

Data Warehousing and Decision Support

Torben Bach Pedersen
Department of Computer Science
Aalborg University



Talk Overview



- Data warehousing and decision support basics
 - Definition
 - Applications
- Multidimensional modeling
 - Dimensional concepts
 - Implementation in RDBMS
- Case study
 - The grocery store
- Exercise:
 - Build a **data warehouse** for a **real** warehouse !

Decision Support?



- Many terms for (almost) the same thing
 - Decision support systems (DSS)
 - Data Warehousing (DW)
 - Business Intelligence (BI)
 - Analytical Intelligence
 - ...
- Composition of technologies
 - BI=DW+OLAP+data mining+ (visualization, what-if, CRM,...)
- The focus is on **supporting decision-making**
 - Different from "ordinary" OLTP databases/systems
 - "Large" analytical queries rather than "small" transaction queries

Decision Support Is Important!



- Meta Group: DW alone is US\$ 15 billion in 2000
- PAMG: Business Intelligence (DW, OLAP, data mining, CRM, visualization) is US\$ 113 billion in 2002
- Decision support can help you understand your customers better
 - Necessary to survive
- The web makes a DW even more necessary
 - Customers do not visit the business physically
 - Customers can easily change to other businesses
 - Web logs makes it possible to analyze user behavior (new !)
 - Combine web data with existing customer data in the DW

Data Analysis Problems



- The same data found in many different systems
 - Example: customer data in 14 (now 23) systems!
 - The same concept is defined differently (Nykredit)
- Data is suited for operational systems (OLTP)
 - Accounting, billing, etc.
 - Do not support analysis across business functions
- Data quality is bad
 - Missing data, imprecise data, different use of systems
- Data are "volatile"
 - Data deleted in operational systems (6 months)
 - Data change over time – no historical information

Data Warehousing



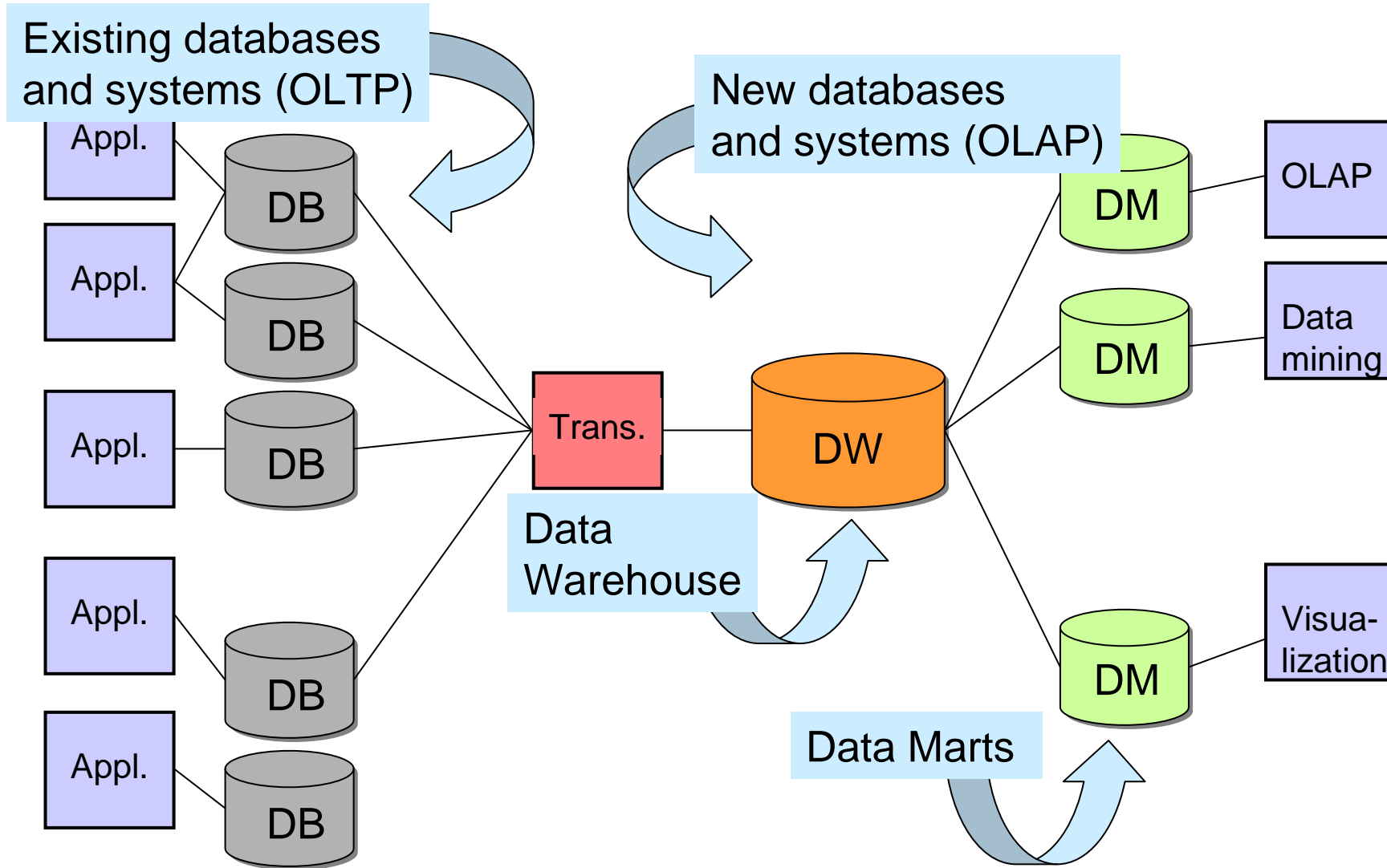
- Solution: new analysis environment (DW) where data are
 - Subject oriented (versus function oriented)
 - Integrated (logically and physically)
 - Stable (data not deleted, several versions)
 - Time variant (data can always be related to time)
 - Supporting management decisions (different organization)
- Data from the operational systems are
 - Extracted
 - Cleansed
 - Transformed
 - Aggregated?
 - Loaded into DW

DW: Purpose and Definition



- The purpose of a data warehouse is to support **decision making**
- Data is collected from a number of different sources
 - Finance, billing, web logs, personnel, ...
- It is made easy to perform advanced analyses
 - Ad-hoc analyses and reports
 - Data mining: identification of trends
 - Management Information Systems
- A data warehouse is a **store of information** organized in a unified data model.

