Databázové systémy

Ostatní typy spojení

Vilém Vychodil

KMI/DATA1, Přednáška 5

Databázové systémy

Přednáška 5: Přehled

1. Vlastnosti přirozeného spojení:
   - úplné spojení a bezeztrátová dekompozice,
   - vztah přirozeného spojení a dalších operací,
   - θ-spojení, spojení na rovnost.
2. Související problémy:
   - kompozice a tranzitivní uzávěr,
   - rekurzivní dotazy.
3. Problematika vnějších spojení:
   - definice, úskalí, vztah k relačnímu modelu,
   - formalizace pomocí tříhodnotových logik,
   - jiné možnosti přístupu k problému.

Věta (Další vlastnosti přirozeného spojení)

Mějme $\mathcal{D}_1$, $\mathcal{D}_2$ a $\mathcal{D}_3$ na schématech R_1, R_2 a $R_3 = R_2$. Pak platí:

1. $\bowtie$ *je isotonní;*
2. $\sigma_{y=d}(\mathcal{D}_1 \bowtie \mathcal{D}_2) = \sigma_{y=d}(\mathcal{D}_1) \bowtie \sigma_{y=d}(\mathcal{D}_2)$ *pokud* $y \in R_1 \cap R_2$*;*
3. $\sigma_{y=d}(\mathcal{D}_1 \bowtie \mathcal{D}_2) = \sigma_{y=d}(\mathcal{D}_1) \bowtie \mathcal{D}_2$ *pokud* $y \in R_1$ *a* $y \notin R_2$*;*
4. $\mathcal{D}_1 \bowtie (\mathcal{D}_2 \cup \mathcal{D}_3) = (\mathcal{D}_1 \bowtie \mathcal{D}_2) \cup (\mathcal{D}_1 \bowtie \mathcal{D}_3)$*;*
5. $\mathcal{D}_1 \bowtie (\mathcal{D}_2 \cap \mathcal{D}_3) = (\mathcal{D}_1 \bowtie \mathcal{D}_2) \cap (\mathcal{D}_1 \bowtie \mathcal{D}_3) = \mathcal{D}_1 \bowtie \mathcal{D}_2 \bowtie \mathcal{D}_3$*;*
6. $\mathcal{D}_1 \bowtie (\mathcal{D}_2 \setminus \mathcal{D}_3) = (\mathcal{D}_1 \bowtie \mathcal{D}_2) \setminus (\mathcal{D}_1 \bowtie \mathcal{D}_3)$.

Důkaz.

1 plyne z isotonie konjunkce; 2 je zřejmé; 3 plyne analogicky jako předchozí bod; 4 je důsledkem distributivity konjunkce a disjunkce; 5 plyne z idempotence konjunkce; 6 platí, protože $rst \in \mathcal{D}_1 \bowtie (\mathcal{D}_2 \setminus \mathcal{D}_3)$ p. k. $rs \in \mathcal{D}_1$ a $st \in \mathcal{D}_2 \setminus \mathcal{D}_3$ p. k. $rs \in \mathcal{D}_1$, $st \in \mathcal{D}_2$ a $st \notin \mathcal{D}_3$ p. k. $rs \in \mathcal{D}_1$, $st \in \mathcal{D}_2$ a $rs \in \mathcal{D}_1$, $st \notin \mathcal{D}_3$ p. k. $rst \in \mathcal{D}_1 \bowtie \mathcal{D}_2$ a $rst \notin \mathcal{D}_1 \bowtie \mathcal{D}_3$ p. k. $rst \in (\mathcal{D}_1 \bowtie \mathcal{D}_2) \setminus (\mathcal{D}_1 \bowtie \mathcal{D}_3)$. $\square$

Úplné spojení

motivace:

V některých spojeních jsou všechny n-tice z výchozích tabulek spojitelné s některými n-ticemi z ostatních tabulek – vede na pojem úplné spojení

nespojitelná n-tice, angl.: *dangling tuple*

Mějme relace $\mathcal{D}_1, \ldots, \mathcal{D}_2$ na schématech $R_1, \ldots, R_n$. Pak n-tice $r_i \in \mathcal{D}_i$ se nazývá nespojitelná vzhledem k $\bowtie_{i=1}^{n} \mathcal{D}_i$ pokud neexistují žádné $r_j \in \mathcal{D}_j$ $(j \neq i)$ tak, že $r_1 \cup \cdots \cup r_n \in \bowtie_{i=1}^{n} \mathcal{D}_i$.

úplné spojení, angl.: *complete join*

Relace $\mathcal{D}_1, \ldots, \mathcal{D}_2$ na schématech $R_1, \ldots, R_n$ mají úplné spojení, pokud žádná $\mathcal{D}_i$ neobsahuje nespojitelnou n-tici vzhledem k $\bowtie_{i=1}^{n} \mathcal{D}_i$.

poznámka: $\mathcal{D}_1$ a $\mathcal{D}_2$ mají úplné spojení p. k. $\mathcal{D}_1 \ltimes \mathcal{D}_2 = \mathcal{D}_1$ a $\mathcal{D}_2 \ltimes \mathcal{D}_1 = \mathcal{D}_2$.

