Das Folhas de Cálculo às Bases de Dados Relacionais

Jácome Cunha João Saraiva Joost Visser

DI – UM

DI - 08 de Outubro de 2008

Agenda

Agenda

Motivação

- As folhas de cálculo são a LP mais usada no mundo
- São muitas vezes usadas como BD
- Demasiada redundância, erros, etc.

	A	B	C	D	E	F	G	H	I	J	K
1											
2	clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	totalDays	rentPerDay	total rent	ownerNo	oName
3	cr76	pg4	john	6 Lawrence St.	1/7/00	8/31/01	602.00	50.00	30100.00	co40	tina
4	cr76	pg16	john	5 Novar Dr.	9/1/01	9/1/02	365.00	70.00	25550.00	co93	tony
5	cr56	pg4	aline	6 Lawrence St.	9/2/99	6/10/00	282.00	50.00	14100.00	co40	tina
6	cr56	pg36	aline	2 Manor Rd	10/10/00	12/1/01	417.00	60.00	25020.00	co93	tony
7	cr56	pg16	aline	5 Novar Dr.	11/1/02	8/10/03	282.00	70.00	19740.00	co93	tony

Objectivo

- BD deveriam ser expressas no respectivo paradigma
- Eliminação de redundância através de normalização
- Devido às restrições, obriga a cometer menos erros

	A	B	C	D	E	F	G	H	I
1	clientNo	cName		cName	rentStart	rentFinish	rentPerDa	total rent	totalDays
2	cr76	john		john	7/1/00	8/31/01	50	21300	426
3	cr56	aline		john	9/1/01	9/1/02	70	25550	365
4				aline	9/1/99	6/10/00	50	14150	283
5				aline	10/10/00	12/1/01	60	25020	417
6				aline	11/1/03	8/10/05	70	45360	648
7									
8	propertyNo	pAddress	rentPerDay	oName		ownerNo	oName		
9	pg4	6 Lawrence St.	50	tina		co40	tina		
10	pg16	5 Novar Dr.	70	tony		co93	tony		
11	pg36	2 Manor Rd	60	tony					

Agenda

Bases de dados

Chave é um conjunto de atributos:

- Chave Candidata: que pode ser usado como chave primária
- Chave Primária: que define unicamente uma linha
- Chave Estrangeira: de uma tabela que referencia a chave prim. de outra

Formas normais:

- 1FN: não podem existir conjuntos
- 2FN: não podem existir dependências parciais na CP
- 3FN: não podem existir dependências transitivas na CP

- Dada uma tabela de dados, calcula as DFs escondidas nos dados
- Depende fortemente da "qualidade" dos dados

Dependências Funcionais

$ownerNo \rightharpoonup oName$

$propertyNo \rightharpoonup pAddress, rentPerDay, ownerNo, oName$

...

	A	B	C	D	E	F	G	H	I	J	K
1											
2	clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	totalDays	rentPerDay	total rent	ownerNo	oName
3	cr76	pg4	john	6 Lawrence St.	1/7/00	8/31/01	602.00	50.00	30100.00	co40	tina
4	cr76	pg16	john	5 Novar Dr.	9/1/01	9/1/02	365.00	70.00	25550.00	co93	tony
5	cr56	pg4	aline	6 Lawrence St.	9/2/99	6/10/00	282.00	50.00	14100.00	co40	tina
6	cr56	pg36	aline	2 Manor Rd	10/10/00	12/1/01	417.00	60.00	25020.00	co93	tony
7	cr56	pg16	aline	5 Novar Dr.	11/1/02	8/10/03	282.00	70.00	19740.00	co93	tony

Problemas

Alguns problemas

- São geradas demasiadas DFs
- Algumas DFs não podem ser usadas (chaves e fórmulas)
- O que fazer com as fórmulas?

Solução

- Algoritmo do Maier para calcular uma BD na 3FN
- Para garantir a propriedade "lossless" é necessário adicionar uma DF:

$$R \setminus Forms. \rightharpoonup !XPTO!, \text{ com } !XPTO! \notin R$$

- Cada fórmula $X_0 = f(X_1, ..., X_n)$ induz a DF $X_1, ..., X_n \rightharpoonup X_0$
- Alguns cuidados: numa coluna sempre a mesma fórmula
- Estas DFs são inseridas no conjunto das já existentes

- Não gera chaves primárias, mas sim chaves candidatas
- Escolhemos a chave com menor número de atributos

Alguns comentários ao esquema gerado

A base de dados

$clientNo \rightharpoonup cName$
$ownerNo \rightharpoonup oName$
$cName, rentStart, rentFinish, rentPerDay \rightharpoonup !XPTO!, total\ rent, totalDays$
$propertyNo \rightharpoonup pAddress, rentPerDay, oName$

- O atributo $!XPTO!$ pode agora ser removido
- Na última tabela, *oName* deveria ser *ownerNo*, mas não há como distinguí-los
- A escolha da chave primária tem de ser melhorada tendo em conta este tipo de questões

