

Databázové systémy

Přednáška 2. Množinové operace a restrikce

Jan Laštovička



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

1 Množinové operace

2 Projekce

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

(**TABLE** R)

... relační výraz

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

(**TABLE** R)

... relační výraz

v : relační výraz

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

(**TABLE** R)

... relační výraz

v : relační výraz

v ;

... vytištění hodnoty v

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

```
( TABLE  $R$  )
```

... relační výraz

v : relační výraz

```
 $v$ ;
```

... vytištění hodnoty v

Například:

```
( TABLE child );
```

Výrazu, jehož hodnota je relace, říkáme *relační výraz*.

R : relační proměnná

```
( TABLE  $R$  )
```

... relační výraz

v : relační výraz

```
 $v$ ;
```

... vytištění hodnoty v

Například:

```
( TABLE child );
```

Vynechání nejvíce vnějších závorek:

```
TABLE child;
```

Typ relace a typ relačního výrazu



- `typ relace` = záhlaví relace

Typ relace a typ relačního výrazu



- `typ relace` = záhlaví relace
- relace stejného typu

- `typ relace` = záhlaví relace
- relace stejného typu

- hodnota relačního výrazu se může změnit (proměnné)

- `typ relace` = záhlaví relace
- relace stejného typu

- hodnota relačního výrazu se může změnit (proměnné)
- typ hodnoty relačního výrazu je stále stejný

- **typ relace** = záhlaví relace
- relace stejného typu

- hodnota relačního výrazu se může změnit (proměnné)
- typ hodnoty relačního výrazu je stále stejný
- **typ relačního výrazu** = typ jeho hodnoty

- **typ relace** = záhlaví relace
- relace stejného typu

- hodnota relačního výrazu se může změnit (proměnné)
- typ hodnoty relačního výrazu je stále stejný
- **typ relačního výrazu** = typ jeho hodnoty
- relační výrazy stejného typu



Operace s relacemi



Operace:

- 1 vstup: jedna i více relací
- 2 výstup: jedna relace

Operace:

- 1 vstup: jedna i více relací
 - 2 výstup: jedna relace
- omezení na vstupní relace

Operace:

- 1 vstup: jedna i více relací
 - 2 výstup: jedna relace
- omezení na vstupní relace
 - vstupy i výstup operace jsou relace

Operace:

- 1 vstup: jedna i více relací
 - 2 výstup: jedna relace
- omezení na vstupní relace
 - vstupy i výstup operace jsou relace
 - operace lze skládat

Operace:

1 vstup: jedna i více relací

2 výstup: jedna relace

- omezení na vstupní relace
- vstupy i výstup operace jsou relace
- operace lze skládat

Relační proměnné uchovávají informace ve formě relace.

Operace:

- 1 vstup: jedna i více relací
 - 2 výstup: jedna relace
- omezení na vstupní relace
 - vstupy i výstup operace jsou relace
 - operace lze skládat

Relační proměnné uchovávají informace ve formě relace.

Dotazy vyjadřujeme skládáním operací s relacemi.

Operace:

- 1 vstup: jedna i více relací
 - 2 výstup: jedna relace
- omezení na vstupní relace
 - vstupy i výstup operace jsou relace
 - operace lze skládat

Relační proměnné uchovávají informace ve formě relace.

Dotazy vyjadřujeme skládáním operací s relacemi.

Výsledek dotazu je relace.

Sjednocení



r_1, r_2 : relace stejného typu

r_1, r_2 : relace stejného typu

Výstupem **sjednocení** r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinové sjednocení těl r_1 a r_2

r_1, r_2 : relace stejného typu

Výstupem **sjednocení** r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinové sjednocení těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

r_1, r_2 : relace stejného typu

Výstupem **sjednocení** r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinové sjednocení těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

sjednocení r_1 a r_2 :

name	age
Anna	3
Bert	4
Cyril	4
Daniela	5

v_1, v_2 : relační výrazy stejného typu

v_1, v_2 : relační výrazy stejného typu

(v_1 UNION v_2)

v_1, v_2 : relační výrazy stejného typu

(v_1 UNION v_2)

