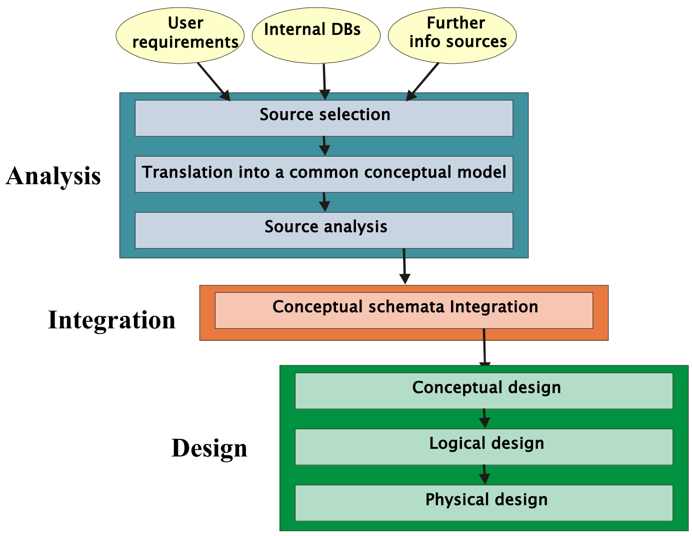
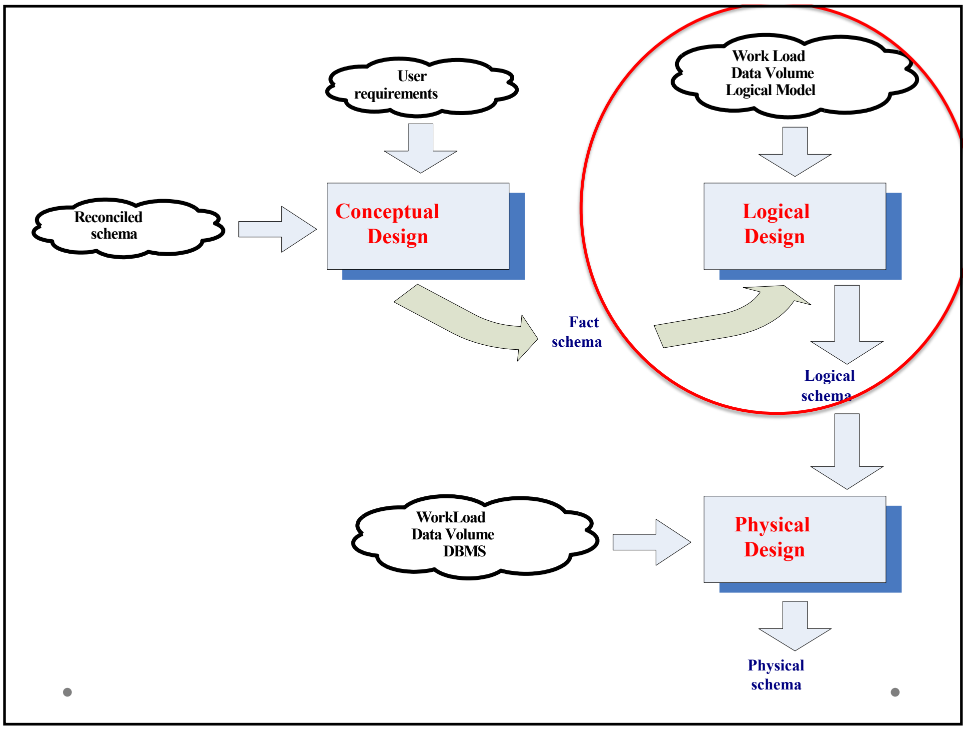
POLITECNICO MILANO 1863

Data Warehouse design

Cinzia Cappiello
A.A. 2023-2024

Data Warehouse Design





Logical Models

Data Mart logical models

MOLAP stands for Multidimensional OLAP. In MOLAP cubes the data aggregations and a copy of the fact data are stored (materialized) in a multidimensional structure on the computer. It is best when extra storage space is available on the server and the best query performance is desired. MOLAP local cubes contain all the necessary data for calculating aggregates and can be used offline. MOLAP cubes provide the fastest query response time and performance but require additional storage space for the extra copy of data from the fact table.
→ NOTE: REQUIRES ADDITIONAL INVESTMENTS!!!