Data Warehouse Architecture



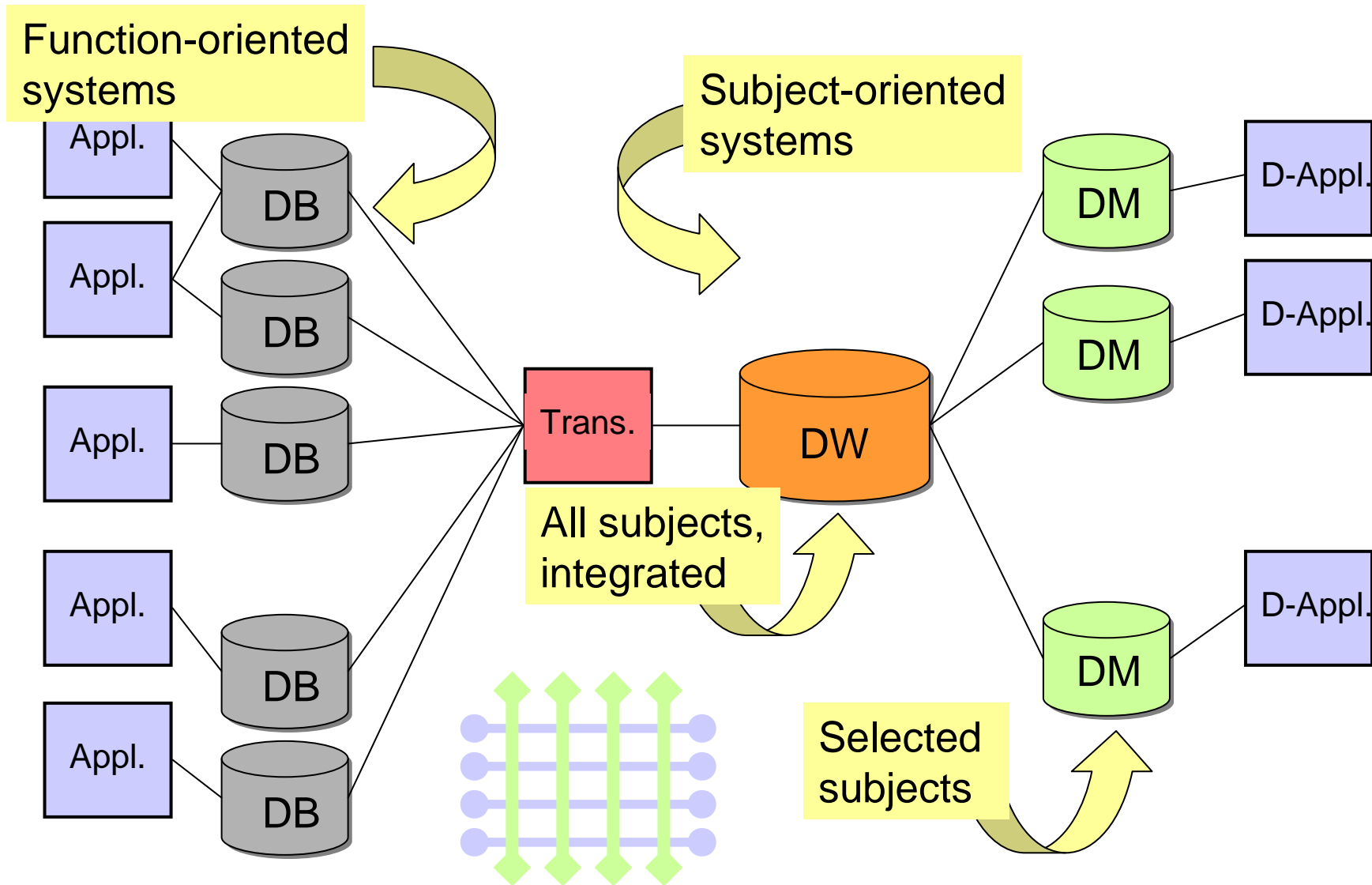
OLTP vs. OLAP



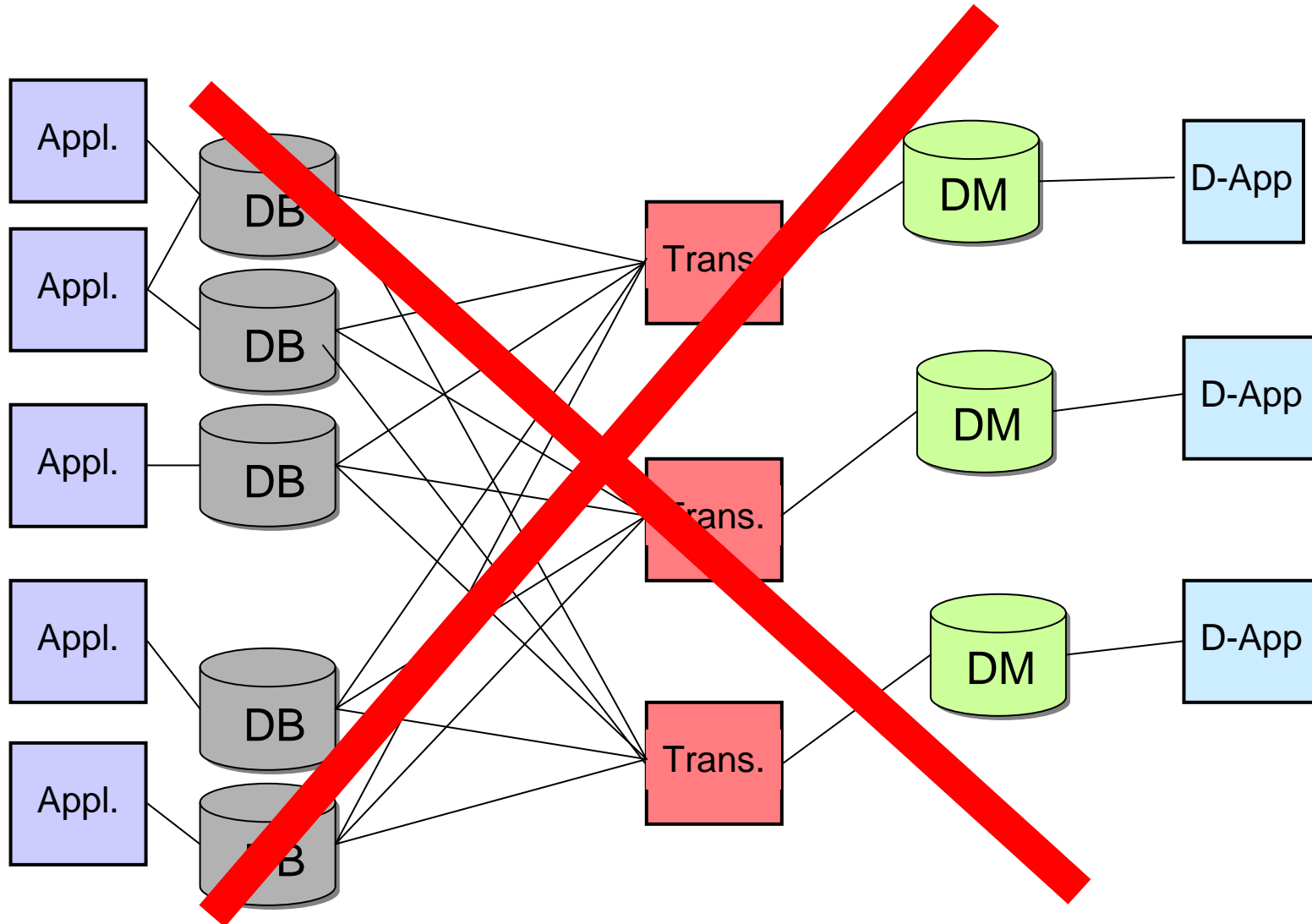
- On-Line Transaction Processing
 - Many, "small" queries
 - Frequent updates
 - The system is always available for both updates and reads
 - Smaller data volume (few historical data)
 - Complex data model (normalized)

- On-Line Analytical Processing
 - Fewer, but "bigger" queries
 - Frequent reads, in-frequent updates (daily)
 - 2-phase operation: either reading or updating
 - Larger data volumes (collection of historical data)
 - Simple data model (multidimensional/de-normalized)

