Data warehouse design

Elena Baralis

Politecnico di Torino

Risk factors

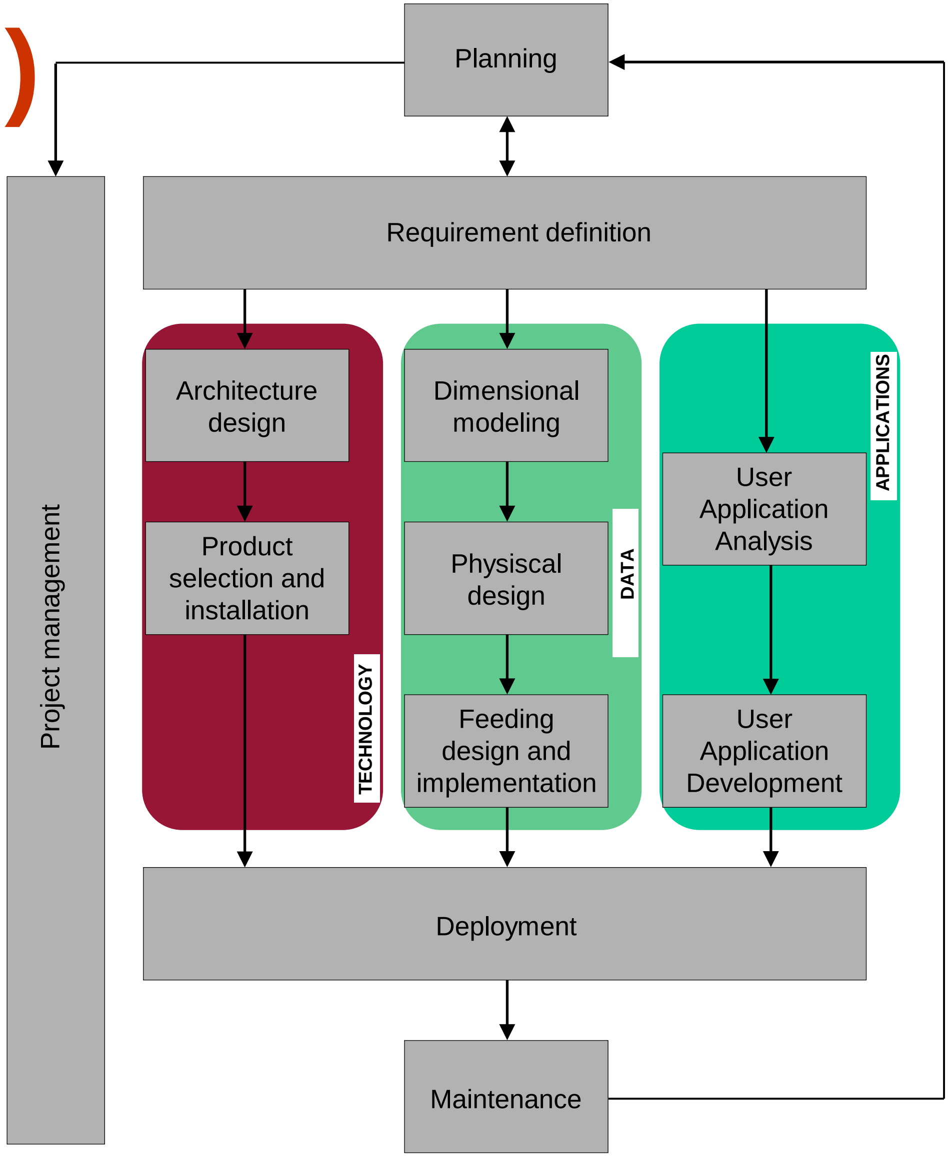
- High user expectation
  - the data warehouse is *the* solution of the company’s problems
- Data and OLTP process quality
  - incomplete or unreliable data
  - non integrated or non optimized business processes
- “Political” management of the project
  - cooperation with “information owners”
  - system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

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

Elena Baralis
Politecnico di Torino

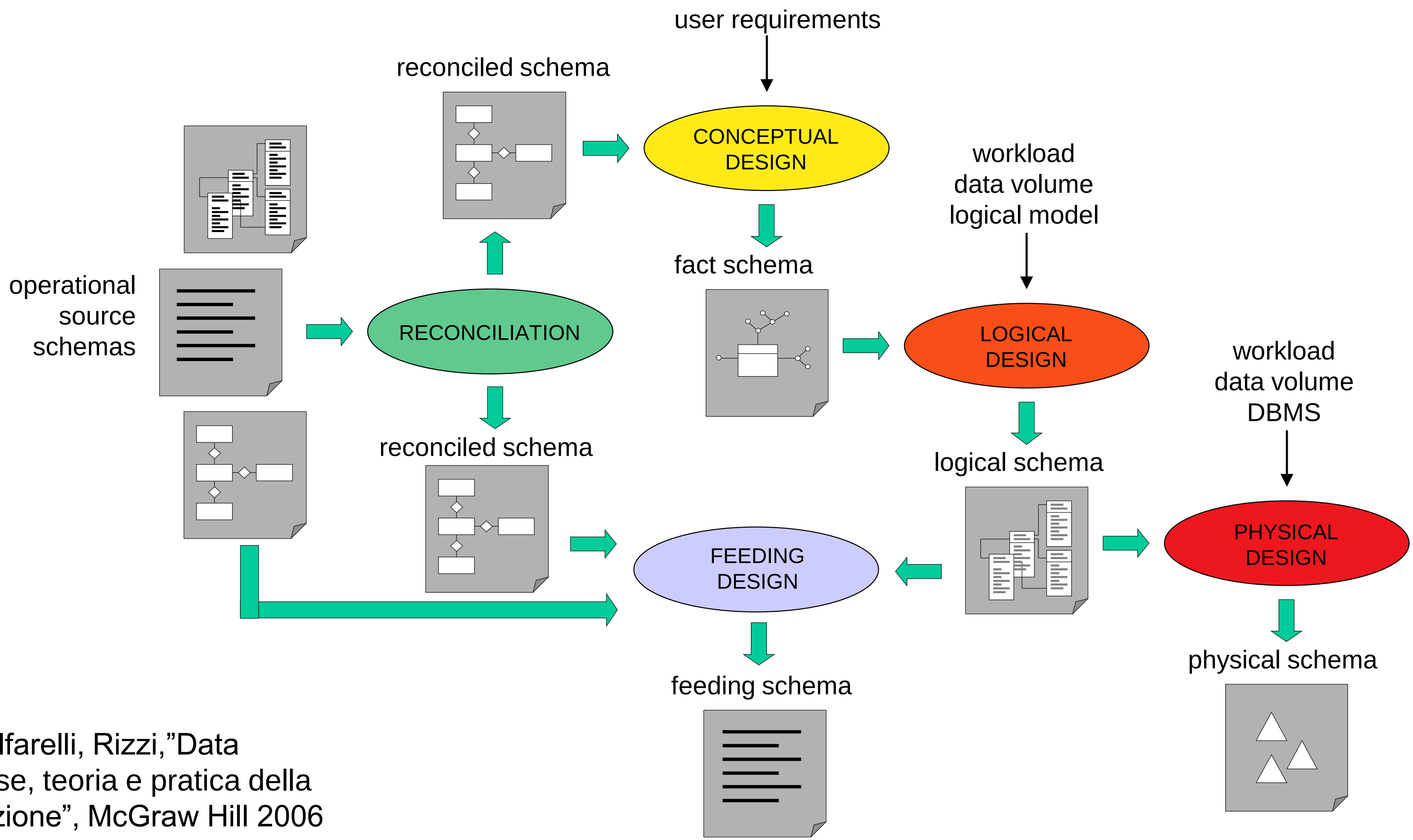
Business Dimensional Lifecycle (Kimball)



