Part I
Business Intelligence (BI), OLAP and the Data Warehouse

David Gross-Amblard
(original slides and documents: Marc Bousse)
ISTIC- Rennes 1 University
V2.1

Summary

- Motivation: why traditional databases are not sufficient for BI
- One approach: OLAP
- Model for OLAP, and implementation
- One example: Business Objects (BO)
- Keyworks: OLAP, ROLAP, MOLAP, key, measure, hierarchy, acyclic graphs, grain, (data)cube, slicing, dicing, roll-up, drill down, ...

Program

- 3 readings (CM): live for now
- 1 exercise session (TD)
- 3 labs (TP) on Business Object and/or PowerBI

Outline

Chapter 1: " *why* " business intelligence (BI)

Chapter 2: "what": multidimensional modeling

Chapter 3: "how": architecture of a BI system

Use case: BI for a driving school

Bibliography

- Building the Data Warehouse. Bill Inmon. Wiley, 1992.
- The Data Warehouse Toolkit. Ralph Kimball. Wiley, 1996
  (see https://www.kimballgroup.com)
- Database systems, The Complete Book (2nd edition). Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Pearson International Edition, 2009 (chap. 10)

- In French:
  - The decision support project. Stakes, models, architectures of the Data Warehouse. Jean-Marie Gouarné. Eyrolles, 1997 La construction du datawarehouse. Jean-François Goglin. Hermes, 1998.

Chapter 1
Business intelligence

- Chapter Outline
  - Why business intelligence
  - Why more than a DBMS
  - Historical development
  - Keywords: *Business Intelligence*, OLAP, multidimensional approaches, *Data Warehouse*, Data *Mart*

Business Intelligence

- Intelligence as in " Central Intelligence Agency " (not as intelligence / wit, in French)
- Data analysis of business information
  - Data management as a basis
  - Reporting
  - OLAP
  - Data sciences
  - Data mining
  - From data to processes (workflows)

Business Intelligence: motivation

- Before 80's
  - Data management dedicated to production monitoring
    - Event triggering (launch machines)
    - Amount of production
- After 80's
  - Any reasonable company has a proper production system
  - Remaining advantage: strategic insights
  - Requires precise information on the company (past) and its environment (customers, competitors): intelligence

Information for right decisions

- Good information is required:
  - *Precise, but not exhaustive*
  - *Correct, but accuracy is not mandatory*
- Information is not data
- Example
  - *A company want to understand its internal travelling costs*
  - *Data: all details about plane tickets, hostel bookings, buses: precise, exhaustive, accurrate, but useless*
  - *information: evolution of the average of travel costs, per month and company departments: precise, correct, not exhaustive, useful*
  - "A valid data in not necessarily a useful information

Charactize Useful Information

- Quoting Bill Inmon:
  - Has a meaning for the decision maker
  - Is thematic (related to a precise topic)
  - Is integrated (same format whatever the source)
  - Is curated (filtered, cleaned)
  - Is persistent (do not forget the past)
  - Has a precise timestamp
  - Is accessible to the decision maker (GUI)

Bill Inmon

- Coined the notion of a data warehouse (*entrepôt de données* in French)
- Book: Building the Data Warehouse. Bill Inmon (1992). Wiley.

1991, © The ComputerWorld

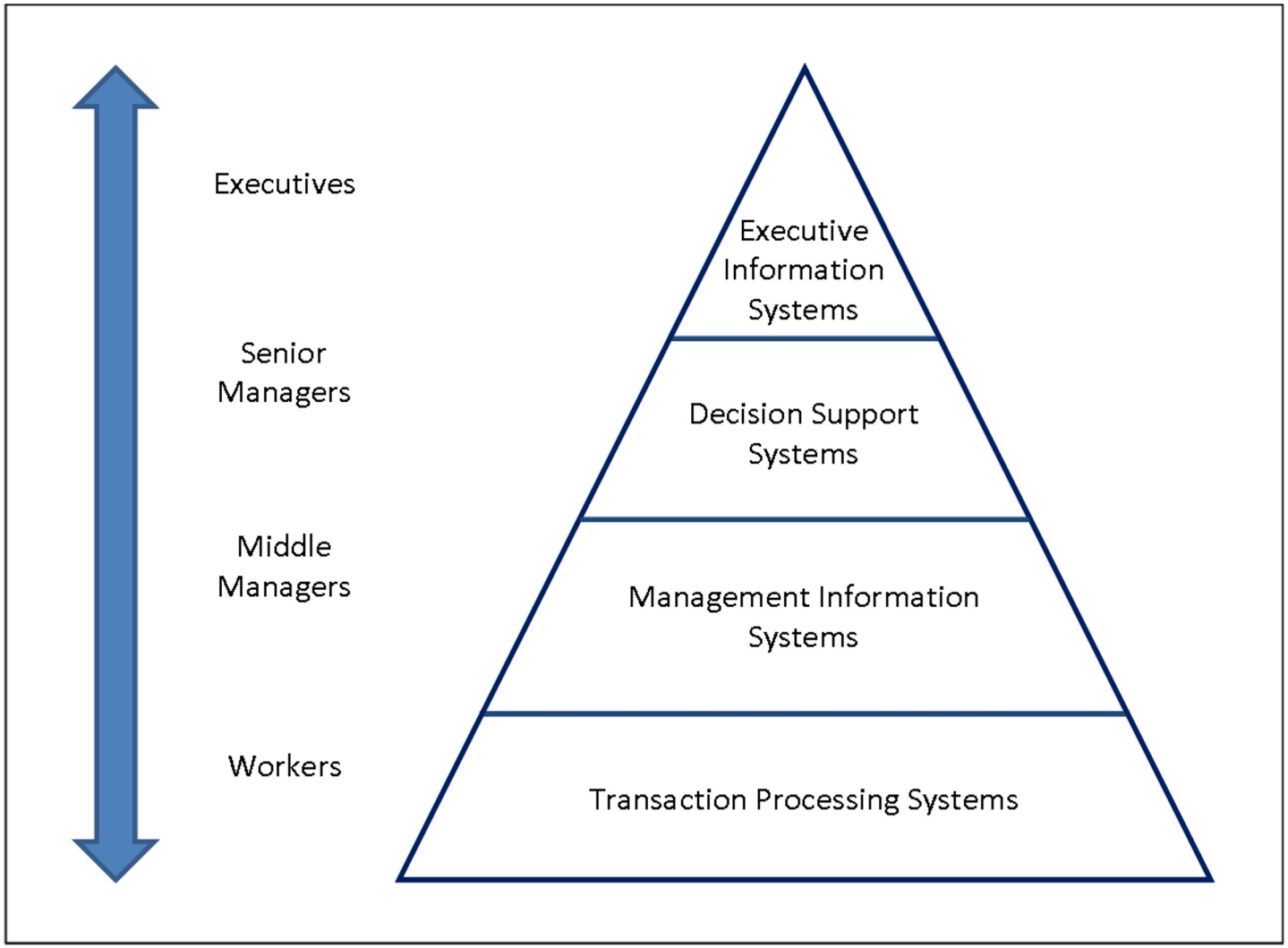
Core Data Management Systems: pros

- Very efficient for operational aspects
  - OLTP (on-line transactional processing)
  - Fine data modelling
  - Transaction management
  - Rich query language
  - Optimization

Core Data Management systems: cons

- Specialized systems, with different versions on different sites (ex. Mysql on one, Postgres on another)
- Not dedicated to Decisional Information
- Technical difficulties
  - Heterogeneous equipment
  - Technical interfaces
  - Complex database schemas
  - No data versioning / timestamping by default
- Conceptual difficulties (intelligence)
  - Dedicated for developers, not decision makers

Better: Layers of Information Systems



(source Wikipedia)

Goals of the decision information / support system

- Provide useful information, without technical details
- Information definition and modeling for the final user (decision maker)
- Persistent information storage (no deletion)
  - Measure of activity
    In a precise context
- Persistent information storage (no deletion)
- Data presentation for any combination of context
- Natural user interface

Example

- A quality manager wants to understand the amount of defective part in a product for the current year
  - Measure: number of defective parts
  - Context: month, day, factory, ..
  - Combination of context:
    - average number of defective parts for a specific factory, for January
  - Navigational interface
    - Change of factory, zoom out on full year, ...

Decisional information systems (DIS)

- **(Decision support systems, DSS)**
- "Decisional information systems are data management systems dedicated to the monitoring of (business) activities, to assist decision making. They provide a synthetic view of operational information, and use specific data modeling and storage methods (data warehouse, OLAP databases). The offer a global view of an activity (company), integrating all its dimensions.

The construction of the datawarehouse. Jean-François Goglin (1998). Hermes

DIS evolution

- 70-80: reporting systems
  - Direct querying of the operational database
  - No autonomy
- 80's : datacenters
  - Extraction of data from several operational databases
  - Centralization in one unique infrastructure
  - Dediated database
  - Data gathering, but no integration
  - First common referential
  - Better user interfaces
- 90's: OLAP model and databases
- 2000's: Cloud version (SAAS, DAAS)
- 2010's: From data analytics to machine learning

  2020: AI

OLAP

- Very common building block
- = On-Line *Analytical* Processing (Codd , 1995)
  (≠ OLTP = On-Line *Transactional* Processing)
- DIS based on a *Data Warehouse*
- *(Bill Inmon): "A data warehouse is a collection of information which is thematic, integrated, persistent and timestamped, organized to assist decision making".*
- Production or operational database: set of sources that will feed the data warehouse

Data Warehouse

- 4-infrastructure phases
  - **Data collect** (from operational databases)
  - **Data integration** (in a single database)
  - **Data diffusion** (according to a multidimensional model)
  - **Data presentation** (for final users)

Usage

- **Reporting:** production, diffusion and customization of static decisional information
- **Dynamic analysis (OLAP / *analytical processing / analytics*):** interactive multidimensional exploration of information
- **Data Sciences:** finding interesting patterns, laws, in information
  - Numerically, symbolically
  - Typical example: correlation between baby diaper sales and beer on football events

Ralph Kimball

- Methodology for multidimensional information modeliling
- Co-founder of RedBricks systems (DBMS optimized for multidimensional querying) / Informix / IBM
- Classic book: The Data Warehouse Toolkit. Ralph Kimball (1996). Wiley.

