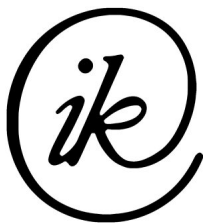


ADATBÁZISOK I.

Szerkesztette: BÓKAY CSONGOR



Az esetleges hibákat kérlek a csongor@csongorbokay.com címen jelezd!
Utolsó módosítás: 2013. március 20.



Ez a Mű a Creative Commons Nevezd meg! - Ne add el! - Így add tovább! 3.0 Unported
Licenc feltételeinek megfelelően szabadon felhasználható.

1. RELÁCIÓS ALGEBRA

Az algebra műveleteket és atomi operandusokat tartalmaz. A relációs algebrában az atomi operandusokon (relációkhoz tartozó változók; konstansok, melyek véges relációkat fejeznek ki) és az algebrai kifejezéseken végzett műveletek alkalmazásával kapott relációkon műveleteket adunk meg, kifejezéseket építünk. A relációs algebrában minden művelet végeredménye egy reláció, amelyen további műveletek adhatóak meg.

1.1. Alapfogalmak

1.1.1. Reláció. A $D_1 \times D_2 \times \dots \times D_n$ Descartes-szorzat bármely R részhalmazát ($R \subseteq D_1 \times D_2 \times \dots \times D_n$) *relációnak* nevezzük. A D_1, D_2, \dots, D_n adott halmazok a reláció értéktartományai.

1.1.2. Attribútum. Az értéktartományok elemeit felvevő jellemzőket *attribútumoknak*, vagy *mezőknek* nevezzük. Az attribútumok a relációk fejrészában találhatók, tehát ezek a relációban szereplő oszlopok nevei, melyek megadják az oszlopban lévő adatok jelentését. A reláción belül ezeknek a neveknek egyedieknek kell lenniük.

1.1.3. Sor, sortípus. A reláció egy eleme a táblázat *sora*. A táblázatnak nem lehetnek azonos sorai, illetve a sorok sorrendje tetszőleges. A sorokat rekordnak is szokás nevezni.

A *sortípus* megadható az $\langle attrnév_1 : értéktípus_1, \dots, attrnév_n : értéktípus_n \rangle$, vagy röviden az $\langle A_1, \dots, A_n \rangle$ formában.

1.1.4. Komponens. A sor egy elemét *komponensnek* vagy *cellának* nevezzük.

1.1.5. Relációséma. A reláció neve és a reláció attribútumainak a halmaza együttesen alkotja a reláció sémáját. A reláció sémáját a reláció nevével és a sortípussal adjuk meg, például: $R(A_1 : értéktípus_1, A_2 : értéktípus_2, \dots, A_n : értéktípus_n)$ vagy rövidebben $R(A_1, A_2, \dots, A_n)$.

1.1.6. Kulcsok. Attribútumok egy halmaza *kulcsot* alkot egy relációra nézve, ha a reláció bármely előfordulásában nincs két olyan sor, amelyek a kulcs összes attribútumának értékein megegyeznének. A reláció sémájában a kulcsot aláhúzással jelöljük, például: $R(\underline{A_1}, \underline{A_2}, A_3, \dots)$.

Egy relációban több kulcs is lehet, de kijelölt kulcs csak egy, ez az *elsődleges kulcs* (primary key). Ha egy kulcs egy másik reláció elsődleges kulcsára utal, akkor *idegen kulcsnak* (foreign key) nevezzük.

1.2. Alapműveletek

A relációs algebrának hat alapművelete van. Ez egy minimális készlet, tehát bármelyiket elhagyva az a többivel nem fejezhető ki.

1.2.1. Halmazműveletek. Mivel a relációk esetén sorok halmazáról van szó, így értelmezhetők a szokásos halmazműveletek: az *unió*, a *metszet* és a *különbség*. Az alapműveletkehez az unió és a különbség tartozik, a metszet műveletet származtatjuk.

Legyen R és S két azonos típusú reláció. Ekkor a halmazműveleteket a következőképp definiáljuk: $R \cup S := \{t \mid t \in R \vee t \in S\}$, $R - S := \{t \mid t \in R \wedge t \notin S\}$ és $R \cap S := R - (R - S)$.

R			S			$R \cup S$			$R - S$		
A	B	C	A	B	C	A	B	C	A	B	C
a	b	c	a	b	c	a	b	c	g	a	d
c	d	e	c	d	e	c	d	e			
g	a	d	g	d	f	g	d	f			
						g	a	d			

1.2.2. Projekció. Egy adott relációt vetít le az alsó indexben szereplő attribútumokra, illetve kiszűri a keletkező duplikátumokat, hogy továbbra is fennáljon a halmaz tulajdonság. Jelölés:

$$\pi_{A_i, \dots, A_j}(R)$$

ahol A_1, \dots, A_j az R reláció sémájában lévő attribútumok egy részhalmazának felsorolása.

R			$\pi_{A,B}(R)$	
A	B	C	A	B
a	b	c	a	b
c	d	e	c	d
c	d	d		

1.2.3. Szelekció. Kiválasztja az argumentumban szereplő reláció azon sorait, melyek eleget tesznek az alsó indexben szereplő feltételnek. Jelölés:

$$\sigma_L(R)$$

ahol L vagy elemi feltétel ($A_i \circ A_j$ vagy $A_i \circ c$, ahol c konstans és $\circ \in \{=, \neq, <, >, \leq, \geq\}$), vagy összetett feltétel (ha B_1, B_2 feltételek, akkor $\neg B_1, B_1 \wedge B_2, B_1 \vee B_2$, és zárójelezésekkel is feltételek).