Například:

#

v_1, v_2 : relační výrazy stejného typu

```
(  $v_1$  UNION  $v_2$  )
```

Například:

```
# ( TABLE child1 ) UNION ( TABLE child2 );
```

v_1, v_2 : relační výrazy stejného typu

(v_1 UNION v_2)

Například:

```
# ( TABLE child1 ) UNION ( TABLE child2 );
```

name		age
Daniela		5
Cyril		4
Bert		4
Anna		3

(4 rows)

r_1, r_2 : relace stejného typu

r_1, r_2 : relace stejného typu

Výstupem průniku r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový průnik těl r_1 a r_2

r_1, r_2 : relace stejného typu

Výstupem průniku r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový průnik těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

r_1, r_2 : relace stejného typu

Výstupem průniku r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový průnik těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

průnik r_1 a r_2 :

name	age
Bert	4
Cyril	4

v_1, v_2 : relační výrazy stejného typu

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ INTERSECT } v_2)$

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ INTERSECT } v_2)$

Například:

#

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ INTERSECT } v_2)$

Například:

```
# ( TABLE child1 ) INTERSECT ( TABLE child2 );
```

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ INTERSECT } v_2)$

Například:

```
# ( TABLE child1 ) INTERSECT ( TABLE child2 );
```

name		age
Bert		4
Cyril		4

(2 rows)

r_1, r_2 : relace stejného typu

r_1, r_2 : relace stejného typu

Výstupem rozdílu r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový rozdíl těl r_1 a r_2

r_1, r_2 : relace stejného typu

Výstupem rozdílu r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový rozdíl těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

r_1, r_2 : relace stejného typu

Výstupem **rozdílu** r_1 a r_2 je relace r :

- 1 záhlaví r je stejné jako záhlaví r_1 a r_2
- 2 tělo r je množinový rozdíl těl r_1 a r_2

Příklad:

r_1 :

name	age
Anna	3
Bert	4
Cyril	4

r_2 :

name	age
Bert	4
Cyril	4
Daniela	5

rozdíl r_1 a r_2 :

name	age
Anna	3

v_1, v_2 : relační výrazy stejného typu

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ EXCEPT } v_2)$

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ EXCEPT } v_2)$

Například:

#

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ EXCEPT } v_2)$

Například:

```
# ( TABLE child1 ) EXCEPT ( TABLE child2 );
```

v_1, v_2 : relační výrazy stejného typu

$(v_1 \text{ EXCEPT } v_2)$

Například:

```
# ( TABLE child1 ) EXCEPT ( TABLE child2 );
```

```
name | age
-----+-----
Anna |    3
(1 row)
```

child1:

name	age
Anna	3
Bert	4
Cyril	4

child2:

name	age
Bert	4
Cyril	4
Daniela	5

Které děti jsou v právě jedné z relací?

child1:

name	age
Anna	3
Bert	4
Cyril	4

child2:

name	age
Bert	4
Cyril	4
Daniela	5

Které děti jsou v právě jedné z relací?

#

child1:

name	age
Anna	3
Bert	4
Cyril	4

child2:

name	age
Bert	4
Cyril	4
Daniela	5

Které děti jsou v právě jedné z relací?

```
# ( ( TABLE child1 ) EXCEPT ( TABLE child2 ) )  
  UNION  
  ( ( TABLE child2 ) EXCEPT ( TABLE child1 ) );
```

child1:

name	age
Anna	3
Bert	4
Cyril	4

child2:

name	age
Bert	4
Cyril	4
Daniela	5

Které děti jsou v právě jedné z relací?

```
# ( ( TABLE child1 ) EXCEPT ( TABLE child2 ) )  
  UNION  
  ( ( TABLE child2 ) EXCEPT ( TABLE child1 ) );
```

```
  name  | age  
-----+-----  
Anna    |   3  
Daniela |   5  
(2 rows)
```

1 Množinové operace

2 Projekce



r : relace

r : relace

r' :

- relace
- záhlaví = záhlaví r
- tělo podmnožina těla r

r : relace

r' :

- relace
- záhlaví = záhlaví r
- tělo podmnožina těla r

Podmnožinu těla r zadáme podmínkou.

