

Databázové systémy

# Přednáška 1. Relace v databázích

Jan Laštovička



KATEDRA INFORMATIKY  
UNIVERZITA PALACKÉHO V OLOMOUCI

## 1 Tabulky

## 2 Relace a relační proměnné

## 3 SQL



# Úvodní příklad: školka



- tři děti: Anna, Bert, Cyril

# Úvodní příklad: školka



- tři děti: Anna, Bert, Cyril
- Anně jsou tři roky, Bertovi a Cyrilovi jsou čtyři roky

- tři děti: Anna, Bert, Cyril
- Anně jsou tři roky, Bertovi a Cyrilovi jsou čtyři roky

Tabulka:

name	age
Anna	3
Bert	4
Cyril	4

- tři děti: Anna, Bert, Cyril
- Anně jsou tři roky, Bertovi a Cyrilovi jsou čtyři roky

Tabulka:

name	age
Anna	3
Bert	4
Cyril	4

Otázky:

- 1 Chodí Cyril do školky?
- 2 Chodí Daniela do školky?
- 3 Kolik let je Anně?
- 4 Kterým dětem jsou čtyři roky?

# Nejednoznačnost tabulky



name	age
Anna	3
Bert	4
Cyril	4



# Nejednoznačnost tabulky



name	age
Anna	3
Bert	4
Cyril	4

name	age
Bert	4
Cyril	4
Anna	3

prohozené řádky

name	age
Anna	3
Bert	4
Cyril	4

name	age
Bert	4
Cyril	4
Anna	3

prohozené řádky

age	name
3	Anna
4	Bert
4	Cyril

prohozené sloupce

name	age
Anna	3
Bert	4
Cyril	4

name	age
Bert	4
Cyril	4
Anna	3

prohozené řádky

age	name
3	Anna
4	Bert
4	Cyril

prohozené sloupce

name	age
Anna	3
Anna	3
Bert	4
Cyril	4

opakující se řádky

name	age
Anna	3
Bert	4
Cyril	4

name	age
Bert	4
Cyril	4
Anna	3

prohozené řádky

age	name
3	Anna
4	Bert
4	Cyril

name	age
Anna	3
Anna	3
Bert	4
Cyril	4

prohozené sloupce

opakující se řádky

Lepší reprezentace?

## 1 Tabulky

## 2 Relace a relační proměnné

## 3 SQL

... hodnoty v tabulce jsou určitého typu

... hodnoty v tabulce jsou určitého typu

*Typem* rozumíme pojmenovanou množinu hodnot.

... hodnoty v tabulce jsou určitého typu

*Typem* rozumíme pojmenovanou množinu hodnot.

Příklady:

- `varchar(i)` ... množina všech řetězců délky nejvýše *i*



... hodnoty v tabulce jsou určitého typu

*Typem* rozumíme pojmenovanou množinu hodnot.

Příklady:

- `varchar(i)` ... množina všech řetězců délky nejvýše *i*
- `integer` ... množina celých čísel od  $-2\,147\,483\,648$  do  $2\,147\,483\,647$

... hodnoty v tabulce jsou určitého typu

*Typem* rozumíme pojmenovanou množinu hodnot.

Příklady:

- `varchar(i)` ... množina všech řetězců délky nejvýše *i*
- `integer` ... množina celých čísel od  $-2\,147\,483\,648$  do  $2\,147\,483\,647$

*Hodnota  $v$  je typu  $T$ , pokud  $v$  je prvkem množiny jménem  $T$ .*

... hodnoty v tabulce jsou určitého typu

*Typem* rozumíme pojmenovanou množinu hodnot.

