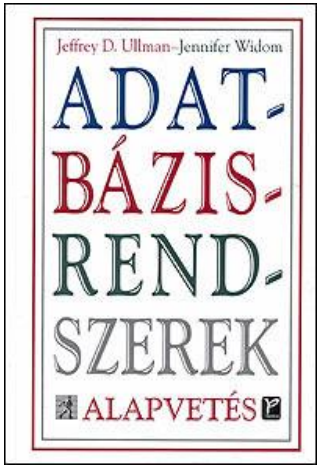
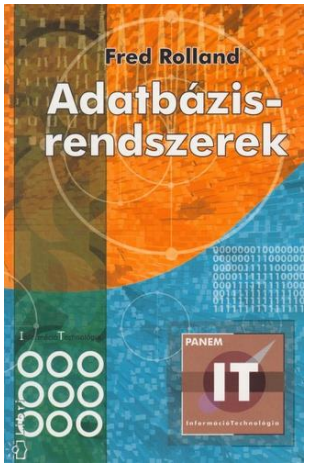
Adatbázisok 2

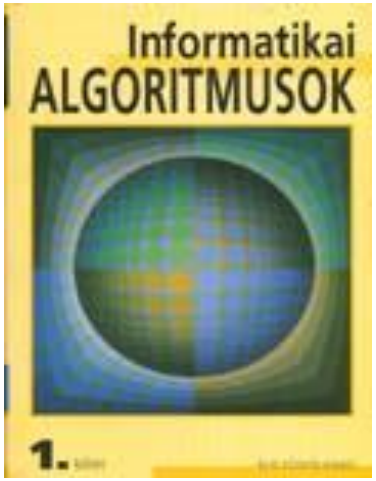
Kiss Attila kiss@inf.elte.hu

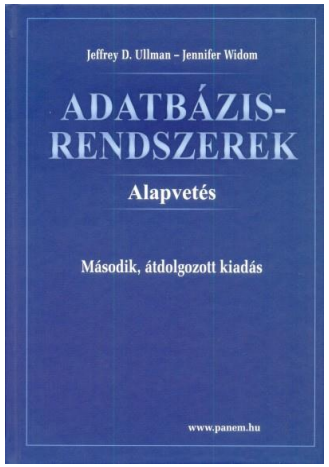
- A kurzus tematikája:
- Lekérdezések optimalizálása
  - relációs algebrai optimalizáció, logikai lekérdező terv
  - fizikai fájlszervezés alapjai
  - fizikai lekérdező terv, operátorok költsége
  - több tábla összekapcsolása
  - Oracle SQL optimalizálása
- Adatbázis rendszerhibák kezelése
  - Naplózás, helyreállítás, ellenőrzőpontok
  - Oracle naplózási technikái
- Adatbázisok konkurens használata
  - Tranzakciók, ütemezése tulajdonságai
  - Szérializálhatóság biztosítása, aktív (zártípusok, időbélyegzők) és passzív módszerek
  - Az Oracle zárolási technológiája

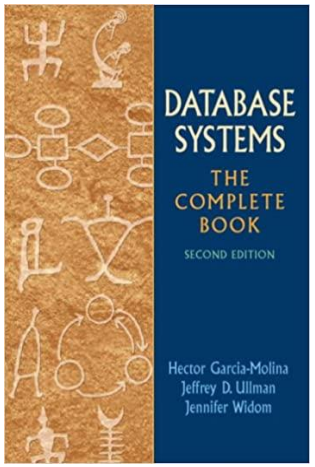
IRODALOM

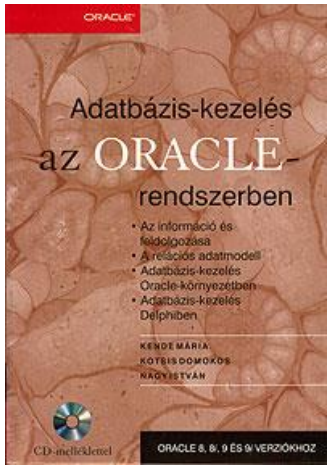


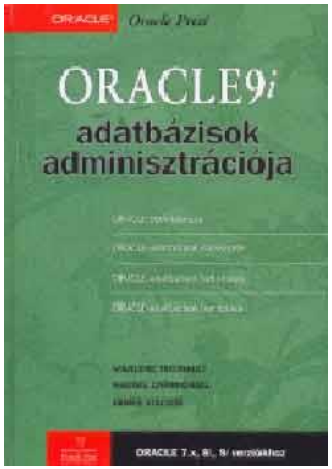


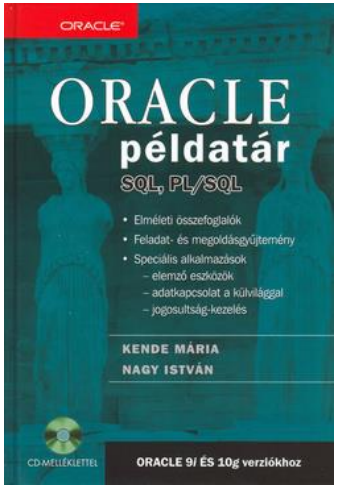












https://people.inf.elte.hu/kiss/

ISMÉTLÉS
(Az Adatbázis 1 tárgyban tanultak)

1. **Adatbázis-kezelő rendszerek** általános jellemzői.
2. **A relációs adatmodell,** a relációs algebra műveletei, használata
3. **Az SQL nyelv részei** (ORACLE specifikusan):

   -DDL, DML QL, triggerek, jogosultságok, PL/SQL, függvények, procedúrák, cursorok használata, programozás,
4. **Adatmodellezés**, egyed-kapcsolat modell, az E/K diagram átalakítása relációs adatmodellé.