R			$\sigma_{A=a \vee C=d}(R)$		
A	B	C	A	B	C
a	b	c	a	b	c
c	d	e			
g	a	d	g	a	d

1.2.4. Természetes összekapcsolás. Két reláció *természetes összekapcsolása* azon sorpárokat tartalmazza, amelyek a relációk azonos attribútumain megegyeznek. Jelölés:

$$R \bowtie S$$

Vegyünk két relációt, amelyek sémái $R(A_1, \dots, A_n, B_1, \dots, B_k)$, illetve $S(B_1, \dots, B_k, C_1, \dots, C_m)$. Ekkor $R \bowtie S$ típusa $(A_1, \dots, A_n, B_1, \dots, B_k, C_1, \dots, C_m)$, vagyis a két attribútumhalmaz uniója. Ha R és S attribútumhalmazai diszjunktak, akkor $R \bowtie S = R \times S$, ha pedig egyenlőek, akkor $R \bowtie S = R \cap S$. Fontos megjegyezni, hogy a relációs algebraiban nem változik az oszlopok sorrendje.

R		S		$R \bowtie S$		
A	B	B	C	A	B	C
a	a	a	a	a	a	a
c	b	a	c	a	a	c
b	c	b	d	c	b	d
		e	d			

1.2.5. Átnevezés. Szükség lehet egy adott relációnak vagy a reláció attribútumainak *átnevezésére*. Ha ugyan azt a táblát szeretnék használni többször, akkor az attribútumok átnevezése nem szükséges. Jelölés:

$$\rho_{R(A_1, \dots, A_n)}(S(B_1, \dots, B_n))$$

Egy adott S relációt a $\rho_R(S)$ kifejezéssel nevezhetünk át R -re.

1.3. További műveletek

1.3.1. Descartes-szorzat. Két vagy több reláció *Descartes-szorzata* során minden reláció sora párban összefűződik. Két reláció esetén az első tábla minden sorához hozzáfűzzük a rákövetkező tábla minden sorát. Jelölés:

$$R \times S \times \dots$$

A relációk azonos nevű attribútumait meg kell különböztetni egymástól, így vagy átnevezést, vagy az $R.A_1, \dots, R.A_n, S.A_1, \dots, S.A_n, \dots$ hivatkozási sémát kell alkalmazni.

R			S		$R \times S$				
A	B	C	B	D	A	R.B	C	S.B	D
a	b	c	b	r	a	b	c	b	r
					a	b	c	q	s
c	d	e	q	s	c	d	e	b	r
					c	d	e	q	s

1.3.2. Osztás. Két reláció hányadosa megadja az osztás bal oldalán álló reláció minden olyan sorát, amelyek mellett az osztás jobb oldalán álló reláció mindegyik sora előfordul.

$R(A_1, \dots, A_n, B_1, \dots, B_m)$ és $S(B_1, \dots, B_m)$ relációk esetén R és S hányadosa megadja azon A_1, \dots, A_n attribútumú v sorok halmazát, amelyekre igaz, hogy az S reláció minden w sorára a vw sor benne van az R relációban. Jelölés:

$$R \div S$$

Az osztás művelet kifejezhető más relációs algebrai alpműveletekkel:

$$R \div S = \pi_L(R) - \pi_L\left((\pi_L(R) \times S) - R\right)$$

ahol L egy olyan attribútumlista, amelynek elemei az R reláció azon attribútumai, amelyek nincsenek benne az S relációban.

R			S		$R \div S$
A	B	C	B	C	A
a	b	c	b	c	a
a	d	e	d	e	f
c	d	e			
f	b	c			
f	d	e			
c	f	e			

1.3.3. Théta-összekapcsolás. A természetes összekapcsolás előírja, hogy egyetlen speciális feltétel szerint párosítsuk a sorokat, mégpedig a közös attribútumok azonos értéke alapján. Néha szükség lehet két reláció sorainak más szempontból történő párosítására, ezért vezetjük be a természetes összekapcsolást kiterjesztve a théta-összekapcsolás műveletet. Jelölés:

$$R \bowtie_L S$$

ahol L egyszerű aritmetikai összehasonlítás, amely $L = A_i \circ B_j$ alakú, és $\circ \in \{=, \neq, <, >, \leq, \geq\}$. Ezt az összehasonlítást $R \times S$ soraira alkalmazzuk, így $R \bowtie_L S = \sigma_L(R \times S)$.

R			S		$R \bowtie_{A=D} S$				
A	B	C	B	D	A	R.B	C	S.B	D
a	b	c	b	c	c	d	e	b	c
c	d	e	q	c	c	d	e	q	c

2. SQL

2.1. Áttérés relációs algebráról

Relációs algebra	SQL (Oracle)
R	SELECT * FROM R;
$\pi_t(R)$	SELECT t FROM R;
$\sigma_{felt}(R)$	SELECT * FROM R WHERE felt;
$\pi_t(\sigma_{felt}(R))$	SELECT t FROM R WHERE felt;
$R \cup S$	SELECT * FROM R UNION SELECT * FROM S;
$R \cap S$	SELECT * FROM R INTERSECT SELECT * FROM S;
$R - S$	SELECT * FROM R MINUS SELECT * FROM S;
$R \times S$	SELECT * FROM R, S; (vagy CROSS JOIN)
$R \bowtie S$	SELECT * FROM R NATURAL JOIN S;
$R \bowtie_{felt} S$	SELECT * FROM R JOIN S ON felt;
$R \div S$	SELECT L FROM R MINUS SELECT L FROM (SELECT * FROM (SELECT L FROM R) CROSS JOIN (SELECT * FROM S) MINUS SELECT * FROM R); -- ahol L az R azon oszlopait jelöli, -- amelyek nincsenek benne az S-ben

2.2. SELECT

2.2.1. A dual tábla. A dual pszeudotáblát használva tudunk olyan lekérdezéseket végezni, amelyekhez nem szükséges tábla. Példák:

```
-- Az attr.név 'HELLO' lesz
SELECT 'hello' FROM dual;

-- Az attr.név 'EREDMÉNY' lesz (átnevezés), a mező értéke pedig 2
SELECT 1+1 eredmény FROM dual;

SELECT 'hello' köszön, n név, sz.* FROM sz;
SELECT 'hello' köszönés, 2*1 "szorzás_eredménye" FROM dual;
```

