

Databázové systémy

Přednáška 4. Relační algebra

Jan Laštovička



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

- 1 Relační algebra
- 2 Relační výraz SELECT
- 3 Relační dělení

*Those who are enamored of practice
without theory are like a pilot who goes
into a ship without rudder or compass and
never has any certainty where he is going.
Practice should always be based upon a
sound knowledge of theory.*

Leonardo da Vinci (1452–1519)

Operace:

Operace:

1 sjednocení

Operace:

1 sjednocení

2 průnik

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl
- 4 restrikce

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl
- 4 restrikce
- 5 projekce

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl
- 4 restrikce
- 5 projekce
- 6 spojení

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl
- 4 restrikce
- 5 projekce
- 6 spojení
- 7 přejmenování atributů

Operace:

- 1 sjednocení
- 2 průnik
- 3 rozdíl
- 4 restrikce
- 5 projekce
- 6 spojení
- 7 přejmenování atributů

E. F. Codd (1970):

- bez přejmenování
- kartézský součin
- relační rozdíl

- relace
- relační proměnné
- relační algebra

- relace
- relační proměnné
- relační algebra

SQL:

- vychází z relačního modelu
- porušuje relační model

- relace
- relační proměnné
- relační algebra

SQL:

- vychází z relačního modelu
- porušuje relační model

Christopher J. Date a Hugh Darwen:

- The Third Manifesto (1995)
- specifikace relačního jazyka D
- Tutorial D
- Rel

- dotazovací jazyk
- vycházející z predikátové logiky

- dotazovací jazyk
- vycházející z predikátové logiky

Dotazy:

- relační symboly
- logické spojky
- kvantifikátory (existenční a obecný)

- dotazovací jazyk
- vycházející z predikátové logiky

Dotazy:

- relační symboly
- logické spojky
- kvantifikátory (existenční a obecný)

Druhy:

- n -ticový relační kalkul
- doménový relační kalkul

- dotazovací jazyk
- vycházející z predikátové logiky

Dotazy:

- relační symboly
- logické spojky
- kvantifikátory (existenční a obecný)

Druhy:

- n -ticový relační kalkul
- doménový relační kalkul

Stejná vyjadřovací síla jako relační algebra



Rovnost relací:

- 1 stejné záhlaví
- 2 stejné tělo

Rovnost relací:

- 1 stejné záhlaví
- 2 stejné tělo

v_1, v_2 : výrazy

ekvivalence v_1, v_2 :

- hodnota v_1 vždy rovna hodnotě v_2

v_1, v_2, v_3 : relační výrazy stejného typu

v_1, v_2, v_3 : relační výrazy stejného typu

Jsou ekvivalentní:

- $v_1 \text{ UNION } (v_2 \text{ UNION } v_3)$
- $(v_1 \text{ UNION } v_2) \text{ UNION } v_3$

v_1, v_2, v_3 : relační výrazy stejného typu

Jsou ekvivalentní:

- $v_1 \text{ UNION } (v_2 \text{ UNION } v_3)$
- $(v_1 \text{ UNION } v_2) \text{ UNION } v_3$

Závorky můžeme vynechat:

$v_1 \text{ UNION } v_2 \text{ UNION } v_3$

v_1, v_2, v_3 : relační výrazy stejného typu

Jsou ekvivalentní:

- $v_1 \text{ UNION } (v_2 \text{ UNION } v_3)$
- $(v_1 \text{ UNION } v_2) \text{ UNION } v_3$

Závorky můžeme vynechat:

$v_1 \text{ UNION } v_2 \text{ UNION } v_3$

Podobně:

$v_1 \text{ INTERSECT } v_2 \text{ INTERSECT } v_3$

v_1, v_2 : výrazy stejného typu

v_1, v_2 : výrazy stejného typu

Ekvivalentní:

- $v_1 \text{ INTERSECT } v_2$
- $v_1 \text{ EXCEPT } (v_1 \text{ EXCEPT } v_2)$

v_1, v_2 : výrazy stejného typu

Ekvivalentní:

- v_1 INTERSECT v_2
- v_1 EXCEPT (v_1 EXCEPT v_2)

Operaci průniku můžeme vyjádřit pomocí operace rozdílu.

