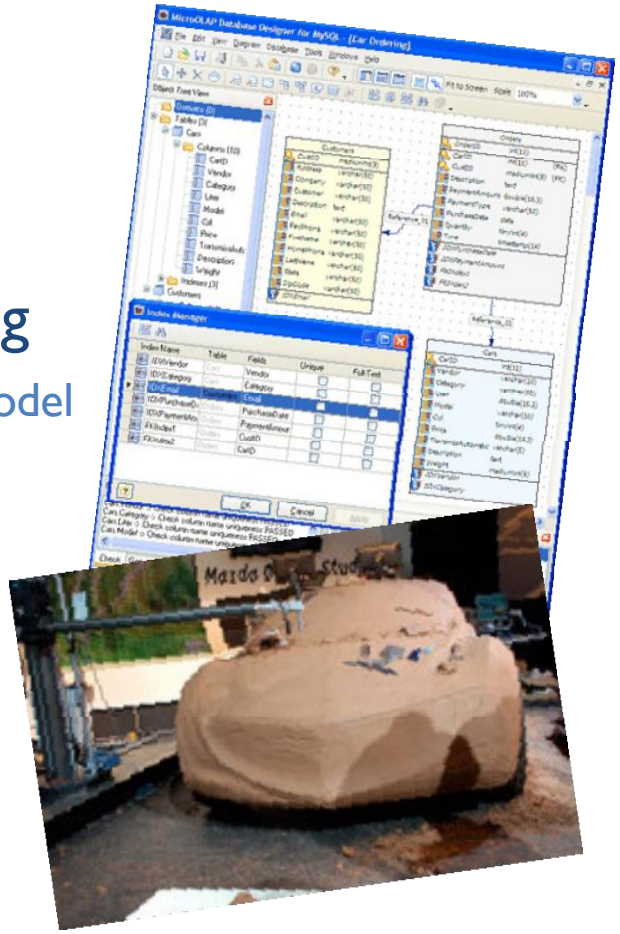
Lecture 05

Data Warehouse Architecture (cont'd.)

Logical Model

Summary – last week

- Last week:
  - Storage structures: MDB
  - Architectures: N-Tier Architectures
  - Data Modeling – Conceptual Modeling
    - Multidimensional Entity Relationship (ME/R) Model



- This week:
  - Logical Model

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

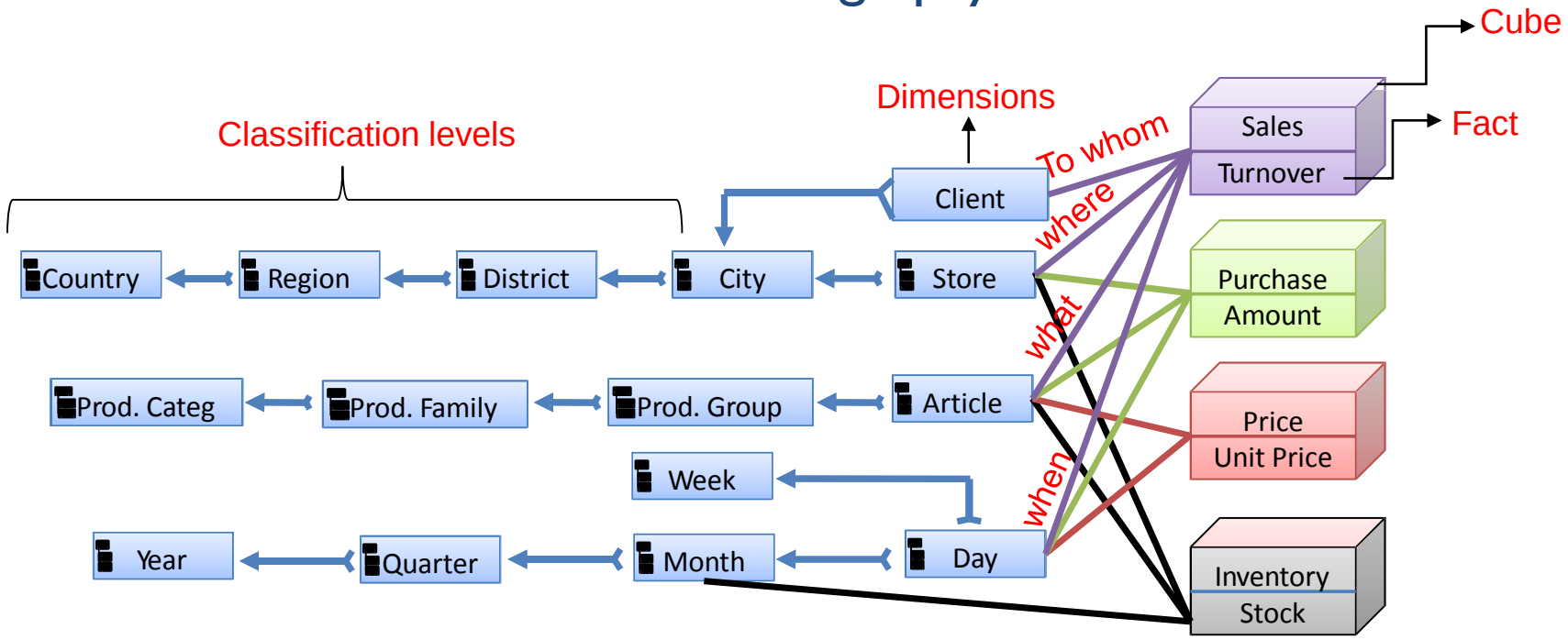
Logical Model

- Goal of the Logical Model
  - Confirm the **subject areas**
  - Create 'real' **facts** and **dimensions** from the subjects that we have identified
  - Establish the needed **granularity** for our dimensions

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Logical Model (cont'd.)

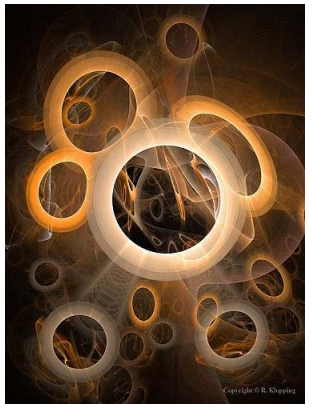
- **Logical structure** of the multidimensional model
  - Cubes: Sales, Purchase, Price, Inventory
  - Dimensions: Product, Time, Geography, Client