ROLAP stands for Relational OLAP. ROLAP uses the relational data model to represent multidimensional data. In ROLAP cubes a copy of data from the fact table is not necessarily made, and the data aggregates are stored in tables, separately or in the source relational database. A ROLAP cube is best when there is limited space on the server and query performance is not very important. ROLAP local cubes contain the dimensions and cube definitions but normally aggregates are computed when needed. ROLAP cubes require less storage space than MOLAP and HOLAP cubes.

HOLAP stands for Hybrid OLAP. A HOLAP cube has a combination of the ROLAP and MOLAP cube characteristics. It does not necessarily create a copy of the source data; however, data aggregations are stored in a multidimensional structure on the server. HOLAP cubes are best when storage space is limited but faster query responses are needed.

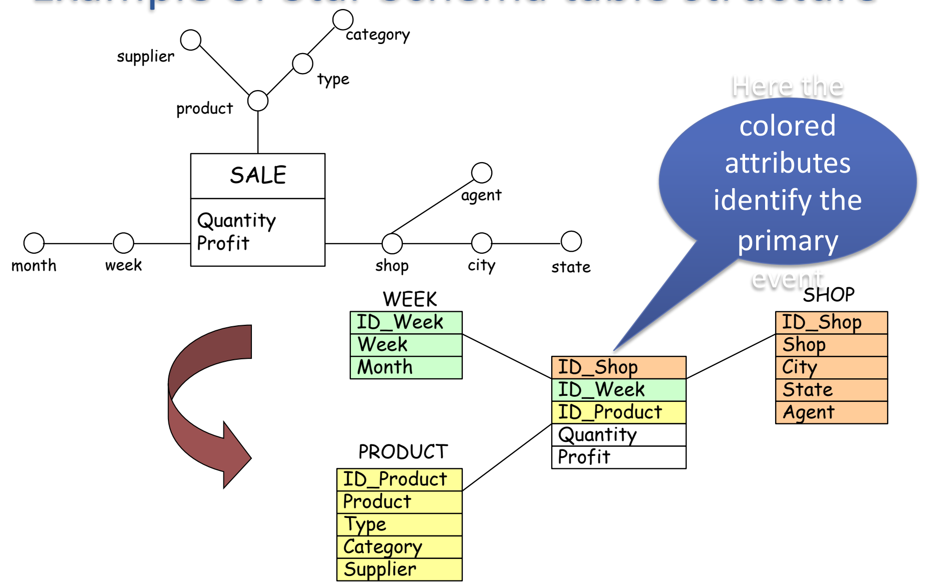
5

ROLAP

- It is based on the Star Schema
- A star schema is :
  - A set of relations DT1, DT2, …DTn - dimension tables - each corresponding to a dimension.
  - Each DTi is characterized by a primary key di and by a set of attributes describing the analysis dimensions with different aggregation levels
  - A relation FT, fact table, that imports the primary keys of dimensions tables. The primary key of FT is d1 d2 … dn ; FT contains also an attribute for each measure

6

Example of Star Schema table structure



7

It is possible to define different variants of the star schema to manage aggregate data, e.g. in a unique fact table

1° row represents sale values for the single shop, 2° row represents aggregate values for Roma, 3° row represents aggregate values for Lazio, etc…

SALE

Shop_key	Date_key	Prod_key	qty	profit	…
1	1	1	170	85	…
2	1	1	300	150	…
3	1	1	1700	850	…
…	…	…	…	…	…

