Object-attribute data:
from philosophy to visualization through mathematics and programming

Speaker: Alexey Neznanov
Senior Data Scientist, SMR Center & Digital Development, Schlumberger
2025-01-28

V. 0.09.01

About the speaker

- Alexey Neznanov
  - PhD, IEEE member
  - Senior Data Scientist at Schlumberger
  - Senior Researcher at International laboratory of Intelligent systems and structural analysis (http://cs.hse.ru/ai/issa)
  - «Small Guide to Big Data» consultant at PostScience portal (http://postnauka.ru/author/a_neznanov)
  - Author of textbook, instructional guidelines, 11 courses, and more than 65 research papers

Some discussed terms

- Entity and concept
- Object and class
- Attribute and attribute value
- Measure, scale, and unit of measure
- Relation and relationship
- Tuple and record
- Type and format
- Interface and protocol
- Table, pivot table, and dataframe
- Machine learning, datasets and features
- DataViz

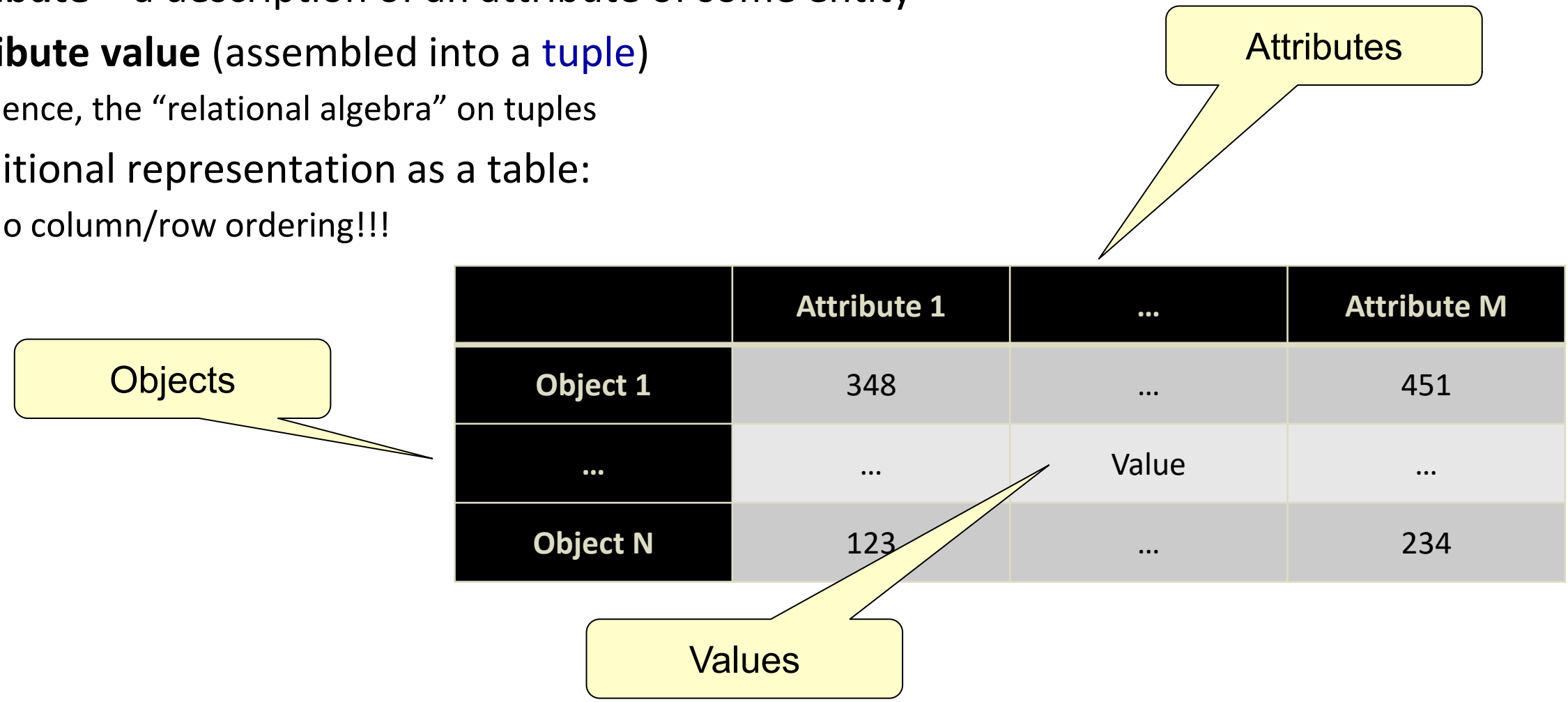
!! Data and metadata → in **other** presentation

Basic sources

- Logic and set theory
  - Full Open-Source, Collaborative Logic Text, aimed at a non-mathematical audience (http://openlogicproject.org)
- Formal Concept Analysis (FCA)
  - Formal Concept Analysis Homepage by Uta Priss (http://upriss.github.io/fca/fca.html)
  - Introduction to Formal Concept Analysis by Radim Belohlavek (http://phoenix.inf.upol.cz/esf/ucebni/formal.pdf)
- Relational model
  - Codd, E. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 1970, pp. 377-387. (http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf)
  - Codd E.F. The relational model for database management: version 2. Addison-Wesley, 1990. 538 p. (http://dl.acm.org/doi/pdf/10.5555/77708)
  - Date C.J. Database Design and Relational Theory. Normal Forms and All That Jazz. O'Reilly Media, 2012. 260 p.
  - Mancas C. Conceptual Data Modeling and Database Design: a Fully Algorithmic Approach. Volume 1, The Shortest Advisable Path. Apple Academic Press, 2016. 634 p.

Object-attribute data (tables?)

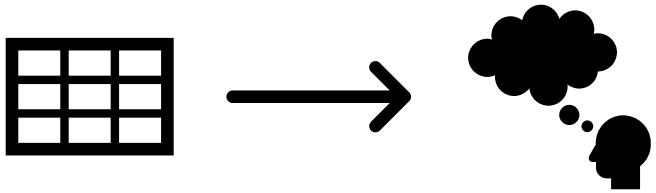
- **Object** – an instance of some entity
- **Attribute** – a description of an attribute of some entity
- **Attribute value** (assembled into a tuple)
  - Hence, the “relational algebra” on tuples
- Traditional representation as a table:
  - No column/row ordering!!!



	Attribute 1	…	Attribute M
Object 1	348	…	451
…	…	Value	…
Object N	123	…	234

Attribute synonyms

- Accent
- Aspect
- **Attribute**
- Characteristic
- Condition
- Connotation
- Degree
- Distinctor (distinction)
- Facet
- **Feature**
- Genus
- Indicator (indication)
- Inherent
- Kind
- Mark
- Peculiarity
- Predicate
- **Property**
- Quality
- Quirk
- Score
- Sign
- Status
- Symptom
- Trait
- Type
- Variety

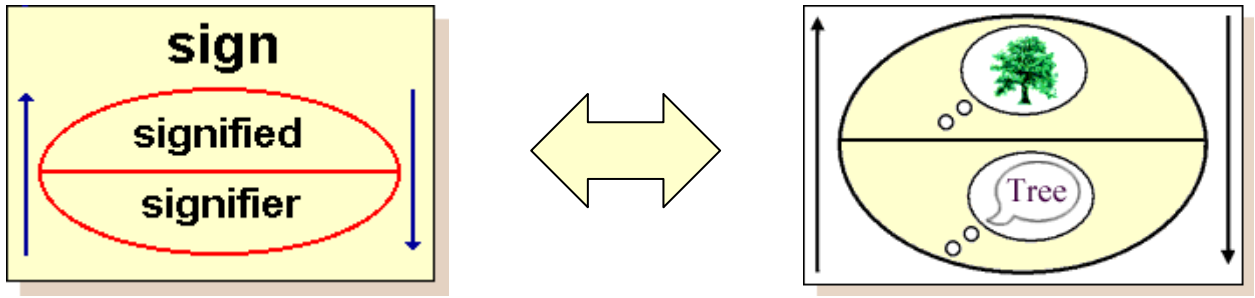


FANTASTIC ENTITIES AND WHERE TO FIND THEM

- ✓ Semiotics and sign systems
- ✓ Entities and concepts
- ✓ Hermeneutic circle, natural language as a language of philosophy
- ✓ Ontology, epistemology, gnoseology
- ✓ Grounding
- ✓ Measurements and scales

Semiotics and sign system

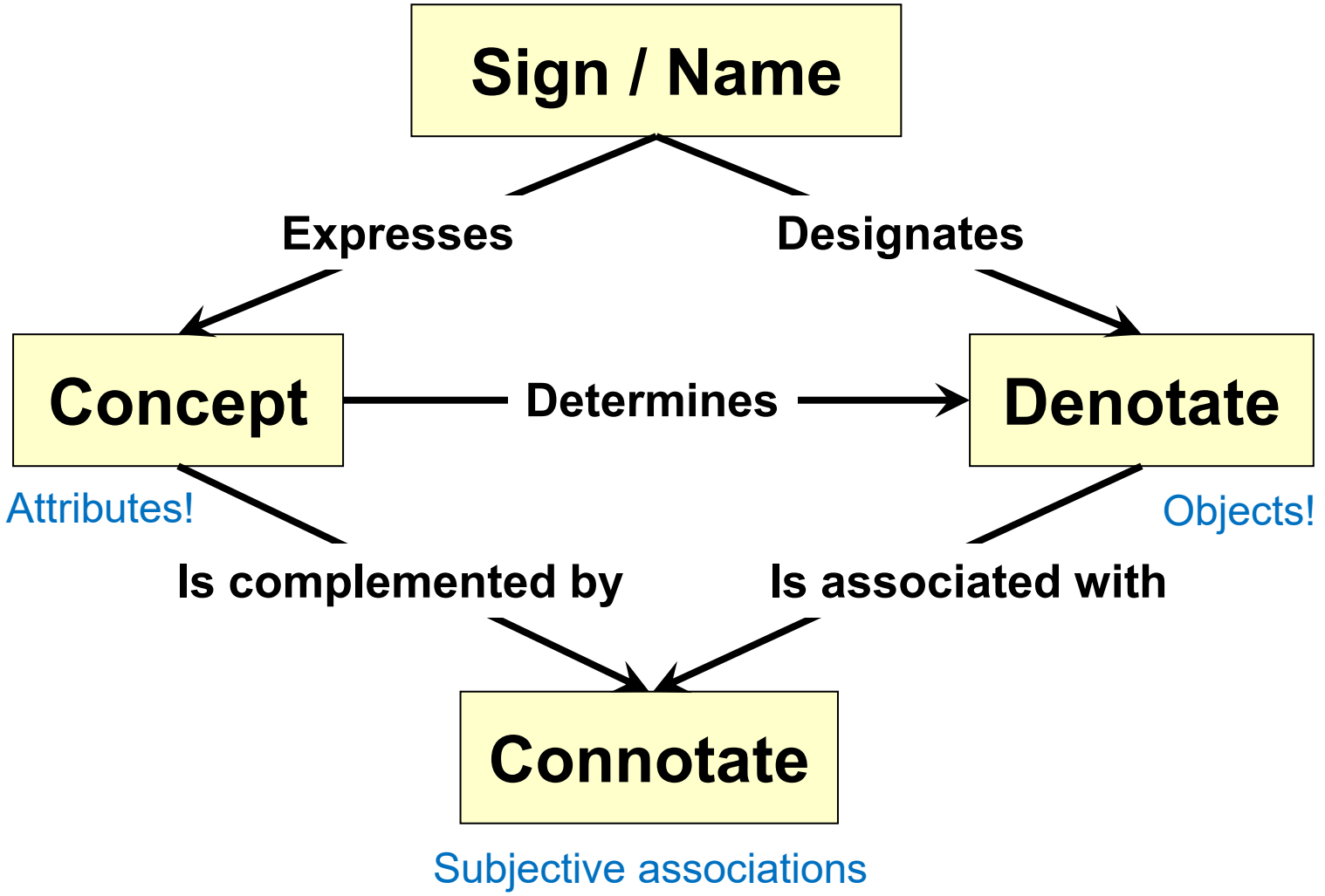
- **Sign system** – any system of signs and relations between signs
  - Sign system – algebraic system intended to capture the systematic structure of complex signs, since in all but the simplest cases, most signs are composed from other signs
    - Goguen J.A. Semiotic Morphisms. 2004
  - Semiotics for Beginners (https://www.cs.princeton.edu/~chazelle/courses/BIB/semio2.htm)



- We interpret things as signs largely unconsciously by relating them to familiar systems of conventions!
  - **Codes** for messages transmission
    - Encoding is the process of creating a message for transmission by an addresser (sender) to an addressee (receiver)
    - Decoding is the process of interpreting a message sent by an addresser (sender) to an addressee (receiver)

Frege triangle – most popular model of “sign”

- F.L.G. Frege, 1848-1925
- Basic triangle + connotations:



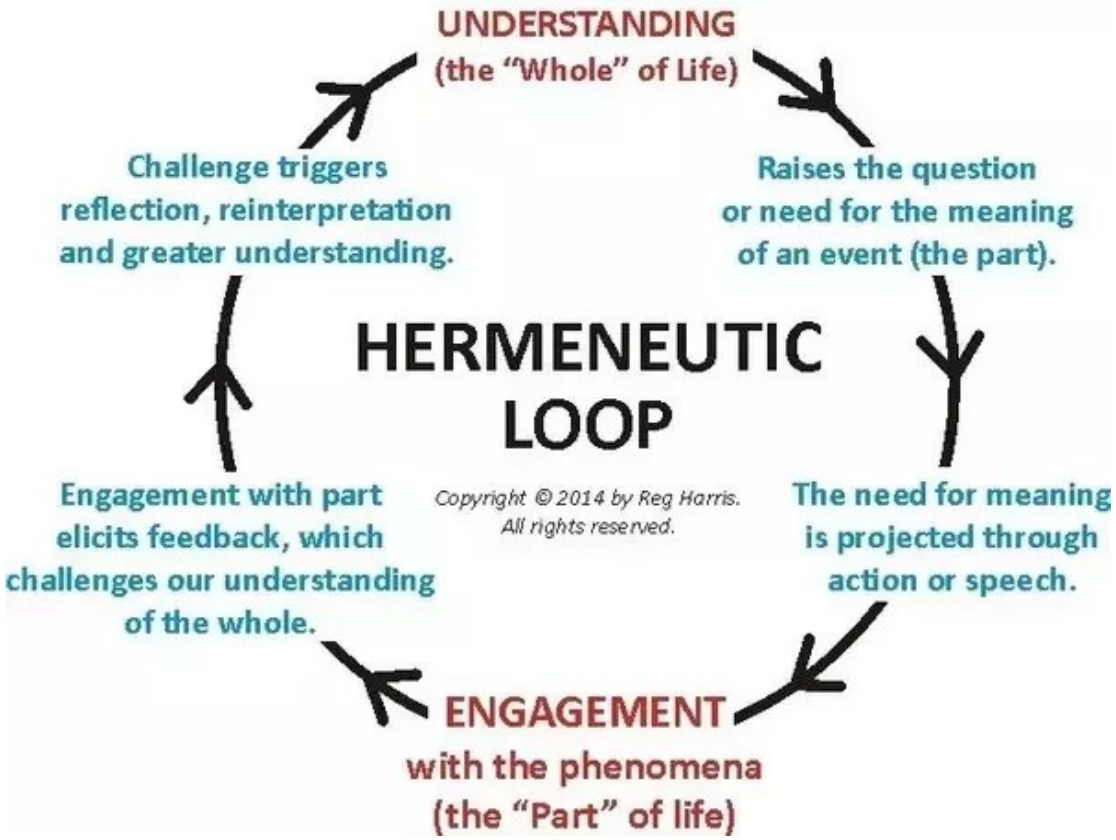
Semiotics deconstruction: semantics, syntactics, and pragmatics

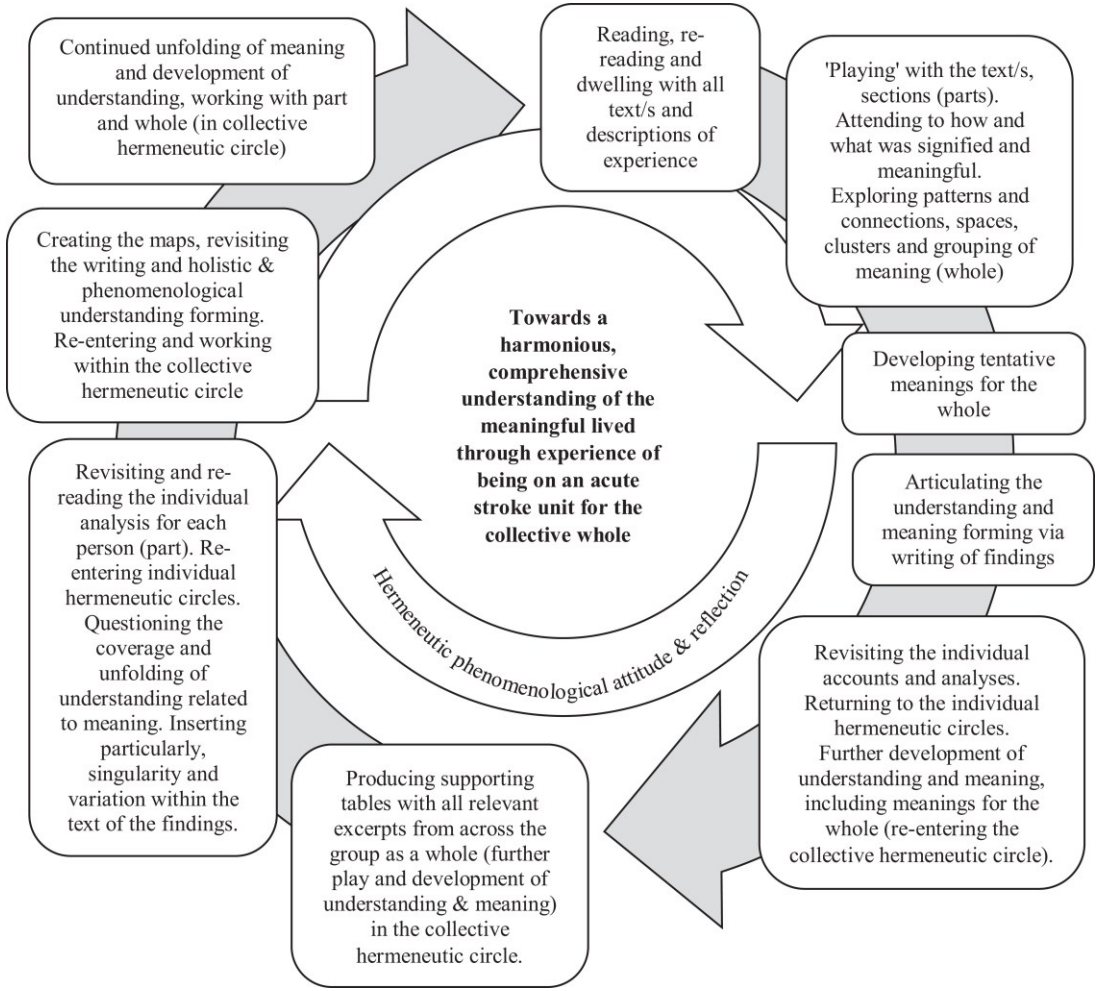
- Three points of view (the branches of semiotics):
  - **Semantics** – this is the “how” of semiotics, and is concerned with this relationship between a signified and signifier – the sign and what it stands in for
  - **Syntactics** – this refers to structural relations in the set of signs
    - One structural relation in language is **grammar**, but syntactics in semiotics refers to the formal relationship between signs that lets them build into sign systems
  - **Pragmatics** – according to Morris, is the relationship of sign to the person reading or understanding that sign
    - Morris C. Foundations of the Theory of Signs (in International Encyclopaedia of Unified Science, V.1). University of Chicago Press, 1938. 59 p.

- “The syntactic and semantic structure of natural languages evidently offers many mysteries, both of fact and of principle, and any attempt to delimit these domains must certainly be quite tentative”
  - Chomsky N. Aspects of the Theory of Syntax. MIT Press, 1965. 261 p.

