# Building a Dedicated Analytic Server with


John Sichi

Project Founder

# A Bit About LucidDB

- GPL v2 (w/LGPL for JDBC driver)
- Developed by LucidEra
  - part of its SaaS business intelligence stack
  - used in production apps since 2006
  - 100+ trial/customer deployments
  - built on Eigenbase frameworks
  - design derived from an earlier commercial column store (Broadbase)

# Why More Than One DBMS?

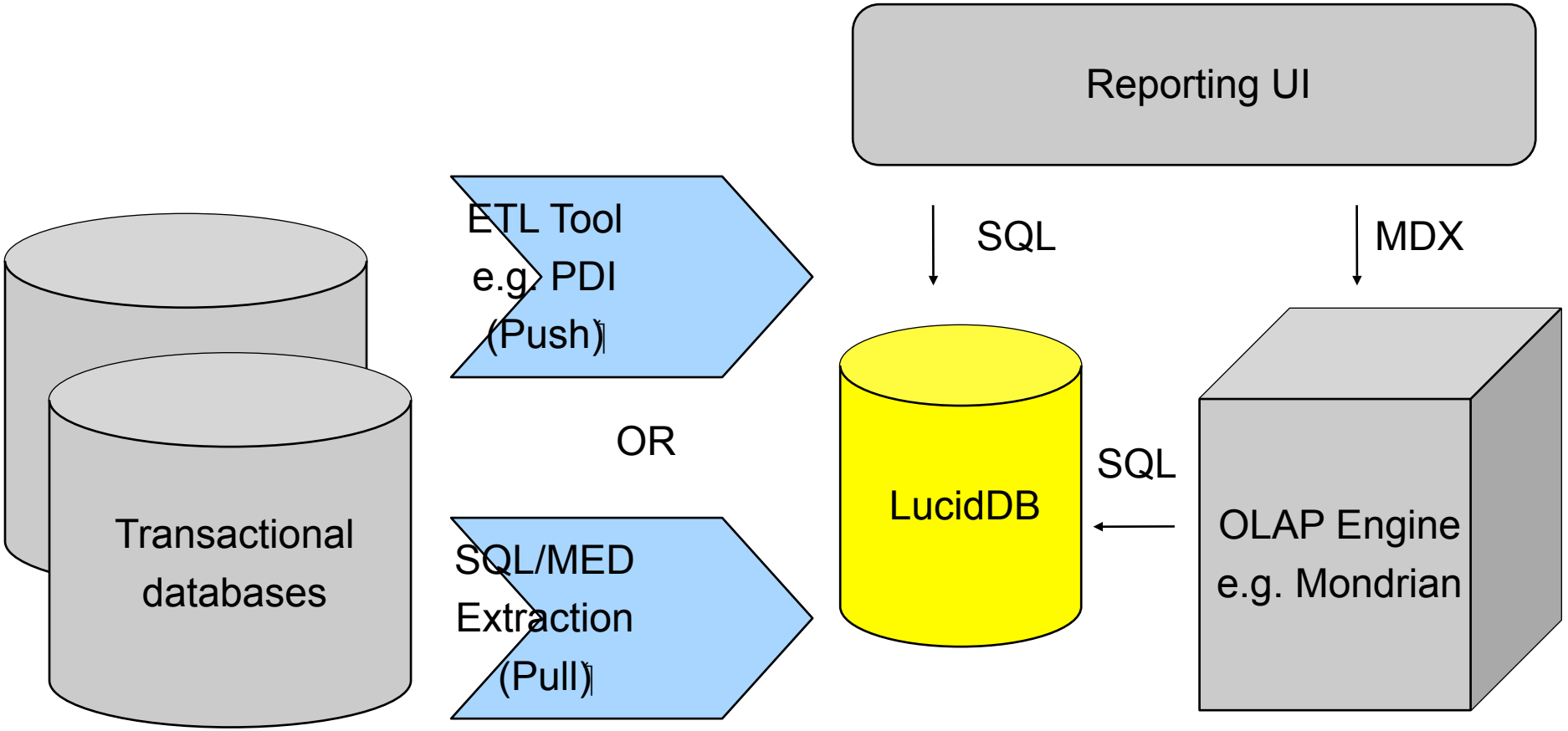

# Because It's Worth The Complexity

- Row store
  - Good for transactions
  - Compression difficult
  - Small scattered writes
  - Row versioning
  - I/O bound
- Column store
