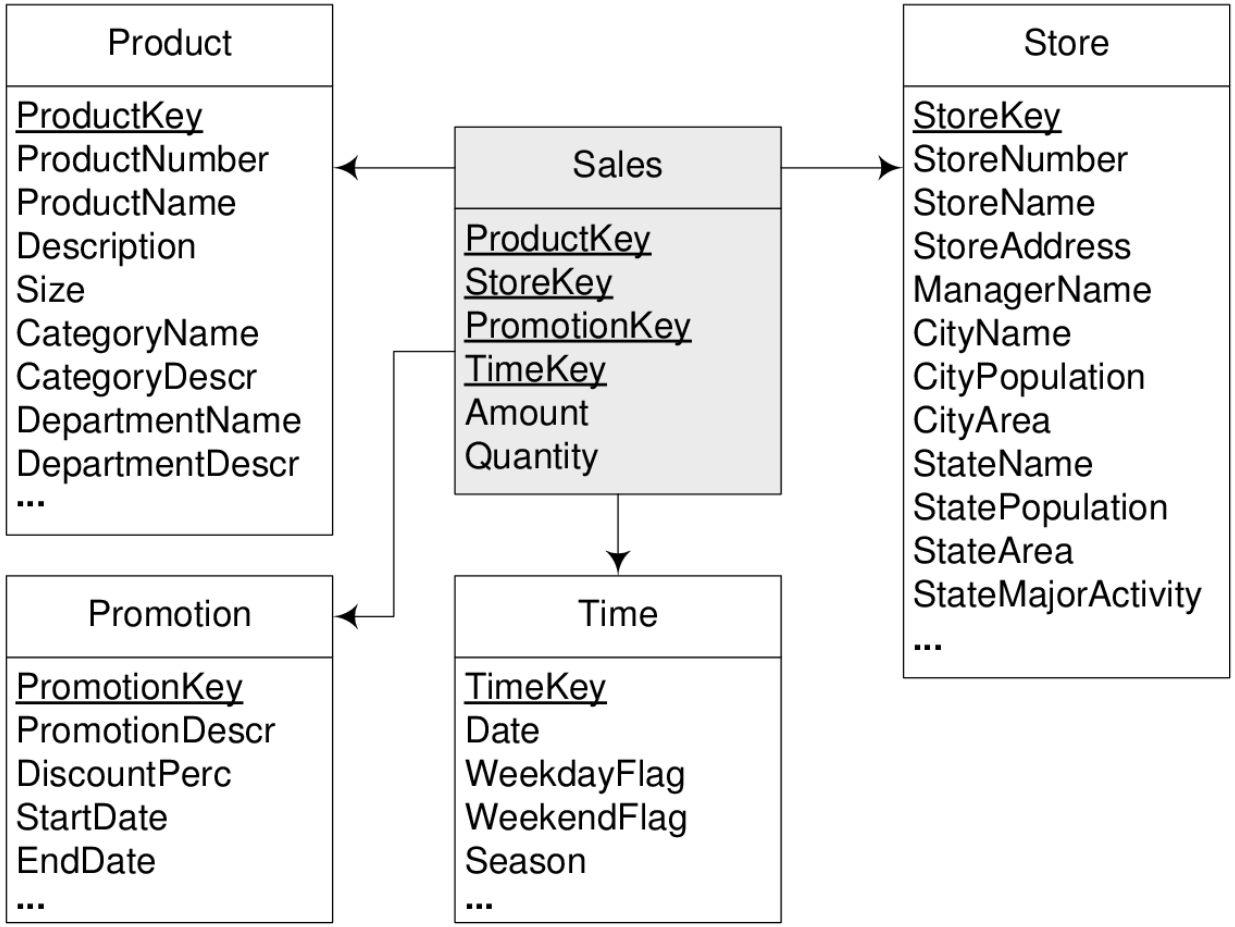
Data Analysis and Integration

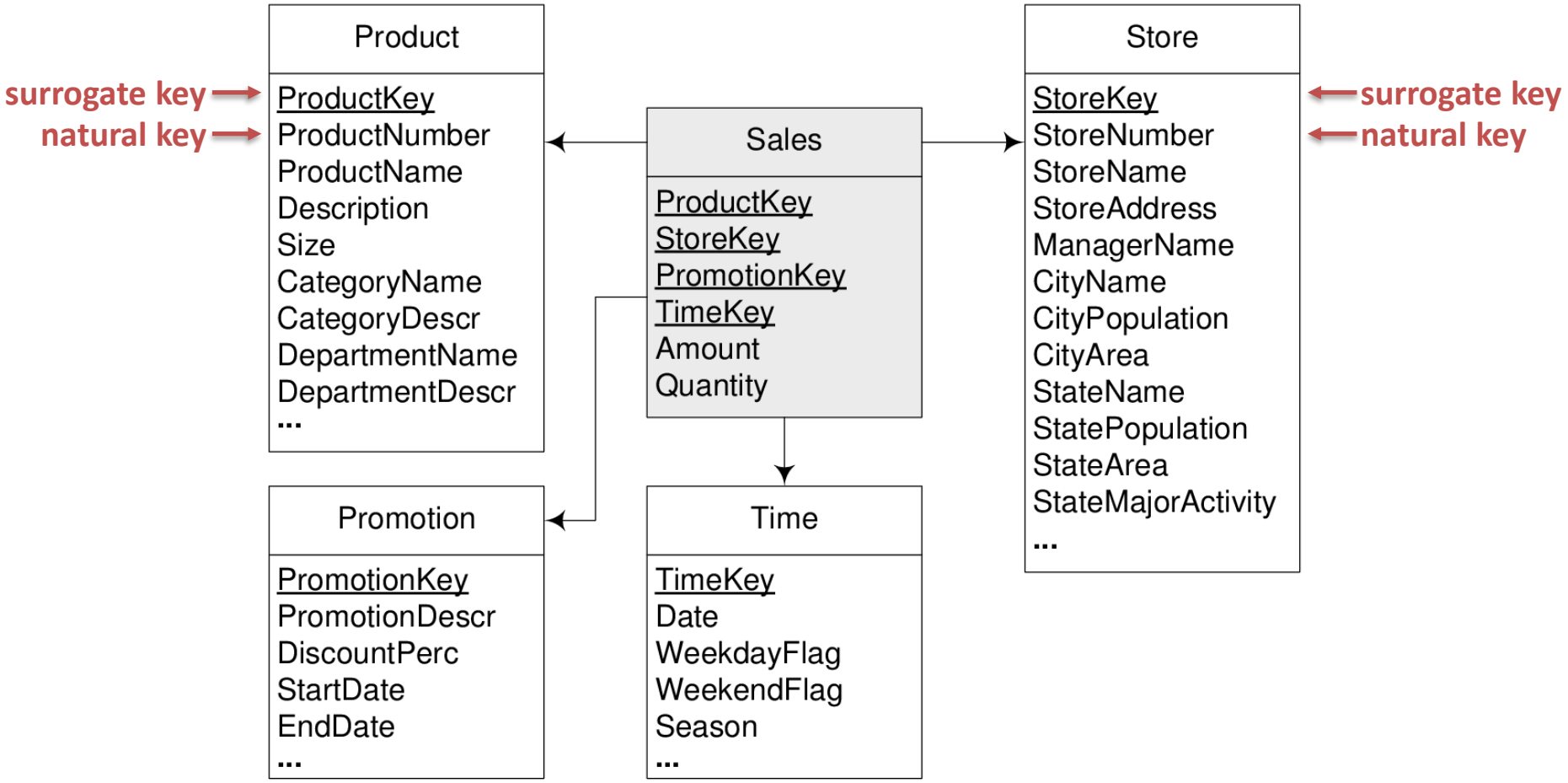
Data warehouse design

Star schema


Product
ProductKey
ProductNumber
ProductName
Description
Size
CategoryName
CategoryDescr
DepartmentName
DepartmentDescr
...
Sales
ProductKey
StoreKey
PromotionKey
TimeKey
Amount
Quantity
Store
StoreKey
StoreNumber
StoreName
StoreAddress
ManagerName
CityName
CityPopulation
CityArea
StateName
StatePopulation
StateArea
StateMajorActivity
...
Promotion
PromotionKey
PromotionDescr
DiscountPerc
StartDate
EndDate
...
Time
TimeKey
Date
WeekdayFlag
WeekendFlag
Season
...

Surrogate keys

- Each **dimension** has its own key



Surrogate keys

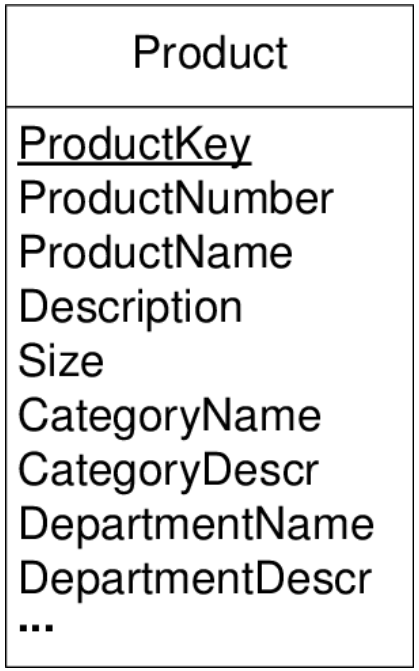
- A data warehouse has its own primary keys
  - these are called **surrogate keys** or **technical keys**
    - **ProductNumber** identifies products in the original database
    - **ProductKey** identifies products in the data warehouse
  - **surrogate keys** replace the original primary keys (**natural keys**)
    - provide independence from keys in the original data sources
    - solve inconsistencies between keys from multiple sources
  - represent keys as **integers** to improve efficiency
    - avoid less efficient data types, such as strings

Types of DW schema

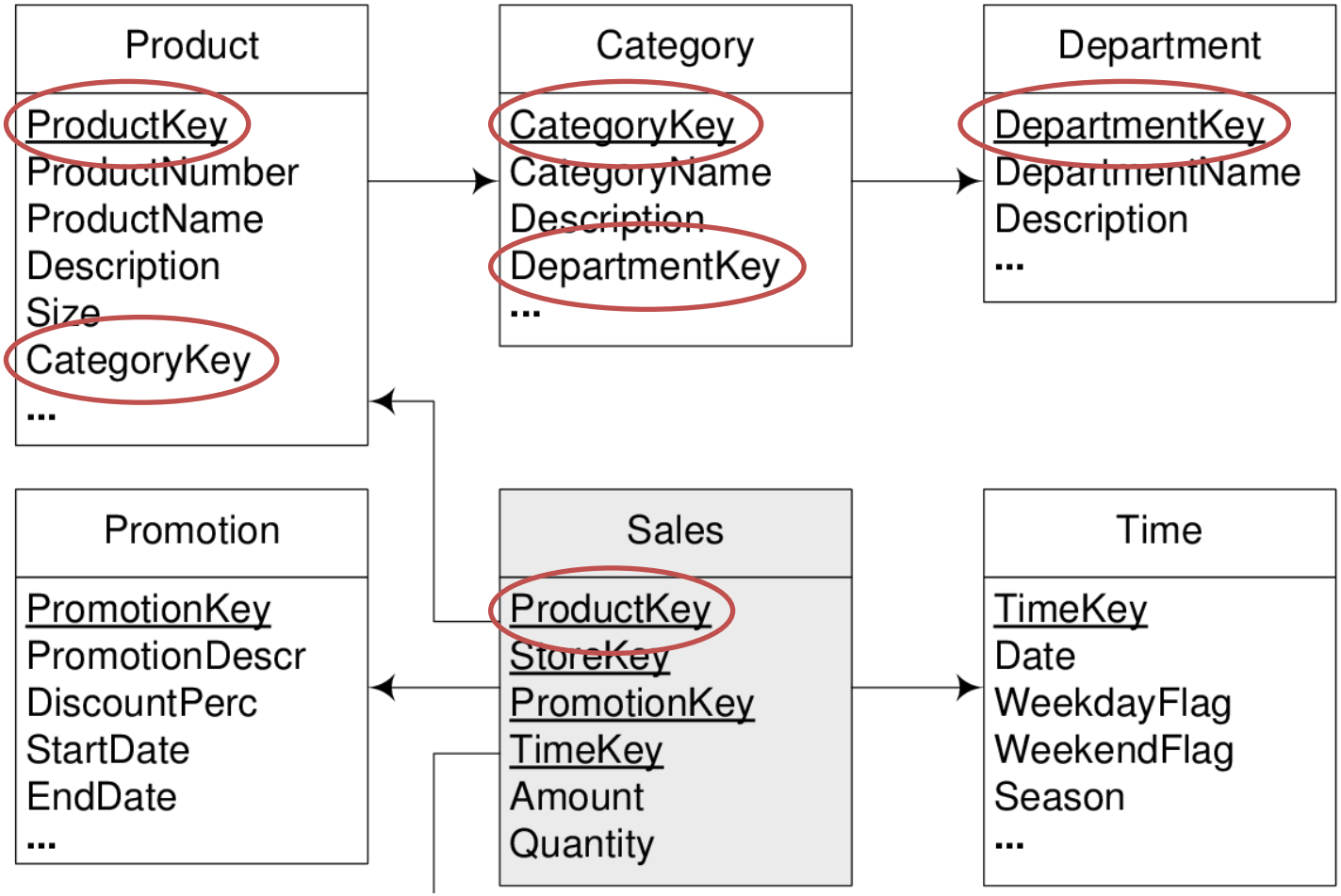
- Star schema
  - has a single table for each dimension, even in the presence of hierarchies (denormalized dimension tables)
- Snowflake schema
  - has normalized tables for all dimensions and their hierarchies
- Starflake schema
  - is a mix between star and snowflake schemas (both normalized and denormalized dimensions)
- Constellation schema
  - has multiple fact tables, possibly with shared dimension tables, and is viewed as a collection of stars

