

ISIT312 Big Data Management

# Conceptual Data Warehouse Design

Dr Fenghui Ren

School of Computing and Information Technology -  
University of Wollongong

# Conceptual Data Warehouse Design

## Outline

[MultiDim: A Conceptual Model for Data Warehouses](#)

[MultiDim Model: Notation](#)

[Dimension Hierarchies](#)

# MultiDim: A Conceptual Multidimensional Model

## Conceptual data models

- Allow better communication between designers and users to understand application requirements
- More stable than implementation-oriented (logical) schema, which changes with the platform
- Provide better support for visual user interfaces

No well-established conceptual model for multidimensional data

Several proposals based on UML, on the ER model, or using specific notations

Problems:

- Cannot express complex kinds of hierarchies
- Lack of a mapping to the implementation platform

# MultiDim: A Conceptual Multidimensional Model

Currently, data warehouses are designed using mostly logical models (star and snowflake schemas)

- Difficult to express requirements (technical knowledge required)
- Limit users to defining only elements that the underlying implementation systems can manage

**MultiDim** data model is based on the entity-relationship model

Includes concepts like:

- dimensions
- hierarchies
- facts
- measures

Supports various kinds of hierarchies existing in real-world applications

Can be mapped to star or snowflake relational structures

# Conceptual Datawarehouse Design

## Outline

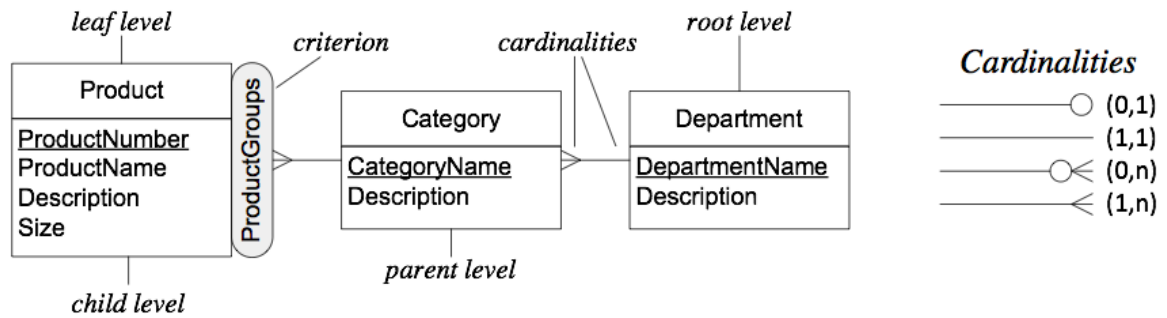
[MultiDim: A Conceptual Model for Data Warehouses](#)

[MultiDim Model: Notation](#)

[Dimension Hierarchies](#)

