Data Warehousing
Logical Database Design

Esteban Zimányi
ezimanyi@ulb.ac.be
Slides by Toon Calders

ULB
ECOLE
POLYTECHNIQUE
DE BRUXELLES

Outline

- Logical Database Design
  - Star schema
  - Snowflake schema

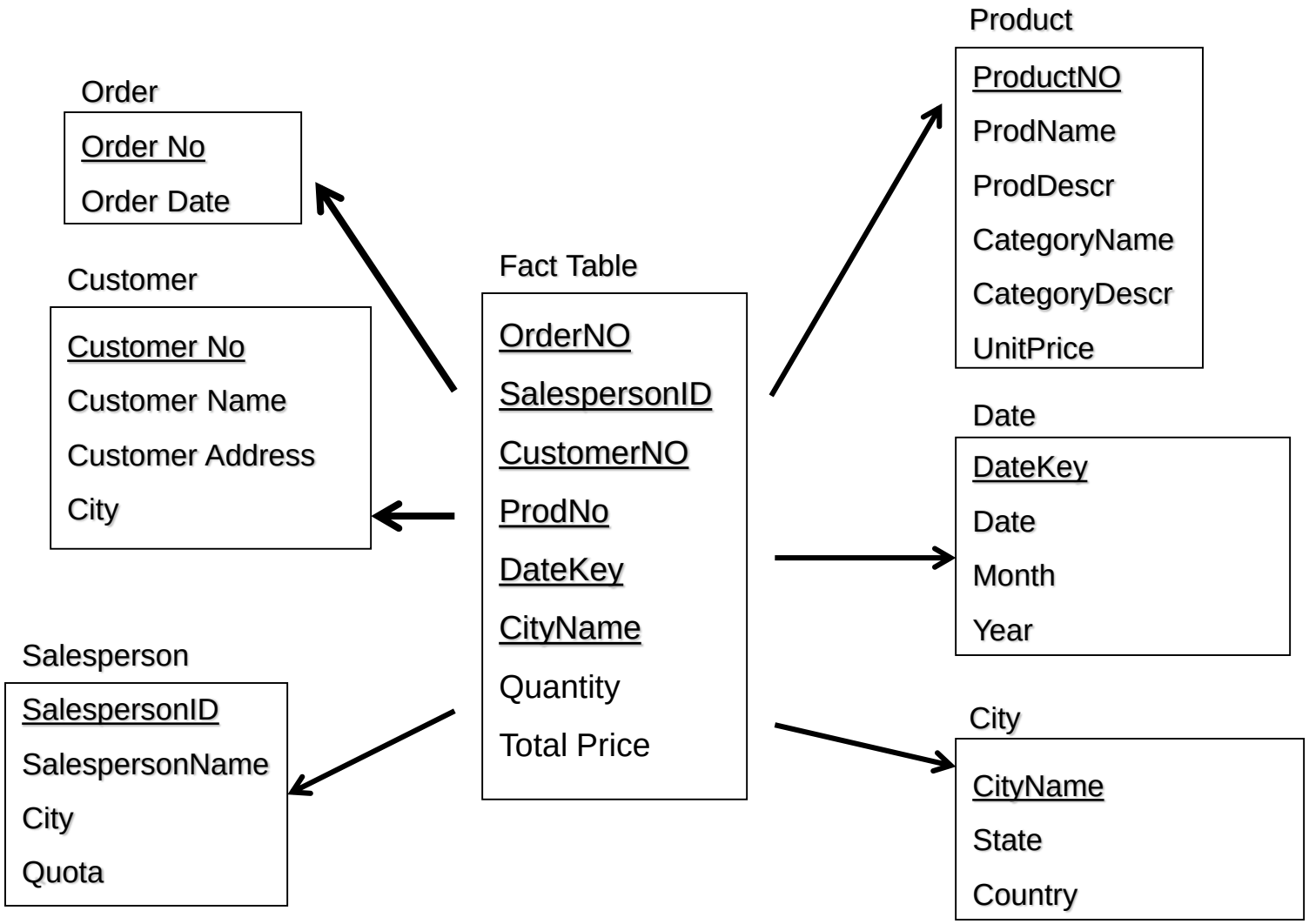
Chapter 8 of Golfarelli & Rizzi

ROLAP

- Relational OLAP
  - OLAP layer built on top of relational database
    - Facts, dimensions, hierarchies encoded as relations
  - Multi-dimensional data warehouse data structure

- Also:
  - MOLAP: Multi-dimensional OLAP = native support for datacubes
  - HOLAP: Hybrid form

Example of a Star Schema



Star Schema

- Dimension tables are not normalized
  - No consistency problems (given your ETL is fine)
  - Avoids the need for joins
  - Use surrogate key
- Dimensions such as Date are materialized
- Key for the fact table consists of the foreign keys to the dimension tables
  - Although some discussion here; case dependent