Hermeneutics

- Hermeneutic loop/circle
  - Martin Heidegger – Being and Time, 1927





Hermeneutic circle in practice

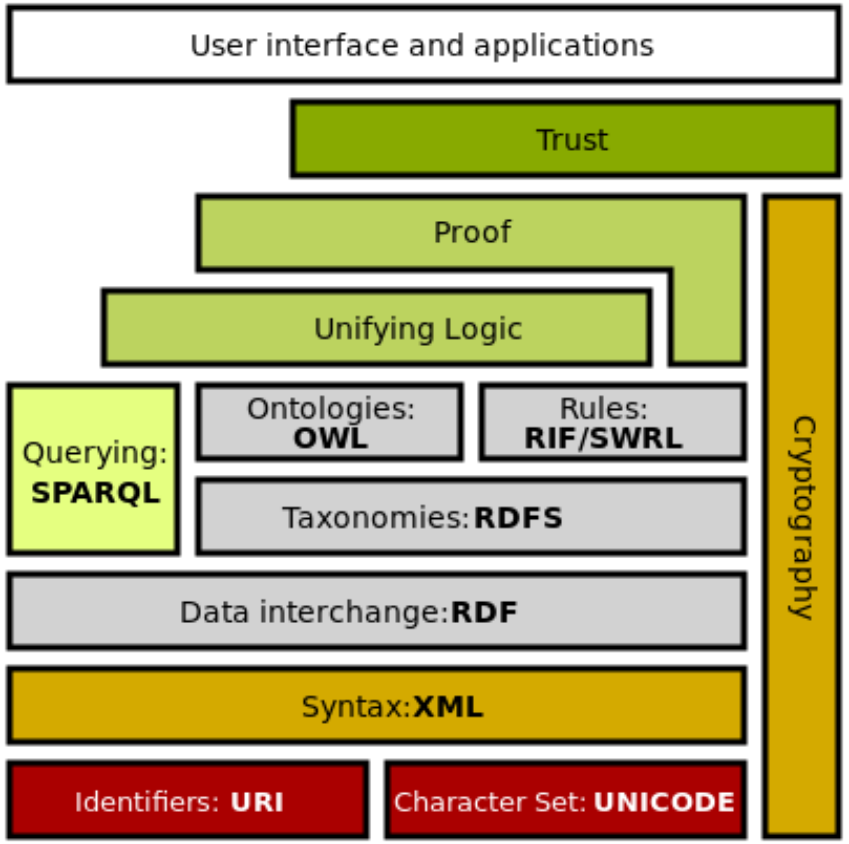
- From “Basic Formal Ontology 2.0 – Specification and User’s Guide” (http://github.com/BFO-ontology/BFO/raw/master/docs/bfo2-reference/BFO2-Reference.pdf):
  - “… we distinguish between primitive and defined. ‘Entity’ is an example of one such primitive term.
  - Primitive terms in a highest-level ontology such as BFO are terms that are so basic to our BFO 2.0 Specification and User’s Guide understanding of reality that there is **no way** of defining them in a **non-circular fashion**.
  - For these, therefore, we can provide only elucidations, supplemented by examples and by axioms”

- Fuenmayor D., Benzmüller C. A Computational-Hermeneutic Approach for Conceptual Explicitation. arXiv:1906.06582, 2019 (http://arxiv.org/abs/1906.06582)
  - !!!

Ontology (+ontics), epistemology, gnoseology

- Technical ontology – formalized computer-readable domain knowledge
- T. Gruber, 1993:
  "**Explicit specification of a conceptualization**"
  - The most cited definition!
  - + "**Shared**"

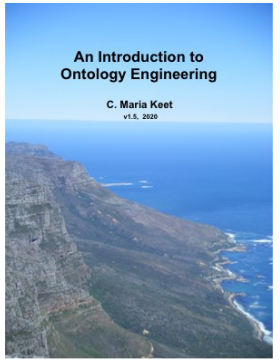
- ***Sematic Web*** (http://www.w3.org/standards/semanticweb/)
  - *XML* (http://www.w3.org/XML/)
  - *RDF* (http://www.w3.org/RDF/)
  - *OWL* (http://www.w3.org/OWL/)
  - *SPARQL* (http://www.w3.org/TR/sparql11-overview/)
  - +++



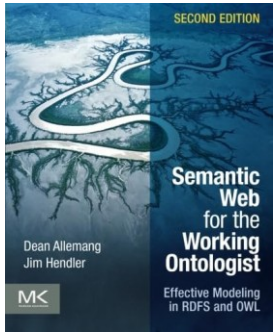
From wikipedia, license: CC BY-SA

An introduction to ontology engineering

- Keet M. An introduction to ontology engineering. 2020 (https://people.cs.uct.ac.za/~mkeet/OEbook/)
  - The best introduction



- Allemang D., Hendler J. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. 2nd Ed. Morgan Kaufmann, 2011. 384 p.



The problem of definitions

- Different domains → different languages → different conceptualizations
  - **Vocabulary**: the body of words used in a particular language
  - **Glossary**: link terms to concepts, described informally by glosses
  - **Thesaurus**: add structural (semantic) relationships among terms (and concepts), i.e. generalization, part, dependence, causation, …

- Michie S., West R., Hastings J. Creating ontological definitions for use in science. 2022 (http://www.qeios.com/read/YGIF9B)
  - Vocabulary definitions are often ambiguous or circular
  - We try to use constructive distinctive non-circular definitions!

- Bjørner D. Domain Science and Engineering. Springer, 2021. 401 p.

Ontologies in various scientific fields

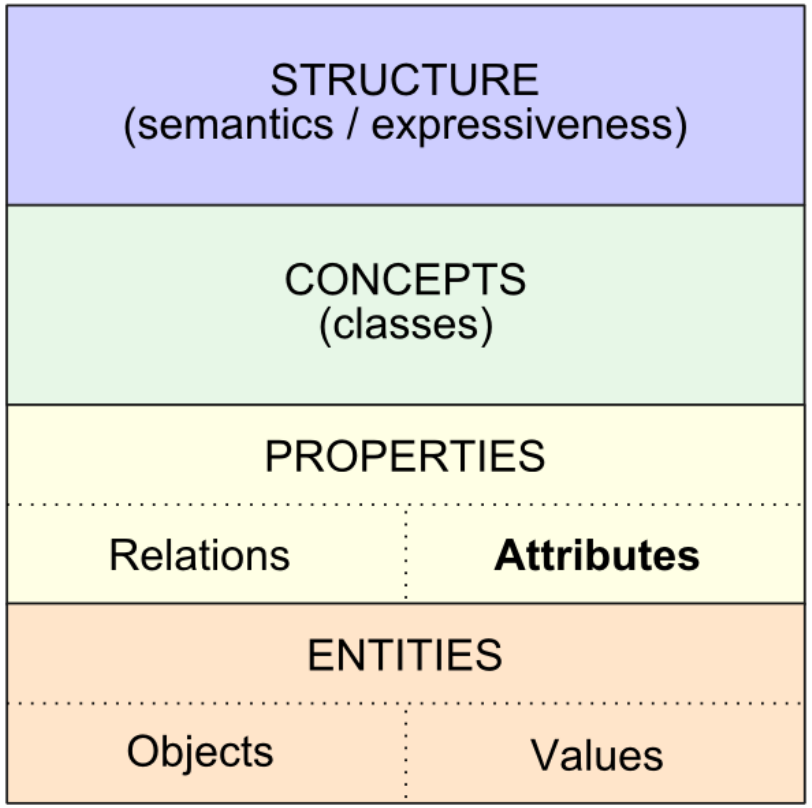
- Interdisciplinary discussions!
- Busse J., Humm B., Lubbert C., Moelter F., Reibold A., Rewald M., Schlüter V., Seiler B., Tegtmeier E., Zeh T. **Actually, What Does "Ontology" Mean**? A Term Coined by Philosophy in the Light of Different Scientific Disciplines / Journal of Computing and Information Technology. 23(29), 2015.
  - Participants:
    - Philosopher
    - Psychologist
    - Information scientist

Philosophy	Computer Science
Substance	Class, entity, concept
Accidents	Attribute, property
Fact	Fact, statement, proposition, relation
Ontology	Base-ontology, foundational ontology

Main question: attribute or relationship?

- Attribute as value object and as reference
- Depends on:
  - Abstraction level
  - Level of specification details
  - Data access layer

- Attribute does not have Identifier, instead it is stored directly as some datatyped variable inside entity
  - Conceptual and Practical Distinctions in the Attributes Ontology (https://www.mkbergman.com/1842/conceptual-and-practical-distinctions-in-the-attributes-ontology/)



Interesting case of standardization of terminology

- ISO 5127:2017 Information and documentation — Foundation and vocabulary (http://www.iso.org/obp/ui#iso:std:iso:5127:ed-2:v1:en)

3.1.1.01 – object
anything perceivable or conceivable
Note 1 to entry: *Objects* may be material (e.g. an engine, a sheet of paper, a diamond), immaterial (e.g. conversion ratio, a project plan) or imagined (e.g. a unicorn).
[SOURCE:ISO 1087-1:2000, definition 3.1.1]
Note 2 to entry: See also *"entity"* (3.1.13.27); and ISO/IEC 27000:2016, definition 2.55.

3.1.1.60 – material object
physical object
object (3.1.1.01) which has physical extension in space and time
Note 1 to entry: See also *"naturafact"* (3.2.1.01), *"artefact (1) <man-made object>* (3.2.1.02); ISO 21127:2014, classes *"E 18"* and *"E 19"*.

3.1.1.03 – property
distinguishing feature of a *material object* (3.1.1.60)
[SOURCE:http://iso.i-term.dk/login.php, modified]
Note 1 to entry: See also ISO 21127:2014, definition 3.14.

3.1.1.02 – concept
unit of *knowledge* (3.1.1.17) created by a unique combination of *characteristics* (3.1.1.04)
Note 1 to entry: *Concepts* are not necessarily bound to particular languages. They are, however, influenced by the social or cultural background, which often leads to different categorizations.
[SOURCE:ISO 1087-1:2000, definition 3.2.10]
Note 2 to entry: *Concepts* represent *objects* (3.1.1.01).
Note 3 to entry: See also ISO 25964-1:2011, definition 2.11.
Note 4 to entry: See also *"class"* (3.8.5.03), *"conceptual object"* (3.1.1.61), *"mentefact"* (3.2.1.03).

3.1.1.61 – conceptual object
object (3.1.1.01) which is the product of the mental activity of applying abstraction in human mind activities
Note 1 to entry: See also ISO 21127:2014, classes *"E 28"* and *"E 65"*.

3.1.1.04 – characteristic
abstraction of a *property* (3.1.1.03) of an *object* (3.1.1.01) or of a *set* (3.1.1.09) of objects
Note 1 to entry: Characteristics are used for describing *concepts* (3.1.1.02).
[SOURCE:ISO 1087-1:2000, definition 3.2.4]

Some comparisons of terminology differences

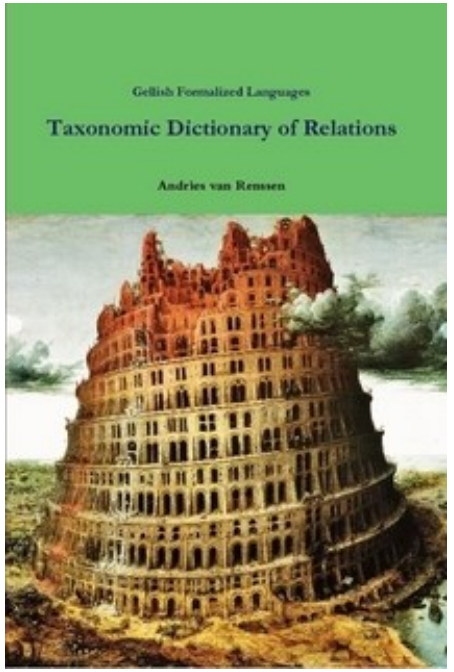
UMBEL/AO Terminology	Terminology Used Elsewhere	
Entity type	concept kind set	collection type class
Entity	object instance exemplar element	member record individual dependent variable
Attribute	property predicate relationship feature facet	dimension characteristic field header independent variables

Conceiving and understanding: concepts and meanings

- Seiler T.B. **Begreifen und Verstehen**: ein Buch über Begriffe und Bedeutungen. Darmstädter Schriften zur allgemeinen Wissenschaft. Verlag Allgemeine Wissenschaft, Mühltal, 2001.
  - Seiler discusses a great variety of concept theories in philosophy and psychology and concludes with his own theory which extends the concept understanding of Piaget's structure-genetic approach
  - In his theory, Seiler describes concepts as cognitive structures whose development in human mind is constructive and adaptive
  - Twelve aspects!
    - They are briefly described in the following twelve subsections and used to review the adequacy of the mathematizations of FCA
    - In each subsection, the first paragraph concisely summarizes Seiler's understanding of the corresponding aspect; then related notions and relationships from FCA are discussed and partly concretized by at least one example. The connections between Seiler's concept theory and FCA which come apparent in this way are far from being exhaustive. But they already show an astonishing multitude of correspondences between both theories which may be taken as arguments for the adequacy of the discussed mathematizations

Controlled language and semantic relations

- Andries van Renssen A. **Taxonomic Dictionary of Relations**, 2016 (http://www.lulu.com/shop/andries-van-renssen/taxonomic-dictionary-of-relations/paperback/product-22534783.html)
  - This is a taxonomic dictionary, which means that the defined concepts are arranged in a strict subtype-supertype hierarchy, also called a *taxonomy*
  - The dictionary defines kinds of relations that are usually not found in conventional dictionaries.
    - A number of the kinds of relations are also defined in various ISO and non-ISO standards
  - More than **5000** relations!



- Implementation in Gellish:
  - Gellish Syntax and Contextual Facts (http://github.com/AndriesSHP/Gellish/blob/master/Data/Gellish%20Syntax%20and%20Contextual%20Facts%20-%20Oct2017.pdf)

Main classes of relations

- Entities are linked together in “relations”,
  at the level of both instances (individuals) and types (classes)

- Three groups of relations are distinguished
  Examples from BFO reference

1. Instance-level relations:
   - Your heart (instance-level) **continuant_part_of** your body **at** t
     - **Boldface** indicates a label for an **instance-level relation**!
   - Your heart beating (instance-level) **has_participant** your heart

2. Type-level relations:
   - human heart ***continuant_part_of*** human body
   - human heart beating process ***has_occurrent_part*** beat profile

3. Instance-type relations:
   - John’s heart *instance_of* human heart

Mereology and holarchy

- Parthood relations (or part-whole relationships)

- The axioms of **Minimal Extensional Mereology** (Simons P. Parts: A study in ontology. Clarendon Press, 1987):
  - Antisymmetry: If x part of y, then if y part of x, then $x = y$.
  - Transitivity: If x part of y, and y part_of z, then x **part_of** z.
  - Weak Supplementation: If x **part_of** y & not $x = y$, then there is some z such that (z **part_of** y and z has no part in common with x).
  - Unique Product: If x and y have a part in common, then there is some unique z such that for all w (w is part of z if and only if (w is part of x and w is part of y)).

  - Additional relation “proper_part_of” is defined as x **proper_part_of** $y \equiv x$ **part_of** y & not $x = y$

- For us more important: Barton A., Toyoshima F., Vieu L., Fabry P., Ethier J.-F. The **Mereological Structure of Informational Entities**. Formal Ontology in Information Systems // 11th International Conference (FOIS 2020), pp.201-215, 2020 (http://hal.science/hal-03041523/document)

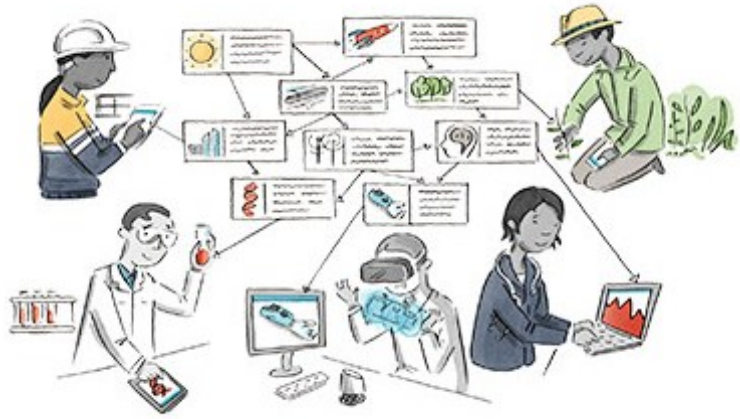
- **Holarchy** is a system of organization where entities, called holons, are both self-contained wholes and parts of larger wholes

Why we need ontology?

- We need to talk formally about domains:
  - Human ↔ Human
  - Computer ↔ Computer
  - Human ↔ Computer

- Applications:
  - Knowledge representation
  - Knowledge exchange and merging
  - Knowledge bases construction
  - Logic reasoning
  - Explanation of decisions
  - Knowledge gaps detection

- *Statistical Machine Learning* is a tool to “guessing” an answer
  - We need interpretation, explanation, harmonization, and tutoring!



© 2021 https://blog.metaphacts.com/visual-ontology-modeling-for-domain-experts-and-business-users-with-metaphactory

Natural languages in the land of LLMs

- Problem: context, discourse, narrative
  - Remember “literary criticism” as a humans’ activity!

- Pirozelli P., Câmara I. Natural Language at a Crossroads: Formal and Probabilistic Approaches in Philosophy And Computer Science. Manuscrito, 45(2), 2022 (http://www.scielo.br/j/man/a/8CcCj7HdRDXmmyZXpPZFp7t/)

- Vulich R. Frege’s Sharpness Requirement and Natural Language (http://cah.ucf.edu/fpr/article/freges-sharpness-requirement-and-natural-language/)

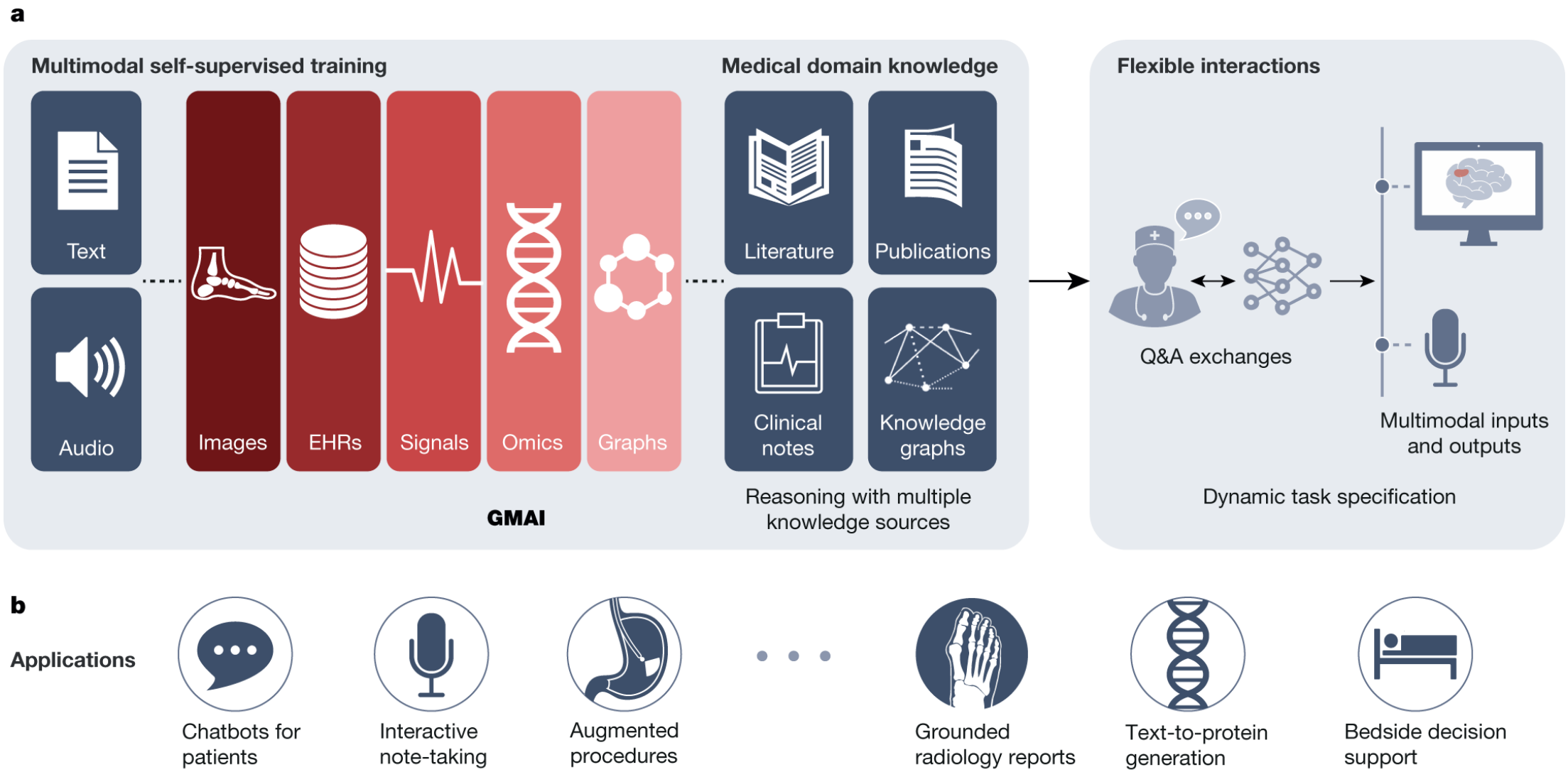
Temporal and geospatial relations

- Precedence relation and events
  - Temporal logics and time axis
- 4D life cycle of objects
  - New objects
  - New attributes
  - Object ↔ Attribute transformation
    - Level of details!
- Consequences:
  - *Static* vs *dynamic* data structures and **VERSIONING** of object-attribute data
  - 4D context is sufficient for temporal and geospatial relations – see below
- It's a very deep "rabbit hole"…
  - See Imaguire G. On the Ontology of Relations. Disputatio, 4(34), 2012, pp. 689-711

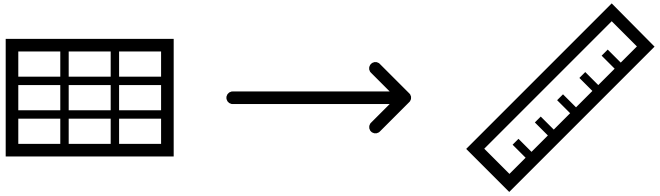
Natural language, modelling, and grounding

- Descriptions and representations
  - Notational complexity ↔ Conceptual complexity
- Physical/Chemical/… ontology and technology domain
- “The Map is Not the Territory”
  - Yudkowsky E. Map and Territory. Machine Intelligence Research Institute, 2018. 231 p.