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

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

Ekvivalentní c_1, c_2 :

pro každou n -tici t nad A_1, \dots, A_n :

- podmínka c_1 je v t splněna, právě když podmínka c_2 je v t splněna

c_1, c_2, c_3 : podmínky nad A_1, \dots, A_n

c_1, c_2, c_3 : podmínky nad A_1, \dots, A_n

Ekvivalentní:

- $c_1 \text{ OR } (c_2 \text{ OR } c_3)$
- $(c_1 \text{ OR } c_2) \text{ OR } c_3$

c_1, c_2, c_3 : podmínky nad A_1, \dots, A_n

Ekvivalentní:

- $c_1 \text{ OR } (c_2 \text{ OR } c_3)$
- $(c_1 \text{ OR } c_2) \text{ OR } c_3$

Vynecháváme závorky:

$$c_1 \text{ OR } c_2 \text{ OR } c_3$$

c_1, c_2, c_3 : podmínky nad A_1, \dots, A_n

Ekvivalentní:

- $c_1 \text{ OR } (c_2 \text{ OR } c_3)$
- $(c_1 \text{ OR } c_2) \text{ OR } c_3$

Vynecháváme závorky:

$c_1 \text{ OR } c_2 \text{ OR } c_3$

Podobně:

$c_1 \text{ AND } c_2 \text{ AND } c_3$

Tweedledum a Tweedledee (Tydliták a Tydlitek)



Tweedledum a Tweedledee (Tydliták a Tydlitek)





Za zrcadlem a co tam Alenka našla od Lewise Carrola.

Relace s prázdným záhlavím



Relace s prázdným záhlavím

\emptyset . . . záhlaví



Relace s prázdným záhlavím



\emptyset . . . záhlaví

n -tice se záhlavím \emptyset :

Relace s prázdným záhlavím



\emptyset . . . záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset (t_0)$

Relace s prázdným záhlavím



\emptyset . . . záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset (t_0)$

Relace se záhlavím \emptyset :

Relace s prázdným záhlavím



\emptyset ...záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset(t_0)$

Relace se záhlavím \emptyset :

- prázdná relace (DUM)

Relace s prázdným záhlavím



\emptyset ...záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset (t_0)$

Relace se záhlavím \emptyset :

- prázdná relace (DUM)
- relace s tělem $\{t_0\}$ (DEE)

Relace s prázdným záhlavím



\emptyset . . . záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset (t_0)$

Relace se záhlavím \emptyset :

- prázdná relace (DUM)
- relace s tělem $\{t_0\}$ (DEE)

DUM:

- Tweedledum
- nepravda

DEE:

- Tweedledee
- pravda

Relace s prázdným záhlavím



\emptyset ...záhlaví

n -tice se záhlavím \emptyset :

- $\emptyset(t_0)$

Relace se záhlavím \emptyset :

- prázdná relace (DUM)
- relace s tělem $\{t_0\}$ (DEE)

DUM:

- Tweedledum
- nepravda

DEE:

- Tweedledee
- pravda

V SQL neexistují

1 Relační algebra

2 Relační výraz SELECT

3 Relační dělení

Relační výraz SELECT



r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka nad vstupními atributy

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

...relační výraz

r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka nad vstupními atributy

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

... relační výraz

Popis vstupní relace:

- relace (vstupní relace)
- jméno relace
- typ relace

r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka nad vstupními atributy

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

... relační výraz

Popis vstupní relace:

- relace (vstupní relace)
- jméno relace
- typ relace

Jména vstupních relací: jedinečná

Relační výraz SELECT



r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

...relační výraz

Relační výraz SELECT



r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka

```
( SELECT DISTINCT  $a_1, \dots, a_m$ 
  FROM    $r_1, \dots, r_n$ 
  WHERE   $c$  )
```