Why Surrogate Key

2000

Customer

CID	Name	Address
001	John	Dallas
002	Mary	Dallas
003	Pete	New York

Sales

CID	Product	Price
001	Gun	5$
002	Beef	20$
003	Lava lamp	150$

2010

Customer

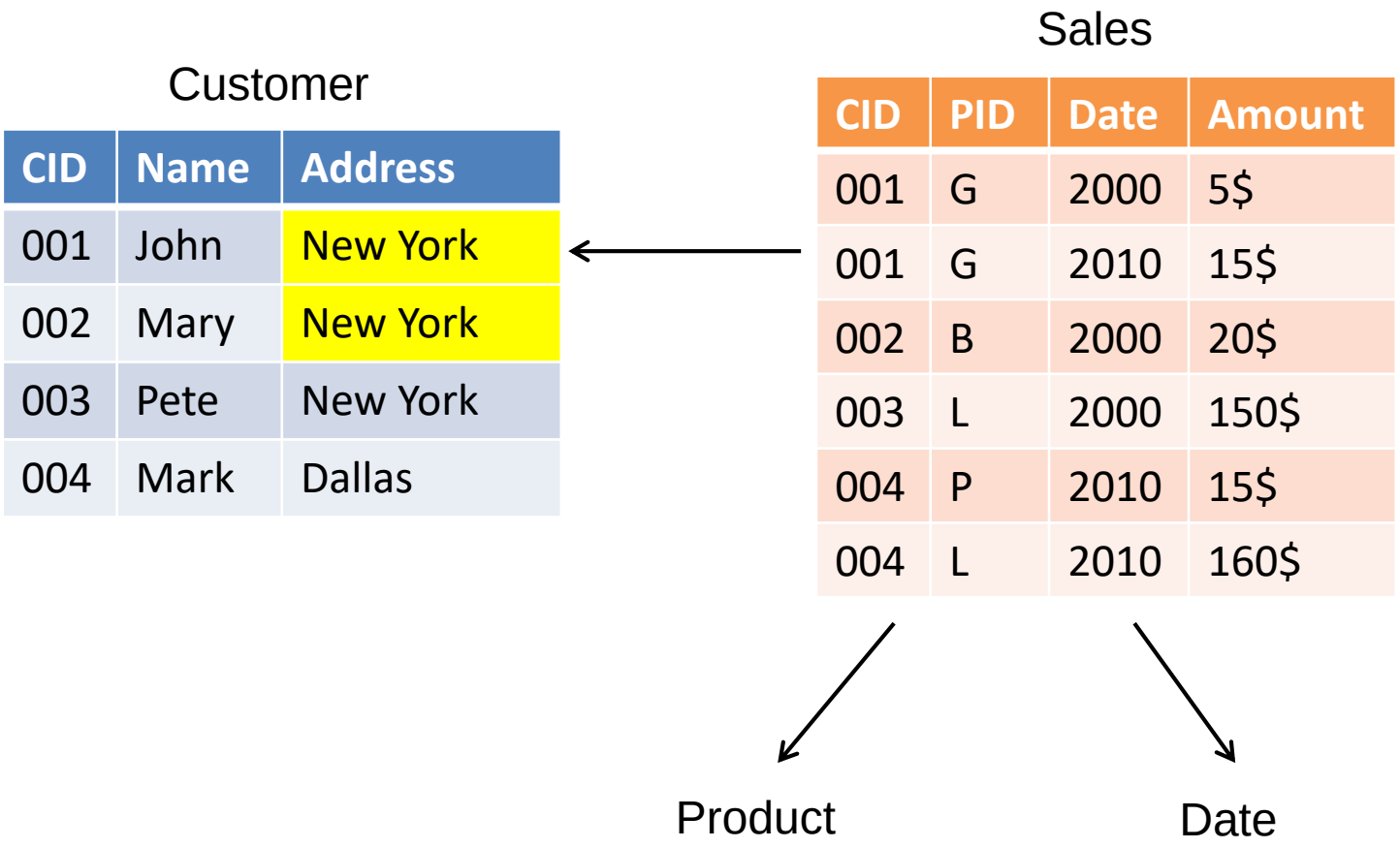
CID	Name	Address
001	John	New York
002	Mary	New York
004	Mark	Dallas

Sales

CID	Product	Price
001	Gun	15$
004	Pork	15$
004	Lava lamp	160$

Why Surrogate Key

- Star schema without surrogate key



Why Surrogate Key

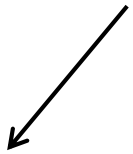
- Star schema with surrogate key

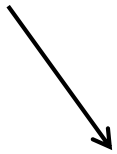
Customer

Key	OLTP key	Name	Address
S1	001	John	Dallas
S2	002	Mary	Dallas
S3	003	Pete	New York
S4	004	Mark	Dallas
S5	001	John	New York
S6	002	Mary	New York

Sales

Key	PID	Date	Amount
S1	G	2000	5$
S5	G	2010	15$
S2	B	2000	20$
S3	L	2000	150$
S4	P	2010	15$
S4	L	2010	160$





