

DBS – relační DB model, relační algebra

Michal Valenta

Katedra softwarového inženýrství FIT
České vysoké učení technické v Praze
©Michal Valenta, 2014

BI-DBS, ZS 2014/15

<https://edux.fit.cvut.cz/courses/BI-DBS/>



Relační model dat (Codd 1970)

Odkud vychází, co přináší?

- Formální abstrakce textových souborů.
- Relační kalkul a relační algebra (dotazovací prostředky).
- Metodika pro posuzování kvality relačního schématu.
- Metodika pro návrh kvalitního relačního schématu.

Relační algebra – ukázka na začátek

```
filmy1(jmeno_f, reziser, rok)
kina1(nazev_k, adresa)
predstaveni1(nazev_k, jmeno_f, datum)
obsazeni12(jmeno_f, herec)
obsazeni90(jmeno_f, herec)
```

- ❶ $filmy1(rok < 1990)[jmeno_f] * predstaveni1[nazev_k]$
- ❷ $filmy1 < * predstaveni1$
- ❸ $obsazeni12[herec] \cup obsazeni90[herec]$
- ❹ $obsazeni12[herec] \cap obsazeni90[herec]$
- ❺ $obsazeni12[herec] - obsazeni90[herec]$
- ❻ $filmy1 \times kina1$
- ❼ $predstaveni1[nazev_k, jmeno_f] \div$
 $(filmy1(reziser = 'Woody Allen')[jmeno_f])$

Relace - zavedení pojmů

- jména **atributů** $[A_1, A_2, A_3, \dots, A_n]$
- **domény** atributů D_i nebo $dom(A_i)$
1.NF požaduje, aby atributy byly atomické.
- **n-tice** (a_1, a_2, \dots, a_n) (prvek relace)
- množina n -tic $\subset D_1 \times D_2 \dots \times D_n$ je **relace**
- **jméno relace** R
- **schéma relace** $R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$,
zkráceně $R(A)$

Intuitivně (ale nepřesně, viz dále):

- relace = tabulka
- schéma relace = záhlaví tabulky

Relace vs. tabulky 1/3

Mějme tabulky:

KINO* a **FILM***

NazevK	Adresa
Blaník	Václ. n. 4
Vesna	Olšiny 3
Mír	Strašnická 3
Domovina	V dvorcích 7

JmenoF	Herec	Rok
Černí baroni	Vetchý	1994
Černí baroni	Landovský	1994
Top gun	Cruise	1986
Top gun	McGillis	1986
Kmotr	Brando	1972
Nováček	Brando	1990
Vzorec	Brando	1980

Dále máme relaci:

KINO(NazevK, Adresa) a **FILM**(JmenoF, Herec, Rok).

Otázky:

Jaký je vztah mezi relacemi KINO a FILM a tabulkami KINO* a FILM*?

Jak zařídit vztah – film je na programu kina?

U jednoho filmu chceme sledovat více herců. Jak to zařídit?

Relace vs. tabulky 2/3

Jaký je vztah mezi relacemi KINO a FILM a tabulkami KINO* a FILM*?

Relace vs. tabulky – terminologie

relace

schéma relace

jméno atributu

atribut

n-tice

tabulka

záhlaví tabulky

jméno sloupce

sloupec

řádek tabulky

Odlišnosti:

- V relaci nezáleží na pořadí n-tic.
- Relace neobsahují duplicitní n-tice (jsou to množiny).

Relace vs. tabulky 3/3

U jednoho filmu chceme sledovat více herců. Jak to zařídit?

Nápad 1: Více atributů herec.

<u>NázevF</u>	Herec1	Herec2	Rok
Černí baroni	Vetchý	Landovský	1994
Kmotr	Brando		1972

Co když více než dva herci? Prázdné sloupce.

Nápad 2: Vícehodnotový atribut.

<u>NázevF</u>	Herec	Rok
Černí baroni	Vetchý, Landovský	1994
Kmotr	Brando	1972

Vícehodnotové atributy se v relačním modelu nepřipouští. **Není v 1NF.**

Nápad 3: Více řádek pro jeden film.

