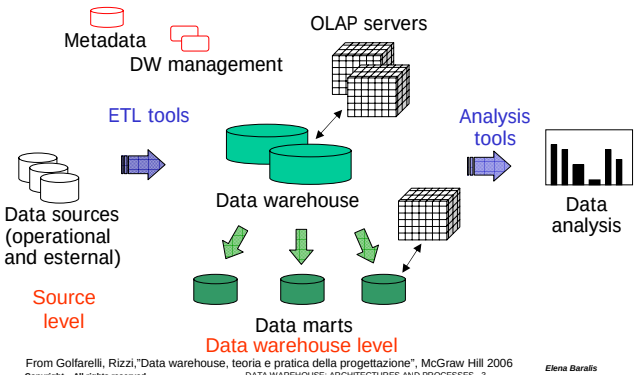
Data warehouse Architectures and processes

Elena Baralis
Politecnico di Torino

Data warehouse architectures

- Separation between transactional computing and data analysis
  - avoid one level architectures
- Architectures characterized by two or more levels
  - separate to a different extent data incoming into the data warehouse from analyzed data
  - more scalable

Two level architecture

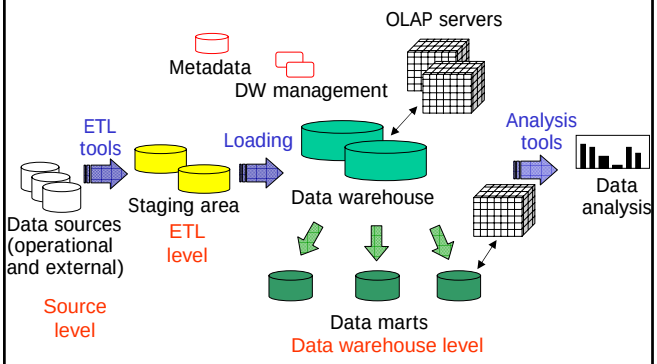


From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Two level architecture features

- Decoupling between source and DW data
  - management of external (not OLTP) data sources (e.g., text files)
  - data modelling suited for OLAP analysis
  - physical design tailored for OLAP load
- Easy management of different temporal granularity of operational and analytical data
- Partitioning between transactional and analytical load
- "On the fly" data transformation and cleaning (ETL)

Three level architecture



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

Three level architecture features

- *Staging area:* buffer area allowing the separation between ET management and data warehouse loading
  - complex transformation and cleaning operations are eased
  - provides an integrated model of business data, still close to OLTP representation
  - sometime denoted as Operational Data Store (ODS)
- Introduces further redundancy
  - more disk space is required for data storage

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

	Static	*Timestamps*	*Applcation assisted*	*Trigger*	*Log*
Management of transient or semi-periodic data	No	Incomplete	Complete	Complete	Complete
Support to file-based systems	Yes	Yes	Yes	No	Rare
Implementation technique	Tools	Tools or internal developments	Internal developments	Tools	Tools
Costs of enterprise specific development	None	Medium	High	None	None
Use with legacy systems	Yes	Difficult	Difficult	Difficult	Yes
Changes to applications	None	Likely	Likely	None	None
DBMS-dependent procedures	Limited	Limited	Variabile	High	Limited
Impact on operational system performance	None	None	Medium	Medium	None
Complexity of extraction procedures	Low	Low	High	Medium	Low

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

Incremental extraction

4/4/2010

Cod	Product	Customer	Qty
1	Greco di tufo	Malavasi	50
2	Barolo	Maio	150
3	Barbera	Lumini	75
4	Sangiovese	Cappelli	45

6/4/2010

Cod	Product	Customer	Qty
1	Greco di tufo	Malavasi	50
2	Barolo	Maio	150
4	Sangiovese	Cappelli	145
5	Vermentino	Maltoni	25
6	Trebbiano	Maltoni	150

Incremental difference

Cod	Product	Customer	Qty	Action
3	Barbera	Lumini	75	D
4	Sangiovese	Cappelli	145	U
5	Vermentino	Maltoni	25	I
6	Trebbiano	Maltoni	150	I

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

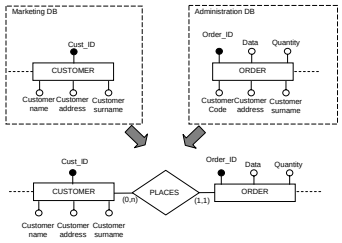
Data cleaning

- 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: join on similar fields, without foreign key
      - purge/merge problem: duplicate records identification
  - outlier identification, deviations from business rules
- Prevention is the best strategy
  - reliable and rigorous OLTP data entry procedures

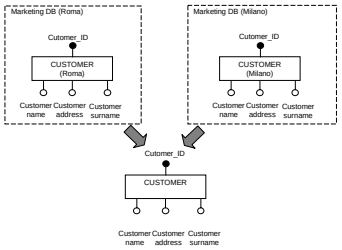
Approximate join



- 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

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 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:	I

Standardization

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

Correction

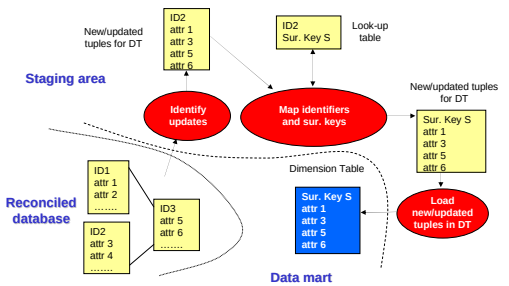
name:	Elena
surname:	Baralis
address:	Corso Duca degli Abruzzi 24
ZIP:	10129
city:	Torino
country:	Italia

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

Data warehouse loading

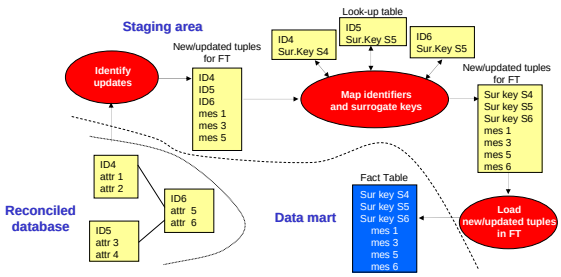
- Update propagation to the data warehouse
- Update order that preserves data integrity
  1. dimensions
  2. fact tables
  3. materialized views and indices
- Limited time window to perform updates
- Transactional properties are needed
  - reliability
  - atomicity

Dimension table loading



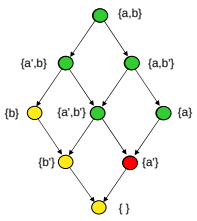
From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Fact table loading



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

Materialized view loading



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