



Database and data mining group, Politecnico di Torino 

## *Data warehouse design*

Elena Baralis  
Politecnico di Torino

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 1      Elena Baralis  
Politecnico di Torino

Database and data mining group, Politecnico di Torino 

## **Risk factors**

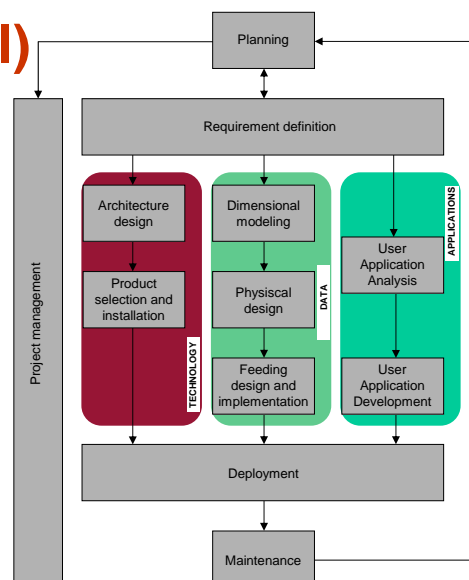
- High user expectation
  - the data warehouse is *the* solution of the company's problems
- Data and OLTP process quality
  - incomplete or unreliable data
  - non integrated or non optimized business processes
- “Political” management of the project
  - cooperation with “information owners”
  - system acceptance by end users
  - deployment
    - appropriate training

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 2      Elena Baralis  
Politecnico di Torino

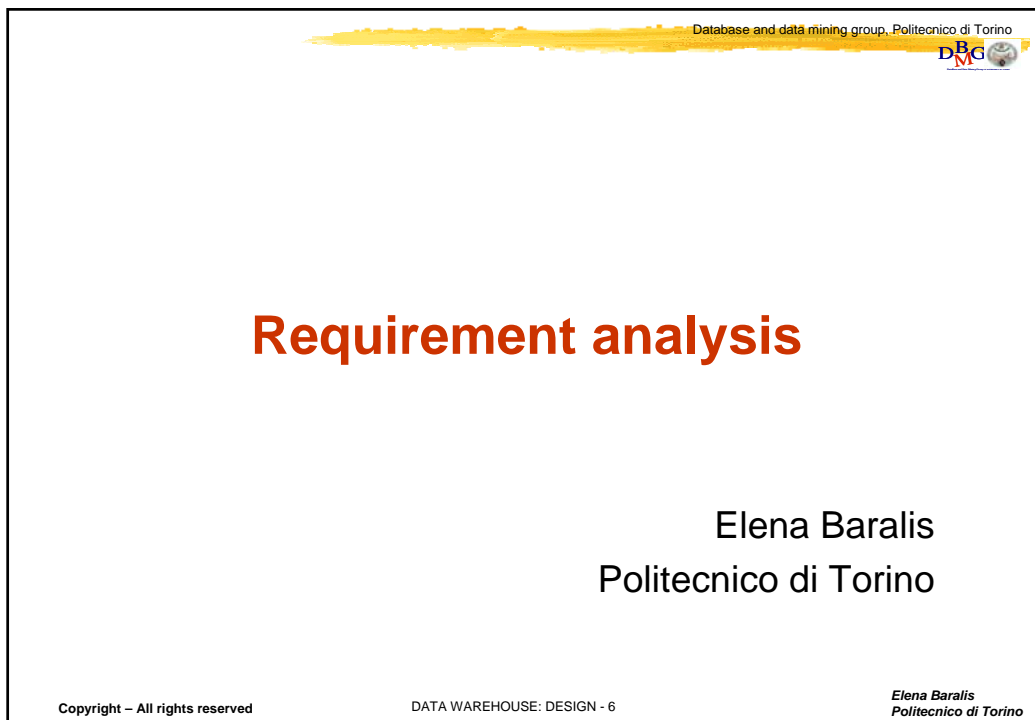
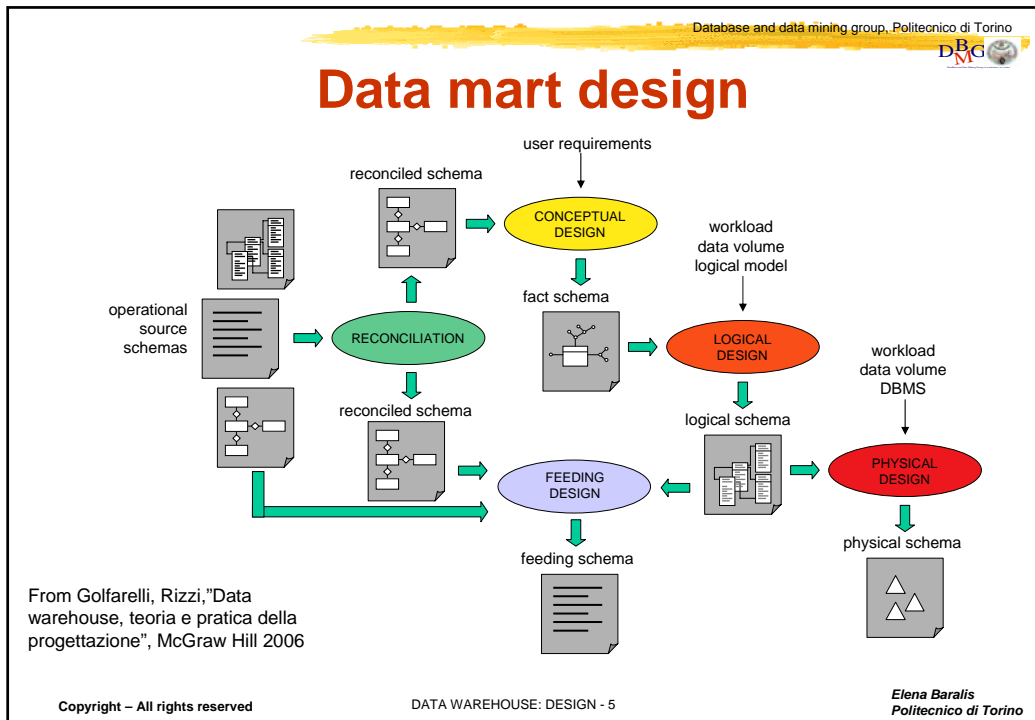
## Data warehouse design


