









Relačná algebra

1.1 Základné operácie

Základnými operáciami relačnej algebry sú:

1. výber 
2. projekcia 
3. kartézsky súčin 
4. zjednotenie 
5. rozdiel 
6. prienik 
7. delenie 
8. spojenie 

1.2 Klasifikácia operácií

Tieto operácie môžu byť klasifikované z rôznych pohľadov:

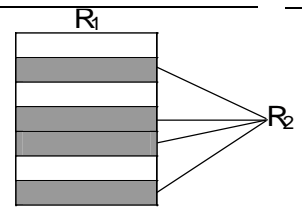
1. Podľa počtu zdrojových relácií

- **Unárne** – výber, projekcia
- **Binárne** – kartézsky súčin, prienik, rozdiel, zjednotenie, delenie, spojenie

2. Podľa typu operácií

- **Množinové operácie** – zjednotenie, prienik, kartézsky súčin, rozdiel
- **Relačné operácie** – spojenie, delenie, výber, projekcia

1.2.1 VÝBER - SELECTION



Definícia – Elementárna podmienka EC

Elementárnou podmienkou EC nazývame výraz v tvare:

<Atribút> <Operátor> <Hodnota|Výraz|Atribút>,
 kde operátor je jeden z množiny relačných operátorov:
 $\{=, <, >, <=, >=, \neq\}$.

Poznámka: Hodnotu v podmienke môžeme vyjadriť konštantou, výrazom, alebo atribútom



Podmienkou C nazývame výraz v tvare:

[NOT] EC_1 [{ **OR** | **AND** | **NOT** } [[**NOT**] EC_2]...]

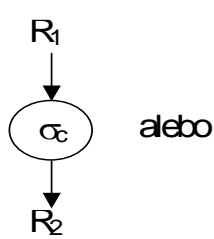
♦

Operácia **SELECTION** (Výber) vytvorí z relácie $\mathbf{R_1(A_1, A_2, \dots, A_n)}$ reláciu $\mathbf{R_2(A_1, A_2, \dots, A_n)}$ takú, že pre každú n -ticu $\mathbf{t \in R_2}$ platí $\mathbf{t \in R_1}$ a je splnená podmienka \mathbf{C} .

♦

Označenie

a) grafické



a) matematické

$$R_2 = \sigma_c(R_1)$$



□ *Príklad 1.1 – Výber*

Vypíšte všetky údaje o predmetoch, ktoré garantuje učiteľ s osobným číslom="KI001"

a) pomocou operácií relačnej algebry

$\sigma_{\text{cis_ucitel}=\text{"KI001"}}(\text{predmet})$

b) pomocou SQL príkazu

```
SELECT * FROM predmet
WHERE cis_ucitel = "KI001";
```

| cis_predmet | Nazov | kredity | cis_ucitel |
|-------------|-----------------------|---------|------------|
| P111 | Základy informatiky 1 | 6 | KI001 |
| P211 | Základy informatiky 2 | 6 | KI001 |
| A502 | C-jazyk | 8 | KI001 |

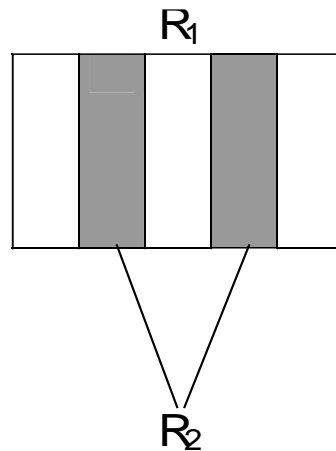
Komutativita výberu

$$\sigma_{\text{cond1}}(\sigma_{\text{cond2}}(\mathbf{R})) = \sigma_{\text{cond2}}(\sigma_{\text{cond1}}(\mathbf{R}))$$

Kaskáda výberu (pre konjunkciu)

$$\sigma_{\text{cond1}}(\sigma_{\text{cond2}}(\dots(\sigma_{\text{condn}}(\mathbf{R})))) = \sigma_{\text{cond1 AND cond2 AND ...AND condn}}(\mathbf{R})$$

1.2.2 PROJEKCIA – PROJECTION



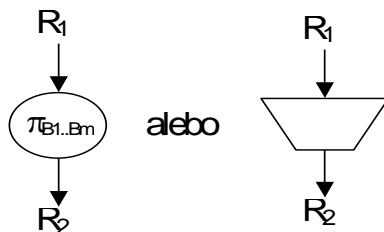
Definícia - Projection



Operácia PROJECTION (Projekcia) vytvorí z relácie $R_1(A_1, A_2, \dots, A_n)$ reláciu $R_2(B_1, B_2, \dots, B_m)$ takú, že množina atribútov $(B_1, B_2, \dots, B_m) \subset (A_1, A_2, \dots, A_n)$ a pre stupeň relácie R_2 platí $m < n$ a $\text{card}(R_2) = \text{card}(R_1)$. ♦

Označenie

a) grafické



b) matematické

$$R_2 = \pi_{B_1..B_m}(R_1)$$

□ *Príklad 1.2 - Projekcia*

Vypíšte zoznam mien a priezvisk všetkých študentov

a) pomocou operácií relačnej algebry

$\pi_{\text{meno}, \text{priezvisko}}(\text{os_udaje})$

b) pomocou SQL príkazu

```
SELECT meno, priezvisko  
FROM os_udaje;
```

| meno | priezvisko |
|------------|-------------|
| Peter | Novák |
| Stanislav | Steinmüller |
| János | Tóth |
| Marek | Rátroch |
| Bohuslav | Biely |
| Branislav | Baláž |
| Peter | Kapustný |
| Marek | Đurica |
| Martin | Klíčiar |
| Lukáš | Satrapa |
| Ján | Krnáč |
| Juraj | Papún |
| Andrej | Janči |
| Zdeno | Svetkovský |
| ... | ... |
| Stanislava | Slámová |
| Erika | Lipovská |
| Peter | Malík |

Kaskáda projekcií

Ak zoznam *zoznam2* atribútov projekcie obsahuje zoznam atribútov *zoznam1*, tak môžeme písať:

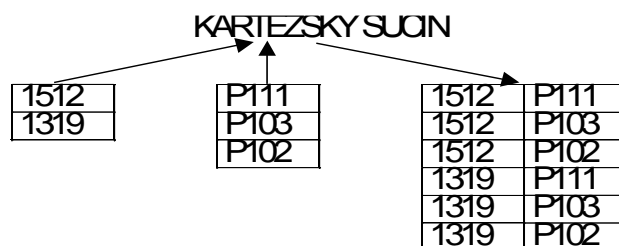
$$\pi_{\text{zoznam1}}(R) = \pi_{\text{zoznam1}}(\pi_{\text{zoznam2}}(R))$$

1.2.3 KARTÉZSKY SÚČIN – PRODUCT

Definícia - Product 

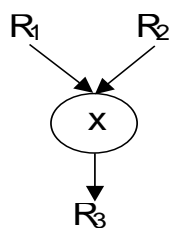
Operácia PRODUCT (Karteziánsky súčin) vytvorí z relácie $\mathbf{R}_1(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$ a z relácie $\mathbf{R}_2(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m)$, tretiu reláciu $\mathbf{R}_3(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m)$ takú, že obsahuje všetky kombinácie n-tíc z \mathbf{R}_1 a \mathbf{R}_2 , kde pre každú n-ticu t platí:

$t \in \mathbf{R}_3$ a t je usporiadanou dvojicou
 $t = t_1, t_2$ ak $t_1 \in \mathbf{R}_1$ a $t_2 \in \mathbf{R}_2$.



