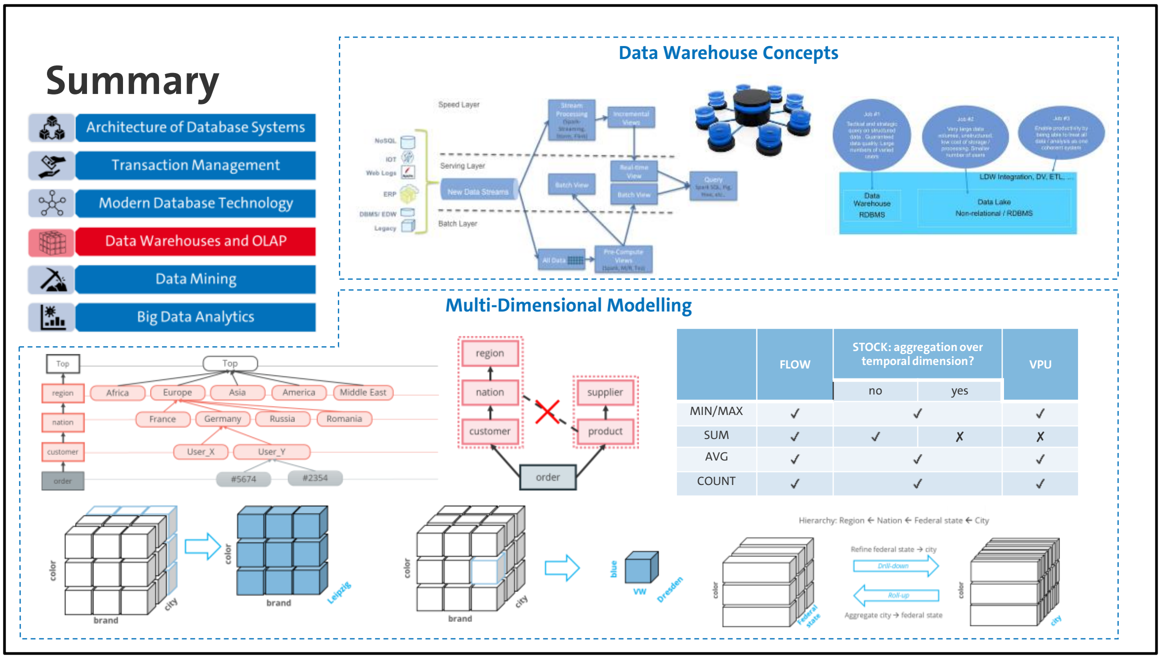
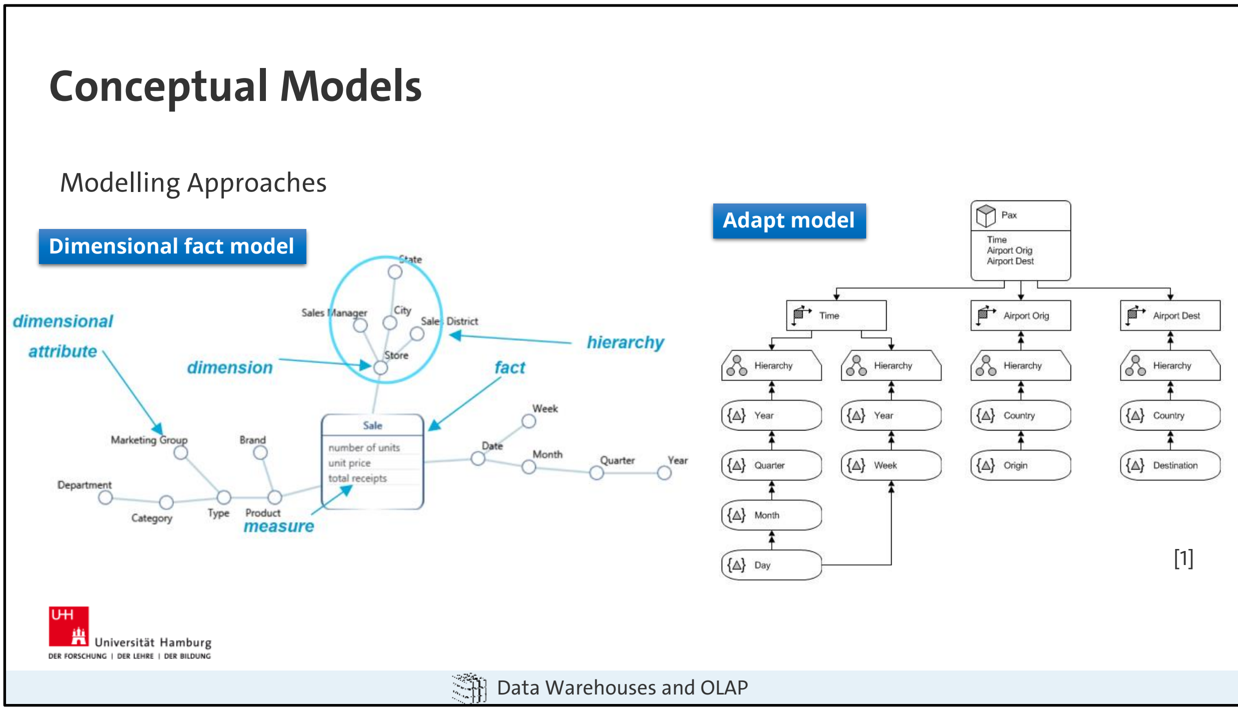

