Databázové systémy

Přirozené spojení

Vilém Vychodil

KMI/DATA1, Přednáška 4

Databázové systémy

Přednáška 4: Přehled

1. Přirozené spojení:
   - definice operace,
   - rozšíření na libovolně mnoho argumentů,
   - implementace v Tutorial D a SQL.
2. Vlastnosti přirozeného spojení:
   - základní vlastnosti,
   - vztah k dalším operacím.
3. Speciální případy přirozeného spojení:
   - průnik, restrikce,
   - kartézský součin, kompozice, polospojení.
4. Fyzická vrstva databáze a otázky efektivity:
   - přirozené spojení a role indexů.

Motivace pro přirozené spojení

motivace:

Řada užitečných dotazů je založena na párování dat z několika tabulek na základě stejných hodnot společných atributů – dotazy tohoto typu lze v RM formalizoavat pomocí relační operace „spojení."

NAME	ID
Abbe	333
Blangis	552
Curval	666
Durcet	101

,

ID	YEAR	COURSE	TYPE
101	2013	KMI/RTFM1	C
333	2012	KMI/DATA1	A
333	2013	KMI/DATA1	A
666	2012	KMI/PAPR1	B
666	2013	KMI/DATA1	A
666	2013	KMI/PAPR1	B

$\Longrightarrow$

NAME	ID	YEAR	COURSE	TYPE
Abbe	333	2012	KMI/DATA1	A
Abbe	333	2013	KMI/DATA1	A
Curval	666	2012	KMI/PAPR1	B
Curval	666	2013	KMI/DATA1	A
Curval	666	2013	KMI/PAPR1	B
Durcet	101	2013	KMI/RTFM1	C

Spojitelné n-tice a spojení n-tic

rozšíření pojmů (PŘEDNÁŠKA 3):

spojitelnost, **spojení n-tic**, angl.: *joinable tuples, tuple join*

Mějme n-tice $r \in \prod_{y \in R} D_y$ a $s \in \prod_{y \in S} D_y$. Pokud $r(y) = s(y)$ pro každý atribut $y \in R \cap S$, pak řekneme, že r a s jsou spojitelné. Pokud jsou r a s spojitelné, pak zobrazení $r \cup s$ (zkráceně rs) nazveme spojení n-tic r a s.

poznámky:

- spojení je: komutativní ($rs = sr$), asociativní ($r(st) = (rs)t$), idempotentní ($rr = r$), neutrální vzhledem k $\emptyset$ ($r\emptyset = \emptyset r = r$)
- důsledek: lze rozšířit na libovolné množství n-tic
- spojení n-tic $r \cup s$ je množinově-teoretické sjednocení zobrazení, odtud:

$$(rs)(y) = \begin{cases} r(y), & \text{pokud } y \in R, \\ s(y), & \text{jinak.} \end{cases}$$

Přirozené spojení

neformálně:

Spojení relací $\mathcal{D}_1$ a $\mathcal{D}_2$ je relace obsahující spojení r_1r_2 všech spojitelných n-tic $r_1 \in \mathcal{D}_1$ a $r_2 \in \mathcal{D}_2$ (a je to nejmenší relace této vlastnosti).

Definice (přirozené spojení, angl.: *natural join*)

Mějme relace $\mathcal{D}_1$ nad schématem $R \cup S$ a $\mathcal{D}_2$ nad schématem $S \cup T$ tak, že $R \cap T = \emptyset$. Relace $\mathcal{D}_1 \bowtie \mathcal{D}_2$ nad $R \cup S \cup T$ definovaná

$$\mathcal{D}_1 \bowtie \mathcal{D}_2 = \left\{ rst \in \textstyle\prod_{y \in R \cup S \cup T} D_y \,\middle|\, rs \in \mathcal{D}_1,\ st \in \mathcal{D}_2 \text{ a } s \in \textstyle\prod_{y \in S} D_y \right\}$$

se nazývá (**přirozené**) **spojení relací $\mathcal{D}_1$ a $\mathcal{D}_2$**.

poznámky:

- $(R \cup S) \cap (S \cup T) = S \cup (R \cap T) = S \cup \emptyset = S$
  (S je množina všech společných atributů relací $\mathcal{D}_1, \mathcal{D}_2$)
- $\mathcal{D}_1 \bowtie \mathcal{D}_2$ obsahuje všechny atributy z obou tabulek

Poznámky k definici přirozeného spojení

otázka:

V definici přirozeného spojení je pro dané $\mathcal{D}_1$ a $\mathcal{D}_2$ obecně možnost volit množiny R, S a T různě – je definice korektní?

Příklad (Různá vyjádření schémat R, S a T)

Pro $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech $R_1 = \{\texttt{FOO}, \texttt{BAR}, \texttt{BAZ}\}$ a $R_2 = \{\texttt{BAR}, \texttt{BAZ}, \texttt{QUX}\}$, lze psát

$R = \{\texttt{FOO}, \texttt{BAR}, \texttt{BAZ}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}, \texttt{BAR}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{BAZ}, \texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}, \texttt{BAR}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}, \texttt{BAZ}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{BAR}, \texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}, \texttt{BAZ}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{BAR}, \texttt{BAZ}, \texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{BAR}, \texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{BAZ}, \texttt{QUX}\}$, nebo

$R = \{\texttt{FOO}\}$, $S = \{\texttt{BAR}, \texttt{BAZ}\}$, $T = \{\texttt{QUX}\}$.

Věta (Charakterizace prvků $\mathcal{D}_1 \bowtie \mathcal{D}_2$ pomocí spojitelnosti)

$\mathcal{D}_1 \bowtie \mathcal{D}_2 = \{r_1r_2 \,|\, r_1 \in \mathcal{D}_1,\, r_2 \in \mathcal{D}_2$ a $r_1(R_1 \cap R_2) = r_2(R_1 \cap R_2)\}$

Důkaz.

