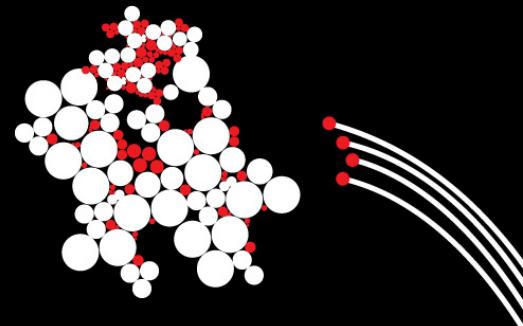


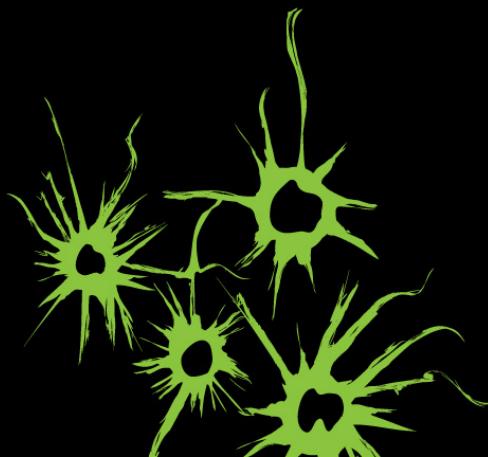
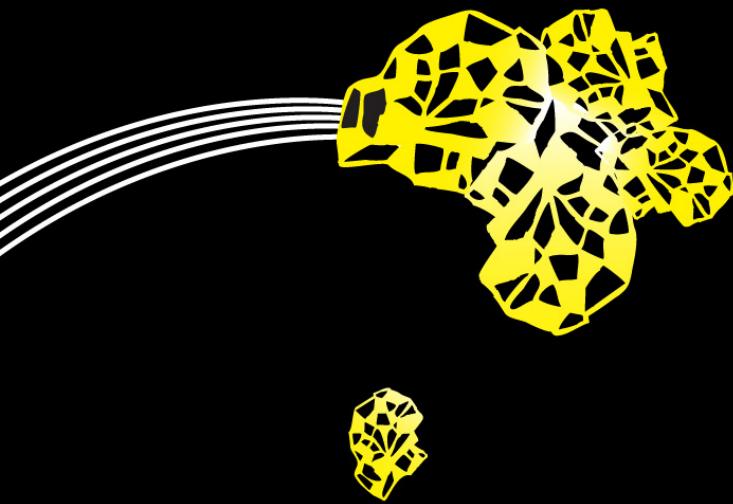
UNIVERSITY OF TWENTE.



# Data Science Topic DPV

# Data Preparation and Visualization

MAURICE VAN KEULEN, KARIN GROOTHUIS-OUDSHOORN  
(CHINTAN AMRIT)



# STUDY MATERIAL

---

## Multidimensional modeling

- **Bookchapter:**

C.S. Jensen, T.B. Pedersen, C. Thomsen, "Fundamental Concepts".

**Chapter 2** in "Multidimensional Databases and Data Warehousing". 2010.

**Access:** through UT library

<https://ut.on.worldcat.org/oclc/664723898>

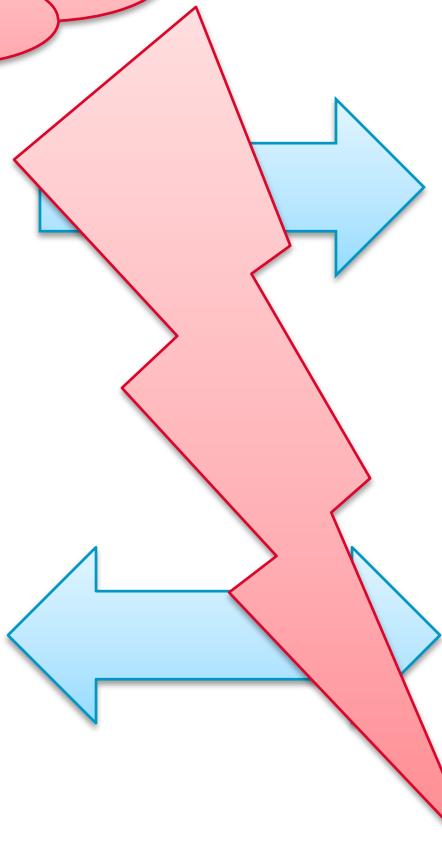
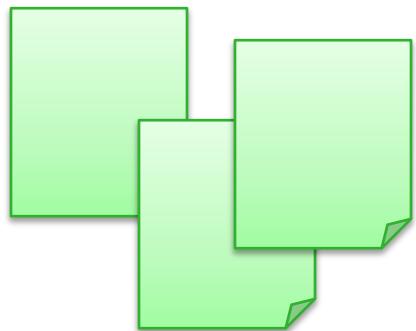
- **Note:** You can do **without this book** and rely on slides and practice only; provided as reference because it nicely and slowly explains all the basic concepts with many examples, so if you don't understand something, go read the chapter.
- Three papers on data visualization
  - **Note:** you can also do without these and rely on slides and practice only; provided for the purpose of deepening.

# DATA: FROM SOURCES TO SENSES

Sources are almost  
NEVER in a shape fit  
for your purpose

**sources**

**senses**

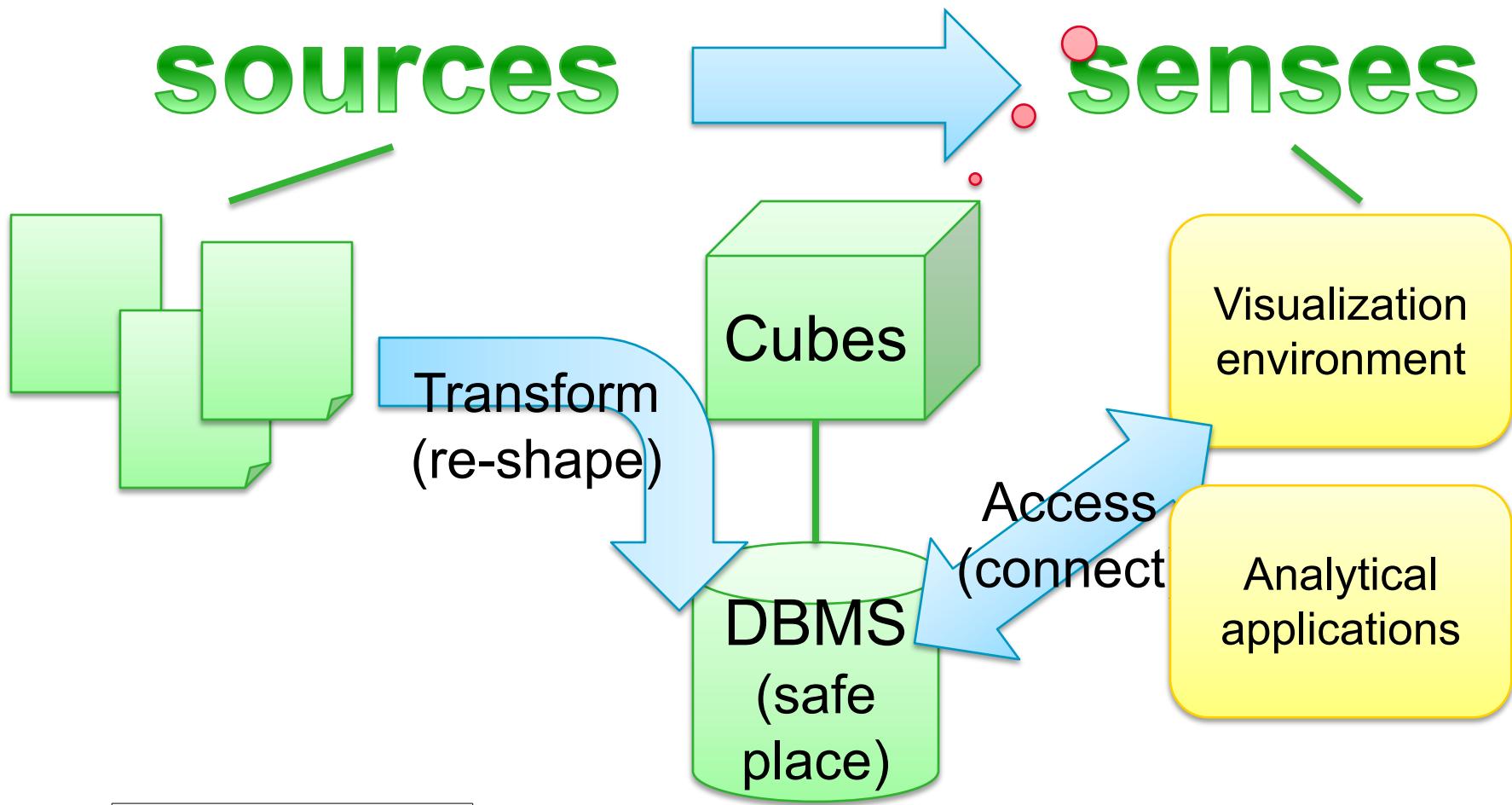


Visualization  
environment

Analytical  
applications

# DATA: FROM SOURCES TO SENSES

Generic shape that fits analytical purposes



# WHY CUBES

---

- The **cube** is a *generic shape* for data that fits *analytical purposes*
- A dataset collection often contains many related cubes
  - Each focusing on one or more *facts*
  - Related through shared standardized *dimensions*
- Data is an *asset*
  - It should not live in files transferred by email or download
  - It should live in a safe place: a DBMS
  - Data is something you *connect* to

Example: [CBS StatLine](#): cubes with access API

# METHOD

---



Two flavours:  
R-based  
Tool-based

1. Design cube (star schema)
  - a) Determine questions the data should answer
  - b) Envision tabular reports that may answer those questions
  - c) Determine for each question and report, the fact, the dimensions, and granularity
  - d) Combine into one star schema
  - e) Formulate what one row in fact table means
2. Design associated table structure (UML)
3. Create (empty) tables in database (SQL)
4. Prepare data and fill tables (SQL)

UNIVERSITY OF TWENTE.

# DATABASES





# PERSPECTIVE

---

- A **database** is a possibly large collection of data
    - that has to be **exchanged/shared, searched, corrected/supplemented**, etc.
    - and that **under no circumstances** may get lost or **corrupted** in any way
  - A DBMS is software that manages databases, allows these actions, and makes sure your data is safe
- 
- “Information is an asset”
  - Availability, reliability, performance, scalability, security

# THE DATA IS OFTEN STRUCTURED IN TABLES

THE PRIMARY 'SHAPE'

**Table** consists of

- **Records:** Rows in the table
- **Attributes:** Columns in the table

**Instance data:**  
The 'real' data in  
the table, the  
contents

**Schema:**  
Description of the  
table structure

Flight		
Number	From	To
KL123	AMS	VIE
OS45	VIE	AMS
KL234	AMS	BRU
NW678	AMS	NYJ
:	:	:

Airport	
Code	City
AMS	Amsterdam
BRU	Brussels
VIE	Vienna
NYJ	New York
:	:

Flight(**number**:STRING, from:STRING, to:STRING)

Airport(**code**:STRING, city:STRING)

The picture can't be displayed.

# CONCEPT “KEY”

---

**Key:** collection of one or more attributes that

- **Uniquely determine** a record in the table
- Primary key: one ‘most important’ key
- Surrogate key: artificially added code or number to function as a key

**Foreign key:** attribute(s) in a table that form a **reference** to the (primary key of) one or more records in another relation.

# THE DATA IS OFTEN STRUCTURED IN TABLES

THE PRIMARY 'SHAPE'

**Table** consists of

- **Records**: Rows in the table
- **Attributes**: Columns in the table

Primary key

**Instance data**:  
The 'real' data in  
the table, the  
contents

**Schema**:  
Description of the  
table structure

Flight			Airport	
Number	From	To	Code	City
KL123	AMS	VIE	AMS	Amsterdam
OS45	VIE	AMS	BRU	Brussels
KL234	AMS	BRU	VIE	Vienna
NW678	AMS	NYJ	NYJ	New York
:	:	:	:	:

Flight(**number**:STRING, from:STRING, to:STRING)

Airport(**code**:STRING, city:STRING)

The picture can't be displayed.

# SQL

---

SQL is one language for:

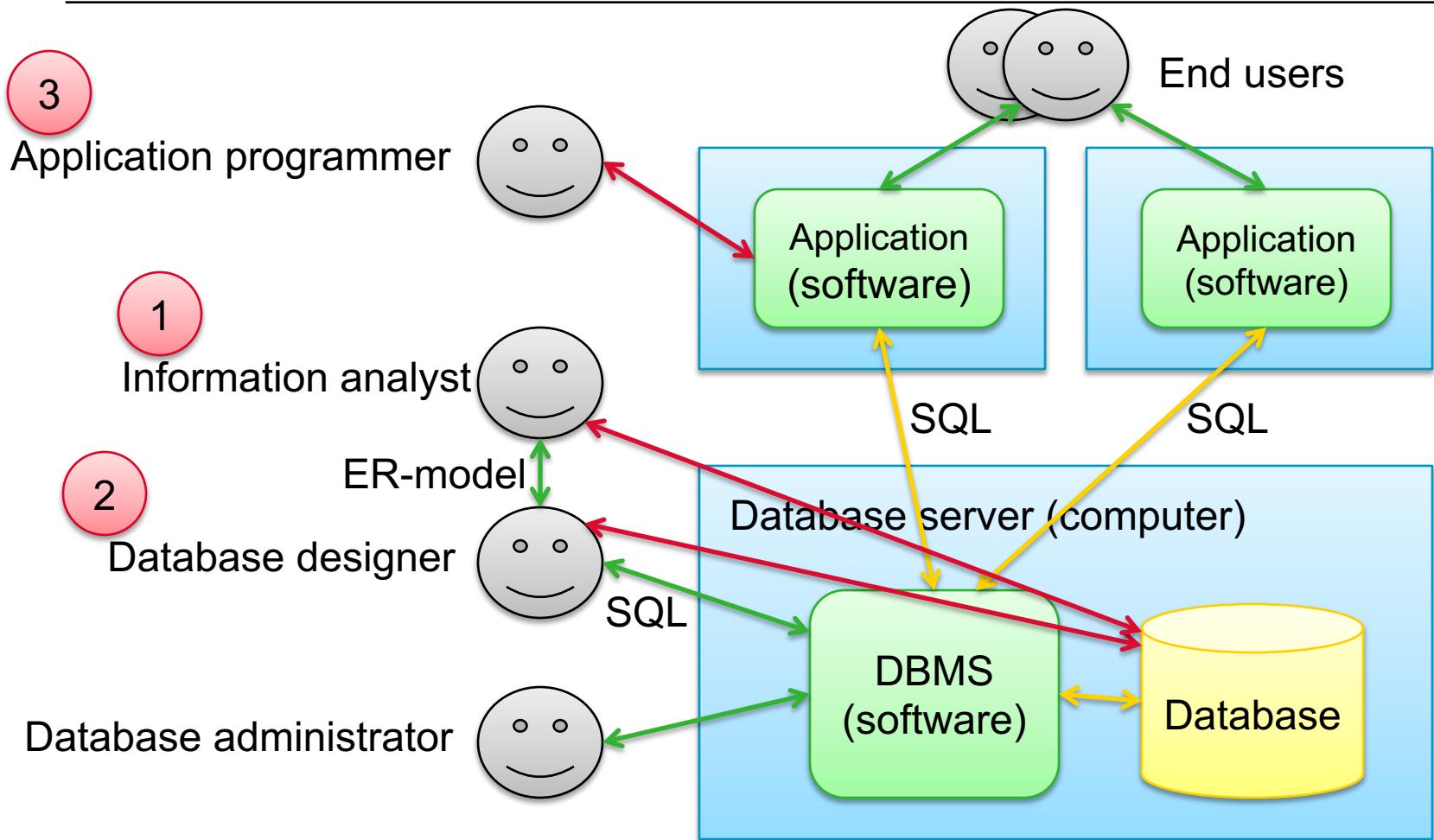
- Data definition (DDL)
- Data manipulation (DML)
- Storage definition (SDL)

Example: “all domestic KLM flights”

```
SELECT V.*, L1.city, L2.city  
FROM FlightV, Airport L1, Airport L2  
WHERE V.number LIKE 'KL%'  
      AND L1.country = 'Netherlands' AND V.from = L1.code  
      AND L2.country = 'Netherlands' AND V.to = L2.code
```



# TERMINOLOGY AND ROLES



The picture can't be displayed.

# DATABASE SERVER AND DATABASE CLIENT

---

## Database Server

- This is the computer running the DBMS software (Database Management System)
- It runs in the background serving (SQL) requests and keeping your data safe
- We use PostgreSQL pre-installed on bronto.ewi.utwente.nl

## Database client

- A tool accessing the database server
- We use PhpPgAdmin for database administration.
- We use R and Kettle / Spoon for data cleaning / transformation
- We use Tableau for data visualization
- All are DB clients connecting in a standard way to the server

# DATABASE STUFF PRE-INSTALLED

---

- The database server (PostgreSQL) and database management tool (PhpPgAdmin) are pre-installed on `bronto.ewi.utwente.nl`
- Each group has their own database
- You need credentials (username / password) for this, which you can obtain from [DAB](#).

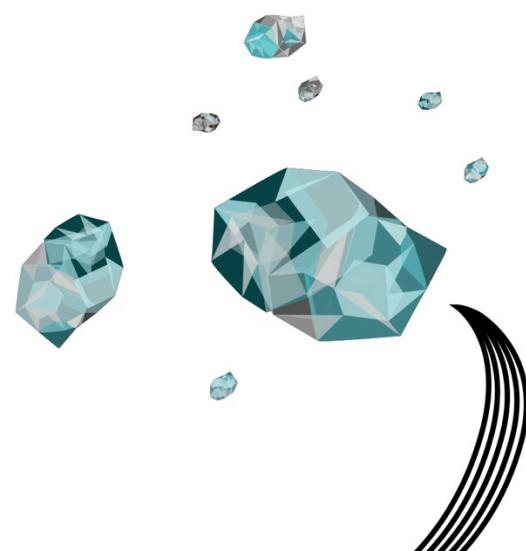
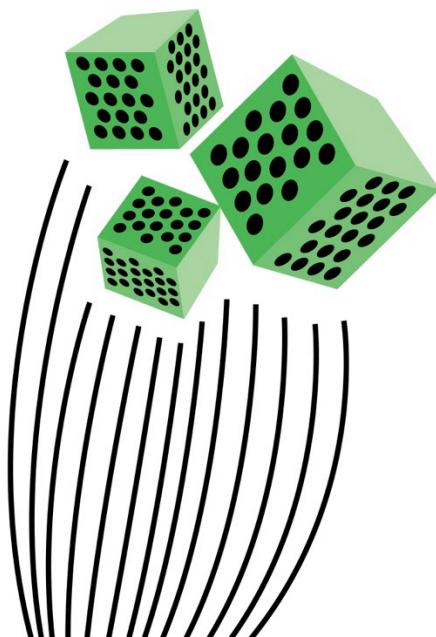
Let's show you how to login ...