Example of “Generalist medical AI”



(C) Foundation models for generalist medical artificial intelligence. 2023 (https://www.nature.com/articles/s41586-023-05881-4)



THE CONCEPT OF “MEASUREMENT”

- ✓ Representational measurement theory
- ✓ Context of value in object-attribute data representation
- ✓ Scales
- ✓ ISO standards
- ✓ Ontologies of units of Measure

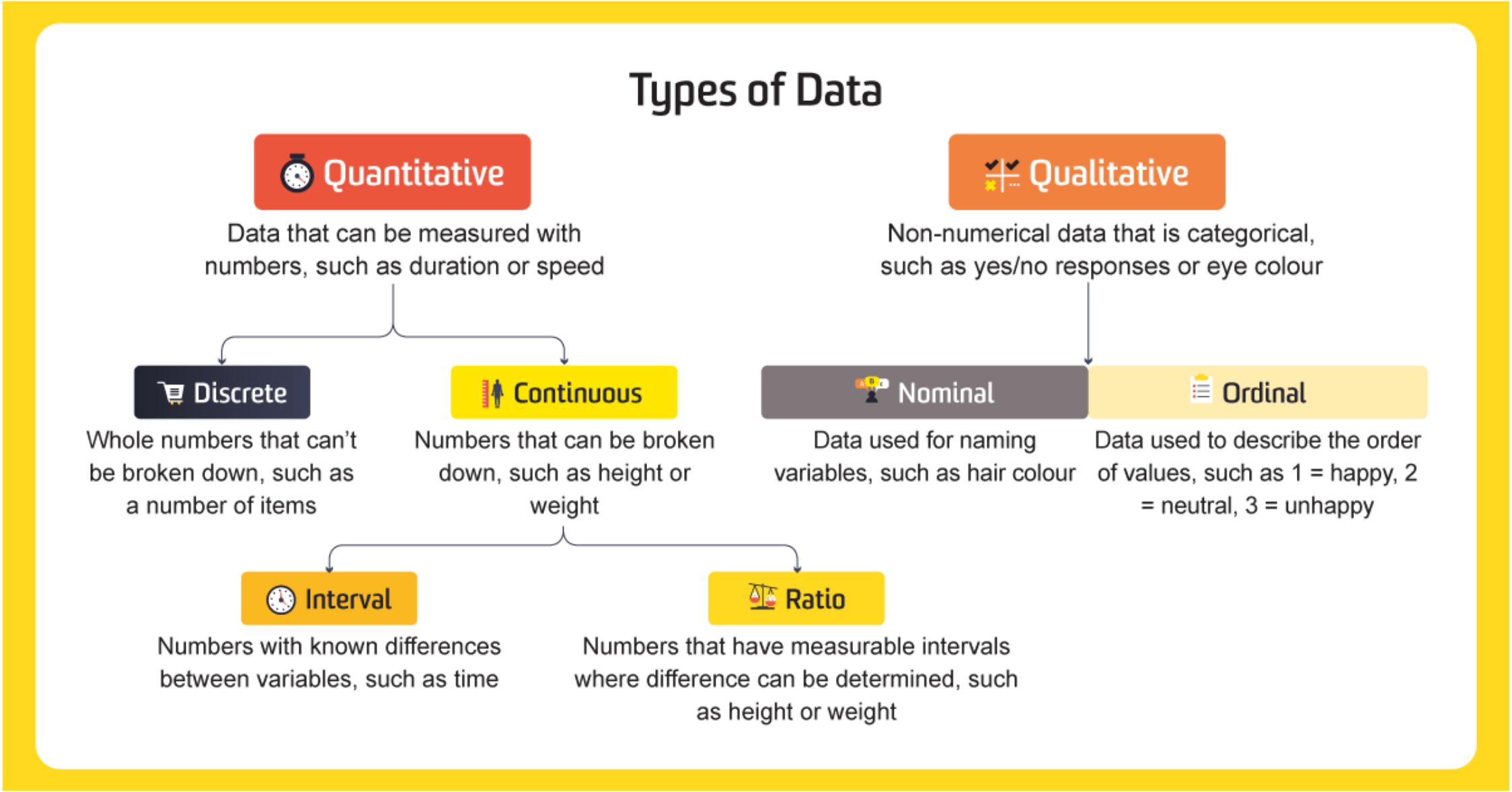
Foundations of measurements

- “**Atribute value**” as a result of a measurement!
  - Each measurement has some **context** – a rich description of the circumstances of a measure
    - Repeated measures design – a research design that involves multiple measures of the same variable taken on the same or matched subjects either under different conditions or over two or more time periods → statistics
      - Accuracy is how close a given set of measurements (observations or readings) are to their true value
      - Precision is how close the measurements are to each other

- **Measurement theory** is the study of how numbers are assigned to objects and phenomena

- **Representational measurement theory** (RMT)
  - Michell J. Representational measurement theory: Is its number up? / Theory & Psychology, 31(1), 2021, pp. 3-23. (http://journals.sagepub.com/doi/10.1177/0959354320930817)
  - Benoit E. New scale classification within the representational theory of measurement / Journal of Physics: Conference Series, 459, 2013 (http://iopscience.iop.org/article/10.1088/1742-6596/459/1/012004/pdf)
  - Heilmann C. A New Interpretation of the Representational Theory of Measurement / Philosophy of Science, 82(5), 2015, pp.787-797 (http://philsci-archive.pitt.edu/11009/1/RTM-Heilmann-PSA.pdf)

Data and scales



Main kinds of measurement scales

- Types of data & the scales of measurement (http://studyonline.unsw.edu.au/blog/types-of-data)
  - Properties of Measurement
    - **Identity**: Identity refers to each value having a unique meaning
    - **Magnitude**: Magnitude means that the values have an ordered relationship to one another, so there is a specific order to the variables
    - **Equal intervals**: Equal intervals mean that data points along the scale are equal, so the difference between data points one and two will be the same as the difference between data points five and six
    - **A minimum value of zero**: A minimum value of zero means the scale has a true zero point
      - Degrees, for example, can fall below zero and still have meaning, but if you weigh nothing, you don't exist

- Four kinds of measurement scales:

Scale	Basis empirical operation	Mathematical group structure	Admissible transformation
Nominal	determination of equality	Permutation group	$y = f(x)$, f is a bijection
Ordinal	+ determination of greater or less	Isotonic group	$y = f(x)$, f is a monotonic increasing function
Interval	+ determination of equality of intervals	General linear group	$y = ax + b,\ a > 0$
Ratio	+ determination of equality of ratio	Similarity group	$y = ax,\ a > 0$

Four main kinds of scales

The Four Scales of Measurement

Nominal Scale

Used for naming variables in no particular order

For example, eye colour

Ordinal Scale

Used for variables in ranked order, but the difference between is not determined

For example, #1 happy, #2 neutral, #3 unhappy

Interval Scale

Used for numerical variables with known equal intervals of the same distance

For example, time

Ratio Scale

Used for variables on a scale that have measurable intervals

For example, weight

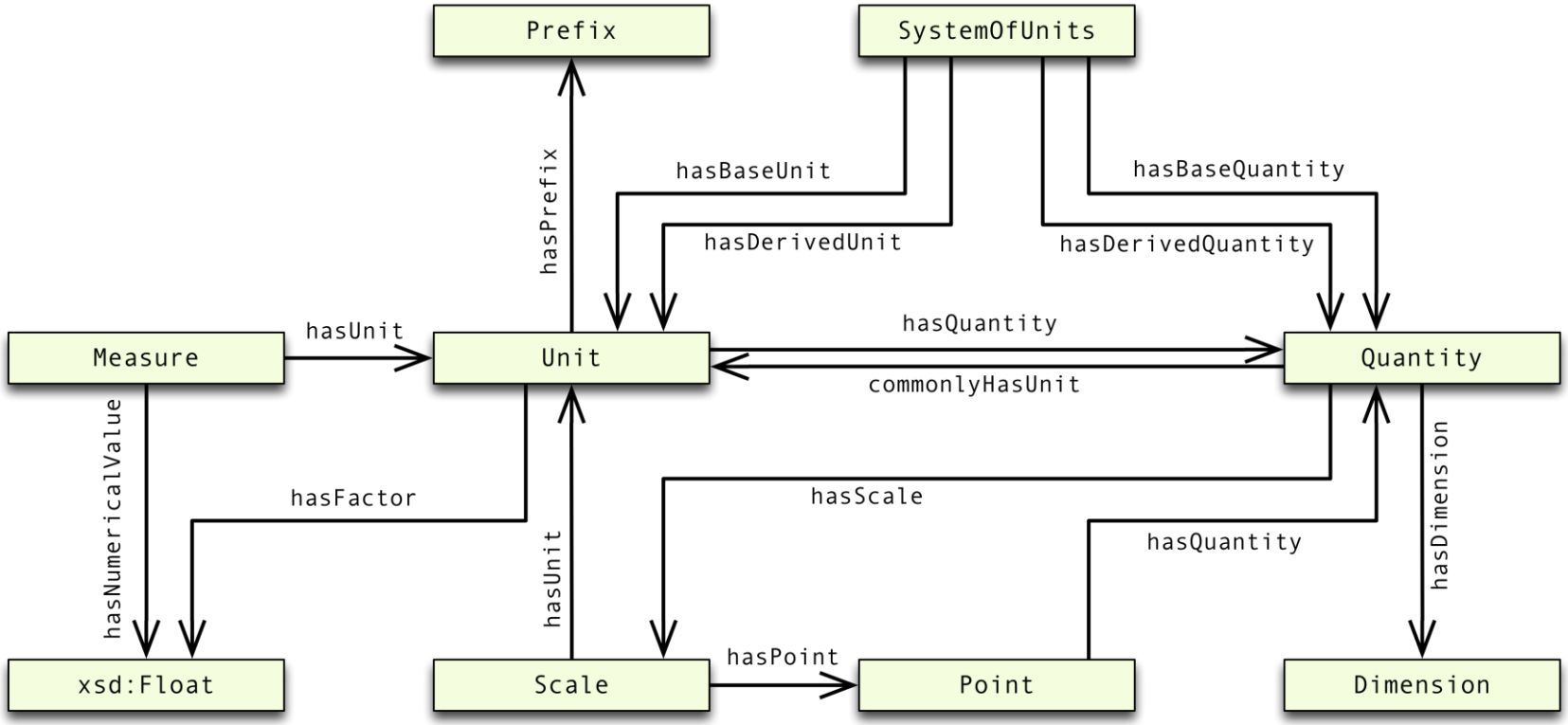
Main ISO standard for quantities and units

- ISO **80000**-1:2022 Quantities and units — Part 1: **General** (http://www.iso.org/standard/76921.html)
- ISO 80000-2:2019 Quantities and units — Part 2: **Mathematics** (http://www.iso.org/standard/64973.html)
- ISO 80000-3:2019 Quantities and units — Part 3: **Space and time** (http://www.iso.org/standard/64974.html)
- ISO 80000-4:2019 Quantities and units — Part 4: **Mechanics** (http://www.iso.org/standard/64975.html)
- …
- ISO 80000-12:2019 Quantities and units — Part 12: **Condensed matter physics** (http://www.iso.org/standard/63480.html)

- IEC 80000-13:2008 Quantities and units — Part 13: **Information science and technology** (http://www.iso.org/standard/31898.html)

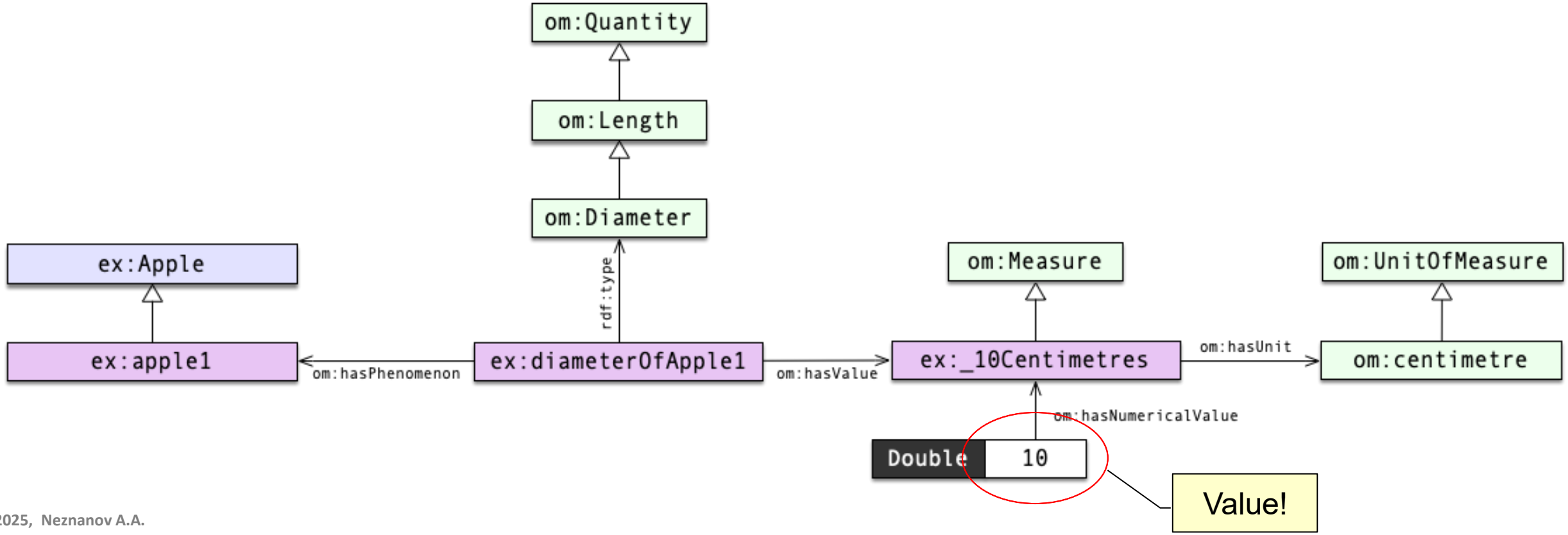
Ontologies of units of Measure – OM

- [GitHub] HajoRijgersberg/OM (http://github.com/HajoRijgersberg/OM)
  - Current version – 2.0.55
  - Classes:814, individuals: 2,246, properties: 34



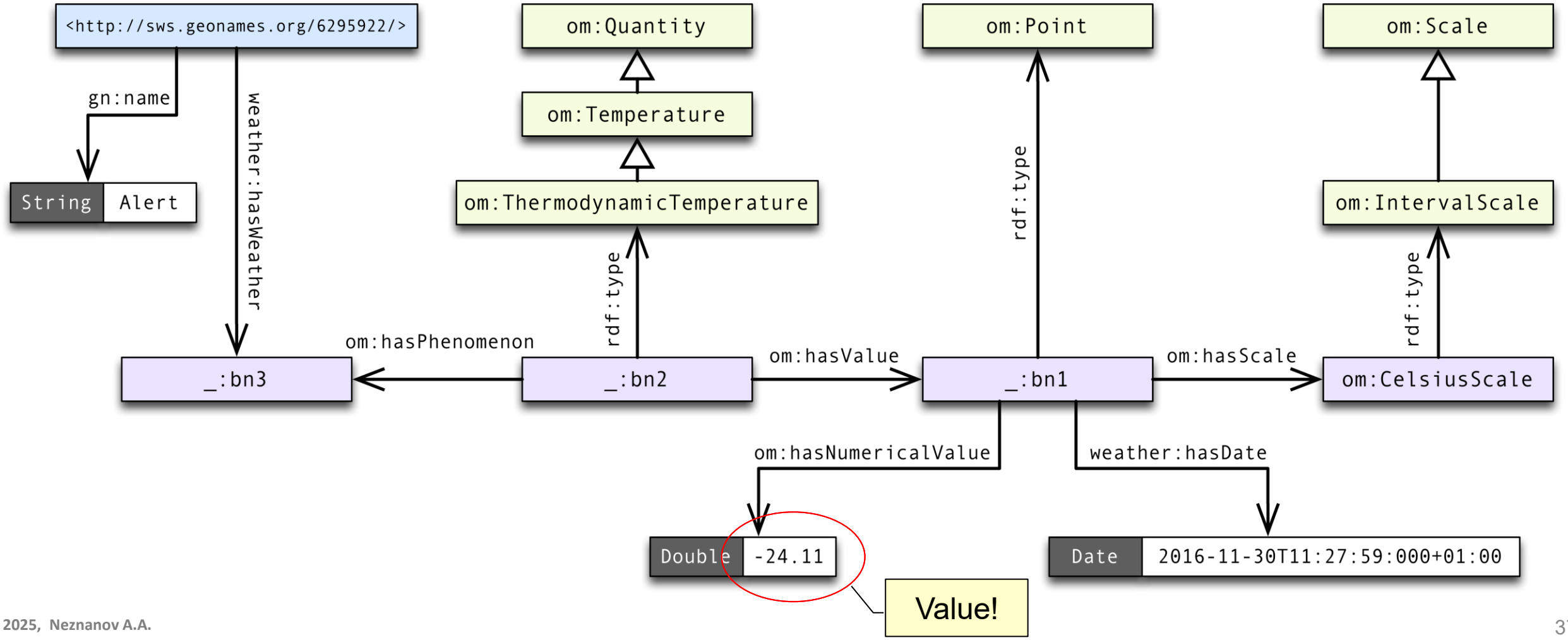
Example of measurement (OM): the diameter of an apple

- Representation of the “concrete act of measurement” with **references** (and grounding!) to:
  - Ontology of units of Measure («centimeter», «diameter»)
  - Ontology of the subject area («apple»)



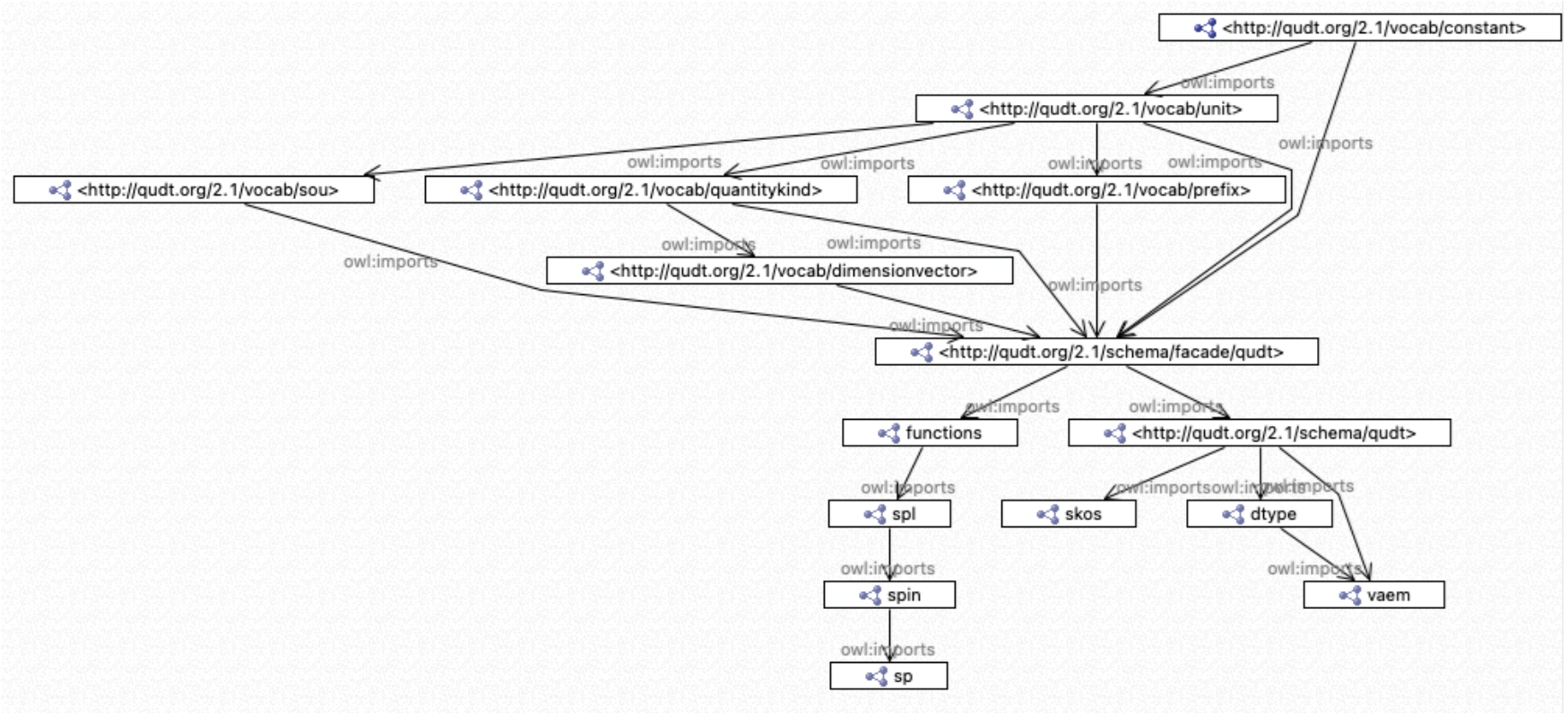
Example of measurement (OM): temperature