<u>NázevF</u>	<u>Herec</u>	Rok
Černí baroni	Vetchý	1994
Černí baroni	Landovský	1994

Zbytečná redundance, update anomálie. **Není ve 2NF.**

Detailní diskuse a “čisté” relační řešení viz přednáška o **normalizaci schématu**.

Relační model dat – definice

Je nutné zajistit, aby se do relací dostala pouze “správná data” – **přípustné n-tice**.

Def.: Schéma (relační) databáze:

(R, I) je **schéma relační databáze**, kde:

- $R = \{R_1, R_2, \dots, R_n\}$ je množina relací a
- I je množina integritních omezení.

Def.: Přípustná (relační) databáze:

Přípustná relační databáze se schématem (R, I) je množina (konkrétních) relací $\{R_1^*, R_2^*, \dots, R_n^*\}$ takových, že jejich n-tice vyhovují tvrzením v I .

Relační model dat – definice klíče schématu

Def.: Klíč schématu

Klíč K schématu $R(A)$ je taková **minimální** podmnožina atributů z A , která jednoznačně určí každou n -tici (konkrétní) relace R^* .

Pozn.: minimalita klíče (viz předchozí definice)

Požadavek na **minimalitu**, lze formulovat takto: neexistuje takové $K' \subset K$, které jednoznačně určuje každou n -tici $R(A)$.

Tvrzení 1:

Nechť K je klíč schématu $R(A)$. Pak pro každou přípustnou relaci R^* platí : jsou-li u a v dvě různé n -tice z R^* , potom $u[K] \neq v[K]$.

Tvrzení 2:

Relace R může mít několik (alternativních) klíčů.

Příklad: řekněme, že $R(\underline{a},b,c)$ a $R(a,\underline{b},c)$, pak ale už ne $R(\underline{a},\underline{b},c)$, protože $a.b$ není minimální!

Relační model dat – příklad IO

KINO (NazevK, Adresa)

FILM (JmenoF, Herec, Rok)

MA_NA_PROGRAMU (NazevK, JmenoF, Datum)

Integritní omezení:

- IO1 : primární klíče (podtržené)
- IO2 : cizí klíče:
 $MA_NA_PROGRAMU[JmenoF] \subseteq FILM[JmenoF]$
 $MA_NA_PROGRAMU[NazevK] \subseteq KINO[NazevK]$
- IO3 : V kině se nehraje více než dvakrát týdně.
- IO4 : Jeden film se nehraje více než ve třech kinech.

Relační model dat – ošetření IO

- 1 Jak definovat IO nad schématem úložiště?
- 2 Jak zajistit dodržování daného IO?

Možností vyjádření (a kontroly) IO:

- 1 deklarativní:
při vytvoření schématu (relace),
hlídá je automaticky DBMS,
výše jsme zavedli PK a FK,
později v SQL přibudou UK, NOT NULL a CHECK.
- 2 procedurální na straně serveru
- 3 procedurální na straně klienta

Relační model dat – definice dotazu

Def.: Dotaz

Databázový **dotaz** nad schématem S je výraz, který vrací odpověď se schématem T . Přičemž:

- definičním oborem jsou všechna úložiště se schématem S ,
- oborem hodnot jsou všechny relace se schématem T ,
- data v odpovědi pocházejí z databáze,
- odpověď nezávisí na fyzickém uložení dat.

Def.: Dotazovací jazyk

Dotazovací jazyk je množina všech použitelných výrazů pro konstrukci dotazu.

Relační model dat – manipulace s relacemi

- Vložení n-tice do dané relace.
- Zrušení/změna daných n-tic v dané relaci.
 - ⇒ Algoritmus manipulačních operací zahrnuje kontrolu dodržování (deklarativních) IO.
- Zadání dotazu:
 - ⇒ relační algebra,
 - ⇒ relační kalkul,
 - ⇒ SQL.