Adatbázisrendszerek

ABR1 1. fejezet (19.- 45. oldal)

- **Adatbázis-kezelés:**
  - **Háttértárolón tárolt, nagy adatmennyiség hatékony kezelése (lekérdezése, módosítása)**
  - Adatmodell támogatása
  - Adatbázis-kezelő nyelvek támogatása
  - Több felhasználó támogatása
  - Tranzakció-kezelés
  - Helyreállíthatóság
  - Ügyfél-kiszolgáló felépítés
  - Adatvédelem, adatbiztonság

Adatmodellek

- **Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása**
  - Hálós, hierarchikus adatmodell (apa-fiú kapcsolatok gráfja, hatékony keresés)
  - Relációs adatmodell (táblák rendszere, könnyen megfogalmazható műveletek)
  - Objektum-orientált adatmodell (az adatbázis-kezelés funkcionalitásainak biztosítása érdekében gyakran relációs adatmodellre épül)
  - Logikai adatmodell (szakértői rendszerek, tények és következtetési szabályok rendszere)
  - Félig strukturált (XML) adatmodell

Adatbázis-kezelő nyelvek

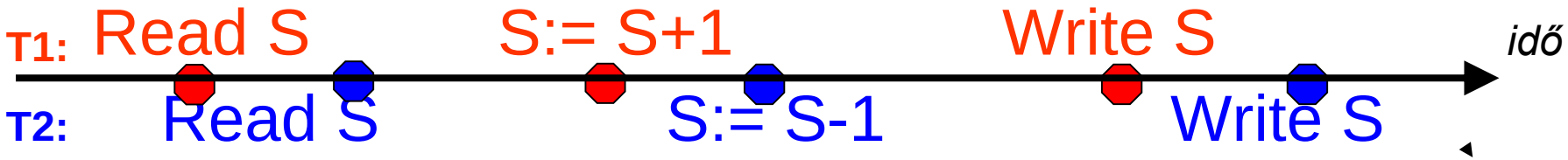
- **DDL** – adatdefiniáló nyelv (sémák, adatstruktúrák megadása)
- **DML** – adatkezelő nyelv (beszúrás, törlés, módosítás)
- **QL** – lekérdező nyelv
  - **Deklaratív** (SQL, kalkulusok)
  - **Procedurális** (relációs algebra)
- **PL/SQL** – programozási szerkezetek + SQL
- **Programozási nyelvbe ágyazás** (előfordító használata)
- **4GL** nyelvek (alkalmazások generálása)

Több felhasználó támogatása

- **Felhasználói csoportok**
- **DBA** – adatbázis-rendszergazda
- Jogosultságok (objektumok olvasása, írása, módosítása, készítése, törlése, jogok továbbadása, jogok visszavonása)
- Jogosultságok tárolása rendszertáblákban történik

Tranzakció-kezelés

- **Tranzakció**: adatkezelő műveletekből (adategység írása, olvasása) álló sorozat
- Cél: tranzakciók párhuzamos végrehajtása



- A tranzakció-kezelő biztosítja:
  - **Atomosság** (a tranzakció egységesen lefut vagy nem)
  - **Következetesség** (a tranzakció futása után konzisztens legyen az adatbázis)
  - **Elkülönítés** (párhuzamos végrehajtás eredménye egymás utáni végrehajtással egyezzen meg)
  - **Tartósság** (a befejezett tranzakció eredménye rendszerhiba esetén sem veszhet el)

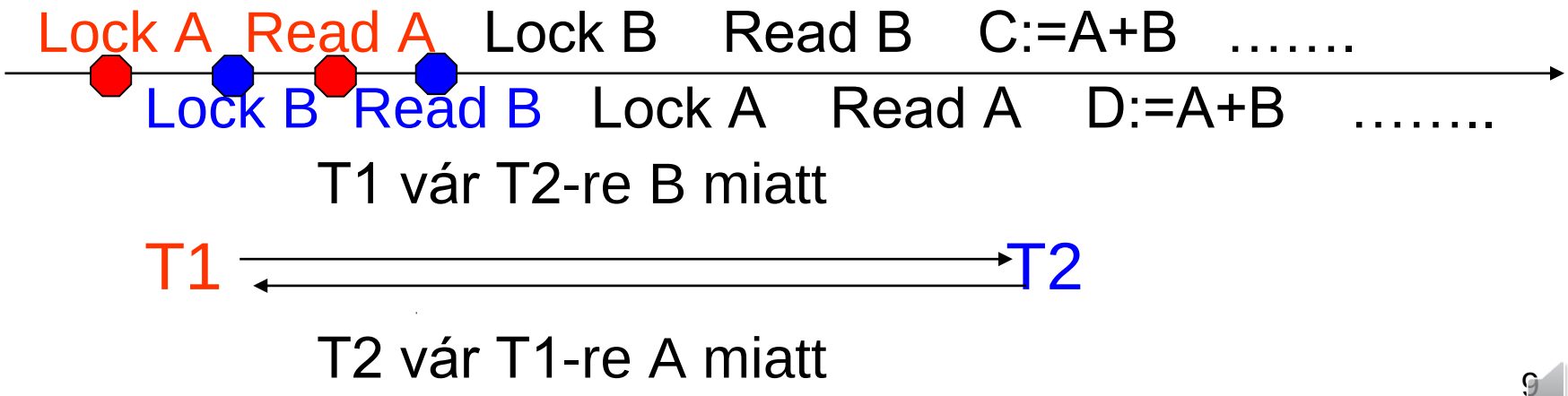
Tranzakció-kezelés

- **Zárolások (Lock, Unlock)**

T1: (Lock S, Read S, S:=S+1, Write S, Unlock S)

T2: (Lock S, Read S, S:=S-1, Write S, Unlock S)

- A zár kiadásához meg kell várni a zár feloldását.
- Csökken a párhuzamosíthatóság
- Zárak finomsága (zárolt adategység nagysága, zárolás típusa) növeli a párhuzamosíthatóságot
- **Holtpont probléma:**



Tranzakció-kezelés

- **Kétfázisú protokoll** – a tranzakció elején zárolunk minden szükséges adatelemet, a végén minden zárat feloldunk
- **Tranzakciók érvényesítése**, naplózás, Commit, Rollback, Checkpoint
- **Ütemező** (tranzakciók műveleteinek végrehajtási sorrendjét adja meg)
- **Szérializálhatóság** (az ütemezés ekvivalens a tranzakciók egymás utáni végrehajtásával)
- Tranzakciók állapotát, elvégzett műveleteket rendszertáblák tárolják