...relační výraz

R : jméno vstupní relace

A : atribut v záhlaví R

$R.A$: vstupní atribut

Relační výraz SELECT



r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka

```
( SELECT DISTINCT  $a_1, \dots, a_m$ 
  FROM    $r_1, \dots, r_n$ 
 WHERE   $c$  )
```

...relační výraz

R : jméno vstupní relace

A : atribut v záhlaví R

$R.A$: vstupní atribut

Popis výstupního atributu:

- vstupní atribut
- atribut (výstupní atribut)

Relační výraz SELECT



r_1, \dots, r_n : popisy vstupních relací ($n > 1$)

a_1, \dots, a_m : popisy výstupních atributů ($m > 1$)

c : podmínka

```
( SELECT DISTINCT  $a_1, \dots, a_m$ 
  FROM    $r_1, \dots, r_n$ 
  WHERE   $c$  )
```

...relační výraz

R : jméno vstupní relace

A : atribut v záhlaví R

$R.A$: vstupní atribut

Popis výstupního atributu:

- vstupní atribut
- atribut (výstupní atribut)

Výstupní atributy: jedinečné



Vstupní relace určená výrazem



v : relační výraz
 R : jméno relace

v : relační výraz

R : jméno relace

v AS R

... popis vstupní relace:

- relace: hodnota v
- jméno: R
- typ: typ v

v : relační výraz

R : jméno relace

v AS R

... popis vstupní relace:

- relace: hodnota v
- jméno: R
- typ: typ v

Například:

(TABLE child) AS ch

Popis výstupního atributu



$R.A$: vstupní atribut

B : atribut

$R.A$: vstupní atribut

B : atribut

$R.A \text{ AS } B$

... popis výstupního atributu:

- vstupní atribut $R.A$
- výstupní atribut: B

$R.A$: vstupní atribut

B : atribut

$R.A \text{ AS } B$

... popis výstupního atributu:

- vstupní atribut $R.A$
- výstupní atribut: B

Například:

`ch.name AS child_name`

Příklad výrazu SELECT



parent	parent_name	child_name
	Pavel	Anna
	Monika	Bert
	Petr	Bert
	Marie	Daniela

child	name	age
	Anna	3
	Bert	4
	Cyril	4

```
SELECT DISTINCT p.parent_name AS parent_name, ch.age AS child_age
FROM ( TABLE parent ) AS p, ( TABLE child ) AS ch
WHERE p.child_name = ch.name
```



```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

1 Získání vstupních relací: r_1, \dots, r_n

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM    $r_1, \dots, r_n$   
  WHERE   $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n
- 3 Spojení relací r'_1, \dots, r'_n : s_1

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n
- 3 Spojení relací r'_1, \dots, r'_n : s_1
- 4 Restrikce relace s_1 vzhledem k podmínce c : s_2

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n
- 3 Spojení relací r'_1, \dots, r'_n : s_1
- 4 Restrikce relace s_1 vzhledem k podmínce c : s_2
- 5 Projekce relace s_2 na vstupní atributy uvedené v popisech výstupních atributů: s_3

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n
- 3 Spojení relací r'_1, \dots, r'_n : s_1
- 4 Restrikce relace s_1 vzhledem k podmínce c : s_2
- 5 Projekce relace s_2 na vstupní atributy uvedené v popisech výstupních atributů: s_3
- 6 Přejmenování vstupních atributů v záhlaví s_3 na výstupní atributy: s_4

Výstupní relace: s_4

```
( SELECT DISTINCT  $a_1, \dots, a_m$   
  FROM     $r_1, \dots, r_n$   
  WHERE    $c$  )
```

- 1 Získání vstupních relací: r_1, \dots, r_n
- 2 Přejmenování každého atributu A_j ve vstupní relaci r_i na $R_i.A_j$: r'_1, \dots, r'_n
- 3 Spojení relací r'_1, \dots, r'_n : s_1
- 4 Restrikce relace s_1 vzhledem k podmínce c : s_2
- 5 Projekce relace s_2 na vstupní atributy uvedené v popisech výstupních atributů: s_3
- 6 Přejmenování vstupních atributů v záhlaví s_3 na výstupní atributy: s_4

Výstupní relace: s_4

- používá pouze operace relační algebry
- lze chápat jako zkratku

Vyhodnocení příkladu výrazu SELECT



parent

parent_name	child_name
Pavel	Anna
Monika	Bert
Petr	Bert
Marie	Daniela

child

name	age
Anna	3
Bert	4
Cyril	4

#

Vyhodnocení příkladu výrazu SELECT



parent	parent_name	child_name
	Pavel	Anna
	Monika	Bert
	Petr	Bert
	Marie	Daniela

child	name	age
	Anna	3
	Bert	4
	Cyril	4