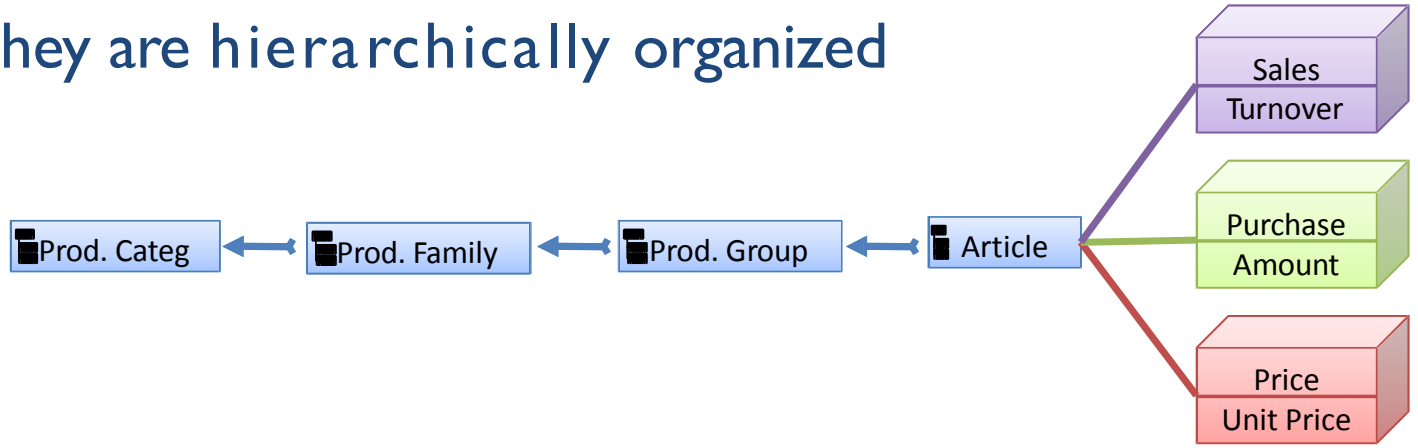


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Dimensions

- **Dimensions are…**
  analysis purpose chosen **entities**, within the data model
  - One dimension can be used to define **more than one cube**
  - They are hierarchically organized

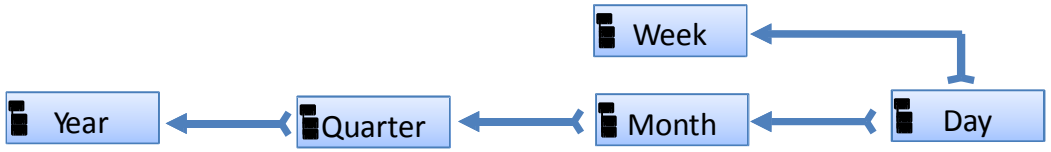




Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Dimensions (cont'd.)

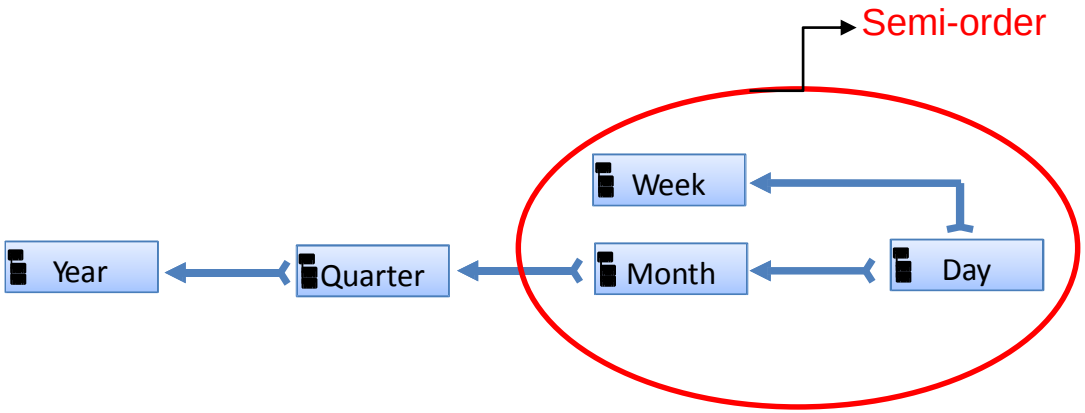
- Dimension hierarchies are organized in **classification levels** (e.g., Day, Month, …)
  - The dependencies between the classification levels are described by the **classification schema** through **functional dependencies**
    - An attribute B is functionally dependent on an attribute A, denoted $A \longrightarrow B$, if for all $a \in dom(A)$ there exists exactly one $b \in dom(B)$ corresponding to it



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

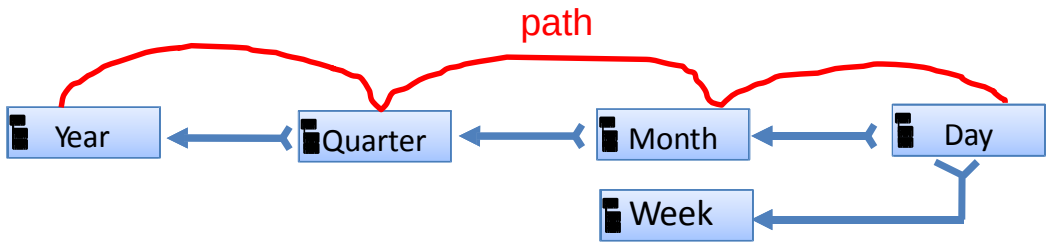
Dimensions (cont'd.)

- Classification schemas
  - The classification schema of a dimension D is a semi-ordered set of **classification levels** ($\{D.K_0, \ldots, D.K_k\}$, $\longrightarrow$)
  - With a **smallest element** $D.K_0$, i.e. there is no classification level with smaller granularity



Dimensions (cont'd.)

- A **fully-ordered** set of classification levels is called a **Path**
  - If we consider the **classification schema** of the time dimension, then we have the following paths
    - T.Day → T.Week
    - T.Day → T.Month → T.Quarter → T.Year



  - Here T.Day is the **smallest element**

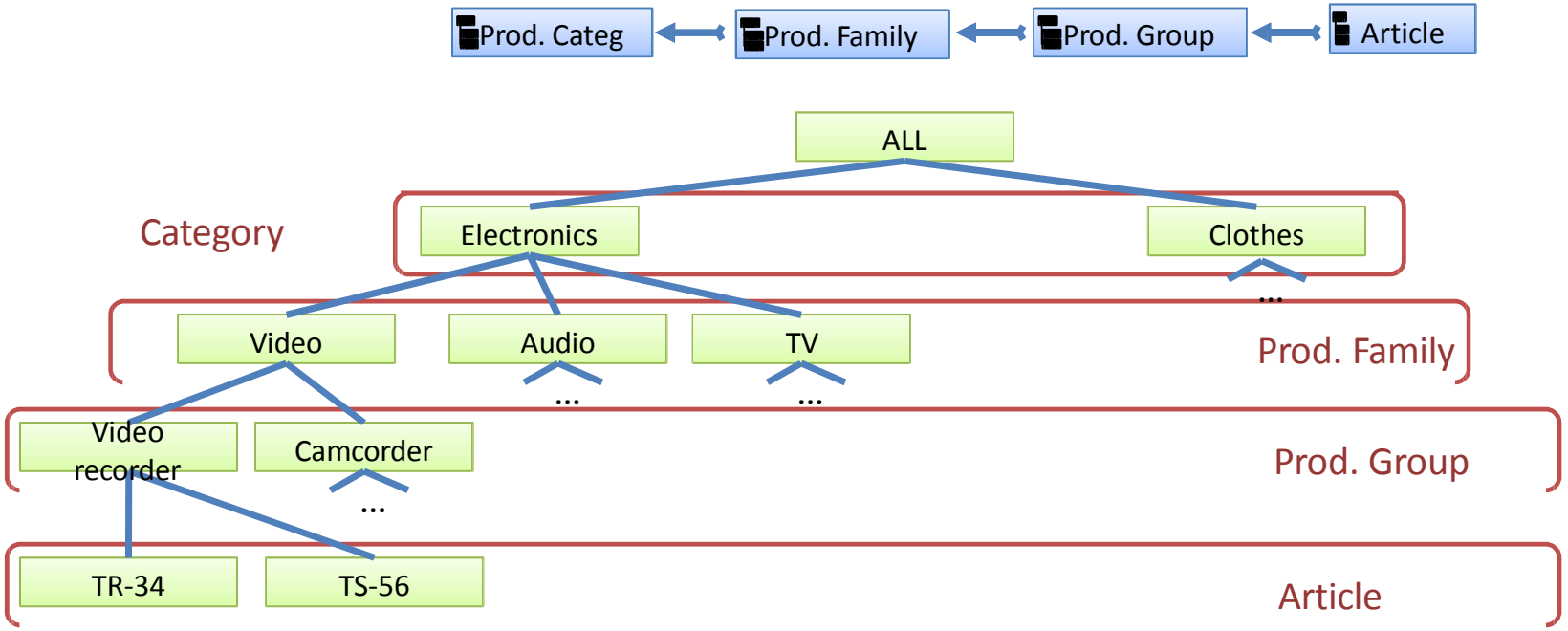
Dimensions (cont'd.)