Označenie

a) grafické



b) matematické

$$R_3 = R_1 \times R_2$$

□ *Príklad 1.3– Kartézsky súčin*

Vypíšte všetky kombinácie mien a priezvisk študentov a čísel predmetov, ktoré je možné si zapísať.

a) pomocou operácií relačnej algebry

```
a =  $\pi_{\text{meno}, \text{priezvisko}}(\text{os\_udaje})$  //vyber všetky mená a priezviská študentov  
b =  $\pi_{\text{cis\_predmet}}(\text{predmet})$  // vyber všetky čísla predmetov  
Kartézsky_súčin = a × b // vytvor všetky kombinácie
```

b) pomocou príkazov SQL

```
CREATE TABLE A  
AS  
SELECT meno, priezvisko FROM  
os_udaje;
```

```
CREATE TABLE B  
AS  
SELECT cis_predmet FROM predmet;
```

```
SELECT meno, priezvisko, cis_predmet  
FROM A, B;
```

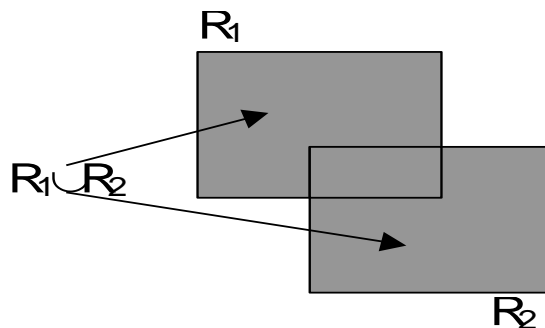
| Meno | priezvisko | cis_predmet |
|-----------|-------------|-------------|
| Peter | Novák | A501 |
| Stanislav | Steinmüller | A501 |
| János | Tóth | A501 |
| ... | ... | ... |
| Peter | Novák | A901 |
| Stanislav | Steinmüller | A901 |
| János | Tóth | A901 |
| ... | ... | ... |

1.2.4 ZJEDNOTENIE - UNION**Definícia - Union**

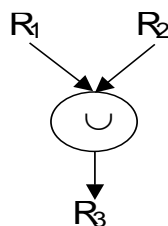
Operácia UNION (zjednotenie) vytvorí z relácie $R_1(A_1, A_2, \dots, A_n)$ a z relácie $R_2(A_1, A_2, \dots, A_n)$, tretiu reláciu $R_3(A_1, A_2, \dots, A_n)$ takú, že pre každú n -ticu t platí:

$$t \in R_3 \quad \text{ak } t \in R_1, \text{ alebo } t \in R_2$$

♦

**Označenie**

a) grafické



b) matematické

$$R_3 = R_1 \cup R_2$$

$$R_3 = \text{UNION } (R_1, R_2)$$

$$R_3 = R_1 \text{ UNION } R_2$$

Definícia – Union kompatibilita

Dve relácie \mathbf{R}_1 a \mathbf{R}_2 sú union kompatibilné, ak majú *totožnú* množinu atribútov.

♦

□ *Príklad 1.4 – Union kompatibilné relácie*

Ak máme reláciu

Študent_ZA(os_cislo,rod_cislo, st_zameranie,
st_odbor, rocnik, forma, stav, dat_1zapisu,
st_skupina, dat_ukoncenia)

a reláciu

študent_PD(os_cislo,rod_cislo, st_zameranie,
st_odbor, rocnik, forma, stav, dat_1zapisu,
st_skupina, dat_ukoncenia),

tak relácie sú union kompatibilné.

□ *Príklad 1.5 – Union nekompatibilné relácie*

Ak máme reláciu

študent(os_cislo, rod_cislo, st_zameranie,
st_odbor, rocnik, forma, stav, dat_1zapisu,
st_skupina, dat_ukoncenia)

a reláciu

študenti(rod_cislo, os_cislo, st_zameranie,
st_odbor, rocnik, forma, stav, dat_1zapisu,
st_skupina, dat_ukoncenia),

tak relácie nie sú union kompatibilné, pretože
poradie atribútov nie je totožné.

□ *Príklad 1.6 – Union nekompatibilné relácie*

Ak máme reláciu

študent(os_cislo, rod_cislo, st_zameranie,
st_odbor, rocnik, forma, stav, dat_1zapisu,
st_skupina, dat_ukoncenia)

a reláciu

študenti(os_cislo, rod_cislo, st_zameranie,
st_odbor, rocnik),

tak relácie nie sú union kompatibilné, pretože počet atribútov nie je totožný.

□ *Príklad 1.7 – Zjednotenie*

Vytvorte dve relácie z relácie osobné údaje také, že prvá relácia bude obsahovať mená a priezviská všetkých žien a druhá relácia mená a priezviská všetkých mužov.

Potom napíšte príkaz pre zjednotenie oboch relácií.

a) pomocou operácií relačnej algebry

$os_udaje_1 = \pi_{meno, priezvisk}(os_udaje_skola)$

$os_udaje_2 = \pi_{meno, priezvisko}(os_udaje_primacky)$

$Zjednotenie = os_udaje_1 \cup os_udaje_2$

b) pomocou príkazov SQL

```
CREATE TEMP TABLE os_udaje_1
AS SELECT meno, priezvisko
FROM os_udaje_skola;
```

```
CREATE TEMP TABLE os_udaje_2
AS SELECT meno, priezvisko
FROM os_udaje_primacky;
```

```
SELECT meno, priezvisko FROM
os_udaje_1
UNION
SELECT meno, priezvisko FROM
os_udaje_2;
```

| meno | priezvisko |
|------------|-------------|
| Andrej | Janči |
| Bohuslav | Biely |
| Branislav | Baláž |
| Erika | Lipovská |
| František | Murgaš |
| Juraj | Papún |
| Ján | Krnáč |
| János | Tóth |
| Marek | Rátroch |
| Martin | Kľúčiar |
| Peter | Kapustný |
| Peter | Malík |
| Rastislav | Kontroš |
| Rudolf | Kováč |
| Stanislav | Steinmüller |
| Stanislava | Slámová |
| Zdeno | Svetkovský |
| Ľuboš | Lehotský |