Chapter 2
The multidimensional model

- At the basis of OLAP DIS:
  - Storage model
  - Information presentation model
  - For the decision maker
- Principle: store and query information as **measures** in a given **context**
- **Three characteristics**:
  - measures **indexed by keys (=/= primary keys)**
  - Indexing with several keys: **dimensions**
  - **Measure aggregation (sum, mean, ...)**
- **Multidimensional modeling:**
  - **Find the measures, find the keys, find the dimensions**

What is a measure

- We want to measure activity (sales, production, ...)
  Measure : numerical value
- Example for a driving school:
  - Number of lessons = 750
  - Number of teaching hours = 30 558
  - Number of driving test attempts = 750
  - Success rate= 56%.
- Problem: no context for these measures (when, where, who, ...)

Indexing measures: keys

- Example: adding the year information
  - **4748 teaching hours in year 2020**
  - **The measure value** 4748 *is now indexed by the key value* 2020
  - Globally: the Teaching hours **mesure** is indexed by the Year **key**

Year	Teaching hours
2009	3966
2010	4748
2011	18379

Definitions

- **Key:** non-empty set of values called members or elements
- (not exactly the keys of the relational model)
- **Measure** (synonym: **fact**)
  = map a number to each key value
- Keys index measures: each key value sets one and only one measure value
- This mapping is a function
  - Teaching hours is a function of the Year
  - Teaching hours = f (Year)

Indexing with several keys

- Several keys can be mandatory to precisely index a measure
- Examples:
  - Teaching hours = f (Year, Monitor)

	2009	2010	2011	2012
Sophie	123	543	604	112
Julia	429	765	352	222
Tsering	286	221	642	643
Georg	523	33	112	232
Anette	123	965	1650	0

Measure arity

- Arity: nb of required keys
  - Binary measure: Teaching hours = f(Year, Monitor)
  - Ternary-measure:
    Teaching hours=f(Year,Monitor,Car)
  - N-ary measure: n keys
- In practice
  - A key may correspond to a primary key in a database
  - Or can be constructed from these existing keys (concatenation,...)
  - Or can be forged for convenience

Dependencies between keys

- Keys may have relationships

Monitor	Monitor Dept.	Year	Hours
Sophia	35	2010	123
Sophia	35	2011	442
John	68	2010	445
John	68	2011	762

- Keys (values) may depend on each other
- Clearly, Dept is defined by the Monitor (the Monitor lives in one unique department)
- Notation: Monitor ➔ Monitor Dept
- ➔ is the functional dependency bw. keys

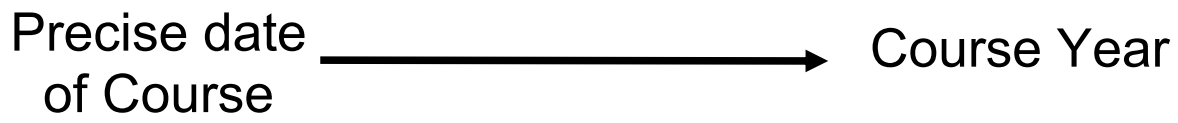
Key independance rule

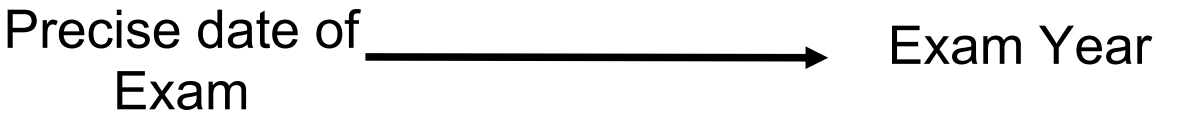
- Rule: keys indexing a measure must be functionnaly independent

- Values of Monitor are the most discriminant
- Values of (Monitor,Monitor Dept) are more informative
- But what to do with dependent keys ?

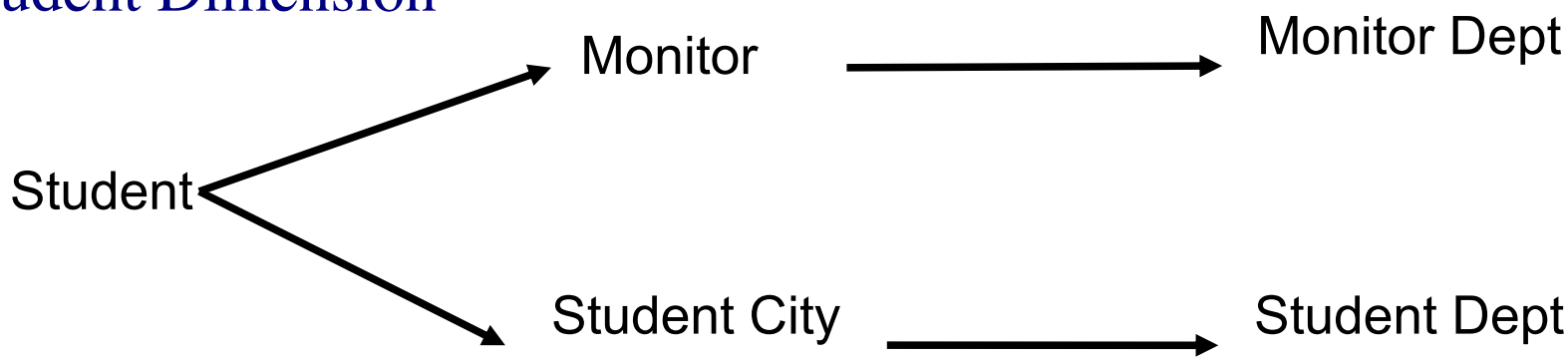
Notion of Dimension

- A set of keys that are functionally dependent form a dependency graph called a dimension
  - Two temporal dimensions for Course and Exam





  - "Student Dimension



- Rule: no dependency between two dimensions !
- (maximally connected components)

Properties of dimensions

- The graph of a dimension is acyclic
- Grain size:
  - Unique source of the graph
  - Also called atomic key or atomic level
    - Grain of the Course dimension: Precise date of course
    - Grain of the Student dimension: Student (key)
  - Each branch of the dimension is called a *hierarchy*
  - A unique hierarchy in the Course dimension

    Precise date of Course → Course Year
  - Two hierachies in the Student dimension

    Student ➔ Monitor ➔ Monitor Dept
  - Student ➔ student city➔ Student Dept

Key aggregation

- Any key of a hierachy represents (aggregates) any key of its underlying level
  - The dependency Monitor ➔ Monitor Dept allows to aggregate Monitors by their departements (as each monitor has a precise Dept)
  - The dependency Student ➔ Monitor allows to aggregate Students by their Monitors

Moniteur	**Dépt.**
Maillet Sophie	35
Meursault Antoine	35
Meyer Julie	22
Moreau François	53
Moreau Julie	35
Morel Gérard	35

Elève	**Moniteur**
Eberhard Adeline	Meursault Antoine
Eveillard Eric	Meursault Antoine
Eveillard Léon	Meursault Antoine
Eluard Sophie	Moreau François
Eugène Jacques	Moreau François
Eveillard Paulette	Moreau François

Key aggregation

- Aggregation is transitive:
  - Student ➔ Monitor Dept
  - But also: Student ➔ Student Dept
  - To avoid confusion, we used two different notions (keys) for Dept (even if departments are the same)

- Complete aggregation: we consider a default key named " all X ", that allows to aggregate all keys of a dimension:

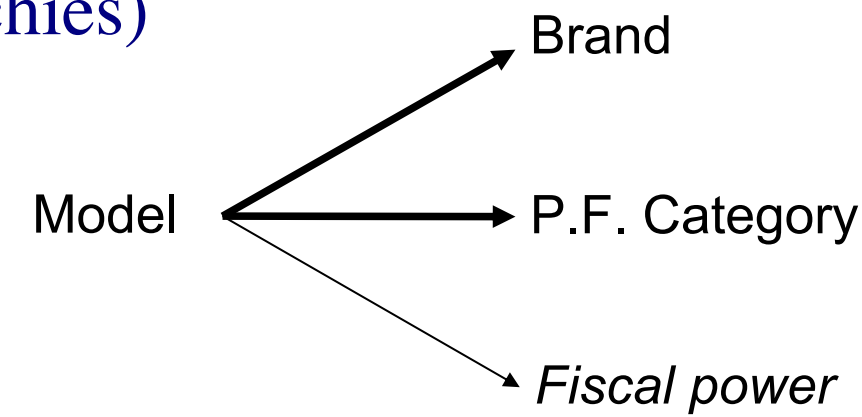
  Student ➔ Monitor ➔ Monitor Dept ➔ All Students
- Student ➔ Student area ➔ Student Dept ➔ All Students
- Note: Some functional dependencies are useless
  - Ideed, for cars: Serial Number ➔ Initial traffic realease
  - But do you really want to aggregate cars by their initial traffic release...?

