data dictionaries / concept dictionaries

- *A data dictionary is used to catalog and communicate the structure and content of data, and provides meaningful descriptions for individually named data objects.*
- *Data Dictionaries **are for sharing**. For groups of people working with similar data, having a shared data dictionary facilitates standardization by documenting common data structures and providing the precise vocabulary needed for discussing specific data elements. Shared dictionaries ensure that the meaning, relevance, and quality of data elements are the same for all users. Data dictionaries also provide information needed by those who build systems and applications that support the data. Lastly, if there is a common, vetted, and documented data resource, it is not necessary to produce separate documentation for each implementation.*

(see: https://www.usgs.gov/products/data-and-tools/data-management/data-dictionaries, last visited Nov. 2023)

Examples for spatial data dictionaries / concept dictionaries

- USGS data dictionaries – see:
  - https://www.usgs.gov/products/data-and-tools/data-management/data-dictionaries
- INSPIRE (Infrastructure for spatial information exchange in Europe):
  - http://inspire.ec.europa.eu/featureconcept

Feature Concepts

☑ Show only valid items

Label	Themes
Abstract Building	Buildings
Abstract Construction	Buildings
Abstract Exposed Element	Natural risk zones
Abstract Hazard Area	Natural risk zones
Abstract Installation	Agricultural and aquaculture facilities
Abstract Monitoring Feature	Environmental monitoring facilities
Abstract Monitoring Object	Environmental monitoring facilities
Abstract Observed Event	Natural risk zones
Abstract Risk Zone	Natural risk zones
Access Restriction	Transport networks
Active Well	Geology
Activity Complex	
Address	Addresses

Address

Search...

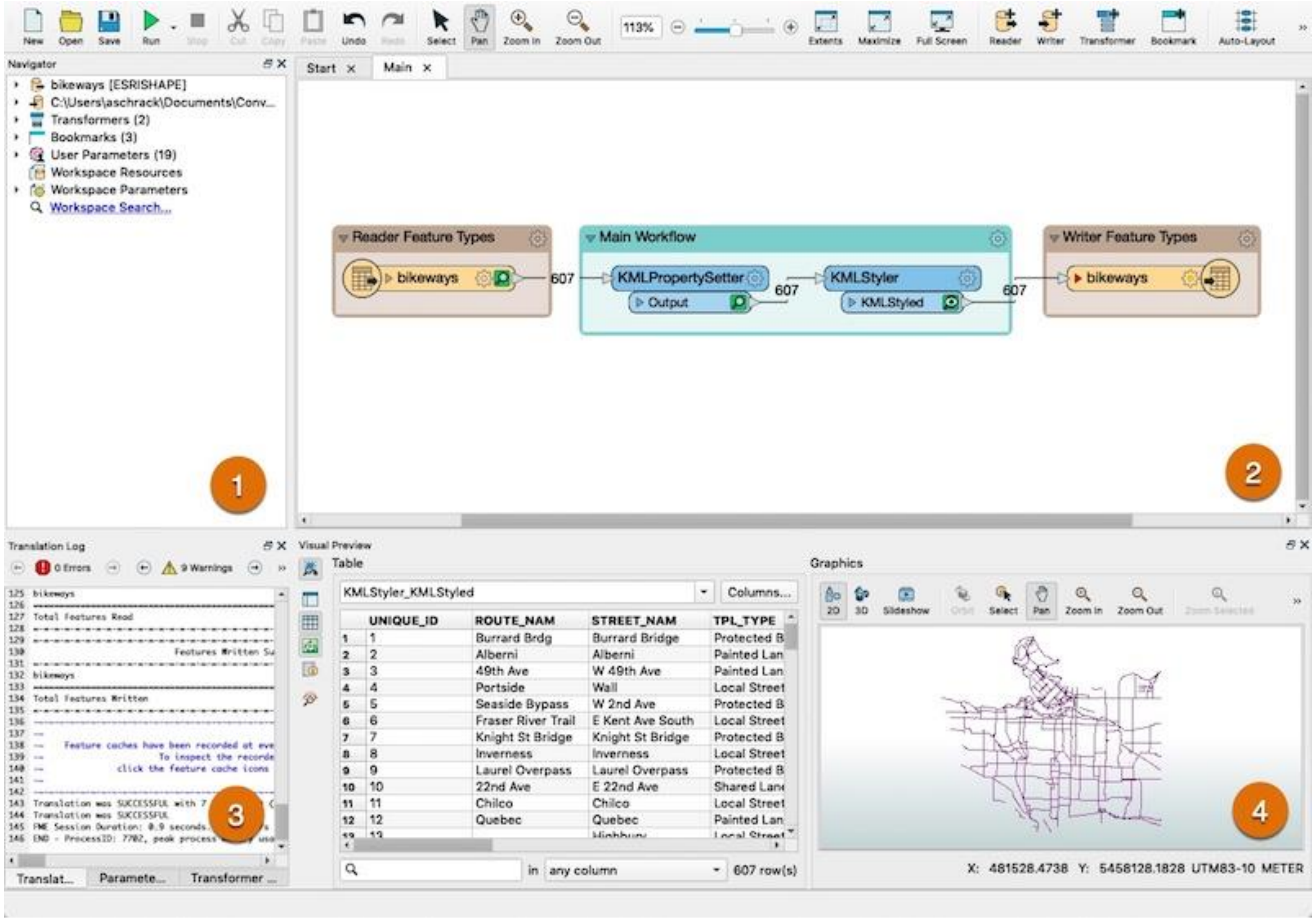
Help us improving the **Re3gistry software**! Please fill our quick survey at http://europa.eu/!Bn84Ct

