AG Heterogene Informationssysteme

Prof . Dr.-Ing. Stefan Deßloch

Fachbereich Informatik

Technische Universität Kaiserslautern

Recent Developments for Data Models – Exercise 2

Monday, May 21, 2012 – 15:30 to 17:00 – Room 36-336

1) User-defined Ordering

Reconsider the typed table hierarchy introduced by the first exercise sheet. Say, the following instances appear in the hierarchy tables (some attributes omitted).

Article

pubkey	title	pdate	source_title
4	aaa	2000	bbb
6	aaa	2002	TechReport

Book

pubkey	title	pdate
2	aaa	2000
5	bbb aaa	2000

TechReport

pubkey	title	pdate
1	aaa	2002
3	aaa	2002

a. Assume that the following user-defined ordering is specified.

```
CREATE ORDERING FOR ArticleT EQUALS ONLY BY STATE;
CREATE ORDERING FOR BookT EQUALS ONLY BY STATE;
CREATE ORDERING FOR TechReportT EQUALS ONLY BY STATE;
```

Perform a pairwise comparison of the tuples! Which tuples are considered to be equal?

b. Assume that the following user-defined ordering is specified.

```
CREATE FUNCTION publicationMap(pub PublicationT)
RETURNS VARCHAR(255)
RETURN pub.title || CAST(YEAR(pub.pdate) AS CHAR(4))

CREATE FUNCTION containedPublMap (pub ContainedPublT)
RETURNS VARCHAR(255)
RETURN pub.source_title || ' ' || pub.title ||
CAST(YEAR(pub.pdate) AS CHAR(4));

CREATE FUNCTION techReportMap (pub TechReportT)
RETURNS VARCHAR(255)
RETURN 'TechReport ' || pub.title  ||
CAST(YEAR(pub.pdate) AS CHAR(4));

CREATE ORDERING FOR PublicationT ORDER FULL BY MAP WITH
FUNCTION publicationMap(PublicationT);

CREATE ORDERING FOR ContainedPublT ORDER FULL BY MAP WITH
FUNCTION containedPublMap(ContainedPublT);

CREATE ORDERING FOR TechReportT ORDER FULL BY MAP WITH
FUNCTION techReportMap(TechReportT);
```

Perform a pairwise comparison of the tuples! Which tuples are considered to be equal?

c. Specify an ordering function using the RELATIVE ordering category that mimics the user-defined ordering specified in b) above.

2) Object-relational Views

Reconsider the publication database's schema as introduced by the first exercise sheet. Create views to provide the following information.

a. Create a (non-typed) view to provide access statistics of publications. Include the authors' self-referencing column, the total number of accesses (local and remote) to papers of the respective author, and the average number of accesses (local and remote) per paper!

   1) Specify the required view definition. Choose `authorstats` as view name.

   2) Retrieve the name and access statistics for all authors from the view.

   3) Retrieve the name of the author with the greatest number of accesses (local and remote).

b. Create a (typed) view hierarchy based on the user-defined structured type `PublicationT` and its subtypes (cf. exercise 1) to provide publications that are available electronically. A publication is assumed to be available electronically if its URL attribute is neither null nor empty.
Specify the required view definitions for electronically available publications and books (called `EPublication` and `EBook`, respectively)!

c. Create a restricted (typed) view hierarchy based on the views defined in b) that does not provide access statistics for publications. Create suitable user-defined structured types (`ResPublicationT` and `ResBookT`)! Then specify the required view definitions (`ResPublication` and `ResBook`)!

d. The SQL-Standard does not allow for typed view hierarchies over untyped tables. What is the reason for this restriction?

e. Nevertheless, database vendors developed extensions to address these limitations. Consider the following untyped legacy table.

```
create table legacyPubl (
    publkey integer not null primary key,
    title varchar(150),
    url varchar(150),
    pdate date,
    publisher integer references legacyPublisher(pid),
    first_page integer,
    last_page integer
);
```

This table shall be exposed in a typed view hierarchy (using DB2 object view extensions) based on the user-defined types `ContainedPublT` (as root view), `ArticleT`, and `BookChapterT`. Assume that articles do not have more than 20 pages while book chapters are longer. Specify the required view definitions (`VContainedPubl`, `VArticle`, and `VBookChapter`)!

f. The DB2 database manager cannot always decide whether OIDs are distinct for each view in a hierarchy. Why not? Is it possible to define typed view hierarchies even in such situations?

3) Composite Types and Collection Types

Consider the following table definition.

```
create table publication (
    title varchar(150),
    year char(4),
    author ROW(name varchar(35), firstname varchar(25))
        ARRAY[10],
    keyword varchar(50) MULTISET
);
```

Specify SQL queries to retrieve the following information.

a. All publications (title, year) where *Jim Melton* appears as first author.

b. Triples of publications (title), the authors (name), and the authors' positions.

c. All keywords together with the associated publications (as MULTISET).

d. All publications with *XQuery* as a keyword.

e. All publications (title) with more than two keywords.

f. All publications (title) with duplicate keywords.

g. All distinct keywords used in 2008.

h. All keywords and the number of times they have been used.

i. Keywords used by all publications of *Jim Melton*.

j. Pairs of publications (title) of the same authors.

k. Authors with their publications (as ARRAY) ordered by publication date.

l. Publications (title) that have no other keywords than *Understanding SQL and Java Together*

m. Keywords that have not been used before 2004.

n. All authors and the keywords they used (as MULTISET).

o. Insert a publication named "Understanding SQL and Java Together" by Jim Melton and Andrew Eisenberg published in 2000 with the keywords "SQLJ" and "JDBC".