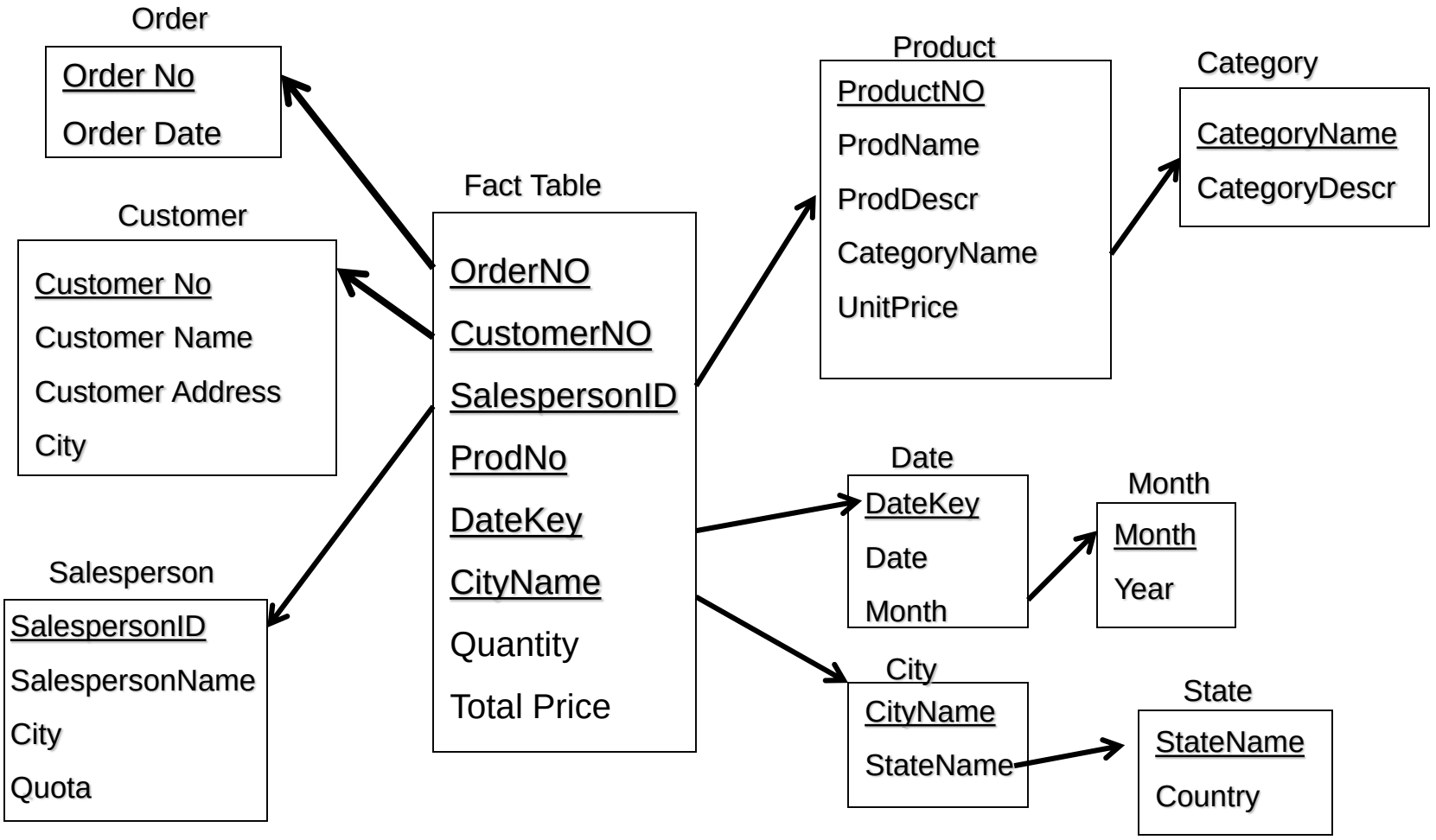
Product

Date

Why Surrogate Key

- Recall that data warehouse is non-volatile
  - Data can be updated
    - same customer, different address
    - discount gets assigned same code as earlier one
  - In operational database: overwrite
  - Not desirable in data warehouse
    - Incorrect data aggregation; when in 2010 querying sales in Dallas for 2000, John's purchases should count for Dallas, not New York

Snowflake Schema



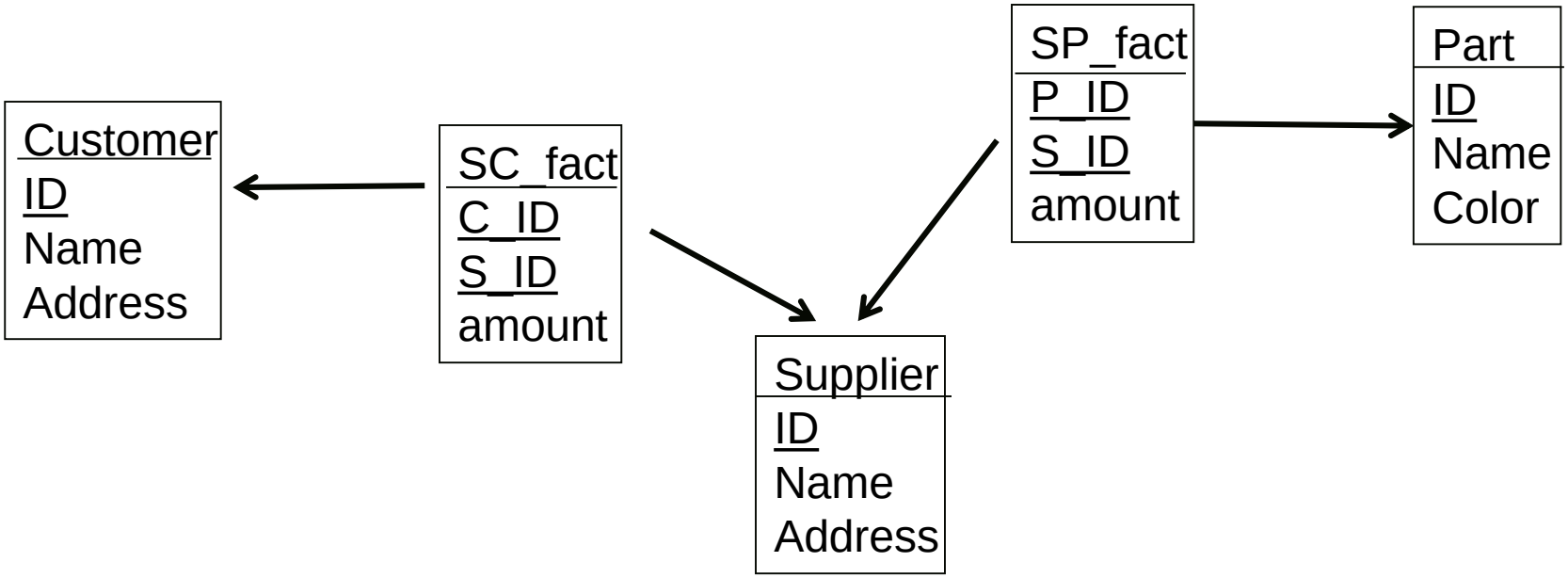
Snowflake Scheme

- More normalized dimensions
  - Avoid redundancy
  - Reduce size of large dimension
    - Customer → City → Country
  - More efficient data insertion

- Disadvantages: need for joins
  - However, join on foreign key; tables index on primary key

