Materialized views

Elena Baralis
Politecnico di Torino
Materialized views

- Precomputed summaries for the fact table
  - explicitly stored in the data warehouse
  - provide a performance increase for aggregate queries

$v_1 = \{\text{product, date, shop}\}$
$v_2 = \{\text{type, date, city}\}$
$v_3 = \{\text{category, month, city}\}$
$v_4 = \{\text{type, month, region}\}$
$v_5 = \{\text{quarter, region}\}$

Materialized views

- Defined by SQL statements
- Example: definition of $v_3$
  - Starting from base tables or views with higher granularity
    
    \[
    \text{group by City, Category, Month}
    \]
  
  - Aggregation (SUM) on \text{Quantity, Income measures}
  
  - Reduction of detail in dimensions
Materialized views

• Materialized views may be exploited for answering several different queries
  – not for all aggregation operators

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006
Materialized view selection

• Huge number of allowed aggregations
  – most attribute combinations are eligible

• Selection of the “best” materialized view set

• Cost function minimization
  – query execution cost
  – view maintainance (update) cost

• Constraints
  – available space
  – time window for update
  – response time
  – data freshness
Materialized view selection

Multidimensional lattice

$\{a,c\}$

$q_1$

$q_2$

$q_3$

$\{a\}$

$\{b\}$

$\{c\}$

$\{d\}$

$\{a,d\}$

$\{b,d\}$

$\{b,c\}$

$\{a,c\}$

$\{}$

$+$

= candidate views, possibly useful to increase workload query performance

Materialized view selection

Materialized view selection

Materialized view selection

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006
ETL Process

Elena Baralis
Politecnico di Torino
Extraction, Transformation and Loading (ETL)

• Prepares data to be loaded into the data warehouse
  – data extraction from (OLTP and external) sources
  – data cleaning
  – data transformation
  – data loading

• Eased by exploiting the staging area

• Performed
  – when the DW is first loaded
  – during periodical DW refresh
Extraction

- Data acquisition from sources
- Extraction methods
  - static: snapshot of operational data
    - performed during the first DW population
  - incremental: selection of updates that took place after last extraction
    - exploited for periodical DW refresh
    - immediate or deferred
- The selection of which data to extract is based on their quality
Extraction

• It depends on how operational data is collected
  – historical: all modifications are stored for a given time in the OLTP system
    • bank transactions, insurance data
    • operationally simple
  – partly historical: only a limited number of states is stored in the OLTP system
    • operationally complex
  – transient: the OLTP system only keeps the current data state
    • example: stock inventory
    • operationally complex
Incremental extraction

• Application assisted
  – data modifications are captured by ad hoc application functions
  – requires changing OLTP applications (or APIs for database access)
  – increases application load
  – hardly avoidable in legacy systems

• Log based
  – log data is accessed by means of appropriate APIs
  – log data format is usually proprietary
  – efficient, no interference with application load
Incremental extraction

- Trigger based
  - triggers capture interesting data modifications
  - does not require changing OLTP applications
  - increases application load

- Timestamp based
  - modified records are marked by the (last) modification timestamp
  - requires modifying the OLTP database schema (and applications)
  - deferred extraction, may lose intermediate states if data is transient
### Comparison of Extraction Techniques

<table>
<thead>
<tr>
<th></th>
<th>Static</th>
<th>Timestamps</th>
<th>Application assisted</th>
<th>Trigger</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management of transient or semi-periodic data</td>
<td>No</td>
<td>Incomplete</td>
<td>Complete</td>
<td>Complete</td>
<td>Complete</td>
</tr>
<tr>
<td>Support to file-based systems</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Rare</td>
</tr>
<tr>
<td>Implementation technique</td>
<td>Tools</td>
<td>Tools or internal developments</td>
<td>Internal developments</td>
<td>Tools</td>
<td>Tools</td>
</tr>
<tr>
<td>Costs of enterprise specific development</td>
<td>None</td>
<td>Medium</td>
<td>High</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Use with legacy systems</td>
<td>Yes</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Yes</td>
</tr>
<tr>
<td>Changes to applications</td>
<td>None</td>
<td>Likely</td>
<td>Likely</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DBMS-dependent procedures</td>
<td>Limited</td>
<td>Limited</td>
<td>Variable</td>
<td>High</td>
<td>Limited</td>
</tr>
<tr>
<td>Impact on operational system performance</td>
<td>None</td>
<td>None</td>
<td>Medium</td>
<td>Medium</td>
<td>None</td>
</tr>
<tr>
<td>Complexity of extraction procedures</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>

From Devlin, Data warehouse: from architecture to implementation, Addison-Wesley, 1997