- The temperature at Alert, Canada on November 30th, 2016 at 11:28 AM



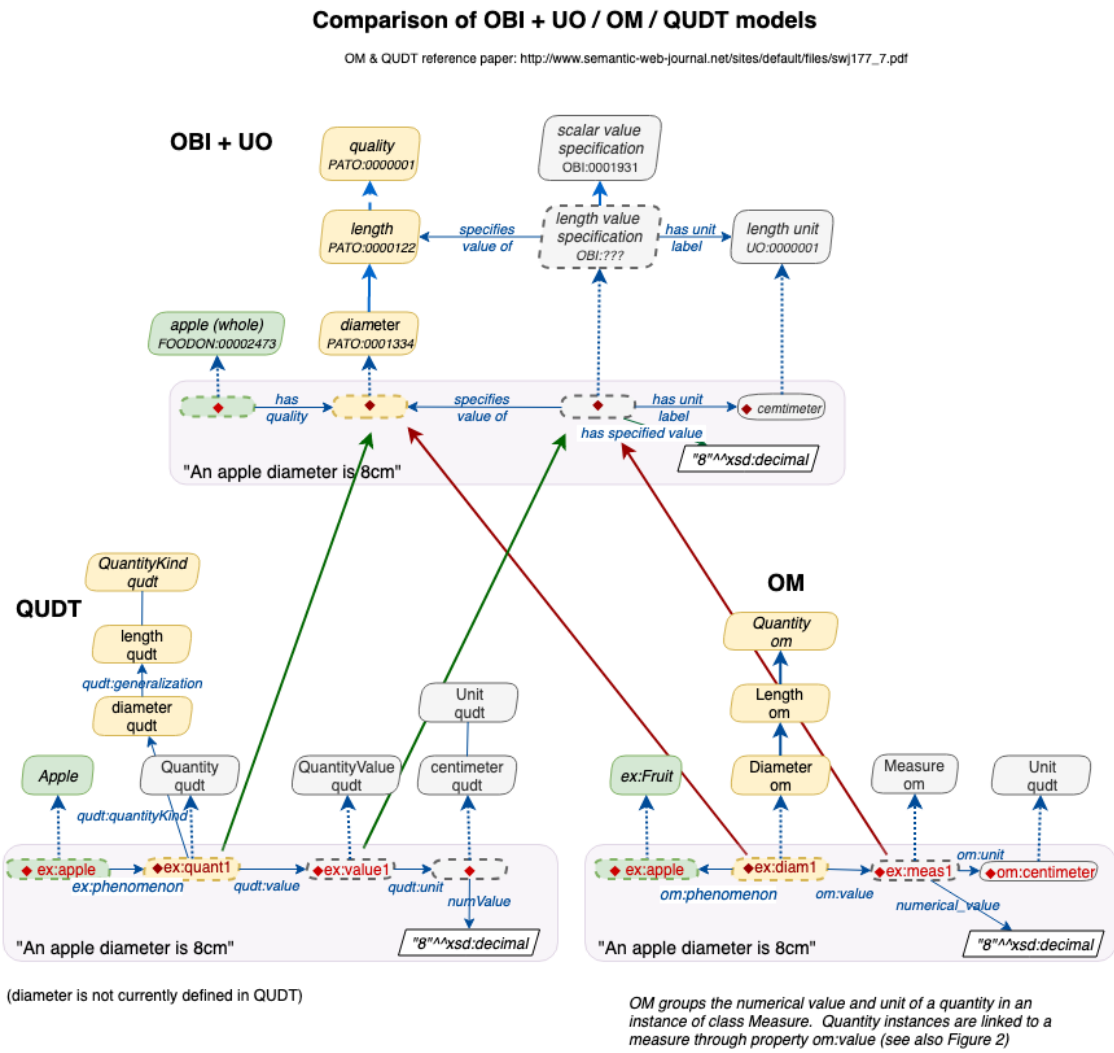
Ontologies of units of Measure – QUDT

- QUDT - **Q**uantities, **U**nits, **D**imensions and Data **T**ypes Ontologies
  - QUDT Units Version 2.1 (http://www.qudt.org/doc/DOC_VOCAB-UNITS.html)
    - Base ontology – classes: 133, individuals: 84, properties: 248
    - + 2509 instances



Ontologies of units of Measure – approaches and comparing

- LinkML – How to model quantities and measurements (http://linkml.io/linkml/howtos/model-measurements.html)

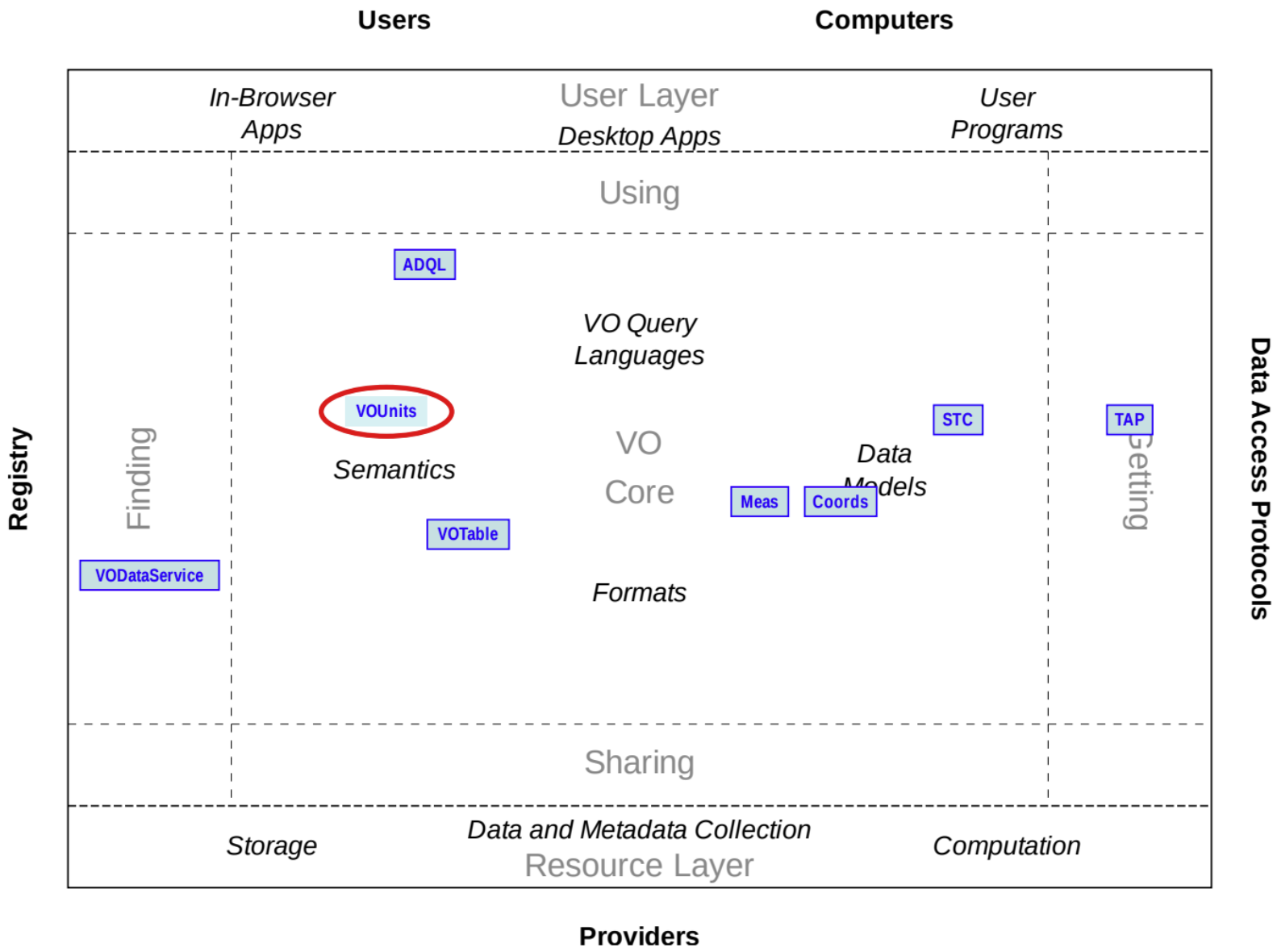


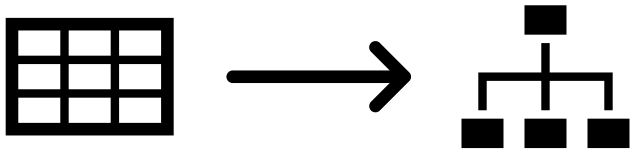
Ontologies of units of Measure – Specific symbols

- Unit designations from International Virtual Observatory Alliance
  - **Units in the VO** (http://www.ivoa.net/documents/VOUnits/index.html)
    - Version 1.1, IVOA Recommendation 2023-12-15
    - Very convenient standard, agreed with BIPM, ISO/IEC, IAU
  - Example:

m (metre)	**g** (gram)	**J** (joule)	**Wb** (weber)
s (second)	**rad** (radian)	**W** (watt)	**T** (tesla)
A (ampere)	**sr** (steradian)	**C** (coulomb)	**H** (henry)
K (kelvin)	**Hz** (hertz)	**V** (volt)	**lm** (lumen)
mol (mole)	**N** (newton)	**S** (siemens)	**lx** (lux)
cd (candela)	**Pa** (pascal)	**F** (farad)	**Ohm** (ohm)

VOUnits as a core building block in the VO:





FORMALIZATION OF CONCEPTS

- ✓ Partially Ordered Sets
- ✓ Formal Concept Analysis
  - ✓ Objects and attributes
  - ✓ Formal context and formal concept
  - ✓ Extension → extent, intension → intent
  - ✓ Formal concept lattice and line diagrams
- ✓ Implications in context and association rules

Formal Concept Analysis

- Historically, **Formal Concept Analysis** (FCA) is a branch of applied lattice theory that appeared in 1980's
  - Barbut M., Monjardet B. Ordre et classification: algèbre et combinatoire. Hachette, 1970.
  - Davey B.A., Priestley H.A. Introduction to lattices and order. Cambridge University Press, 1990.
  - **Ganter B., Wille R. Formal concept analysis. Springer, 1999.**
  - Kuznetsov S.O. On stability of a formal concept / Annals of Mathematics and Artificial Intelligence, 49(1–4), 2007, pp. 101-115.
  - Belohlávek R. Relational data, formal concept analysis, and graded attributes / Handbook of research on fuzzy information processing in databases. IGI Global, 2008, pp. 462-489.
  - Poelmans J., Ignatov D.I., Kuznetsov S.O., Dedene G. Formal concept analysis in knowledge processing: a survey on applications / Expert Systems with Applications 40(16), 2013, pp. 6538-6560.
  - Poelmans J., Ignatov D.I., Kuznetsov S., Dedene G. Fuzzy and rough formal concept analysis: a survey / International Journal of General Systems, 43(2), 2014, pp. 105-134.
  - Codocedo V., Napoli A. Formal concept analysis and information retrieval – a survey // Proceedings of the 13th international conference on formal concept analysis (ICFCA). LNCS, vol 9113, Springer, 2015, pp. 61-77.
  - **Ganter B., Obiedkov S.A. Conceptual exploration. Springer, 2016.**
  - Belohlavek R., Trnecka M. Basic level of concepts in formal concept analysis: formalization and utilization / International Journal of General Systems, 49(7), 2020, pp. 689-706.
  - Ferré S., Huchard M., Kaytoue M., Kuznetsov S.O., Napoli A. Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing / A Guided Tour of Artificial Intelligence Research. Volume II: AI Algorithms. Springer, 2020.
  - Roscoe S., Khatri M., Voshall A., Batra S., Kaur S., Deogun J. Formal Concept Analysis Applications in Bioinformatics / ACM Computing Surveys, 55(8), 2022, pp. 1-40.

Formal context

- Formal context – triple $\langle G, M, I \rangle$, where:
  - G – a set of objects
  - M – a set of attributes
  - $I \subseteq G \times M$ – binary relation between G and M

- The statement $(g, m) \in I$ (also denoted by gIm) is interpreted as "object g **has attribute** m"
  - Remember algebraic definition of "graph"!

- Cross-table representation:

I	y_1	y_2	y_3	y_4
x_1	×	×	×	×
x_2	×		×	×
x_3		×	×	×
x_4		×	×	×
x_5	×			

Formal concept

- With $A \subseteq G$ and $B \subseteq M$ in formal context $\langle G, M, I \rangle$:
- Operator $(\cdot)'$ define a **Galois connection** between the powersets $\left(2^G, \subseteq\right)$ and $\left(2^M, \subseteq\right)$:
  - $A' = \{m \in M | \forall g \in A: gIm\}$
  - $B' = \{g \in G | \forall m \in B: gIm\}$
    - Three properties: (1) $Z_1 \subseteq Z_2 \Rightarrow Z_1' \supseteq Z_2'$, (2) $Z \subseteq Z''$, (3) $Z''' = Z'$
- Formal concept – pair (A, B), such that $A' = B$ & $B' = A$ (*duality*!)
  - Set A is called the "**extent**" (to the *set of instances* of the concept)
  - Set B is called the "**intent**" (corresponds to the *description* of the concept)

- Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \ (\Leftrightarrow B_2 \subseteq B_1)$
- With respect to this partial order, the set of all formal concepts forms a complete lattice called the "**concept lattice**" of $\langle G, M, I \rangle$

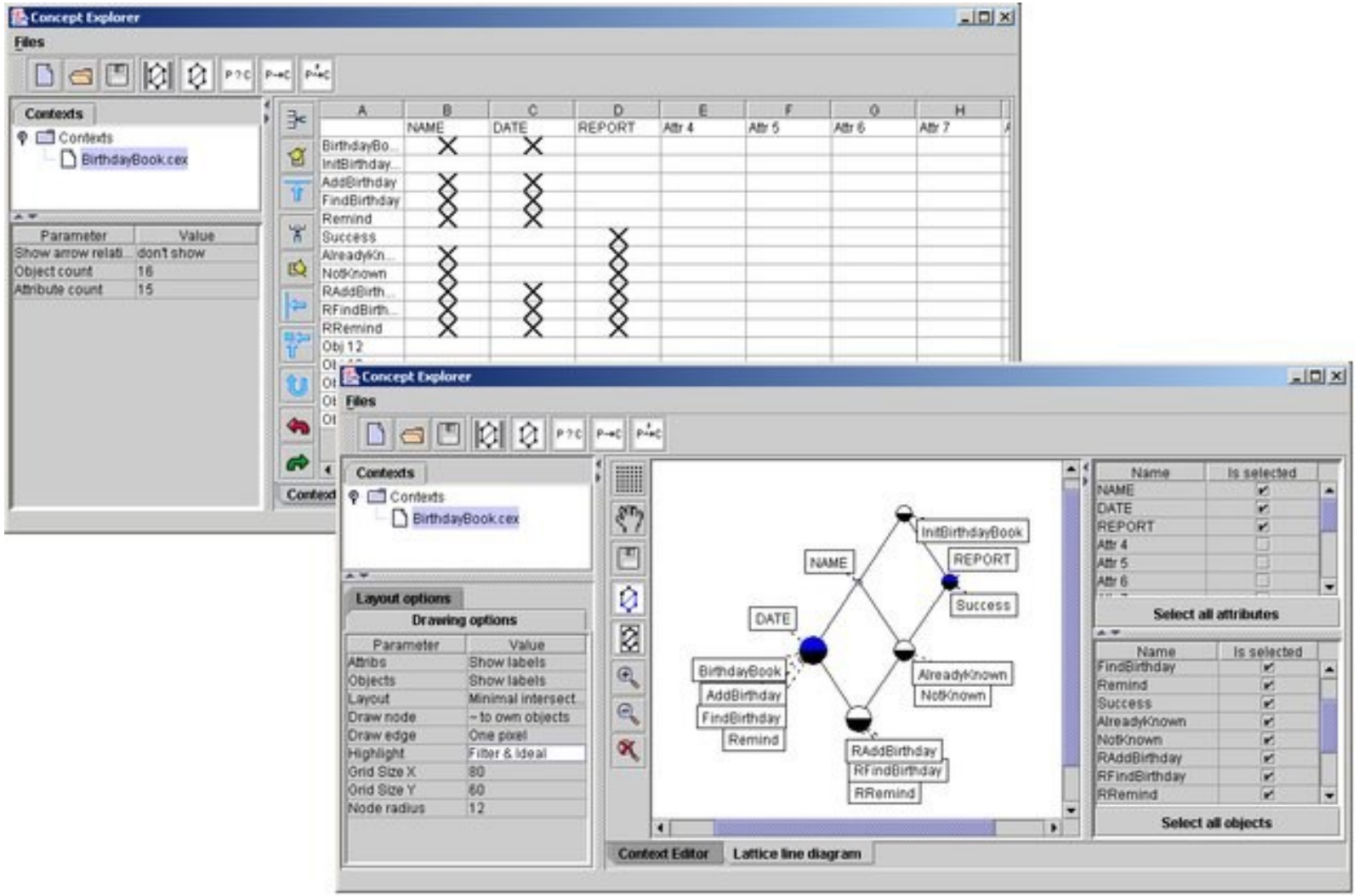
Implications and rules

- **Attribute implication** $A \Rightarrow B$ holds if $A' \subseteq B'$
- **Association rule** $A \rightarrow B$ is valid if:
  - Its "**support**" $s = \frac{|(A \cup B)'|}{|G|} = \frac{|(A' \cap B')|}{|G|} \geq \sigma_s$
  - Its "**confidence**" $c = \frac{|(A \cup B)'|}{|A'|} = \frac{|(A' \cap B')|}{|A'|} \geq \sigma_c$
  - Where σ_s and σ_c are user defined thresholds

- Attribute implication is an association rule with c = 1
- Implications obey Armstrong rules (see below in RM topic)

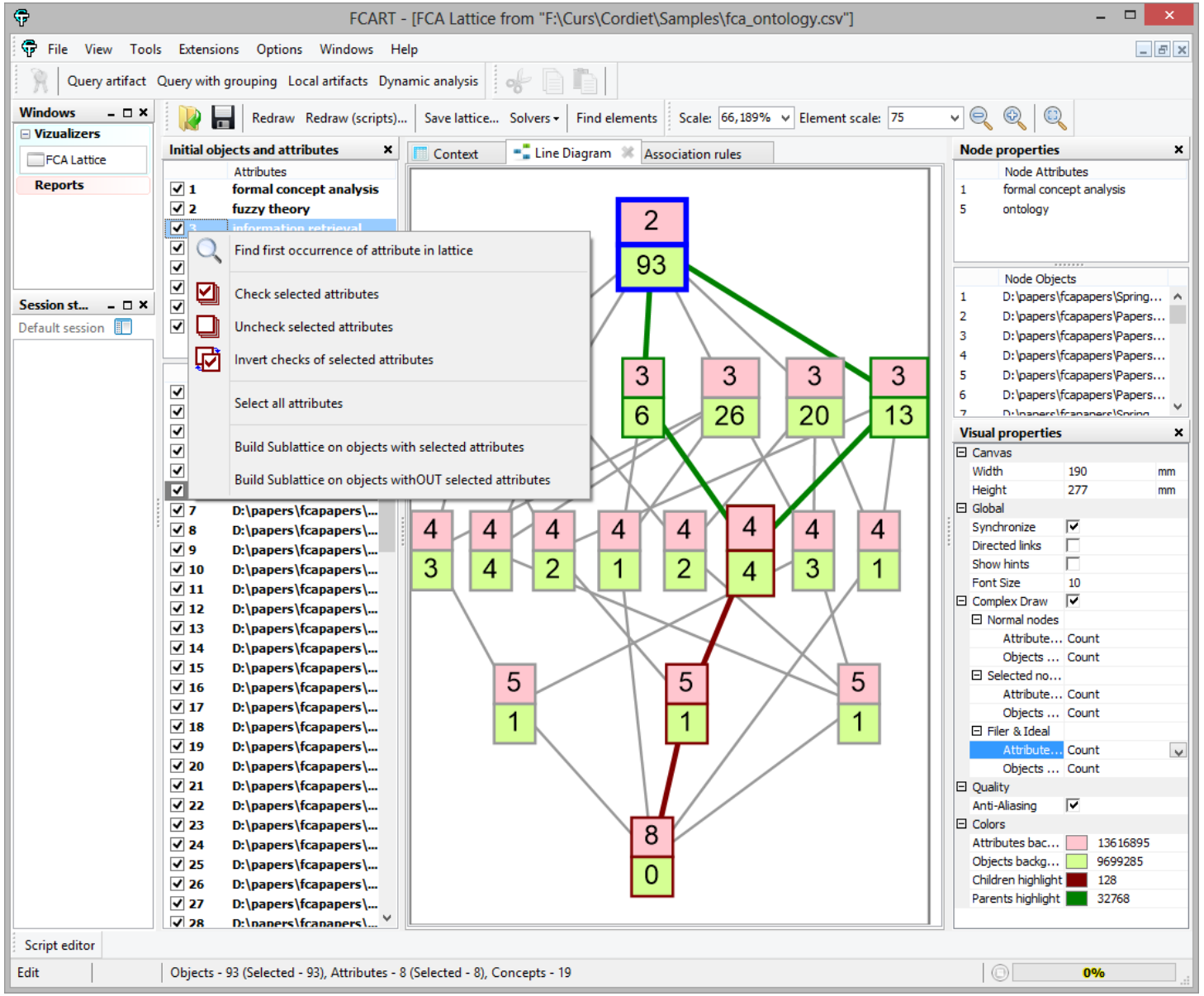
Line (Hasse) diagrams (1) – ConExp

- Canonical representation in The Concept Explorer (ConExp) by Serhiy Yevtushenko