2.2.2. ROWNUM. Néha szükség lehet egy lekérdezés sorainak explicit megszámozására. Ezt a **ROWNUM** pszeudomező segítségével tehetjük meg. Példa:

```
SELECT ROWNUM, n FROM sz;
```

2.2.3. DISTINCT. A relációs algebrával ellentétben az SQL-ben a táblák nem rendelkeznek halmaz tulajdonsággal, így egy sor többször is előfordulhat. A **DISTINCT** kulcsszó segítségével a duplikátumokat kiszűrhetjük. Példa:

```
SELECT DISTINCT gy FROM sz; -- Az összes gyümölcs
```

2.2.4. Műveletek, IN, NOT. A lekérdezésekben használhatóak a megszokott műveletek (=, !=, <, <=, >, >=), továbbá a rendezett n-esek esetén használható az **IN**, illetve ennek tagadása, a **NOT IN** kulcsszó. A **NOT** kulcsszó bármely logikai kifejezés tagadására használható. Példák:

```
SELECT 'igaz' eredmény FROM dual WHERE 3>2;
SELECT 'igaz' eredmény FROM dual WHERE NOT 1>2;
SELECT 'igaz' eredmény FROM dual WHERE 1 IN (1, 2, 3);

SELECT 'igaz' eredmény FROM dual
  WHERE 'Fülés' IN ('Malacka', 'Füles', 'Kanga');

SELECT 'igaz' eredmény FROM dual
  WHERE (1, 3) IN ((1, 2), (1, 4), (1, 3));

SELECT 'igaz' eredmény FROM dual
  WHERE 'Füles' NOT IN (SELECT n FROM sz WHERE gy='dió');

SELECT 'igaz' eredmény FROM dual
  WHERE ('Micimackó', 'dió') IN (SELECT * FROM sz);
```

2.2.5. LIKE. Szöveg típusú mezők esetén a **LIKE** kulcsszóval mintaillesztés szerűen szűrhetünk. A '_' karakter pontosan egy, míg a '%' bármennyi karakterre illeszkedik. Példák:

```
SELECT * FROM sz WHERE n LIKE 'F%';
SELECT * FROM sz WHERE n LIKE '%F%';
SELECT * FROM sz WHERE n LIKE '%c%a';
SELECT * FROM sz WHERE n LIKE '_i%s';
SELECT * FROM sz WHERE n LIKE '_____';
```

2.2.6. BETWEEN. A **BETWEEN** kulcsszóval eldönthető, hogy egy mező benne van-e az adott intervallumban. Példák:

```
-- Azon dolgozók, akinek a fizetése 1000 és 2000 között van
SELECT * FROM emp WHERE sal BETWEEN 1000 AND 2000;
```

```
-- Aki nevének a kezdőbetűje 'J' és 'S' között van
SELECT * FROM emp WHERE ename BETWEEN 'J' AND 'S';
```

2.2.7. EXISTS. Annak eldöntésére, hogy egy allekérdezés adott-e vissza legalább egy sort az **EXISTS** kulcsszót használjuk, amely hamis értéket ad vissza, ha nem tért vissza sorral az allekérdezés, különben igazat. Példa:

```
SELECT 'igaz' FROM dual
WHERE EXISTS (SELECT * FROM sz WHERE gy='alma');
```

2.2.8. ALL, ANY. Az **ALL** és az **ANY** kulcsszóval összehasonlíthatunk egy mezőt egy allekérdezés vagy rendezett n-es minden elemével. Az **ANY** akkor ad igazat, ha legalább egy elemre, míg az **ALL** akkor ad igazat, ha minden elemre igaz az összehasonlítás. Példák:

```
SELECT * FROM emp
WHERE sal > ALL (SELECT sal FROM emp WHERE job='MANAGER');
```

```
SELECT * FROM emp
WHERE sal > ANY (SELECT sal FROM emp WHERE job='MANAGER');
```

2.2.9. A NULL érték. A relációs algebrával ellentétben SQL-ben egy mező felvehet **NULL** értéket is (kivéve, ha explicit megtiltjuk). Fontos megjegyezni, hogy a **NULL** nem egyenlő az üres szöveggel vagy a 0-val, illetve **NULL** = **NULL** és **NULL** != **NULL** is hamisat ad vissza, így csak az **IS NULL** kulcsszóval tudjuk eldönteni, hogy egy mező értéke **NULL**-e. Példák:

```
SELECT 'igaz' FROM dual WHERE NULL != NULL; -- 0
SELECT 'igaz' FROM dual WHERE NULL = NULL;  -- 0
```

```
SELECT * FROM emp WHERE comm IS NULL;
SELECT * FROM emp WHERE comm IS NOT NULL;
```

2.2.10. COALESCE. Mivel bármely **NULL**-al végzett művelet **NULL**-t eredményez, ezért szükség lehet a **NULL** értékeket „átalakítani” egy másik értékke. Erre szolgál a **COALESCE** függvény, amely visszaadja az első nem **NULL** értéket a paramétereinek közül. Példa:

```
-- A sal+comm NULL, ha a comm értéke NULL, ám a
-- COALESCE segítségével ezt a NULL értéket 0-nak vesszük
SELECT ename, sal, comm, sal + COALESCE(comm, 0) FROM emp;
```

2.2.11. Aggregátor függvények. Az SQL tartalmaz olyan csoportfüggvényeket, amelyek a lekérdezés minden során végrehajtódnak, majd egy értékkel térnek vissza. Ilyen a **SUM**, **COUNT**, **AVG**, **MAX** és a **MIN**. Ezen függvények a **COUNT**-ot kivéve (0-t ad vissza) nem veszik figyelembe a **NULL**-t. Példa:

```
SELECT SUM(comm), AVG(comm), SUM(comm)/COUNT(comm),
COUNT(comm), SUM(comm)/COUNT(*), COUNT(*) FROM emp;
```