Mais alguns comentários ao esquema gerado

- Toda a redundância é eliminada
- Existe agora uma estrutura lógica para os dados
- Mais difícil cometer erros

	A	B	C	D	E	F	G	H	I
1	clientNo	cName		cName	rentStart	rentFinish	rentPerDa	total rent	totalDays
2	cr76	john		john	7/1/00	8/31/01	50	21300	426
3	cr56	aline		john	9/1/01	9/1/02	70	25550	365
4				aline	9/1/99	6/10/00	50	14150	283
5				aline	10/10/00	12/1/01	60	25020	417
6				aline	11/1/03	8/10/05	70	45360	648
7									
8	propertyNo	pAddress	rentPerDay	oName		ownerNo	oName		
9	pg4	6 Lawrence St.	50	tina		co40	tina		
10	pg16	5 Novar Dr.	70	tony		co93	tony		
11	pg36	2 Manor Rd	60	tony					

Agenda

Two-Level Tranformations I

Refinamento de dados

- $A_\phi \underset{from}{\overset{to}{\leqslant}} B_\psi \wedge (\phi \cdot from)$
- $A \underset{from}{\overset{to}{\leqslant}} B$ e $B \underset{from'}{\overset{to'}{\leqslant}} C$ então $A \underset{from \cdot from'}{\overset{to' \cdot to}{\leqslant}} C$

Two-Level Tranformations II

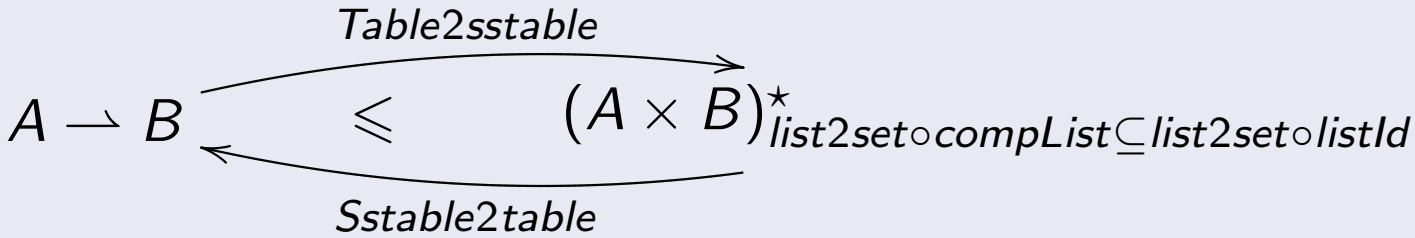
- Usada uma representação de tipos:

$$
\begin{aligned}
&\textbf{data } \mathit{Type} \textbf{ where}\\
&\quad \mathit{String} :: \mathit{Type}\ \mathit{String}\\
&\quad [\circ] \quad :: \mathit{Type}\ a \rightarrow \mathit{Type}\ [a]\\
&\quad \ldots\\
&\textbf{type } \mathit{Rule} = \forall\ a\ .\ \mathit{Type}\ a \rightarrow \mathit{Maybe}\ (\mathit{View}\ (\mathit{Type}\ a))\\
&\textbf{data } \mathit{View}\ a \textbf{ where } \mathit{View} :: \mathit{Rep}\ a\ b \rightarrow \mathit{Type}\ b \rightarrow \mathit{View}\ (\mathit{Type}\ a)\\
&\textbf{data } \mathit{Rep}\ a\ b = \mathit{Rep}\{\, \mathit{to} = \mathit{PF}\ (a \rightarrow b), \mathit{from} = \mathit{PF}\ (b \rightarrow a)\,\}
\end{aligned}
$$

- PF é um tipo para representar funções em PF
- Combinadores como $\mathit{once} :: \mathit{Rule} \rightarrow \mathit{Rule}$ ou $\triangleright\ \mathit{Rule} \rightarrow \mathit{Rule} \rightarrow \mathit{Rule}$ completam os sistema

Refinar uma table (relacional) numa tabela (folha cálculo)

Graficamente



Refinar uma Tabela (Relacional) numa Tabela (Folha Cál.)

Em Haskell

$$
\begin{aligned}
&table2sstable :: Rule \\
&table2sstable\ (a \rightharpoonup b)_i = return\ View\ rep\ [a \times b]_{inv'} \\
&\quad \textbf{where}\ inv' = trySimplify\ (i \circ Sstable2table \wedge fd) \\
&\qquad\quad rep\ = Rep\{\, to = Table2sstable, from = Sstable2table \,\}
\end{aligned}
$$

Um exemplo

$$
\begin{aligned}
&*\ ghci\rangle table2sstable\ (clientNo \rightharpoonup cName) \\
&Just\ (View\ (Rep\langle to\rangle\langle from\rangle)\ [clientNo \times cName]_{fd})
\end{aligned}
$$