<http://bronto.ewi.utwente.nl/phppgadmin/>



UNIVERSITY OF TWENTE.

# THE MANY SHAPES OF DATA



# THE DATA IS OFTEN STRUCTURED IN TABLES

THE PRIMARY 'SHAPE'

**Table** consists of

- **Records**: Rows in the table
- **Attributes**: Columns in the table

Primary key

**Instance data**:  
The 'real' data in  
the table, the  
contents

**Schema**:  
Description of the  
table structure

Flight			Airport	
Number	From	To	Code	City
KL123	AMS	VIE	AMS	Amsterdam
OS45	VIE	AMS	BRU	Brussels
KL234	AMS	BRU	VIE	Vienna
NW678	AMS	NYJ	NYJ	New York
:	:	:	:	:

Flight(**number**:STRING, from:STRING, to:STRING)

Airport(**code**:STRING, city:STRING)

The picture can't be displayed.

# DATA IS ALMOST NEVER IN THE DESIRED SHAPE

---

Even if it is a nice table  
the rows and columns  
are not as you desire

# **EXAMPLE: %SCHOOLING IN THE WORLD**

[HTTP://BARROLEE.COM/DATA/OUT\\_DEFI.HTM](http://BARROLEE.COM/DATA/OUT_DEFI.HTM)

---

Suppose I want to

- Analyze data on percentage of population who go to school
- ... in the different countries
- ... male vs. female
- ... different kinds of schools
- ... over the years

# EXAMPLE: %SCHOOLING IN THE WORLD

THIS IS WHAT THE SOURCE DATA LOOKS LIKE

Country	Population	Female A	Male A	Female B	Male B	Female C	Male C
Albania	4356456	1	2	9	8	12	11
Finland	456765	2	3	8	7	12	11
Netherlands	546745	3	4	7	6	11	12
Germany	1234255	4	5	6	5	11	12

Country	Population	Female A	Male A	Female B	Male B	Female C	Male C
Brazil	3865234	2	1	1	1	23	19
Argentina	2637564	4	4	3	4	21	24
Uruguay	645354	6	5	5	4	25	22
Venezuela	834565	8	8	7	8	18	16
Colombia	1762435	9	7	6	6	20	13

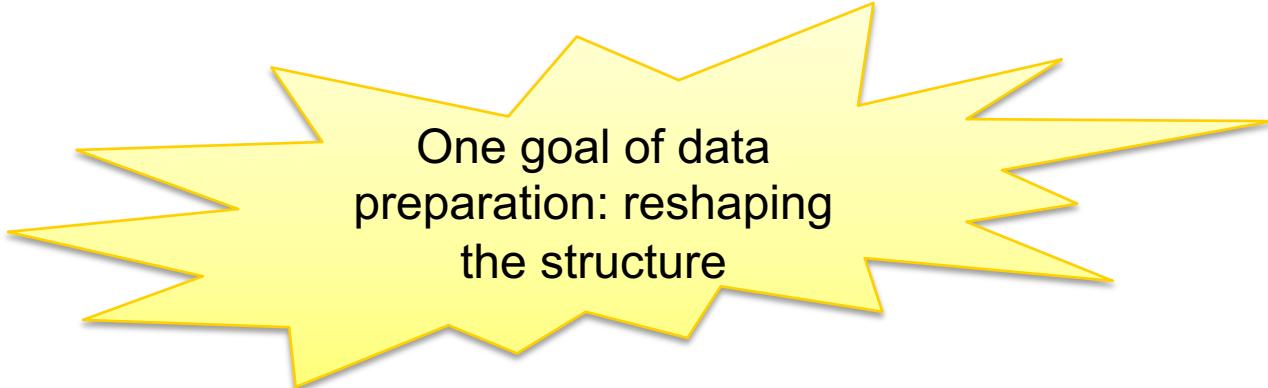
# EXAMPLE: %SCHOOLING IN THE WORLD

WHAT RESHAPING (DATA TRANSFORMATION) NEEDS TO BE DONE?

---

Source has

- More attributes than needed
- Data in different attributes of the same row, that I want to have on separate rows
- Data is in different files that I want in one table



A large yellow starburst graphic with several points and a wavy top edge, centered on the slide. Inside the starburst, the text "One goal of data preparation: reshaping the structure" is written in black.

One goal of data preparation: reshaping the structure

# **DATA IS ALMOST NEVER IN THE DESIRED SHAPE**

---

**Even if it is a nice table  
the contents (values)  
are not as you desire**

# DATA SEMANTICS: EXAMPLE

DB of department 1

enr	name	salary

DB of department 2

enr	name	salary

Data warehouse

enr	name	salary



What could be an obstacle

for a simple union of these tables?

- Situations
- Exceptions
- Semantical differences

# DATA SEMANTICS: EXAMPLE

CONTINUED

---

DB of department 1

enr	name	salary
3	M. van Keulen	100.000
4	R. Pieper	100.000
5	H. Blanken	200.000

DB of department 2

enr	name	salary
3	Keulen, M. van	3.781,50
6	Pieper, R.	18.907,51
9	Blanken, B.	7.563,00
12	Poel. M.	5.673,25
15	Vet, P. van der	NULL



# THERE IS MORE TO SHAPE THAN STRUCTURE

---

There is more to shape than the **structure** of the data

➤ The **contents** can also be in a wrong ‘shape’

## Contents

- What do the rows and columns really mean?
- What have people put in them? (exceptional cases)
- Missing values, inconsistent values, wrong values, ...

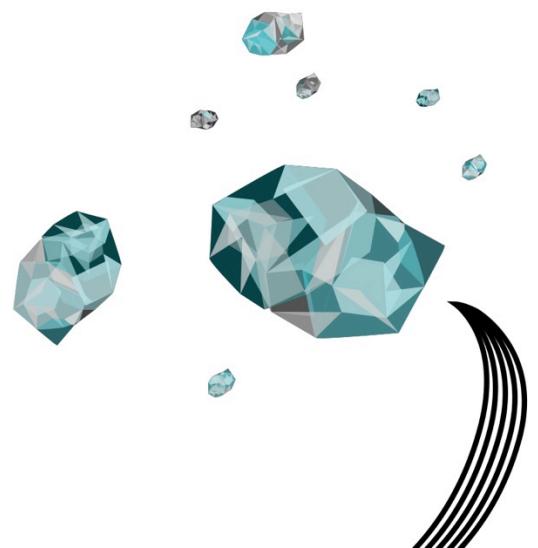
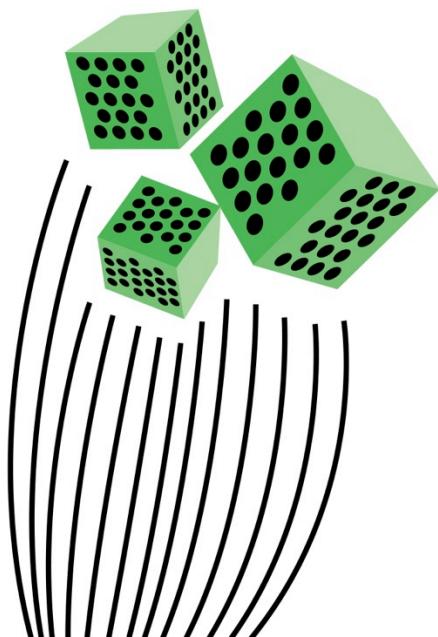
➤ Problems with **data quality** are often much much time-consuming to solve than re-shaping the structure

UNIVERSITY OF TWENTE.



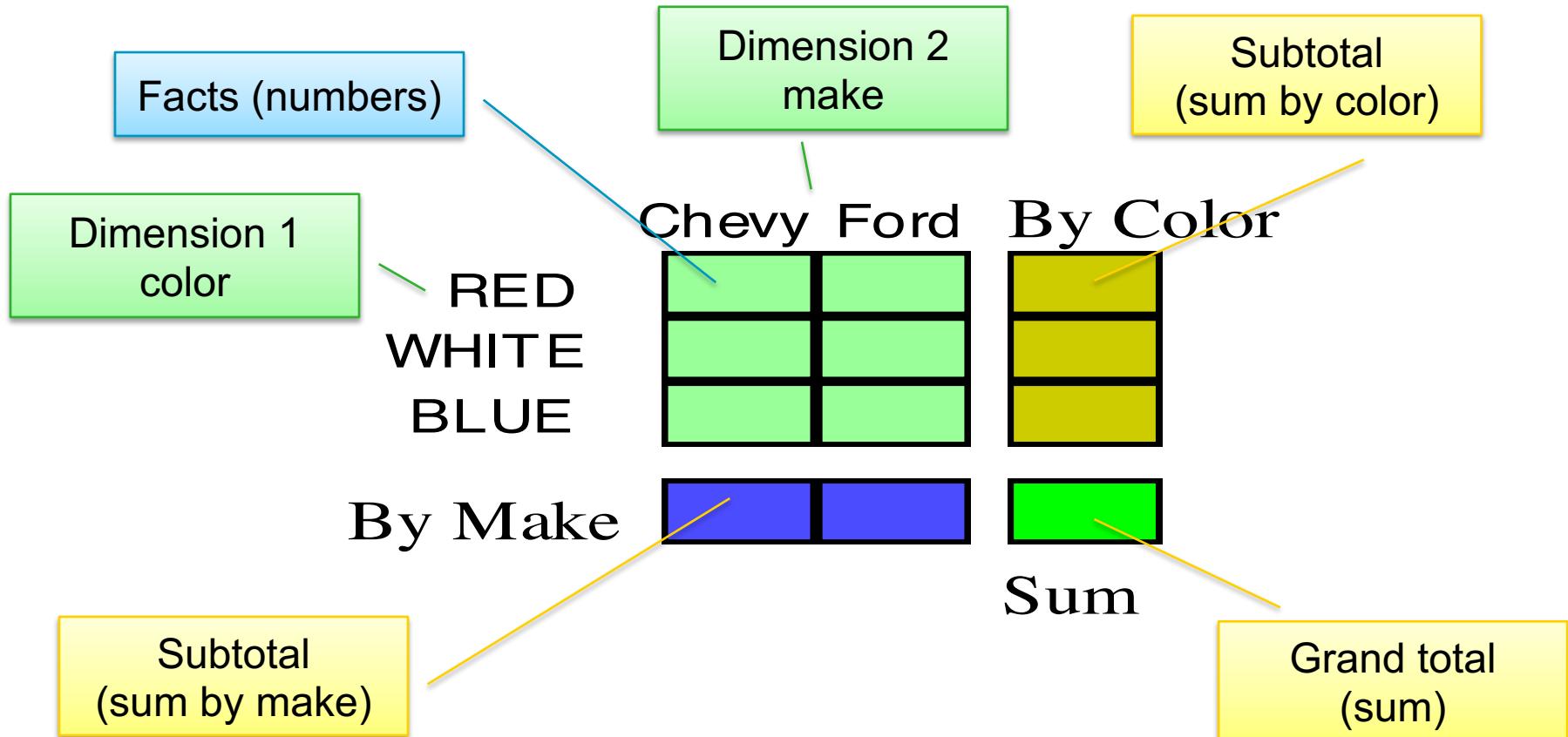
# CUBES

GENERIC SHAPE SUITABLE FOR ANALYTICS



# SPREADSHEET

IS A CUBE WITH TWO DIMENSIONS



# CUBE = MULTI-DIMENSIONAL DATABASE

= MULTI-DIMENSIONAL SPREADSHEET

## Aggregate

Sum

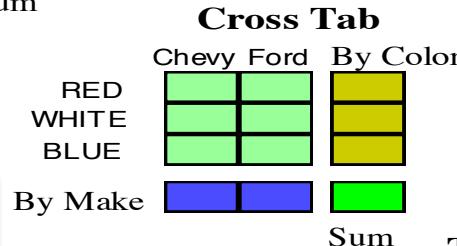
## Group By (with total)



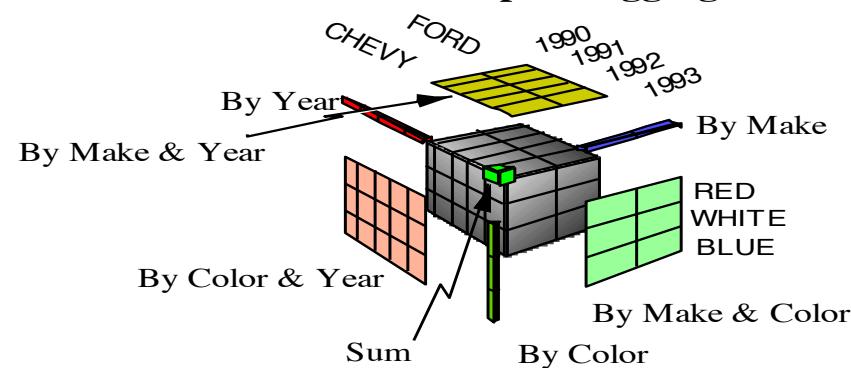
MDB contains spreadsheets with arbitrary numbers of dimensions (data **cubes**)

Data is either

- a **fact** with associated numerical *measure*, or
- a **dimension** which characterize the facts (mostly textual)