Line (Hasse) diagrams (2) – FCART

- FCART by Alexey Neznanov

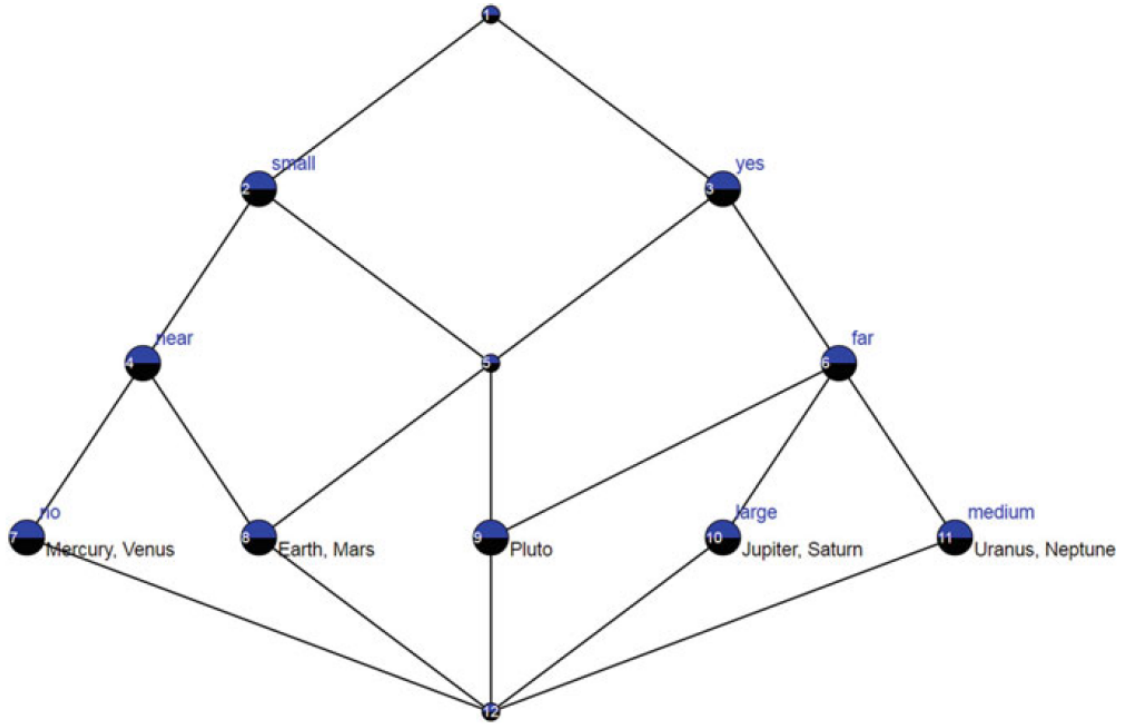


Binary attributes and scaling

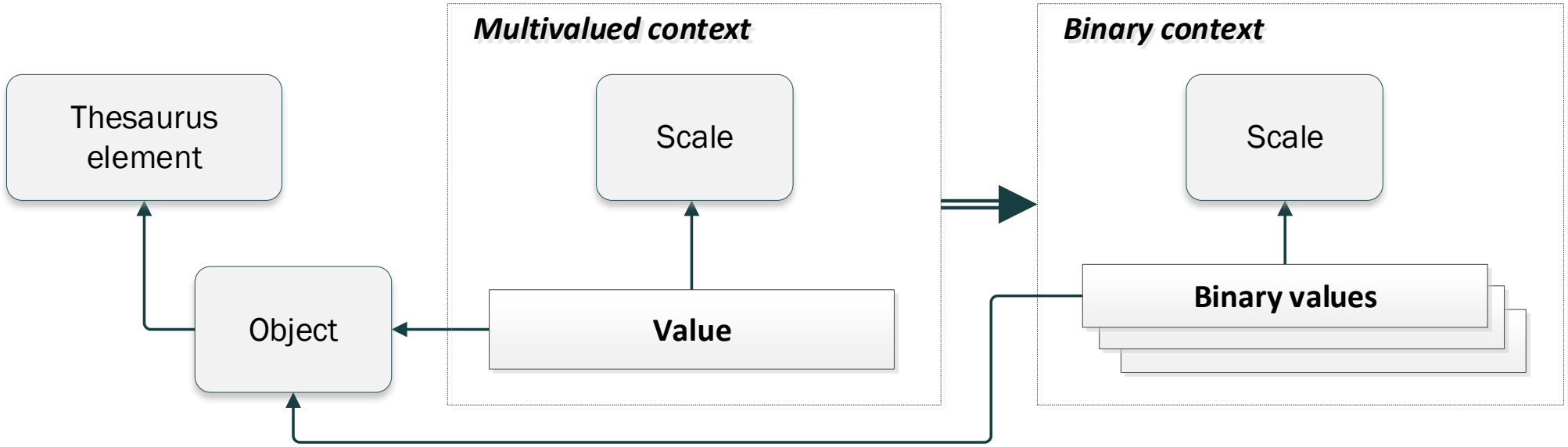
- Example from Davey and Priestley (1990)
  - Planets and their properties

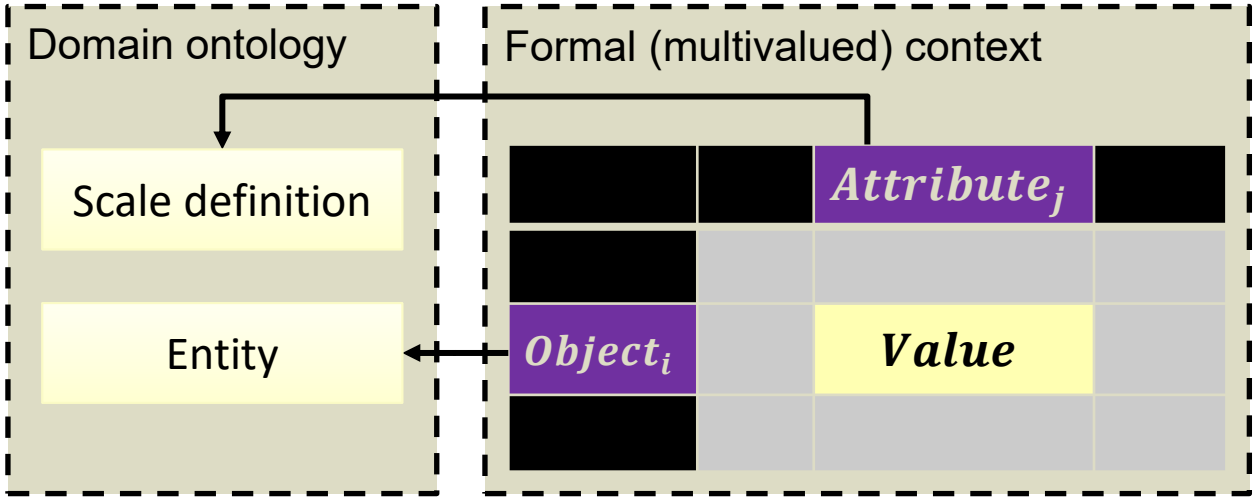
- Scaling of attributes:
  - Size: diameter, m => (“small”, “medium”, “large”)
  - Distance to Sun: distance, m => (“near”, “far”)
  - Moon(s): predicate, Boolean => (“yes”, “no”)

Planets	Size			Distance to Sun		Moon(s)	
	small	medium	large	near	far	yes	no
Jupiter			x		x	x	
Mars	x			x		x	
Mercury	x			x			x
Neptune		x			x	x	
Pluto	x				x	x	
Saturn			x		x	x	
Earth	x			x		x	
Uranus		x			x	x	
Venus	x			x			x



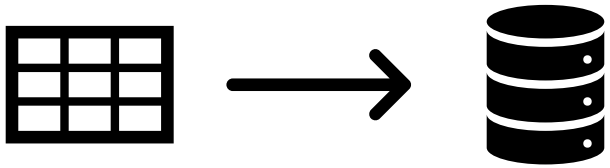
Scaling, ontologies, and context tracking





Exceptions and sources

- Priss U. Formal Concept Analysis Homepage (http://www.upriss.org.uk/fca/fca.html)
- Kaytoue M. An Introduction to Formal Concept Analysis. 2013 (http://perso.liris.cnrs.fr/mehdi.kaytoue/teaching/fca/fca-29-10-slides.pdf)



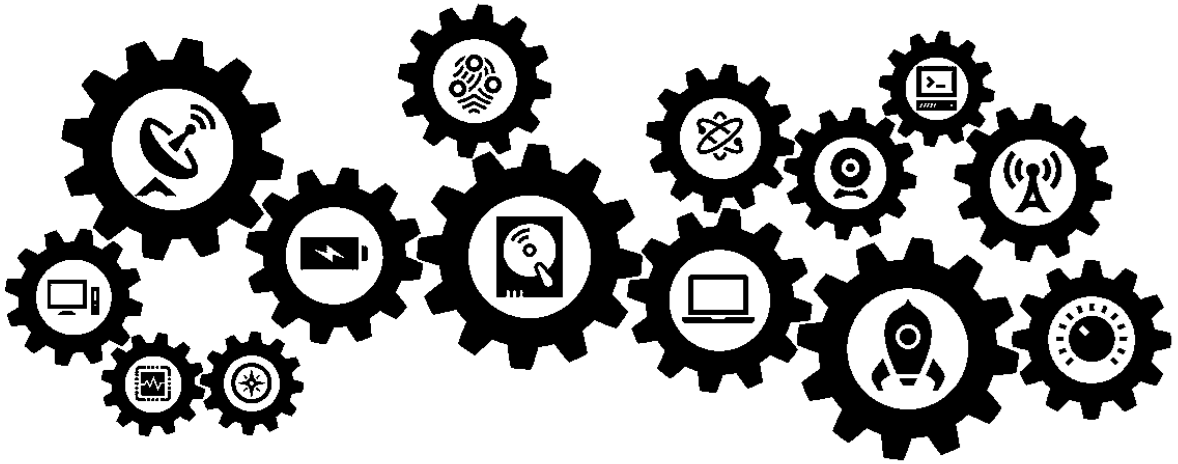
ENTITIES AND DATA MODELS

- ✓ Formal Concept Analysis
  - ✓ Objects and attributes
  - ✓ Formal context and formal concept
  - ✓ Extension → extent, intension → intent
  - ✓ Formal concept lattice and line diagrams
- ✓ Implications in context and association rules

Main tasks of data processing

- Information system (IS) as a data processing facility
  - Relevant knowledge about data → adequate data processing
  - Workflows and pipelines (another big lecture)

- Six main tasks:
  1. **Input/Output**
  2. **Format conversion** (sic!)
  3. **Telecommunication**
  4. **Storage**
  5. **Search (data retrieval)**
  6. **Manipulation**
     - Analysis!



Data and metadata

- *The knowledge* appears at the **metadata** level
  - Data about data ☺
  - What is Metadata (with examples) (http://dataedo.com/kb/data-glossary/what-is-metadata)
  - Dataedo – Data vs Metadata (https://landing.dataedo.com/data-vs-metadata)
- Metadata management
  - Full chapter in DMBoK (Chapter 12: Metadata Management)!
- Metadata formalization and standardization
- Metadata about metadata (infinite hierarchy)
  - Where can we stop? Metadata, **metametadata**, **metametametadata**, ...
  - The problem of harmonization of *ontologies* and *ontics* (interpretation)
  - The problem of interoperability
- Metadata and meta-modelling: modelling, notations, architecture
  - Model, metamodel, metametamodel, … (see UML meta-model, BPMN metamodel)

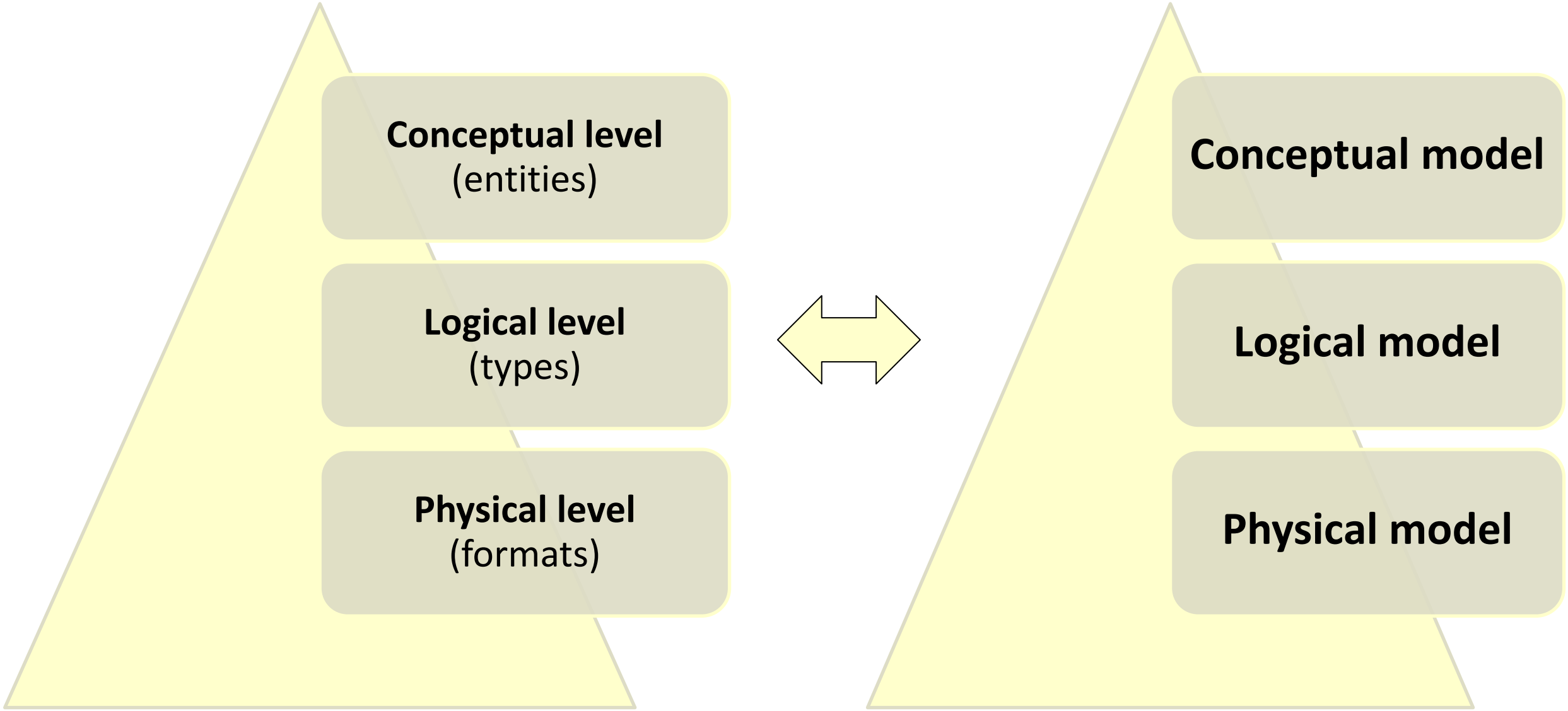
Object-oriented analysis and design

- **Object** – an arbitrary entity for which the boundaries, state and behavior are easy to define
- **Attribute** – an object property
  - The set of attribute values defines the object state
- **Method** – a variant of object behavior
  - The set of methods defines the object behavior
- **Interface of object** – the external representation of the object, i.e., all its attributes and methods
  - The interface description defines the object boundaries by the very fact of attributes and methods inclusion or non-inclusion in the interface
- **Class** – a description of objects of the same type, i.e. their common interface
  - A class can be represented as a template that describes the possible objects' states and behavior
  - The object that satisfies this pattern is called an instance of a class
  - All instances of a class have the same interface
  - The class declaration formalizes the boundaries, state and behavior discussed in the object definition above
  - ☑ In OO programming languages a class is a data type

Basic types of relationships in OO-analysis

- Relationships between objects and classes
  - **Instantiation** (“instance-of”) – relationship between a class instance and a class
- Relationships between classes
  - **Generalization** (“is-a”) – relationship between a successor and an ancestor (oriented from a successor to an ancestor)
    - In the opposite direction – **specialization** (from an ancestor to a successor)
    - ☑ **Inheritance** – a mechanism for implementing a generalization/specialization relationship
  - **Realization/implementation** – relationship between a class and an interface/abstract class
- Relationships between objects (usage)
  - **Association** – arbitrary “using” relationship between two or more objects in which the objects have their own lifetime and there is no “owner”
    - Binary associations can be unidirectional and bidirectional (symmetrical)
  - **Aggregation** (“part-of”) – relationship between a part and a whole (weak containment)
  - **Composition** – aggregation, when a part is integral (one “part” cannot belong to more than one “whole” and the lifetime of the “part” coincides with the lifetime of the “whole”) (strong containment)

Three levels of data models abstraction

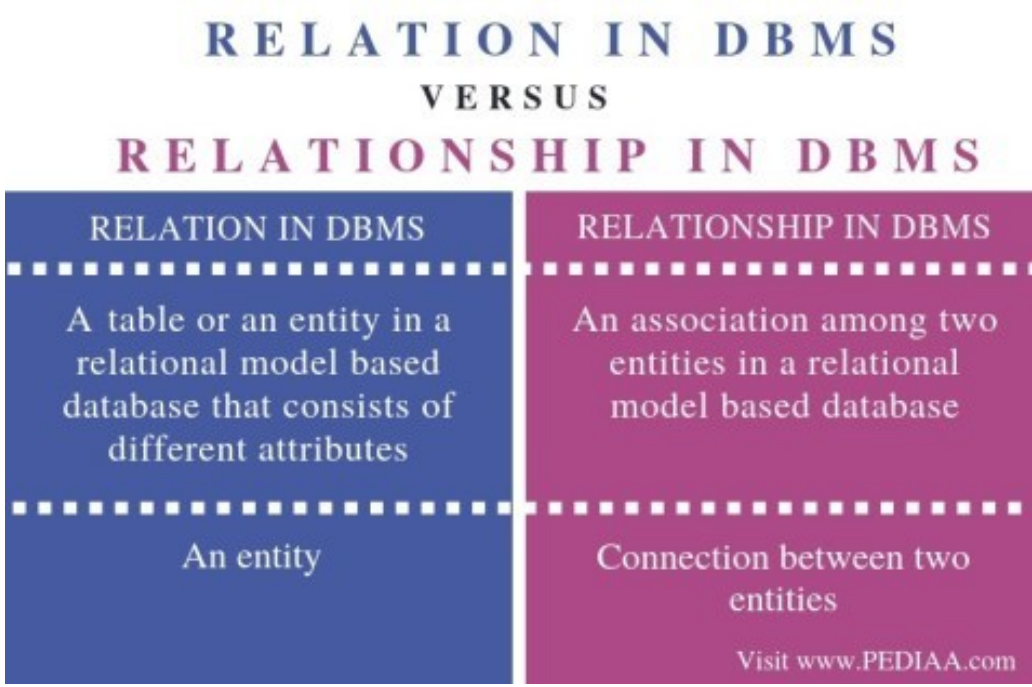


The note about relation(ship)

- Complex word usage!
  - Historically and traditionally:
    - Britannica: “Relations” and “Relationship“ (http://www.britannica.com/dictionary/eb/qa/relations-and-relationship)
    - BBC: Relative / relation – relationship (http://www.bbc.co.uk/worldservice/learningenglish/grammar/learnit/learnitv235.shtml)
    - Discussion (http://english.stackexchange.com/questions/15208/relation-versus-relationship)

- The main difference is that relationship is broader than relation
  - Both can mean “the way in which two or more concepts, objects, or people are connected”
  - In addition, relationship can mean “the state of being connected”

- Later we discuss additional senses!
  - ☑“Relation” in relational algebra



Entity-relationship model

- Peter Pin-Shan Chen, 1976
- **Entity-Relationship Diagrams** – visual language for conceptual data modelling
  - Standard abbreviations – **ER** and **ERD**
  - Both terms “entity” and “relationship” are incomprehensible and need clarification
    - Wikipedia is a very bad source here!
  - ☑ ER is a static model
    - We will discuss further both time consideration and special notations (UML, BPMN, etc.)
  - A huge number of extensions and variations:
    - Semantics extensions
    - Various “syntax sugar” for working with large diagrams
    - A plenty of visual images variations
    - Confusion with IDEF1X
      - See ISO 31320-2:2012 Information technology — Modeling Languages — Part 2: Syntax and Semantics for IDEF1X97 (IDEFobject) (http://www.iso.org/standard/60614.html)