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

Elena Baralis
Politecnico di Torino

Data mart design



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

Elena Baralis
Politecnico di Torino

Requirement analysis

Elena Baralis

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

Elena Baralis
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

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

Elena Baralis
Politecnico di Torino

Conceptual design

Elena Baralis
Politecnico di Torino

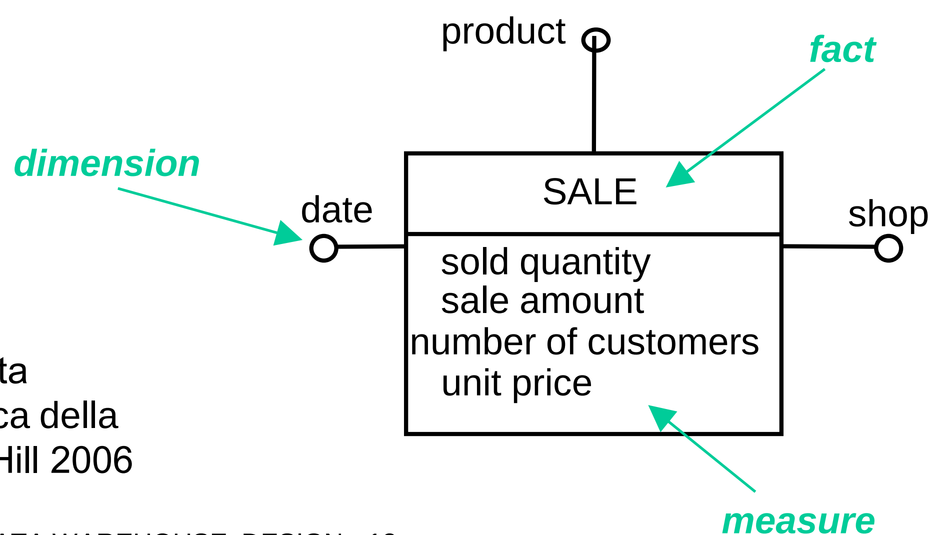
Conceptual design

- 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

Elena Baralis
Politecnico di Torino

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

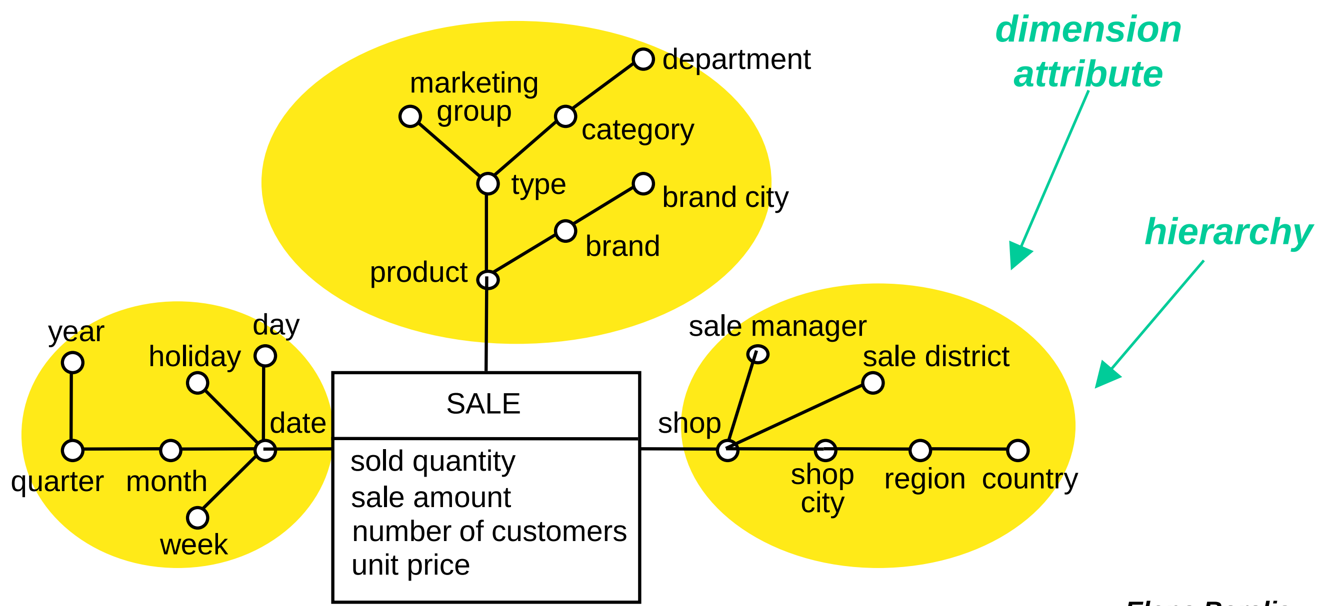


From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Elena Baralis
Politecnico di Torino

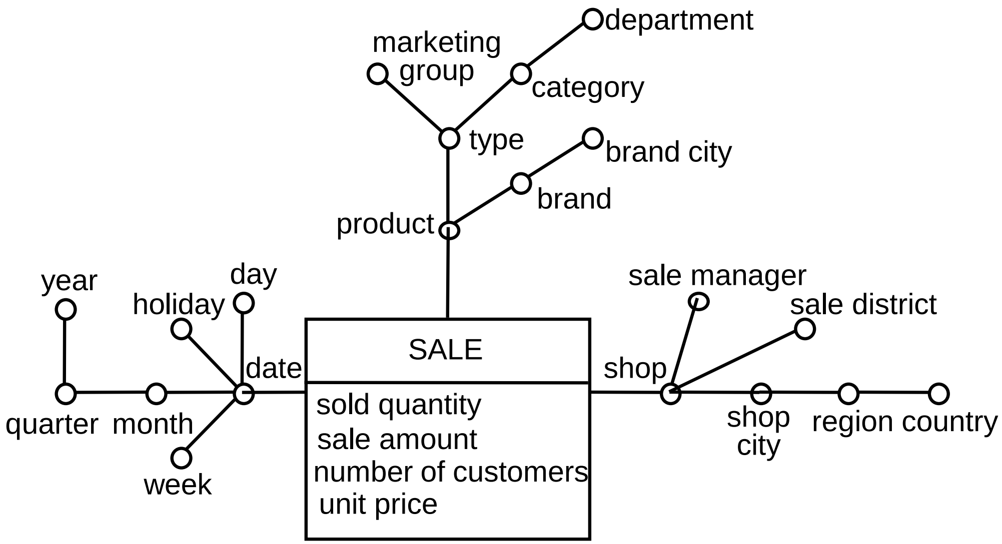
DFM: Hierarchy

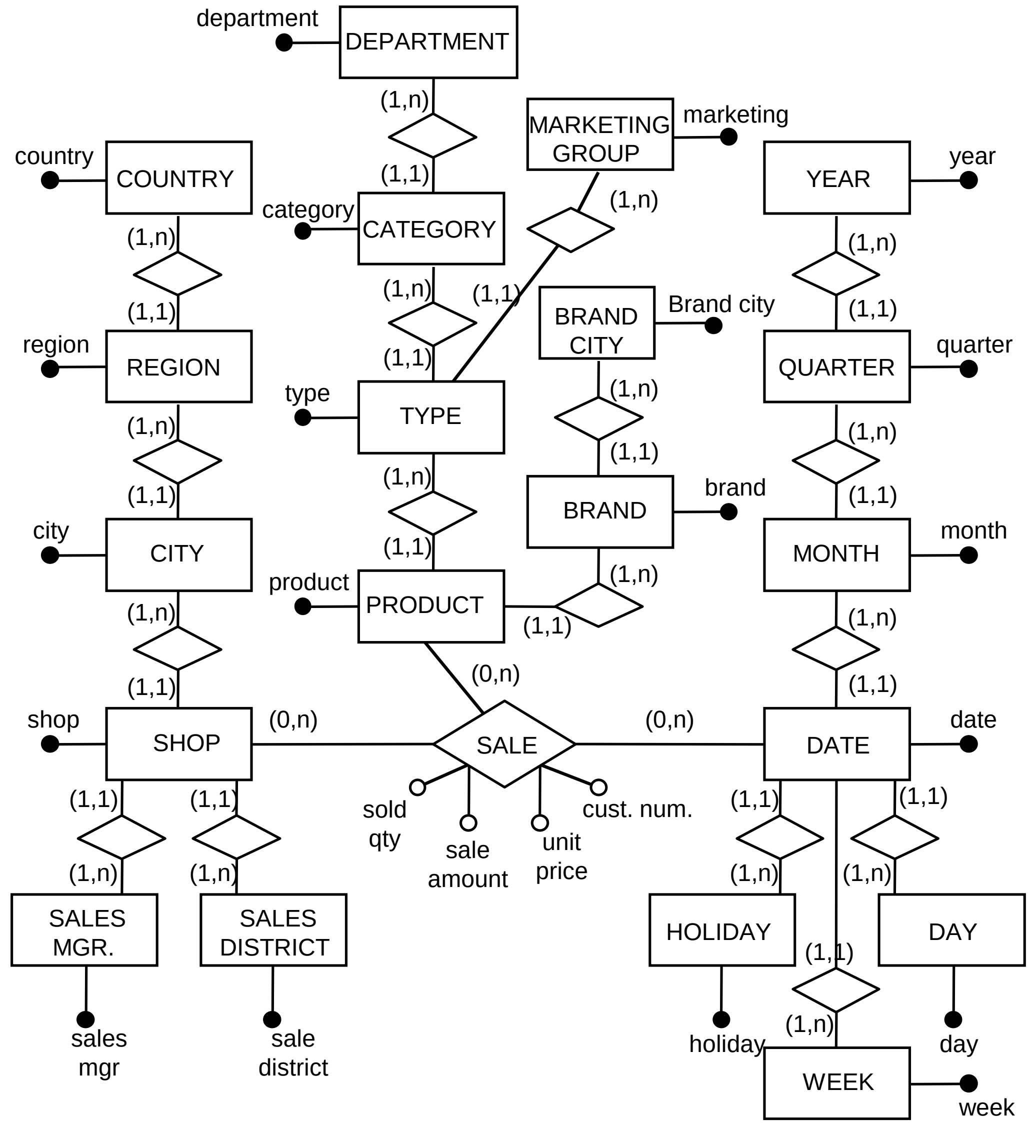
- Each dimension can have a set of associated attributes
- The attributes describe the dimension at different abstraction levels and can be structured as a hierarchy
- The hierarchy represents a generalization relationship among a subset of attributes in a dimension (e.g., geografic hierarchy for the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



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

Comparison with ER

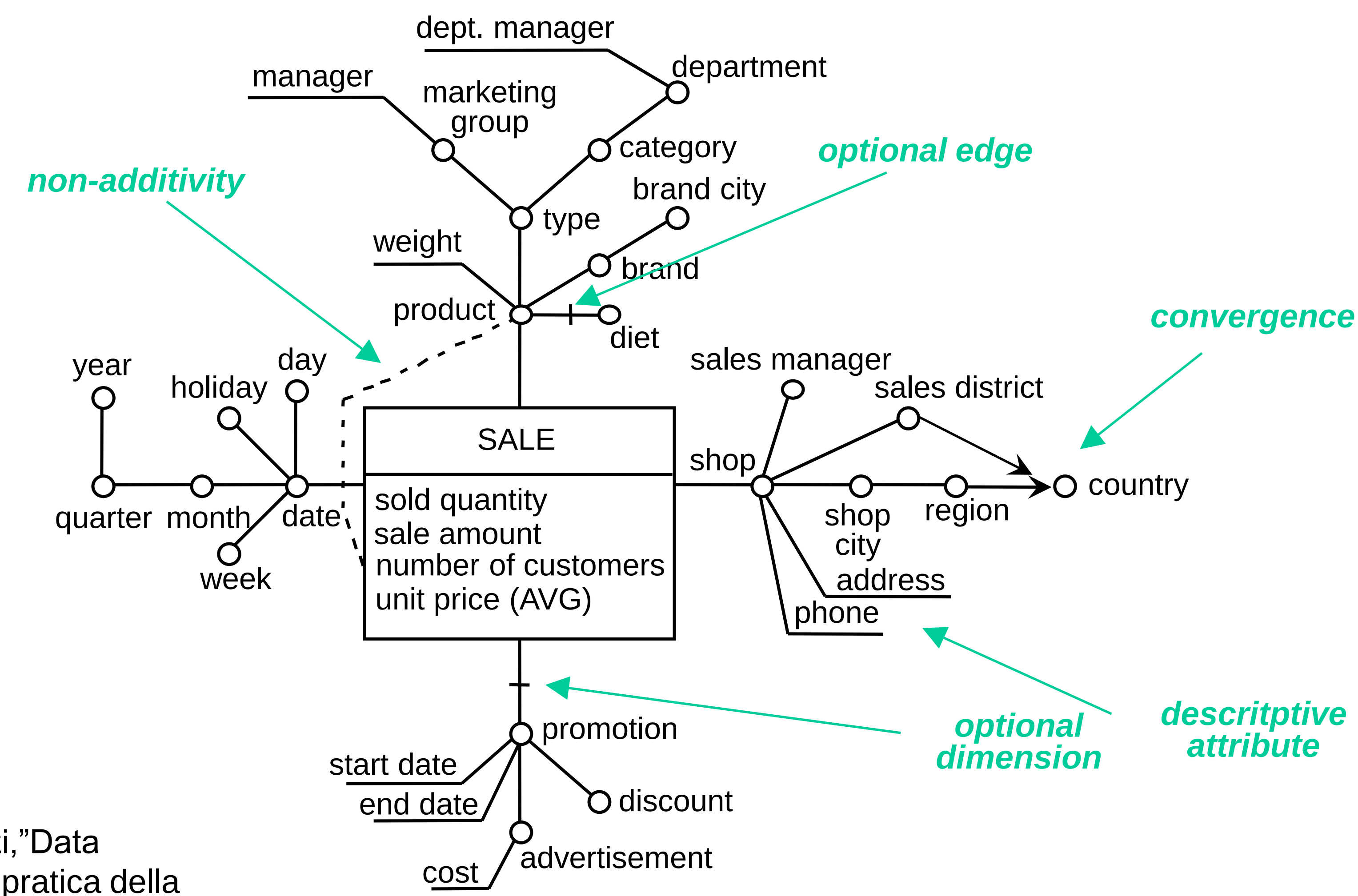




From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Elena Baralis
Politecnico di Torino