SHOP

Shop_key	shop	city	region	…
1	COOP1	Bologna	E.R.	…
2	-	Roma	Lazio	…
3	-	-	Lazio	…
…	…	…	…	…

8

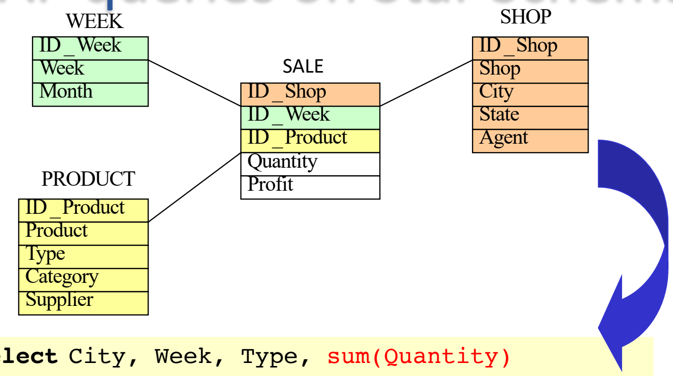
Star schema: considerations

- Dimension table keys are <u>surrogates (i.e. generated ids)</u>, for space efficiency reasons
- Dimension tables are de-normalized, i.e. they contain redundancy: note that

  product → type→ category

  means that for each different product all the info related to type is repeated, and the same for the category
- De-normalization introduces redundancy, but fewer joins to do
- The fact table contains information expressed at different aggregation levels

OLAP queries on Star Schema

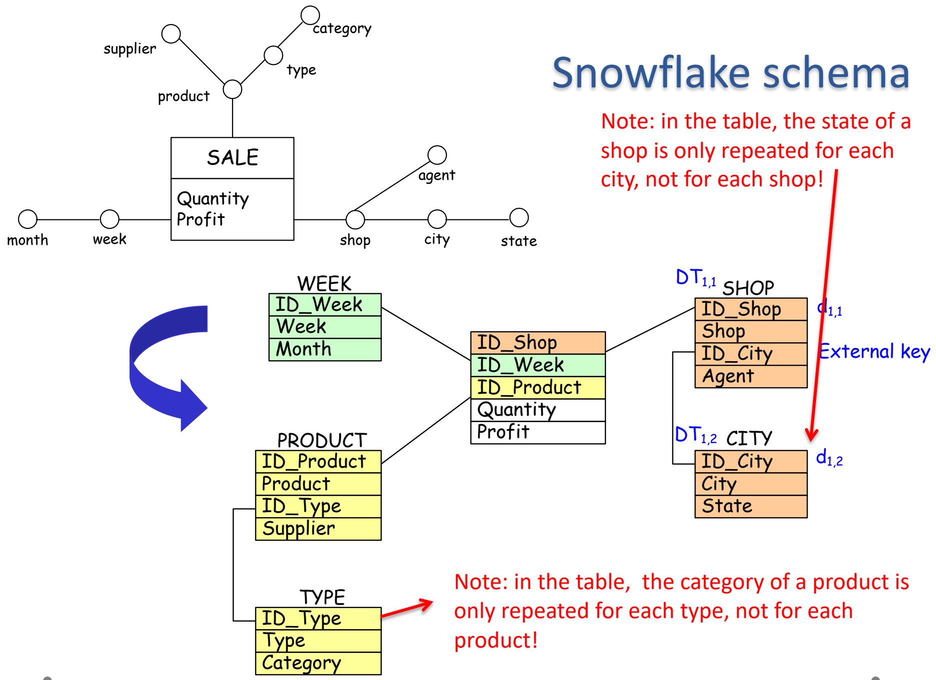



```
select City, Week, Type, sum(Quantity)
from   Week, Shop, Product, Sale
where  Week.ID_Week=Sale.ID_Week and
       Shop.ID_Shop=Sale.ID_Shop and
       Product.ID_Product=Sale.ID_Product and
       Product.Category = 'FoodStuff'
group by City,Week,Type
```

Snowflake schema

- The snowflake schema reduces the de-normalization of the dimensional tables DTi of a star schema
- Dimensions tables of a snowflake schema are composed by
  - A primary key di,j
  - A subset of DTi attributes that directly depend on di,j
  - Zero or more external keys that allow to obtain the entire information
- In a snowflake schema
  - Primary dimension tables: their keys are imported in the fact table
  - Secondary dimension tables

Snowflake schema



Note: in the table, the state of a shop is only repeated for each city, not for each shop!