- Classification hierarchies
  - Let $D.K_0 \longrightarrow \ldots \longrightarrow D.K_k$ be a path in the classification schema of dimension D
  - A classification hierarchy concerning these path is a balanced tree which
    - Has as nodes $dom(D.K_0) \cup \ldots \cup dom(D.K_k) \cup \{ALL\}$
    - And its edges respect the functional dependencies

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Dimensions (cont'd.)

- **Example**: classification hierarchy from the path product dimension



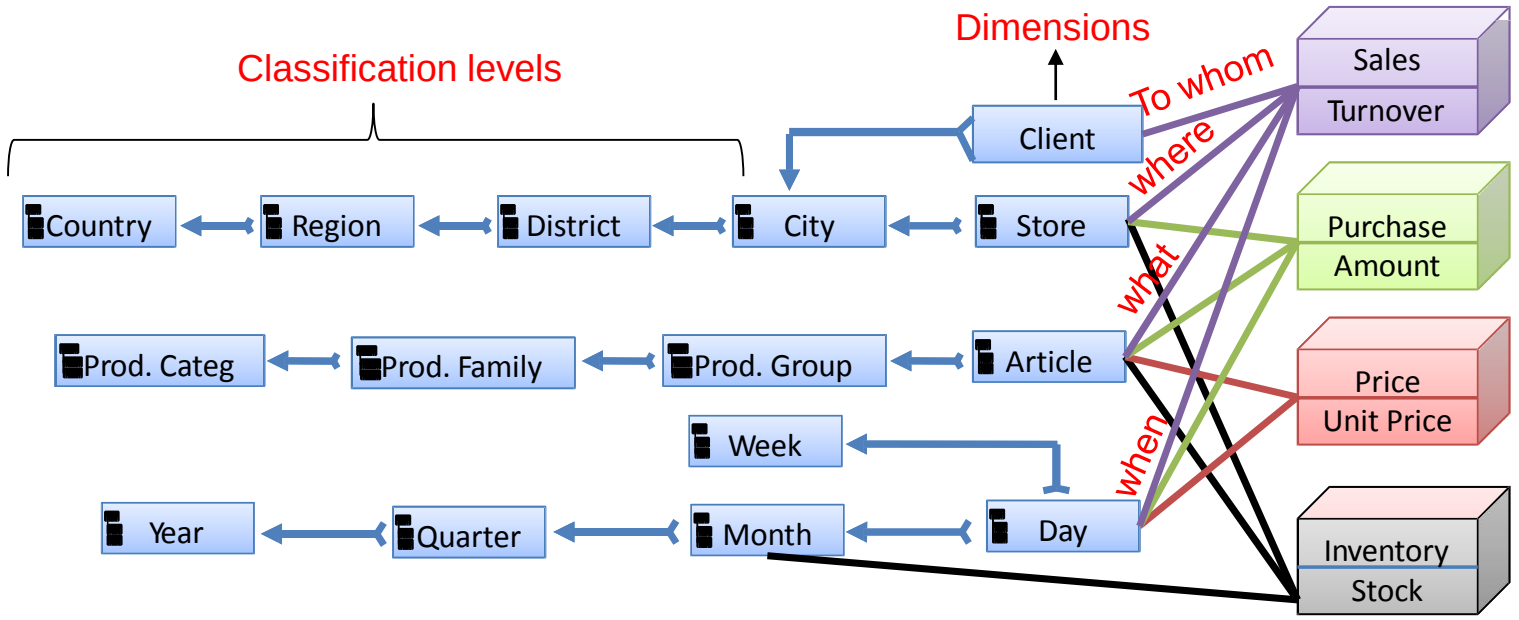
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Cubes

- **Cubes** consist of data cells with one or more measures
- If a **cube schema** $S(G,M)$ consists of a granularity $G= (D_1.K_1, ..., D_n.K_n)$ and a set $M= (M_1, ..., M_m)$ representing the **measure**
  - A **Cube** (C) is a set of cube cells, $C \subseteq dom(G) \times dom(M)$

Cubes (cont'd.)

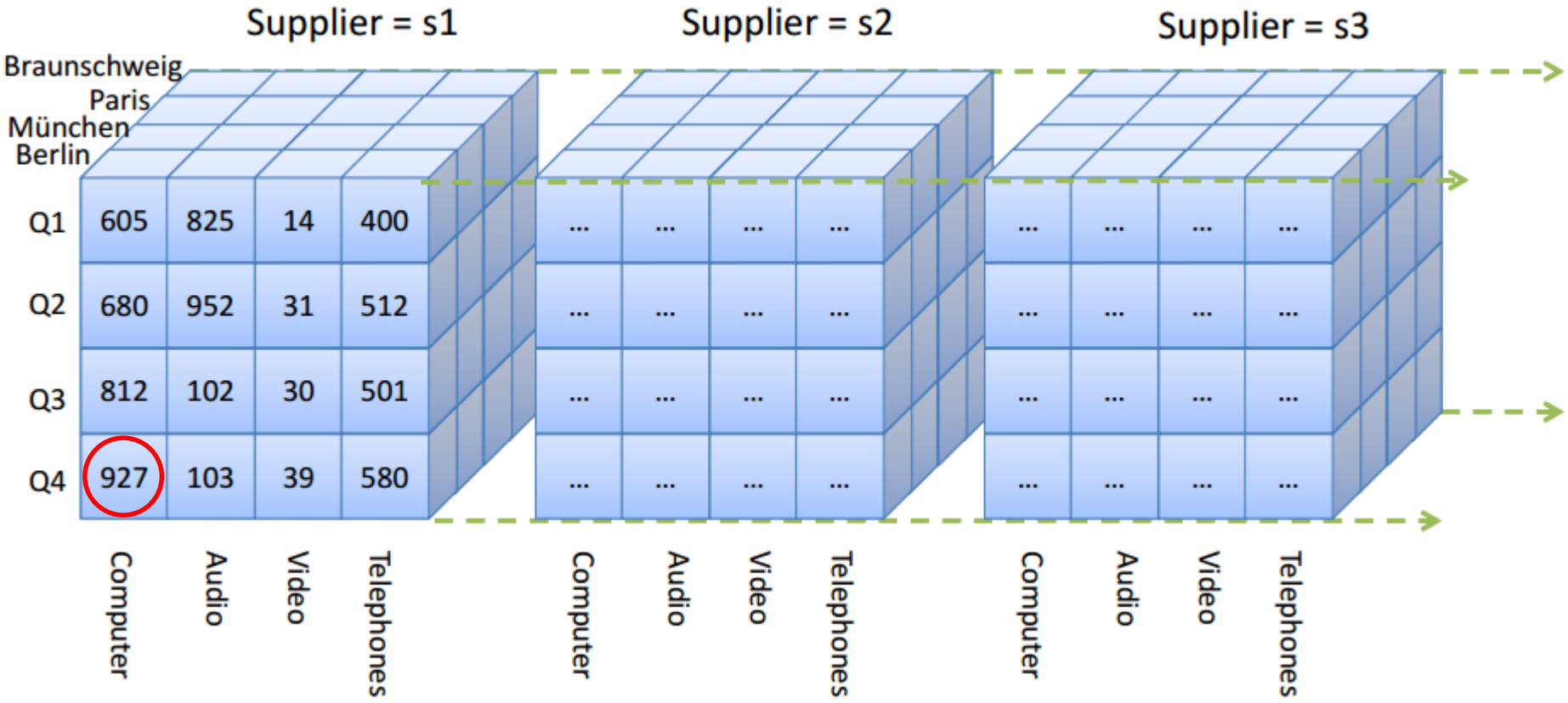
- The **coordinates** of a cell are the classification nodes from dom(G) corresponding to the cell
  - Sales ((Article, Day, Store, Client), (Turnover))
  - …