Komutativita

$$R_1 \cup R_2 = R_2 \cup R_1$$

Asociativita

$$R_1 \cup (R_2 \cup R_3) = (R_1 \cup R_2) \cup R_3$$

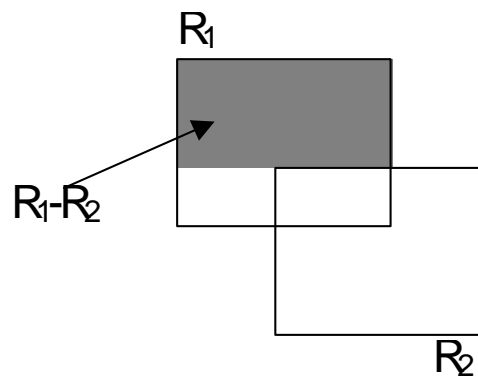
1.2.5 ROZDIEL - DIFFERENCE

Definícia - Difference

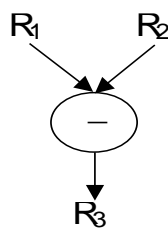
Operácia DIFFERENCE (rozdiel) vytvorí z relácie $R_1(A_1, A_2, \dots, A_n)$ a z relácie $R_2(A_1, A_2, \dots, A_n)$, tretiu reláciu $R_3(A_1, A_2, \dots, A_n)$ takú, že pre každú n -ticu t platí:

$$t \in R_3 \quad \text{ak } t \in R_1 \text{ a } t \notin R_2.$$

♦

**Označenie**

a) grafické



b) matematické

$$R_3 = R_1 - R_2$$

$$R_3 = R_1 \text{ DIFFERENCE } R_2$$

$$R_3 = \text{DIFFERENCE } (R_1, R_2)$$

$$R_3 = \text{SUBTRACT } (R_1, R_2)$$

Poznámka:

$$R_1 - R_2 \neq R_2 - R_1$$

□ *Príklad - Rozdiel*

Pomocou operácie rozdiel vypíšte rodné čísla študentov z relácie osobné údaje, ktorí **nie sú** študentmi druhého ročníka.

a) pomocou operácií relačnej algebry

$$R_1 = \pi_{\text{rod_cislo}}(\text{os_udaje})$$

$$R_2 = \pi_{\text{rod_cislo}}(\sigma_{\text{rocnik} = 2}(\text{student}))$$

$$\text{ROZDIEL} = R_1 - R_2$$

b) pomocou príkazov SQL

```
CREATE TABLE R1
AS
SELECT rod_cislo FROM os_udaje;

CREATE TABLE R2
AS
SELECT rod_cislo FROM student
WHERE rocnik = 2;

SELECT * FROM R1
WHERE rod_cislo NOT IN
(SELECT rod_cislo FROM R2);
```

| rod_cislo |
|-------------|
| 755022/8569 |
| 760103/2238 |
| 770913/3326 |
| 771203/5472 |
| 781001/3623 |
| 781015/4431 |
| 781130/4454 |
| 781201/1248 |

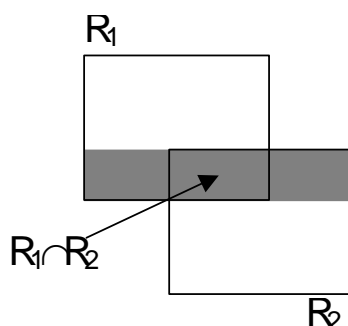
1.2.6 PRIENIK – INTERSECTION

Definícia - Intersection

Operácia INTERSECTION (prienik) vytvorí z relácie $\mathbf{R}_1(A_1, A_2, \dots, A_n)$ a z relácie $\mathbf{R}_2(A_1, A_2, \dots, A_n)$ tretiu reláciu $\mathbf{R}_3(A_1, A_2, \dots, A_n)$ takú, že pre každú n -ticu t platí:

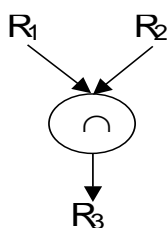
$$t \in \mathbf{R}_3 \quad \text{ak } t \in \mathbf{R}_1 \text{ a súčasne } t \in \mathbf{R}_2.$$

♦



Označenie

a) grafické



b) matematické

$$R_3 = R_1 \cap R_2$$

$$R_3 = R_1 \text{ INTERSECTION } R_2$$

$$R_3 = \text{INTERSECTION } (R_1, R_2)$$

Operáciu relačnej algebry prienik je možné vyjadriť pomocou operácie rozdiel, potom pre prienik relácii R_1 a R_2 platí:

$$R_3 = R_1 - (R_1 - R_2), \text{ alebo}$$

$$R_3 = R_2 - (R_2 - R_1)$$

alebo

pomocou operácií zjednotenie a rozdiel

$$R_3 = (R_1 \cup R_2) - ((R_1 - R_2) \cup (R_2 - R_1))$$

□ *Príklad 1.9 - Prienik*

Pomocou operácie prienik vypíšte všetky rodné čísla študentov z relácie osobné údaje, ktorí **sú** študentmi druhého ročníka.

- a) pomocou operácií relačnej algebry

$$R_1 = \pi_{\text{rod_cislo}}(\text{os_udaje})$$
$$R_2 = \pi_{\text{rod_cislo}}(\sigma_{\text{rocnik}=2}(\text{student}))$$
$$\text{PRIENIK} = R_1 \cap R_2$$

- b) pomocou príkazov SQL

```
CREATE TABLE R1
AS SELECT rod_cislo FROM os_udaje
INTO TEMP R1;
```

```
CREATE TABLE R2
AS SELECT rod_cislo FROM student
WHERE rocnik = 2;
```

```
SELECT * FROM R1
WHERE EXISTS
  (SELECT 'x' FROM R2
   WHERE R1.rod_cislo=R2.rod_cislo);
```

| |
|-------------|
| rod_cislo |
| 771124/3578 |
| 790907/1259 |
| 791229/5431 |
| 800312/7845 |
| 800407/3522 |

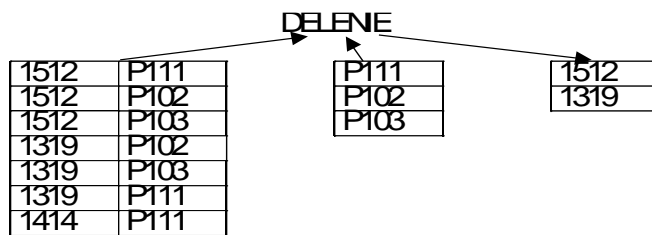
Komutativita

$$R_1 \cap R_2 = R_2 \cap R_1$$

Asociativita

$$R_1 \cap (R_2 \cap R_3) = (R_1 \cap R_2) \cap R_3$$

1.2.7 DELENIE – DIVISION

*Definícia – Division*

Operácia **DIVISION** (delenie) vytvorí z relácie $D(A_1, A_2, \dots, A_p, A_{p+1}, A_{p+2}, \dots, A_n)$ delením reláciou $d(A_{p+1}, A_{p+2}, \dots, A_n)$ tretiu reláciu $Q(A_1, A_2, \dots, A_p)$ takú, že konkatenciou $t_Q \in Q$ a $t_d \in d$ dostaneme n-ticu $t_D \in D$.

$$(t_Q, t_d = t_D)$$

*Definícia - Division*

Nech $X = (A_1, A_2, \dots, A_p)$ a $Y = (A_{p+1}, A_{p+2}, \dots, A_n)$ potom operácia **DIVISION** (delenie) vytvorí z binárnej relácie $D(X, Y)$ delením unárnou reláciou $d(Y)$ tretiu unárnu reláciu $Q(X)$ takú, že konkatenciou $t_Q \in Q$ a $t_d \in d$ dostaneme n-ticu $t_D \in D$

$$(t_Q, t_d = t_D)$$

Veta

$$\mathbf{D} \div \mathbf{d} = \mathbf{R}_1 - \mathbf{R}_2$$

kde

$$\mathbf{R}_1 = \pi_{A_1, A_2, \dots, A_p}(\mathbf{D})$$

$$\mathbf{R}_2 = \pi_{A_1, A_2, \dots, A_p}((\mathbf{R}_1 \times \mathbf{d}) - \mathbf{D})$$