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j \leq n$

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j \leq n$
- v : hodnota typu T_j

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j \leq n$
- v : hodnota typu T_j

($A_j = v$)

... podmínka nad A_1, \dots, A_n

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j \leq n$
- v : hodnota typu T_j

($A_j = v$)
... podmínka nad A_1, \dots, A_n

Příklad:

(name = 'Anna')
(age = 3)
... podmínky nad name a age

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j \leq n$
- v : hodnota typu T_j

```
( A_j = v )  
... podmínka nad  $A_1, \dots, A_n$ 
```

Příklad:

```
( name = 'Anna' )  
( age = 3 )  
... podmínky nad name a age
```

Nejvíce vnější závorky vynecháváme:

```
name = 'Anna'  
age = 3
```



- t : n -tice nad A_1, \dots, A_n

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ je v t splněna, pokud t atributu A_j přiřazuje v .

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ *je v t splněna*, pokud t atributu A_j přiřazuje v .

n -tice:

name	age
Anna	3

Splněné?

- 1 $(\text{name} = \text{'Anna'})$
- 2 $(\text{age} = 4)$
- 3 $(\text{name} = \text{'Bert'})$
- 4 $(\text{age} = 3)$

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ *je v t splněna*, pokud t atributu A_j přiřazuje v .

n -tice:

name	age
Anna	3

Splněné?

- 1 $(\text{name} = \text{'Anna'})$... ano
- 2 $(\text{age} = 4)$
- 3 $(\text{name} = \text{'Bert'})$
- 4 $(\text{age} = 3)$

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ *je v t splněna*, pokud t atributu A_j přiřazuje v .

n -tice:

name	age
Anna	3

Splněné?

- 1 $(\text{name} = \text{'Anna'})$... ano
- 2 $(\text{age} = 4)$... ne
- 3 $(\text{name} = \text{'Bert'})$
- 4 $(\text{age} = 3)$

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ *je v t splněna*, pokud t atributu A_j přiřazuje v .

n -tice:

name	age
Anna	3

Splněné?

- 1 $(\text{name} = \text{'Anna'})$... ano
- 2 $(\text{age} = 4)$... ne
- 3 $(\text{name} = \text{'Bert'})$... ne
- 4 $(\text{age} = 3)$

- t : n -tice nad A_1, \dots, A_n
- $(A_j = v)$: podmínka nad A_1, \dots, A_n

Podmínka $(A_j = v)$ *je v t splněna*, pokud t atributu A_j přiřazuje v .

n -tice:

name	age
Anna	3

Splněné?

- 1 $(\text{name} = \text{'Anna'})$... ano
- 2 $(\text{age} = 4)$... ne
- 3 $(\text{name} = \text{'Bert'})$... ne
- 4 $(\text{age} = 3)$... ano



r : relace nad A_1, \dots, A_n

r : relace nad A_1, \dots, A_n

c : podmínka nad A_1, \dots, A_n

r : relace nad A_1, \dots, A_n

c : podmínka nad A_1, \dots, A_n

r' :

- záhlaví $r' = \text{záhlaví } r$
- tělo obsahuje právě ty n -tice z těla r , které splňují podmínku c
- restrikce r vzhledem k c

r : relace nad A_1, \dots, A_n

c : podmínka nad A_1, \dots, A_n

r' :

- záhlaví $r' = \text{záhlaví } r$
- tělo obsahuje právě ty n -tice z těla r , které splňují podmínku c
- restrikce r vzhledem k c

Příklad:

name	age	= restrikce	name	age	vzhledem k $\text{age} = 4$
Bert	4		Anna	3	
Cyril	4		Bert	4	
			Cyril	4	

- $v \dots$ relační výraz
- hodnota $v =$ relace r nad A_1, \dots, A_n

- v ... relační výraz
- hodnota $v = \text{relace } r \text{ nad } A_1, \dots, A_n$
- R je jméno relace
- c je podmínka nad A_1, \dots, A_n

