) = s - (s - r)$



# Átnevezés

---

- A relációnak és az attribútumoknak új nevet adhatunk.
- Ha az  $r$  reláció sémája  $R(A_1, \dots, A_n)$ , akkor  $\rho_{S(B_1, \dots, B_n)}(r)$  sémája  $S(B_1, \dots, B_n)$
- Ha az attribútumokat nem szeretnénk átnevezni, csak a relációt, ezt így jelöljük:  $\rho_s(r)$



```
SELECT A as C, B as D
FROM R as S;
```



# Szorzás, direktszorzat vagy Descartes-szorzat

---

- Olyan relációt készít, amely sorpárokat tartalmaz, a két reláció rekordjainak összes lehetséges kombinációját. Az azonos nevű attribútumokat meg kell különböztetni egymástól.  $r \times s$  sémája  $r$  és  $s$  sémájának az egyesítése.
- $r \times s = \{t \mid t[R] \in r \wedge t[S] \in s\}$
- $|r \times s| = |r| \cdot |s|$

<b>r</b>	
A	B
a1	10
a2	11
a3	12

<b>s</b>	
B	C
12	c3
14	c4

<b><math>r \times s</math></b>			
A	R.B	S.B	C
a1	10	12	c3
a1	10	14	c4
a2	11	12	c3
a2	11	14	c4
a3	12	12	c3
a3	12	14	c4

```
SELECT * FROM R CROSS JOIN S;
```

```
SELECT * FROM R, S;
```



# Théta-összekapcsolás

---

- Olyan sorpárokat ad vissza, amelyek kielégítenek egy feltételt.
- Jelölése:  $r \bowtie_{\Theta} s$ , ahol  $\Theta$  egy feltétel az  $r$  és  $s$  relációk egy-egy attribútuma között.
- Származtatható korábbi műveletekből:  $r \bowtie_{\Theta} s = \sigma_{\Theta}(r \times s)$

 $r$ 

A	B
a1	10
a2	11

 $s$ 

C	D
11	c1
12	c2

 $r \bowtie_{B=C} s$ 

A	R.B	S.B	C
a3	12	12	c3

```
SELECT *
FROM R JOIN S
ON R.B = S.B;
```

$F_1 = F_2$  feltétel esetén egyen-összekapcsolásnak hívjuk.



# Természetes összekapcsolás

---

- Olyan sorpárokat ad vissza, amelyek a relációk azonos attribútumain megegyeznek. Jelölése:  $r \bowtie s$
- Származtatható korábbi műveletekből:  $r \bowtie s = \Pi_L(\sigma_C(r \times s))$
- Ahol C a közös attribútumok egyenlőségét írja el, L pedig csak egyszer veszi a közös attribútumokat.

***r***

A	B
a1	11
a2	12
a3	12

***s***

B	C
11	c1
12	c2

***r*  $\bowtie$  *s***

A	B	C
a1	11	c1
a2	12	c2
a3	12	c2

```
SELECT *
FROM R NATURAL JOIN S;
```



# Félig összekapcsolás

---

- Azokat a sorokat adja vissza az első relációból, amelyikhez van kapcsolható sor a második relációból (származtatható korábbi műveletekből)
- Ha s és r relációk és r sémája  $R(A_1, \dots, A_n)$ , akkor:
- $r \bowtie s = \Pi_{A_1, \dots, A_n}(r \bowtie s)$

<i>r</i>	
A	B
a1	11
a2	12
a3	13

<i>s</i>	
B	C
11	c1
12	c2

$r \bowtie s$	
A	B
a1	11
a2	12

```
SELECT R.A, R.B
FROM R NATURAL JOIN S;
```



# Egyéb összekapcsolások

---

- **Külső összekapcsolás** – az  $r \bowtie s$  relációt kiegészítjük az  $r$  és  $s$  soraival, a hiányzó helyekre NULL értéket írva.
  - Nem relációs algebrai művelet, mivel kilép a modellből.
- **Anti-join** (tagadó összekapcsolás) – a félig összekapcsolás ellentettje, azokat a sorokat adja vissza a relációból, amelyeknek nincs párjuk a másik relációban.
- Ezeknek az összekapcsolásoknak általában nincs saját szintaktikájuk SQL-ben, más műveletekkel érjük el a hatásukat (a félig összekapcsolásnak sincs).