ID:	http://inspire.ec.europa.eu/featureconcept/Address
This version:	http://inspire.ec.europa.eu/featureconcept/Address:1
Latest version:	http://inspire.ec.europa.eu/featureconcept/Address
Label:	**Address**
Definition:	An identification of the fixed location of property by means of a structured composition of geographic names and identifiers.
Governance level:	eu-legal
Status:	Valid
UML class name:	Address
Theme:	Addresses
Other formats:	XML Re3gistry, XML ISO 19135, RDF/XML, JSON, Atom

Sharing data – master data / data mapping and transformation

- Data described by its data model is frequently shared within and across organizations
- Master data represents the business objects that contain the most valuable, agreed upon information shared across an organization
  - Typically data related to the entities of an organization like customers / suppliers / products / employees / …
  - Take the full scope of data into account when defining the concept for master data
  - Special enterprise applications like „Enterprise-Resource-Planning (or ERP) systems“ provide data models for the master data that are required to be collected as part of the application.
- Data that is shared across organizations
  - In case the concepts are defined, data can be exchanged – frequently by establishing „mapping rules“ for the data to be exchanged
    - *customer* „relates to“ *purchaser*
    - *school* „is type of“ *public building*
  - Such mapping rules can be established by Extract Transform Load (ETL) tools like FME from Safe Software – which addresses spatial data

ETL example



Example taken from:

https://www.safe.com/fme/fme-desktop/tour/2019.0

Data modeling concepts – LDM

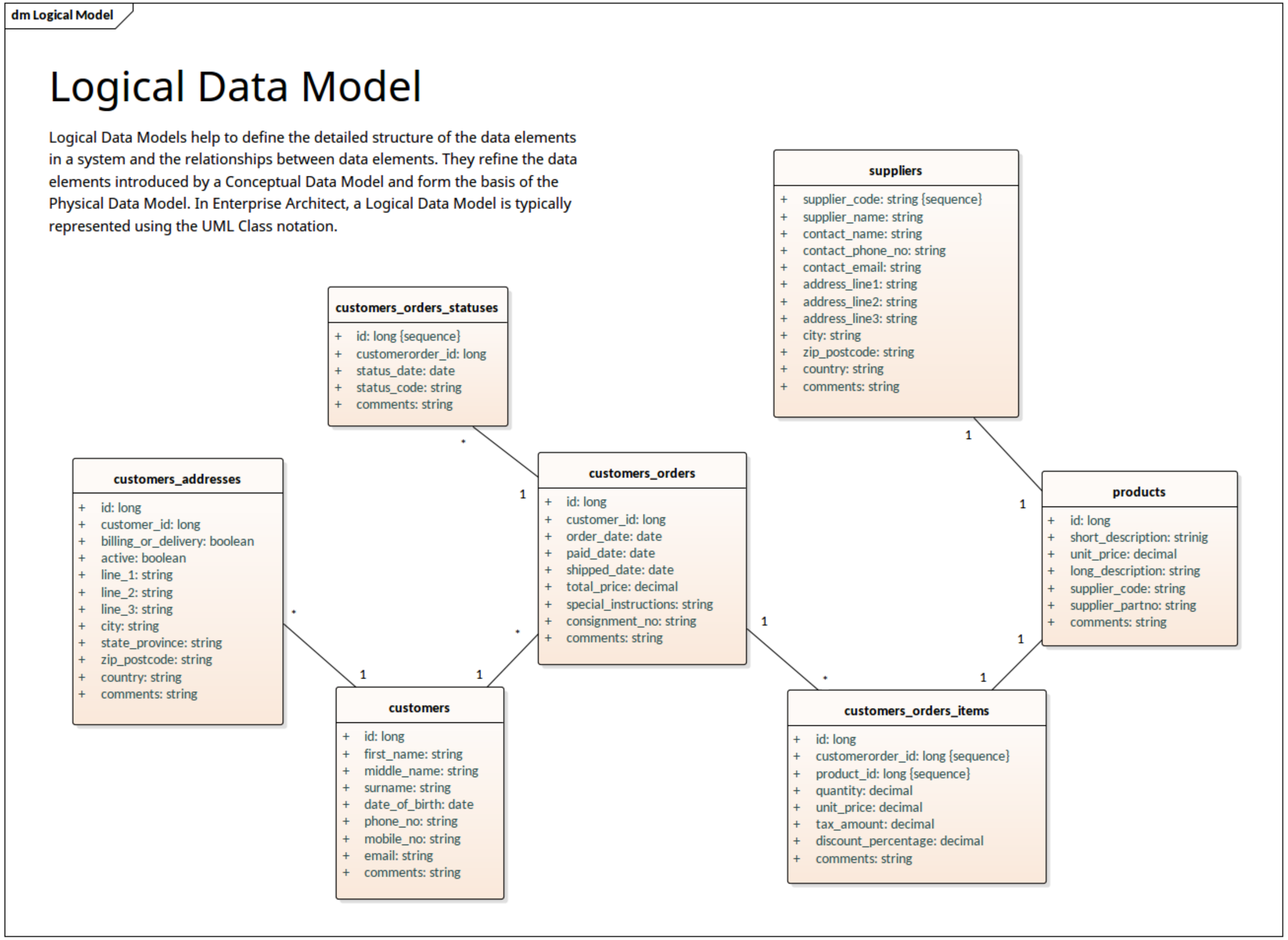
- A **logical data model** or logical schema is a data model of a *specific problem domain expressed independently of a particular database management product or storage technology* (physical data model) but in terms of data structures such as relational tables and columns, object-oriented classes, or XML tags. This is as opposed to a conceptual data model, which describes the semantics of an organization without reference to technology.

