Advanced Data Management Technologies

Unit 4 — Requirements Analysis and Conceptual Design

J. Gamper

Free University of Bozen-Bolzano
Faculty of Computer Science
IDSE

Acknowledgements: I am indebted to Michael Böhlen and Stefano Rizzi for providing me their slides, upon which these lecture notes are based.

Outline

Outline

Goal of User Requirements Analysis

- We skip the analysis and reconciliation of data sources, which should be the first step
- Instead, we start with the user requirement analysis
- Aims to collect end user needs for DW applications and usage
- Main "source" of information are the so-called business users (end users)
- Has strategic importance as it influences almost every decision made during the project
- Plays an essential role for the conceptual design (which is the next step)
- Different ways to elicit user requirements, e.g.,
  - Interviews
  - Glossary-based requirements analysis

Interviews

- Frequently used method are interviews with single users or small groups of users
- Different types of questions
  - Open-ended questions
    - What do you think of data source quality?
    - What are the key objectives your unit has to face?
  - Closed questions
    - Are you interested in sorting out purchases by hour?
    - Do you want to receive a sales report every week?
  - Evidential questions
    - Could you please give me an example of how you calculate your business unit budget?
    - Could you please describe the issues with poor data quality that your business unit is experiencing?

Glossary-based Requirements Analysis

- Aims at creating tables that collect information about facts, dimensions, attributes and their relationship
- It is recommended that this analysis is focused on **facts**
- Facts are the concepts on which end users base decision-making processes
- Each fact describes a category of events taking place in enterprises
- Facts essentially represent **business processes**
- Frequently, this analysis is going hand-in-hand with the conceptual design

Typical Facts of Different Application Fields

Application field	Data Mart	Facts
Business, manufacturing	Supplies	Purchases, stock inventory, distribution
	Production	Packaging, inventory, delivery, manufacturing
	Demand management	Sales, invoices, orders, shipments, complaints
	Marketing	Promotions, customer retention, advertising campaigns
Finance	Banks	Checking accounts, bank transfers, mortgage loans, loans
	Investments	Securities, stock exchange transactions
	Services	Credit cards, bill payment through standing orders
Health service	Division	Admissions, discharges, transfers, surgical operations, diagnosis, prescriptions
	Epidemiology	Diseases, outbreaks, treatments, vaccinations
Transportation	Goods	Demand, supply, transport
	Passengers	Demand, supply, transport
Telecommunications	Traffic management	Network traffic, calls
	Cust. rel. management	Customer retention, complaints, services
Tourism	Demand management	Ticketing, car rentals, stays

Facts Enriched with Dimensions and Measures

- Facts should be enriched with additional information, such as dimensions and measures
- Such information can be derived from existing documentation, database schemata of source systems, users, etc.
- As DW store historical information, every fact needs a historical interval, for which the data should be stored
- Example of user requirements glossary

Fact	**Dimensions**	**Measures**	**History**
Stock inventory	Product, Date, Warehouse	Stocked quantity	1 year
Sales	Product, Date, Store	Sold quantity, Receipts, Discount	5 years
Order Lines	Product, Date, Supplier	Ordered quantity, Receipts, Discount	3 years

Preliminary Workload

- Together with the facts, a set of preliminary workloads should be identified
- Workloads are analysis queries the user wants to answer
- Example for workload (analysis queries)

Fact	Query
Stock inventory	What is the average quantity of each product made available monthly in every warehouse? Which product stocks ran out at least once last week at the same time in every warehouse? What's the daily trend of all the stocks grouped by product type?
Sales	What's the total amount per product sold last month? What are the daily receipts per store? What is the annual report of receipts per state per product?
Order lines	What is the total amount of goods ordered from a specific supplier every year? What's the daily total amount ordered last month for a specific product type? What's the best discount given by each supplier last year and grouped by product category?