Příklady:

- `varchar(i)` ... množina všech řetězců délky nejvýše  $i$
- `integer` ... množina celých čísel od  $-2\,147\,483\,648$  do  $2\,147\,483\,647$

*Hodnota  $v$  je typu  $T$ , pokud  $v$  je prvkem množiny jménem  $T$ .*

Příklady:

- `'Anna'` je typu `varchar(4)`
- `'Anna'` není typu `varchar(3)`
- `3,4,5,6, 7` jsou typu `integer`

... buňky záhlaví

name	age
Anna	3
Bert	4
Cyril	4

... buňky záhlaví

name	age
Anna	3
Bert	4
Cyril	4

*Atribut je dán svým jménem a jménem svého typu.*

... buňky záhlaví

name	age
Anna	3
Bert	4
Cyril	4

*Atribut je dán svým jménem a jménem svého typu.*

Příklady:

- atribut name typu `varchar(10)`
- atribut age typu `integer`

... záhlaví tabulky

... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*



... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*

- například: {name varchar(10), age integer}

... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*

- například: {name varchar(10), age integer}
- zobrazení:

name varchar(10)	age integer
------------------	-------------

... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*

- například: {name varchar(10), age integer}

- zobrazení:

name varchar(10)	age integer
------------------	-------------

- typy vynecháváme: {name, age}

... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*

- například: {name varchar(10), age integer}

- zobrazení:

name varchar(10)	age integer
------------------	-------------

- typy vynecháváme: {name, age}

- zobrazení:

name	age
------	-----

... záhlaví tabulky

*Konečnou množinu atributů s unikátními jmény nazveme **záhlavím**.*

- například: {name varchar(10), age integer}

- zobrazení:

name varchar(10)	age integer
------------------	-------------

- typy vynecháváme: {name, age}

- zobrazení:

name	age
------	-----

- nezáleží na pořadí:

age	name
-----	------

...řádek tabulky

... řádek tabulky

$A$ : atribut

***Komponenta** přiřazuje atributu  $A$  hodnotu, která je stejného typu, jako je typ atributu  $A$ .*

... řádek tabulky

$A$ : atribut

*Komponenta* přiřazuje atributu  $A$  hodnotu, která je stejného typu, jako je typ atributu  $A$ .

Například: komponenta atributu `name` typu `varchar(10)` přiřadí hodnotu `'Anna'`



... řádek tabulky

$A$ : atribut

***Komponenta** přiřazuje atributu  $A$  hodnotu, která je stejného typu, jako je typ atributu  $A$ .*

Například: komponenta atributu `name` typu `varchar(10)` přiřadí hodnotu 'Anna'

*Konečnou množinu komponent  $t$  s jedinečnými názvy atributů nazýváme  **$n$ -ticí nad  $A_1, \dots, A_n$** , kde  $A_1, \dots, A_n$  jsou všechny atributy komponent v  $t$ .*

... řádek tabulky

$A$ : atribut

*Komponenta* přiřazuje atributu  $A$  hodnotu, která je stejného typu, jako je typ atributu  $A$ .

Například: komponenta atributu `name` typu `varchar(10)` přiřadí hodnotu 'Anna'

Konečnou množinu komponent  $t$  s jedinečnými názvy atributů nazýváme  *$n$ -ticí nad*  $A_1, \dots, A_n$ , kde  $A_1, \dots, A_n$  jsou všechny atributy komponent v  $t$ .

Množina  $\{A_1, \dots, A_n\}$  se nazývá *záhlaví  $n$ -tice  $t$* .

...řádek tabulky

$A$ : atribut

*Komponenta* přiřazuje atributu  $A$  hodnotu, která je stejného typu, jako je typ atributu  $A$ .

Například: komponenta atributu `name` typu `varchar(10)` přiřadí hodnotu 'Anna'

Konečnou množinu komponent  $t$  s jedinečnými názvy atributů nazýváme  *$n$ -ticí nad  $A_1, \dots, A_n$* , kde  $A_1, \dots, A_n$  jsou všechny atributy komponent v  $t$ .

Množina  $\{A_1, \dots, A_n\}$  se nazývá *záhlaví  $n$ -tice  $t$* .

