Multidimensional Databases and Data Warehousing

Christian Thomsen, Aalborg University

Slides adapted from earlier versions made by
Christian S. Jensen, Torben Bach Pedersen, and Man Lung Yiu

Agenda

- What is Business Intelligence?
- Data analysis problems
- Data Warehouse (DW) introduction
- Fundamental multidimensional modeling
- Relational representations
- Querying

Outcomes

- By the end of the lecture (and the exercise session), you'll be able to
  - See when to use multidimensional databases
  - Do basic multidimensional modeling
  - Create a relational representation of a multidimensional database
  - Use basic analysis and querying on a multidimensional database

- You will need that for your work on the F-klub case

What is Business Intelligence (BI)?

- From *Encyclopedia of Database Systems*:

  “[BI] refers to a **set of tools and techniques** that enable a company to **transform its business data** into **timely and accurate information** for the decisional process, to be made available to the right persons **in the most suitable form**.”

*CT: My **emphasis***

What is Business Intelligence (BI)?

- BI is different from Artificial Intelligence (AI)
  - AI systems **make** decisions **for** the users
  - BI systems **help** the users make the **right** decisions, based on available data

- Combination of technologies
  - Data Warehousing (DW)
  - On-Line Analytical Processing (OLAP)
  - Data Mining (DM)
  - ……

Case Study of an Enterprise

- Example of a chain of hi-fi stores or car dealers or …
  - Each store maintains its own customer records and sales records
  - The same customer may be viewed as different customers for different stores
  - Imprecise or missing data in the addresses of some customers
  - Purchase records are maintained in the operational system for a limited time (e.g., 6 months)
  - The same "product" may have different prices, or different discounts in different stores

- Can you see the problems of using such data for business analysis?

Data Analysis Problems

- The same data found in many different systems
  - Example: customer data across different stores and departments
  - The same concept is defined differently

- Heterogeneous sources
  - Relational DBMS, On-Line Transaction Processing (OLTP)
  - Data in files (e.g., Excel or text files)
  - Legacy systems
  - …

Data Analysis Problems (cont')

- Data is suited for operational systems
  - Accounting, billing, etc.
  - Does not support analysis across business functions
- Data quality is bad
  - Missing data, imprecise data, different use of systems
- Data is "volatile"
  - Data deleted in operational systems (6 months)
  - Data changes over time – no historical information

Data Analysis Problems (cont')

- Kimball & Ross point out typical issues:
  - "We have mountains of data, but we can't access it"
  - "We need to slice and dice the data in every which way"
  - "Make it easy to get the data directly"
  - "Show me what is important"
  - "Two people present the same business metrics, but with different numbers"

Data Warehousing

- Solution: new analysis environment (DW) where the data is
  - Subject oriented (versus function oriented)
  - Integrated (logically and physically)
  - Time variant (data can always be related to time)
  - Stable (data not deleted, several versions)
  - Supporting management decisions (different organization)

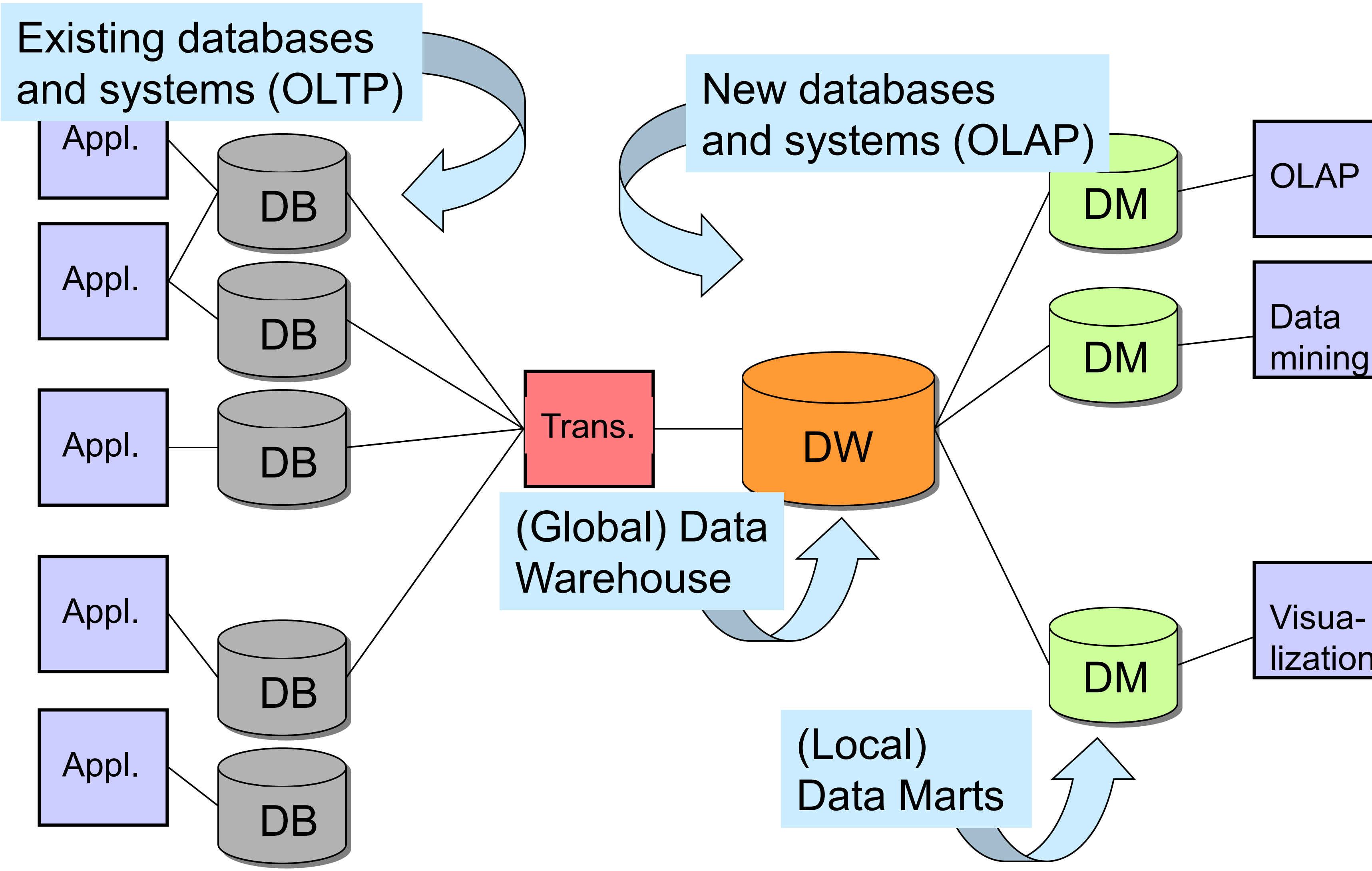
- Data from the operational systems is
  - Extracted
  - Cleansed
  - Transformed
  - Aggregated (?)
  - Loaded into the DW