- $v \dots$ relační výraz
- hodnota $v =$ relace r nad A_1, \dots, A_n
- R je jméno relace
- c je podmínka nad A_1, \dots, A_n

restrikce relace r vzhledem k c :

```
( SELECT *  
  FROM    $v$  AS  $R$   
  WHERE   $c$  )
```

(R zatím nemá žádný význam)

#

```
# TABLE child;
```

```
# TABLE child;
```

name		age
Anna		3
Bert		4
Cyril		4

(3 rows)

```
#
```

```
# TABLE child;
```

name		age
Anna		3
Bert		4
Cyril		4

(3 rows)

```
# SELECT *  
FROM ( TABLE child ) AS t  
WHERE age = 4;
```



```
# TABLE child;
```

name		age
-----+-----		
Anna		3
Bert		4
Cyril		4

(3 rows)

```
# SELECT *  
  FROM ( TABLE child ) AS t  
  WHERE age = 4;
```

name		age
-----+-----		
Bert		4
Cyril		4

(2 rows)

Děti:

#

Děti:

```
# TABLE child;
```

Děti:

TABLE child;

name		age		street
Anna		3		Kosinova
Bert		4		Mahlerova
Cyril		4		Kosinova

(3 rows)

Děti:

TABLE child;

name		age		street
Anna		3		Kosinova
Bert		4		Mahlerova
Cyril		4		Kosinova

(3 rows)

Jak zjistit, které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?

Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



#

Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



```
# ( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE age = 4 )  
    INTERSECT  
( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE street = 'Kosinova' );
```

Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



```
# ( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE age = 4 )  
    INTERSECT  
( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE street = 'Kosinova' );
```

name	age	street
Cyril	4	Kosinova

(1 row)

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



#

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



```
# ( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE age = 3 )  
UNION  
( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE street = 'Mahlerova' );
```

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



```
# ( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE age = 3 )  
UNION  
( SELECT *  
    FROM ( TABLE child ) AS t  
    WHERE street = 'Mahlerova' );
```

name	age	street
Bert	4	Mahlerova
Anna	3	Kosinova

(2 rows)

Kterým dětem nejsou tři roky?



#

Kterým dětem nejsou tři roky?



```
# ( TABLE child )  
  EXCEPT  
( SELECT *  
  FROM ( TABLE child ) AS t  
  WHERE age = 3 );
```

Kterým dětem nejsou tři roky?



```
# ( TABLE child )  
  EXCEPT  
( SELECT *  
  FROM ( TABLE child ) AS t  
  WHERE age = 3 );
```

name		age		street
Bert		4		Mahlerova
Cyril		4		Kosinova

(2 rows)

c_1, c_2 : podmínky nad A_1, \dots, A_n

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

t : n -tice nad A_1, \dots, A_n

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

t : n -tice nad A_1, \dots, A_n

Podmínka je splněna:

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

t : n -tice nad A_1, \dots, A_n

Podmínka je splněna:

- (c_1 AND c_2) ... obě podmínky v_1 a v_2 jsou v t splněny

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

t : n -tice nad A_1, \dots, A_n

Podmínka je splněna:

- (c_1 AND c_2) ... obě podmínky v_1 a v_2 jsou v t splněny
- (c_1 OR c_2) ... aspoň jedna z podmínek v_1 a v_2 je v t splněna

c_1, c_2 : podmínky nad A_1, \dots, A_n

(c_1 AND c_2)

(c_1 OR c_2)

(NOT c_1)

...podmínky nad A_1, \dots, A_n

t : n -tice nad A_1, \dots, A_n

Podmínka je splněna:

- (c_1 AND c_2) ... obě podmínky v_1 a v_2 jsou v t splněny
- (c_1 OR c_2) ... aspoň jedna z podmínek v_1 a v_2 je v t splněna
- (NOT c_1) ... c_1 není v t splněna

Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



#

Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



```
# SELECT *  
FROM ( TABLE child ) AS t  
WHERE ( age = 4 )  
AND ( street = 'Kosinova' );
```