Veta

$$\mathbf{D} \div \mathbf{d} = \mathbf{R}_1 - \mathbf{R}_2$$

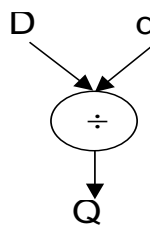
kde

$$\mathbf{R}_1 = \pi_X(\mathbf{D})$$

$$\mathbf{R}_2 = \pi_X((\mathbf{R}_1 \times \mathbf{d}) - \mathbf{D})$$

Označenie

a) grafické



b) matematické

$$Q = D \div d$$

□ *Príklad 1.10*

Vypíšte osobné čísla všetkých študentov, ktorí majú zapísané oba predmety: P202 a P301.

a) riešenie číslo 1 - pomocou príkazov SQL

```
CREATE TABLE menovatel
AS SELECT cis_predmet FROM predmet
WHERE predmet.cis_predmet = "P202"
OR
    predmet.cis_predmet = "P301";
```

```
CREATE TABLE pocet_v_menovateli
AS
SELECT COUNT(*) pocet FROM
menovatel;
```

```
CREATE TABLE st_pr AS
SELECT unique os_cislo, cis_predmet
FROM zap_predmety
WHERE cis_predmet IN
    (SELECT cis_predmet FROM
    menovatel);
```

```
CREATE TABLE pocet_st_pr AS
SELECT os_cislo, COUNT(*) pocet
FROM st_pr
GROUP BY os_cislo;
```

```
SELECT os_cislo
FROM pocet_st_pr z JOIN
pocet_v_menovateli y
ON(z.pocet = y.pocet);
```

b) riešenie číslo 2 - pomocou príkazov relačnej algebry

menovatel = $\sigma_{\text{cis_predmet}=\text{"P202"} \text{ OR } \text{cis_predmet}=\text{"P301"}}(\pi_{\text{cis_predmet}}(\text{predmet}))$

R1 = $\sigma_{\text{unique}}(\pi_{\text{os_cislo}, \text{cis_predmet}}(\text{zap_predmety} \bowtie_{\text{cis_predmet}} \text{menovatel}))$

R = $\sigma_{\text{unique}}(\pi_{\text{os_cislo}}(\text{R1}))$

R2 = $\pi_{\text{os_cislo}}((\text{R} \times \text{menovatel}) - \text{R1})$

PODIEL = **R** - **R2**

c) riešenie číslo 2 - pomocou príkazov SQL

```
SELECT cis_predmet FROM predmet
WHERE cis_predmet = "P202"
      OR cis_predmet = "P301"
INTO TEMP menovatel;
```

```
SELECT UNIQUE os_cislo , cis_predmet
FROM zap_predmety
WHERE cis_predmet IN
(SELECT cis_predmet FROM menovatel)
INTO TEMP R1;
```

```
SELECT unique os_cislo FROM R1
INTO TEMP R;
```

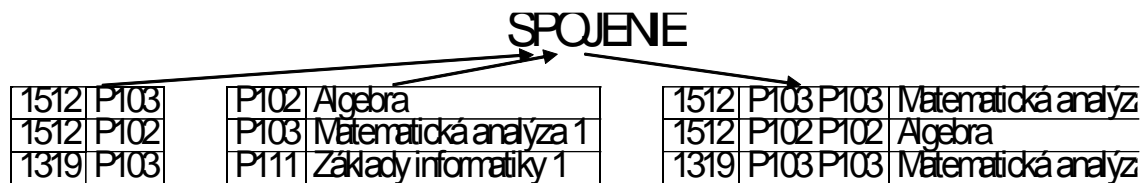
```
SELECT os_cislo FROM R, menovatel
WHERE cis_predmet NOT IN
(SELECT cis_predmet FROM R1)
```

```
INTO TEMP R2;
```

```
SELECT os_cislo FROM R
WHERE os_cislo NOT IN
      (SELECT os_cislo FROM R2);
```

| os_cislo |
|----------|
| 1402 |
| 1555 |

1.2.8 SPOJENIE – JOIN

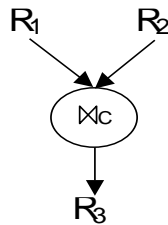
*Definícia – Join*

Operácia JOIN (spojenie) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$, ktoré majú spoločnú množinu atribútov X tretiu reláciu $R_3(X, A_1, A_2, \dots, A_n, X, B_1, B_2, \dots, B_m)$ takú, že ak $t_1 \in R_1$ a ak $t_2 \in R_2$ a pre hodnoty atribútov X platí $t_1.X = t_2.X$, potom n -tica $t \in R_3$ má atribúty $t = t_1.X, t_1.A_1, t_1.A_2, \dots, t_1.A_n, t_2.X, t_2.B_1, t_2.B_2, \dots, t_2.B_m$. ♦

Pri operácii spojenie si je potrebné uvedomiť, že množiny atribútov, cez ktoré sa spojenie realizuje môžu, ale nemusia mať rovnaké mená, ale vždy musia mať rovnakú doménu.

Označenie

a) grafické



b) matematické

$$R_3 = R_1 \bowtie_c R_2$$

$$R_3 = R_1 \text{ JOIN}_C R_2$$

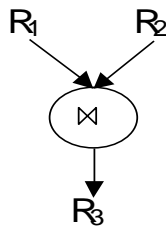
$$R_3 = \text{JOIN}(R_1, R_2, C)$$

Poznámka

V prípade, že množiny atribútov, cez ktoré sa realizuje spojenie majú rovnaké mená, nie je nutné pri operátore spojenia uvádzať podmienku vyjadrujúcu spojenie atribútov dvoch relácií.

Označenie v prípade rovnakých mien atribútov

a) grafické



b) matematické

$$R_3 = R_1 \bowtie R_2$$

$$R_3 = R_1 \text{ JOIN } R_2$$

$$R_3 = \text{JOIN}(R_1, R_2)$$

Operáciu relačnej algebry spojenie je možné vyjadriť pomocou operácií kartézsky súčin a výber. V tom prípade pre spojenie $R_3 = R_1 \bowtie_C R_2$ platí:

$$R_3 = \sigma_C(R_1 \times R_2)$$

- *Príklad 1.11 – Spojenie s použitím rovnakých mien atribútov*

```
SELECT os_udaje.*,student.* FROM  
os_udaje JOIN student ON  
(os_udaje.rod_cislo=student.rod_cislo)
```

- *Príklad 1.12 – Spojenie s použitím rôznych mien atribútov*

V prípade, že by v relácii študent atribút, v ktorom bude uložené rodné číslo mal meno RC, príklad spojenia by vyzeral nasledovne:

```
SELECT os_udaje.*,student.* FROM  
os_udaje JOIN student ON  
(os_udaje.rod_cislo=student.rc)
```