# MultiDim Model: Notation

A graphical notation used for a sample **hierarchy**



**Dimension:** level or one or more **hierarchies**

**Hierarchy:** several related levels

**Level:** entity type

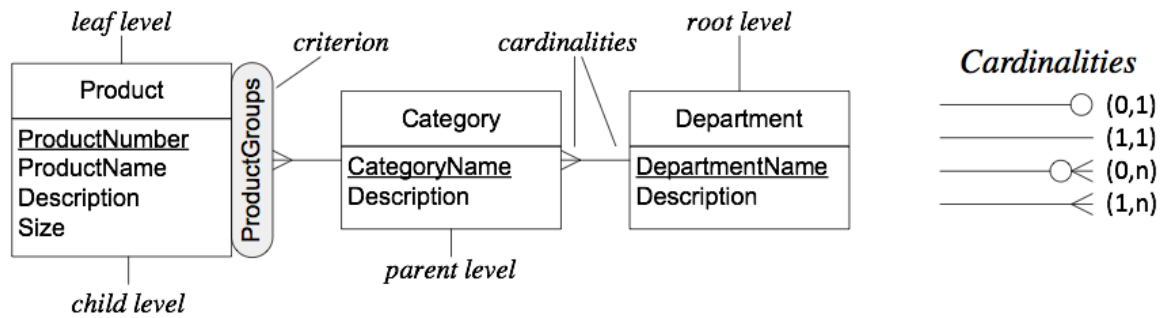
**Member:** every instance of a level

**Child and parent levels:** the lower and higher levels

**Leaf and root levels:** first and last levels in a hierarchy

# MultiDim Model: Notation

A graphical notation used for a sample **hierarchy**



**Cardinality:** minimum/maximum numbers of members in a level related to members in another level

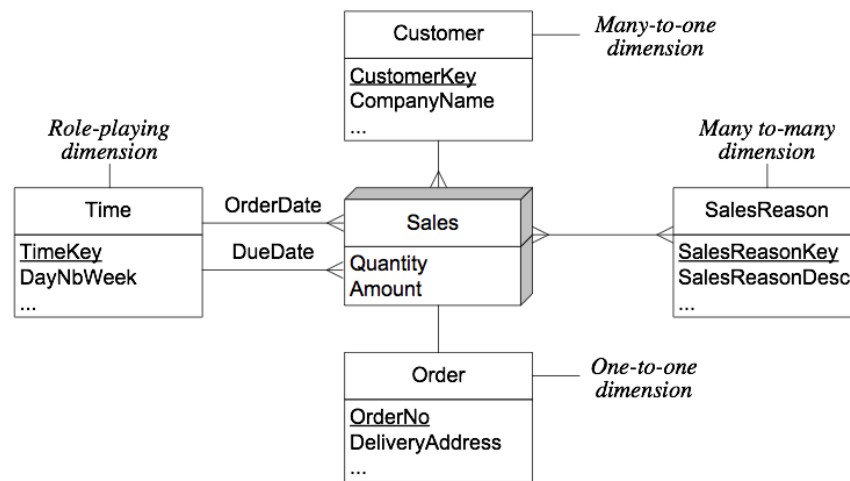
**Criterion:** expresses different hierarchical structures used for analysis

**Key attribute:** indicates how child members are grouped

**Descriptive attributes:** describe characteristics of members

# MultiDim Model: Notation

A sample **fact** with 5 **dimensions**



**Fact**: relates measures to leaf levels in dimensions

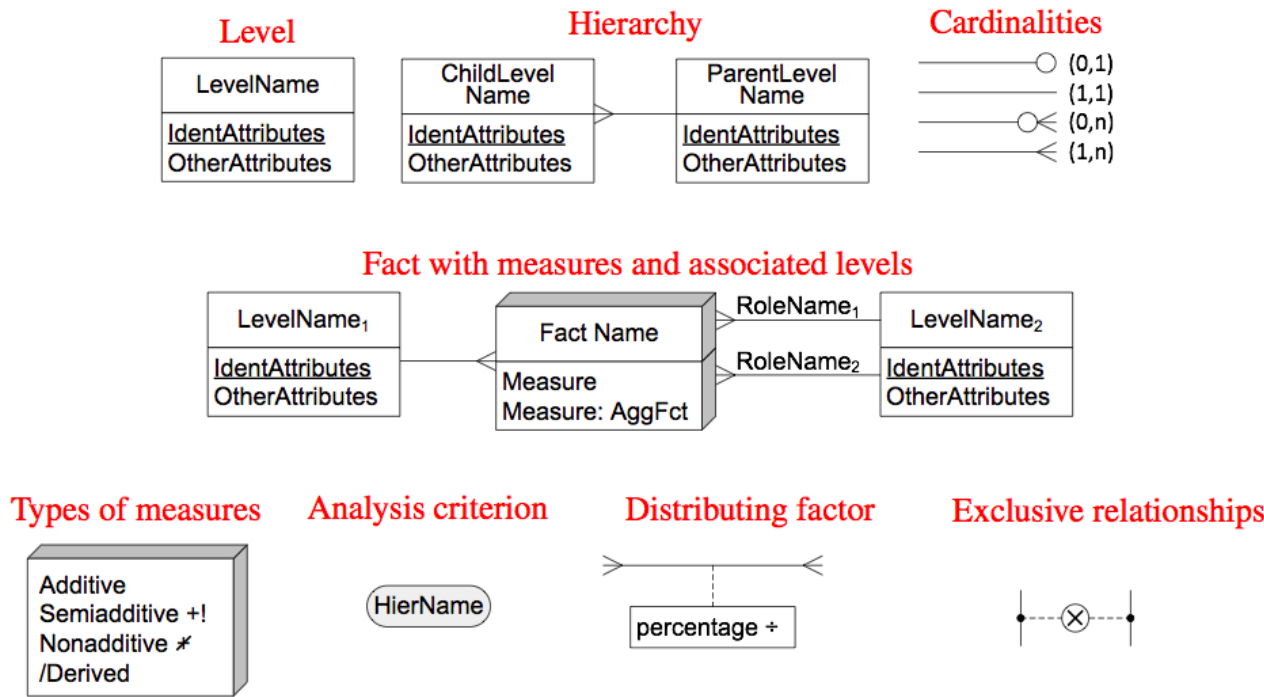
Dimensions can be related to fact with **one-to-one**, **one-to-many**, of **many-to-many**