- Top-down approach
  - the data warehouse provides a global and complete representation of business data
  - significant cost and time consuming implementation
  - complex analysis and design tasks
- Bottom-up approach
  - incremental growth of the data warehouse, by adding data marts on specific business areas
  - separately focused on specific business areas
  - limited cost and delivery time
  - easy to perform intermediate checks

## Business Dimensional Lifecycle (Kimball)



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




Database and data mining group, Politecnico di Torino  


## Requirement analysis

- It collects
  - data analysis requirements to be supported by the data mart
  - implementation constraints due to existing information systems
- Requirement sources
  - business users
  - operational system administrators
- The first selected data mart is
  - crucial for the company
  - feeded by (few) reliable sources

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 7      *Elena Baralis  
Politecnico di Torino*

Database and data mining group, Politecnico di Torino  


## Application requirements

- Description of relevant events (facts)
  - each fact represents a category of events which are relevant for the company
    - examples: (in the CRM domain) complaints, services
  - characterized by descriptive dimensions (setting the granularity), history span, relevant measures
  - informations are gathered in a glossary
- Workload description
  - periodical business reports
  - queries expressed in natural language
    - example: number of complaints for each product in the last month

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 8      *Elena Baralis  
Politecnico di Torino*

## Structural requirements

- Feeding periodicity
- Available space for
  - data
  - derived data (indices, materialized views)
- System architecture
  - level number
  - dependent or independent data marts
- Deployment planning
  - start up
  - training

## Conceptual design

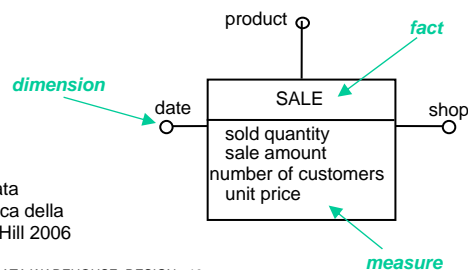
Elena Baralis  
Politecnico di Torino

## Conceptual design


- No currently adopted modeling formalism
  - ER model not adequate
- *Dimensional Fact Model* (Golfarelli, Rizzi)
  - graphical model supporting conceptual design
  - for a given fact, it defines a *fact schema* modelling
    - dimensions
    - hierarchies
    - measures
  - it provides design documentation both for requirement review with users, and after deployment

## Dimensional Fact Model

- Fact
  - it models a set of relevant events (sales, shippings, complaints)
  - it evolves with time
- Dimension
  - it describes the analysis coordinates of a fact (e.g., each sale is described by the sale date, the shop and the sold product)
  - it is characterized by many, typically categorical, attributes
- Measure
  - it describes a numerical property of a fact (e.g., each sale is characterized by a sold quantity)
  - aggregates are frequently performed on measures



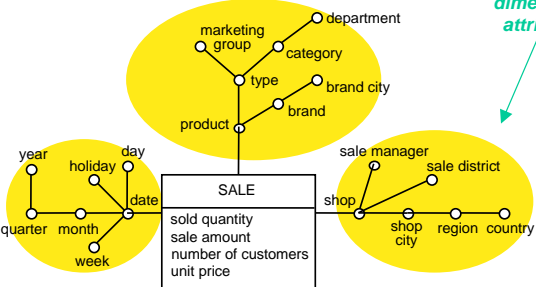
From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Database and data mining group, Politecnico di Torino  


## DFM: Hierarchy

- Each dimension can have a set of associated attributes
- The attributes describe the dimension at different abstraction levels and can be structured as a hierarchy
- The hierarchy represents a generalization relationship among a subset of attributes in a dimension (e.g., geographic hierarchy for the shop dimension)
- The hierarchy represents a functional dependency (1:n relationship)


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



dimension attribute

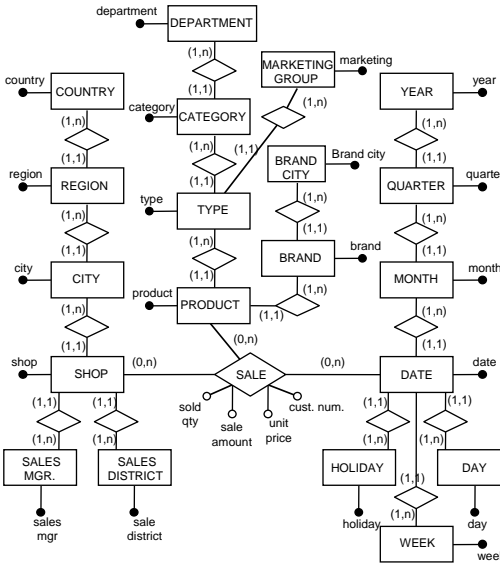
hierarchy

Copyright - All rights reserved
DATA WAREHOUSE: DESIGN - 13
Elena Baralis  
Politecnico di Torino

