Year 2000 Data Warehouse Challenges

Big Picture

- Metadata
- Data Extract, Transport & Transformation
- RDBMS services for (OLAP & DM) Analysis
- Resource Management
DW Metadata
Requirements

• Standardization Problem (create islands of mdata)
• Must describe Appl. objects and processes
• Synchronized with RDBMS
• API for mdata management
• Common functionality/objects promoted to SQL
Data Extraction, Transport, Transformation

Requirements

- Unified (everything as a table, errors too)
- Universal (any \{OS, Platform, Source\} -> RDBMS)
- Scaleable & efficient (parallel, wire compression)
- Resumable (E*T errors, resumable errors)
- Full and Incremental
- Extendible (public API for custom drivers)
- Traceable (metadata lineage)
- Single Administration (strong metadata)
RDBMS for OLAP

Integration Questions

• Is it an End User or an Application tool?
• How to share data & results with other tools
• How much integration between RDBMS & tool?
• Should we add OLAP Objects to RDBMS objects?
• How much OLAP calculation on RDBMS?
• Who keeps pre-computed results RDBMS or tool?
RDBMS for OLAP - Integrate Closely

An Integration Model

- Tool provides own abstractions and API for them
- Single point of administration via DW metadata
- Extend SQL for some (most) OLAP calculations
- OLAP aware optimizer
- OLAP access methods
Personality of OLAP queries

*Lessons from OLAP tools*

- OLAP queries:
  - build/submitted incrementally -> many Query Blocks
  - reference the same subcube n-times -> (dynamic MVs)
  - compare different subcubes -> (relate GS)
  - create derived members dynamically - UNION ALL
  - random access to measures within subcube -> pivot
  - join & aggr once and then n-passes over the result
  - order dependent computation (moving avg, rank, etc)
  - rollup along hierarchical dimensions
  - build-in the time dimension with calendar features
  - caching challenging - first query should be cached
Optimizer and OLAP

Have we done everything?

• Optimization with many unmerged query blocks
  – Dynamic MVs
  – Predicate Move around
  – Extend view merging / push join preds into views
  – Push predicates into CUBE/ROLLUP

• Other optimizer enhancements
  – More MV rewrites
  – pruning for extended Grouping Sets
SQL extensions for OLAP
More functional SQL Subcubes

- SQL for subcubes (extend Grouping Sets)
  - efficient calculation of subcubes (Grouping Sets)
  - compare subcubes
  - calculate different aggregates on different slices of subcubes
SQL extensions for OLAP
Order Dependent Calculation

• **Window Functions**
  – defines a (ordered) set per row
  – window per partition over data
  – within it, a window with size value and offset based
  – moving, cumulative, rank aggregates over window
  – extends to general hierarchical calendar

• **New aggregates**
  – statistical aggregates (regression, correlation, ptile)
  – order by time, aggregate measure (e.g., closing balance)
Access Paths & Execution For OLAP

More MVs

• Materialized Views
  – MVs with GROUPING SETS & Concatenated ROLLUP
  – Only some combination of levels materialized -> rewrite directs to nearest materialization
  – dynamic MVs - exploit locality of computation (who creates & destroys, automatic or tool managed)
  – MV advisor becomes critical

• N-dimensional storage

• Make RDBMS more of a number processor
  – offsets to measures
  – efficient arithmetic
  – compile vs. interpreted execution
RDBMS for OLAP

Summary

• Fruitful ground for RDBMS technology
  – SQL extensions needed
  – optimizer work
  – access methods

• Interesting (risky) for tools accessing RDBMS
  – rely on RDBMS more
  – issue complex statements & let RDBMS optimize it

• Metadata
Resource Management

- Database Resource Manager
- Self-tuning Queries (cpu, parallel, memory)
- Monitoring of Long-Running Operations
- Optimizer Statistics Management
- Advisors (index, Materialized Views, etc.)
- Plan Stability
SQL extensions for OLAP analysis

Summary