  - Good for queries
  - Compression easy
  - Bulk load
  - Page versioning
  - CPU bound

# Data Transformations

- No transformation:
  - run analysis queries directly against OLTP system
- Physical:
  - replicate to read-only slaves; additional indexing
  - load into column store, materialize views
- Logical:
  - denormalize into stars (facts and dimensions)
  - fact archive/summary tables (sales by region for 2005)
  - dimension history tracking (customer single -> married)

# System Dataflow




INSERT INTO warehouse_tbl
SELECT ... FROM mysql_external_table ...

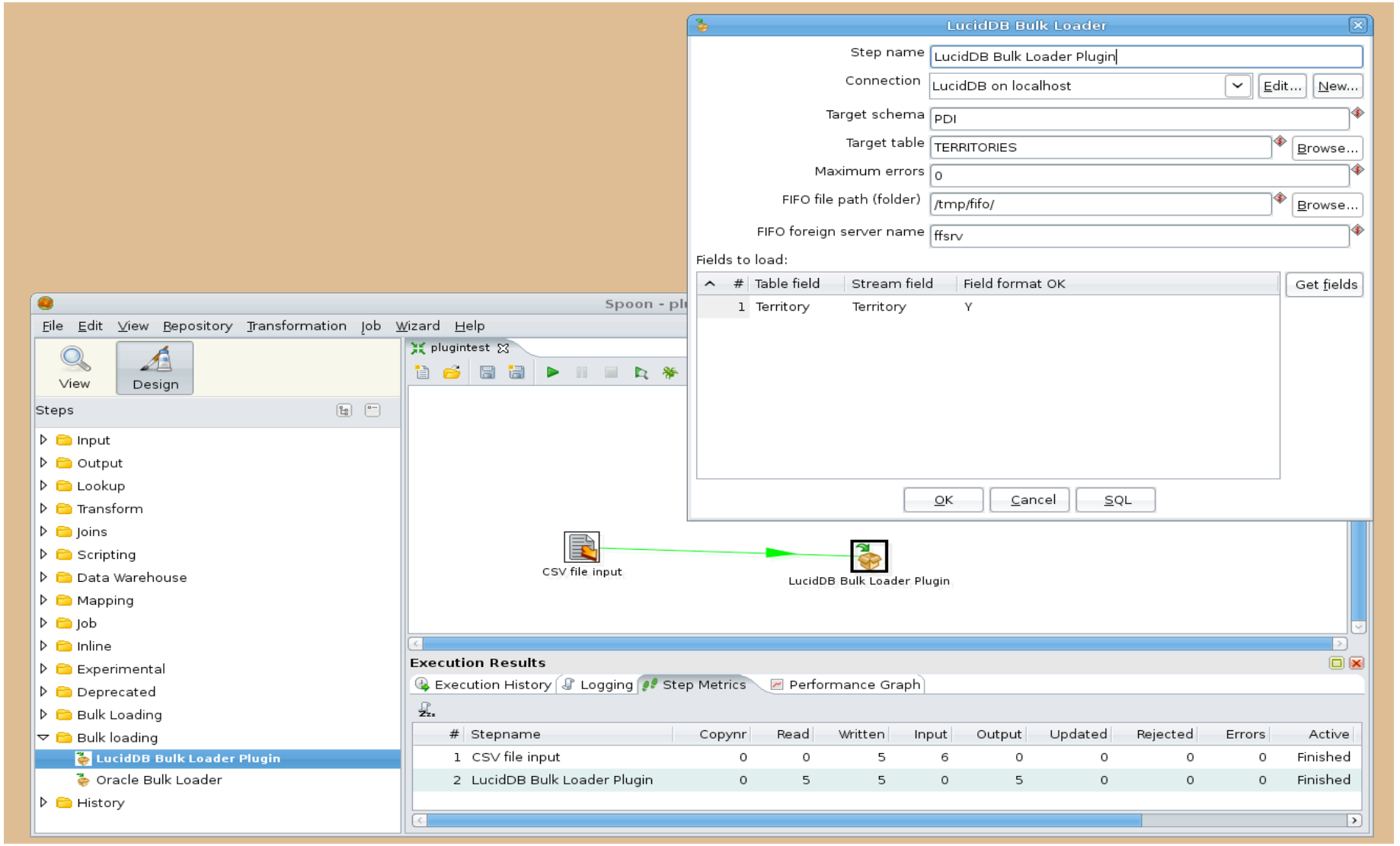
# So, Why Use LucidDB?

