Advanced Data Management Technologies

Unit 6 — Case Studies

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

The Grocery Store Example/1

- A grocery chain with 500 stores spread over a five-state area.
  - Each of the stores is a typical modern supermarket with a full complement of departments including grocery, frozen foods, dairy, meat, bakery, hard goods, liquor, and drugs.
  - Each store has roughly 60.000 individual products on its shelves.
- The individual products are called stock keeping units (SKUs).
- About 40.000 of the SKU come from outside manufacturers and have bar codes imprinted on the product package.
  - These bar codes are called universal product codes (UPCs).
  - UPCs are at the same grain as individual SKUs.
  - Each different package variation of a product has a separate UPC and hence is a separate SKU.
- The remaining 20.000 SKUs come from departments like meat or bakery departments and do not have nationally recognized UPC codes.
  - The grocery store assigns SKU numbers to these products by sticking scanner labels on the items.
  - Although the bar codes are not UPCs they are certainly SKU numbers.

The Grocery Store Example/2

- Data is collected at several places in a grocery store.
  - Some of the most useful data is collected at the cash registers as customers purchase products.
  - Our modern grocery store scans the bar codes directly into the point-of-sale (POS) system.
  - The POS system is at the front door of the grocery store where customer takeaway is measured.
  - The back door, where vendors make deliveries, is another interesting data-collection point.

The Grocery Store Example/3

- At the grocery store, management is concerned with the logistics of ordering, stocking the shelves, and selling the products **while maximizing the profit** at each store.
- The profit ultimately comes from
  - charging as much as possible for each product,
  - lowering costs for product acquisition and overhead, and
  - at the same time attracting as many customers as possible.
- The most significant decisions have to do with pricing and promotions.
  - Both store management and headquarters marketing spend a great deal of time tinkering with pricing and running promotions.
  - Promotions in a grocery store include temporary price reductions, ads in newspapers and newspaper inserts, displays in the grocery store, and coupons.

Simplified DM Design Process (Kimball and Ross)

- A somehow simplified DM design process consists of the following 4 steps:
  1. Choose the business process(es) to model
     - e.g., Sales,
  2. Choose the granularity of the business process
     - e.g., Items by Store by Promotion by Day.
     - Low granularity is needed.
     - Are individual transactions necessary/feasible?
  3. Choose the dimensions
     - Time, Store, ...
  4. Choose the measures
     - Dollar_sales, unit_sales, dollar_cost, customer_count

Step 1: Choose the Business Process

- A business process is an activity in the organization that typically is supported by a source data management system
  - raw material purchasing, orders, shipments, invoicing, inventory, bank transfers, patient transfers, . . .
- Business processes are not necessarily limited to a single department
  - e.g., sales and marketing departments might be interested in the orders
- Focusing on the business process rather than the department avoids duplication of work and keeps data more consistent
- The first dimensional model built should be the one with the most impact
- It should answer the most pressing business questions and be readily accessible for data extraction

Step 1: Choose the Business Process – Example

- 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.

Step 2: Choose the Grain of the Business Process

- Preferably develop dimensional models for the most atomic information captured by a business process
  - Not because queries report individual rows, but queries need to cut through the details in very precise ways
- The more detailed/atomic data is, the more things we know
- Atomic data provides maximum analytic flexibility
- Can be constrained and rolled up in every possible way
- It is always possible to declare higher-level grains by aggregation of atomic data; the opposite is not true
- Less granular model is vulnerable to unexpected requests for more details
- Example grain declarations
  - Individual line item on a customer's sales ticket as measured by a scanner
  - An individual boarding pass of a flight
  - A monthly snapshot for each bank account

Step 2: Choose the Grain of the Business Process – Example

- Individual line item on a POS transaction is the most detailed data, and we choose this as grain (→ event fact)
- Allows a very detailed analysis of sales
  - Difference in sales on Monday vs. Sunday
  - Is it worthwhile to stock so many individual sizes of certain brands?
  - How many shoppers took advantage of the 50-cents-off promotion on shampoo?
  - Impact in terms of increasing sales when a competitive diet soda product was promoted
  - etc.
- Note that none of these queries calls for data from a specific transaction, but require detailed ways to slice data
  - Could not be answered if only aggregated values would be stored, e.g., daily summaries

Step 3: Choose the Dimensions

- Dimensions can be derived by answering the question *"How do business people describe the data resulting from the business processes?"*
- "Decorate" fact tables with dimensions representing all possible descriptions of the facts/measures
- A clear grain statement helps to identify the dimensions
- Sometimes a revision of step 2 is required

Step 3: Choose the Dimensions – Example/1

- The Date dimension
  - Explicit date dimension is needed (events, holidays, ...)
- The Product dimension
  - Hierarchy allows drill-down/roll-up through category, brand, department, etc.
  - Many descriptive attributes (often more than 50)
- The Store dimension
  - Primary geographic dimension to specify location of the store
  - Many descriptive attributes
- The Promotion dimension
  - Used to see if promotions work and are profitable
  - Ads, price reductions, end-of-sale displays, coupons
    - Highly correlated (only 5000 combinations)

Step 3: Choose the Dimensions – Example/2

- Preliminary version of grocery store schema

Date dimension
DateKey (PK)
FullDateDesc
DayOfWeek
Month
Year
Quartal
...

Product Dimension
ProductKey (PK)
ProductDesc
BrandDesc
CategoryDesc
DepartmentDesc
...

POS Retail Sales Transaction Facts
DateKey (FK)
ProductKey (FK)
StoreKey (FK)
PromotionKey (FK)
Measures TBD

Store Dimension
StoreKey (PK)
StoreName
StoreStreetAddress
StoreCity
StoreCountry
...

Promotion Dimension
PromotionKey (PK)
PromotionName
PriceReductionType
PromotionMediaType
AdType
...