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

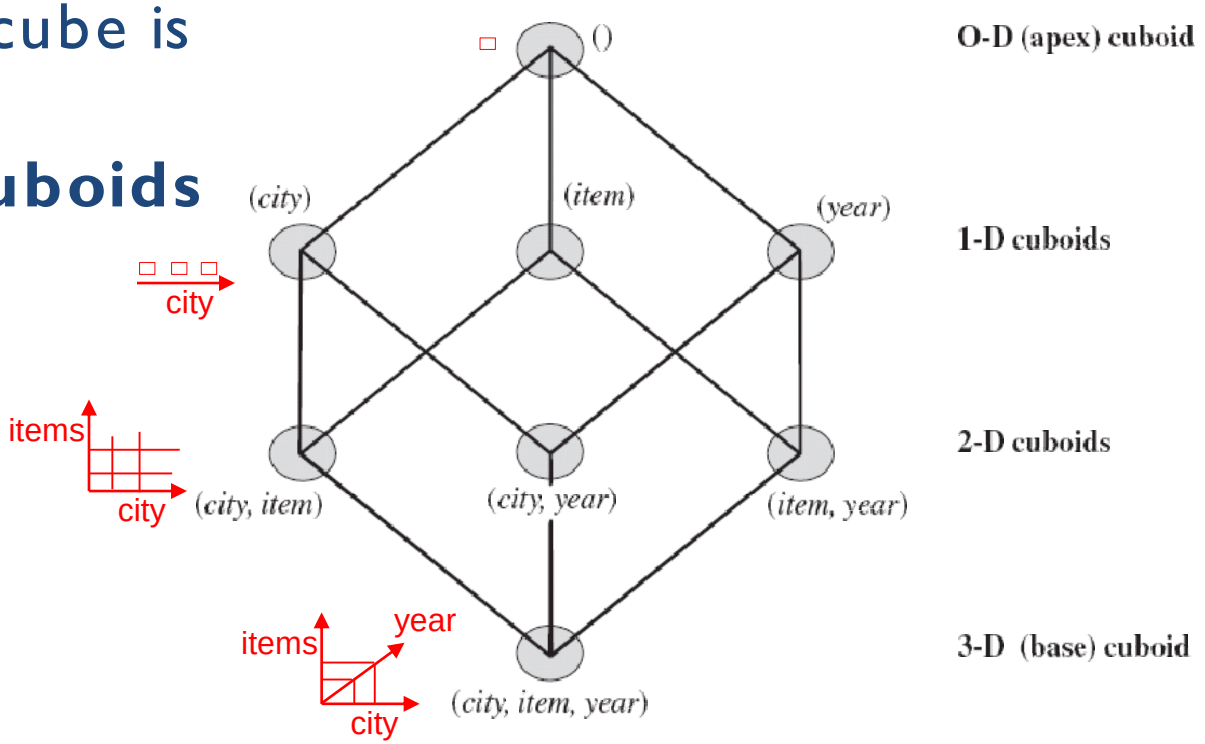
Cubes (cont'd.)

- 4 dimensions (supplier, city, quarter, product)



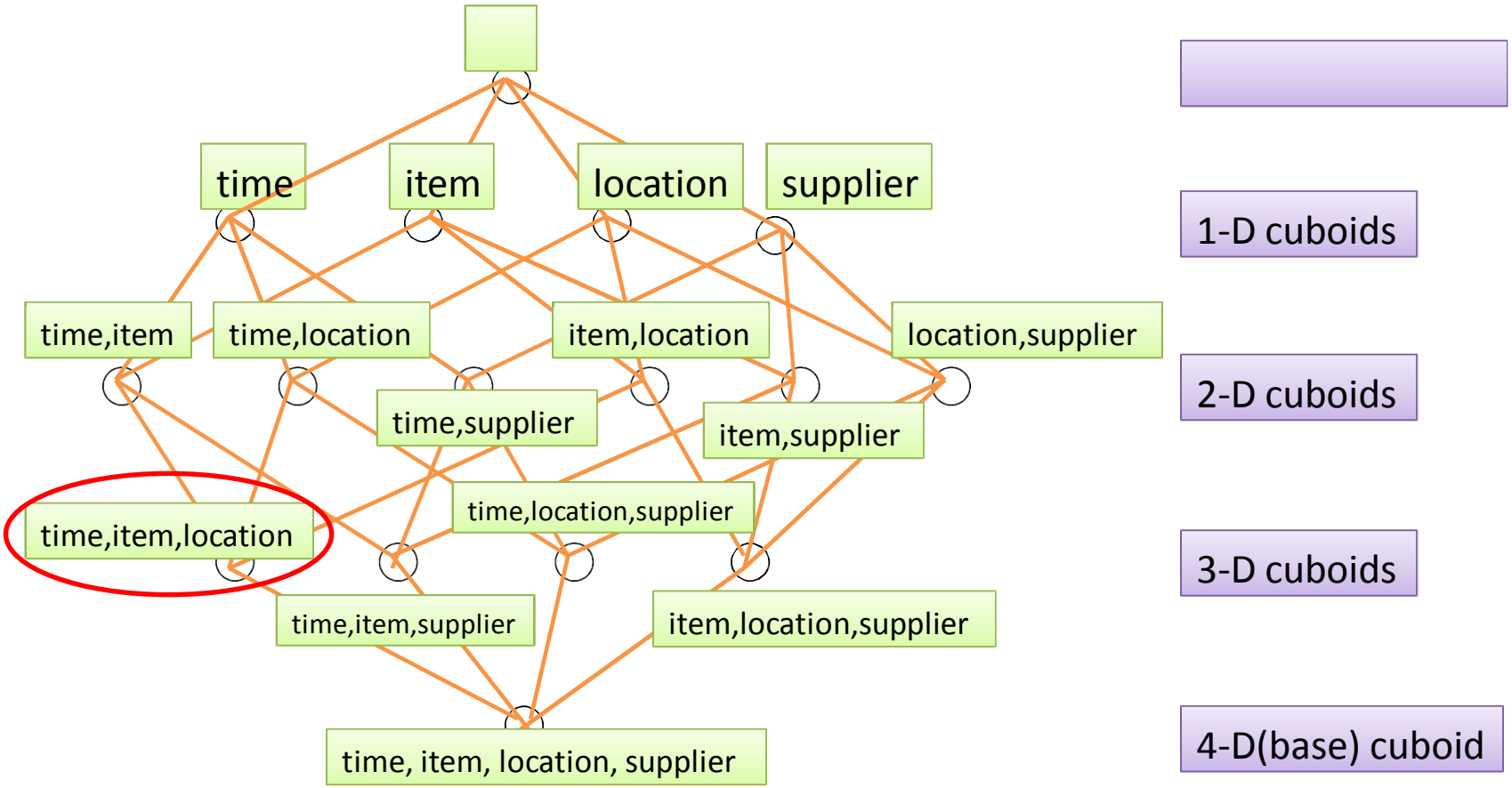
Cubes (cont'd.)