Relační model dat – shrnutí

- Relace (tabulky) jsou v 1NF,
Domény atributů jsou atomické (toto omezení překračují ORDBMS).
- Jedinečné n-tice (řádky) v rámci relace.
Relace jsou (definovány jako) množiny. V implementacích v RDBMS a ORDBMS toto neplatí.
- Vyhodnocení dotazu nezávisí na fyzickém uložení dat.
- Přístup k prvkům relace (řádkům tabulky) dle obsahu, ne podle pořadí.
- Silné prostředky pro dotazování.
(relační algebra, relační kalkul, v implementacích RDBMS SELECT).
- Existují metody návrhu schématu úložiště v relační databázi, které vedou na schémata "dobrých vlastností".
Viz normalizace schématu v dalších přednáškách.

Relační algebra – charakteristika 1/2

- Programování vs. relační algebra
 - ▶ Relační algebra je “vyšší” jazyk.
 - ▶ Nespecifikujeme “jak se mají věci udělat” (typicky 3GL), ale “co má být výsledkem”.
 - ▶ Na druhou stranu je to jazyk velmi úzce specializovaný.
- Dotazovací jazyk, který umožňuje realizovat relační algebru se nazývá **relačně úplný**.
- Komerční svět
 - ▶ SQL SELECT (je relačně úplný)
 - ▶ jazyky formulářů (nebývají relačně úplné)
 - ▶ obrázkové jazyky (nebývají relačně úplné)

Relační algebra – charakteristika 2/2

- Řeší “pouze” dotazování nikoliv DML a DDL.
- Výsledkem dotazu je relace, která může být vstupem do dalšího dotazu – dotazy lze řetězit.
- Výrazy (dotazy) se skládají z operací a operandů.
- Operandem je vždy celá relace.
- Relační algebra byla vzorem pro návrh příkazu SELECT v SQL.
- SELECT má aktuálně větší vyjadřovací možnosti než RA (agregace, vnější spojení, vnořené dotazy, řazení, ...).

RA – operace 1/5, předp. $R(A)$ a $S(B)$

- **selekce** (restrikce) relace R dle podmínky φ
Značení: $R(\varphi)$. **Definice:** $R(\varphi) =_{\text{def}} \{u \mid u \in R \wedge \varphi(u)\}$,
kde φ je (složený) logický výraz typu $(t_1 \Theta t_2)$ a/nebo $(t \Theta a)$,
kde t zastupuje atribut, a konstantu a Θ relační operátor.
- **projekce** relace R na množinu atributů C , kde $C \subseteq A$
Značení: $R[C]$. **Definice:** $R[C] =_{\text{def}} \{u[C] \mid u \in R\}$.
- **přírozené spojení** relací $R(A)$ a $S(B)$ s výsledkem $T(C)$
Značení: $R * S$. **Definice:** $R * S =_{\text{def}} \{u \mid u[A] \in R \wedge u[B] \in S\}$,
kde $C = A \cup B$ a kde výběr n -tic pro spojení je dán **rovností** na
všech průnikových attributech A a B .
- **přejmenování** atributu
Značení: $t \rightarrow \text{alias}$.

Příklad: Hvězdy (herci), které hrají ve filmech promítaných v kině Mír.

$MA_NA_PROGRAMU(\text{NazevK} = 'Mir')[JmenoF, Datum] * FILM[Herec \rightarrow Hvezda]$

Rozbor dotazu po krocích viz dále.

RA – operace 2/5, předp. $R(A)$ a $S(B)$

- **obecné spojení (Θ -spojení)** $R(A)$ a $S(B)$ s výsledkem $T(C)$

Značení: $R[t_1 \Theta t_2] S$.

Definice: $R[t_1 \Theta t_2] S =_{\text{def}} \{u \mid u[A] \in R, u[B] \in S, u.t_1 \Theta u.t_2\}$,
kde $\Theta \in \{<, >, =, \leq, \geq, \neq\}$, přípustné je také použití logických spojek $\{\wedge, \vee, \neg\}$ pro konstrukci složitějších podmínek spojení.
 C je “zřetěžením” A a B - spol. atributy ve výsledku se opakují, nutno je přejmenovat nebo používat kvalifikaci, např. $R.A$.