Choose a Business Process (Kimball and Ross)

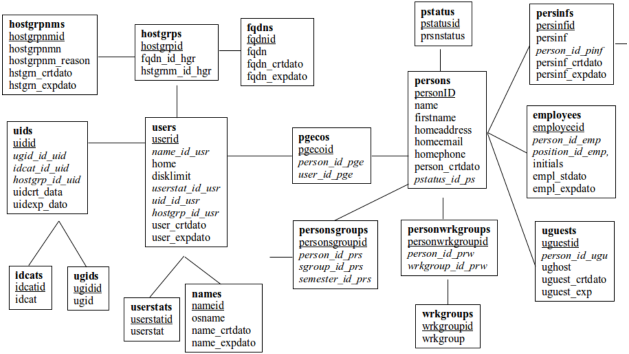
- Kimball and Ross propose in their DW/DM design methodology as the first step to choose business process(es) together with analysis questions that can be answered
- Example of business process in a grocery store domain:
  - Management wants to better understand customer purchases as captured by the POS system.
  - Business process: POS retail sales
  - Allows us to analyze:
    - What products are selling?
    - In which stores?
    - On what days?
    - Under what promotional conditions?
    - etc.

Outline

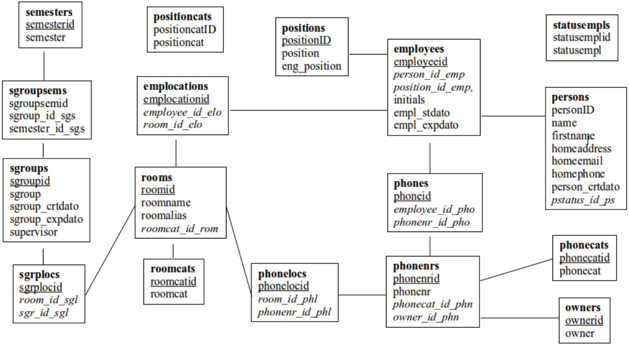
Why a New Model?/1

- ER and OO model are widely used as a conceptual tool for documentation and design of relational databases
- ER/OO models serve many purposes, thus they are flexible and general
- All types of data are equal
- No difference between
  - What is important
  - What just describes the important things
- ER/OO models are large
  - 50–1000 entities/relations/classes
  - Hard to get an overview
- ER/OO models implemented in RDBMSs
  - Normalized databases spread information
  - When analyzing data, the information must be integrated/joined

OLTP Example: CS Dept/1



OLTP Example: CS Dept/2



Why a New Model?/2

- ER/OO models are not very useful in modeling DWs.
- It is now generally recognized that a DM/DW is based on a multidimensional view of the data.
- But there is still no agreement on how to realize its conceptual design.
- Very often DM design is at the logical level, i.e., star/snowflake schema is directly designed.
  - But a star schema is nothing but a relational schema.
  - Standard implementation of the multidimensional model in RDBMS.
  - Contains only the definition of a set of relations and integrity constraints!
- A better approach:
  1. design first a conceptual model using richer and more user-friendly language;
  2. translate conceptual model into a logical model.

Dimensional Fact Model (DFM)

- The Dimensional Fact Model (DFM) is a graphical conceptual model for DM/DW design.
- The aim of the DFM is to
  - provide effective support to conceptual design;
  - create an environment in which user queries may be formulated intuitively;
  - make communication possible between designers and end users with the goal of formalizing requirement specifications;
  - build a stable platform for logical design (independently of the target logical model);
  - provide clear and expressive design documentation.
- The conceptual representation generated by the DFM consists of a set of fact schemata that basically model facts, measures, dimensions, and hierarchies.

DFM: Facts, Measures and Dimensions

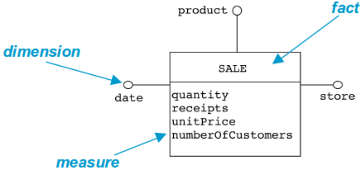
- A fact is a concept relevant to decision-making processes.
  - It typically models events taking place within a company, e.g.,
    - in commercial domain: sales, shipments, purchases, taking exams, ...
    - in healthcare industry: patient transfers, discharges, surgeries, ...
    - in financial business: stock exchange transactions, credit card balance, ...
  - It is essential that a fact has dynamic properties and evolves over time.
- A measure is a numerical property of a fact and describes a quantitative fact aspect that is relevant to analysis, e.g.,
  - every sale is quantified by its units sold, unit price, ...
  - exams are quantified by its grades, credit points, ...
  - bank transfers are quantified by the amount