Helyreállíthatóság

- Szoftver- vagy hardverhiba esetén az **utolsó konzisztens állapot visszaállítása**
- Rendszeres mentések
  - Statikus adatbázis (módosítás nem gyakori)
  - Dinamikus adatbázis (módosítás gyakori)
- Naplóállományok
- Összefügg a tranzakciókezeléssel

Ügyfél-kiszolgáló felépítés

- **Kiszolgáló:**
  - nagy tárhellyel rendelkező, gyors gép
  - adatbázis-műveletek optimalizált, párhuzamos végrehajtása
- **Ügyfél:**
  - adatbázis-művelet megfogalmazása
  - elküldése
  - az eredményadatok fogadása
  - megjelenítése
- Más felépítések is léteznek (például **köztes réteg** az ügyfél és a kiszolgáló között)

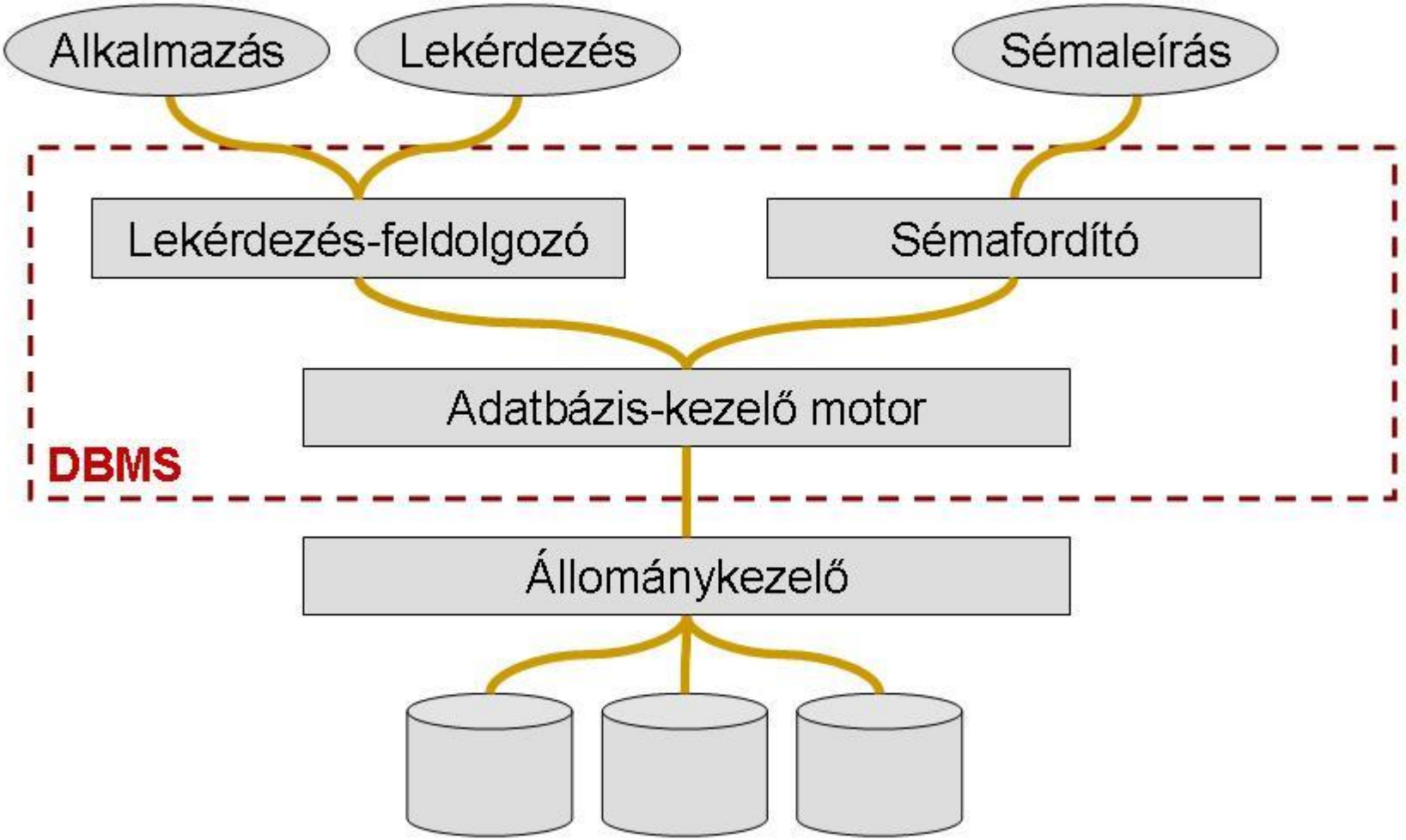
Adatvédelem, adatbiztonság

- **Jogosultságok kezelése**, felhasználók, jelszavak, hozzáférési jogok
- Adatbázissémák korlátozása (virtuális) **nézettáblák** segítségével
- Tárolt adatok, hálózati adatforgalmak **titkosítása** (nagy prímszámok, RSA, DES)

Adatbázis-kezelők felépítése

- **Lekérdezés-feldolgozó**
  - Lekérdezés szintaktikai ellenőrzése
  - Adatbázis-objektumok létezésének, és a hozzáférési jogoknak az ellenőrzése (metaadatbázis, rendszertáblák)
  - Lekérdezés optimális átfogalmazása
  - Végrehajtási tervek készítése
  - Az adatstruktúrák, méretek statisztikái alapján várhatóan minimális költségű végrehajtási terv kiválasztása
  - Az optimális végrehajtási terv lefuttatása
- **Tranzakció-kezelő:**
  - Tranzakciók párhuzamos végrehajtásának biztosítása (**a**tomosság, **k**övetkezetesség, **e**lkülönítés, **t**artósság)
- Tárkezelő (**állománykezelő**):
  - fizikai adatstruktúrák, táblák, **indexek**, **pufferek** kezelése

Adatbázisok ANSI/X3/SPARC modellje



Adatbázisok különböző szintjei

- **Sémák** (tervek, leírások) és **előfordulások** (konkrét adatok, megvalósulások)
- **Fizikai**, **logikai**, **alkalmazói réteg**:

	Séma	**Egy előfordulás**
Alkalmazások	Select sum(fiz) as összfiz from Bér;	30
Logikai adatbázis	Bér(név, fiz)	**név** | **fiz** Kiss | 10 Nagy | 20
Fizikai adatbázis	szekvenciális	(Bér,név,fiz,#2,Kiss,10,Nagy,20)

Adatbázisok különböző szintjei

- **Fizikai adatfüggetlenség**
  - Fizikai adatbázis módosítása (indexek készítése, az adatok más adatstruktúrákban tárolása) nem látszik a felette levő szinteken
  - Hatékonyság növelhető jobb tárolási struktúrákkal
- **Logikai adatfüggetlenség**
  - A logikai adatbázis **bővítése** (új táblák, oszlopok hozzáadása) esetén a régi alkalmazások változtatás nélkül ugyanúgy működjenek

A relációs adatmodell
Edgar Frank Codd 12 szabálya

1. **Az egységes megjelenésű információ szabálya**

   Az adatbázisban szereplő összes információt egy, és csak egy megadott formában (adatmodellben) lehet ábrázolni, nevezetesen táblázatok sorainak oszlopértékeiben.

2. **Garantált lokalizálhatóság szabálya**

   Az adatbázisban minden egyes skaláris értékre logikailag úgy kell hivatkozni, hogy megadjuk az azt tartalmazó táblázat és az oszlop nevét, valamint a megfelelő sor elsődleges kulcsának az értékét.

3. **A NULL értékek egységes kezelése**

   Az adatbázis-kezelő rendszernek (DBMS) olyan egységes módszerrel kell támogatnia a hiányzó vagy nem ismert információ kezelését, amely eltér az összes „rendes” érték kezelésétől, továbbá független az adattípustól.

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

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

5. **A teljes körű „adatnyelv” szabálya**

   A rendszernek legalább egy olyan relációs nyelvet kell támogatnia, amelynek

   (a) lineáris a szintaxisa,

   (b) interaktívan és az alkalmazásokhoz készített programokon belül is lehet használni,

   (c) támogatja az adatdefiniáló műveleteket, a visszakereső és adatmódosító (manipulációs) műveleteket, biztonsági és jósági (integritási) korlátokat, valamint a tranzakciókezelési műveleteket (begin, commit, rollback: elkezdés, jóváhagyás és visszagörgetés).

6. **A nézetek frissítésének szabálya**

   A rendszernek képesnek kell lennie az adatok összes nézetének frissítésére.

A relációs adatmodell
Edgar Frank Codd 12 szabálya

7. Magas szintű beszúrás, frissítés és törlés

A rendszernek támogatnia kell az INSERT, UPDATE, és DELETE (új adat, módosítás, törlés) operátorok halmaz szintű, egyidejű működését.

8. Fizikai szintű adatfüggetlenség

A fizikai adatfüggetlenség akkor áll fenn, ha az alkalmazások (programok) és a felhasználók adatelérési módja független az adatok tényleges (fizikai) tárolási és elérési módjától.

9. Logikai szintű adatfüggetlenség

Logikai adatfüggetlenség akkor áll fenn, ha az adatbázis logikai szerkezetének bővítése nem igényli az adatbázist használó alkalmazások (programok) megváltoztatását.

10. Jóság (integritás) függetlenség

Az adatok jóságának (érvényességének) korlátait az adatfeldolgozási programoktól függetlenül kell tudni meghatározni, és azokat katalógusban kell nyilvántartani. Legyen lehetséges a szóban forgó korlátokat megváltoztatni, anélkül hogy a meglévő alkalmazásokon változtatni kelljen.

11. Elosztástól való függetlenség

A meglévő alkalmazások működése zavartalan kell, hogy maradjon

(a) amikor sor kerül az adatbázis-kezelő osztott változatának bevezetésére

(b) amikor a meglévő osztott adatokat a rendszer újra szétosztja.

12. Megkerülhetetlenség szabálya

Ha a rendszernek van egy alacsony szintű (egyszerre egy rekordot érintő) interfésze, akkor ezt az interfészt ne lehessen a rendszer megkerülésére használni, például a relációs biztonsági vagy jósági (integritás védelmi) korlátok megsértésével.

Relációs adatmodell

ABR1 3. fejezet (104.- 110. oldal)
ABR1 4. fejezet (196.- 215. oldal)

- **Relációséma**: R(A1,A2,…,An)
  - R – relációnév
  - Ai – attribútum- vagy tulajdonságnevek, oszlopnevek
  - Dom(Ai) – lehetséges értékek halmaza, típusa
  - Egy sémán belül az attribútumok különbözőek
- **Reláció-előfordulás**: r
  - r - reláció, tábla, sorhalmaz
  - Egy sor egyszer szerepel
  - Sorok sorrendje lényegtelen
  - Oszlopok sorrendje lényegtelen

$$r \subseteq \mathrm{X}_{i=1}^{n} \mathrm{Dom}(\mathrm{Ai})$$

Relációs adatmodell

- **Jelölések**
  - t∈r esetén t sor (angolul: tuple – n-es)
    - t(Ai) vagy t($i) – a t sor i-edik komponense
    - t[Ai1,...,Aik] - a t sor i1,…,ik-adik komponenseiből álló vektor
- Különböző sémák azonos attribútumai esetén
  - R.A – prefixszel különböztetjük meg
- Egy t sor függvénynek is tekinthető

$t:\{A1,...,An\} \rightarrow \bigcup_{i=1}^{n} \mathrm{Dom}(\mathrm{Ai})$ ahol t(Ai)∈Dom(Ai), i=1..n