Advanced DFM



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

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Aggregate operators

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

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

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

category	1999	2000
home clean.	870	760
food	1030	975

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

Elena Baralis
Politecnico di Torino

Aggregate operators

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

Elena Baralis
Politecnico di Torino

Non distributive operators

Measure: unit price

category	type	product	year 1999: quart. I'99	II'99	III'99	IV'99
home cleaning	washing powder	Brillo	2	2	2,2	2,5
		Sbianco	1,5	1,5	2	2,5
		Lucido	–	3	3	3
	soap	Manipulite	1	1,2	1,5	1,5
		Scent	1,5	1,5	2	–

category	type	year 1999: quart. I'99	II'99	III'99	IV'99
home cleaning	wash. p.	1,75	2,17	2,40	2,67
	soap	1,25	1,35	1,75	1,50
	avg:	1,50	1,76	2,08	2,09

category	year 1999: quart. I'99	II'99	III'99	IV'99
home clean.	1,50	1,84	2,14	2,38

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

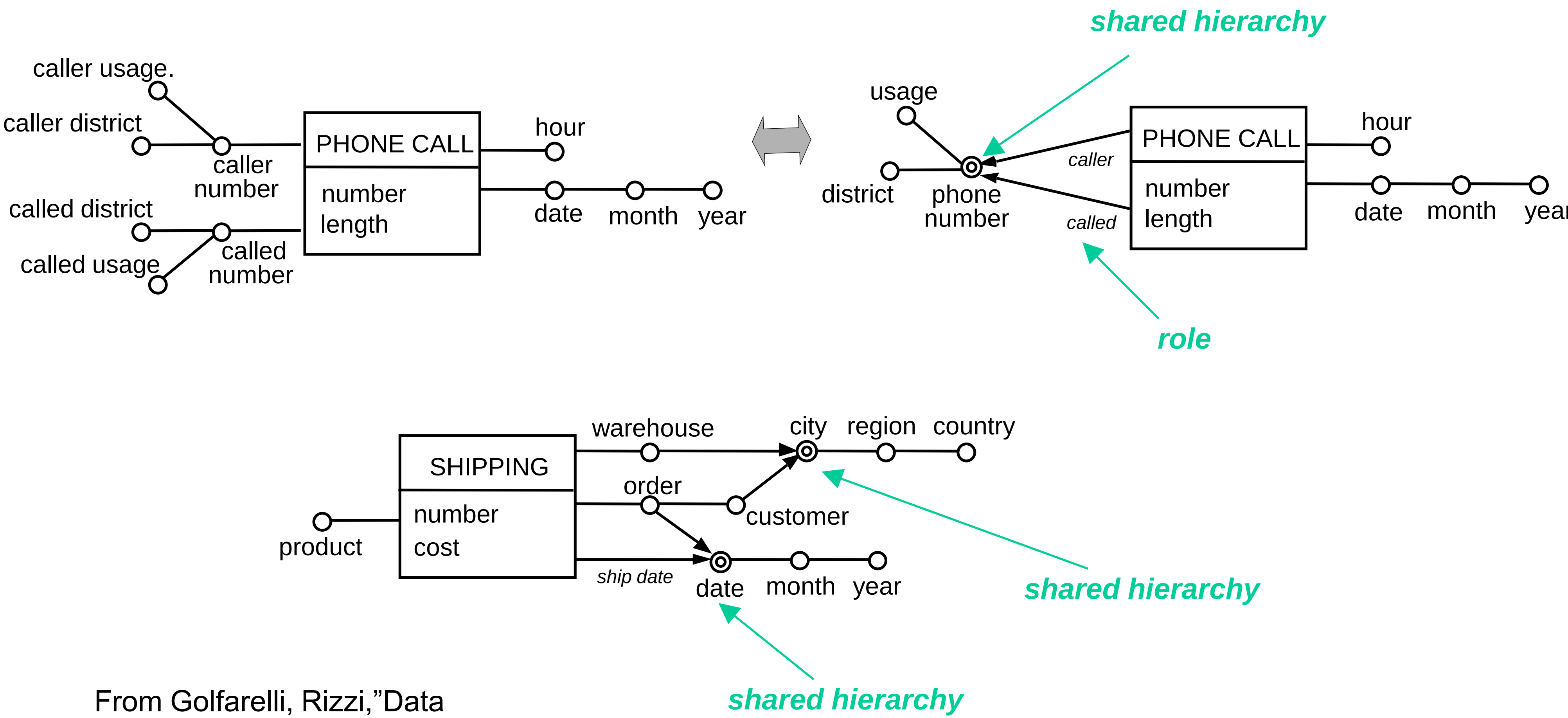
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

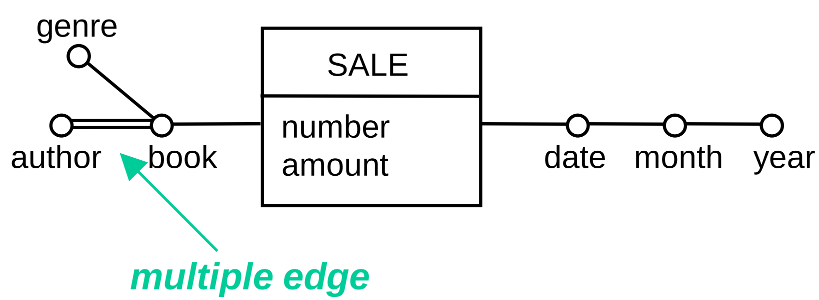
Advanced DFM

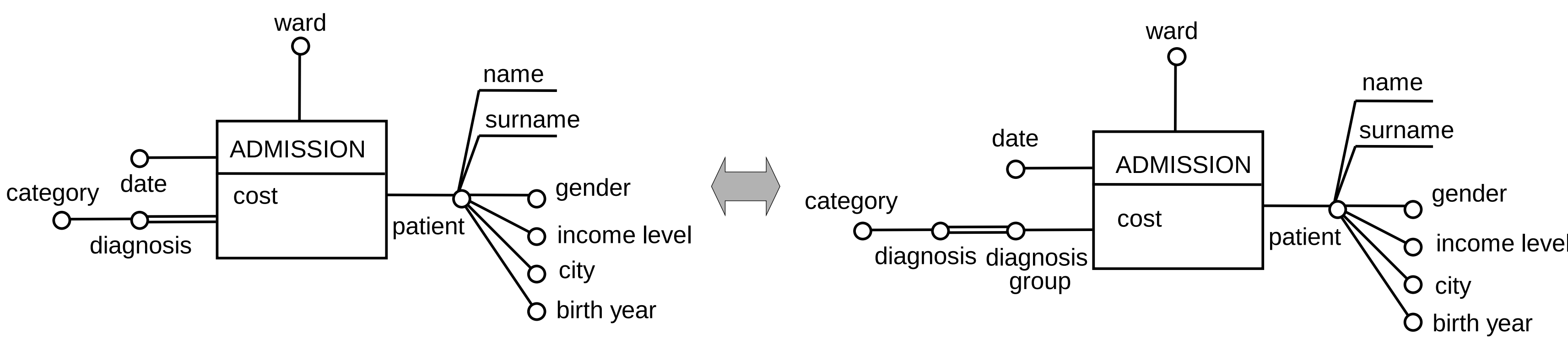


From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Elena Baralis
Politecnico di Torino

Advanced DFM

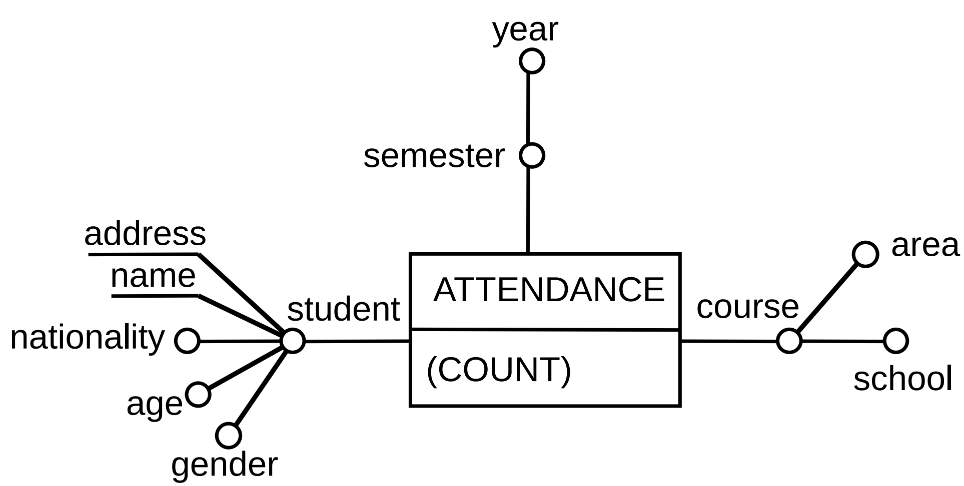




From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

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

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

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)

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

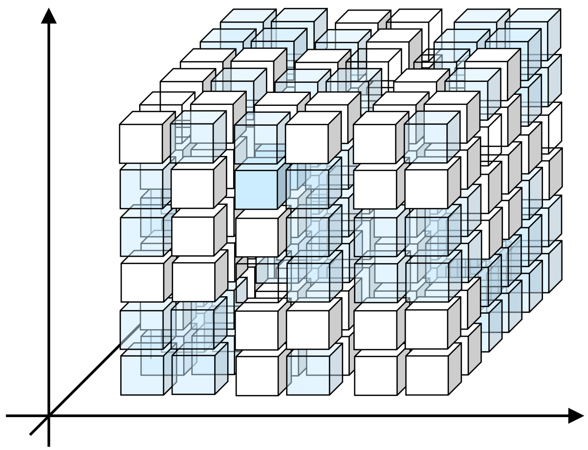
Data volume

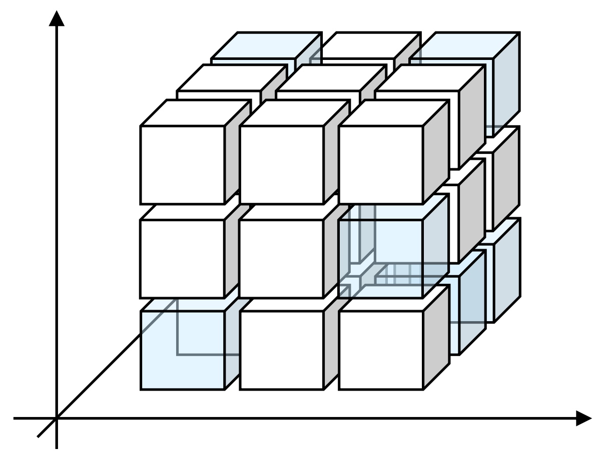
- Estimation of the space required by the data mart
  - for data
  - for derived data (indices, materialized views)
- To be considered
  - event cardinality for each fact
  - domain cardinality (number of distinct values) for hierarchy attributes
  - attribute length
- It depends on the temporal span 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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Logical design