- A good DW is a **prerequisite** for successful BI

DW: Purpose and Definition

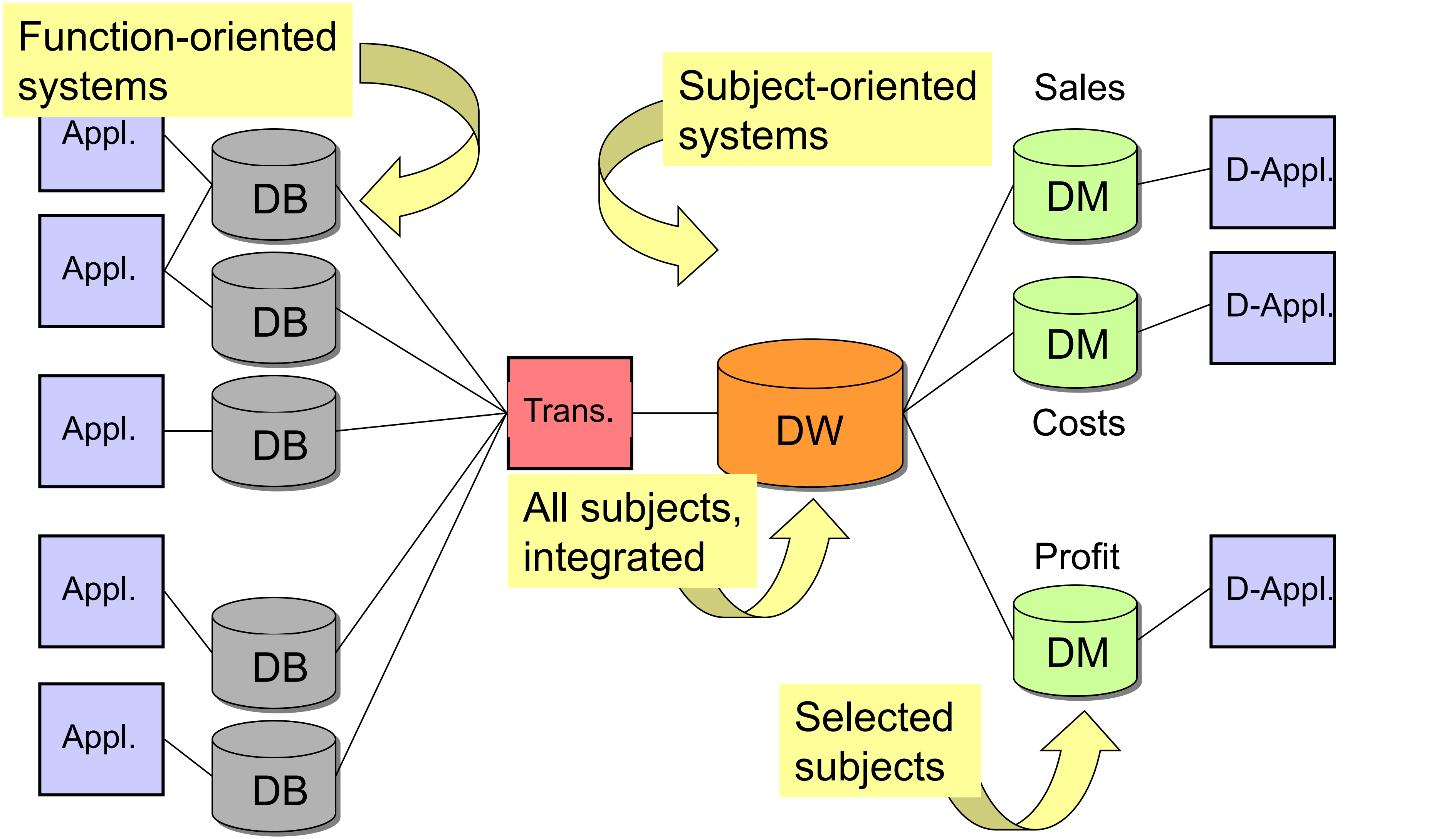
- A DW is a **store of information** organized in a unified data model
- Data collected from a number of different sources
  - Finance, billing, website logs, personnel, …
- Purpose of a data warehouse (DW):
  support **decision making**
- Easy to perform advanced analysis
  - Ad-hoc analysis and reports
    - We will cover this soon ……
  - Data mining: discovery of hidden patterns and trends