Vezměme $rst \in \mathcal{D}_1 \bowtie \mathcal{D}_2$, to jest $rs \in \mathcal{D}_1$, $st \in \mathcal{D}_2$ a $s \in \prod_{y \in S} D_y$. Položme $r_1 = rs$ a $r_2 = st$. Z předchozího víme, že $R_1 \cap R_2 = S$ a tedy

$$r_1(R_1 \cap R_2) = r_1(S) = rs(S) = s = st(S) = r_2(S) = r_2(R_1 \cap R_2).$$

Opačně: Předpokládejme, že $r_1 \in \mathcal{D}_1$ a $r_2 \in \mathcal{D}_2$ splňují podmínku $r_1(R_1 \cap R_2) = r_2(R_1 \cap R_2)$. Můžeme uvažovat schémata $R = R_1 \setminus R_2$, $S = R_1 \cap R_2$ a $T = R_2 \setminus R_1$ a položit $r = r_1(R)$, $s = r_1(S)$, $t = r_2(T)$. Platí, že $rs = r_1(R)r_1(S) = r_1(R_1) = r_1 \in \mathcal{D}_1$, $st = r_1(S)r_2(T) = r_2(S)r_2(T) = r_2 \in \mathcal{D}_2$ a $rst = r_1r_2$, to jest $r_1r_2 \in \mathcal{D}_1 \bowtie \mathcal{D}_2$. □

- definice $\bowtie$ nezávisí na volbě R a T (**!!**)
- lze přijmout dodatečnou (praktickou) podmínku $R \cap S = S \cap T = \emptyset$

Příklad (Protipříklady k definici spojení)

Mějme relace $\mathcal{D}_1$ nad schématem $R \cup S$ a $\mathcal{D}_2$ nad schématem $S \cup T$ tak, že $R \cap T = \emptyset$. Pro *definici spojení* můžeme udělat následující pozorování:

1. podmínku $R \cap T = \emptyset$ v definici nelze vynechat:

   pro $R = \{\mathtt{FOO}\} = T$ a $S = \{\mathtt{BAR}\}$ a následující n-tice
   $r = \{\langle \mathtt{FOO}, \mathsf{a}\rangle, \langle \mathtt{BAR}, \mathsf{b}\rangle\}$, $s = \{\langle \mathtt{BAR}, \mathsf{b}\rangle\}$, $t = \{\langle \mathtt{FOO}, \mathsf{c}\rangle, \langle \mathtt{BAR}, \mathsf{b}\rangle\}$
   platí: rs a st nejsou spojitelné

2. podmínku $s \in \prod_{y \in S} D_y$ v definici nelze vynechat:

   pro $R = \{\mathtt{FOO}\}$, $S = \{\mathtt{BAR}\}$, $T = \{\mathtt{BAZ}\}$ a následující n-tice
   $r = \{\langle \mathtt{FOO}, \mathsf{a}\rangle, \langle \mathtt{BAR}, \mathsf{a}\rangle\}$, $s = \emptyset$, $t = \{\langle \mathtt{FOO}, \mathsf{b}\rangle, \langle \mathtt{BAR}, \mathsf{b}\rangle\}$
   platí: rs a st nejsou spojitelné

poznámka:

- spojení je definováno pro relace nad *libovolnými schématy* (**!!**)

Příklad (Příklady přirozených spojení)

FOO	BAR	BAZ
444	ghi	103
555	def	102
555	ghi	103
666	abc	101

$\bowtie$

BAR	BAZ	QUX
abc	111	zzz
def	102	www
def	102	yyy
ghX	103	xxx
ghi	103	ttt
ghi	103	uuu
ghi	103	vvv

$=$

FOO	BAR	BAZ	QUX
444	ghi	103	ttt
444	ghi	103	uuu
444	ghi	103	vvv
555	def	102	www
555	def	102	yyy
555	ghi	103	ttt
555	ghi	103	uuu
555	ghi	103	vvv

FOO	BAR	BAZ
555	def	102
555	ghi	103
666	abc	101

$\bowtie$

BAR	BAZ	QUX
AAA	101	AAA
def	102	BBB
ghi	333	CCC

$=$

FOO	BAR	BAZ	QUX
555	def	102	BBB

FOO	BAR	BAZ
333	jkl	103
444	ghi	102
555	def	101
666	abc	101

$\bowtie$

BAZ
101
103

$=$

FOO	BAR	BAZ
333	jkl	103
555	def	101
666	abc	101

Přirozené spojení v Tutorial D a SQL

Tutorial D:

$\langle \textit{relační-výraz}_1 \rangle$ `JOIN` $\langle \textit{relační-výraz}_2 \rangle$

`JOIN` $\{\langle \textit{relační-výraz}_1 \rangle, \langle \textit{relační-výraz}_2 \rangle, \ldots\}$

SQL:

`SELECT * FROM` $\langle \textit{jméno}_1 \rangle$ `NATURAL JOIN` $\langle \textit{jméno}_2 \rangle$

`SELECT` $\langle \textit{jméno}_1 \rangle$`.*,` $\langle \textit{jméno}_2 \rangle$`.`$\langle T\textit{-atribut}_1 \rangle, \ldots, \langle \textit{jméno}_2 \rangle$`.`$\langle T\textit{-atribut}_m \rangle$
`FROM` $\langle \textit{jméno}_1 \rangle$, $\langle \textit{jméno}_2 \rangle$
`WHERE` $\langle \textit{jméno}_1 \rangle$`.`$\langle S\textit{-atribut}_1 \rangle$ `=` $\langle \textit{jméno}_2 \rangle$`.`$\langle S\textit{-atribut}_1 \rangle$
`AND` $\cdots$ `AND`
$\langle \textit{jméno}_1 \rangle$`.`$\langle S\textit{-atribut}_n \rangle$ `=` $\langle \textit{jméno}_2 \rangle$`.`$\langle S\textit{-atribut}_n \rangle$, kde

$\langle T\textit{-atribut}_i \rangle$ jsou všechny atributy z $\langle \textit{jméno}_2 \rangle$, které nejsou v $\langle \textit{jméno}_1 \rangle$