ER origins and advancing

- Almost **fifty years** of development
  - Chen P. The Entity-Relationship Model-Toward a Unified View of Data. ACM Transactions on Database Systems, 1, 1976. pp. 9-36 (http://www.csc.lsu.edu/~chen/pdf/erd-5-pages.pdf, https://pdfs.semanticscholar.org/1348/1add6a6ce710af7004577ccb0ea7f92e887d.pdf)

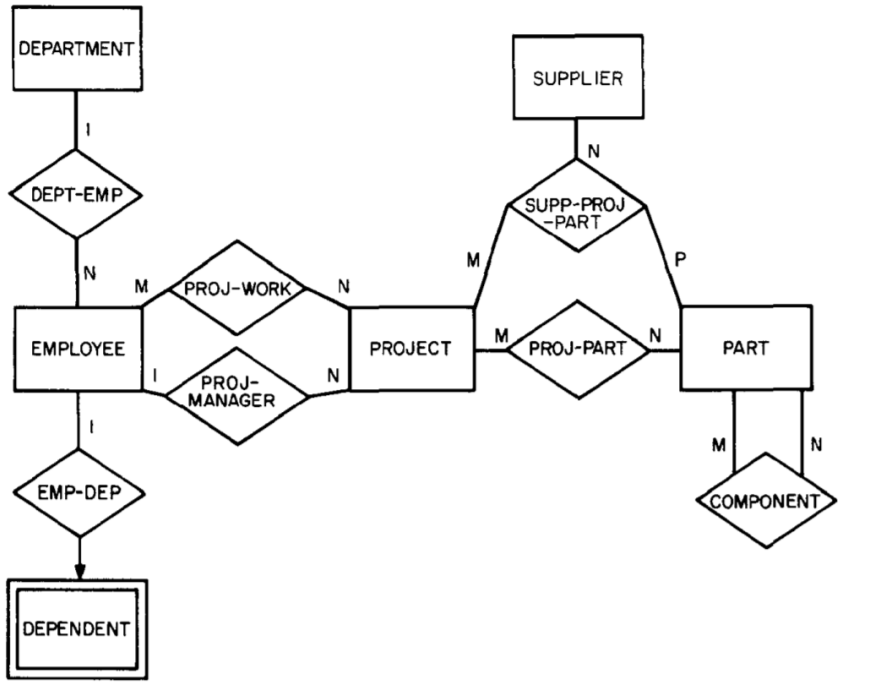
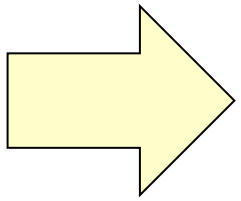
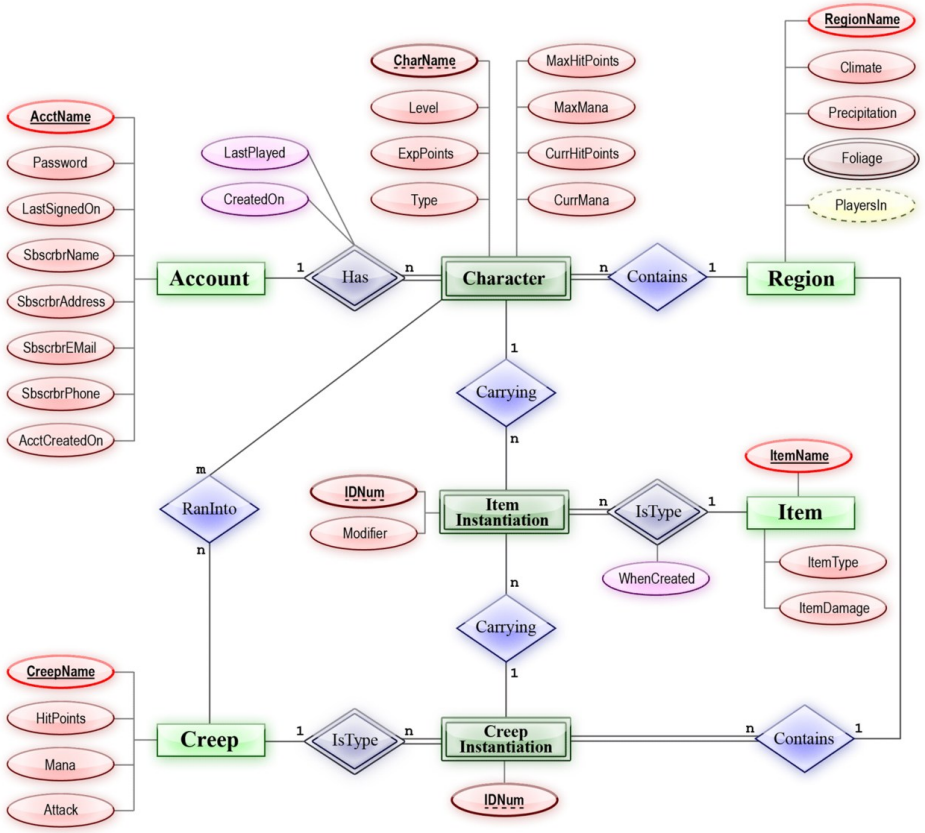


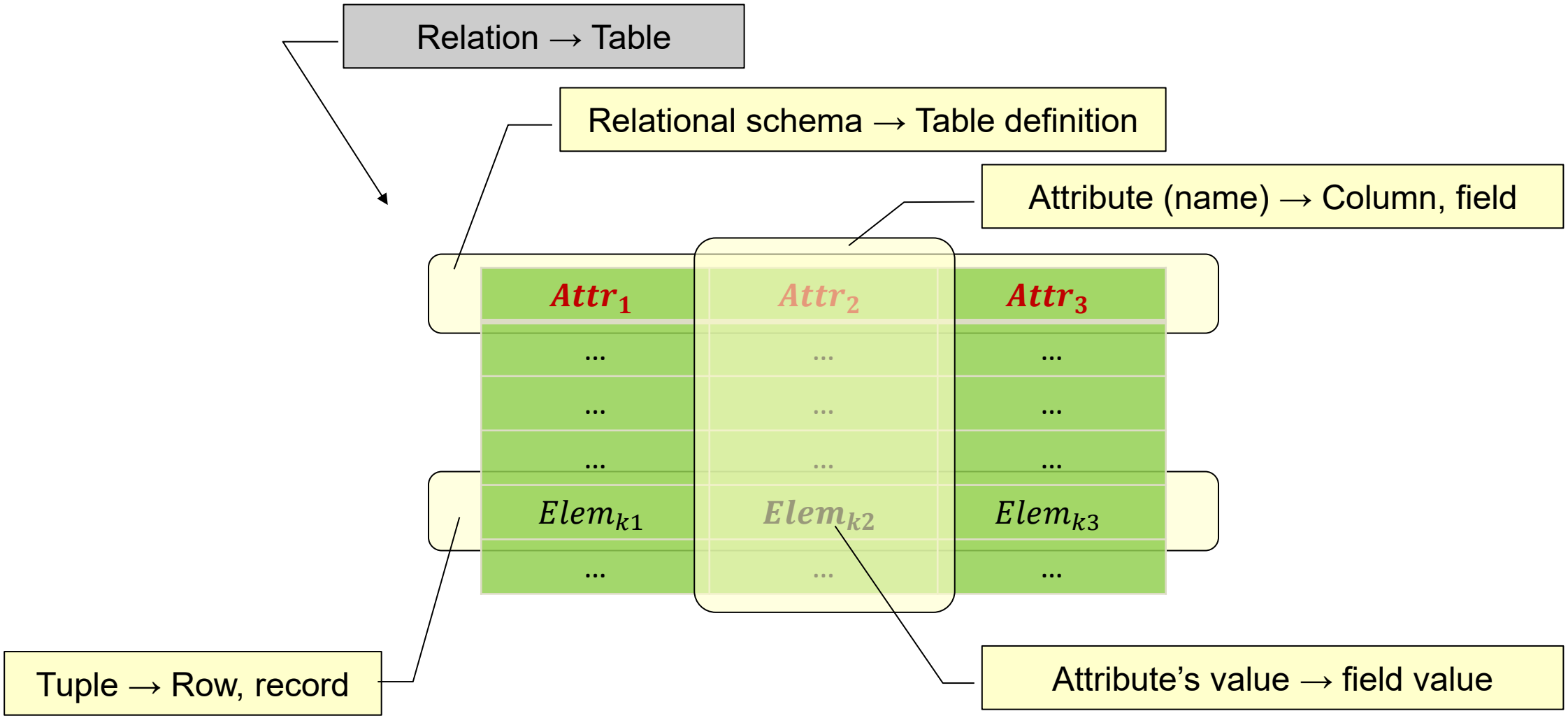
Fig. 11. An entity-relationship diagram for analysis of information in a manufacturing firm

ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976.



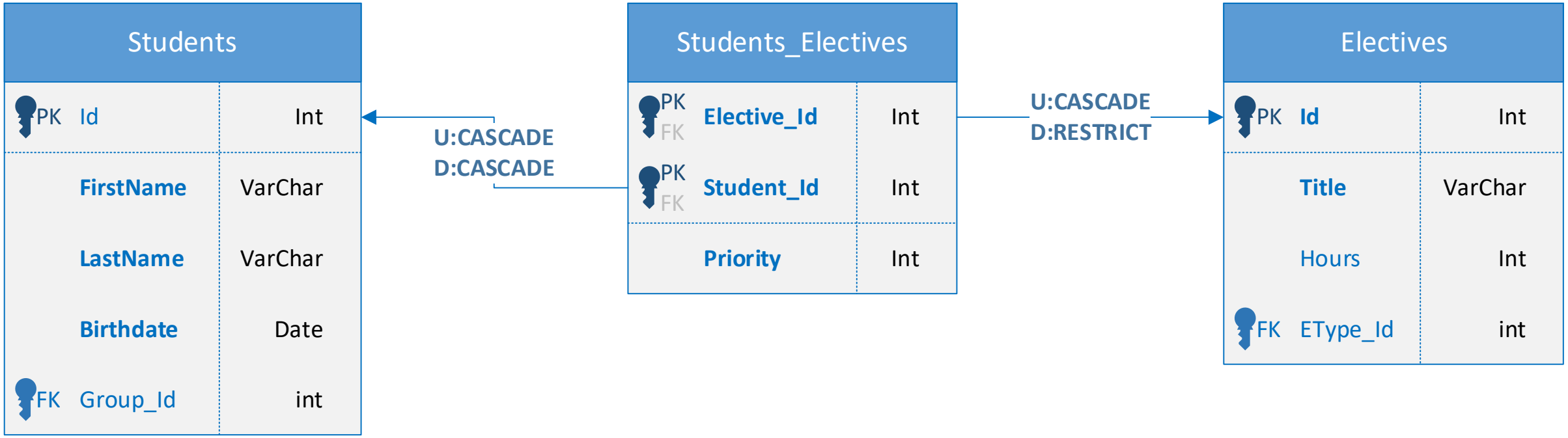


Relation → Table



RS/TR origins and advancing

- Relational schemas or table-reference diagrams
  - Codd, E. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 1970, pp. 377-387. (http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf)
  - Codd E.F. The relational model for database management: version 2. Addison-Wesley, 1990. 538 p.



Conceptual and logical levels of classical ER/TR approach

	ER Model	**Relational Model**
Level	Conceptual	Logical (representation or implementation)
Purpose	Shows real-world entities, their characteristics, and the relationships between them.	Shows data about objects in tables, and how the tables relate to each other.
Used by	For anyone who needs to see how the data model will be formed.	Mostly for data engineers and programmers who need to see where data will be stored.
Language and notations	Chen, Martin, UML, and other notations.	Relational/Schema diagrams, SQL.
Main components	Entities, attributes, relationships, arrows	Tables (relations), columns, domains, keys, records
Relationships	Relationships can be easily seen and identified through the use of arrows and symbols.	It is more difficult to determine the relationships between tables.
Mapping	Has the possibility to show cardinality mapping in explicit way.	Not possible to see cardinality mapping (only "foregn keys" with constraints).

SQL table and their rows VS query and their recordset

- A relation's **tuple** <> a **row** from a query result (output rows of a SQL SELECT statement)
  - Remember PKs and "DISTINCT" keyword!

- A recordset (or result set) is a data structure that consists of a group of records as the result of a query to a data source
  - Recordset is not a set (in set-theoretic sense)!
  - Recordset has local schema!
  - Recordset has ordering!
  - Recordset can have duplicates!

Dependency mining and relational constraints

- Dependency
  - Functional dependency (FD)
  - Approximate functional dependency (AFD)
  - Multi-valued dependency (MVD)
  - …
- Apparatus of dependencies was introduced by William W. Armstrong (Armstrong's Axioms)
  - Beeri, Fagin и Howard created apparatus of multi-valued dependencies
  - Hugh Darwen formulated General Unification Theorem
- Keys (PKs, FKs)!
- Normalization/**denormalization**!
  - How to build object-attribute dataset from relational model? See further!
- Data mining! …

Functional dependency mining

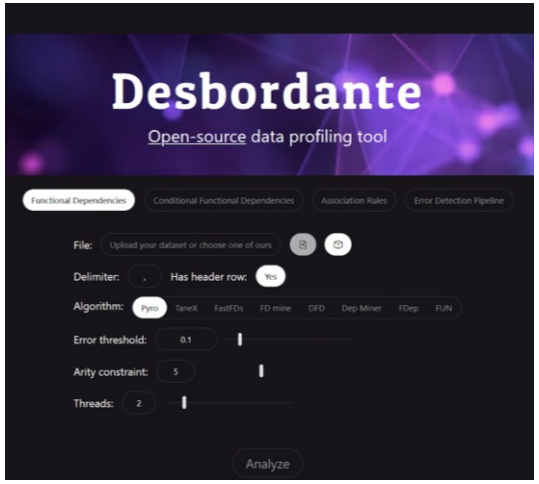
- **Functional dependency** (FD) – a relationship between two sets of attributes in a relation
  - $X \rightarrow Y$ – X determines Y, i.e. there is a many-to-one relationship between the values of the attribute set X and the values of the attribute set Y
  - X – determinant, Y – dependent set (or simply dependent)

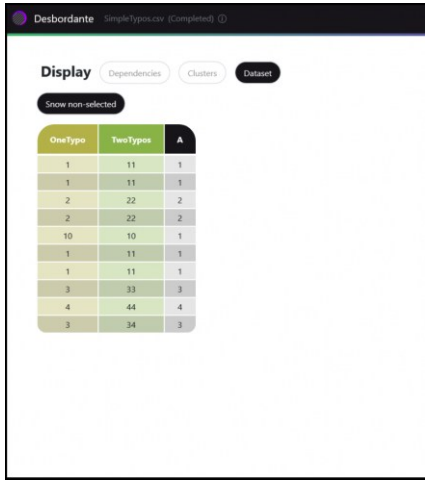
- Armstrong's Axioms:

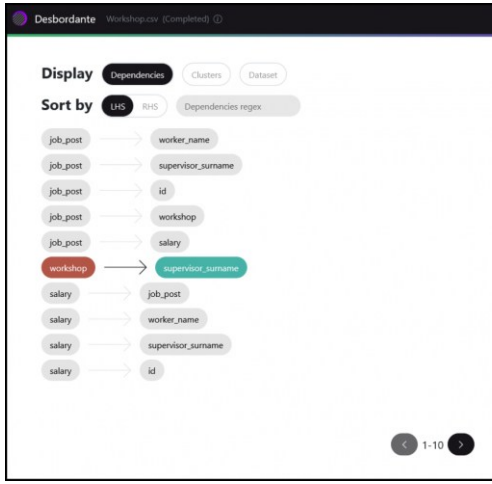
Axiom/inference rule	**Expression**					
Axiom of reflexivity	$X \rightarrow X$					
Axiom of augmentation	IF	$X \rightarrow Y$	THEN	$XZ \rightarrow Y$		
Axiom of transitivity	IF	$X \rightarrow Y$	AND	$Y \rightarrow Z$	THEN	$X \rightarrow Z$
Decomposition rule	IF	$X \rightarrow YZ$	THEN	$X \rightarrow Y$	AND	$X \rightarrow Z$
Union rule	IF	$X \rightarrow Y$	AND	$X \rightarrow Z$	THEN	$X \rightarrow YZ$
Composition rule	IF	$X \rightarrow Y$	AND	$Z \rightarrow V$	THEN	$XZ \rightarrow YV$
Pseudotransitivity rule	IF	$X \rightarrow Y$	AND	$YZ \rightarrow V$	THEN	$XZ \rightarrow V$

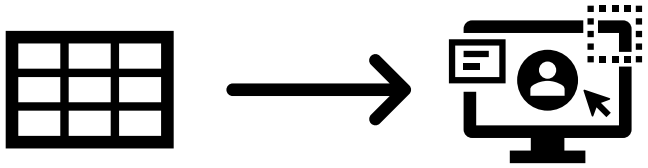
Dependency mining in practice

- **Desbordante** – a high-performance data profiler that is capable of discovering and validating many different patterns in data using various algorithms (http://desbordante.unidata-platform.ru)
- Data patterns:
  - Functional dependencies, both exact and approximate (discovery and validation)
  - Conditional functional dependencies (discovery)
  - Metric functional dependencies (validation)
  - Fuzzy algebraic constraints (discovery)
  - Unique column combinations (discovery and validation)
  - Association rules (discovery)







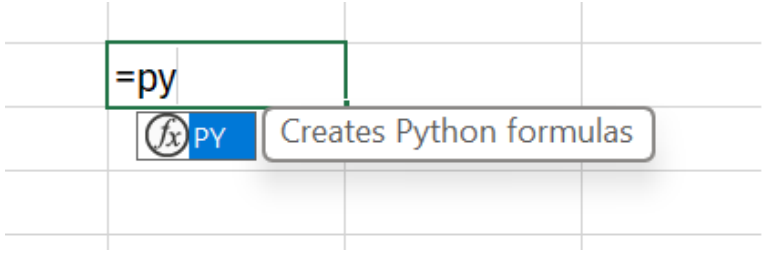


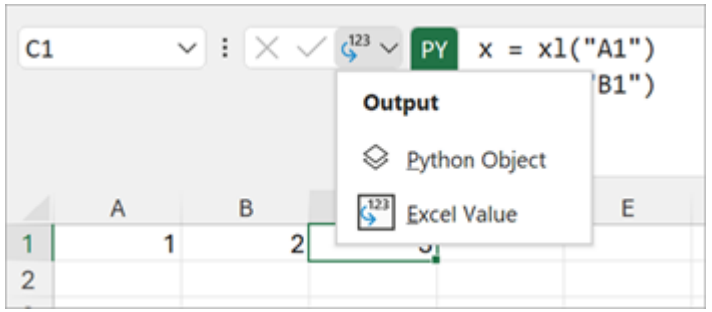
OBJECT-ATTRIBUTE DATA IN HCI AND UI

- ✓ HCI and UI
- ✓ Tables and spreadsheets
  - ✓ Pivot tables
- ✓ Rows and columns in UI
  - ✓ Implicit and explicit ordering

Tables in human-computer interaction

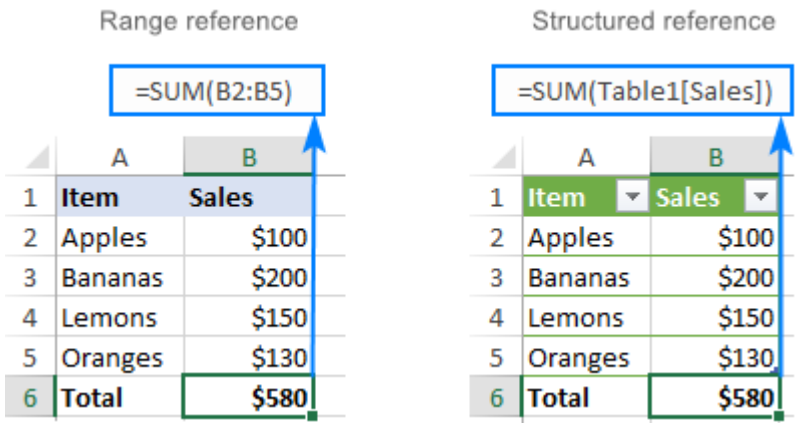
- **Spreadsheet** – electronic table with formulas and hyperlinks
- **Lotus 1-2-3** was arguably the application which made the IBM PC a success
- **Microsoft Excel** – industry standard
  - Hierarchy: Workbook → Worksheet → Range → Cell
  - Formula language
  - Now with Python ☺
    - (http://support.microsoft.com/en-gb/office/get-started-with-python-in-excel-a33fbcbe-065b-41d3-82cf-23d05397f53d)