  Measures are used to make calculations and analyses
- A dimension is a property of a fact with a finite domain and describes an analysis coordinate of the fact.
  - Typical dimensions are
    - for the sales fact: product, store, date
    - for the patient transfer fact: patient, department, date

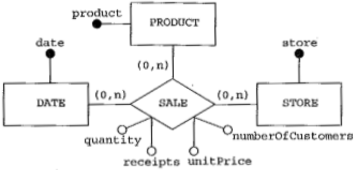
DFM: Sales Facts Example/1

- **Example:** Sales facts in a store
  - Fact: 10 packages of milk were sold for $25 on 10.10.2013 in the DM store.

DFM schema



Corresponding ER schema



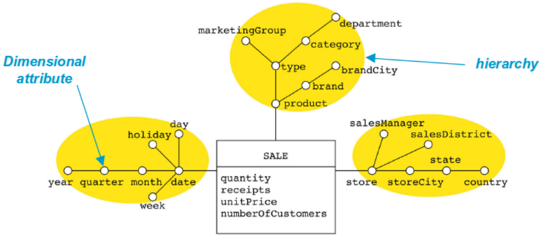
- In the DFM
  - A fact expresses a many-to-many relationship between its dimensions.
  - Facts, dimensions, and measures are first-class citizens
- In the ER model these concepts are not first-class citizens
  - Fact is a relationship, dimensions are entities, measures are attributes of relationships.

DFM: Dimensional Attributes and Hierarchies

- The general term dimensional attributes stands for the dimensions and other attributes that describe the dimensions
  - e.g., a product is described by its type, by the category to which it belongs, by its brand, and by the department in which it is sold.
- Dimensional attributes have always discrete values
- Hierarchies are used to represent relationships between dimensional attributes
- A hierarchy is a directed tree: nodes are dimensional attributes and arcs model many-to-one associations between dimensional attributes
  - The dimension itself is at the root of the tree
  - All other dimensional attributes are (direct or indirect) descendents
  - The root defines the finest granularity level; the other attributes are at a coarser granularity

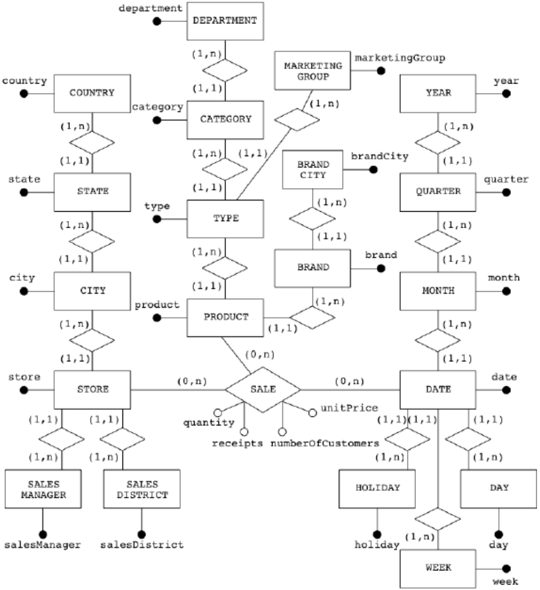
DFM: Sales Events Example/2

- **Example:** Fact schema for sales events enhanced with dimensional attributes



- Many-to-one relationships (i.e., hierarchies) from parent nodes to child nodes
- Hierarchies describe functional dependencies, e.g.,
  - product → type, type → category, category → department
    product → brand, brand → brandCity

DFM vs. ERM


department
DEPARTMENT
MARKETING GROUP
marketingGroup
country
COUNTRY
category
CATEGORY
YEAR
year
BRAND CITY
brandCity
state
STATE
type
TYPE
QUARTER
quarter
BRAND
brand
city
CITY
product
PRODUCT
MONTH
month
store
STORE
SALE
DATE
date
quantity
receipts
numberOfCustomers
unitPrice
SALES MANAGER
SALES DISTRICT
HOLIDAY
DAY
salesManager
salesDistrict
holiday
day
WEEK
week
(1,n)
(1,1)
(0,n)