Věta (Charakterizace úplných spojení)

Mějme relace $\mathcal{D}_1, \ldots, \mathcal{D}_n$ na schématech $R_1, \ldots, R_n$. Pak platí:

1. $\pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right) \subseteq \mathcal{D}_j$ *pro každé* $j = 1, \ldots, n$;
2. $\pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right) = \mathcal{D}_j$ *pro každé* $j = 1, \ldots, n$ *p. k.* $\mathcal{D}_1, \ldots, \mathcal{D}_n$ *lze úplně spojit;*
3. $\bowtie_{i=1}^{n} \mathcal{D}_i = \bowtie_{j=1}^{n} \pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right)$ *pro každé* $j = 1, \ldots, n$;
4. $\pi_{R_j}\left(\bowtie_{k=1}^{n} \pi_{R_k}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right)\right) = \pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right)$ *pro každé* $j = 1, \ldots, n$.

Důkaz.

Bod 1 plyne z toho, že pokud $r \in \bowtie_{i=1}^{n} \mathcal{D}_i$, pak $r(R_j) \in \mathcal{D}_j$. V případě 2 ukážeme obměnou: $\mathcal{D}_1, \ldots, \mathcal{D}_n$ nelze úplně spojit pokud existuje $r_j \in \mathcal{D}_j$, která je nespojitelná vzhledem k $\bowtie_{i=1}^{n} \mathcal{D}_i$; to je p. k. $r(R_j) \neq r_j$ pro každou $r \in \bowtie_{i=1}^{n} \mathcal{D}_i$ a to je p. k. $r_j \notin \pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right)$ p. k. existuje j tak, že $\pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right) \subset \mathcal{D}_j$. Pro dokázání inkluze „$\subseteq$“ bodu 3 vyjdeme z toho, že pokud $r \in \bowtie_{i=1}^{n} \mathcal{D}_i$, pak $r = r_1 \cup \cdots \cup r_n$, kde $r(R_j) = r_j$ pro každé j. To jest $r_j \in \pi_{R_j}(\bowtie_{i=1}^{n} \mathcal{D}_i)$ a tedy $r \in \bowtie_{j=1}^{n} \pi_{R_j}\left(\bowtie_{i=1}^{n} \mathcal{D}_i\right)$. Opačná inkluze plyne z 1 a isotonie $\bowtie$. Bod 4 je triviální důsledek 3. □

Příklad (Úplné spojení)

relace, které nemají úplné spojení:

$\mathcal{D}_1$

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

$\mathcal{D}_2$

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

$\mathcal{D}_1 \bowtie \mathcal{D}_2$

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

$\mathcal{D}_1 \ltimes \mathcal{D}_2$

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

$\mathcal{D}_2 \ltimes \mathcal{D}_1$

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

dle předchozího tvrzení:

- relace $\mathcal{D}_3 = \mathcal{D}_1 \ltimes \mathcal{D}_2$ a $\mathcal{D}_4 = \mathcal{D}_2 \ltimes \mathcal{D}_1$ mají úplné spojení $\mathcal{D}_1 \bowtie \mathcal{D}_2 = \mathcal{D}_3 \bowtie \mathcal{D}_4$

Bezeztrátová dekompozice relace

motivace:

Duální pojem k pojmu úplné spojení: Je možné vyjádřit výchozí relace pomocí spojení některých jejich projekcí?

bezeztrátová dekompozice, angl.: *nonloss decomposition*

Relace $\mathcal{D}$ na schématu $R_1 \cup \cdots \cup R_n$ má bezeztrátovou dekompozici vzhledem k množině schémat $\{R_1, \ldots, R_n\}$ pokud $\mathcal{D} = \bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D})$.

existence dekompozic:

- každá $\mathcal{D}$ na R má (jednoprvkovou) bezeztrátovou dekompozici vzhledem k $\{R\}$ (triviální případ; hledáme dekompozice vzhledem k větším množinám schémat)

další typy dekompozic:

- horizontální/vertikální typy dekompozic, faktorové dekompozice, . . .

Věta (Vlastnosti bezeztrátových dekompozic)

Mějme relaci $\mathcal{D}$ na schématu $R_1 \cup \cdots \cup R_n$. Pak platí

1. $\mathcal{D} \subseteq \bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D})$;
2. $\bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D}) = \bowtie_{j=1}^{n} \pi_{R_j}\left(\bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D})\right)$.

Důkaz.

Pro dokázání 1 vezměme $r \in \mathcal{D}$. Vzhledem ke schématům $R_1, \ldots, R_n$ lze r chápat jako n-tici ve tvaru $r_1 \cup \cdots \cup r_n$, kde $r_i = r(R_i)$ pro každé $i = 1, \ldots, n$. Zřejmě tedy $r_i \in \pi_{R_i}(\mathcal{D})$ pro každé $i = 1, \ldots, n$ a tím pádem $r \in \bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D})$.

Tvrzení 2 je speciálním případem bodu 3 předchozí věty pro $\mathcal{D}_i = \pi_{R_i}(\mathcal{D})$. $\square$

důsledek:

- $\bowtie_{i=1}^{n} \pi_{R_i}(\mathcal{D})$ má bezeztrátovou dekompozici vzhledem k $\{R_1, \ldots, R_n\}$

Příklad (Bezeztrátová dekompozice)