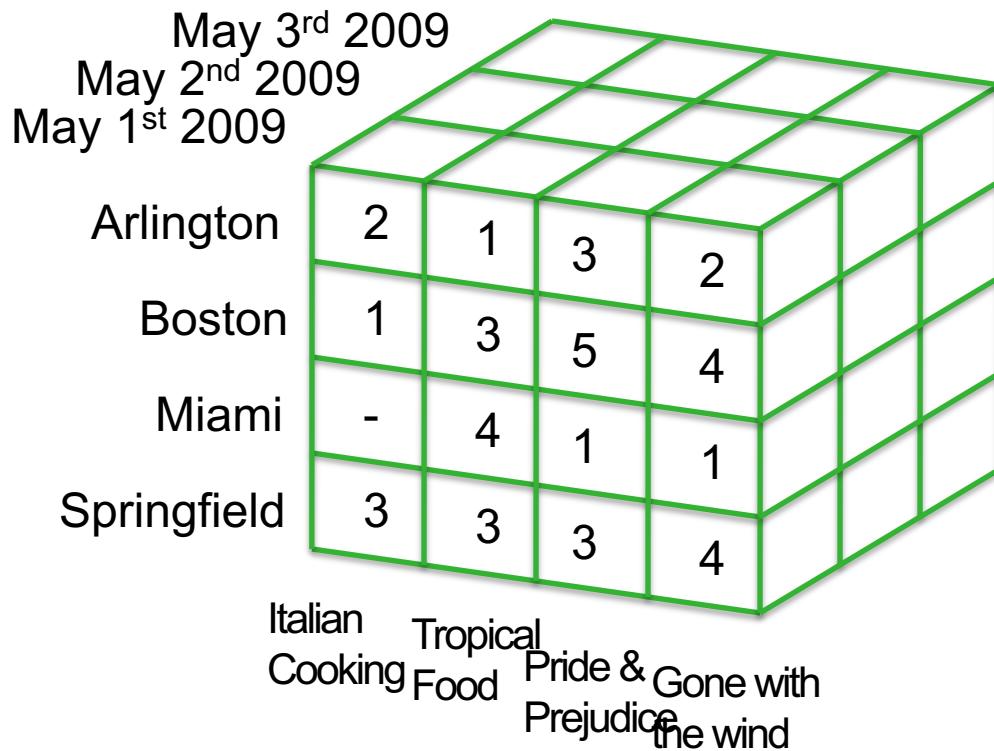


## The Data Cube and The Sub-Space Aggregates

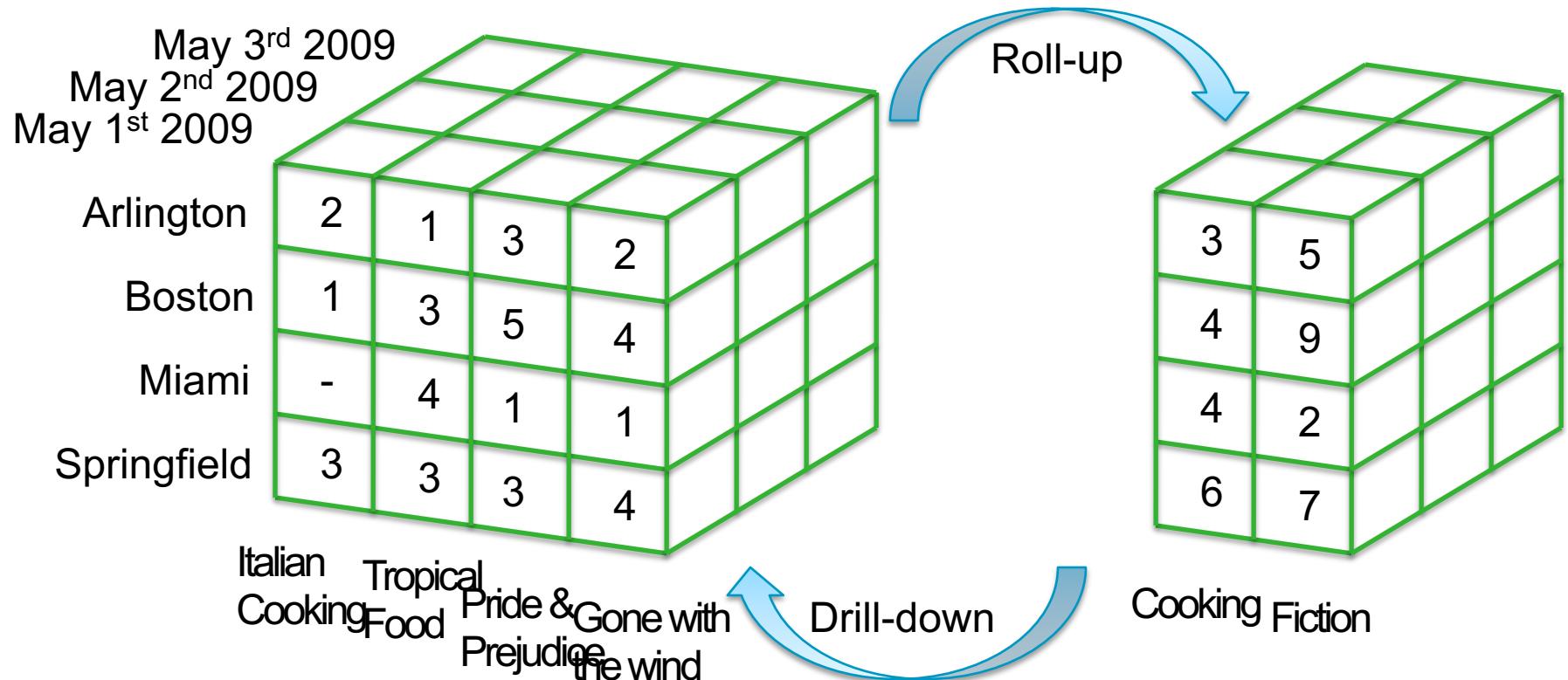


# EXAMPLE CUBE

---

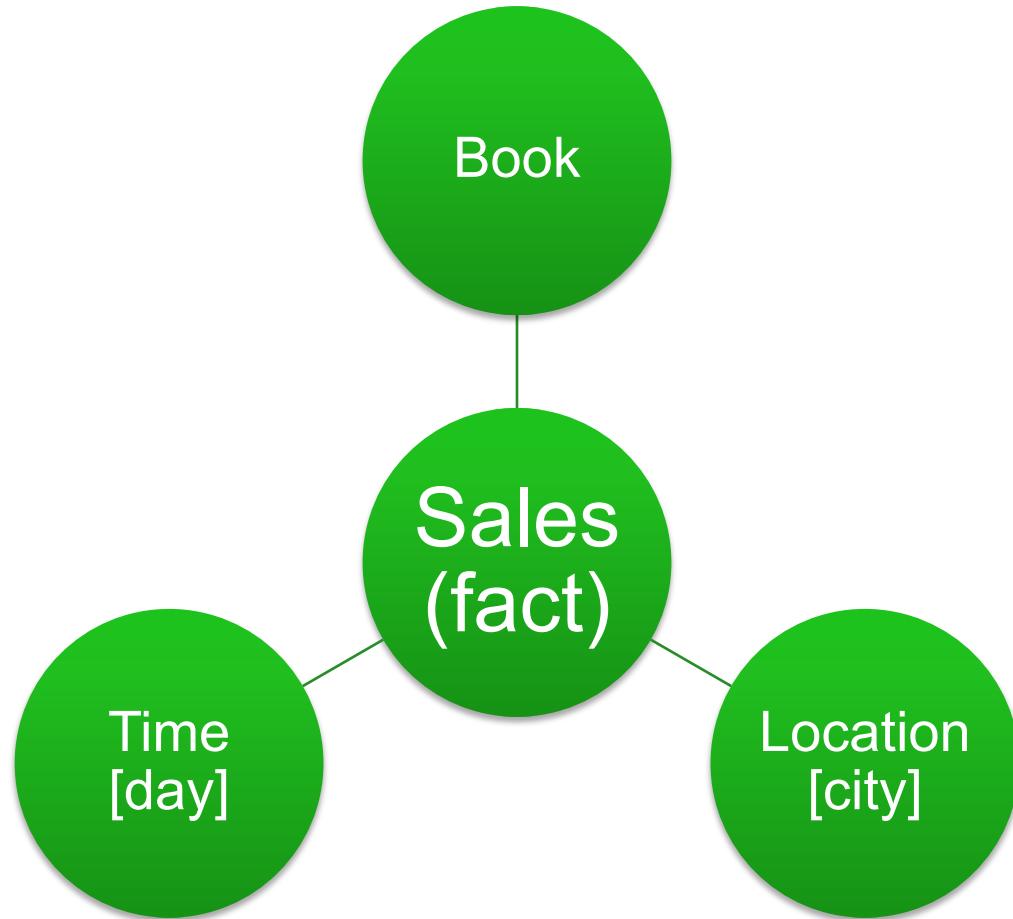


# DRILL-DOWN AND ROLL-UP



# STAR SCHEMA

---

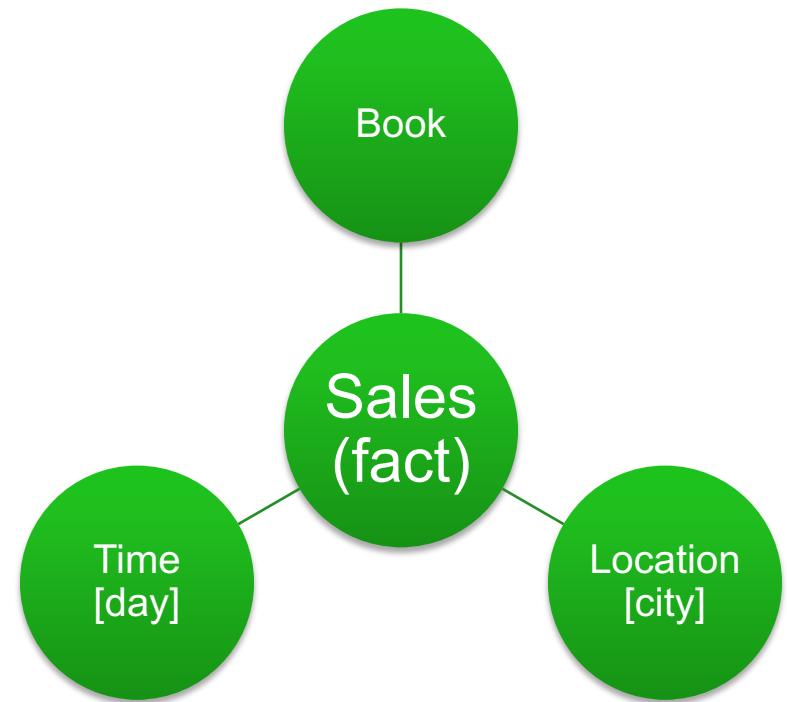


# WHAT DOES THIS STAR SCHEMA MEAN

---

In one spreadsheet / table

- One row of sales  
Per combination of  
Book, Time Unit, Location
  - Attributes for  
sales (fact: amount)  
book (dim: name, category)  
time (dim: day)  
location (dim: city, country)
  - For each dimension value  
there are multiple facts!
- More detail outside in!



# DIMENSIONS MAY HAVE GROUPINGS

Levels



*top (root)*

Grouping

Top

Dimension values

Country

City

Netherlands

Belgium

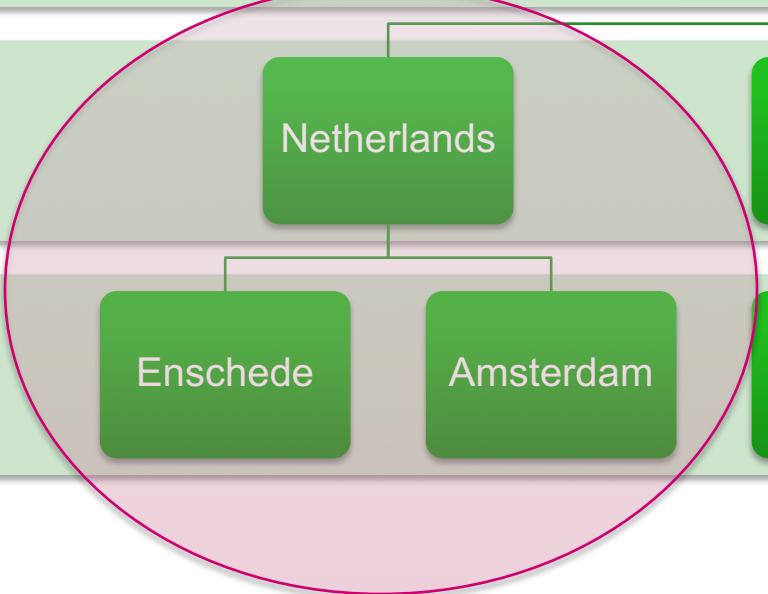
Germany

Enschede

Amsterdam

Brussels

Berlin



# FACTS

---

Fact has two components

- Numerical property (**measure**)
- Combination formula (e.g., aggregate like SUM)

Facts have a certain **granularity**

- **Sales** by *month* by *make* by *color*  
(fact by *dim1* by *dim2* by *dim3* ...)
- **Sales** by *year* by *make*

Second is **coarser**; first is **finer**

# EXAMPLE: %SCHOOLING IN THE WORLD

WHAT SHAPE WOULD BE MY TARGET SHAPE FOR THIS DATA?

Country	Population	Female A	Male A	Female B	Male B	Female C	Male C
Albania	4356456	1	2	9	8	12	11
Finland	456765	2	3	8	7	12	11
Netherlands	546745	3	4	7	6	11	12
Germany	1234255	4	5	6	5	11	12

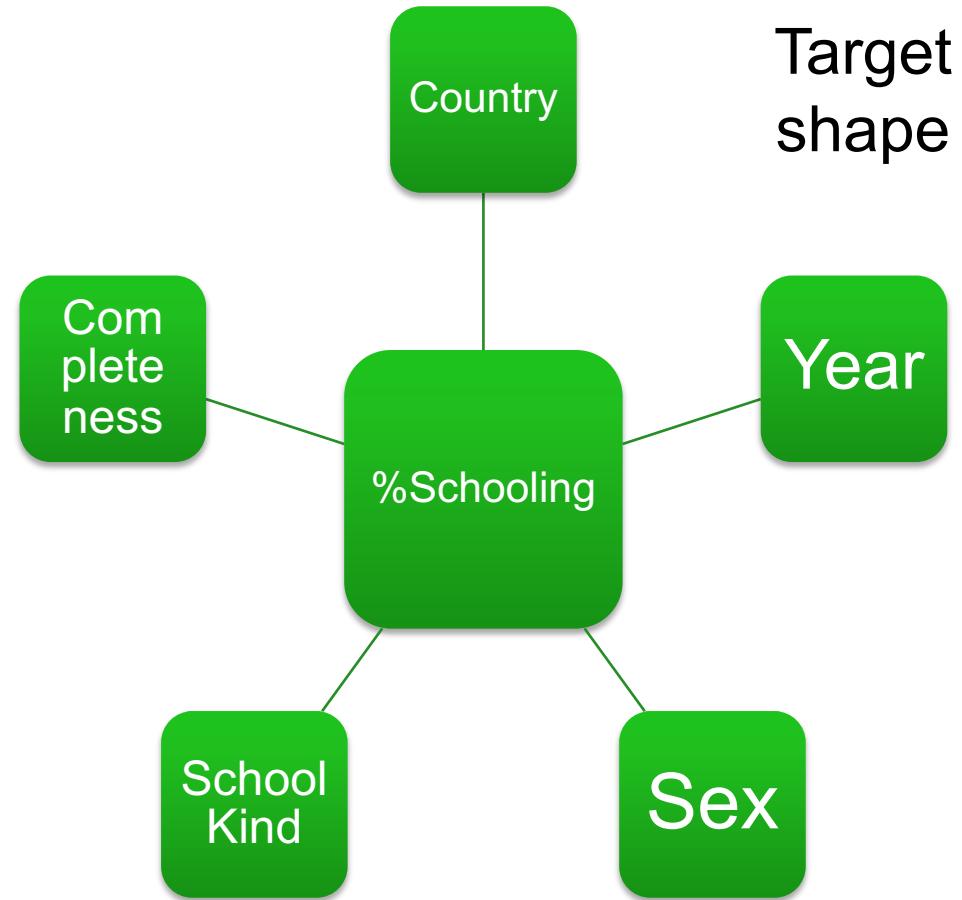
Country	Population	Female A	Male A	Female B	Male B	Female C	Male C
Brazil	3865234	2	1	1	1	23	19
Argentina	2637564	4	4	3	4	21	24
Uruguay	645354	6	5	5	4	25	22
Venezuela	834565	8	8	7	8	18	16
Colombia	1762435	9	7	6	6	20	13

# EXAMPLE: %SCHOOLING IN THE WORLD

[HTTP://BARROLEE.COM/DATA/OUT\\_DEFI.HTM](HTTP://BARROLEE.COM/DATA/OUT_DEFI.HTM)

Source has

- More attributes than needed
- Data in different attributes of the same row, that I want to have on separate rows
- Data in different files that I want in one table



Not a separate dimension,  
but hierarchy on Country

# EXAMPLE: %SCHOOLING IN THE WORLD

THIS SHAPE WOULD BE MY TARGET SHAPE FOR THIS DATA

%Schooling	Country	Continent	Sex	School kind	Completeness	Year
11	Albania	Europe	Male	Primary	Yes	2013
12	Albania	Europe	Female	Primary	Yes	2013
8	Albania	Europe	Male	Secondary	Yes	2013
9	Albania	Europe	Female	Secondary	Yes	2013
19	Brazil	South America	Male	Primary	Yes	2013
23	Brazil	South America	Female	Primary	Yes	2013
2	Brazil	South America	Male	Secondary	Yes	2013
1	Brazil	South America	Female	Secondary	Yes	2013
1	Brazil	South America	Male	Primary	No	2013
1	Brazil	South America	Female	Primary	No	2013

Fact

Dim 1

Dim 2

Dim 3

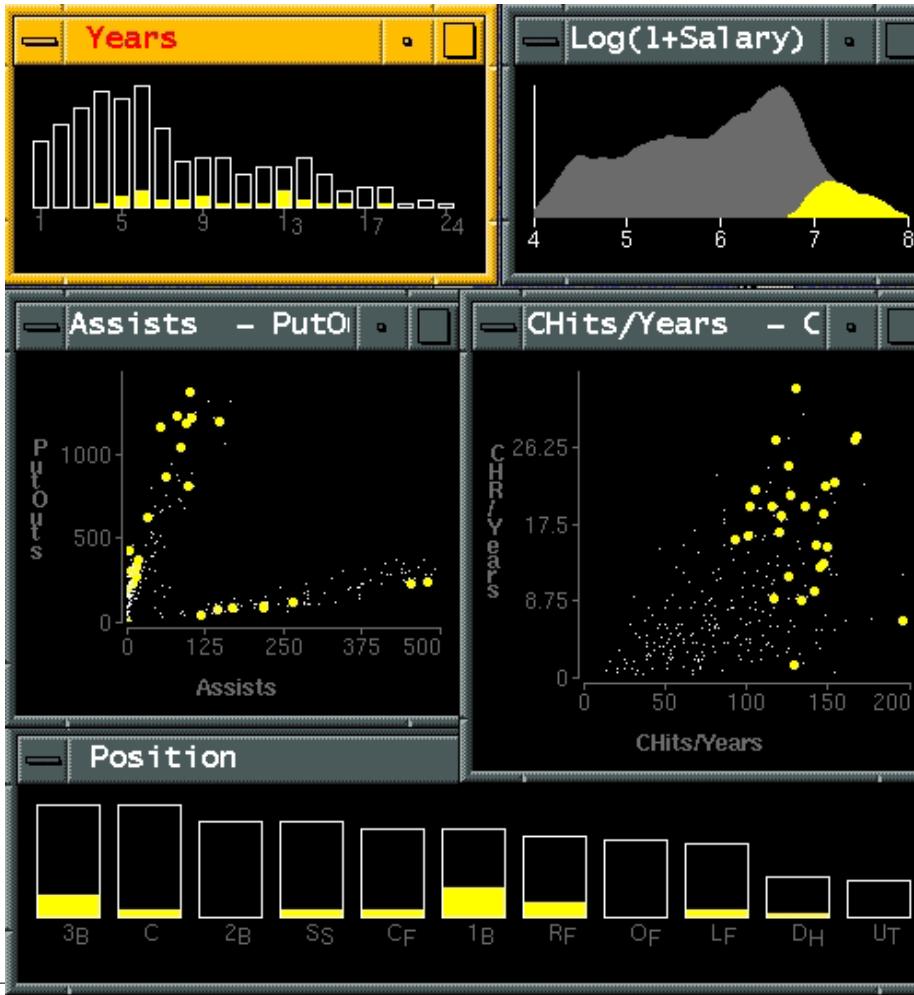
Dim 4

Dim 5

# FACTS & DIMENSIONS ARE IMPORTANT FOR VISUALIZATION

(FROM EICK & WILLS 95)

how long  
in majors



select high  
salaries

avg assists vs  
avg putouts →  
(fielding ability)

distribution  
of positions  
played

avg career  
HRs vs avg  
career hits  
(batting ability)



# ATTRIBUTE TYPES

---

Not every analysis method can be applied to any data.  
Some have limitations regarding attribute types:

- Continuous vs. Discrete
  - Continuous: real numbers, coordinates, time
  - Discrete: integer, nominal, ordinal

Nominal: limited set of ‘labels’ or ‘categories’

- Example: Male, Female

Ordinal: same but with an order

- Example: Very Low, Low, Medium, High, Very High

# ATTRIBUTE TYPES (CONTINUED)

---

In programming languages, databases and tools, variables/attributes always have a type.

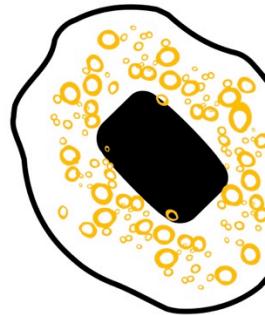
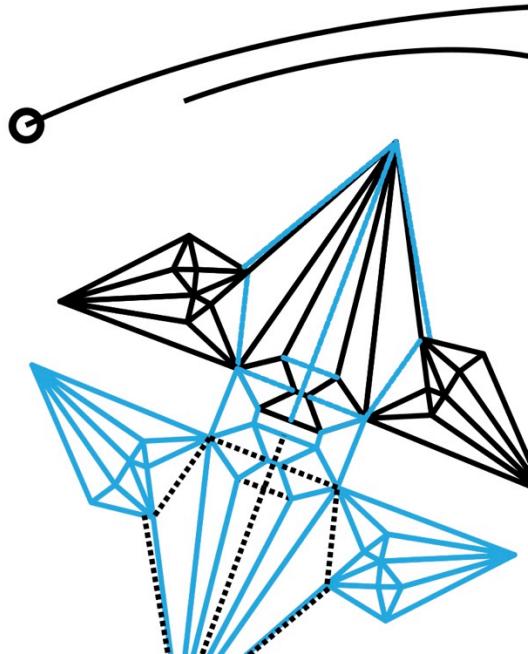
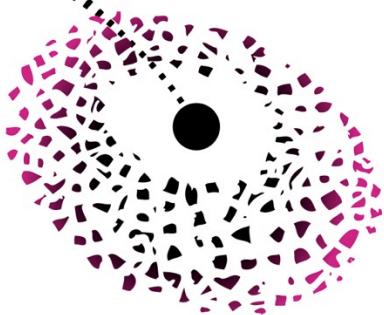
Some often occurring

- Integer: whole numbers upto certain maximum
- Float/double: real numbers of certain precision
- Date/Time/DateTime
- String: sequence of characters with certain length  
(in databases called “characted varying” or “varchar”)
- Boolean: true or false
- Char: one character



UNIVERSITY OF TWENTE.