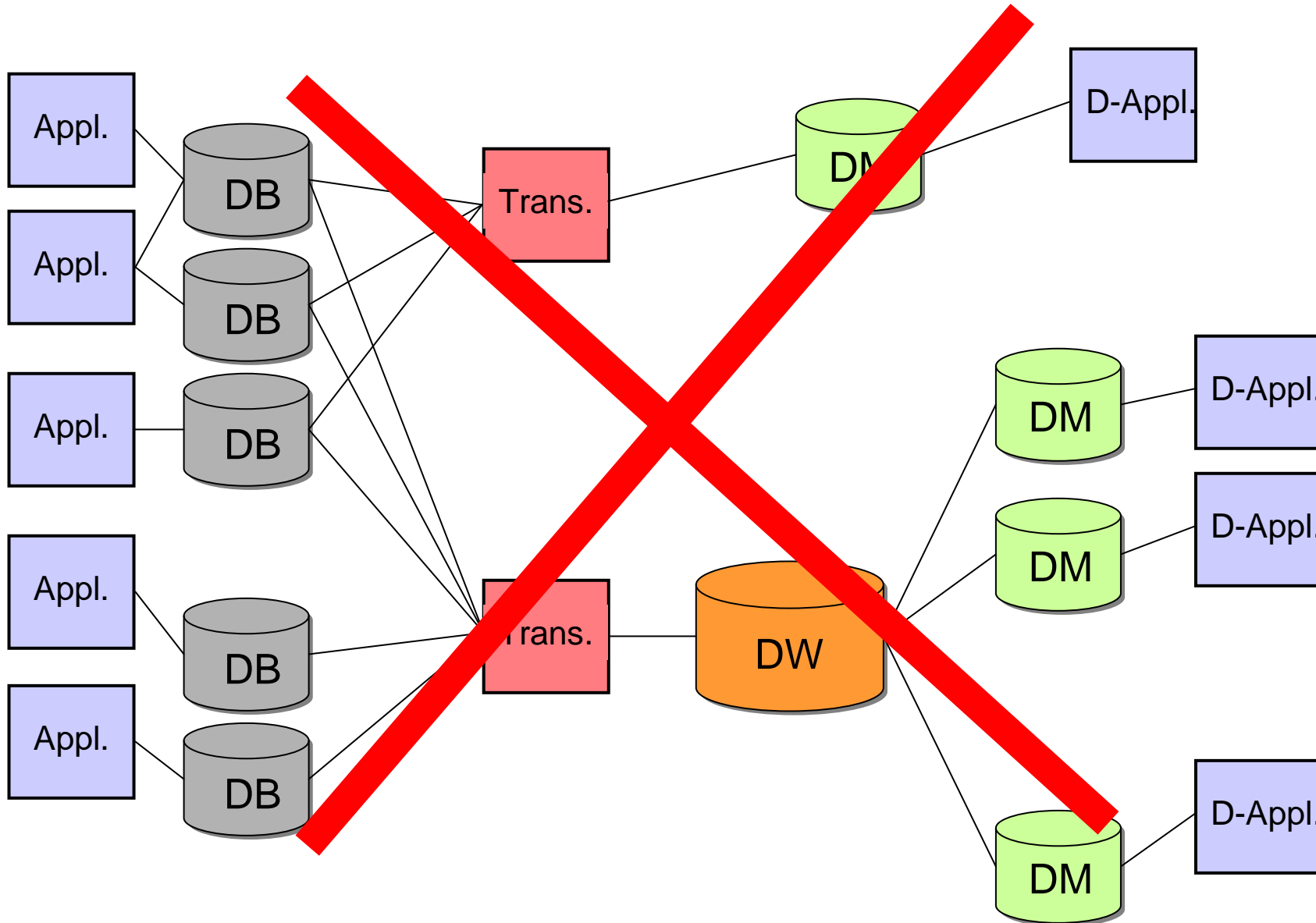
Function- vs. Subject Orientation



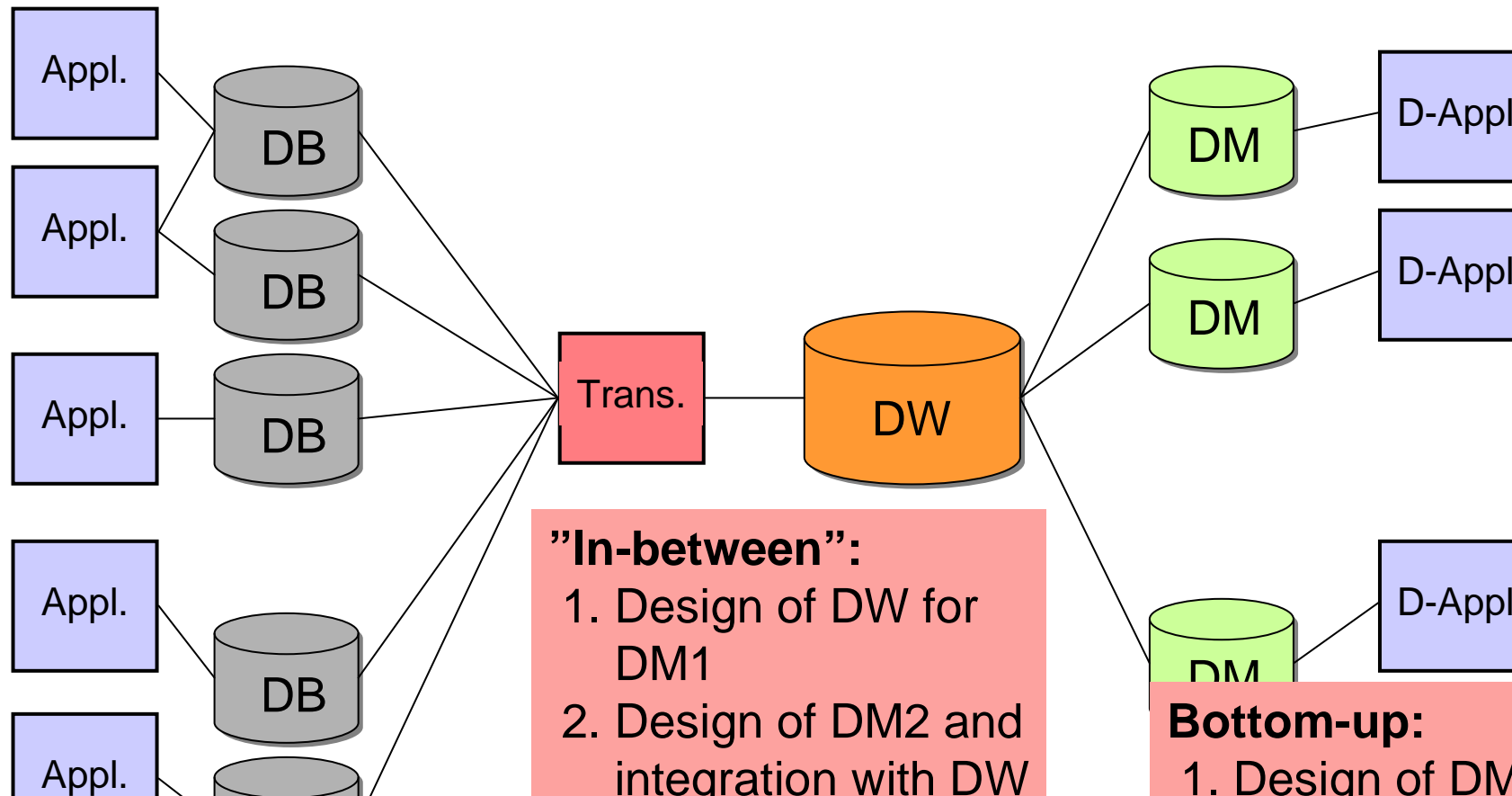
$n \times m$ versus $n + m$



Architecture Alternative



Top-down vs. Bottom-up



Top-down:
1. Design of DW
2. Design of DMs

"In-between":
1. Design of DW for DM1
2. Design of DM2 and integration with DW
3. Design of DM3 and integration with DW
4. ...

Bottom-up:
1. Design of DMs
2. Maybe integration of DMs in DW
3. Maybe no DW

Data's Way To The DW



- Extraction
 - Extract from many heterogeneous systems
- Staging area
 - Large, sequential bulk operations => flat files best ?
- Cleansing
 - Data checked for missing parts and erroneous values
 - Default values provided and out-of-range values marked
- Transformation
 - Data transformed to decision-oriented format
 - Data from several sources merged, optimize for querying
- Aggregation?
 - Are individual business transactions needed in the DW ?
- Loading into DW
 - Large bulk loads rather than SQL INSERTs
 - Fast indexing (and pre-aggregation) required

Online Analytical Processing



- Millions of clicks
 - Fast query response

The screenshot shows the 'Seagate Analysis' application window. It features a menu bar (File, Edit, View, Format, Favorites, Tools, Data, Help) and a toolbar with various icons. The main area displays a data table with columns for 'Type', '3', '4', '1999', and 'All Time'. The table has rows for 'Continuing' and 'New' types. Below the table are several filter controls: 'Time of Day: All Time of Day', 'Measures: Requests', 'User Domain: All User Domain', and 'Users: All Users'. A 'Cube' button is visible at the bottom left, and the status bar shows 'Ready'.

