Part 4
Entity Base

Part 4.1
From Relational Data to Entities

Databases & entities (1)

- As discovered in the previous lecture, the KBs and KGs changed the way to represent the data, by introducing the entity modelling.
- In particular the KGs provide a new representation of the entities, previously represented only through (relational and non-relational) databases (DBs).
- Nowadays, relational and non-relational DBs are still largely used. Therefore, in the current time the two different approaches coexists.
- Nevertheless, the exploitation of KGs is still hindered by the costs of transforming databases into graph-based entities.

Databases & entities (2)

- The major difference between the database representation and the KG's representation, is that, in the former the real-world **entities are only logically defined**, while in the latter the **entities are concretely represented**.
- **(Relational) Databases**: the entity modelling appears in the database logical model (ER, ERR models). Nevertheless, in the database physical model the entities representation changes based on the Database Management System (DBMS).
- **Knowledge Graphs**: the entity modelling appears during the modelling of the KG's knowledge layer that shapes the KG's structure. The entities representation does not change in the KG implementation.

(Relational) Database entity representation (1)

- **ER model**: logical model of the database, it represents the entities through their attributes and their relationships with other entities.
    - The entities are logically represented as they were conceived. Noticed how it is shaped as a **graph**!

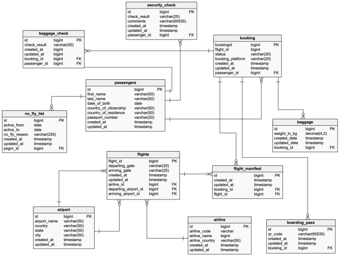
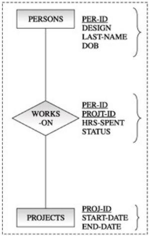


Figure: ER model example

(Relational) Database entity representation (2)

- The logical db model is then implemented in the physical model by representing each entity type (class) as a table called **relation**, and each entity as a record, or **tuple**, of such a table.
    - The entities **graph view is lost**. The entities and their relationships are **values into a tabular data model**. Notice how also the relationships are turned into tables (not graph edges anymore).



(a) E-R Model

Relation: PERSONS

PER-ID	DESIGN	LAST-NAME	DOB
106519	Manager	Thomas	100282
122816	Programmer	Abhishek	220187
131256	Analyst	Jose	271084

Relation: WORKS-ON

PROJ-ID	PER-ID	HRS-SPENT	STATUS
P-4021	106519	45	MAJOR
P-4042	122816	15	MINOR
P-4021	106519	21	MINOR

Relation: PROJECTS

PROJ-ID	START-DATE	END-DATE
P-4021	100205	250705
P-4042	010104	300305
P-4022	300804	010805

(b) Converted relations

(Relational) Database entity representation (3)

- **Normalization process**: a process of organizing data into tables in such a way that the results of using the DB are always unambiguous and as intended. Such normalization is intrinsic to relational DB theory. It may have the effect of **duplicating data within the DB and often results in the creation of additional tables**.
- The new tables (and/or new data) added during the normalization process, **do not represent the entities and relationships modeled in the DB logical data model**.
- By consequence, the DB logical and physical model do not correspond anymore. This means that, **there is no a concrete and persistent definition of entity in relational DBs**.

(Non-Relational) Database entity representations

- Non-Relational DB, instead:
  - Have the ability to store large amounts of data with **little structure**.
  - Provide **schema-free or schema-on-read** options.
  - Have the ability to capture all types of data "Big Data" including **unstructured data**.
  - They are best for Rapid Application Development. Non-Relational DB is the best selection for flexible data storage **with little to no structure limitations**.
- This means that **Non-Relational DBs**, due to their application purpose, **do not define concretely the entities**.

Knowledge Graphs entity representation

- KGs are defined by two layers, the knowledge layer and the data layer.
- The knowledge layer can be designed as the logical model of a relational DB.
- Then the data values are organized and mapped over the knowledge layer to compose the KG.
- Unlike the relational DBs, t**he physical model is fully mapped over the logical one.**
- This means that, the modelling of the entity types in the knowledge layer is then instantiated by the data layer, thus creating entities which are in turn shaped as graph's nodes and edges (like in the logical model's structured).

By the composition of knowledge (logical) and data (physical) layers, it is possible to concretely define an entity.

From data to entities (1)

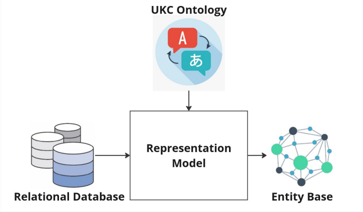
- The transformation of existing relational and non-relational data (DBs) into KGs, is crucial to reduce the data reuse cost.
  - **Scalability**: the evolution of a KG is cheaper respect to the cost of evolving a DB.
  - **Interoperability**: the data integration between high-quality KGs is cheaper respect of integrating DBs.