Které děti jsou ve věku čtyř let a bydlí v ulici Kosinova?



```
# SELECT *  
  FROM ( TABLE child ) AS t  
 WHERE ( age = 4 )  
 AND   ( street = 'Kosinova' );
```

name	age	street
Cyril	4	Kosinova

(1 row)

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



#

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



```
# SELECT *  
FROM ( TABLE child ) AS t  
WHERE ( age = 3 )  
OR     ( street = 'Mahlerova' );
```

Které děti jsou ve věku tří let nebo bydlí v ulici Mahlerova?



```
# SELECT *  
FROM ( TABLE child ) AS t  
WHERE ( age = 3 )  
OR     ( street = 'Mahlerova' );
```

name	age	street
Anna	3	Kosinova
Bert	4	Mahlerova

(2 rows)

Kterým dětem nejsou tři roky?



#

Kterým dětem nejsou tři roky?



```
# SELECT *  
FROM ( TABLE child ) AS t  
WHERE NOT ( age = 3 );
```

Kterým dětem nejsou tři roky?



```
# SELECT *  
  FROM ( TABLE child ) AS t  
  WHERE NOT ( age = 3 );
```

name	age	street
Bert	4	Mahlerova
Cyril	4	Kosinova

(2 rows)

student	name	mathematics	informatics
	Anna	2	2
	Bert	1	3
	Cyril	3	3

Kdo má stejnou známku z matematiky a informatiky?

student	name	mathematics	informatics
	Anna	2	2
	Bert	1	3
	Cyril	3	3

Kdo má stejnou známku z matematiky a informatiky?

```

SELECT      *
FROM        ( TABLE student ) AS t
WHERE ( ( ( ( mathematics = 1 AND informatics = 1 )
OR         ( mathematics = 2 AND informatics = 2 ) )
OR         ( mathematics = 3 AND informatics = 3 ) )
OR         ( mathematics = 4 AND informatics = 4 ) )
OR         ( mathematics = 5 AND informatics = 5 )
    
```


- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j, k \leq n$
- T_j se rovná T_k (Atributy A_j a A_k mají stejný typ)

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j, k \leq n$
- T_j se rovná T_k (Atributy A_j a A_k mají stejný typ)

podmínka nad A_1, \dots, A_n :

$$(A_j = A_k)$$

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j, k \leq n$
- T_j se rovná T_k (Atributy A_j a A_k mají stejný typ)

podmínka nad A_1, \dots, A_n :

$$(A_j = A_k)$$

t : n -tice nad A_1, \dots, A_n

- $\{A_1, \dots, A_n\}$: záhlaví
- A_i je typu T_i ($1 \leq i \leq n$)
- $1 \leq j, k \leq n$
- T_j se rovná T_k (Atributy A_j a A_k mají stejný typ)

podmínka nad A_1, \dots, A_n :

$$(A_j = A_k)$$

t : n -tice nad A_1, \dots, A_n

$(A_j = A_k)$ je v n -tici t **splněna**

$\dots t$ přiřazuje stejnou hodnotu atributu A_j i atributu A_k

n -tice nad name, mathematics, informatics a physics:

name	mathematics	informatics	physics
Anna	2	2	1

n -tice nad name, mathematics, informatics a physics:

name	mathematics	informatics	physics
Anna	2	2	1

- (mathematics = informatics)
- (mathematics = physics)

n -tice nad name, mathematics, informatics a physics:

name	mathematics	informatics	physics
Anna	2	2	1

- $(\text{mathematics} = \text{informatics}) \dots$ je v n -tici splněna
- $(\text{mathematics} = \text{physics})$

n -tice nad name, mathematics, informatics a physics:

name	mathematics	informatics	physics
Anna	2	2	1

- $(\text{mathematics} = \text{informatics}) \dots$ je v n -tici splněna
- $(\text{mathematics} = \text{physics}) \dots$ není v n -tici splněna

#

```
# SELECT * FROM ( TABLE student ) AS t  
WHERE mathematics = informatics;
```

```
# SELECT * FROM ( TABLE student ) AS t
  WHERE mathematics = informatics;
```

name	mathematics	informatics
Anna	2	2
Cyril	3	3

(2 rows)