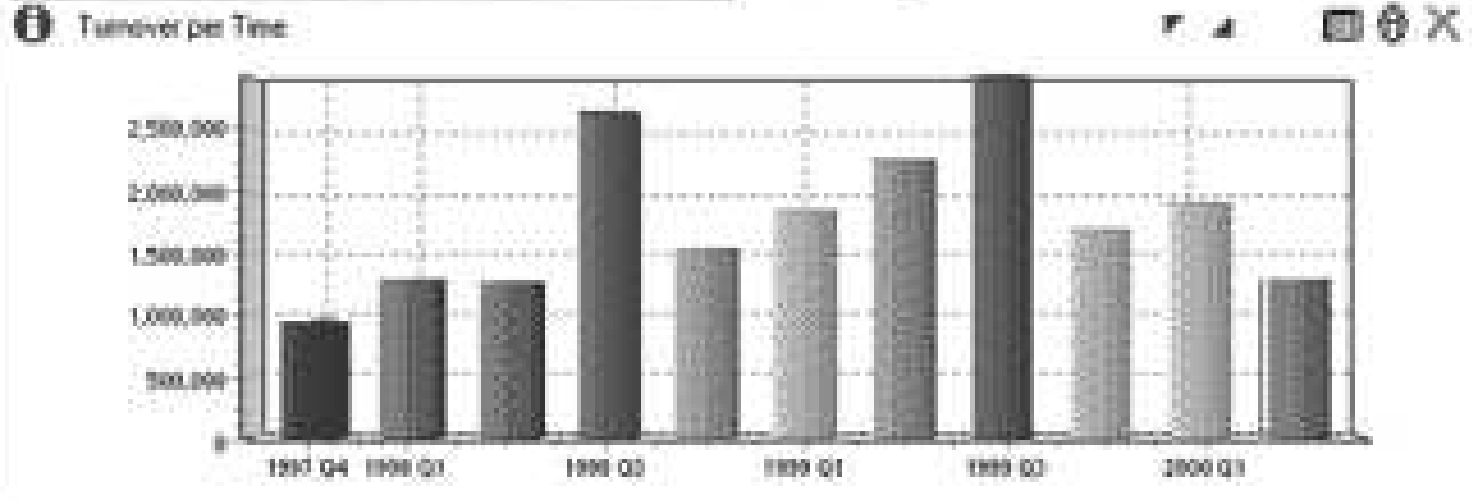
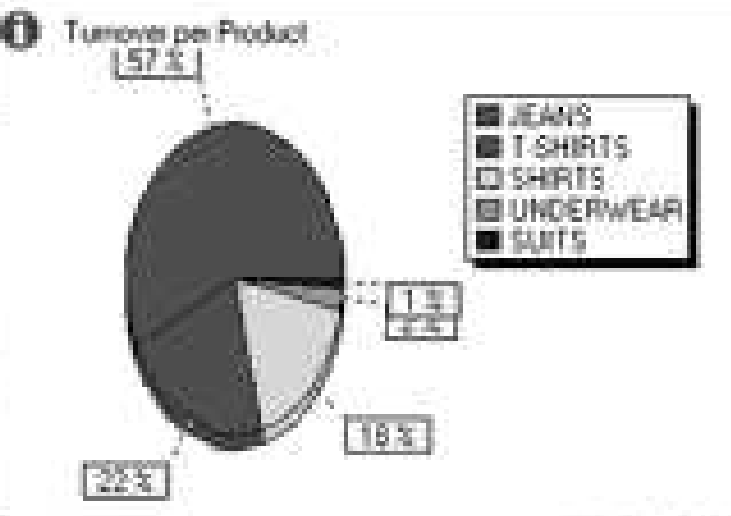
Type	3	4	1999	All Time
Continuing	1,783,680.00	3,191,896.00	4,975,576.00	4,975,576.00
New	148,752.00	265,800.00	414,552.00	414,552.00



Initial

- Sample Contribution
- Sample Delineating
- Sample Gages
- Sample Turnover**

Personal



DW Applications: OLAP

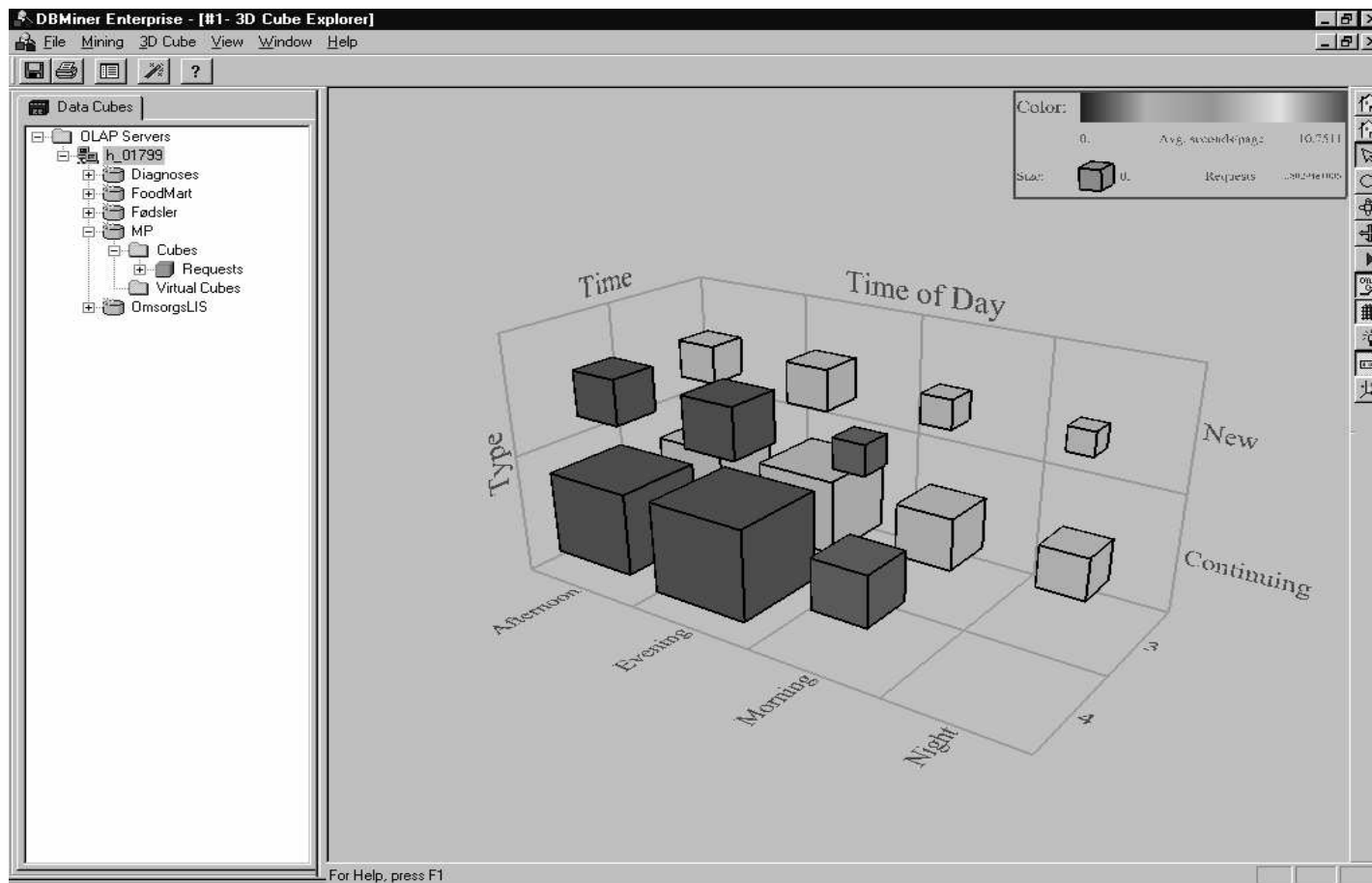


- Large data volumes, e.g., sales, telephone calls
 - **Giga-, Tera-, Peta-, Exa-**byte
- OLAP = On-Line Analytical Processing
 - Interactive analysis
 - Explorative discovery
 - Fast response times required
- OLAP operations
 - Aggregation of data
 - Standard aggregations operator, e.g., SUM
 - Starting level, (Quarter, City)
 - Roll Up: Less detail, Quarter->Year
 - Drill Down: More detail, Quarter->Month
 - Slice/Dice: Selection, Year=1999
 - Drill Across: “Join”

DW Applications: Visualization



- Graphical presentation of complex result
- Color, size, and form help to give a better overview



DW Applications: Data Mining



- Data mining is automatic knowledge discovery
- Roots in AI and statistics
- Classification
 - Partition data into pre-defined classes
- Prediction
 - Predict/estimate unknown value based on similar cases
- Clustering
 - Partition data into groups so the similarity *within* individual groups are *greatest* and the similarity *between* groups are *smallest*
- Affinity grouping/associations
 - Find associations/dependencies between data that occur together
 - Rules: $A \rightarrow B (c\%,s\%)$: if A occurs, B occurs with confidence c and support s
- Important to choose the granularity for mining
 - Too small granularity don't give any useful results (shirt brand,..)