Step 4: Choose the Measures

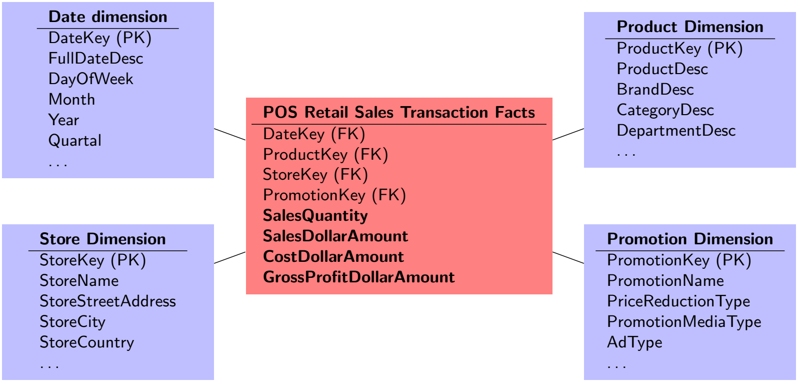
- Identify the business performance measures for the selected processes
- Determined by answering *"What are we measuring?"*
- Measures are determined by the grain declaration
- Revision of the grain might be required

Step 4: Choose the Measures – Example

- Sales quantity
- Sales dollar amount
- Cost dollar amount
- Gross profit
  - Equals to sales dollar − cost dollar
  - Explicit storage avoids user errors

- All measures are additive
- Facts are classified as event facts since each POS transaction is stored (most detailed level)

MD Schema of the Grocery Store Example



Outline

Database Sizing

- Time dimension: 2 years = 730 days
- Store dimension: 300 stores reporting each day
- Product dimension: 30,000 products, only 3,000 sell per day
- Promotion dimension: 5,000 combinations, but a product only appears in one combination per day
- Number of fact records: $730 \times 300 \times 3,000 \times 1 = 657,000,000$
- Number of fields: 4 key + 4 measures = 8 fields
- Total DB size: 657,000,000 $\times$ 8 fields $\times$ 4 bytes = 21 GB
- Small database by today's standards!

Date Dimension

- Date dimension is present in all DWs
- Can be created in advance
- "Meaningful" values are important for report generation, etc.
  - e.g., Holiday/Nonholiday vs. Yes/No
- Time-of-day a separate dimension
  - Separation keeps both dimensions small
- Date dimension vs. SQL date type
  - Many date attributes are not supported in SQL, e.g., fiscal month
  - Business user is not versed in SQL
- Date dimension is relatively small
  - 10 years = 3,650 rows

Date Dimension
DateKey (PK)
Date
Full Date Description
Day Of Week
Day Number in Epoch
Week Number in Epoch
Day Number in Calendar Month
Day Number in Calendar Year
Last Day in Week Indicator
Last Day in Month Indicator
Calendar Week Ending Date
Calendar Quarter
Calendar Year-Quarter
Calendar Half Year
Calendar Year
Fiscal Week
Fiscal Month
Fiscal Quarter
Fiscal Half Year
Fiscal Year
Holiday Indicator
Weekday Indicator
Selling Season
...

Instance of Date Dimension

Date Dimension Table

DK	Date	FullDateDescription	DayOfWeek	DayNum	HolidayInd	WeekdayInd	...
1	29.09.2013	September 29, 2013	Sunday	29	Nonholiday	Nonweekday	
2	30.09.2013	September 30, 2013	Monday	30	Nonholiday	Weekday	
3	01.10.2013	October 1, 2013	Tuesday	1	Nonholiday	Weekday	
4	02.10.2013	October 2, 2013	Wednesday	2	Nonholiday	Weekday	

Product Dimension

- Description of the products
- >50 attributes is typical for Product dimension
- Concept hierarchy
  - SKU → Brand → Category → Department
  - Many repetitions, but space of dimensions is not critical

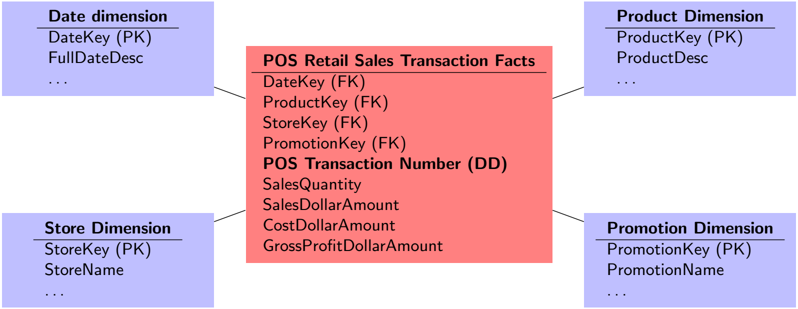
Product Dimension
Product Key (PK)
Product Description
SKU Number
Brand Description
Category Description
Department Description
Package Type Description
Package Size
Fat Content
Diet Type
Weight
Weight Units of Measure
...

Instance of Product dimension table

PK	Product Description	Brand Desc	Cat Desc	Dept Desc	...
1	Baked Well Light	Baked Well	Bread	Bakery	
2	Fluffy Sliced Whole Wheat	Fluffy	Bread	Bakery	
3	Fluffy Light Sliced Whole Wheat	Fluffy	Bread	Bakery	
4	Fat Free Mini Cinnamon Rolls	Light	Sweeten Bread	Bakery	
5	Diet Lovers Vanilla 2 Gallon	Coldpack	Frozen Desserts	Frozen Foods	
6	Light and Creamy Butter Pecan 1 Pint	Freshlike	Frozen Desserts	Frozen Foods	

Degenerate Dimensions

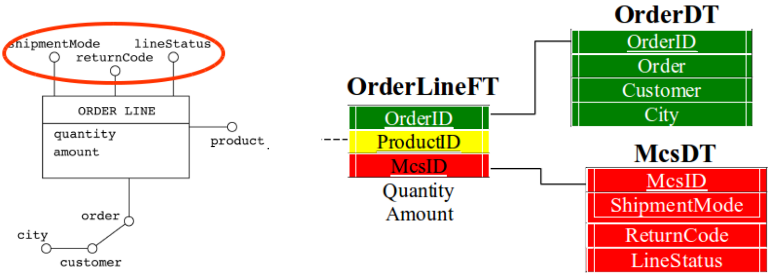
- Degenerate dimensions are "empty"
  - i.e., dimensions with a "hierarchy" of only one attribute
  - Values are directly stored in fact table, no dimension table is needed
- Examples are operational control numbers, e.g., order #, invoice #, POS transaction #, etc.
- Useful to serve as part of primary key in fact table or for grouping
  - e.g, grouping by POS transaction number to retrieve all products purchased in a single transaction