DFM: Naming Conventions

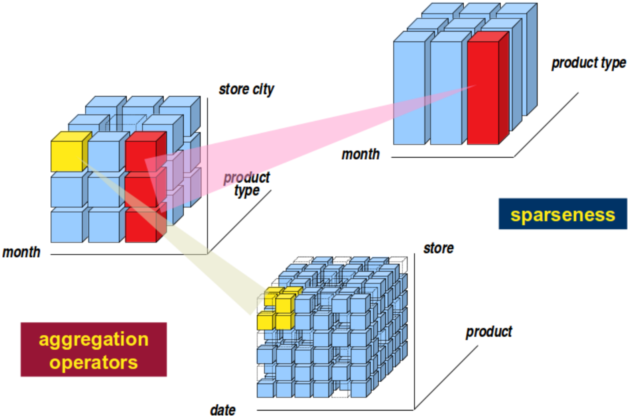
- All attributes and measures within a fact schema must have different names
- You can differentiate similar names, if you qualify them with the name of the dimensional attribute that comes before them in hierarchies
  - e.g., `storeCity` and `brandCity`
- Attributes names should not explicitly refer to the fact they belong to
  - Avoid `shipped product` and `shipment date`
- Attributes with the same meaning in different fact schemata should have the same name

Primary and Secondary Events/1

- A **primary event** is a particular occurrence of a fact that is identified by one tuple with a value for each dimension and each measure.
  - e.g., 10 packages of milk were sold for a total of $25 on 10/10/2013 in the SmartMart store.
- Each combination of a set of dimensional attribute values identifies a **secondary event** that aggregates each measure over all corresponding primary events.
  - e.g., `(product:'milk', storeCity:'Bozen', month:'10/2013')` identifies a secondary event that aggregates all sales of milk in October 2013 in Bozen.
- Hierarchies are used to define the way how to aggregate primary events and effectively select them for decision-making processes.
  - The root of the hierarchy defines the finest aggregation granularity.
  - The other dimensional attributes correspond to a gradually increasing granularity.

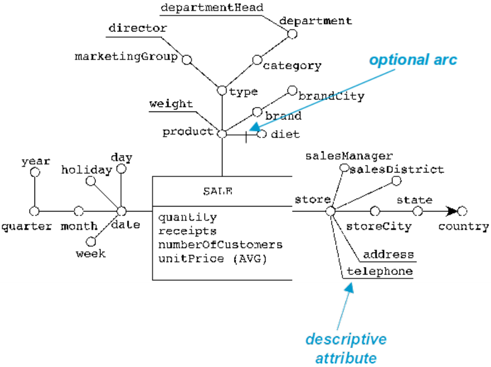
Primary and Secondary Events/2

- Primary events: (product, store, date)
- Secondary events: (product type, store city, month), (product type, month)



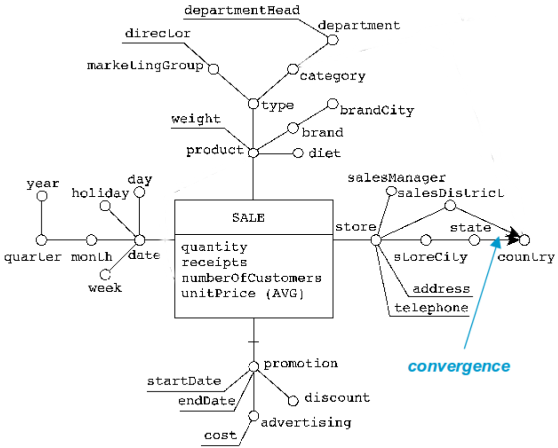
DFM: Descriptive and Optional Attributes

- Descriptive attributes store additional information (i.e., a property) about dimensional attributes, e.g., address, phone number
    - Usually not used for aggregation
    - One-to-one association to a dimensional attribute, and always a leave node
- Some arcs in a fact schema can be optional
    - e.g., diet has a value (cholesterol-free, gluten-free, sugar-free, . . .) only for food; for other products the value is null