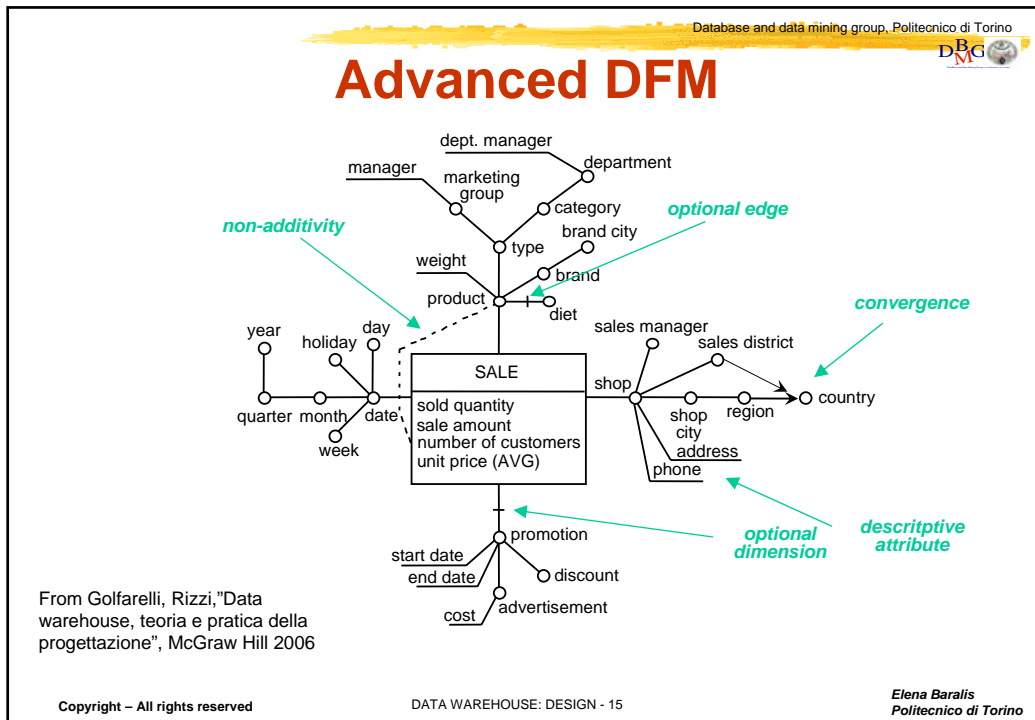
Database and data mining group, Politecnico di Torino  


## Comparison with ER

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



Copyright - All rights reserved
DATA WAREHOUSE: DESIGN - 14
Elena Baralis  
Politecnico di Torino



Database and data mining group, Politecnico di Torino  
DBG

## Aggregation

- Aggregation computes measures with a coarser granularity than those in the original fact schema
  - detail reduction is usually obtained by climbing a hierarchy
  - standard aggregate operators: SUM, MIN, MAX, AVG, COUNT
- Measure characteristics
  - additive
  - not additive: cannot be aggregated along a given hierarchy by means of the SUM operator
  - not aggregable

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 16      Elena Baralis Politecnico di Torino



Database and data mining group, Politecnico di Torino  
DBG

## Measure classification

- Stream measures
  - can be evaluated cumulatively at the end of a time period
  - can be aggregated by means of all standard operators
  - examples: sold quantity, sale amount
- Level measures
  - evaluated at a given time (snapshot)
  - not additive along the time dimension
  - examples: inventory level, account balance
- Unit measures
  - evaluated at a given time and expressed in relative terms
  - not additive along any dimension
  - examples: unit price of a product

Copyright – All rights reserved
DATA WAREHOUSE: DESIGN - 17
Elena Baralis  
Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Aggregate operators

category	type	product	1999				2000			
			I '99	II '99	III '99	IV '99	I '00	II '00	III '00	IV '00
home cleaning	washing powder	Brillo	100	90	95	90	80	70	90	85
		Sbianco	20	30	20	10	25	30	35	20
	soap	Lucido	60	50	60	45	40	40	50	40
		Manipulite	15	20	25	30	15	15	20	10
food	milk	Scent	30	35	20	25	30	30	20	15
		Latte F Slurp	90	90	85	75	60	80	85	60
		Latte U Slurp	60	80	85	60	70	70	75	65
	soda	Yogurt Slurp	20	30	40	35	30	35	35	20
		Bevimi	20	10	25	30	35	30	20	10
		Colissima	50	60	45	40	50	60	45	40

↓

category	year	1999				2000			
		I '99	II '99	III '99	IV '99	I '00	II '00	III '00	IV '00
home clean		225	225	220	200	190	185	215	170
food		240	270	280	240	245	275	260	195

↓

category	year	1999	2000
		home clean	870
food	1030	975	

↓

category	type	1999		2000	
		home cleaning	washing p	670	605
food	soap	200	155		
	milk	750	685		
	soda	280	290		

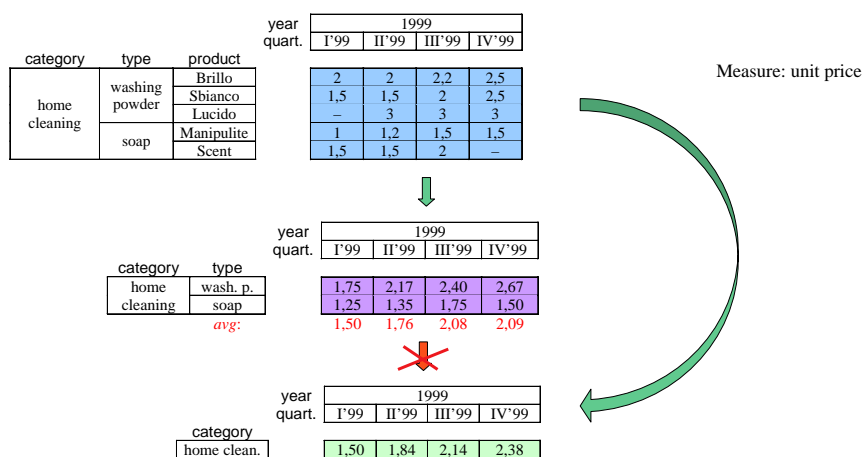
From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Copyright – All rights reserved
DATA WAREHOUSE: DESIGN - 18
Elena Baralis  
Politecnico di Torino

## Aggregate operators

- Distributive
  - can always compute higher level aggregations from more detailed data
  - examples: sum, min, max