Summary
Architecture of Database Systems
Transaction Management
Modern Database Technology
Data Warehouses and OLAP
Data Mining
Big Data Analytics
Data Warehouse Concepts
Speed Layer
Serving Layer
Batch Layer
New Data Streams
Batch View
All Data
LDW Integration, DV, ETL, ...
Data Warehouse RDBMS
Data Lake Non-relational / RDBMS
Multi-Dimensional Modelling
Top
region
nation
customer
order
Africa
Europe
Asia
America
Middle East
France
Germany
Russia
Romania
User_X
User_Y
#5674
#2354
supplier
product
FLOW
STOCK: aggregation over temporal dimension?
no
yes
VPU
MIN/MAX
SUM
AVG
COUNT
color
brand
city
Leipzig
blue
VW
Dresden
Hierarchy: Region ← Nation ← Federal state ← City
Refine federal state → city
Drill-down
Roll-up
Aggregate city → federal state
Federal state

Multidimensional database design

	Classical relational database design	Multidimensional database design	
Conceptual schema (semi-formal)	Variants of entity-relationship modelling	Different modelling languages, e.g. mE/R, mUML, ADAPT,...	
Logical schema (formal)	Relations with attributes	Data cube: facts and measures	
		Dimensional hierarchy with categorical attributes: classifying and describing attributes	
Internal/physical schema	Memory organisation (primary/secondary indexes, partitioning,...)	Relational storage (ROLAP): Star/Snowflake schema patterns	Multidimensional storage (MOLAP): native implementation