2.2.12. GROUP BY. Adatok csoportosítására a **GROUP BY** kulcsszót használjuk. A **WHERE** kulcsszóval a csoportosítás előtt, míg a **HAVING**-el a létrejövő csoportokat lehet szűrni.

```
-- A NULL egy értéknek számít
SELECT job FROM emp GROUP BY job;

SELECT job, SUM(sal), COUNT(sal), COUNT(*) FROM emp
  WHERE sal > 1000
  GROUP BY job
  HAVING SUM(sal) > 5000;
```

2.2.13. ORDER BY. Az adatok rendezéséhez az **ORDER BY** kulcsszót használjuk. Rendezhetünk csökkenő (**DESC**), illetve növekvő (**ASC**, alapértelmezett) sorrendben is.

```
SELECT job, deptno FROM emp
  GROUP BY job, deptno
  ORDER BY job, deptno DESC;
```

2.2.14. Théta-összekapcsolás, külső összekapcsolás. Théta-összekapcsolás SQL-ben a következő kifejezéssel írható le: <tábla 1> **JOIN** <tábla 2> **ON** <feltétel>. Példa:

```
-- A látogat táblában ivó oszlopként szerepelnek a sörivók nevei
SELECT * FROM ivók JOIN látogat ON név = ivó;
```

A külső összekapcsolás megőrzi az egyik tábla azon sorait, amelyeknek nincs a másik táblában párja (lógó sorok), és **NULL** értékekkel helyettesíti a hiányzó mezőket. Példa:

<i>R</i>		<i>S</i>		<i>R OUTER JOIN S</i>		
A	B	B	C	A	B	C
a	b	b	c	a	b	c
c	d	f	s	c	d	(null)
				(null)	f	s

2.2.15. Megjegyzések. A **SELECT-FROM-WHERE** állítások multihalmaz szemantikát használnak, de a halmazműveleteknél (**UNION**, **INTERSECT**, **MINUS**) a halmaz szemantika érvényes.

Az összesítésekben az ismétlődések kiküszöbölése érdekében **DISTINCT**-et használunk. Példa:

```
-- Hány különféle áron árulják a 'Bud' sört?
SELECT COUNT(DISTINCT ár) FROM felszolgal WHERE sör = 'Bud';
```

Néha szükségünk lehet lekérdezések átírására, mivel nem minden relációsadatbázis-kezelő támogatja a használt kifejezéseket. A **MINUS**-t tartalmazó lekérdezéseket átírhatjuk **NOT EXISTS**-et használva. Példa:

```
SELECT d.dname FROM dept d
  MINUS
  SELECT d.dname FROM dept d, emp e WHERE d.deptno = e.deptno
  ORDER BY dname;

SELECT d.dname FROM dept d
  WHERE NOT EXISTS (SELECT NULL FROM emp e
                    WHERE e.deptno = d.deptno)
  ORDER BY dname;
```

3. FELADATOK

3.1. Micimackó és barátai (relációs algebra)

Szeret	
Név	Gyümölcs
Micimackó	málna
Micimackó	méz
Füles	körte
Malacka	méz
Malacka	málna
Malacka	körte
Kanga	banán
Tigris	méz

A feladatok egyszerű megoldása érdekében alkalmazzuk az alábbi átnevezést:

$$\rho_{SZ(N,GY)}(Szeret(Név, Gyümölcs))$$

3.1.1. Feladat. Melyek azok a gyümölcsök, amelyeket Micimackó szeret?

$$\pi_{GY}(\sigma_{N='Micimackó'}(SZ))$$

3.1.2. Feladat. Melyek azok a gyümölcsök, amelyeket Micimackó nem szeret?

$$\pi_{GY}(SZ) - \pi_{GY}(\sigma_{N='Micimackó'}(SZ))$$

3.1.3. Feladat. Melyek azok a gyümölcsök, amelyeket valaki szeret, és nem csak egyedül Micimackó?

$$\pi_{GY}(SZ - \sigma_{N='Micimackó'}(SZ)) = \pi_{GY}(\sigma_{N \neq 'Micimackó'}(SZ))$$

3.1.4. Feladat. Kik azok, akik *legalább* azokat a gyümölcsöket szeretik, mint Micimackó?

$$\begin{aligned}
 X &:= SZ \div \pi_{GY}(\sigma_{N='Micimackó'}(SZ)) \\
 &= \pi_N(SZ) - \pi_N\left(\underbrace{\pi_N(SZ) \times \pi_{GY}(\sigma_{N='Micimackó'}(SZ))}_{\substack{\text{Csak azok szeretik a Micimackó által szeretett} \\ \text{gyümölcsöket, akik eredetileg nem szerették}}}\right) - SZ \\
 &\quad \underbrace{\hspace{10em}}_{\substack{\text{Mindenki szereti a Micimackó által} \\ \text{szeretett gyümölcsöket}}}
 \end{aligned}$$

3.1.5. Feladat. Kik azok, akik *legfeljebb* azokat a gyümölcsöket szeretik, mint Micimackó?

$$Y := \pi_N(SZ) - \pi_N\left(\underbrace{SZ - \left(\pi_N(SZ) \times \pi_{GY}(\sigma_{N='Micimackó'}(SZ))\right)}_{\substack{\text{Akik mást is szeretnek a Micimackó által szeretett} \\ \text{gyümölcsökön kívül}}}\right)$$

3.1.6. Feladat. Kik azok, akik *pontosan* azokat a gyümölcsöket szeretik, mint Micimackó?

$$X \cap Y$$

3.1.7. Feladat. Kik szeretik az almát?

$$\pi_N(\sigma_{GY='alma'}(SZ))$$

3.1.8. Feladat. Kik nem szeretik az almát?

$$\pi_N(SZ) - \pi_N(\sigma_{GY='alma'}(SZ))$$

3.1.9. Feladat. Kik azok, akik szeretnek legalább egy almán kívüli gyümölcsöt?

$$\pi_N(SZ) - \pi_N(\sigma_{GY='alma'}(SZ))$$

3.1.10. Feladat. Kik szeretik az almát és a diót is?

$$\pi_N(\sigma_{GY='alma'}(SZ)) \cap \pi_N(\sigma_{GY='dió'}(SZ))$$

3.1.11. Feladat. Kik szeretik vagy az almát, vagy a diót?

$$\pi_N(\sigma_{GY='alma'}(SZ)) \cup \pi_N(\sigma_{GY='dió'}(SZ))$$

3.1.12. Feladat. Kik szeretik az almát, de a diót nem?

$$\pi_N(\sigma_{GY='alma'}(SZ)) - \pi_N(\sigma_{GY='dió'}(SZ))$$

A következő feladatokhoz alkalmazzuk a $\rho_{SZ_1}(SZ), \dots, \rho_{SZ_n}(SZ)$, $n \in \mathbb{N}$ átnevezéseket.