Information Attributes

- keys = discriminating information
- **Information attributes**
  = secondary information
  - each depend on a key
  - do not allow for significant aggregation (thus absent from hierarchies)
  - provide additional information
- Examples:
  - fiscal power of a car model
  - date of birth of a student
  - name of the department in clear (already identified by its number)

Interval type key

- How do you use information such as *fiscal power* in a hierarchy?
- Solution: define a new key whose values are *intervals*
- Example: *P.F. category* key. category key, with the values: 1-5 CV; 6-8 CV; 9-11 CV; etc.
- Resulting graph :
  (2 hierarchies)



Dimensions and measurements

- The aggregation of the keys multiplies the possibilities of indexing the measures:
  - no. of hours = f(course week, student department, model)
  - no. of hours = f(current year, community, P.F. category)
  - ... more than 250 possibilities in total for this measure!
- Let's generalize this notion: a measure is a function of N dimensions composed of hierarchical keys
  - dimensions of the driving school case: date of course, time of course, vehicle, test date, inspector, vehicle model
  - measures :

    pass rate = f(exam date, student, vehicle model, inspector)
    no. of hours = f(course date, student, vehicle, time)
    no. of classes = f(class date, student, vehicle, time)

Aggregation of measurements

- The aggregation of the *keys* allows the aggregation of the *measures*:

  (1) number of hours = f(course year, monitor)

  (2) number of hours = f(course year, monitor dept.)

		1999	2000	2001	2002
22	**Meyer Julie**	1112 h	990 h	5227 h	1193 h
	Somme par dépt :	*1112 h*	*990 h*	*5227 h*	*1193 h*
35	**Maillet Sophie**	1433 h	1797 h	7270 h	1112 h
	Meursault Antoine	429 h	388 h	1450 h	381 h
	Moreau Julie	183 h	74 h	626 h	255 h
	Morel Gérard	523 h	836 h	1991 h	309 h
	Somme par dépt :	*2568 h*	*3095 h*	*11337 h*	*2057 h*
53	**Moreau François**	286 h	663 h	1815 h	245 h
	Somme par dépt :	*286 h*	*663 h*	*1815 h*	*245 h*

(2)

(1)

We obtain the values of (2) by aggregating the values of (1) according to the aggregation of the monitors by department

Aggregation of measures (continued)

- To "neutralize" a dimension,
  we aggregate on the key " ensemble " :
  (3) number of hours = f(course year, instructor)
  (4) number of hours = f(course year, student group)

	1999	2000	2001	2002
Maillet Sophie	1433 h	1797 h	7270 h	1112 h
Meursault Antoine	429 h	388 h	1450 h	381 h
Meyer Julie	1112 h	990 h	5227 h	1193 h
Moreau François	286 h	663 h	1815 h	245 h
Moreau Julie	183 h	74 h	626 h	255 h
Morel Gérard	523 h	836 h	1991 h	309 h
Somme:	3966 h	4748 h	18379 h	3495 h

(4)

(3)

We obtain the values of (4) are obtained by adding the values of (3) over each year

Notion of an OLAP query

- We call a **query** the application of a measure to a set of keys:
  - one and only one key per measurement dimension.
    Example: *Number of hours per year, student and vehicle*
    = no. of hours (year, student name+first name, vehicle)
  - a ALL key is used to "neutralize" a dimension. Example:
    no. *of hours per year and student, whatever the vehicule*
    = no. of hours (year, student name+first name, ALL vehicle)
    ...or simply :
    = nb. of hours (year, student name+first name)
  - a query is said to be *atomic* if each dimension is represented by its unique atomic key. Example:
    nb. of hours (week+year, student name+first name, immatr.)

Query computation

- **Case 1**: **Atomic query**. The corresponding measur values must be stored in memory
  - example :
    no. of hours (week+year, student name+first name, immatr.)
  - these measurement values are called *atomic*
- **Case 2**: **non-atomic query**
  Each non-atomic key value defines a subset of atomic keys
  Example: nb. of hours (2001, Eberhard Josette, 987ADD35)
  is the sum of the values :
  nb. of hours (week+year, Eberhard Josette, 987ADD35)
  ... of the weeks belonging to the year 2001

Aggregation function

- We have identified the atomic values to be aggregated
  It remains to define an *aggregation function* to obtain the non-atomic values.
- Most used aggregation functions:
  - **sum**
    Back to the previous example :
    no. of hours (2001, Eberhard Josette, 987ADD35)
    = Σ (no. of hours (week+year, Eberhard Josette, 987ADD35))
    ... for any week+year value belonging to 2001
  - **average**
  - **median**
  - **minimum/maximum**

Measure consistency

- Measure consistency: meaningful computation for any query
  - Examples of ***consistent*** measures:
    - Aggregated sales turnover (CA in FR) = sum of the turnover of the elementary tuples of the aggregate
      (*month with days, country with customers, etc.)*
    - minimum temperature of the region =
      minimum of the minimum temperatures of the areas in the region
  - Examples of ***inconsistent*** measures:
  - national unemployment rate = average of regional unemployment rates
  - => inconsistent, must be weighted by population
  - sum of the total number of workers over N years *(non-additivity)*
  - => inconsistent: must focus on newcomers each year

Computed measures

- Problem: the success rate is in fact the ratio of the number of successes to the number of attempts
- We must therefore define a new measure of *success (which is not available in the DW):*
  - same keys as the *success rate* and *number of attempts* measures
  - same aggregation function as *nb. attempts*: sum
- Definition of *success rate*: ratio of the *number of successes* to the *number of attempts*, for the same set of keys
- This measure is therefore **computed** from other measure, hence a *computation function*

Computed measures and aggregation

- Four possible situations for a measure:

	atomic	**non-atomic**
primary	**1.** memory	**2.** aggregation
computed	**3.** computation	**4**. aggregation then computation

- type 1 = value stored in memory
- type 2 = aggregation function (type 1 measure)
- type 3 = computation function (type 1 and 3 measurements)
- type 4 = aggregation then computation function (type 2 and 4 measures)

Computations and aggregations (continued)

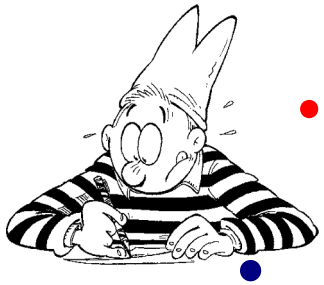
- Back to the *success rate example*:
  - type 1 = *number of successes* (or *number of attempts*) applied to the atomic keys of each dimension
  - type 2 = *number of successes* (or *number of attempts*) aggregated by inspector, year, student, monitor
  - type 3 = *number of successes* (type 1) / *number of attempts* (type 1)
  - type 4 = *number of successes* (type 2) / *number of attempts* (type 2)
- Order of priority for type 4 : Aggregation *then* computation

Context (cube)

- Consider measures indexed with the same dimensions:
  - No. of hours and No. of lessons are indexed by the dimensions (course date, student, vehicle, time)
  - number of attempts, success rate and average score are indexed by dimensions (exam date, student, vehicle model, inspector)
- A set of measures that share the same keys (and therefore dimensions) is called a *context*
  - **cube** context: three dimensions
  - **hypercube** context: more than 3 dimensions (4 to 10 typically)

Modeling with a CDM

- A multidimensional model can be represented by a CDM (Conceptual Data Model, FR- MCD)
  - key = *identifying property of an entity* of the CDM
    - several identifying properties if the key is composed
    - information attribute = *non-identifying property* of the attachment key entity
    - hierarchical link A → B between keys = *binary association* linking the entities concerned **A :*(1,1)-(0,n)*:B**
  - context = CDM *association*
    - number of dimensions = number of links in the association
    - target of each link = grain (atomic key) of one dimension
    - cardinality of each link = *0,n*
  - measure = *property* of such an association