- We can now imagine **n-dimensional cubes**
  - n-D cube is called a **base cuboid**
  - The top most cuboid, the 0-D, which holds the highest level of summarization is called **apex cuboid**
  - The full data cube is formed by the **lattice of cuboids**



Cubes (cont'd.)

- But things can get complicated pretty fast



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Basic Operations

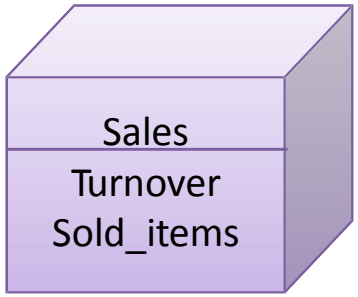
- Basic operations of the multidimensional model on the logical level
  - Selection
  - Projection
  - Cube join
  - Sum
  - Aggregation

Basic Operations (cont'd.)

- Multidimensional Selection
  - The **selection** on a cube $C((D_1.K_1,..., D_g.K_g), (M_1, ..., M_m))$ through a predicate P, is defined as $\sigma_P(C) = \{z \in C:P(z)\}$, if all variables in P are either:
    - Classification levels K, which functionally depend on a classification level in the granularity of K, i.e. $D_i.K_i \longrightarrow K$
    - Measures from $(M_1, ..., M_m)$
  - E.g. $\sigma_{P.Prod\_group= \text{“Video”}}(Sales)$

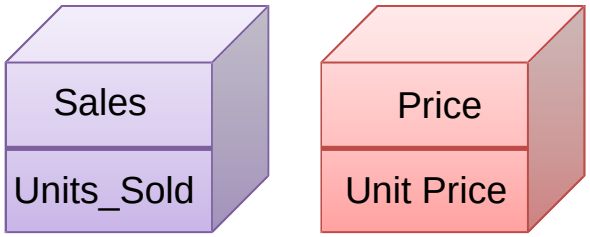
Basic Operations (cont'd.)

- Multidimensional projection
  - The **projection** of a function of a measure $F(M)$ of cube C is defined as
    - $\pi_{F(M)}(C) = \{ (g,F(m)) \in dom(G) \times dom(F(M)) : (g,m) \in C \}$
  - E.g. projection $\pi_{turnover,\ sold\_items}(Sales)$



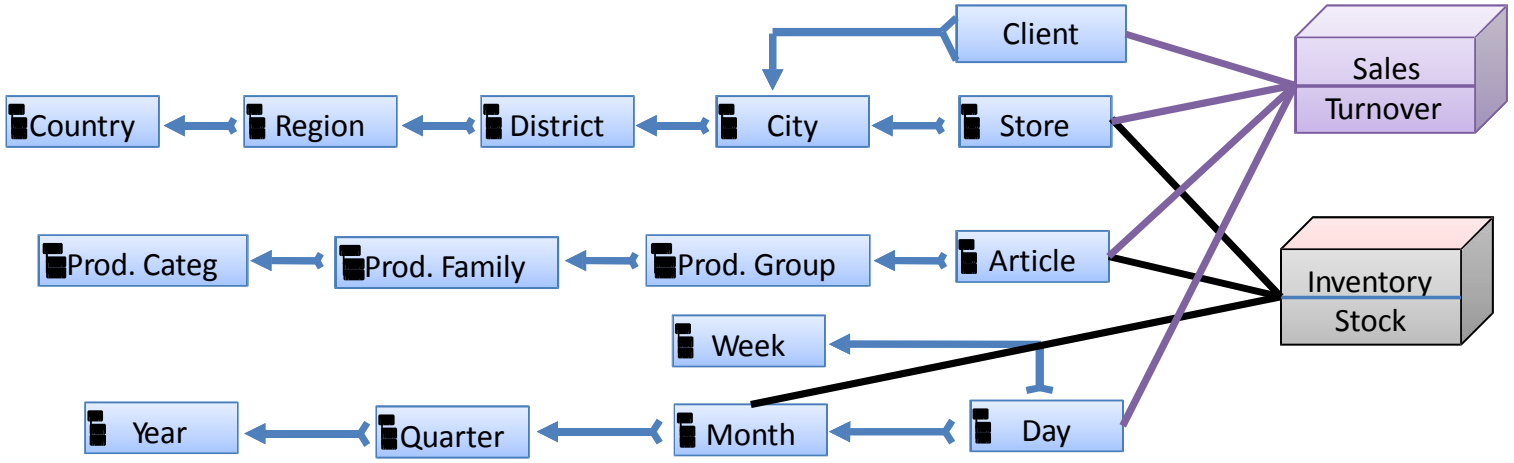
Basic Operations (cont'd.)

- Join operations between cubes is usual
  - E.g. if turnover would not be provided, it could be calculated with the help of the unit price from the price cube
- 2 cubes $C_1(G_1, M_1)$ and $C_2(G_2, M_2)$ can only be joined, if they have the same granularity $(G_1 = G_2 = G)$
  - $C_1 \bowtie C_2 = C(G, M_1 u M_2)$



Basic Operations (cont'd.)

- When the granularities are different, but we still need to join the cubes, **aggregation** has to be performed
  - E.g., Sales ⋈ Inventory: aggregate Sales((Day, Article, Store, Client)) to Sales((Month, Article, Store, Client))



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Basic Operations (cont'd.)