Junk Dimensions

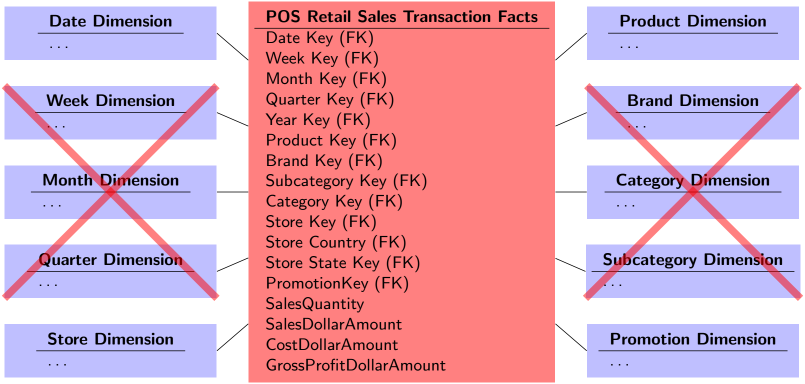
- Junk dimension: a dimension table that stores several degenerate dimensions
  - Within a junk dimension there is no functional dependency (hierarchy)
  - Only feasible if the number of distinct values for the attributes is small

- **Example:** 3 generate dimensions combined in a single junk dimension



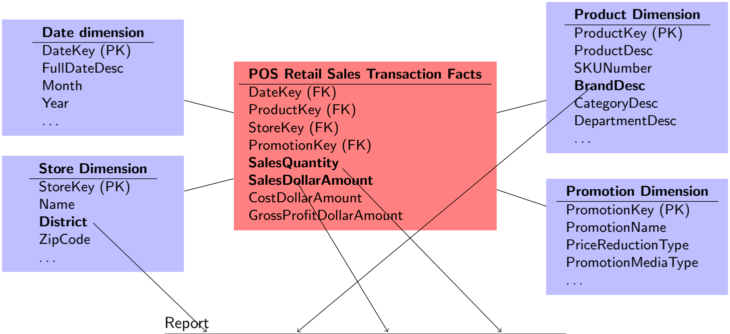
How Many Dimensions?

- Dimensions are important to provide detailed information for the analysis!
- But, too many dimensions is bad!
  - A sign that dimensions are not independent, and hence should be combined
  - Significantly increases space requirements of fact table
    - Size of dimension table is not a problem
  - 15 dimensions should normally be enough



Working with a Dimensional Model/1

- Each dimension attribute is a rich source for constructing row headers
- A common activity is to drag and drop dimensional attributes and measures into a simple report (+ specification of aggregate functions for measures)



Report

District	BrandDesc	SalesAmount	QuantitySold
Gries	Kellogs	3198	1023
Haslach	Danone	1099	671
Oswald	Kellogs	2976	800
...			

Working with a Dimensional Model/2

- Drilling down is adding row headers from a dimension
- Rolling up is removing row headers

Dept desc	Sales Dollar Amount	Sales Quantity
Bakery	$12.331	5088
Frozen Foods	$31.776	15.565

Roll up on Product dimension ↑ ↓ Drill down on Product dimension

Dept Desc	Brand Desc	Sales Dollar Amount	Sales Quantity
Bakery	Baked Well	$3.009	1.138
Bakery	Fluffy	$3.024	1.476
Bakery	Light	$6.298	2.474
Frozen Foods	Coldpack	$5.321	2640
Frozen Foods	Freshlike	$10.476	5.234

Product Dimension
ProductKey (PK)
ProductDesc
SKUNumber
BrandDesc
CategoryDesc
DepartmentDesc
...

Surrogate Keys

- **Surrogate keys** are integers that are assigned sequentially in a dimension table, e.g., 1, 2, 3, ...
- Should be used instead of natural operational production codes
- Many advantages over operational codes.
  - Make the DW independent from operational changes
    - e.g., re-use of old operational keys after some time
  - Avoid key overlap problem when consolidating data
  - Dimension keys should not contain “intelligence”
    - Should be stored explicitly as additional attribute
  - Performance: Small integer vs. long alpha-numeric code
    - Results in smaller fact tables
    - 1 Byte in a 1 billion fact table translates into 1 GB disk space

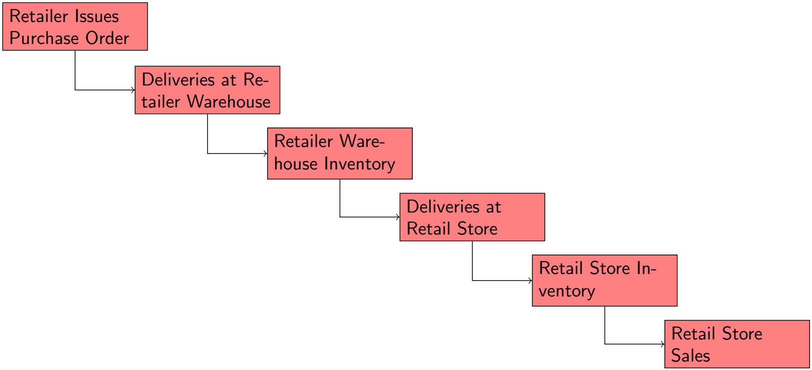
Outline

Inventory Management

- Consider a large grocery chain with a central warehouse and several retail stores
- Advanced retail business requires inventory information
  - Making sure the right product is in the right store at the right time minimizes out-of-stocks and reduces overall inventory carrying costs
  - The retailer needs the ability to analyze daily quantity-onhand inventory levels by product and store
- Design dimensional models that support the analysis of inventories for retail businesses (grocery stores)

The Value Chain

- The value chain identifies the natural, logical flow of an organization's primary activities
- Provides useful information for the identification of business processes
- Operational systems provide snapshots at each step with interesting data and performance metrics



Inventory Models

- 3 different inventory models
  - Model 1: Inventory periodic snapshot model
  - Model 2: Inventory transactions model
  - Model 3: Inventory accumulating snapshot model

Inventory Periodic Snapshot Model/1

- **Model 1: Inventory Periodic Snapshot**: Every day (or at some other regular time interval) the inventory levels of each product is measured and stored as a new row in the fact table
- **Example:** Inventory of retail store
  - Business process: Analysis of retail store inventory
  - Granularity: Daily inventory by product at each individual store
  - Dimensions: Date, product, and store
  - Facts/measures: Quantity on hand