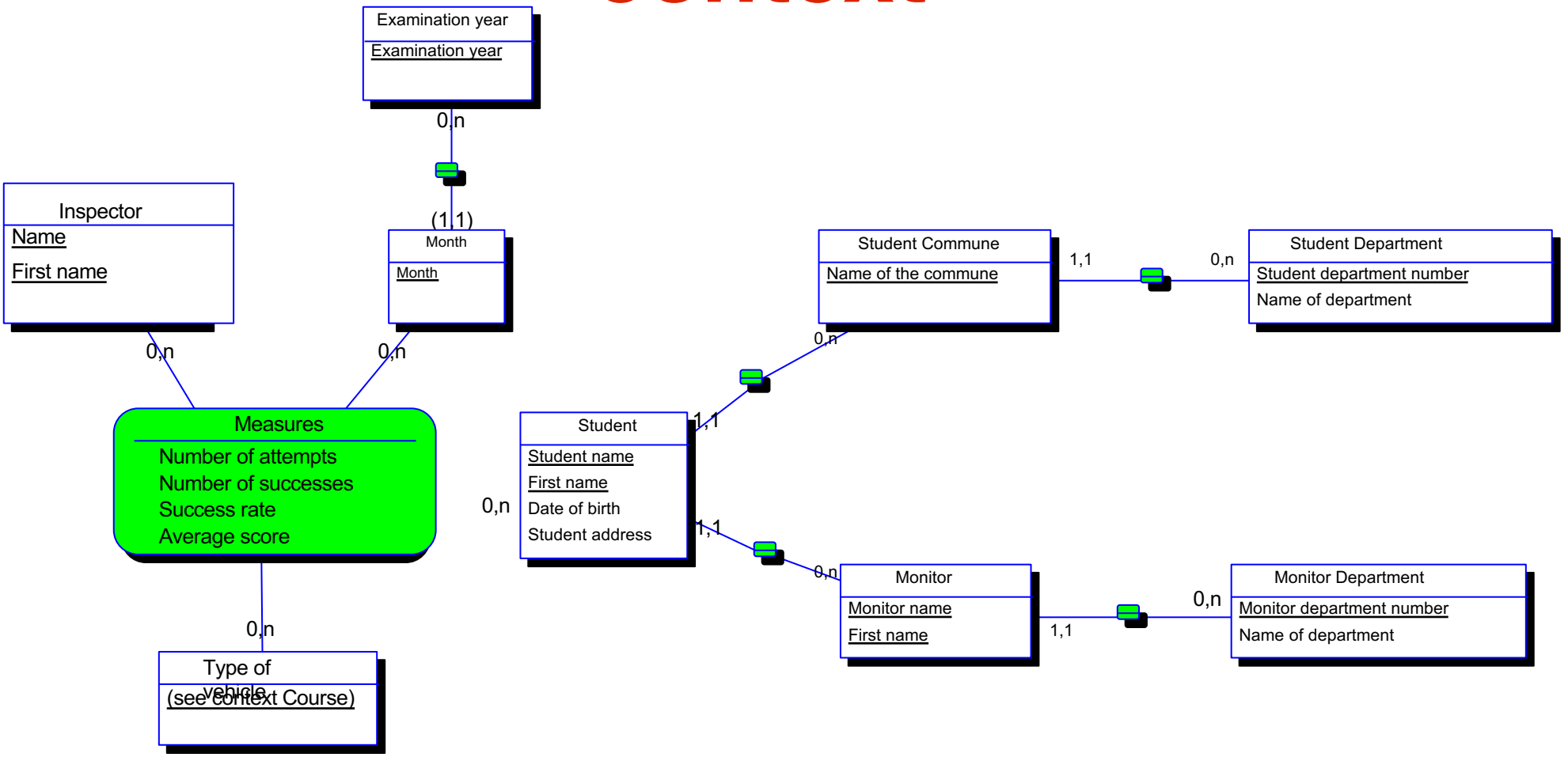
CDM of the driving school case

- *Student* dimension
  - Two hierarchies of five keys = five entities
    - name+first name student ➔ name+first name instructor ➔ Dept.
      = entities *Student, Instructor* and *Instructor Dept.*
    - student surname+first name ➔ student municipality name ➔ student dept. no.
      = entities *Student, Student Commune* and *Student Dept.*
    - The *Student* entity has an identifier composed of the properties *Student Name* and *Student First Name*
    - the *Monitor* ID is a combination of *Last name* and *First name*
    - the hierarchical relations between entities follow the FDs (symbol ➔)
  - Information Attributes :
    - name+first name student ➔ student address: the address is thus a non-identifying property of the Student entity
    - same for *date of birth student*

CDM of the driving school case (continued)

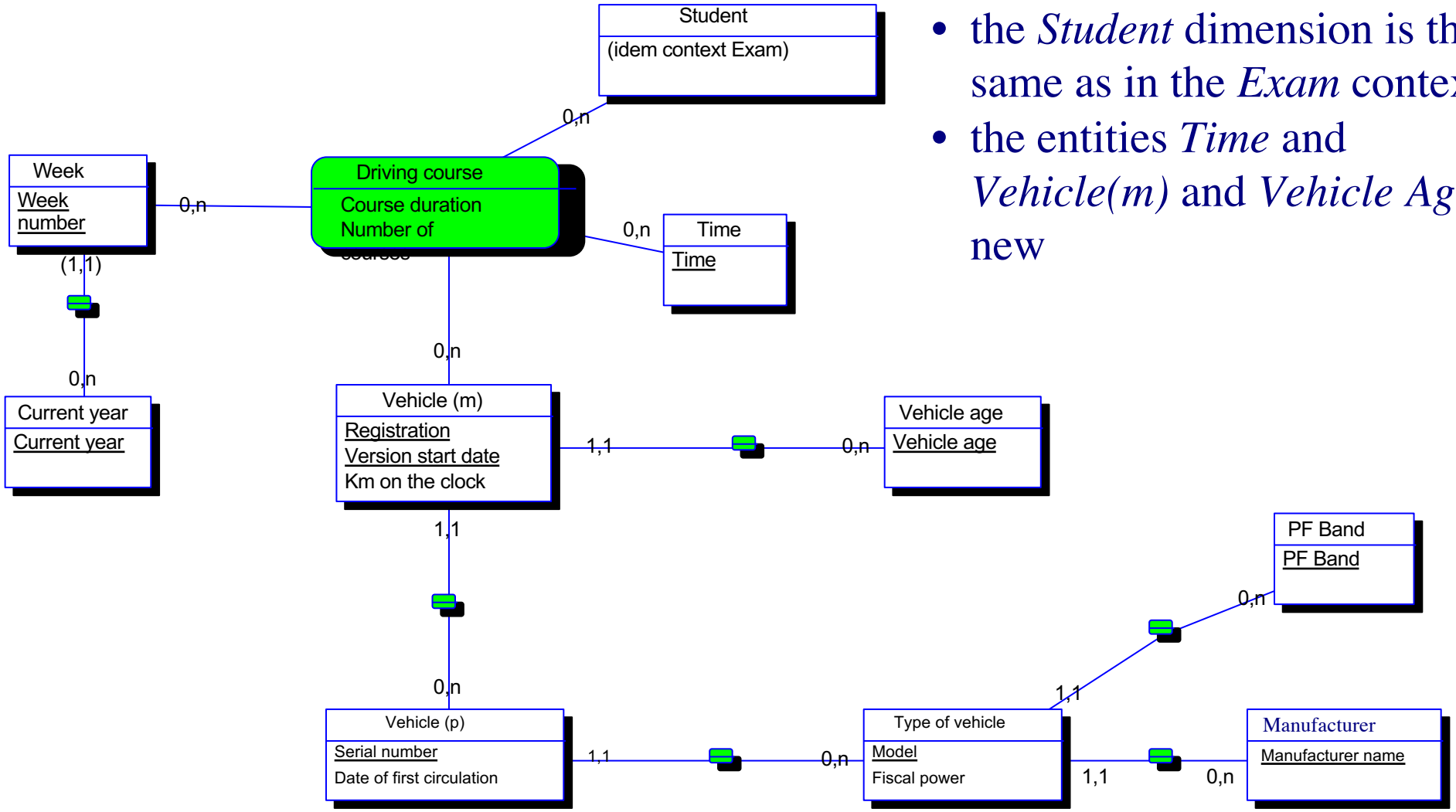
- Dimension *Date of examination*
  - Single hierarchy: Month+exam year ➔ exam year
    - entity *Month* made up of the identifying properties *Month No.* and *Year Examination*
    - entity *Year review* formed property *year review*
    - hierarchical binary association
      - cardinality of *1.1 on the Month* side
      - link identifier: *year exam* participates in the *Month* identifier
- *Review* context = association linking the 2 dimensions
  - links with the entities *Exam Month* and *Student* (atomic keys)
  - properties = measures *No. of attempts*, success *rate* and *average score*

MCD of the "examination" context



- the *vehicle type* dimension is similar to that of the *course* context, but with a "bigger" grain
- the *Inspector* entity is new

MCD of the context "driving course" context



- the *Student* dimension is the same as in the *Exam* context
- the entities *Time* and *Vehicle(m)* and *Vehicle Age* are new

Moving/Permanent Entities

- *Permanent* entity = timeless keys that cannot be modified (unless there is an error)
  - example : *VEHICLE(P) = Serial number, registration date*
- *Moving* entity = "photography », valid for a a limited time
  - keys whose value is variable over time
  - keys to identify and/or date the photo
  - 1 to N occurrences of (M) for one occurrence of (P)
  - measures indexed by (M) and not by (P)
  - example : *VEHICLE(M) = Registration, Km meter, Version start date*

Design steps for a multidimensional model

- Identify objectives and available data
- Identify measures :
- the measures expressed in the *objectives*
- the measures expressed in typical queries
- Identify the keys:
- define one entity per key
- organize entities within the dimensions
- add information attributes to entities
- Defining contexts: grouping measures of the same dimensions

Measure/key comparison

How to distinguish measures from keys?

