Meta-SQL: Towards practical meta-querying

Jan Van den Bussche
Stijn Vansummeren
Limburgs Universitair Centrum

Gottfried Vossen
University of Muenster

Context

Enterprise databases contain not only data but also queries:

- view definitions
- stored procedures
- usage logs
- workloads
- software engineering

Stored as long strings…

$\Rightarrow$ Difficult to query in standard SQL

Goal

Allow meta-querying: query the stored queries together with ordinary data

- Which queries do the most joins?
- Which queries return an empty answer?
- What are common subexpressions in this workload?
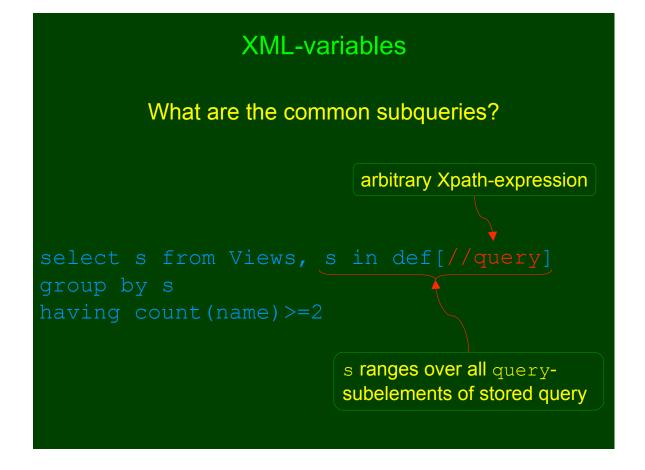- If I do these updates, which materialized views change their value?

Contribution

Meta-SQL: conservative extension of SQL to allow meta-querying

- store queries in XML
- call XSLT within SQL
- add XML-variables
- add an `EVAL` function

Works on top of any SQL99-compliant DBMS

Prototype with DB2 UDB

Call XSLT within SQL

Which queries do the most joins?

```
select name from Views where
count_tables(def) =
(select max(count_tables(def)) from Views)
```

written in XSLT

view definition in XML

XML-variables

What are the common subqueries?

arbitrary Xpath-expression

```
select s from Views, s in def[//query]
group by s
having count(name)>=2
```

s ranges over all query-subelements of stored query

EVAL function

Which queries return an empty answer?

```
select name from Views where not
exists
(select * from EVAL(def))
```

EVAL produces query results in SQL row variable
UEVAL produces them in XML-variable

Implementation

compilation
execution

XSLT functions   Meta-SQL query

SQL statements
(CREATE FUNCTION)

Java wrappers   Rewriting in SQL99

SAXON (XML processing)   Extract (XML variables)
EVAL