

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta
BAKALÁRSKA PRÁCA



Martin Lysík
Demonstrační aplikace vyhodnocování dotazu
v relačnej algebre
Katedra softwarového inženýrství

Vedúci bakalárskej práce: RNDr. Tomáš Skopal, Ph.D.
Študijný program: Informatika, programování

2007

Chcem sa poďakovať pánovi RNDr. Tomášovi Skopalovi Ph.D. za odborné vedenie mojej bakalárskej práce, za ochotu a čas, ktorý mi venoval počas tvorby programu a písania tejto práce.

Prehlasujem, že som moju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa 30.5.2007

Martin Lysík

OBSAH

1 ÚVOD.....	6
2 ZÁKLADNÁ TEÓRIA.....	7
2.1 Relačná algebra.....	7
2.2 Selekcia.....	7
2.3 Projekcia.....	8
2.4 Premenovanie atribútov.....	8
2.5 Množinové operácie.....	9
2.6 Delenie.....	10
2.7 Spojenie relácií.....	11
2.8 Prírodné spojenie.....	12
2.9 Priority operátorov.....	13
3 POŽIADAVKY NA PROGRAM.....	14
3.1 Syntax dotazu relačnej algebry.....	14
3.2 Notácia vyhodnocovacieho stromu.....	14
3.3 Vizualizácia vyhodnocovania.....	14
4 UŽÍVATEĽSKÁ DOKUMENTÁCIA.....	16
4.1 Inštalácia.....	16
4.2 Spustenie programu.....	16
4.3 Hlavné okno.....	16
4.4 Zakladanie, otváranie a ukladanie projektu.....	17
4.5 Práca s tabuľkami v projekte.....	18
4.6 Práca s dotazmi.....	19
4.7 Práca s poznámkami.....	20
4.8 Vyhodnotenie dotazu.....	20
4.6 Vizualizácia medzivýsledkov.....	21
4.6.1 Ovládanie animácie.....	22
4.7 Vizualizácia vyhodnocovania operácií.....	23
4.7.1 Animácia pre projekciu a premenovanie.....	23
4.7.2 Animácia pre selekciu.....	24
4.7.3 Animácia množinových operácií.....	24
4.7.4 Animácia pre delenie.....	25
4.7.5 Animácia pre operácie spojenia.....	27
4.8 Nastavenia.....	27
5 GRAFICKÉ UŽÍVATEĽSKÉ ROZHRAŇIE.....	30
5.1 Rozvrhnutie aplikácie.....	30
5.2 Spravovanie objektov projektu.....	30
5.3 Editácia tabuliek.....	31
5.4 Editácia dotazov a poznámok.....	31
5.5 Ukladanie a otváranie súborov.....	32
5.6 Vykresľovanie vyhodnocovacieho stromu.....	32
5.7 Chyby.....	33
6 SPRACOVANIE DOTAZU.....	35
6.1 Prevod dotazu do textového formátu.....	35
6.2 Vnútoraná reprezentácia dotazu.....	35
<i>Vnútoraná reprezentácia podmienky.....</i>	<i>36</i>
<i>Vnútoraná reprezentácia výrazu.....</i>	<i>36</i>

6.3	Syntaktická analýza dotazu.....	37
6.3.1	<i>Gramatika popisujúca syntax dotazu.....</i>	37
6.3.3	<i>Syntaktická analýza podmienky.....</i>	38
6.3.2	<i>Syntaktické chyby.....</i>	39
6.4	Sémantická analýza dotazu.....	39
6.4.1	<i>Sémantická analýza podmienky.....</i>	40
6.4.2	<i>Sémantická analýza pri selekcii.....</i>	40
6.4.3	<i>Sémantická analýza pri projekcii.....</i>	41
6.4.4	<i>Sémantická analýza pri premenovaní atribútov.....</i>	41
6.4.5	<i>Sémantická analýza pri množinových operáciách.....</i>	41
6.4.6	<i>Sémantická analýza pri spojeniach.....</i>	41
7	VIZUALIZÁCIA.....	42
7.1	Vnútoraná reprezentácia animácie.....	42
7.2	Animácia medzivýsledkov.....	43
7.3	Animácia vyhodnocovania operácie.....	43
7.3.1	<i>Animácia pri projekcii.....</i>	43
7.3.2	<i>Animácia pri premenovávaní.....</i>	44
7.3.3	<i>Animácia pri ostatných operáciách.....</i>	44
8	ZÁVER.....	45
	POUŽITÉ ZDROJE.....	46

Názov práce: Demonstrační aplikace vyhodnocování dotazu v relačnej algebre

Autor: Martin Lysík

Katedra: Katedra softwarového inženýrství

Vedúci bakalárskej práce: RNDr. Tomáš Skopal, Ph.D.

E-mail vedúceho: Tomas.Skopal@mff.cuni.cz

Abstrakt:

Predmetom práce je implementácia aplikácie na vyhodnotenie dotazu zapísaného v relačnej algebre a na vizualizáciu vyhodnocovania jednotlivých operácií na konkrétnych dátach. Syntax dotazov je rovnaká, ako sa vyučuje na predmete Databázové systémy na MFF UK. Program taktiež umožňuje editáciu tabuliek a dotazov v rámci uceleného projektu.

Kľúčové slová: relačná algebra, vyhodnocovací strom, vizualizácia vyhodnocovania

Title: An Educational Application for Relational Algebra Query Evaluation

Author: Martin Lysík

Department: Department of Software Engineering

Supervisor: RNDr. Tomáš Skopal, Ph.D.

Supervisor's e-mail address: Tomas.Skopal@mff.cuni.cz

Abstract:

The task of this work is implementation of application for relational algebra query evaluation and for visualization of particular operations on particular data. Syntax of queries is the same as is used in the course of Database systems at the Faculty of Mathematics and Physics of Charles University. Program also allows editing tables and queries within whole project.

Keywords: relational algebra, evaluation tree, evaluation visualization

1 ÚVOD

Databázu si môžeme predstaviť ako miesto, kde sú uchované informácie, ktoré spolu určitým spôsobom súvisia. V prípade potreby môžeme informácie v databáze kedykoľvek zmeniť. Taktiež môžeme do databázy pridávať nové záznamy alebo mazať už nepotrebné či neplatné záznamy.

Väčšina databází pracuje s relačným dátovým modelom. Dáta sú situované do tabuliek, ktorých môže byť v databáze viac. Napríklad pre požičovňu filmov bude vhodné pracovať s tabuľkami zákazníkov, filmov a s tabuľkou obsahujúcou informácie o tom, ktorý zákazník si požičal aký film a kedy. Aby sme mohli z databázy získať nejaké užitočné informácie, potrebujeme jazyk, pomocou ktorého budeme môcť špecifikovať svoje požiadavky. Relačný dátový model poskytuje relačnú algebru alebo relačný kalkul, ako silné prostriedky pre formuláciu databázových dotazov. Podrobný popis relačnej algebry a jej operátorov spolu s príkladmi dotazov sa nachádza v druhej kapitole.

Cieľom tejto práce bolo vytvoriť výukový program, ktorý by vyhodnocoval dotaz zapísaný v relačnej algebre. Predpokladá sa, že dotaz bude zapísaný v syntaxi, ktorá sa používa na kurze Databázové Systémy na MFF UK. Za cieľ sa tiež kládlo sprístupnenie vyhodnocovania konkrétneho dotazu pomocou vizualizácie vyhodnocovania jednotlivých operácií. Nepodarilo sa mi nájsť program podobného charakteru, ktorý by umožňoval vyhodnocovanie dotazu zapísaného v relačnej algebre, respektíve vizualizáciu vyhodnocovacích algoritmov pre jednotlivé operácie. Požiadavky na výsledný program sú zhrnuté v tretej kapitole tejto práce.

V rámci tejto práce bol implementovaný program *REAL*, ktorého názov vznikol zo slovného spojenia *RE*lačná *AL*gebra respektíve *RE*lational *AL*gebra. Program umožňuje vytváranie a editáciu relácií a tiež zapisovanie dotazov relačnej algebry nad týmito reláciami. Následne je možné zapísané dotazy vyhodnotiť. Vyhodnocovanie dotazu je ilustrované vykreslením vyhodnocovacieho stromu dotazu. Po vyhodnotení dotazu je možné zobrazit' medzivýsledky niektorej operácie alebo si prehrať animáciu vyhodnocovania niektorej operácie na konkrétnych reláciách.

V štvrtej kapitole tejto práce je uvedená užívateľská dokumentácia programu *REAL*, ktorá obsahuje návod na používanie programu. Nasledujúce kapitoly obsahujú popis riešenia hlavných funkcií programu. Text práce je doplnený o obrázky, ktoré sú väčšinou prevzaté priamo z programu *REAL*.

2 ZÁKLADNÁ TEÓRIA

2.1 Relačná algebra

Relačná algebra je jedným z formálnych dotazovacích jazykov spojených s relačným dátovým modelom. Relačný model dát je tvorený reláciami, ktoré majú svoje schémy. Schému relácie tvorí usporiadaný zoznam dvojíc názvov a domén jednotlivých atribútov. Samotná relačná algebra potom pracuje s konkrétnymi inštanciami jednotlivých relácií. Inštanície relácií sú chápané ako množiny, takže inštančia relácie neobsahuje žiadne duplicitné prvky. [1][2]

Napríklad nech relácia *Zamestnanci* obsahuje atribúty *ID*, *Meno*, *DatumNarodenia*, kde atribút *ID* je prirodzené číslo, *Meno* je znakový reťazec a *DatumNarodenia* je typu dátum. Potom inštančia tejto relácie by bola napríklad nasledujúca tabuľka, ktorá už obsahuje konkrétne dáta.

Zamestnanci			
	ID	Meno	DatumNarodenia
	1	Adam	20/01/1977
	2	Božena	17/06/1972
	4	Cyril	02/10/1976
	8	Daniela	03/04/1980

Obrázok 2.1 Inštančia Zamestnanci1 relácie Zamestnanci

Dotazy v relačnej algebre sú tvorené názvami inštancií relácií a operátormi relačnej algeby. Operácie relačnej algeby sú definované nad reláciami a výsledkom každej operácie je opäť relácia. Takto môžeme konštruovať aj zložitejšie dotazy, ktoré obsahujú viac operátorov relačnej algeby. Dotaz relačnej algeby teda možno definovať ako samotnú reláciu alebo ako unárny operátor aplikovaný na nejaký dotaz relačnej algeby alebo ako binárny operátor aplikovaný na dva dotazy relačnej algeby. Aby sa predišlo nesprávnemu vyhodnoteniu dotazu kôli rôznym prioritám operátorov relačnej algeby, je vhodné uzatvárať poddotazy do okrúhlych zátvoriek. V ďalšom texte budú postupne opísané základné operácie relačnej algeby.

2.2 Selekcia

Selekcia je unárna operácia, ktorá zo vstupnej relácie vytvorí reláciu s rovnakou schémou, ktorá bude obsahovať všetky prvky vstupnej relácie, ktoré spĺňajú selekčnú podmienku. Selekcia má nasledujúcu syntax [1]

Relácia(Podmienka)

Podmienka je logická formula, ktorá je zadaná pomocou logických operátorov (\wedge , \vee , \Rightarrow , \Leftrightarrow a \neg) a atomických logických formulí. Aj logické operátory majú rôzne priority (\neg najväčšia priorita, \wedge , \vee , \Rightarrow , \Leftrightarrow najmenšia priorita), preto je vhodné uzatvárať jednotlivé podformule do okrúhlych zátvoriek. Každá atomická logická formula má tvar [1]

Výraz1 Θ Výraz2

kde Θ je ľubovoľný porovnávací operátor ($<$, \leq , $=$, \geq , $>$, \neq). Výraz1 a Výraz2 sú výrazy obsahujúce aritmetické operátory ($+$, $-$, $*$, $/$, mod , div) a základné operandy.

Základným operandom môže byť ľubovoľná konštanta (celé číslo, reálne číslo, reťazec alebo dátum) alebo identifikátor niektorého atribútu vstupnej relácie, za ktorý sa dosadí vždy príslušná hodnota atribútu pre spracovávaný prvok relácie. V každej atomickej formulí musí byť použitý aspoň jeden identifikátor atribútu [1][2]. Za základný operand môžeme považovať aj funkciu na atribúte (Min, Max, Avg, Sum), ktoré sa vyhodnotia na začiatku zo vstupnej relácie a ďalej sa používajú ako konštanty.

Na obrázku 2.2 je uvedený ako príklad dotaz, ktorého výsledkom je inštancia relácie *Zamestnanci*, ktorá obsahuje práve zamestnancov narodených neskôr než 1.1.1997, teda prvky inštancie *Zamestnanci1*, ktorých hodnota atribútu *DatumNarodenia* je väčšia ako 1.1.1997.

	ID	Meno	DatumNarodenia
	1	Adam	20/01/1977
	8	Daniela	03/04/1980

Obrázok 2.2 Zamestnanci1(DatumNarodenia > 1.1.1997)

2.3 Projekcia

Projekcia je unárna operácia, ktorá zo vstupnej relácie vytvorí reláciu, ktorej schéma bude obsahovať zadanú podmnožinu atribútov pôvodnej relácie. Pomocou projekcie je možné meniť poradie atribútov v relačnej schéme výslednej relácie. Z výslednej relácie budú odstránené duplicitné riadky. Projekcia má nasledujúcu syntax [1]

Relácia[atribút₁, ... , atribút_N]

Kde atribút_i je identifikátor atribútu vstupnej relácie. Atribút môže byť identifikovaný v projekcii menom atribútu (*meno* alebo *menorelacie.meno*) alebo pozíciou v rámci relácie (*menorelacie.i*) [2].

Príklad na obrázku 2.3 ilustruje projekciu na atribúty *Id* a *Meno* v inštancii *Zamestnanci1*.

	ID	Meno
	1	Adam
	2	Božena
	4	Cyril
	8	Daniela

Obrázok 2.3 Zamestnanci1[ID, Meno]

2.4 Premenovanie atribútov

Premenovanie atribútov je unárna operácia, ktorá nijakým spôsobom nemení dáta vo vstupnej inštancii, ale mení názvy atribútov podľa zadaných premenovacích pravidiel. Premenovanie má nasledujúcu syntax [1]

Relácia< atribút₁ → novémemo₁, ... , atribút_N → novémemo_N>

Kde atribút_i je identifikátor atribútu vstupnej relácie. Pri premenovaní môže byť atribút identifikovaný menom atribútu vo vstupnej relácii alebo jeho pozíciou vo vstupnej relácii. Novémemo_i je nové meno pre tento atribút.

Operáciu premenovanie atribútov možno využiť na zadanie mena pre nepomenovaný atribút alebo na výmenu mien pre niektoré atribúty, keďže premenovacie pravidlá sa

aplikujú na reláciu operandu postupne, ako sú uvedené v zozname premenovacích pravidiel.

Príklad na obrázku 2.4 ilustruje prevod schémy relácie *Zamestnanci* do angličtiny. Atribút *Meno* zmeníme na *Name* a názov atribútu *DatumNarodenia* zmeníme na *Birthday*.

	ID	Name	Birthday
	1	Adam	20/01/1977
	2	Božena	17/06/1972
	4	Cyril	02/10/1976
	8	Daniela	03/04/1980

Obrázok 2.4 $Zamestnanci1 < \text{Meno} \rightarrow \text{Name}, \text{DatumNarodenia} \rightarrow \text{Birthday} >$

2.5 Množinové operácie

V relačnej algebre sú k dispozícii štandardné binárne množinové operácie ako zjednotenie, prienik, rozdiel a kartézsky súčin. Kým kartézsky súčin je definovaný pre ľubovoľné dve relácie, ostatné operácie vyžadujú aby schémy vstupných relácií boli **kompatibilné** [1] (**union-compatible** [2]). Relácie majú kompatibilné schémy, ak majú rovnaký počet atribútov a pre všetky i platí, že i -ty atribút v prvej relácii má rovnakú doménu ako i -ty atribút v druhej relácii. Množinové operácie v relačnej algebre majú tradičnú syntax.

Relácia1 \cup Relácia2

Relácia1 \cap Relácia2

Relácia1 - Relácia2

Relácia1 \times Relácia2

Pri operáciach zjednotenie, prienik a rozdiel relácií má výsledná relácia totožnú schému so schémou relácie prvého operandu. Výsledná relácia **zjednotenia** relácií obsahuje všetky prvky, ktoré sa vyskytujú aspoň v jednej z relácií operandov. Výsledná relácia **prieniku** obsahuje všetky prvky, ktoré sa nachádzajú v oboch reláciách. Výsledkom **rozdielu** relácií je relácia, ktorá obsahuje všetky prvky, ktoré obsahuje prvá relácia a zároveň sa nevyskytujú v druhej relácii.