Dimension can be related several times to a fact with **different roles**

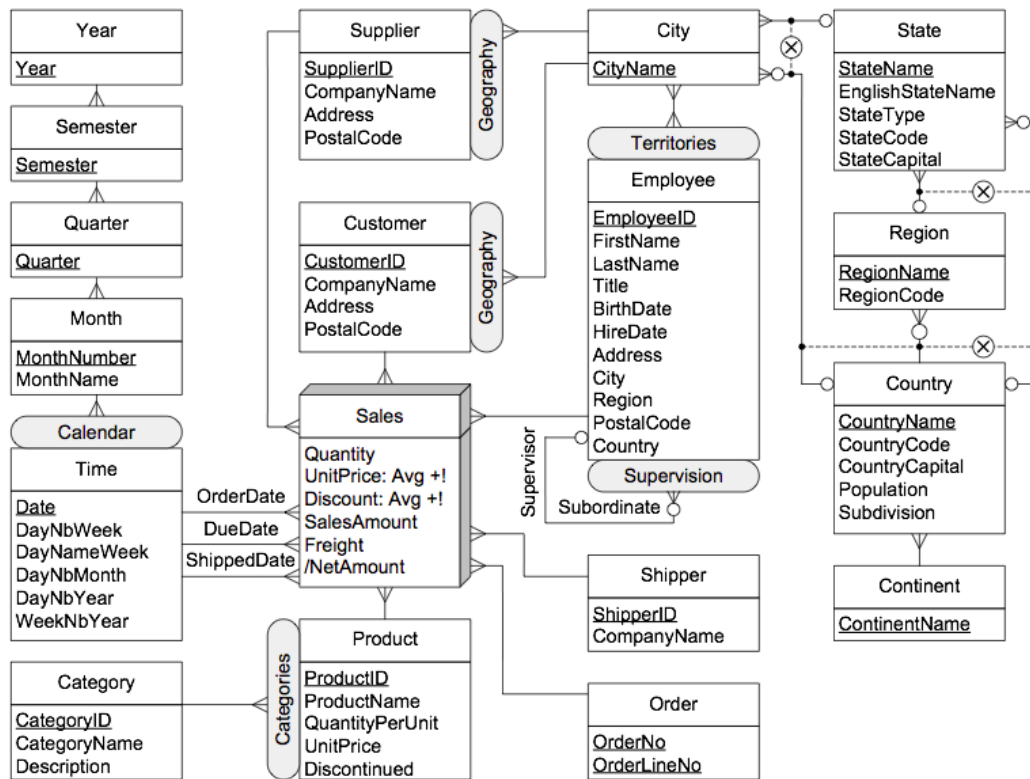


# MultiDim Model: Notation

## Summary



# MultiDim Conceptual Schema of the Northwind Data Warehouse



# Conceptual Data Warehouse Design

## Outline

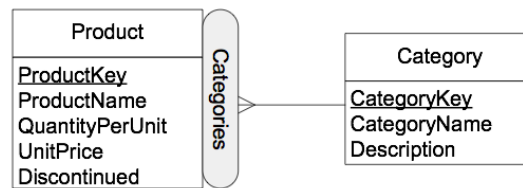
[MultiDim: A Conceptual Model for Data Warehouses](#)