- množinové operace:

sjednocení – $R \cup S$

průnik – $R \cap S$

množinový **rozdíl** – $R \setminus S$ (někdy též klasicky $R - S$)

kartézský součin – $R \times S$

Minimální množina operací RA: $\{\times, \text{selekce}, \text{projekce}, \rightarrow, \cup, \setminus\}$

Ostatní operace se dají zapsat s jejich pomocí.

Např. $*$ nebo Θ -spojení pomocí \times , *selekce*, *projekce*.

RA – příklad – Θ -spojení

Mějme relace $R(A, B, C)$ a $S(B, C, D, E)$. $T := R[A < S.B]S$.

R	A	B	C
	8	2	3
	1	2	3
	2	1	4
	3	6	7
	3	8	9

S	B	C	D	E
	3	4	2	3
	3	3	3	3
	1	4	5	6
	2	3	4	7

T	A	R.B	R.C	S.B	S.C	D	E
	1	2	3	3	4	2	3
	1	2	3	3	3	3	3
	1	2	3	2	3	4	7
	2	1	4	3	4	2	3
	2	1	4	3	3	3	3

RA – příklad 1/4 – selekce

Rozeberme **po krocích** dotaz:

Hvězdy (herci), které hrají ve filmech promítaných v kině Mír.

Databáze:

KINO (NazevK, Adresa) **IO1:** $MA_NA_PROGRAMU[NazevK] \subseteq KINO[NazevK]$

FILM (JmenoF, Herec, Rok) **IO2:** $MA_NA_PROGRAMU[JmenoF] \subseteq FILM[JmenoF]$

MA_NA_PROGRAMU (NazevK, JmenoF, Datum)

$R1 := MA_NA_PROGRAMU(NazevK = 'Mir')$

MA_NA_PROGRAMU	<u>NazevK</u>	<u>NazevF</u>	Datum
	Blaník	Tog gun	29.3. 1994
	Blaník	Kmotr	8.3. 1994
	Mír	Nováček	10.3. 1994
	Mír	Top gun	9.3. 1994
	Mír	Kmotr	8.3. 1994

R1	NazevK	NazevF	Datum
	Mír	Nováček	10.3. 1994
	Mír	Top gun	9.3. 1994
	Mír	Kmotr	8.3. 1994

RA – příklad 2/4 – projekce

$R2 := R1[JmenoF, Datum]$

R1	NazevK	NazevF	Datum
	Mír	Nováček	10.3. 1994
	Mír	Top gun	9.3. 1994
	Mír	Kmotr	8.3. 1994

R2	NazevF	Datum
	Nováček	10.3. 1994
	Top gun	9.3. 1994
	Kmotr	8.3. 1994

RA – příklad 3/4 – přirozené spojení

$R3 := R2 * FILM$

R2	NazevF	Datum
	Nováček	10.3. 1994
	Top gun	9.3. 1994
	Kmotr	8.3. 1994

FILM	<u>NazevF</u>	Herec	Rok
	Černí baroni	Vetchý	1994
	Černí baroni	Landovský	1994
	Top gun	Cruise	1986
	Top gun	McGillis	1986
	Kmotr	Brando	1972
	Nováček	Brando	1990
	Vzorec	Brando	1980

R3	NazevF	Datum	Herec	Rok
	Nováček	10.3. 1994	Brando	1990
	Top gun	9.3. 1994	Cruise	1986
	Top gun	9.3. 1994	McGillis	1986
	Kmotr	8.3. 1994	Brando	1972

Spojení bylo provedeno přes rovnost na attributech J_{menoF} .