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

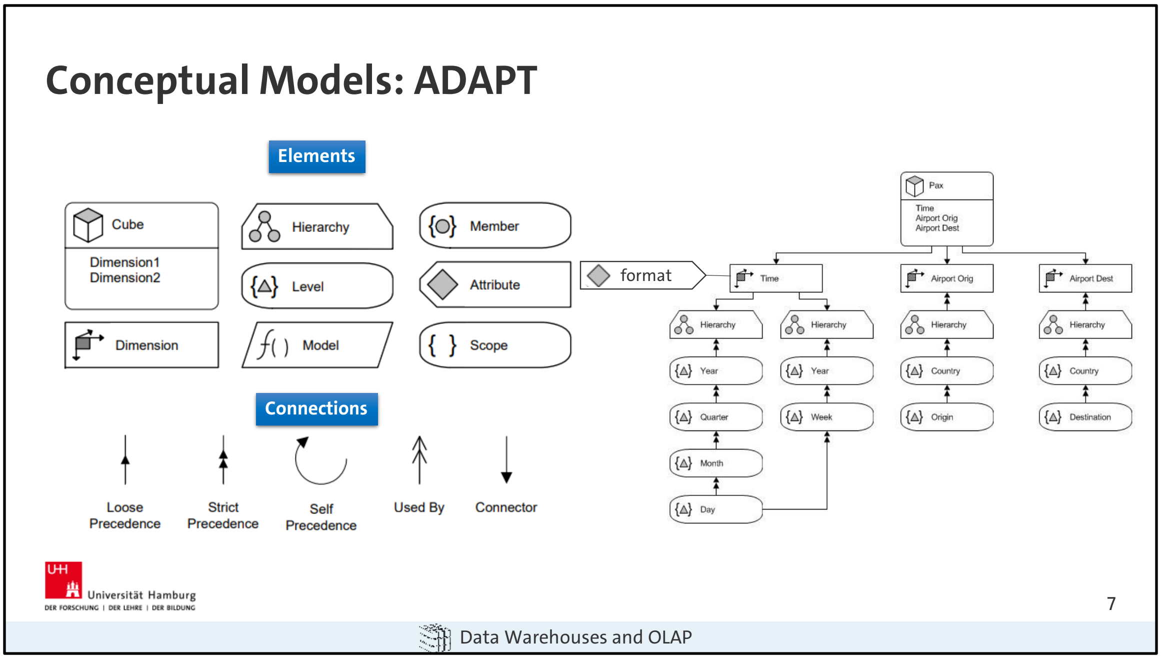
ROLAP = Relational OLAP
MOLAP = Multidimensional OLAP
HOLAP → Hybrid solution



[1] source: https://www.datenbanken-verstehen.de/data-warehouse/data-warehouse-design/anforderungsanalyse/adapt-notation/

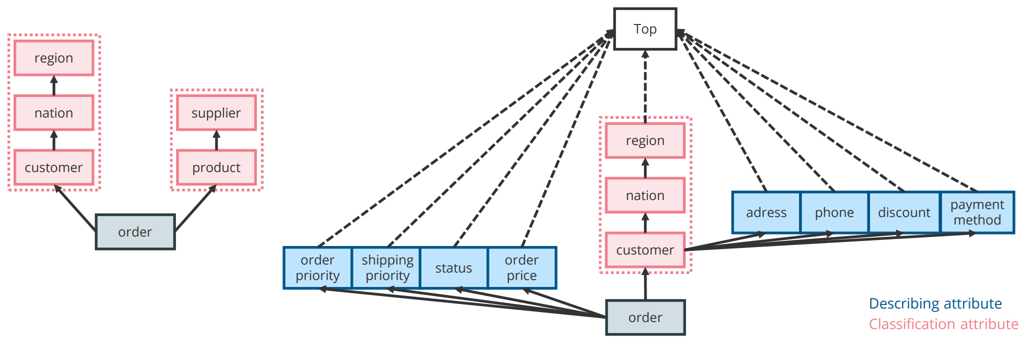
Software for ADAPT: Midrosoft visio with ADAPT Add-On

Further reading
Getting Started with ADAPT:
http://www.symcorp.com/downloads/ADAPT_white_paper.pdf



- There are also different dimension types and scopes which we will not cover here
- *Used by* is used with *Models*