[MultiDim Model: Notation](#)

[Dimension Hierarchies](#)

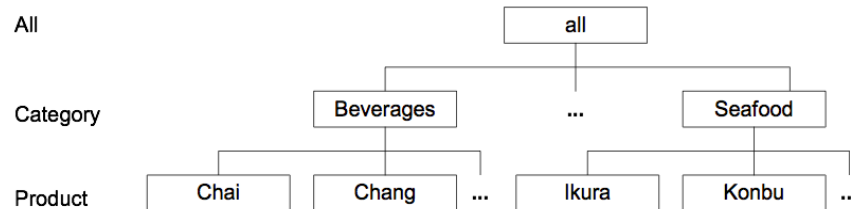
# Balanced Hierarchies

At **schema level**: only one path where all parent-child relationships are many-to-one and mandatory



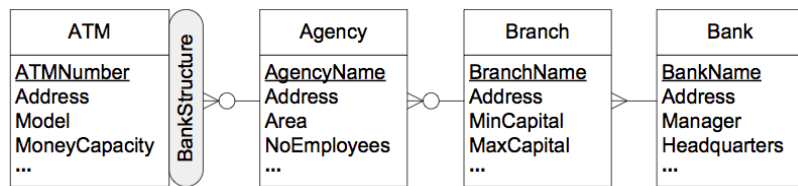
At **instance level**: members form a balanced tree (all the branches have the same length)

All parent members have at least one child member, and a child belongs exactly to one parent

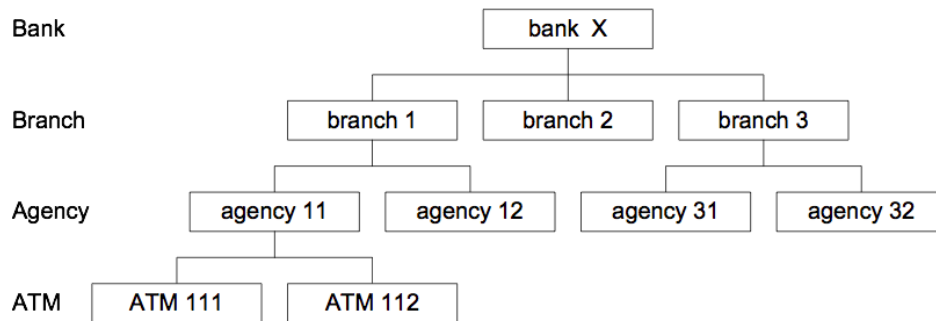


# Unbalanced Hierarchies

At **schema level**: one path where all parent-child relationships are many-to-one, but some are optional



At **instance level**: members form a unbalanced tree



# Recursive Hierarchies

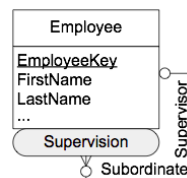
A special case of unbalanced hierarchies

The **same level** is linked by the two roles of a parent-child relationship

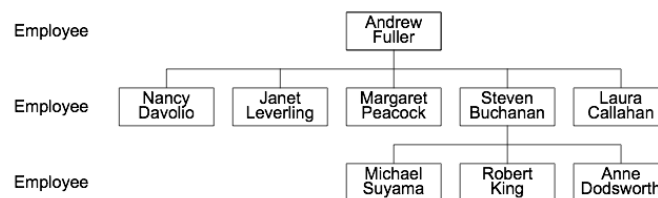
Used when all hierarchy levels express the same semantics

The characteristics of the parent and child are similar (or the same)