  Logical data models represent the abstract structure of a domain of information. They are often diagrammatic in nature and are most typically used in business processes that seek to capture things of importance to an organization and how they relate to one another

  (see.: http://en.wikipedia.org/wiki/Logical_data_model, Nov.2022)

- The LDM takes the concepts and relationships of the CDM and make them more concrete. It already addressed how the schema can be implemented on an „ideal machine“.
- A CDM can be represented by more than one LDM
- A LDM forms the basis for the physical data model
- Logical data models are used to visualize data entities, attributes, keys, and relationships

Example of a LDM



Example taken from: https://sparxsystems.com/resources/gallery/diagrams/software/sw-logical_data_model-uml_notation.html , visited Nov. 2023)

Logical data models are used to **visualize data entities, attributes, keys, and relationships.** (see: https://www.ibm.com/docs/en/iis/11.5?topic=types-physical-logical-data-models, Nov. 2023)

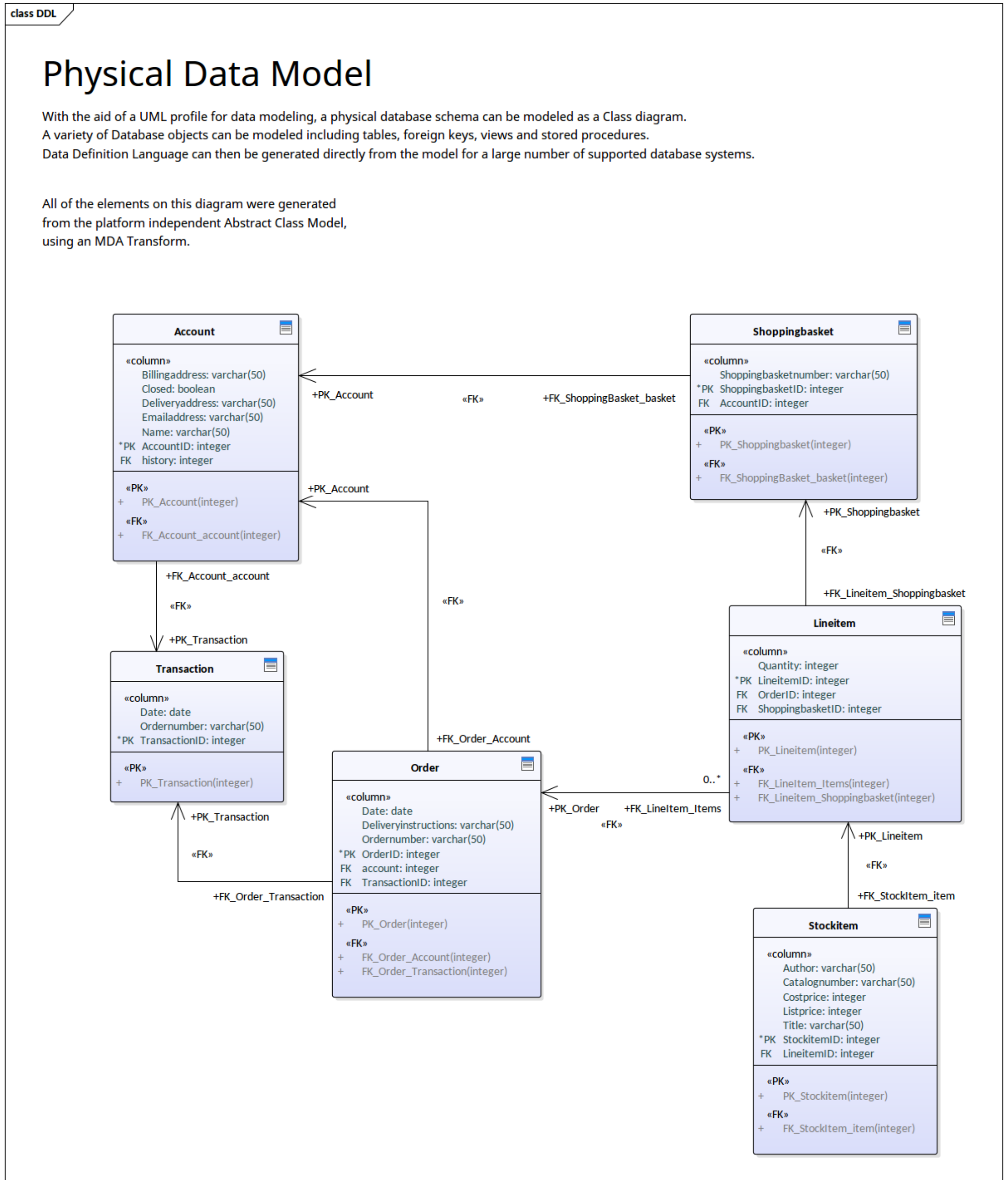
Data modeling concepts – PDM

Physical Data Model:

- A physical data model (or database design) is a representation of a data design as implemented, or intended to be implemented, in a database management system. In the lifecycle of a project it typically derives from a logical data model, though it may be reverse-engineered from a given database implementation. A complete physical data model will include all the database artifacts required to create relationships between tables or to achieve performance goals, such as indexes, constraint definitions, linking tables, partitioned tables or clusters.