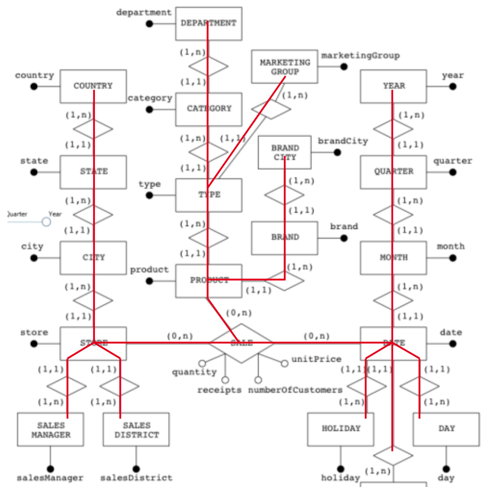
Exercise: ADAPT



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Conceptual Models

Multidimensional ER



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

10

Data Warehouses and OLAP

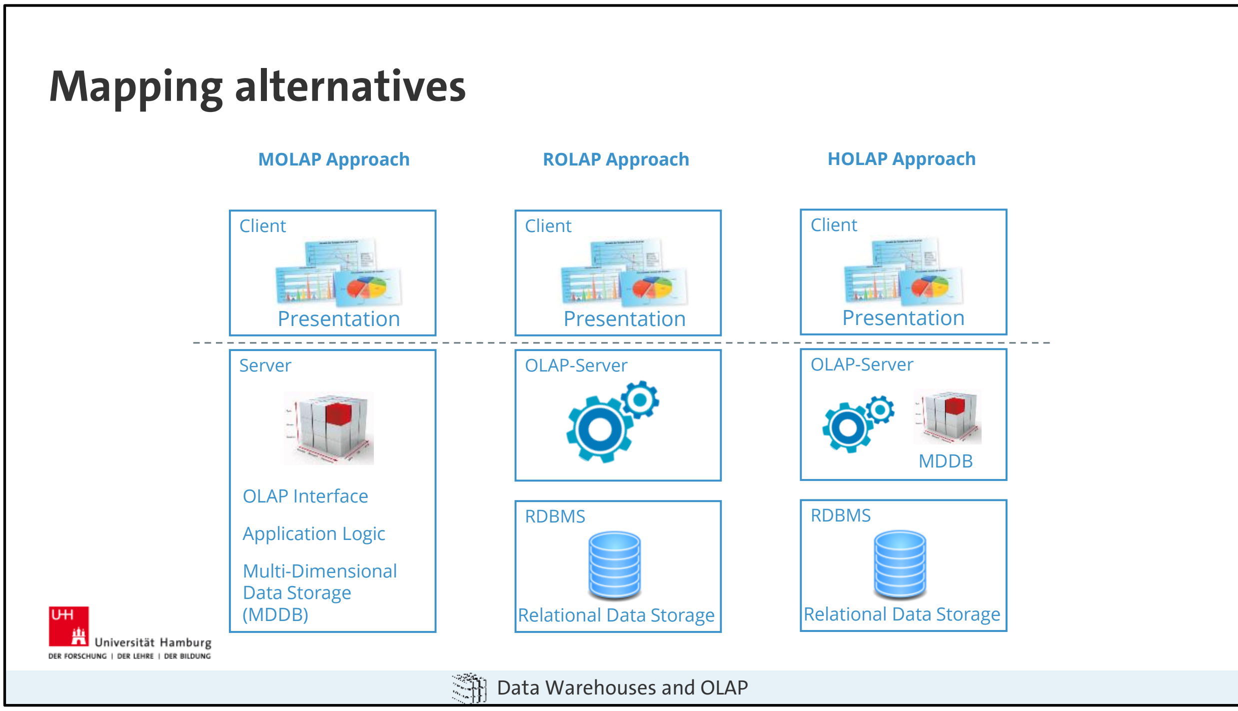
Relational Data model

- Avoiding redundancies
- OLTP queries
- Preferably efficient querying of individual entries

In Data Warehouses

- OLAP queries usually affect multiple entries
- Value ranges and/or aggregates
- Redundancy can shorten query response time

→ Different relational data models



MOLAP

Properties

- Raw data & aggregates stored in OLAP server
- Direct storage as cubes

Pros

- No transfer of modeling concepts
- Separation of descriptive und quantified attributes fixed in the model
- Multiple integrated statistical operators (esp. OLAP functions)
- Fast response times through pre-calculation
- Very efficient due to optimized storage structures