$\langle S\textit{-atribut}_j \rangle$ jsou všechny atributy společné pro $\langle \textit{jméno}_1 \rangle$ a $\langle \textit{jméno}_2 \rangle$

Příklad (Tutorial D: Příklady přirozeného spojení)

```
VAR rel1 BASE
  INIT (RELATION {
    TUPLE {foo 666, bar "abc", baz 101},
    TUPLE {foo 555, bar "def", baz 102},
    TUPLE {foo 555, bar "ghi", baz 103},
    TUPLE {foo 444, bar "ghi", baz 103}})
  KEY {foo, bar, baz};

VAR rel2 BASE
  RELATION {bar CHAR, baz INTEGER, qux CHAR}
  KEY {bar, baz, qux}; ···

rel1 JOIN rel2
rel2 JOIN rel1
JOIN {rel1, rel2}
JOIN {rel2, rel1}
```

Příklad (SQL: Příklady přirozeného spojení)

```
CREATE TABLE rel1 (
  foo NUMERIC NOT NULL,
  bar VARCHAR NOT NULL,
  baz NUMERIC NOT NULL,
  PRIMARY KEY (foo, bar, baz));

CREATE TABLE rel2 (
  bar VARCHAR NOT NULL,
  baz NUMERIC NOT NULL,
  qux VARCHAR NOT NULL,
  PRIMARY KEY (bar, baz, qux)); ···

SELECT * FROM rel1 NATURAL JOIN rel2;

SELECT rel1.*, rel2.qux FROM rel1, rel2
  WHERE rel1.bar = rel2.bar AND rel1.baz = rel2.baz;
```

Věta (Základní vlastnosti přirozeného spojení)

Pro libovolné relace $\mathcal{D}, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ platí následující:

1. $\mathcal{D} \bowtie \mathcal{D} = \mathcal{D}$,
2. $\mathcal{D}_1 \bowtie \mathcal{D}_2 = \mathcal{D}_2 \bowtie \mathcal{D}_1$,
3. $\mathcal{D}_1 \bowtie (\mathcal{D}_2 \bowtie \mathcal{D}_3) = (\mathcal{D}_1 \bowtie \mathcal{D}_2) \bowtie \mathcal{D}_3$,
4. $\mathcal{D} \bowtie \mathcal{D}_\top = \mathcal{D}$,
5. $\mathcal{D} \bowtie \emptyset_S = \emptyset_{R \cup S}$ *pro* $\mathcal{D}$ *na* R *a pro libovolné* S.

Důkaz (začátek).

V případě 1 tvrzení plyne z toho, že každá n-tice z $\mathcal{D}$ je triviálně spojitelná sama se sebou. Bod 2 plyne z komutativity konjunkce. Bod 4 plyne z toho, že každá n-tice z $\mathcal{D}$ je triviálně spojitelná s $\emptyset$ (prázdná n-tice) a platí $r\emptyset = r$. Bod 5 je důsledkem faktu, v prázdné relaci (nad libovolným schématem) neexistuje žádná n-tice, která by mohla být spojitelná s jinou; to jest výsledkem spojení je prázdná relace nad sjedocením schémat obou relací. Zbývá prokázat bod 3 (pokračování, ...)

Důkaz (dokončení).

Mějme relace $\mathcal{D}_1$, $\mathcal{D}_2$ a $\mathcal{D}_3$ a označme jejich schémata $R_1 \cup R_{12} \cup R_{13} \cup R_{123}$, $R_2 \cup R_{12} \cup R_{23} \cup R_{123}$ a $R_3 \cup R_{13} \cup R_{23} \cup R_{123}$ (viz obrázek). Pak tvrzení na následujících řádcích jsou ekvivalentní:

$r_1 r_2 r_3 r_{12} r_{13} r_{23} r_{123} \in \mathcal{D}_1 \bowtie (\mathcal{D}_2 \bowtie \mathcal{D}_3)$

$r_1 (r_{12} r_{13} r_{123}) r_2 r_3 r_{23} \in \mathcal{D}_1 \bowtie (\mathcal{D}_2 \bowtie \mathcal{D}_3)$

$r_1 (r_{12} r_{13} r_{123}) \in \mathcal{D}_1$ a $(r_{12} r_{13} r_{123}) r_2 r_3 r_{23} \in \mathcal{D}_2 \bowtie \mathcal{D}_3$

$r_1 (r_{12} r_{13} r_{123}) \in \mathcal{D}_1$, $r_2 r_{12} (r_{23} r_{123}) \in \mathcal{D}_2$ a $(r_{23} r_{123}) r_3 r_{13} \in \mathcal{D}_3$

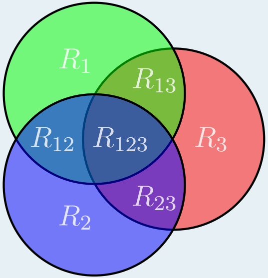
$r_1 r_{13} (r_{12} r_{123}) \in \mathcal{D}_1$, $(r_{12} r_{123}) r_2 r_{23} \in \mathcal{D}_2$ a $(r_{23} r_{123}) r_3 r_{13} \in \mathcal{D}_3$

$r_1 r_{13} (r_{12} r_{123}) r_2 r_{23} \in \mathcal{D}_1 \bowtie \mathcal{D}_2$ a $(r_{23} r_{123}) r_3 r_{13} \in \mathcal{D}_3$

$r_1 r_2 r_{12} (r_{13} r_{23} r_{123}) \in \mathcal{D}_1 \bowtie \mathcal{D}_2$ a $(r_{13} r_{23} r_{123}) r_3 \in \mathcal{D}_3$

$r_1 r_2 r_{12} (r_{13} r_{23} r_{123}) r_3 \in (\mathcal{D}_1 \bowtie \mathcal{D}_2) \bowtie \mathcal{D}_3$