Data Mining Examples



- Wal-Mart: USA's largest supermarket chain
 - Has DW with all ticket item sales for the last 2 years (big!)
 - Use DW and mining intensively to gain business advantages
 - Analysis of association within sales tickets
 - ◆ Discovery: Beer and diapers on the same ticket
 - ◆ Men buy diapers, and must "just have a beer"
 - ◆ Put the expensive beers next to the diapers
 - ◆ Put beer at some distance from diapers with chips, videos in-between!
 - Wal-Mart's suppliers use the DW to optimize delivery
 - ◆ The supplier puts the product on the shelf
 - ◆ The supplier only get paid when the product **is sold**
- Web log mining
 - What is the association between time of day and requests?
 - What user groups use my site?
 - How many requests does my site get in a month? (Yahoo)

Break



- Data Warehouse basics
 - Data analysis problems
 - DW architectures and characteristics
 - DW processes
- Data warehouse applications
 - OLAP
 - Visualization
 - Data mining

Talk Overview



- Data warehouse basics
 - Definition
 - Applications
- **Multidimensional modeling**
 - **Dimensional concepts**
 - **Implementation in RDBMS**
- Case study
 - The grocery store
- Exercise:
 - Build a **data warehouse** for a **real** warehouse !

Why a new model?



- We know E/R and OO modeling
- All types of data are “equal”
- E/R and OO models: many purposes
 - *Flexible*
 - *General*
- No difference between:
 - What **is** important
 - What just **describes** the important
- ER/OO models are **large**
 - 50-1000 entities/relations/classes
 - Hard to get an overview
- ER/OO models implemented in RDBMSes
 - Normalized databases **spread** information
 - When analyzing data, the information must be **integrated** again

The multidimensional model



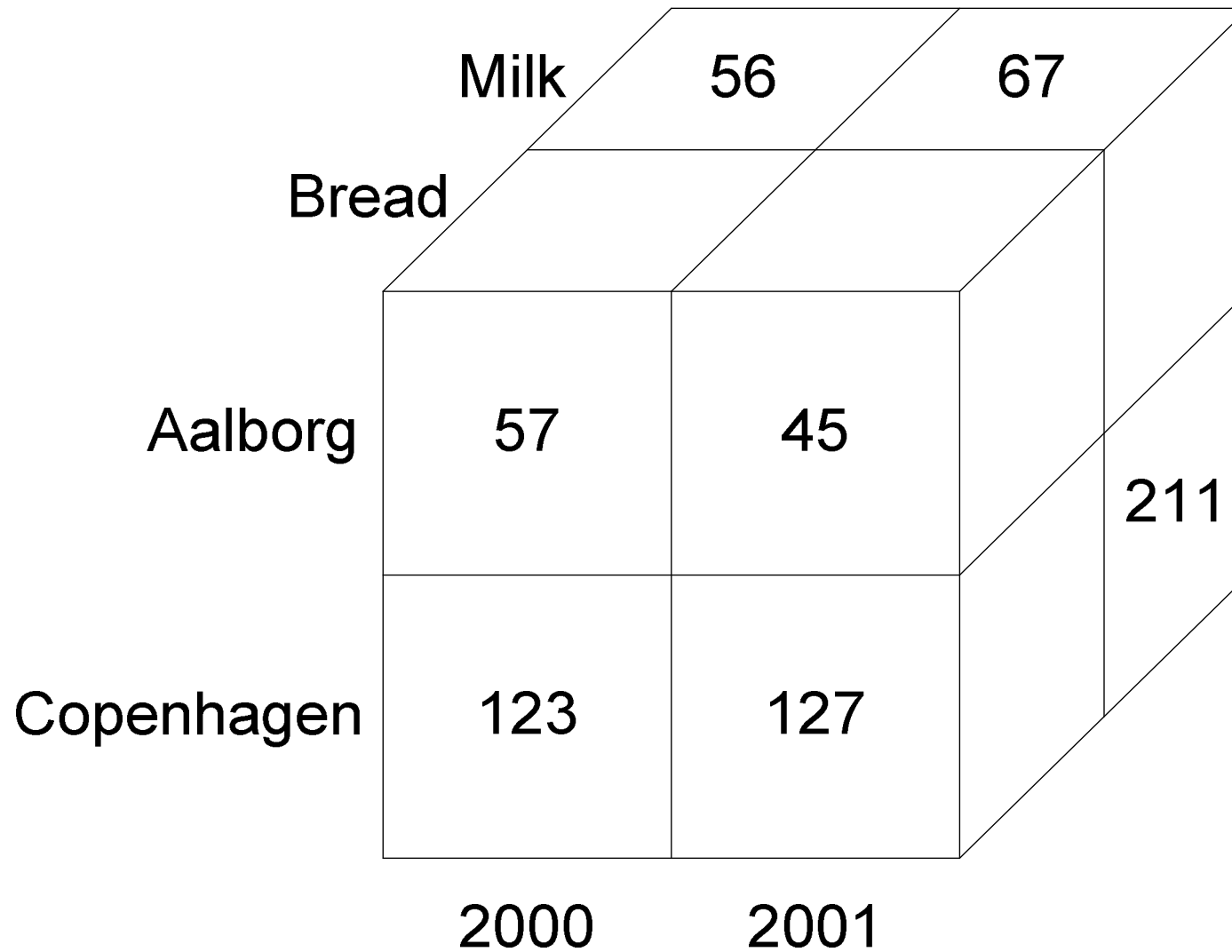
- One purpose
 - **Data analysis**
- Better at **that** purpose
 - Less flexible
 - Not suited for OLTP systems
- More **built in** “meaning”
 - What **is** important
 - What **describes** the important
 - What we want to **optimize**
 - Automatic aggregations means easy querying
- Recognized by OLAP/BI tools
 - Tools offer powerful query facilities based on MD design
 - Example: TARGIT Analysis

The multidimensional model



- Data is divided into:
 - **Facts**
 - **Dimensions**
- Facts are the **important** entity: a sale
- Facts have **measures** that can be aggregated: sales price
- Dimensions **describe** facts
 - A sale has the dimensions Product, Store and Time
- Facts “live” in a multidimensional **cube** (dice)
 - Think of an array from programming languages
- Goal for dimensional modeling:
 - Surround facts with as much context (dimensions) as possible
 - Hint: redundancy may be ok (in well-chosen places)
 - But you should **not** try to model **all** relationships in the data (unlike E/R and OO modeling!)