- **Aggregation**: most important operation for OLAP operations
- Aggregation functions
  - Build a single values from set of value, e.g. in SQL: SUM, AVG, Count, Min, Max
  - Example: $SUM_{(P.Product\_group,\ G.City,\ T.Month)}(Sales)$

Change Support

Detour

- Classification schema, cube schema, classification hierarchy are all designed in the building phase and **considered as fix**
  - Practice has proven otherwise
    DW grow old, too
  - Changes are strongly connected to the time factor
  - This lead to the time validity of these concepts

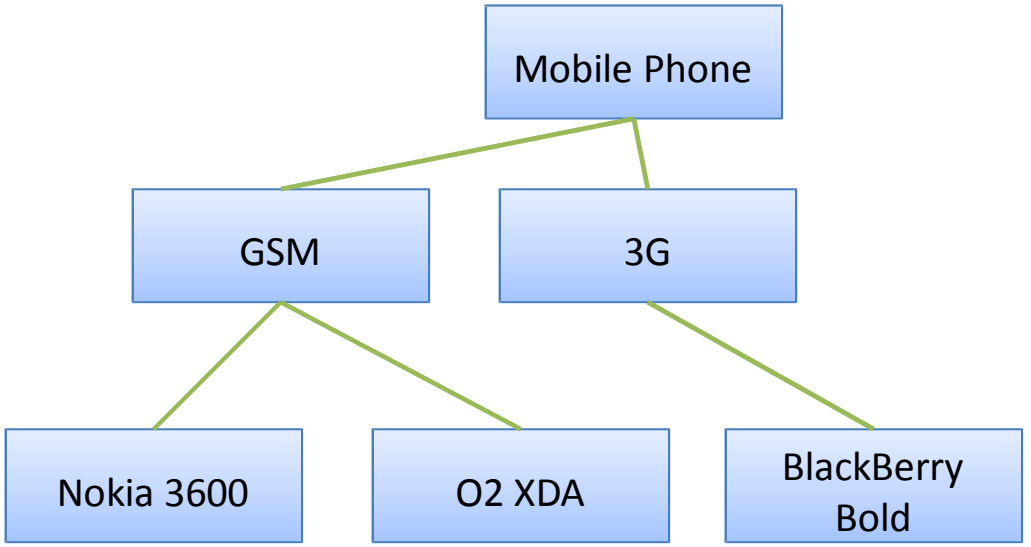
- Reasons for schema modification
  - New requirements
  - Modification of the data source

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy

Detour

- E.g. Saturn sells a lot of electronics
  - Lets consider mobile phones
    - They built their DW on 01.03.2003
    - A classification hierarchy of their data until 01.07.2008 could look like this:

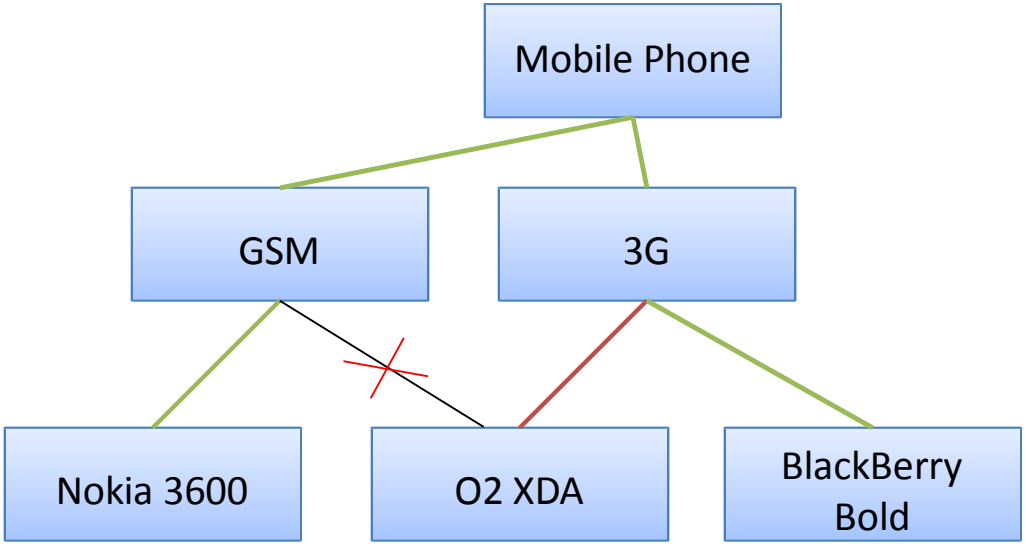


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- After 01.07.2008 3G becomes hip and affordable and many phone makers start migrating towards 3G capable phones
  - Lets say O2 makes its XDA 3G capable

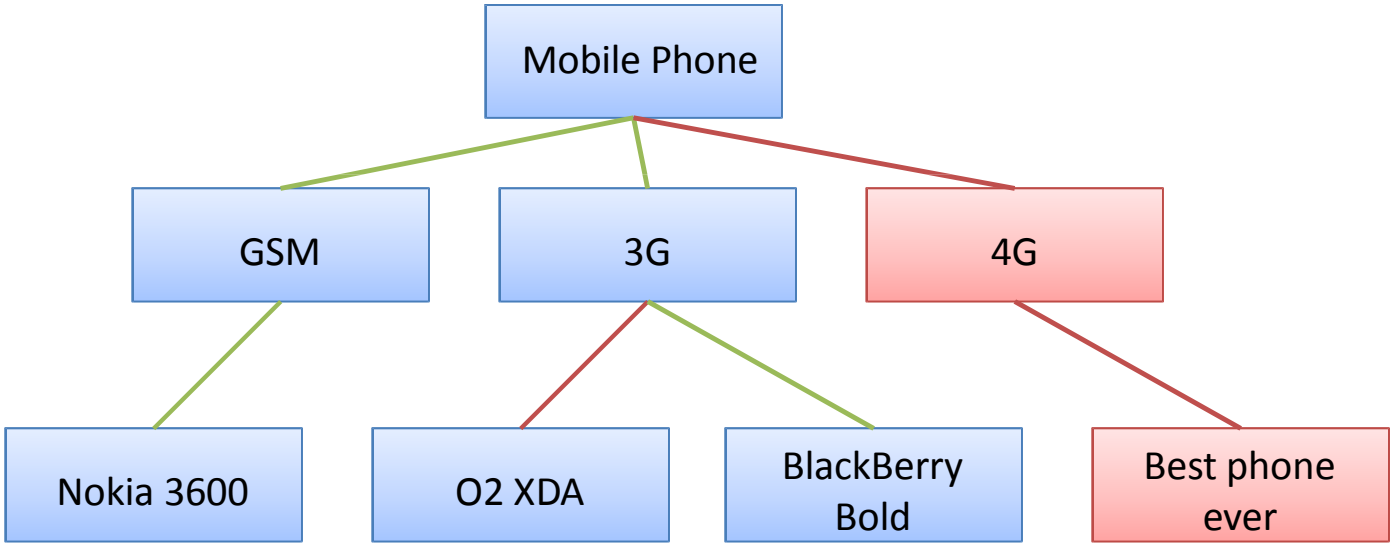


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- After 01.04.2010 phone makers already develop 4G capable phones

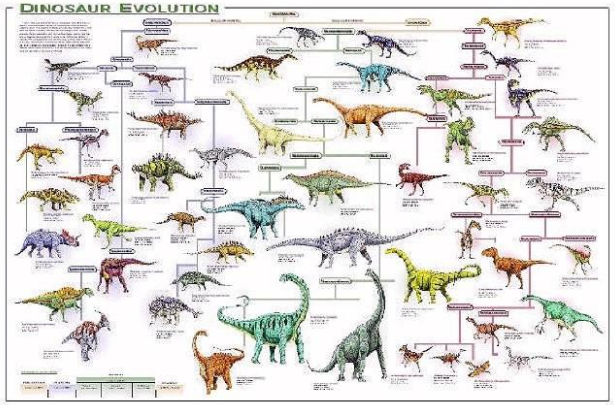


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- It is important to **trace the evolution** of the data
  - It can explain which data was available at which moment in time
  - Such a **versioning system** of the classification hierarchy can be performed by constructing a **validity matrix**
    - When is something, valid?
    - Use **timestamps** to mark it!