$r_1 r_2 r_3 r_{12} r_{13} r_{23} r_{123} \in (\mathcal{D}_1 \bowtie \mathcal{D}_2) \bowtie \mathcal{D}_3$ $\square$



poznámka: věta nás opravňuje zavést $\bowtie_{i=1}^{n} \mathcal{D}_i$ nebo $\bowtie_{i \in I} \mathcal{D}_i$ (I je konečná)

Příklad (Tutorial D: Vlastnosti spojení)

```
/* idempotency of joins */
rt JOIN rt = rt  ⟾ TRUE

/* commutativity of joins */
rt1 JOIN rt2 = rt2 JOIN rt1  ⟾ TRUE

/* associativity of joins */
rt1 JOIN (rt2 JOIN rt3) = (rt1 JOIN rt2) JOIN rt3  ⟾ TRUE
rt1 JOIN rt2 JOIN rt3 = JOIN {rt1, rt2, rt3}       ⟾ TRUE

/* neutrality of DEE */
rt JOIN TABLE_DEE = TABLE_DEE JOIN rt = rt  ⟾ TRUE
rt JOIN RELATION {TUPLE {}} = rt            ⟾ TRUE

/* annihilation of DUM */
rt JOIN TABLE_DUM = (rt WHERE FALSE)  ⟾ TRUE
```

Speciální případy spojení: Průnik

Věta (Vztah spojení a průniku)

Nechť $\mathcal{D}_1$ a $\mathcal{D}_2$ jsou relace nad stejným schématem, pak $\mathcal{D}_1 \bowtie \mathcal{D}_2 = \mathcal{D}_1 \cap \mathcal{D}_2$.

Důkaz.

Jelikož mají $\mathcal{D}_1$ a $\mathcal{D}_2$ stejná schémata, pak $R = T = \emptyset$ a dostáváme:

$$\begin{aligned}
\mathcal{D}_1 \bowtie \mathcal{D}_2 &= \{rst \in \textstyle\prod_{y \in R \cup S \cup T} D_y \,|\, rs \in \mathcal{D}_1,\ st \in \mathcal{D}_2 \text{ a } s \in \textstyle\prod_{y \in S} D_y\} \\
&= \{\emptyset s \emptyset \in \textstyle\prod_{y \in \emptyset \cup S \cup \emptyset} D_y \,|\, \emptyset s \in \mathcal{D}_1,\ s\emptyset \in \mathcal{D}_2 \text{ a } s \in \textstyle\prod_{y \in S} D_y\} \\
&= \{s \in \textstyle\prod_{y \in S} D_y \,|\, s \in \mathcal{D}_1 \text{ a } s \in \mathcal{D}_2\} = \mathcal{D}_1 \cap \mathcal{D}_2.
\end{aligned}$$

□

důsledek (ekvivalence výrazů) za předpokladu stejných schémat argumentů:

- `SELECT * FROM foo INTERSECT SELECT * FROM bar` $\equiv$
  `SELECT * FROM foo NATURAL JOIN bar`
- `foo JOIN bar` $\equiv$ `foo INTERSECT bar`

Speciální případy spojení: Restrikce na rovnost

Věta (Vztah spojení a restrikce na rovnost)

Pokud je $\mathcal{D}$ *relace na schématu* R, $y \in R$ *a* $d \in D_y$, *pak* $\sigma_{y=d}(\mathcal{D}) = \mathcal{D} \bowtie \{\{\langle y, d\rangle\}\}$.

Důkaz.

Pro $\mathcal{D}$ na R, $T = \emptyset$ a $S = \{y\}$ dle definice spojení dostáváme:

$$\begin{aligned}
\mathcal{D} \bowtie \{\{\langle y, d\rangle\}\} &= \{rst \in \textstyle\prod_{y \in R \cup S \cup T} D_y \,|\, rs \in \mathcal{D},\ st \in \{\{\langle y, d\rangle\}\} \text{ a } s \in \prod_{y \in S} D_y\} \\
&= \{rs \in \textstyle\prod_{y \in R} D_y \,|\, rs \in \mathcal{D} \text{ a } s = \{\langle y, d\rangle\}\} \\
&= \{rs \in \mathcal{D} \,|\, s = \{\langle y, d\rangle\}\} = \{r \in \mathcal{D} \,|\, r(y) = d\} = \sigma_{y=d}(\mathcal{D}).
\end{aligned}$$

□

poznámky:

- restrikci na rovnost lze vyjádřit pomocí spojení se singletonem (Přednáška 2)
- $\langle$*relační-výraz*$\rangle$ `WHERE` $\langle$*atribut*$\rangle$=$\langle$*hodnota*$\rangle$ $\equiv$
  $\langle$*relační-výraz*$\rangle$ `JOIN RELATION {TUPLE {`$\langle$*atribut*$\rangle$ $\langle$*hodnota*$\rangle$`}}`

Příklad (Tutorial D: Průnik a restrikce na rovnost pomocí spojení)

```
/* if rt1 and rt2 have the same type: */
rt1 JOIN rt2 = rt1 INTERSECT rt2  ⟾  TRUE

/* generalization to arbitrary number of arguments */
JOIN {rt1, rt2, ...} = INTERSECT {rt1, rt2, ...}  ⟾  TRUE
```

Příklad (SQL: Průnik pomocí pomocí spojení)

```
/* intersection using INTERSECT */
SELECT * FROM tab1 INTERSECT SELECT FROM tab2;

/* intersection using NATURAL JOIN */
SELECT * FROM tab1 NATURAL JOIN tab2;
```

poznámka: PostgreSQL provádí předchozí pomocí obecně různých strategií

Intermezzo: Přejmenování n-tic

neformálně:

Přejmenováním n-tice zkonstruujeme novou n-tici, která obsahuje stejná data (stejných typů), ale může mít jiná jména atributů.

přejmenování n-tice, angl.: *tuple renaming*

Mějme n-tici $r \in \prod_{y \in R} D_y$ a injektivní zobrazení $h\colon R \to Y$. Pak složené zobrazení $\rho_h(r) = h^{-1} \circ r$ nazveme přejmenování n-tice r podle h.