$\mathcal{D}$

FOO	BAR	BAZ	QUX
100	aaa	444	GGG
100	aaa	444	HHH
200	bbb	555	III
300	ccc	555	III

$\pi_{\{\mathtt{FOO},\mathtt{BAR},\mathtt{BAZ}\}}(\mathcal{D})$

FOO	BAR	BAZ
100	aaa	444
200	bbb	555
300	ccc	555

$\pi_{\{\mathtt{BAZ},\mathtt{QUX}\}}(\mathcal{D})$

BAZ	QUX
444	GGG
444	HHH
555	III

$\pi_{\{\mathtt{BAR},\mathtt{QUX}\}}(\mathcal{D})$

BAR	QUX
aaa	GGG
aaa	HHH
bbb	III
ccc	III

$\pi_{\{\mathtt{FOO},\mathtt{BAR}\}}(\mathcal{D})$

FOO	BAR
100	aaa
200	bbb
300	ccc

platí například:

$$\mathcal{D} = \pi_{\{\mathtt{FOO},\mathtt{BAR},\mathtt{BAZ}\}}(\mathcal{D}) \bowtie \pi_{\{\mathtt{BAZ},\mathtt{QUX}\}}(\mathcal{D})$$
$$\mathcal{D} = \pi_{\{\mathtt{FOO},\mathtt{BAR},\mathtt{BAZ}\}}(\mathcal{D}) \bowtie \pi_{\{\mathtt{BAR},\mathtt{QUX}\}}(\mathcal{D})$$
$$\mathcal{D} = \pi_{\{\mathtt{FOO},\mathtt{BAR},\mathtt{BAZ}\}}(\mathcal{D}) \bowtie \pi_{\{\mathtt{FOO},\mathtt{BAR}\}}(\mathcal{D})$$

ale například: $\mathcal{D} \neq \pi_{\{\mathtt{FOO},\mathtt{BAR}\}}(\mathcal{D}) \bowtie \pi_{\{\mathtt{BAZ},\mathtt{QUX}\}}(\mathcal{D})$

poznámky (k tomtuto konkrétnímu příkladu):

- neexistuje dekompozice na dvě dvouprvková schémata
- celkem $60,672$ možných dekompozic (složených z vzájemně různých schémat)

Odvozené operace: θ-spojení

motivace:

Chceme spojovat data ze dvou tabulek tak, aby kritérium spojitelnosti nebylo dáno pouze rovností na určitých hodnotách, ale obecnou podmínkou vztahující se na hodnoty n-tic z obou tabulek.

Definice (θ-spojení, angl.: *θ-join*)

Mějme relace $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech R a T takových, že $R \cap T = \emptyset$ a nechť θ je skalární výraz typu „pravdivostní hodnota“, který může obsahovat jména atributů z $R \cup T$. Pak **θ-spojení $\mathcal{D}_1$ a $\mathcal{D}_2$ splňující θ** je relace $\sigma_\theta(\mathcal{D}_1 \bowtie \mathcal{D}_2)$ a označujeme ji $\mathcal{D}_1 \bowtie_\theta \mathcal{D}_2$.

poznámky:

- $\bowtie_\theta$ je definováno jako *restrikce* z *kartézského součinu*
- podle definice σ_θ a $\bowtie$:

$$\mathcal{D}_1 \bowtie_\theta \mathcal{D}_2 = \{rt \,|\, r \in \mathcal{D}_1 \text{ a } r \in \mathcal{D}_2 \text{ tak, že } rt \text{ splňuje } \theta\}.$$

Odvozené operace: Spojení na rovnost

motivace:

Speciální případ θ-spojení, ve kterém je podmínka θ formulována jako konjunkce podmínek vyjadřující rovnost hodnot atributů stejných typů.

Definice (spojení na rovnost, angl.: *equijoin*)

Mějme relace $\mathcal{D}_1$ a $\mathcal{D}_2$ na schématech R a T takových, že $R \cap T = \emptyset$ a nechť $\{y_1, \ldots, y_n\} \subseteq R$, $\{y'_1, \ldots, y'_n\} \subseteq T$ tak, že atributy mají y_i a y'_i stejný typ. Pak θ-spojení $\mathcal{D}_1$ a $\mathcal{D}_2$, kde θ je va tvaru $y_1 = y'_1 \curlywedge \cdots \curlywedge y_n = y'_n$ nazýváme **spojení $\mathcal{D}_1$ a $\mathcal{D}_2$ na rovnost** (atributů y_1, y'_1 až y_n, y'_n) a označujeme jej $\mathcal{D}_1 \bowtie_{y_1 = y'_1, \ldots, y_n = y'_n} \mathcal{D}_2$.

Tutorial D:

($\langle$*relační-výraz*$_1\rangle$ `JOIN` $\langle$*relační-výraz*$_2\rangle$) `WHERE` $\langle$*podmínka*$\rangle$

SQL:

`SELECT * FROM` $\langle$*jméno*$_1\rangle$, $\langle$*jméno*$_2\rangle$ `WHERE` $\langle$*podmínka*$\rangle$

Vzájemná zastupitelnost operací

pozorování:

Spojení na rovnost lze vyjádřit pomocí přirozeného spojení a restrikce, protože na disjunktních schématech přechází přirozené spojení v kartézský součin.

otázka:

Lze naopak vyjádřit přirozené spojení pomocí spojení na rovnost?

úvaha:

Přirozené spojení lze chápat jako spojení na rovnost, ve kterém nejprve vhodně přejmenujeme atributy a provedeme dodatečnou projekci.