Fact Constellation

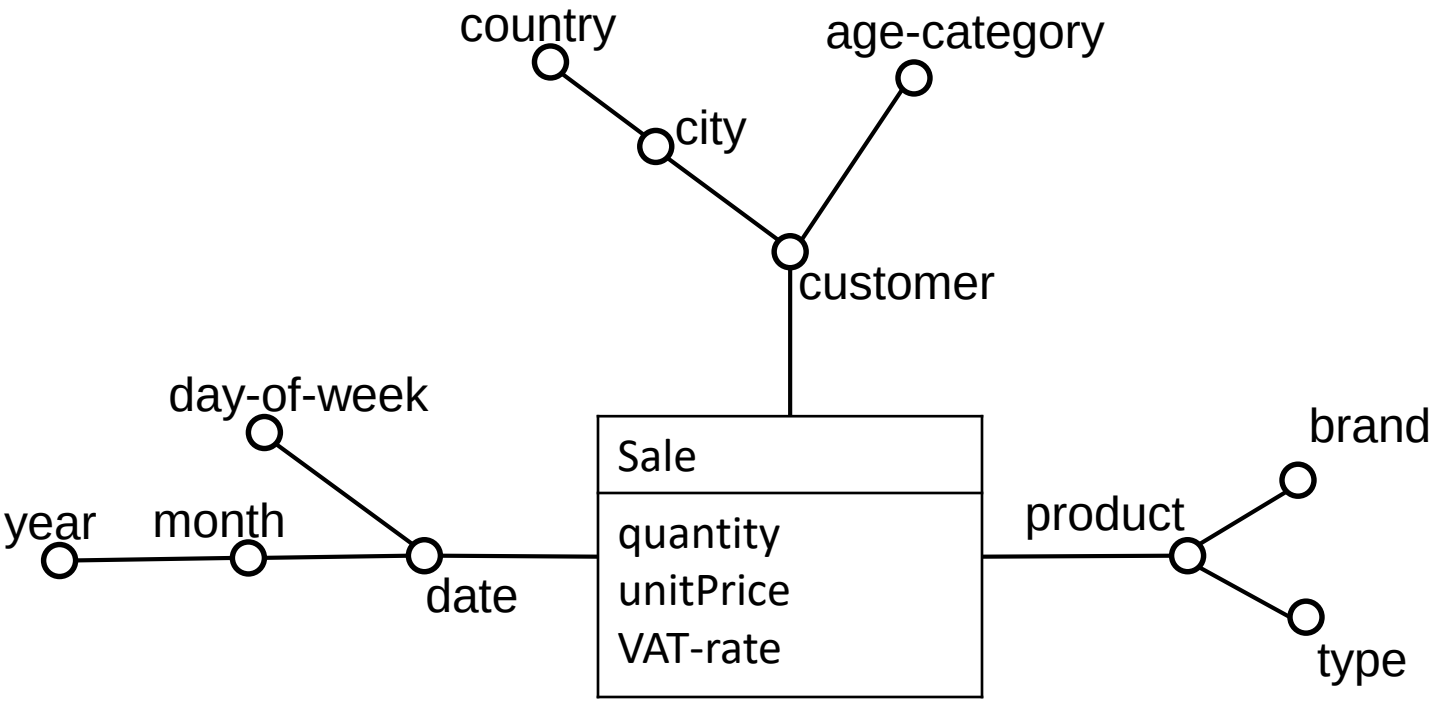
- Multiple fact tables share the same dimensions
  - E.g., (part, customer) shares Customer with (supplier, customer)



