AG Heterogene Informationssysteme
Prof. Dr.-Ing. Stefan Deßloch
Fachbereich Informatik
Technische Universität Kaiserslautern

Middleware for Heterogeneous and Distributed Information Systems – Exercise Sheet 9

Wednesday, January 7, 2009 – 10:00 to 11:30 – Room 48-379

Schematic Heterogeneity

Schematic Heterogeneity results from modeling similar application concepts using different data model concepts. Consider the relational databases shown in Figure 1 that contain sales data of cloth shops in Indianapolis, Chicago, and Milwaukee[1]. Each city uses a different relational representation of their sales data.

Indy

Sales

Store	Dept	AvgSales
PineSt	Wmn	62500
WestRd	Wmn	75000
AndAve	Wmn	81500
PineSt	Men	50000
AndAve	Men	73500
PineSt	Toddler	41250
WestRd	Toddler	55000
AndAve	Toddler	68500

Chicago

AvgSales

Store	Wmn	Men	Boy	Girl
CedarRd	48500	35000	25500	⊥
CtrSq	55500	50000	32000	52500
WashSt	63500	58500	42250	58500
IllSt	78000	63250	50000	65500

Milw

LincAveSales

Dept	AvgSales	LCode
Girl	45000	1
Boy	55000	2
Men	65000	3
Wmn	80000	4

FreePkSales

Dept	AvgSales	FCode
Girl	35000	A
Men	48500	B
Wmn	55000	C

WashStSales

Dept	AvgSales	WCode
Boy	28500	B38
Men	46500	C18
Wmn	60000	X27

Figure 1: Average sales for clothes shops

1. What differences do you see between the sales data representations depicted in Figure 1? Why do we speak of schematic heterogeneity here? What part of the sales data is represented on the data level? What part is represented on the metadata level?
2. Give SQL view definitions to translate sales data between the following representations.
    a. Indianapolis to Milwaukee

[1] Catharine M. Wyss, Edward L. Robertson: Relational languages for metadata integration. ACM Trans. Database Syst. (TODS) 30(2):624-660 (2005)

b. Chicago to Indianapolis

c. Milwaukee to Indianapolis

3. What problems do you see with these SQL view definitions? (Consider that shops may open up new departments or close existing ones.)

Federated Interoperable Relational Algebra (FIRA)

The Federated Interoperable Relational Algebra (FIRA) is an extension of the Relational Algebra (RA) for metadata integration. FIRA allows querying and restructuring metadata along with data directly within the relational model. FIRA has an SQL-like counterpart called Federated Interoperable Structured Query Language (FISQL). FIRA can help to overcome data integration problems caused by schematic heterogeneity.

1. FIRA augments the Relational Algebra with six new operators, namely drop projection, down, attribute dereference, generalized union, transpose, and partition. Explain the purpose of each of these operators!
2. Give FIRA expressions to translate sales data between the following representations.
    a. Indianapolis to Milwaukee
    b. Chicago to Indianapolis
    c. Milwaukee to Indianapolis
3. Comment on the following statement: “FISQL/FIRA queries are more robust under schema evolution than traditional SQL/RA queries.”