význam: $\rho_h(r)$ je n-tice nad schématem $h(R)$, kde

$$(\rho_h(r))(h(y)) = r(y), \text{ pro každé } y \in R.$$

Tutorial D:

$\langle n\textit{-ticový-výraz}\rangle$ `RENAME` $\{\langle\textit{staré-jméno}_1\rangle$ `AS` $\langle\textit{nové-jméno}_1\rangle, \ldots\}$

$\langle n\textit{-ticový-výraz}\rangle$ `RENAME` $\{$`PREFIX` $\langle\textit{starý-řetězec}\rangle$ `AS` $\langle\textit{nový-řetězec}\rangle\}$

Přejmenování

Definice (přejmenování, angl.: *renaming*)

Mějme relaci $\mathcal{D}$ na schématu R a injektivní zobrazení $h\colon R \to Y$. Položíme:

$$\rho_h(\mathcal{D}) = \{\rho_h(r) \,|\, r \in \mathcal{D}\}.$$

Relace $\rho_h(\mathcal{D})$ se nazývá **přejmenování $\mathcal{D}$ podle h**. Pro jednoduchost někdy značíme $\rho_{y'_1 \leftarrow y_1, \ldots, y'_n \leftarrow y_n}(\mathcal{D})$ pokud $h(y_i) = y'_i$ $(i = 1, \ldots, n)$ a $h(y) = y$ jinak.

Tutorial D:

$\langle \textit{relační-výraz} \rangle$ `RENAME {`$\langle \textit{staré-jméno}_1 \rangle$ `AS` $\langle \textit{nové-jméno}_1 \rangle$`, ...}`

$\langle \textit{relační-výraz} \rangle$ `RENAME {PREFIX` $\langle \textit{starý-řetězec} \rangle$ `AS` $\langle \textit{nový-řetězec} \rangle$`}`

SQL:

`SELECT` $\langle \textit{staré-jméno}_1 \rangle$ `AS` $\langle \textit{nové-jméno}_1 \rangle$`, ...` `FROM` $\langle \textit{jméno} \rangle$

`SELECT` $\langle \textit{staré-jméno}_1 \rangle$␣$\langle \textit{nové-jméno}_1 \rangle$`, ...` `FROM` $\langle \textit{jméno} \rangle$

Příklad (Tutorial D: Přejmenování n-tic a relací)

```
TUPLE {x 10, y "foo", z 30} RENAME {y AS blah}
  ⟼ TUPLE {x 10, blah "foo", z 30}

TUPLE {xa 10, xb "foo", yc 30} RENAME {PREFIX "x" AS "z"}
  ⟼ TUPLE {za 10, zb "foo", yc 30}

TUPLE {xa 10, xb "foo", yc 30} RENAME {PREFIX "" AS "z"}
  ⟼ TUPLE {zxa 10, zxb "foo", zyc 30}

TUPLE {xa 10, xb "foo", yc 30} RENAME {}  /* error */

rt RENAME {y AS blah, z AS qux} ⟼ ···
```

Příklad (SQL: Přejmenování)

```
SELECT y AS blah FROM tab1;
SELECT y blah FROM tab1;
SELECT y AS blah, z AS qux FROM tab1;
SELECT y AS blah, z AS qux FROM tab1 WHERE blah = 10;  /* error */
```

Odvozené operace: Kartézský součin

Definice (kartézský součin, angl.: *cross join*)

Mějme relace $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech R a T takových, že $R \cap T = \emptyset$. Pak $\mathcal{D}_1 \bowtie \mathcal{D}_2$ se spojení nazývá **kartézský součin relací $\mathcal{D}_1$ a $\mathcal{D}_2$**.

podle definice spojení:

$$\begin{aligned} \mathcal{D}_1 \bowtie \mathcal{D}_2 &= \{rst \,|\, rs \in \mathcal{D}_1,\ st \in \mathcal{D}_2 \text{ a } s \in \textstyle\prod_{y \in S} D_y\} \\ &= \{r\emptyset t \,|\, r\emptyset \in \mathcal{D}_1 \text{ a } \emptyset t \in \mathcal{D}_2\} \\ &= \{rt \,|\, r \in \mathcal{D}_1 \text{ a } t \in \mathcal{D}_2\} \end{aligned}$$

poznámky:

- speciální případ $\bowtie$ pro $S = \emptyset$ (disjunktní schémata)
- každá n-tice z $\mathcal{D}_1$ je spojitelná s každou n-ticí z $\mathcal{D}_2$
- $|\mathcal{D}_1 \bowtie \mathcal{D}_2| = |\mathcal{D}_1| \cdot |\mathcal{D}_2|$ (pro obecná spojení platí pouze $\leq$)

Kartézský součin v Tutorial D a SQL

Tutorial D:

$\langle \textit{relační-výraz}_1 \rangle$ `TIMES` $\langle \textit{relační-výraz}_2 \rangle$

`TIMES` $\{\langle \textit{relační-výraz}_1 \rangle,\ \langle \textit{relační-výraz}_2 \rangle, \ldots\}$

SQL:

`SELECT * FROM` $\langle \textit{jméno}_1 \rangle$, $\langle \textit{jméno}_2 \rangle$

`SELECT * FROM` $\langle \textit{jméno}_1 \rangle$ `CROSS JOIN` $\langle \textit{jméno}_2 \rangle$

poznámky:

- `TIMES` je ekvivalentní `JOIN`, ale testuje disjunktnost schémat
- SQL povolí provést i nad schématy se společnými atributy (výsledek obsahuje více sloupců stejných jmen, lze odstranit přejmenováním)
- obecně lze přejmenováním atributů „vynutit kartézský součin“