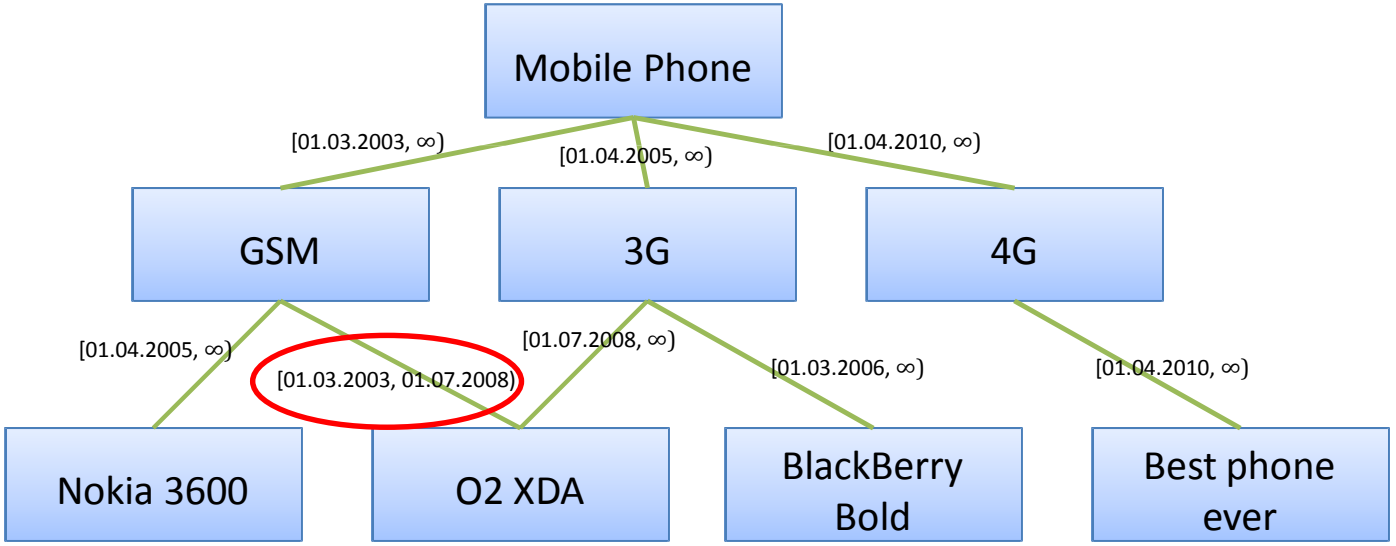


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- Annotated Change data



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

- The tree can be stored as dimension metadata
  - The storage form is a **validity matrix**
    - Rows are parent nodes
    - Columns are child nodes

	GSM	3G	4G	Nokia 3600	O2 XDA	Berry Bold	Best phone
Mobile phone	[01.03.2003, ∞)	[01.04.2005, ∞)	[01.04.2010, ∞)				
GSM				[01.04.2005, ∞)	[01.03.2003, 01.07.2008)		
3G					[01.07.2008, ∞)	[01.03.2006, ∞)	
4G							[01.04.2010, ∞)
Nokia 3600							
O2 XDA							
Berry Bold							
Best phone							

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

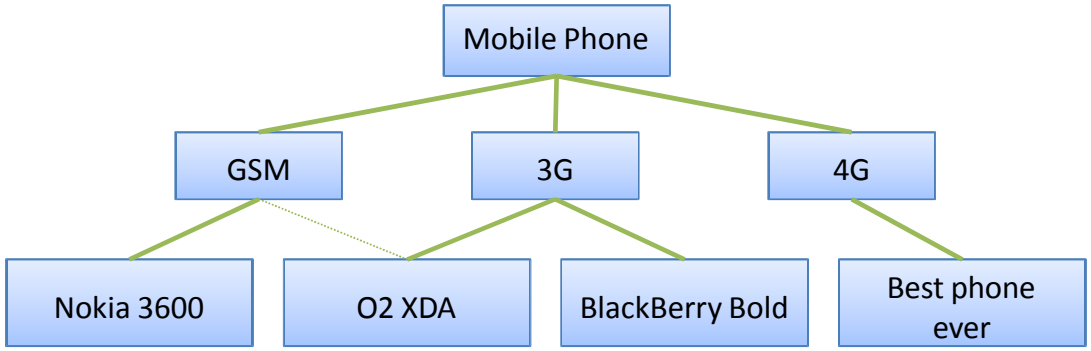
- Deleting a node in a classification hierarchy
  - Should be performed only in exceptional cases
    - It can lead to information loss
- How do we solve it?
  - Soon GSM phones will not be produced anymore
    - But we might have some more in our warehouses, to be delivered
    - Or we might want to query data since when GSM was sold
    - Just mark the end validity date of the GSM branch in the validity matrix

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- Query classification
  - Having the validity information we can support queries like **as is versus as was**
    - Regards all the data as if the only valid classification hierarchy is the present one
    - In the case of O2 XDA, it will be considered as it has always been a 3G phone

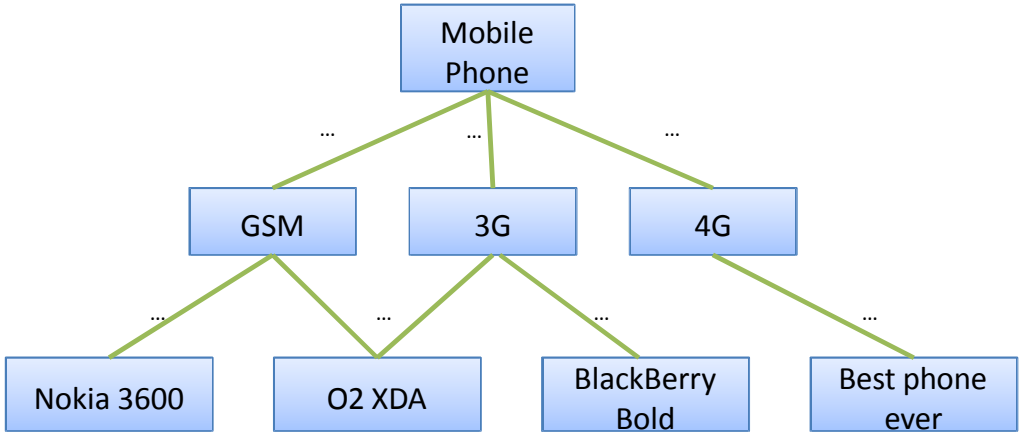


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- As is versus as was
  - Orders the classification hierarchy by the validity matrix information
    - O2 XDA was a GSM phone until 01.07.2008 and a 3G phone afterwards

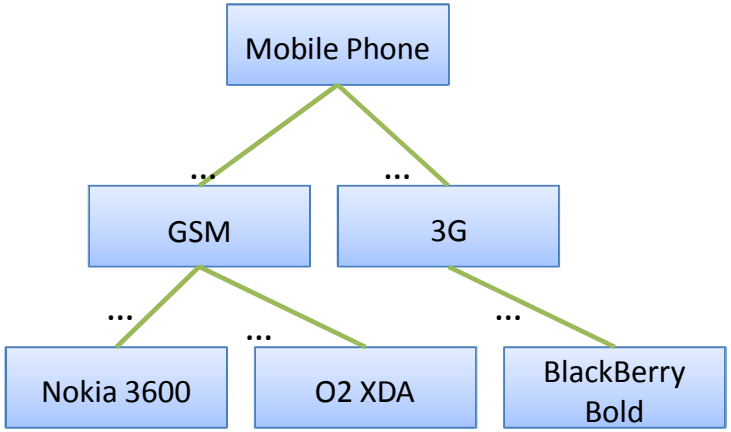


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification Hierarchy (cont'd.)

