Logical design

Elena Baralis
Politecnico di Torino

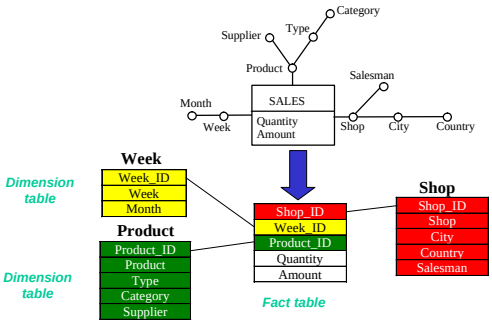
Logical design

- 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

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

Star schema



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

Star schema

Shop_ID	Shop	City	Country	Salesman
1	N1	RM	I	R1
2	N2	RM	I	R1
3	N3	MI	I	R2
4	N4	MI	I	R2

Dimension Table

Shop_ID	Week_ID	Product_ID	Quantity	Amount
1	1	1	100	100
1	2	1	150	150
3	3	4	350	350
4	4	4	200	200

Fact Table

Week_ID	Week	Month
1	Jan1	Jan.
2	Jan2	Jan.
3	Feb1	Feb.
4	Feb2	Feb.

Dimension Table

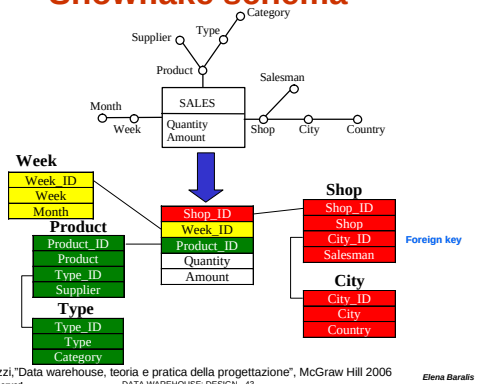
Product_ID	Product	Type	Category	Supplier
1	P1	A	X	F1
2	P2	A	X	F1
3	P3	B	X	F2
4	P4	B	X	F2

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

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



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

Snowflake schema

Type_ID	Type	Category
1	A	X
2	B	X

Product_ID	Product	Supplier	Type_ID
1	P1	F1	1
2	P2	F1	1
3	P3	F2	2
4	P4	F2	2

Week_ID	Week	Month
1	Jan1	Jan.
2	Jan2	Jan.
3	Feb1	Feb.
4	Feb2	Feb.

Shop_ID	Week_ID	Product_ID	Quantity	Amount
1	1	1	100	100
1	2	1	150	150
3	3	4	350	350
4	4	4	200	200

Shop_ID	Shop	City_ID	Salesman
1	N1	1	R1
2	N2	1	R1
3	N3	2	R2
4	N4	2	R2

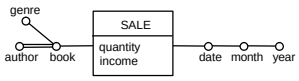
City_ID	City	Country
1	RM	I
2	MI	I

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

Star or snowflake?

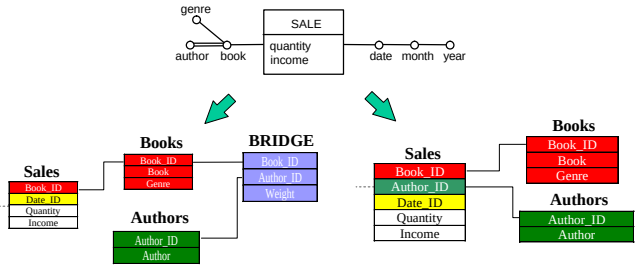
- The snowflake schema is usually not recommended
  - storage space decrease is rarely beneficial
    - most storage space is consumed by the fact table (difference with dimensions is several orders of magnitude)
  - cost of join execution may be significant
- The snowflake schema may be useful
  - when 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

Multiple edges



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

Multiple edges

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

```
SUM(Income*weight)
group by ID_author
```

  - Impact query: do not consider the weight of the multiple edge
    - example: book copies sold for each author
    - by using bridge table:

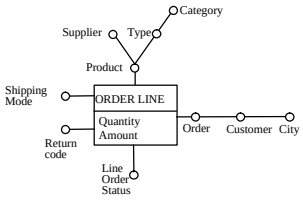
```
SUM(Quantity)
group by ID_author
```

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 time for push down
    - less joins
    - higher cardinality for the fact table

Degenerate dimensions

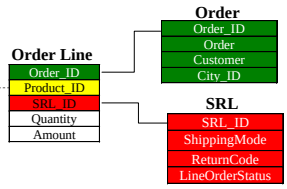
- Dimensions with a single attribute



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

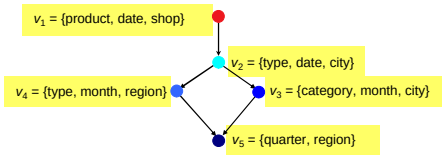
Junk dimension



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

Materialized views

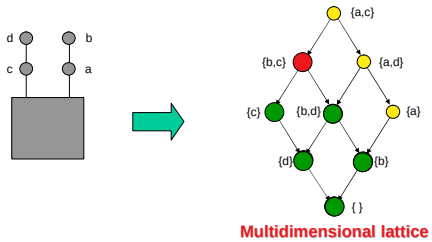
- Precomputed summaries for the fact table
  - explicitly stored in the data warehouse
  - provide a performance increase for aggregate queries



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

Materialized views

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



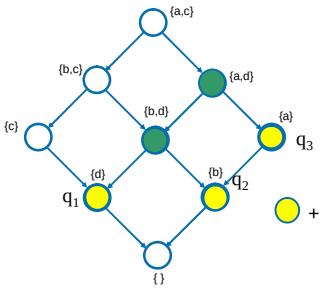
Multidimensional lattice

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

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

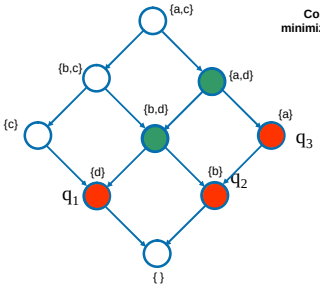


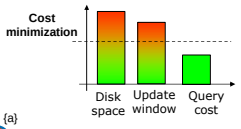
Multidimensional lattice

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

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

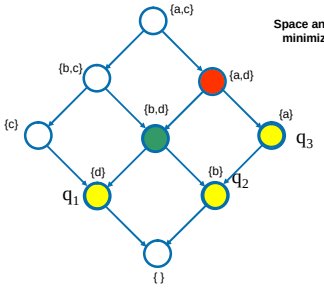
Materialized view selection

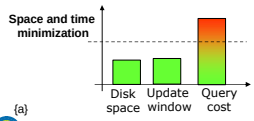




From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

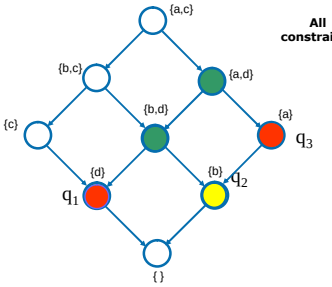
Materialized view selection

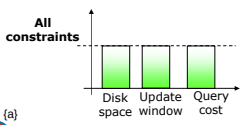




From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Materialized view selection





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