3.1.13. Feladat. Kik szeretnek legalább kétféle gyümölcsöt?

$$Z := \pi_{SZ_1.N}(\sigma_{SZ_1.GY \neq SZ_2.GY \wedge SZ_1.N = SZ_2.N}(SZ_1 \times SZ_2))$$

3.1.14. Feladat. Kik szeretnek legalább háromféle gyümölcsöt?

$$V := \pi_{SZ_1.N} \left(\sigma_{SZ_1.GY \neq SZ_2.GY \wedge SZ_2.GY \neq SZ_3.GY \wedge SZ_1.GY \neq SZ_3.GY} (SZ_1 \times SZ_2 \times SZ_3) \right) \\ \wedge SZ_1.N = SZ_2.N \wedge SZ_2.N = SZ_3.N$$

3.1.15. Feladat. Kik szeretnek legfeljebb kétféle gyümölcsöt?

$$\pi_N(SZ) - V$$

3.1.16. Feladat. Kik szeretnek pontosan kétféle gyümölcsöt?

$$(\pi_N(SZ) - V) \cap Z$$

3.1.17. Feladat. Kik, mit nem szeretnek?

$$\pi_N(SZ) \times \pi_{GY}(SZ) - SZ$$

3.1.18. Feladat. Kik nem szeretnek minden gyümölcsöt?

$$W := \pi_N(\pi_N(SZ) \times \pi_{GY}(SZ) - SZ) = \\ = \pi_N(SZ) - (SZ \div \pi_{GY}(SZ))$$

3.1.19. Feladat. Kik azok, akik minden gyümölcsöt szeretnek?

$$SZ \div \pi_{GY}(SZ) = \pi_N(SZ) - W$$

3.1.20. Feladat. Melyek azok a gyümölcsök, amiket legalább ketten szeretnek?

$$\pi_{GY}(\sigma_{SZ_1.GY = SZ_2.GY \wedge SZ_1.N \neq SZ_2.N} (SZ_1 \times SZ_2))$$

3.1.21. Feladat. Melyek azok a gyümölcsök, amit mindenki szeret?

$$SZ \div \pi_N(SZ)$$

Egészítsük ki a relációt egy *Mennyiség* attribútummal, amely az állatok által fogyasztott gyümölcsmennyiséget jelenti. Alkalmazzuk a $\rho_{SZ(N,GY,M)}(Szeret(Név, Gyümölcs, Mennyiség))$ átnevezést.

3.1.22. Feladat. Kik fogyasztják a legtöbb gyümölcsöt?

Nem lehet megoldani, hisz relációs algebrában nem lehet számolni, az eredmény sorrendje nem befolyásolható, nem lehet összesítést csinálni, nem lehet csoportítani, nem lehet a relációkat módosítani, illetve nem alkalmazhatunk rekurziót.

3.1.23. Feladat. Kik fogyasztják a legtöbb almát?

Nem lehet megoldani.

3.1.24. Feladat. Kik fogyasztják a legkevesebb almát?

Nem lehet megoldani.

3.1.25. Feladat. Kinek a neve mellett van a legnagyobb számérték?

$$\pi_N \left(SZ - \pi_{SZ_2.N, SZ_2.GY, SZ_2.M} \left(\sigma_{SZ_1.M > SZ_2.M} (SZ_1 \times SZ_2) \right) \right)$$