Elena Baralis
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

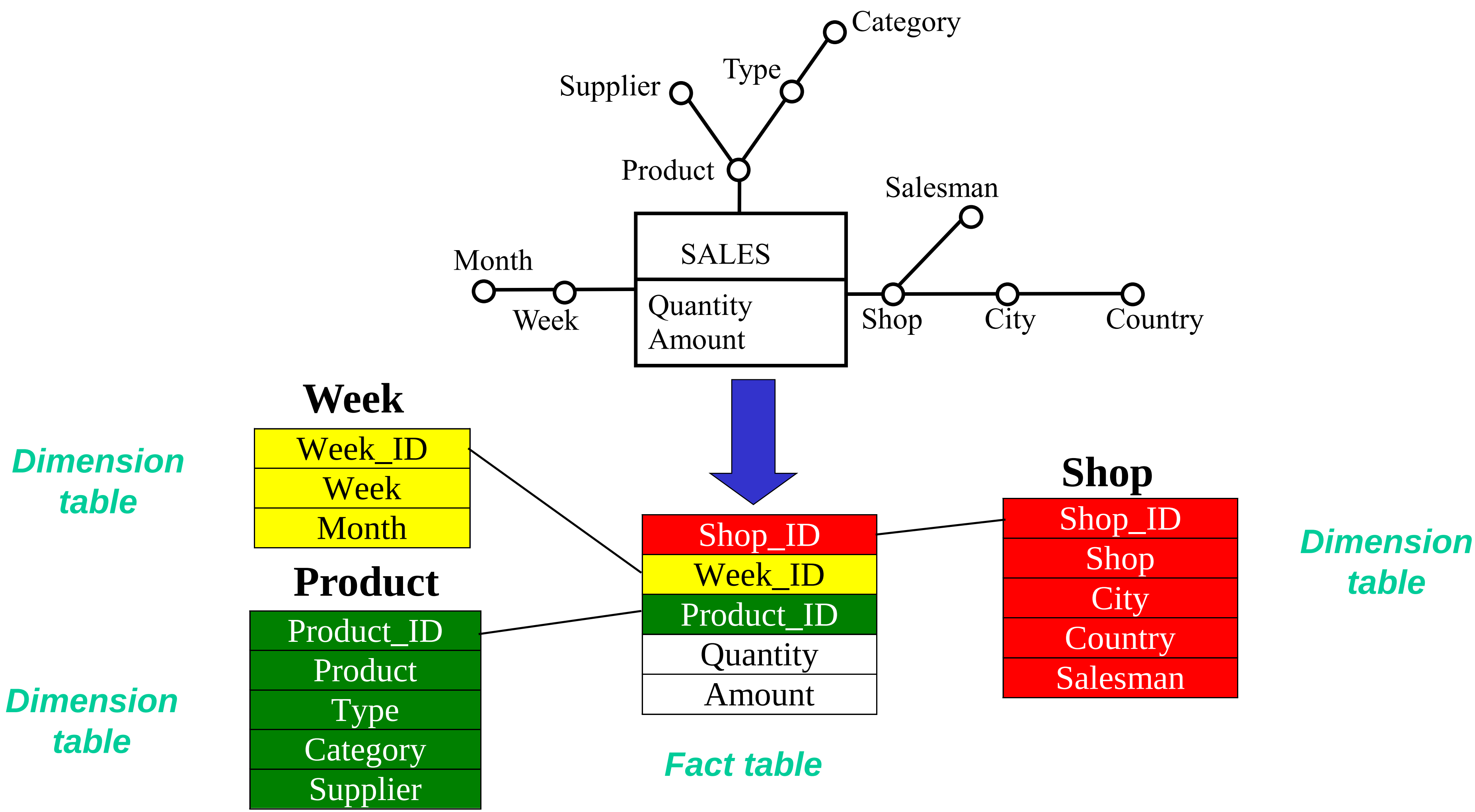
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Star schema



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

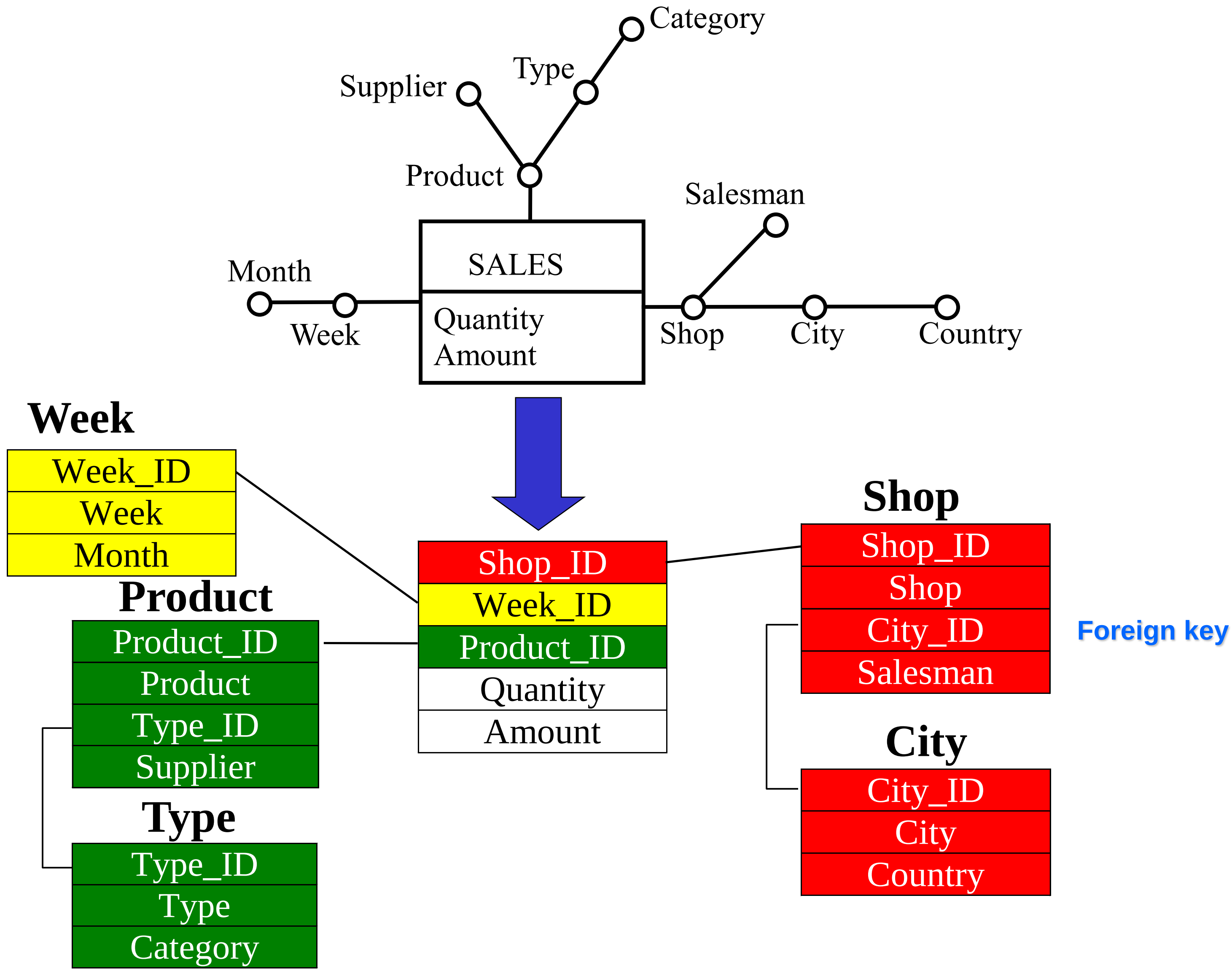
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Snowflake schema



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

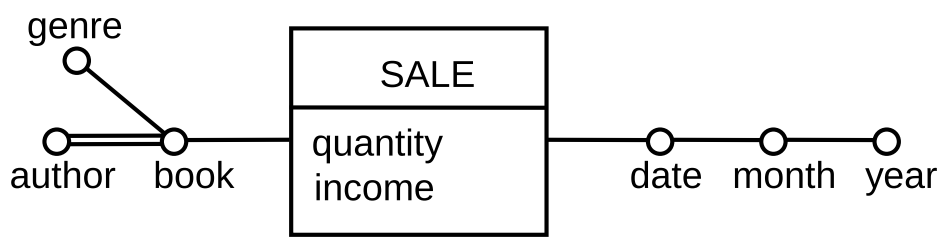
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