Příklad (Tutorial D: Kartézský součin)

```
VAR rel1 BASE
  RELATION {x CHAR, y INTEGER}
  KEY {x, y};

VAR rel2 BASE
  RELATION {y INTEGER, z CHAR}
  KEY {y, z}; ···

/* using JOIN vs. TIMES */
rel1 JOIN rel2   ⟾ ···
rel1 TIMES rel2  /* error, no disjoint schemes */

/* forced cross join by renaming */
(rel1 RENAME {y AS w}) JOIN rel2   ⟾ ···
(rel1 RENAME {y AS w}) TIMES rel2  ⟾ ···
```

Příklad (SQL: Kartézský součin)

```
CREATE TABLE rel1 (
  x VARCHAR NOT NULL,
  y NUMERIC NOT NULL,
  PRIMARY KEY (x, y));

CREATE TABLE rel2 (
  x VARCHAR NOT NULL,
  y NUMERIC NOT NULL,
  PRIMARY KEY (x, y));

SELECT * FROM rel1, rel2;
SELECT rel1.x AS x1, rel1.y AS y1, FROM rel1, rel2;

SELECT * FROM rel1, rel2 WHERE y = ···; /* error */
SELECT x FROM rel1, rel2;               /* error */
SELECT y FROM rel1, rel2;               /* error */
```

Kartézský součin: Tři různé pojmy

pojem „kartézský součin“ používáme ve třech různých významech:

1. **naivní kartézský součin** (dvou nebo více množin v daném pořadí)
   - značení: $A \times B$, $A_1 \times \cdots \times A_n$
   - formalizace: množina všech *uspořádaných* dvojic (n-tic) prvků z daných množin
   - použití: pro zavedení pojmu *zobrazení*
2. **obecný kartézský součin** (indexovaného systému množin)
   - značení: $\prod_{i \in I} A_i$
   - formalizace: množina všech zobrazení $f : I \to \bigcup_{i \in I} A_i$ kde $f(i) \in A_i$ $(i \in I)$
   - použití: pro zavedení pojmu *relace nad relačním schématem*
3. **kartézský součin relací nad relačními schématy** (operace s relacemi)
   - značení: $\mathcal{D}_1 \bowtie \mathcal{D}_2$
   - formalizace: *přirozené spojení* relací nad disjunktními schématy
   - použití: relační dotazování

zřejmě: $\mathcal{D}_1 \times \mathcal{D}_2 \neq \mathcal{D}_1 \bowtie \mathcal{D}_2$ (neplést, **!!**)

Speciální případy: Polospojení

Definice (polospojení, angl.: *semijoin*)

Mějme relace $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech R_1 a R_2. Položme:

$$\mathcal{D}_1 \ltimes \mathcal{D}_2 = \pi_{R_1}(\mathcal{D}_1 \bowtie \mathcal{D}_2).$$

Relace $\mathcal{D}_1 \ltimes \mathcal{D}_2$ se nazývá **polospojení $\mathcal{D}_1$ a $\mathcal{D}_2$ (v tomto pořadí)**.

Tutorial D:

$\langle \mathit{relační\text{-}výraz}_1 \rangle$ `MATCHING` $\langle \mathit{relační\text{-}výraz}_2 \rangle$

SQL:

`SELECT DISTINCT` $\langle \mathit{jméno}_1 \rangle$`.*` `FROM` $\langle \mathit{jméno}_1 \rangle$ `NATURAL JOIN` $\langle \mathit{jméno}_2 \rangle$

význam:

- $r_1 \in \mathcal{D}_1 \ltimes \mathcal{D}_2$ právě tehdy, když $r_1 \in \mathcal{D}_1$ a r_1 je spojitelná s nějakou $r_2 \in \mathcal{D}_2$

Věta (Polospojení jako speciální případ spojení)

Mějme relace $\mathcal{D}_1$ *nad schématem* $R \cup S$ *a* $\mathcal{D}_2$ *nad schématem* $S \cup T$ *tak, že* $R \cap T = \emptyset$*. Pak platí, že* $\mathcal{D}_1 \ltimes \mathcal{D}_2 = \mathcal{D}_1 \bowtie \pi_S(\mathcal{D}_2)$*. Pokud navíc máme* $T = \emptyset$*, pak platí* $\mathcal{D}_1 \ltimes \mathcal{D}_2 = \mathcal{D}_1 \bowtie \mathcal{D}_2$ *Duálně,* $\mathcal{D}_2 \ltimes \mathcal{D}_1 = \mathcal{D}_1 \bowtie \mathcal{D}_2$ *pokud* $R = \emptyset$*.*

Důkaz.

Pro jednoduchost předpokládejme, že $R \cap S = S \cap T = \emptyset$. Rozepsáním polospojení podle definice projekce a spojení dostáváme:

$$
\begin{aligned}
\mathcal{D}_1 \ltimes \mathcal{D}_2 &= \pi_{R \cup S}(\mathcal{D}_1 \bowtie \mathcal{D}_2) \\
&= \{rs \,|\, \text{existuje } t \in \textstyle\prod_{y \in T} D_y \text{ tak, že } rst \in \mathcal{D}_1 \bowtie \mathcal{D}_2\} \\
&= \{rs \,|\, \text{existuje } t \in \textstyle\prod_{y \in T} D_y \text{ tak, že } rs \in \mathcal{D}_1 \text{ a } st \in \mathcal{D}_2\} \\
&= \{rs \in \mathcal{D}_1 \,|\, \text{existuje } t \in \textstyle\prod_{y \in T} D_y \text{ tak, že } st \in \mathcal{D}_2\} \\
&= \{rs \in \mathcal{D}_1 \,|\, s \in \pi_S(\mathcal{D}_2)\} = \mathcal{D}_1 \bowtie \pi_S(\mathcal{D}_2).
\end{aligned}
$$

Druhé tvrzení plyne přímo z prvního a faktu, že $\pi_S(\mathcal{D}) = \mathcal{D}$ pro každou relaci $\mathcal{D}$ na schématu S (Přednáška 3). □

Příklad (Příklady polospojení)