Inventory Periodic Snapshot Model/2

- Inventory generates dense snapshot tables (i.e., entry for each product)
  - In contrast, POS Retail Sales table was sparse
- Hence, inventory fact table is growing fast!
  - 60.000 products $\times$ 100 stores = 6 Mio. rows each time
  - With a row width of 14 bytes, this is 84 MB each time
  - 1 year of daily snapshots would be 30 GB
- Reduce snapshot frequencies over time
  - Last 60 days of inventory at daily level
  - Weekly snapshots for older data
  - Instead of 1.095 snapshots in 3 years, only 208 snapshots would be required

Inventory Periodic Snapshot Model/3

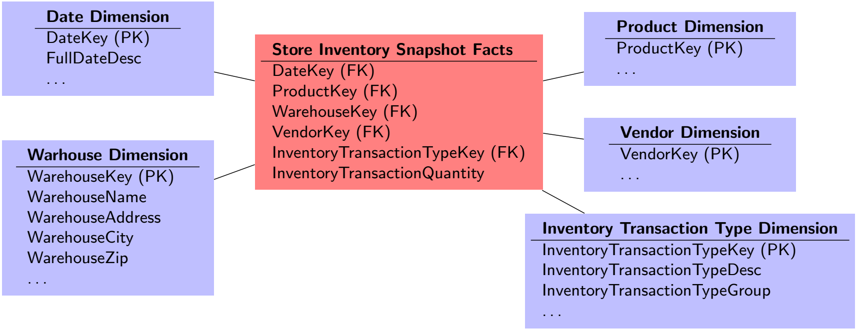
- The quantity on hand is a semi-additive measure
  - Can be summarized across products and stores, but not across time
  - Different in POS Retail Sales table: Sold entities are counted only once
- All measures that record a static level (inventory, financial account balance, measures of intensity, e.g., temperature) are inherently non-additive across time and possibly other dimensions
  - Can be aggregated along time dimension by averaging
- A note about SQL AVG function:
  - Cannot be used to compute the average over time, since it averages over the number of rows
  - Avg inventory over a cluster of 3 products in 4 stores across 7 days would divide the summed value by 84

Inventory Transactions Model/1

- **Model 2: Inventory Transactions:** Every transaction that affects the inventory is recorded
- **Example:** Inventory transactions in the store chain
  - Receive product
  - Place product in to inspection hold
  - Release product from inspection hold
  - Return product to vendor due to inspection failure
  - Place product in bin
  - Authorize product for sale
  - Pick product for shipment
  - Ship product to customer
  - Receive product form customer
  - Return product to inventory from customer return
  - Remove product from inventory

Inventory Transactions Model/2

- Star schema of the inventory transaction model



- Contains most detailed information, e.g.,
  - How many shipments from a given vendor?
  - On which products more than one round of inspection?
- Reconstruction of exact inventory numbers is possible, but not practical!
  - Used in combination with other fact table

Inventory Accumulative Snapshot Model/1

- **Model 3: Inventory Accumulating Snapshot:** One row in the fact table for each shipment of a particular product to the warehouse

- Assumption that the inventory goes through a series of events, e.g., receiving, inspection, bin placement, authorization to sell, picking, boxing, and shipping.
- A row tracks the disposition of a shipment through these events in the warehouse.
- Row is updated as the shipment moves through the warehouse until it leaves the warehouse.
- Characterized by many date dimensions and many updates.

Inventory Accumulative Snapshot Model/2

Date Received Dimension
DateKey (PK)
...

Date Inspected Dimension
DateKey (PK)
...

Date Placed in Inventory Dimension
DateKey (PK)
...

Date authorized to Sell Dimension
DateKey (PK)
...

Date Picked Dimension
DateKey (PK)
...

Date Boxed Dimension
DateKey (PK)
...

Date Shipped Dimension
DateKey (PK)
...

Date of Last Return Dimension
DateKey (PK)
...

Warehouse Inventory Accumulating Facts
Date Received Key (FK)
Date Inspected Key (FK)
Date Placed in Inventory Key (FK)
Date Authorized to Sell key (FK)
Date Picked Key (FK)
Date Boxed Key (FK)
Date Shipped Key (FK)
Date of Last Return Key (FK)
Product Key (FK)
Warehouse Key (FK)
Vendor Key (FK)
Quantity Received
Quantity Inspected
Quantity Returned to Vendor
Quantity Placed in Bin
Quantity Authorized to Sell
Quantity Picked
Quantity Boxed
Quantity Shipped
Quantity Returned by Customer
Quantity Returned to Inventory
Quantity Damaged
Quantity Lost
Unit Cost
Unit List Price
Unit Average Price
Unit Recovery Price

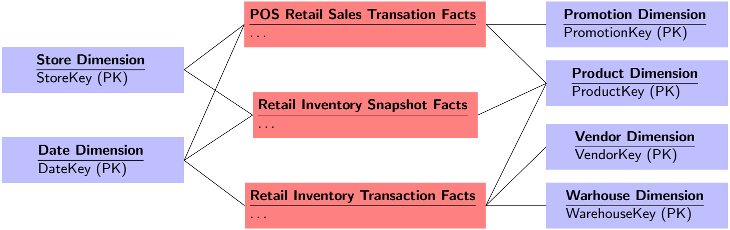
Product Dimension
ProductKey (PK)
...

Warhouse Dimension
WarehouseKey (PK)
...

Vendor Dimension
VendorKey (PK)
...

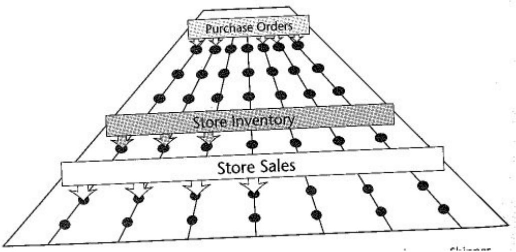
Value Chain Integration and Shared Dimensions

- Integration accross the value chain is important for the analysis
  - Allows analysis across the business to better evaluate the overall performance (not just at the individual department level)
  - End-to-end perspective high-level management to customer
- This requires the integration and consistent handling/use of data
- Solution: Individual fact tables for processes + shared dimensions
- **Shared dimensions** are used by different data marts



Data Warehouse Bus Architecture

- **Data Warehouse Bus Architecture** is a standard bus interface that supports the incremental development of a DW
- Based on conformed (similar) dimensions that are shared by the DMs
- Useful tool for the design process as it breaks down the process into small chunks (DMs)
- DMs can be realized at different times and by different groups



Data Warehouse Bus Matrix/1

- **Data Warehouse Bus Matrix** is a way to document the bus architecture
  - Rows represent business processes (translate into DMs)
  - Columns represent a suite of standardized, common and shared dimensions