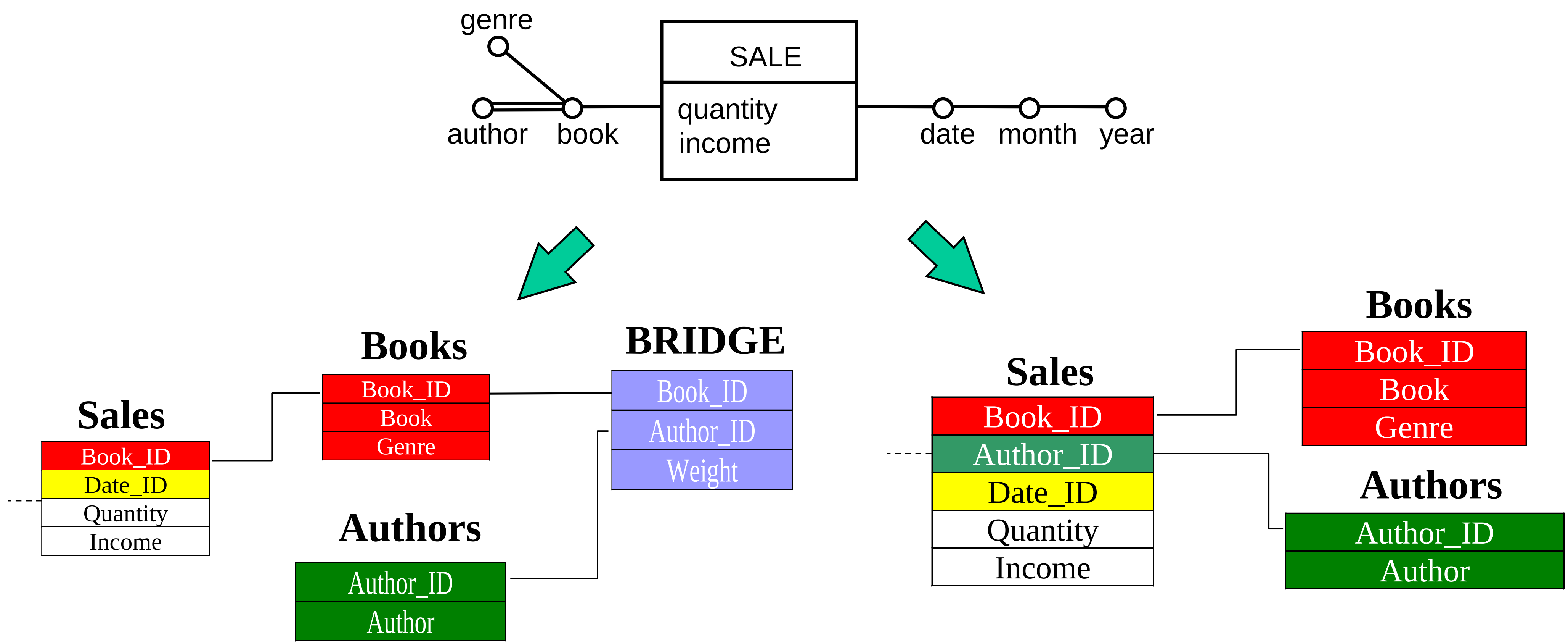
Multiple edges



- 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

Elena Baralis
Politecnico di Torino

Multiple edges



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

Elena Baralis
Politecnico di Torino

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
```


Elena Baralis
Politecnico di Torino

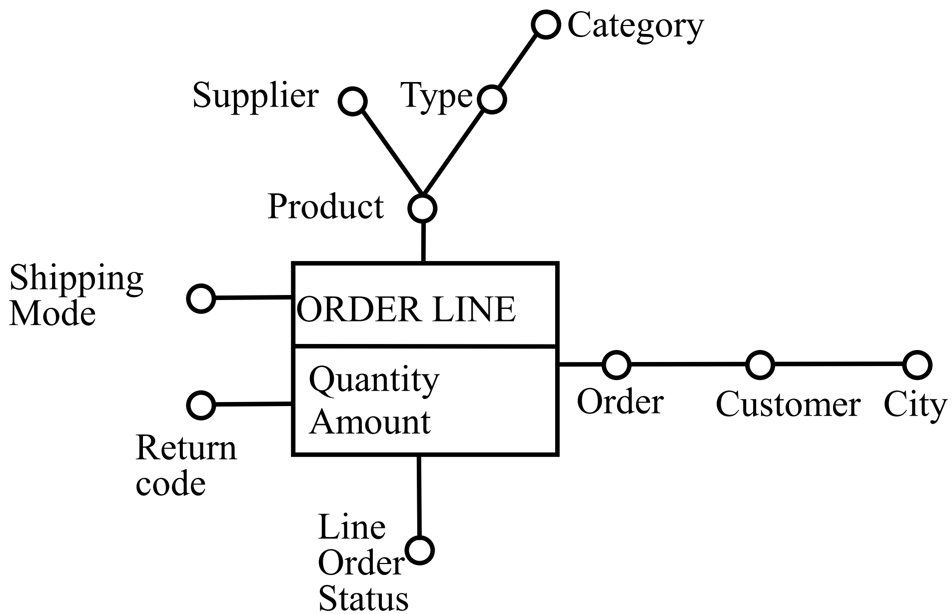
Multiple edges

- Comparison
  - weight is explicited 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

Elena Baralis
Politecnico di Torino

Degenerate dimensions

- Dimensions with a single attribute



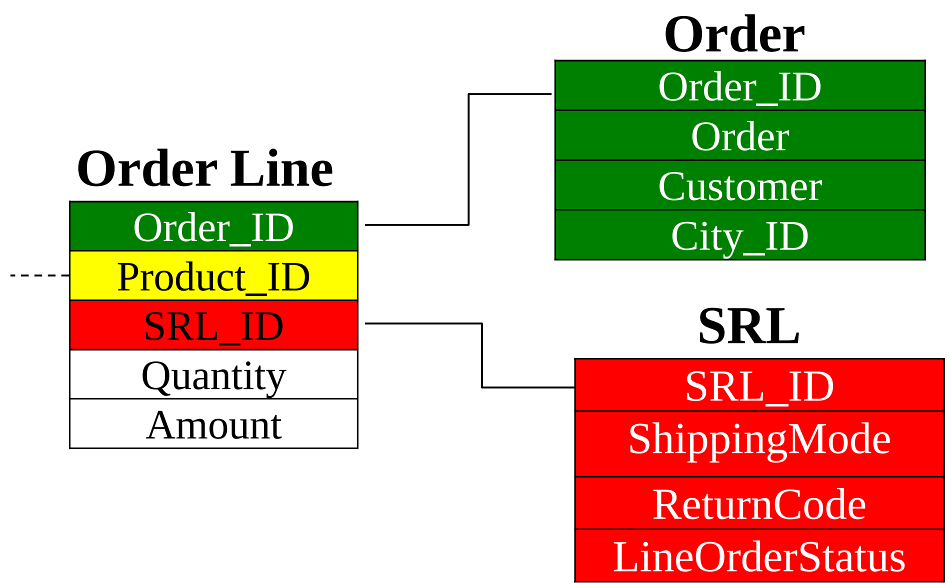
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Junk dimension



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

Elena Baralis
Politecnico di Torino

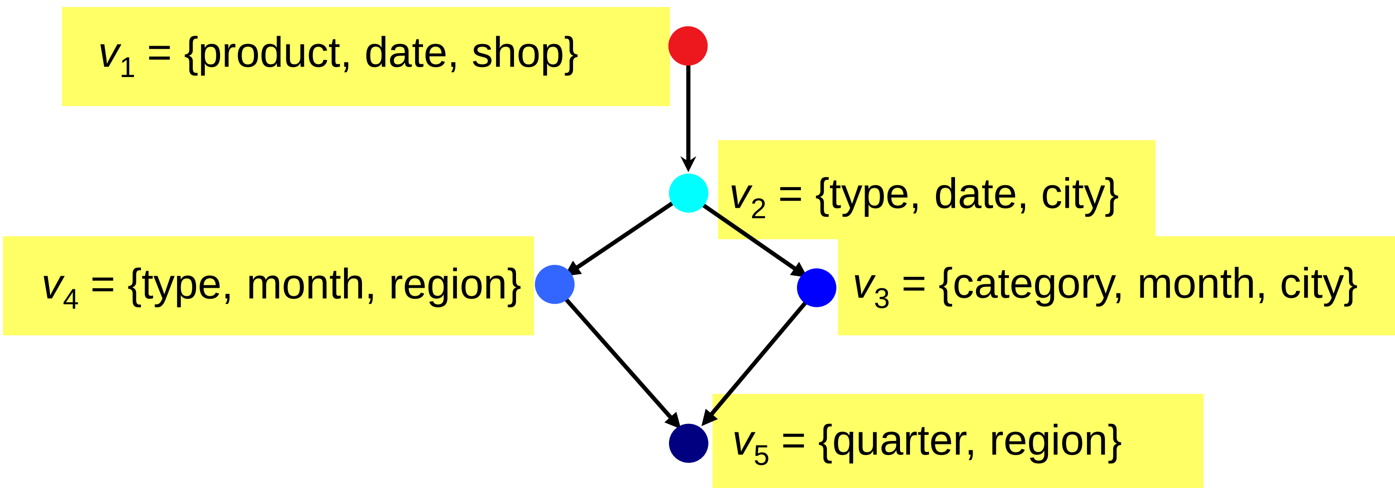
Materialized views

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



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

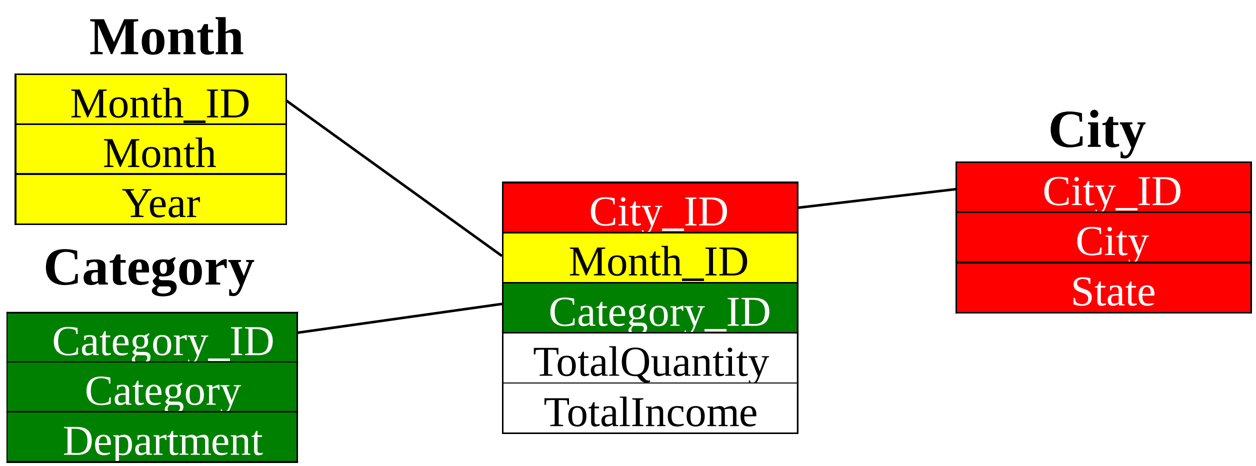
Elena Baralis
Politecnico di Torino

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
```