Measures are :

- always digital
- time-varying
- for different combinations of keys
  (*no. of lessons* relative to *commune*, or to *year+monitor*, etc.)
- always cumulative by aggregation function(s)

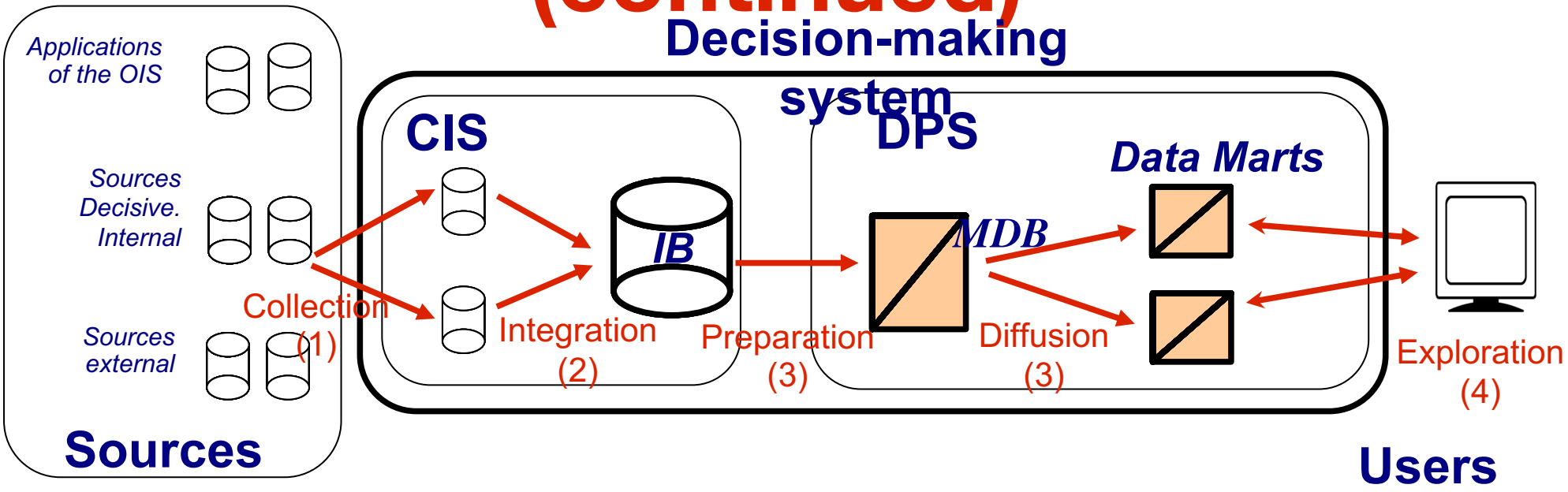
Key **values** are :

- numerical or textual
- either invariant or linked to moving entities
  (*name*, *address*, *age*, etc.)
- related to at most one other key
  (*vehicle age* depends only on *registration*)
- never cumulative
  (*vehicle age* is not cumulative)

Chapter 3
Architecture of the DIS

- Objective: to create the multidimensional model from the data of the OIS (Operational Information System)
- At the heart of DIS: an integration base (IB)
  - two steps *upstream of* IB :
    1. **collection** and filtering of data from the OIS applications
    2. harmonization and integration into the IB
  - two steps *downstream of* the IB :
    3. preparation and **dissemination of** multidimensional data
    4. interactive **presentation** to the user (decision maker)
- These steps will be presented in reverse order

Architecture of the DIS (continued)



- Two subsystems :
  - **CIS = Collection** (1) **and Integration** (2) **System**
  - **DPS = Diffusion System** (3) **and Presentation System** (4)
- Interface between the two: the **Integration Base (IB)**
- ***IB = the data warehouse*** according to most authors!

Two bases, two models

- **MDB** (*Database*) = multidimensional database
  - composed of *Data Marts*, reconstituted from the IB
  - native multi-dimensional format: ***M-OLAP***
    (Cognos, Oracle OLAP,...)
  - ...or Denormalized Relational Format: ***R-OLAP***
    (Mondrian/Pentaho, B.O.)
- **IB** *(Integration Base)* = DW
  - provides the data set of regular updates from previously cleaned, harmonized and historicized data
  - unique "memory" of the company (never *delete*!)
  - ... hence a classical, *normalized* relational model

Architecture of the driving school case

- Integration database = data warehouse
  - Realized by a *normalized* relational database
    - Classical CDM
    - PDM and loading scripts (SQL)

- Broadcasting Base = "Data Mart".
  - Realized by a *denormalized* relational database
    - Multidimensional CDM (see Chapter 2)
    - Denormalized PDM
    - fed from IB (SQL, scripts)

Step 4: Presentation

- Atomic measurements are not presentable
  - too insignificant (example: no. of successes per student)
  - too many values
- The decision-maker prefers aggregated measures, a priori
- But it must also be able to "zoom in" on certain measurements to observe the details
- Objectives of the presentation stage:
  - present the data in a user-friendly way
  - allow for interactive exploration
- This step requires a specialized tool
  (during this course: SAP *Business Objects*)

Exploring a context

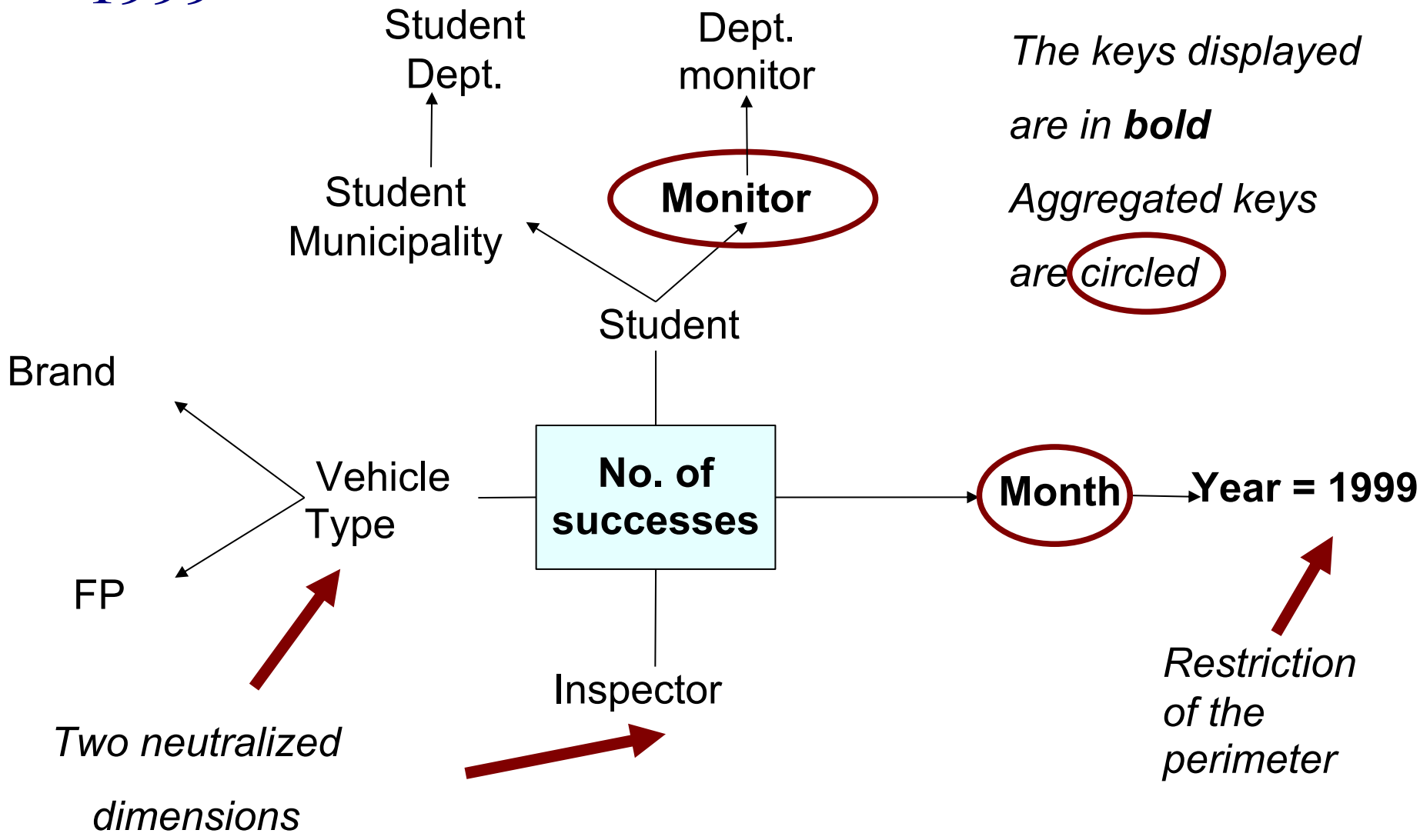
- Challenge: to show an intelligible subset of the cube, without hiding anything!
- The exploration consists in choosing :
  - the **level of aggregation of** each dimension
  - the **exploration perimeter**, by constraining the values of certain keys
  - the keys whose values are **displayed**
  - Example: *No. of successes per monitor and month in 1999* :
    - the perimeter covers the measurements verifying *Year exam =1999*
    - aggregation by monitor and month, for all types and inspectors
    - display: year, month, monitor, number of successes

Exploration tactics : tabular view

- The tabular view focuses on two dimensions of the cube:
  - neutralization of all dimensions (choice of ALL key), except two
  - choice of an aggregation level in each of these two dimensions
  - perimeter definition
  - other dimensions can appear, provided that the key values of several dimensions are combined on the same axis of the table