Zamestnanci2			
	ID	Meno	DatumNarodenia
	1	Adam	20/01/1977
	3	Boris	14/02/1981
	4	Cyril	02/10/1976

Obrázok 2.5 Inštancia *Zamestnanci2* relácie *Zamestnanci*

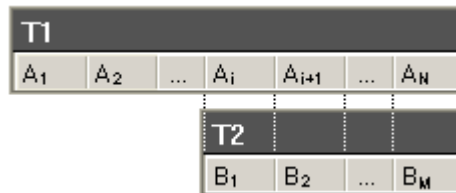
	ID	Meno	DatumNarodenia
	1	Adam	20/01/1977
	2	Božena	17/06/1972
	4	Cyril	02/10/1976
	8	Daniela	03/04/1980
	3	Boris	14/02/1981

Obrázok 2.6 $Zamestnanci1 \cup Zamestnanci2$

Pri operácii **kartézsky súčin** má výsledná relácia schému, ktorá vznikne pridaním všetkých atribútov relácie druhého operandu do schémy relácie prvého operandu. Keďže schémy oboch relácií môžu obsahovať atribúty s rovnakým menom, dokonca môžeme chcieť určiť kartézsky súčin relácie s ňou samotnou, mená atribútov, ktoré sú v oboch schémach, budú vo výslednej relácii nedefinované a tieto atribúty budú prístupné len podľa pozície. Výsledná relácia obsahuje prvky, ktoré vzniknú spojením všetkých možných kombinácií prvkov z relácie prvého operandu s prvkami relácie druhého operandu.

2.6 Delenie

Delenie je takzvaná opačná operácia k operácii kartézskeho súčinu. Čiže ak platí, že $R_3 = R_1 \times R_2$, potom tiež platí $R_1 = R_3 \div R_2$. Deliť nemožno ľubovoľné dve relácie, ale relácie musia byť kompatibilné pre operáciu delenie [1][2]. To znamená, že prvá relácia musí mať viac atribútov ako druhá relácia. A ak n je počet atribútov prvej relácie a m počet atribútov druhej relácie, potom tiež pre všetky $i = 0, \dots, m-1$ má $(n-i)$ -ty atribút prvej relácie rovnakú doménu ako $(m-i)$ -ty atribút druhej relácie.



Obrázok 2.7 Ilustrácia schém relácií kompatibilných pre delenie. Atribúty zobrazené nad sebou musia mať rovnaké domény.

Výsledkom delenia relácii R_1 a R_2 bude relácia R_3 taká, že kartézsky súčin $R_3 \times R_2$ je najväčšou podmnožinou (čo do počtu prvkov) relácie R_1 . Iná definícia delenia hovorí, že výsledkom delenia relácií R_1 a R_2 je relácia R_3 obsahujúca všetky prvky p_1 také, že pre každý prvok p_2 relácie R_2 , prvok $p_1 \oplus p_2$ je prvkom relácie R_1 . Symbol \oplus symbolizuje operáciu zretžazenia prvkov. Operácia delenie má v relačnej algebre nasledujúcu syntax [1]

$$\text{Relácia1} \div \text{Relácia2}$$

Schému výslednej relácie tvorí prvých $(n - m)$ atribútov prvej relácie, kde n je počet atribútov prvej relácie a m je počet atribútov druhej relácie. V príklade použijeme pre jednoduchosť relácie s malým počtom atribútov.

T1	
A1	A2
1	d1
2	d1
3	d1
4	d1
1	d2
2	d2
1	d3
2	d3
1	d4

T2
A2
d1
d2
d3
d4

T3
A2
d1
d2
d3

T4
A2
d1

Obrázok 2.8 Inštancie relácií pre príklad delenia

	A1
1	

	A1
1	
2	

	A1
1	
2	
3	
4	

Obrázok 2.9 Príklady delenia relácií, zľava $T1 \div T2$, $T1 \div T3$ a $T1 \div T4$

2.7 Spojenie relácií

Spojenie relácií (*join*) je binárna operácia obsahujúca spájajúcu podmienku. Podmienka pri spájaní relácií má rovnakú syntax ako selekčná podmienka popísaná v odstavci 2.2. Schéma výslednej relácie je rovnaká ako schéma kartézského súčinu relácií operandov. Pre všetky dvojice $p1$ a $p2$, kde $p1$ je prvkom prvej relácie a $p2$ je prvkom druhej relácie, sa vyhodnotí spájajúca podmienka a v prípade, že je splnená, potom prvok $p1 \oplus p2$ patrí do výslednej relácie (operátor \oplus symbolizuje zretženie prvkov relácií). Spojenie má nasledujúcu syntax [1]

Relácia1 [Podmienka] Relácia2

a platí preň nasledujúci vzťah [1][2]

$$A [\varphi] B = (A \times B)(\varphi)$$

Odelenia	
	VeduciID
Finančné	1
Zákaznícke	2

Obrázok 2.10 Inštancia Odelenia

Na obrázku 2.11 je uvedený príklad spojenia, v ktorom chceme ku každému odeleniu pripojiť informáciu o jeho vedúcom.

	ID	Meno	DatumNarodenia	Nazov	VeduciID
	1	Adam	20/01/1977	Finančné	1
	2	Božena	17/06/1972	Zákaznícke	2

Obrázok 2.11 Zamestnanci1 [ID = VeduciID] Odelenia

Existuje viacero variant spájaní relácií. Jednou z nich sú **vonkajšie spojenia**. Rozlišujeme ľavé, pravé a plné vonkajšie spojenie. Schéma výslednej relácie je rovnaká ako pri samotnom klasickom spojení, akurát do výslednej relácie sú pridané ešte niektoré prvky relácií operandov spojené s riadkom obsahujúcim nedefinované hodnoty (NULL). V prípade ľavého vonkajšieho spojenia (*left outer join*) sa do výslednej relácie klasického spojenia pridajú všetky prvky relácie prvého (ľavého) operandu spojené s prázdny (nedefinovaný) prvkom so schémou relácie druhého (pravého) operandu, ktoré nevyhoveli spájajúcej podmienke so žiadnym prvkom relácie druhého operandu. Symetricky pre pravé vonkajšie spojenie (*right outer join*). Plné vonkajšie spojenie (*full outer join*) je vlastne zjednotenie ľavého a pravého vonkajšieho spojenia. Vonkajšie spojenia sa zapisujú vo výraze relačnej algebry pridaním znaku $_L$, $_R$ alebo $_F$ (pre ľavé, pravé alebo plné vonkajšie spojenie) za uzatváraciu zátvorku podmienky [1].

Ako príklad môžeme uviesť dotaz, v ktorom nás zaujíma zoznam zamestnancov a prípadných odelení, kde sú vedúcimi. Inštancia výslednej relácie je uvedená v

príklade na obrázku 2.12

	ID	Meno	DatumNarodenia	Nazov	VeduciID
	1	Adam	20/01/1977	Finančné	1
	2	Božena	17/06/1972	Zákaznícke	2
	4	Cyril	02/10/1976		
	8	Daniela	03/04/1980		

Obrázok 2.12 Zamestnanci1 [ID = VeduciID]_L Odelenia

Ďalšou variantou spojenia sú **polospojenia**. Ich výsledok je rovnaký ako keby sme najprv vyhodnotili klasické spojenie a výslednú reláciu by sme dali ako operand projekcii na atribúty niektorej z relácií operandov. Výsledkom ľavého polospojenia (*left semi-join*) je relácia s rovnakou schémou akú má relácia prvého (ľavého) operandu a obsahuje všetky prvky prvej relácie, ktoré spĺňajú spájaciu podmienku s aspoň jedným prvkom relácie druhého operandu. Symetricky definujeme pravé polospojenie (*right semi-join*). Ľavé polospojenie zapisujeme v dotaze relačnej algebry ako klasické spojenie, akurát miesto ľavej hranatej zátvorky použijeme znak '<'. Pri pravom polospojení použijeme miesto pravej hranatej zátvorky znak '>' [1].

Napríklad keby nás zaujímali len zamestnanci, ktorý sú vedúcimi nejakého odelenia, mohli by sme použiť ľavé polospojenie, ako je uvedené v príklade na obrázku 2.13.

	ID	Meno	DatumNarodenia
	1	Adam	20/01/1977
	2	Božena	17/06/1972

Obrázok 2.13 Zamestnanci1 <ID = VeduciID] Odelenia

2.8 Prirodzené spojenie

Prirodzené spojenie (*natural join*) je binárna operácia. Ide vlastne o špeciálny prípad spojenia, kde spájacou podmienkou je rovnosť vo všetkých rovnako pomenovaných atribútoch oboch relácií. Odtiaľ tiež vyplýva, že atribúty s rovnakým názvom musia byť navzájom porovnateľné, teda musia mať rovnakú doménu. Schéma výslednej relácie je tvorená atribútmi relácie prvého operandu, ku ktorým sú pridané tie atribúty zo schémy relácie druhého operandu, ktoré majú odlišný názov od atribútov prvého operandu. Prirodzené spojenie sa v dotaze relačnej algebry zapisuje nasledovne [1]

Relácia1 * Relácia2

Dotaz z obrázku 2.11 by sme mohli zapísať pomocou prirodzeného spojenia, ak by sme atribút VeduciID relácie Odelenia premenovali na ID. Tým pádom by sa atribút výslednej relácie popisujúci ID zamestnanca vyskytoval vo výslednej relácii iba raz. Inštancia výslednej relácie by potom vyzerala nasledovne

	ID	Meno	DatumNarodenia	Nazov
	1	Adam	20/01/1977	Finančné
	2	Božena	17/06/1972	Zákaznícke

Obrázok 2.14 Zamestnanci1 * Odelenia<VeduciID→ID>

Ako klasické tak aj prirodzené spojenie má svoje varianty, vonkajšie spojenia a polospojenia. Ich výsledné relácie sa získavajú obdobne ako pri klasickom spojení. Ľavé, pravé a plné vonkajšie prirodzené spojenie sa zapisujú pridaním znaku _L, _R alebo _F za znak '*'. Pri ľavom polospojení sa používa dvojica znakov '<*' a pri pravom

polospojená sa používa dvojica znakov $*\rightarrow$ [1].

Príklad ľavého vonkajšieho spojenia z obrázku 2.12 môžeme tiež zapísať pomocou ľavého vonkajšieho prirodzeného spojenia.

	ID	Meno	DatumNarodenia	Nazov
	1	Adam	20/01/1977	Finančné
	2	Božena	17/06/1972	Zákaznícke
	4	Cyril	02/10/1976	
	8	Daniela	03/04/1980	

Obrázok 2.15 Zamestnanci1 $*_L$ Odelenia<VeduciID \rightarrow ID>

Podobne by sme mohli prepísať príklad ľavého polospojenia z obrázku 2.13 na dotaz Zamestnanci1 $<* \text{ Odelenia}<\text{VeduciID}\rightarrow\text{ID}>$. Výsledná relácia bude rovnaká ako na obrázku 2.13 pretože nás zaujímajú len prvky relácie ľavého operandu, ktoré spolu s aspoň jedným prvkom relácie pravého operandu spĺňajú spájajúcu podmienku.

2.9 Priority operátorov

Operátory relačnej algebry majú rôzne priority, ktoré sa uplatňujú pri vyhodnocovaní dotazu obsahujúceho viac operátorov. Najvyššiu prioritu majú unárne operátory selekcia, projekcia a premenovanie atribútov. Aplikovaním viacerých unárnych operátorov na jednu vstupnú reláciu, sú tieto operátory vyhodnocované zľava doprava.

	ID	Meno
	1	Adam
	8	Daniela

Obrázok 2.16 Zamestnanci1(DatumNarodenia > 1.1.1977)[ID, Meno]

V príklade na obrázku 2.16 sa teda najprv vyhodnotí selekcia a výsledná relácia selekcie sa predá projekcii ako vstupný operand. Z binárnych operátorov má najvyššiu prioritu kartézsky súčin, potom nasleduje delenie spolu so všetkými typmi spojenia, za nimi nasleduje množinový rozdiel a najnižšiu prioritu majú zjednotenie a prienik. V prípade, že v dotaze je viac operácií, ktoré majú rovnakú prioritu, použitých za sebou, sú tieto operácie vyhodnocované postupne zľava doprava.

Ak chceme zrušiť implicitné priority operátorov vo vyhodnocovaní dotazu, musíme poddotazy, ktoré chceme aby sa vyhodnotili najskôr, uzavrieť do zátvoriek.

Priorita	Operácia
najvyššia priorita	selekcia, projekcia, premenovanie
	kartézsky súčin
	delenie, ľubovoľné spojenie
	rozdiel
najnižšia priorita	zjednotenie, prienik

Obrázok 2.17 Tabuľka priorít operátorov

3 POŽIADAVKY NA PROGRAM

3.1 Syntax dotazu relačnej algebry

Aplikácia REAL bola vytváraná ako výuková aplikácia. Z tohto dôvodu bola použitá syntax, aká sa vyučuje na predmete Databázové systémy na MFF UK. Syntax dotazu relačnej algebry bola popísaná v predošlej kapitole a odpovedá syntaxi použitej v skriptách [1] alebo slajdoch k prednáške Databázové systémy [3].

Názvy relácií ani názvy atribútov použité v dotaze nie sú citlivé na veľkosť písmen (nie sú case-sensitive). Pre zápis konštant boli zvolené nasledujúce pravidlá:

- **Celočíselná konštanta** sa zapisuje pomocou číslíc. Napríklad 1234.
- **Reálna konštanta** sa zapisuje v tvare celá časť, bodka a desatinná časť. Za to sa môže uviesť exponentová časť ktorá ma tvar 'E' alebo 'e', nepovinné znamienko (+ alebo -) a celočíselná hodnota desiatkového exponentu. Napríklad 12.34E-5.
- **Dátumová konštanta** sa zapisuje v tvare D.M.Y, kde D je číslo dňa, M je číslo mesiaca a Y je rok. Napríklad 30.5.2007.
- **Reťazcová konštanta** sa zapisuje ako text uzavretý v úvodzovkách ("). Ak chceme v tomto texte použiť úvodzovky, musíme ich prefixovať spätným lomítkom. Napríklad "ABC\"abc\"12" odpovedá textu ABC"abc"12.

3.2 Notácia vyhodnocovacieho stromu

Vyhodnocovací strom odpovedajúci dotazu relačnej algebry má v koreni názov relácie v prípade, že dotaz obsahoval len tento názov a žiadnu operáciu. Ak dotaz obsahoval aspoň jednu operáciu, potom koreň vyhodnocovacieho stromu reprezentuje operáciu, ktorá sa bude vyhodnocovať ako posledná. Ak sa jedná o unárnu operáciu, tak tento uzol bude mať jedného potomka (podstrom), ktorý bude reprezentovať poddotaz, ktorého výsledná relácia bude použitá ako operand pre túto operáciu. Ak sa jedná o binárnu operáciu, tak tento uzol bude mať dvoch potomkov (podstromy), ktorí budú reprezentovať poddotazy, ktorých výsledné relácie budú použité ako operandy pre túto operáciu. Odtiaľto nám plynie, že mená relácií sa budú vo vyhodnocovacom strome dotazu vyskytovať len ako listy.

Aplikácia REAL zobrazuje vyhodnocovací strom v popísanej notácii. Tento strom je zobrazovaný v závislosti od nastavenia parametrov, určujúcich rozmery jednotlivých typov uzlov, vertikálne rozostupy medzi susednými vrstvami stromu alebo horizontálne rozostupy medzi susednými listami vyhodnocovacieho stromu.

3.3 Vizualizácia vyhodnocovania

Dotaz relačnej algebry sa vyhodnocuje z vyhodnocovacieho stromu od koreňa rekurzívne. Ak je uzol stromu list (názov relácie), tak výsledkom je príslušná relácia. Ak uzol stromu reprezentuje unárnu alebo binárnu operáciu, tak sa najprv rekurzívne vyhodnotia podstromy tohto uzlu a potom sa vyhodnotí operácia, ktorú tento uzol reprezentuje, na výsledných reláciách poddotazov. Program by mal umožňovať vizualizáciu, v ktorej sa zobrazí priebeh vyhodnocovania dotazu vybraného podstromu. Postupne sa vyznačí vo vyhodnocovacom strome časť, ktorá sa práve vyhodnocuje a zobrazia sa relácie, ktoré sú použité ako operandy príslušnej operácie

a tiež výsledná relácia.

Ďalej by program mal umožňovať vizualizáciu vyhodnocovania konkrétnej operácie na konkrétnych reláciach operandov. Vizualizácia by pozostávala z postupného zobrazovania vyhodnocovacích stavov. Napríklad postupným zvýrazňovaním prvkov relácie operandu pri selekcii zelenou farbou v prípade, že selekčná podmienka je splnená a prvok je pridaný do výslednej relácie, alebo červenou farbou v prípade, že selekčná podmienka nieje splnená a tento prvok sa nepridáva do výslednej relácie.

Súčasťou každej vizualizácie by mala byť možnosť krokovať dopredu alebo dozadu a tiež možnosť spustiť vizualizáciu ako animáciu, ktorá sama vo zvolených intervaloch zobrazí postupne všetky kroky vyhodnocovania.

4 UŽIVATEĽSKÁ DOKUMENTÁCIA

4.1 Inštalácia

Program REAL je napísaný pre operačný systém Windows a pre svoj chod vyžaduje .NET Framework 2.0, ktorého inštalácia sa v prípade nutnosti zaistí počas inštalácie programu. Program sa inštaluje pomocou inštalačného balíku RealSetup.exe.

Inštalačný balík má dve varianty. Prvá v sebe obsahuje aj inštalačný balík .NET Framework 2.0, ktorý ak ešte nieje nainštalovaný, tak sa najprv spustí jeho inštalácia. Druhá varianta v prípade ak nieje nainštalovaný .NET Framework 2.0 umožní užívateľovi jeho stiahnutie z internetu. Sprievodca inštaláciou Vám umožní zvoliť si kam sa program nainštaluje, či bude vytvorený zástupca na pracovnej ploche, respektíve v ponuke Štart menu.