Note: in the table, the category of a product is only repeated for each type, not for each product!

Snowflake schema: considerations

- Reduction of memory space
- New surrogate keys
- Advantages in the execution of queries related to attributes contained into fact and primary dimension tables

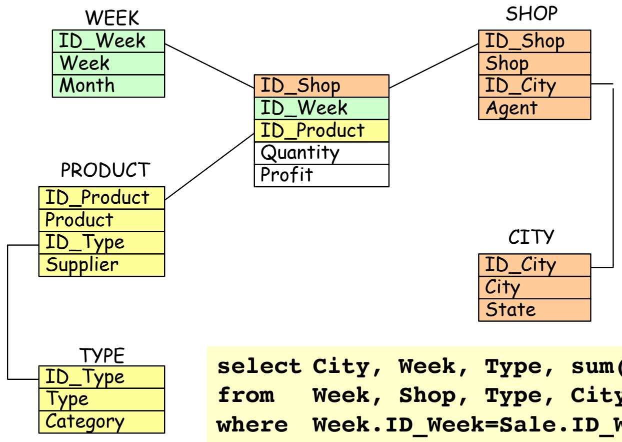
13

Normalization & Snowflake schema

- Attributes uniquely determined (transitively or not) by the snowflake attribute are placed in a new relation

14

OLAP queries on snowflake schema




```
select City, Week, Type, sum(Quantity)
from   Week, Shop, Type, City, Product, Sale
where  Week.ID_Week=Sale.ID_Week and
       Shop.ID_Shop=Sale.ID_Shop and
       Shop.ID_City = City.ID_City and
       Product.ID_Product=Sale.ID_Product and
       Product.ID_Type=Type.ID_Type and
       Product.Category = 'FoodStufs'
group by City,Week, Type
```


Views

- Aggregation allows to consider concise (summarized) information
- Aggregation computation is very expensive→ pre-computation (materialization)
- A view denotes a fact table containing aggregate data
- We can pre-compute views to make computation more efficient

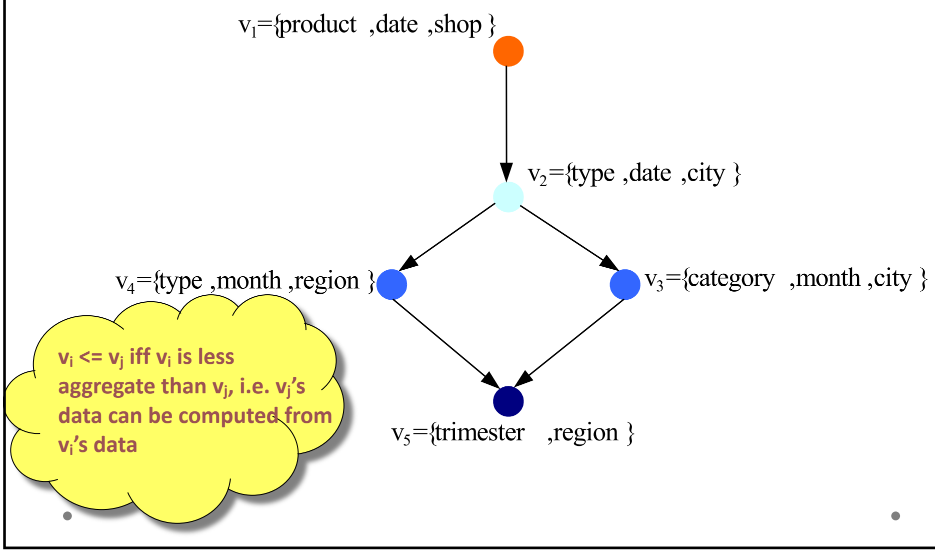
Views

- A view can be characterized by its aggregation level (pattern)
  - Primary views: correspond to the primary aggregation levels
  - Secondary views: correspond to secondary aggregation levels (secondary events)