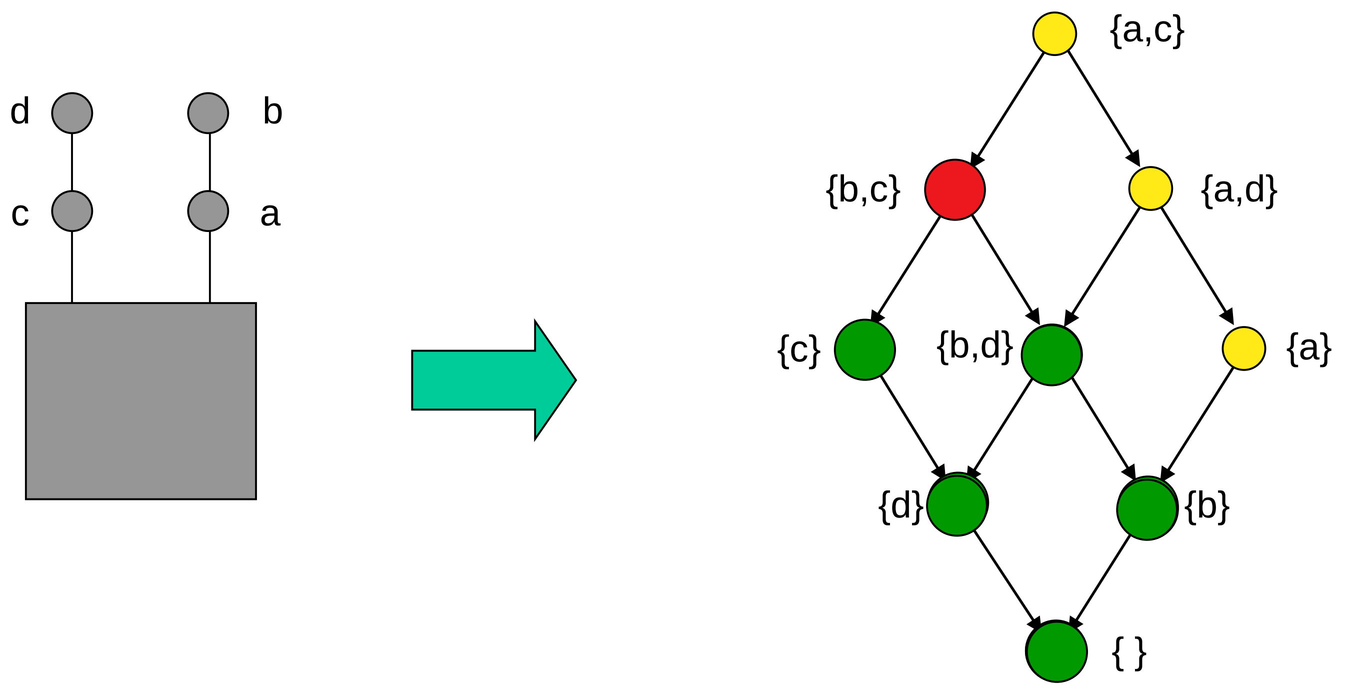
  - Aggregation (SUM) on `Quantity`, `Income` measures
  - Reduction of detail in dimensions



Elena Baralis
Politecnico di Torino

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

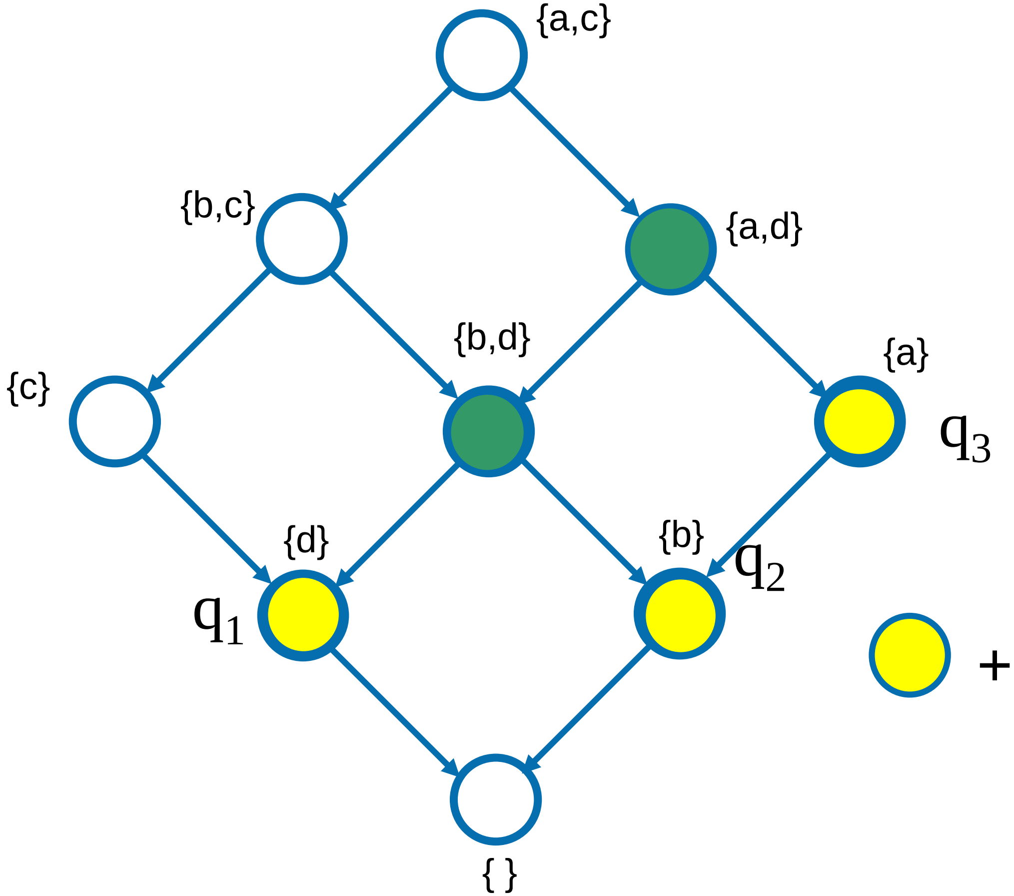
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Materialized view selection



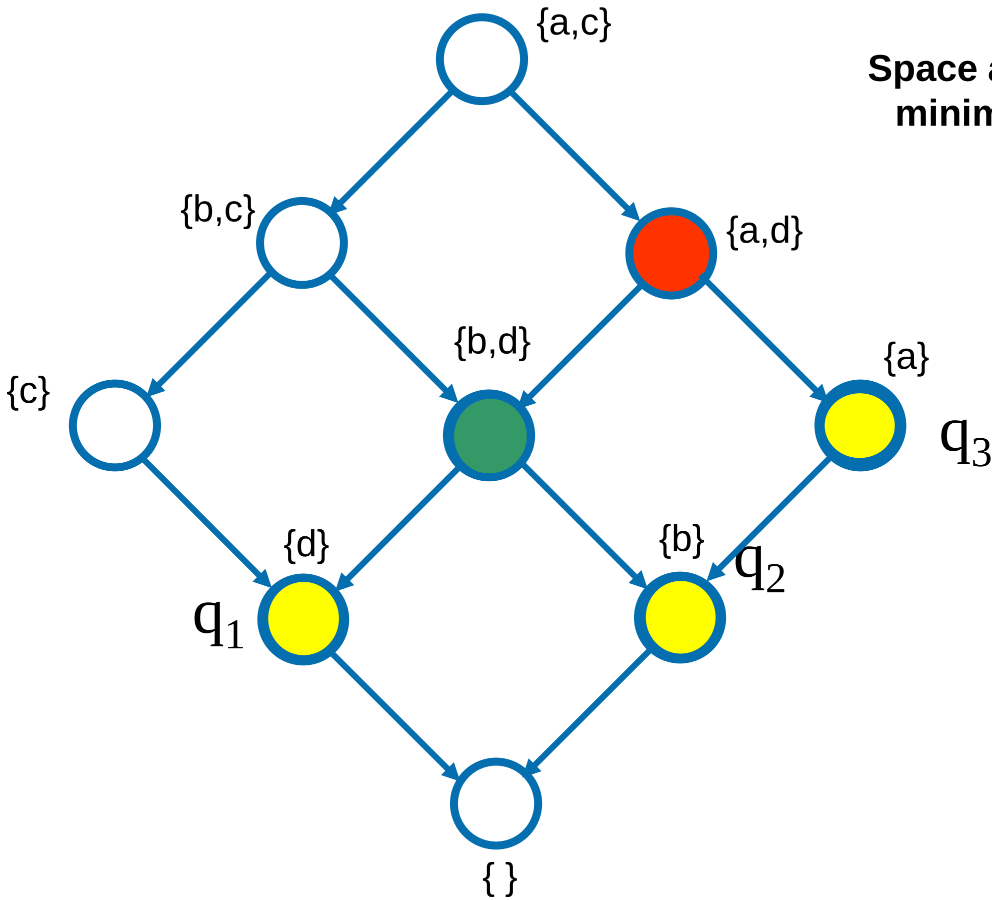
Multidimensional lattice

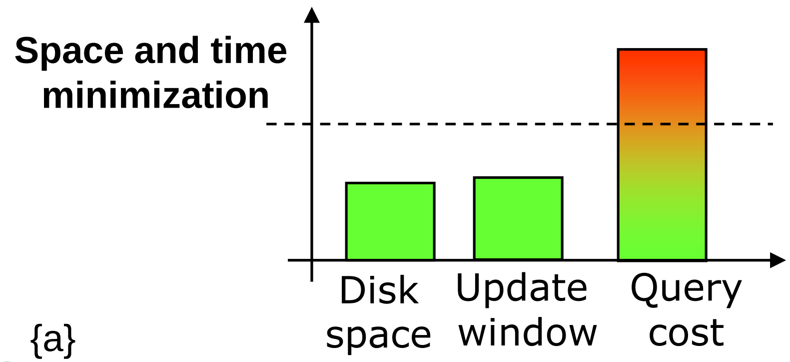
+ = ***candidate views***, **possibly useful to increase workload query performance**