  (see.: http://en.wikipedia.org/wiki/Physical_data_model , Nov. 2023)
- The blue print of the schema to store data
- Physical data models are **used to visualize the physical structure of databases and data files**. Logical data models are used to visualize data entities, attributes, keys, and relationships. (see: https://www.ibm.com/docs/en/iis/11.5?topic=types-physical-logical-data-models, Nov. 2023)
- Note: PDM is not limited to DBMS. Can be implemented for instance in terms of: XML documents, spreadsheet data ,file formats, etc.

Example of a PDM



Example taken from: https://sparxsystems.com/resources/gallery/diagrams/software/sw-physical_data_model-uml_notation.html, visited Nov 2023)

Physical data models are **used to visualize the physical structure of databases and data files.**

Logical data models are used to visualize data entities, attributes, keys, and relationships. (see: https://www.ibm.com/docs/en/iis/11.5?topic=types-physical-logical-data-models, Nov. 2023)

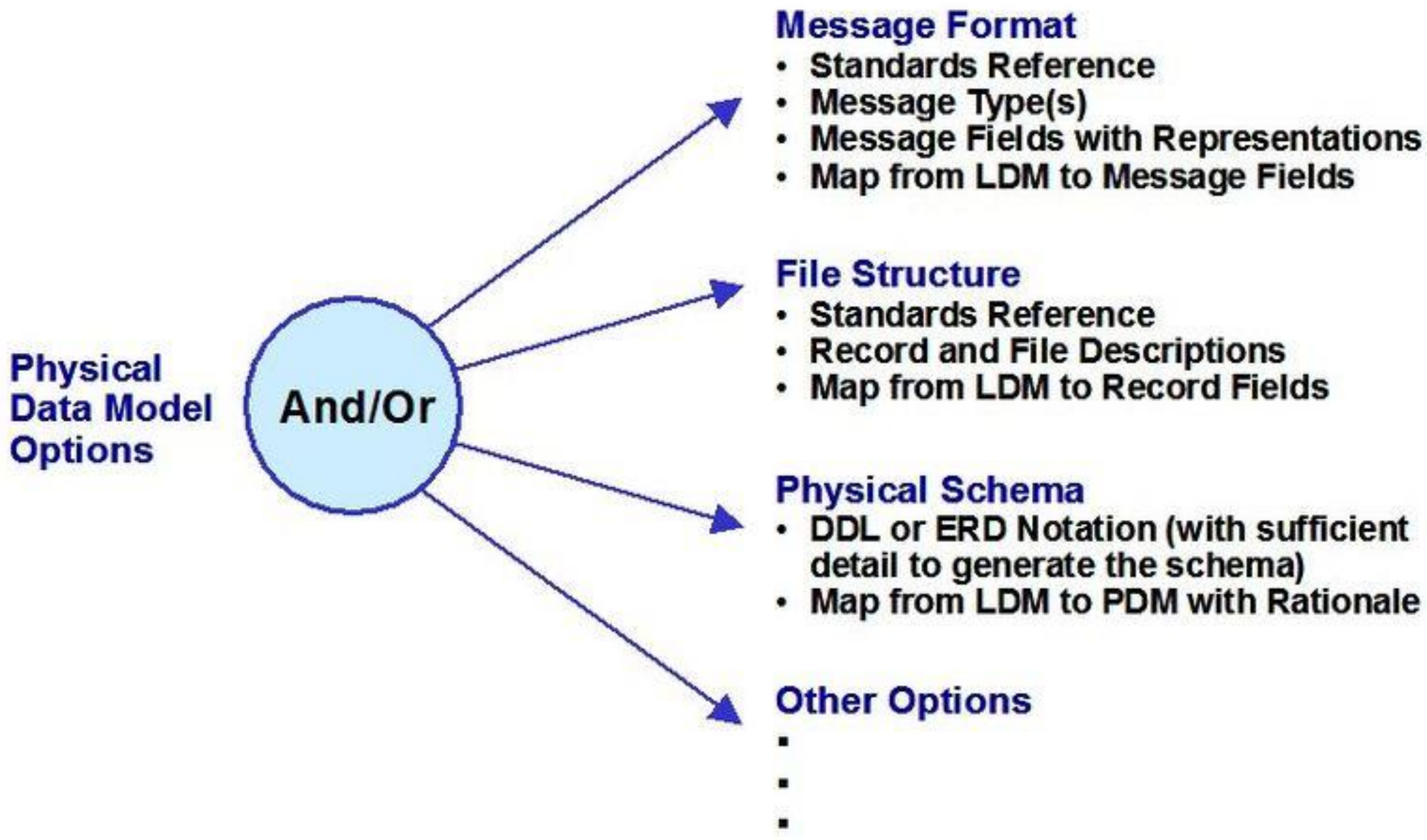
Differences LDM / PDM (an example)