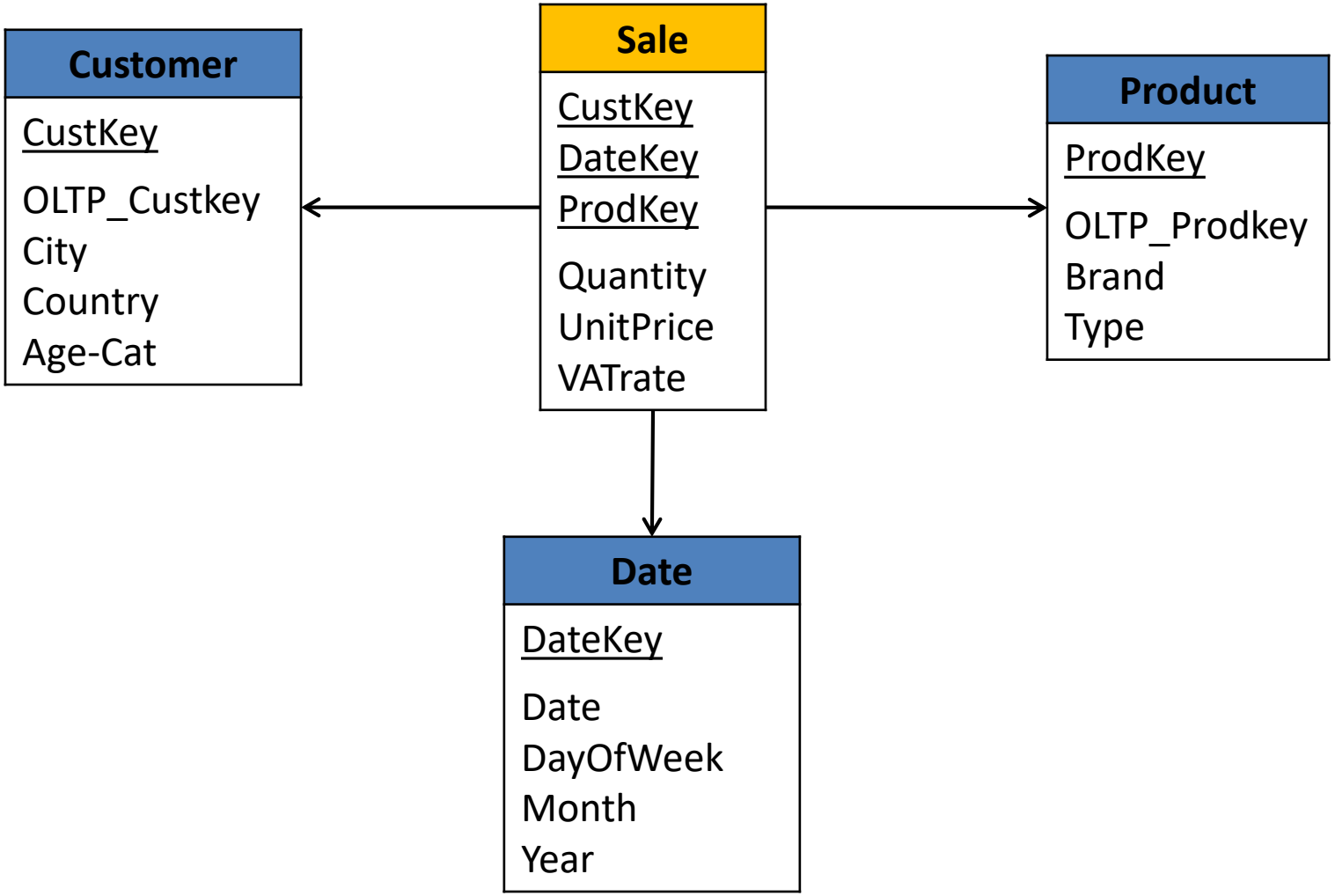
Fact Constellation

- Reuse as much as possible existing dimensions
  - Populating a dimension requires quite some effort → Reuse
  - Makes it easier to combine different cubes
- Notion of a "conformed dimension"
  - Definition of customer, date, product, … agreed upon by all departments
  - Makes it easier to integrate different data marts

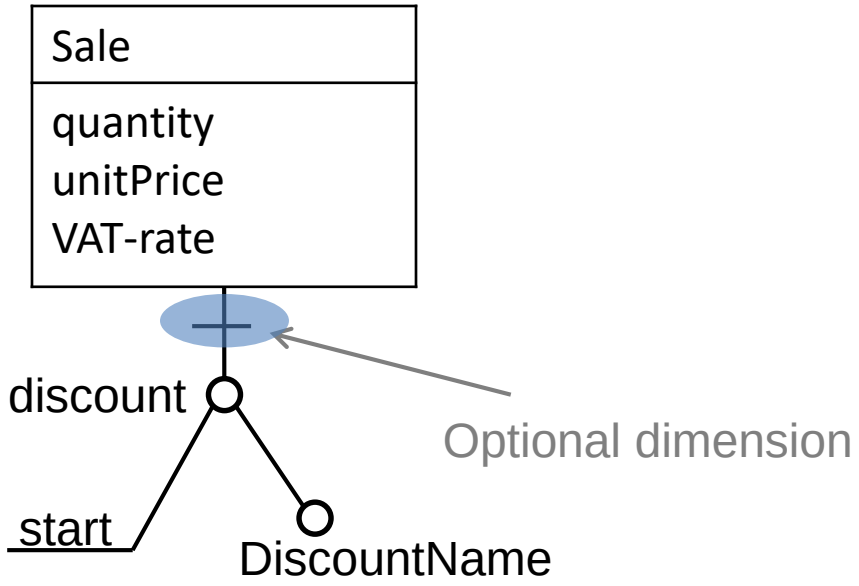
Examples


country
age-category
city
customer
day-of-week
year
month
date
Sale
quantity
unitPrice
VAT-rate
brand
product
type

Corresponding Star-Schema

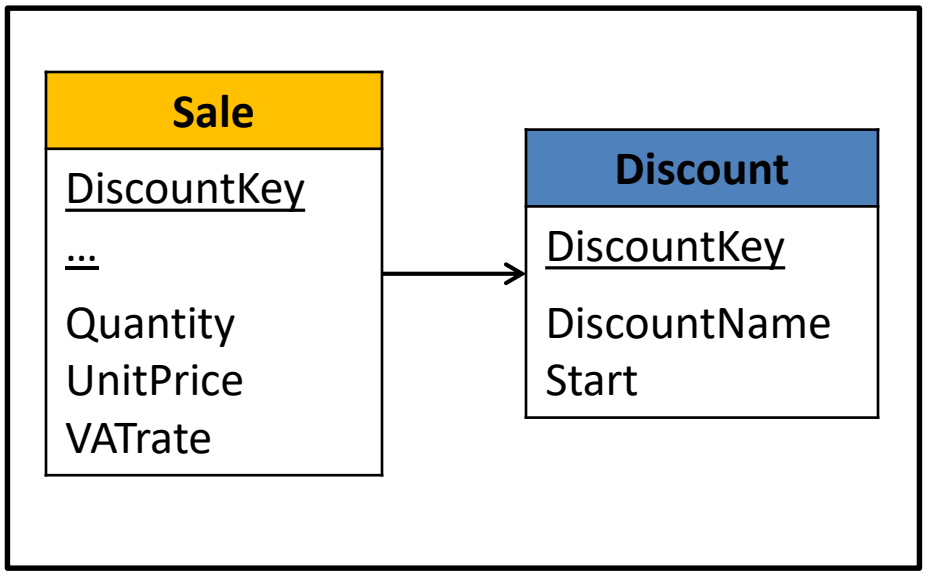


Example


Sale
quantity
unitPrice
VAT-rate
discount
start
DiscountName
Optional dimension

Optional Dimension

- Avoid the use of “null”
  - Confusing and non-descriptive
  - Special case for queries (null≠null)