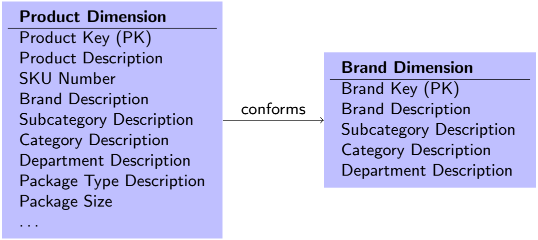
	Shared Dimensions							
Business Processes	Date	Product	Store	Promotion	Warehouse	Vendor	Contract	Shipper
Retail Sales	X	X	X	X				
Retail Inventory	X	X	X					
Retail Deliveries	X	X	X					
Warehouse Inventory	X	X			X	X		
Warehouse Deliveries	X	X			X	X		
Purchase Orders	X	X			X	X	X	X

Data Warehouse Bus Matrix/2

- Creating the DW bus matrix is one of the most important up-front deliverables of a DW implementation
- Create a comprehensive list of dimensions before filling in the matrix
- The rows provide a concise overview about the dimensionality of the individual DMs
- The columns show the interaction between the DMs and the common/shared dimensions

Conformed Dimensions

- **Conformed Dimensions** are either identical or strict mathematical subsets of the most granular, detailed dimension
- Roll-up dimensions conform to the base-level dimension
- **Example:** The sales process captures data at the product level, while the forecasting process does it at the brand level
    - Brand table conforms to the atomic product table as it is a strict subset of product table



Outline

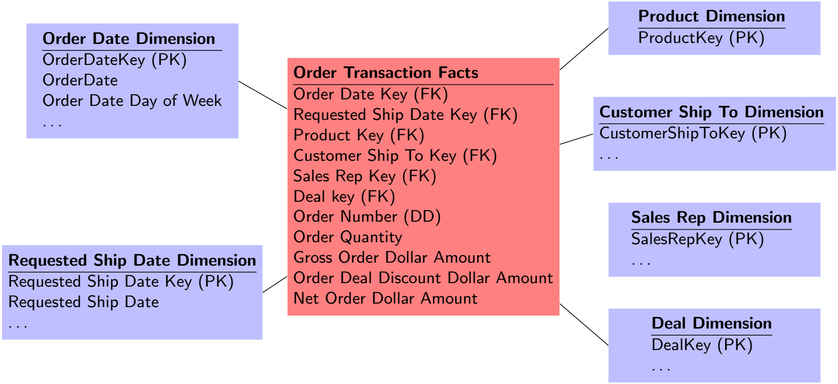
Order Management/1

- Order management consists of several critical business processes (order, shipment, invoiceprocessing, etc.) and measures (sales volume, invoice revenue, etc.)
- Data warehouse bus matrix

	Shared Dimensions						
Business Processes	Date	Product	Customer	Deal	Sales Rep	Ship From	Shipper
Quotes	X	X	X	X	X		
Orders	X	X	X	X	X		
Shipment	X	X	X	X	X	X	X
Invoicing	X	X	X	X	X	X	X

Order Management/2

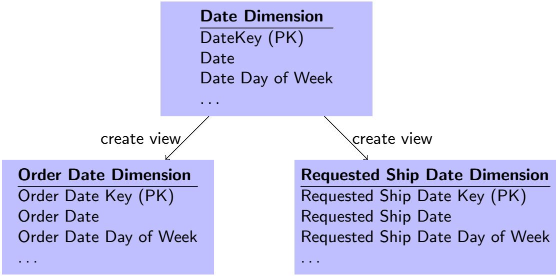
- Star Schema



- Issues
  - Should there be a “Ship Date Key” in the fact table?
  - Can/should Order Date Key and Requested Ship Date Key be foreign keys to the same dimension table?

Role-Playing

- **Role-playing** in DW occurs when a single dimension appears several times in the same fact table,
  - e.g., order date and requested ship date
- Should not be FK to the same dimension table
  - SQL would require the two dates to be the same
  - We might want to constrain the two dimensions differently
- The underlying dimension may exist as a single physical table, but each of the roles should be presented in a separate view with different labels



Fact Normalization

- **Fact normalization:** Normalize the fact table and collapse all measures into a single measure along with a special fact dimension that identifies the type of the measure
- Makes only sense if:
  - fact table is sparsely populated and no computations are made between differt measures
  - e.g., medical tests where different parameters are measured and data is sparse
- **Example:** Normalize Order Transaction Fact Table
  - Fact Dimension table has entries for Order Quantity, Gross Order Dollar Amount, Order Deal Discount Dollar Amount, Net Order Dollar Amount

Not normalized

Order Transaction Facts
Order Date Key (FK)
Requested Ship Date (FK)
...
Order Quantity
Gross Order Dollar Amount
Order Deal Discount Dollar Amount
Net Order Dollar Amount

Normalized

Order Transaction Facts
Order Date Key (FK)
Requested Ship Date (FK)
...
Entry Type (FK)
Amount

Fact Dimension
Entry Type Key (PK)
Entry Type
Unit of Measurement
...

Outline

OncoNet/1

- Collaboration with Hospital Meran (BSc thesis of A. Heinisch)
- OncoNet is an application for the management of patients undergoing a cancer therapy
- Cancer therapy follows a treatment plan/protocol, which specifies certain events (medications, tests, etc.) at regular time points

Event type	Event description	Date
Data collection	Ct Thorax	Day 60, 100, 360, 720
Data collection	Ct Abdomen	Day 60, 100, 360, 720
Data collection	Hemogram	Day 110
Medication	Zofran	Day 1
Medication	Adriblastina	Day 1

OncoNet/2

- Business process: Analysis of cancer therapies
- Queries to answer:
  - How many patients with normal blood pressure after medication X?
  - Which dosages of drug A were successful to reduce parameter Y?
  - etc.
- Granularity: Individual events of the chemotherapy
  - Includes measurements, examinations, questionnaires, etc.

OncoNet/3

- Patient and Drug Dimension

Patient Dimension
Patient Key (SK)
Patient ID
Patient First Name
Patient Last Name
Patient Gender
Patient Address
Patient ZIP Code
Patient Phone Number
Patient Profession
Patient First Language
Patient Height
Patient Weight
Patient Body Surface Area
Patient Place of Birth
Patient Birthday Date Key
Patient Death Date Key
Patient First Admission Doctor
Patient First Admission Area
Patient Smoking Indicator
Patient Cigarettes per Day
Patient Alcohol Indicator
Patient Alcohol Amount per Day
...