4.2 Spustenie programu

Program spustíte súborom REAL.exe v inštalačnom adresári. Po spustení sa otvorí hlavné okno programu (Obrázok 4.1). Program pracuje s projektom (REAL project) a je možné pracovať naraz len s jedným projektom v jednej inštancii programu. Program je možné spustiť vo viacerých inštanciách naraz. Tabuľky jednotlivých relácií je možné ukladať do samostatných súborov a potom ich pridávať do iných projektov. Text dotazu respektíve poznámky je možné kopírovať pomocou schránky Windows.

Pri spustení môžete ako parameter uviesť názov súboru s projektom, ktorý chcete otvoriť.

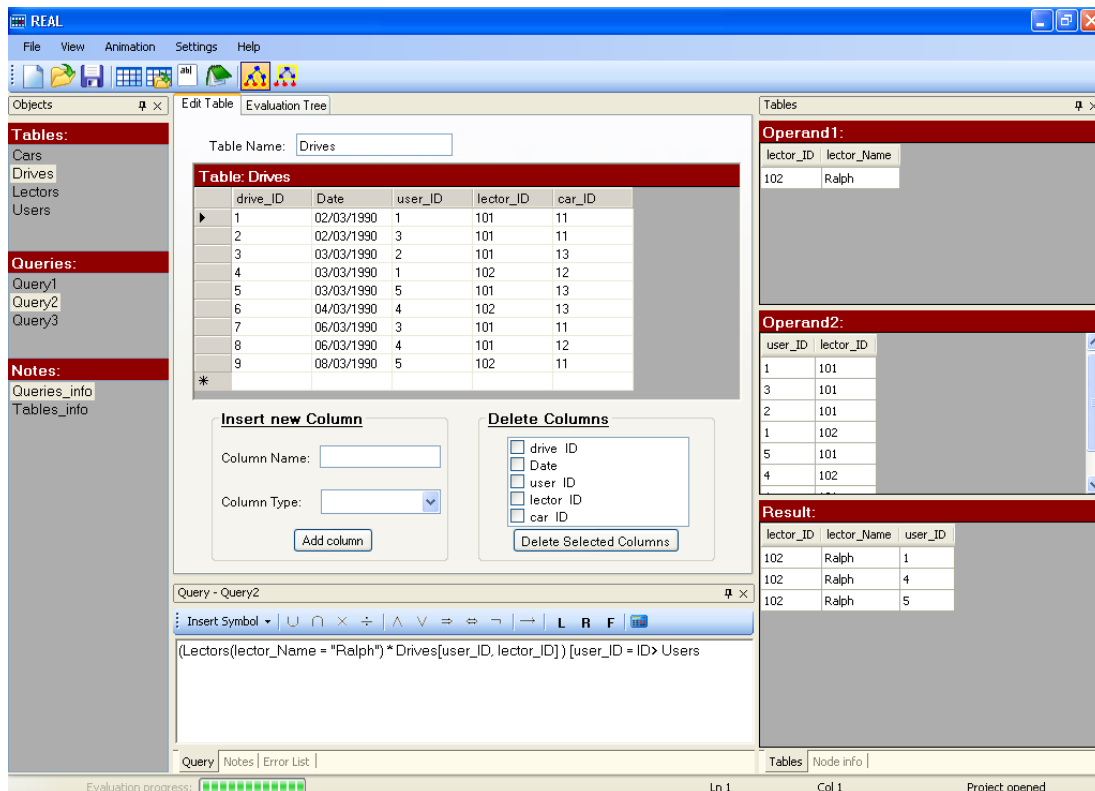
V inštalačnom adresári sa nachádzajú priečinky Examples a Help. Priečinok Examples obsahuje príklady projektov a priečinok Help obsahuje súbor s užívateľskou dokumentáciou REAL_help.chm, ktorá obsahuje užívateľský manuál s ilustračnými obrázkami.

4.3 Hlavné okno

Hlavné okno programu obsahuje hlavné menu a panel nástrojov (toolbar), ktoré sa nachádzajú v hornej časti okna. Hlavné menu umožňuje všetky hlavné funkcie programu a na paneli nástrojov sú tlačidlá pre rýchlejšie použitie najčastejšie používaných funkcií. V dolnej časti okna sa nachádza stavový riadok (statusbar), ktorý zobrazuje informácie o priebehu vyhodnocovania dotazu a informáciu o aktuálnom stave projektu.

V strede okna sa nachádza prepínací panel, ktorý umožňuje prepínať medzi panelom na editáciu tabuliek relácií a panelom, ktorý zobrazuje vyhodnocovací strom dotazu. Tento prepínací panel je vždy umiestnený v strede okna medzi ostatnými plávajúcimi panelmi. Naľavo sa nachádza panel **Objects**, ktorý obsahuje zoznam všetkých tabuliek, dotazov a poznámok projektu. V dolnej časti okna sa nachádzajú panely **Query**, **Notes** a **Error List**. Prvé dva umožňujú editáciu dotazu, respektíve poznámok. Panel Error List slúži na zobrazenie chýb vyskytujúcich sa vo vyhodnocovanom dotaze. V pravej časti okna sú panely **Tables** a **Node Info**, ktoré zobrazujú informácie o vybranom uzle vyhodnocovacieho stromu dotazu. Panel Tables zobrazí tabuľky relácií, ktoré sú vstupnými parametrami príslušnej operácie

a tiež tabuľku, ktorá je výsledkom tejto operácie prevedenej na príslušné operandy. Panel Node Info zobrazuje informácie o vybranom uzle, napríklad druh operácie, selekčnú alebo spájajúcu podmienku, typ spojenia atď. Všetky tieto plávajúce panely sa dajú zobrazovať a skrývať pomocou tlačidiel umiestnených v ich hlavičkách alebo pomocou položky **View|...** v hlavnom menu.



Obrázok 4.1 Ukážka hlavného okna programu REAL

4.4 Zakladanie, otváranie a ukladanie projektu

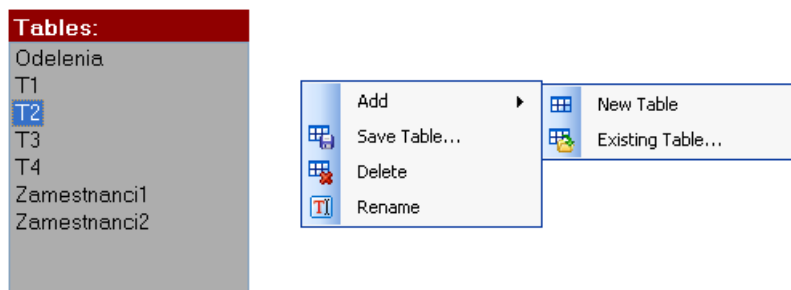
Nový projekt založíte voľbou **File|New** v hlavnom menu alebo pomocou tlačidla na hlavnom toolbare. Pred vytvorením nového projektu sa program opýta, či chcete uložiť rozpracovaný projekt. Po spustení programu bez zadania mena súboru, ktorý sa má otvoriť, program založí nový projekt.

Pre otvorenie už existujúceho projektu voľte **File|Open** v hlavnom menu alebo kliknite na odpovedajúce tlačidlo na toolbare. Ak ste neuložili zmeny v rozpracovanom projekte, program vás na to upozorní a poskytne vám možnosť uloženia rozpracovaného projektu. Následne v dialógu pre otvorenie súboru zvolíte súbor, ktorý chcete otvoriť.

Rozpracovaný projekt **uložíte** voľbou **File|Save** v hlavnom menu alebo odpovedajúcim tlačidlom na hlavnom toolbare. Ak daný projekt ukladáte prvýkrát, budete vyzvaní, aby ste v dialógu pre uloženie súboru zadali meno súboru, do ktorého chcete projekt uložiť. Pri každom ďalšom ukladaní sa projekt zapíše do tohto súboru. Ak by ste chceli rozpracovaný projekt uložiť do iného súboru, môžete tak urobiť pomocou **File|Save As** v hlavnom menu programu.

4.5 Práca s tabuľkami v projekte

Zoznam mien tabuliek je zobrazený v paneli **Objects** pod nadpisom **Tables**. Kliknutím pravým tlačidlom myši na tento zoznam otvoríte kontextové menu, ktoré poskytuje možnosti pre spravovanie zoznamu tabuliek.

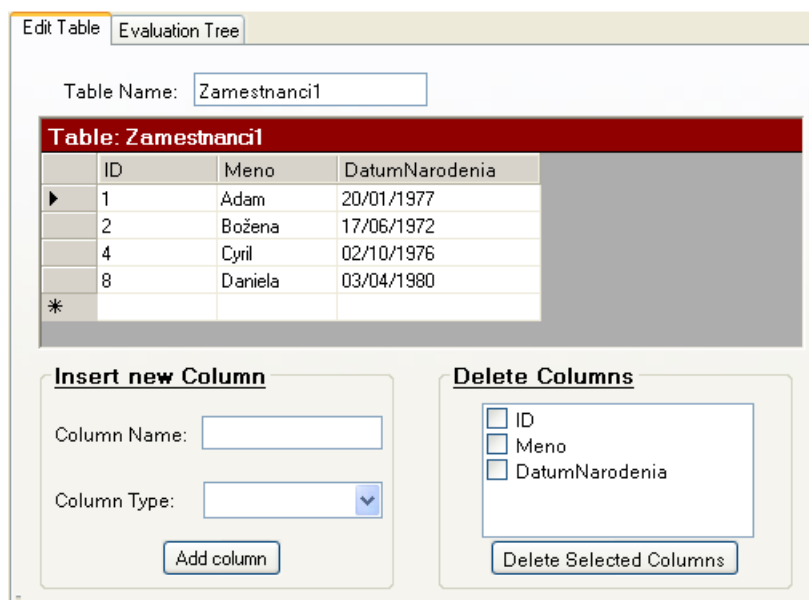


Obrázok 4.2 Zoznam tabuliek a jeho kontextové menu

Pomocou kontextového menu zoznamu tabuliek môžete do projektu pridávať nové tabuľky (**Add|New Table**) alebo existujúce tabuľky zo súboru (**Add|Existing Table**). Hotové tabuľky môžete ukladať do súborov (**Save Table**) a potom pridávať do iných projektov. Taktiež môžete tabuľky z projektu odstrániť (**Delete**) alebo ich premenovávať (**Rename**).

Tabuľky uložené v samostatných súboroch je možné do projektu pridať metódou *drag&drop*. Označené súbory s tabuľkami jednoducho potiahnutím myšou presuňte na zoznam tabuliek.

Meno tabuľky musí začínať písmenom a môže obsahovať len alfanumerické znaky bez diakritiky a podtržítka ('_'). Mená tabuliek nezávisia na veľkosti jednotlivých písmen v ich názve, preto názvy "Odelenia", "odelenia" alebo "ODELENIA" sa vzťahujú na jednu a tú istú tabuľku. Program nedovolí pomenovať viac tabuliek rovnakým menom.



Obrázok 4.3 Panel pre úpravu tabuliek

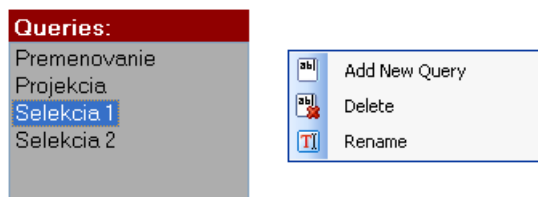
Tabuľku, ktorú vyberiete v zozname tabuliek, môžete upravovať v paneli **Edit Table** v hlavnom prepínacom paneli uprostred hlavného okna. V hornej časti panela môžete zmeniť meno upravovanej tabuľky. Do tabuľky môžete pridávať nové riadky

zadaním hodnôt do posledného riadku zobrazenej tabuľky. Kliknutím naľavo od riadku, na jeho hlavičku, sa vybraný riadok označí. Označené riadky môžete z tabuľky zmazať stlačením klávesy Delete.

Pod zobrazenou tabuľkou sa nachádzajú dva panely, ktoré umožňujú pridávať a mazať stĺpce tabuľky. Pre pridanie nového stĺpca do tabuľky zadajte v paneli **Insert new Column** jeho názov do políčka *Column Name* a dátový typ do políčka *Column Type*. Na meno stĺpca sú kladené rovnaké obmedzenia ako na meno tabuľky. Program nedovolí pridanie nesprávne zadaného stĺpca do tabuľky. Ak chcete nejaké stĺpce z tabuľky odstrániť, označte ich v zozname stĺpcov v paneli **Delete Columns** a stlačte tlačidlo *Delete Selected Columns*.

4.6 Práca s dotazmi

Zoznam dotazov je umiestnený v paneli **Objects** pod nadpisom **Queries**. Kliknutím pravým tlačidlom myši otvoríte kontextové menu zoznamu dotazov, ktoré obsahuje funkcie pre správu zoznamu dotazov. Umožňuje pridávanie nových dotazov do projektu (**Add New Query**), mazanie vybraných dotazov z projektu (**Delete**) alebo premenovávanie dotazov (**Rename**). Na názvy dotazov nie sú kladené žiadne špeciálne obmedzenia, akurát projekt nemôže obsahovať viac dotazov s rovnakým názvom.



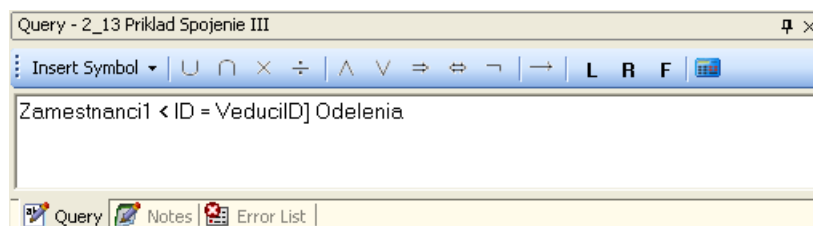
Obrázok 4.4 Zoznam dotazov a jeho kontextové menu

Vybraný dotaz sa zobrazí v paneli **Query** a môžete ho upravovať ako text. Z klávesnice môžete zadávať štandardné znaky dotazu. Špeciálne symboly ako množinové operátory, logické operátory a iné pomocné symboly, môžete do dotazu vkladať pomocou voľby **Query|Insert|...** v hlavnom menu alebo pomocou toolbaru umiestneného v hornej časti panela Query.

V texte dotazu možno miesto špeciálnych symbolov použiť takzvané tagy. Tieto reprezentujú jednotlivé symboly v dotaze. Pomocou tagov možno každý dotaz zapísať v čisto textovej podobe bez špeciálnych symbolov. Tagy pre jednotlivé symboly sú nasledovné:

\cup - <code>\union</code>	\cap - <code>\intersection</code>	\times - <code>\crossproduct</code>
\div - <code>\division</code>	\wedge - <code>&&</code>	\vee - <code> </code>
\Rightarrow - <code>\imp</code>	\Leftrightarrow - <code>\eq</code>	\neg - <code>!</code>
\perp - <code>\l</code>	\Re - <code>\r</code>	\mathbb{F} - <code>\f</code>
\rightarrow - <code>\arrow</code>		

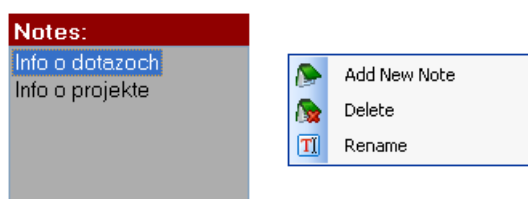
V rámci editácie dotazu je možné využiť funkcie *undo* a *redo*, pre vrátenie posledných zmien alebo pre obnovenie vrátených zmien, ktoré je možné zvoliť v kontextovom menu panela na editáciu dotazu. Pri výbere iného dotazu sa zmeny aplikované na predošlý dotaz vymažú zo zásobníka pre *undo* aj *redo*. Pri práci s dotazom môžete tiež využívať funkcie pracujúce so schránkou Windows a to *cut* (vystrihnúť), *copy* (kopírovať) a *paste* (prilepiť), ktoré je možné zvoliť v kontextovom menu.



Obrázok 4.5 Panel pre úpravu dotazov

4.7 Práca s poznámkami

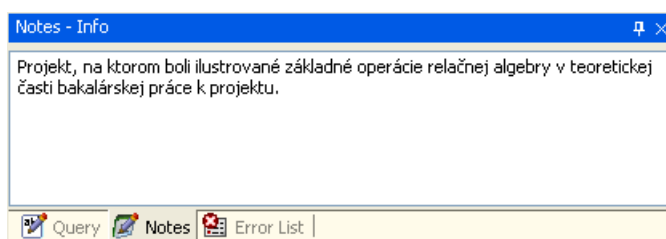
Poznámky v projekte môžu slúžiť ako alternatíva komentárov, ktoré v textoch dotazov niesú povolené. Tiež pomocou nich môžete popísať význam jednotlivých tabuliek alebo prácu jednotlivých dotazov.



Obrázok 4.6 Zoznam poznámok a jeho kontextové menu

Zoznam poznámok je zobrazený v paneli **Objects** pod nadpisom **Notes**. Súčasťou zoznamu je kontextové menu, ktoré sprostredkúva funkcie pre spravovanie poznámok v rámci projektu. Pridávanie (**Add New Note**), mazanie (**Delete**) alebo premenovávanie (**Rename**) funguje rovnako ako pri práci so zoznamom dotazov.