## Schema level



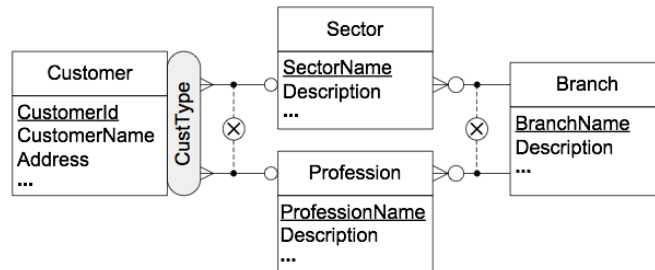
## Instance level



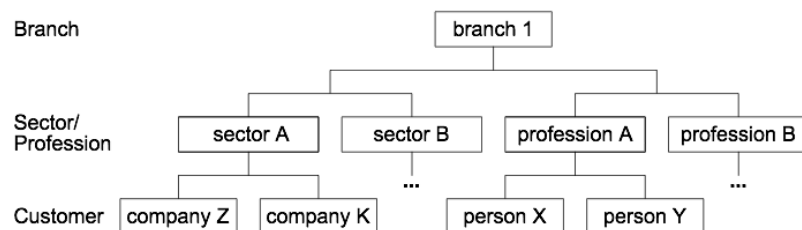
# Generalized Hierarchies

At **schema level**: multiple exclusive paths sharing at least the leaf level;  
may also share other levels

Two aggregation paths, one for each type of customer



At **instance level**: each member belongs to only one path

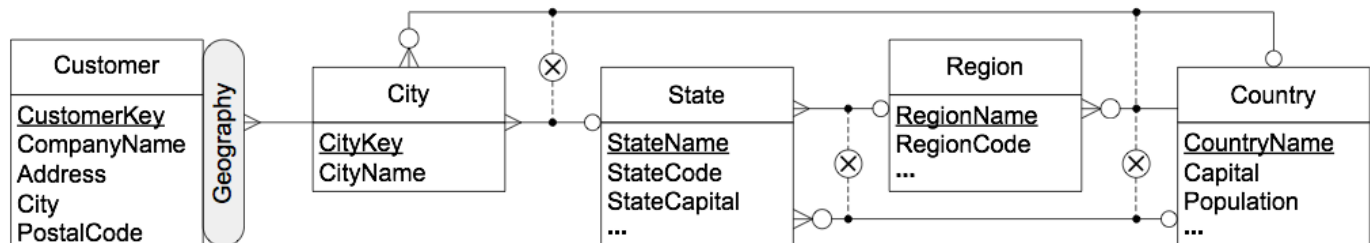


# Noncovering Hierarchies

Also known as **ragged** or **level-skipping hierarchies**

A **special case of generalized hierarchies**

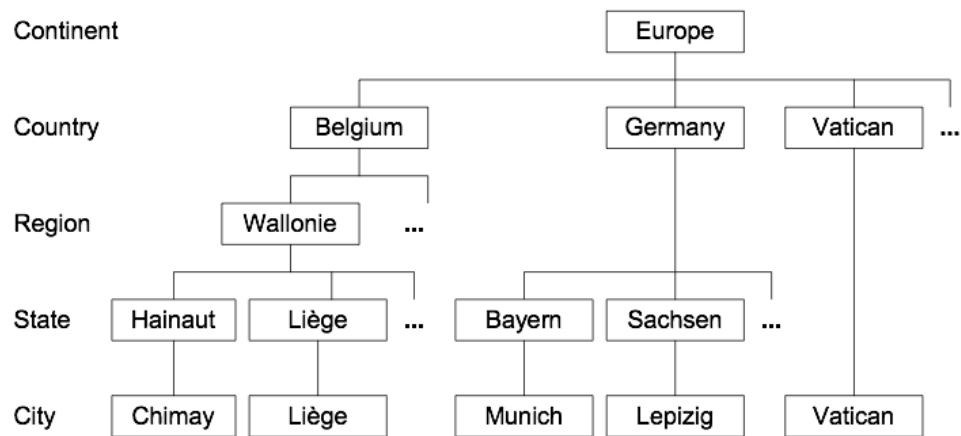
At the **schema level**: Alternative paths are obtained by skipping one or several intermediate levels





# Noncovering Hierarchies

At **instance level**: Path length from the leaves to the same parent can be different for different members



# References

A. VAISMAN, E. ZIMANYI, Data Warehouse Systems: Design and Implementation, Chapter 4 Conceptual Data Warehouse Design, Springer Verlag, 2014