□ *Príklad 1.13 - Spojenie*

Pomocou operácie spojenie, spojte nasledovné relácie R_1 a R_2 .

$$R_1 = \pi_{\text{rod_cislo, meno, priezvisko}}(\text{os_udaje})$$

$$R_2 = \pi_{\text{os_cislo, rod_cislo, rocnik, st_skupina}}(\text{student})$$

a) pomocou operácií relačnej algebry:

$$\text{SPOJENIE} = R_1 \bowtie R_2$$

b) pomocou príkazov SQL

```
SELECT rod_cislo,
       meno,
       priezvisko
FROM os_udaje
INTO TEMP r1;
```

```
SELECT os_cislo,
       rod_cislo,
       rocnik,
       st_skupina
FROM student
INTO TEMP r2;
```

```
SELECT ou.rod_cislo,
       meno,
       priezvisko,
       os_cislo,
       st.rod_cislo,
       rocnik,
       st_skupina
FROM os_udaje ou,
     JOIN student st
     ON(ou.rod_cislo=st.
        rod_cislo);
```

```
SELECT r1.*, r2.* FROM
r1 JOINr2
ON(r1.rod_cislo =
r2.rod_cislo);
```

| r1.rod_cislo | Meno | priezvisko | os_cislo | r2.rod_cislo | rocnik | st_skupina |
|--------------|-----------|-------------|----------|--------------|--------|------------|
| 801106/3456 | Peter | Novák | 1512 | 801106/3456 | 1 | 5Z012 |
| 800312/7845 | Stanislav | Steinmüller | 1469 | 800312/7845 | 2 | 5Z021 |
| 810514/5341 | Branislav | Baláž | 1567 | 810514/5341 | 1 | 5Z013 |
| 781015/4431 | Peter | Kapustný | 1319 | 781015/4431 | 3 | 5ZA31 |
| 800407/3522 | Marek | Đurica | 1555 | 800407/3522 | 2 | 5Z022 |
| 791229/5431 | Martin | Kľúčiar | 1402 | 791229/5431 | 2 | 5Z023 |
| 771124/3578 | Lukáš | Satrapa | 1096 | 771124/3578 | 2 | 5Z023 |
| 771203/5472 | Ján | Krnáč | 1103 | 771203/5472 | 4 | 5ZI41 |
| 790310/2145 | Juraj | Papún | 1333 | 790310/2145 | 3 | 5ZA32 |
| 791225/7452 | Rastislav | Kontroš | 1448 | 791225/7452 | 1 | 5P011 |
| ... | | | | | | |
| 755022/8569 | Erika | Lipovská | 807 | 755022/8569 | 1 | 5Z013 |

SQL 99

SELECT

```
ou.rod_cislo, meno, priezvisko,  
os_cislo, st.rod_cislo, rocnik,  
st_skupina
```

```
FROM os_udaje as ou,
```

```
INNER JOIN student as st
```

```
ON (ou.rod_cislo=st.rod_cislo);
```

1.3 Ďalšie varianty operácie spojenia

V literatúre sa môžeme stretnúť s ďalšími operáciami relačnej algebry, ktoré sú rozšírením základných operácií relačnej algebry a sú to:

- a) **prirodzené spojenie**
- b) **theta spojenie**
- c) **equi spojenie**
- d) **inequi spojenie**
- e) **externé spojenie**
- f) **polospojenie (semi spojenie)**

1.3.1 PRIRODZENÉ SPOJENIE - NATURAL JOIN

Definícia – Prirodzené spojenie - Natural join

Operácia NATURAL JOIN (prirodzené spojenie) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$, ktoré majú spoločnú množinu atribútov X , tretiu reláciu

$R_3(X, A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ takú, že

ak $t_1 \in R_1$ a ak $t_2 \in R_2$ a pre hodnoty atribútov X platí

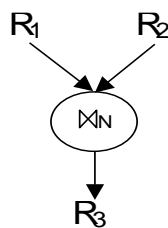
$$t_1.X = t_2.X$$

potom n-tica $t \in R_3$ má atribúty $t = t_1.X, t_1.A_1, t_1.A_2, \dots, t_1.A_n, t_2.B_1, t_2.B_2, \dots, t_2.B_m$, pričom atribúty s rovnakými menami *sa neopakujú*.

♦

Označenie

a) grafické



b) matematické

$$R_3 = R_1 \bowtie_N R_2$$

$$R_3 = R_1 \text{ JOIN}_N R_2$$

$$R_3 = \text{JOIN}(R_1, R_2, N)$$

□ *Príklad 1.15 – Prirodzené spojenie*

Vypíšte pre každého študenta nasledovné údaje:

rodné číslo, meno, priezvisko, osobné číslo, ročník, študijná skupina

(Spravte teda to isté, čo v príklade 1.11, ale s potlačením duplicity stĺpca rodné číslo).

a) pomocou operácií relačnej algebry

$$R_1 = \pi_{\text{rod_cislo}, \text{meno}, \text{priezvisko}}(\text{os_udaje})$$

$$R_2 = \pi_{\text{os_cislo}, \text{rod_cislo}, \text{rocnik}, \text{st_skupina}}(\text{student})$$

$$\text{SPOJENIE} = R_1 \bowtie_N R_2$$

b) pomocou SQL príkazov

```
SELECT rod_cislo, meno,
       priezvisko
FROM os_udaje
INTO TEMP r1;
```

```
SELECT os_cislo, rod_cislo,
       rocnik, st_skupina
FROM student
INTO TEMP r2;
```

```
SELECT r1.*, r2.os_cislo,
       r2.rocnik, r2.st_skupina
FROM r1 JOIN r2
ON
   r1.rod_cislo = r2.rod_cislo;
```

```
SELECT ou.rod_cislo, meno,
       priezvisko, os_cislo, rocnik,
       st_skupina
FROM os_udaje ou JOIN student st
ON ou.rod_cislo=st.rod_cislo;
```

| rod_cislo | meno | priezvisko | os_cislo | rocnik | st_skupina |
|-------------|-----------|-------------|----------|--------|------------|
| 801106/3456 | Peter | Novák | 1512 | 1 | 5Z012 |
| 800312/7845 | Stanislav | Steinmüller | 1469 | 2 | 5Z021 |
| 790907/1259 | János | Tóth | 1414 | 2 | 5Z021 |
| ... | | | | | |
| 755022/8569 | Erika | Lipovská | 807 | 1 | 5Z013 |

SQL99

```
SELECT ou.rod_cislo, meno,  
       priezvisko, os_cislo, rocnik,  
       st_skupina  
FROM os_udaje as ou  
     NATURAL INNER JOIN student as st  
ON (ou.rod_cislo=st.rod_cislo);
```

Špeciálny prípad

**Ak relácie R a S sú UNION
kompatibilné tak:**

$$R \cup S = R \bowtie S$$

1.3.2 THETA JOIN

Definícia –Theta join

Operácia THETA JOIN (θ - spojenie) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$, ktoré majú spoločnú množinu atribútov X tretiu reláciu $R_3(X, A_1, A_2, \dots, A_n, X, B_1, B_2, \dots, B_m)$ takú, že

ak $t_1 \in R_1$ a ak $t_2 \in R_2$ a pre hodnoty atribútov X platí

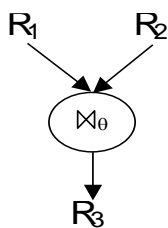
$$t_1.X \theta t_2.X$$

potom n-tica $t \in R_3$ má atribúty $t = t_1.X, t_1.A_1, t_1.A_2, \dots, t_1.A_n, t_2.X, t_2.B_1, t_2.B_2, \dots, t_2.B_m$, pričom *operátor* θ nadobúda hodnotu z množiny relačných operátorov $\{=, <, >, <=, >=, \neq\}$.