RA – příklad 4/4 – projekce, selekce a přejmenování

$R4 := R3[Herec \rightarrow Hvezda]$

R3	NazevF	Herec	Rok	Datum
	Nováček	Brando	1990	10.3. 1994
	Top gun	Cruise	1986	9.3. 1994
	Top gun	McGillis	1986	9.3. 1994
	Kmotr	Brando	1972	8.3. 1994

R4	Hvezda
	Cruise
	Brando
	McGillis

Poznámka:

Díky “množinovému” chápání RMD dojde **automaticky k vyloučení duplicit** v relaci R4.

Bez mezikroků:

Hvězdy (herci), které hrají ve filmech promítaných v kině Mír.

$((MA_NA_PROGRAMU(NazevK = 'Mir')[JmenoF, Datum]) * \\ FILM)[Herec \rightarrow Hvezda]$

RA – operace 3/5 – polospojení

- **levé Θ -polospojení**

Značení: $R <_{t_1 \Theta t_2} S$

Definice: $R <_{t_1 \Theta t_2} S =_{def} (R[t_1 \Theta t_2] S[A])$

- **pravé Θ -polospojení**

Značení: $R[t_1 \Theta t_2 > S$

Definice: $R[t_1 \Theta t_2 > S =_{def} (R[t_1 \Theta t_2] S[B])$

- **levé přirozené polospojení**

Značení: $R <_* S$

Definice: $R <_* S =_{def} (R * S)[A]$

- **pravé přirozené polospojení**

Značení: $R * > S$

Definice: $R * > S =_{def} (R * S)[B]$

Lze chápat jako “syntaktickou zkratku” – spojení následované projekcí na A resp. B
Skutečná implementace může být mnohem efektivnější.

Interpretace $R <_* S$

Podmnožina n -tic z R , které jsou **spojitelné** s nějakou n -ticí z S .

RA – operace 4/5 – antijoin

Motivace:

Tato operace se používá v modulu query executor v Oracle. Můžeme ji vidět v prováděcích plánech SQL dotazů při vyhodnocení klauzule NOT EXISTS.

Značení: $R \overleftarrow{*} S$

Definice: $R \overleftarrow{*} S =_{def} R \setminus (R <{*} S)$

Interpretace $R \overleftarrow{*} S$

Podmnožina n-tic z R , které **nejsou spojitelné** s žádnou n-ticí z S .

Příklad: Najděte kina, která nic nehrají.

$KINO \overleftarrow{*} MA_NA_PROGRAMU$

Bez operace antijoin: $KINO \setminus (KINO <{*} MA_NA_PROGRAMU)$

RA – příklady 1/6

D1: Názvy kin, která něco hrají.

$MA_NA_PROGRAMU[NazevK]$

D2: Seznam kin, která nic nehrají.

$KINO[NazevK] \setminus (MA_NA_PROGRAMU[NazevK])$

D3: Názvy kin, která hrají film Top Gun.

$MA_NA_PROGRAMU(JmenoF = 'TopGun')[NazevK]$

D4: Jména filmů, které hraje kino s adresou Zvonková.

$((KINO(Adresa = 'Zvonkova')[NazevK]) * \\ MA_NA_PROGRAMU)[JmenoF]$

RA – příklady 2/6

D5: Názvy kin, která hrají něco s hercem Brando.

$D5 :=$

$(FILM(Herec = 'Brando')[JmenoF] * MA_NA_PROGRAMU)[NazevK]$

D6: Názvy kin, která **nedávají žádný** film s Brando.

Přeformulujme dotaz D6 na D6':

Názvy všech kin kromě těch, která hrají něco s hercem Brando.

$KINO[NazevK] \setminus D5$

D6B: Názvy kin, která hrají, ale nehrají žádný film s Brando.

$MA_NA_PROGRAMU[NazevK] \setminus D5$

RA – příklady 3/6 – negace (\neg) a ex. kvant. (\exists)

D7: Názvy kin, ve kterých se **nehraje některý** film s Brando.