Text vybranej poznámky je zobrazený v paneli **Notes** a priamo v ňom je možné upravovať text vybranej poznámky. Pri editácii poznámok je tiež možné využiť schránku Windows a pracovať s funkciami *cut*, *copy* a *paste*, ktoré sú prístupné cez kontextové menu editačného panela.



Obrázok 4.7 Panel pre úpravu poznámok

4.8 Vyhodnotenie dotazu

Program akceptuje dotazy zapísané v syntaxi popísanej v kapitole 2. Dokončený dotaz vyhodnotíte zvolením **Query|Evaluate** v hlavnom menu alebo tlačidlom na toolbare v paneli **Query**. Ak dotaz obsahoval nejakú syntaktickú chybu, vyhodnocovanie skončí a bude zobrazený panel **Error List** s hláškou o nájdenej chybe a jej pozícii v rámci dotazu. Po dvojkliku na chybovú hlášku sa kurzor presmeruje na pozíciu chyby v dotaze v paneli Query.

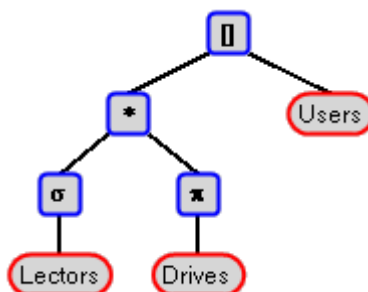
Ak je zadaný dotaz syntakticky správne, tak sa začne postupne vyhodnocovať. Ak sa počas vyhodnocovania vyskytnú nejaké sémantické chyby, vyhodnotenie dotazu sa nedokončí a informácie o nájdených sémantických chybách budú zobrazené v paneli

Error List. Opäť sa dvojklikom na príslušnú chybovú hlášku presunie kurzor na pozíciu chyby v dotaze.

#	Description	Line	Column
1	DatmNarodenia is not column name of corresponding table	1	16

Obrázok 4.8 Panel zobrazujúci chyby vo vyhodnocovanom dotaze

Ak je zadaný dotaz správne aj sémanticky zobrazí sa panel **Evaluation Tree** v prepínacom paneli v strede okna programu a do ňo sa vykreslí vyhodnocovací strom dotazu. V paneli **Tables** sa v tabuľke **Result** zobrazí tabuľka výslednej relácie dotazu.



Obrázok 4.9 Príklad vyhodnocovacieho stromu dotazu

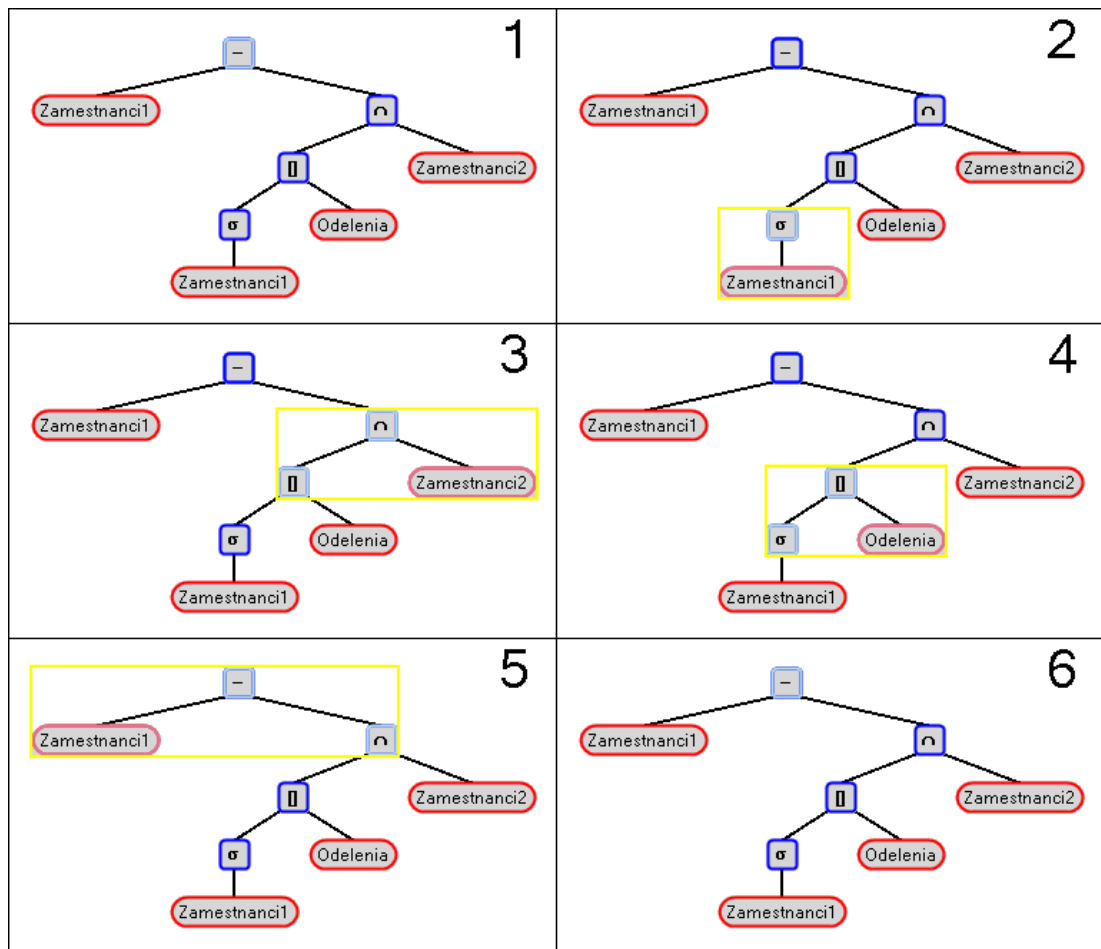
Po úspešnom vyhodnotení dotazu možno kliknutím ľavým tlačidlom myši vybrať ľubovoľný uzol stromu a v paneli Tables sa zobrazia tabuľky relácií operandov vstupujúcich do operácie a príslušná výsledná relácia. V paneli Node Info sa zobrazia informácie o vybranom uzle. Ak vybraný uzol reprezentuje názov relácie, tak sa v paneli Tables zobrazí len jedna tabuľka, ktorá reprezentuje túto reláciu. Ak pôvodná tabuľka obsahovala nejaké duplicitné riadky, tak budú odstránené z tabuľky zobrazenej v paneli Tables, aby bolo splnené chovanie relácií ako množín.

4.6 Vizualizácia medzivýsledkov

Program umožňuje postupné zobrazenie medzivýsledkov v paneli Tables v poradí v akom boli vyhodnocované v rámci vyhodnocovania dotazu. V hlavnom menu treba zaškrtnúť položku **Animation|SubTree Animation** a potom po vybratí ľubovoľného uzlu stromu sa vytvorí animácia medzivýsledkov pre podstrom, ktorý reprezentuje vybraný uzol.

Animácia obsahuje pôvodný stav vyhodnocovacieho stromu ako prvý a posledný snímok animácie. Medzi nimi sa nachádzajú snímky reprezentujúce vyhodnotenie jednotlivých operácií podstromu v poradí, v akom boli vyhodnocované pri vyhodnocovaní dotazu. Pre každý uzol podstromu reprezentujúci operáciu sa v paneli Tables zobrazia tabuľky reprezentujúce relácie operandov (tabuľky **Operand1** a **Operand2**), z ktorých táto operácia vytvára výslednú reláciu (tabuľka **Result**). Uzol operácie, ktorej medzivýsledky sú zobrazované je spolu so svojimi potomkami zobrazený v žltom ráme na obrázku vyhodnocovacieho stromu. Na obrázku 4.10 je zobrazený príklad priebehu vykresľovania vyhodnocovacieho stromu v rámci

animácie medzivýsledkov.



Obrázok 4.10 Príklad vizualizácie medzivýsledkov

4.6.1 Ovládanie animácie

Animácia sa ovláda pomocou panela zobrazeného na obrázku 4.11. Tento je umiestnený pod panelom, na ktorý sa vykresľuje vyhodnocovací strom. Na ľavej strane je posuvník určujúci pozíciu v animácii. Na pravej strane je posuvník určujúci rýchlosť animácie. Dolnú a hornú hranicu časového intervalu medzi jednotlivými snímkami môžete nastaviť v nastaveniach projektu. Tlačidlá v strede panela umožňujú navigáciu v animácii. Ľavé dve tlačidlá umožňujú posun v animácii o krok dopredu (*Step forward*), respektíve o krok dozadu (*Step back*). Prostredné tlačidlo (*Play/Pause*) spustí prehrávanie animácie z aktuálnej pozície animácie. Ak sa animácia prehráva, týmto tlačidlom sa tiež pozastaví prehrávanie. Dve tlačidlá napravo umožňujú rýchly posun na začiatok animácie (*Jump to begin*), respektíve na koniec animácie (*Jump to end*). Pozíciu animácie je tiež možné meniť priamo posúvaním posuvníka určujúceho pozíciu v animácii na ľavej strane panela.



Obrázok 4.11 Panel pre ovládanie animácie

4.7 Vizualizácia vyhodnocovania operácií

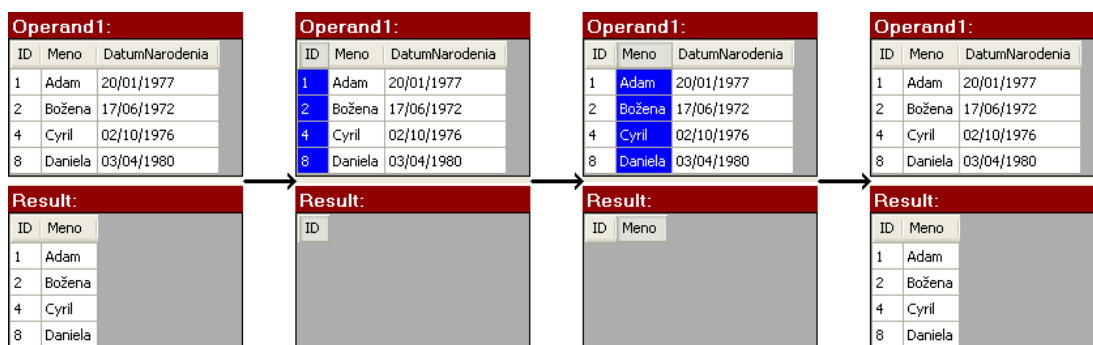
Program REAL umožňuje tiež vizualizáciu vyhodnocovania konkrétnej operácie s konkrétnymi operandami. V hlavnom menu treba zaškrtnúť položku **Animation|Operation Animation**, čím sa nastaví tento druh animácie. Po zvolení ľubovoľného uzla vyhodnocovacieho stromu sa vytvorí animácia príslušnej operácie, ktorú daný uzol reprezentuje. Ak zvolený uzol reprezentuje názov relácie, tak sa preň vytvorí animácia obsahujúca len 2 snímky, ktoré zobrazujú tabuľku tejto relácie.

Podobne ako pri animácii medzivýsledkov obsahuje aj táto animácia ako prvý a posledný snímok pôvodný stav vyhodnocovacieho stromu a zobrazovaných tabuliek v paneli Tables. Ostatné snímky zobrazia žltý rám okolo vybraného uzlu a uzlov reprezentujúcich jeho operandu a zvýrazia niektorý riadok alebo stĺpec v tabuľkách v paneli Tables podľa príslušnej operácie. Ak chcete aby boli zvýraznené riadky tabuliek vždy viditeľné počas animácie, zaškrtnite **Animation|Tables AutoScroll** v hlavnom menu. Ovládanie animácie je popísané v časti 4.6.1.

4.7.1 Animácia pre projekciu a premenovanie

Tieto operácie relačnej algebry pracujú skôr s atribútmi ako s prvkami relácie operandu. Pri vizualizácii týchto operácií sa postupne zvýrazňujú stĺpce tabuľky *Operand1* a tabuľky *Result* v paneli Tables, ktoré reprezentujú reláciu vstupného operandu a výslednú reláciu.

Pri **projekcii** sa postupne v tabuľke *Operand1* zvýrazňujú stĺpce, uvedené v zozname identifikátorov atribútov a hlavičky týchto stĺpcov sú súčasne pridávané do tabuľky výslednej relácie. Nakoniec sa výsledná tabuľka naplní odpovedajúcimi hodnotami. Na obrázku 4.12 sú zobrazené jednotlivé kroky animácie projekcie z príkladu na obrázku 2.3. Postupne sú do výslednej tabuľky pridané stĺpce *ID* a *Meno*. V poslednom snímku je zobrazená výsledná tabuľka aj s hodnotami po odstránení duplicitných riadkov.



Obrázok 4.12 Príklad animácie pre projekciu

Pri **premenovaní** sa taktiež postupne zvýrazňujú stĺpce v tabuľke operandu, zároveň je prevedené premenovanie dotknutého atribútu a v tabuľke reprezentujúcej výslednú reláciu je už zobrazené nové meno atribútu. Mená atribútov sa postupne menia aj v tabuľke reprezentujúcej reláciu operandu tak, aby v každom kroku animácie tabuľka operandu odpovedala relácii, na ktorú sa dané premenovacie pravidlo aplikuje.

Operand 1:		
ID	Meno	DatumNarodenia
1	Adam	20/01/1977
2	Božena	17/06/1972
4	Cyril	02/10/1976
8	Daniela	03/04/1980

Result:		
ID	Name	Birthday
1	Adam	20/01/1977
2	Božena	17/06/1972
4	Cyril	02/10/1976
8	Daniela	03/04/1980

Obrázok 4.13 Príklad animácie pre premenovanie atribútov

V animácii operácie premenovania atribútov z príkladu na obrázku 2.4, sa najprv premenuje atribút *Meno* na *Name*. V treťom kroku vystupuje ako operand tabuľka so stĺpcom *Name*, pretože druhé premenovacie pravidlo už bude aplikované na túto tabuľku. Potom sa v tejto tabuľke premenuje *DatumNarodenia* na *Birthday*.

4.7.2 Animácia pre selekciu

V animácii sa postupne prechádzajú všetky riadky tabuľky relácie operandu a vyhodnocuje sa selekčná podmienka. Ak je selekčná podmienka pre daný riadok splnená, riadok je zvýraznený zelenou farbou a pridaný do tabuľky výslednej relácie, v ktorej je tiež zvýraznený zelenou farbou. Ak pre daný riadok nie je splnená selekčná podmienka, je tento riadok zvýraznený červenou farbou a v tomto prípade sa nepridáva do tabuľky výslednej relácie.

Na obrázku sú zobrazené jednotlivé časti animácie pre selekciu uvedenú na obrázku 2.2. Pre prvý a štvrtý riadok je splnená selekčná podmienka, preto sú zvýraznené zelenou farbou a sú pridané do výslednej tabuľky. Druhý a tretí riadok selekčnú podmienku nespĺňajú.

Operand 1:		
ID	Meno	DatumNarodenia
1	Adam	20/01/1977
2	Božena	17/06/1972
4	Cyril	02/10/1976
8	Daniela	03/04/1980

Result:		
ID	Meno	DatumNarodenia
1	Adam	20/01/1977
8	Daniela	03/04/1980

Obrázok 4.14 Príklad animácie pre selekciu

4.7.3 Animácia množinových operácií

Pri animovaní vyhodnocovania **zjednotenia** sa najprv postupne pridávajú do výslednej relácie všetky riadky tabuľky prvého operandu (*Operand1*). Tieto sú zvýraznené zelenou farbou ako v tabuľke prvého operandu, tak v tabuľke výslednej relácie. Potom sa začnú postupne prechádzať riadky tabuľky druhého operandu (*Operand2*) a do tabuľky výslednej relácie sa pridávajú tie, ktoré do výsledku ešte neboli zaradené. Pridávané riadky sú zvýraznené zelenou farbou v tabuľke druhého operandu aj výsledku. Ak výsledná tabuľka už obsahuje daný riadok tabuľky druhého operandu, je tento zvýraznený červenou farbou a taktiež je zvýraznený na červeno kolidujúci riadok výslednej tabuľky.

Operand1:	Operand1:	Operand1:	Operand1:	Operand1:	Operand1:												
ID	Meno	DatumNarodenia	ID	Meno	DatumNarodenia												
1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977
2	Božena	17/06/1972	2	Božena	17/06/1972	2	Božena	17/06/1972	2	Božena	17/06/1972	2	Božena	17/06/1972	2	Božena	17/06/1972
4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976
8	Daniela	03/04/1980	8	Daniela	03/04/1980	8	Daniela	03/04/1980	8	Daniela	03/04/1980	8	Daniela	03/04/1980	8	Daniela	03/04/1980

Operand2:	Operand2:	Operand2:	Operand2:	Operand2:	Operand2:												
ID	Meno	DatumNarodenia	ID	Meno	DatumNarodenia												
1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977
3	Boris	14/02/1981	3	Boris	14/02/1981	3	Boris	14/02/1981	3	Boris	14/02/1981	3	Boris	14/02/1981	3	Boris	14/02/1981
4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976

Result:	Result:	Result:	Result:	Result:	Result:												
ID	Meno	DatumNarodenia	ID	Meno	DatumNarodenia												
1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977	1	Adam	20/01/1977
4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976	4	Cyrl	02/10/1976

Obrázok 4.15 Animácia vyhodnocovania prieniku relácií Zamestnanci1 a Zamestnanci2

Pri animovaní **prieniku** sa prechádzajú riadky tabuľky prvej relácie. Riadky, ktoré sa nachádzajú aj v tabuľke druhého operandu, sú pridané do výslednej tabuľky a vo všetkých troch tabuľkách sú zvýraznené zelenou farbou. Riadky tabuľky prvého operandu, ktoré sa nenachádzajú v tabuľke druhého operandu, a teda nie sú pridané do výslednej tabuľky, sú v tabuľke prvého operandu zvýraznené červenou farbou.