Pro relace $\mathcal{D}_1$ a $\mathcal{D}_2$ nad schématy $R \cup S$ a $S \cup T$ tak, že $R \cap T = \emptyset$ a $S = \{y_1, \dots, y_k\}$, s použitím přejměnování $y_1, \dots, y_k$ na $y'_1, \dots, y'_k$ máme:

$$\begin{aligned}
\mathcal{D}_1 \bowtie \mathcal{D}_2 &= \pi_{R \cup S \cup T}\big(\mathcal{D}_1 \bowtie_{y_1 = y'_1, \dots, y_k = y'_k} \rho_{y'_1 \leftarrow y_1, \dots, y'_k \leftarrow y_k}(\mathcal{D}_2)\big) \\
&= \pi_{R \cup S \cup T}\Big(\sigma_{y_1 = y'_1, \dots, y_k = y'_k}\big(\mathcal{D}_1 \bowtie \rho_{y'_1 \leftarrow y_1, \dots, y'_k \leftarrow y_k}(\mathcal{D}_2)\big)\Big).
\end{aligned}$$

Obecné formule pro restrikce na rovnost

motivace:

Ve spojení na rovnost je θ vyjádřená jako konjunkce atomických formulí (tzv. identit). Snadno lze úvahy rozšířit na libovolně složité formule konstruované z identit a výrokových spojek.

Definice (formule definující podmínky pro restrikce)

Mějme relační schéma R obsahující atributy $y \in R$ typů D_y. Pak **formule** pro restrikce nad R definujeme takto:

1. pokud jsou $y_1, y_2 \in R$, pak $y_1 = y_2$ je (atomická) formule;
2. pokud je $y \in R$ a $d \in D_y$, pak $y = d$ je (atomická) formule;
3. pokud jsou φ a ψ formule, pak jsou $\neg\varphi$ a $(\varphi \Rightarrow \psi)$ formule.

- přijímáme konvenci o vynechávání vnějších závorek formulí
- $(\varphi \Leftrightarrow \psi)$ $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ chápeme jako zkratky (běžným způsobem)

Věta (Rozšíření spojení na rovnost)

Nechť R je relační schéma a θ je formule pro restrikci nad R. Pak existuje posloupnost relačních operací zahrnující pouze restrikce na rovnost, sjednocení a rozdíl tak, že pro každou $\mathcal{D}$ nad R je výsledek aplikace těchto operací roven $\sigma_\theta(\mathcal{D})$.

Důkaz.

Nejprve nahládněme, že podle předchozí definice lze θ chápat jako formuli, ve které se vyskytují pouze spojky $\neg$ a $\Rightarrow$. Použitím známého faktu, že $\varphi \Rightarrow \psi$ je sémanticky ekvivalentní $\neg\varphi \vee \psi$, můžeme θ chápat jako formuli obsahující $\neg$ a $\vee$ jako základní spojky. Dále můžeme využít faktu, že pro každou $\mathcal{D}$ nad R zřejmě platí následující

$$\begin{aligned}\sigma_{\neg\theta}(\mathcal{D}) &= \{r \in \mathcal{D} \,|\, r \text{ nesplňuje } \theta\} = \mathcal{D} \setminus \sigma_\theta(\mathcal{D}),\\ \sigma_{\theta_1 \vee \theta_2}(\mathcal{D}) &= \{r \in \mathcal{D} \,|\, r \text{ splňuje } \theta_1 \text{ nebo } r \text{ splňuje } \theta_2\} = \sigma_{\theta_1}(\mathcal{D}) \cup \sigma_{\theta_2}(\mathcal{D}).\end{aligned}$$

To jest, $\sigma_\theta(\mathcal{D})$ pro libovolně složitou θ lze vyjádřit konečně mnoha aplikacemi $\setminus$ a $\cup$ pouze za použití výchozí relace $\mathcal{D}$. □

poznámka: spojení na rovnost je „dost obecná“ relační operace

Intermezzo: Tranzitivní uzávěr

tranzitivní uzávěr relace na množině, angl.: *transitive closure*

Mějme binární relaci $R \subseteq A \times A$. Pak tranzitivním uzávěrem R, rozumíme binární relací $R^\infty \subseteq A \times A$ splňující následující podmínky:

1. $R \subseteq R^\infty$,
2. R^∞ je tranzitivní,
3. R^∞ je nejmenší relace splňující 1 a 2.

konstruktivně lze R^∞ vyjádřit jako

$$R^\infty = \textstyle\bigcup_{n=1}^{\infty} R^n = \bigcup_{n=1}^{\infty} \underbrace{R \circ \cdots \circ R}_{n\text{-krát}},$$

přitom platí:

- pokud je R konečná, pak existuje index m tak, že $R^\infty = \bigcup_{n=1}^{m} R^n$
- pokud je navíc R reflexivní, pak $R^\infty = R^m$

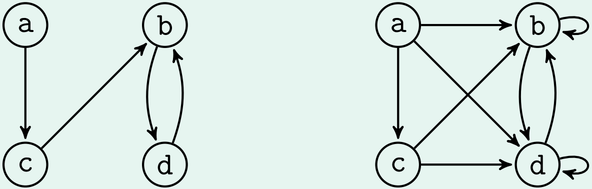
Příklad (Tranzitivní uzávěr binární relace na konečné množině)

uvažujme binární relaci $R = \{\langle \mathtt{a}, \mathtt{c}\rangle, \langle \mathtt{b}, \mathtt{d}\rangle, \langle \mathtt{c}, \mathtt{b}\rangle, \langle \mathtt{d}, \mathtt{b}\rangle\}$ na množině A

tranzitivní uzávěr relace R nalezneme takto:

$$
\begin{aligned}
R^1 &= \{\langle \mathtt{a}, \mathtt{c}\rangle, \langle \mathtt{b}, \mathtt{d}\rangle, \langle \mathtt{c}, \mathtt{b}\rangle, \langle \mathtt{d}, \mathtt{b}\rangle\}\\
R^2 &= \{\langle \mathtt{a}, \mathtt{b}\rangle, \langle \mathtt{b}, \mathtt{b}\rangle, \langle \mathtt{c}, \mathtt{d}\rangle, \langle \mathtt{d}, \mathtt{d}\rangle\}\\
R^3 &= \{\langle \mathtt{a}, \mathtt{d}\rangle, \langle \mathtt{b}, \mathtt{d}\rangle, \langle \mathtt{c}, \mathtt{b}\rangle, \langle \mathtt{d}, \mathtt{b}\rangle\}\\
R^4 &= \{\langle \mathtt{a}, \mathtt{b}\rangle, \langle \mathtt{b}, \mathtt{b}\rangle, \langle \mathtt{c}, \mathtt{d}\rangle, \langle \mathtt{d}, \mathtt{d}\rangle\}\\
R^\infty &= \{\langle \mathtt{a}, \mathtt{b}\rangle, \langle \mathtt{a}, \mathtt{c}\rangle, \langle \mathtt{a}, \mathtt{d}\rangle, \langle \mathtt{b}, \mathtt{b}\rangle, \langle \mathtt{b}, \mathtt{d}\rangle, \langle \mathtt{c}, \mathtt{b}\rangle, \langle \mathtt{c}, \mathtt{d}\rangle, \langle \mathtt{d}, \mathtt{b}\rangle, \langle \mathtt{d}, \mathtt{d}\rangle\}
\end{aligned}
$$

graficky:



Příklad (Tranzitivní uzávěr v relačním modelu)

motivace: Binární relaci R na množině A z předchozího příkladu můžeme formalizovat jako relaci $\mathcal{D}$ nad schématem $\{x, y\}$ tak, že

$$\mathcal{D} = \big\{\{\langle x, a\rangle, \langle y, b\rangle\} \,|\, \langle a, b\rangle \in R\big\}.$$

Pak můžeme vyjádřit protějšky $\mathcal{D}^1$, $\mathcal{D}^2$, $\mathcal{D}^3$, $\mathcal{D}^4$, $\mathcal{D}^\infty$ relací R^1, R^2, R^3, R^4, R^∞:

$$\begin{aligned}
\mathcal{D}^1 &= \mathcal{D},\\
\mathcal{D}^2 &= \rho_{z\leftarrow y}(\mathcal{D}) \circ \rho_{z\leftarrow x}(\mathcal{D}) = \pi_{\{x,y\}}(\rho_{z\leftarrow y}(\mathcal{D}) \bowtie \rho_{z\leftarrow x}(\mathcal{D})),\\
\mathcal{D}^3 &= \rho_{z\leftarrow y}(\mathcal{D}) \circ \rho_{z\leftarrow x}(\mathcal{D}^2) = \pi_{\{x,y\}}(\rho_{z\leftarrow y}(\mathcal{D}) \bowtie \rho_{z\leftarrow x}(\mathcal{D}^2)),\\
\mathcal{D}^4 &= \rho_{z\leftarrow y}(\mathcal{D}) \circ \rho_{z\leftarrow x}(\mathcal{D}^3) = \pi_{\{x,y\}}(\rho_{z\leftarrow y}(\mathcal{D}) \bowtie \rho_{z\leftarrow x}(\mathcal{D}^3)),\\
\mathcal{D}^n &= \pi_{\{x,y\}}(\bowtie_{i=1}^{n} \rho_{x'_{i,n}\leftarrow x, y'_{i,n}\leftarrow y}(\mathcal{D})),\\
\mathcal{D}^\infty &= \textstyle\bigcup_{n=1}^{m} \mathcal{D}^n = \bigcup_{n=1}^{m} \pi_{\{x,y\}}(\bowtie_{i=1}^{n} \rho_{x'_{i,n}\leftarrow x, y'_{i,n}\leftarrow y}(\mathcal{D})),
\end{aligned}$$

kde $x'_{1,n} = x$ pro každé n; $y'_{n,n} = y$ pro každé n; $x'_{i,n} = y'_{i-1,n}$ pro každé $2 \leq i \leq n$.

problém: parametr m ve výrazu pro $\mathcal{D}^\infty$ závisí na konkrétní $\mathcal{D}$ (**!!**)

Tranzitivní uzávěr relací

Definice (tranzitivní uzávěr relace, angl.: *transitive closure*)

Mějme relaci $\mathcal{D}$ nad dvouprvkovým schématem $R = \{y_1, y_2\}$. Pak relace $\mathcal{D}^\infty$ nad relačním schématem R splňující následující podmínky:

1. $\mathcal{D} \subseteq \mathcal{D}^\infty$,
2. $\rho_{y \leftarrow y_1}(\mathcal{D}^\infty) \circ \rho_{y \leftarrow y_2}(\mathcal{D}^\infty) \subseteq \mathcal{D}^\infty$,
3. $\mathcal{D}^\infty$ je nejmenší relace splňující 1 a 2,

se nazývá **tranzitivní uzávěr relace $\mathcal{D}$**.