- 1.krok :
 \Rightarrow Univerzum pro množinu dvojic {kino, "film s Brando"}.
 $U := KINO[NazevK] \times (FILM(Herec = 'Brando')[JmenoF])$
- 2.krok :
 \Rightarrow Získejme reálnou množinu dvojic {kino, film }.
 $R := MA_NA_PROGRAMU[NazevK, JmenoF]$
- 3.krok :
 \Rightarrow Odečtením výše zkonstruovaných relací získáme množinu v programu nezrealizovaných dvojic {kino, "film s Brando"}.
 $N := U \setminus R$
- 4.krok :
 \Rightarrow Projekcí na první atribut získáme požadovaný seznam kin.
 $N[NazevK]$

Poznámka: Opravdu není nutné omezovat R na filmy s Brando.

RA – příklady 4A/6 – všeobecný kvantifikátor (\forall)

Princip formulace dotazů se všeobecným kvantifikátorem.

$$\forall x. P(x) = \neg \exists x. (\neg P(x))$$

D8: Názvy kin, ve kterých se dávají **všechny** filmy s Brando.

Přepis 1 D8':

Názvy kin, která něco hrají, a zároveň **není pravda**, že **nehrají některý** film s Brando.

Přepis 2 D8'':

Seznam kin, ve kterých něco hrají s výjimkou těch kin, která **nehrají některý** film s Brando.

Přepis 3 D8''':

$\{NazevK \mid \text{kino něco hraje}\} \setminus$
 $\{NazevK \mid \text{kino nehraje některý film s Brando}\}$

RA – příklady 4B/6 – všeobecný kvantifikátor (\forall)

D8'': $\{ \text{NazevK} \mid \text{kino něco hraje} \} \setminus$
 $\{ \text{NazevK} \mid \text{kino } \textbf{nehraje} \textbf{ některý} \text{ film s Brando} \}$

- 1. krok: Názvy kin, která něco hrají.
 $K := \text{MA_NA_PROGRAMU}[\text{NazevK}]$
- 2. krok: Univerzum dvojic {"kino s programem", "film s Brando"}.
 $U := K \times (\text{FILM}(\text{Herec} = \text{'Brando'})[\text{JmenoF}])$
- 3. krok: Reálná množina dvojic {kino, film }.
 $R := \text{MA_NA_PROGRAMU}[\text{NazevK}, \text{JmenoF}]$
- 4.krok: Nerealizované {"kino s programem", "film s Brando"}.
 $N := U \setminus R$
- 5. krok: Kina, která **nehrají některý** film s Brando.
 $B := N[\text{NazevK}]$
- 6. krok: Konečný výsledek.
 $K \setminus B$

RA – operace 5/5 – db. dělení (\div) a př. 4C/6 – (\forall)

Mějme $A(x, y)$ a $B(y)$.

Značení: $A \div B$

Definice: $A \div B =_{def} A[x] \setminus ((A[x] \times B) \setminus A)[x]$

Interpretace $A \div B$

- 1 Výsledkem jsou všechny hodnoty x z A , které v A tvoří dvojici s **každým** prvkem y z B .
- 2 Pomocí prvků y z B se snažíme **diskvalifikovat** prvky x z A . Prvek x je diskvalifikován, pokud v A neexistuje ve dvojici s **každým** y z B . Výsledkem $A \div B$ jsou ty prvky x z A , které se diskvalifikovat nepodařilo.

D8''': $\{NazevK \mid \text{ kino něco hraje} \} \setminus$
 $\{NazevK \mid \text{ kino } \textbf{nehraje} \textbf{ některý} \text{ film s Brando} \}$

$MA_NA_PROGRAMU[NazevK, JmenoF] \div$
 $(FILM(Herec = 'Brando')[JmenoF])$

RA – příklady 5/6 – všeobecný kvantifikátor (\forall)

D9: Názvy kin, ve kterých se hrají **všechny** filmy, které jsou na programu.

D9': Názvy kin, ve kterých něco hrají, a zároveň **není** pravdou, že **nehrají některý** z filmů z programu.