Cube Example



Cubes



- A “cube” may have **many** dimensions!
 - More than 3 - the term “hypercube” is sometimes used
 - Theoretically no limit for the number of dimensions
 - Typical cubes have 4-12 dimensions
- But only 2-3 dimensions can be viewed at a time
 - Dimensionality reduced by queries via projection/aggregation
- A cube consists of **cells**
 - A given combination of dimension values
 - A cell can be empty (no data for this combination)
 - A **sparse** cube has few non-empty cells
 - A **dense** cube has many non-empty cells
 - Cubes become sparser for many/large dimensions

Dimensions



- Dimensions are the core of multidimensional databases
 - Other types of databases do not support dimensions
- Dimensions are used for
 - **Selection** of data
 - **Grouping** of data at the right level of detail
- Dimensions consist of **dimension values**
 - Product dimension have values "milk", "cream", ...
 - Time dimension have values "1/1/2001", "2/1/2001", ...
- Dimension values may have an **ordering**
 - Used for comparing cube data across values
 - Example: "percent sales increase compared with last month"
 - Especially used for Time dimension

Dimensions

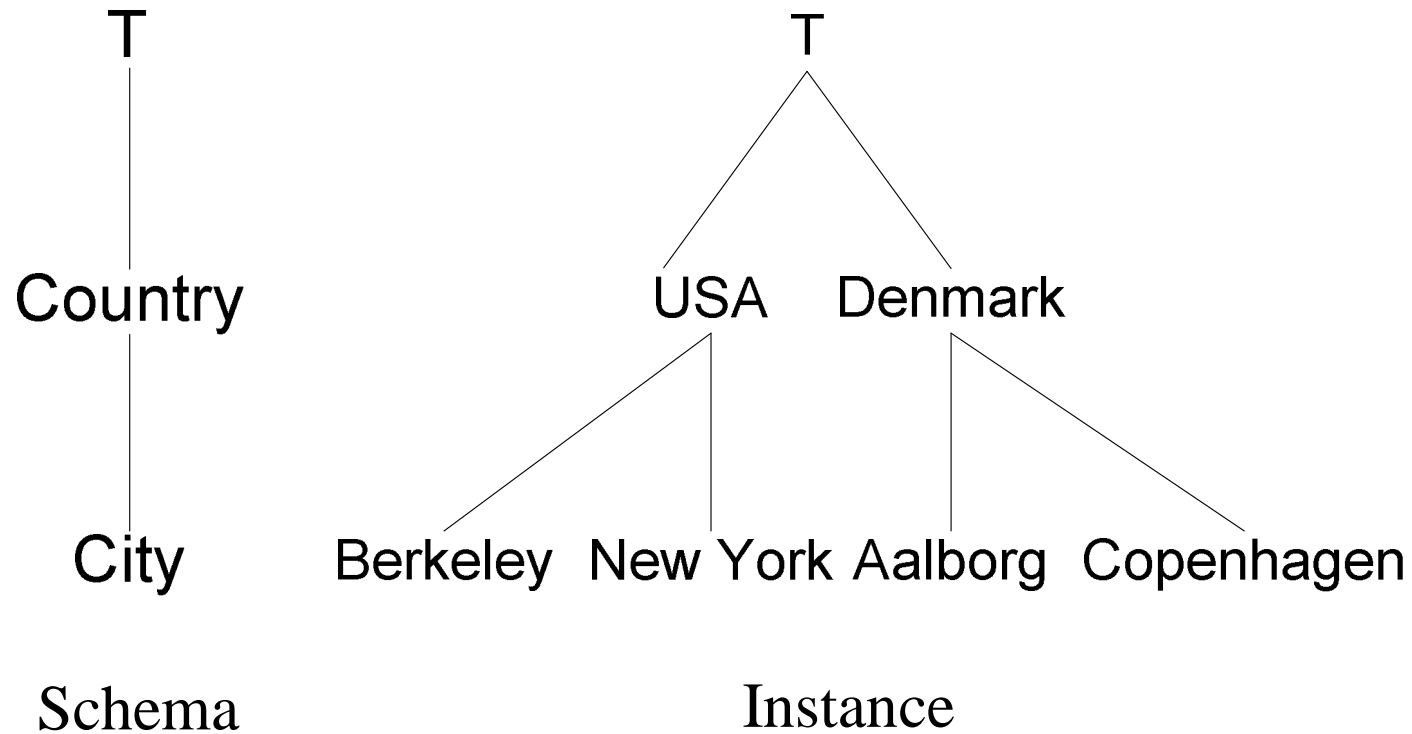


- Dimensions have **hierarchies** with **levels**
 - Typically 3-5 levels (of detail)
 - Dimension values are organized in a **tree structure**
 - **Product**: Product->Type->Category
 - **Store**: Store->Area->City->County
 - **Time**: Day->Month->Quarter->Year
 - Dimensions have a **bottom level** and a **top level** (ALL)
- Levels may have **attributes**
 - Simple, non-hierarchical information
 - Day has Workday as attribute
- Dimensions should contain much information
 - Time dimensions may contain holiday, season, events,...
 - Good dimensions have 50-100 or more attributes/levels

Dimension Example



Location



- Facts represent the **subject** of the desired analysis
 - The "important" in the business that should be analyzed
- A fact is most often identified via its dimension values
 - A fact is a non-empty cells
 - Some models give facts an explicit identity
- Generally a fact should
 - Be attached to **exactly one** dimension value in each dimension
 - Only be attached to dimension values in the bottom levels
 - Some models do not require this

Types Of Facts



- **Event** fact (transaction)
 - A fact for every **business event** (sale)
- **"Fact-less"** facts
 - A fact per event (customer contact)
 - **No** numerical measures
 - An event has happened for a given dimension value combination
- **Snapshot** fact
 - A fact for every dimension combination at given time intervals
 - Captures **current** status (inventory)
- **Cumulative snapshot** facts
 - A fact for every dimension combination at given time intervals
 - Captures **cumulative** status up to now (sales in year to date)
- Every type of facts answers **different** questions
 - Often both event facts and both kinds of snapshot facts exist

Granularity



- **Granularity** of facts is important
 - What does a single fact mean?
 - **Level of detail**
 - Given by combination of bottom levels
 - Example: "total sales per store per day per product"
- Important for number of facts
 - Scalability
- Often the granularity is a single business transaction
 - Example: sale
 - Sometimes the data is aggregated (**total** sales per store per day per product)
 - Might be necessary due to scalability
- Generally, transaction detail can be handled
 - Except perhaps huge clickstreams etc.

- Measures represent the fact property that the users want to **study and optimize**
 - Example: total sales price
- A measure has two components
 - **Numerical value**: (sales price)
 - **Aggregation formula** (SUM): used for aggregating/combining a number of measure values into one
 - Measure value determined by dimension value combination
 - Measure value is meaningful for all aggregation levels
- Most multidimensional models have measures
 - A few do not