Cons

- Proprietary systems
- Often compression is missing (sparsity) , scalability (MB to GB)
- High storage requirements (factor 10 to 100) due to aggregation/pre calculation
- Often limited number of dimensions (10 to 64)

Products

- Oracle Express, Hyperion Essbase, Cognos Powerplay, Seagate Holos

ROLAP

Properties
- Data is stored in RDBMS
- Information is created via SQL
- Additional tables for aggregates

Pros
- Proven database technology
- Scalability for big volume data / high number of dimensions

Cons
- Installation overhead (interaction of different components)
- Overhead for mapping the Data Cube and complex queries to SQL
- Extensive Meta data management for parameters
- Slower

Products
- Almost all RDBMS vendors (Oracle, DB2, Microsoft, etc.), SAP BW / SIMCE

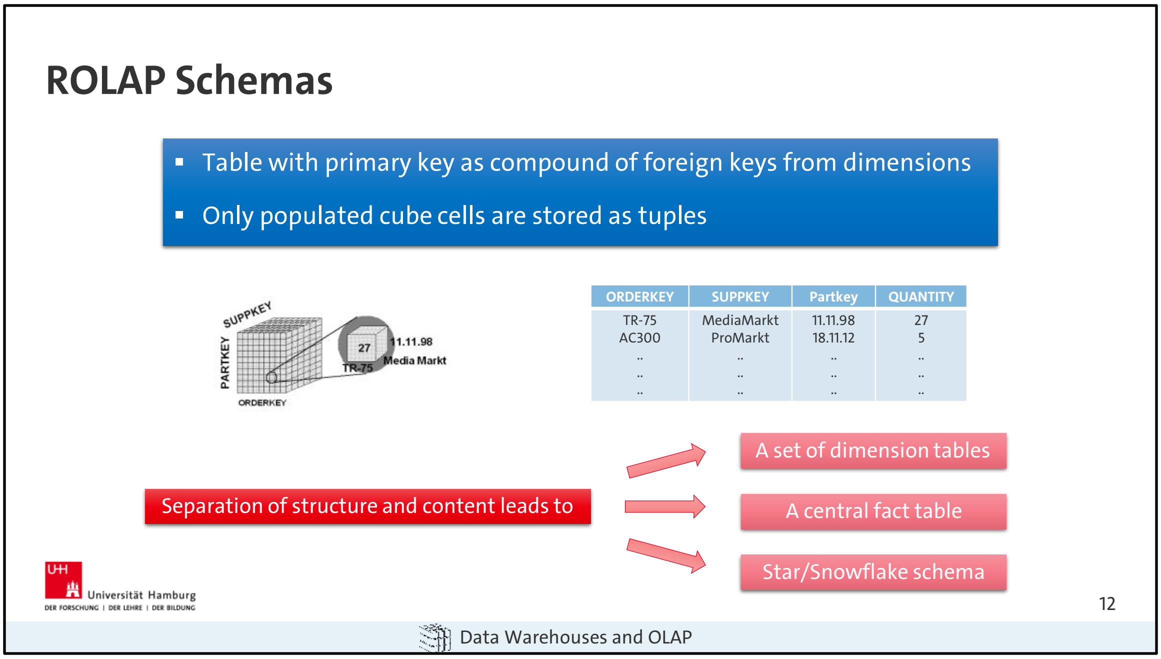
HOLAP

Properties
- „best of both worlds“
- Data partially stored in MOLAP Cubes and RDBMS

Pros
- Proven database technology
- Improved scalability / response times

Cons
- Installation overhead (interaction of different components)
- Complexity / Handling, Which data should be stored where?