# MULTIDIMENSIONAL MODELING



# METHOD FOR DATA PREPARATION

---



Start

## 1. Design cube (star schema)

- a) Determine questions the data should answer
- b) Envision tabular reports that may answer those questions
- c) Determine for each question and report, the fact, the dimensions, and granularity
- d) Combine into one star schema
- e) Formulate what one row in fact table means

## 2. Design associated table structure (UML)

- 3. Create (empty) tables in database (SQL)
- 4. Prepare data and fill tables (SQL)

(a) Determine questions  
the data should answer

(b) Envision tabular reports that  
may answer those questions

## MULTIDIMENSIONAL MODELING EXAMPLE

### ORCHARD

Large industrial orchard grows several fruits (apples, oranges, etc.) on many fields. Harvested fruits are automatically filtered for bad fruits before being sold. Orchard management wants to quickly and effectively determine the bad fields and weak fruits. Moreover, they like to analyze the effect of improvements.

#### Question(s)

- Determine bad fields and weak fruits
- Effects of improvements

Field	Fruit	Date	Condition	Harvest
A	Apples	1 Sep	Good	1400 kg
A	Apples	1 Sep	Bad	200 kg
A	Bananas	1 Sep	Good	800 kg
B	Apples	1 Sep	Bad	1900 kg

(c) Determine for each question and report, the fact, the dimensions, and granularity

## MULTIDIMENSIONAL MODELING EXAMPLE

### ORCHARD

Large industrial orchard grows several fruits (apples, oranges, etc.) on many fields. Harvested fruits are automatically filtered for bad fruits before being sold. Orchard management wants to quickly and effectively determine the bad fields and weak fruits. Moreover, they like to analyze the effect of improvements.

Field	Fruit	Date	Condition	Harvest
A	Apples	1 Sep	Good	1400 kg
A	Apples	1 Sep	Bad	200 kg
A	Bananas	1 Sep	Good	800 kg
B	Apples	1 Sep	Bad	1900 kg

### Dimensions

- Field, Fruit, Date, Condition

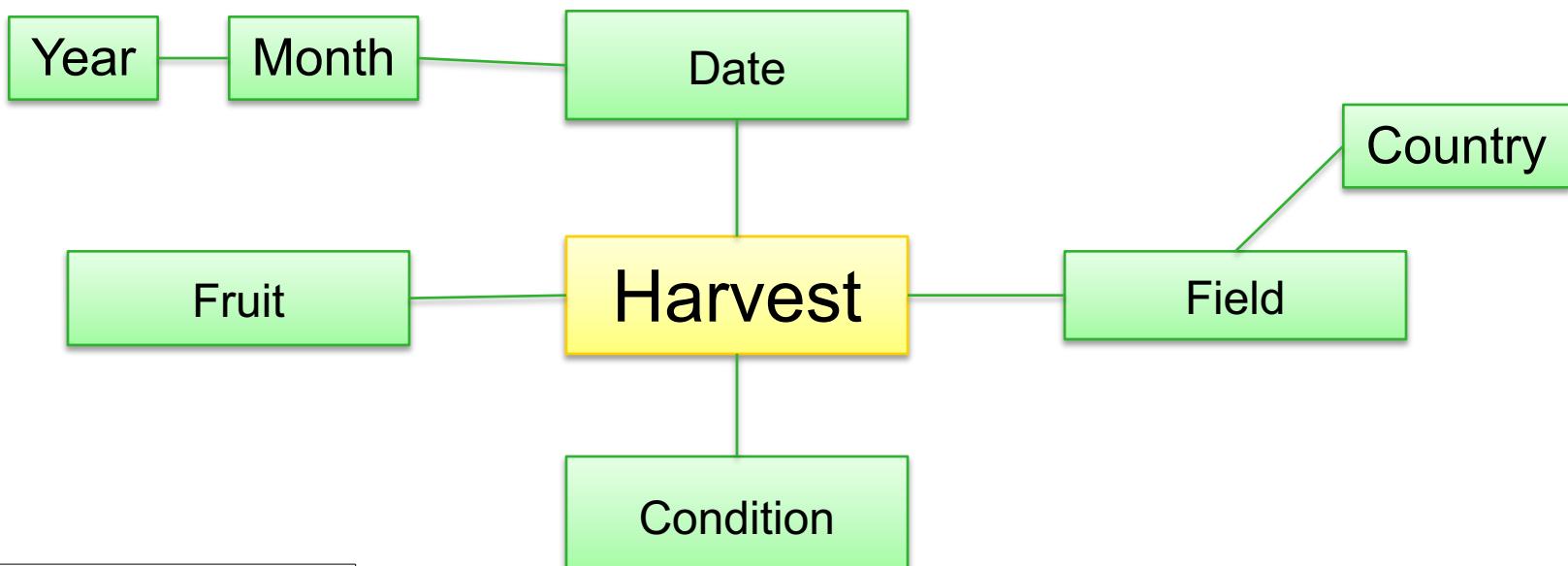
### Fact

- Harvest (weight)

# MULTIDIMENSIONAL MODELING EXAMPLE

## ORCHARD

Large industrial orchard grows several fruits (apples, oranges, etc.) on many fields. Harvested fruits are automatically filtered for bad fruits before being sold. Orchard management wants to quickly and effectively determine the bad fields and weak fruits. Moreover, they like to analyze the effect of improvements.



(e) Formulate what one row in fact table means

## MULTIDIMENSIONAL MODELING EXAMPLE

### ORCHARD

Large industrial orchard grows several fruits (apples, oranges, etc.) on many fields. Harvested fruits are automatically filtered for bad fruits before being sold. Orchard management wants to quickly and effectively determine the bad fields and weak fruits. Moreover, they like to analyze the effect of improvements.

#### Dimensions

- Field, Fruit, Date, Condition

#### Fact

- Harvest (weight)

#### What does this mean?

- **For each** time unit (say, day), **we store** the total weight of the harvest **per** fruit **for** bad and good fruit **seperately per field**.

# RULES OF THUMB

---

- Focus on the questions  
(don't be distracted by source data structure)
  - Dimensions:  
look for 'aspects' and formulations like "per X"; determine required granularity
  - Fact:  
on what numbers would an answer/report be based?
- Checks you can do afterwards
  - Dimensions are (almost always) independent
  - For all combinations of values of the dimensions, you (potentially) have one fact
  - Can all questions be answered?

# MULTIDIMENSIONAL MODELING EXAMPLE 2

## AUDIO/VIDEO SALES

Director of chain of high-end audio/video shops wants to know per month and city how many sales come from customers in the same city as the shop vs. sales from customers coming from other cities. He needs this to decide if he needs to open shops in all major cities or that customers are willing to travel to go to his shops.

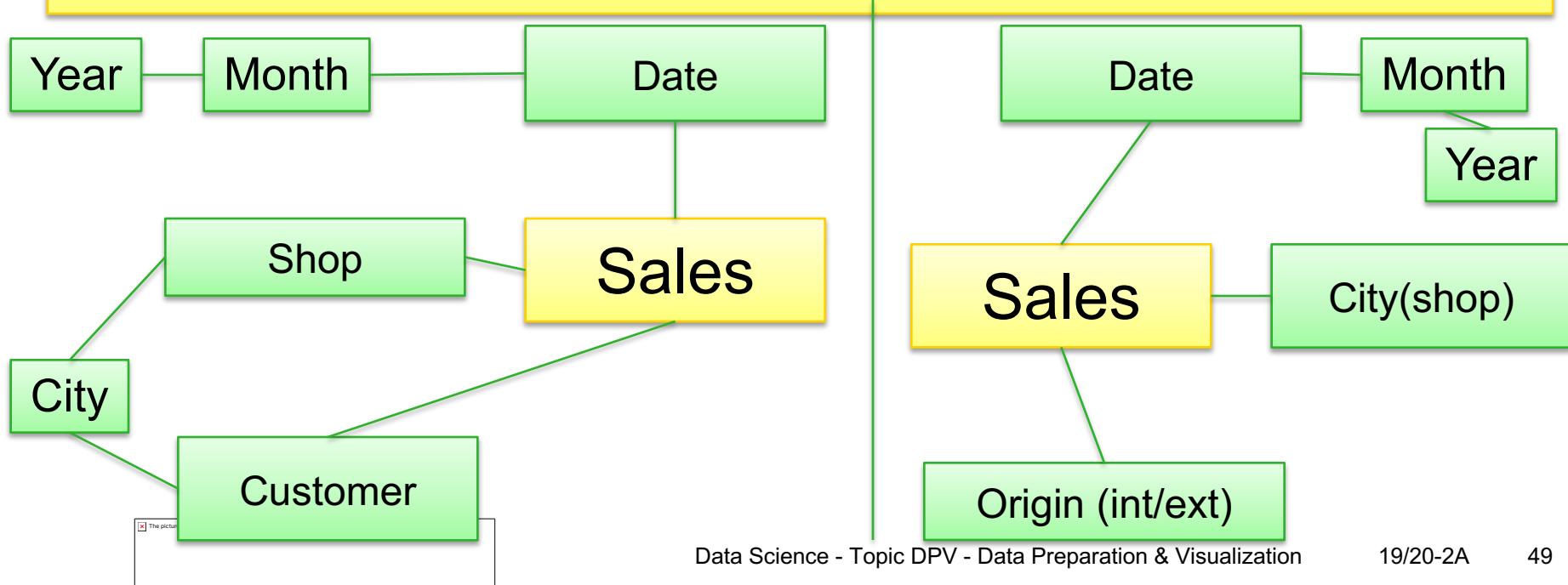
### Assignment:

- I will make initial design
- You tell me what I did wrong

# MULTIDIMENSIONAL MODELING EXAMPLE 2

## AUDIO/VIDEO SALES

Director of chain of high-end audio/video shops wants to know per month and city how many sales come from customers in the same city as the shop vs. sales from customers coming from other cities. He needs this to decide if he needs to open shops in all major cities or that customers are willing to travel to go to his shops.



# MULTIDIMENSIONAL MODELING EXAMPLE 2

## SOFTWARE LICENCES

Company spends much money on licenses for software. You start paying when you open software and stop when you terminate it. Software use is inter-active or running (e.g., simulation), but software can also be idle. Mgmt wants to know if they pay a lot of money of started software per category that is idle for a long time.

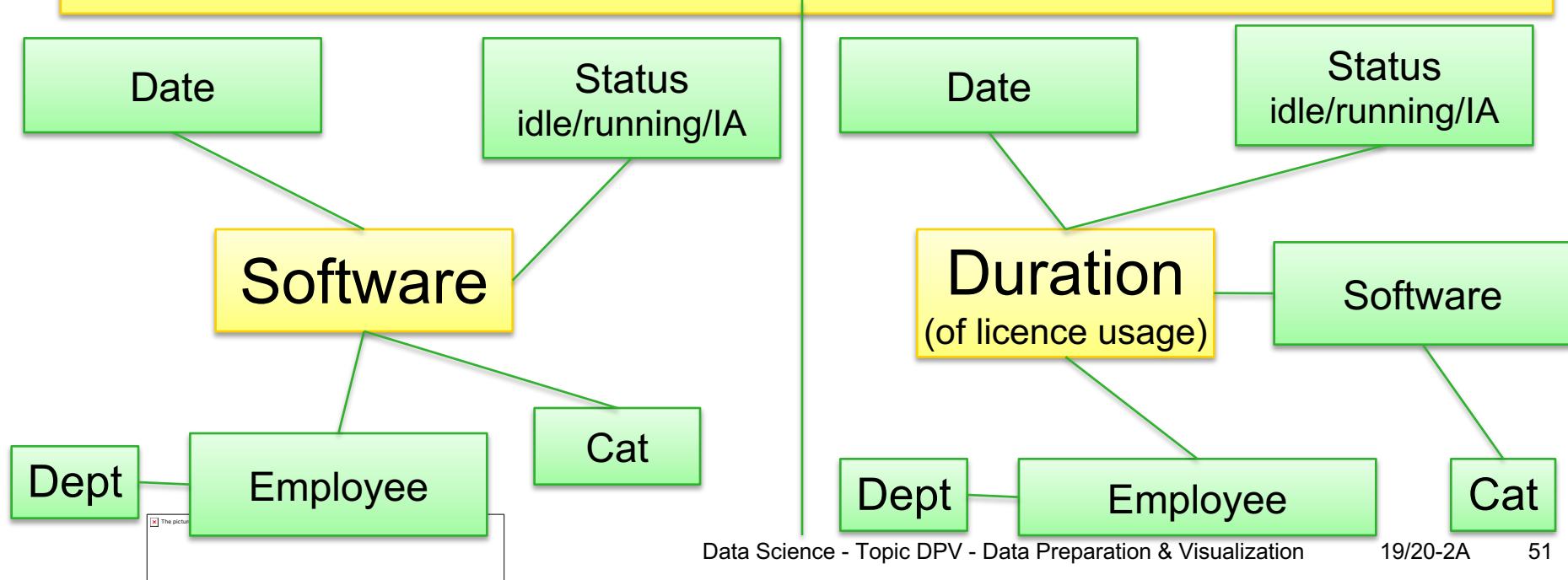
### Assignment:

- I will make initial design
- You tell me what I did wrong

# MULTIDIMENSIONAL MODELING EXAMPLE 2

## SOFTWARE LICENCES

Company spends much money on licences for software. You start paying when you open software and stop when you terminate it. Software use is inter-active or running (e.g., simulation), but software can also be idle. Mgmt wants to know if they pay a lot of money of started software per category that is idle for a long time.



# METHOD FOR DATA PREPARATION

---



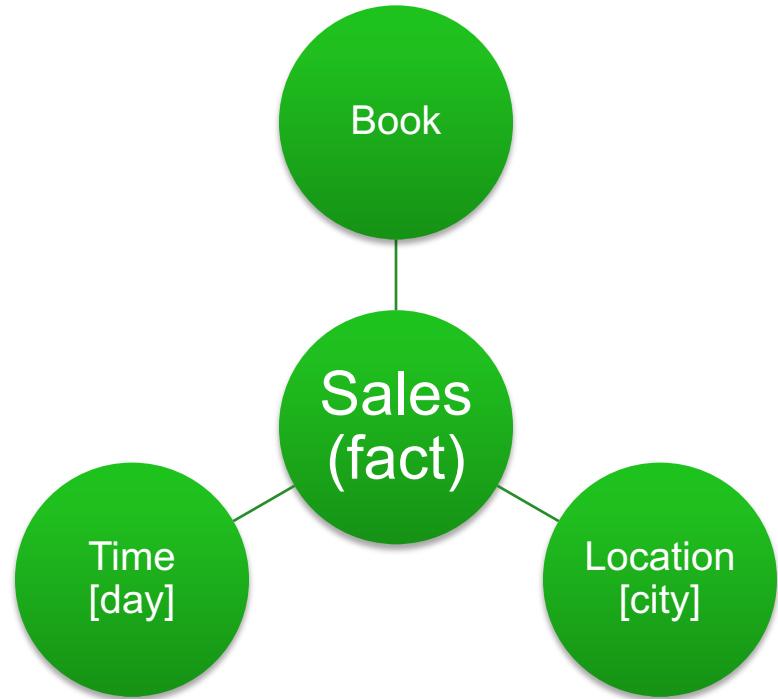
Done

1. Design cube (star schema)
    - a) Determine questions the data should answer
    - b) Envision tabular reports that may answer those questions
    - c) Determine for each question and report, the fact, the dimensions, and granularity
    - d) Combine into one star schema
    - e) Formulate what one row in fact table means
  2. Design associated table structure (UML)
  3. Create (empty) tables in database (SQL)
  4. Prepare data and fill tables (SQL)
- 
- Now

# REMEMBER THIS STAR SCHEMA?

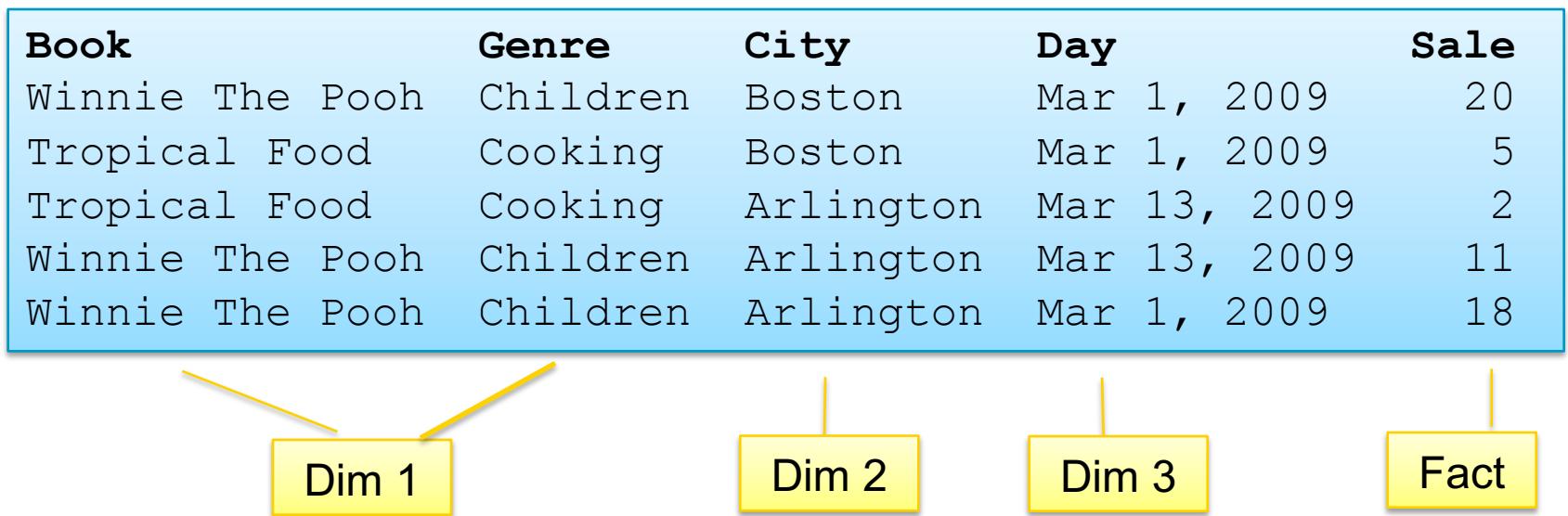
---

- One row of sales  
Per combination of  
Book, Time Unit, Location
  - Attributes for  
sales (fact: amount)  
book (dim: name, category)  
time (dim: day)  
location (dim: city, country)
  - For each dimension value  
there are multiple facts!
- More detail outside in!