Snowflake schema

- When dimensions store redundant data (are denormalized)
  - **CategoryName**, **CategoryDescr** are the same for multiple products
    - replace by **CategoryKey** and move them to another table
  - **DepartmentName**, **DepartmentDescr** are the same for multiple categories
    - replace by **DepartmentKey** and move them to another table



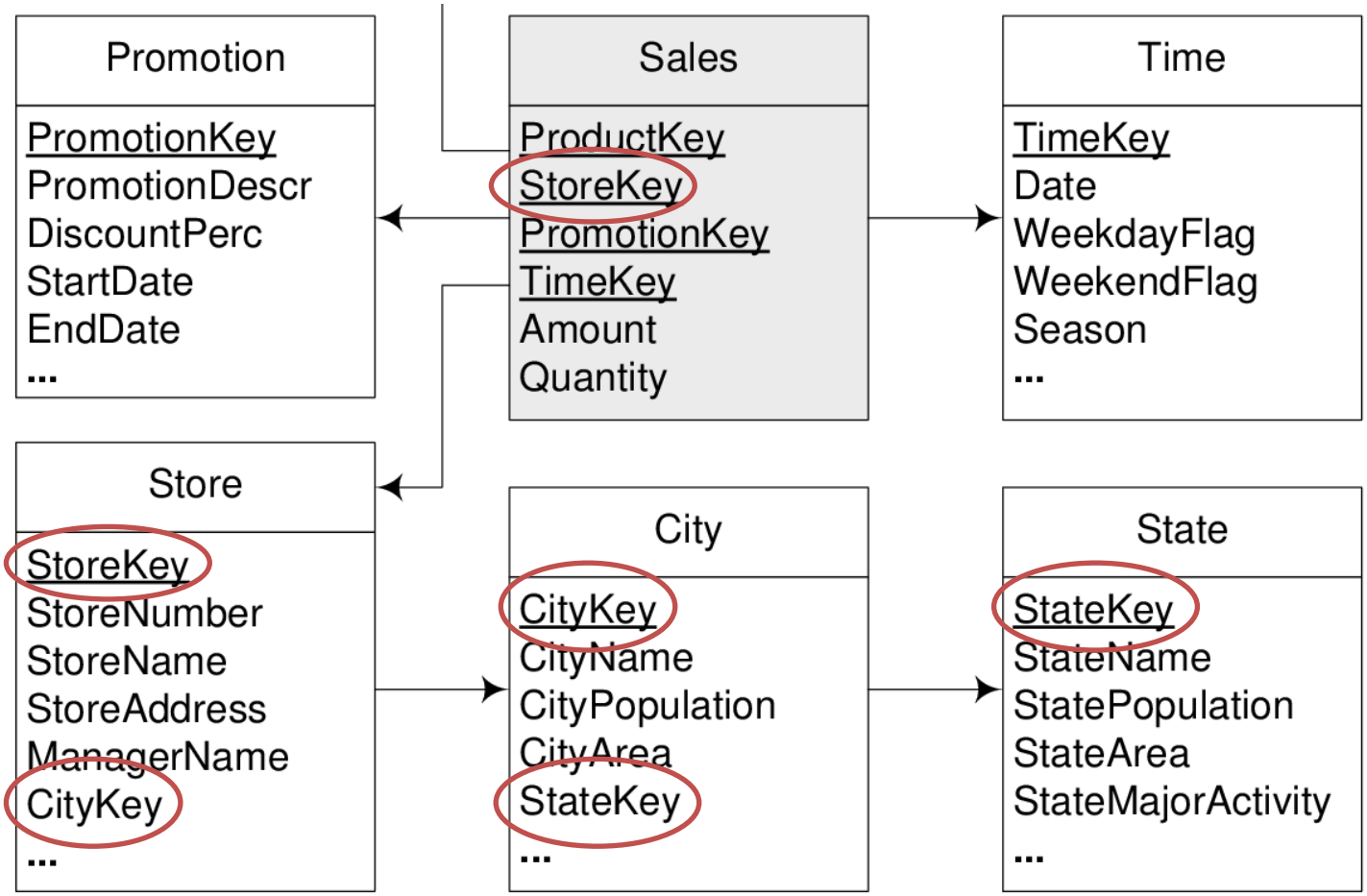
Snowflake schema



Snowflake schema

- Another example
  - **CityName**, **CityPopulation**, **CityArea** are the same for multiple stores
    - replace by **CityKey** and move them to another table
  - **StateName**, **StatePopulation**, **StateArea**, etc. are the same for multiple products
    - replace by **StateKey** and move them to another table