Products
- MicroStrategy, Essbase, Mondrian, Microsoft Analysis Services



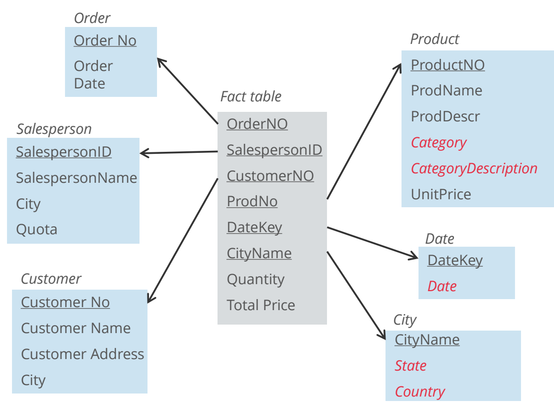
Fact table (content)

- Central table
- Compound primary key from foreign keys of dimension tables (explicit assignment of facts)
- Few columns, many tuples (millions, billions)
- Typically, only numeric data types

Dimension tables (structure)

- Classifying and descriptive attributes
- Many columns, but few tuples (< 10% of fact table)
- Used for selection (by utilizing bitmap-indices)
- Used for aggregation (group-by-clause)
- Foreign key links dimension table and fact table

Star Schema



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

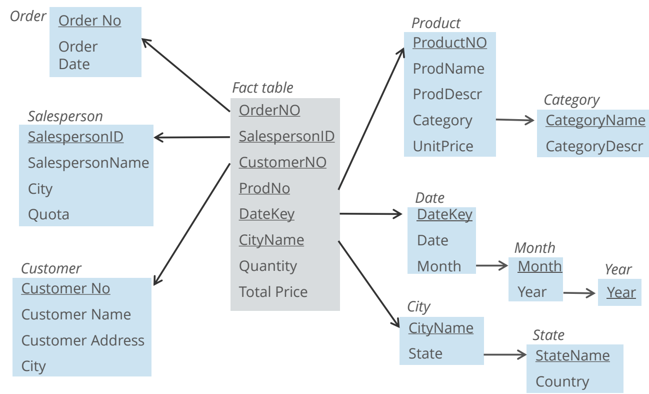
Pros

- Intuitive for users
- Low response time because of read-optimization
- Adaptability to changing structure
- Partitioning and optimization possible
- Less redundancy

Cons

- Not that update friendly
- Multi-hierarchies can be only modeled indirectly
- No distinction between classifying and descriptive attributes (only implicit hierarchies)