Példa

Bér

név	**fiz**	**kor**	
Kiss	10	35	t1
Nagy	20	45	t2
Kovács	15	22	t3

t1(név)=„Kiss”

t3($3)=22

t2(név,kor)=(„Nagy”,45)

t1(Bér.fiz)=10

SQL lekérdezések felbontása: Relációs algebra

- **Az SQL nyelvben összetett, több táblás, alkérdéseket is tartalmazó lekérdezéseket lehet megfogalmazni.**
- **Hogyan lehetne egyszerű SQL lekérdezésekből felépíteni az összetett SQL lekérdezéseket?**
- **Miért jó egy ilyen felbontás?**
  - **Áttekinthetőbbé válik az összetett lekérdezés.**
  - **Az egyszerű lekérdezések kiszámítási költségét könnyebb kifejezni, így segít az optimalizálásban.**
- **Melyek legyenek az egyszerű SQL lekérdezések?**
  - **Legyenek közöttük egyszerű kiválasztásra épülő SQL lekérdezések.**
  - **Legyenek közöttük többtáblás lekérdezések.**
  - **Halmazműveleteket lehessen használni.**
  - **Lehessen átnevezni táblákat, oszlopokat.**
  - **Lehessen egy lekérdezés eredményét egy másik lekérdezésben felhasználni (nézettáblák view-k)**

Egyesítés, unió

1. select * from r union select * from s;

- r, s és r ∪ s azonos sémájú
- **r ∪ s** := {t | t∈r vagy t∈s}
- |r ∪ s| <= |r|+|s|, ahol |r| az r reláció sorainak száma
- azonos sor csak egyszer szerepelhet

A	B
0	0
0	1

∪

A	B
0	0
1	0

=

A	B
0	0
0	1
1	0

Kivonás, különbség

2. select * from r minus select * from s;

- r, s és r - s azonos sémájú
- **r - s** := { t | t ∈ r és t ∉ s}
- |r - s| <= |r|

A	B
0	0
0	1

—

A	B
0	0
1	0

=

A	B
0	1

select * from r minus select * from s;

VAGY

select * from r where not exists

(select * from s where r.A=s.A and r.B=s.B);

Szorzás, direktszorzat vagy Descartes-szorzat

3. select * from r,s;

- r, s sémáiban nincs közös attribútum
- r × s sémája a sémák egyesítése
- **r × s** := { t | t[R] ∈ r és t[S] ∈ s }
- |r × s| = |r|*|s|

A	B
0	0
0	1

×

C	D
0	0
1	0

=

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

Vetítés, projekció

4. select distinct A1,...,Ak from r;

- **X ⊆ {A1,…,An}**
- $\Pi_X(r)$ sémája X
- $\Pi_X(r)$:= { t | van olyan t'∈r, melyre t'[X] = t }
- $|\Pi_X(r)| <= |r|$

r:

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

select distinct B,D from r;

$\Pi_{BD}(r) =$

B	D
0	0
1	0

select distinct D,A from r;

$\Pi_{DA}(r) =$

D	A
0	0

Kiválasztások

5. select * from r where A=B;
   select * from r where A<B;
   select * from r where A>B;
   select * from r where A<>B;
   select * from r where A<=B;
   select * from r where A>=B;

   select * from r where A=konstans;
   select * from r where A<konstans;
   select * from r where A>konstans;
   select * from r where A<>konstans;
   select * from r where A<=konstans;
   select * from r where A>=konstans;

   select * from r where feltétel1 and feltétel2;
   select * from r where feltétel1 or feltétel2;
   select * from r where not (feltétel);

Kiválasztás, szűrés, szelekció

- $\sigma_F(r)$ és r sémája megegyezik
- **$\sigma_F(r)$** := { t | t∈r és F(t) = IGAZ }
- **F feltétel:**
  - atomi, elemi feltétel
    - **Ai Θ Aj**, ahol **Θ ∈ { =, ≠, <,>, <=, >= }**
    - **Ai Θ c**, **c Θ Ai** ahol c egy konstans
  - feltételekből ∧, ∨, ¬ logikai összekapcsolókkal, és zárójelekkel kapható kifejezés

r:

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0

$\sigma_{A=C \wedge \neg (B<1)}(r) =$

A	B	C	D
0	1	0	0

select * from r where A=B and not (B<1);

Kiválasztás, szűrés, szelekció

- $|\sigma_F(r)| <= |r|$
- a feltételben függvények nem használhatók:

  $\sigma_{A + B < 5}(r)$ nem megengedett

- **az összetett feltételek átírhatók elemi feltételeket használó kifejezésekké a következő szabályok segítségével**:
  - $\sigma_{F1 \wedge F2}(r) \cong \sigma_{F1}(\sigma_{F2}(r)) \cong \sigma_{F2}(\sigma_{F1}(r))$
  - $\sigma_{F1 \vee F2}(r) \cong \sigma_{F1}(r) \cup \sigma_{F2}(r)$
  - A De Morgan azonosság segítségével a negáció beljebb vihető:
    - $\neg$ **(F1** $\wedge$ **F2)** helyett **(**$\neg$ **F1)** $\vee$ **(**$\neg$ **F2)**
    - $\neg$ **(F1** $\vee$ **F2)** helyett **(**$\neg$ **F1)** $\wedge$ **(**$\neg$ **F2)**
    - elemi feltétel tagadása helyett a fordított összehasonlítást használjuk:

      például $\neg$ **(A < B)** helyett **(A >= B)**

Kiválasztás, szűrés, szelekció

$$\sigma_{(\neg(A = C \wedge \neg (B < 1))) \wedge (D < 2)}(r) =$$

$$\sigma_{(\neg(A = C) \vee \neg \neg(B < 1)) \wedge (D < 2)}(r) =$$

$$\sigma_{A \neq C}(\sigma_{D < 2}(r)) \cup \sigma_{B < 1}(\sigma_{D < 2}(r))$$

- az elemi feltételekhez lekérdezést gyorsító adatszerkezetek, indexek készíthetők

Átnevezés

6. select oszlop [AS] újnév,... from r [AS] újnév;

- A relációnak és az attribútumoknak új nevet adhatunk.
- Ha r sémája R(A1,…,An), akkor $\rho_{S(B1,\ldots,Bn)}(r)$ sémája **S(B1,…,Bn).**
- $|\rho_{S(B1,\ldots,Bn)}(r)| = |r|$