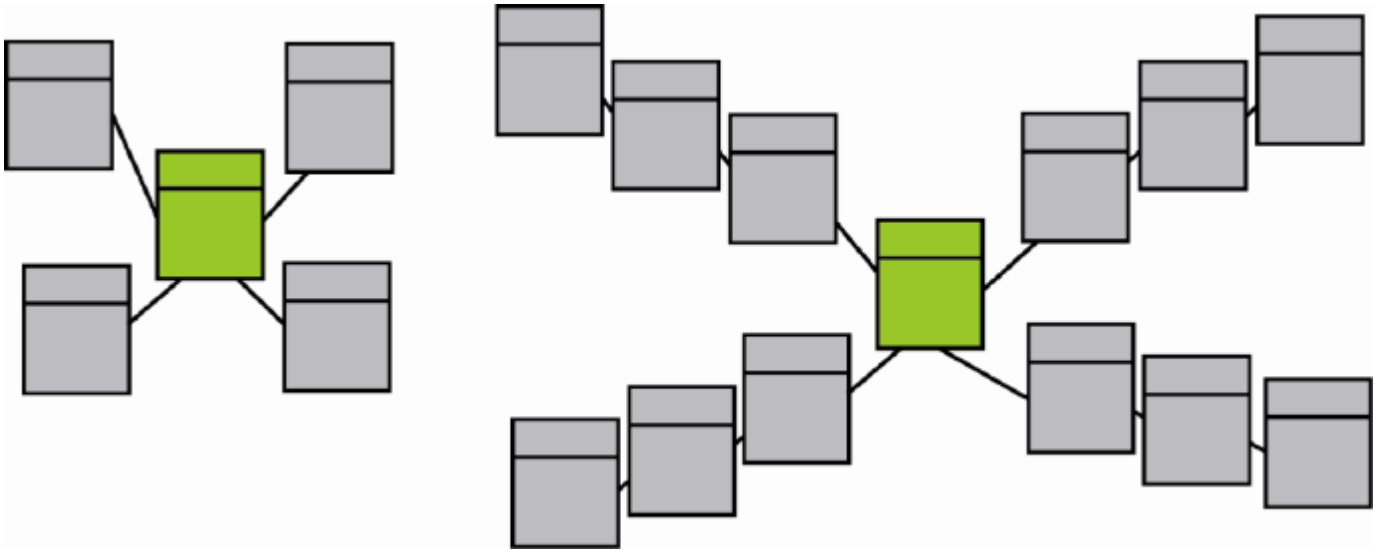
Snowflake schema



Snowflake schema

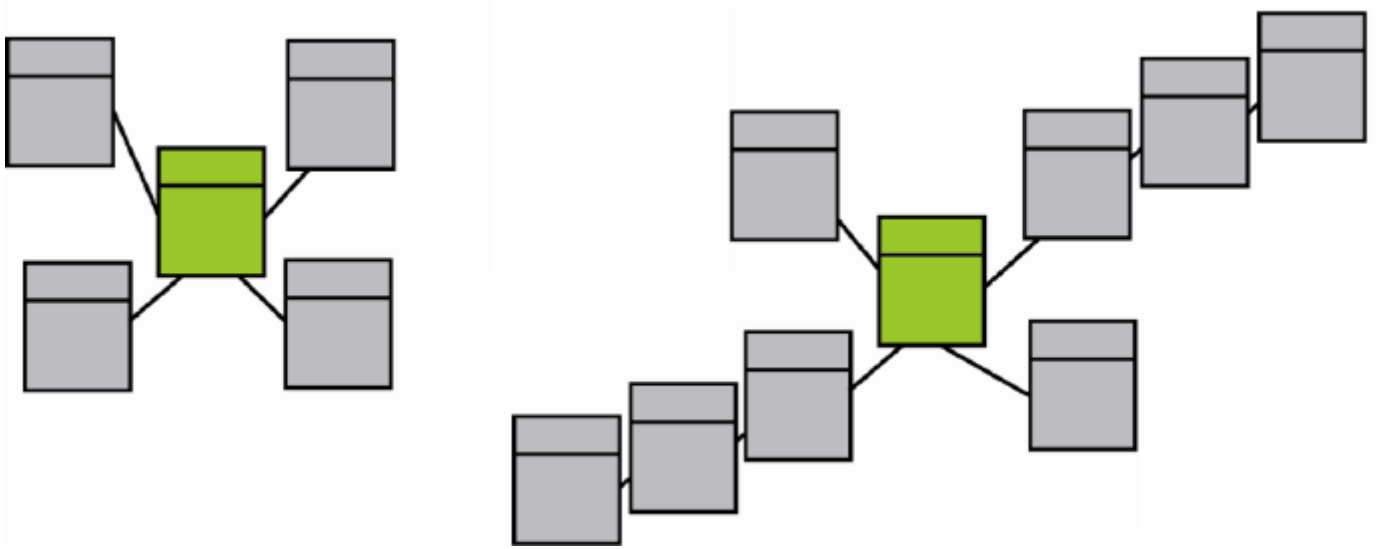
Snowflake schema

- Star vs. snowflake schema
  - all dimensions are normalized



Starflake schema

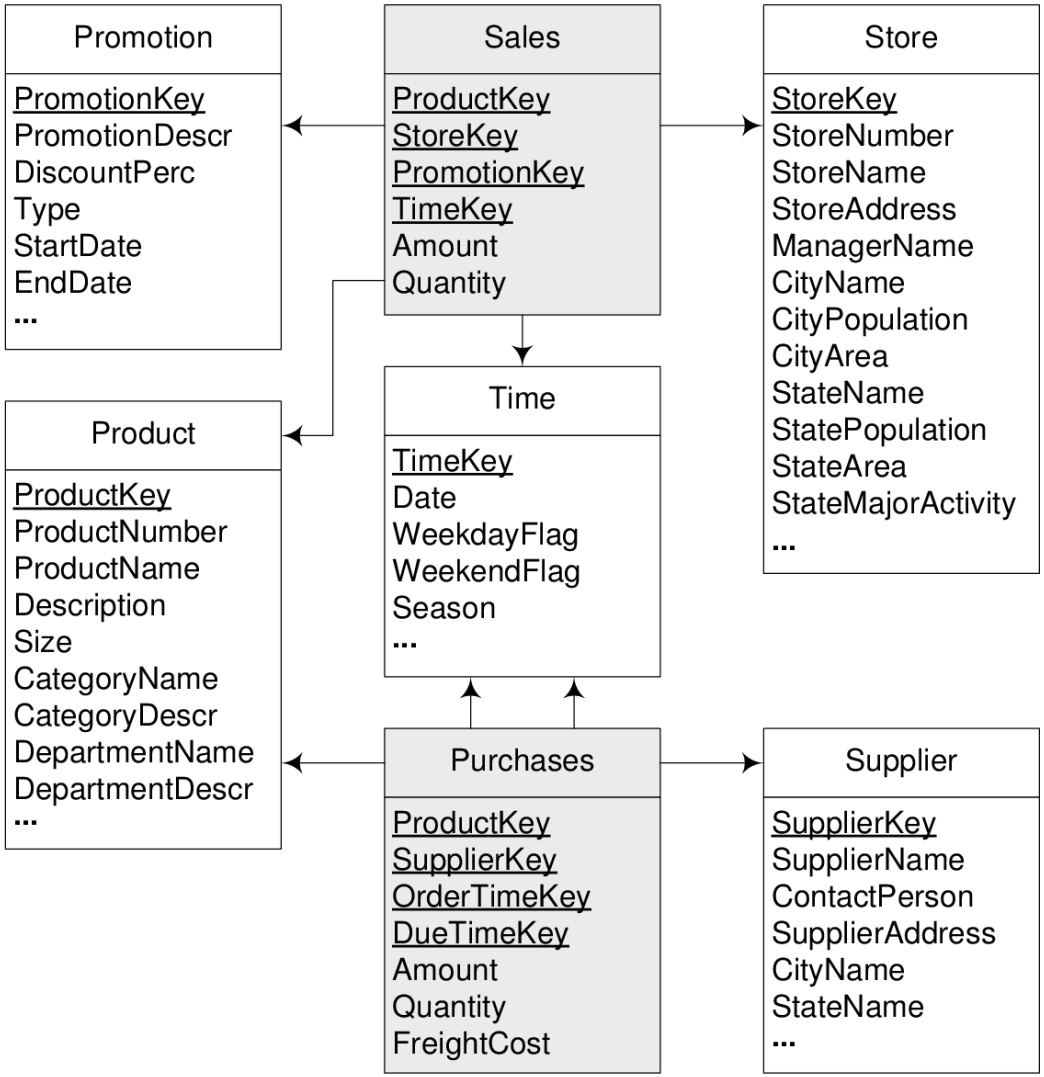
- Star vs. starflake schema
  - some dimensions are normalized



Constellation schema

- Multiple fact tables
  - e.g. sales facts, purchase facts
- Some dimension tables may be shared
  - e.g. product, time

Constellation schema

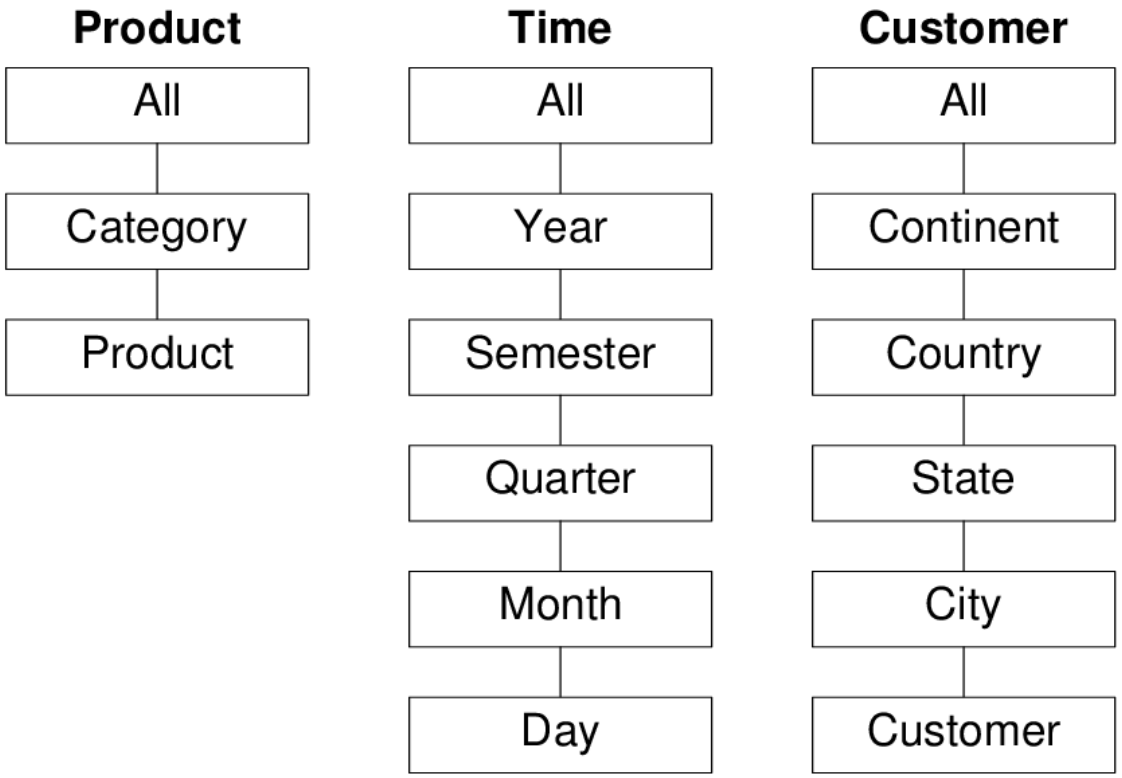


Hierarchies

- Dimension hierarchies are essential to enable analysis at different levels of detail
  - define a hierarchical structure of levels relating lower-level members to higher-level ones
- In real-world applications, users must deal with complex hierarchies of various kinds
  - however, current DW and OLAP systems support only a limited set of hierarchies
  - here, we are going to study additional types of hierarchies

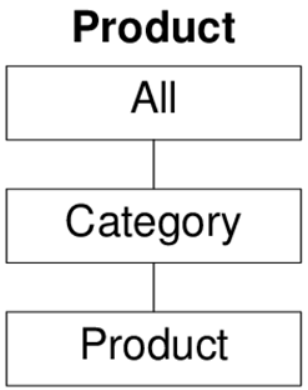
Hierarchies

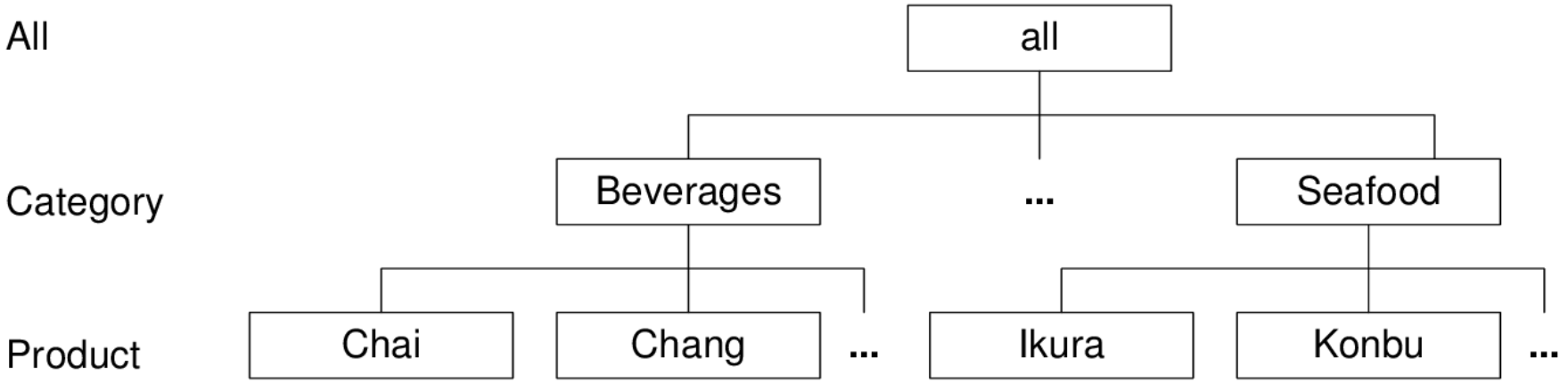
- Product, time and customer dimensions



Hierarchy members

- Example of a hierarchy and its members



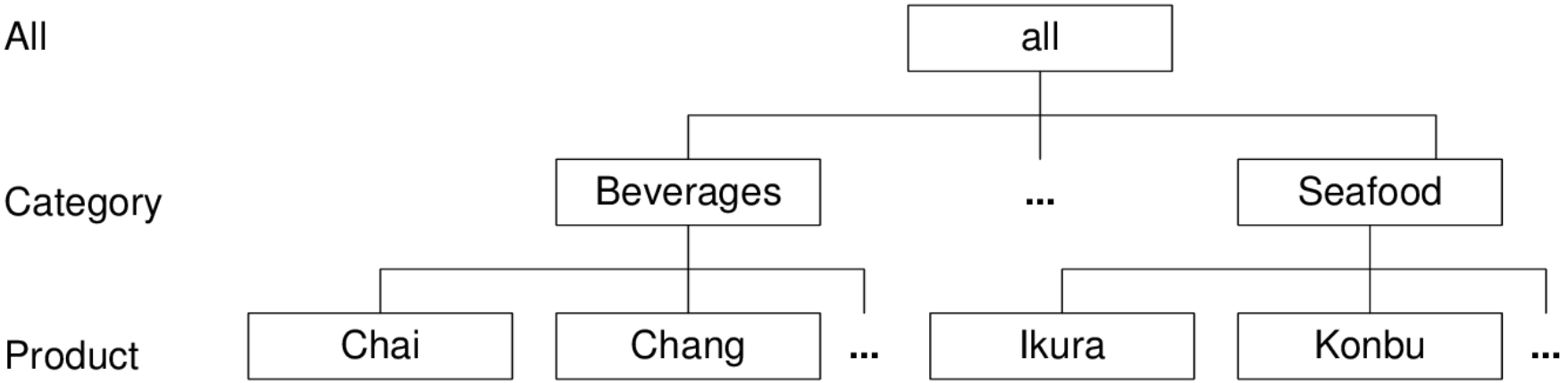


Types of hierarchy

- Balanced hierarchy
- Unbalanced hierarchy
- Recursive hierarchy
- Generalized hierarchy
- Ragged hierarchy
- Alternative hierarchy
- Parallel hierarchies
- Non-strict hierarchy

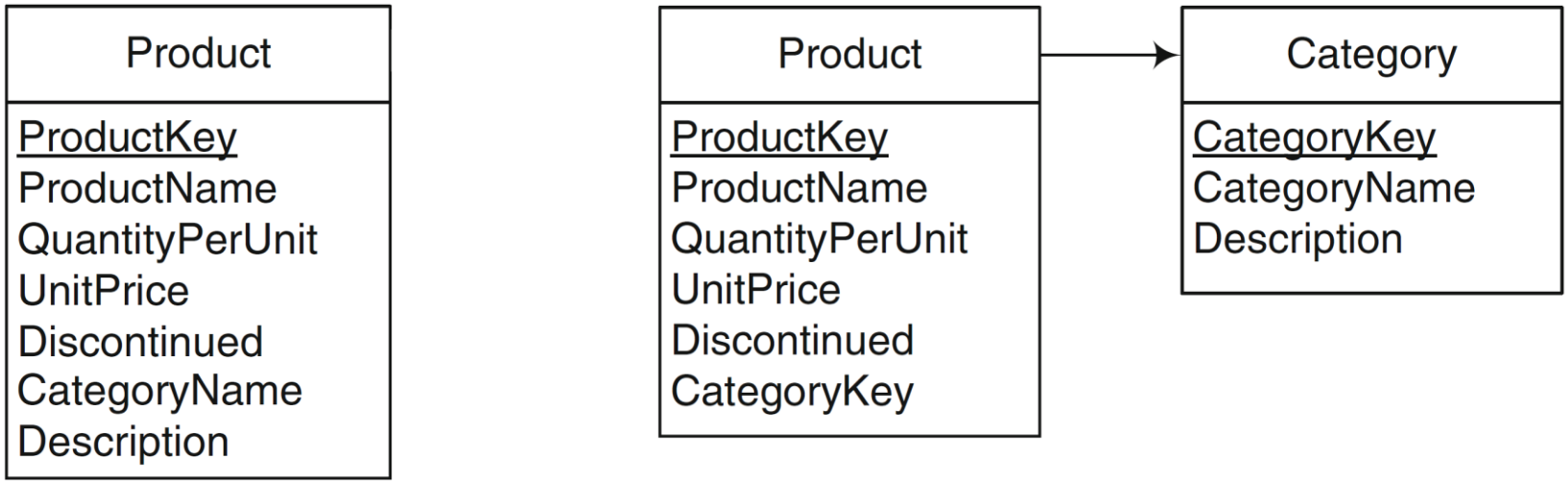
Types of hierarchy

- Balanced hierarchy
  - all levels are mandatory
  - all branches have the same length
  - a child member belongs to only one parent