Types Of Measures

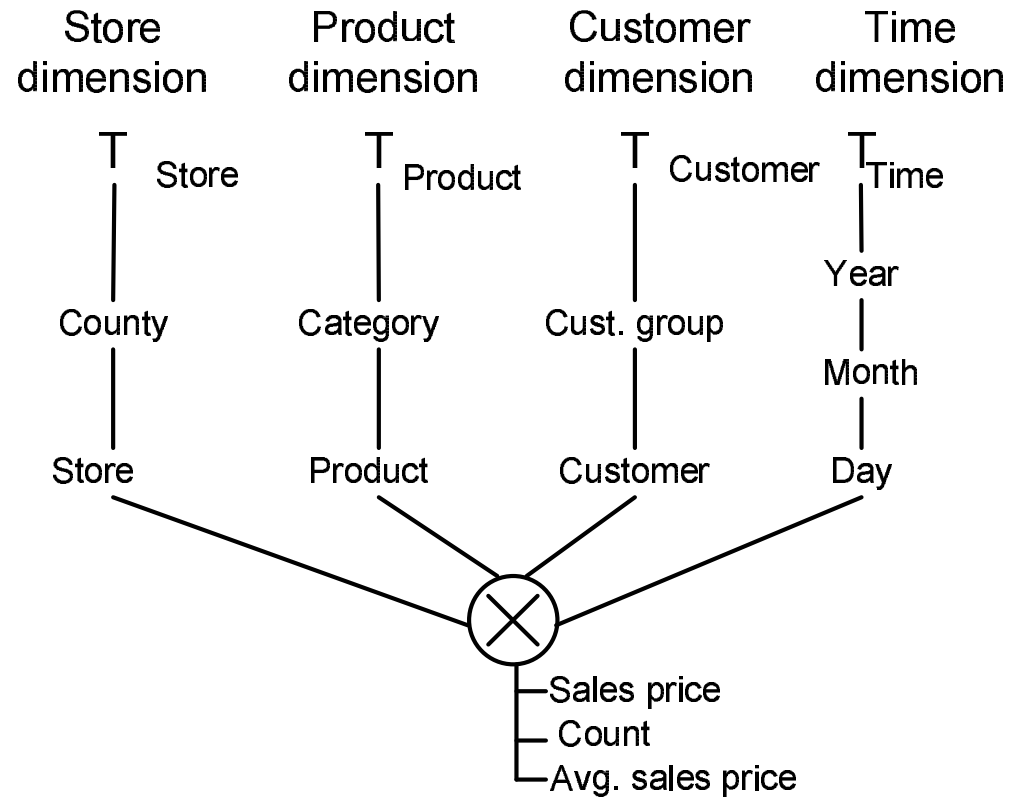


- Three types of measures
- Additive
 - Can be aggregated over **all** dimensions
 - Example: **sales price**
 - Often occur in event facts
- Semi-additive
 - **Cannot** be aggregated over **some** dimensions - typically time
 - Example: **inventory**
 - Often occur in snapshot facts
- Non-additive
 - **Cannot** be aggregated over **any** dimensions
 - Example: **average sales price**
 - Occur in all types of facts

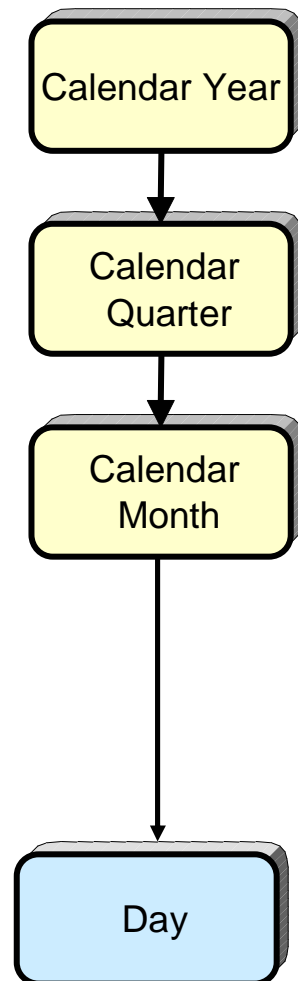
Documentation Of Schema



- No well-defined standard
- Our own notation
 - Seen to the right
 - T level corresponds to ALL
- Modeling and OLAP tools have their own notation



Kimball Dimension Notation



- The granulariteten is Day
- There is an implicit "top" value which means "all days" or "the whole time axis".
 - This is selected by not mentioning the dimension in a query

Relational OLAP (ROLAP)



- The cube is often implemented in an RDBMS
- Fact table stores facts
 - One column for each measure
 - One column for each dimension (foreign key to dimension table)
- Dimension table stores dimension
 - Integer key column (surrogate keys)
 - Don't use production keys in DW!
- Goal for dimensional modeling: "surround the facts with as much context (dimensions) as we can"
- **Granularity** of the fact table is important
 - What does one fact table row represent ?
 - Important for the size of the fact table
 - Often corresponding to a single business transaction (sale)
 - But it can be aggregated (sales per product per day per store)

Relational Design



- One completely de-normalized table
 - Bad: inflexibility, storage use, bad performance, slow update
- Star schemas
 - One fact table
 - De-normalized dimension tables
 - One column per level/attribute
- Snowflake schemas
 - Dimensions are normalized
 - One dimension table per level
 - Each dimension table has integer key, level name, and one column per attribute

Star Schema Example



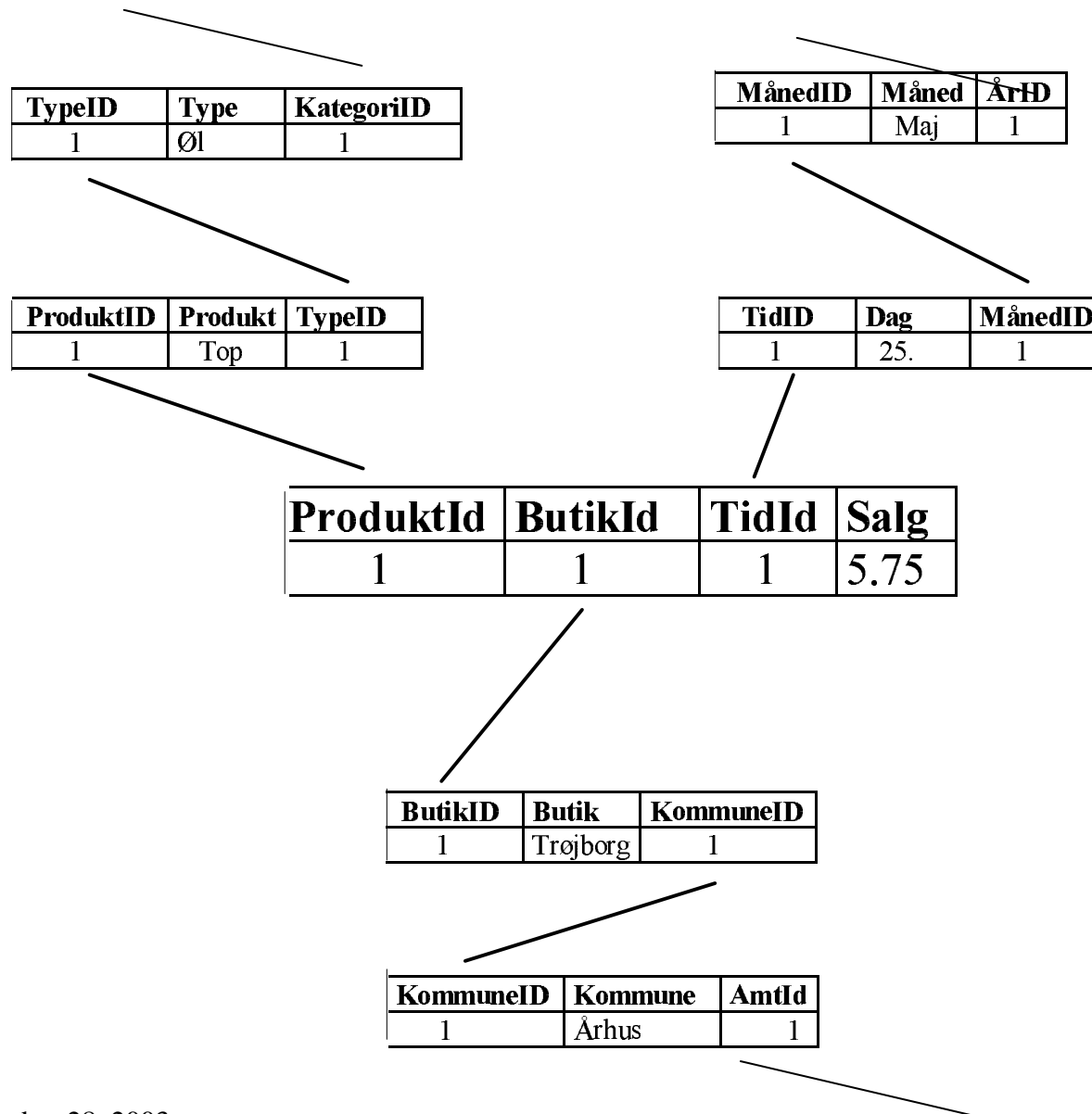
ProduktID	Produkt	Type	Kategori
1	Top	Øl	Drikkevarer