♦

Označenie

a) grafické



b) matematické

$$R_3 = R_1 \bowtie_{\theta} R_2$$

$$R_3 = R_1 \text{ JOIN}_{\theta} R_2$$

$$R_3 = \text{JOIN}(R_1, R_2, \theta)$$

□ *Príklad 1.16 [Codd90]*

Predpokladajme, že máme reláciu
Výrobok (cis_vyrobku, nazov, mnozstvo)
a reláciu
Objednávka (cis_odberatela, cis_vyrobku, poz_mnostvo).

V prípade, že požiadavka je formulovaná
nasledovne:

```
SELECT vyrobok.*, objednavka.*  
FROM vyrobok JOIN objednavka  
ON  
vyrobok.mnozstvo < objednavka.poz_mnozstvo
```

Výsledná relácia bude obsahovať atribúty z oboch
relácií a tie n-tice, kde požadované množstvo je
väčšie ako množstvo vyrobených výrobkov.

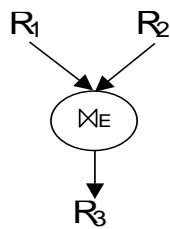
1.3.3 EQUI JOIN

Definícia – Equi join

Operácia EQUI JOIN je takou *operáciou θ -spojenia*, kde operátor θ nadobúda hodnotu relačného operátora $=$. ♦

Označenie

a) grafické



b) matematické

$$R_3 = R_1 \bowtie_E R_2$$

$$R_3 = R_1 \text{ JOIN}_E R_2$$

$$R_3 = \text{JOIN}(R_1, R_2, E)$$

1.3.4 INEQUI JOIN

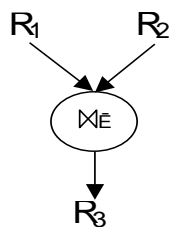
Definícia – Inequi join

Operácia INEQUI JOIN je takou *operáciou Θ -spojenia*, kde operátor Θ nadobúda hodnotu z množiny relačných operátorov $\{<, >, <=, >=, \neq\}$.

♦

Označenie

a) grafické



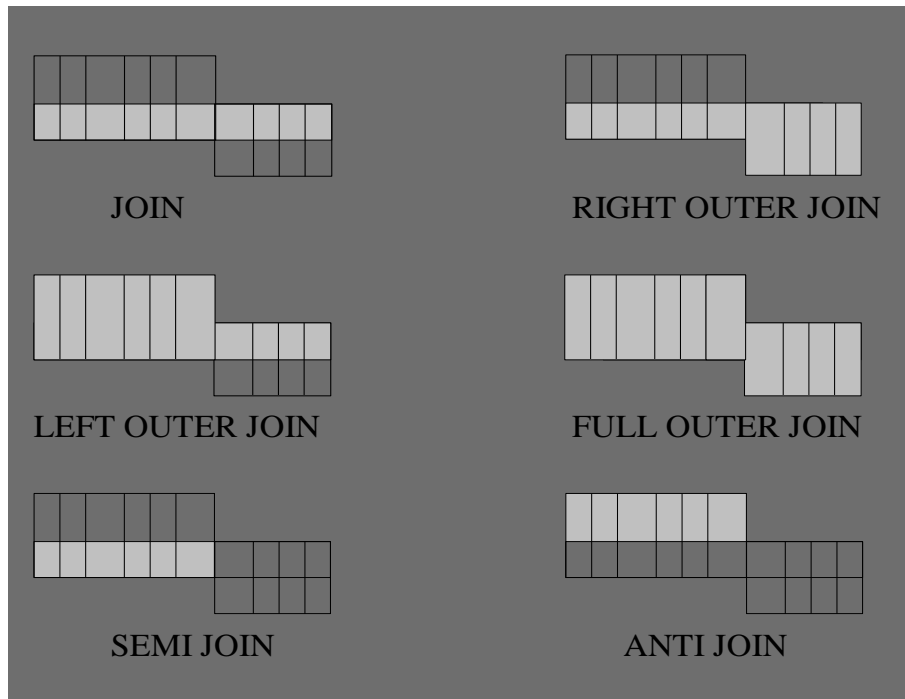
b) matematické

$$R_3 = R_1 \bowtie_E R_2$$

$$R_3 = R_1 \text{ JOIN}_E R_2$$

$$R_3 = \text{JOIN}(R_1, R_2, \bar{E})$$

1.3.5 EXTERNAL JOIN



Definícia – External join - FULL

Operácia **EXTERNAL JOIN – FULL** (Vonkajšie spojenie - úplné) vytvorí z relácie $\mathbf{R_1(X,A_1,A_2,...,A_n)}$ a z relácie $\mathbf{R_2(X,B_1,B_2, ...,B_m)}$, ktoré majú spoločnú množinu atribútov \mathbf{X} tretiu reláciu

$\mathbf{R_3(X,A_1,A_2,...,A_a,X,B_1,B_2,...,B_m)}$ takú, že

ak $\mathbf{t_1 \in R_1}$ a ak $\mathbf{t_2 \in R_2}$

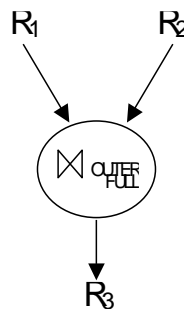
potom n-tica $\mathbf{t \in R_3}$ má atribúty $\mathbf{t = t_1.X, t_1.A_1, t_1.A_2, ..., t_1.A_a, t_2.X, t_2.B_1, t_2.B_2, ..., t_2.B_m}$. a ak pre hodnoty atribútov \mathbf{X} platí:

- $\mathbf{t_1.X = t_2.X}$
- alebo hodnota $\mathbf{t_1.X \notin \{hodnôt\ t_2.X\}}$
- alebo hodnota $\mathbf{t_2.X \notin \{hodnôt\ t_1.X\}}$

potom n-tica $\mathbf{t \in R_3}$ nadobúda **NULL** hodnoty pre chýbajúce atribúty n-tice \mathbf{t} .