V animácii **množinového rozdielu** je to presne opačne ako pri animácii prieniku. Riadky nachádzajúce sa v oboch tabuľkách operandov, sú označené červenou farbou a nie sú pridané do výslednej tabuľky. Naopak riadky tabuľky prvého operandu, ktoré neobsahuje tabuľka druhého operandu, sú zvýraznené zelenou farbou v tabuľke prvého operandu a pridané do tabuľky výsledku.

Pri operácii **kartézskeho súčinu** pozostáva animácia z postupného prechodu všetkými riadkami prvej a druhej tabuľky a spojený riadok sa pridá do výslednej tabuľky. Dotknuté riadky sú zvýraznené zelenou farbou vo všetkých troch tabuľkách.

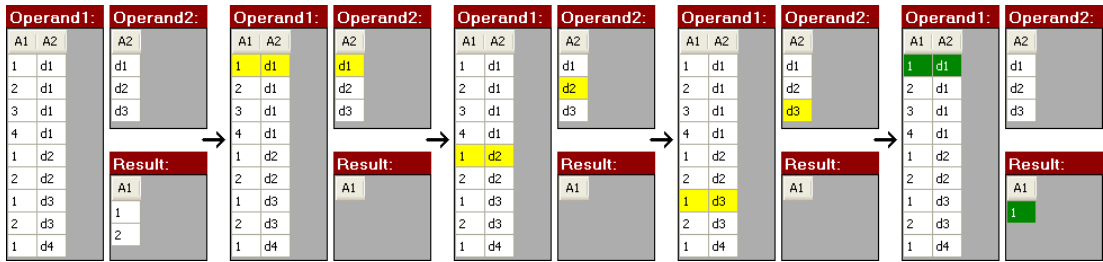
4.7.4 Animácia pre delenie

Pre všetky riadky tabuľky výslednej relácie po delení a pre všetky riadky tabuľky druhého operandu musí v tabuľke prvého operandu existovať riadok, ktorý dostaneme spojením týchto riadkov.

V animácii sa prechádza každý riadok tabuľky reprezentujúcej prvý operand a porovnáva sa, či sa zhoduje v posledných atribútoch s prvým riadkom tabuľky druhého operandu. Ak sa nezhoduje je riadok prvej tabuľky zvýraznený červenou farbou a pokračuje sa nasledujúcim riadkom prvej tabuľky.

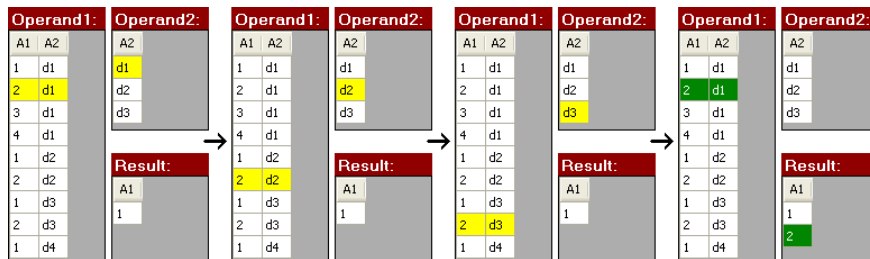
Ak sa zhoduje v posledných atribútoch s prvým riadkom druhej tabuľky, sú obidva riadky označené žltou farbou. Začiatkové atribúty tohto riadku sa stávajú kandidátom pre výslednú tabuľku. Označme tento podriadok názvom *riadokA*. Najprv sa ale pre každý ďalší riadok (označme napríklad *riadokB*) druhej tabuľky musí nájsť v tabuľke prvého operandu riadok, ktorý vznikne spojením *riadkuA* a *riadkuB*. Teda nasleduje prechod riadkami tabuľky druhého operandu a hľadanie riadku, ktorý spĺňa popísanú vlastnosť v tabuľke prvého operandu. Ak je tento riadok v prvej tabuľke nájdený, označí sa spolu s odpovedajúcim riadkom tabuľky druhého operandu žltou farbou, ktorá signalizuje, že *riadokA* je stále kandidátom na pridanie do výslednej tabuľky. Ak by nebola pre podriadok *riadokA* a niektorý riadok z tabuľky druhého operandu splnená uvedená podmienka, zvýrazní sa pôvodný riadok tabuľky prvého operandu červenou farbou a pokračuje sa nasledujúcim riadkom prvej tabuľky.

Ak podmienka pre podriadok *riadokA* bola splnená pre každý riadok tabuľky druhého operandu, zvýrazní sa pôvodný riadok tabuľky prvej relácie zelenou farbou a pridá sa do výslednej tabuľky.



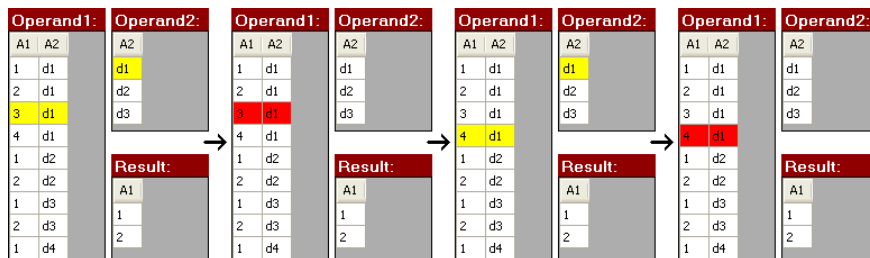
Obrázok 4.16 Animácia podielu relácií $T1 \div T2$, 1. časť

Na obrázku 4.16 je zobrazený začiatok vizualizácie vyhodnocovania operácie delenia na reláciach $T1$ a $T2$ zadaných na obrázku 2.8. Začína sa prvým riadkom tabuľky prvého operandu. Ten sa zhoduje v atribúte $A2$ s prvým riadkom tabuľky druhého operandu. Takže atribút $A1$ prvého riadku je kandidátom na pridanie do výslednej tabuľky a oba riadky sú zvýraznené žltou farbou. V ďalších dvoch krokoch sa pokúšame v tabuľke prvého operandu nájsť riadok, ktorého hodnota v atribúte $A1$ by bola 1 a hodnota atribútu $A2$ by bola zhodná s druhým respektíve tretím riadkom druhej tabuľky. Keďže aj pre druhý aj pre tretí riadok druhej tabuľky sme našli takýto riadok v prvej tabuľke, môžeme pridať hodnotu atribútu $A1$ prvého riadku tabuľky prvého operandu do výslednej tabuľky. Pri pridaní sú oba riadky zvýraznené zelenou farbou. Pokračuje sa druhým riadkom prvej tabuľky. Aj pre tento riadok sa v tabuľke prvého operandu nachádzajú odpovedajúce riadky. Animácia teda pokračuje nasledovne.



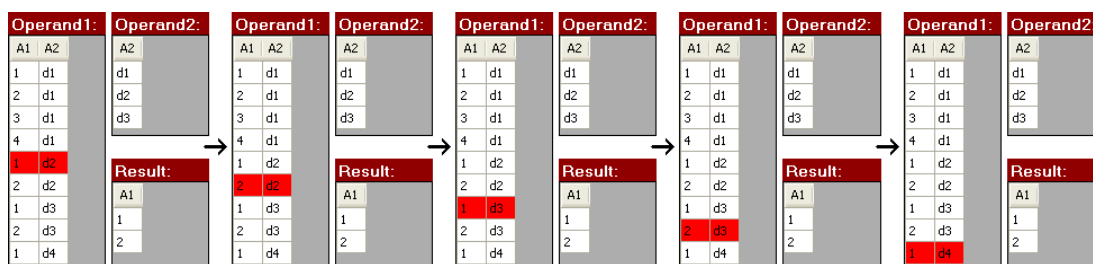
Obrázok 4.17 Animácia podielu relácií $T1 \div T2$, 2. časť

Pre tretí a štvrtý riadok prvej tabuľky bude síce splnená zhodnosť v atribúte $A2$ s prvým riadkom tabuľky druhého operandu, teda sú najprv zvýraznené žltou farbou. Ale už pre druhý riadok tabuľky druhého operandu sa v prvej tabuľke nenachádza riadok s hodnotou $d2$ atribútu $A2$ a odpovedajúcou hodnotou v atribúte $A1$. Preto sú následne zvýraznené červenou farbou a nie sú pridané do výslednej tabuľky.



Obrázok 4.18 Animácia podielu relácií $T1 \div T2$, 3. časť

Ostatné riadky tabuľky prvého operandu sa už nezhodujú v atribúte $A2$ s prvým riadkom tabuľky druhej relácie. Teda tieto nijakým spôsobom neprispievajú do výslednej tabuľky a sú postupne zvýrazňované červenou farbou.



Obrázok 4.19 Animácia podielu relácií $T1 \div T2$, 4. časť

Za tieto snímky je ešte pridaný snímok, ktorý zobrazuje tabuľky relácií operandov a úplnú výslednú tabuľku bez zvýraznených riadkov

4.7.5 Animácia pre operácie spojenia

Animácie pre podmienené spojenie a prirodzené spojenie sú realizované rovnakým spôsobom. V oboch sa pre každú dvojicu riadkov z tabuliek operandov testuje, či spĺňajú určitú podmienku. Či už je to spájacia podmienka pri podmienenom spojení alebo konjunkcia rovností všetkých rovnako pomenovaných atribútov relácií operandov pri prirodzenom spojení. Ak je podmienka splnená, pridá sa do výslednej tabuľky odpovedajúci riadok.

Pri **klasickej variante spojenia** sa postupne pre každý riadok tabuľky prvej relácie a pre každý riadok tabuľky druhej relácie skontroluje, či platí spájacia podmienka. Ak platí, je príslušný riadok pridaný do výslednej tabuľky a všetky tri riadky sú zvýraznené zelenou farbou. Ak pre dvojicu riadkov nieje podmienka splnená, sú tieto riadky zvýraznené červenou farbou a do výslednej tabuľky sa nič nepridáva.

Pri ľubovoľnej variante **vonkajšieho spojenia** je začiatok animácie rovnaký ako pri klasickej variante spojenia. Pri ľavom vonkajšom spojení sú na konci animácie pridané do výslednej tabuľky riadky tabuľky prvého operandu spojené s prázdny m riadkom také, že pre žiadny riadok tabuľky druhého operandu, nespĺnila táto dvojica riadkov spájajúcu podmienku. Riadok v tabuľke prvého operandu a rovnako aj novo pridaný riadok do výslednej tabuľky sú označené zelenou farbou. Pri pravom a úplnom vonkajšom spojení je to podobné.

Pri ľavom **polospojení** sa podobne ako pri klasickej variante spojenia v animácii prechádza cez všetky riadky tabuľky prvého operandu a cez všetky riadky druhého operandu. Ak je podmienka pre túto dvojicu riadkov splnená, pridá sa aktuálny riadok tabuľky prvého operandu do výslednej tabuľky. Všetky tri riadky sa označia zelenou farbou a animácia pokračuje nasledujúcim riadkom prvej tabuľky. Pre pravé polospojenie je animácia podobná, akurát tabuľka prvého operandu a tabuľka druhého operandu si vymenia úlohy.

4.8 Nastavenia

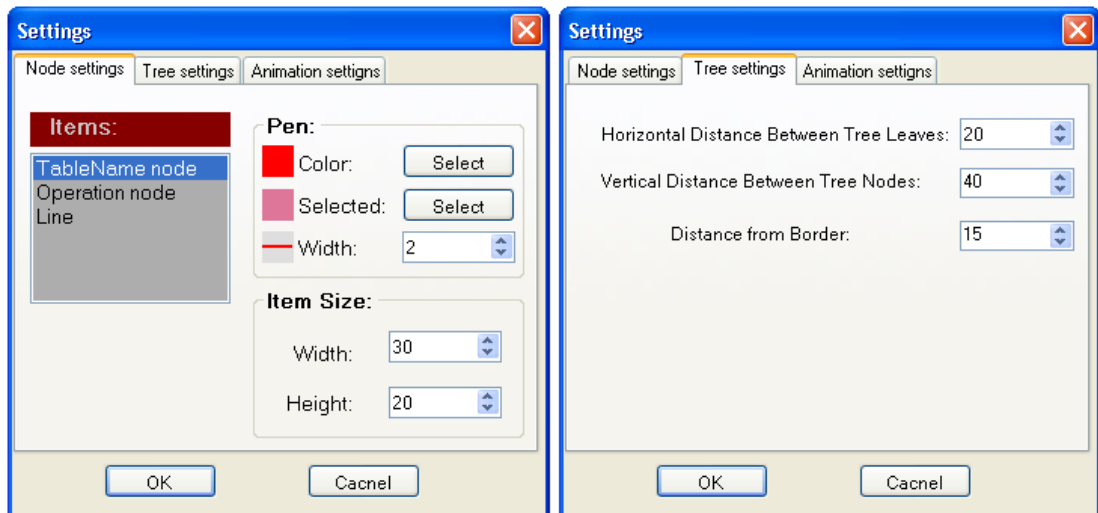
Voľbou [Settings|Change settings](#) v hlavnom menu môžete nastaviť parametre ovplyvňujúce vzhľad vykresleného vyhodnocovacieho stromu a tiež nastaviť dolnú a hornú hranicu pre časový interval medzi dvomi prehrávanými snímkami animácie.

V zozname *Items* vyberte prvok, ktorého vlastnosti chcete upraviť. Pre **uzly** vyhodnocovacieho stromu je možné nastaviť farbu, ktorou bude uzol vykreslený v normálnom stave (položka *Color*) a tiež farbu, akou bude uzol vykreslený, keď je označený užívateľom (položka *Selected*). Taktiež je možné nastaviť hrúbku pera,

ktorá bude použitá pri kreslení rámu pre uzol (položka *Width*). Pre uzly vyhodnocovacieho stromu môžeme tiež nastaviť ich rozmery (položka *Width* v paneli *Measures* pre šírku a položka *Height* pre výšku uzlu). Pre uzly reprezentujúce názvy relácií, údaj o šírke uzla označuje minimálnu šírku uzla, pretože šírka uzla je prispôbená veľkosti názvu odpovedajúcej relácie.

Pre čiary, ktoré reprezentujú **hrany** vo vyhodnocovacom strome je možné nastaviť hrúbku a farbu pera, ktorým sa budú tieto čiary kresliť.

V paneli **Tree Settings** prepínacieho panela v dialógu nastavení je možné nastaviť rozostupy medzi uzlami vyhodnocovacieho stromu.

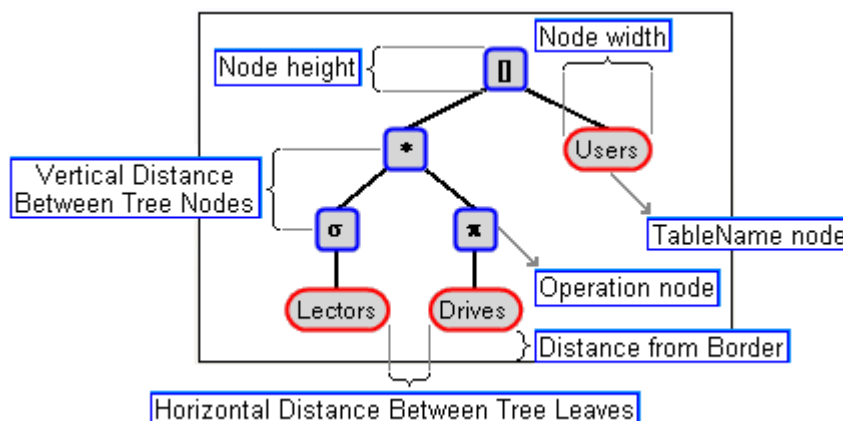


Obrázok 4.20 Dialog nastavení ovplyvňujúcich vzhľad vyhodnocovacieho stromu

Horizontal Distance Between Tree Leaves určuje horizontálnu vzdialenosť v pixeloch medzi susednými listami – uzlami reprezentujúcimi názvy relácií.

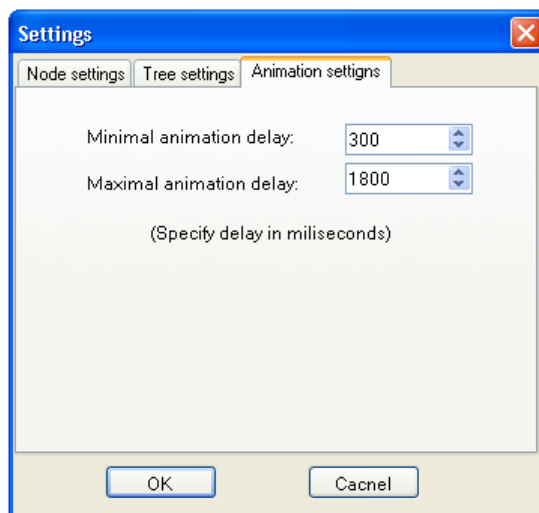
Vertical Distance Between Tree Nodes určuje vertikálnu vzdialenosť v pixeloch medzi stredmi uzlov v susedných úrovniach. Táto vzdialenosť nemôže byť ľubovoľná, pretože keby bola príliš malá uzly by sa mohli prekrývať. Preto je zdola automaticky obmedzená na súčet polovičnej výšky uzla operácie a maxima z polovičnej výšky uzla relácie alebo z polovičnej výšky uzla operácie. To sú dve kombinácie typov uzlov, ktoré sa môžu vyskytnúť nad sebou vo vyhodnocovacom strome.