Drug Dimension
Drug Key (SK)
Drug ID
Drug Name
Drug Category
Drug Active Substance
Drug Manufacturer
Drug Quantity Unit
Drug Quantity Unit Description
Drug Administration Type
Drug Administration Location
Drug Packaging
Drug Packaging AIC Code
...

OncoNet/4

- Normalized fact table
  - Only one measure is used in the fact table
  - Type of measure is described in Event Dimension Table and Investigation Dimension Table

Chemotherapy Event Facts
Date Key (FK)
Prescribing Date Key (FK)
Relative Date Key (FK)
Patient Key (FK)
Therapy Key (FK)
Drug Key (FK)
Event Key (FK)
Investigation Key (FK)
Numverical Value
Textual Value

OncoNet/5

Patient Key	Patient ID	First Name	Last Name	Gender	Language	Weight	Height	...
1	345	Hans	Maier	male	German	68	185	
...								

Date Key	Date Type	Full Date	Day of Week	Weekday Ind	Month	Year	...
1	Normal	05.06.2012	Tuesday	Weekday	June	2012	...
...							

Date key	Patient Key	Therapy Key	Event Key	Investigation Key	Numerical Value	...
1	1	1	1	1	32,7	
...						

Therapy Key	Therapy ID	Therapy Name	Therapy Type	...
1	25436	NHL Chop 14	Profile	
...				

Investigation Key	Group Level	Label	Unit	...
1	Haemogram	MCH	1	
2	Haemogram	HB	g/dl	
3	Haemogram	MCV	pg	

Event Key	Event ID	Event Name	Event Type	Responsible	...
1	9372	Urgent Laboratory	Laboratory Test	Nurse	

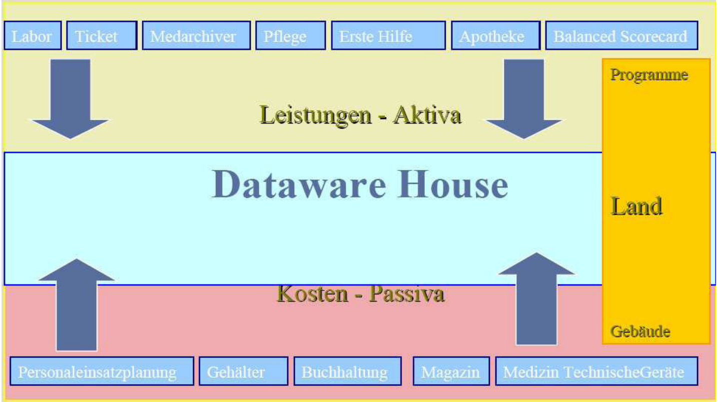
Outline

MEDAN (MEdical Data Warehousing and ANalysis)

- Collaboration between the Hospital Meran and the FUB
- Objectives
  - Conduct research and create competences in the field of medical data warehousing and analysis
  - Build a BI/DW solution for the hospital
    - Administrative DW
    - Medical DW
  - Develop and apply data analysis/mining techniques

MEDAN Data Sources

- Data sources in a health care environment
  - Internal production systems (SQL, Excel, Text files, ...)
  - External information systems



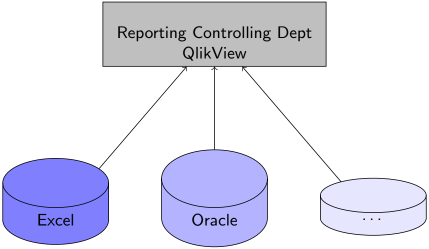
Budgeting Using a Spreadsheet

- **Example:** Budgeting in the controlling department
  - Complex and error-prone Excel spreadsheets have been used in the past
  - Monthly reports have to be sent to the Province

CdC 8090		2008				BUDGET			2009				BUDGET 2010
VOLUMI, MIX E QUALITA' PRODUZIONE	FINO AL		PIANO	PESO	Limite inferiore	Limite superiore	PROIEZ.	SCOST.	MATURATO	BUDGET	PESO	Limite inferiore	Limite superiore
Dimessi ordinari	31.08.	293	296	0%	0	0	270	-8,8%	0%	270			
Trasferimento	31.08.	1	2				0	-100,0%	0%	0			
gg di degenza	31.08.	990	939				993	5,8%	0%	993			
n. posti letto	31.08.	5	5				5		0%				
Accessi day hospital/surgery	31.08.	887	981	0%	0	0	1.168	19,0% !	0%	1.168			
n. posti letto day hosp./surg.	31.08.	4	6				6		0%	6			
Totale attività per esterni	31.08.	45.670	46.986	0%	0	0	44.039	-6,3%	0%	44.039			
Totale attività per interni	30.09.	566	548	0%	0	0	559	1,9%	0%	559			
Totale attività ricevuta	30.06.	2.561	0	0%	0	0	2.502		0%	2.502			
- di cui di laboratorie	31.08.	2.295	2.202	0%	0	0	2.340	6,3%	0%	2.340			
- di cui di radiologie	31.08.	185	171	0%	0	0	102	-40,4%	0%	102			
n° prest. di lab. x dimessi ordinari	31.08.	7,83					8,67		0%	8,67	40%		8,67
n° prest. di rad. x dimessi ordinari	31.08.	0,63					0,38		0%	0,38	30%		0,38
COSTI ED EFFICIENZA													
Consumi beni sanitari	30.09.	486.304	411.792	50%	0	432.382	501.299	21,7% !		501.299			
PHT + H-OSP2	30.09.						0		0%	0			
Consumi beni non sanitari	30.09.	3.648	3.728	0%	0	0	3.284	-11,9%	0%	3.284			
altri costi	30.09.	4.459	3.831	0%	0	0	6.524	70,3% !	0%	6.524			
Totale consumi		494.411	419.351	0%	0	0	511.107	21,9% !	0%	511.107			
costi personale (non da pianificare)	30.06.	994.834					1.029.189						
unità personale	30.09.	5,75	7,00	0%	0,00	0,00	6,26	-10,6%	0%	6,44			
presenza media	31.08.	6,16					6,44						
Tasso utilizzo letti	31.08.	54,10%	51,31%	0%	0,00%	0,00%	54,26%		0%	54,26%			
degenza media	31.08.	3,33	3,16	0%	0,00	0,00	3,60		0%	3,60			
tasso op	30.06.	70,76%	87,23%	0%	0,00%	0,00%	70,73%		0%	70,73%			
% ricoveri di 1 giorno	31.08.	7,64%	2,00%	50%	0,00%	3,00%	6,40%			6,40%	30%		6,40%
peso medio drg	30.06.	0,68	0,81				0,89		0%	0,89			
mobilità provinciale passiva	30.06.	151.398	0	0%	0	0	133.248		0%				
mobilità provinciale attiva	30.06.	680.010	0				954.940		0%				
mobilità Innsbruck	30.06.	30.775	0				24.448		0%				
servizio trasporti: n° pazienti	31.08.	32	0	0%	0	0	21		0%				
servizio trasporti: €	31.08.	1.566	0				999		0%				
Summe Gewichtung				100%					0%				