Example of exploration

- Query: Nb of successes by monitor and month in 1999

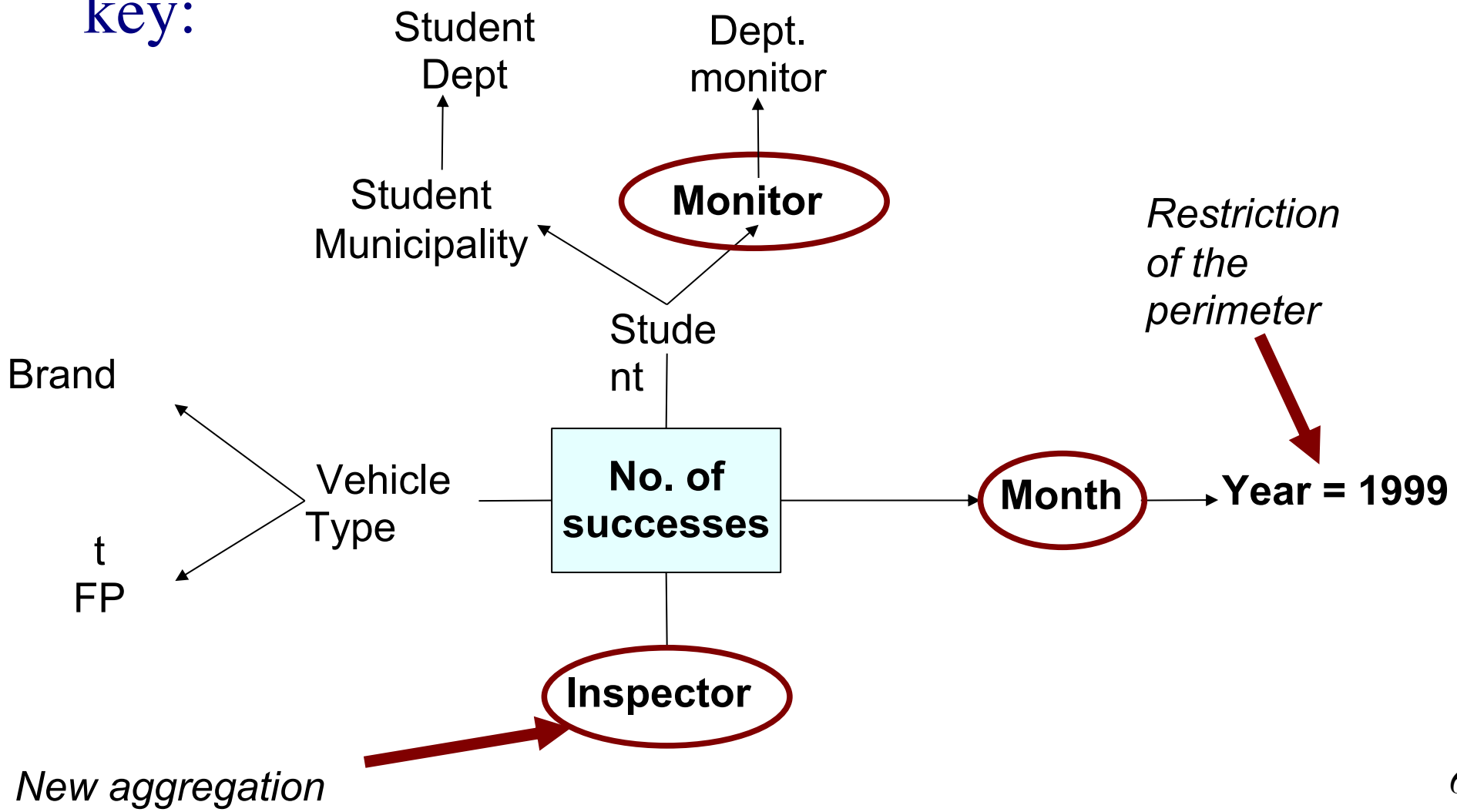


Resulting table : monitor * month/year

	1999	1999	1999	1999	1999	1999	1999	1999	1999	1999
	3	4	5	6	7	8	9	10	11	12
Maillet Sophie		0	2	0	2		1	1	1	1
Meursault Antoine			1						0	
Meyer Julie	0	0					1		2	
Moreau François	0			0				0		
Morel Gérard						0		0		

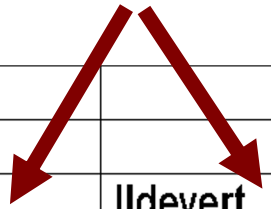
3 dimensional chart

- To understand the impact of the inspectors, we aggregate the measurements at the level of this key:



3-dimensional table (continued)

Two dimensions are represented on the same axis



		1999	1999	1999	1999	1999	1999	1999	1999	19
		3	4	5	6	7	9	10	11	
Maillet Sophie	Ildevert Jacques						1		1	
Maillet Sophie	Imbert Jacques			1						
Maillet Sophie	Imbert Julie		0	1	0	2		1		
Maillet Sophie	Irel Jacques			0						
Meursault Antoine	Imbert Julie			1					0	
Moreau François	Ildevert Jacques							0		
Moreau François	Imbert Jacques				0					
Moreau François	Imbert Julie	0								

Thomsen's projection

- Principle: N+1 logical dimensions are projected onto a 3-dimensional window
- Logical dimensions: N key dimensions + 1 measurement dimension
- Physical" dimensions of the window :
  - rows
  - columns
  - pages (tabs)
- Each physical dimension hosts 0, 1 or more logical dimensions

Example of projection

Logical dimensions

geography *and*

measurements

in rows

Logical **time** *dimension*

in columns

		1999				
		Jan	**Fév**	**...**	**Nov**	**Déc**
Centre	**CA**	81	88		90	101
	Qté	139	144		177	183
Est	**CA**	60	65		65	69
	Qté	181	183		205	190
Nord	**CA**	50	52		55	56
	Qté	92	90		44	89

Logical dimension

product *in pages*

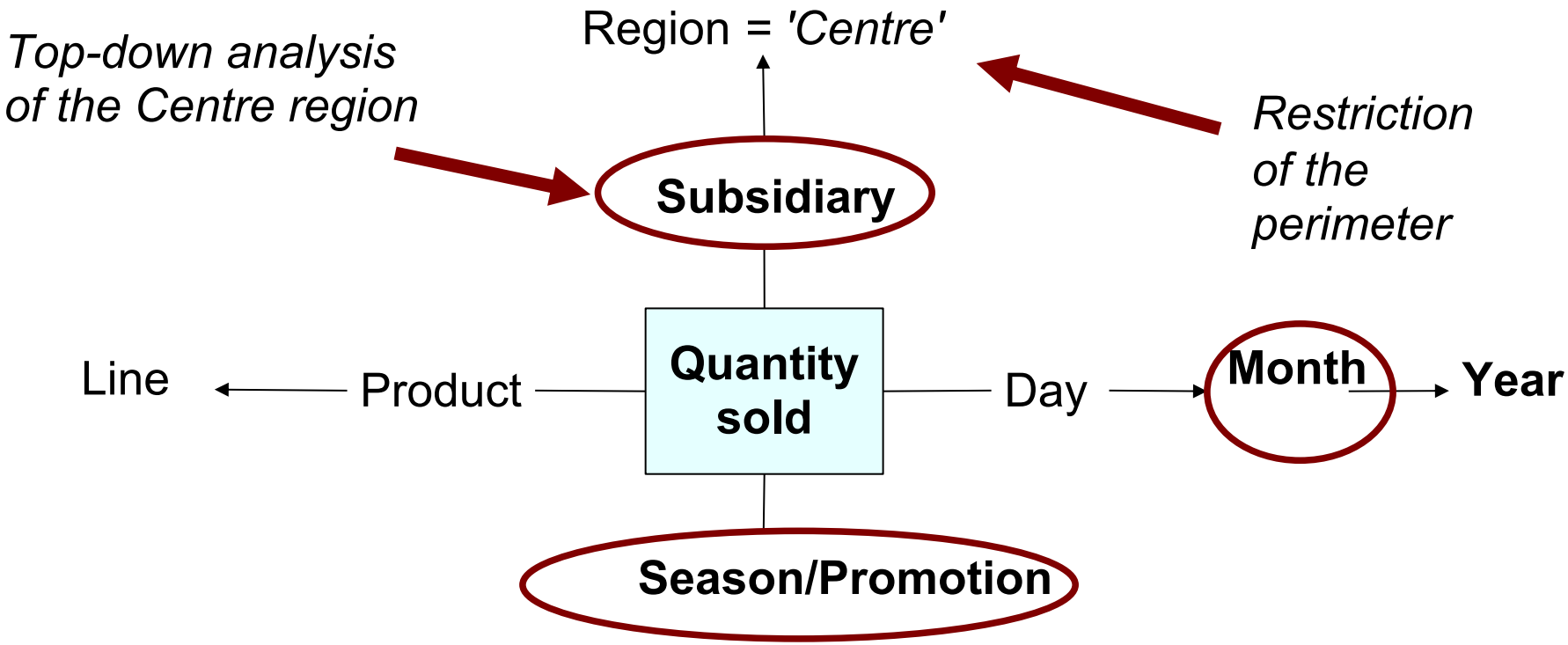
home garden

Navigation actions

- The user can "navigate" through the cube by incremental actions such as :
  - top-down analysis (or *drill-down*, or *roll-down)*
  - bottom-up analysis (or consolidation / drill-down / *drill-up* or *roll-up)*
  - lateral analysis
  - Filtering (or key selection, or *screening)*
  - projection on certain dimensions, or *slicing*
  - rotation, or *pivot*
  - merge

Top-down analysis

- This action consists of two elementary actions on one dimension:
  - scope is limited to one value of the aggregated key
  - aggregation on a lower level key