Distance from Border určuje vzdialenosť od okraja v pixeloch.



Obrázok 4.21 Význam parametrov ovplyvňujúcich vzhľad vyhodnocovacieho stromu

V treťom paneli **Animation Settings** možno nastaviť dolnú a hornú hranicu časového intervalu medzi jednotlivými snímkami pri prehrávaní animácie. Na základe týchto hodnôt budú hodnoty posuvníka určujúceho rýchlosť animácie odpovedať hodnotám, ktoré dostaneme rovnomerným rozdelením intervalu, určeného minimálnou a maximálnou dobou medzi jednotlivými snímkami, do 6 hodnôt vrátane dolnej a hornej hranice.



Obrázok 4.22 Nastavenia pre animáciu

Zvolené nastavenia sa vzťahujú len na práve spracovávaný projekt, ku ktorému budú v prípade uloženia projektu pripojené. Avšak nastavenia je možné uložiť aj do samostatného súboru. Voľbou **Settings|Save settings** v hlavnom menu programu sa otvorí dialog pre uloženie súboru, ktorý umožní zadanie názvu súboru a určenie miesta, kam sa majú nastavenia uložiť. Uložené nastavenia možno potom aplikovať na ľubovoľný projekt voľbou **Settings|Load settings** v hlavnom menu. V prípade, že chcete pre aktuálny projekt zmeniť nastavenia na základné nastavenia, zvolíte v hlavnom menu **Settings|Default settings**.

5 GRAFICKÉ UŽIVATEĽSKÉ ROZHRAINIE

Program REAL bol napísaný v jazyku C# vo vývojovom prostredí Microsoft Visual Studio 2005. Priamo prostredie .NET poskytuje mnoho užitočných komponent pre užívateľské rozhranie, napríklad triedy DataGrid, respektíve DataGridView pre jednoduché zobrazovanie a editáciu tabuliek z pohľadu užívateľa alebo triedy pre prácu s grafikou vhodné na vykresľovanie vyhodnocovacieho stromu dotazu.

5.1 Rozvrhnutie aplikácie

Hlavné okno aplikácie (obrázok 4.1) som sa snažil rozvrhnúť tak aby komponenty, ktoré by mohli byť využívané súčasne, mohli byť aj súčasne k dispozícii. V strede okna sa nachádza prepínací panel, ktorého prvý panel **Edit Table** obsahuje prvky pre editáciu relácie ako tabuľky. Pridávanie nových stĺpcov, mazanie stĺpcov alebo pridávanie a mazanie riadkov. Druhý panel **Evaluation Tree** obsahuje plochu na ktorú sa vykresľuje vyhodnocovací strom. Tento prepínací panel je vždy viditeľný narozdiel od ostatných plávajúcich panelov, ktoré je možné skrývať a zasúvať, a jeho veľkosť je určená veľkosťou voľného miesta v strednej časti okna, ktorú vyplňa.

Pri skúmaní vyhodnocovacieho stromu si užívateľ môže zobrazit' tabuľku základnej relácie kliknutím na uzol stromu, ktorý odpovedá názvu relácie na pravo v paneli **Tables**. Tak isto si môže zobrazit' kliknutím na uzol reprezentujúci operáciu jej medzivýsledky. Na pravej strane okna sa nachádza aj plávajúci panel **Node Info**, ktorý obsahuje informácie o práve vybranom uzle vyhodnocovacieho stromu.

V ľavej strane okna sa nachádza plávajúci panel **Objects**, ktorý obsahuje zoznam objektov projektu. Vybraná tabuľka sa zobrazí v paneli Edit Table a môže byť modifikovaná. V dolnej časti sa nachádzajú plávajúce panely **Query**, **Notes** a **Error List**. V paneli Query je zobrazený vybraný dotaz zo zoznamu dotazov v paneli Objects. Panel Notes zobrazuje vybranú poznámku zo zoznamu poznámok. Panel Error List slúži na zobrazenie syntaktických alebo sémantických chýb zistených pri vyhodnocovaní dotazu.

Všetky plávajúce panely môžu meniť svoje rozmery, dokonca aj pozíciu v okne. Užívateľ ich môže zobrazovať a skrývať. Takže si užívateľ môže rozmiestniť plávajúce panely tak, ako to najviac vyhovuje jemu. Po ukončení programu sa rozmiestnenie jednotlivých plávajúcich panelov uloží a pri ďalšom spustení programu sa tieto panely rozmiestnia podľa uloženej konfigurácie.

Prácu s plávajúcimi panelmi zaisťuje trieda DockingManager z knižnice MagicLibrary.dll verzia 1.7.4 [4], ktorá je prístupná zadarmo. Táto trieda zobrazuje plávajúce panely do vnútornej časti hlavného okna a automaticky koriguje rozmery tzv. **vnútornej komponenty**, čo je v našom prípade prepínací panel umiestnený vždy v strednej časti hlavného okna.

Hlavné okno aplikácie tiež obsahuje klasické prvky okienkových aplikácií ako hlavné menu a hlavný toolbar v hornej časti okna a stavový riadok v dolnej časti okna.

5.2 Spravovanie objektov projektu

Objekty používané v projekte sú zobrazené v paneli **Objects**, ktorý obsahuje pre každý zoznam vlastné kontextové menu, pomocou ktorého sa dajú jednoducho

pridávať nové prvky do konkrétneho zoznamu, premenovávať prvky a tiež odstraňovať prvky. Tabuľky relácií sú uložené v inštancii triedy DataSet, ktorá umožňuje jednoduchú prácu so súborom tabuliek. Pre uchovanie a prácu s Dotazmi a Poznámkami som využil triedu HashTable, kde ako vyhľadávacie kľúče boli použité názvy jednotlivých dotazov, respektíve poznámok. Počas práce s projektom sú všetky tabuľky, dotazy a ostatné prvky projektu držané v pamäti počítača. Keďže sa jedná o výukovú aplikáciu, nepredpokladá sa práca s rozsiahlymi projektami obsahujúcimi veľké množstvo tabuliek.

5.3 Editácia tabuliek

Jednoduchý editor tabuliek sa nachádza v paneli **Edit Table** (obrázok 4.3) v strede hlavného okna. Aktuálne vybraná tabuľka je previazaná s objektom DataGrid, ktorý zaisťuje zobrazenie tabuľky v aktuálnom stave. Operácie pridávania, mazania stĺpcov ovládané pomocou dvoch pomocných panelov (**Insert New Column** a **Delete Columns**) sú realizované len na tabuľke, zmeny sa vďaka previazaniu tabuľky s DataGrid-om prejavia následne aj na zobrazenej tabuľke. Trieda DataGrid zaručí aj správnosť zadávaných dát do políčok tabuľky, keďže nedovolí pridať hodnotu iného typu, než zvoleného typu pre daný stĺpec.

Názvy tabuliek (relácií) a názvy ich stĺpcov (atribútov) môžu obsahovať len alfanumerické znaky bez diakritiky a znak podtržítka ('_') a musia začínať písmenom. Je to hlavne z dôvodu jednoznačného odlišenia v dotazoch a v selekčných alebo spájacích podmienkach, kde by mohlo byť nejednoznačné či sa jedná o konštantu alebo o názov atribútu.

5.4 Editácia dotazov a poznámok

V paneli **Query** sa nachádza jednoduchý editor dotazov (obrázok 4.5). Je v ňom zobrazený vybraný dotaz zo zoznamu dotazov v paneli Objects. Dotazy relačnej algebry obsahujú ako štandardné znaky klávesnice, tak aj špeciálne znaky reprezentujúce väčšinou množinové operátory alebo logické operátory používané v podmienkach. Dotaz sa zapisuje do komponenty RichTextBox, ktorá umožňuje zobrazenie textu vo formáte Rich Text Format a teda aj zobrazenie textu, ktorý obsahuje znaky písané rôznym fontom. Pre zápis dotazu potrebujeme nejaký štandardný font na písanie klasického textu, potom font Symbol na písanie špeciálnych znakov a tiež font, ktorým budeme písať znaky L , R a F , ktoré sa používajú pre vonkajšie spojenia.

Voľbu jedného z týchto troch druhov fontov nie je potrebné ponechať na užívateľa. Zbytočné by bolo neustále prepínanie druhu písma, hlavne ak chceme zadať len jeden špeciálny znak a potom opäť pokračovať štandardným písmom, čo chceme prakticky vždy vzhľadom k tomu, že špeciálnym druhom písma sa zapisujú len operátory. A vzhľadom k tomu, že počet špeciálnych symbolov, ktoré sa môžu vyskytovať v dotaze je len 13, lepším riešením je ponechať jeden základný druh písma a umožniť užívateľovi pridávať špeciálne znaky do dotazu pomocou hlavného menu alebo pomocou toolbaru v hornej časti panela Query.

Pri editácii dotazu môže užívateľ využívať štandardné funkcie RichTextBox-u ako jeho vlastné Undo a Redo, tak aj funkcie pre prácu so schránkou Windows ako vystrihovanie, kopírovanie, vkladanie a označ všetko. Všetky tieto funkcie sú ponechané na pôvodnú implementáciu triedy RichTextBox, len funkcia vkladania je

preťažená. Je to hlavne z dôvodu, že táto komponenta dovolí užívateľovi vkladať ľubovoľný text vo formáte Rich Text Format a my užívateľovi nepovoľujeme zadávanie ľubovoľných špeciálnych znakov. Preto pri vkladaní textu všetky nezadateľné symboly zobrazujeme hlavným fontom nášho editačného panela. Kopírovanie medzi viacerými inštanciami programu REAL funguje bez zmien textu dotazu.

Na editáciu poznámok slúži panel **Notes** (obrázok 4.7), ktorý zobrazuje vybranú poznámku a sprístupňuje ju na editáciu. Panel obsahuje len komponentu TextBox, ktorá umožňuje prácu s obvyčajným textom.

Podobne aj komponenta TextBox, ktorá je použitá na editáciu poznámok umožňuje prácu so schránkou Windows. Tieto funkcie som ponechal v pôvodnej implementácii, pretože samotná komponenta zaistí prevedenie kopírovaného formátu do obvyčajného textu.

5.5 Ukladanie a otváranie súborov

Projekt programu REAL možno kedykoľvek uložiť do súboru v binárnom formáte a potom ho z tohto súboru zase otvoriť. Pri voľbe uloženia projektu sa vytvorí inštancia triedy Project, ktorá bude zastrešovať tabuľky, dotazy, poznámky a tiež nastavenie parametrov pre vykresľovanie stromu. Následne sa táto trieda pomocou mechanizmu serializácie zapíše do zvoleného súboru. Súbor je serializovaný do binárneho formátu pomocou triedy BinaryFormatter. Pri otváraní projektu zo súboru je najprv pomocou deserializácie vytvorená inštancia triedy Project a následne z nej sú postupne načítané dáta pre jednotlivé komponenty programu. V prípade, že je súbor poškodený alebo neobsahuje korektné dáta, deserializácia vyvolá výnimku, ktorá bude zachytená a užívateľovi sa oznámi chyba pomocou chybového dialogu. V prípade chyby sa ponechá otvorený predošlý projekt.

Tabuľky samotné sa ukladajú do súboru vo formáte XML. Využívajú sa priamo funkcie ReadXml a WriteXml triedy DataTable.

Program tiež umožňuje uloženie nakresleného vyhodnocovacieho stromu do rastrového obrázkového súboru. Keďže obrázok vyhodnocovacieho stromu je uchovaný v inštancii triedy Bitmap, využijú sa jej metódy, ktoré umožňujú uloženie do rastrového obrázku.

Nastavenia použité v projekte je tiež možné ukladať samostatne a následne načítať zo súboru. Nastavenia sa podobne ako celý projekt ukladajú pomocou serializácie triedy Settings do binárneho formátu pomocou triedy BinaryFormatter. Užívateľ môže nastavenia jedného projektu samostatne uložiť a potom ich aplikovať na iné projekty.

5.6 Vykresľovanie vyhodnocovacieho stromu

Vyhodnocovací strom dotazu sa vykresľuje na vykresľovaciu plochu v paneli **Evaluation Tree**. Aby sa predišlo preblikávaniu, tak je vyhodnocovací strom najprv nakreslený do bitmapy uloženej v pamäti a až následne je prekreslený do panela. Ak je obrázok väčší ako panel do ktorého sa kreslí, tak sa vykresľuje len potrebná časť bitmapy.

Obrázok stromu sa mení, len pri novom vyhodnotení dotazu. Väčšinou je potrebné len zvýrazniť užívateľom vybraný uzol v strome, prípadne vykresliť rámik obklopujúci vybraný uzol a uzly reprezentujúce jeho potomkov pri animácii. Pri

označení, respektíve odznačení uzla v strome sa prekreslí len vybraný uzol buď ako označený alebo neo značený. Ani pri vykresľovaní rámika nieje nutné prekresľovať celý obrázok, stačí do neho prikresliť tento rámik.

Pri zobrazení obrázka ktorého veľkosť je väčšia ako veľkosť zobrazovacej plochy sa zobrazia posuvné lišty, ktoré umožňujú meniť viditeľnú časť obrázku. Pri zmene polohy niektorej posuvnej lišty sa nezneplatňuje celý panel, pretože to by sa najprv vykreslil tento panel a až potom by sa vykreslil obrázok vyhodnocovacieho stromu, čo by spôsobilo preblikávanie obrázku pri posúvaní niektorej lišty. Miesto toho je nový výrez obrázku vykreslený priamo na pôvodný obrázok.

Užívateľovi je ponechaná voľba niektorých parametrov, ktoré ovplyvňujú vzhľad vyhodnocovacieho stromu. Jedná sa hlavne o rozmery jednotlivých uzlov, farbu a šírku pera, ktorým sa bude uzol maľovať a tiež rozostupy medzi jednotlivými uzlami. Význam jednotlivých parametrov je uvedený v kapitole 4.8, respektíve na obrázku 4.21.

Vykresleniu vyhodnocovacieho stromu predchádza určenie súradníc stredu každého uzlu vyhodnocovacieho stromu. Algoritmus na určenie súradníc je rekurzívny a prechádza strom v poradí post order. V každom okamihu výpočtu si pamätáme x-ovú súradnicu ľavého okraja listu, ktorý budeme spracovávať najbližšie (**EndX**). Na začiatku je premenná EndX inicializovaná na veľkosť okraja. Počas výpočtu si tiež pamätáme aktuálnu úroveň vo vyhodnocovacom strome.

Y-ovú súradnicu stredu ľubovoľného uzlu získame sčítaním veľkosti okraja s číslom aktuálnej úrovne vynásobeným vertikálnou vzdialenosťou medzi stredmi uzlov v susedných úrovniach. X-ovú súradnicu určí algoritmus podľa typu uzla nasledovne:

Uzol reprezentuje názov relácie – určí sa šírka názvu relácie a podľa nej sa určí šírka tohto uzla. X-ovú súradnicu stredu získame sčítaním premennej EndX a polovice zo šírky tohto uzla. Nakoniec premennú EndX zvýšime o šírku tohto uzla a o horizontálnu vzdialenosť medzi susednými listami.

Uzol reprezentuje unárnu operáciu – najprv sa určia súradnice stredov uzlov v podstrome tohto uzla. X-ová súradnica stredu tohto uzla je rovnaká ako x-ová súradnica stredu jeho synovského uzla.

Uzol reprezentuje binárnu operáciu – najprv sa určia súradnice stredov uzlov v ľavom podstrome a potom v pravom podstrome. X-ová súradnica stredu tohto uzlu bude aritmetický priemer x-ových súradníc stredov jeho synovských uzlov.

Po určení súradníc stredov jednotlivých uzlov vyhodnocovacieho stromu, môžeme jednoducho určiť rozmery obrázku, do ktorého budeme strom vykresľovať. Výšku obrázku určíme podľa úrovne najhlbšieho listu stromu a šírku podľa hodnoty v premennej EndX po skončení výpočtu súradníc.

Pre vykreslenie stromu do obrázku sa potom zavolá rekurzívna funkcia, ktorá vykreslí každý uzol stromu podľa pozície jeho stredu. Uzly reprezentujúce operácie najprv vykreslia hrany stromu, ktoré z nich vychádzajú, potom podstromy svojich potomkov a potom sami seba.

5.7 Chyby

Program bol testovaný pomocou zadávania rozličných súborov vstupných relácií a tvorením dotazov nad týmito reláciami. Pri testovaní boli použité rozličné príklady z

cvičení k predmetu Databázové systémy na MFF UK ako aj príklady z rôznych učebníc a boli porovnávané výsledky jednotlivých dotazov.

Správnosť animácie bola testovaná len vizuálne, keďže o podobnom programe ktorý by vizualizoval vyhodnocovanie operácií relačnej algebry na konkrétnych dátach neviem. Vytváranie animácie operácií vychádza z postupu, ktorý je použitý na vyhodnocovanie týchto operácií, čo zaručuje konzistentnosť tabuliek medzivýsledkov a animácie.