D9'': Názvy kin, ve kterých něco hrají s výjimkou kin, kde **nehrají některý** z filmů v programu.

D9''': $\{NazevK \mid \text{kino něco hraje}\} \setminus \{NazevK \mid \text{kino nehraje některý film z programu}\}$

① bez použití operace \div

- ▶ 1. krok: $K := MA_NA_PROGRAMU[NazevK]$
- ▶ 2. krok: $U := K \times (MA_NA_PROGRAMU[JmenoF])$
- ▶ 3. krok: $R := MA_NA_PROGRAMU[NazevK, JmenoF]$
- ▶ 4. krok: $N := U \setminus R$
- ▶ 5. krok: $B := N[NazevK]$
- ▶ 6. krok: $K \setminus B$

② s pomocí operace \div

$MA_NA_PROGRAMU[NazevK, JmenoF] \div (MA_NA_PROGRAMU[JmenoF])$

RA – příklady 6/6 – všeobecný kvantifikátor (\forall)

D10: Název kina,
ve kterém se hrají **všechna** představení z programu.

Poznámka: Výsledek je buď jedno kino, nebo nic.

D10': Název kina, pro které **není** pravda,
že **nehraje některé** přestavení z programu.

D10'': Názvy kin, která něco hrají s výjimkou kin,
ve kterých se **nehraje některé** představení z programu.

D10''': $\{NazevK \mid \text{kino něco hraje}\} \setminus$
 $\{NazevK \mid \text{kino } \textbf{nehraje } \textbf{některé} \text{ představení z programu}\}$

$MA_NA_PROGRAMU[NazevK, JmenoF, Datum] \div$
 $(MA_NA_PROGRAMU[JmenoF, Datum])$

Operace "za" (přesahující) RA – vnější spojení 1/3

Motivace: Chceme seznam představení včetně kin, která nic nehrají.

Zavedeme "bottom" prvek **NULL** do každé domény atributu.

Hodnota **NULL** má význam **UNKNOWN** / **NESPECIFIKOVÁNO** / **N/A**.
Nezaměňovat s 0 pro čísla nebo s prázdným řetězcem pro char!

R	NazevK	Adresa	JmenoF	Datum
	Blaník	Václ.n. 4	Top gun	29.03.1992
	Blaník	Václ.n. 4	Kmotr	08.03.1992
	Mír	Starostrašnická 3	Nováček	10.03.1992
	Mír	Starostrašnická 3	Top gun	09.03.1992
	Mír	Starostrašnická 3	Top gun	08.03.1992
	Mír	Starostrašnická 3	Kmotr	08.03.1992
	Vesna	V olšínách 6	NULL	NULL
	Domovina	V dvorcích	NULL	NULL

Operace "za" (přesahující) RA – vnější spojení 2/3

- **levé vnější přirozené spojení**

Značení: $R *_L S$

Definice:

$$R *_L S =_{def} (R * S) \cup ((R \setminus R < * S) \times \{NULL, \dots, NULL\})$$

- **pravé vnější přirozené spojení**

Značení: $R *_R S$

Definice:

$$R *_R S =_{def} (R * S) \cup ((S \setminus S < * R) \times \{NULL, \dots, NULL\})$$

- **plně vnější přirozené spojení**

Značení: $R *_F S$

Definice: $R *_F S =_{def} (R *_L S) \cup (R *_R S)$

Poznámky:

- Analogicky bychom mohli definovat vnější (obecné) Θ -spojení.
- Za předpokladu $R(A)$ a $S(B)$ platí:

$$(R *_L S)[A] = R$$

$$(R *_L S)[B] = (R * > S) \cup \{NULL, \dots, NULL\}$$

Operace "za" (přesahující) RA – vnější spojení 3/3

Program kin včetně těch kin, která nic nehrají.

$KINO *_L MA_NA_PROGRAMU$

Program kin včetně těch kin, která nic nehrají, a včetně filmů, které nejsou na programu žádného kina.

$KINO *_L MA_NA_PROGRAMU *_R FILM$