# REALISING A DATA CUBE WITH ONE TABLE

---



# REALISING A DATA CUBE WITH RELATIONAL TABLES

THIS IS EXACTLY THE SAME DATA; JUST A DIFFERENT REPRESENTATION

Dimension table

BookID	Book	Genre
7493	Tropical Food	Cooking
9436	Winnie The Pooh	Children

Fact table

BookID	CityID	DayID	Sale
9436	854	2475	20
7493	854	2475	5
7493	876	3456	-
9436	876	2475	11
9436	876	2475	18

Fact

CityID	City
876	Arlington
854	Boston

key

DayID	Day	Month	Year
2475	Mar 1, 2009	Mar'09	2009
3456	Mar 13, 2009	Mar'09	2009

Where do all  
these ID's  
come from?

# CUBE TABLE DESIGN

---

How to map a star / snowflake schema to a table structure for a data warehouse?

- Fact entity is a separate table
    - Attributes: fact,  $\text{dim}_1$ ,  $\text{dim}_2$ , ...,  $\text{dim}_n$  ( $n$  dimensions)
    - A  $\text{dim}_i$  may consist of several attributes (groupings)
  - Row in fact table represents one fact value for one combination of dimension
- For some dimensions you may choose to put them in a table of their own
- Dimension table:  $\text{ID}_i$ , Value [, description, ...]
  - Replace  $\text{dim}_i$  with  $\text{ID}_i$

# CUBE TABLE DESIGN

IF YOU PUT EVERY DIMENSION IN A SEPARATE TABLE YOU WILL GET

---

How to map a star / snowflake schema to a table structure for a data warehouse?

- Fact entity is a separate table
  - Attributes: fact, ID<sub>1</sub>, ID<sub>2</sub>, ..., ID<sub>n</sub> (*n* dimensions)
- Each dimension a separate table
  - Attributes: ID, Value [, description, ...]
  - For each grouping in the hierarchy, add one attribute
- Row in dimension table represents one distinct dimension value with an ID
- Row in fact table represents one fact value for one combination of dimension values (using ID's to each dimension value)

# TABLE DESIGN EXAMPLE: INLINING ALL DIMENSIONS

STILL 4 DIMENSIONS!

## Fact table

## Dimensions

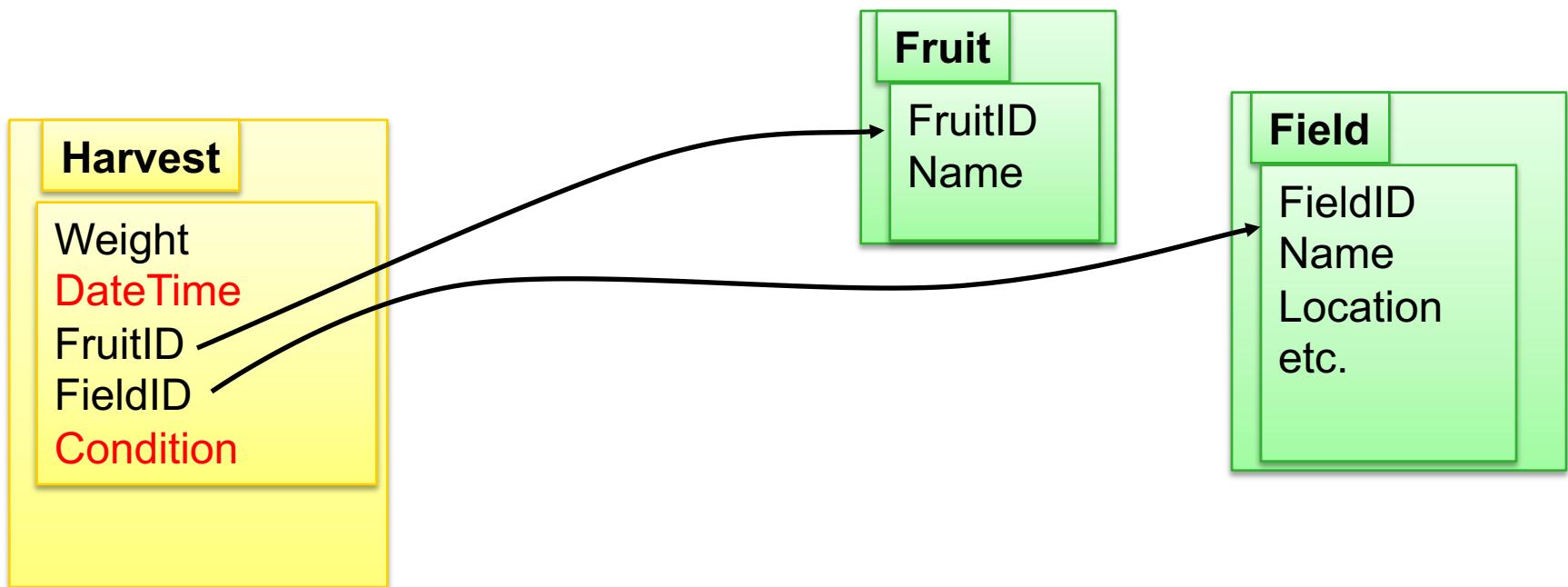


Fruit and Field are large dimensions likely to be used in other cubes as well. Therefore, these are advised to keep as separate tables.

# TABLE DESIGN EXAMPLE: INLINING CONDITION

STILL 4 DIMENSIONS!

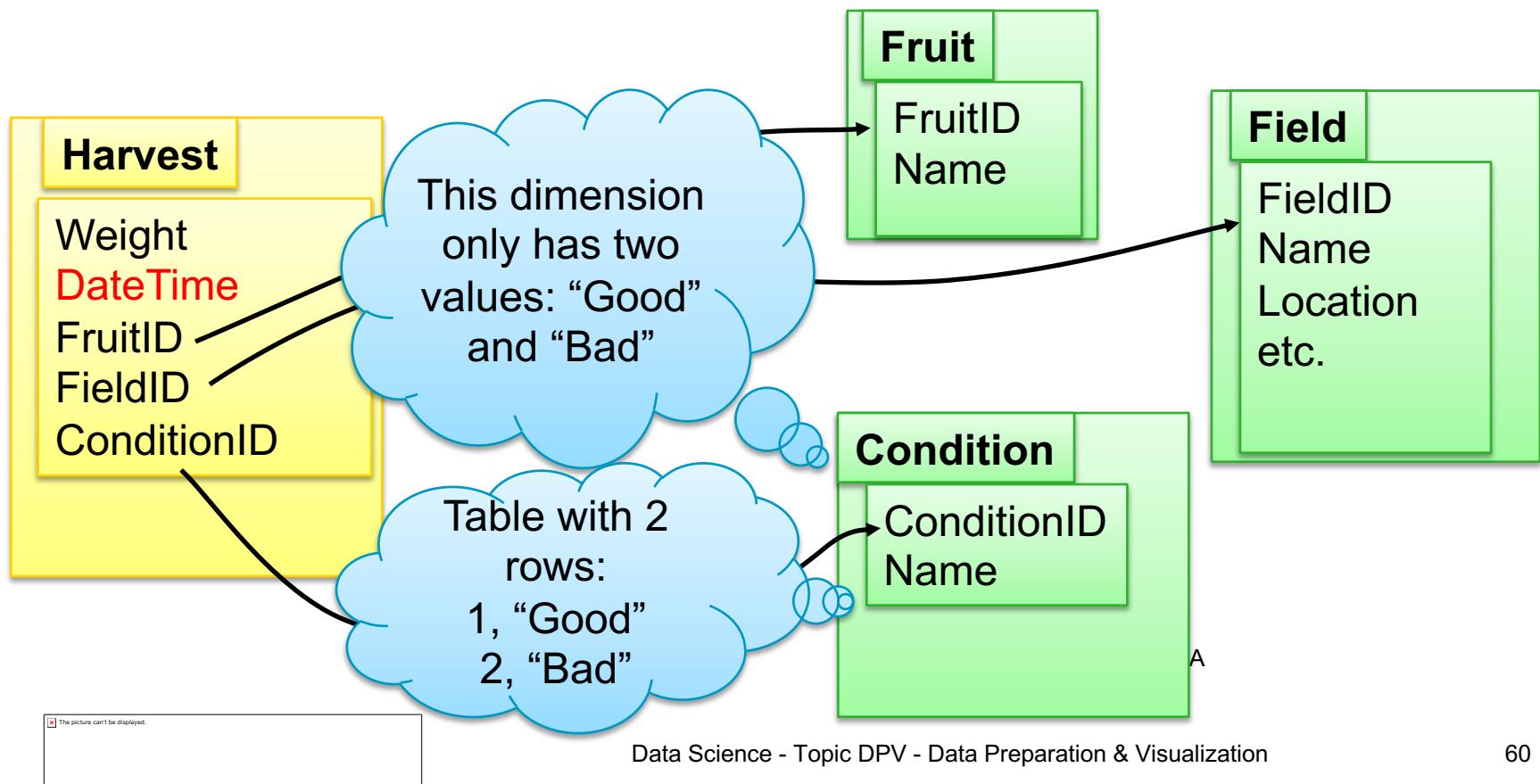
# Fact table Dimensions



# TABLE DESIGN EXAMPLE

STILL 4 DIMENSIONS!

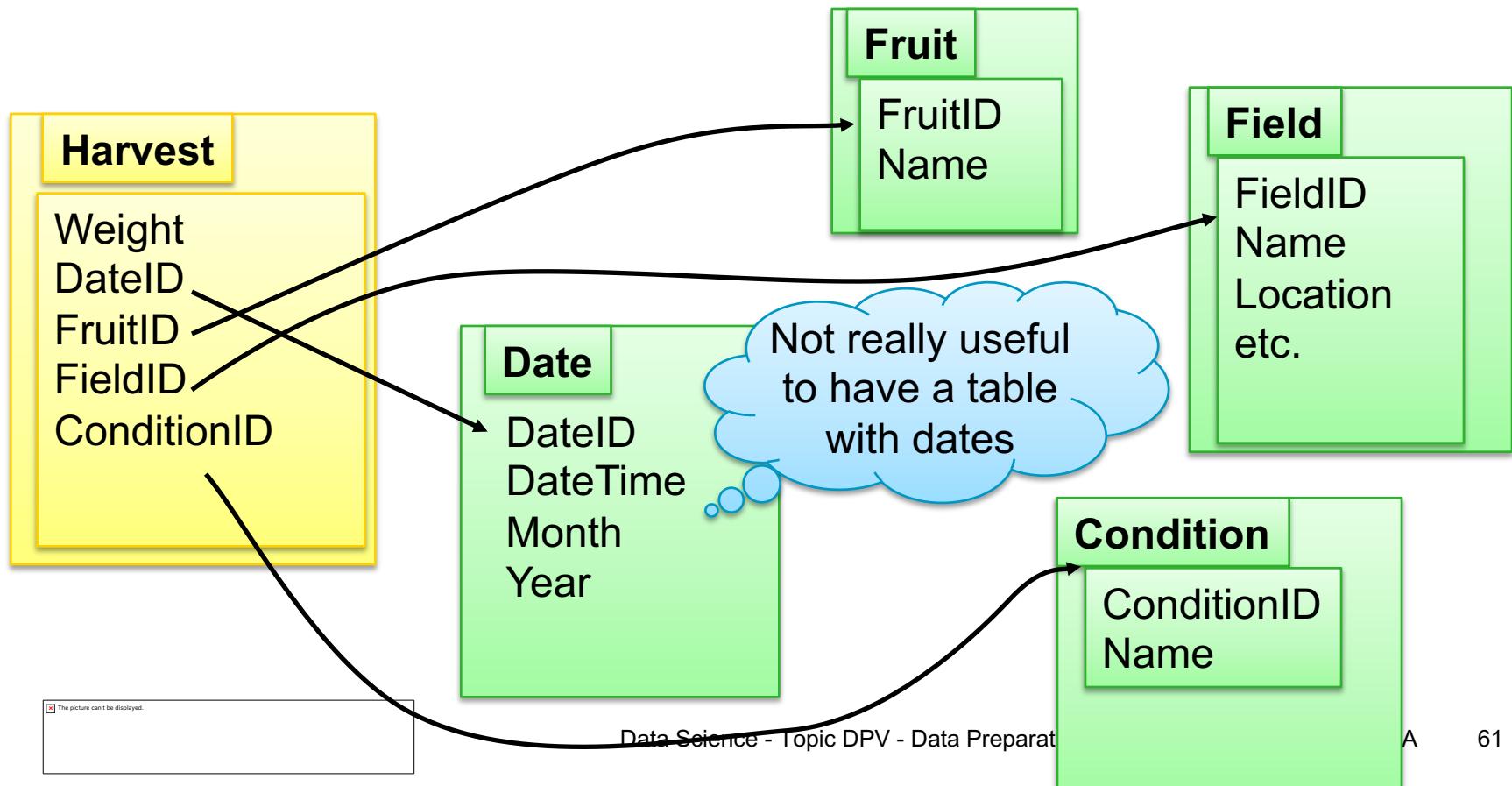
# Fact table Dimensions



# TABLE DESIGN EXAMPLE

ALL DIMENSIONS HAVE SEPARATE TABLES

# Fact table Dimensions



# IMPROVEMENTS AND DESIGN QUALITY

---

- Improvements: *inlining*
  - Not many possible values
  - Dimension is not re-used in other cubes
  - If hierarchical grouping is computable, extra attribute is not needed (month, year for date)
  - Dimension value itself can sometimes be used as ID (date)
- Design quality
  - Invent some example data (interrelated)
  - Each combination of dimension values gives one fact? Sparse? Discriminating enough?

# METHOD FOR DATA PREPARATION

---

- 
1. Design cube (star schema)
    - a) Determine questions the data should answer
    - b) Envision tabular reports that may answer those questions
    - c) Determine for each question and report, the fact, the dimensions, and granularity
    - d) Combine into one star schema
    - e) Formulate what one row in fact table means
  2. Design associated table structure (UML)
  3. Create (empty) tables in database (SQL)
  4. Prepare data and fill tables (SQL)
- 

# CREATE (EMPTY) TABLES IN DATABASE (SQL)

## Flight

number CHARACTER(6)

day DATE

time TIME

airplane INTEGER

**from CHARACTER(3)  
to CHARACTER(3)**

PRIMARY KEY number, day

## Airport

code CHARACTER(3) PRIMARY KEY

city CHARACTER VARYING

country CHARACTER VARYING

```
CREATE TABLE Airport (
    code CHARACTER(3),
    city CHARACTER VARYING,
    country CHARACTER VARYING,
    PRIMARY KEY (code)
);
```

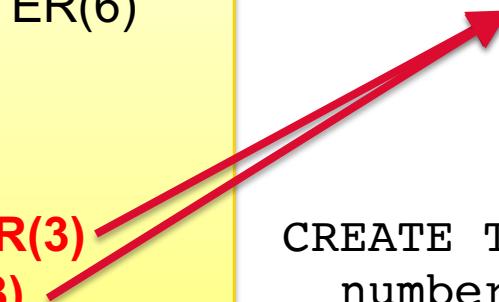
# CREATE (EMPTY) TABLES IN DATABASE (SQL)

## Flight

number CHARACTER(6)  
day DATE  
time TIME  
airplane INTEGER  
**from CHARACTER(3)**  
**to CHARACTER(3)**  
PRIMARY KEY number, day

## Airport

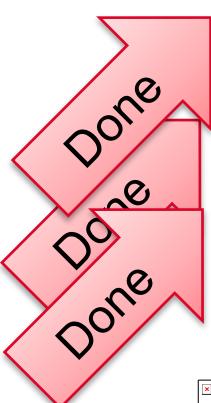
code CHARACTER(3) PRIMARY KEY  
city CHARACTER VARYING  
country CHARACTER VARYING



```
CREATE TABLE Flight (
    number CHARACTER(6),
    day DATE, time TIME,
    airplane INTEGER,
    from_ap CHARACTER(3),
    to_ap CHARACTER(3),
    PRIMARY KEY (number, day),
    FOREIGN KEY (from_ap)
        REFERENCES Airport(code),
    FOREIGN KEY (to_ap)
        REFERENCES Airport(code));
```

# METHOD FOR DATA PREPARATION

---

- 
1. Design cube (star schema)
    - a) Determine questions the data should answer
    - b) Envision tabular reports that may answer those questions
    - c) Determine for each question and report, the fact, the dimensions, and granularity
    - d) Combine into one star schema
    - e) Formulate what one row in fact table means
  2. Design associated table structure (UML)
  3. Create (empty) tables in database (SQL)
  4. Prepare data and fill tables (SQL)
- 

# STEP 4: PREPARE & FILL TABLES = ACTUAL RESHAPING

---

## Reshaping

- Reading sources
- Restructuring to match target star schema
- Data cleaning
- Writing to cube (i.e., the tables in the database)

## How

- SQL (if both sources and target are databases)
  - Any programming language
  - ETL or Data Wrangling tool
- 
- What we do (R-based flavour)
- What we do (Tool-based flavour)

# STEP 4: PREPARE & FILL TABLES = ACTUAL RESHAPING

---

Some advice on ETL construction method

- **Small do-test steps**

Do: Add only one or two small bits, then execute and verify the result, before continuing

Do not: Add many steps and then don't know where the mistake is when you receive an error

- **Read the error message carefully**

It may contain a lot of gibberish you don't understand, but part of it may provide clues to what is wrong

- **GIYF: Google Is Your Friend**

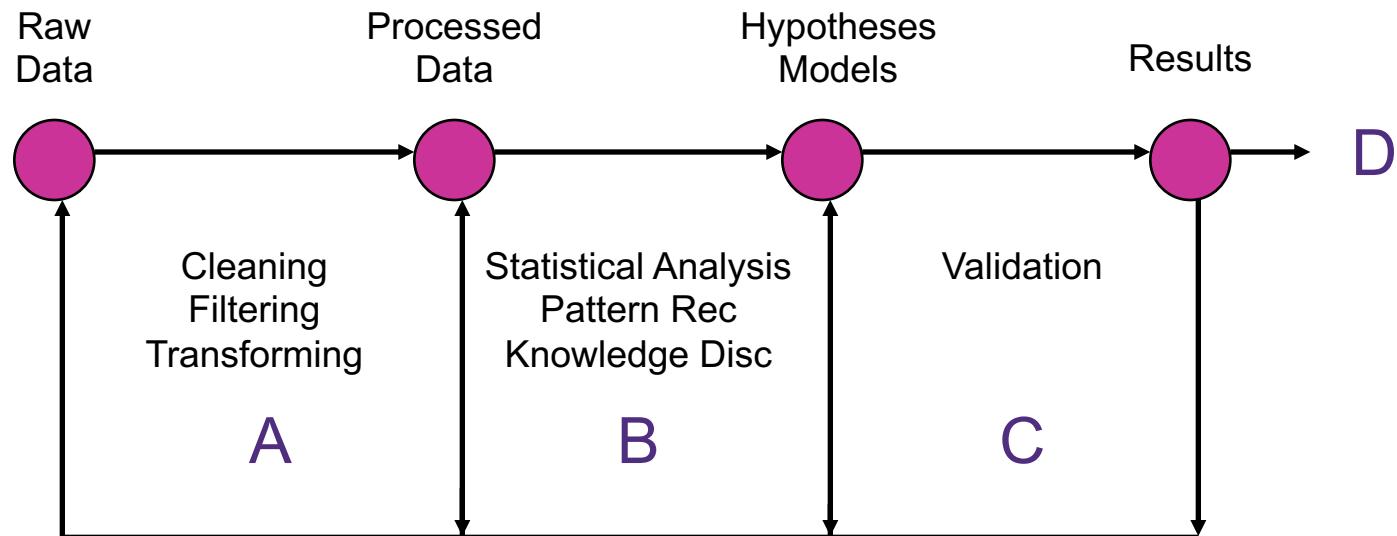
You may think Googling is not academic, but the internet is full of information on what may have caused certain errors and what you can do to fix them

UNIVERSITY OF TWENTE.