DW Architecture – Data as Materialized Views



Analogy: (data) producers ↔ warehouse ↔ (data) consumers

Function vs. Subject Orientation



Hard/Infeasible Queries for OLTP

- Why not use the existing databases (OLTP) for business analysis?
- Business analysis queries
  - In the **past five years**, which 10 products are most profitable?
  - Which **public holiday** has the largest sales?
  - Which **week** has the largest sales?
  - Does the sales of **dairy products** increase over time?
- Difficult for the analyst to express these queries
- There is a need for multidimensional modeling …

Multidimensional Modeling

- Example: sales from supermarkets
- Facts and measures
  - Each sales record is a *fact*, and its sales value is a *measure*
- Dimensions
  - Group correlated attributes into the same dimension ➔ easier for analysis tasks
  - Each sales record is associated with its values of *Product*, *Store*, *Time*

Product	Type	Category	Store	City	Region	Date	Sales
Top	Beer	Beverage	Vejgård	Aalborg	Nord	25 May, 2015	7.75

Product　　　　Store　　　　Time

Multidimensional Modeling

- How do we model the *Time* dimension?
  - *Hierarchy* with multiple levels
  - Attributes, e.g., holiday, event

T
|
Year
|
Month
|
Day

tid	day	day #	month #	year	week day	work day	…
1	January 1st 2009	1	1	2009	Thurs day	No	…
2	January 2nd 2009	2	1	2009	Fri day	Yes	…
…	…	…	…	…	…	…	…

- Advantage of this model?
  - Easy for querying (more about this later)
- Disadvantage?
  - Data redundancy (but controlled redundancy is acceptable)

OLTP vs. OLAP

	OLTP	**OLAP**
Target	operational needs	business analysis
Data	small, operational data	large, historical data
Model	normalized	denormalized/ multidimensional
Query language	SQL	not unified – but MDX is used by many
Queries	small	large
Updates	frequent and small	infrequent and batch
Optimized for	update operations	query operations

Quiz - Pick the correct answer(s) for each question

Q1.1 Data in a data warehouse

[ ] A) can come from different source systems
[ ] B) should generally be deleted after some time (e.g., 12 months)
[ ] C) should be represented exactly as it is in the originating source
[ ] D) should be related to time

Q1.2 The purpose of a data warehouse is to enable

[ ] A) fast updates
[ ] B) less storage use
[ ] C) easy analysis
[ ] D) efficient backup

Q1.3 Data redundancy in a data warehouse is

[ ] A) considered harmful
[ ] B) OK to some degree
[ ] C) not a problem at all

Q1.4 A data warehouse is

[ ] A) a complete replacement for an OLTP system
[ ] B) not needed if you already have an OLTP system
[ ] C) something that co-exists with the OLTP system

ER Model vs. Multidimensional Model

- ER model: a data model for **general** purposes

- All types of data are “equal”, and it is difficult to identify the data that is important for business analysis
  - No difference between what **is** important and what just **describes** the important
  - Normalized databases **spread** information
  - When analyzing data, the information must be **integrated** again

- Hard to overview a **large** ER diagram (e.g., over 100 entities/relations for an enterprise)