Top-down analysis : result

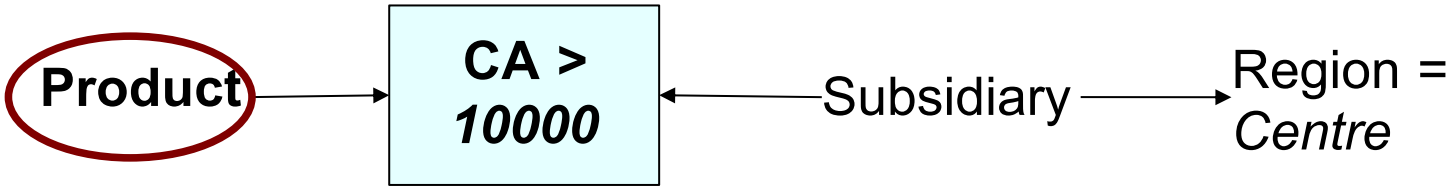
		1999			
		Jan	**Fév**	**...**	**Nov**
Filiale 1	**Saison**	33	39		65
	Promo	17	10		9
Filiale 2	**Saison**	20	20		12
	Promo	5	12		13
Filiale 3	**Saison**	28	29		13
	Promo	17	22		15
Total Centre	**Saison**	81	88		90
	Promo	39	44		37

Bottom-up / lateral analysis

- **Bottom-up analysis** = reverse of top-down analysis
- This action consists of two elementary actions on one dimension:
  - the aggregation level "goes up" to the next level
  - the constraint on the key at this level is removed
- **Lateral analysis**: the constraint is maintained, but the selection key value is changed
  - example: selection on *February* instead of *January*

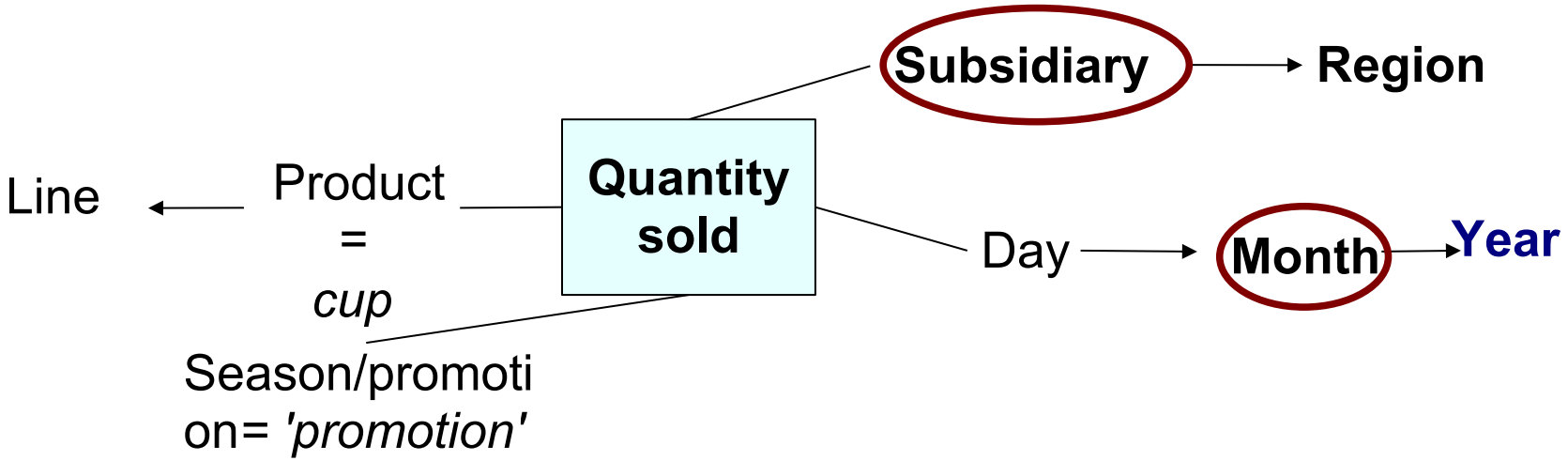
Filtering

- Synonyms: *screening*, *key selection*
- This type of action concerns only the exploration perimeter
- More or less complex conditions are imposed on the keys and/or measures
- Information attributes may be involved
- Example: products from the Centre region with a turnover higher than 10000



Projection on N-M dimensions

- In an N-dimensional context, we impose a condition on the keys of M dimensions
- These M dimensions are aggregated totally
- Result: a "cube slice" of N-M dimensions
  - example: details of sales of promotional cups (N=4, M=2)



Rotation / Merge

- **Rotation** = choosing another dimension on an axis
- Two possibilities:
  - moving a dimension already displayed.
    Example: the *season/promotion* rows become sub-columns of the *month* columns
  - replacement on an axis of a displayed dimension by a non displayed dimension
    Example: *Subsidiary* is replaced by *Product line*
- **Merge** = grouping of measures from different contexts. Requirement: have browsed each context to obtain compatible views

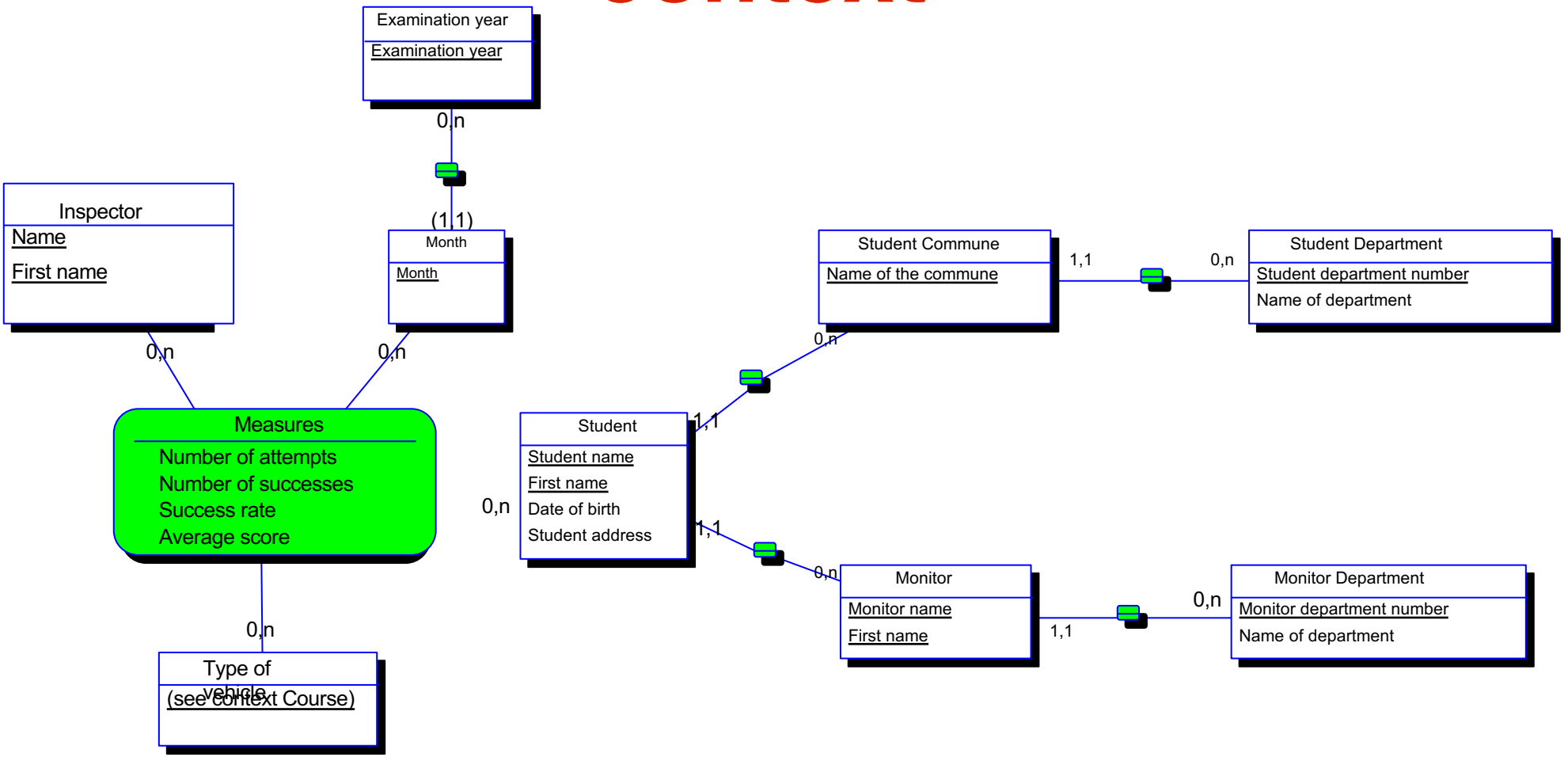
Step 3: Dissemination

- The MDB provides the multidimensional data
  - partial copy of the IB, with
    - development of basic measures
    - setting up dimensional keys
  - storage in native multidimensional or relational format
  - often confused with IB (data warehouse)
- The MDB is composed of *Data Marts*, structured by domains:
  - sets of contexts with common dimensions
  - Implementation very dependent on the presentation software