# Összekapcsolások - megjegyzések

---

- Ha  $r$  és  $s$  sémái megegyeznek, akkor:  $r \bowtie s = r \cap s$
- Ha  $r$  és  $s$  sémáiban nincs közös attribútum, akkor:  $r \bowtie s = r \times s$
- Ha  $r = \emptyset$ , akkor  $r \times s = \emptyset$  és  $r \bowtie s = \emptyset$

# Egyéb műveletek

---

- Értékadás
- Osztás (hányados)
- Kiterjesztett relációs algebra (nem lesz rá szükségünk):
  - Ismétlődések megszüntetése -  $\delta$
  - Aggregáció (összesítés) -  $\gamma$
  - Rendezés -  $\tau$

# Műveletek monotonitása

---

- Egy relációkon értelmezett operátor akkor **monoton**, ha bármelyik argumentumrelációhoz egy újabb sort hozzávéve az eredmény tartalmazza az összes olyan sort, amelyet addig tartalmazott és esetleg újabb sorokat is.
- Formálisan, ha  $R_i \subseteq S_i, i = 1, \dots, n$ , akkor  $E(R_1, \dots, R_n) \subseteq E(S_1, \dots, S_n)$ , ahol  $E$  egy relációs algebrai operátor.
- A kivonás kivétel az alapműveletek monoton műveletek.
- **DE** monoton kifejezésben szerepelhet kivonás:  $r \cap s = r - (r - s)$



# Relációkra vonatkozó megszorítások

---

- Megszorítások relációs algebrában is kifejezhetőek.
- Tegyük fel, hogy  $r$  és  $s$  relációs algebrai kifejezések (relációk):
  - $r = \emptyset$ , azaz  $r$ -nek üresnek kell lennie.
  - $r \subseteq s$ , azaz  $r$  eredményének minden sorának benne kell lennie  $s$  eredményében.
- Vegyük észre, hogy a két megszorítás kifejező erő szempontjából azonos:
  - $r \subseteq s$  így is kifejezhető:  $r - s \subseteq \emptyset$



# Kulcs megszorítás

---

- A kulcs megszorítás kifejezhető relációs algebrában.
- Adott  $R$  reláció a következő sémával:  $R(A, B)$ , ahol  $A$  kulcs.
- $\sigma_{R_1.A=R_2.A \wedge R_1.B \neq R_2.B}(\rho_{R_1}(R) \times \rho_{R_2}(R)) = \emptyset$

# Hivatkozási épség megszorítás

---

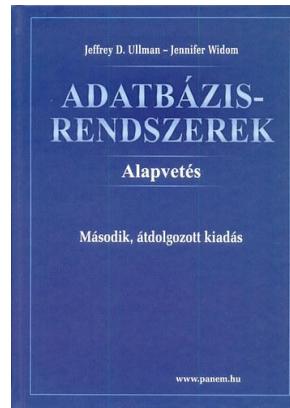
- Hivatkozási épség megszorítás: ha egy érték megjelenik valahol egy környezetben, akkor ugyanez az érték egy másik, az előzővel összefüggő környezetben is meg kell, hogy jelenjen.
- Adott  $R$  reláció  $R(A, B)$  sémával és  $S$  reláció  $S(C, D)$  sémával.
- Követeljük meg, hogy az  $R$  reláció B oszlopában csak olyan értékek szerepeljenek, amelyek előfordulnak az  $S$  reláció C oszlopában:
- $\Pi_B(R) \subseteq \Pi_c(S)$



# Tankönyv fejezetek

---

- Jeffrey D. Ullman, Jennifer Widom: Adatbázisrendszerek – Alapvetés
  - I. rész, 2. fejezet – A relációs adatmodell (17 – 68. oldalak)



# Összefoglalás

---

- A kurzus az ABKR-ek implementálásáról szól.
- A relációs ABKR-ek széles körben elterjedtek és még nagyon hosszú ideig velünk lesznek.
- A lekérdezések optimalizációjához részben relációs algebrai azonosságokat fogunk használni.