```
# SELECT DISTINCT p.parent_name AS parent_name, ch.age AS child_age
FROM ( TABLE parent ) AS p, ( TABLE child ) AS ch
WHERE p.child_name = ch.name;
```

Vyhodnocení příkladu výrazu SELECT



parent	parent_name	child_name
	Pavel	Anna
	Monika	Bert
	Petr	Bert
	Marie	Daniela

child	name	age
	Anna	3
	Bert	4
	Cyril	4

```
# SELECT DISTINCT p.parent_name AS parent_name, ch.age AS child_age
FROM ( TABLE parent ) AS p, ( TABLE child ) AS ch
WHERE p.child_name = ch.name;
```

parent_name	child_age
Pavel	3
Petr	4
Monika	4

(3 rows)

Příklad vyhodnocení: krok jedna



Získáme vstupní relace

Získáme vstupní relace

- Vstupní relace r_1 jménem p:

parent_name	child_name
Pavel	Anna
Monika	Bert
Petr	Bert
Marie	Daniela

- Vstupní relace r_2 jménem ch:

name	age
Anna	3
Bert	4
Cyril	4

Příklad vyhodnocení: krok dva



Přejmenování atributů vstupních relací na vstupní atributy

Přejmenování atributů vstupních relací na vstupní atributy

- Relace r'_1 :

p.parent_name	p.child_name
Pavel	Anna
Monika	Bert
Petr	Bert
Marie	Daniela

- Relace r'_2 :

ch.name	ch.age
Anna	3
Bert	4
Cyril	4

Příklad vyhodnocení: krok tři



Spojení relací r'_1 a r'_2 .

Příklad vyhodnocení: krok tři



Spojení relací r'_1 a r'_2 .

Výsledek s_1 :

p.parent_name	p.child_name	ch.name	ch.age
Pavel	Anna	Anna	3
Monika	Bert	Anna	3
Petr	Bert	Anna	3
Marie	Daniela	Anna	3
Pavel	Anna	Bert	4
Monika	Bert	Bert	4
Petr	Bert	Bert	4
Marie	Daniela	Bert	4
Pavel	Anna	Cyril	4
Monika	Bert	Cyril	4
Petr	Bert	Cyril	4
Marie	Daniela	Cyril	4

Restrikce s_1 vzhledem k podmínce `p.child_name = ch.name`

Restrikce s_1 vzhledem k podmínce `p.child_name = ch.name`

Výsledek s_2 :

p.parent_name	p.child_name	ch.name	ch.age
Pavel	Anna	Anna	3
Monika	Bert	Bert	4
Petr	Bert	Bert	4

Příklad vyhodnocení: krok pět



Projekce relace s_2 na `p.parent_name` a `ch.age`

Výsledek s_3 :

<code>p.parent_name</code>	<code>ch.age</code>
Pavel	3
Monika	4
Petr	4

Přejmenování atributů `p.parent_name` a `ch.age` v záhlaví s_3 na `parent_name` a `child_age`

Přejmenování atributů `p.parent_name` a `ch.age` v záhlaví s_3 na `parent_name` a `child_age`

Výsledek s_4 :

parent_name	child_age
Pavel	3
Monika	4
Petr	4

#

Příklad za použití relační algebry



```
# SELECT DISTINCT parent_name, child_age
FROM (
    SELECT *
    FROM ( TABLE parent ) AS t1
    NATURAL JOIN (
        SELECT name AS child_name, age AS child_age
        FROM ( TABLE child ) AS t
    ) AS t2
) AS t;
```

Příklad za použití relační algebry