3.2. Micimackó és barátai (SQL)

Ha egy feladathoz több lekérdezés is tartozik, akkor mindegyik ugyan azt az eredményt adja. Néhány feladathoz nem tartozik megoldás, ám ezek könnyen kitalálhatóak az előző feladatokból, vagy egyszerűen átírhatóak a relációs algebrai kifejezésekből.

```
-- [3.1.1.] Melyek azok a gyümölcsök, amelyeket Micimackó szeret?
SELECT gy FROM sz WHERE n = 'Micimackó';

-- [3.1.2.] Melyek azok a gyümölcsök, amelyeket Micimackó nem szeret?
SELECT gy FROM sz MINUS SELECT gy FROM sz WHERE m = 'Micimackó';

-- [3.1.3.] Melyek azok a gyümölcsök, amelyeket valaki szeret,
--           és nem csak egyedül Micimackó?
SELECT DISTINCT gy FROM sz WHERE n <> 'Micimackó';
SELECT DISTINCT gy
  FROM (SELECT * FROM sz MINUS SELECT * FROM sz WHERE n = 'Micimackó');

-- [3.1.4.] Kik azok, akik legalább azokat a gyümölcsöket szeretik,
--           mint Micimackó?
SELECT n FROM sz
  MINUS
  SELECT n FROM (SELECT * FROM
    ( SELECT n FROM sz ),
    ( SELECT gy FROM sz WHERE n = 'Micimackó' )
  MINUS
  SELECT * FROM sz);

-- [3.1.5.] Kik azok, akik legfeljebb azokat a gyümölcsöket szeretik,
--           mint Micimackó?
SELECT n FROM sz
  MINUS
  SELECT n FROM (SELECT * FROM sz
    MINUS
    SELECT * FROM
      ( SELECT n FROM sz ),
      ( SELECT gy FROM sz WHERE n = 'Micimackó' ));

-- [3.1.6.] Kik azok, akik pontosan azokat a gyümölcsöket szeretik,
--           mint Micimackó?
SELECT n FROM sz
  MINUS
  SELECT n FROM (SELECT * FROM
    ( SELECT n FROM sz ),
    ( SELECT gy FROM sz WHERE n = 'Micimackó' )
  MINUS
  SELECT * FROM sz)
INTERSECT
  SELECT n FROM sz
  MINUS
```

```

SELECT n FROM (SELECT * FROM sz
                MINUS
                SELECT * FROM
                  (SELECT n FROM sz),
                  (SELECT gy FROM sz WHERE n = 'Micimackó'));

-- [3.1.7.] Kik szeretik az almát?
SELECT DISTINCT n FROM sz WHERE gy = 'alma';

-- [3.1.8.] Kik nem szeretik az almát?
SELECT n FROM sz
  MINUS
  SELECT n FROM sz WHERE gy = 'alma';

-- [3.1.10.] Kik szeretik az almát és a diót is?
SELECT n FROM sz sz1, sz sz2
  WHERE sz1.n = sz2.n AND sz1.gy = 'alma' AND sz2.gy = 'dió';
SELECT n FROM sz WHERE gy = 'alma'
  INTERSECT
  SELECT n FROM sz WHERE gy = 'dió';

-- [3.1.11.] Kik szeretik vagy az almát, vagy a diót?
SELECT n FROM sz WHERE gy = 'alma'
  UNION
  SELECT n FROM sz WHERE gy = 'dió';

-- [3.1.12.] Kik szeretik az almát, de a diót nem?
SELECT n FROM sz WHERE gy = 'alma'
  MINUS
  SELECT n FROM sz WHERE gy = 'dió';

-- [3.1.17.] Kik, mit nem szeretnek?
SELECT * FROM (SELECT n FROM sz), (SELECT gy FROM sz)
  MINUS
  SELECT * FROM sz;

-- [3.1.21.] Melyek azok a gyümölcsök, amelyeket mindenki szeret?
SELECT n FROM sz
  MINUS
  SELECT n FROM (SELECT * FROM
                  (SELECT n FROM sz),
                  (SELECT gy FROM sz)
                  MINUS
                  SELECT * FROM sz);

-- [3.1.22.] Kik fogyasztják a legtöbb gyümölcsöt?
SELECT n
  FROM (SELECT n, SUM(m) FROM sz GROUP BY n ORDER BY 2 DESC)
  WHERE ROWNUM <= 1;

```

-- [3.1.23.] Kik fogyasztják a legtöbb almát?

```
SELECT *
  FROM (SELECT n FROM sz WHERE gy = 'alma' ORDER BY m DESC)
 WHERE ROWNUM <= 1;
```

-- [3.1.25.] Kinek a neve mellett van a legnagyobb számérték?

```
SELECT n
  FROM (SELECT * FROM sz
        MINUS
        SELECT sz2.n, sz2.gy, sz2.m
          FROM sz sz1, sz sz2
         WHERE sz1.m > sz2.m);
```

3.3. Hajók (relációs algebra)

A feladatok megoldásához használjuk az alábbi átnevezéseket:

$$\rho_{HO}(Hajóosztályok), \quad \rho_H(Hajók), \quad \rho_K(Kimenetelek), \quad \rho_{Cs}(Csaták)$$

$$\rho_{HO_1}(HO), \dots, \rho_{HO_n}(HO), \quad \rho_{H_1}(H), \dots, \rho_{H_n}(H), \quad \rho_{Cs_1}(Cs), \dots, \rho_{Cs_n}(Cs)$$

3.3.1. Feladat. Melyek azok a hajók, amelyeket 1921 előtt avattak fel?

$$\pi_{Hajónév}(\sigma_{Felavatva < 1921}(H))$$

3.3.2. Feladat. Adjuk meg azokat a hajóosztályokat a gyártó országok nevével együtt, amelyeknek az ágyú legalább 16-os kaliberűek.

$$\pi_{Osztály, Ország}(\sigma_{Kaliber \geq 16}(HO))$$

3.3.3. Feladat. Adja meg a Denmark Strait-csatában elsüllyedt hajók nevét.

$$\pi_{Hajónév}(\sigma_{Eredmény='elsüllyedt' \wedge Csatanév='Denmark Strait'}(K))$$

3.3.4. Feladat. Adjuk meg az adatbázisban szereplő összes hadihajó nevét. Ne feledjük, hogy a Hajók relációban nem feltétlen szerepel az összes hajó.

$$\pi_{Hajónév}(K) \cup \pi_{Hajónév}(H)$$

3.3.5. Feladat. Melyek azok az országok, amelyeknek csatahajóik és cirkálójajóik is voltak?

$$\pi_{Ország}(\sigma_{HO.Típus='bb'}(HO)) \cap \pi_{Ország}(\sigma_{HO.Típus='bc'}(HO))$$

$$\pi_{HO_1.Ország}(\sigma_{HO_1.Típus='bb' \wedge HO_2.Típus='bc' \wedge HO_1.Ország=HO_2.Ország}(HO_1 \times HO_2))$$

3.3.6. Feladat. Melyik hajó, melyik országban készült?

$$\pi_{Hajónév, Ország}(H \bowtie HO)$$

3.3.7. Feladat. Adjuk meg a Guadalcanal csatában részt vett hajók nevét, vízkiszorítását és ágyúinak számát.

$$\pi_{Hajónév,ÁgyúkSzáma,Vízkiszorítás}(\sigma_{Csatanév='Guadalcanal'}(K \bowtie H \bowtie HO))$$

3.3.8. Feladat. Soroljuk fel a biztosan 1943 előtt épült hajókat!

$$\pi_{Hajónév}(\sigma_{Felavatra < 1943}(H)) \cup \pi_{Hajónév}(\sigma_{Dátum < '1/1/43'}(Cs \bowtie K))$$