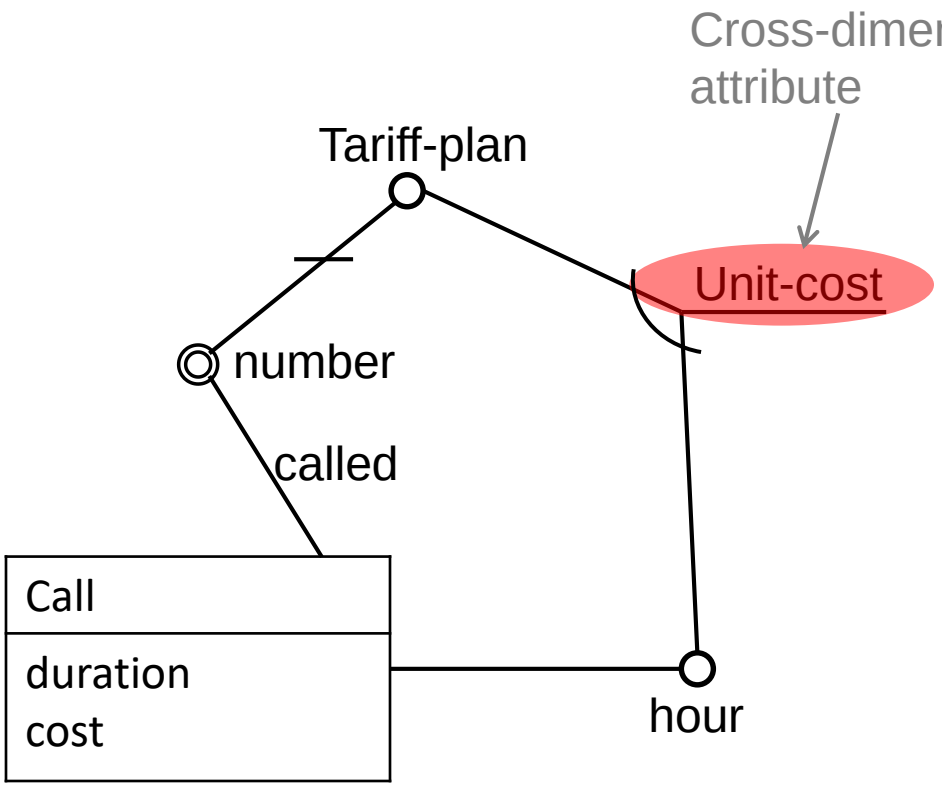
Schema

Discount

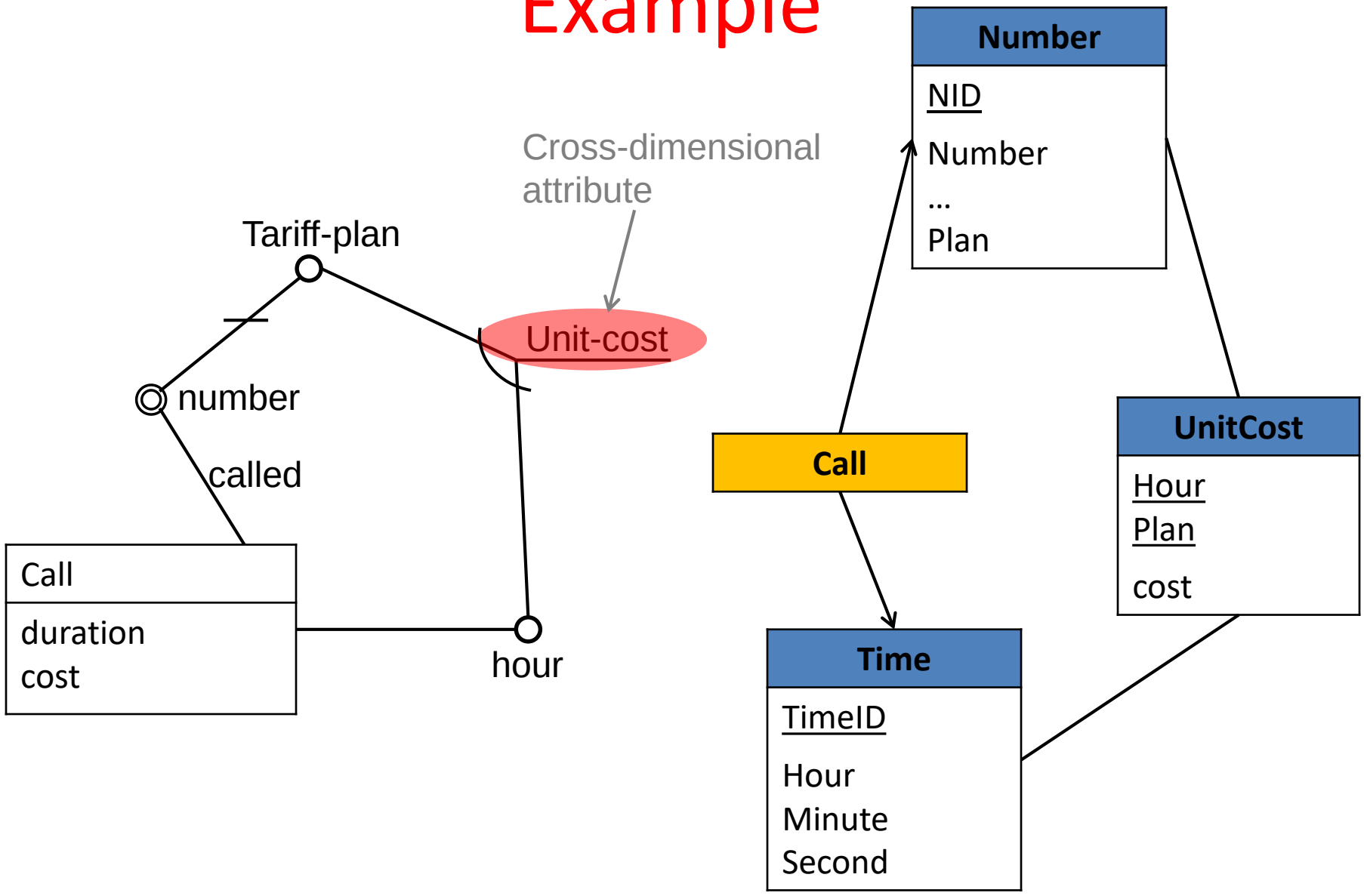
Key	Name	Start
0001	No discount	n/a
0002	Loyalty	Jan 2011
0003	Christmas	Dec 2012