```
# SELECT DISTINCT parent_name, child_age
  FROM (
    SELECT *
    FROM ( TABLE parent ) AS t1
    NATURAL JOIN (
      SELECT name AS child_name, age AS child_age
      FROM ( TABLE child ) AS t
    ) AS t2
  ) AS t;
```

parent_name	child_age
Pavel	3
Monika	4
Petr	4
(3 rows)	



Relační proměnná jako popis vstupní relace



R : relační proměnná

R : relační proměnná

R

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: R
- typ: typ proměnné R

R : relační proměnná

R

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: R
- typ: typ proměnné R

Například: `child`

R : relační proměnná

R

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: R
- typ: typ proměnné R

Například: child

Zjednodušení ukázky:

```
SELECT DISTINCT parent.parent_name AS parent_name,  
                child.age AS child_age  
FROM   parent, child  
WHERE  parent.child_name = child.name;
```

Přejmenování relační proměnné v SELECT



Přejmenování relační proměnné v SELECT



R : relační proměnná

S : jméno relace

R : relační proměnná

S : jméno relace

R AS S

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: S
- typ: typ proměnné R

Přejmenování relační proměnné v SELECT



R : relační proměnná

S : jméno relace

R AS S

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: S
- typ: typ proměnné R

Například: parent AS p

Přejmenování relační proměnné v SELECT



R : relační proměnná

S : jméno relace

R AS S

... popis vstupní relace:

- vstupní relace: hodnota R
- jméno: S
- typ: typ proměnné R

Například: parent AS p

Zjednodušení ukázky:

```
SELECT DISTINCT p.parent_name AS parent_name, ch.age AS child_age
FROM   parent AS p, child AS ch
WHERE  p.child_name = ch.name;
```

Vstupní atribut jako popis výstupního atributu



Vstupní atribut jako popis výstupního atributu



$R.A$: vstupní atribut

$R.A$: vstupní atribut

$R.A$

... popis výstupního atributu:

- vstupní atribut $R.A$
- výstupní atribut A

$R.A$: vstupní atribut

$R.A$

... popis výstupního atributu:

- vstupní atribut $R.A$
- výstupní atribut A

Například: `p.parent_name`

$R.A$: vstupní atribut

$R.A$

... popis výstupního atributu:

- vstupní atribut $R.A$
- výstupní atribut A

Například: `p.parent_name`

Zjednodušení ukázky:

```
SELECT DISTINCT p.parent_name, ch.age AS child_age
FROM   parent AS p, child AS ch
WHERE  p.child_name = ch.name;
```

Vstupní atribut s jedinečným jménem atributu



$R.A$:

- vstupní atribut
- neexistuje jiný vstupní atribut $R'.A$

$R.A$:

- vstupní atribut
- neexistuje jiný vstupní atribut $R'.A$

Místo $R.A$ můžeme psát jen A

$R.A$:

- vstupní atribut
- neexistuje jiný vstupní atribut $R'.A$

Místo $R.A$ můžeme psát jen A

Zjednodušení ukázky:

```
SELECT DISTINCT parent_name, age AS child_age
FROM   parent, child
WHERE  child_name = name;
```



r : neprázdná relace nad A_1, \dots, A_n

$\{t_1, \dots, t_m\}$: tělo r

v_{ij} : hodnotapřiřazená n -ticí t_j atributu A_i ($1 \leq i \leq n$ a $1 \leq j \leq m$)

R : jméno relace

r : neprázdná relace nad A_1, \dots, A_n

$\{t_1, \dots, t_m\}$: tělo r

v_{ij} : hodnotapřiřazená n -ticí t_j atributu A_i ($1 \leq i \leq n$ a $1 \leq j \leq m$)

R : jméno relace

```
( VALUES ( v11, ..., v1n ),  
          ⋮  
          ( vm1, ..., vmn ) ) AS R ( A1, ..., An ) )
```

... popis vstupní relace:

- vstupní relace: r
- jméno: R
- typ: $\{A_1, \dots, A_n\}$

#