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

Elena Baralis
Politecnico di Torino

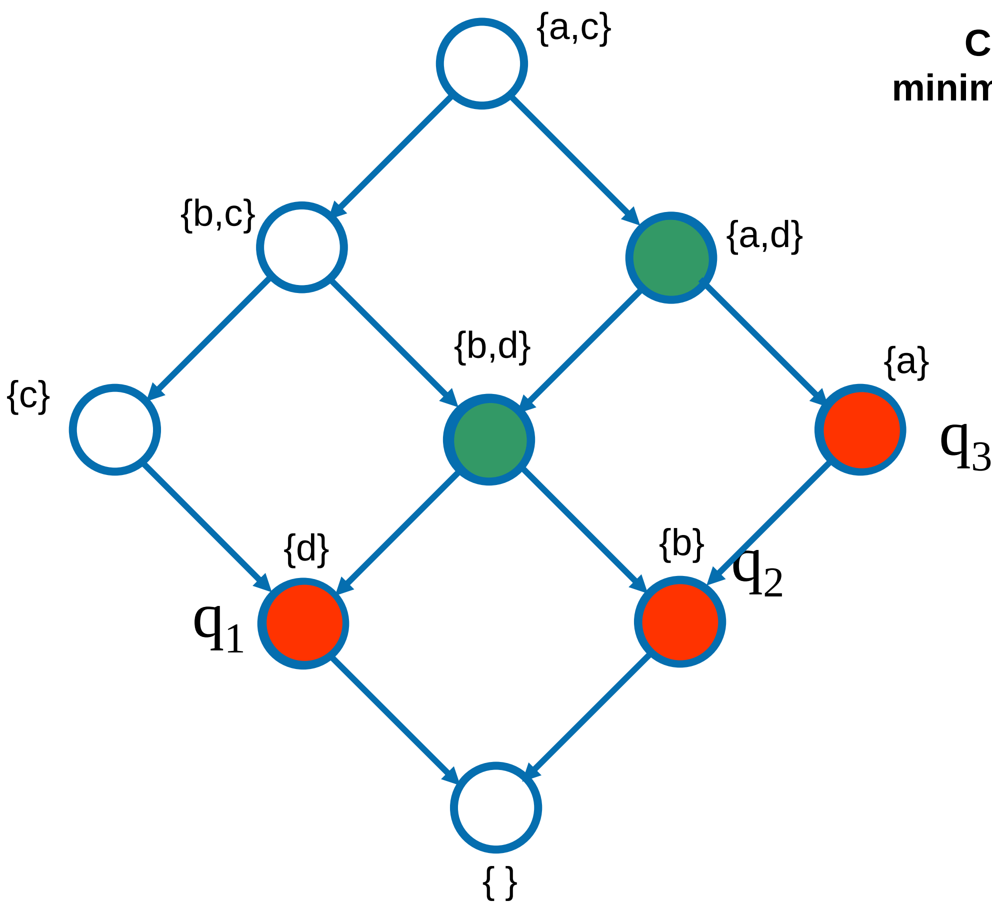
Materialized view selection

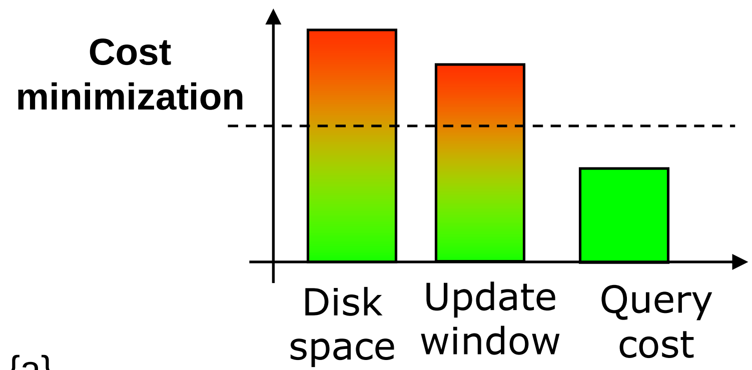




From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Materialized view selection

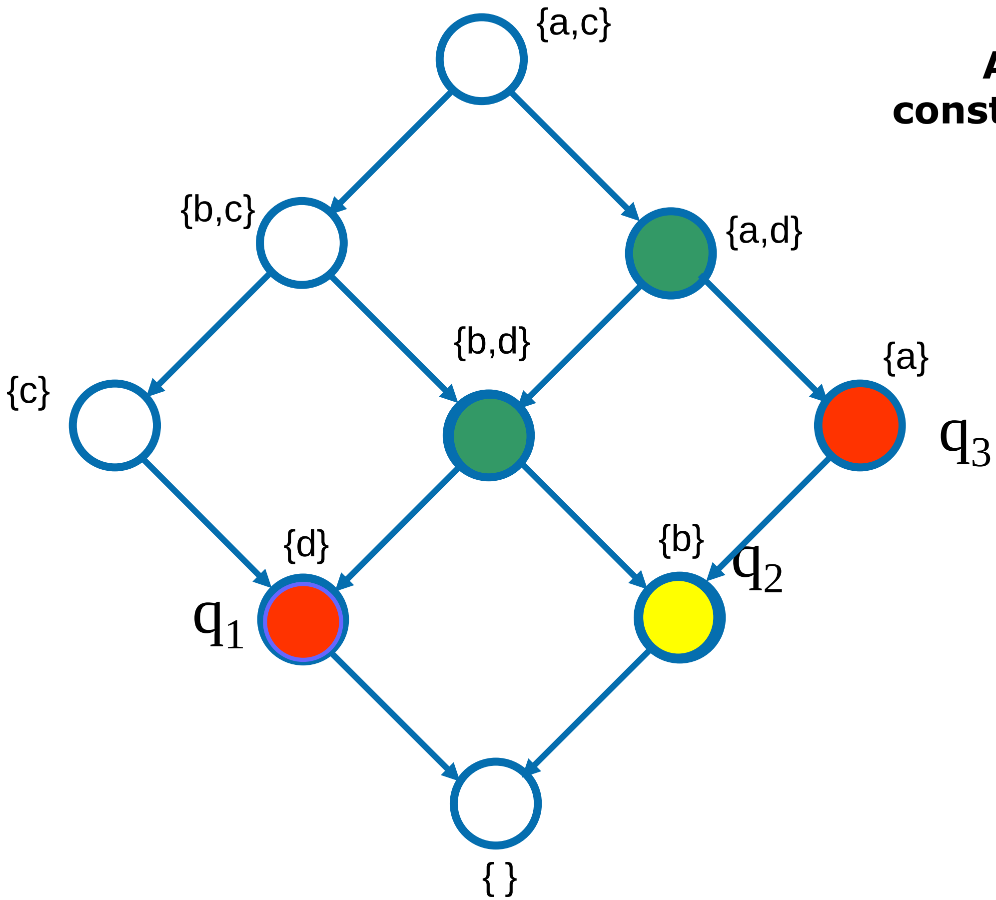


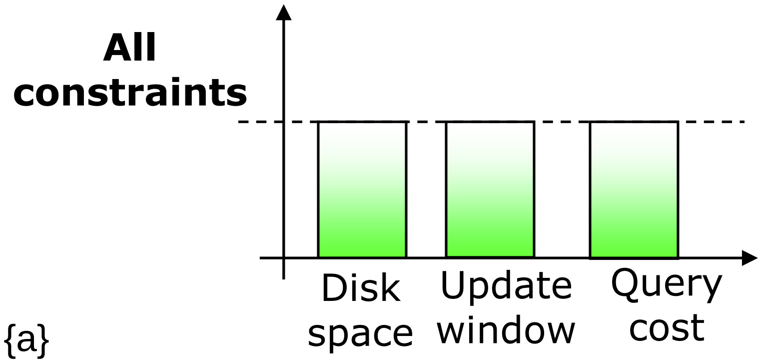


From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Elena Baralis
Politecnico di Torino

Materialized view selection





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

Elena Baralis
Politecnico di Torino

Physical design

Elena Baralis

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^+-tree index not appropriate for
      - attributes with low cardinality domains
      - queries with low selectivity
  - materialized views
    - query optimizer should be able to exploit them

Elena Baralis
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

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

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

Elena Baralis
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

Elena Baralis
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

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

Elena Baralis
Politecnico di Torino

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*	*Appilcation 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

Elena Baralis
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

Elena Baralis
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

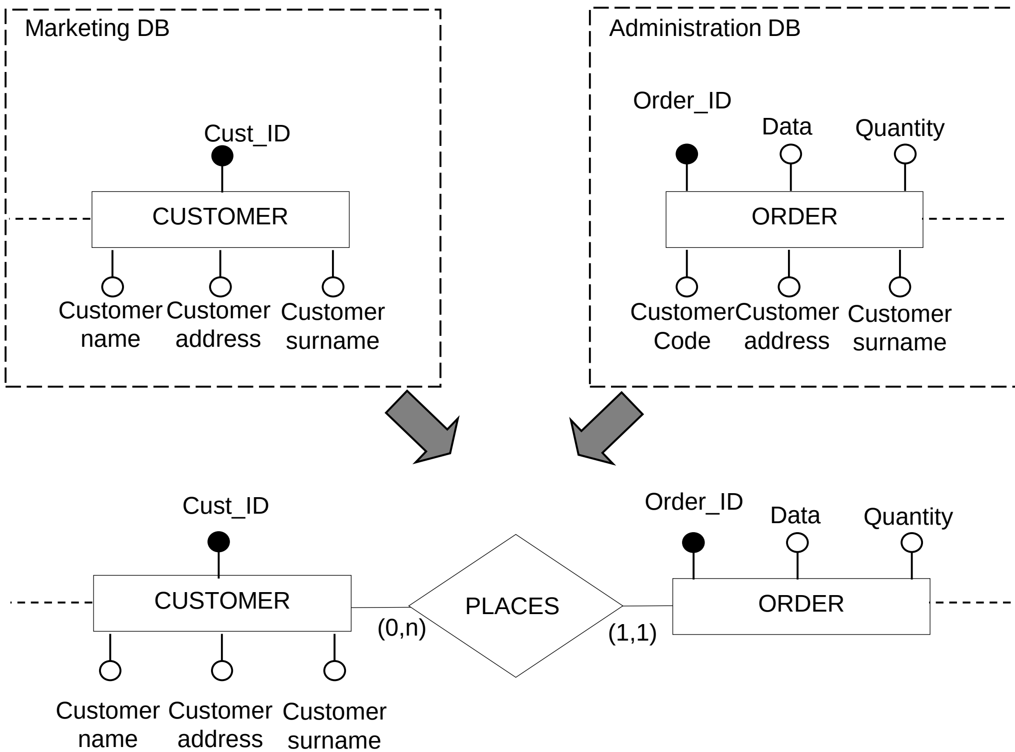
Elena Baralis
Politecnico di Torino

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

Elena Baralis
Politecnico di Torino

Approximate join

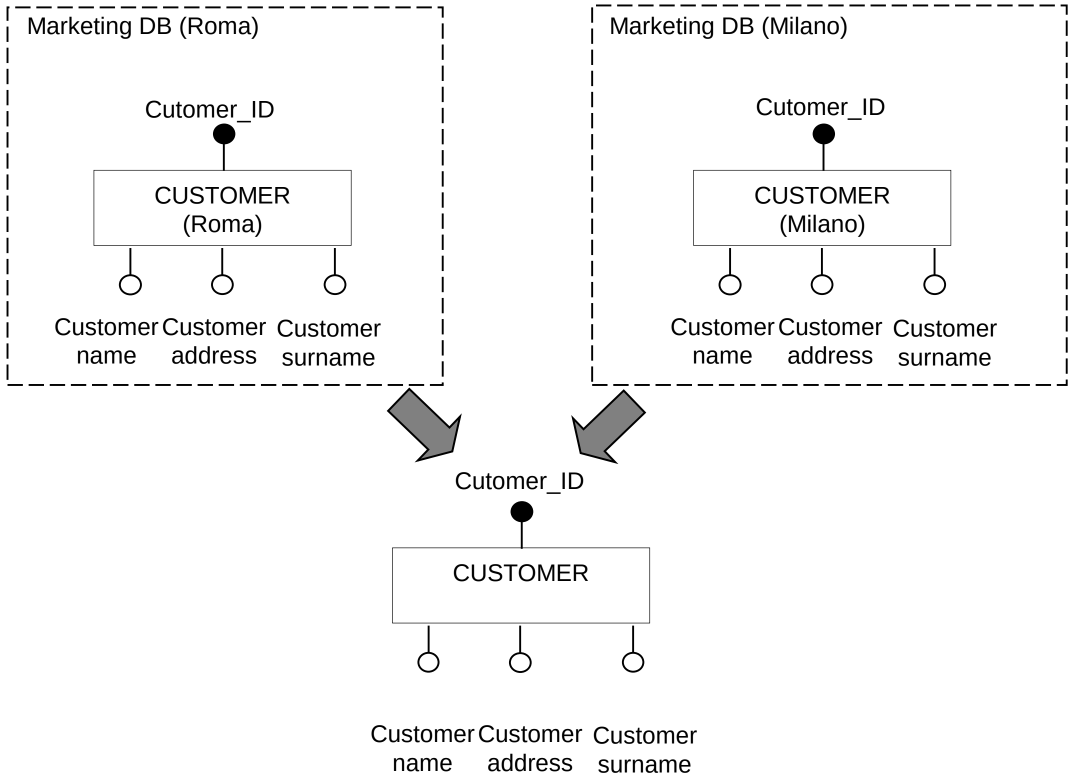


- The join operation should be executed based on common fields, not representing the customer identifier