Database

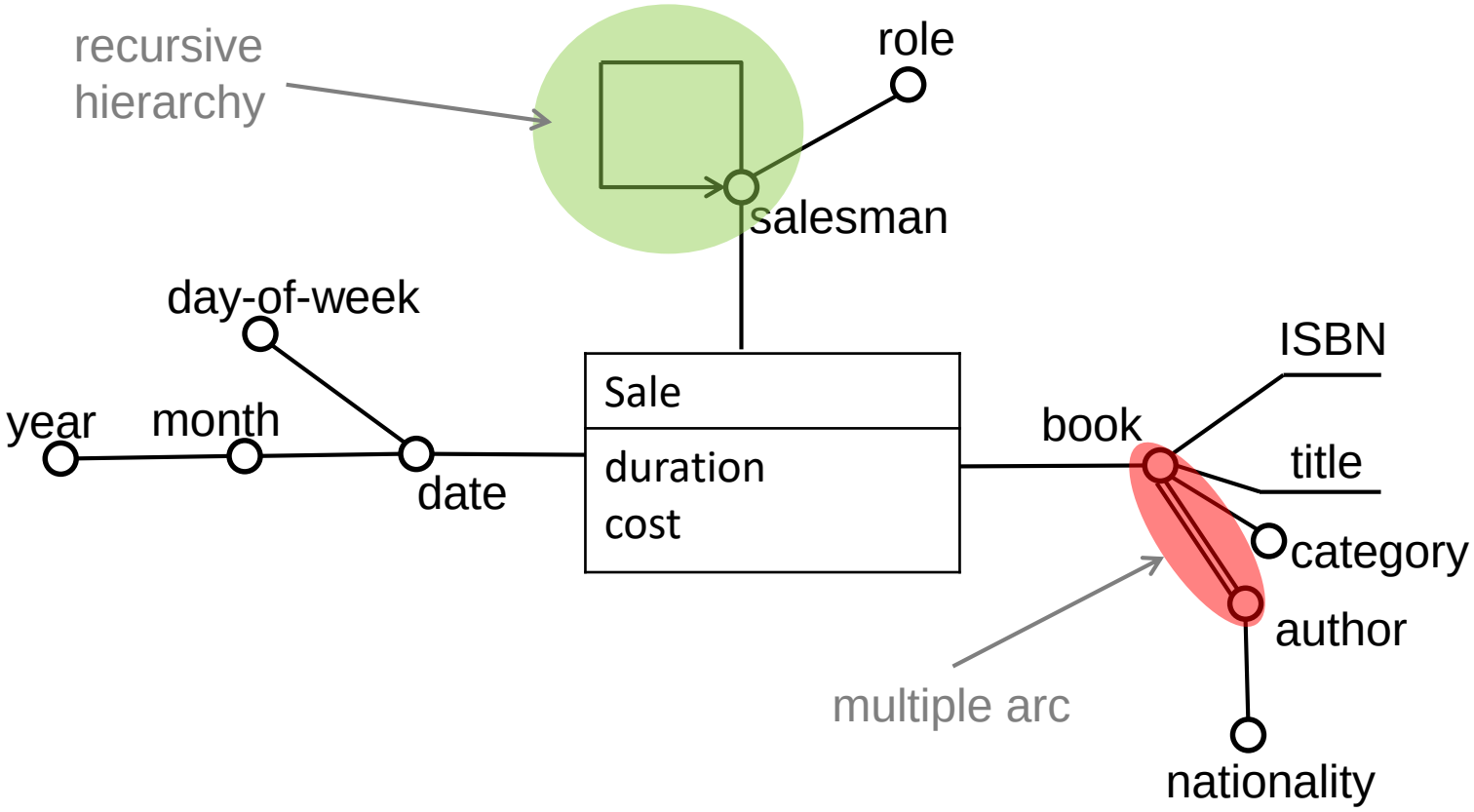
Example


Cross-dimensional attribute
Tariff-plan
Unit-cost
number
called
Call
duration
cost
hour

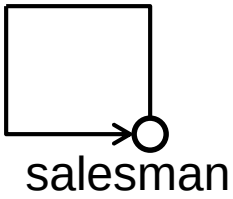
Example


Cross-dimensional attribute
Tariff-plan
Unit-cost
number
called
Call
duration
cost
hour
Number
NID
Number
...
Plan
Call
UnitCost
Hour
Plan
cost
Time
TimeID
Hour
Minute
Second