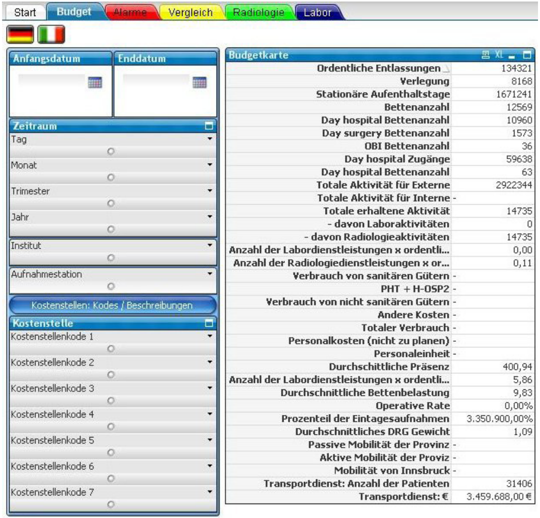
Budgeting Using QlickView/1

- A QlickView application to replace Excel
  - QlickView is a state-of-the-art data analysis tool
  - All data are kept in main memory $\rightarrow$ fast
  - Easy-to-use for small/medium sized applications
  - No solution for ETL/data staging
- Direct access to the data sources
- Data integration in QlickView
- No DW in place



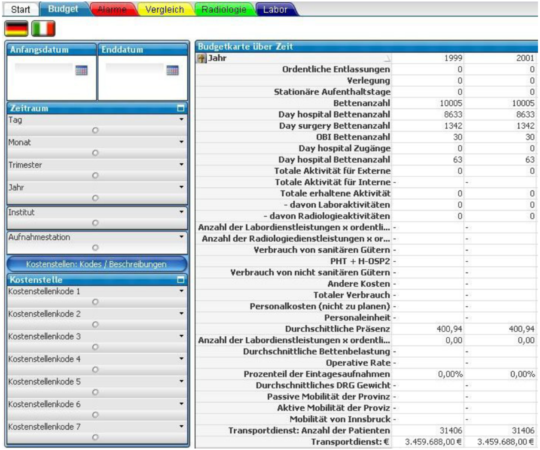
Budgeting Using QlickView/2

- QlickView application: Actual budget
  - Big improvement over Excel solution



Budgeting Using QlickView/2

- QlickView application: Budget over time

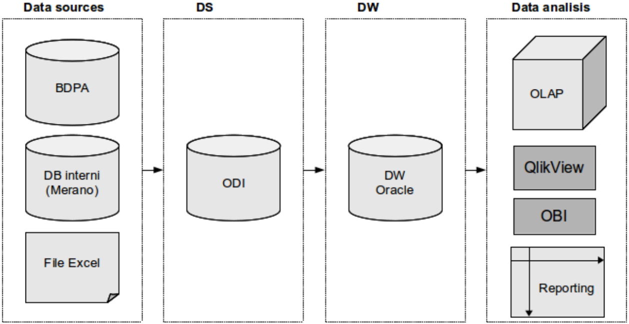


MEDAN DW Solution

- It was difficult to convince decision makers to do proper modeling and build a DW as the core of a BI solution
  - They were too much technology-driven (QlikView, ...)
- QlickView is mainly an analysis tool and cannot replace a DW
  - Good for quick and small ad-hoc solutions
  - Difficult to do data cleaning and hence to control data quality
  - Not scalable for many applications, changing sources, etc.
- Since Oracle technology was already in place, we finally were able to convince them to use this technology

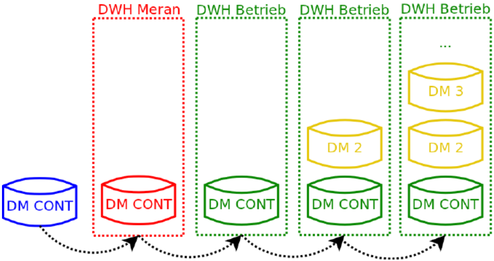
MEDAN Architecture

- Oracle ODI for ETL part and Data Storage
- Oracle DB for the DW
- QlickView and OBI for data analysis



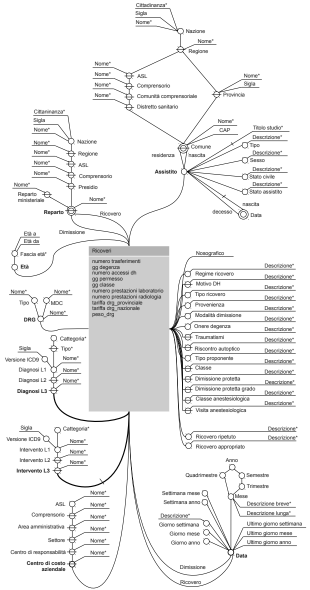
MEDAN Bottom Up Approach

- We developed a prototype of DM CONT for the Hospital of Meran, consisting of 3 cubes:
  - Hospital Stays
  - Services
  - Transfers
- Each cube corresponds to a business process
- Goal: Deploy DM CONT in the Hospital of Meran, then in the other hospitals in South Tyrol
- Repeat the same cycle for the other DMs: DM Personnel, DM Pharmacy, DM Laboratory, etc.