Snowflake Schema



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

14

Data Warehouses and OLAP

Main property: normalized dimension tables

+No redundancy

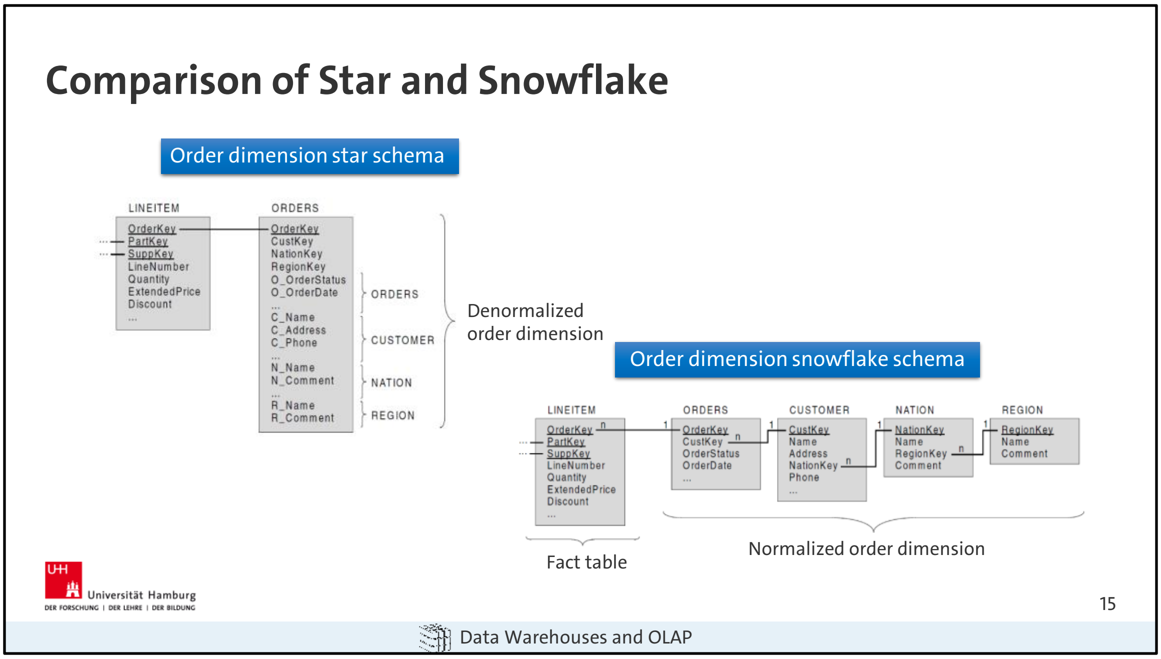
+Update-friendly

+Inherent mapping of structural information of dimension hierarchies

–Less performance due to additional joins

Redundancy: "Resist the urge to normalize"

- Fact table size of hundreds of GBs/TBs
- Plus, index on fact table of similar size
- Dimension tables size of several hundred MBs
- Little storage savings from normalization
- Only favorable for large dimensions or many updates



Benefits Star over Snowflake

- Higher query performance
    - Analytic queries typically address higher aggregation levels
    - Less joins because of denormalization
- Data volume
    - Dimension tables small in comparison to fact tables
    - Additional storage required for denormalized dimension tables rather small
- Changes of classification structure (meta/instance level)
    - Occur rarely
    - Drawback: higher change costs due to duplicate identification for denormalized dimension tables
- Simplicity of structure
    - Fewer dimension tables than snowflake schema
    - Simpler creation of SQL queries by the OLAP-server

Combining Star & Snowflake Schema

<u>Properties</u>

- Some dimensions normalized

- Some dimensions denormalized

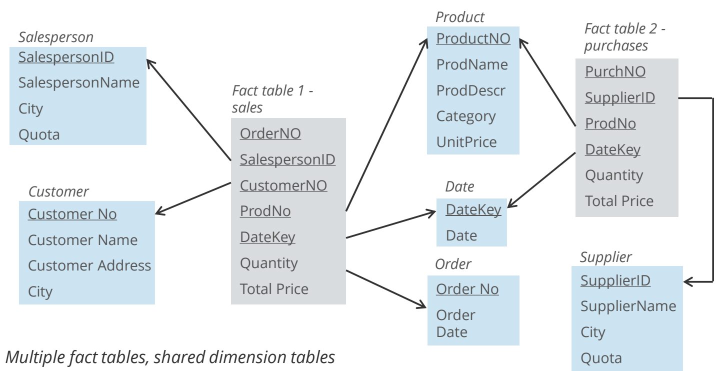
<u>Frequency of updates</u>

- If update frequency of a dimension is high, it makes sense to normalize this dimension to drastically reduce update costs

<u>Number of dimension elements</u>

- High number of finest-grain dimension elements leads to high savings through normalization
- High number of aggregation levels leads to high savings through normalization
- Multiplicative relationship

Galaxy Schema



Multiple fact tables, shared dimension tables

Relational Mapping of Hierarchies

Vertical Mapping

Product

ID	Product	ID2
1	Flour X	1
2	Sugar Y	2
3	Water Z	3

Product group

ID	Product group	ID3
1	Flour	1
2	Sugar	1
3	Water	2

Product family

ID	Product family	ID4
1	Bakery goods	1
2	Beverages	1

Category

ID	Category
1	Food

Horizontal Mapping

Relation product

ID	Product	Product group	Product family	Category
1	Flour X	Flour	Bakery goods	Food
2	Sugar Y	Sugar	Bakery goods	Food
3	Water Z	Water	Beverages	Food

Recursive Vertical Mapping

ID	Product	Product group	Product family	Category
1	Flour X	Flour	Bakery goods	Food
2	Sugar Y	Sugar	Bakery goods	Food
3	Water Z	Water	Beverages	Food