Detour

- Like versus like
  - Only data whose classification hierarchy remained unmodified, is evaluated
  - E.g. the Nokia 3600 and the Black Berry



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Schema Modification

Detour

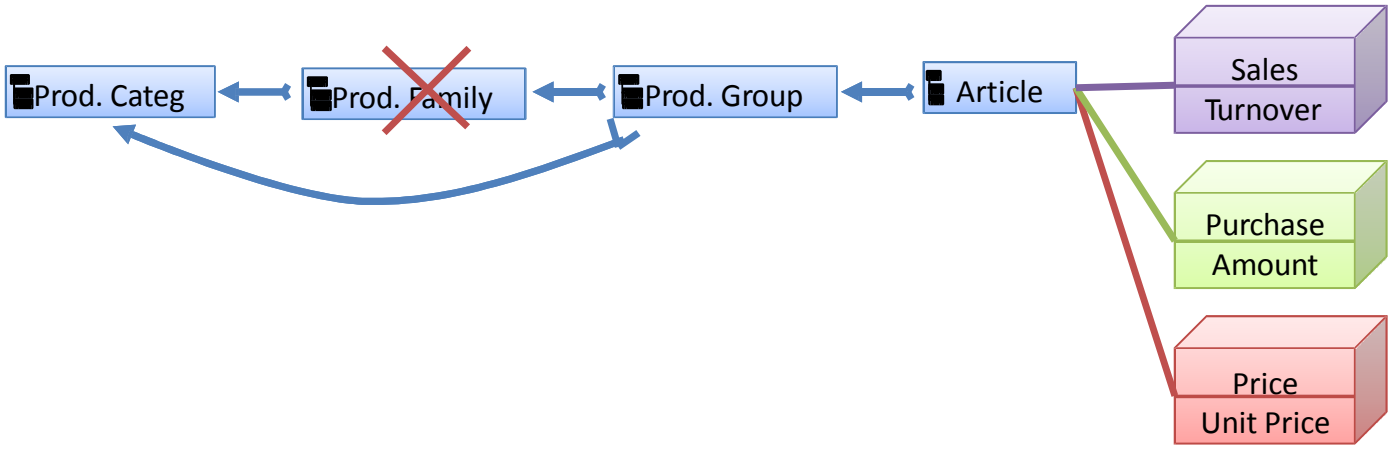
- Improper modification of a schema (deleting a dimension) can lead to
  - Data loss
  - Inconsistencies
    - Data is incorrectly aggregated or adapted
- Proper schema modification is complex but
  - It brings flexibility for the end user
    - The possibility to ask "As Is vs. AsWas" queries and so on
- Alternatives
  - Schema evolution
  - Schema versioning

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Schema Modification (cont'd.)

Detour

- Schema evolution
  - Modifications can be performed **without data loss**
  - It involves schema modification and **data adaptation** to the new schema
  - This data adaptation process is called **Instance adaptation**

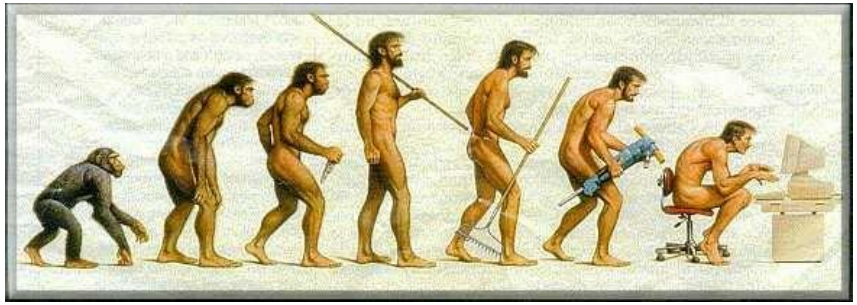


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Schema Modification (cont'd.)

Detour

- Schema evolution
  - Advantage
    - Faster to execute queries in DW with many schema modifications
  - Disadvantages
    - It limits the end user flexibility to query based on the past schemas
    - Only actual schema based queries are supported

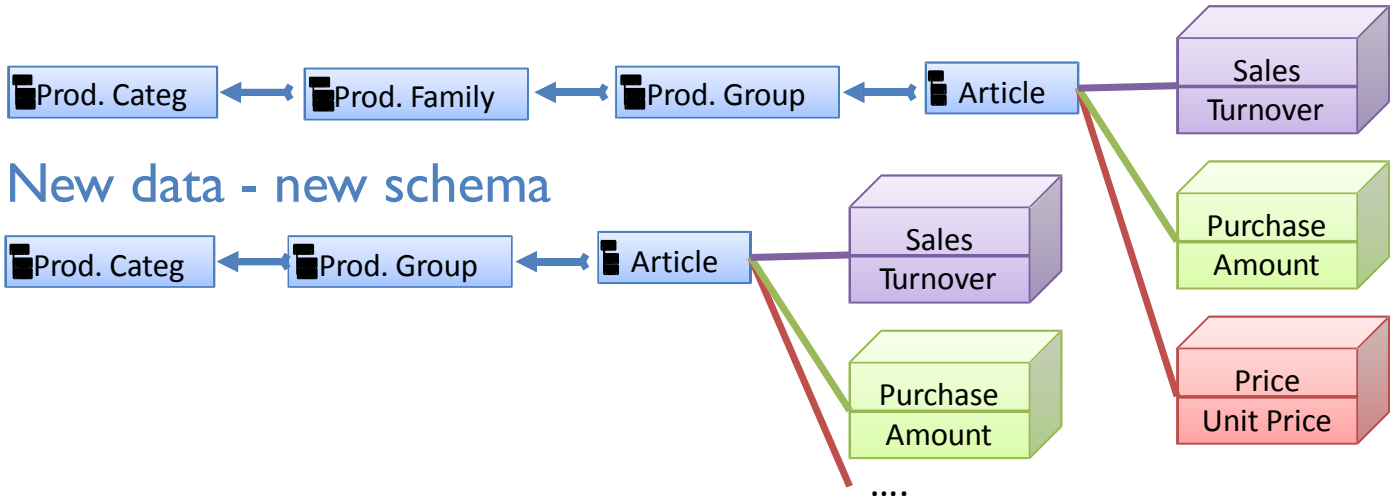


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Schema Modification (cont'd.)

Detour

- Schema versioning
  - Also no data loss
  - All the data corresponding to all the schemas are always available
  - After a schema modification the data is held in their belonging schema
    - Old data - old schema



    - New data - new schema

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Schema Modification (cont'd.)

Detour

- Schema versioning
  - Advantages
    - Allows higher flexibility, e.g., "As Is vs. As Was", etc. queries
  - Disadvantages
    - Adaptation of the data to the queried schema is done on the spot
    - This results in longer query run time

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Summary

- Logical Model
  - Cubes, Dimensions, Hierarchies, Classification Levels

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Next Lecture

- Physical Model
  - Relational Implementation through:
    - Star schema: improves query performance for often-used data
    - Snowflake schema: reduce the size of the dimension tables