# DATA VISUALIZATION

# DATA ANALYSIS PIPELINE

---



# WHERE DOES VISUALIZATION COME IN?

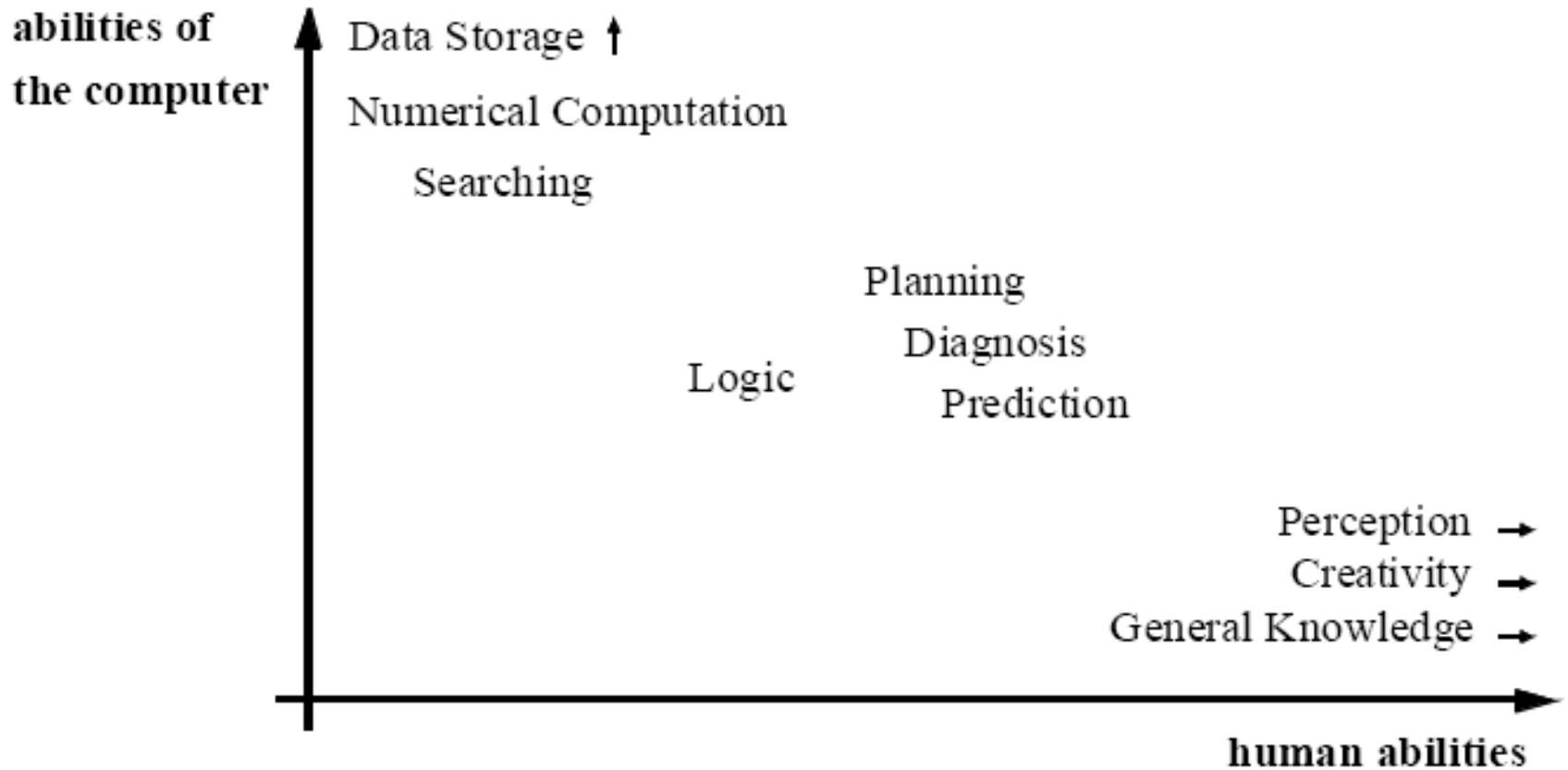
---

All stages can benefit from visualization

- A: identify bad data, select subsets, help choose transforms (**exploratory**)
- B: help choose computational techniques, set parameters, use vision to recognize, isolate, classify patterns (**exploratory**)
- C: Superimpose derived models on data (**confirmatory**)
- D: Present results (**presentation**)

# VISUALIZATION VS. DATA MINING

---



# VISUALIZATION VS. DATA MINING

---

## Traditional Mining Methods

- + Very precise results when the problem is stated exactly.
- + Potential for automation.
  - Large data sets may be analyzed at a time.

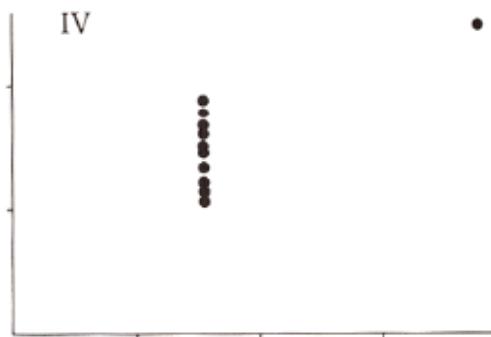
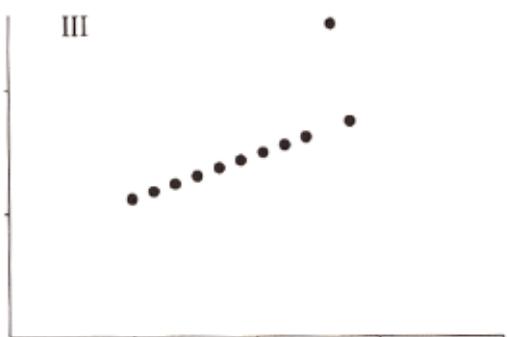
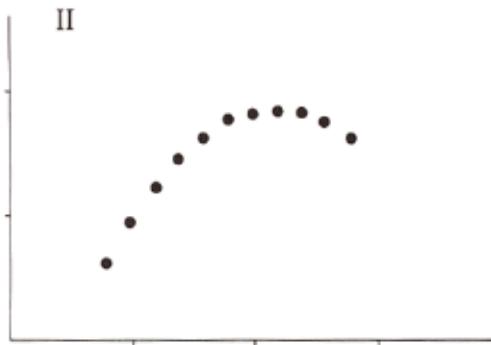
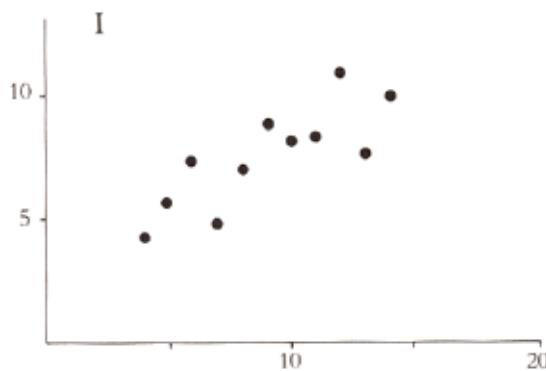
## Information Visualization

- + Intuitive; the user is directly involved in the exploration process.
- + Applicable even
  - If only little is known about the data
  - The exploration goals are vague, or
  - Highly inhomogeneous and noisy data is given.

# VISUALIZATION VS. DATA MINING

## REPORTING ONLY NUMBERS MAY BE MISLEADING

### ANScombe's Quartet



anscombe's quartet

each of the values below is the same for each set

number of points

average x

average y

regression line

standard error of slope

sum of squares

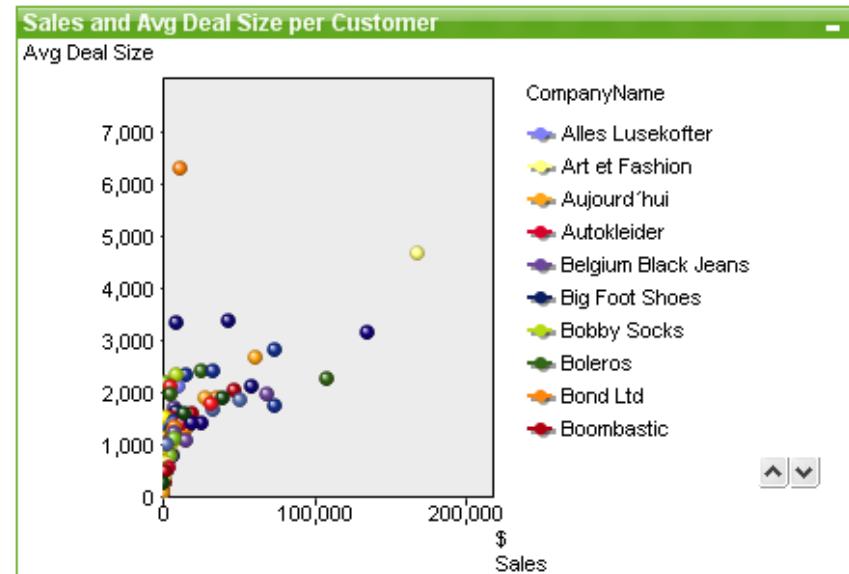
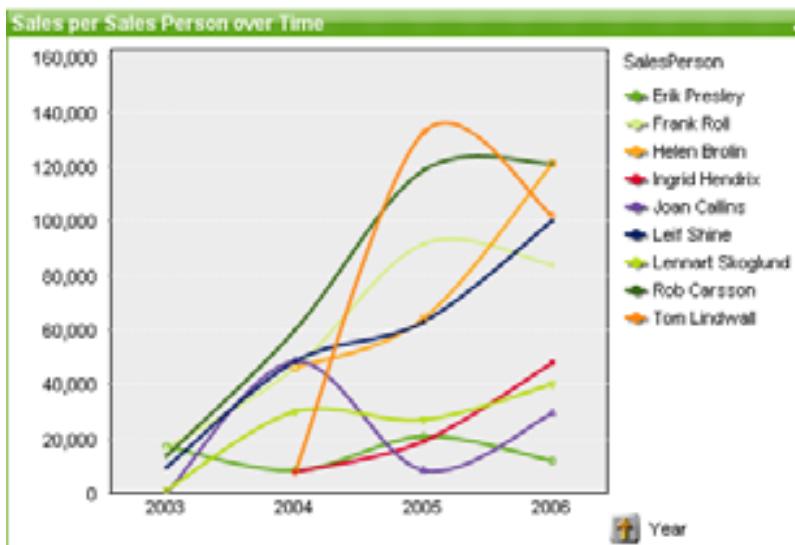
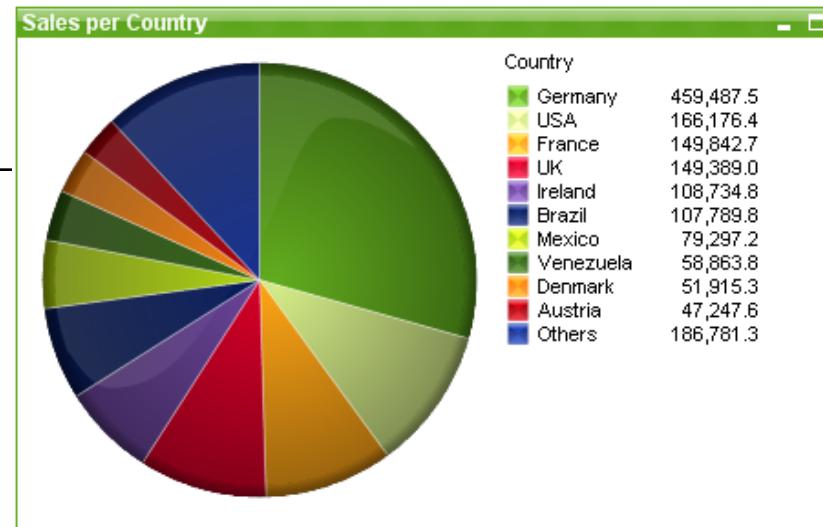
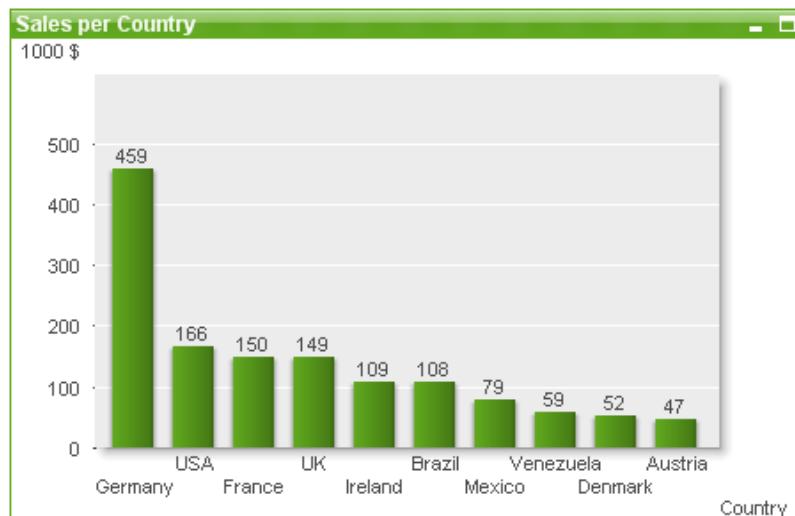
residual sum of squares

correlation coefficient

$r^2$

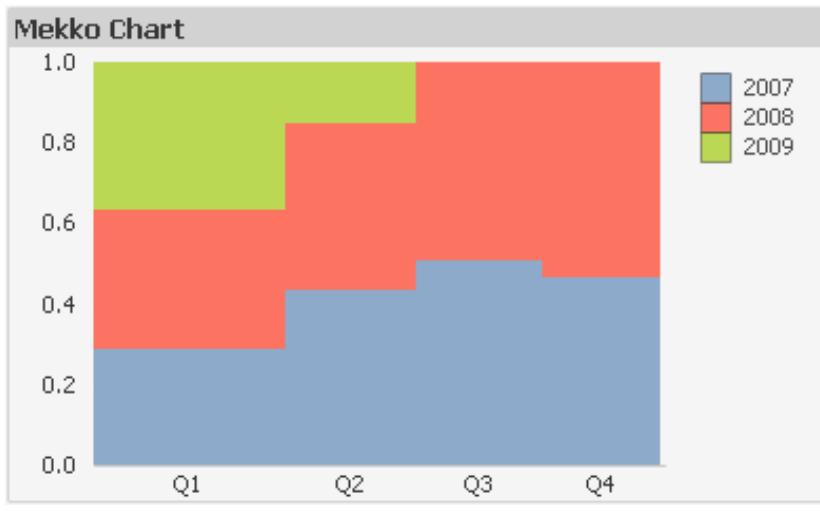
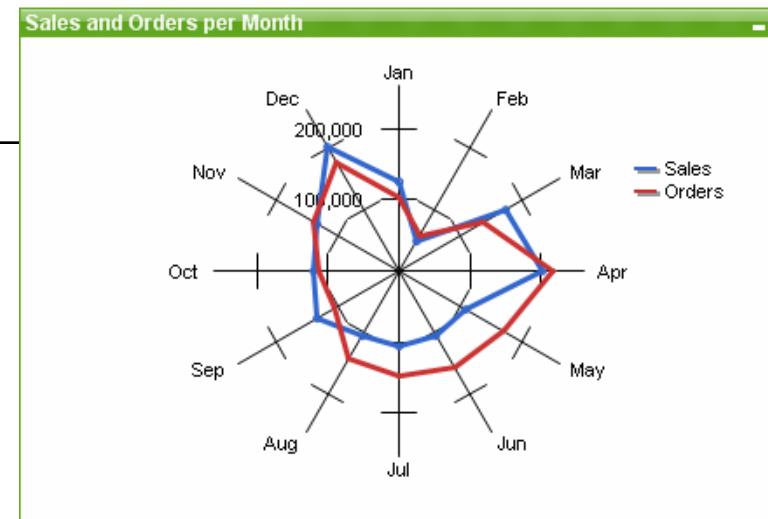
The picture can't be displayed.