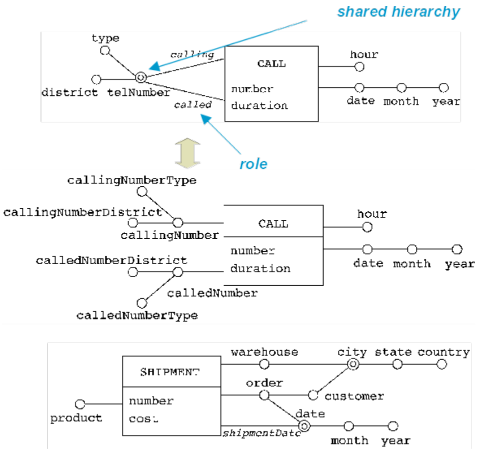
DFM: Convergence

- In a convergence, two (or more) dimensional attributes are connected by two (or more) distinct directed paths
- Each path still represents a functional dependency
  - e.g., $\ldots \rightarrow$ salesDistrict $\rightarrow$ country, $\ldots \rightarrow$ state $\rightarrow$ country



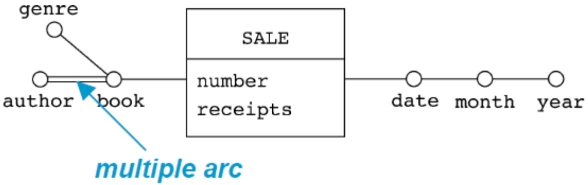
DFM: Shared Hierarchy

- A shared hierarchy is a shorthand to denote that a part of a hierarchy is replicated
- If a hierarchy were replicated, names would have to be qualified with the dimension in order to avoid ambiguity
- Roles are used to specify the meaning if a shared hierarchy starts at the dimension attribute, e.g., `calling` and `called` for `telNumber`
- Otherwise, the meaning is specified by the parents, e.g., warehouse city vs. customer city



DFM: Multiple Arc

- A multiple arc models a many-to-many association between two dimensional attributes (and not many-to-one)
- Aggregation along multiple arcs needs particular care
  - e.g., how much did Rizzi sell, how much Rizzi and Golfarelli together?

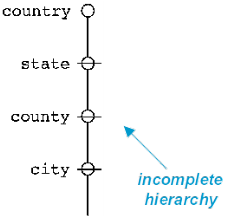


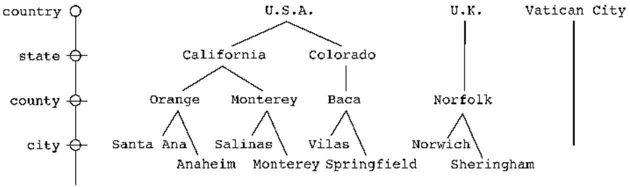
Facts & Crimes	Golfarelli, Rizz	3
Sounds Logical	Golfarelli	5
The Right Measure	Rizzi	10
Facts: How and Why	Golfarelli, Rizz	4
The Fourth Dimension	Golfarelli	8

How much did Rizzi sell?

DFM: Incomplete Hierarchy

- An incomplete hierarchy is a hierarchy where, for some instances, one ore more aggregation levels are missing, because they are unknown or undefined
- Different from optional arcs, where all descendant attributes are missing; here, only selected attributes are missing





Outline

Conceptual Design

- Process of designing/creating a set of DFM schemata
- Requirements-driven approach
  - Designers extract detailed information about facts, measures, and hierarchies from user interviews
  - A connection between the source schema and the data mart schema is established later
- Data-driven approach
  - A conceptual schema for the data mart is created starting from the schema of data sources
  - A connection between the source schema and the data mart schema is easily established
  - A preliminary conceptual schema can be automatically derived