ER Model vs. Multidimensional Model

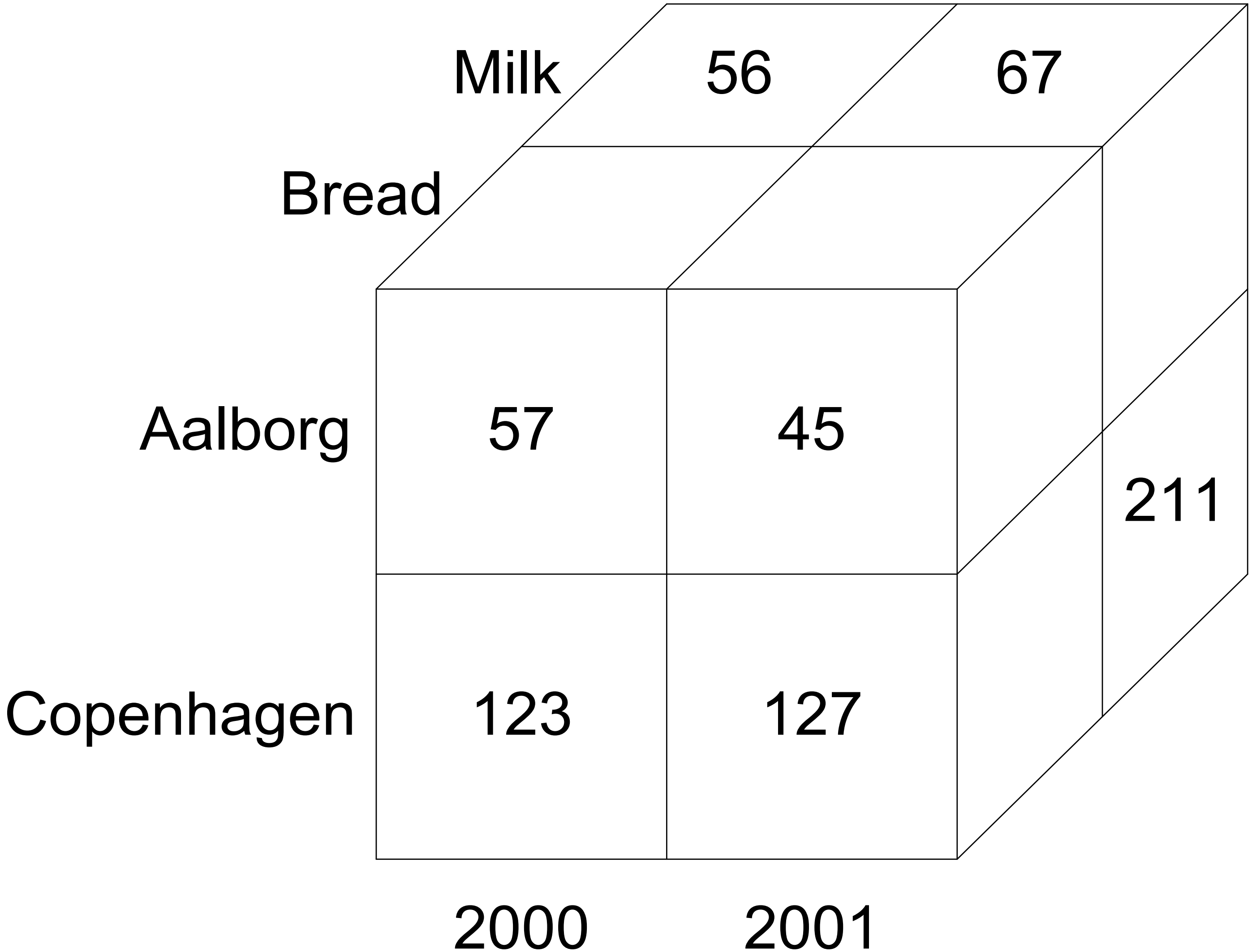
- The multidimensional model's only purpose is **data analysis**
  - It is not suitable for OLTP systems
- More **built in** "meaning"
  - What **is** important
  - What **describes** the important
  - What we want to **optimize**
  - Easy for query operations
- Recognized by OLAP/BI tools, e.g., TARGIT
  - Tools offer powerful query facilities based on MD design

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**

- Goal for dimensional modeling:
  - Surround facts with as much context (dimensions) as possible
  - Hint: redundancy may be okay (in well-chosen places)
  - You do not necessarily have to model *all* relationships in the data

Cube Example



Cubes

- A “cube” may have **many** dimensions!
  - It can have 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 2-4 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 (i.e., there is no data for this combination)
  - A **sparse** cube has many empty cells
  - A **dense** cube has many filled cells
  - Cubes become sparser for many/large dimensions