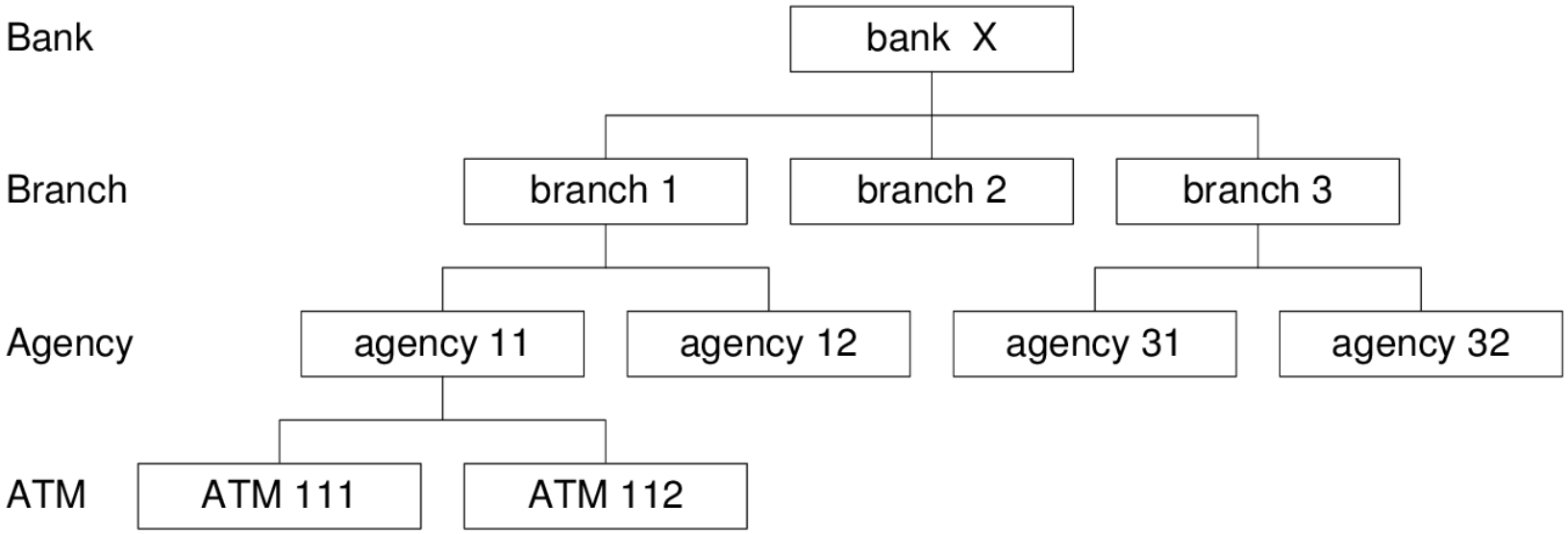
Types of hierarchy

- Balanced hierarchy
  - flat table (as in star schema) or snowflake structure



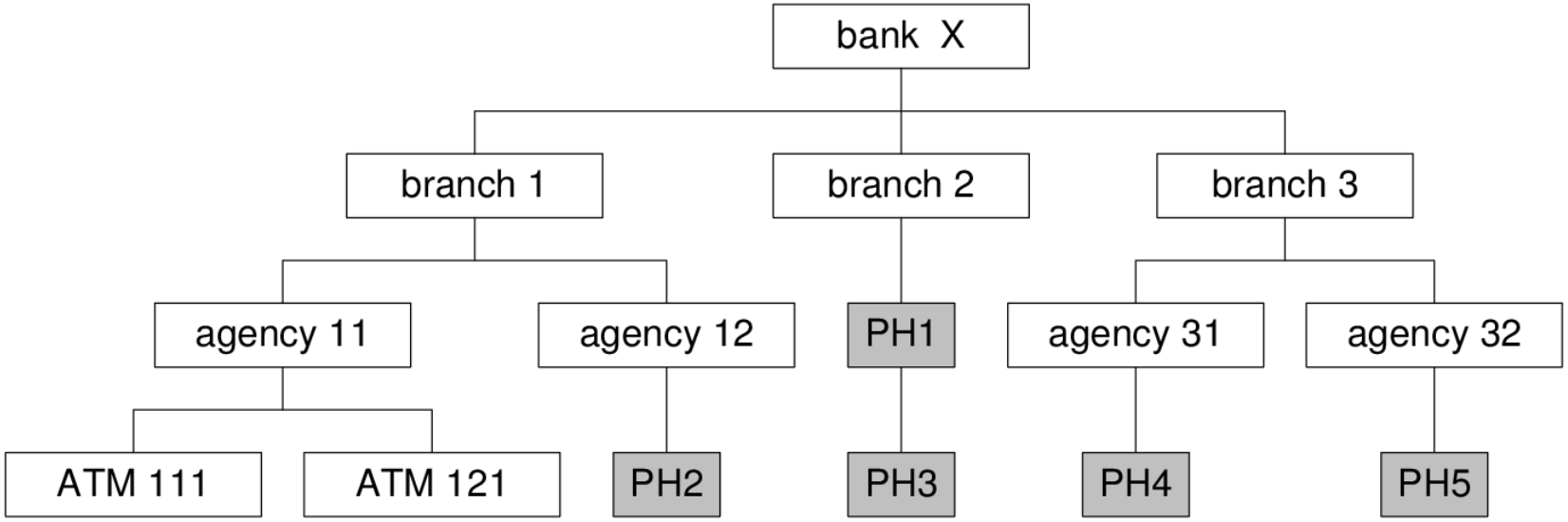
Types of hierarchy

- Unbalanced hierarchy
  - some levels are optional
  - branches may have different lengths
  - a child member belongs to only one parent



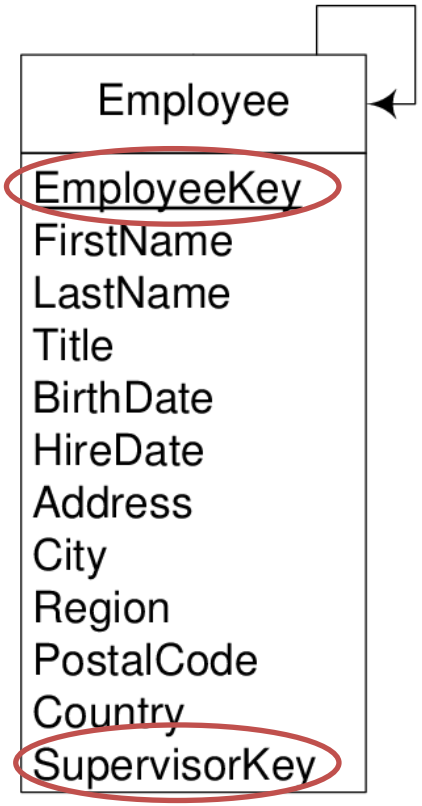
Types of hierarchy

- Unbalanced hierarchy
  - transform into balanced hierarchy by using placeholders
  - then use flat table or snowflake structure



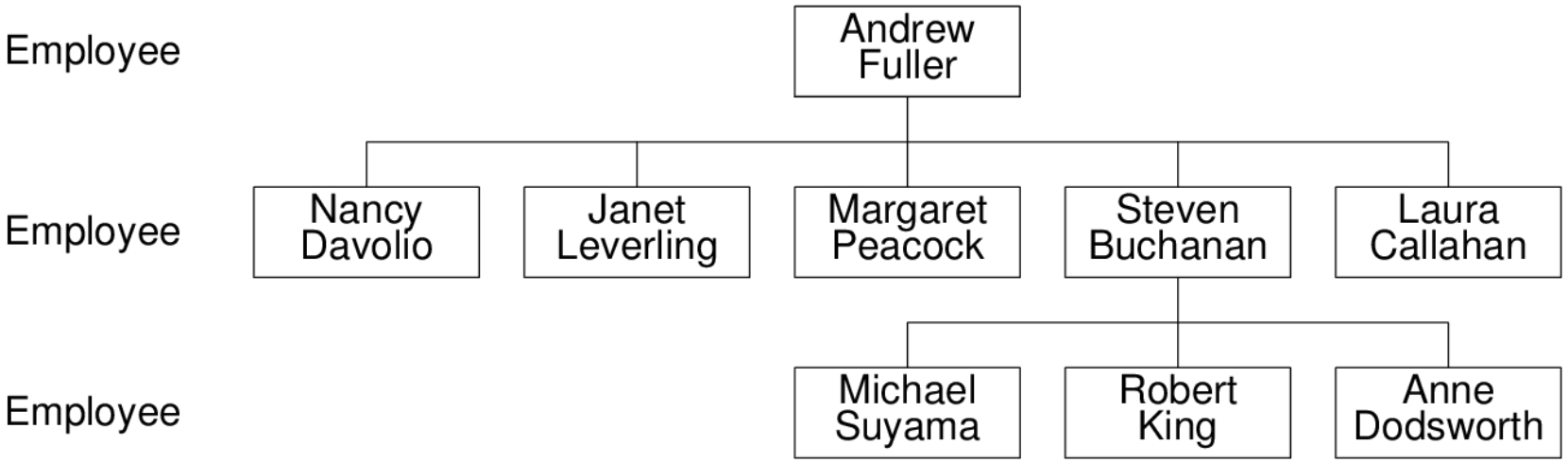
Types of hierarchy

- Recursive hierarchy
  - a scenario that we have seen before



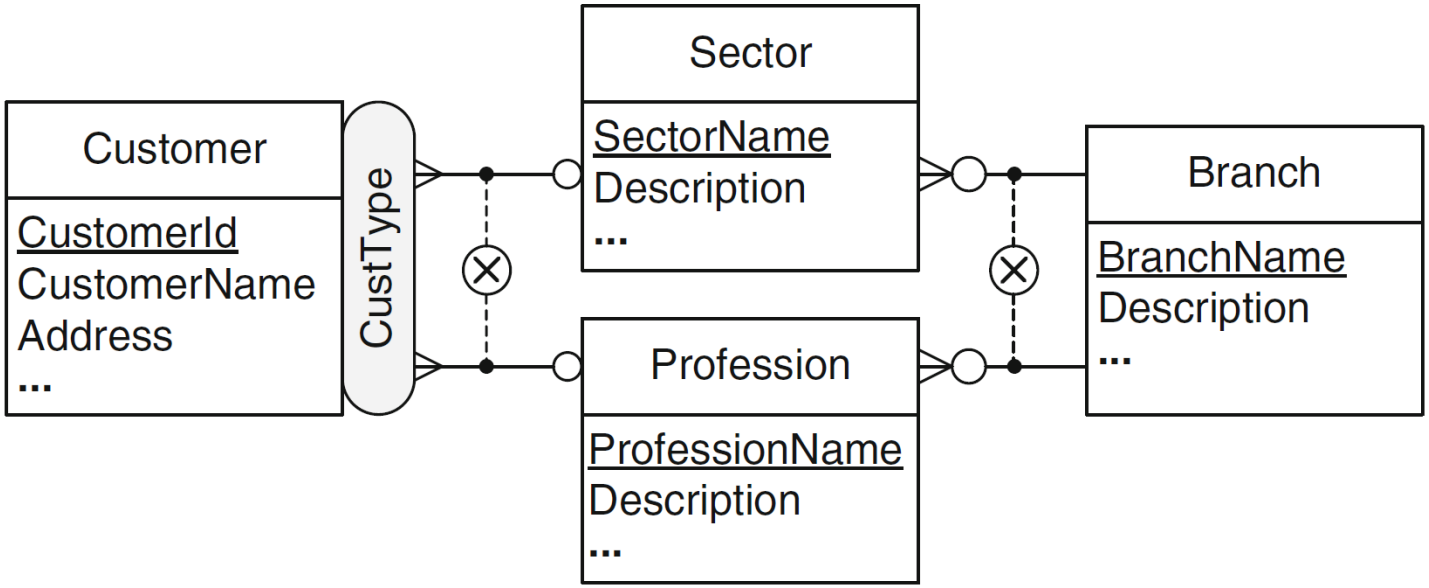
Types of hierarchy

- Recursive hierarchy
  - all levels are of the same type
  - can easily become an unbalanced hierarchy
  - requires recursive queries to traverse the hierarchy