♦

Označenie – grafické



□ *Príklad 1.17 – External join FULL*

Pomocou príkazov SQL realizujte úplné vonkajšie spojenie relácií r1 a r2 definovaných nasledovnými SELECT-ami:

```
CREATE TABLE R1 AS
SELECT UNIQUE os_cislo , cis_predmet
FROM zap_predmety;
CREATE TABLE R2 AS
SELECT cis_predmet, nazov
FROM predmet;
```

SQL99

```
SELECT r1.*, r2.*
FROM r1 FULL OUTER JOIN r2
ON r1.cis_predmet = r2.cis_predmet
```

Oracle

```
SELECT r1.*, r2.*
FROM r1+, r2
WHERE r1.cis_predmet = r2.cis_predmet
UNION
SELECT r1.*, r2.*
FROM r1, r2+
WHERE r1.cis_predmet = r2.cis_predmet;
```

□

Riešenie

```
SELECT r1.*, r2.*
FROM r1, OUTER r2
WHERE r1.cis_predmet=r2.cis_predmet
UNION
SELECT r1.*, r2.*
FROM OUTER r1, r2
WHERE r1.cis_predmet=r2.cis_predmet;
```

| os_cislo | r1.cis_predmet | r2.cis_predmet | nazov |
|----------|----------------|----------------|-------------------------------|
| | | A502 | C-jazyk |
| | | A506 | Časti elektronických systémov |
| | | A601 | Matematické programovanie |
| ... | ... | ... | ... |
| 807 | P202 | P202 | Matematická analýza 2 |
| 807 | P203 | | |
| 807 | P211 | P211 | Základy informatiky 2 |
| 807 | V101 | V101 | Praktikum z programovania 1 |
| ... | ... | ... | ... |
| 1448 | P111 | P111 | Základy informatiky 1 |
| 1448 | P202 | P202 | Matematická analýza 2 |
| 1448 | P203 | | |
| 1469 | P203 | | |
| 1469 | P301 | P301 | Pravdepodobnosť |
| 1469 | P303 | P303 | Matematická analýza 3 |
| 1512 | P102 | P102 | Algebra |
| 1512 | P103 | P103 | Matematická analýza 1 |
| 1512 | P111 | P111 | Základy informatiky 1 |
| 1545 | P103 | P103 | Matematická analýza 1 |
| 1545 | P202 | P202 | Matematická analýza 2 |
| 1545 | P203 | | |
| 1555 | P202 | P202 | Matematická analýza 2 |
| 1555 | P203 | | |
| 1555 | P301 | P301 | Pravdepodobnosť |
| 1555 | P303 | P303 | Matematická analýza 3 |
| 1559 | P201 | | |
| 1559 | P203 | | |
| ... | ... | ... | ... |

Definícia – External join - LEFT

Operácia EXTERNAL JOIN – LEFT (vonkajšie spojenie - ľavé) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$, ktoré majú spoločnú množinu atribútov X tretiu reláciu

$R_3(X, A_1, A_2, \dots, A_n, X, B_1, B_2, \dots, B_m)$ takú, že

ak $t_1 \in R_1$ a ak $t_2 \in R_2$

potom n-tica $t \in R_3$ má atribúty

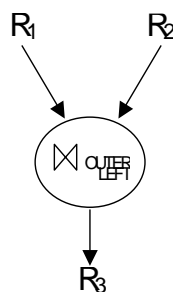
$t = t_1.X, t_1.A_1, t_1.A_2, \dots, t_1.A_n, t_2.X, t_2.B_1, t_2.B_2, \dots, t_2.B_m$.

a ak pre hodnoty atribútov X platí:

- $t_1.X = t_2.X$
- alebo hodnota $t_2.X \notin \{\text{hodnôt } t_1.X\}$

potom n-tica $t \in R_3$ nadobúda NULL hodnoty pre chýbajúce atribúty n-tice t .

Označenie – grafické



- *Príklad 1.18 – Externé spojenie LEFT*
 Pomocou príkazov SQL realizujte ľavé vonkajšie spojenie relácií r1 a r2 definovaných v predchádzajúcom príklade 1.17:

SQL99

```
SELECT r1.*,r2.*
FROM r1 LEFT OUTER JOIN r2
ON r1.cis_predmet = r2.cis_predmet;
```

ORACLE

```
SELECT r1.*,r2.*
FROM r1+, r2
WHERE r1.cis_predmet = r2.cis_predmet;
```

INFORMIX

```
SELECT R1.*,R2.*
FROM OUTER R1, R2
WHERE R1.CIS_PREDMET=R2.CIS_PREDMET;
```

| os_cislo | r1.cis_predmet | r2.cis_predmet | nazov |
|----------|----------------|----------------|-----------------------------|
| 807 | P202 | P202 | Matematická analýza 2 |
| 807 | P203 | | |
| 807 | P211 | P211 | Základy informatiky 2 |
| 807 | V101 | V101 | Praktikum z programovania 1 |
| ... | ... | ... | ... |
| 1545 | P103 | P103 | Matematická analýza 1 |
| 1545 | P202 | P202 | Matematická analýza 2 |
| 1545 | P203 | | |
| 1555 | P202 | P202 | Matematická analýza 2 |
| 1555 | P203 | | |
| 1555 | P301 | P301 | Pravdepodobnosť |
| 1555 | P303 | P303 | Matematická analýza 3 |
| 1559 | P201 | | |
| 1559 | P203 | | |
| 1567 | P202 | P202 | Matematická analýza 2 |
| 1567 | P203 | | |
| 1567 | V201 | V201 | Praktikum z programovania 2 |

Definícia – External join - RIGHT

Operácia **EXTERNAL JOIN - RIGHT**(vonkajšie spojenie - pravé) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$, ktoré majú spoločnú množinu atribútov X tretiu reláciu

$R_3(X, A_1, A_2, \dots, A_n, X, B_1, B_2, \dots, B_m)$ takú, že

ak $t_1 \in R_1$ a ak $t_2 \in R_2$

potom n-tica $t \in R_3$ má atribúty

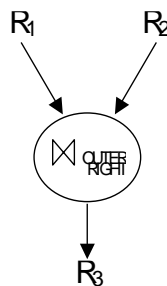
$t = t_1.X, t_1.A_1, t_1.A_2, \dots, t_1.A_n, t_2.X, t_2.B_1, t_2.B_2, \dots, t_2.B_m$.

a ak pre hodnoty atribútov X platí:

- $t_1.X = t_2.X$
- alebo hodnota $t_1.X \notin \{\text{hodnota } t_2.X\}$

potom n-tica $t \in R_3$ nadobúda **NULL** hodnoty pre chýbajúce atribúty n-tice t .

Označenie – grafické



□

♦

□ *Príklad 1.19 – OUTER RIGHT*

Pomocou príkazov SQL realizujte úplné vonkajšie spojenie relácií r1 a r2 definovaných v príklade 1.17:

SQL99

```
SELECT r1.*,r2.*
FROM r1 RIGHT OUTER JOIN r2
ON r1.cis_predmet = r2.cis_predmet;
```

ORACLE

```
SELECT r1.*,r2.*
FROM r1, r2+
WHERE r1.cis_predmet = r2.cis_predmet;
```