- OLAP query acceleration
  - Column store, bitmap indexes, hash join/agg
  - Cost-based star join optimization
  - *Mondrian aggregate table builder*
- Extract/transform/load expressed as SQL
  - SQL/MED data extraction
  - Upsert, user-defined transformations
  - Page multiversioning, hot/incremental backup
  - *Pentaho Data Integration bulk load via fifo pipe*

# Extract/Transform/Load

- With an external ETL tool
  - tool pushes data into fifo pipe; LucidDB pulls from other end via its flatfile reader
- Or cut out the middleman
  - LucidDB can query source system directly via JDBC
  - You can also write your own SQL/MED foreign data wrappers and plug them in

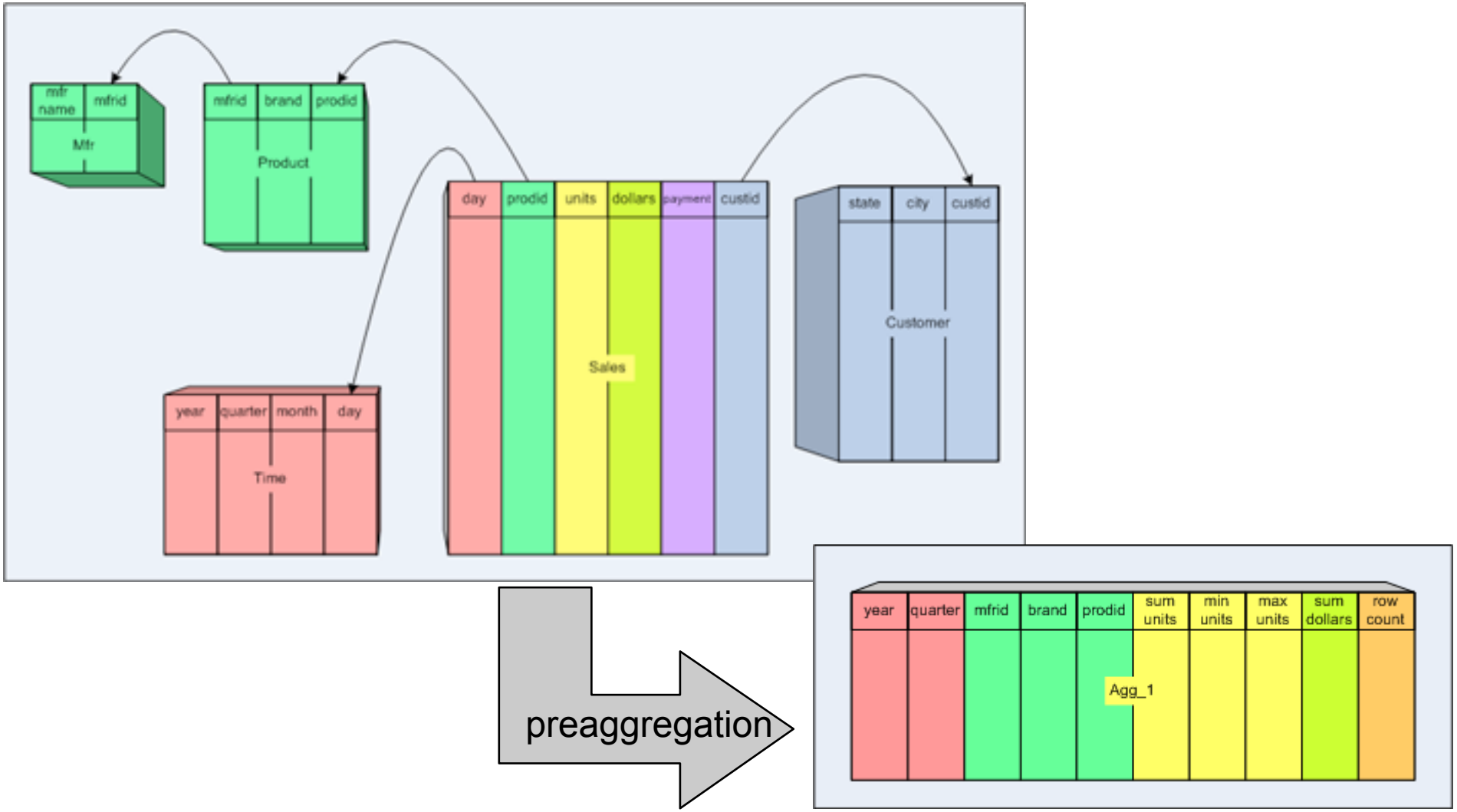
# Pentaho Data Integration



# Applying Transformations

- Example transformations
  - Surrogate key assignment/lookup
  - Cleansing dirty data “Calif -> CA”
  - Deduplication “C.J. Date -> Chris Date”
- How?
  - express directly as SQL (e.g. CASE ... WHEN)
  - plug custom Java transformations into SQL
  - or use ETL tool's transformation library