LDM

PDM

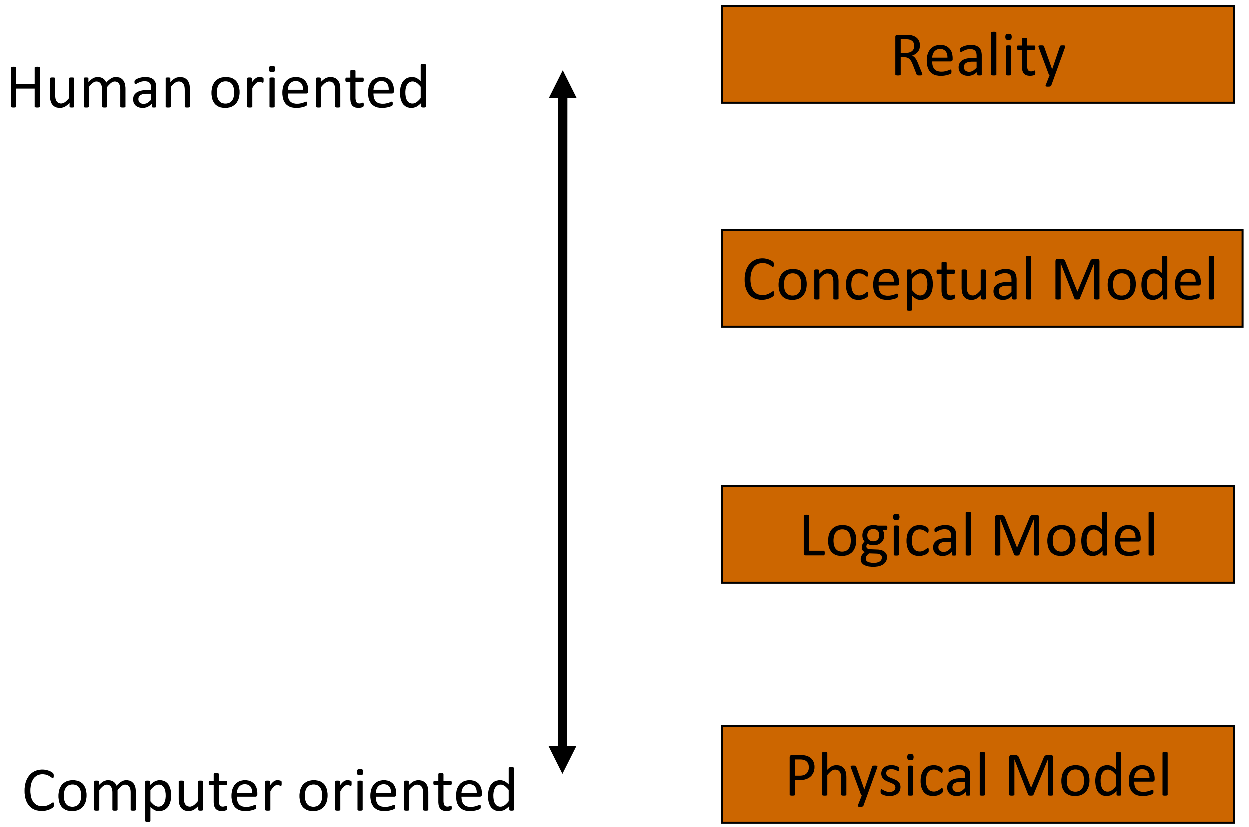
(Source: UNIGIS Course Materials, Course: “Database Theory”)

Examples of different storage schemas relevant for LDM and PDM



(Source: Treasury Enterprise Architecture Framework. Version 1, July 2000)

The modeling stack



Summary

Conceptual Data Model (CDM)	Logical Data Model (LDM)	Physical Data Model (PDM)
Includes high-level data constructs	Includes entities (tables), attributes (columns/fields) and relationships (keys)	Includes tables, columns, keys, data types, validation rules, database triggers, stored procedures, domains, and access constraints
Non-technical names, so that executives and managers at all levels can understand the data basis of Architectural Description	Uses business names for entities & attributes	Uses more defined and less generic specific names for tables and columns, such as abbreviated column names, limited by the database management system (DBMS) and any company defined standards
Uses general high-level data constructs from which Architectural Descriptions are created in non-technical terms	Is independent of technology (platform, DBMS)	Includes primary keys and indices for fast data access.
May not be normalized	Is normalized to fourth normal form (4NF)	May be de-normalized to meet performance requirements based on the nature of the database. If the nature of the database is Online Transaction Processing (OLTP) or Operational Data Store (ODS) it is usually not de-normalized. De-normalization is common in Datawarehouses.
Represented in the DIV-1 Viewpoint (DoDAF V2.0)	Represented in the DIV-2 Viewpoint (DoDAF V2.0), and OV-7 View (DoDAF V1.5)	Represented in the DIV-3 Viewpoint (DoDAF V2.0), and SV-11 View (DoDAF V1.5)

(see: https://en.wikipedia.org/wiki/Logical_schema, Nov 2023)

Creation of data models

- Common considerations
  - Assess the data requirements (thematic requirements, security requirements, performance requirements, legal requirements, ….)
  - Analyze legacy systems in use and define target storage system
  - Define and assess users (including maintainers)
  - Define purpose
- Principles:
  - „as simple as possible – but not simpler“
  - Work iteratively
  - Ask others to quality control your work

Creation of data models – approach (main steps)