Program som sa snažil tiež otestovať na behové chyby. Napriek tomu sa v ňom môžu vyskytnúť. Ak behom programu nastane behová chyba, kvôli ktorej bude musieť byť program ukončený, pokúsi sa program sprostredkovať užívateľovi uloženie aktuálneho stavu projektu do súboru.

6 SPRACOVANIE DOTAZU

6.1 Prevod dotazu do textového formátu

Užívateľovi je umožnená práca s dotazom vo formáte Rich Text Format. Funkcie, ktoré vytvárajú štruktúru dotazu, akceptujú dotaz zapísaný v textovej forme. Parsovanie dotazu je potom realizované jednoduchším spôsobom, keď sa porovnáva len hodnota príslušného znaku a nemusí sa zisťovať akým fontom bol daný znak napísaný.

Pre prevod dotazu z Rich Text formátu do obyčajného (plain) textu som napísal triedu **Text_Rtf**. Pri prevode dotazu do textového formátu sa jednotlivé symboly nahradia odpovedajúcimi tagmi. Pre prevod symbolu na tag som napísal statickú triedu **Tags**, ktorá obsahuje statické funkcie realizujúce tieto prevody. Napríklad symbol \div sa prevedie na reťazec “\division“. Dotaz v textovom formáte je určený dotazom v Rich Text formáte jednoznačne. Opačne to však neplatí. Preto je potrebné si pamätať, ktoré symboly boli v pôvodnom dotaze nahradené.

Inštancia triedy **Text_Rtf** sa vytvorí pred začatím vyhodnocovania dotazu. Uchová si pôvodný dotaz, ktorý zadal užívateľ a vytvorí si usporiadaný zoznam špeciálnych symbolov použitých v dotaze a pozície ich tagov v textovej forme dotazu.

Keďže inštancia triedy dotazu sa vytvára z textovej formy dotazu, sú aj nájdené chyby počas vyhodnocovania viazané s pozíciou v textovej forme dotazu. Trieda **Text_Rtf** umožňuje previesť pozíciu v textovej forme dotazu do pozície v pôvodnej forme dotazu. Postupne sa prejde zoznam použitých symbolov v dotaze a pre všetky symboly, ktorých tag mal menšiu pozíciu ako prevádzaná pozícia, znížime prevádzanú pozíciu o dĺžku tohto tagu zníženú o jedna, pretože musíme uvažovať aj dĺžku symbolu. Tento prevod je lineárny k počtu symbolov použitých v dotaze.

Keďže zoznam symbolov je v triede **Text_Rtf** usporiadaný, možno jednoducho realizovať prevod usporiadaného zoznamu pozícií v textovej forme dotazu do zoznamu pozícií v pôvodnej forme dotazu. Prevod sa realizuje podobne ako pri prevode jednej hodnoty. Postupne sa prechádza zoznamom použitých symbolov a zoznamom pozícií, ktoré chceme previesť a prepočítava sa aktuálna pozícia v zadanom dotaze. Aj tento prevod je lineárny vzhľadom k súčtu dĺžok oboch zoznamov.

6.2 Vnútoraná reprezentácia dotazu

Pre uchovávanie dotazu relačnej algebry som naimplementoval triedu **RAFormula**, ktorá poskytuje funkcie na vytvorenie inštancie tejto triedy z textu dotazu, umožňuje jeho vyhodnotenie na konkrétnych reláciách a tiež vykreslenie jeho vyhodnocovacieho stromu.

Základnou štruktúrou uchovania dotazu je strom, ktorý obsahuje aspoň jeden uzol (v prípade, že dotaz obsahoval len názov relácie) a každý uzol má od nuly po dvoch potomkov. V inštancii triedy **RAFormula** si držíme odkaz na koreň vyhodnocovacieho stromu, prostredníctvom ktorého vykonávame všetky funkcie rekurzívneho charakteru s dotazom.

Trieda reprezentujúca prázdny uzol **RAFormulaNode** obsahuje hlavičky všetkých spoločných funkcií pre uzly stromu. Potomkami tejto triedy sú uzly reprezentujúce

názvy relácií alebo jednotlivé operácie a tieto uzly preťažujú funkcie triedy **RAFormulaNode** podľa svojich potrieb. Medzi základné položky každého uzla patria:

- Príznak určujúci typ uzla, či sa jedná o uzol reprezentujúci názov relácie v dotaze alebo o uzol reprezentujúci operáciu relačnej algebry.
- Výsledná tabuľka poddotazu určeného podstromom tohto uzla.
- Položka popisujúca, či vo vyhodnocovaní poddotazu nedošlo k sémantickým chybám.
- Súradnice stredu tohto uzlu vzhľadom k obrázku vyhodnocovacieho stromu celého dotazu.
- Maximálna x-ová súradnica pre podstrom určený týmto uzlom. Táto položka nám zjednoduší hľadanie uzla, na ktorý klikol užívateľ v obrázku vyhodnocovacieho stromu. Na nájdenie nám bude stačiť nanajvýš jeden prechod od koreňa k niektorému listu, ktorý je jednoznačne určený touto položkou.

Základné funkcie, každého uzla sú:

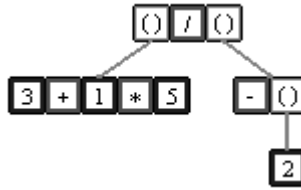
- Rekurzívna funkcia, ktorá vyhodnotí dotaz reprezentovaný podstromom tohto uzla.
- Rekurzívna funkcia, ktorá určí súradnice stredu tohto uzla v obrázku stromu.
- Rekurzívna funkcia, ktorá určí v podstrome tohto uzla uzol, ktorý leží na zadanych súradniciach.
- Funkcia, ktorá vykreslí do obrázku tento uzol. Ak sa jedná o uzol reprezentujúci operáciu, funkcia vykreslí aj hrany smerujúce z tohto uzla.
- Funkcia, ktorá doplní hodnoty do tabuliek reprezentujúcich medzivýsledky pre tento uzol.
- Funkcie, ktoré vytvoria animáciu medzivýsledkov, respektíve vyhodnocovania operácie pre tento uzol.

Vnútoraná reprezentácia podmienky

Súčasťou uzlov reprezentujúcich selekciu alebo podmienené spojenie je inštancia triedy **BoolFormula**, ktorá reprezentuje podmienku prispôbenú pre účely programu. Taktiež je jej hlavným prvkom vyhodnocovací strom logickej formule, v ktorom listy reprezentujú atomické logické formule a nelistové uzly reprezentujú niektorý z logických operátorov. Každý z týchto uzlov má spoločného predka v hierarchii tried **BoolFormulaNode**. Od neho sú postupne odvodené triedy reprezentujúce konkrétne logické operátory alebo atomické logické formule. Atomická logická formula sa skladá z dvoch výrazov a identifikátoru určujúceho porovnávajúci operátor.

Vnútoraná reprezentácia výrazu

Pre reprezentáciu výrazu prispôbenú účelom programu som navrhol triedu **ArithmeticFormula**. Jej základom je vyhodnocovací strom výrazu, v ktorom jednotlivé uzly reprezentujú podvýrazy.



Obrázok 6.1 Príklad reprezentácie výrazu $(3+1*5)/(-2)$

Hoci všetky výrazy použité v atomickej logickej formuly nemusia byť aritmetické, táto štruktúra vyhovuje aj pre uchovanie nearitmetických prvkov. Vzhľadom k tomu, že pri vytváraní tejto štruktúry ešte nemusíme vedieť, aké typy majú použité názvy atribútov.

6.3 Syntaktická analýza dotazu

Syntaktické parsovanie dotazu som implementoval ako upravený konečný automat, ktorý sa snaží z daného reťazca prečítať maximálny úsek, ktorý vyhovuje gramatike. Konštruktor každej z tried RAFormula, BoolFormula a ArithmeticFormula zo vstupného textového reťazca načíta odpovedajúcu časť dotazu, na jej základe vytvorí inštanciu príslušnej triedy a vráti pozíciu v reťazci, kde skončilo čítanie zadanej triedy.

Na začiatku sa začne parsovať dotaz relačnej algebry pomocou funkcie na jej parsovanie zavolanej z konštruktoru triedy RAFormula. Tá podľa potreby volá konštruktor pre vytvorenie inštancie logickej formule triedy BoolFormula, ktorá zase volá konštruktor pre vytvorenie inštancie výrazov pre operandov atomických logických formulí.

6.3.1 Gramatika popisujúca syntax dotazu

Program pracuje s gramatikou pre dotaz v relačnej algebre. Táto gramatika zároveň zohľadňuje priority jednotlivých operátorov relačnej algebry a vyhodnocovanie poddotazov zľava doprava v prípade operátorov s rovnakou prioritou. Neterminály sú zobrazené tučným písmom. Počiatočný neterminál je **RA**. Neterminál **Cond** symbolizuje podmienku pri selekcii alebo pri podmienenom spojení. Gramatika popisujúca syntax dotazu v relačnej algebre vyzerá nasledovne:

RA → **RA2** ∪ **RA** | **RA2** ∩ **RA** | **RA2**

RA2 → **RA3** – **RA2** | **RA3**

RA3 → **RA4** NatJoin **RA3** | **RA4** CondJoin **RA3** | **RA4** ÷ **RA3** | **RA4**

RA4 → **RA5** × **RA4** | **RA5**

RA5 → **RA6** UnOp

RA6 → (**RA**) | NázovRelácie

NatJoin → * | *_L | *_R | *_F | <* | *>

CondJoin → [**Cond**] | [**Cond**]_L | [**Cond**]_R | [**Cond**]_F | < **Cond**] | [**Cond** >

UnOp → (**Cond**) UnOp | [ColList1] UnOp | < ColList2 > UnOp | λ

ColList1 → Atribút | Atribút , ColList1

ColList2 → Atribút → NovéMeno | Atribút → NovéMeno , ColList2

Prvé pravidlá popisujú priority operátorov relačnej algebry. Neterminál **NatJoin** popisuje možné varianty operátorov pre prirodzené spojenie a neterminál **CondJoin** pre podmienené spojenie. Neterminál **UnOp** určuje postupnosť unárnych operácií za sebou, terminál λ (lambda) symbolizuje prázdne slovo. Neterminál **Collist1** určuje zoznam identifikátorov atribútov pre projekciu a **Collist2** popisuje zoznam premenovacích pravidiel pre operáciu premenovania atribútov.

Pre každý neterminál existuje parsovacia funkcia, ktorá sa snaží prečítať z parsovaného reťazca čo najväčšiu časť, ktorá odpovedá gramatike. Funkcie pre neterminály **RA** až **RA6** vracajú koreň vyhodnocovacieho stromu načítaného poddotazu. Takto získané podstromy sa postupne spájajú, až nakoniec vytvoria jeden vyhodnocovací strom pre celý zadaný dotaz. Funkcie pre ostatné neterminály vracajú štruktúry použité v uzloch konkrétnych operácií. Napríklad funkcia pre neterminál **NatJoin** vracia typ prirodzeného spojenia a na základe tejto informácie sa vytvorí inštancia triedy reprezentujúcej operáciu prirodzeného spojenia.

Funkcia, ktorá parsuje neterminál **Cond**, vytvorí novú inštanciu triedy BoolFormula, z aktuálnej pozície v dotaze.

6.3.3 Syntaktická analýza podmienky

Logické operátory majú rôzne priority, preto bol aj na parsovanie podmienky z textu použitý mechanizmus konečného automatu, ktorý zo začiatku vstupného reťazca prečíta maximálny reťazec, ktorý odpovedá použitej gramatike. Neterminály sú zobrazené tučným písmom. Počiatočný neterminál je **BF**. Použitá gramatika má nasledujúci tvar

BF \rightarrow **BF2** \Leftrightarrow **BF** | **BF2**
BF2 \rightarrow **BF3** \Rightarrow **BF2** | **BF3**
BF3 \rightarrow **BF4** \vee **BF3** | **BF4**
BF4 \rightarrow **BF5** \wedge **BF4** | **BF5**
BF5 \rightarrow \neg **BF5** | (**BF**) | **AtomicBF**
AtomicBF \rightarrow **ArithmeticF** **CompareOp** **ArithmeticF**
CompareOp \rightarrow < | <= | = | != | >= | >

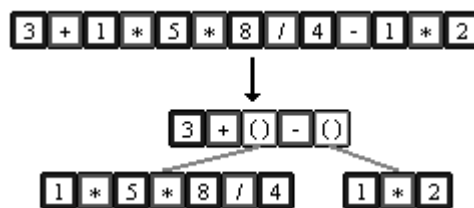
Pre každý neterminál gramatiky, okrem posledného, som napísal parsovaciu funkciu, ktorá zo vstupného reťazca prečíta maximálny reťazec vyhovujúci gramatike. Každá z týchto funkcií vracia koreň vyhodnocovacieho stromu odpovedajúcej podformule. Tieto stromy sú postupne spájané pomocou uzlov reprezentujúcich logické operácie až do vyhodnocovacieho stromu celej logickej formule.

Neterminál **AtomicBF** reprezentuje atomickú logickú formulu. Tá obsahuje dva výrazy a porovnávací operátor. Vo funkcii, ktorá číta atomickú formulu zo zadaného textu, sa vytvárajú inštancie triedy **ArithmeticFormula**, ktoré reprezentujú porovnávané výrazy. Pre ich načítanie sa využíva konštruktor triedy **ArithmeticFormula**, ktorý z danej pozície v dotaze prečíta maximálny vyhovujúci výraz a vráti pozíciu prvého znaku za prečítaným výrazom.

Výrazy sú načítané do štruktúry, ktorá odpovedá zapísanej forme výrazu a nezohľadňuje rôzne priority jednotlivých aritmetických operátorov. Rekonštrukciu tejto štruktúry do podoby, v ktorej sú zohľadnené priority operátorov zaisťuje

funkcia `PrecedenceUpgrade()`, ktorú volá konštruktor triedy výrazu hneď po úspešnom prečítaní výrazu zo zadaného textu. Táto funkcia pracuje so statickou triedou **OperatorPrecedences**, ktorá popisuje priority jednotlivých binárnych operátorov.

Algoritmus mení len uzly, ktoré reprezentujú podvýraz. Ak sa jedná o podvýraz obsahujúci len jeden operand alebo podvýraz obsahujúci unárnu operáciu a výraz, ponechá algoritmus uzol v pôvodnom stave. Ináč určí počet rôznych priorít operátorov v uzle a postupne pre všetky priority od najvyššej, presunie maximálne postupnosti s operátormi danej priority o úroveň nižšie. Aplikuje sa podobný princíp, ako keby sme výrazy uzátvorkovali kôli zdôrazneniu priorít operátorov.



Obrázok 6.2 Príklad prevodu uzlu reprezentujúceho podvýraz

6.3.2 Syntaktické chyby

V prípade, že počas parsovania dotazu bola identifikovaná nejaká syntaktická chyba, je vyvolaná výnimka, ktorá obsahuje príslušnú chybovú hlášku a pozíciu chyby v dotaze. Táto výnimka je zachytená vo funkcii, ktorá zaisťuje spracovanie dotazu. Spracovanie dotazu bude ukončené a dotaz bude nastavený na tzv. prázdny alebo nedefinovaný.

V prípade, ak sa zo začiatku parsovaného reťazca podarí vytvoriť korektný vyhodnocovací strom dotazu, ale spracovaná časť reťazca je kratšia ako reťazec zadaný užívateľom, bude to vyhodnotené ako syntaktická chyba očakávaného konca dotazu. Dotaz bude nastavený na prázdny.

Keďže text dotazu nieje dlhý, nepovažoval som za nutné implementovať zotavovanie sa zo syntaktických chýb. Preto program upozorní len na prvú zistenú syntaktickú chybu pri parsovaní dotazu.

6.4 Sémantická analýza dotazu

Sémantická analýza dotazu je realizovaná vo funkcii, ktorá vyhodnocuje dotaz, ktorý bol korektne syntakticky spracovaný. Vyhodnotenie podstromu uzlu je realizované funkciou `Evaluate`, ktorá pracuje so zoznamom tabuliek prvotných relácií (uchovaných v inštancii triedy `DataSet`) a je preťažená pre každý typ uzlu. Pre správnu detekciu sémantických chýb je potrebné v každej operácii poznať relácie jej operandov a vzhľadom k nim otestovať napríklad správnosť zadaných mien atribútov v selekčnej alebo spájacej podmienke.

Pre uzol reprezentujúci názov relácie, sa skontroluje či existuje tabuľka v danom `DataSet-e` s rovnakým názvom ako bol zadaný v dotaze. Ak existuje, tak sa z tejto tabuľky odstráni duplicitné riadky, aby bola splnená podmienka, že každý medzivýsledok je relácia, teda množina. Ak daný `DataSet` neobsahuje tabuľku so zadaným názvom, nastaví sa príznak indikujúci, že v tomto podstrome sa vyskytla sémantická chyba.

Pre každý uzol reprezentujúci operáciu sa najprv vyhodnotia podstromy reprezentujúce operandy operácie. Ak boli tieto podstromy vyhodnotené korektne, začne sa vyhodnocovať operácia, ktorú uzol reprezentuje. V prípade, že sa v niektorom podstrome vyskytne nejaká sémantická chyba, operácia sa nevyhodnocuje, len sa nastaví príznak, že došlo k sémantickej chybe v niektorom podstrome.