3.3.9. Feladat. Melyik csatában volt mindenféle eredmény?

$$\pi_{Eredmény,Csatanév}(K) \div \pi_{Eredmény}(K)$$

3.3.10. Feladat. Melyik években avattak legalább 3 hajót?

$$\pi_{Felavatra}(\sigma_{H_1.Felavatra=H_2.Felavatra \wedge H_2.Felavatra=H_3.Felavatra \wedge (H_1 \times H_2 \times H_3) \\ \wedge H_1.Hajónév \neq H_2.Hajónév \wedge H_2.Hajónév \neq H_3.Hajónév \wedge \\ \wedge H_1.Hajónév \neq H_2.Hajónév})$$

3.3.11. Feladat. Az 1921-es washingtoni egyezmény betiltotta a 35000 tonnánál súlyosabb hajókat. Adjuk meg azokat a hajókat, amelyek megszegték az egyezményt.

$$\pi_{Hajónév}(\sigma_{Vízkiszorítás > 35000 \wedge Felavatra \geq 1921}(H \bowtie HO))$$

3.3.12. Feladat. Adjuk meg azokat a hajókat, amelyek „újjáéledtek”, azaz egyszer már megsérültek egy csatában, de egy későbbi csatában újra harcoltak.

$$\pi_{K_1.Hajónév}(\sigma_{K_1.Eredmény='ok' \wedge K_2.Eredmény='sérült' \wedge ((K_1 \bowtie Cs_1) \times (K_2 \bowtie Cs_2)) \\ \wedge K_1.Hajónév=K_2.Hajónév \wedge \\ \wedge Cs_1.Dátum > Cs_2.Dátum})$$

3.3.13. Feladat. Adjuk meg azokat az osztályokat, amelyekbe csak egyetlenegy hajó tartozik.

$$\pi_{Osztály}(H) - \pi_{H_1.Osztály}(\sigma_{H_1.Osztály=H_2.Osztály \wedge H_1.Hajónév \neq H_2.Hajónév}(H_1 \times H_2))$$

3.3.14. Feladat. Évenkénti bontásban hány hajót avattak?

Nem lehet megoldani, csak kiterjesztett relációs algebrával.

3.3.15. Feladat. Mely hajóosztályból, mikor avatták az utolsó hajót?

Nem lehet megoldani.

3.4. Hajók (SQL)

-- [3.3.1.] Melyek azok a hajók, amelyeket 1921 előtt avattak fel?

SELECT hajónév **FROM** hajók **WHERE** felavatra < 1921;

-- [3.3.2.] Adjuk meg azokat a hajóosztályokat a gyártó országok

-- nevével együtt, amelyeknek az ágyúi legalább 16-os kaliberűek.

SELECT osztály, ország **FROM** hajóosztályok **where** kaliber >= 16;

-- [3.3.3.] Adja meg a Denmark Strait-csatában elsüllyedt hajók nevét.

```
SELECT hajónév FROM kimenetelek
WHERE eredmény = 'elsüllyedt' AND csatanév = 'Denmark_Strait';
```

-- [3.3.4.] Adjuk meg az adatbázisban szereplő összes hadihajó nevét.

```
SELECT hajónév FROM kimenetelek UNION SELECT hajónév FROM hajók;
```

-- [3.3.5.] Melyek azok az országok, amelyeknek csatahajóik

-- és cirkálóhajóik is voltak?

```
SELECT ország FROM hajóosztályok WHERE típus = 'bb'
INTERSECT
SELECT ország FROM hajóosztályok WHERE típus = 'bc';
```

-- [3.3.7.] Adjuk meg a Guadalcanal csatában részt vett hajók nevét,

-- vízkiszorítását és ágyúinak számát.

```
SELECT hajónév, ágyúkszám, vízkiszorítás
FROM kimenetelek NATURAL JOIN hajók NATURAL JOIN hajóosztályok
WHERE csatanév = 'Guadalcanal';
```

-- [3.3.12.] Adjuk meg azokat a hajókat, amelyek „újjáéledtek”,

-- azaz egyszer már megsérültek egy csatában, de egy későbbi

-- csatában újra harcoltak.

```
SELECT k1.hajónév
FROM (kimenetelek k1 NATURAL JOIN csaták cs1)
CROSS JOIN
(kimenetelek k2 NATURAL JOIN csaták cs2)
WHERE k1.eredmény = 'ok' AND k2.eredmény = 'sérült'
AND k1.hajónév = k2.hajónév AND cs1.dátum > cs2.dátum;
```

-- [3.3.13.] Adjuk meg azokat az osztályokat, amelyekbe csak egyetlenegy

-- hajó tartozik.

```
SELECT osztály FROM hajók
MINUS
SELECT h1.osztály FROM hajók h1, hajók h2
WHERE h1.osztály = h2.osztály AND h1.hajónév != h2.hajónév;
```

-- [3.3.14.] Évenkénti bontásban hány hajót avattak?

```
SELECT felavatva, COUNT(*) FROM hajók GROUP BY felavatva;
```

-- [3.3.15.] Mely hajóosztályból, mikor avatták az utolsó hajót?

```
SELECT osztály, MAX(felavatva) FROM hajók GROUP BY osztály;
```

3.5. Minta ZH.

Adottak a következő relációsémák:

Családtag(szigsz, vezetéknév, keresztnév, életévek)

Házasság(házip, ffigsig, nőiszig, vendégekszáma)

Szüelő(szigsz, gyerekszig)

Járt_ott(szigsz, házip, ajándékdb)

3.5.1. Feladat. Ki(k) a legfiatalabb családtagok?

$$\pi_{szigsz} \left(Cs - \pi_{Cs_2} \left(\sigma_{Cs_1.életévek < Cs_2.életévek} (Cs_1 \times Cs_2) \right) \right)$$