FOO	BAR	BAZ
444	ghi	103
555	def	102
555	ghi	103
666	abc	101

$\ltimes$

BAR	BAZ	QUX
abc	111	zzz
def	102	www
def	102	yyy
ghX	103	xxx
ghi	103	ttt
ghi	103	uuu
ghi	103	vvv

$=$

FOO	BAR	BAZ
444	ghi	103
555	def	102
555	ghi	103

BAR	BAZ	QUX
abc	111	zzz
def	102	www
def	102	yyy
ghX	103	xxx
ghi	103	ttt
ghi	103	uuu
ghi	103	vvv

$\ltimes$

FOO	BAR	BAZ
444	ghi	103
555	def	102
555	ghi	103
666	abc	101

$=$

BAR	BAZ	QUX
def	102	www
def	102	yyy
ghi	103	ttt
ghi	103	uuu
ghi	103	vvv

Příklad (Tutorial D: Příklady polospojení)

```
VAR rel1 BASE
  RELATION {foo INTEGER, bar CHAR, baz INTEGER}
  KEY {foo, bar, baz};

VAR rel2 BASE
  RELATION {bar CHAR, baz INTEGER, qux CHAR}
  KEY {bar, baz, qux}; ···

rel1 MATCHING rel2
rel1 JOIN rel2 {bar, baz}
(rel1 JOIN rel2) {foo, bar, baz}

rel2 MATCHING rel1
rel1 {bar, baz} JOIN rel2
(rel1 JOIN rel2) {bar, baz, qux}
```

Příklad (SQL: Příklady polospojení)

```
CREATE TABLE rel1 (
  foo NUMERIC NOT NULL,
  bar VARCHAR NOT NULL,
  baz NUMERIC NOT NULL,
  PRIMARY KEY (foo, bar, baz));

CREATE TABLE rel2 (
  bar VARCHAR NOT NULL,
  baz NUMERIC NOT NULL,
  qux VARCHAR NOT NULL,
  PRIMARY KEY (bar, baz, qux)); ···

SELECT DISTINCT rel1.* FROM rel1 NATURAL JOIN rel2;

SELECT DISTINCT rel2.* FROM rel1 NATURAL JOIN rel2;
```

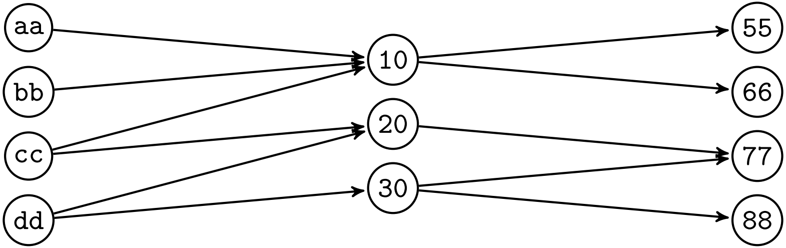
Intermezzo: Kompozice binárních relací

pojem související s binárními relacemi:

- **skládání (kompozice) binárních relací** – mějme binární relace $R_1 \subseteq A \times B$ a $R_2 \subseteq B \times C$ (množiny uspořádaných dvojic). Pak kompozicí $R_1 \circ R_2$ relací R_1 a R_2 (v tomto pořadí) rozumíme binární relaci $R_1 \circ R_2 \subseteq A \times C$ definovanou:

$$R_1 \circ R_2 = \{\langle a, c \rangle \in A \times C \mid \text{ existuje } b \in B \text{ tak, že } \langle a, b \rangle \in R_1 \text{ a } \langle b, c \rangle \in R_2\}.$$

- skládání hran v grafu, násobení Booleovských matic (sousednosti), ...



Intermezzo: Kompozice n-tic

motivace:

Lze zavést pojem „skládání relací nad relačními schématy" tak, aby souvisel s pojmem skládání binárních relací (formalizovaných jako podmnožiny naivních kartézských součinů dvou množin)?

složení (kompozice) n-tic, angl.: *tuple composition*

Mějme n-tice $r \in \prod_{y \in R} D_y$ a $s \in \prod_{y \in S} D_y$, které jsou spojitelné. Pak zobrazení $(r \cup s)\big((R \setminus S) \cup (S \setminus R)\big)$ nazveme složení (kompozice) n-tic r a s.

Tutorial D:

$\langle n\text{-}ticov\acute{y}\text{-}v\acute{y}raz_1 \rangle$ `COMPOSE` $\langle n\text{-}ticov\acute{y}\text{-}v\acute{y}raz_2 \rangle$

vlastnosti:

- skládání n-tic *je komutativní* (výsledek spojení a projekce n-tic)

Odvozené operace: Kompozice

Definice (složení (kompozice), angl.: *composition*)

Mějme relace $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech $R \cup S$ a $S \cup T$ tak, že $R \cap T = \emptyset$. Položme:

$$\mathcal{D}_1 \circ \mathcal{D}_2 = \pi_{R \cup T}(\mathcal{D}_1 \bowtie \mathcal{D}_2).$$

Relace $\mathcal{D}_1 \circ \mathcal{D}_2$ na $R \cup T$ se nazývá **složení** (**kompozice**) **relací $\mathcal{D}_1$ a $\mathcal{D}_2$**.