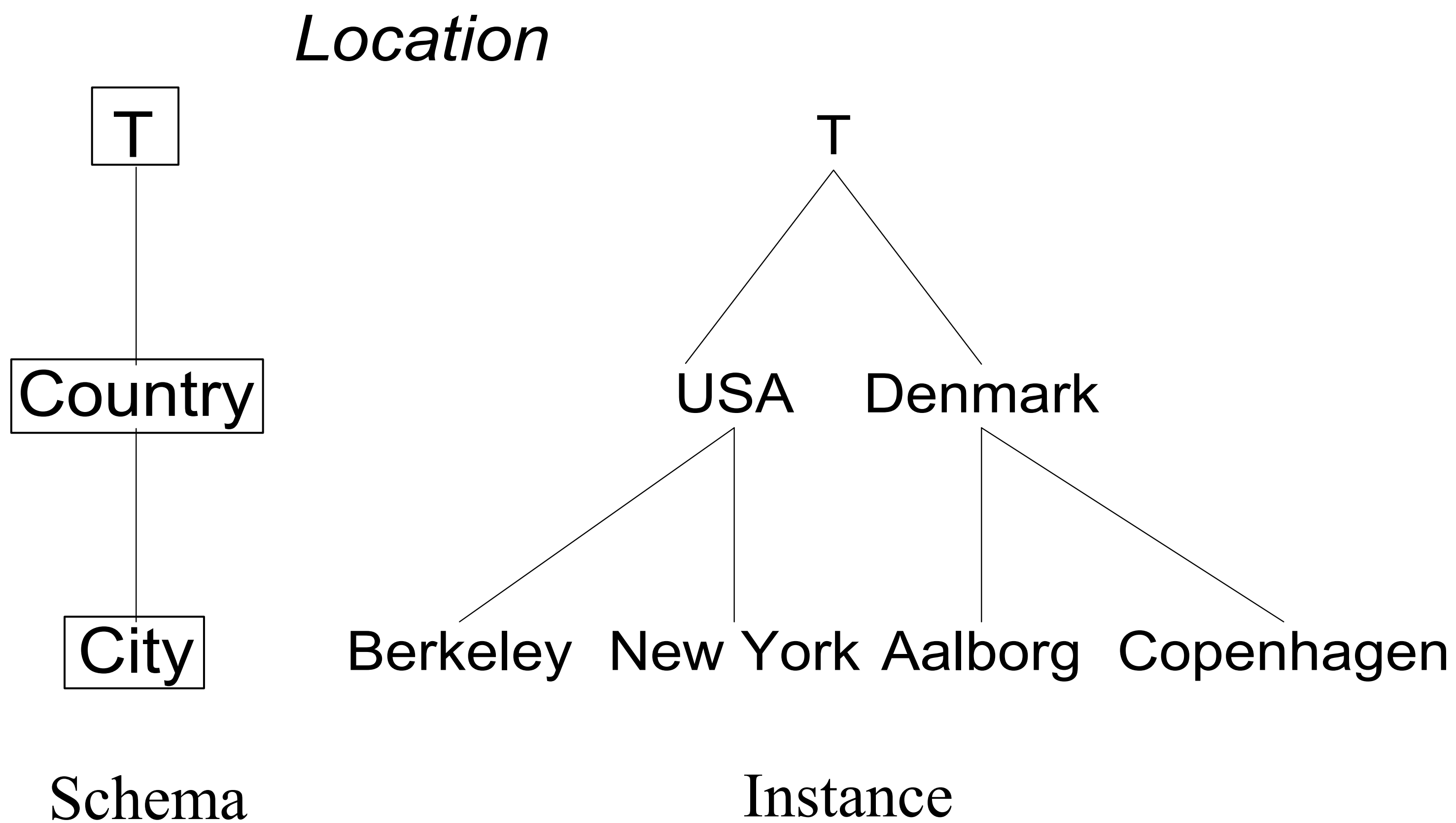
Dimensions

- Dimensions are the core of multidimensional databases
- Dimensions are used for
  - **Selection** of data
  - **Grouping** of data at the right level of detail
- Dimensions consist of **dimension values**
  - The Product dimension has the values ”milk”, ”cream”, …
  - The Time dimension has the values ”1/1/2020”, ”2/1/2020”,…
- Dimension values *can* have an **ordering**
  - Used for comparing cube data across values
  - Example: ”percent sales increase compared with last month”
  - Especially used for the Time dimension

Dimensions

- Dimensions have **hierarchies** with **levels**
  - Typically 3-5 levels
  - **Product**: Product→Type→Category→T
  - **Store**: Store→Area→City→County→T
  - **Time**: Day→Month→Quarter→Year→T
  - Dimensions have a **bottom level** and a **top level** ("T" or "ALL")
  - Dimension values are organized in a **tree structure**
- 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



Dimensions

- Dimensions are independent logical clumps of data
- We group correlated attributes into the same dimension

- Be careful not to group things that don't belong together
  - ProductAndShopDimension *vs.*
  - ProductDimension + ShopDimension

- What about dates and time of day?
  - They are often used independently of each other
  - 24 * 60 * 60 = 86,400 seconds in a day
  - 365 days in a year
  - → 31,536,000 seconds in a year
  - Go for two different dimensions

Facts

- Facts represent the **subject** of the desired analysis
  - What is important to the business
- A fact is identified via its dimension values
  - A fact is a non-empty cell

- 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, but we do!)

Types of Facts

- **Event** fact (a.k.a. transaction fact)
  - A fact for every **business event** (sale)
- **Measureless** fact (a.k.a. factless fact ☹)
  - A fact per event (e.g., customer contact)
  - **No** numerical measures
  - An event has happened for a given dimension value combination
- **State** fact (a.k.a. snapshot fact)
  - A fact for every dimension combination at given time intervals
  - Captures **current** status (e.g., inventory)

- Every type of facts answers **different** questions
  - Often both event facts and 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”
- Affects the number of facts

- 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

Measures

- Measures represent the fact property that the users want to **study and optimize**
  - Example: total sales price
- A measure has two components
  - **Numerical value** (e.g., sales price)
  - **Aggregation formula** (e.g., 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 (including the top level T)

Types of Measures

- **Additive**
  - Can be aggregated over **all** dimensions
  - Example: **sales price**
  - Do often occur in event facts
- **Semi-additive**
  - Can be aggregated over **some** dimensions, but **not all** dimensions (typically not time)
  - Example: **inventory**
  - Do often occur in snapshot facts
- **Non-additive**
  - **Cannot** be aggregated over **any** dimensions
  - Example: **average sales price**
  - Occur in all types of facts

Quiz

Q1.5 In the multidimensional model, data is divided into
[ ] A) facts and records
[ ] B) facts and cubes
[ ] C) facts and dimensions
[ ] D) cubes and dimensions
[ ] E) hierarchies and dimensions

Q1.6 Facts
[ ] A) describe dimension values
[ ] B) always exist for every possible combination of dimension values
[ ] C) represent the subject of the analyses
[ ] D) can have measures
[ ] E) are described by dimension values

Q1.7 Measures
[ ] A) have one component: a numerical value
[ ] B) have two components: a numerical value and an aggregation formula
[ ] C) have three components: a numerical value, an aggregation formula, and a dimension value

Relational Implementation

- 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?
- Many-to-one relationships from facts to dimension values
- Many-to-one relationships from lower to higher levels in the hierarchies

Relational Design

Product	Type	Category	Store	City	County	Date	Sales
Top	Beer	Beverage	Trøjborg	Århus	Århus	25 May 2009	5.75

Product Store Time

- Naïve solution: One completely de-normalized table
  - This is bad due to inflexibility, storage use, bad performance, slow updates
- Instead, we use *star schemas* or *snowflake schemas* with fact tables and dimension tables