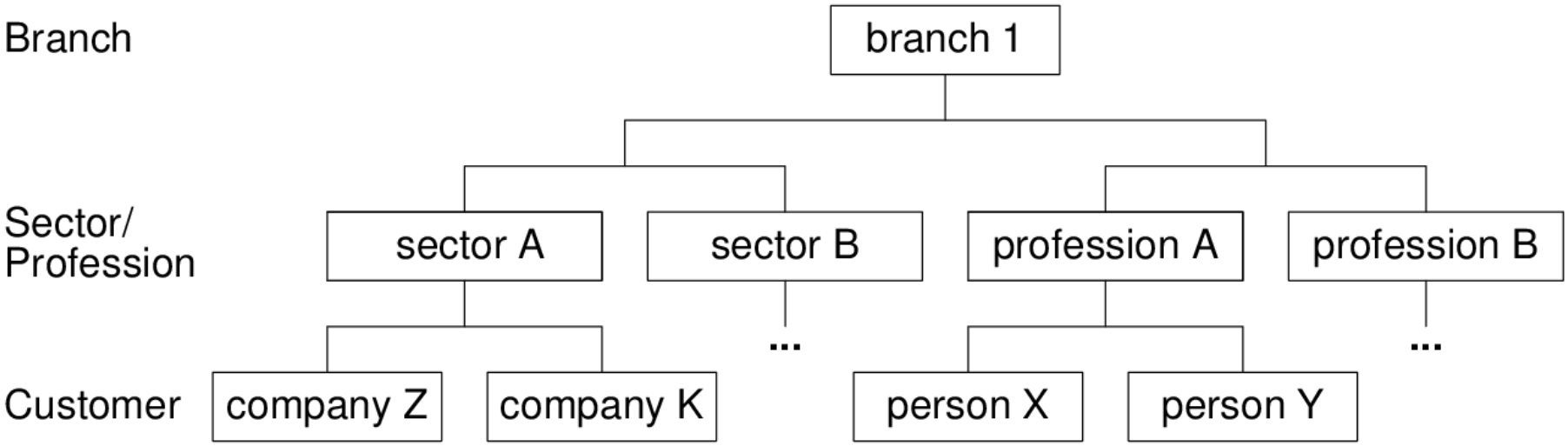
Types of hierarchy

- Generalized hierarchy
  - the same level may have different types
    - e.g. customers of a bank may be companies (with an industry **sector**) or individual persons (with a **profession**)



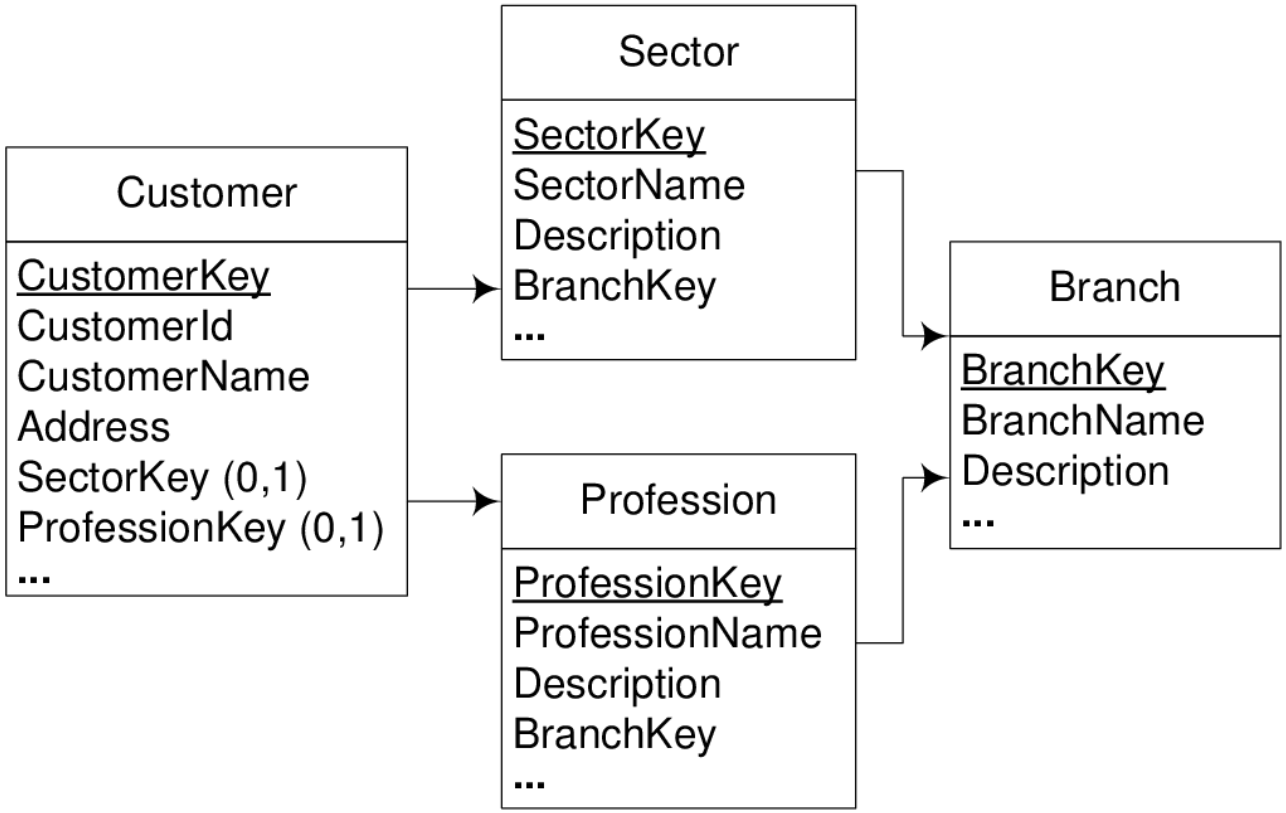
Types of hierarchy

- Generalized hierarchy
  - the same level may have different types
    - e.g. customers of a bank may be companies (with an industry **sector**) or individual persons (with a **profession**)



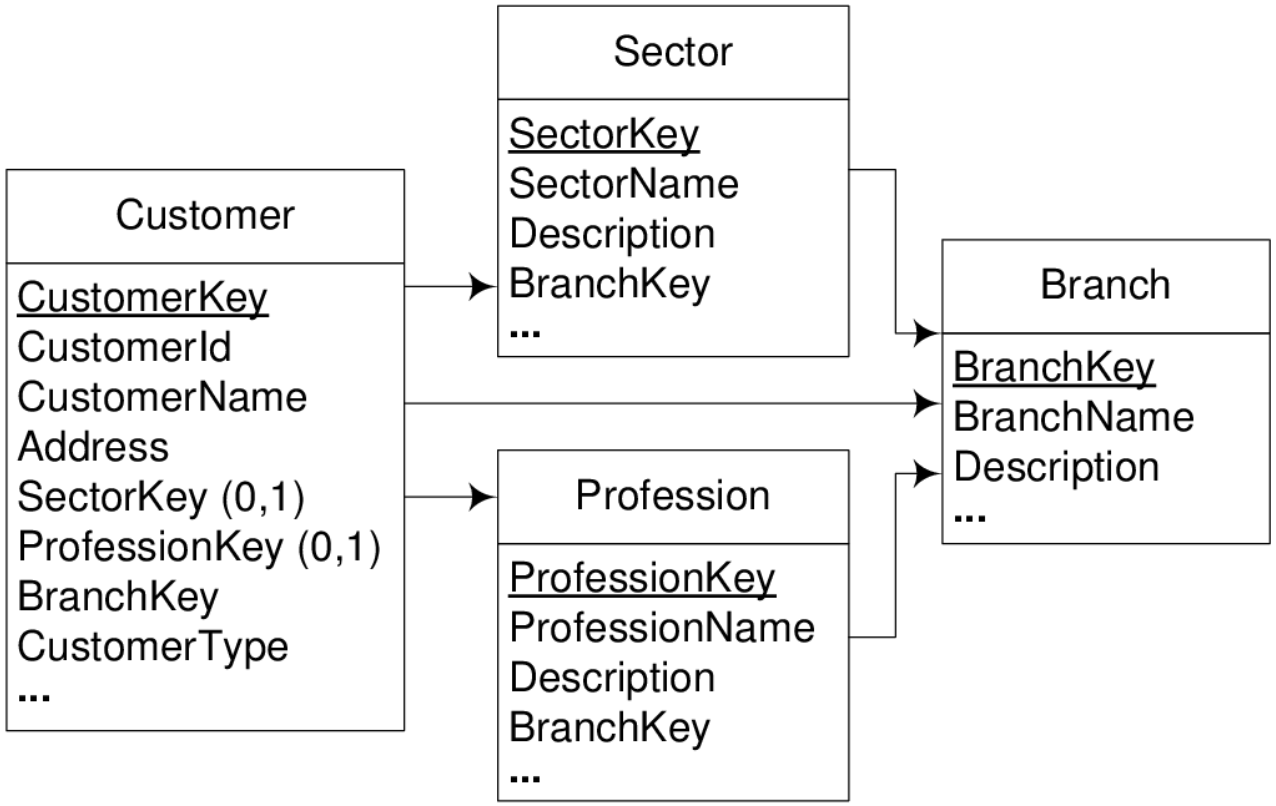
Types of hierarchy

- Generalized hierarchy
  - flat table with NULLs or snowflake structure (preferred)
    - different aggregation paths for different types of customer



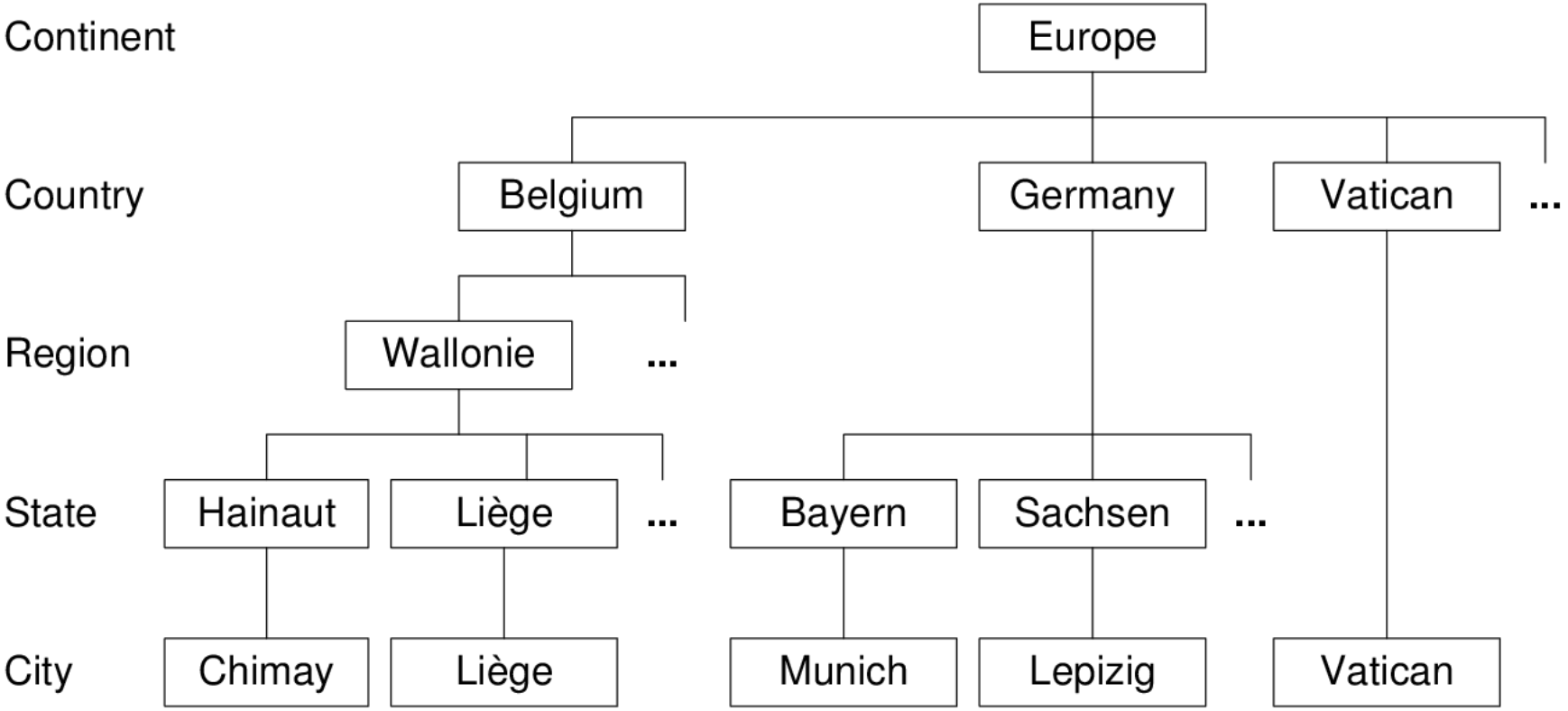
Types of hierarchy

- Generalized hierarchy
  - possibility of using extra foreign key to roll-up directly to upper level