$\rho_{MUNKA(dolg,jöv)}(r) =$

BÉR

r:

név	fiz
Kiss	10
Nagy	20

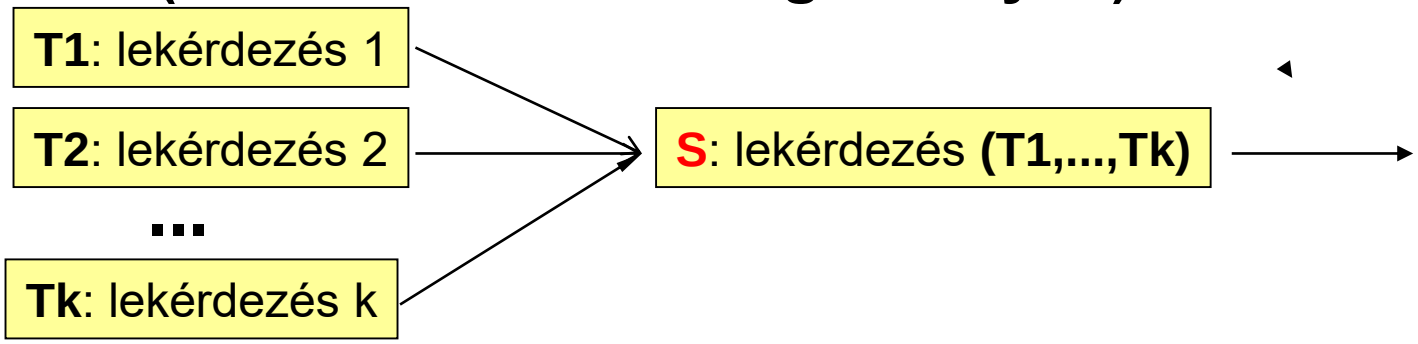
MUNKA

dolg	jöv
Kiss	10
Nagy	20

select név dolg, fiz jöv from BÉR MUNKA;

Kifejezések kompozíciója

- **Az egyszerű SQL lekérdezésekből hogy lehet felépíteni összetett lekérdezéseket?**
- **Az SQL lekérdezés eredménye SQL tábla.**
- **Készítsünk nézettáblát (VIEW) a részlekérdezéshez.**
- **Az SQL lekérdezés FROM listájában nézettáblák is használhatók. (A nézettábla nem foglal helyet.)**




```
create view T1 as select ... from ... where ... ;
create view T2 as select ... from ... where ... ;
...
create view Tk as select ... from ... where ... ;
create view S as select ... from T1,...,Tk where ... ;
```

Relációs algebra

- **ÖSSZEFOGLALVA:**
- **Alapoperátorok:**
  1. Egyesítés
  2. Különbség
  3. Szorzat
  4. Vetítés
  5. Kiválasztás
  6. Átnevezés

1. select * from r union select * from s;
2. select * from r minus select * from s;
3. select * from r,s;
4. select distinct A1,...,Ak from r;
5. select * from r where feltétel;
6. select oszlop [AS] újnév,... from r [AS] újnév;

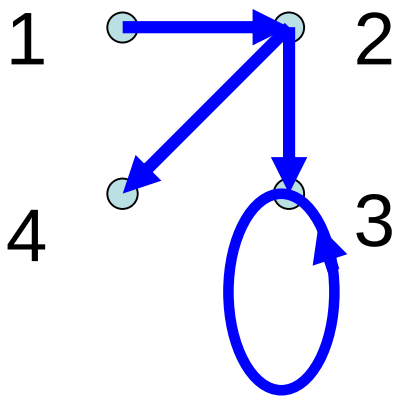
- **Kifejezés**:
  - konstans reláció
  - relációs változó
  - alapoperátorok véges sok alkalmazása kifejezésekre
  - ezek és csak ezek

```
create view T1 as select ... from ... where ... ;
....
create view Tk as select ... from ... where ... ;
create view S as select ... from T1,...,Tk where ... ;
```

- **Relációs algebra = kifejezések halmaza**

A relációs algebra kifejezőereje

- **Relációs algebrában a legfontosabb lekérdezéseket ki tudjuk fejezni, de nem mindent!**
- ÉL(honnan, hova)
- ÚT(honnan, hova) – tranzitív lezárás



ÉL

honnan	hova
1	2
2	4
2	3
3	3

ÚT

honnan	hova
1	2
2	4
2	3
3	3
1	3
1	4

- nem triviális rekurzió
- **TÉTEL**: **Nem létezik olyan relációs algebrai kifejezés, amelyet tetszőleges ÉL táblára alkalmazva a neki megfelelő ÚT táblát eredményezi.**

Származtatott műveletek

- A gyakran használt kifejezések helyett új műveleteket vezetünk be.
- Nem alapműveletek, hanem származtatottak
- **Metszet**
  - **$r \cap s = \{ t \mid t \in r$ és $t \in s \}$**

```
select * from r intersect select * from s;
```

  - többféleképpen kifejezhető relációs algebrában:
    - **$r \cap s = r - (r - s) = s - (s - r) = r \cup s - ((r - s) \cup (s - r))$**
- **Összekapcsolások (JOIN)**
  - Téta-összekapcsolás (**Θ-join**)
  - Egyen-összekapcsolás (**equi-join**)
  - Természetes összekapcsolás (**natural join**)
  - Félig-összekapcsolás (**semi-join**)
  - Külső összekapcsolás (**outer join**)
- A szorzáshoz hasonlóan költséges műveletek, nagy méretű táblákat eredményezhetnek, kivételt képez a félig-összekapcsolás.