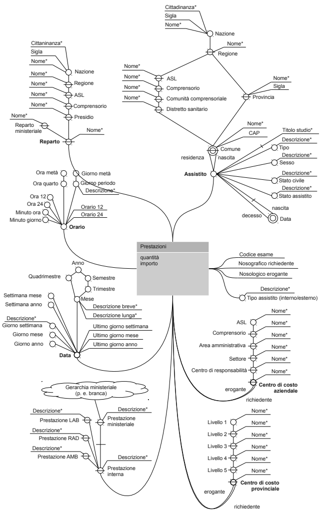


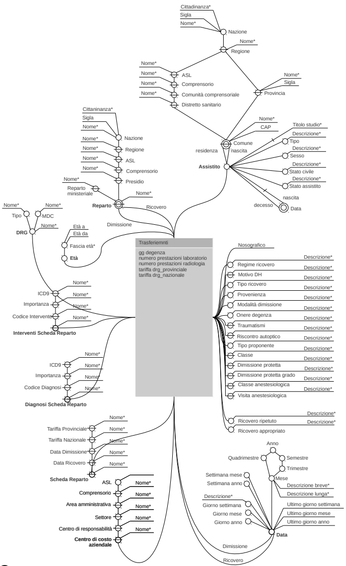
Conceptual Model of Hospital Stays

- Each event stores a hospital stay of a patient
- Similar model for the other data cubes:
  - Services
  - Transfer
- Shared dimensions are used



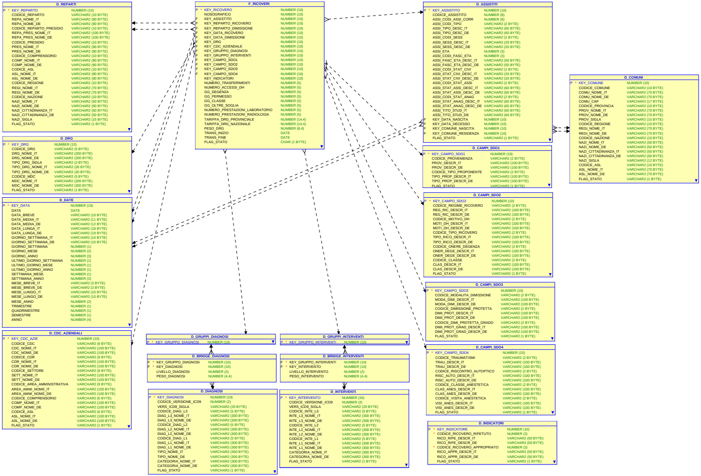
Conceptual Model of Services and Transfers





Logical Model of Hospital Stays

- Snowflake schema



Multi-valued Dimensions

- Diagnoses are an example of multi-valued attributes/dimensions, i.e., many-to-many relationships between attributes (multiple arcs in DFM)
  - a patient typically has multiple diagnoses (up to > 10)
  - books and authors is another example
- Solutions
  - Reserve multiple columns (one for each diagnosis)
    - Results in many empty cells, i.e., sparse fact table
  - Use several facts for a single hospital stay (one for each diagnosis)
    - Similar to fact normalization for measures
    - Increases the number of tuples in fact table

Multiple columns

Hospital Stay Facts
Data Hospital Stay Admission (FK)
Data Hospital Stay Discharge (FK)
Health Record Number (FK)
Health Record Year (FK)
Patient Key (FK)
...
Diagnosis Key 1 (FK)
Diagnosis Key 2 (FK)
Diagnosis Key 3 (FK)
...

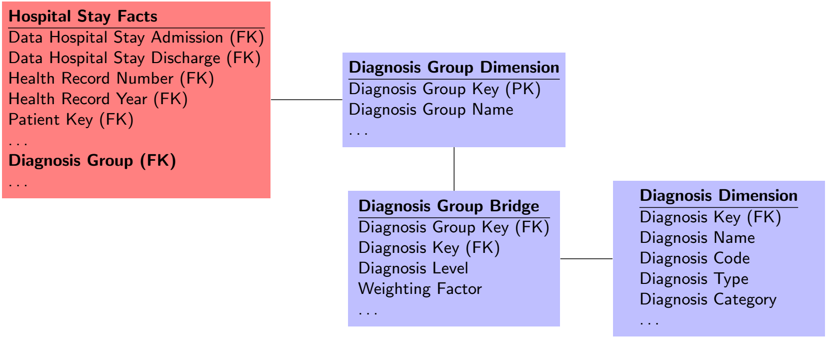
Multiple facts

Hospital Stay Facts
Data Hospital Stay Admission (FK)
Data Hospital Stay Discharge (FK)
Health Record Number (FK)
Health Record Year (FK)
Patient Key (FK)
...
Diagnosis Key (FK)
...

  - Using bridge tables (next slide)

Bridge Tables

- **Bridge tables** help to deal with multi-valued dimensions
- Bridge table implements two one-to-many relationships



MEDAN Lessons Learned

- Developing a BI platform is a process that takes years
- A well-designed and consistent DW is the foundation for BI
  - QlikView is a tool for quick analyses, but cannot replace a DW
- Do not put anything in a single data mart
  - Use one DM for one business process (set of closely related business queries)
- Different opinions on bottom-up vs. top-down, but bottom-up seems to have more acceptance
- Data modeling is difficult but unavoidable
  - Helps to get a conformed view on the business
    - e.g., what is an admission/hospital stay?
  - Different granularity by different users, e.g., Province, hospital
- A Business Intelligence Competence Center (BICC) was missing, but would have been very helpful to
  - coordinate the whole project
  - take important decisions about the data

Summary/1

- Simplified DW design process by Kimball and Ross consists of 4 steps:
  - Choose business processes, granularity, dimensions, and measures
- Surrogate key should be used instead of operational codes
- Degenerate dimensions are stored in the fact table
  - or stored all together in a junk dimension
- DW sizing
  - dimensions are a small portion of the DW, hence can/should contain as much information as possible
  - fact table determines the size of the DW
- Dimensional model is easy to use, e.g., drag and drop for report generation

Summary/2

- Different Inventory models: periodic snapshot model, transactions model, accumulating
- Shared and conformed dimensions are crucial to integrate several DMs across a value chain
- DW bus architecture is a standard interface to support incremental DW design
- DW bus matrix is a way to document the DW bus architecture
- Role-playing allows to physically store a dimension only once, but use it several times in different roles and with different names
- Fact normalization collapses all measures into a single measure together with a special fact dimension to determine the type of the measure
  - Only useful when the fact table is sparse
- Multi-valued dimensions if a dimension occurs more than once in a single fact, e.g., a patient has typically several diagnoses
  - Corresponds to multiple arcs in DFM (many-to-many relationships)
- Bridge tables can be used to represent such dimensions