# FAMILIAR CHARTS



The picture can't be displayed.

# NOT SO FAMILIAR CHARTS

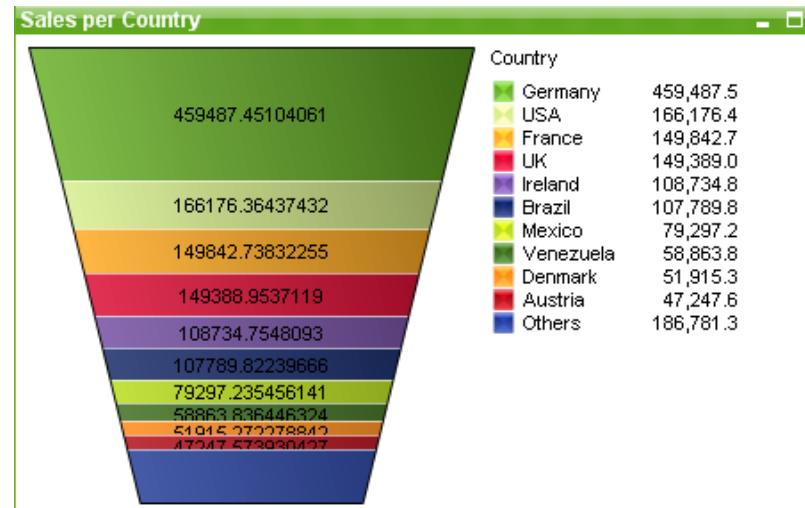
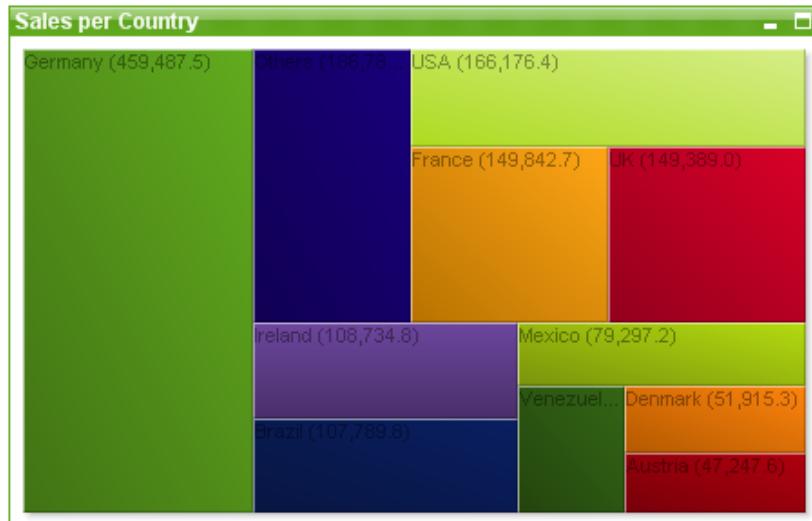
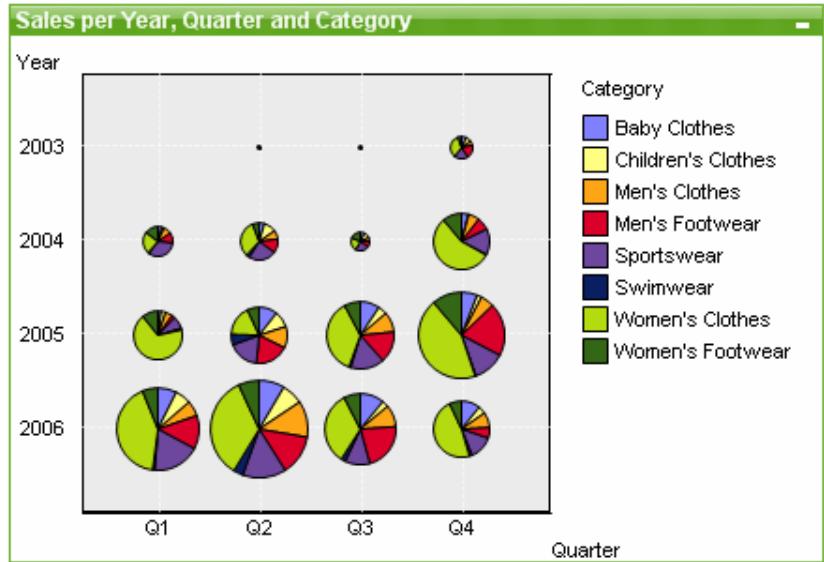


**Sales per Year**

Country	Year	Salesman	Sales
Japan			240,781
U.S.A.			202,455
Bulgaria			116,550
	2004		22,316
	2005		22,316
	2000		2,190
	2001		1,640
	2002	Joe Cheng	19,960
		Sehoon Daw	10,880
		Marcus Sa...	1,250
	2003	Joe Cheng	7,748
		Jerry Tessel	4,149
		Keith Hel...	4,040
		Tony Ced...	3,690

The picture can't be displayed.

# NOT SO FAMILIAR CHARTS



The picture can't be displayed.

# WHAT DO YOU NEED TO KNOW FOR CHOOSING THE RIGHT VISUALIZATION?

---

Characteristics of data

- Types, size, structure

- Semantics, completeness, accuracy

Characteristics of user

- Perceptual and cognitive abilities

- Knowledge of domain, data, tasks, tools

Characteristics of graphical mappings

- What are possibilities

- Which convey data effectively and efficiently

Characteristics of interactions

- Which support the tasks best

- Which are easy to learn, use, remember

# GENERAL VISUALIZATION PRINCIPLES

---

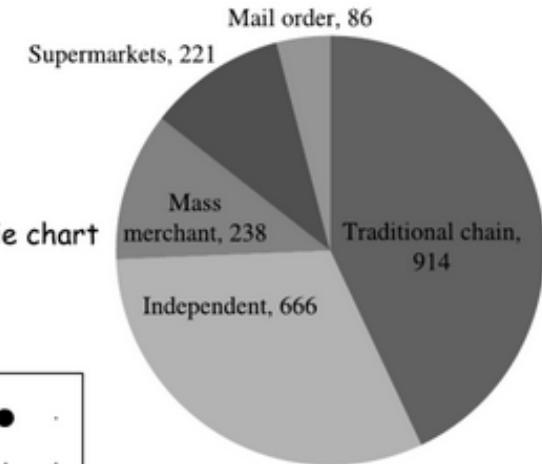
- **Show the data**  
Grids, tick marks, explanatory texts should be avoided
- **Simplify**  
Choose the graphic that most effectively communicates the information

# WHICH IS THE BETTER VISUALIZATION?

Number of drug prescriptions, millions	
Traditional chain	914
Independent	666
Mass merchant	238
Supermarkets	221
Mail order	86

Table

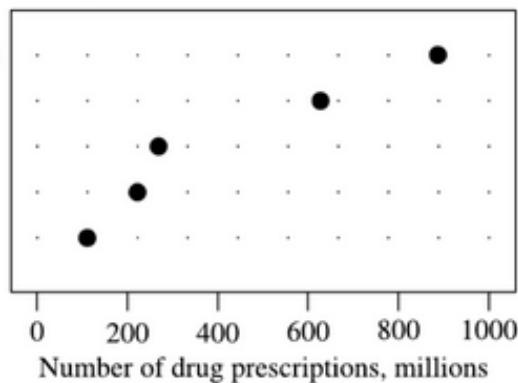
Number of drug prescriptions, millions



Pie chart

Dot plot

Traditional chain  
Independent  
Mass merchant  
Supermarkets  
Mail order

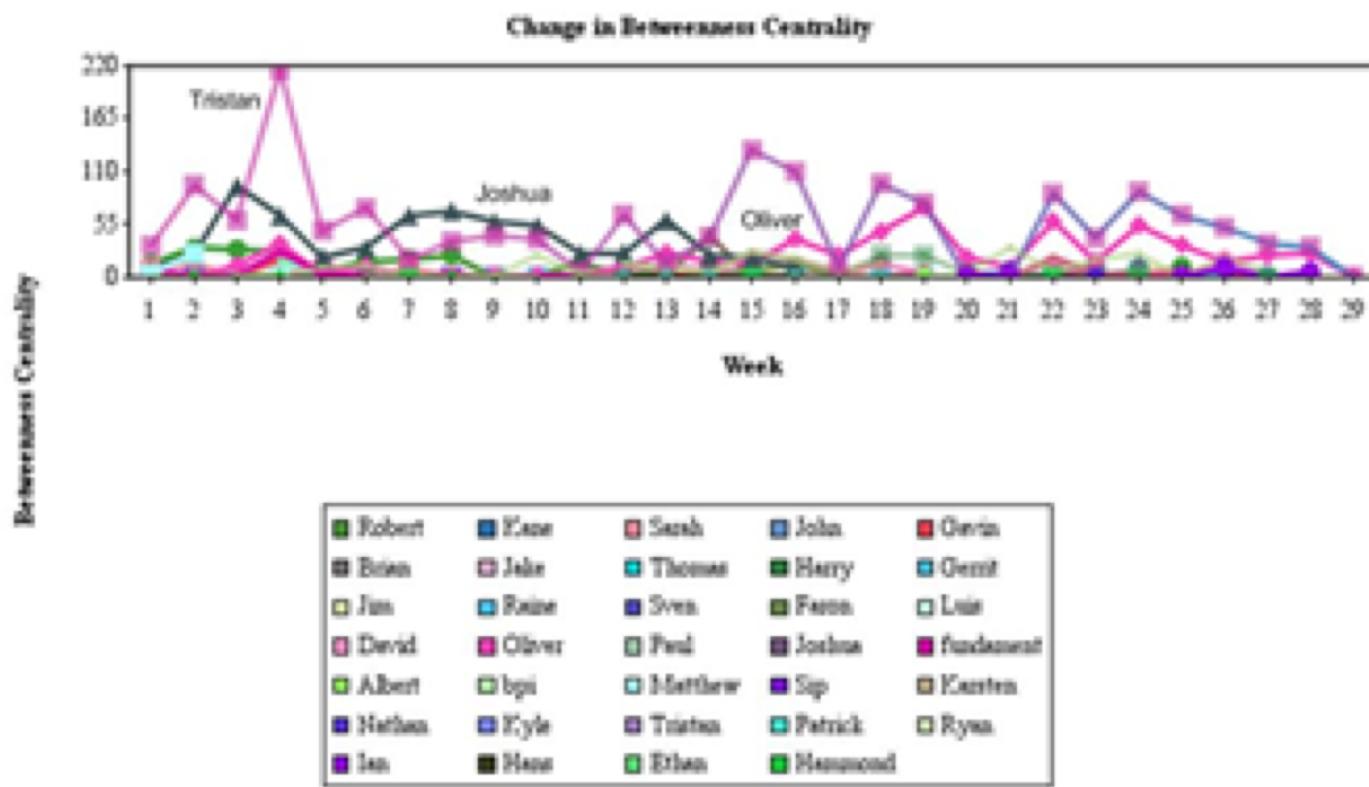


Number of drug prescriptions, millions

The picture can't be displayed.

# GENERAL VISUALIZATION PRINCIPLES

**Reduce clutter:** reduce marks and textual redundancy

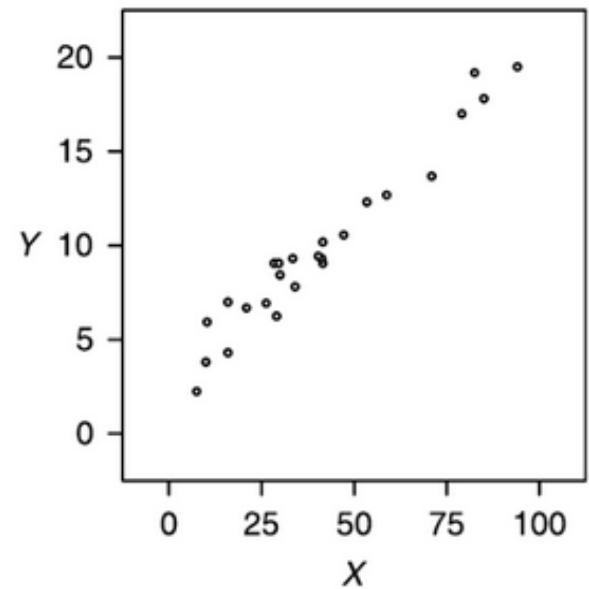
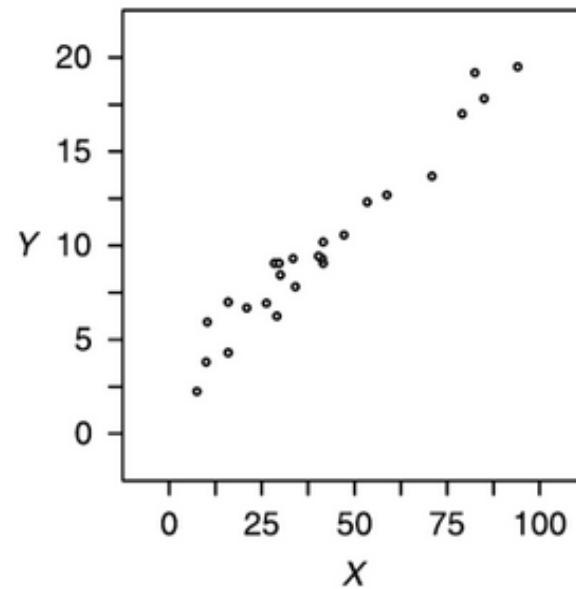
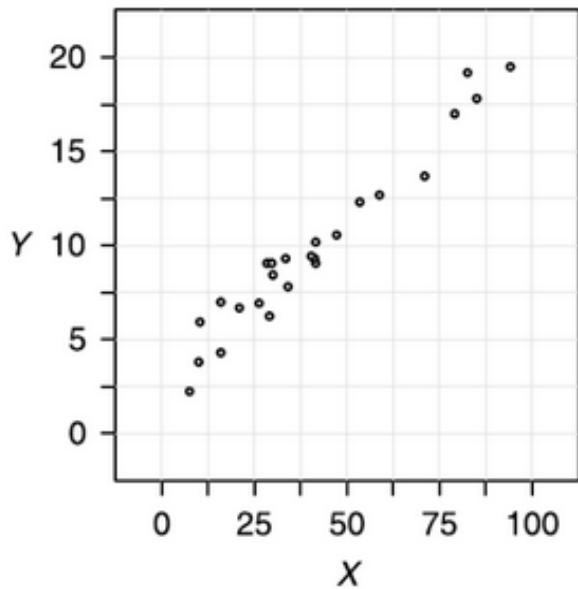


The picture can't be displayed.

# GENERAL VISUALIZATION PRINCIPLES

---

**Revise:** increase amount of ink related to the data

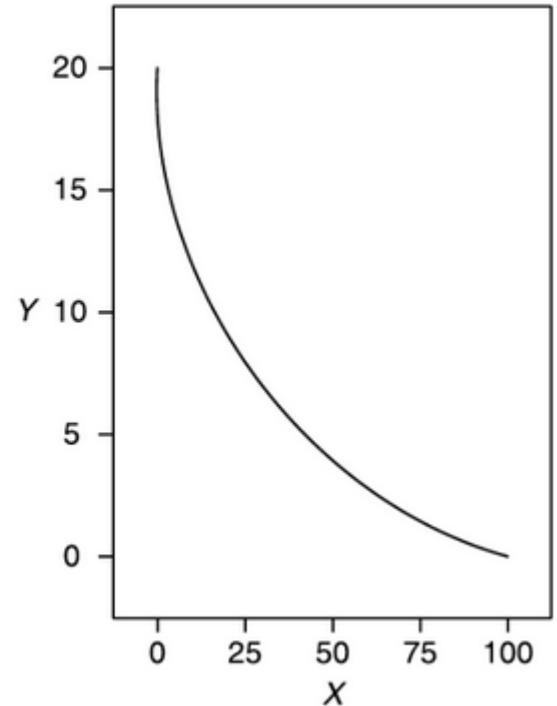
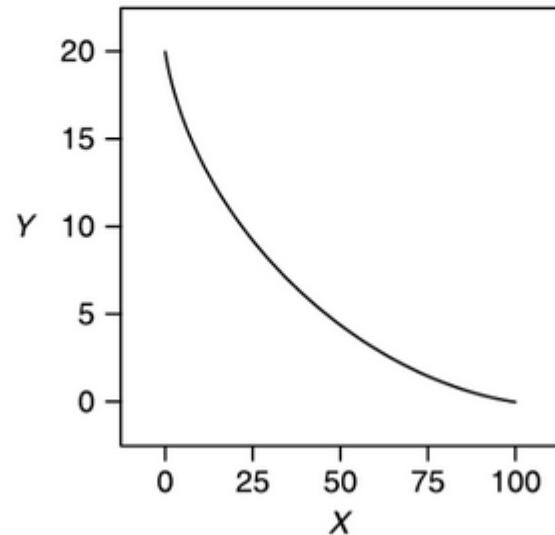
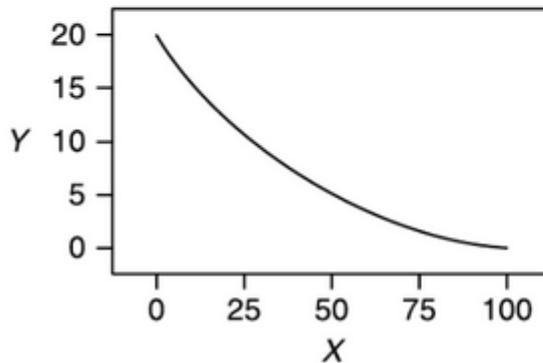


The picture can't be displayed.