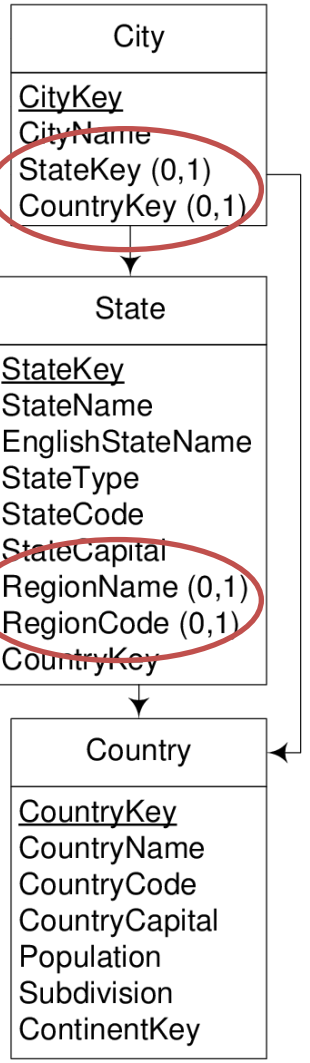
Types of hierarchy

- Ragged hierarchy
  - one or more levels can be skipped



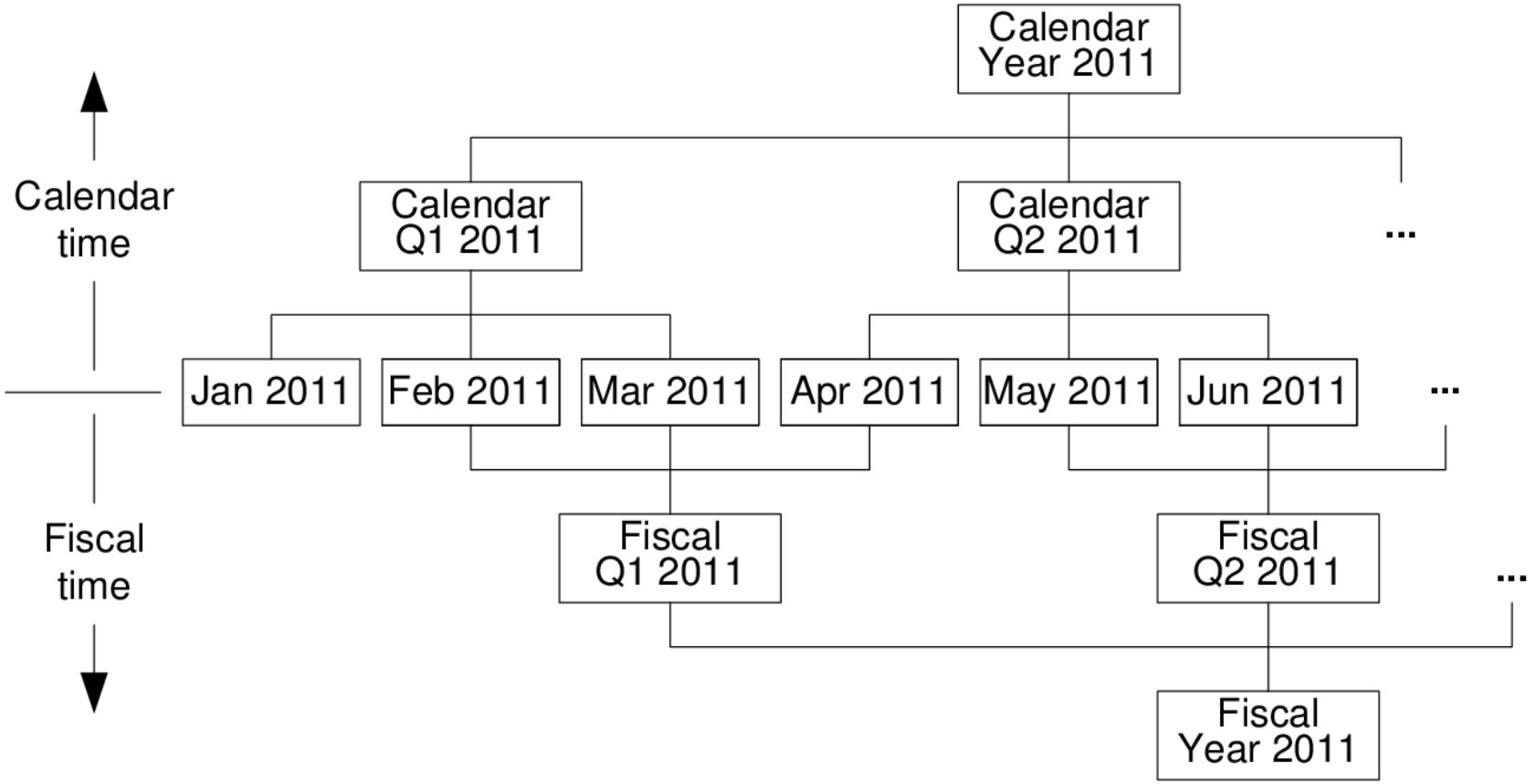
Types of hierarchy

- Ragged hierarchy
  - several implementations
    - add extra foreign keys to skip levels
    - or use optional attributes for the levels to skip
    - or use placeholders