## Non distributive operators

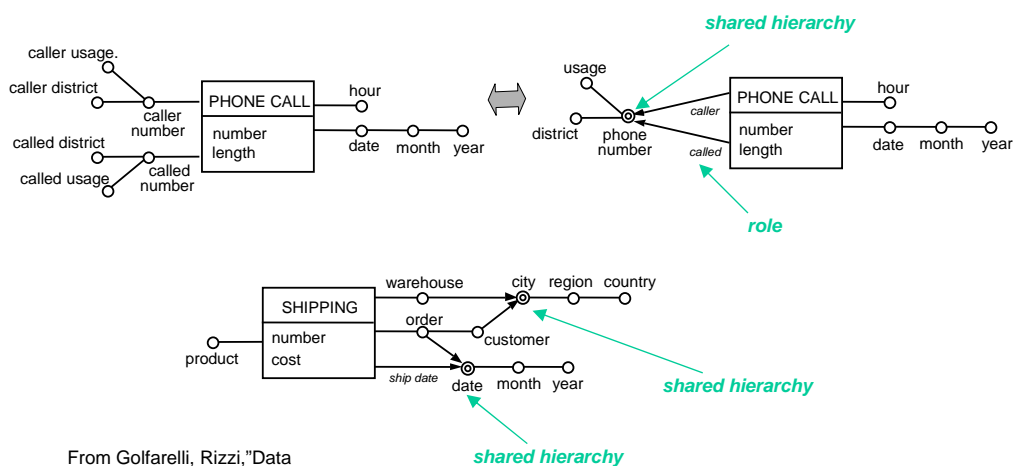


From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

## Aggregate operators

- Distributive
  - can always compute higher level aggregations from more detailed data
  - examples: sum, min, max
- Algebraic
  - can compute higher level aggregations from more detailed data *only* when supplementary support measures are available
  - examples: avg (it requires count)
- Olistic
  - *can not* compute higher level aggregations from more detailed data
  - examples: mode, median

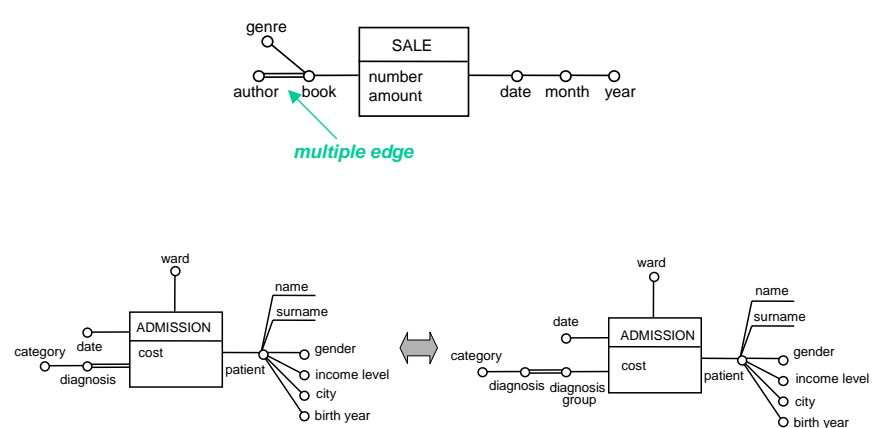
## Advanced DFM



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

Database and data mining group, Politecnico di Torino  
DBG

## Advanced DFM



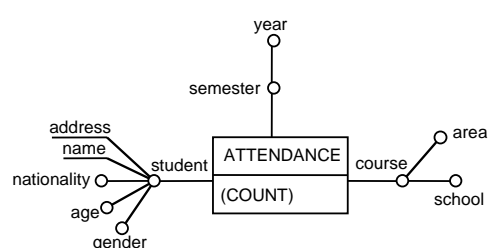
From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 23      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Factless fact schema

- Some events are not characterized by measures
  - empty (i.e., factless) fact schema
  - it records occurrence of an event
- Used for
  - counting occurred events (e.g., course attendance)
  - representing events not occurred (coverage set)



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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 24      Elena Baralis Politecnico di Torino

## Representing time

- Data modification over time is explicitly represented by event occurrences
  - time dimension
  - events stored as facts
- Also dimensions may change over time
  - modifications are typically slower
    - slowly changing dimension [Kimball]
  - examples: client demographic data, product description
  - if required, dimension evolution should be explicitly modeled

## How to represent time (type I)

- Snapshot of the current value
  - data is overwritten with the current value
  - it overrides the past with the current situation
  - used when an explicit representation of the data change is not needed
  - example
    - customer Mario Rossi changes marital status after marriage
    - all his purchases correspond to the “married” customer

## How to represent time (type II)

- Events are related to the temporally corresponding dimension value
  - after each state change in a dimension
    - a new dimension instance is created
    - new events are related to the new dimension instance
  - events are partitioned after the changes in dimensional attributes
  - example
    - customer Mario Rossi changes marital status after marriage
    - his purchases are partitioned in purchases performed by “unmarried” Mario Rossi and purchases performed by “married” Mario Rossi (a new instance of Mario Rossi)

## How to represent time (type III)

- All events are mapped to a dimension value sampled at a given time
  - it requires the explicit management of dimension changes during time
    - the dimension schema is modified by introducing
      - two timestamps: validity start and validity end
      - a new attribute which allows identifying the sequence of modifications on a given instance (e.g., a “master” attribute pointing to the root instance)
    - each state change in the dimension requires the creation of a new instance

## How to represent time (type III)

- Example
  - customer Mario Rossi changes marital status after marriage
  - validity end timestamp of first Mario Rossi instance is given by the marriage date
  - validity start timestamp of the new instance is the same day
  - purchases are partitioned as in type II
  - a new attribute allows tracking all changes of Mario Rossi instance