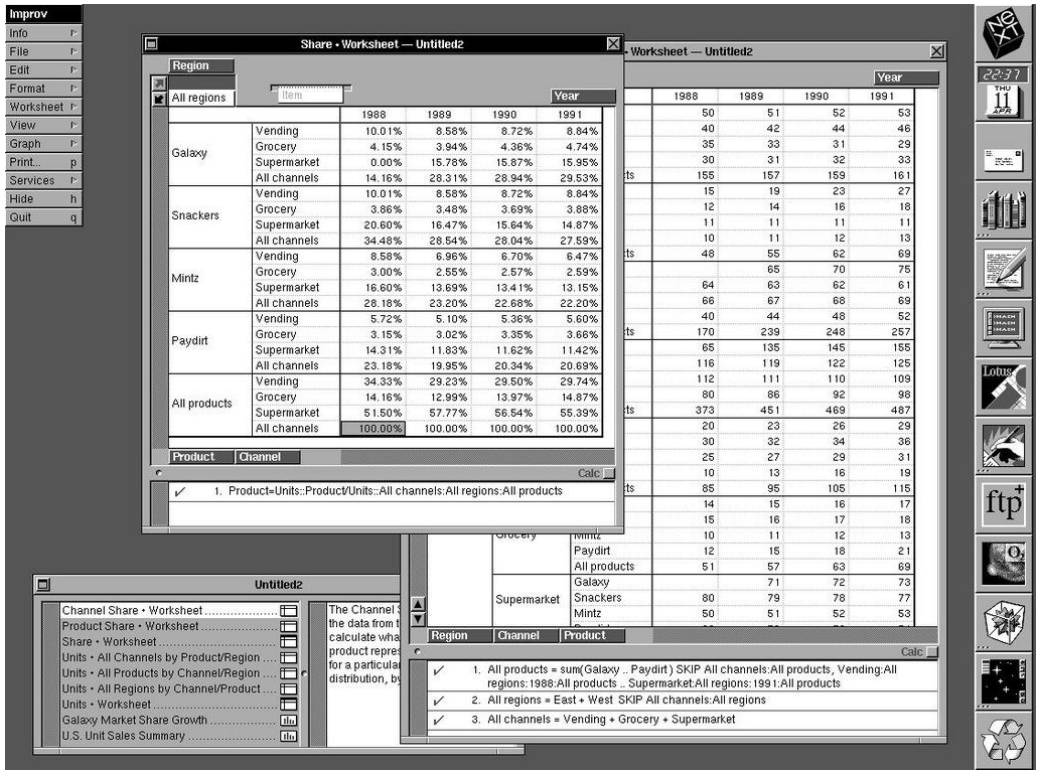
Tables in Microsoft Excel

- 2003 – Lists
- **2007 – Tables**
  - With new access language (http://support.microsoft.com/en-gb/office/using-structured-references-with-excel-tables-f5ed2452-2337-4f71-bed3-c8ae6d2b276e)
    - Example: `=DeptSales[[#Totals],[Sales Amount]:[Commission Amount]]`
  - Table is the successor of List!
    - VBA Tables and ListObjects (http://exceloffthegrid.com/vba-excel-tables-code/)
- 2013 – New query engine in Power Query
- 2016 – New generation of Data Model (internal DBMS)
  - + Tables access



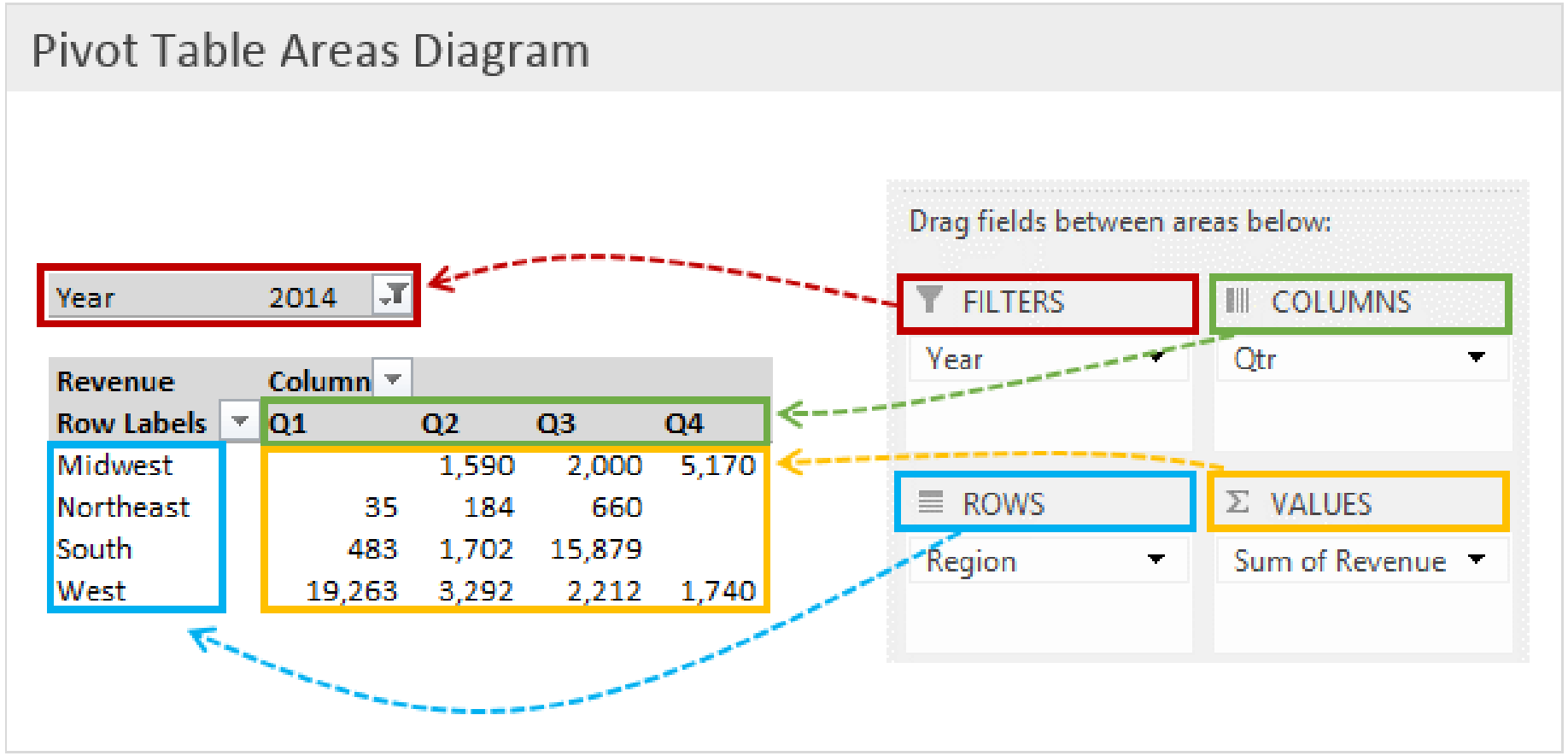
Pivot tables – beginning

- 1986 – Pito Salas from Lotus Corp – father of pivot tables
- First implementations:
  - 1991 – Lotus Improv (http://en.wikipedia.org/wiki/Lotus_Improv)
  - 1993 – Microsoft Excel 5.x (http://winworldpc.com/product/microsoft-excel/5x)



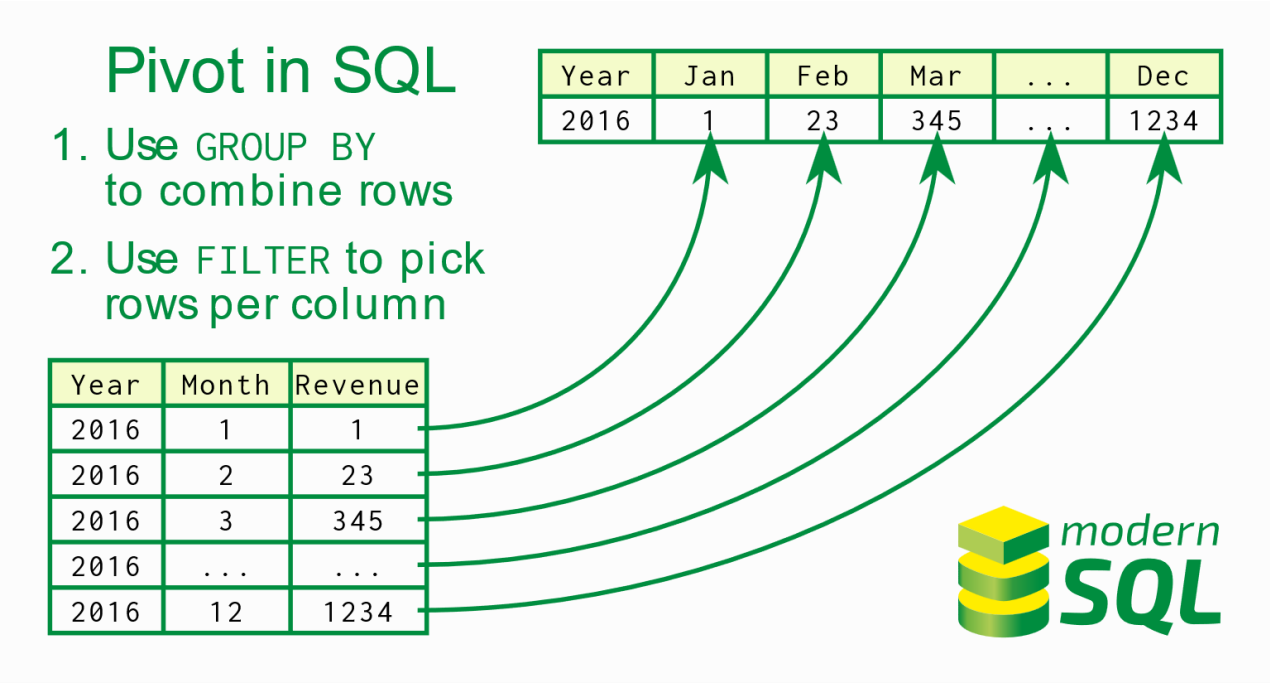
Excel Pivot Tutorial

- *How Do Pivot Tables Work?* (http://www.excelcampus.com/pivot-tables/pivot-tables-work/)



Pivot & SQL (addendum for previous topic)

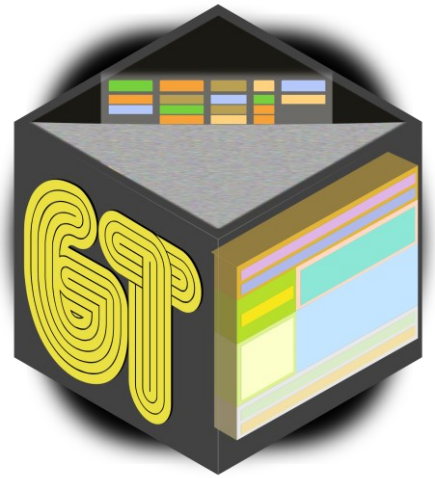
- In SQL language:
  - *Pivot — Rows to Columns* (https://modern-sql.com/use-case/pivot)

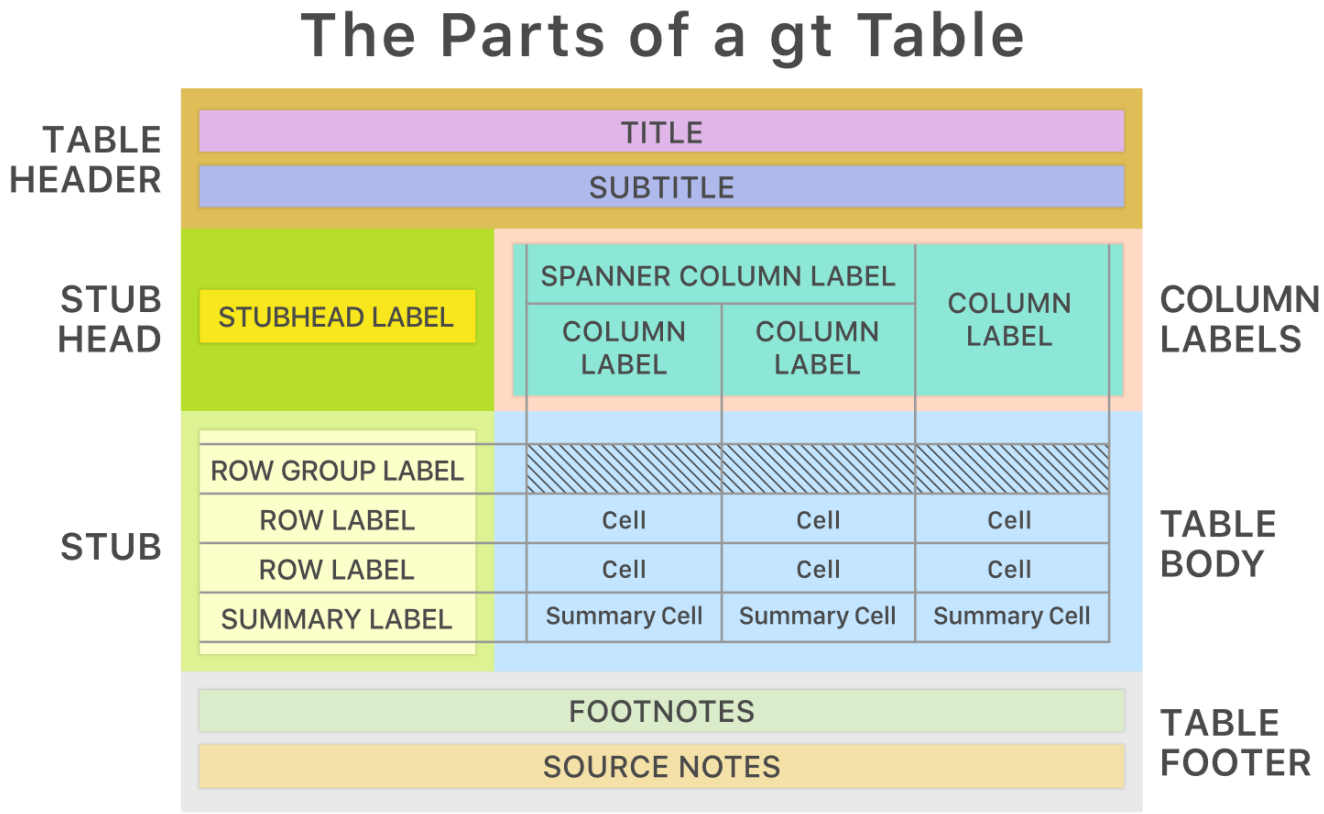


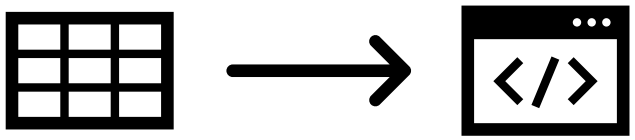
  - *Microsoft SQL Server: FROM – Using PIVOT and UNPIVOT* (http://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot)

R – gt (A grammar of tables)

- gt package (http://gt.rstudio.com)
  - Constructs a wide variety of useful tables with a cohesive set of table parts:
    - Header, stub, column labels with spanners, body with summary, footer, …







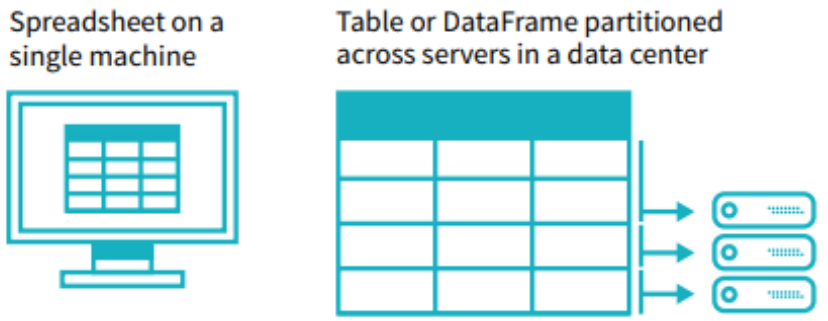
OBJECT-ATTRIBUTE DATA IN PROGRAMMING

- ✓ Dataframes and tables
- ✓ Dataframes interoperability
- ✓ Rich table construction and visualization

Dataframe

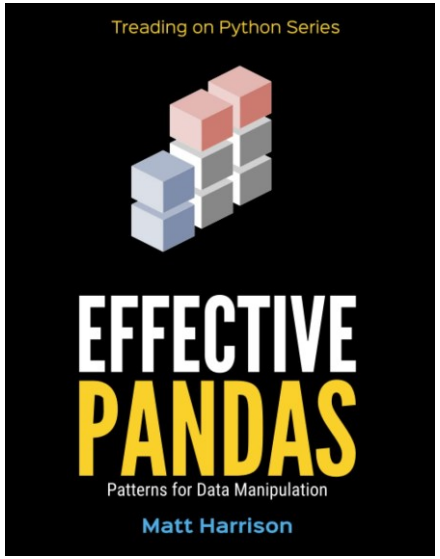
- A **tabular data structure** common to many data processing libraries:
  - pandas and other Python libraries
  - The Dataframe API in Apache Spark
  - Data frames in the R programming language
  - …

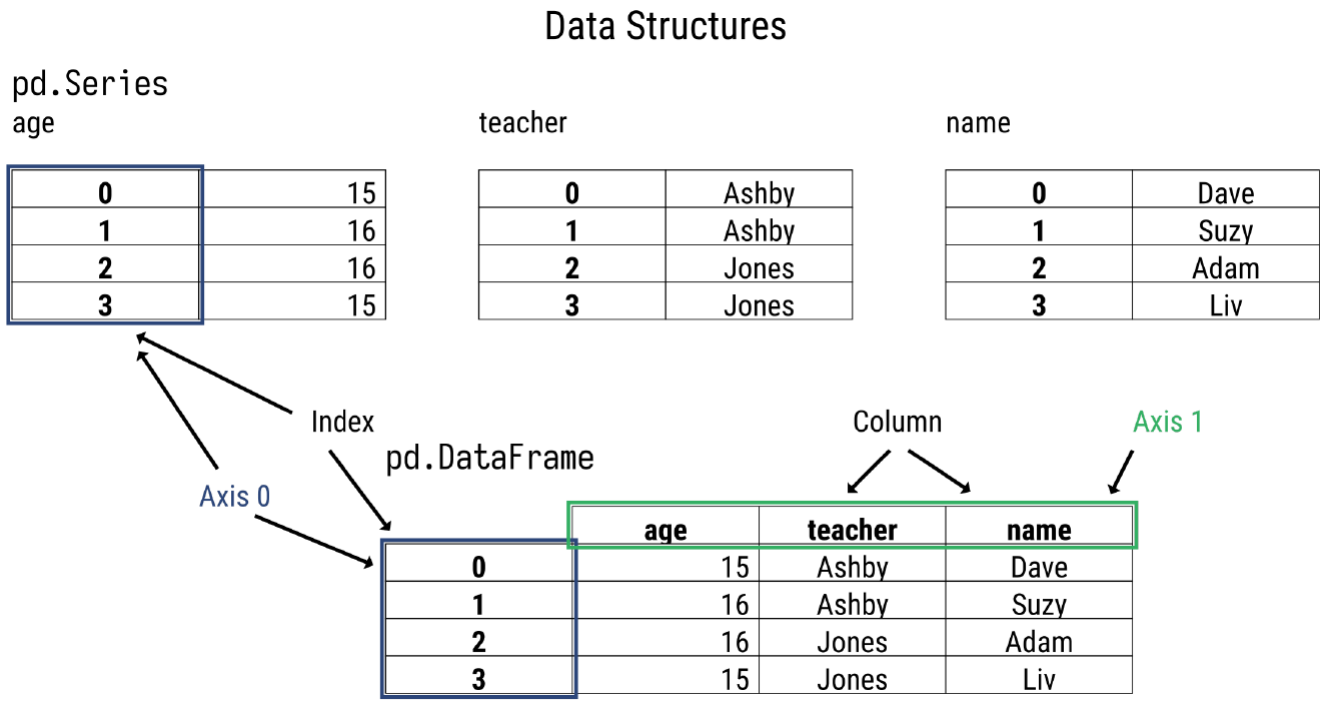
- Databricks – What is a DataFrame? (http://www.databricks.com/glossary/what-are-dataframes)
  - DataFrame contains a blueprint, known as a schema, that defines the name and data type of each column
  - Missing or incomplete values are stored as null values in the DataFrame