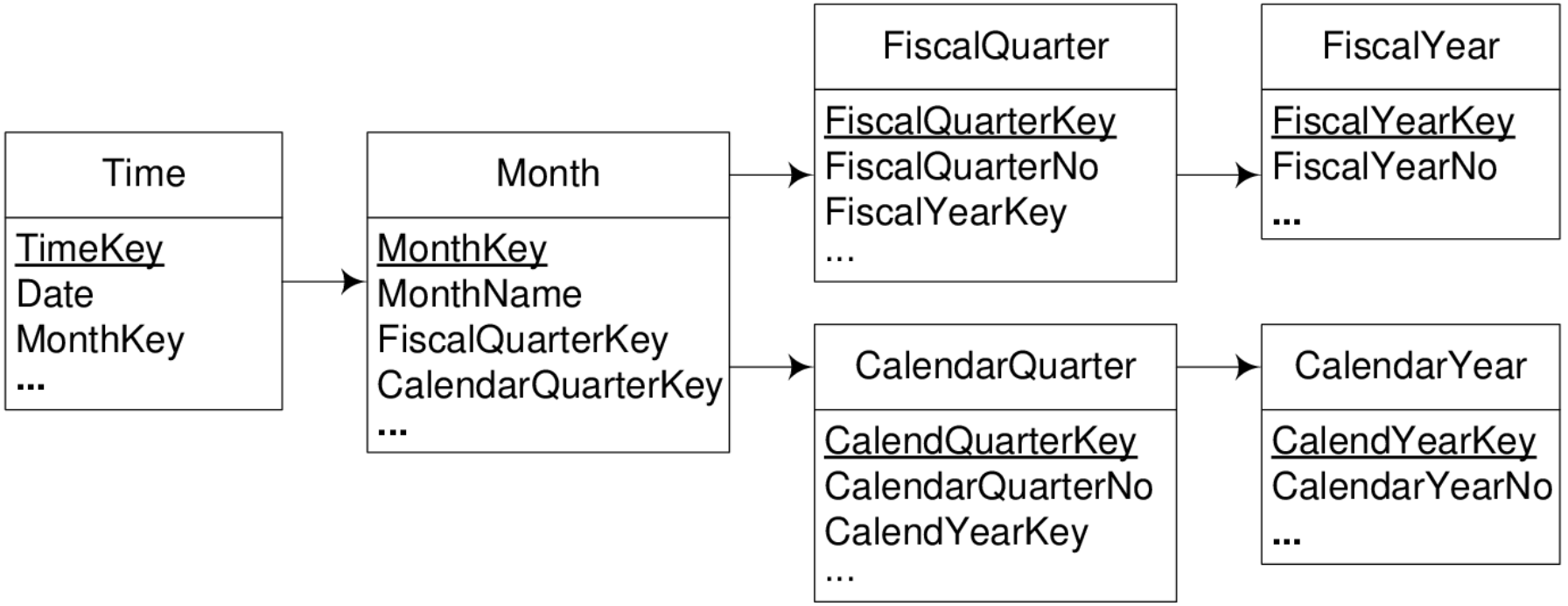
Types of hierarchy

- Alternative hierarchy
  - the same level has alternative aggregation paths



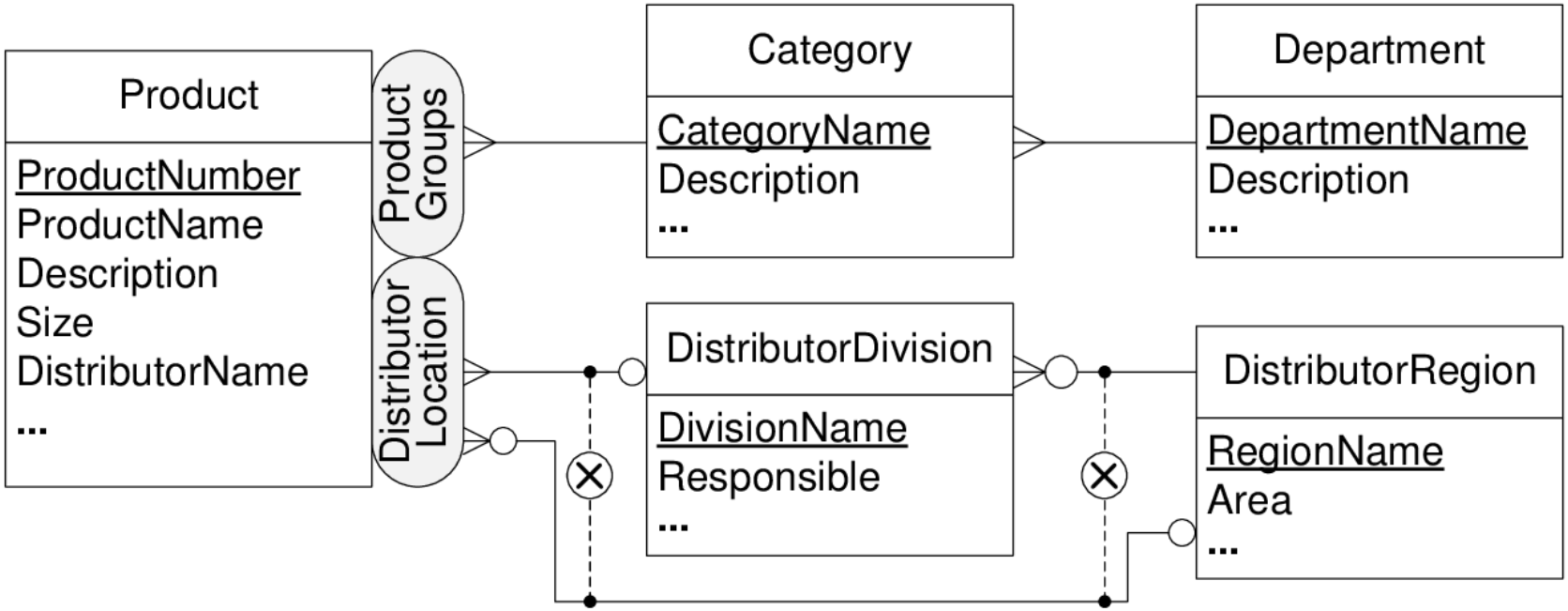
Types of hierarchy

- Alternative hierarchy
  - use snowflake structure



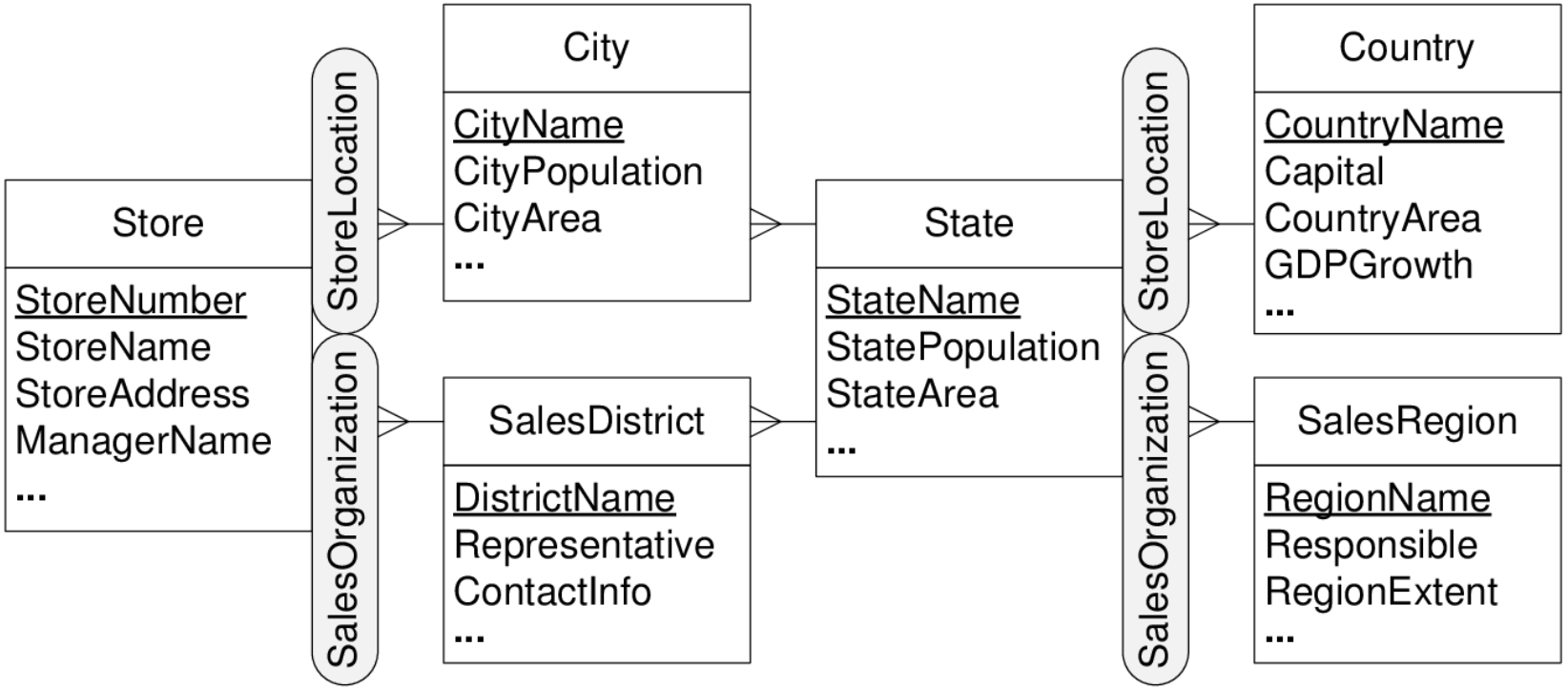
Types of hierarchy

- Parallel hierarchies
  - the same dimension has several hierarchies
    - example of two parallel **independent** hierarchies



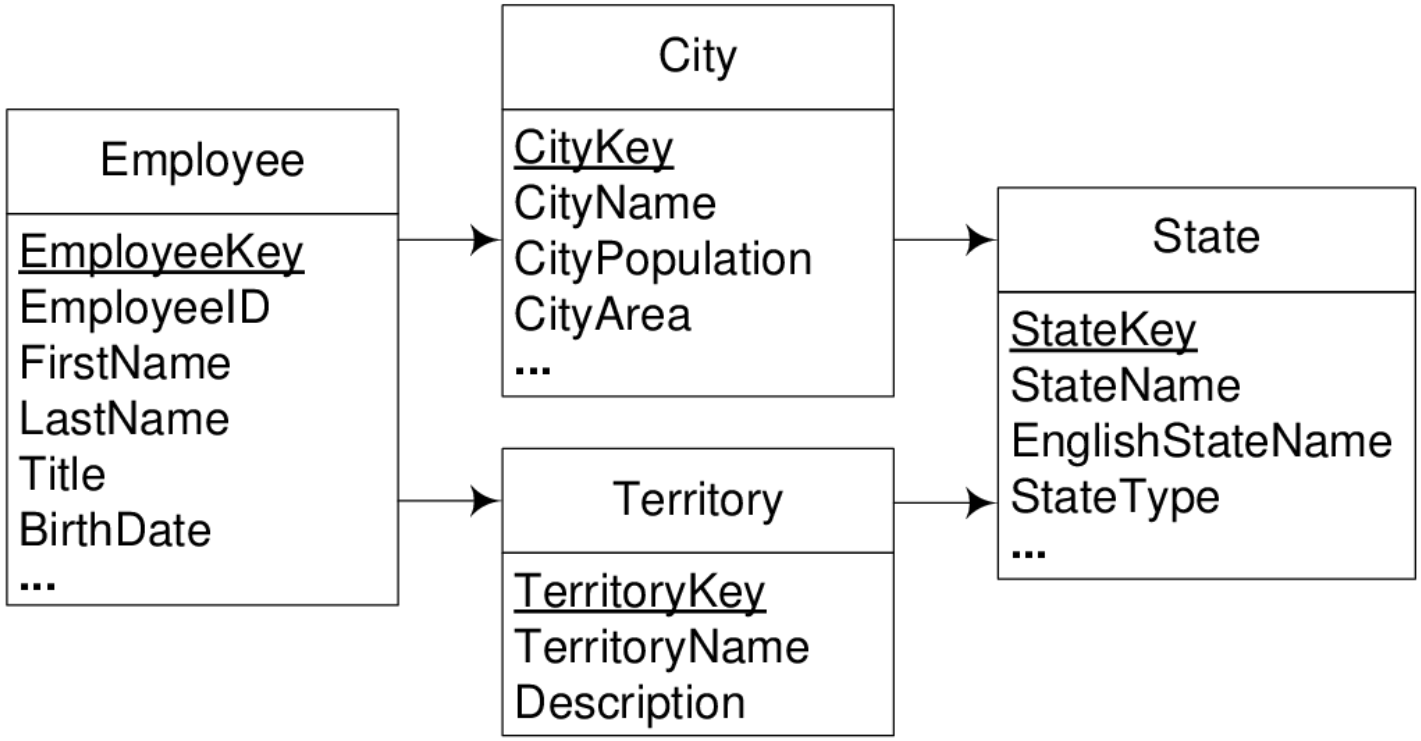
Types of hierarchy

- Parallel hierarchies
  - the same dimension has several hierarchies
    - example of two parallel **dependent** hierarchies



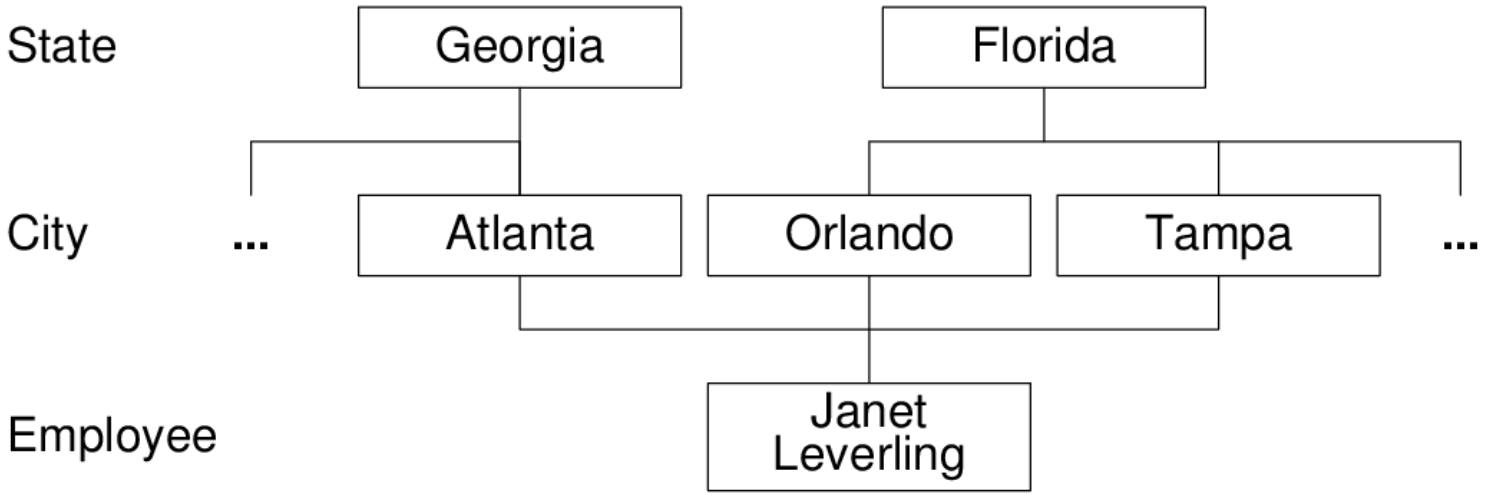
Types of hierarchy

- Parallel hierarchies
  - use snowflake structure



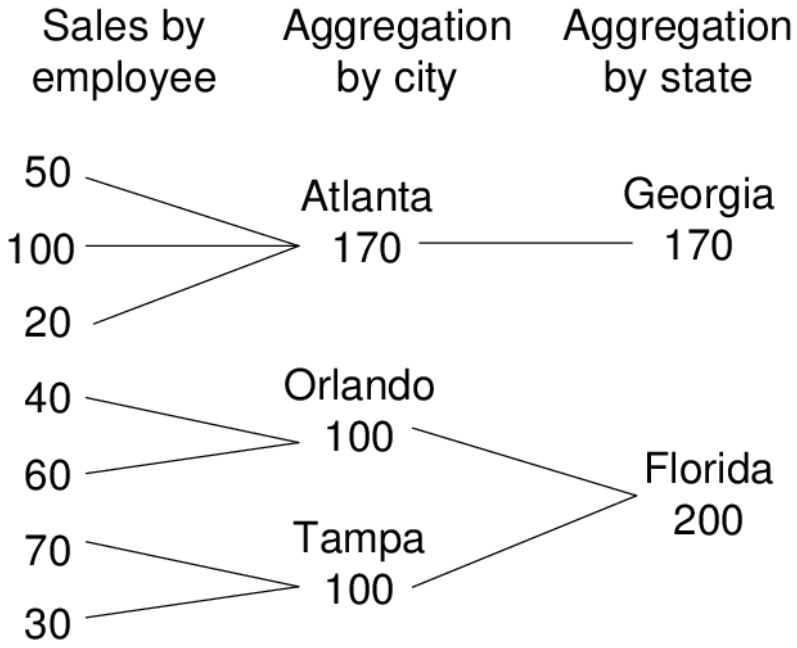
Types of hierarchy

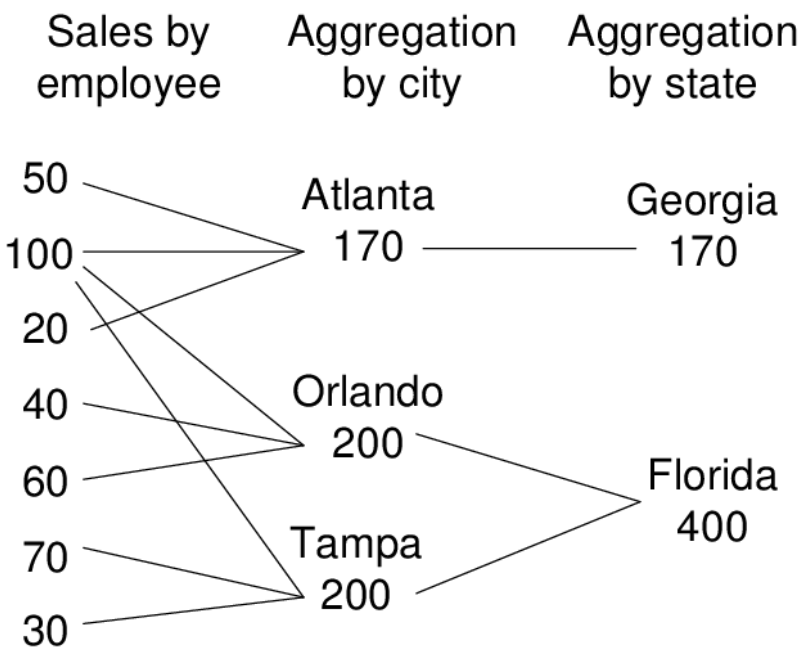
- Non-strict hierarchy
  - a member may have several parents
    - e.g. an employee that works in multiple cities
    - e.g. a week that belongs to two months