Komponenty:

- $c_1$  atributu `name` přiřazuje 'Anna'
  - $c_2$  atributu `age` přiřazuje 3
- $\{c_1, c_2\}$  ...  $n$ -tice nad `name` a `age`

# Zobrazení $n$ -tice tabulkou



Komponenty:

- $c_1$  atributu name přiřazuje 'Anna'
- $c_2$  atributu age přiřazuje 3

$\{c_1, c_2\}$  ...  $n$ -tice nad name a age

# Zobrazení $n$ -tice tabulkou



Komponenty:

- $c_1$  atributu name přiřazuje 'Anna'
  - $c_2$  atributu age přiřazuje 3
- $\{c_1, c_2\}$  ...  $n$ -tice nad name a age

Zobrazení:

name	age
Anna	3

# Zobrazení $n$ -tice tabulkou



Komponenty:

- $c_1$  atributu name přiřazuje 'Anna'
  - $c_2$  atributu age přiřazuje 3
- $\{c_1, c_2\}$  ...  $n$ -tice nad name a age

Zobrazení:

name	age
Anna	3

Chybné tabulky:

name	age
Anna	

chybí hodnota

# Zobrazení $n$ -tice tabulkou



Komponenty:

- $c_1$  atributu name přiřazuje 'Anna'
  - $c_2$  atributu age přiřazuje 3
- $\{c_1, c_2\}$  ...  $n$ -tice nad name a age

Zobrazení:

name	age
Anna	3

Chybné tabulky:

name	age
Anna	

chybí hodnota

name	age
Anna	3, 4

dvě hodnoty

... tabulka



... tabulka

$\{A_1, \dots, A_n\}$  ... záhlaví

... tabulka

$\{A_1, \dots, A_n\}$  ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

... tabulka

$\{A_1, \dots, A_n\}$ ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

*Tělo relace je konečná množina  $n$ -tic nad  $A_1, \dots, A_n$ .*

... tabulka

$\{A_1, \dots, A_n\}$ ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

*Tělo relace je konečná množina  $n$ -tic nad  $A_1, \dots, A_n$ .*

*Relaci, jejíž tělo je prázdná množina, říkáme **prázdná relace**.*

... tabulka

$\{A_1, \dots, A_n\}$  ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

*Tělo relace je konečná množina  $n$ -tic nad  $A_1, \dots, A_n$ .*

*Relaci, jejíž tělo je prázdná množina, říkáme **prázdná relace**.*

Například:

*r*... relace (dětí ve školce) nad name a age

... tabulka

$\{A_1, \dots, A_n\}$  ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

*Tělo relace je konečná množina  $n$ -tic nad  $A_1, \dots, A_n$ .*

*Relaci, jejíž tělo je prázdná množina, říkáme **prázdná relace**.*

Například:

$r$  ... relace (dětí ve školce) nad name a age

tělo  $r$  je  $\{t_1, t_2, t_3\}$

... tabulka

$\{A_1, \dots, A_n\}$  ... záhlaví

*Relace nad atributy  $A_1, \dots, A_n$  se skládá ze záhlaví  $\{A_1, \dots, A_n\}$  a těla.*

*Tělo relace je konečná množina  $n$ -tic nad  $A_1, \dots, A_n$ .*

*Relaci, jejíž tělo je prázdná množina, říkáme **prázdná relace**.*

Například:

$r$  ... relace (dětí ve školce) nad name a age

tělo  $r$  je  $\{t_1, t_2, t_3\}$

- $t_1$  přiřazuje name hodnotu 'Anna' a age hodnotu 3
- $t_2$  přiřazuje name hodnotu 'Bert' a age hodnotu 4
- $t_3$  přiřazuje name hodnotu 'Cyril' a age hodnotu 4

# Zobrazení relace tabulkou



name varchar(10)	age integer
Anna	3
Bert	4
Cyril	4



name varchar(10)	age integer
Anna	3
Bert	4
Cyril	4

Vynechání typů:

name	age
Anna	3
Bert	4
Cyril	4

name varchar(10)	age integer
Anna	3
Bert	4
Cyril	4

Vynechání typů:

name	age
Anna	3
Bert	4
Cyril	4

Chybná tabulka:

name	age
Anna	
Bert	3, 4
Cyril	4

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Relační proměnná může uchovávat pouze relace s odpovídajícím záhlavím.*

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Relační proměnná může uchovávat pouze relace s odpovídajícím záhlavím.*

Například:

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Relační proměnná může uchovávat pouze relace s odpovídajícím záhlavím.*

Například:

- 1 definujeme relační proměnnou `child` typu `{name varchar(10), age integer}`

*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Relační proměnná může uchovávat pouze relace s odpovídajícím záhlavím.*

Například:

- 1 definujeme relační proměnnou `child` typu `{name varchar(10), age integer}`
- 2 nastavíme její hodnotu na relaci dětí školky