Tutorial D:

TCLOSE ($\langle$*relační-výraz*$\rangle$)

poznámky:

- SQL nepodporuje přímo jako relační operaci (ale je definovatelné)
- důležitý aspekt: ukázat *konstruktivní předpis* pro výpočet TCLOSE

Lokální pojmenování výsledků dotazů v SQL

$$
\begin{array}{l}
\texttt{WITH} \\
\quad \langle \textit{jméno}_1 \rangle \ \texttt{AS} \ (\langle \textit{dotaz}_1 \rangle), \\
\quad\quad \vdots \qquad\qquad\qquad \vdots \\
\quad \langle \textit{jméno}_n \rangle \ \texttt{AS} \ (\langle \textit{dotaz}_n \rangle) \\
\quad \langle \textit{dotaz-používající-jména} \rangle;
\end{array}
$$

$$
\begin{array}{l}
\texttt{WITH RECURSIVE} \\
\quad \langle \textit{jméno}_1 \rangle \ (\langle \textit{atribut}_{1,1} \rangle, \ \langle \textit{atribut}_{1,2} \rangle, \ldots) \ \texttt{AS} \\
\quad\quad (\langle \textit{nerekurzivní-výraz}_1 \rangle \ \texttt{UNION DISTINCT} \ \langle \textit{rekurzivní-výraz}_1 \rangle), \\
\quad\quad\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\quad \langle \textit{jméno}_n \rangle \ (\langle \textit{atribut}_{n,1} \rangle, \ \langle \textit{atribut}_{n,2} \rangle, \ldots) \ \texttt{AS} \\
\quad\quad (\langle \textit{nerekurzivní-výraz}_n \rangle \ \texttt{UNION DISTINCT} \ \langle \textit{rekurzivní-výraz}_n \rangle) \\
\quad \langle \textit{dotaz-používající-jména} \rangle;
\end{array}
$$

související pojem: „rekurzivní dotazy“ v SQL

Iterativní vyhodnocení pojmenovaných dotrazů v SQL

vstup:

- $\langle \textit{jméno}_i \rangle$ (jméno pro výsledek), $\langle atribut_{i,1} \rangle$, $\langle atribut_{i,2} \rangle, \ldots$ (jména atributů)
- $\langle \textit{nerekurzivní-výraz}_i \rangle$ (dotaz)
- $\langle \textit{rekurzivní-výraz}_i \rangle$ (předpis iterace, může obsahovat $\langle \textit{jméno}_i \rangle$ a atributy)

způsob vyhodnocení:

1. vyhodnoť $\langle \textit{nerekurzivní-výraz}_i \rangle$; výsledek ulož do $RESULT$ a $WORK$;
2. dokud $WORK \neq \emptyset$, opakuj:
   - vyhodnoť $\langle \textit{rekurzivní-výraz}_i \rangle$ v němž je každé $\langle \textit{jméno}_i \rangle$ nahrazeno obsahem tabulky $WORK$; z výsledku odstraň duplicitní řádky a každý řádek, který se již nachází v $RESULT$; výsledek ulož do $WORK$;
   - připoj obsah $WORK$ na konec $RESULT$;
3. naváž obsah $RESULT$ na $\langle \textit{jméno}_i \rangle$

poznámka: varianta s `UNION ALL` neodstraňuje duplicity

Příklad (SQL: Tranzitivní uzávěr relace)

využijeme faktu, že $\mathcal{D}^\infty = f(\mathcal{D}, \mathcal{D})$, kde

$$f(\mathcal{D}_1, \mathcal{D}_2) = \begin{cases} \mathcal{D}_2, & \text{pokud } \mathcal{D}_1 = \emptyset, \\ f(\mathcal{D}' \setminus \mathcal{D}_2, \mathcal{D}' \cup \mathcal{D}_2), & \text{pro } \mathcal{D}' = \rho_{z \leftarrow x}(\mathcal{D}_1) \circ \rho_{z \leftarrow y}(\mathcal{D}) \text{ a } \mathcal{D}_1 \neq \emptyset. \end{cases}$$

```
CREATE TABLE r (
  x NUMERIC NOT NULL,
  y NUMERIC NOT NULL,
  PRIMARY KEY (x, y)); ···

/* computing transitive closure */
WITH RECURSIVE
  tr (x, y) AS (
    SELECT * FROM r
      UNION DISTINCT
    SELECT r.x, tr.y FROM r, tr WHERE r.y = tr.x)
  SELECT * FROM tr;
```

Lokální pojmenování výsledků dotazů v Tutorial D

lokální pojmenování pro skalární výrazy (analogie `let*` z dialektů LISPu):

$$\texttt{WITH}\ (\langle \textit{proměnná}_1 \rangle\ \texttt{:=}\ \langle \textit{výraz}_1 \rangle, \ldots, \langle \textit{proměnná}_n \rangle\ \texttt{:=}\ \langle \textit{výraz}_n \rangle)\texttt{:}\ \langle \textit{skalární-výraz-používající-proměnné} \rangle$$

lokální pojmenování pro n-ticové výrazy:

$$\texttt{WITH}\ (\langle \textit{proměnná}_1 \rangle\ \texttt{:=}\ \langle \textit{výraz}_1 \rangle, \ldots, \langle \textit{proměnná}_n \rangle\ \texttt{:=}\ \langle \textit{výraz}_n \rangle)\texttt{:}\ \langle \textit{n-ticový-výraz-používající-proměnné} \rangle$$

lokální pojmenování pro relační výrazy:

$$\texttt{WITH}\ (\langle \textit{proměnná}_1 \rangle\ \texttt{:=}\ \langle \textit{výraz}_1 \rangle, \ldots, \langle \textit{proměnná}_n \rangle\ \texttt{:=}\ \langle \textit{výraz}_n \rangle)\texttt{:}\ \langle \textit{relační-výraz-používající-proměnné} \rangle$$

poznámka:

- pouze lokální pojmenování (neslouží k vyjadřování „rekurzivních“ předpisů)

Příklad (Tutorial D: Použití WITH a tranzitivní uzávěr relace)

```
WITH (r1 := r,
      r2 := (r RENAME {y AS z}) COMPOSE (r1 RENAME {x AS z}),
      r3 := (r RENAME {y AS z}) COMPOSE (r2 RENAME {x AS z}),
      r4 := (r RENAME {y AS z}) COMPOSE (r3 RENAME {x AS z})):
  UNION {r1, r2, r3, r4}

WITH (r1 := r,
      w2 := (r RENAME {y AS z}) COMPOSE (r1 RENAME {x AS z}),
      r2 := w2 MINUS r1,
      w3 := (r RENAME {y AS z}) COMPOSE (r2 RENAME {x AS z}),
      r3 := w3 MINUS UNION {r1, r2},
      w4 := (r RENAME {y AS z}) COMPOSE (r3 RENAME {x AS z}),
      r4 := w4 MINUS UNION {r1, r2, r3}):
  UNION {r1, r2, r3, r4}

TCLOSE (r)
```

Příklad (Tutorial D: Implementace tranzitivního uzávěru)

```
OPERATOR tr (r RELATION {x INT, y INT}) RETURNS SAME_TYPE_AS (r);
  BEGIN;
    VAR result PRIVATE INIT (r) KEY {x,y};
    VAR work PRIVATE INIT (r) KEY {x,y};
    WHILE NOT (IS_EMPTY (work));
      BEGIN;
        work := (r RENAME {y AS z}) COMPOSE (work RENAME {x AS z});
        work := work MINUS result;
        result := result UNION work;
      END;
    END WHILE;
    RETURN result;
  END;
END OPERATOR;
```

Problematika nespojitelných n-tic

motivace:

Pokud se v relacích, které spojujeme, nacházejí nějaké nespojitelné n-tice, ve výsledky dochází ke „ztrátě informace.“ Jak tento problém vyřešit?

převládající, ale koncepčně špatné řešení:

- snahy se objevují kolem roku 1979 (10 let po představení originálního modelu)
- zavedení *vnějších spojení* a konceptu *nedefinovaných* (*chybějících*) *hodnot*
- tabulky nelze formalizovat jako relace nad relačními schématy
- formální model je pochybný, postrádá logiku, velký potenciál vzniku chyb

další řešení:

- místo nedefinovaných hodnot používat **designované hodnoty z domén** (např. `"??"` může mít význam, že řetězcová hodnota je neznámá)
- zavedení relační operace **polorozdíl**, která vyjadřuje nespojitelné n-tice
- plné využití RM a konceptu relace jako hodnoty atributu (Přednáška 6)

Vnitřní a vnější spojení

vnější přirozená a θ-spojení v SQL:

```
SELECT * FROM ⟨jméno1⟩ NATURAL FULL OUTER JOIN ⟨jméno2⟩
SELECT * FROM ⟨jméno1⟩ NATURAL LEFT OUTER JOIN ⟨jméno2⟩
SELECT * FROM ⟨jméno1⟩ NATURAL RIGHT OUTER JOIN ⟨jméno2⟩

SELECT * FROM ⟨jméno1⟩ FULL OUTER JOIN ⟨jméno2⟩ ON ⟨podmínka⟩
SELECT * FROM ⟨jméno1⟩ LEFT OUTER JOIN ⟨jméno2⟩ ON ⟨podmínka⟩
SELECT * FROM ⟨jméno1⟩ RIGHT OUTER JOIN ⟨jméno2⟩ ON ⟨podmínka⟩
```

princip vnějších spojení:

- ve výsledku levého (pravého/oboustranného) vnějšího spojení jsou zahrnuty i n-tice z první (druhé/obou) relace(í), které jsou nespojitelné
- v tabulkách se projevuje přítomností `NULL` (značí „absenci hodnot")
- doposud diskutovaná spojení jsou tzv. *vnitřní spojení* (v SQL lze místo `NATURAL JOIN` psát `NATURAL INNER JOIN`)

Příklad (Vnější přirozená spojení)

$\mathcal{D}_1$

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

$\mathcal{D}_2$

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

levé

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
666	abc	101	

oboustranné

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
666	abc	101	
	abc	111	zzz
	ghX	103	xxx

pravé

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
	abc	111	zzz
	ghX	103	xxx

```
SELECT * FROM r1 NATURAL LEFT OUTER JOIN r2
SELECT * FROM r1 NATURAL FULL OUTER JOIN r2
SELECT * FROM r1 NATURAL RIGHT OUTER JOIN r2
```

Související problémy

patologický rys:

- výsledek vnějšího spojení nelze formalizovat jako relaci nad schématem ($\{\{\langle x, 10\rangle, \langle y, 20\rangle\}, \{\langle x, 66\rangle\}, \{\langle y, 77\rangle\}\}$ může být vysledkem vnějšího spojení)
- související otázky:
  - *Jak implementovat množinové relační operace?*
  - *Jak vlastně implementovat jakékoliv relační operace?*
  - *Co je výsledkem restrikce, pokud není hodnota atributu definovaná?*