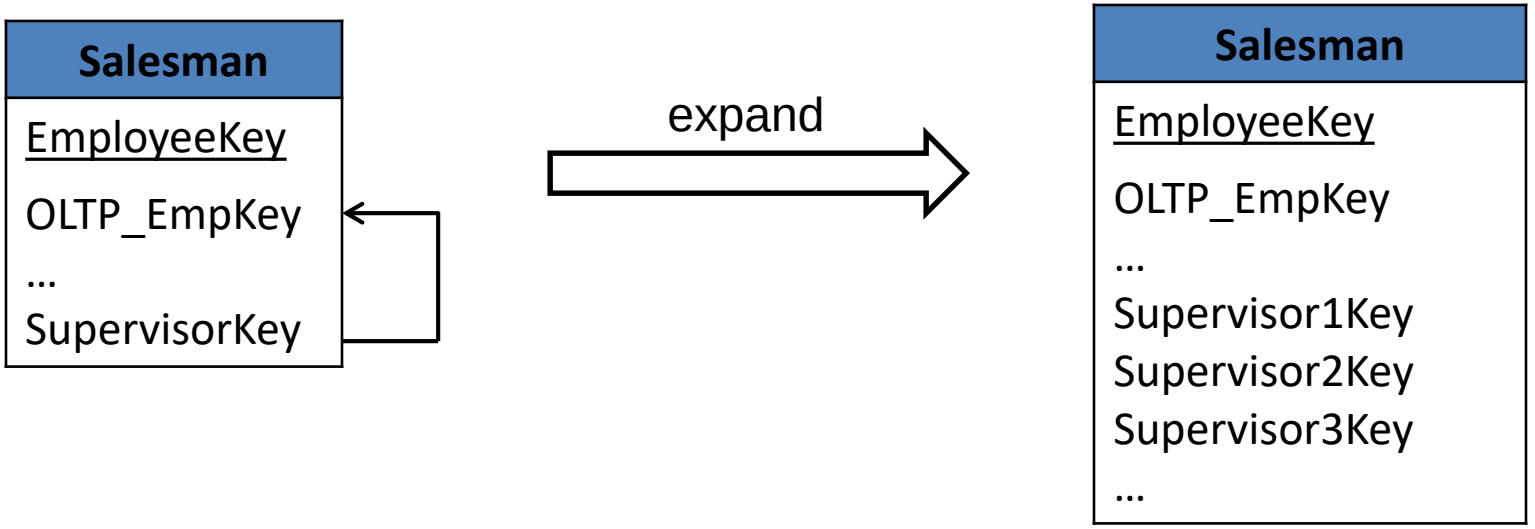
Example


recursive
hierarchy
role
salesman
day-of-week
year
month
date
Sale
duration
cost
ISBN
book
title
category
author
nationality
multiple arc

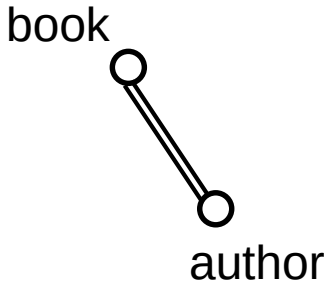
Recursive Hierarchy



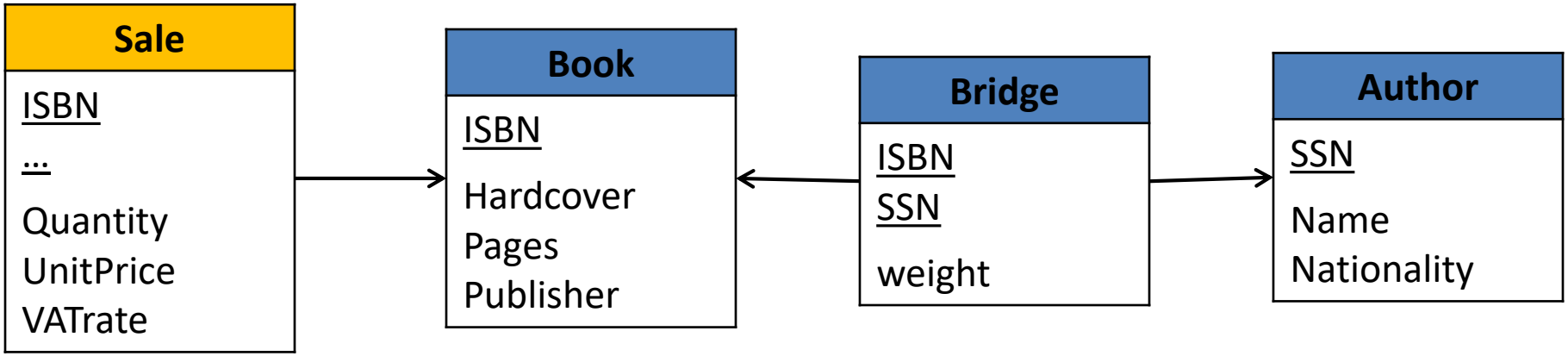
- Special type of hierarchy: user hierarchy
  - Encode using attribute “parent” (unique)
  - Do not use “child” (not unique)



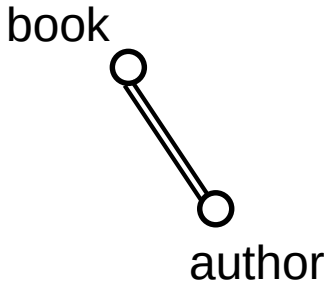
Multiple Arc – Option 1



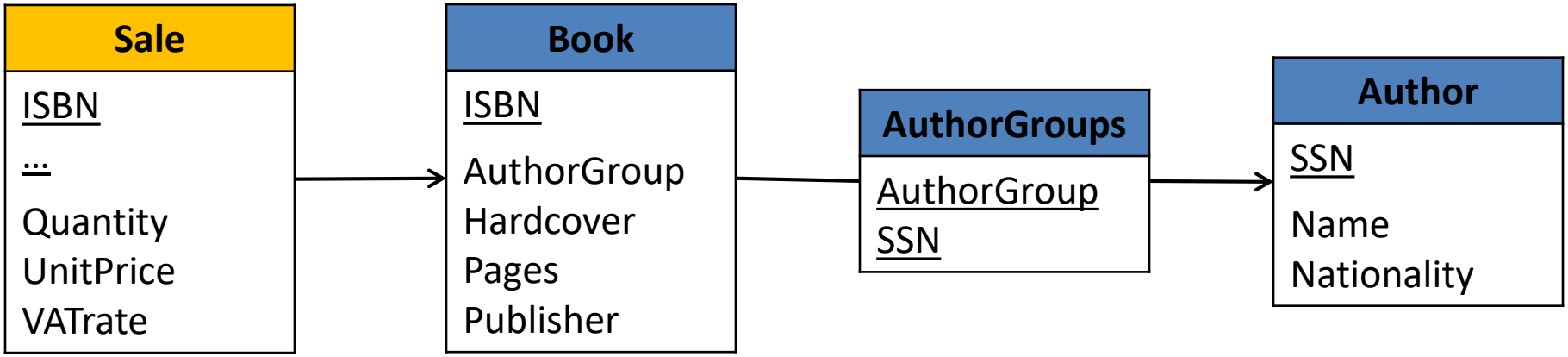
- Use bridge table



Multiple Arc – Option 2



- Create artificial groups



Note: this is not a snowflake!

Summary

- Different types of aggregation operators
  - Distributive, algebraic, holistic
- Logical schemes for data warehouses
  - Star schema
  - Snowflake
- Translation of conceptual model to tables