Relational Implementation

- The **fact table** stores facts
  - One row for each fact
  - One column for each measure
  - One column for each dimension (foreign key to dimension table)
  - The dimension keys make up a composite primary key

- A **dimension table** stores dimension data

- What are the disadvantages of using production codes as the key?
  - E.g., product dimension, production code: AABC1234
  - E.g., customer dimension, CPR number: 020208-1357
- Use a surrogate key ("meaningless" integer key), which only allows the linking between its dimension table and the fact table

Star Schema Example

- **Star schema**
  - One fact table (for each business process)
  - One de-normalized dimension table for each dimension with one column for each level and attribute (except T)

ProductID	Product	Type	Category
1	Top	Beer	Beverage

TimeID	Day	Month	Year
1	25.	Maj	1997

ProductId	StoreId	TimeId	Sale
1	1	1	5.75

StoreID	Store	City	County
1	Trøjborg	Århus	Århus

Snowflake Schema Example

TypeID	Type	CategoryID
1	Beer	1

ProductID	Product	TypeID
1	Top	1

MonthID	Month	YearID
1	May	1

TimeID	Day	MonthID
1	25.	1

- **Snowflake schema**
  - Dimensions are normalized
  - *One* dimension table *per* level
  - Each dimension table has an integer key, a level name, one column per attribute, and a foreign key to the next level

ProductId	StoreId	TimeId	Sale
1	1	1	5.75

StoreID	Store	CityID
1	Trøjborg	1

CityID	City	CountyId
1	Århus	1

Star vs. Snowflake

- Star Schemas
  + Simple and easy overview → ease-of-use
  + Dimension tables often relatively small
  + “Recognized” by many RDBMSes → good performance
  - Hierarchies are ”hidden” in the columns
  - Dimension tables are de-normalized

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

Quiz

Q1.8 In a star schema, a dimension is represented by
[ ] A) several normalized dimension tables
[ ] B) a single denormalized dimension table
[ ] C) one or more fact tables

Q1.9 In a snowflake schema, a dimension is represented by
[ ] A) several normalized dimension tables
[ ] B) a single denormalized dimension table
[ ] C) one or more fact tables

Q1.10 If you change a typical star schema into a snowflake schema, you will end up with
[ ] A) more tables
[ ] B) fewer tabes
[ ] C) the same amount of tables

Q1.11 If you change a star schema into a snowflake schema, the fact table will
[ ] A) get more columns
[ ] B) get more rows
[ ] C) not change

Q1.12 Consider a collection of star schemas. When we talk about "a fact", we refer to
[ ] A) a row in a fact table
[ ] B) one of the fact tables
[ ] C) a row in a fact table OR one of the fact tables, depending on the context

Redundancy in the DW

- Only very little or no redundancy in fact tables
- 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 acceptable

Case Study: Grocery Store

- Products sold from a Point Of Sale (POS) system in Stores with Promotions

- Task: Analyze how promotions affect sales

DW Design Steps

- Choose the **business process(es)** to model
  - Sales
- Choose the **granularity** of the business process
  - Sales by Product 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 make browsing easy and fast

The Grocery Store Dimensions

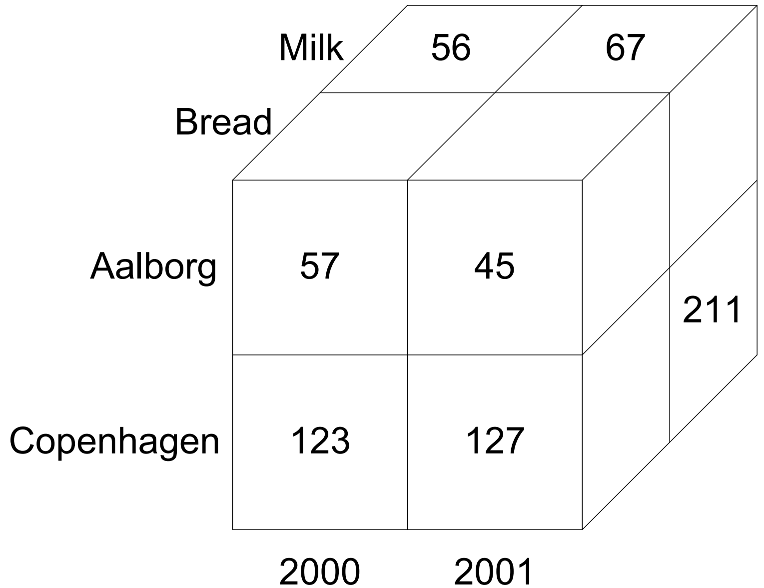
- Time dimension
  - Explicit time dimension is needed (events, holidays,..)
- Product dimension
  - Many-level hierarchy allows drill-down/roll-up
  - **Many** descriptive attributes (often more than 50)
- Store dimension
  - Many descriptive attributes
- Promotion dimension
  - Used to see if promotions work/are profitable
  - Ads, price reductions, end-of-aisle displays, coupons

The Grocery Store Measures

- All **additive** across all dimensions
  - Dollar_sales
  - Unit_sales
  - Dollar_cost
- Gross profit (derived)
  - Computed from sales and cost: sales – cost
  - Additive
- Gross margin (derived)
  - Computed from gross profit and sales: (sales – cost)/cost
  - **Non-additive** across all dimensions
- Customer_count
  - Additive across time, promotion, and store
  - **Non-additive** across product. Why?
  - **Semi-additive**