Python – Pandas

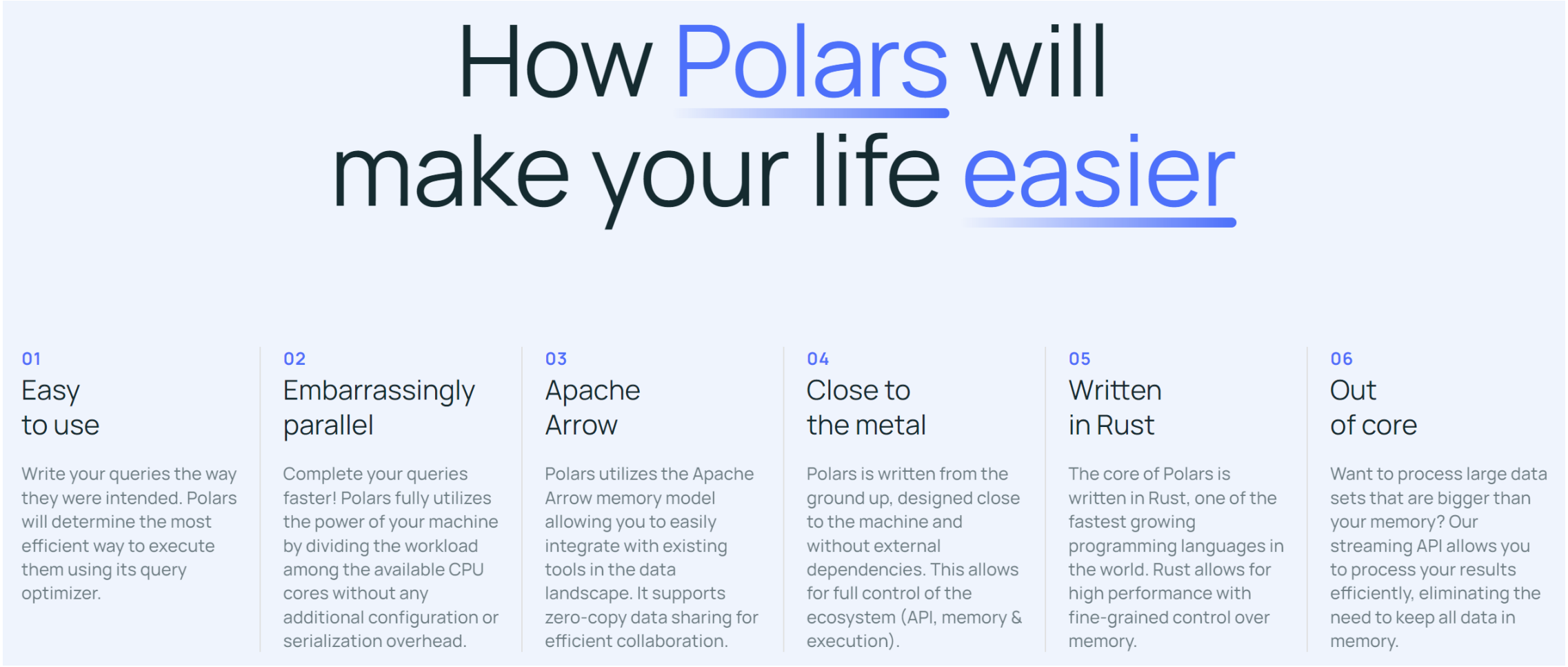
- Harrison M. Effective Pandas. Patterns for Data Manipulation. 2021. 397 p.
  - Series manipulation
  - Creating columns
  - Summary statistics
  - Grouping, pivoting, and cross-tabulation
  - Time series data
  - Visualization
  - Chaining





Python – Polars

- **Polars** – Lightning-fast DataFrame library for Rust and Python (https://www.pola.rs)

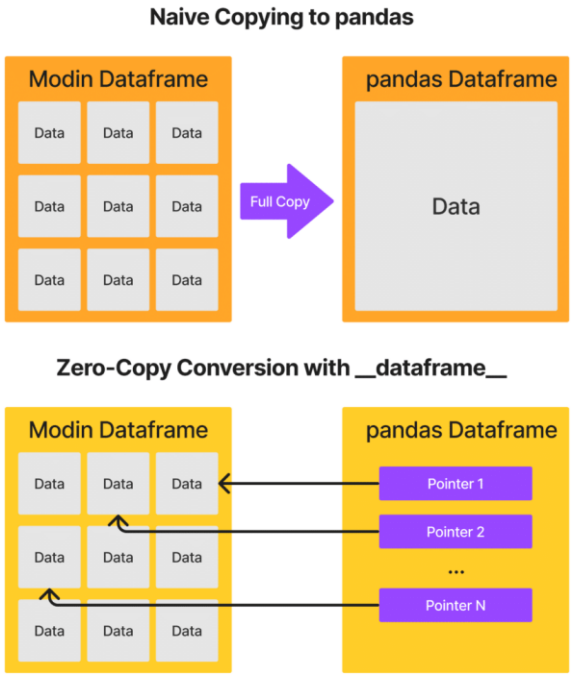


Python dataframes interoperability

- Consortium for Python Data API Standards (http://data-apis.org)

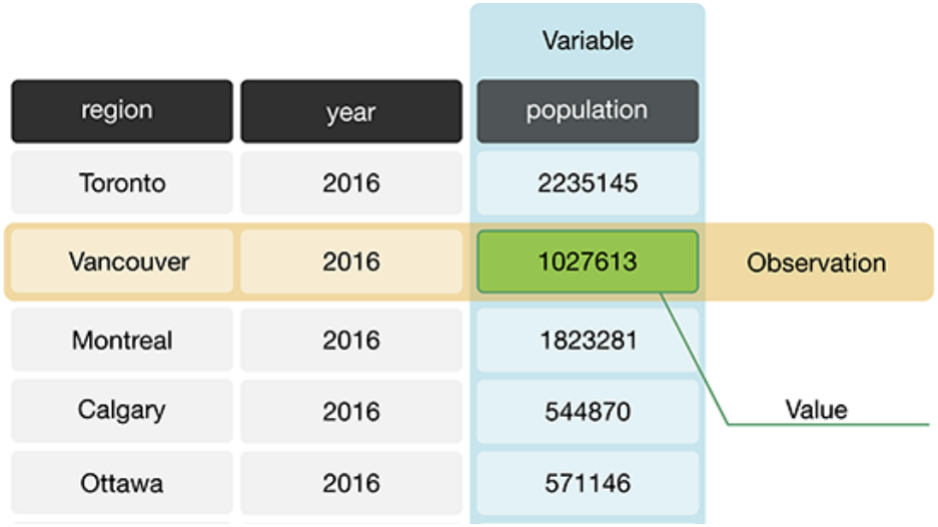
- Dataframe Interchange Protocol (http://arrow.apache.org/docs/python/interchange_protocol.html)

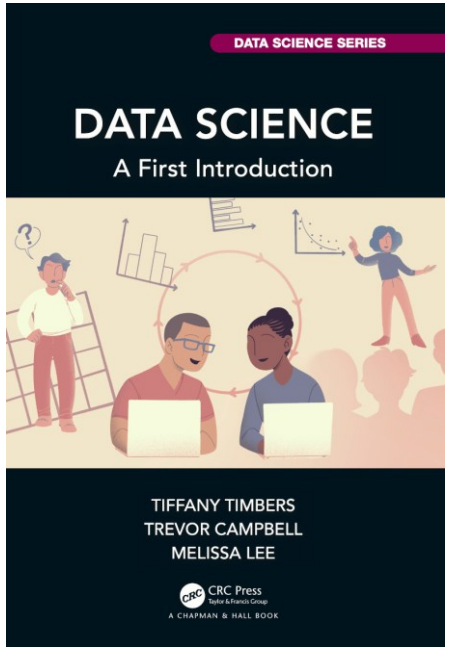
- How the Python Dataframe Interchange Protocol Makes Life Better (http://ponder.io/how-the-python-dataframe-interchange-protocol-makes-life-better/)
  - Vaex (PR merged 10/13/21)
  - cuDF (PR merged 11/17/21)
  - Modin (PR merged 2/25/22)
  - pandas (PR merged 4/27/22)
  - Polars (PR merged 1/30/23)
  - ibis (PR merged 6/16/23)



R – data frame

- Timbers T., Campbell T., Lee M. Data Science. A First Introduction. 2023
  - What is a data frame? (http://datasciencebook.ca/wrangling.html#what-is-a-data-frame)

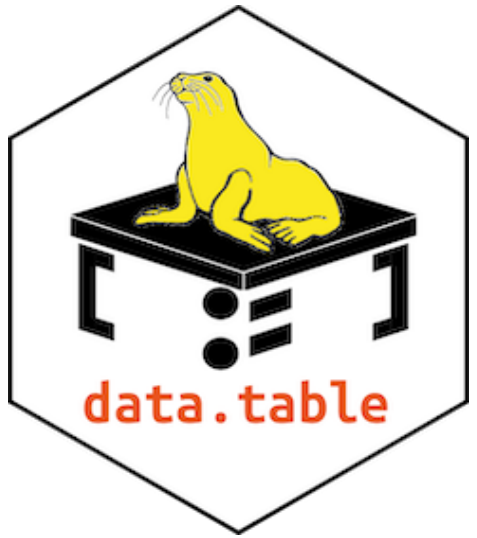


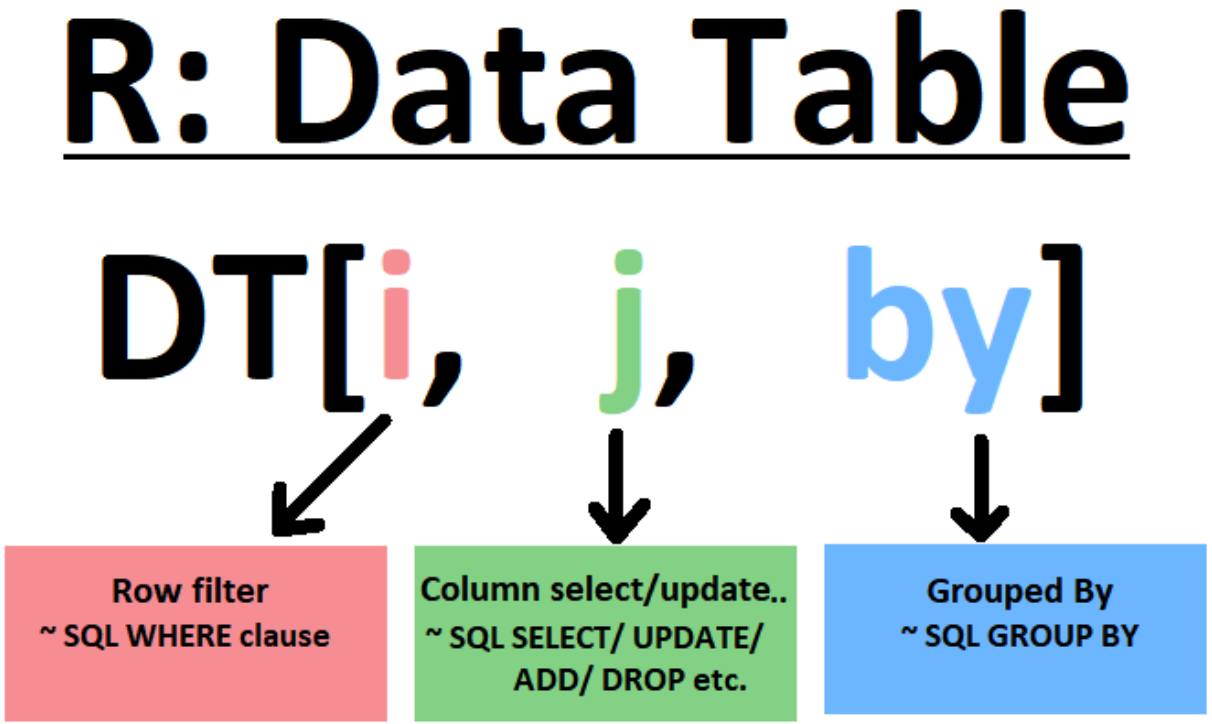


  - + Tidyverse – R packages for data science (https://www.tidyverse.org)

R – Tables

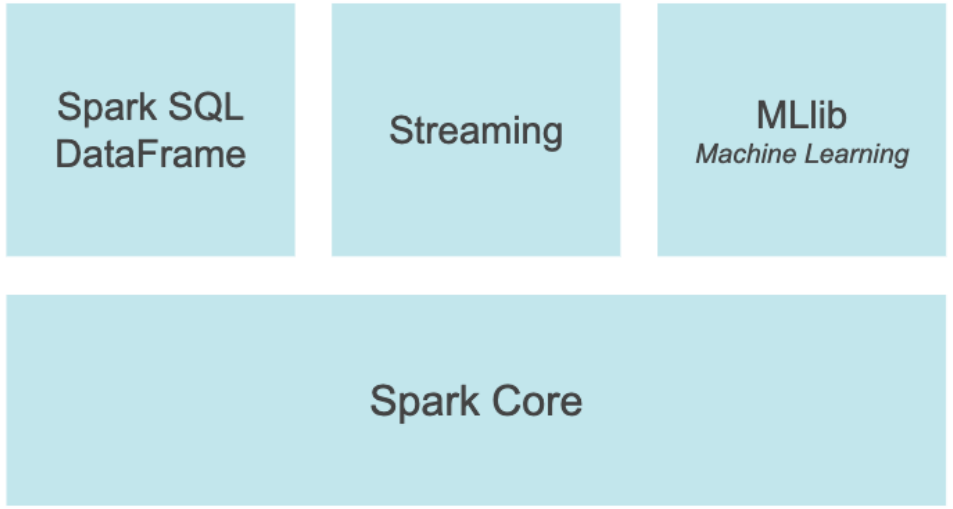
- `data.table` is an enhanced version of `data.frames`
  - Signorell A. Tables in R – A quick practical overview. 2021 (http://cran.r-project.org/web/packages/DescTools/vignettes/TablesInR.pdf)





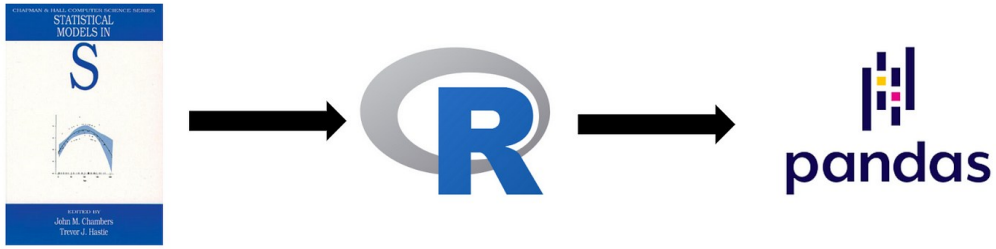
Spark – DataFrame

- Apache PySpark – pyspark.sql.DataFrame
  (http://spark.apache.org/docs/3.1.1/api/python/reference/api/pyspark.sql.DataFrame.html)
  - A DataFrame is a Dataset organized into named columns
  - It is conceptually equivalent to a table in a relational database
    or a data frame in R/Python, but with richer optimizations under the hood

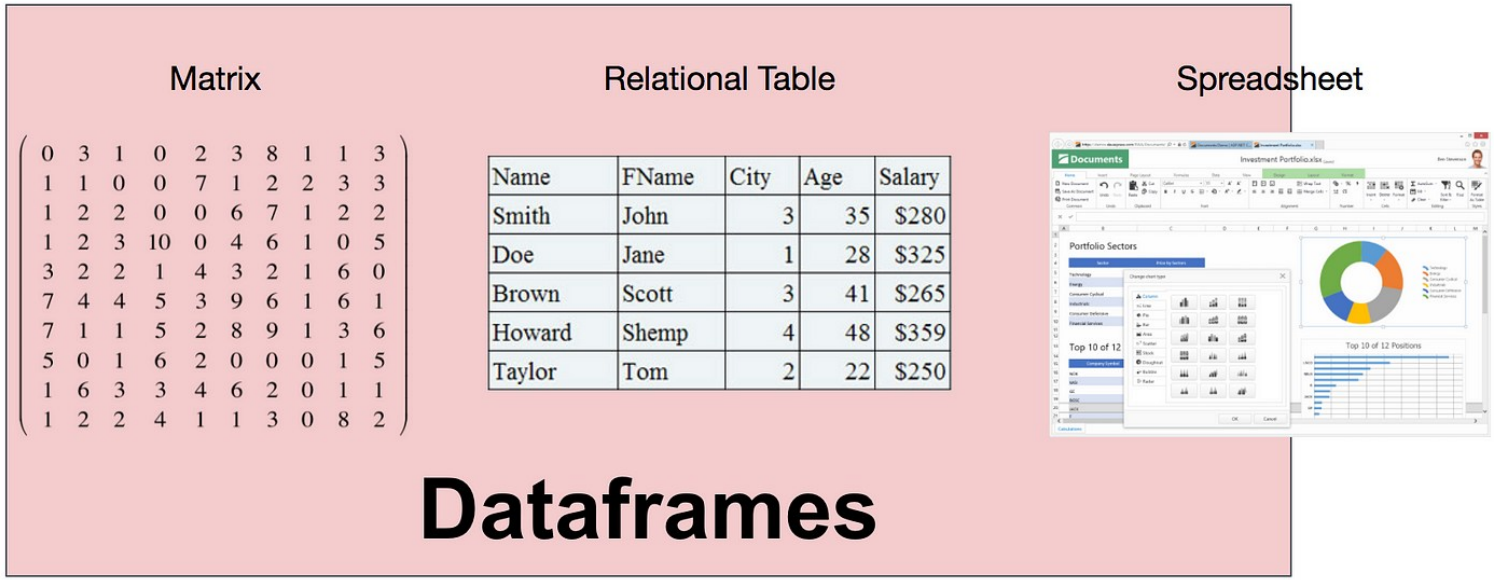


Terminology drift

- Preventing the Death of the Dataframe (http://towardsdatascience.com/preventing-the-death-of-the-dataframe-8bca1c0f83c8)
  - Dataframes are losing their statistical computing and machine learning roots:

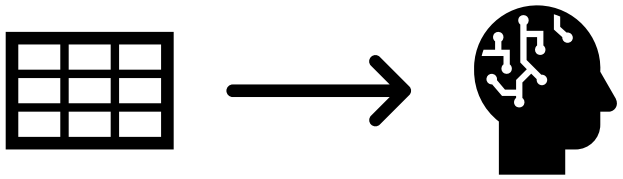


- Dataframe supports:
  - Linear algebra (matrices!)
  - Relational algebra (tables!)
  - Some cell formulae (spreadsheets!)



0	3	1	0	2	3	8	1	1	3
1	1	0	0	7	1	2	2	3	3
1	2	2	0	0	6	7	1	2	2
1	2	3	10	0	4	6	1	0	5
3	2	2	1	4	3	2	1	6	0
7	4	4	5	3	9	6	1	6	1
7	1	1	5	2	8	9	1	3	6
5	0	1	6	2	0	0	0	1	5
1	6	3	3	4	6	2	0	1	1
1	2	2	4	1	1	3	0	8	2

Name	FName	City	Age	Salary
Smith	John	3	35	$280
Doe	Jane	1	28	$325
Brown	Scott	3	41	$265
Howard	Shemp	4	48	$359
Taylor	Tom	2	22	$250



OAD AND AI

- ✓ Data and decision making
- ✓ AI and AGI
- ✓ Machine leaning : train, test, and validate
- ✓ TO EXTEND!!!

Machine learning – focusing on dataset

- Variables: features and targets
  - **Label** as target value
- Feature engineering
  - Datasets
    - Train, test, validate!

	$Feature_1$	…	$Feature_i$	…	$Target_j$
Obj_1		…	V	…	V
…	…	…	…	…	
Obj_q	V	…		…	
…	…	…	…	…	
Obj_p	V	…	V	…	

Label!

Train, test, and validation sets – cats and dogs

MLU-EXPLAIN | Introduction | The Split | Train Set | Model | Validation Set | Test Set | Summary

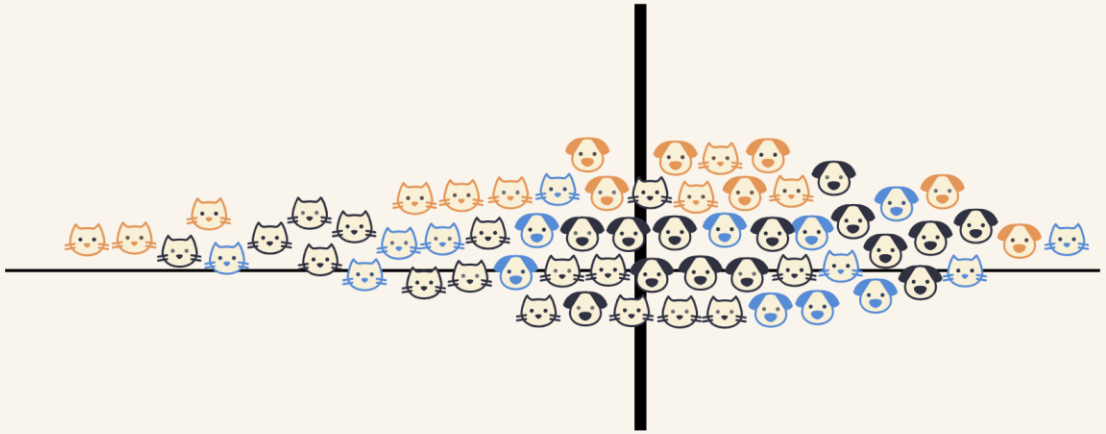
Model Features: None | Weight | Fluffiness | Both

The Testing Set

Once we have used the validation set to determine the algorithm and parameter choices that we would like to use in production, the test set is used to approximate the models's true performance in the wild. It is the final step in evaluating our model's performance on unseen data.

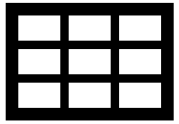
We should never, under any circumstance, look at the test set's performance before selecting a model.

Peeking at our test set performance ahead of time is a form of overfitting, and will likely lead to unreliable performance expectations in production. It should only be checked as the final form of evaluation, after the validation set has been used to identify the best model.



dataset	feature	# cat right	# cat wrong	# dog right	# dog wrong	accuracy
test	weight	7	2	5	2	75.0%
validation	weight	5	3	6	2	68.8%

(C) 2023, Train,Test, and Validation Sets (https://mlu-explain.github.io/train-test-validation/)



THE END OF PART 1

WHAT'S NEXT?

- ✓ Dataviz as art and technology
- ✓ Visualization of tables
- ✓ Charts and their classification
- ✓ Time series as object-attribute data
- ✓ BI
- ✓ Processes and versioning
- ✓ ML specific topics