TidID	Dag	Måned	År
1	25.	Maj	1997

ProduktId	ButikId	TidId	Salg
1	1	1	5.75

ButikID	Butik	Kommune	Amt
1	Trøjborg	Århus	Århus

Snow-flake Schema Example



Relational OLAP Queries



- **Aggregating** data, e.g., with SUM
- **Starting level:** (Quarter, Product)
- **Roll Up:** less detail, Quarter->Year
- **Drill Down:** more detail, Quarter->Month
- **Slice/Dice:** selection, Year=1999
- **Drill Across:** “join” on common dimensions
- **Visualization** and **exceptions**
- Note: only **two** kinds of queries
 - **Navigation queries** examine one dimension
 - ◆ SELECT DISTINCT I FROM d [WHERE p]
 - **Aggregation queries** summarize fact data
 - ◆ SELECT d1.I1,d2.I2,SUM(f.m) FROM d1,d2,f
WHERE f.dk1=d1.dk1 AND f.dk2=d2.dk2 [AND p]
GROUP BY d1.I1,d2.I2

Star Schemas



- + Simple and easy overview -> ease-of-use
- + Relatively flexible
- + Fact table is normalized
- + Dimension tables often relatively small
- + “Recognized” by many RDBMSes -> good performance
- - Hierarchies are “hidden” in the columns
- - Dimension tables are de-normalized

Snow-flake Schemas



- + Hierarchies are made explicit/visible
- + Very flexible
- + Dimension tables use less space
- - Harder to use due to many joins
- - Worse performance

Redundancy In The DW



- Only very little redundancy in fact tables
 - Order head data copied to order line facts
 - The same fact data (generally) only stored in one fact table
- Redundancy is mostly in dimension tables
 - Star dimension tables have redundant entries for the higher levels
- Redundancy problems?
 - Inconsistent data – the central load process helps with this
 - Update time – the DW is optimized for querying, not updates
 - Space use: dimension tables typically take up less than 5% of DW
- So: **controlled** redundancy is good
 - Up to a certain limit

Limits – And Strengths



- Many-to-one relationship from fact to dimension
- Many-to-one relationships from lower to higher levels in the hierarchies
- Therefore, it is impossible to "count wrong"
- Hierarchies have a fixed height
- Hierarchies don't change?

Cube Design



- New dimensions should be added **gracefully**
 - Old queries will still give the same result
 - Example: location dimension can be added for **old**+new facts
- The design is never "done"
 - The dimensional modeler always looks for new information to include in dimensions and facts

Stringing Stars Together



- Data marts are built independently by departments
 - This good (keeps project size down, focus,...)
 - Problems with "stovepipes" (reuse across data marts impossible)
 - How should this be resolved?
- **Conformed** dimensions and facts/measures
- Conformed dimensions
 - The same structure **and content** across data marts
 - Dimensions are **copied** among data marts (not a space problem)
- Conformed facts
 - The same **definition** across data marts (sales without sales tax)
 - Facts are **not** copied across data marts (facts > 90% of data)
- "Data Warehouse Bus" architecture (Kimball 98)

Stringing Stars Together



- Dimension content managed by **one dimension authority**
- Query tools connect to each data mart
 - Separate SQL requests to each data mart
 - Results are merged (outer join) by the query tool, not the DBMS
- It is **hard** to make conformed dimensions and facts
 - Organizational and political challenge, not technical
 - Get everyone together **and**
 - Have a high-level manager affirm the intension of conforming

Break



- Dimensional modeling
 - Facts
 - Measures
 - Dimensions
 - Levels
- Implementation in RDBMS
 - Star schemas
 - Snow-flake schemas
 - SQL queries on stars and snowflakes
 - Redundancy
 - Conforming facts and dimensions

Talk Overview



- Data warehouse basics
 - Definition
 - Applications
- Multidimensional modeling
 - Dimensional concepts
 - Implementation in RDBMS
- **Case study**
 - **The grocery store**
- Exercise:
 - Build a data warehouse for a real warehouse !

The Grocery Store



- Stock Keeping Units (SKUs)
- Universal Product Codes (UPCs)
- Point Of Sale (POS) system
- Stores
- Promotions

DW Design Steps



- Choose the **business process(es)** to model
 - Sales
- Choose the **grain** of the business process
 - SKU by Store by Promotion by Day
 - Low granularity is needed
 - Are individual transactions necessary/feasible ?
- Choose the **dimensions**
 - Time, Store, Promotion, Product
- Choose the **measures**
 - Dollar_sales, unit_sales, dollar_cost, customer_count
- Resisting normalization and preserving browsing
 - Flat dimension tables makes browsing easy and fast

The Grocery Store Dimensions



- The Time dimension
 - Explicit time dimension is needed (events, holidays,...)
- The Product dimension
 - Six-level hierarchy allows drill-down/roll-up
 - **Many** descriptive attributes (often more than 50)
- The Store dimension
 - Many descriptive attributes
 - The Time dimension is an **outrigger** table (First opened,...)
- The Promotion dimension
 - Example of a **causal** dimension
 - Used to see if promotions work/are profitable
 - Ads, price reductions, end-of-aisle displays, coupons
 - ◆ Highly correlated (only 5000 combinations)
 - ◆ Separate dimensions ? (size&efficiency versus simplicity&understanding)

The Grocery Store Measures



- Dollar_sales
- Unit_sales
- Dollar_cost
- All **additive** across all dimensions
- Gross profit
 - Computed from sales and cost
 - Additive
- Gross margin
 - Computed from gross profit and sales
 - **Non-additive** across all dimensions
- Customer_count
 - Additive across time, promotion, and store
 - **Non-additive** across product
 - **Semi-additive**

Database Sizing



- Time dimension: 2 years = 730 days
- Store dimension: 300 stores reporting each day
- Product dimension: 30,000 products, only 3000 sell per day
- Promotion dimension: 5000 combinations, but a product only appears in one combination per day
- Number of fact records: $730 * 300 * 3000 * 1 = 657,000,000$
- Number of fields: 4 key + 4 fact = 8 fields
- Total DB size: $657,000,000 * 8 \text{ fields} * 4 \text{ bytes} = 21 \text{ GB}$
- **Small** database by today's standards?
- Transaction level detail is feasible today

Typical Fact Tables (Again)



- Event/transaction table
 - One record for every business event (sale)
- Snapshot table
 - One record for every dimension combination at given time intervals
 - Records **current** status (inventory)
 - Often, both event and snapshot tables are needed
- Cumulative snapshot table
 - One record for every dimension combination at given time intervals
 - Records **cumulative** status up till now (sales in year to date)
- Fact-less fact table
 - One record per event (customer contact)
 - **No** numeric measures
 - Used to capture that an event has happened for a particular dimension combination

Break



- The grocery store
- The grocery store dimensions
- The grocery store measures
- Database sizing
- Typical fact tables

References



- References
 - Meta Group. *1999 DW Marketing Trends* <metagroup.com>
 - Palo Alto Management Group. 1999 BI and DW Program Competitive Analysis Report, <pamg.com>
 - Ralph Kimball. *The Data Warehouse Toolkit*, Wiley, 1996
 - Ralph Kimball et al. *The Data Warehouse Lifecycle Toolkit*, Wiley, 1998
 - Ralph Kimball and R. Merz. *The Data Webhouse Toolkit*, Wiley, 2000.
 - Ralph Kimball. Data Webhouse Column, <intelligententerprise.com>
 - Erik Thomsen. *OLAP Solutions*, Wiley, 1997.
 - Erik Thomsen et al. *Microsoft OLAP Solutions*, Wiley, 1999.
 - DBMiner Technology <dbminer.com>
 - The OLAP Council <olapcouncil.org>
 - The OLAP Report <olapreport.com>
 - The Data Warehousing Information Center <dwinfocenter.org>
 - DSS Lab <dsslabor.com>
 - NDB www.cs.auc.dk/NDB

Talk Overview



- Data warehouse basics
 - Definition
 - Applications
- Multidimensional modeling
 - Dimensional concepts
 - Implementation in RDBMS
- Case study
 - The grocery store
- **Exercise:**
 - **Build a data warehouse for a real warehouse !**