Data modeling involves the following main tasks

- Identification of entity types (e.g., via business process modelling)
- Identification of attributes (e.g., via business process modelling)
- Definition of naming conventions and applying naming conventions
- Identification of relationships
- Key assignment
- Normalization to reduce data redundancy
- De-normalization to improve performance

Data modeling (1) – identification of entities / attributes

- *Identification of Entities*
  - Collection of data objects described by the same set of attributes
  - Similar to a class in UML
    (but no behavior is identified in a relational model)

- *Identification of Attributes*
  - Describe characteristics of entities
  - A column is the implementation of a data attribute in a DBMS

Data modeling (2) – naming conventions

- Applying Data Naming Conventions
  - Need to agree on naming conventions (Entity Names typically defined in singular)
  - Preferably such conventions are defined within an organization (rarely the case) -> design principles
  - Naming conventions for different types of models
    - in a physical data model technical prerequisites are to e considered
    - Conventions often aligned to physical limitations of systems; these may require adoptions
      - e.g. Maximum of 8 Characters for a field name, etc.)
      - d.g. No white spaces / no special characters / no…. are allowed

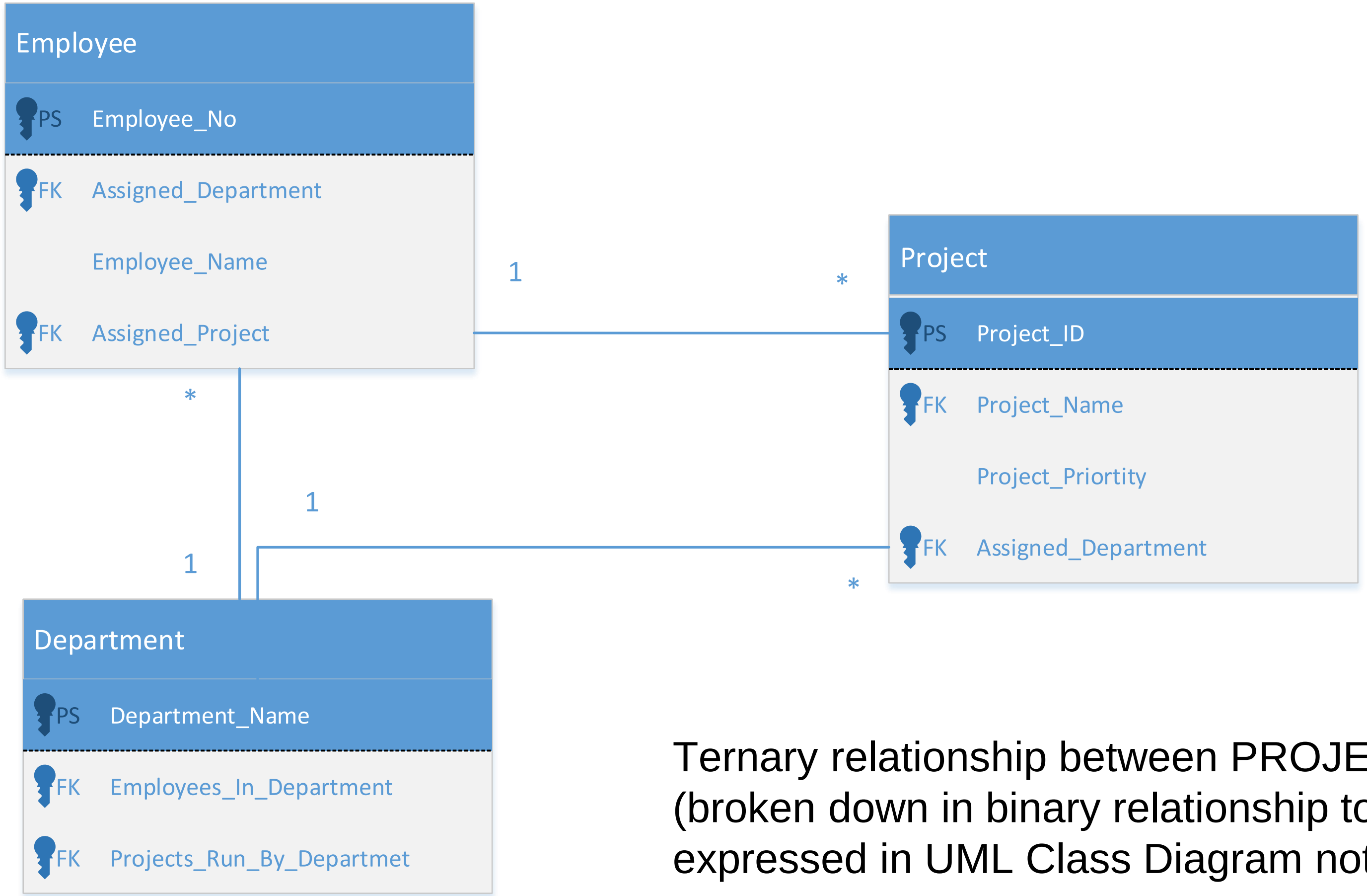
Data modeling (3) - identification of relationships

- Identification of relationships
  - A relationships between entities is conceptually identical to the relationships (associations) between objects
  - The relationship itself should be named (equals to the role of the relationship)
  - Relationship classified by degree and cardinality
  - **Degree of a relationship** = Number of entities associated with the relationship
    - Binary -> relationship between 2 entities
    - Ternary -> relationship between 3 entities
    - N-ary -> ….