Téta-összekapcsolás

select * from r,s where r.Ai összehasonlítás s.Bj;

- r, s sémáiban (R(A1,…,An), S(B1,…,Bn) nincs közös attribútum
- $r \underset{AiΘBj}{⋈} s = \sigma_{Ai\ \Theta\ Bj}(r \times s)$

select * from r,s where r.B=s.C;

A	B
0	0
0	1

$\underset{B=C}{⋈}$

C	D
0	0
0	1

=

A	B	C	D
0	0	0	0
0	0	0	1

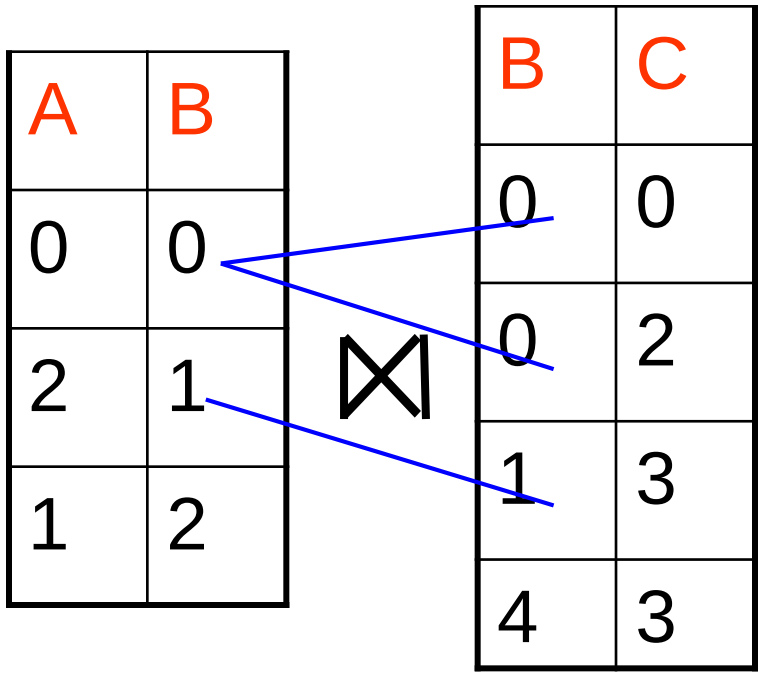
- **Ai=Bj** feltétel esetén **egyen-összekapcsolás**nak hívjuk.

Természetes összekapcsolás

select distinct R.A1,...,R.An,R.B1,...,R.Bk,S.C1,...,S.Cm from r,s where R.B1=S.B1 and R.B2=S.B2 and ... and R.Bk=S.Bk;

- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- r ⋈ s =

$$\rho_{P(A1,\ldots,An,B1,\ldots,Bk,C1,\ldots,Cm)}\Pi_{A1,\ldots,An,R.B1,\ldots,R.Bk,C1,\ldots,Cm}\sigma_{R.B1=S.B1\wedge\ldots\wedge R.Bk=S.Bk}(r\times s)$$



A	B
0	0
2	1
1	2

⋈

B	C
0	0
0	2
1	3
4	3

=

A	B	C
0	0	0
0	0	2
2	1	3

select distinct A,R.B,C from r,s where R.B=S.B;

Félig-összekapcsolás

select distinct R.A1,...,R.An,R.B1,...,R.Bk from r,s

where R.B1=S.B1 and R.B2=S.B2 and ... and R.Bk=S.Bk;

- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- $r \ltimes s = \rho_{P(A1,\ldots,An,B1,\ldots,Bk,)} \Pi_{A1,\ldots,An,R.B1,\ldots,R.Bk} (r|\times|s)$
- Az első relációban mely sorokhoz létezik kapcsolható sor a második táblából

A	B
0	0
2	1
1	2

$\ltimes$

B	C
0	0
0	2
1	3
4	3

=

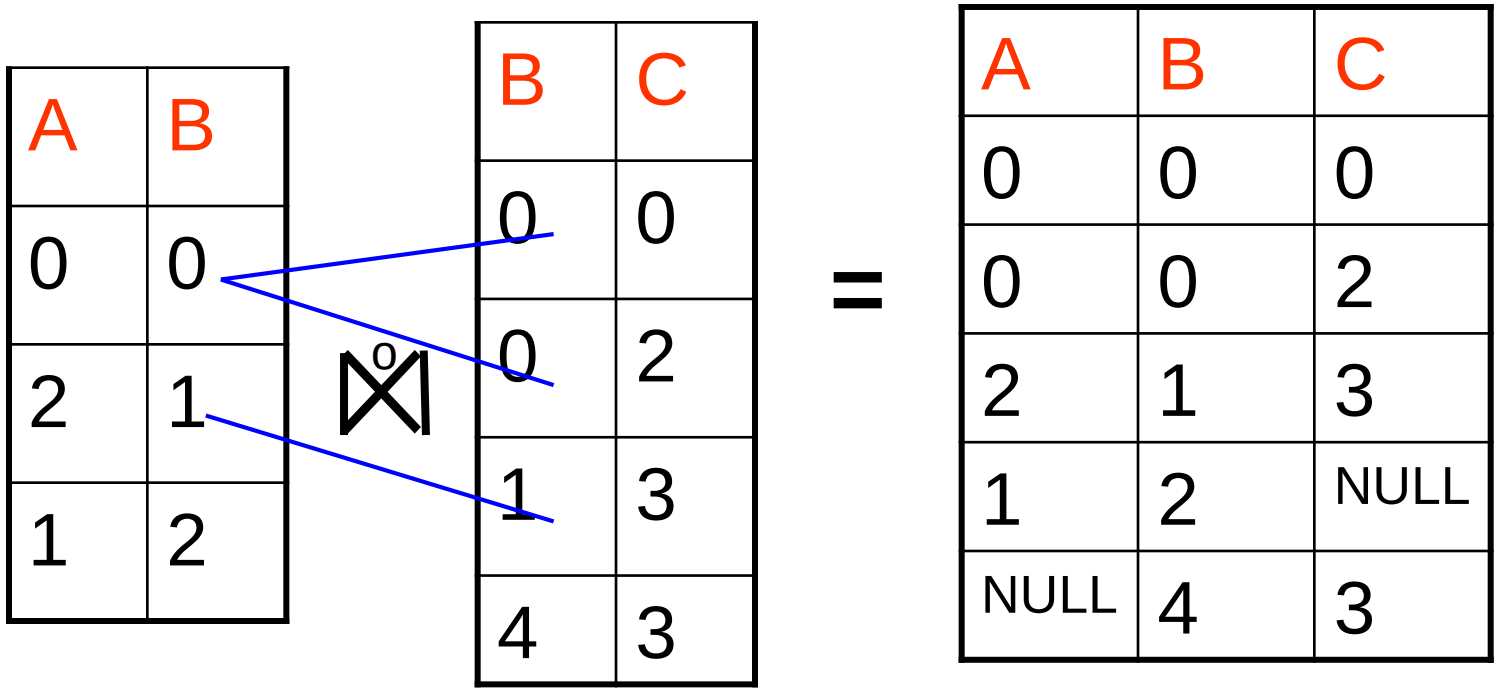
A	B
0	0
2	1

select distinct A,R.B
from r,s where R.B=S.B;

