Derby/S

A DBMS for Sample-Based Query Answering

Faculty of Computer Science
Database Technology Group

Anja Klein
anja.klein@sap.com

Rainer Gemulla
gemulla@inf.tu-dresden.de

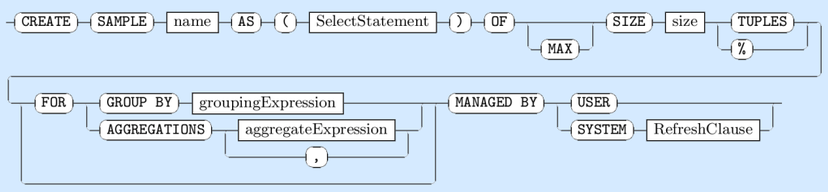
Philipp Rösch
philipp.roesch@tu-dresden.de

Wolfgang Lehner
wolfgang.lehner@tu-dresden.de

Sample Definition

1

- *declarative sample definition language*
  - choice of sampling scheme is left to the system
  - based on information about expected queries and the data set itself
- *parameters*
  - base data (as `SELECT`-statement) & sample size
  - grouping columns, aggregate functions for optimization



Catalog Tables

2

- one logical sample may consist of multiple physical samples (e.g., stratification)
- additional system tables

SYSPHYSICALSAMPLE

PhysicalSampleID	unique ID
LogicalSampleID	reference to logical sample
TableID	reference to table with sample data
Algorithm	sampling scheme
SampleSize	current sample size
Parameters	specific parameters (optional)
Optimization	none/group-by/ aggregation
Attributes	list of aggregates or grouping attributes

SYSLOGICALSAMPLE

LogicalSampleID	unique ID
Name	name of sample
SchemaID	schema of sample
SQL	`CREATE` statement
ResultSetSize	size of base data
SampleSize	desired sample size
SizeType	tuples/fraction
Adaptive	additional memory for optimization (y/n)
Only	fix optimization attributes (y/n)
Manager	system/user
RefreshType	immediate/fast/ complete
RefreshTime	on demand/periodic
PeriodLength	period length
RefreshPeriod	unit of period length

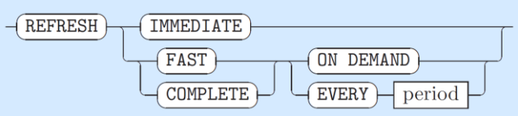
Maintenance

3

- *automatic sample maintenance*
  - samples are kept up-to-date
  - immediate and deferred refresh supported
  - incremental strategies are used whenever possible

	Incremental	Complete
Immediate	`IMMEDIATE`	---
Deferred	`FAST`	`COMPLETE`

- specification (RefreshClause of `CREATE`-Statement):



Maintenance Behind the Curtain

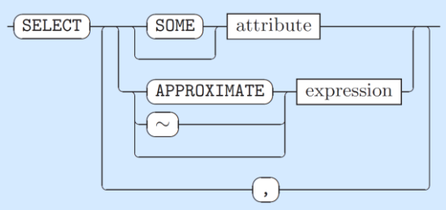
4

- *intercepting DML operations*
  - capture modifications of base data (`INSERT`, `UPDATE`, and `DELETE`)
  - compute deltas of base data
  - derive net effect on the sample
  - depending on the maintenance strategy
    - apply to sample (immediate refresh)
    - write to staging table (deferred refresh)
- `REFRESH` command
  - manually refresh sample

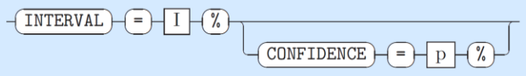
Approximate Queries

5

- *SQL-like queries for approximate query processing*
  - queries against base tables
  - distinction between existential (`SOME`) and statistical (`APPROXIMATE`) errors



- optional specification of accepted error (future work)