Data modeling (4) - identification of relationships

- Identification of relationships (cont.)
  - Relationship **Cardinality**
    - Describe mapping of associated entity instances in a relationship
    - One to One (1:1)
    - One to Many (1:N)
    - Many to Many (N:M)
    - Recursive -> entity is related to itself

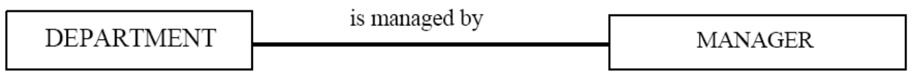
Data modeling (4a) – ternary relationship



Ternary relationship between PROJECT / EMPLOYEE / DEPARTMENT (broken down in binary relationship to establish the ternary relationship / expressed in UML Class Diagram notation)

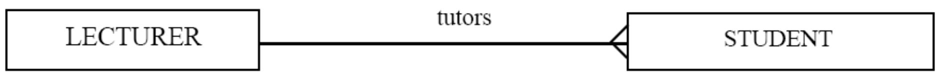
Data modeling (5) - identification of relationships

- Identification of relationships (cont.)
  - ***One to One (1:1)***



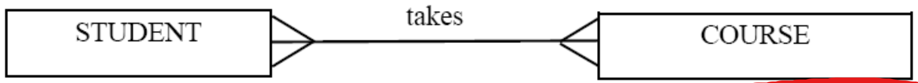
Data modeling (6) - identification of relationships

- Identification of relationships (cont.)
  - ***One to Many (1:N)***
  - Again - relationship managed via shared attributes
  - Foreign key needs to be included into related table



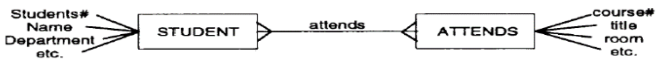
Data modeling (7) - identification of relationships

- Identification of relationships (cont.)
  - ***Many to Many (N:M)***

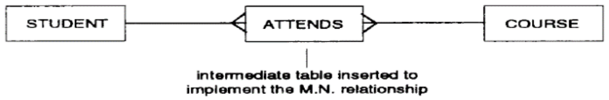


  - Many participants on both sides of the link
  - Require a “link table” in the physical Data Model

a) ATTRIBUTES ATTACHED TO ENTITLES



b) CONVERSION OF A M:N RELATIONSHIP INTO TWO 1:M RELATIONSHIPS



STUDENT

Student#	Name	etc.
1	Heywood	
2	Briggs	
3	Petch	

ATTENDS

Student#	*Course#*
1	101
2	101
1	103
1	102
2	103
3	101
3	103

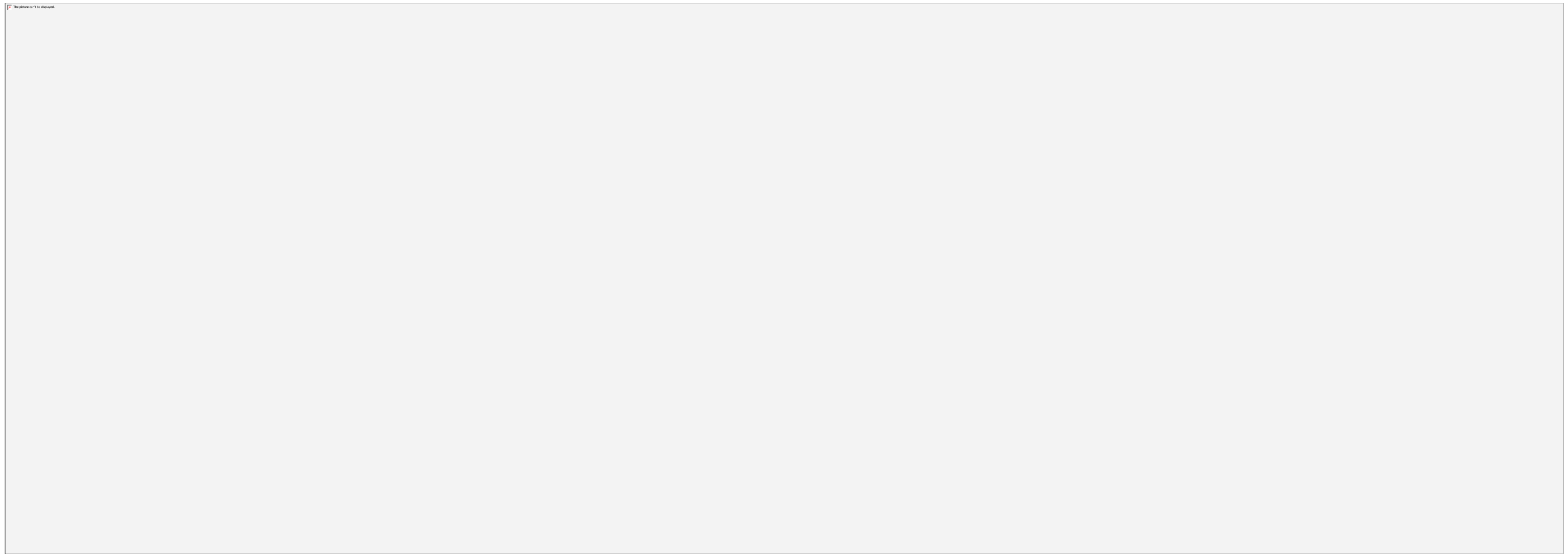
COURSE

Course#	Title	etc.
101	Biology	
103	Geology	
102	GIS	

(taken from: UNIGIS Course Materials, Course: “Database Theory”)