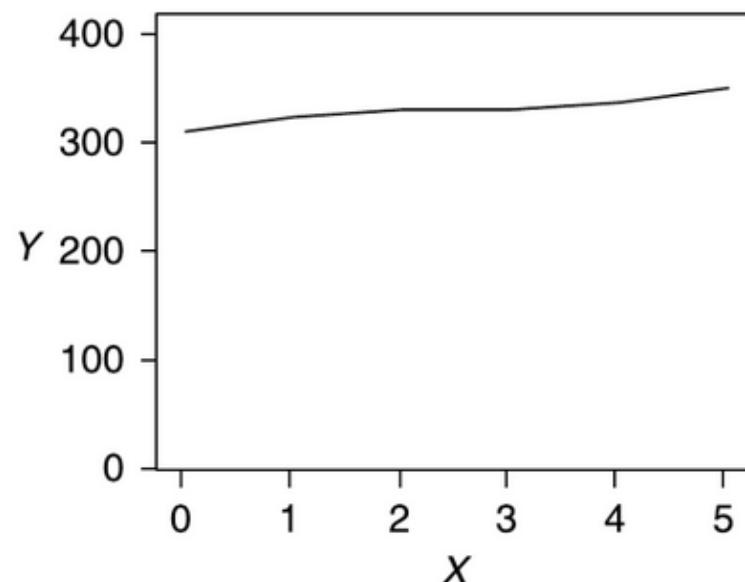
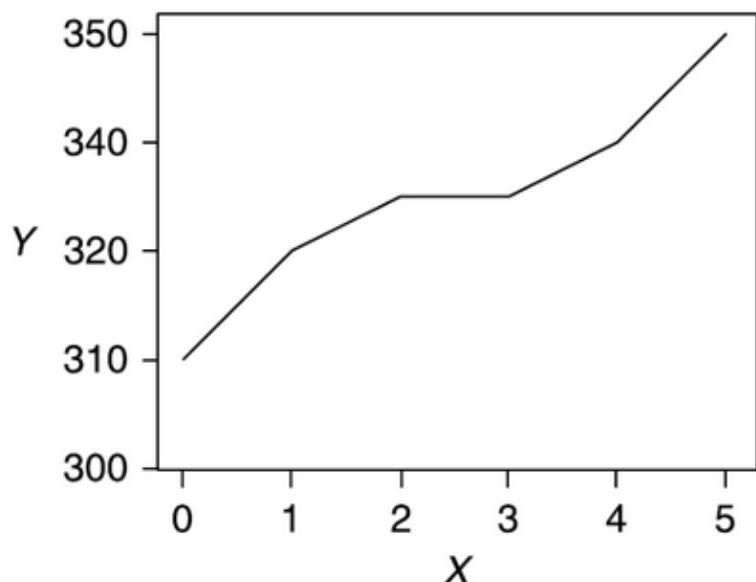
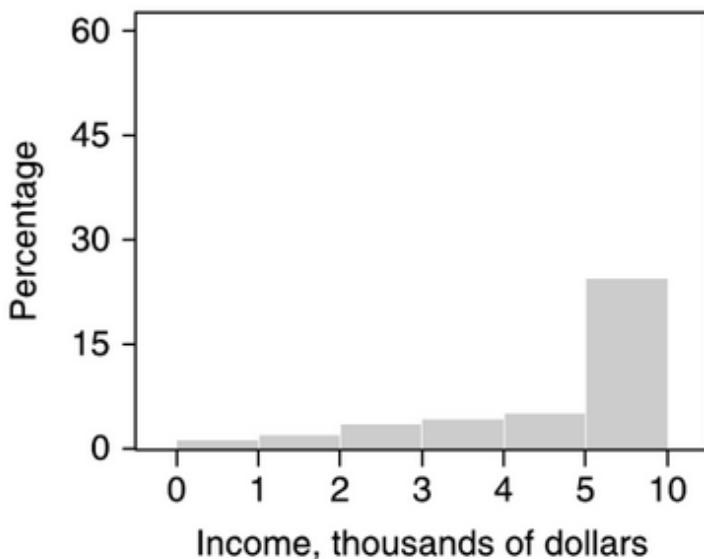
# GENERAL VISUALIZATION PRINCIPLES

---

**Be honest:** w.r.t aspect ratio, scale



The picture can't be displayed.



# Issues Regarding Data

## GENERAL VISUALIZATION PRINCIPLES

---

Know what type of data you are visualizing

- Discrete vs. Continuous
- Ordinal vs. Nominal

*Is there a meaningful ranking of values or not?*

Genre (Action, Fantasy, Romance, ...) : Nominal

Levels (Low, Medium, High) : Ordinal

Low < Medium < High

Why? For example

- If your x-axis is a nominal value, don't draw a line  
... it simply doesn't make any sense

# METHODS TO COPE WITH SCALE

---

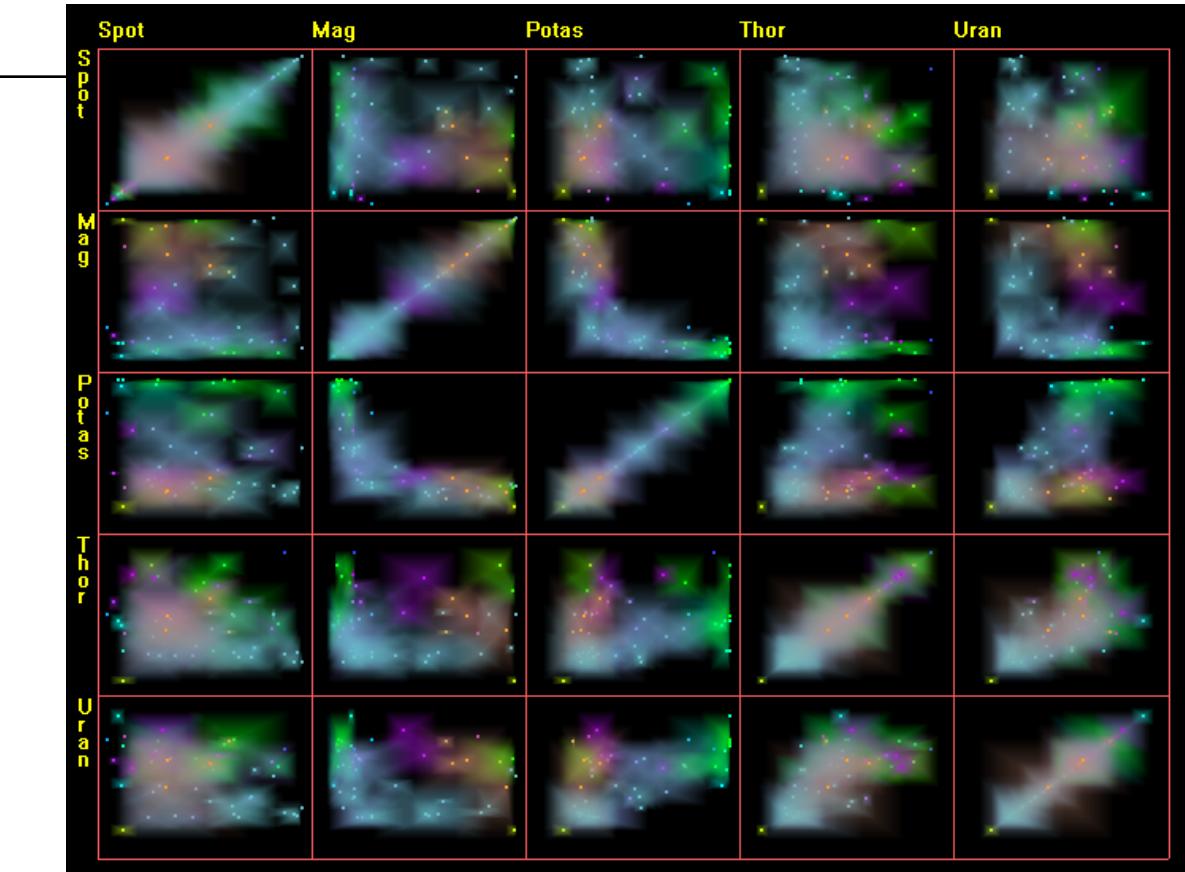
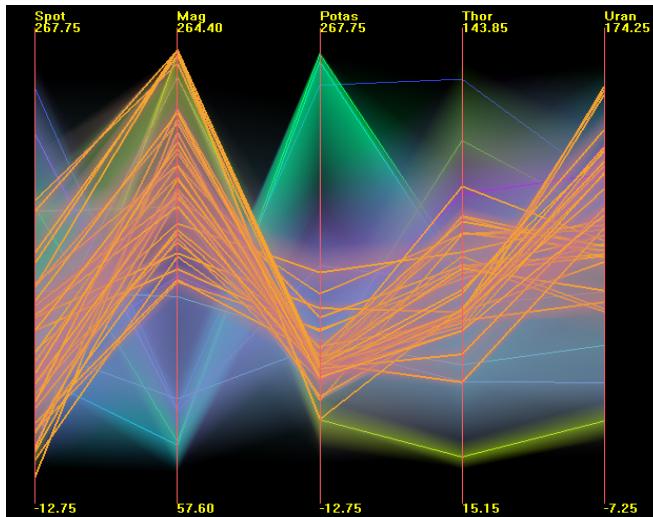
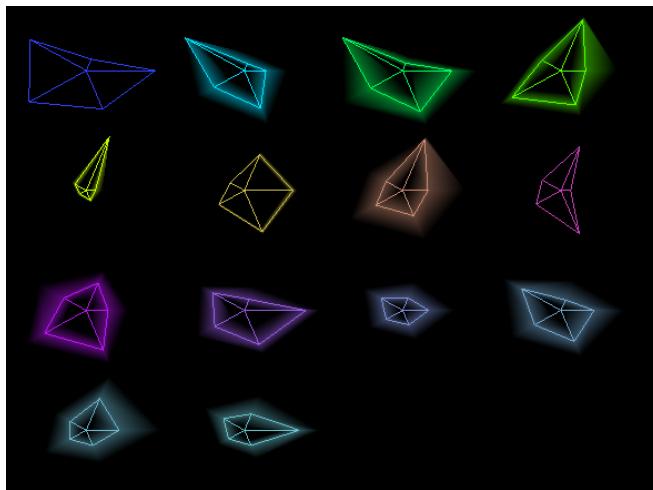
Many modern datasets contain large number of records (millions and billions) and/or dimensions (hundreds and thousands)

Several strategies to handle scale problems

- Sampling
- Filtering
- Clustering/aggregation

Techniques can be automated or user-controlled

# EXAMPLES OF DATA CLUSTERING



The picture can't be displayed.

---

# Don't dismiss tables



# SUMMARY TABLE

**TABLE 2.1 Format for a Summary Table**

Variable $a$	Count	Variable $x$ summary	Variable $y$ summary	...
$a_1$	Count ( $a_1$ )	Statistic ( $x$ ) for group $a_1$	Statistic ( $y$ ) for group $a_1$	...
$a_2$	Count ( $a_2$ )	Statistic ( $x$ ) for group $a_2$	Statistic ( $y$ ) for group $a_2$	...
$a_3$	Count ( $a_3$ )	Statistic ( $x$ ) for group $a_3$	Statistic ( $y$ ) for group $a_3$	...
...	...	...	...	...
$a_n$	Count ( $a_n$ )	Statistic ( $x$ ) for group $a_n$	Statistic ( $y$ ) for group $a_n$	...

**TABLE 2.2 Summary Table Showing Average  $mpg$  for Different Cylinder Vehicles**

Cylinders	Count	Mean, $mpg$
3.0	4	20.55
4.0	199	29.28
5.0	3	27.37
6.0	83	19.97
8.0	103	14.96

 The picture can't be displayed.

# TWO-WAY CONTINGENCY TABLE

TABLE 2.3 Contingency Table Format

		Variable $x$		Totals	
		Value 1	Value 2		
Variable $y$	Value 1	Count <sub>11</sub>	Count <sub>21</sub>	Count <sub>1+</sub>	
	Value 2	Count <sub>12</sub>	Count <sub>22</sub>	Count <sub>2+</sub>	
		Count <sub>+1</sub>	Count <sub>+2</sub>	Total count	

Direct  
visualization  
of 2-dim  
cube

TABLE 2.4 Contingency Table Summarizing Counts of Cars Based on the Number of Cylinders and Ranges of Fuel Efficiency (mpg)

	Cylinders = 3	Cylinders = 4	Cylinders = 5	Cylinders = 6	Cylinders = 8	Totals
mpg (5.0–10.0)	0	0	0	0	1	1
mpg (10.0–15.0)	0	0	0	0	52	52
mpg (15.0–20.0)	2	4	0	47	45	98
mpg (20.0–25.0)	2	39	1	29	4	75
mpg (25.0–30.0)	0	70	1	4	1	76
mpg (30.0–35.0)	0	53	0	2	0	55
mpg (35.0–40.0)	0	25	1	1	0	27
mpg (40.0–45.0)	0	7	0	0	0	7
mpg (45.0–50.0)	0	1	0	0	0	1
<i>Totals</i>	4	199	3	83	103	392

# SUPER TABLE

How different groups voted for president				
	Carter	Reagan	Anderson	Carter–Ford in 1976
Democrats (47%)	66	26	6	77–22
Independents (23%)	30	54	12	43–54
Republicans (11%)	11	84	4	9–90
Liberals (17%)	57	27	11	70–26
Moderates (46%)	42	48	8	51–48
Conservatives (28%)	23	71	4	29–70
Family income				
Less than \$10,000 (13%)	50	41	6	58–40
\$10,000–\$14,999 (14%)	47	42	8	55–43
\$15,000–\$24,999 (30%)	38	53	7	48–50
\$25,000–\$50,000 (32%)	32	58	8	36–62
Over \$50,000 (5%)	25	65	8	—
Professional or manager (40%)	33	56	9	41–57
Clerical, sales or other				
white-collar (11%)	42	48	8	46–53
Blue-collar worker (17%)	48	47	5	57–41
Agriculture (3%)	29	66	3	—
Looking for work (3%)	55	35	7	65–34

Figure 2.18 Portion of supertable showing voter profiles for the 1976 and 1980 U.S. elections

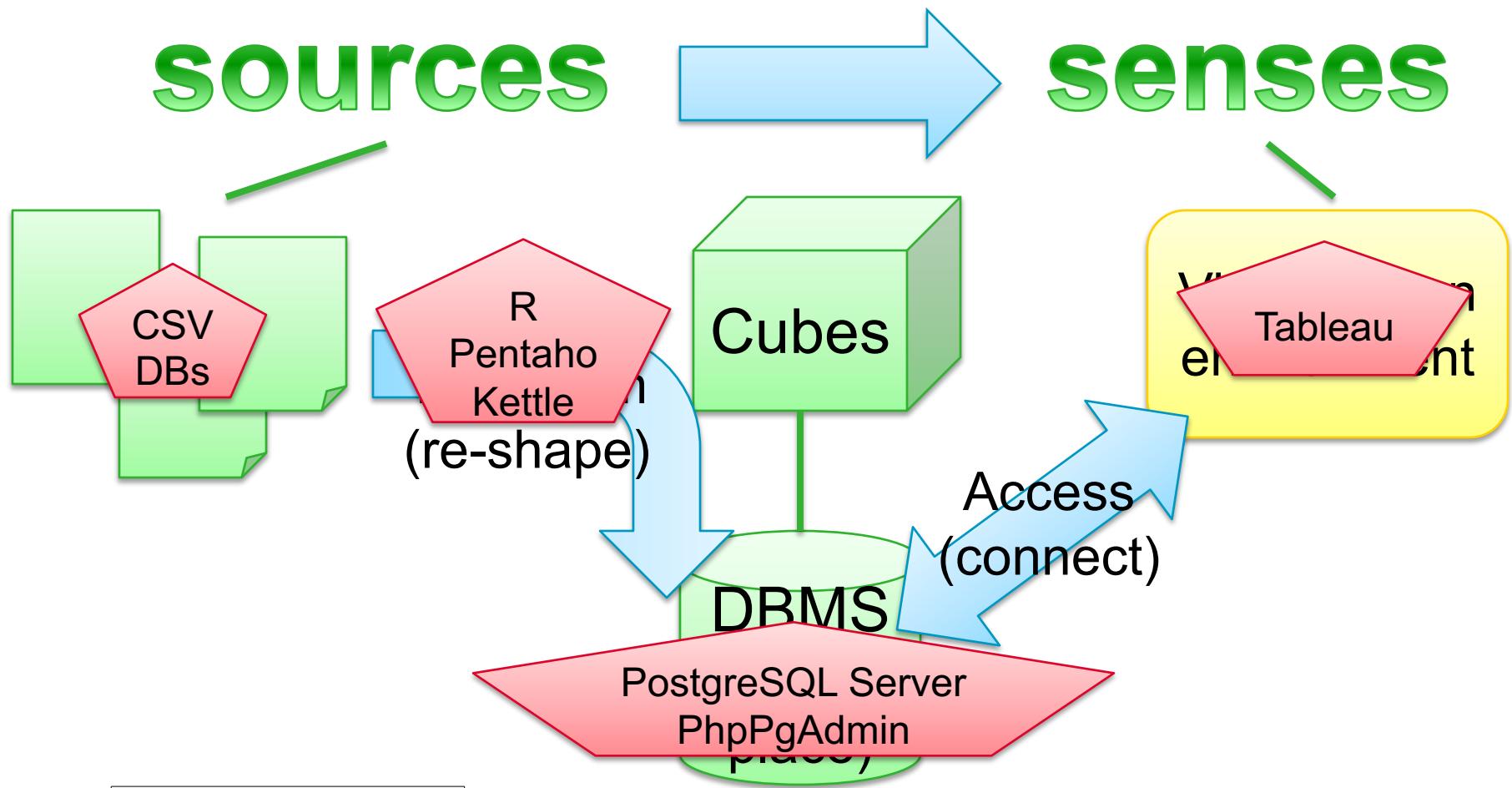
 The picture can't be displayed.

UNIVERSITY OF TWENTE.

# TOOLS AND THEIR ROLES

# TOPIC 3: DATA PREPARATION AND VISUALIZATION

WHICH TOOLS WE ARE USING



# TOOLS AND THEIR ROLES IN THE PROCESS

---

Process for data warehouse design and realization

- Define business questions
- Design star/snowflake schema (based on BQs!)  
(or add facts / dimensions to existing one)  
(or design new one by re-using existing dimensions)
- Transform source data to target schema (e.g., **R**, **Spoon**)

Analytics

- Reports (e.g., Business Objects)
- Visualization (e.g., **Tableau**)
- Data Mining (e.g., **R**, **Python (=programming)** or **Weka (=tool)**)

They all ‘communicate’ through the database (e.g., **PostgreSQL**)

# TOPIC ASSIGNMENTS & NEXT LECTURES

---

## Topic assignments (4)

- 1: on-paper assignment on cube concepts
- 2&3: assignments on data preparation; in two flavours:
  - Tool-based (using Pentaho Kettle / Spoon)
  - Programming-based (using R)
- 4 on-paper assignment on multidimensional modeling

You  
choose

## Two additional lectures

You only need to participate in the one belong to the flavor you chose

- Tool-based: demo of assignment 2 with Kettle / Spoon
- Programming-based: introduction to R

DM also uses R, so if you do DM as well, then choose R-based for DPV