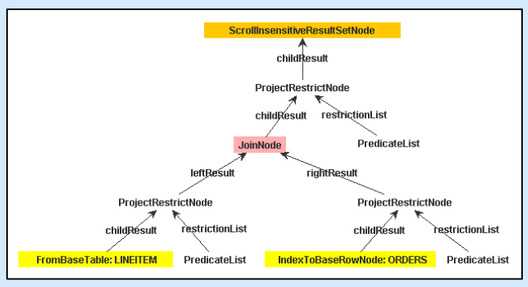


Rewriting the Query Tree

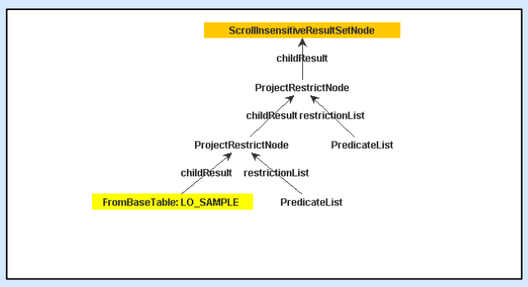
6

- transparent replacement of base tables by appropriate samples
- example: existing sample over lineitem and orders (lo_sample)

```
SELECT * FROM lineitem, orders
WHERE l_orderkey = o_orderkey
```



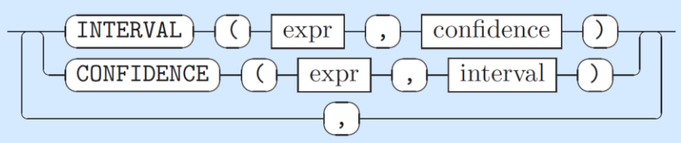

```
SELECT SOME * FROM lineitem, orders
WHERE l_orderkey = o_orderkey
```



INTERVAL and CONFIDENCE

7

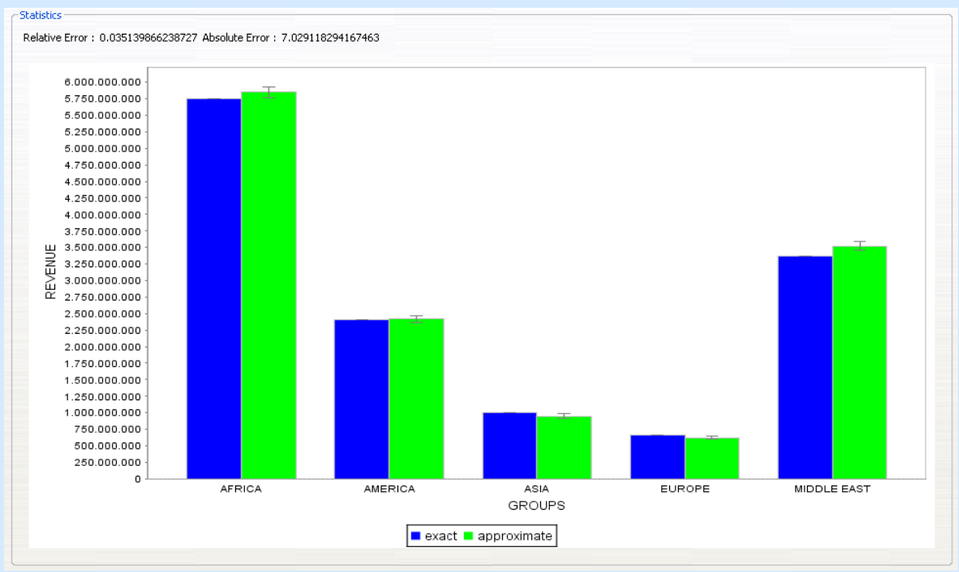
- *computation of error bounds with user-defined parameters*
  - CONFIDENCE returns the confidence for a user-defined interval
  - INTERVAL returns the interval with a user-defined confidence



- implemented as additional aggregation functions
- large-sample confidence intervals

Example

8



http://wwwdb.inf.tu-dresden.de/research/

Derby/S