From data to entities (2)

- This course aims at:
  - providing a **concrete definition of entity**, as well as the definition of novel data representation model, based on KGs, namely the **Entity Base (EB)** model;
  - teaching how to reuse relational data by transforming it into EBs.
- The key idea is to exploit the suitability of the **KG data structure** to represent entities (and EBs), by highlighting the three **different layers of information diversity**.
- According to this approach, an entity is defined by three components, each representing a level of diversity of the entity representation.
  - Entity Language
  - Entity Knowledge (Schema)
  - Entity Data

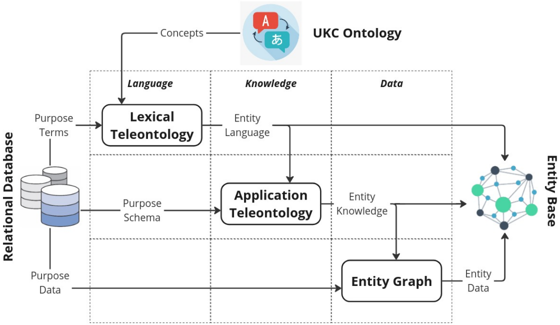
From data to entities (3)

- The transformation of relational **DBs into EBs**, is based on the below data representation model, taking two main inputs:
  - **The purpose, concretely represented by the DB to be transformed**. It intrinsically contains the representational (semantic and syntactic) meaning of the data.
  - **The UKC language ontology**. The input DB lacks of a language layer resource required to disambiguate the meaning of the terms used to represent its data. The UKC guided by the purpose provides the concepts to address the issue.



From data to entities (4)

- The Representation Model defines an entity with three components, each one representing a level of its diversity. **Each component is a KG!**
  - Entity Language - Lexical Teleontology (Language KG)
  - Entity Knowledge - Application Teleontology (ETypes KG)
  - Entity Data - Entity Graph (Entity values KG)



Part 4.2
Entity Language - Lexical Teleontology

Entity Language

- The entity language represents concretely the **language diversity for a specific entity**.
- As already mentioned above, the language diversity is concretely represented into **language datasets, or language ontologies**.
- To define the entity's language we have to consider two types of language ontology:
  - The global language ontology - UKC.
  - The Lexical Teleontology.

Global Language Ontology - UKC

- It is concretely implemented by the UKC project [29].
- This ontology represents a multilingual world-wide language resource, providing the concepts that can be used to define one (or a set of) entity and its properties.
- The UKC is continuously extended by new languages and/or new concepts into existing languages.
- This means that the UKC is not only a multi lingual resource but also a multi domain resource.

[29] see lecture on types of language resources.

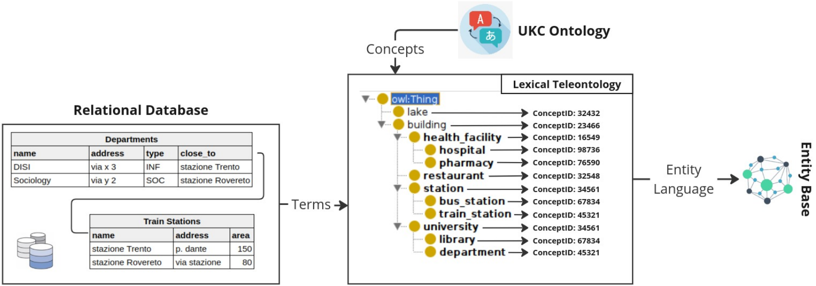
Lexical Teleontology

- The definition of entities for a specific purpose considers only a single (or a few) conceptual domains (set of concepts regarding a specific domain of interest).
  - And most of the time a single (natural or domain[30]) languages.
- Lexical Teleontology:
  - **"Tele"** from "Telos", a greek word that means purpose, objective.
  - **"Ontology"**, a knowledge resource that in this case represent the **lexical** concepts.
- In other words the Lexical Teleontology is a sub-set of the UKC focused on (including only the concept of) the purpose.
- This means that it contains all the concepts required to disambiguate the terms provided by the input DB.

[30] a dedicated lecture will explain the difference

Lexical Teleontology

- An example of Lexical Teleontology definition.