```
# SELECT DISTINCT age
FROM   child, ( VALUES ( 'Anna' ), ( 'Bert' ) ) AS const ( val )
WHERE  name = val;
```

```
# SELECT DISTINCT age
  FROM  child, ( VALUES ( 'Anna' ), ( 'Bert' ) ) AS const ( val )
 WHERE name = val;
```

age

3

4

(2 rows)

*: zkratka za všechny vstupní atributy

*: zkratka za všechny vstupní atributy

Vstupní relace musí mít atributy s jedinečnými jmény.

*: zkratka za všechny vstupní atributy

Vstupní relace musí mít atributy s jedinečnými jmény.

Například:

#

*: zkratka za všechny vstupní atributy

Vstupní relace musí mít atributy s jedinečnými jmény.

Například:

```
# SELECT DISTINCT *  
FROM   parent, child  
WHERE  child_name = name;
```


*: zkratka za všechny vstupní atributy

Vstupní relace musí mít atributy s jedinečnými jmény.

Například:

```
# SELECT DISTINCT *  
FROM   parent, child  
WHERE  child_name = name;
```

parent_name	child_name	name	age
Petr	Bert	Bert	4
Monika	Bert	Bert	4
Pavel	Anna	Anna	3

(3 rows)

*: zkratka za všechny vstupní atributy

Vstupní relace musí mít atributy s jedinečnými jmény.

Například:

```
# SELECT DISTINCT *  
FROM   parent, child  
WHERE  child_name = name;
```

parent_name	child_name	name	age
Petr	Bert	Bert	4
Monika	Bert	Bert	4
Pavel	Anna	Anna	3

(3 rows)

* ...parent.parent_name, parent.child_name, child.name, child.age

Klauzule výrazu SELECT



Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

- Klauzule FROM:

```
FROM  $r_1, \dots, r_n$ 
```

Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

- Klauzule FROM:

```
FROM  $r_1, \dots, r_n$ 
```

- Klauzule WHERE (nepovinná):

```
WHERE  $c$ 
```

Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

- Klauzule FROM:

```
FROM  $r_1, \dots, r_n$ 
```

- Klauzule WHERE (nepovinná):

```
WHERE  $c$ 
```

Například bez klauzule WHERE:

```
#
```


Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

- Klauzule FROM:

```
FROM  $r_1, \dots, r_n$ 
```

- Klauzule WHERE (nepovinná):

```
WHERE  $c$ 
```

Například bez klauzule WHERE:

```
# SELECT DISTINCT name FROM child;
```

Klauzule výrazu SELECT



Výraz SELECT: rozdělen *klauzulí*:

- Klauzule SELECT:

```
SELECT DISTINCT  $a_1, \dots, a_m$ 
```

- Klauzule FROM:

```
FROM  $r_1, \dots, r_n$ 
```

- Klauzule WHERE (nepovinná):

```
WHERE  $c$ 
```

Například bez klauzule WHERE:

```
# SELECT DISTINCT name FROM child;
```

```
name
```

```
-----
```

```
Anna
```

```
Bert
```

```
Cyril
```

```
(3 rows)
```

1 Relační algebra

2 Relační výraz SELECT

3 Relační dělení

completed

student	task
Anna	DISK1
Anna	DISK2
Bert	DISK1
Bert	PAPR1
Cyril	DISK2
Cyril	DISK1
Cyril	PAPR1

disk_course

task
DISK1
DISK2

Kteří studenti splnili všechny předměty z diskrétních struktur?

t_1 : n -tice nad A_1, \dots, A_n

t_2 : n -tice nad B_1, \dots, B_m

$$\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$$

t_1 : n -tice nad A_1, \dots, A_n

t_2 : n -tice nad B_1, \dots, B_m

$$\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$$

$t_1 \cup t_2$: n -tice nad $A_1, \dots, A_n, B_1, \dots, B_m$

t_1 : n -tice nad A_1, \dots, A_n

t_2 : n -tice nad B_1, \dots, B_m

$$\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$$

$t_1 \cup t_2$: n -tice nad $A_1, \dots, A_n, B_1, \dots, B_m$

r_1 : relace nad A_1, \dots, A_n

r_2 : relace nad A_m, \dots, A_n ($1 < m \leq n$)

t_1 : n -tice nad A_1, \dots, A_n

t_2 : n -tice nad B_1, \dots, B_m

$$\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$$

$t_1 \cup t_2$: n -tice nad $A_1, \dots, A_n, B_1, \dots, B_m$

r_1 : relace nad A_1, \dots, A_n

r_2 : relace nad A_m, \dots, A_n ($1 < m \leq n$)

r' :

- relace nad A_1, \dots, A_{m-1}

- tělo obsahuje všechny n -tice t' nad A_1, \dots, A_{m-1} :

Pro každou n -tici t_2 v těle r_2 je $t' \cup t_2$ v těle r_1 .

- **podíl** r_1 a r_2

r_1	student	task
	Anna	DISK1
	Anna	DISK2
	Bert	DISK1
	Bert	PAPR1
	Cyril	DISK2
	Cyril	DISK1
	Cyril	PAPR1

r_2	task
	DISK1
	DISK2

r_1 děleno r_2 :

r_1	student	task
	Anna	DISK1
	Anna	DISK2
	Bert	DISK1
	Bert	PAPR1
	Cyril	DISK2
	Cyril	DISK1
	Cyril	PAPR1

r_2	task
	DISK1
	DISK2

r_1 děleno r_2 :

student
Anna
Cyril

Vyjádření relačního dělení



Demonstrujeme na ukázce:

completed	student	task
	Anna	DISK1
	Anna	DISK2
	Bert	DISK1
	Bert	PAPR1
	Cyril	DISK2
	Cyril	DISK1
	Cyril	PAPR1

disk_course	task
	DISK1
	DISK2

r_1 : hodnota proměnné completed

r_2 : hodnota proměnné disk_course

$n = m = 2$

$r_1 \dots$ relace nad student a task

$r_2 \dots$ relace nad task

$A_1 = \text{student}$

$A_2 = \text{task}$

Kartézský součin všech projekcí r_1 na atribut A_i ($1 \leq i \leq m - 1$) a relace r_2 :

```
# SELECT *  
  FROM (SELECT DISTINCT student FROM completed) AS t1  
  NATURAL JOIN (TABLE disk_course) AS t2;
```

student		task
Anna		DISK1
Bert		DISK1
Cyril		DISK1
Anna		DISK2
Bert		DISK2
Cyril		DISK2

(6 rows)

Relace s_1 : student mohl splnit předmět z kurzu diskrétních struktur

Rozdíl s_1 a r_2 :

```
# ( SELECT *  
    FROM (SELECT DISTINCT student FROM completed) AS t1  
    NATURAL JOIN (TABLE disk_course) AS t2  )  
EXCEPT  
( TABLE completed );
```

```
student | task  
-----+-----  
Bert    | DISK2  
(1 row)
```

Relace s_2 : Student nesplnil předmět z diskrétních struktur

Projekce s_2 na A_1, \dots, A_{m-1} :

```
# SELECT DISTINCT student
FROM (
    ( SELECT *
      FROM ( SELECT DISTINCT student FROM completed ) AS t1
        NATURAL JOIN (TABLE disk_course) AS t2 )
    EXCEPT
    ( TABLE completed )
) AS t;
```

student

Bert

(1 row)

Relace s_3 : Student nesplnil aspoň jeden předmět z diskrétních struktur

Relační dělení: krok čtyři



Rozdíl projekce r_1 na A_1, \dots, A_{m-1} a s_3 :

```
# ( SELECT DISTINCT student FROM completed )
  EXCEPT
  ( SELECT DISTINCT student
    FROM (
      ( SELECT *
        FROM ( SELECT DISTINCT student FROM completed ) AS t1
        NATURAL JOIN (TABLE disk_course) AS t2 )
      EXCEPT
      ( TABLE completed )
    ) AS t );
```

student

Anna

Cyril

(2 rows)

... lze vyjádřit operacemi relační algebry