Relation product

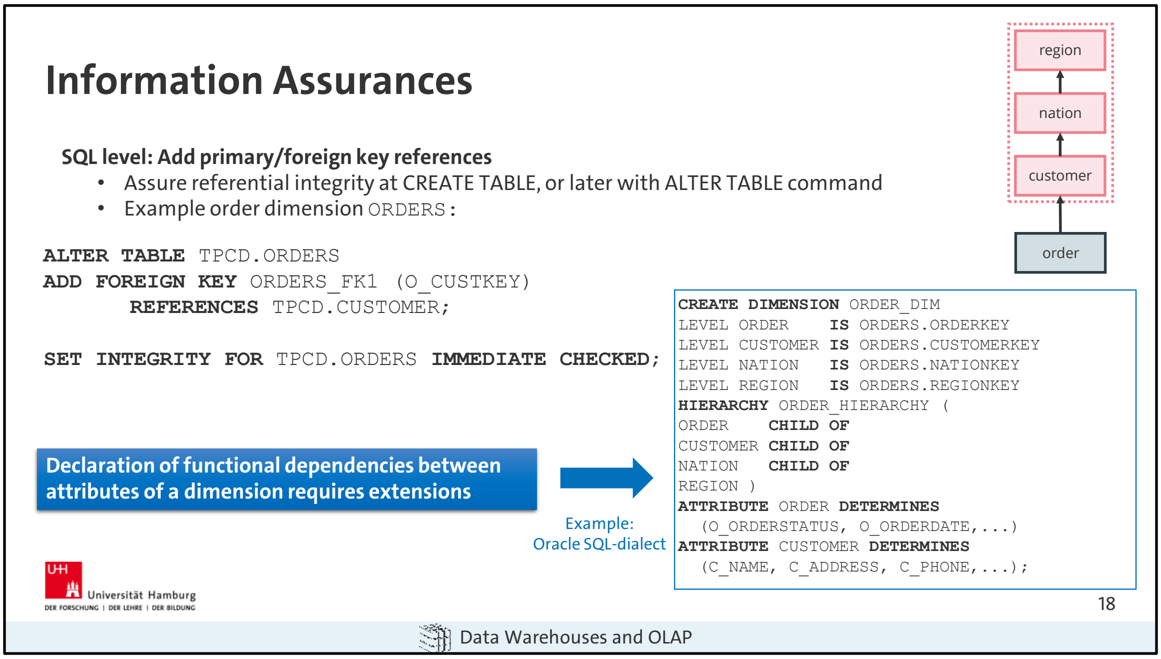
ID	Product	ParentID
1	Food	NULL
2	Bakery goods	1
3	Beverages	1
4	Flour	2
5	Sugar	2
6	Water	3
7	Flour X	4
8	Sugar Y	5
9	Water Z	6

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Vertical mapping
- Explicit mapping using separate relations
- Normalized
- Ref. snowflake schema

Horizontal mapping
- Implicit mapping of equally ranked attributes
- Denormalized
- Ref. star schema

Recursive vertical mapping
- Explicit mapping of arbitrary deep hierarchies
- Combination with horizontal mapping possible
- Extension for irregular hierarchies possible
- Recursive resolution of hierarchy via a set of self-joins



<u>Primary/foreign key relation between dimension- & fact table</u>

- Referential integrity: assure that every quantified part can be evaluated against the descriptive part of a multidimensional schema
- Every part of the composite primary key of the fact table has to be a foreign key to the primary key of the corresponding dimension table

<u>Problem</u>

- PK/FK relations can be declared using classical DB means, BUT declaration of functional dependencies between attributes of a dimension → requires extensions , e.g. `CREATE DIMENSION` (Oracle)
- Knowledge about functional dependencies is necessary to check disjointness
- Information assurances describe functional dependencies, but inside a relation (e.g. denormalized dimension tables in star schemas)