Külső összekapcsolás

select A,r.B,C from r outer join s on r.B=s.B;

- Nem relációs algebrai művelet, mert kilép a modellből
- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- r $\overset{o}{\bowtie}$ s = r $\bowtie$ s relációt kiegészítjük az r és s soraival, a hiányzó helyekre NULL értéket írva



A	B
0	0
2	1
1	2

$\overset{o}{\bowtie}$

B	C
0	0
0	2
1	3
4	3

=

A	B	C
0	0	0
0	0	2
2	1	3
1	2	NULL
NULL	4	3

Összekapcsolások

- Ha r, s **sémái megegyeznek**, akkor **r|×|s = r ∩ s**.
- Ha r, s sémáiban **nincs közös attribútum**, akkor **r|×|s = r×s**.
- Ha r = ∅, akkor r×s = ∅ és r|×|s = ∅.
- A külső összekapcsolás lehet bal oldali, ha csak r sorait vesszük hozzá a természetes összekapcsolás-hoz: $r|^{o}\times|_{B}s$. Hasonlóan értelmezhetjük a jobb oldali összekapcsolást is $r|^{o}\times|_{J}s$.

select A,r.B,C from r left outer join s on r.B=s.B;
vagy select A,r.B,C from r,s where r.B = s.B(+);

select A,r.B,C from r right outer join s on r.B=s.B;
vagy select A,r.B,C from r,s where r.B(+) = s.B;

Osztás, hányados

- Maradékos osztás: $7 \div 3 = 2$, mert 2 a legnagyobb egész, amelyre még $2 * 3 \leq 7$.
- Relációk szorzata esetén $\leq$ helyett tartalmazás.
- r és s sémája **R(A1,…,An,B1,…,Bm)**, illetve **S**(B1,…,Bm), **r ÷ s** sémája R(**A1,…,An**)
- **r ÷ s** a legnagyobb (**legtöbb sort tartalmazó**) reláció, amelyre **(r ÷ s) × s ⊆ r**.
- Kifejezhető relációs algebrában:
- $\Pi_{A1,\ldots,An}(r) - \Pi_{A1,\ldots,An}(\Pi_{A1,\ldots,An}(r) \times s - r)$
- Lehetséges értékekből kivonjuk a rossz értékeket.
- $(p \times r) \div r = p$

Osztás, hányados

- Ki szereti legalább azokat, mint Micimackó?

KI	MIT
Füles	málna
Füles	méz
Füles	alma
Micimackó	málna
Micimackó	méz
Kanga	málna
Kanga	körte
Nyuszi	lekvár

÷

MIT
málna
méz

=

KI
Füles
Micimackó

$$\text{szeret} \div \prod_{\text{MIT}}(\sigma_{\text{KI='Micimackó'}}(\text{szeret}))$$

r(a,b)÷s(b) hányados kifejezése SQL-ben (MINUS segítségével):

- $r(a,b) \div s(b) = \Pi_a(r) - \Pi_a(\Pi_a(r) \times s - r)$

- $\Pi_a(r) \times s = \Pi_{r.a,s.b}(r \times s)$
- select distinct r.a,s.b from r,s;

- $\Pi_a(r) \times s - r$
- create view rsz as
  select distinct r.a,s.b from r, s
  minus
  select * from r;

- $\Pi_a(\Pi_a(r) \times s - r)$
- select distinct a from rsz;

- $\Pi_a(r) - \Pi_a(\Pi_a(r) \times s - r)$
- select distinct a from r
  minus
  select distinct a from rsz;

- $r(a,b) \div s(b)$:
- $\Pi_a(r) - \Pi_a(\Pi_a(r) \times s - r)$
- create view rsz as
  select distinct r.a,s.b from r, s
  minus
  select * from r;
- select distinct a from r
  minus
  select distinct a from rsz;

r(a,b)÷s(b) hányados kifejezése SQL-ben (NOT EXISTS segítségével):

- $r(a,b) \div s(b) = \Pi_a(r) - \Pi_a(\Pi_a(r) \times s - r)$
- $\Pi_a(r) \times s = \Pi_{r.a,s.b}(r \times s)$
- select distinct r.a,s.b from r,s;
- $\Pi_a(r) \times s - r$
- select distinct r.a,s.b from r r1, s s1
  where not exists
  (select * from r r2
  where r2.a=r1.a and s1.b=r2.b);
- $\Pi_a(\Pi_a(r) \times s - r)$
- select distinct r.a from r r1, s s1
  where not exists
  (select * from r r2
  where r2.a=r1.a and s1.b=r2.b);

- $\Pi_a(r) - \Pi_a(\Pi_a(r) \times s - r)$
- select distinct r2.a from r r2
  where not exists
  (select * from r r1, s s1
  where r2.a=r1.a and
  not exists
  (select * from r r3
  where r3.a=r1.a
  and s1.b=r3.b));

Monotonitás

- **Monoton nem csökkenő** (röviden **monoton**) kifejezés: bővebb relációra alkalmazva az eredmény is bővebb:

  Ha **Ri ⊆ Si**, i=1,…,n, akkor **E(R1,…,Rn)⊆E(S1,…,Sn).**

- A kivonás kivétel az alapműveletek monoton műveletek (monoton relációs algebra).

A	B
0	1
0	0

-

A	B
0	1

$\not\subseteq$

A	B
0	1
0	0

-

A	B
0	1
0	0

Monotonitás

- **DE**: Monoton kifejezésben is szerepelhet kivonás: $\mathbf{r} \cap \mathbf{s} = r - (r - s)$ monoton.
- Ha E, E1, Ek monoton kifejezések, és E(E1(…),…,Ek(…)) helyes kifejezés, akkor monoton is.
- **Következmény**: **A kivonás nem fejezhető ki a többi alapművelettel**.

Ismétlés vége