Copyright – All rights reserved
Incremental extraction

<table>
<thead>
<tr>
<th>Cod</th>
<th>Product</th>
<th>Customer</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Greco di tufo</td>
<td>Malavasi</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>Barolo</td>
<td>Maio</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>Barbera</td>
<td>Lumini</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>Sangiovese</td>
<td>Cappelli</td>
<td>45</td>
</tr>
</tbody>
</table>

4/4/2010

<table>
<thead>
<tr>
<th>Cod</th>
<th>Product</th>
<th>Customer</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Greco di tufo</td>
<td>Malavasi</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>Barolo</td>
<td>Maio</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>Barbera</td>
<td>Lumini</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>Sangiovese</td>
<td>Cappelli</td>
<td>145</td>
</tr>
<tr>
<td>5</td>
<td>Vermentino</td>
<td>Maltoni</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>Trebbiano</td>
<td>Maltoni</td>
<td>150</td>
</tr>
</tbody>
</table>

6/4/2010

Incremental difference

<table>
<thead>
<tr>
<th>Cod</th>
<th>Product</th>
<th>Customer</th>
<th>Qty</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Barbera</td>
<td>Lumini</td>
<td>75</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>Sangiovese</td>
<td>Cappelli</td>
<td>145</td>
<td>U</td>
</tr>
<tr>
<td>5</td>
<td>Vermentino</td>
<td>Maltoni</td>
<td>25</td>
<td>I</td>
</tr>
<tr>
<td>6</td>
<td>Trebbiano</td>
<td>Maltoni</td>
<td>150</td>
<td>I</td>
</tr>
</tbody>
</table>

Data cleaning

• Techniques for improving data quality (correctness and consistency)
  – duplicate data
  – missing data
  – unexpected use of a field
  – impossible or wrong data values
  – inconsistency between logically connected data

• Problems due to
  – data entry errors
  – different field formats
  – evolving business practices
Data cleaning

• Each problem is solved by an ad hoc technique
  – data dictionary
    • appropriate for data entry errors or format errors
    • can be exploited only for data domains with limited cardinality
  – approximate fusion
    • appropriate for detecting duplicates/similar data correlations
      - approximate join
      - purge/merge problem
  – outlier identification, deviations from business rules

• Prevention is the best strategy
  – reliable and rigorous OLTP data entry procedures
Approximate join

- The join operation should be executed based on common fields, not representing the customer identifier.

From Golfarelli, Rizzi, ”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006
Purge/Merge problem

- Duplicate tuples should be identified and removed
- A criterion is needed to evaluate record similarity

Data cleaning and transformation example

Elena Baralis
C.so Duca degli Abruzzi 24
20129 Torino (I)

Normalization

name: Elena
surname: Baralis
address: C.so Duca degli Abruzzi 24
ZIP: 20129
city: Torino
country: Italia

Standardization

name: Elena
surname: Baralis
address: Corso Duca degli Abruzzi 24
ZIP: 20129
city: Torino
country: Italia

Correction

name: Elena
surname: Baralis
address: Corso Duca degli Abruzzi 24
ZIP: 10129
city: Torino
country: Italia

Adapted from Golfarelli, Rizzi, ”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006
Transformation

• Data conversion from operational format to data warehouse format
  – requires data integration

• A uniform operational data representation (reconciled schema) is needed

• Two steps
  – from operational sources to reconciled data in the staging area
    • conversion and normalization
    • matching
    • (possibly) significant data selection
  – from reconciled data to the data warehouse
    • surrogate keys generation
    • aggregation computation
Data warehouse loading

- Update propagation to the data warehouse
- Update order that preserves data integrity
  1. dimensions
  2. fact tables
  3. materialized views and indices
- Limited time window to perform updates
- Transactional properties are needed
  - reliability
  - atomicity
Dimension table loading

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006
Fact table loading

Staging area

Identify updates

ID4
attr 1
attr 2

ID5
attr 3
attr 4

ID6
attr 5
attr 6

New/updated tuples for FT

ID4
ID5
ID6
mes 1
mes 3
mes 5

Map identifiers and surrogate keys

ID4
Sur.Key S4

ID5
Sur.Key S5

ID6
Sur.Key S5

Look-up table

New/updated tuples for FT

Sur key S4
Sur key S5
Sur key S6
mes 1
mes 3
mes 5
mes 6

ODS

Data mart

Fact Table

Sur key S4
Sur key S5
Sur key S6
mes 1
mes 3
mes 5
mes 6

Load new/updated tuples in FT

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006
Materialized view loading

Materialized view loading