17

Views

(MultiDimensional Lattice)



18

Partial aggregations

- Sometimes it is useful to introduce new measures in order to manage aggregations correctly
  - Derived measures: obtained by applying mathematical operators to two or more values of the same tuple

Partial aggregations

Profit=Quantity*Price

Type	Product	Quantity	Price	Profit
T1	P1	5	1,00	5,00
T1	P2	7	1,50	10,50
T2	P3	9	0,80	7,20

22,70
(total profits)

SUM

AVG

Type	Quantity	Price	Profit
T1	12	1,25	15,00
T2	9	0,80	7,20

22,20

We can't just sum up profits as before!!

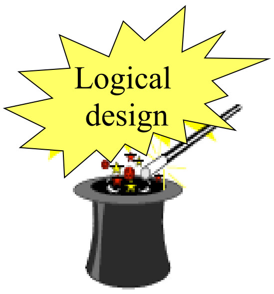
The correct solution consists in the aggregation of data on the primary

Logical design
Rolap

Logical modelling

- Sequence of steps that, starting from the conceptual schema, allow one to obtain the logical schema for a specific data mart

INPUT
Conceptual Schema
WorkLoad
Data Volume
System constraints



OUTPUT
Logical Schema

Workload

- In OLAP systems, workload is dynamic in nature and intrinsically extemporaneous
  - Users' interests change over time
  - Number of queries grows when users gain confidence in the system
  - OLAP should be able to answer any (unexpected) request
- During requirement collection phase, deduce it from:
  - Interviews with users
  - Standard reports

Workload

- Characterize OLAP operations:
  - Based on the required aggregation pattern
  - Based on the required measures
  - Based on the selection clauses
- At system run-time, workload can be desumed from the system log

Data volume

- Depends on:
  - Number of distinct values for each attribute
  - Attribute size
  - Number of events (primary and secondary) for each fact
- Determines:
  - Table dimension
  - Index dimension
  - Access time

Logical modelling: steps

- Choice of the logical schema (star/snowflake schema)
- Conceptual schema translation
- Choice of the materialized views
- Optimization

From fact schema to star schema

- Create a fact table containing measures and descriptive attributes directly connected to the fact
- For each hierarchy, create a dimension table containing all the attributes

35

Guidelines

- Descriptive attributes (e.g. color)
  - If it is connected to a dimensional attribute, it has to be included in the dimension table containing the attribute (see slide n. 14, snowflake example, agent)
  - If it is connected to a fact, it has to be directly included in the fact schema
- Optional attributes (e.g. diet)
  - Introduction of null values or ad-hoc values

36

Guidelines

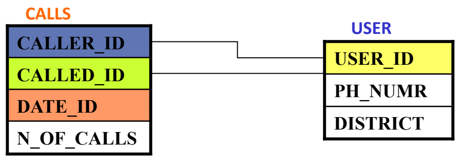
- Cross-dimensional attributes (e.g. VAT)
  - A cross-dimensional attribute b defines an N:M association between two or more dimensional attributes $a_1, a_2, \ldots, a_k$
  - It requires to create a new table including b and having as key the attributes $a_1, a_2, \ldots, a_k$

Guidelines

- Shared hierarchies and convergence
  - A shared hierarchy is a hierarchy which refers to different elements of the fact table (e.g. caller number, called number)
  - The dimension table should not be duplicated
  - Two different situations:
    - The two hierarchies contain the same attributes, but with different meanings (e.g. phone call → caller number, phone call → called number)
    - The two hierarchies contain the same attributes only for part of the hierarchy trees