## Workload

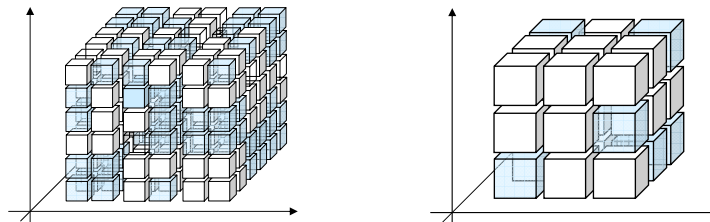
- Workload defined by
  - standard reports
  - approximate estimates discussed with users
- Actual workload difficult to evaluate at design time
  - if the data warehouse succeeds, user and query number may grow
  - query type may vary over time
- Data warehouse tuning
  - performed after system deployment
  - requires monitoring the actual system workload

## Data volume

- Estimation of the space required by the data mart
  - for data
  - for derived data (indices, materialized views)
- To be considered
  - event cardinality for each fact
  - domain cardinality (number of distinct values) for hierarchy attributes
  - attribute length
- It depends on the temporal span of data storage
- Sparsity
  - occurred events are not all combinations of the dimension elements
  - example: the percentage of products actually sold in each shop and day is roughly 10% of all combinations

## Sparsity

- It decreases with increasing data aggregation level
- May significantly affect the accuracy in estimating aggregated data cardinality



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




Database and data mining group, Politecnico di Torino 

## *Logical design*

Elena Baralis  
Politecnico di Torino

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 33      Elena Baralis  
Politecnico di Torino

Database and data mining group, Politecnico di Torino 

## *Logical design*

- We address the relational model (ROLAP)
  - inputs
    - conceptual fact schema
    - workload
    - data volume
    - system constraints
  - output
    - relational logical schema
- Based on different principles with respect to traditional logical design
  - data redundancy
  - table denormalization

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 34      Elena Baralis  
Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

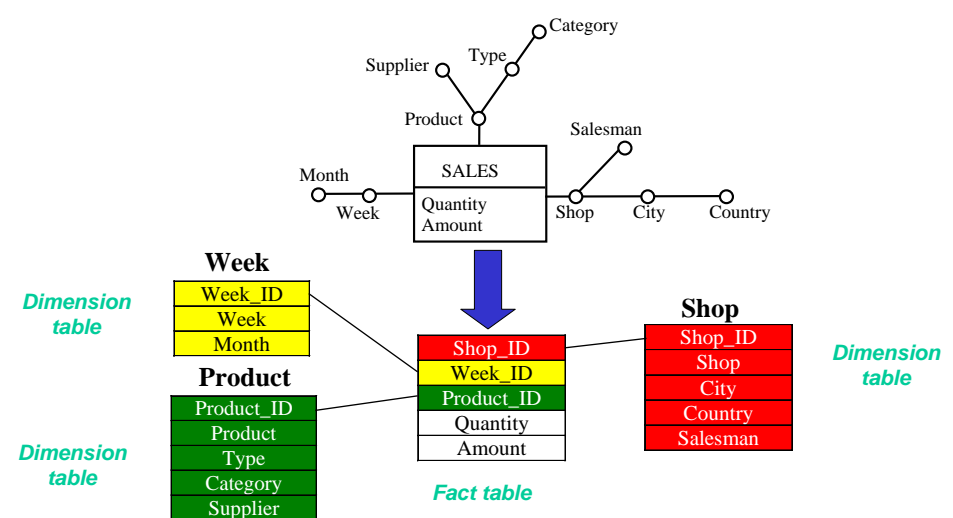
## Star schema

- Dimensions
  - one table for each dimension
  - surrogate (generated) primary key
  - it contains all dimension attributes
  - hierarchies are not explicitly represented
    - all attributes in a table are at the same level
  - totally denormalized representation
    - it causes data redundancy
- Facts
  - one fact table for each fact schema
  - primary key composed by foreign keys of all dimensions
  - measures are attributes of the fact table

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 35      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Star schema



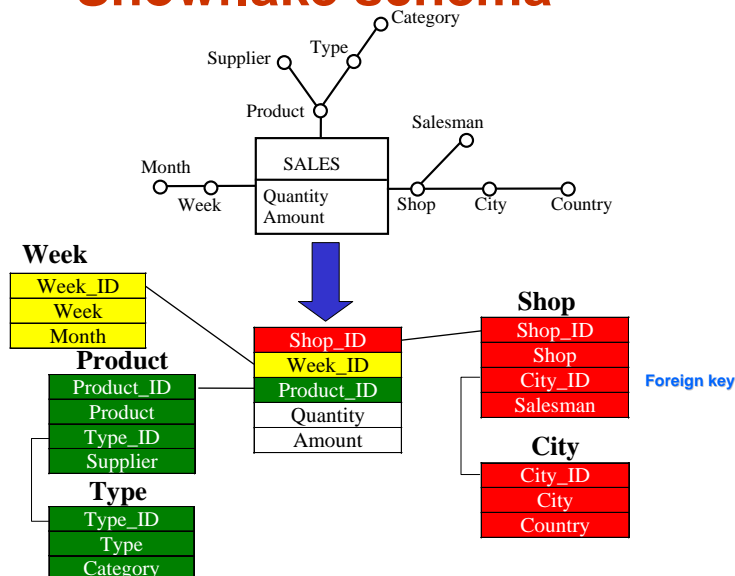
From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 36      Elena Baralis Politecnico di Torino

## Snowflake schema

- Some functional dependencies are separated, by partitioning dimension data in several tables
  - a new table separates two branches of a dimensional hierarchy (hierarchy is cut on a given attribute)
  - a new foreign key correlates the dimension with the new table
- Decrease in space required for storing the dimension
  - decrease is frequently not significant
- Increase in cost for reading entire dimension
  - one or more joins are needed