Types of hierarchy

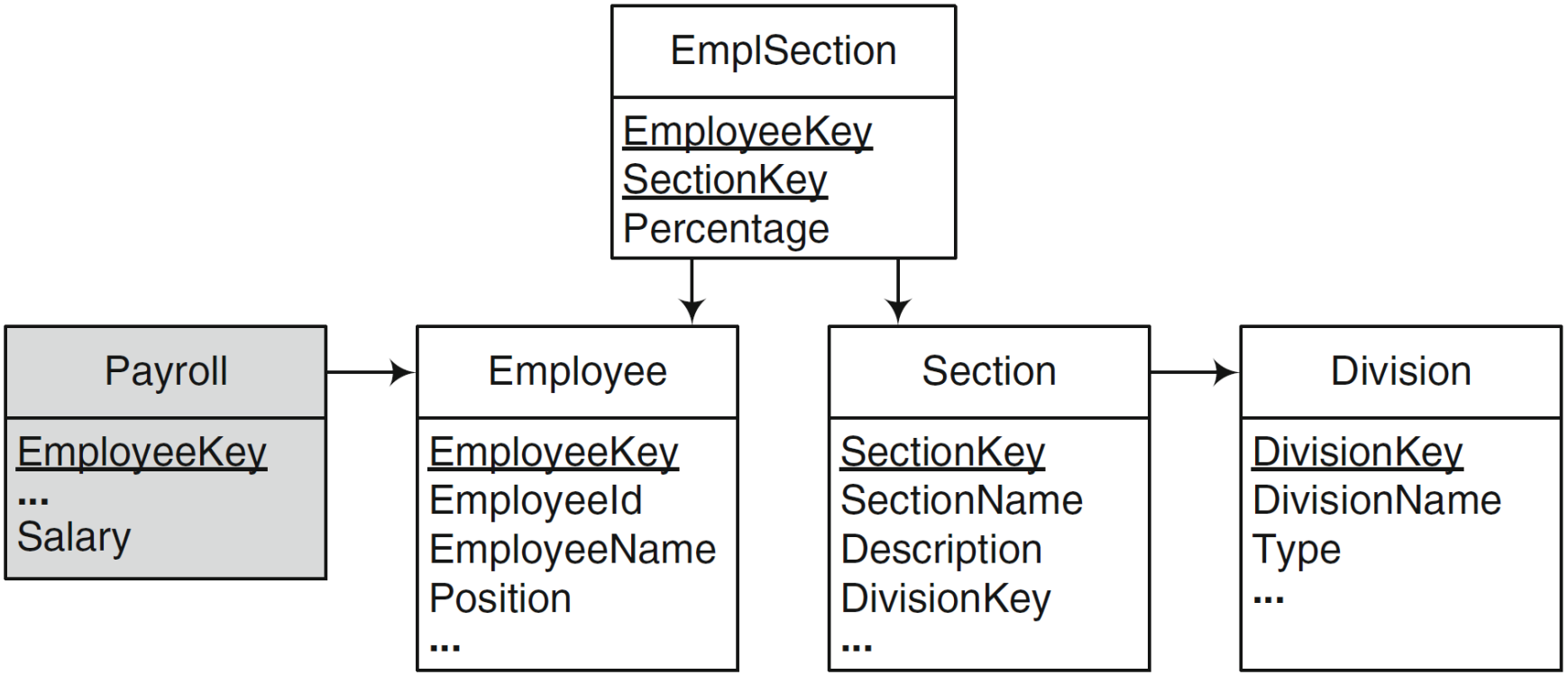
- Non-strict hierarchy
  - must be careful to avoid **double counting** on roll-up





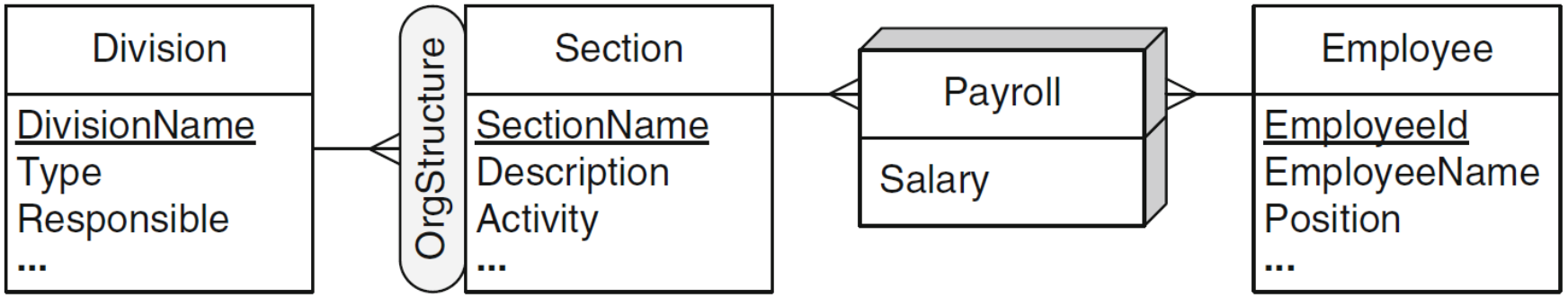
Types of hierarchy

- Non-strict hierarchy
  - use **bridge table** with percentage (%)
    - distributing attribute



Types of hierarchy

- Non-strict hierarchy
  - another solution is to re-design the DW schema using two separate dimensions



Measures

- Each measure is associated to an **aggregation function** that combines several values into a single one
  - the aggregation takes place whenever we change to a different level in a dimension hierarchy
- When defining a measure we must decide the associated aggregation function
  - **sum** is the most typical, but it may not always apply
  - some aggregation functions may not apply to a measure, or to a measure on a certain dimension

Measures

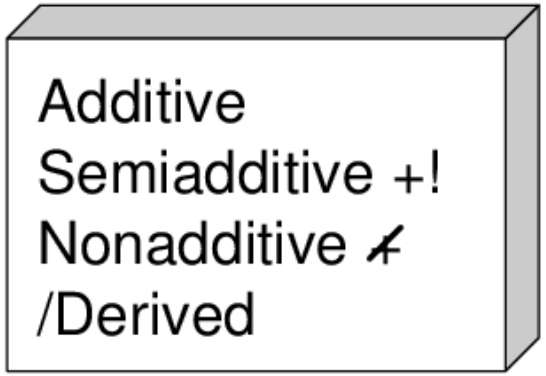
- Additive measures
  - can be aggregated along all dimensions using addition (sum)
    - e.g. sales amount along customer, product and time
- Semi-additive measures
  - can be added along some, but not all dimensions
    - e.g. inventory level cannot be summed along time
- Non-additive measures
  - cannot be added along any dimension
    - e.g. unit price, exchange rate

Measures

- What to do about semi- or non-additive measures
  - use other forms of aggregation
    - average (e.g. average inventory level over time)
    - minimum (e.g. minimum exchange rate over space or time)
    - maximum (e.g. maximum unit price over space or time)

Measures

- Derived measures
  - can be calculated from other measures or attributes
    - e.g. two measures: **sales amount** and **tax amount**
    - then **net amount** can be derived as a third measure
      net amount = **sales amount** – **tax amount**

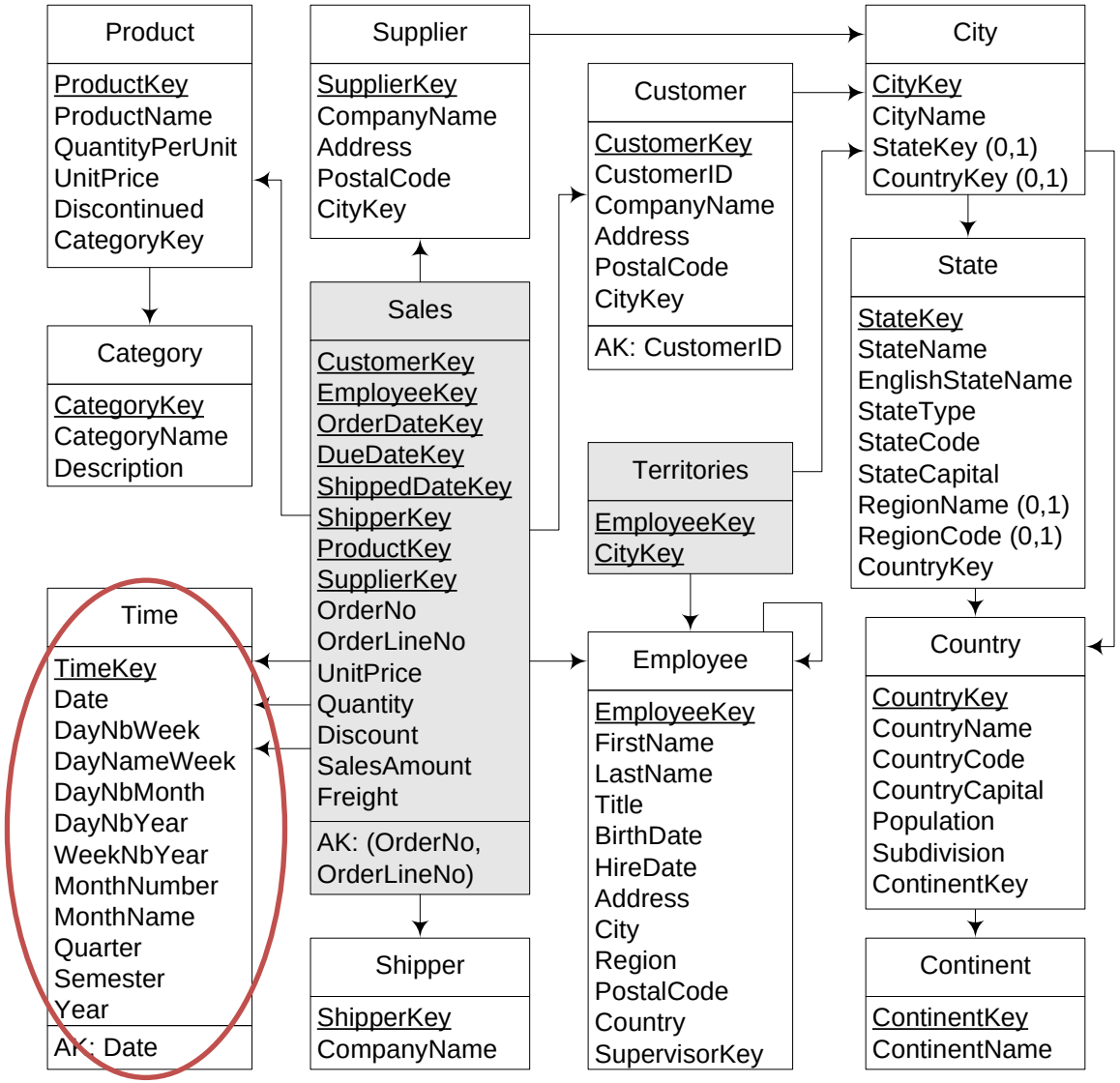


A. Vaisman, E. Zimányi, *Data Warehouse Systems: Design and Implementation*, Springer, 2014

Time dimension

- A data warehouse is a historical database so the time dimension is present in almost all DWs
  - in star/snowflake schema, time is included both as a foreign key in the fact table and as a time dimension containing the associated hierarchy levels
- In transactional databases, time information is stored in attributes of a DATE data type
  - e.g. weekend is computed on-the-fly using appropriate functions
- In a data warehouse, time information is stored explicitly in multiple attributes in the time dimension
  - easier to compute queries, e.g. total sales during weekends

Time dimension



Time dimension

- Granularity of time dimension depends on use
  - if interested in monthly data only, define the time dimension with granularity of month
  - time dimension with a month granularity spanning 5 years will have 5*12 = 60 tuples
  - if the granularity is second, then the dimension time will have: 5*12*30*24*3600 = 155 520 000 tuples
- Time dimension may have more than one hierarchy
  - e.g. fiscal and calendar year
- Time dimension can often be populated automatically