Shared hierarchies and convergence

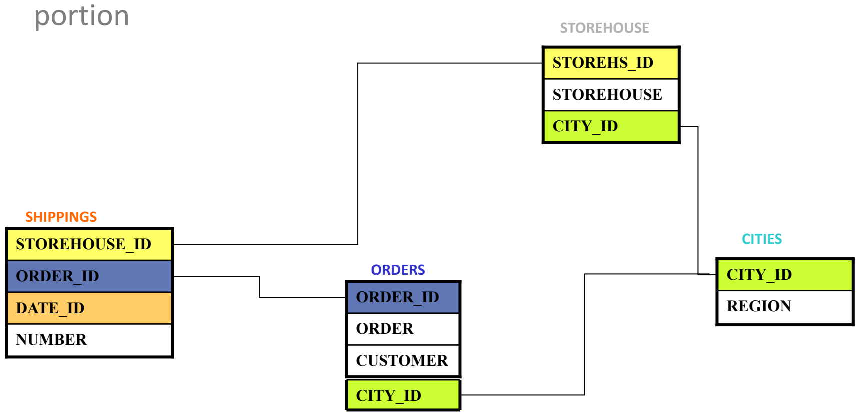
- The two hierarchies contain the same attributes, but with different meanings (e.g. phone call → caller number, phone call → called number)



39

Shared hierarchies and convergence

- The two hierarchies contain the same attributes only for part of the trees. Here we could also decide to replicate the shared portion

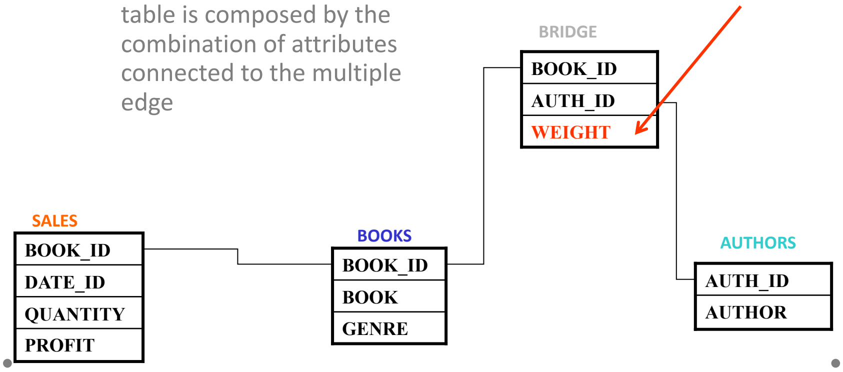


40

Guidelines

- Multiple edges
  - A bridge table models the multiple edge
    - the key of the bridge table is composed by the combination of attributes connected to the multiple edge

The weight of the edge is the contribution of each edge to the cumulative relationship



Guidelines

- Multiple edges: bridge table
  - Weighed queries take into account the weight of the edge

Query computing the profit for each author

```
SELECT AUTHORS.Author,SUM(SALES.Profit * BRIDGE.Weight)
FROM AUTHORS, BRIDGE, BOOKS, SALES
WHERE AUTHORS.Author_id=BRIDGE.Author_id
AND BRIDGE.Book_id=BOOKS.Book_id
AND BOOKS.Book_id=SALES.Book_id
GROUP BY AUTHORS.Author
```

Guidelines

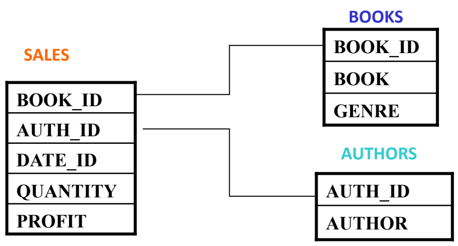
- Multiple edges: bridge table
  - Impact queries do not take into account the weight of the edge