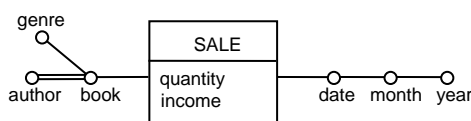
## Snowflake schema



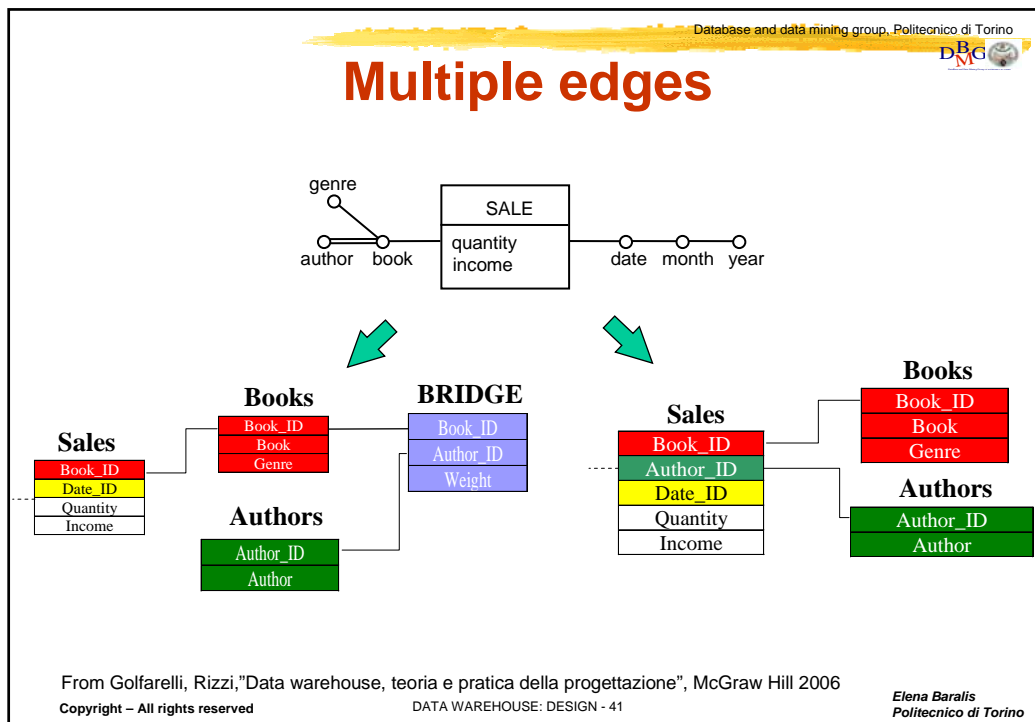
## Star or snowflake?

- The snowflake schema is usually not recommended
  - storage space decrease is rarely beneficial
    - most storage space is consumed by the fact table (difference with dimensions is several orders of magnitude)
  - cost of join execution may be significant
- The snowflake schema may be useful
  - when part of a hierarchy is shared among dimensions (e.g., geographic hierarchy)
  - for materialized views, which require an aggregate representation of the corresponding dimensions

## Multiple edges



- Implementation techniques
  - bridge table
    - new table which models many to many relationship
    - new attribute weighting the contribution of tuples in the relationship
  - push down
    - multiple edge integrated in the fact table
    - new corresponding dimension in the fact table



Database and data mining group, Politecnico di Torino  
DBG

## Multiple edges

- Queries
  - Weighted query: consider the weight of the multiple edge
    - example: author income
    - by using bridge table:
 

```
SELECT Author_ID, SUM(Income*Weight)
...
group by Author_ID
```
  - Impact query: do not consider the weight of the multiple edge
    - example: book copies sold for each author
    - by using bridge table:
 

```
SELECT Author_ID, SUM(Quantity)
...
group by Author_ID
```

Copyright - All rights reserved

DATA WAREHOUSE: DESIGN - 42

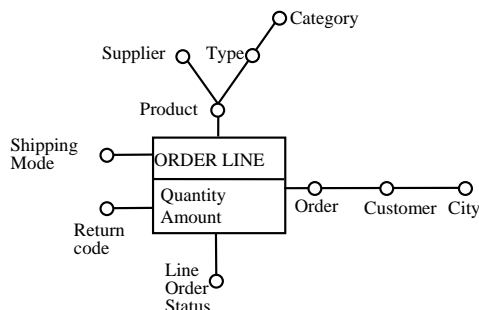
Elena Baralis  
Politecnico di Torino

## Multiple edges

- Comparison
  - weight is explicit in the bridge table, but wired in the fact table for push down
    - (push down) hard to perform impact queries
    - (push down) weight is computed when feeding the DW
    - (push down) weight modifications are hard
  - push down causes significant redundancy in the fact table
  - query execution cost is lower for push down
    - less joins

## Degenerate dimensions

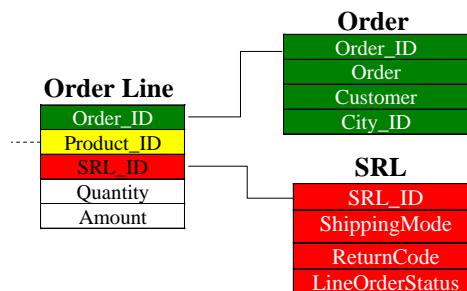
- Dimensions with a single attribute




## Degenerate dimensions

- Implementations
  - (usually) directly integrated into the fact table
    - only for attributes with a (very) small size
  - junk dimension
    - single dimension containing several degenerate dimensions
    - no functional dependencies among attributes in the junk dimension
      - all attribute value combinations are allowed
      - feasible only for attribute domains with small cardinality

## Junk dimension




Database and data mining group, Politecnico di Torino  


# Materialized views

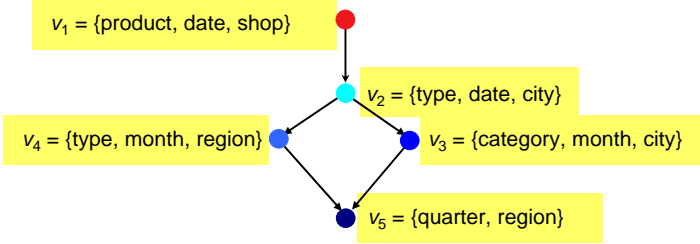
Elena Baralis  
 Politecnico di Torino

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 47      *Elena Baralis  
 Politecnico di Torino*