Data modeling (8) - identification of relationships

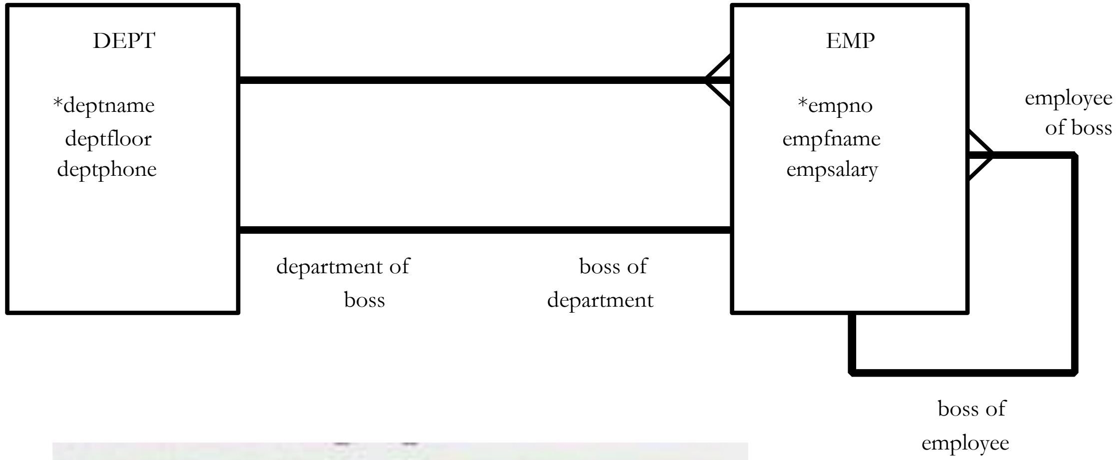
- Identification of relationships – Example (Information Engineering Notation)



(taken from: www.agiledata.org,: “Data modeling 101”)

Data modeling (9) - identification of relationships

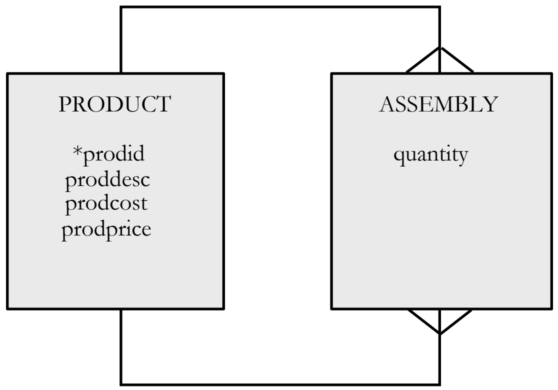
- Recursive Relationships (relationship of instances within an entity)
  - Manager manages employees. Manager and employees are instances of the employee entity



	EMP-CODE	EMP_LNAME	EMP_MANAGER
▶	101	Waddell	102
	102	Orincona	
	103	Jones	102
	104	Reballoh	102
	105	Robertson	102
	106	Deltona	102

Data modeling (10) - identification of relationships

- Identification of relationships (cont.)
  - Recursive Relationships (M:N)



PRODUCT

PRODID	PRODDESC	PRODCOST	PRODPRICE
1000	Animal photography kit		725
101	35mm camera	150	300
102	Camera case	10	15
103	70-210 zoom lens	125	200
104	28-85 zoom lens	115	185
105	Photographer's vest	25	40
106	Lens cleaning cloth	1	1.25
107	Tripod	35	45
108	24 exposure, 100 ASA, 35mm color negative film	.85	1

ASSEMBLY

QUANTITY	*PRODID*	*SUBPRODID*
1	1000	101
1	1000	102
1	1000	103
1	1000	104
1	1000	105
2	1000	106
1	1000	107
10	1000	108

A product can be assembled based on other products. Therefore a product can appear as part of many other products and can be made up of many products.

Data modeling (11) – key assignment

- Assignment of Keys
  - A key identifies an entity
    - *Composite key* (key definition by more than 1 attribute)
    - *Candidate key* (key suitable to identify an entity)
    - ***Primary Key*** (primary key for entity identification
    - *Alternate Key* (alternative key for entity identification)
  - Foreign Key
    - A column that contains the primary keys of another table. Used to establish the link between two tables.

Data modeling (12) – key assignment

- Assignment of Keys – Example (UML notation)

(taken from: www.agiledata.org,: “Data modeling 101”)

Data modeling (13) – key assignment

- Assignment of Keys – Strategies
  - Use of a ***natural key*** (i.e. existing data attributes)
    - E.g. Social Security Number
  - Introduce a new column to be used as a key = ***surrogate key***
    - Typically in absence of the ability to define a natural key as unique identifier
    - No real business meaning, just there to uniquely identify an entity
    - Very important in case more complex database architectures needs to be implemented (e.g. distributed databases, replication scenarios, …)

Data modeling (14) – key assignment

- Assignment of Keys – Strategies
  - Options to define surrogate keys
    - Key values assigned by the database
    - MAX() + 1
    - Universally unique identifiers (UUIDs).
    - Globally unique identifiers (GUIDs)
    - High-low strategy