Query computing the copies sold for each author

```
SELECT AUTHORS.Author, SUM(SALES.Quantity)
FROM AUTHORS, BRIDGE, BOOKS, SALES
WHERE AUTHORS.Author_id=BRIDGE.Author_id
AND BRIDGE.Book_id=BOOKS.Book_id
AND BOOKS.Book_id=SALES.Book_id
GROUP BY AUTHORS.Author
```


Alternative solution: keep the star model (only one level after the fact)

Multiple edges with a star schema:
add authors to the fact schema



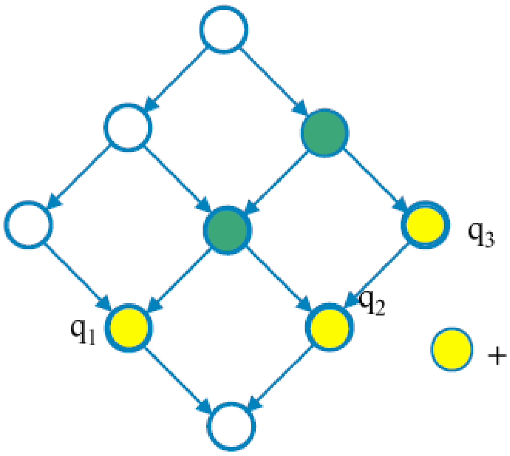
Here we don't need the weight because the fact table records quantity and profit per book and per author

Secondary-view precomputation

- The choice about views that have to be materialized takes into account contrasting requirements:
  - Cost functions' minimization
    - Workload cost
    - View maintenance cost
  - System constraints
    - Disk space
    - Time for data update
  - Users constraints
    - Max answer time
    - Data freshness

45

Materialized views (MD lattice)



= **exact views:** They solve exactly the queries

= **less aggregate views:** They solve more than one query

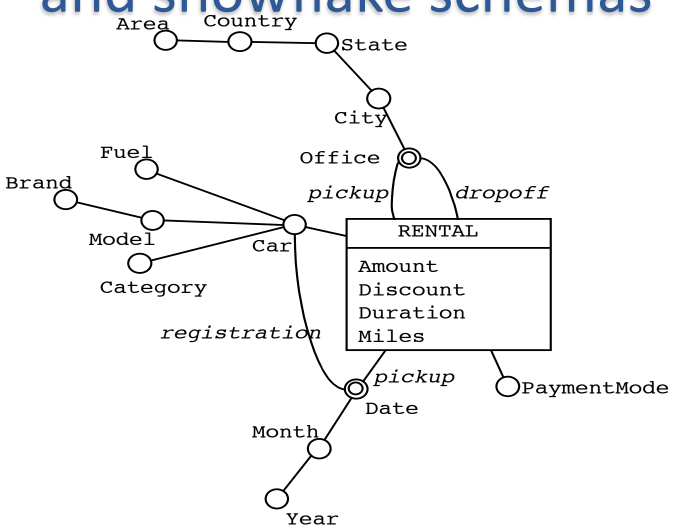
+ = **candidate views** They could reduce elaboration costs

46

Materialized Views

- It is useful to materialize a view when:
  - It directly solves a frequent query
  - It reduce the costs of some queries
- It is not useful to materialize a view when:
  - Its aggregation pattern is the same as another materialized view
  - Its materialization does not reduce the cost

Exercise: from the DFM to Star and snowflake schemas



Exercise: from the DFM to Star schema

References

- Stefano Rizzi : Data Warehouse Design: Modern Principles and Methodologies McGraw-Hill, 2009
- M. Golfarelli, S. Rizzi: Data Warehouse: teoria e pratica della progettazione McGraw-Hill, 2002.
- Matteo Golfarelli: Data Warehouse Life-Cycle and Design. Encyclopedia of Database Systems 2009: 658-664
- Stefano Rizzi: Business Intelligence. Encyclopedia of Database Systems 2009: 287-288
- Ralph Kimball: The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses John Wiley 1996.
- On the Internet: Oracle® Database Data Warehousing