Ref. de Tabelas com Chaves Estrang. nas Chaves Prim. I

Graficamente

$$((A \rightharpoonup B) \times (C \rightharpoonup D))_{inv1} \underset{Sstables2tables}{\overset{Tables2sstable}{\leqslant}} ((A \times B)^{\star}_{fd} \times (C \times D)^{\star}_{fd})_{inv2}$$

$$inv1 = \pi_A \circ \delta \circ \pi_1 \subseteq \pi_C \circ \delta \circ \pi_2$$

$$inv2 = \pi_A \circ list2set \circ \pi_1^{\star} \circ \pi_1 \subseteq \pi_C \circ list2set \circ \pi_1^{\star} \circ \pi_2$$

Ref. de Tabelas com Chaves Estrang. nas Chaves Prim. II

Graficamente

$$((A \rightharpoonup B) \times (A \rightharpoonup C))_{\delta \circ \pi_1 \subseteq \delta \circ \pi_2} \underset{Table2tables}{\overset{Tables2table}{\cong}} A \rightharpoonup (C \times B?)$$

Ref. de Tabelas com Chaves Estrang. em Atrib. não Chave

Graficamente

$$((A \rightharpoonup B) \times (C \rightharpoonup D))_{inv1} \underset{Sstables2tables'}{\overset{Tables2sstables'}{\leqslant}} ((A \times B)^{\star}_{fd} \times (C \times D)^{\star}_{fd})_{inv2}$$

$$inv1 = \pi_B \circ \rho \circ \pi_1 \subseteq \pi_C \circ \delta \circ \pi_2$$

$$inv2 = \pi_B \circ list2set \circ \pi_2^{\star} \circ \pi_1 \subseteq \pi_C \circ list2set \circ \pi_1^{\star} \circ \pi_2$$

Refinamento de Dados como um Sistema de Re-escrita I

- Transforma uma BD numa folha de cálculo
- Devolve funções para migrar os dados entre ambos os modelos

De BD relacionais até Folhas de Cálculo

$$
\begin{aligned}
&\mathit{rdb2ss} :: \mathit{Rule}\\
&\mathit{rdb2ss} = \mathit{simplifyInv} \rhd (\mathit{many}\ (\mathit{aux}\ \mathit{tables2table})) \rhd\\
&\qquad\qquad (\mathit{many}\ ((\mathit{aux}\ \mathit{tables2sstables}) \rhd\\
&\qquad (\mathit{aux}\ \mathit{tables2sstables}'))) \rhd (\mathit{many}\ (\mathit{aux}\ \mathit{table2sstable}))\\
&\quad \textbf{where}\\
&\quad \mathit{aux}\ r = ((\mathit{once}\ r) \rhd \mathit{simplifyInv}) \rhd ((\mathit{many}\ (\mathit{once}\ r)) \rhd \mathit{simplifyInv})
\end{aligned}
$$

Refinamento de Dados como um Sistema de Re-escrita II

- Transforma uma BD numa folha de cálculo com a mesma estrutura

De BD relacionais até Folhas de Cálculo

$rdb2ssDirect :: Rule$
$rdb2ssDirect = many\ (once\ table2sstable)$

Resumindo o processo

1. Calcular as dependências funcionais
2. Gerar o esquema da base de dados
3. Aplicar o refinamento à BD e obter as funções de migração
4. Migrar os dados entre os modelos
5. Exportar Excel, base de dados, etc.

Agenda

Importação

Importação

- Usamos as UMHL para ler uma folha de cálculo (XML)
- As tabelas são inferidas usando um algoritmo espacial
- O processo descrito anteriormente é então possível de usar

Exportação

- Podemos gerar uma nova folha de cálculo
- Ou então uma base de dados (SQL)
- Criação de uma BD em HaskellDB (em progresso)

Ferramentas

- Está disponível uma biblioteca em Haskell com todas as funcionalidades: http://haskell.di.uminho.pt/websvn
- Uma ferramenta em linha de comandos também foi produzida
- Podem usar-se as fontes e facilmente compilá-la usando o Cabal
- Online é possível transformar uma folha de cálculo numa BD: http://haskell.di.uminho.pt/jacome/HaExcel.cgi

Agenda

Divulgação

Divulgação

- Escrevemos o artigo *From Spreadsheets to Relational Databases and Back* (submetido à PADL 2009)
- Convidados a apresentar o trabalho no *GRACE International Meeting on Bidirectional Transformations*, Tokyo, 14-18 de Dezembro de 2008

Projectos para a UCE-15 / Teses de Mestrado

Projectos

- Métricas e Técnicas de Teste para Folhas de Cálculo
- Folhas de Cálculo na Framework HaExcel: Das Folhas de Cálculo ás Bases de Dados Relacionais

Conclusões e Trabalho Futuro

- Criamos um método para cálcular uma BD relacional a partir de uma tabela de dados
- Migramos BDs para folhas de cálculo e voltamos sem perda nem corrupção de dados
- Criamos front-ends para ler folhas de cálculo
- E back-ends para exportação
- Precisamos ainda de melhorias tanto na importação como na exportação
- Por exemplo, não lemos macros ou funções pouco usadas
- Transformação de fórmulas em funções SQL e vice-versa (em produção)

Questões?

Questões?