*Proměnným, které uchovávají relace, říkáme přirozeně **relační proměnné**.*

- relační proměnná má jméno a je jistého typu
- typ proměnné je dán záhlavím

*Relační proměnná může uchovávat pouze relace s odpovídajícím záhlavím.*

Například:

- 1 definujeme relační proměnnou `child` typu `{name varchar(10), age integer}`
- 2 nastavíme její hodnotu na relaci dětí školky

Zobrazíme pojmenovanou tabulkou:

child	name	age
	Anna	3
	Bert	4
	Cyril	4

## 1 Tabulky

## 2 Relace a relační proměnné

## 3 SQL



- systém řízení báze dat (Database Management System)

- systém řízení báze dat (Database Management System)
- <https://www.postgresql.org>

- systém řízení báze dat (Database Management System)
- <https://www.postgresql.org>
- psql (SQL shell)

# Příkaz deklarující relační proměnnou



- $R$ . . . jméno relační proměnné
- $n > 0$
- různá jména atributů:  $A_1, \dots, A_n$
- jména typů:  $T_1, \dots, T_n$

- $R$ ...jméno relační proměnné
- $n > 0$
- různá jména atributů:  $A_1, \dots, A_n$
- jména typů:  $T_1, \dots, T_n$

```
CREATE TABLE  $R$  (  
     $A_1$   $T_1$  NOT NULL,  
     $\vdots$   
     $A_n$   $T_n$  NOT NULL,  
    UNIQUE ( $A_1, \dots, A_n$ )  
);
```



- $R$ ...jméno relační proměnné
- $n > 0$
- různá jména atributů:  $A_1, \dots, A_n$
- jména typů:  $T_1, \dots, T_n$

```
CREATE TABLE  $R$  (  
     $A_1$   $T_1$  NOT NULL,  
     $\vdots$   
     $A_n$   $T_n$  NOT NULL,  
    UNIQUE ( $A_1, \dots, A_n$ )  
);
```

- deklaruje relační proměnnou  $R$  nad  $A_1, \dots, A_n$
- $A_i$  je typu  $T_i$  pro každé  $1 \leq i \leq n$
- hodnota proměnné  $R$  je prázdná relace nad  $A_1, \dots, A_n$

```
# CREATE TABLE child (  
    name varchar(10) NOT NULL,  
    age integer NOT NULL,  
    UNIQUE (name, age)  
);
```

```
# CREATE TABLE child (  
    name varchar(10) NOT NULL,  
    age integer NOT NULL,  
    UNIQUE (name, age)  
);
```

- deklaruje relační proměnnou child nad name a age
- name je typu varchar(10)
- age je typu integer
- hodnota relační proměnné child je prázdná relace nad name a age

# Zobrazení hodnoty relační proměnné



$R$  ... relační proměnná

```
TABLE  $R$ ;
```

# Zobrazení hodnoty relační proměnné



$R$  ... relační proměnná

```
TABLE  $R$ ;
```

například:

```
# TABLE child;
```

$R$  ... relační proměnná

```
TABLE  $R$ ;
```

například:

```
# TABLE child;
```

zobrazí prázdnou relaci:

```
name | age  
-----+-----  
(0 rows)
```

- relační proměnná `child` obsahuje prázdnou relaci



- relační proměnná `child` obsahuje prázdnou relaci
- chceme, aby relační proměnná `child` obsahovala neprázdnou relaci



- relační proměnná `child` obsahuje prázdnou relaci
- chceme, aby relační proměnná `child` obsahovala neprázdnou relaci
- musíme změnit hodnotu proměnné `child`

- relační proměnná `child` obsahuje prázdnou relaci
- chceme, aby relační proměnná `child` obsahovala neprázdnou relaci
- musíme změnit hodnotu proměnné `child`
- relace měnit nelze

- relační proměnná `child` obsahuje prázdnou relaci
- chceme, aby relační proměnná `child` obsahovala neprázdnou relaci
- musíme změnit hodnotu proměnné `child`
- relace měnit nelze
- musíme s použitím hodnoty proměnné `child` vytvořit novou relaci