# Aggregate Tables

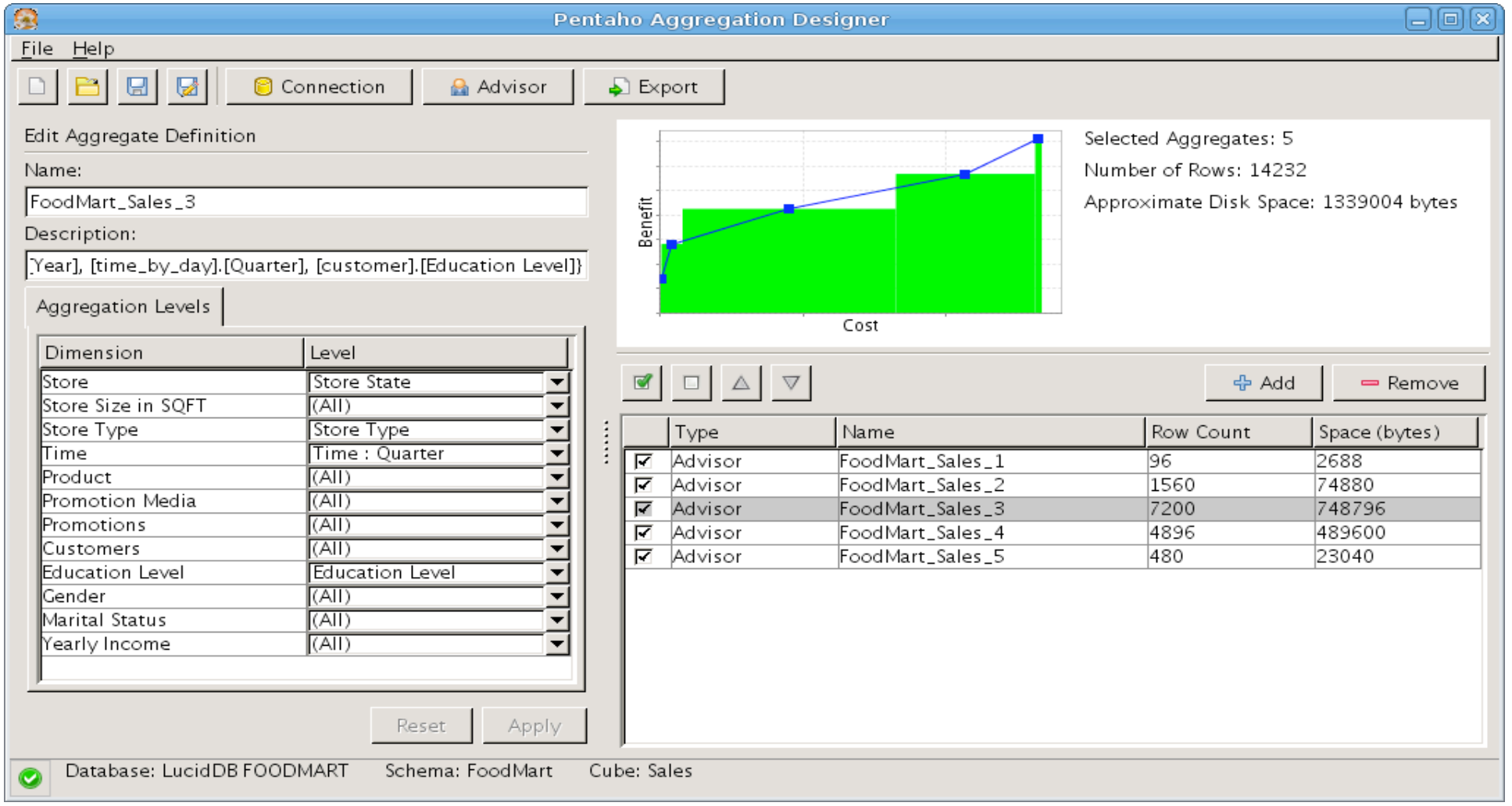


# LucidDB Agg Table Benefits

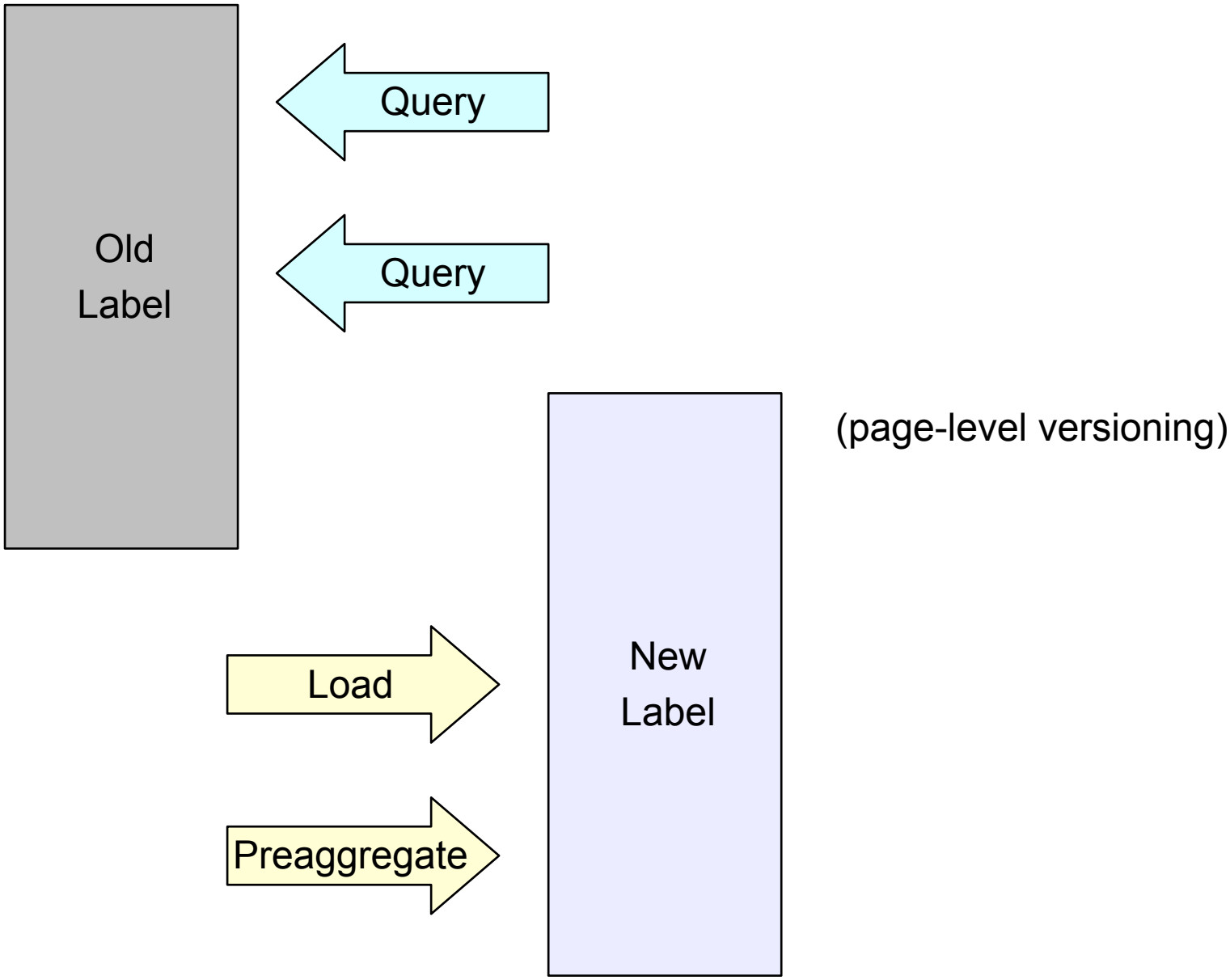
- Column store and bitmap compression work well with repeated values in aggregate compound keys
  - save disk space, I/O
- LucidDB hash aggregation is quite fast
  - reduce load time
- Page versioning efficiently guarantees a consistent query view across all tables