    $\vdots$

je nutné:

- explicitně zavést nedefinovanou hodnotu (označení ω, v SQL `NULL`)
- uvažovat artimetické, logické a další operace, jejichž argumenty mohou být nedefinované (jaké jsou jejich výsledky?)
- rozšířit (zprznit) relačního modelu dat

3VL v SQL

přístup k „nedefinovaným hodnotám“ v SQL:

- použití fragmentu Kleeneho tříhodnotové logiky (3VL) s hodnotami $\{0, \omega, 1\}$ a interpretací logických spojek $\mathbin{\owedge}$, $\mathbin{\ovee}$, $\Rightarrow$, $\Leftrightarrow$, $\neg$ danou tabulkami:

$\wedge$	0	ω	1
0	0	0	0
ω	0	ω	ω
1	0	ω	1

$\vee$	0	ω	1
0	0	ω	1
ω	ω	ω	1
1	1	1	1

$\rightarrow$	0	ω	1
0	1	1	1
ω	ω	ω	1
1	0	ω	1

$\leftrightarrow$	0	ω	1
0	1	ω	0
ω	ω	ω	ω
1	0	ω	1

$\neg$	
0	1
ω	ω
1	0

bohužel, nedává smysl, protože:

- *nespecifita* $\neq$ *pravdivost*
- bereme-li v úvahu nespecifitu, přestává platit princip kompozicionality, to jest pravdivost složených formulí (v daném ohodnocení) nezávisí pouze na pravdivosti podformulí (a spojkách), ale i **na struktuře podformulí**. (**!!**)

například: pokud je $e(\varphi) = \omega$ a $e(\psi) = \omega$, pak dle 3VL máme $e(\varphi \mathbin{\ovee} \psi) = \omega$;
ale pokud je ψ ve tvaru $\neg\varphi$, musí být $e(\varphi \mathbin{\ovee} \psi) = 1$ (*tertium non datur*, **!!**)

Příklad (Příklad důsledků 3VL v SQL, C. J. Date)

```
CREATE TABLE r1 (x NUMERIC NOT NULL PRIMARY KEY, yn VARCHAR);
CREATE TABLE r2 (y NUMERIC NOT NULL PRIMARY KEY, zn VARCHAR);

INSERT INTO r1 VALUES (45, 'London');
INSERT INTO r2 VALUES (33, NULL);

SELECT x, y FROM r1, r2 WHERE (yn <> zn OR zn <> 'Paris');
```

logický výsledek:

x	y
45	33

, protože $\begin{cases} \text{buď zn} = \text{'Paris' a tím pádem yn} \neq \text{zn} \\ \text{nebo zn} \neq \text{'Paris'} \end{cases}$

výsledek dotazu v SQL:

x	y

(prázdná relace), protože $\omega \vee \omega = \omega$

You can never trust the answers you get from a database with nulls.

— C. J. Date

Alternativní přístup: designované hodnoty domén

motivace:

Snaha obejít problémy s 3VL a nedefinovanými hodnotami.

úskalí předchozího řešení:

- `NULL` jako „pravdivostní hodnota“ nedává smysl
- další možnost: používat místo `NULL` speciální hodnoty domén

formalizace:

- uvažovat domény $D_y \cup \{\omega_y\}$ rozšířené o ω_y (nedef. hodnota domény atributu y)
- typicky: ω_y jsou prázdné řetězce, zpeciální znaky, čísla, ...

přetrvávající nevýhody:

- soudobé SŘBD přímo nepodporují (například u vnějších spojení)
- problémy s použitím ω_y omylem jako „skutečnou hodnotu“ (zdroj chyb)

Odvozená operace: Polorozdíl

Definice (polorozdíl, angl.: *semidifference*)

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

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

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

Tutorial D (zobecňuje `MINUS`):

$\langle \textit{relační-výraz}_1 \rangle$ `NOT MATCHING` $\langle \textit{relační-výraz}_2 \rangle$

SQL:

```
SELECT * FROM ⟨jméno₁⟩
  EXCEPT
SELECT DISTINCT ⟨jméno₁⟩.* FROM ⟨jméno₁⟩ NATURAL JOIN ⟨jméno₂⟩
```

význam: $r_1 \in \mathcal{D}_1 \,\overline{\ltimes}\, \mathcal{D}_2$ p. k. $r_1 \in \mathcal{D}_1$ není spojitelná s žádnou n-ticí z $\mathcal{D}_2$

Příklad (Vnější spojení a polorozdíly)

$\mathcal{D}_1$

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

$\mathcal{D}_2$

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

levé

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
666	abc	101	

oboustranné

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
666	abc	101	
	abc	111	zzz
	ghX	103	xxx

pravé

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
	abc	111	zzz
	ghX	103	xxx

$\mathcal{D}_1 \overline{\ltimes} \mathcal{D}_2 =$

FOO	BAR	BAZ
666	abc	101

$\mathcal{D}_2 \overline{\ltimes} \mathcal{D}_1 =$

BAR	BAZ	QUX
abc	111	zzz
ghX	103	xxx

Přednáška 5: Závěr

pojmy k zapamatování:

- úplné přirozené spojení, bezeztrátová dekompozice
- θ-spojení a spojení na rovnost
- tranzitivní uzávěr, lokální pojmenování, rekurzivní dotazy
- vnitřní a vnější spojení, nedefinované hodnoty, polorozdíl

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