INFORMIX

```
SELECT r1.*,r2.*
FROM r1, OUTER r2
WHERE r1.cis_predmet=r2.cis_predmet;
```

| Os_cislo | r1.cis_predmet | r2.cis_predmet | nazov |
|----------|----------------|----------------|-------------------------------|
| 945 | P111 | P111 | Základy informatiky 1 |
| 1381 | P111 | P111 | Základy informatiky 1 |
| 1414 | P111 | P111 | Základy informatiky 1 |
| 1448 | P111 | P111 | Základy informatiky 1 |
| 1512 | P111 | P111 | Základy informatiky 1 |
| ... | ... | ... | ... |
| 1333 | P602 | P602 | Číslicové počítače |
| 1381 | P609 | P609 | Manažment |
| | | A601 | Matematické programovanie |
| 1333 | A602 | A602 | Databázové systémy |
| 1381 | A602 | A602 | Databázové systémy |
| 1612 | A602 | A602 | Databázové systémy |
| | | V502 | Právo 1 |
| | | V601 | Právo 2 |
| ... | ... | ... | ... |
| | | A702 | Operačné systémy* |
| | | A709 | %Marketing |
| 1103 | A806 | A806 | Riadenie počítačom |
| | | V719 | Základy programovania vo Win. |
| ... | | | |
| 945 | A904 | A904 | Prognostika |

1.3.6 SEMI JOIN

Definícia – Semi join

Operácia SEMI JOIN (polospojenie) vytvorí z relácie $R_1(X, A_1, A_2, \dots, A_n)$ a z relácie $R_2(X, B_1, B_2, \dots, B_m)$ tretiu reláciu $R_3(X, A_1, A_2, \dots, A_n)$ takú,

že $t_1 \in R_1$ a $t_2 \in R_2$

ak pre n-ticu $t_1 \in R_1$ existuje spojenie *aspoň* s jednou n-ticou $t_2 \in R_2$, potom $t_1 \in R_3$.

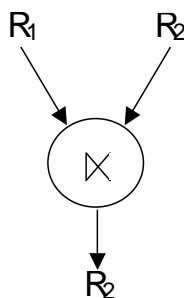
♦

Poznámka

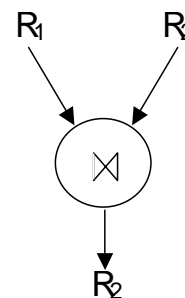
Základná operácia Polospojenia predpokladá, že výsledná relácia bude obsahovať n-tice len z prvej z relácií, ktoré sú operandami. Ale podobne ako pri Vonkajšom spojení aj pri polospojení rozlišujeme pravé, alebo ľavé polospojenie, z čoho vyplýva, že výsledná relácia je tvorená len výskytmi n-tíc tej relácie, ktorá je vo výraze umiestnená vpravo (pravé polospojenie), resp. vľavo (ľavé polospojenie). Z toho vyplýva, že základná operácia Polospojenia je vlastne definovaná ako Ľavé polospojenie.

Označenie – grafické

a) Semi join – LEFT



b) Semi join – RIGHT



□ Príklad 1.20 – Ľavé polospojenie

```
SELECT UNIQUE os_udaje.* FROM
os_udaje JOIN student
ON
os_udaje.rod_cislo=student.rod_cislo
```

□ Príklad 1.21 – Pravé polospojenie

```
SELECT UNIQUE student.*
FROM os_udaje JOIN student
ON
os_udaje.rod_cislo=student.rod_cislo
```


1.4 Ďalšie operácie relačnej algebry

V literatúre sa môžeme stretnúť s ďalšími operáciami relačnej algebry, ktoré sú rozšírením základných operácií relačnej algebry a sú to:

a) **Doplnok (Complement)**

b) **Rozdelenie (Split)**

1.4.1 COMPLEMENT

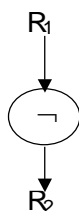
Definícia - Complement

Operácia COMPLEMENT (doplnok) vytvorí z relácie $R_1(A_1, A_2, \dots, A_n)$ reláciu $R_2(A_1, A_2, \dots, A_n)$ takú, že $t_2 \in R_2$ obsahuje všetky n -tice, ktoré patria do Kartézskeho súčinu hodnôt domén atribútov relácie R_1 a $t_2 \notin R_1$.

♦

Označenie

a) grafické



b) matematické

$$R_2 = \text{NOT } (R_1)$$

$$R_2 = \text{COMP } (R_1)$$

$$R_2 = \neg R_1$$

□ *Príklad 1.22*

Nech existuje relácia $R(A,B)$, pričom doména atribútu $A = \{1,2,3\}$ a doména atribútu $B = \{x,y,z,w\}$ a nech relácia R má nasledovné n-tice:

| A | B |
|---|---|
| 1 | x |
| 1 | y |
| 1 | z |
| 2 | w |

Potom výsledok operácie doplnok je relácia, ktorá má nasledovné n-tice:

| A | B |
|---|---|
| 1 | w |
| 2 | x |
| 2 | y |
| 2 | z |
| 3 | x |
| 3 | y |
| 3 | z |
| 3 | w |

1.4.2 SPLIT

Definícia - Split

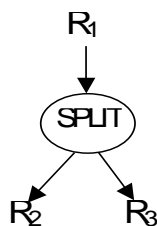
Operácia split vytvorí z relácie $R_1(A_1, A_2 \dots A_n)$ dve relácie $R_2(A_1, A_2 \dots A_n)$ a $R_3(A_1, A_2 \dots A_n)$ také, že pre každé $t \in R_1$ platí:

- t patrí do práve jednej z relácií R_2 a R_3
- ak pre t *je splnená* podmienka C , potom $t \in R_2$
- ak pre t *nie je splnená* podmienka C , potom $t \in R_3$

♦

Označenie

a) grafické



b) matematické

$$R_2 = \sigma_C R_1$$

$$R_3 = \sigma_{\neg C} R_1$$

Relačná algebra a kalkul

□ *Príklad 1.14 - Split*

Vytvorte reláciu, ktorá obsahuje zoznam mien, priezvisk a rodných čísel všetkých osôb z relácie osobné údaje. Potom ju pomocou operácie split rozdeľte na dve relácie – muži a ženy.

a) pomocou operácií relačnej algebry

```
R1 =  $\pi_{\text{meno, priezvisko, rod\_cislo}}(\text{os\_udaje})$   
R2 =  $\sigma_{\text{rod\_cislo}[3,3]>4}$  R1 //ženy  
R3 =  $\sigma_{\neg \text{rod\_cislo}[3,3]>4}$  R1 //muži
```

b) pomocou SQL príkazov

```
CREATE TABLE R1 AS  
SELECT meno, priezvisko, rod_cislo FROM os_udaje;  
  
CREATE TABLE R2 AS  
SELECT * FROM r1  
WHERE rod_cislo[3,3]>4 ; //ženy
```

| meno | priezvisko | rod_cislo |
|------------|------------|-------------|
| Stanislava | Slámová | 796123/5471 |
| Erika | Lipovská | 755022/8569 |

```
SELECT * FROM r1  
WHERE rod_cislo[3,3]<=4; //muži
```

| meno | priezvisko | rod_cislo |
|-----------|-------------|-------------|
| Peter | Novák | 801106/3456 |
| Stanislav | Steinmüller | 800312/7845 |
| János | Tóth | 790907/1259 |
| Marek | Rátroch | 810130/3695 |
| Bohuslav | Biely | 781201/1248 |
| Branislav | Baláž | 810514/5341 |
| Peter | Kapustný | 781015/4431 |
| Marek | Đurica | 800407/3522 |
| Martin | Kľúčiar | 791229/5431 |
| ... | | |
| Ján | Krnáč | 771203/5472 |