# Mondrian Aggregate Designer

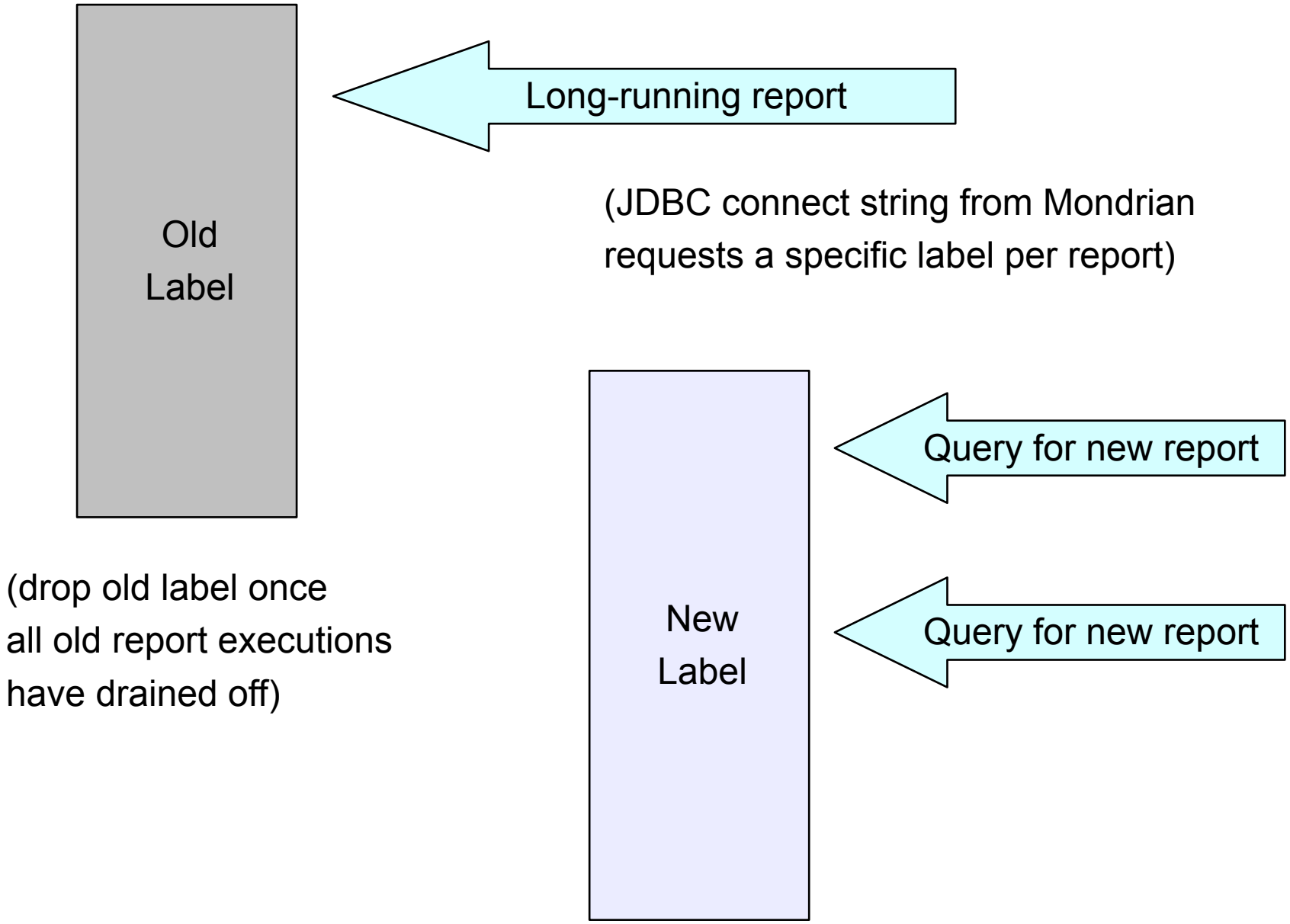
- LucidDB system procedure for automation



# Warehouse Labels



# Rolling the Active Label



# Performance (anecdotal)

- Biggest difference seen vs row store when data does not fit in memory (can be 10X and up)
  - http://pub.eigenbase.org/wiki/LucidDbTp
- When data fits in memory, more like 2X
  - http://www.tholis.net/news/open-source-data-warehousing/
  - http://pentahomusings.blogspot.com/2009/02/lucid-warehouse-results-well-sorta.html
- No rigorous comparisons performed yet; would like to collaborate on DBT-3 benchmarks etc
- SSD helps row store more than column store

# Future Efforts

- Incremental view materialization
- SQL/OLAP support (e.g. fast TOP-N)
- Multicore parallel load/query
  - currently experimental
- Scale-out parallelism
  - multi-node load/query partitioning
- Tool support: generate SQL for ETL

# References

- jsichi@gmail.com
- luciddb-users@lists.sf.net
- http://www.luciddb.org
- http://pub.eigenbase.org/wiki
  - [[LucidDbPdiBulkLoad]]
  - [[LucidDbAggregateDesigner]]

# Attributions

- Images
  - http://www.flickr.com/photos/jekemp/3424782/
  - http://www.flickr.com/photos/lenore-m/409731388/
  - Mondrian aggregate table documentation
- Trademarks
  - LucidDB is a trademark of LucidEra, Inc.
  - Pentaho is a trademark of Pentaho, Inc.