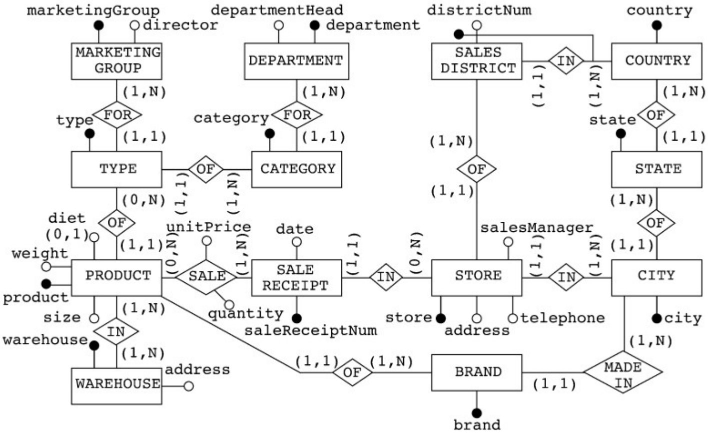
Requirements-driven Approach

- The DFM schemas are derived by the designer from the result of the requirements analysis (interviews, glossaries, . . .)
- Design has to manipulate the interviews with users in order to extract
  - precise instructions about facts
  - measures defining those facts
  - hierarchies for those facts that can be used for aggregating

Data-driven Approach

- Conceptual design starts from the documentation of the data sources
  - ER diagrams
  - Relational schemata
  - XML schemata
  - ...
- Design steps:
  - Define facts
  - For each fact:
    - Build an attribute tree
    - Edit the attribute tree
    - Define dimensions
    - Define measures
    - Create a fact schema
- We show the design steps for ER diagrams; they work in a similar way for relational schemata and other documentation

ER Schema for Sales Example



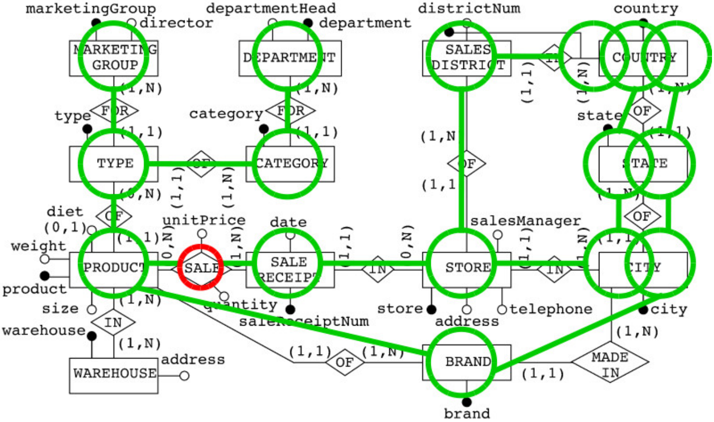
Defining Facts

- In an ER schema, a fact my correspond either to an entity or to an n-ary relationship
- Entities that are frequently updated, such as SALE, are good candidates for facts
- Entities that represent structural domain properties are rather static, such as STORE and CITY, and are not good candidates for facts
- In the sale ER schema, we choose as a fact the SALE relationship

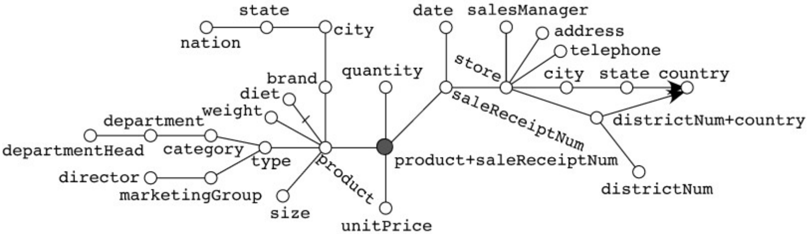
Building the Attribute Tree

- In an attribute tree:
  - The root corresponds to the entity/relationship identified as fact
  - Each node corresponds to a source schema attribute
  - For each node v, the corresponding attribute functionally determines all the attributes corresponding to the descendants of v
- The attribute tree can be automatically constructed by recursively navigating functional dependencies expressed by identifiers and many-to-one relationships in the source schema