Database and data mining group, Politecnico di Torino  


# Materialized views

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



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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 48      *Elena Baralis  
 Politecnico di Torino*



Database and data mining group, Politecnico di Torino  
DBG

## Materialized views

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

Month	
Month_ID	
Month	
Year	

City	
City_ID	
City	
State	

Category	
Category_ID	
Category	
Department	

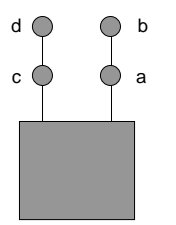
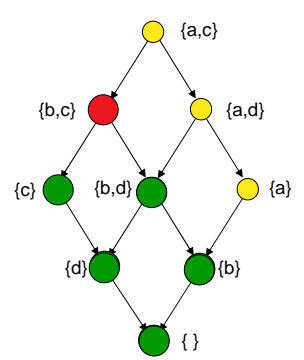
City_ID	
Month_ID	
Category_ID	
TotalQuantity	
TotalIncome	

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 49      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Materialized views

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


➔


Multidimensional lattice

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 50      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 51      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Materialized view selection

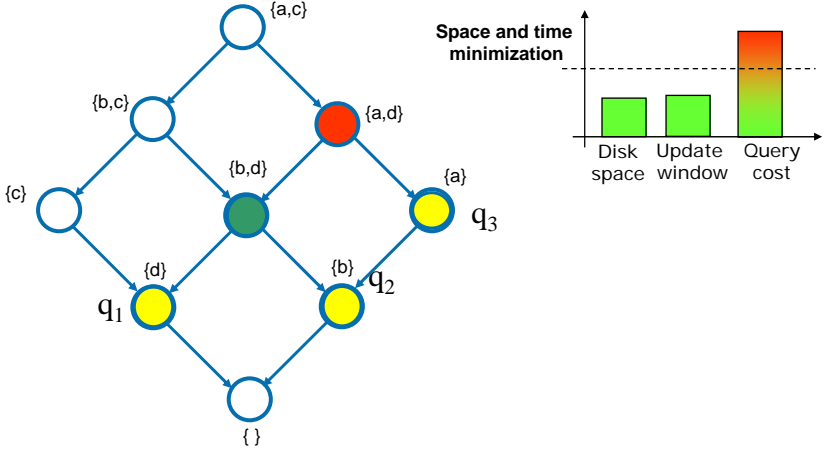
**Multidimensional lattice**

● + ● = *candidate views*,  
 possibly useful to  
 increase workload  
 query performance

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
 Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 52      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Materialized view selection



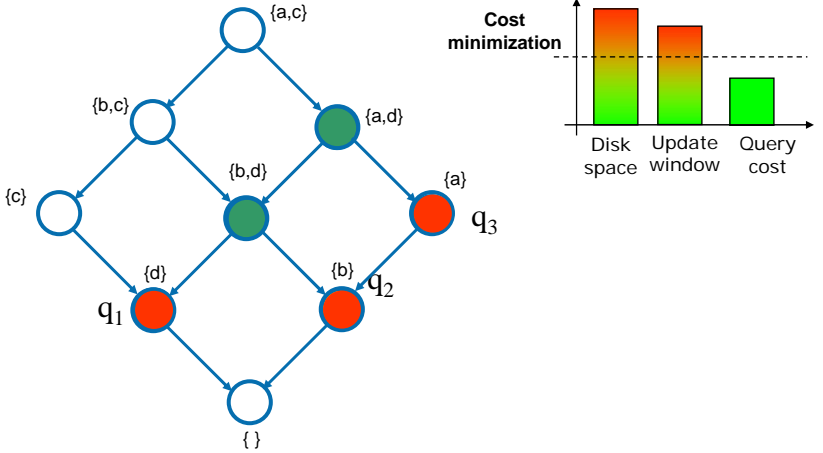
Space and time minimization

Disk space window    Update cost    Query cost

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 53      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG


## Materialized view selection



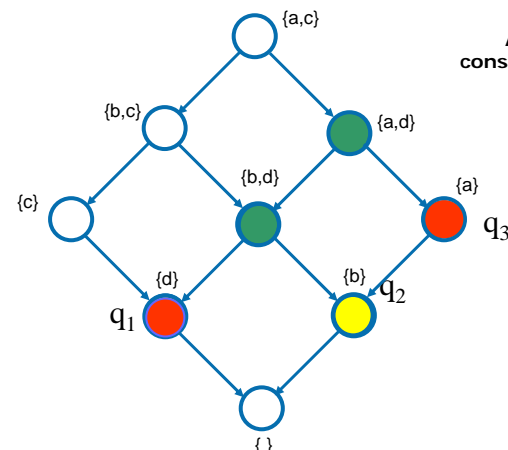
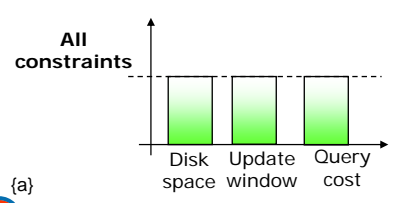
Cost minimization

Disk space window    Update cost    Query cost


From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 54      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## Materialized view selection


From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
 Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 55      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## Physical design

Elena Baralis  
 Politecnico di Torino


Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 56      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## Physical design

- Workload characteristics
  - aggregate queries which require accessing a large fraction of each table
  - read-only access
  - periodic data refresh, possibly rebuilding physical access structures (indices, views)
- Physical structures
  - index types different from OLTP
    - bitmap index, join index, bitmapped join index, ...
    - B<sup>+</sup>-tree index not appropriate for
      - attributes with low cardinality domains
      - queries with low selectivity
  - materialized views
    - query optimizer should be able to exploit them