3.5.2. Feladat. Ki vette el a nejét közepes (20-50 fős) esküvőn?

$$\pi_{ffiszig} \left(\sigma_{vendégekszáma \leq 50 \wedge vendégekszáma \geq 20} (H) \right)$$

3.5.3. Feladat. Ki(k)nek van Gergő nevű unokájuk?

$$\pi_{Sz_1.szigsz} \left(\sigma_{k.név='Gergő'} \left(Sz_1 \bowtie Sz_1.gyerekszig = Sz_2.szigsz \left(Sz_2 \bowtie Sz_2.gyerekszig = Cs.szigsz Cs \right) \right) \right)$$

$$\pi_{Sz_2.szigsz} \left(\sigma_{Sz_2.gyerekszig = Sz_1.szigsz (Sz_1 \times Sz_2) \bowtie Sz_1.gyerekszig = Cs.szigsz \wedge Cs} \right) \\ \wedge Cs.k.név='Gergő'$$

3.5.4. Feladat. Ki járt minden olyan esküvőn, ahol 'Kiss Éva' is?

$$\pi_{szigsz, házip} (J) \div \pi_{házip} \left(\sigma_{v.név='Kiss' \wedge k.név='Éva'} (Cs) \bowtie J \right)$$

3.5.5. Feladat. Melyik férfi esküvőjén járt minden 'Öreg János'-nál idősebb családtag?

$$\pi_{ffiszig} \left(\pi_{szigsz, házip} (J) \div \pi_{Cs_2.szigsz} \left(Cs_1 \bowtie Cs_1.v.név='Öreg' \wedge Cs_1.k.név='János' \wedge Cs_2 \right) \bowtie H \right) \\ \wedge Cs_1.kor < Cs_2.kor$$

3.5.6. Feladat. Kinek (név) van pontosan egy gyereke?

$$\left(\pi_{szigsz} (Sz) - \pi_{Sz_1.szigsz} \left(\sigma_{Sz_1.szigsz = Sz_2.szigsz \wedge Sz_1.gyerekszig \neq Sz_2.gyerekszig} (Sz_1 \times Sz_2) \right) \right) \bowtie Cs$$

3.5.7. Feladat. Ki (szigsz, vezetéknév) nem adta egyik gyerekének sem a 'László' keresztnévet?

```
SELECT szigsz, vezetéknév
FROM családtag NATURAL JOIN
(SELECT szigsz FROM szülő WHERE gyerekszig NOT IN
(SELECT szigsz FROM családtag WHERE keresztnév = 'László'));
```

3.5.8. Feladat. Ki(k) az(ok) (név), aki(k) minden esküvőre, ahova elment(ek), annyi ajándékot vitt(ek), aminél soha senki nem vitt többet (azaz legtöbbet)?

```
SELECT vezetéknév, keresztnév
FROM családtag NATURAL JOIN
(SELECT szigsz FROM járt_ott
WHERE ajándékdb = (SELECT MAX(ajándékdb) FROM járt_ott));
```

3.5.9. Feladat. Ki járt minden olyan esküvőn, ahol járt minden olyan családtag, akinek van 'Anita' nevű gyereke?

A megoldáshoz két osztást használunk.

```
SELECT szigsz FROM járt_ott -- első osztás kezdete
MINUS
SELECT szigsz FROM (
  SELECT * FROM (SELECT szigsz FROM járt_ott)
    CROSS JOIN
    (SELECT házid FROM járt_ott -- második osztás
      MINUS
      SELECT házid FROM (
        SELECT * FROM
          (SELECT házid FROM járt_ott)
        CROSS JOIN
        (SELECT szigsz FROM szülő WHERE gyerekszig IN
          (SELECT szigsz FROM családtag
            WHERE vezetéknév = 'Anita'))
        MINUS
        SELECT házid, szigsz FROM járt_ott))
      MINUS
      SELECT házid, szigsz FROM járt_ott);
```

3.5.10. Feladat. $\pi_{szigsz,v.név}(Cs) \div \rho_{JH(szigsz)}(\pi_{ffiszig}(\sigma_{ajándékd>5}(J) \bowtie H))$

```
SELECT vezetéknév FROM családtag
MINUS
SELECT vezetéknév FROM (
  SELECT * FROM (SELECT vezetéknév FROM családtag)
    CROSS JOIN
    (SELECT ffiszig szigsz
      FROM ((SELECT * FROM járt_ott WHERE ajándékd > 5)
        NATURAL JOIN
        házasság))
    MINUS
    SELECT vezetéknév, szigsz FROM családtag);
```

3.5.11. Feladat. Táblázatos formában írja ki, hogy egy gyerek szüleinek az esküvőjén hány vendég volt! (gyerekszig, vendégekszáma)

```
-- A gyereknek egy vagy kettő szülője van (tágabb)
SELECT DISTINCT gyerekszig, vendégekszáma
  FROM szülő JOIN házasság ON szigsz = nőiszig OR szigsz = ffiszig;

-- A gyereknek pontosan kettő szülője van (szűkebb)
SELECT gyerek gyerekszig, vendégekszáma
  FROM házasság NATURAL JOIN (
    SELECT sz1.szigsz ffiszig, sz2.szigsz nőiszig, sz1.gyerekszig gyerek
    FROM szülő sz1 CROSS JOIN szülő sz2
    WHERE sz1.gyerekszig = sz2.gyerekszig AND sz1.szigsz != sz2.szigsz);
```

FORRÁSOK

- [1] Hajas Cs.: *Előadás*. ELTE IK, 2013.
http://people.inf.elte.hu/sila/AB1EA/AB1_eloadasanyag.html
- [2] Balogh T.: *Előadás és gyakorlat jegyzet (Brányi L. gyakorlatai, illetve Hajas Cs. előadásai alapján)*. ELTE IK, 2013. március 20.
<http://baloghtamas.hu/download/gyakorlatadatb.pdf>
- [3] Bekő T.: *Gyakorlat jegyzet (Brányi L. gyakorlatai alapján)*. ELTE IK, 2013.