Example: Building the Attribute Tree

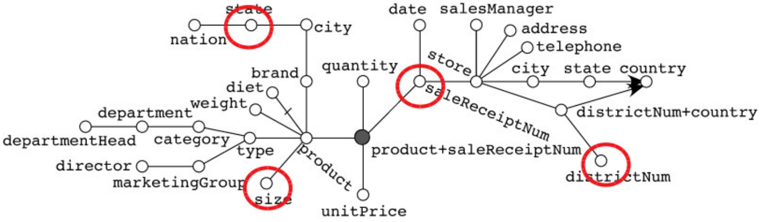


Example: Attribute Tree

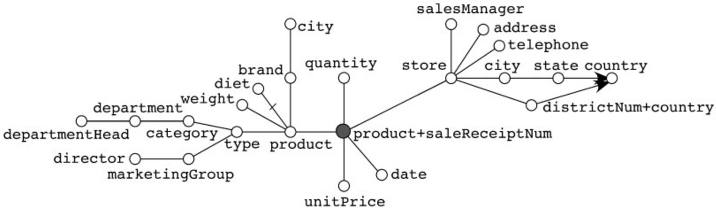


Editing the Attribute Tree

- Generally, not all attributes in the tree are relevant to the DM
- Unnecessary levels of detail should be removed
  - Prune nodes and the entire sub-tree, e.g., `state` or `size`
  - Graft individual nodes by connecting children with the parent node, e.g., `saleReceiptNum`

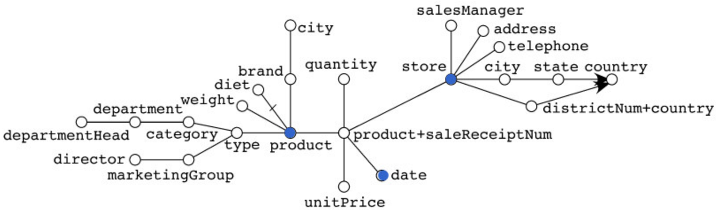


Edited Attribute Tree



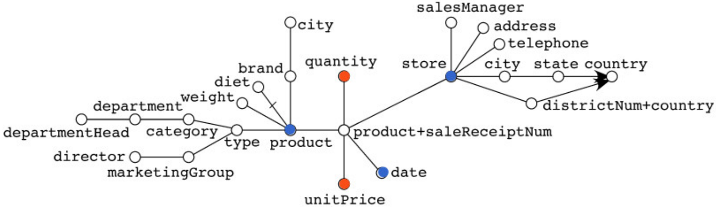
Defining Dimensions

- Dimensions are selected from the root child nodes of the attribute tree, e.g, `product`, `store`, `date`
- Selecting dimensions is crucial since it defines the granularity of primary events
- Time should always be a dimension
  - Typically represents validity time, i.e., time when an event occurs in the business domain
  - Transaction time, i.e., time when an event is stored in the DM, is normally not considered relevant for decision making



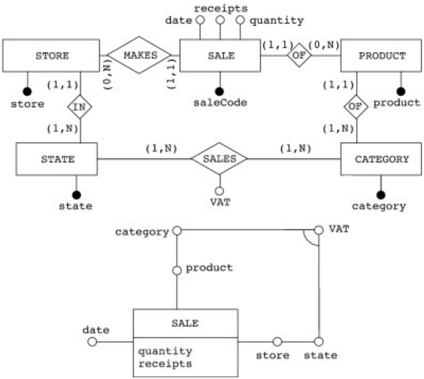
Defining Measures

- Attributes are usually from the numeric attributes which are root children, e.g., `quantity`, `unitPrice`
- Otherwise, aggregation operators (SUM, AVG, MAX, ...) might be applied on primary events to obtain measures
- A fact may also be without measures



Generating DFM Schemata

- The attribute tree can now be automatically translated into a DFM schema including the dimensions and measures specified in the preceding phases
  - Hierarchies correspond to sub-trees of the attribute trees with their roots in dimensions
  - Fact names correspond to the names of entities chosen as facts



Summary

- Requirements analysis to solicit business users needs, e.g., using interviews or glossary-based analysis
  - Results in facts/business processes and workload/queries
- There is a need for new models and modelling approaches in DWs, since ER/OO models are not very useful
  - too flexible and general resulting in complex models
- DW/DM design should be done in 2 steps:
  - conceptual design produces a conceptual model;
  - logical design transforms conceptual model into logical model.
- Conceptual design is frequently skipped.
- DFM is a graphical conceptual model to support the conceptual design.
  - Distinguishes between dimensions, facts, and measures
- DFM schemas can be created from the user requirements (requirements-driven approach) or derived from the documentation of the data sources (data-driven approach)