OLAP Queries

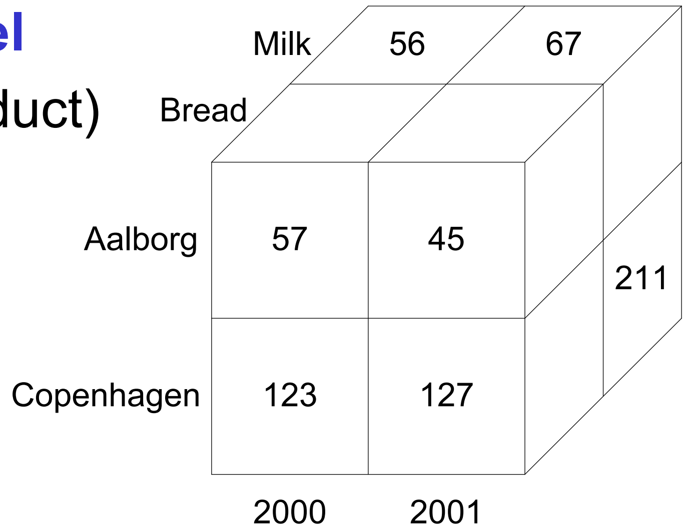
- Two kinds of queries
  - **Navigation queries** examine one dimension
    - SELECT DISTINCT l FROM d [WHERE p]
  - **Aggregation queries** summarize fact data
    - SELECT d1.l1, d2.l2, SUM(f.m) FROM d1, d2, f
      WHERE f.dk1 = d1.dk1 AND f.dk2 = d2.dk2 [AND p]
      GROUP BY d1.l1,d2.l2
- Fast, interactive analysis of large amounts of data



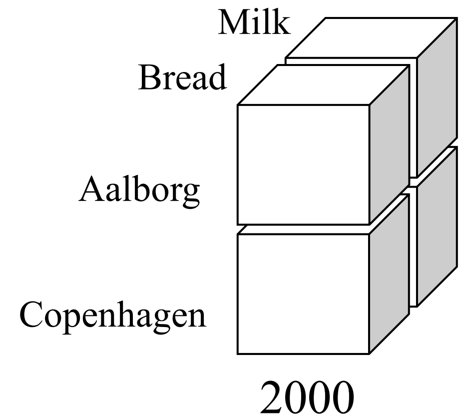
OLAP Queries

Starting level

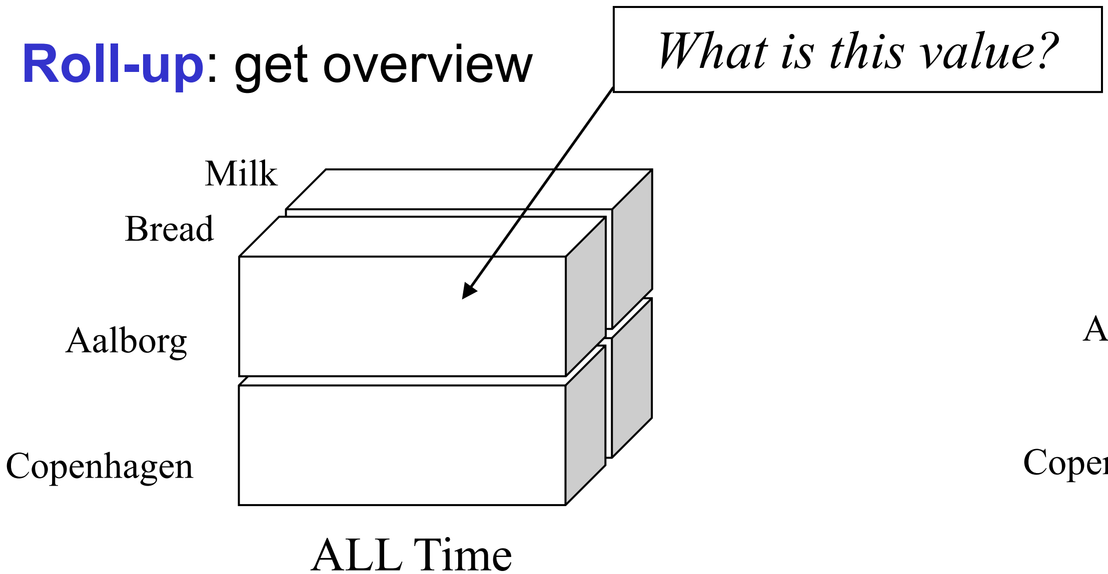
(City, Year, Product)



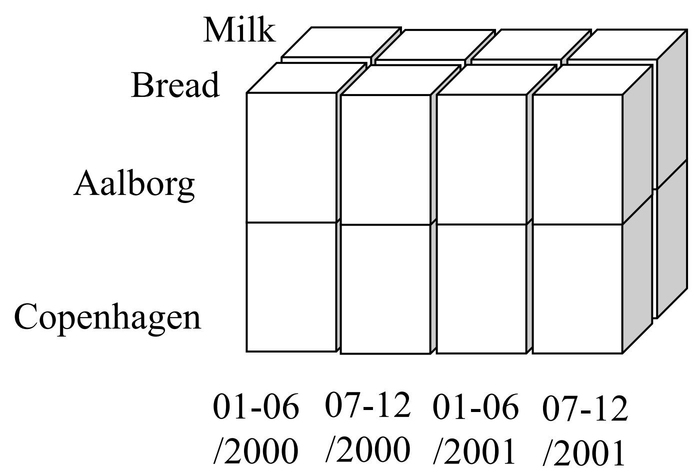
Slice/Dice:



Roll-up: get overview



Drill-down: more detail



OLAP Cube in MS Analysis Services Project

	Prod Group / Name						
	cacao	flask	kaffe	milk	others	vand	Grand Total
Year / Month / Day	Sales	Sales	Sales	Sales	Sales	Sales	Sales
1996	369	471		229		813	1882
1997	2161.75	3985		1727	144	15576	23593.75
1998	16082	20591		12887.25	6908	80492	136960.25
1999	17325	20626	2535	13063.25	7609.5	90644	151802.75
2000	21095	17395	5940	10631.5	21132.5	81444	157638
2001	16900.75	29712.5	0	9861.25	23260.25	84286	164020.75
2002	30086.5	34731	0	15506.5	41619.5	74847	196790.5
2003	28740	28596	0	14213.5	45046	63580	180175.5
2004	24126.75	28292	0	9592	82226	54526.5	198763.25
2005	22695.5	20449	0	7803.25	75835	52044	178826.75
2006	25196	19958	0	6910.5	102746	47456	202266.5
2007	876	641	0	155.75	2094.5	1387.5	5154.75
Grand Total	205654.25	225447.5	8475	102580.75	408621.25	647096	1597874.75

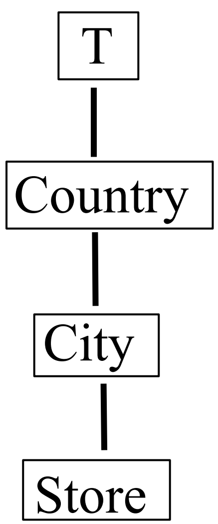
drill down

	Prod Group / Name								
	cacao			flask	kaffe	milk	others	vand	Grand Total
	½L Matilde cacao	Cocio	Total						
Year / Month / Day	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales
1996	174	195	369	471		229		813	1882
1997	1501.75	660	2161.75	3985		1727	144	15576	23593.75
1998	13767	2315	16082	20591		12887.25	6908	80492	136960.25
1999	13050	4275	17325	20626	2535	13063.25	7609.5	90644	151802.75
2000	17430	3665	21095	17395	5940	10631.5	21132.5	81444	157638
2001	12403.5	4497.25	16900.75	29712.5	0	9861.25	23260.25	84286	164020.75
2002	25425.75	4660.75	30086.5	34731	0	15506.5	41619.5	74847	196790.5
2003	25524.25	3215.75	28740	28596	0	14213.5	45046	63580	180175.5
2004	20286	3840.75	24126.75	28292	0	9592	82226	54526.5	198763.25
2005	18152.75	4542.75	22695.5	20449	0	7803.25	75835	52044	178826.75
2006	22968.5	2227.5	25196	19958	0	6910.5	102746	47456	202266.5
2007	876		876	641	0	155.75	2094.5	1387.5	5154.75
Grand Total	171559.5	34094.75	205654.25	225447.5	8475	102580.75	408621.25	647096	1597874.75

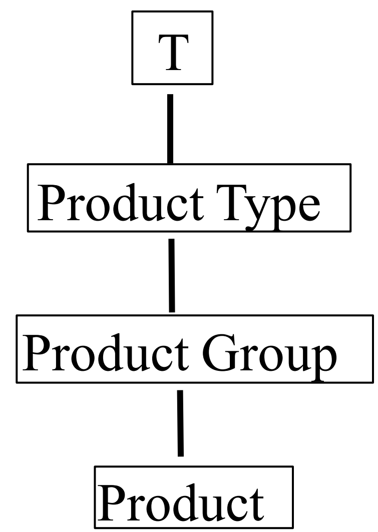
“Drill-down” vs. “Drill-out”

- We “drill-down” when we go downwards from one (non-T) level in a hierarchy to another level in the same hierarchy
- We “drill-out” when we include a level from another hierarchy in the analysis

- Consider the following hierarchies:



Store Dimension



Product Dimension

Drill-down

Country	Sales
Denmark	4200
Sweden	5500

Drill-down from Country level to City level

City	Sales
Aalborg	2000
Copenhagen	2200
Lund	2500
Stockholm	3000

Drill-out

Country	Sales
Denmark	4200
Sweden	5500

Drill-out to include the Product Type level

Country	Product Type	Sales
Denmark	Food	3000
Denmark	Non-food	1200
Sweden	Food	4000
Sweden	Non-food	1500

Drill-across

- To drill-across means to combine two cubes by means of one or more shared dimensions
  - In relational terms this corresponds to a join
- The resulting cube holds measures from both cubes
  - You can think of non-shared dimensions as rolled up to their top level

- Consider, for example, a book retailer with two cubes
  - ShopSales with dimensions Book, Date, and Shop
  - InternetSales with dimensions Book, Date, and Customer
- To find the total sales, the book retailer drills-across and considers the (calculated) measure shop_sales + internet_sales

Summary of Multidimensional Modeling

- Cubes, Dimensions, Facts, Measures
- Relational Implementation
  - Star schema, Snowflake schema
- OLAP Queries

- Next lecture: advanced issues on multidimensional modeling

Exercises

- JPT Section 2.10