6.4.1 Sémantická analýza podmienky

Prvotnú sémantickú analýzu podmienky realizuje funkcia `TableUpgrade` triedy **BoolFormula**, ktorá prína ako parameter jednu alebo dve tabuľky, vzhľadom ku ktorým bude sémantická analýza prevedená. Analýza sa týka hlavne spracovania všetkých atomických formulí v podmienke. Kontrolujú sa len operandy vo výrazoch, ktoré reprezentujú identifikátor atribútu alebo funkciu na atribúte. Skontroluje sa, či zadaný identifikátor skutočne identifikuje nejaký stĺpec v niektorej zo zadaných tabuliek. V prípade, že identifikuje stĺpce v oboch tabuľkách, bude identifikátor automaticky priradený k atribútu prvej tabuľky. V triede, reprezentujúcej identifikátor názvu atribútu, sa potom nastaví príznak určujúci, ku ktorej tabuľke sa vzťahuje (či ku prvej alebo druhej) a tiež sa z identifikátora vyseparuje meno atribútu.

Ak je týmto operandom funkcia na atribúte relácie. Prevedie sa navyše vypočítanie výslednej hodnoty tejto funkcie. Táto hodnota bude stále rovnaká, pretože podmienka bude vyhodnocovaná len pre riadky tých istých tabuliek a nemá zmysel, aby sa táto hodnota vždy znova počítala pri každom testovaní podmienky. Navyše by potom pri každom testovaní bolo nutné pridať celú danú tabuľku, aby sa mohla hodnota tejto funkcie vypočítať. Jedná sa len o funkcie určujúce minimum, maximum, priemer a súčet hodnôt určitého atribútu v relácii. Všetky tieto funkcie sa dajú určiť jedným prechodom cez všetky hodnoty daného atribútu.

Súčasťou sémantickej analýzy podmienky je skontrolovanie, či každá atomická formula v podmienke obsahuje aspoň jeden identifikátor atribútu. Aby bola v súlade s definíciou selekčnej respektíve spájacej podmienky podľa druhej kapitoly.

Následne ak je všetko v poriadku, je možné podmienku otestovať pre konkrétny riadok, respektíve riadky danej tabuľky, respektíve tabuliek. Sémantické chyby týkajúce sa typovej nekompatibility operandov v operáciách sa odhalia pri prvom testovaní splnenia podmienky na konkrétnych riadkoch. V takomto prípade vyhodnocovanie podmienky pre ostatné riadky už nepokračuje.

Nedefinované hodnoty (NULL) sa v atomickej formule môžu vyskytovať len, ak sa jedná o porovnanie identifikátora atribútu na hodnotu NULL. Ak sa pri vyhodnocovaní atomickej formule pokúsime do výrazu dosadiť za identifikátor atribútu hodnotu NULL a potom výslednú hodnotu tohto výrazu porovnať s nejakou inou hodnotou, nebude to vyhodnotené ako sémantická chyba. V takomto prípade sa celá podmienka (nie len dotknutá atomická formula) vyhodnotí na false.

Podobne ak sa pri vyhodnocovaní výrazu pokúsime deliť nulou, bude celá podmienka vyhodnotená na false a nebude to chápané ako sémantická chyba.

6.4.2 Sémantická analýza pri selekcii

Ak bola výsledná relácia poddotazu, reprezentujúceho operand selekcie, korektne určená, vyhodnotí sa sémantická analýza pre selekčnú podmienku vzhľadom k tejto

výslednej relácii. Postup je uvedený v predošlej kapitole. Ak je selekčná podmienka zadaná korektne vzhľadom k relácii operandu, vyhodnotí sa pre každý riadok tabuľky operandu selekčná podmienka a určí sa tabuľka výslednej relácie.

6.4.3 Sémantická analýza pri projekcii

Pri projekcii je nutné skontrolovať, či každý zadaný identifikátor atribútu v projekčnom zozname je skutočne atribútom relácie, ktorá je operandom tejto operácie. Pri identifikovaní stĺpca podľa jeho pozície je potrebné otestovať, či zadaná pozícia skutočne odpovedá nejakému atribútu v relácii.

Tiež je potrebné skontrolovať, či sa v zozname nenachádza viac identifikátorov, ktoré identifikujú rovnaký atribút. Či už podľa názvu alebo pozície.

Pri realizovaní vyhodnotenia sa postupne kopírujú stĺpce tabuľky operandu uvedené v zozname atribútov do výslednej tabuľky aj s hodnotami. Nakoniec sú z výslednej tabuľky odstránené duplicitné riadky.

6.4.4 Sémantická analýza pri premenovaní atribútov

Pre operáciu premenovania atribútov platí nasledovný vzťah

$$R\langle A_1 \rightarrow B_1, \dots, A_N \rightarrow B_N \rangle = R\langle A_1 \rightarrow B_1 \rangle \dots \langle A_N \rightarrow B_N \rangle$$

Sémantická analýza teda prebieha priamo počas vyhodnocovania operácie. Postupne sa pre všetky i od 1 do N otestuje či relácia operandu obsahuje atribút identifikovaný pomocou A_i . Potom sa otestuje či nové meno B_i nemá už niektorý z atribútov relácie s výnimkou premenovávaného atribútu. Ak je toto všetko splnené tento atribút sa v relácii výsledku premenuje a pokračuje sa ďalším premenovávacím pravidlom.

6.4.5 Sémantická analýza pri množinových operáciách

Na operáciu **kartézskeho súčinu** sa nekladú žiadne podmienky a táto operácia je dobre definovaná pre ľubovoľné dve relácie.

Pre operácie **zjednotenie**, **prienik** a **množinový rozdiel** je potrebné otestovať kompatibilitu schém relácií operandov. Teda či majú rovnaký počet atribútov a či i -ty atribút v prvej relácii má rovnaký dátový typ ako i -ty atribút v druhej relácii.

Pre operáciu **delenia** relácií sa musí otestovať či relácie operandov sú kompatibilné pre túto operáciu. Prvá relácia musí mať viac atribútov ako druhá. A dátové typy atribútov druhej relácie musia byť zhodné s dátovými typmi posledných atribútov prvej relácie. Podrobnejší popis kompatibility pre delenie je v odstavci 2.6.

6.4.6 Sémantická analýza pri spojeniach

Pri **prirodzenom spojení** stačí skontrolovať, či rovnako pomenované atribúty v reláciách operandov majú tiež rovnaké dátové typy.

Pri **podmienenom spojení** sa skontroluje sémantická správnosť podmienky vzhľadom k obom reláciám operandov. Sémantická kontrola podmienky je popísaná v odstavci 6.4.1.

Sémantická analýza pri spojeniach je rovnaká bez ohľadu na to, či sa jedná o klasické spojenie, nejakú variantu vonkajšieho spojenia alebo niektoré polospojenie.

7 VIZUALIZÁCIA

Vizualizácia v programe REAL je riešená postupným zobrazovaním medzistavov. Pri vizualizácii medzivýsledkov v podstrome zvoleného uzlu ide o zobrazenie medzivýsledkov v paneli Tables, pre všetky uzly reprezentujúce operácie v tomto podstrome a v poradí v akom boli vyhodnocované pri vyhodnocovaní dotazu. V paneli **Evaluation Tree**, v obrázku vyhodnocovacieho stromu sa vyhodnocovaná operácia zobrazí spolu s uzlami reprezentujúcimi operandy operácie do rámiku.

Pri vizualizácii konkrétnych operácií rámik okolo uzla operácie a jej operandov v obrázku vyhodnocovacieho stromu nemení svoju pozíciu, pretože vizualizovaná operácia sa nemení. Postupne sa zväčšujú riadky, respektíve stĺpce v tabuľkách v paneli Tables a tabuľka výsledku sa v priebehu vizualizácie postupne zapĺňa hodnotami.

Pri analýze a programovaní som sa snažil minimalizovať množstvo informácií, ktoré sú potrebné pre reprezentáciu jednotlivých stavov počas vizualizácie.

7.1 Vnútoraná reprezentácia animácie

Pre ovládanie vizualizácie som navrhol základnú triedu **Animation**, ktorá obsahuje informácie o animácii, prostriedky pre riadenie animácie a aj prostriedky pre jej prehrávanie. Táto trieda tiež vlastní referenciu na triedu hlavného okna programu, v ktorom sa budú jednotlivé stavy postupne zobrazovať. Animácia môže byť v dvoch stavoch, ktoré popisujú, či sa animácia prehráva alebo či je zastavená. Táto trieda reprezentuje obecnú animáciu a triedy pre animáciu zobrazujúcu medzivýsledky (**SubTreeAnimation**), respektíve pre animáciu vyhodnocovania operácie (**OperationAnimation**), ktoré sú jej potomkami, musia prežiť iba funkciu, zobrazujúcu konkrétny snímok animácie (*ShowAnimationShot*). O všetko ostatné sa stará trieda Animation.

Trieda hlavného okna programu obsahuje verejné funkcie, ktoré slúžia na zobrazovanie informácií o aktuálnom stave animácie. Napríklad nastavenie pozície posuvníka reprezentujúceho pozíciu v animácii na aktuálnu hodnotu pozície v animácii, reakcie na zmenu stavu animácie alebo zobrazenie aktuálneho snímku animácie v hlavnom okne.

Zmena pozície v animácii môže byť vyvolaná z hlavného okna užívateľom alebo vnútri triedy Animation, ak je animácia v stave prehrávania animácie. Pri zmene pozície z hlavného okna sa aktualizuje pozícia animácie a zavolá sa metóda animácie, ktorá zobrazí aktuálny snímok. Pri zmene pozície počas prehrávania animácie, sa zavolá metóda triedy hlavného okna zobrazujúca aktuálny snímok a tiež funkcia, ktorá aktualizuje zobrazované informácie v paneli ovládajúcom animáciu.

Pri voľbe štruktúry pre jednotlivé snímky je potrebné brať do úvahy, že užívateľ môže pomocou posuvníka, určujúceho pozíciu v animácii, meniť zobrazovaný snímok nielen o jednu pozíciu niektorým smerom, ale bude sa môcť pohybovať v animácii aj o viac ako o jeden snímok. Preto je potrebné aby sme dokázali zobraziť jednotlivé časti animácie len na základe informácií uložených v triedach snímkov.

7.2 Animácia medzivýsledkov

Na zobrazenie aktuálneho snímku animácie zobrazujúcej medzivýsledky si stačí pamätať uzly operácií v poradí, v akom boli tieto operácie pri vyhodnocovaní dotazu vyhodnocované.

Vytvorenie tohto zoznamu uzlov operácií poskytuje ako funkciu trieda reprezentujúca dotaz relačnej algebry (*GetSelectedItemAnimationList*). Táto funkcia prejde rekurzívne vyhodnocovací strom štýlom post order a postupne pridá všetky uzly operácií podstromu vybraného uzlu do výsledného zoznamu. Nakoniec pridá na začiatok a koniec pôvodne vybraný uzol.

Zobrazenie aktuálneho snímku animácie medzivýsledkov sa vykonáva vo verejnej funkcii triedy hlavného okna (*ShowSubTreeAnimationShot*). Pozostáva z vykreslenia rámika okolo uzla reprezentujúceho operáciu a uzlov reprezentujúcich jej operandy. Ďalej sa zobrazia v paneli Tables tabuľky, ktoré reprezentujú relácie operandov a výslednú reláciu tejto operácie a do panela Node Info sa zobrazia informácie o operácii, ktorú tento uzol reprezentuje. Príklad animácie medzivýsledkov je zobrazený v odstavci 4.6 na obrázku 4.10.

7.3 Animácia vyhodnocovania operácie

Animáciu vyhodnocovania operácie riadi pre všetky druhy operácií trieda **OperationAnimation**, ktorá je podedená od triedy Animation. Táto trieda si uchováva usporiadaný zoznam takzvaných snímkov tabuliek v paneli Tables. Tento zoznam získa animácia priamo metódou vybraného uzla vyhodnocovacieho stromu, ktorá vytvorí tento zoznam snímkov na základe aktuálnych relácií operandov tejto operácie.

Každý snímok animácie obsahuje informáciu o tom, koľko riadkov výslednej tabuľky sa má zobraziť. Aby sa predišlo nastavovaniu viditeľnosti pre každý riadok zakaždým, keď sa zmení pozícia animácie, počet práve zobrazených riadkov je uložený v pomocnej premennej a pri každej zmene pozície v animácii sa zviditeľnia respektíve zneviditeľnia len riadky, ktorých sa to týka.

Pri vytváraní týchto zoznamov snímkov nie je potrebné kontrolovať sémantiku, pretože vyhodnocovací strom sa vykreslí, len ak bol zadaný dotaz korektné vyhodnotený. Užívateľ teda môže vyberať uzly len v prípade korektného vyhodnotenia dotazu. Ak vo vyhodnocovaní dotazu nastali chyby, animácia sa nastaví do nedefinovaného stavu a panel pre jej ovládanie nebude možné používať.

7.3.1 Animácia pri projekcii

Pre zobrazenie aktuálneho stavu animácie pri vizualizácii projekcie potrebujeme poznať index atribútu, ktorý sa v danom snímku pridá do výslednej relácie a počet už pridaných atribútov. Na základe indexu premenovávaného atribútu môžeme v tabuľke, ktorá reprezentuje operand operácie, zvýrazniť tento stĺpec. Podľa aktuálneho počtu atribútov vo výslednej relácii vieme, koľko stĺpcov výslednej tabuľky máme zobraziť. Aktuálne pridávaný atribút bol pridaný ako posledný, teda vieme, ktorý stĺpec potrebujeme zvýrazniť v tabuľke výslednej relácie.

V animácii sa v tabuľke výslednej relácie zobrazujú len hlavičky stĺpcov, ale žiadne dáta. Je to preto, že postupne zobrazované vyplnené stĺpce by mohli vytvoriť v

aktuálnej tabuľke duplicitné riadky. Ale tiež preto, že výsledná tabuľka už neobsahuje duplicitné riadky, ktoré vznikli pri projekcii a teda aktuálne dáta vo vytváranej tabuľke výsledku by nemuseli odpovedať dátam v tabuľke operandu. Príklad animácie projekcie je uvedený na obrázku 4.12.

7.3.2 Animácia pri premenovávaní

Zoznam snímkov pre vizualizáciu premenovávania obsahuje dva druhy snímkov. Prvý sa používa len ako prvý snímok v zozname snímkov. Obsahuje zoznam všetkých premenovacích pravidiel použitých v tomto premenovaní. Premenovacie pravidlo je v tomto prípade určené indexom premenovávaného atribútu a novým menom pre tento atribút. Ostatné snímky v zozname snímkov obsahujú referenciu na prvý snímok a tiež informáciu o tom, koľké premenovacie pravidlo im odpovedá.

Pri zobrazovaní *i*-teho snímku v animácii premenovávania sa na tabuľku výslednej relácie postupne aplikuje prvých *i* premenovacích pravidiel a na tabuľku operandu o jedno pravidlo menej. Potom sa v oboch tabuľkách zvýrazní stĺpec, ktorý bol premenovaný ako posledný. Príklad animácie pre premenovanie atribútov je uvedený na obrázku 4.13.

7.3.3 Animácia pri ostatných operáciách

Vizualizácia ostatných operácií relačnej algebry pozostáva z postupného zvýrazňovania riadkov v tabuľkách zobrazených v paneli Tables. Pre popis zvýraznenia jednej tabuľky som napísal triedu **DataGridShot** respektíve triedu od nej odvodenú **DataGridShot_**. Tieto triedy obsahujú informáciu o tom koľký riadok sa má zvýrazniť a tiež aká farba má byť použitá.

Podľa toho či je operácia unárna alebo binárna, obsahuje trieda reprezentujúca snímok v animácii tejto operácie dve alebo tri položky triedy **DataGridShot**, ktoré popisujú zvýraznenie riadkov v tabuľkách operandov a výsledku. Snímok taktiež obsahuje informáciu o počte prvkov výslednej relácie v danej fáze vyhodnocovania. Príklad animácie selekcie je uvedený na obrázku 4.14, animácia prieniku je zobrazená na obrázku 4.15 a prvá časť animácie pri delení relácií je uvedená na obrázku 4.16.

8 ZÁVER

Cieľom tejto bakalárskej práce bolo vytvoriť výukovú aplikáciu, ktorá by mala objasniť vyhodnocovanie databázových dotazov zapísaných v relačnej algebre. Počas práce som vytvoril program REAL, ktorý umožňuje vytváranie a upravovanie tabuliek, zadávanie dotazov relačnej algebry a ich následné vyhodnotenie na zadaných tabuľkách. Snažil som sa, aby upravovanie tabuliek aj dotazov bolo čo najprirodzenejšie z pohľadu užívateľa.

Pre lepšie pochopenie vyhodnocovania jednotlivých operácií, umožňuje program ich vizualizáciu pre ľubovoľné dáta. Užívateľ môže sledovať tabuľky medzivýsledkov vo vyhodnocovaní dotazu alebo môže sledovať priebeh vyhodnocovania jednotlivých operácií dotazu na ich aktuálnych operandoch.

POUŽITÉ ZDROJE

[1] Pokorný J., Halaška I.: *Databázové systémy*, Vydavatelství ČVUT, 2003.

[2] Ramakrishnan R., Gehrke J.: *Database management systems*, McGraw-Hill, 2003.

[3] Skopal T.: *Prezentácie k prednáške Databázové systémy*,
<http://urtax.ms.mff.cuni.cz/~skopal/DBI025.htm>

[4] MagicLibrary.dll, <http://www.dotnetmagic.com/downloads/MagicInstall174.msi>