Tutorial D:

$\langle \textit{relační-výraz}_1 \rangle$ `COMPOSE` $\langle \textit{relační-výraz}_2 \rangle$

`COMPOSE` $\{\langle \textit{relační-výraz}_1 \rangle, \langle \textit{relační-výraz}_2 \rangle, \ldots\}$

SQL:

$\texttt{SELECT DISTINCT}\ \langle \textit{jméno}_1 \rangle.\langle \textit{R-atribut}_1 \rangle, \ldots, \langle \textit{jméno}_1 \rangle.\langle \textit{R-atribut}_m \rangle,$
$\langle \textit{jméno}_2 \rangle.\langle \textit{T-atribut}_1 \rangle, \ldots, \langle \textit{jméno}_2 \rangle.\langle \textit{T-atribut}_n \rangle$
$\texttt{FROM}\ \langle \textit{jméno}_1 \rangle\ \texttt{NATURAL JOIN}\ \langle \textit{jméno}_2 \rangle$

Příklad (Tutorial D: Kompozice relací nad schématy)

```
VAR rel1 BASE
  INIT (RELATION {
    TUPLE {x "aa", y 10}, TUPLE {x "bb", y 10},
    TUPLE {x "cc", y 10}, TUPLE {x "cc", y 20},
    TUPLE {x "dd", y 20}, TUPLE {x "dd", y 30}})
  KEY {x, y};

VAR rel2 BASE
  INIT (RELATION {
    TUPLE {y 10, z 55}, TUPLE {y 10, z 66},
    TUPLE {y 20, z 77}, TUPLE {y 30, z 77},
    TUPLE {y 30, z 88}})
  KEY {y, z};

rel1 COMPOSE rel2          ⟾ ···
(rel1 JOIN rel2) {x, z}    ⟾ ···
```

Příklad (SQL: Kompozice relací nad schématy)

```
CREATE TABLE rel1 (
  x VARCHAR NOT NULL, y NUMERIC NOT NULL,
  PRIMARY KEY (x, y));

INSERT INTO rel1 VALUES
  ('aa', 10), ('bb', 10), ('cc', 10),
  ('cc', 20), ('dd', 20), ('dd', 30);

CREATE TABLE rel2 (
  y NUMERIC NOT NULL, z NUMERIC NOT NULL,
  PRIMARY KEY (y, z));

INSERT INTO rel2 VALUES
  (10, 55), (10, 66), (20, 77), (30, 77), (30, 88);

SELECT DISTINCT x, z FROM rel1 NATURAL JOIN rel2;
```

Aktivní domény a komplementy

motivace:

Kartézský součin $\prod_{y \in R} D_y$ je obecně nekonečný, jak můžeme uvažovat „komplement“ relace $\mathcal{D} \subseteq \prod_{y \in R} D_y$ nad relačním schématem?

důsledky:

- nelze chápat jako $\prod_{y \in R} D_y \setminus \mathcal{D}$
- nutno zajistit, aby byl „komplement“ konečný
- řešení pomocí aktivních domén:

aktivní doména, **komplement**, angl.: *active domain, complement*

Aktivní doména $y \in R$ v relaci $\mathcal{D}$ nad R je množina $A_y^{\mathcal{D}} = \{r(y) \,|\, r \in \mathcal{D}\} \subseteq D_y$.
Komplement relace $\mathcal{D}$ je relace $\overline{\mathcal{D}} = \bowtie_{y \in R} \pi_{\{y\}}(\mathcal{D}) \setminus \mathcal{D}$.

Příklad (Tutorial D: Komplement)

```
VAR rt BASE
  INIT (RELATION {
    TUPLE {foo 666, bar "abc", baz 101},
    TUPLE {foo 555, bar "def", baz 102},
    TUPLE {foo 555, bar "ghi", baz 103}})
  KEY {foo, bar, baz};

/* cartesian product or relations representing active domains */
JOIN {rt {foo}, rt {bar}, rt {baz}}   ⟾ ···
TIMES {rt {foo}, rt {bar}, rt {baz}}  ⟾ ···

/* complement using set difference */
JOIN {rt {foo}, rt {bar}, rt {baz}} MINUS rt  ⟾ ···
TIMES {rt {foo}, rt {bar}, rt {baz}} MINUS rt ⟾ ···
```

Příklad (SQL: Komplement)

```
CREATE TABLE tab (
  foo NUMERIC NOT NULL,
  bar VARCHAR NOT NULL,
  baz NUMERIC NOT NULL,
  PRIMARY KEY (foo, bar, baz));

/* cartesian product or relations representing active domains */
SELECT DISTINCT r1.foo, r2.bar, r3.baz
  FROM tab AS r1, tab AS r2, tab AS r3;

/* complement using set difference */
SELECT DISTINCT r1.foo, r2.bar, r3.baz
  FROM tab AS r1, tab AS r2, tab AS r3
  EXCEPT SELECT * from tab;
```

poznámka: použití AS není přejmenování (atributu), ale *pojmenování* (tabulky, **!!**)

Problémy fyzické vrstvy: Otázky efektivity

otázka:

Jak efektivně počítat výsledky přirozených spojení?

možnosti vyhodnocování $\mathcal{D}_1 \bowtie \mathcal{D}_2$:

- naivní iterace ve vnořené smyčce (pro kartézský součin vždy)
- využití indexů (efektivní pokud je $|\mathcal{D}_1 \bowtie \mathcal{D}_2| \lll |\mathcal{D}_1| \cdot |\mathcal{D}_2|$), například:
  - iterace přes prvky $\mathcal{D}_1$ a dohledávání (pomocí indexu) spojitelných n-tic z $\mathcal{D}_2$
  - spojení *sléváním* (merge) $\mathcal{D}_1$ a $\mathcal{D}_2$
    (pokud mají obě na S indexy reprezentující uspořádané množiny)
  - lze urychlit ještě víc, pokud S (množina společných atributů relací $\mathcal{D}_1$ a $\mathcal{D}_2$) obsahuje klíč (nebo unikátní index)

obecné doporučení:

- vytvářet indexy pro množiny atributů, přes které se spojuje
- PostgreSQL: používat `EXPLAIN` (důležité v případě velkých dat)

Přednáška 4: Závěr

pojmy k zapamatování:

- přirozené spojení
- speciální případy: průnik, restrikce na rovnost
- odvozené operace: kartézský součin, kompozice, polospojení
- aktivní domény a komplementace

použité zdroje:

- Date C. J.: *Database in Depth: Relational Theory for Practitioners*
  O'Reilly Media 2005, ISBN 978–0596100124
- Date C. J., Darwen H.: *Databases, Types and the Relational Model*
  Addison Wesley 2006, ISBN 978–0321399427
- Maier D: *Theory of Relational Databases*
  Computer Science Press 1983, ISBN 978–0914894421