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 57      *Elena Baralis  
Politecnico di Torino*

Database and data mining group, Politecnico di Torino  


## Physical design

- Optimizer characteristics
  - should consider statistics when defining the access plan (cost based)
  - aggregate navigation
- Physical design procedure
  - selection of physical structures supporting most frequent (or most relevant) queries
  - selection of structures improving performance of more than one query
  - constraints
    - disk space
    - available time window for data update


Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 58      *Elena Baralis  
Politecnico di Torino*

## Physical design

- Tuning
  - a posteriori change of physical access structures
  - workload monitoring tools are needed
  - frequently required for OLAP applications
- Parallelism
  - data fragmentation
  - query parallelization
    - inter-query
    - intra-query
  - join and group by lend themselves well to parallel execution

## Index selection


- Indexing dimensions
  - attributes frequently involved in selection predicates
  - if domain cardinality is high, then B-tree index
  - if domain cardinality is low, then bitmap index
- Indices for join
  - indexing only foreign keys in the fact table is *rarely* appropriate
  - bitmapped join index is suggested (if available)
- Indices for group by
  - use materialized views

Database and data mining group, Politecnico di Torino 

## **ETL Process**

Elena Baralis  
Politecnico di Torino


Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 61      Elena Baralis  
Politecnico di Torino

Database and data mining group, 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


Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 62      Elena Baralis  
Politecnico di Torino

Database and data mining group, Politecnico di Torino 

## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 63      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino 

## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 64      Elena Baralis Politecnico di Torino



## 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

Database and data mining group, Politecnico di Torino

## Comparison of extraction techniques

	Static	Timestamps	Application 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	Variable	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, Addison-Wesley, 1997  
 Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 67      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino

## 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  
 Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 68      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 69      *Elena Baralis  
Politecnico di Torino*

Database and data mining group, Politecnico di Torino  


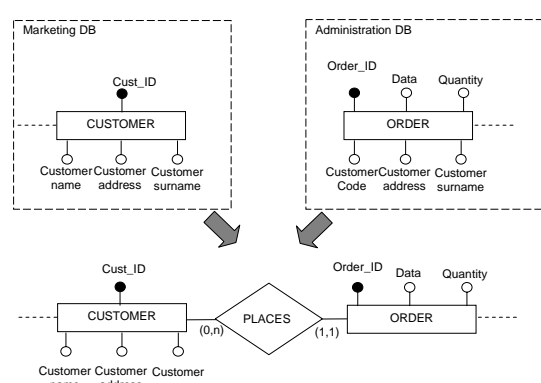
## Data cleaning

- 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 70      *Elena Baralis  
Politecnico di Torino*

Database and data mining group, Politecnico di Torino  
DBG

## Approximate join

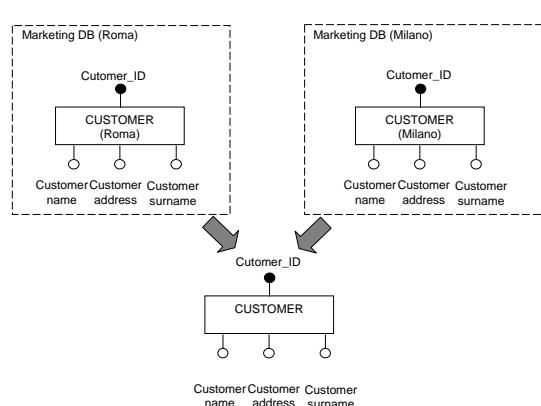


- The join operation should be executed based on common fields, not representing the customer identifier

From Golfarelli, Rizzi, "Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006  
Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 71      Elena Baralis Politecnico di Torino


Database and data mining group, Politecnico di Torino  
DBG

## 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  
Copyright - All rights reserved      DATA WAREHOUSE: DESIGN - 72      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 73      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  


## 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 74      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

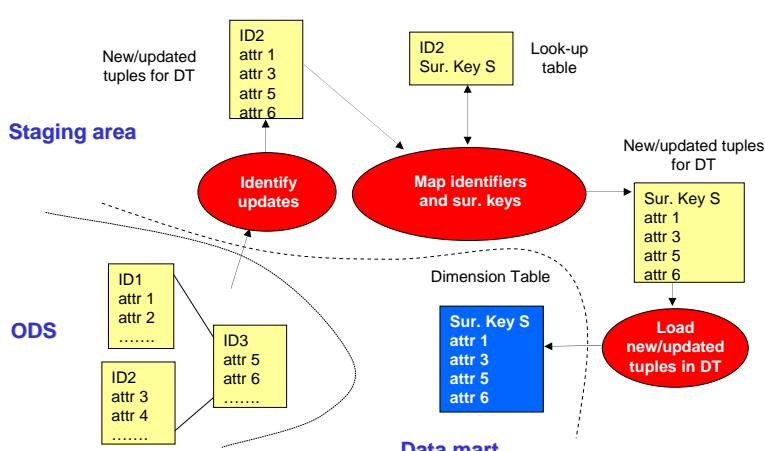
## Data warehouse loading

- 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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 75      Elena Baralis Politecnico di Torino

Database and data mining group, Politecnico di Torino  
DBG

## Dimension table loading



The diagram illustrates the dimension table loading process. It shows the flow of data from the ODS (Operational Data Store) through a Staging area to the final Dimension Table in the Data mart. The ODS contains source identifiers (ID1, ID2, ID3) and their attributes. In the Staging area, these are processed through 'Identify updates' and 'Map identifiers and sur. keys' steps. A 'Look-up table' (ID2 Sur. Key S) is used to map source identifiers to surrogate keys. The final 'Load new/updated tuples in DT' step populates the 'Dimension Table' (Sur. Key S, attr 1, attr 3, attr 5, attr 6) in the Data mart.

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

Copyright – All rights reserved      DATA WAREHOUSE: DESIGN - 76      Elena Baralis Politecnico di Torino