- relační proměnná `child` obsahuje prázdnou relaci
- chceme, aby relační proměnná `child` obsahovala neprázdnou relaci
- musíme změnit hodnotu proměnné `child`
- relace měnit nelze
- musíme s použitím hodnoty proměnné `child` vytvořit novou relaci
- novou relaci musíme vložit do proměnné `child`

# Příkaz přidání $n$ -tic do proměnné



- $R$  je relační proměnná nad  $A_1, \dots, A_n$
- $A_i$  je typu  $T_i$  pro každé  $1 \leq i \leq n$
- $\{t_1, \dots, t_m\}$  je konečná neprázdná množina  $n$ -tic nad  $A_1, \dots, A_n$
- $\{t_1, \dots, t_m\}$  je disjunktní s tělem relace  $R$
- hodnotu  $v_{ij}$  přiřazuje  $n$ -tice  $t_j$  atributu  $A_i$

- $R$  je relační proměnná nad  $A_1, \dots, A_n$
- $A_i$  je typu  $T_i$  pro každé  $1 \leq i \leq n$
- $\{t_1, \dots, t_m\}$  je konečná neprázdná množina  $n$ -tic nad  $A_1, \dots, A_n$
- $\{t_1, \dots, t_m\}$  je disjunktní s tělem relace  $R$
- hodnotu  $v_{ij}$  přiřazuje  $n$ -tice  $t_j$  atributu  $A_i$

```
INSERT INTO  $R$  ( $A_1, \dots, A_n$ ) VALUES  
    (  $v_{11}, \dots, v_{1n}$  ),  
    ⋮  
    (  $v_{m1}, \dots, v_{mn}$  );
```

# Příkaz přidání $n$ -tic do proměnné



- $R$  je relační proměnná nad  $A_1, \dots, A_n$
- $A_i$  je typu  $T_i$  pro každé  $1 \leq i \leq n$
- $\{t_1, \dots, t_m\}$  je konečná neprázdná množina  $n$ -tic nad  $A_1, \dots, A_n$
- $\{t_1, \dots, t_m\}$  je disjunktní s tělem relace  $R$
- hodnotu  $v_{ij}$  přiřazuje  $n$ -tice  $t_j$  atributu  $A_i$

```
INSERT INTO  $R$  ( $A_1, \dots, A_n$ ) VALUES  
    (  $v_{11}, \dots, v_{1n}$  ),  
    ⋮  
    (  $v_{m1}, \dots, v_{mn}$  );
```

$r \dots$  relace proměnné  $R$

$r'$  je relace:

- 1 má stejné záhlaví jako relace  $r$
- 2 tělo vznikne sjednocením těla  $r$  a množiny  $\{t_1, \dots, t_m\}$

Příkaz nastaví hodnotu proměnné  $R$  na  $r'$ .

```
# INSERT INTO child (name, age) VALUES  
  ( 'Anna', 3 ),  
  ( 'Bert', 4 ),  
  ( 'Cyril', 4 );
```



```
# INSERT INTO child (name, age) VALUES
    ( 'Anna', 3 ),
    ( 'Bert', 4 ),
    ( 'Cyril', 4 );
```

Tělo child obsahuje tři *n*-tice:

```
# TABLE child;
```

name		age
Bert		4
Cyril		4
Anna		3

(3 rows)

Přidání dítěte do relace child:

```
#
```

Přidání dítěte do relace child:

```
# INSERT INTO child (name, age) VALUES  
  ( 'Daniela', 5);
```

Přidání dítěte do relace child:

```
# INSERT INTO child (name, age) VALUES  
    ( 'Daniela', 5);  
#
```

Přidání dítěte do relace child:

```
# INSERT INTO child (name, age) VALUES  
    ( 'Daniela', 5);  
# TABLE child;
```

Přidání dítěte do relace child:

```
# INSERT INTO child (name, age) VALUES  
    ( 'Daniela', 5);  
# TABLE child;
```

name		age
Bert		4
Cyril		4
Anna		3
Daniela		5

(4 rows)

# Příkaz zrušení relační proměnné



*R* . . . relační proměnná

```
DROP TABLE R;
```

*R*... relační proměnná

```
DROP TABLE R;
```

Zrušení relační proměnné child:

```
DROP TABLE child;
```