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

Elena Baralis
Politecnico di Torino

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

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

Transformation

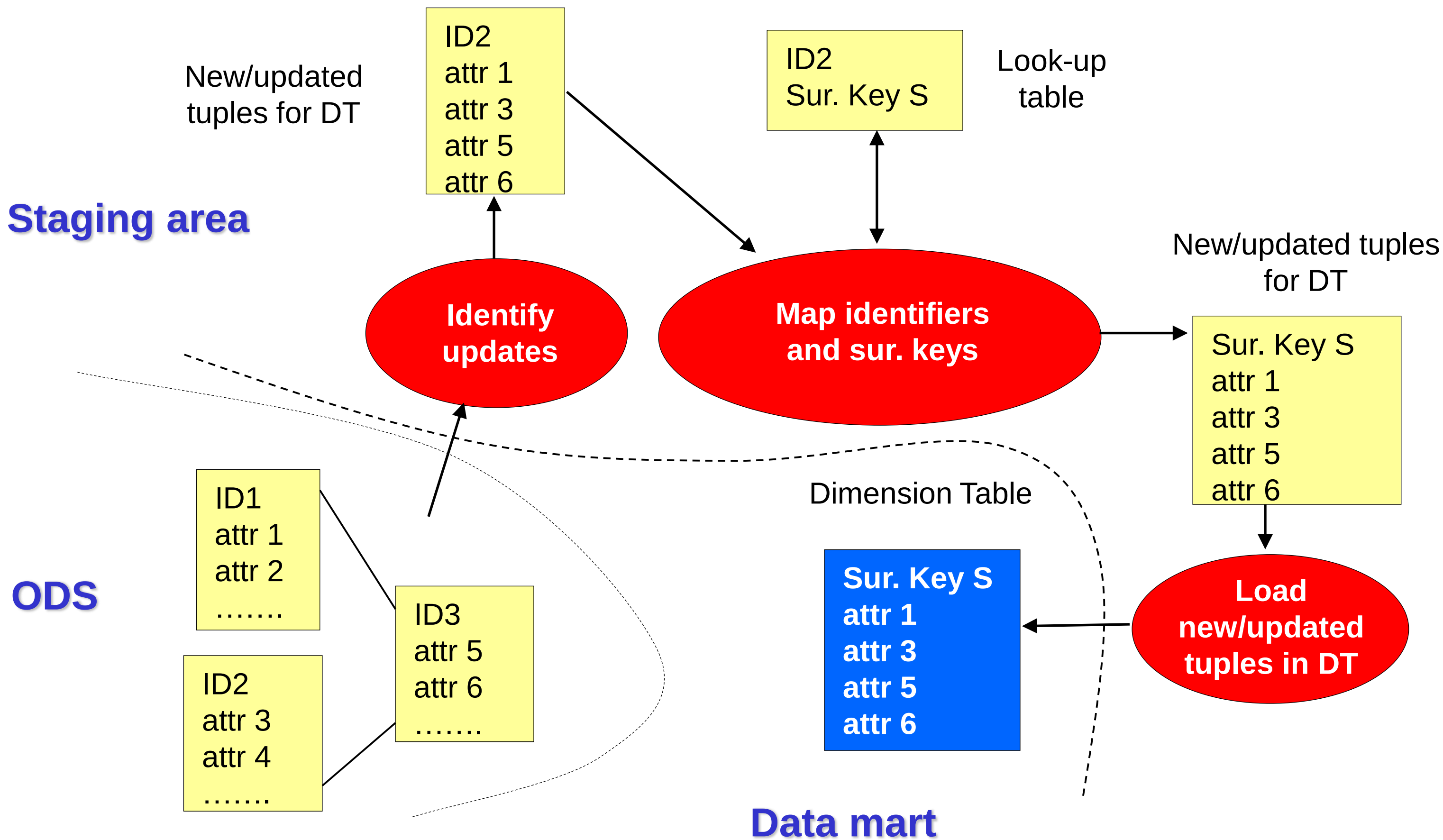
- Data conversion from operational format to data warehouse format
  - requires data integration
- A uniform operational data representation (reconciled schema) is needed
- Two steps
  - from operational sources to reconciled data in the staging area
    - conversion and normalization
    - matching
    - (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

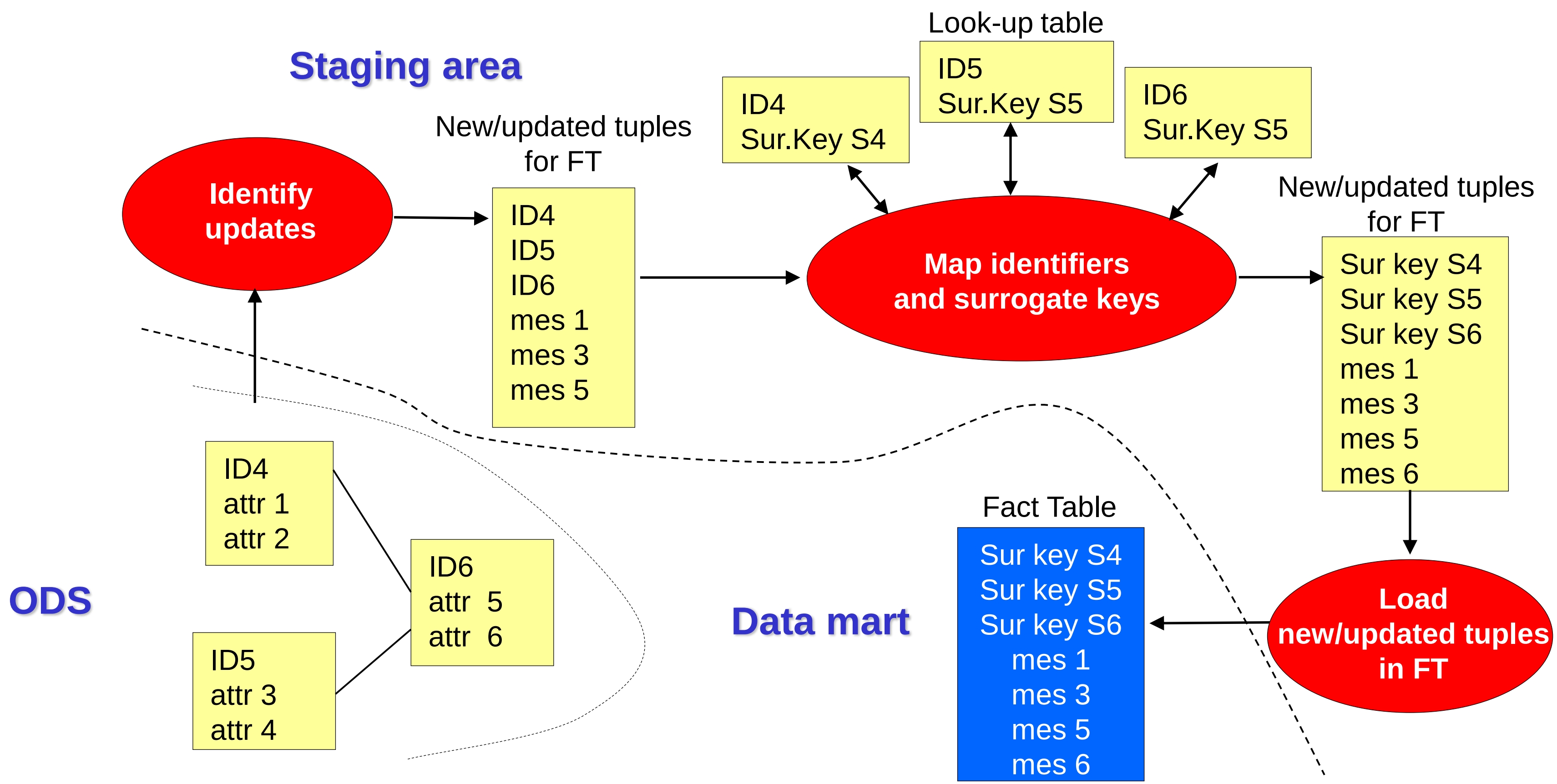
Elena Baralis
Politecnico di Torino

Dimension table loading



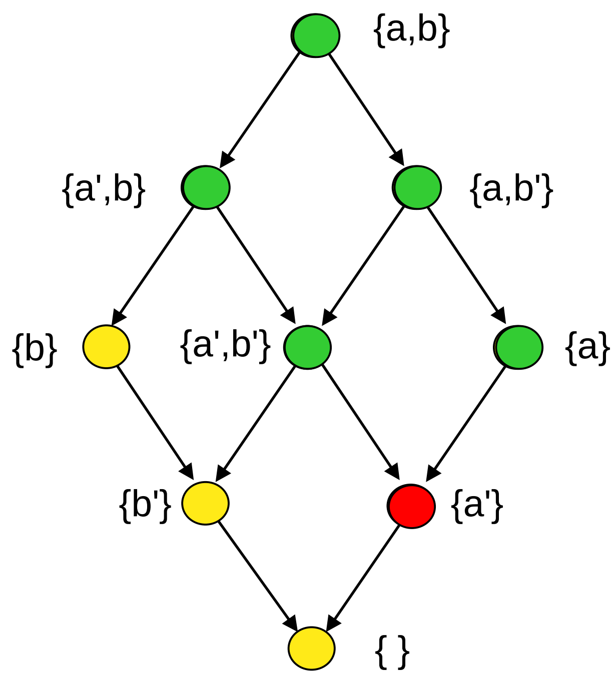
From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Fact table loading



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

Materialized view loading



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

Elena Baralis
Politecnico di Torino