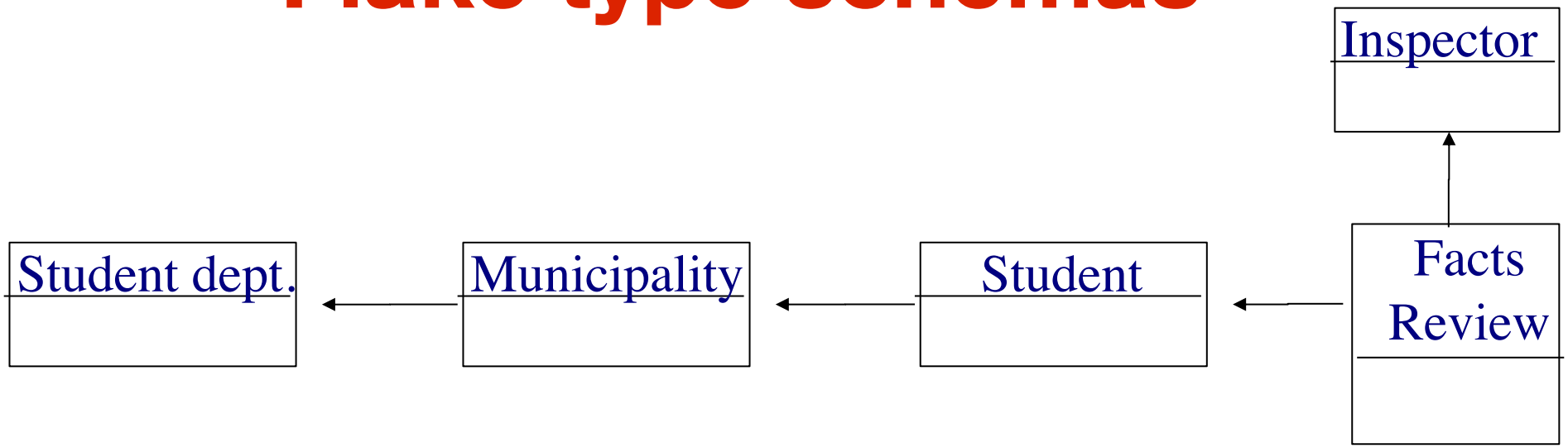
From the multidimensional CDM to relational schema

- R-OLAP technology (the only one seen here): Database stored in *relational* format
- Each context is translated into a table called *fact table*. In general :
  - one column per elementary measure (not calculated)
  - one foreign key per dimension
  - no primary key
- At least one table per dimension:
  - sequential and digital primary key
  - one column per key or information attribute

MCD of the "examination" context

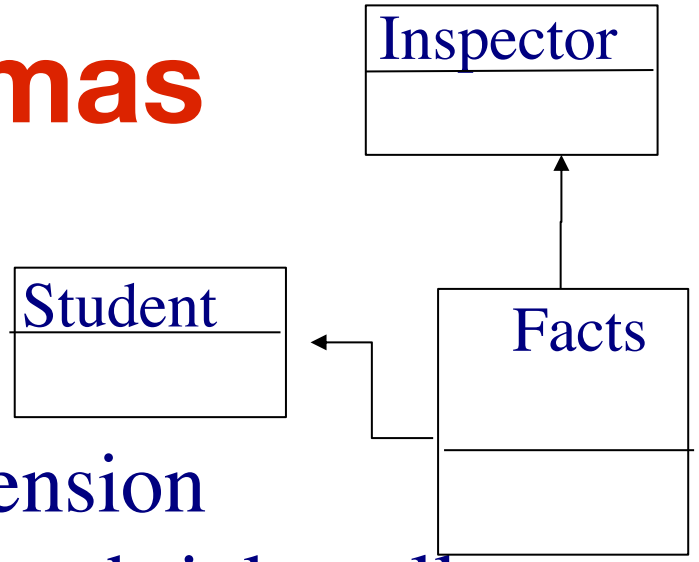


Flake type schemas



- *Classical translation of a CDM to LDM*
- *Flake* model: dimensions are normalized according to the 3NF
- No redundancy
- Limits : join cost for interactive exploration

star type schemas



- *Star* model: at most one denormalized table per dimension
- Denormalized: for a dimension, « shrink » all entities in one (the grain/seed)
- Advantage: the number of joins is minimal and moreover constant
- Disadvantage: the recopies of values increase the volume. But:
  - the dimensional tables occupy only 5 to 10% of the total volume
  - tables are generated from the IB, never modified: no update anomalies

Fact tables

- Principle: one row for any significant combination of measurements (non-zero)
- Aggregation by *sum*, *average*, *min*, *max*, etc.
- One column per measure, except for measures *by coverage* :
  - no row: measurement = 0
  - presence of row: measure = 1
  - aggregation by the *count* function
  - example: measures = (*no. of tickets sold*, entrance fee) dimensions = *(Film, Day, Customer, Room)*

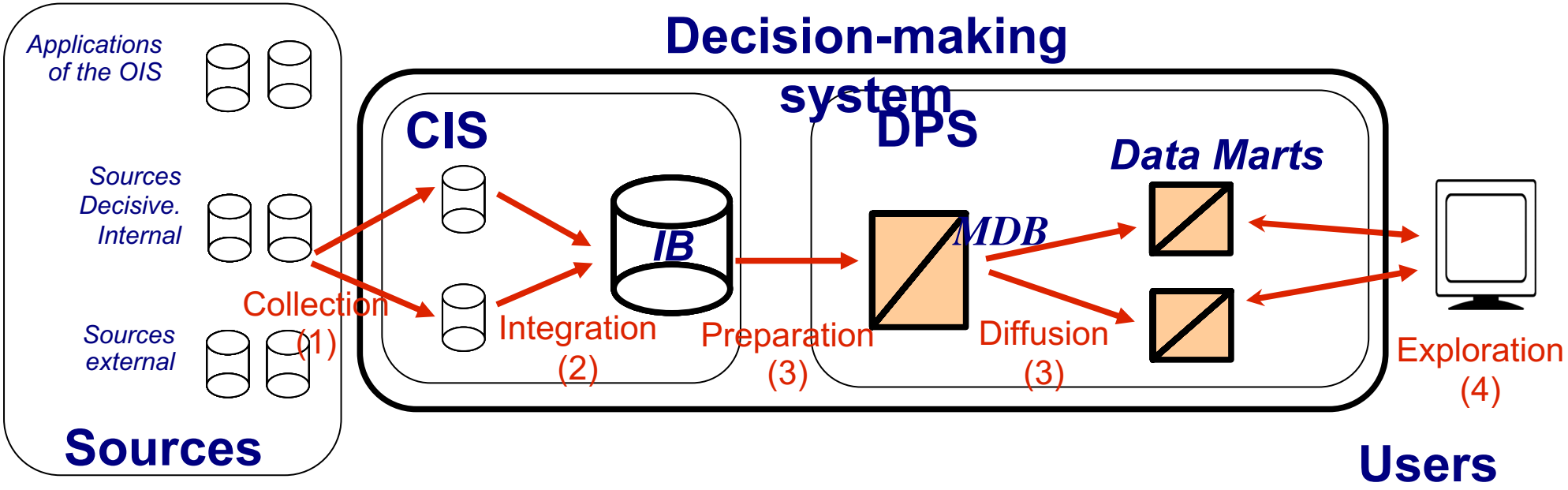
Special cases (1)

- **Degraded dimension**: no table!
  - no more need for a foreign key in the fact table
  - one column per key in the fact table
  - advantage: one join less
  - reserved for very small dimensions.
    Examples: dimension *dates exam*, *date course*
- **Multiple fact tables** for the same context
  - same foreign keys
  - different numbers

Aggregate tables

- Dynamic aggregation is expensive: pre-computed aggregates can be used
- **Solution 1:** Individual aggregate tables
  Example: Aggregates by *Region*/*Month*/*Product Family*.
  Problem: one table per aggregation level
- **Solution 2**: Grouping of aggregates in the fact table
  - no change in the structure of the fact table
  - one additional row per pre-calculated aggregate
  - aggregation level indicator in each dimensional table

Architecture of the DIS



- Two subsystems :
  - **CIS = Collection** (1) **and Integration** (2) **System**
  - **DPS = Diffusion System** (3) **and Presentation System** (4)
- Interface between the two: the **Integration Base (IB)**
- ***IB = the data warehouse*** according to most authors!

The Integration base (IB)

- The Diffusion database is fed by the **Integration Database** (the proper DW)
- Need to distinguish between *IB* and M*DB* :
  - *Multidimensional Difusion database*
    - composed of *Data Marts* adapted to presentation tools
    - **native multidimensional** (M-OLAP) or **simulated** (R-OLAP)
    - data that can be consulted and sometimes modified by the end user
  - *Integration database*
    - unique, centralized and historical reference base
    - **relational** because it is incrementally enriched from data sources

Steps 1 and 2: feeding the IB

- Step 1
  - selective **collection of** "dirty" data from sources
  - **harmonization** in "clean" intermediate tables
- Step 2: **Integration** into IB tables
  - *Permanent* entities :
    - no deletions, only insertions
    - change of ownership: considered as a correction (old value lost because "false")
  - *Moving* Entities :
    - no deletion or modification
    - add occurrence for any change in the source

Step 0: Decision-making project approach

- Study of decision needs: identification of measures and dimensions (grain)
- Study of the existing system (OIS)
- *Conceptual* analysis
  - multidimensional model (Mutidim. CDM)
- *Technical* analysis
  - logical and physical CIS model (IB + other tables)
  - logical and physical models of the DPS (MDB, Data Marts)
  - processing analysis: collection, integration, aggregation, distribution

Glossary

- IDB: integration Database (FR-base d'integration)
- BI: business intelligence (FR-le décisionnel)
- OLAP: on-line analytical processing
- ROLAP: OLAP based on a traditionnal RDBMS
- BO: SAP Business Object
- DW: Data Warehouse (FR-entrepôt de données)
- DIS: Decisional information systems (FR-système d'information décisionnel)