Departments			
name	address	type	close_to
DISI	via x 3	INF	stazione Trento
Sociology	via y 2	SOC	stazione Rovereto

Train Stations		
name	address	area
stazione Trento	p. dante	150
stazione Rovereto	via stazione	80

Language Graph

- The Lexical Teleontology defines the language of the entities structured as a KG.
  - Where the nodes are concepts, and the edges are the hierarchical relationships (IS-A) between them.
  - "Mother" –IS-A–> "Person"
- The input terms from the DB are used to match the UKC concepts to be used to disambiguate the meaning of the entities and properties, logically defined in the input DB.
- The Lexical Teleontology is the language diversity layer of the final EB.

Part 4.3

Entity Knowledge - Application Teleontology

Entity Knowledge

- The entity knowledge is the schema component of an entity, which expresses its **schema diversity**.
- The entity knowledge aims at representing **the types of entity (ETypes)** considered, through the definition of their **properties**:
  - **Data properties**: describing the attributes of a specific EType.
  - **Object properties**: describing how two ETypes are linked each other.
- The definitions of ETypes and their properties form a **EType Graph, or KG of the ETypes**.

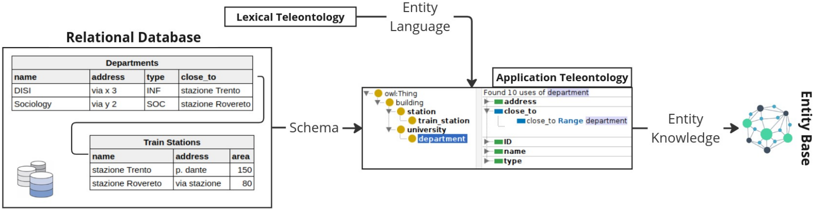
Application Teleontology (1)

- An ETypes Graph created for a specific purpose is called **Application Teleontology**.
  - Again from **"Telos"**: purpose, objective.
  - Therefore, an **ontology**, focused on a **specific application purpose**.

- The EB representation model defines the Application Teleontology starting from two inputs:
  - **the (purpose) DB tables structure, or schema**, indicating which entity types are considered and their properties (tables columns);
  - **the Lexical Teleontolgy**, providing the concepts to be associated to each EType and to each property, in order to be clearly disambiguated.

Application Teleontology (2)

- An example of Application Teleontology definition.



Application Teleontology (3)

- It is important to notice how the Application Teleontology, has two property dimension:
  - **The vertical one**: given by the the Lexical Teleontology, is represented by the IS-A relationships between the concepts used to define each EType.
    - The ETypes in the Application Teleontology inherit the conceptual hierarchy.
  - **The horizontal one**: the data and object properties which concretely represent how the ETypes are shaped.
    - How the ETypes are perceived given a specific purpose.
    - Data and object properties **do not define the EType hierarchy** (based on IS-A relationships), but instead, a **flat graph structure** where the ETypes are connected each other.

Application Teleontology (4)

- The Application Teleontology defines the knowledge (or schema) of the entities **structured as a KG**.
- The input DB tables and columns together with the Lexical Teleontology concepts are used to define the KG of ETypes, **the second component of the EB**.
- It is important to notice how **the Application Teleontology**, not only use but also **incorporate the Lexical Teleontology**.

Part 4.4
Entity Data - Entity Graph

Entity Data

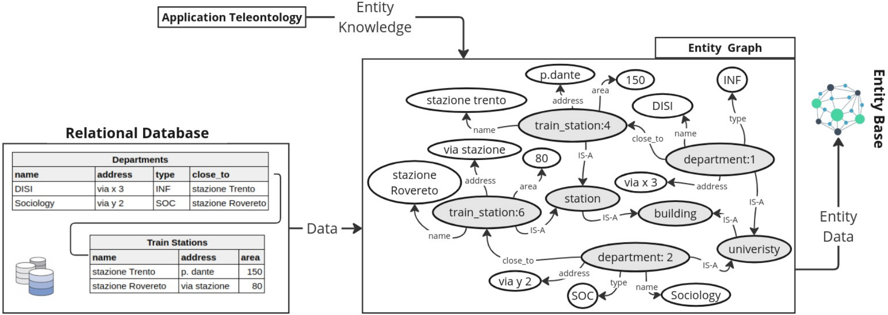
- The entity data is the entity, and EB, component which expresses its **data value diversity**.
- Such a diversity is represented by the values that instantiate the Application Teleontology ETypes and properties.
- The data diversity is expressed in the entity data through data values represented using **purpose-specific, but also interoperable, data types**.

Entity Graph (1)

- The generation of the last component defining the entities into EB, considers two inputs:
  - **The purpose data values**: extracted form the input DB, and, if possible transformed by using interoperable data types.
    - The purpose diversity has to be maintained, if possible.
  - **The Application Teleontolgy**: then, the data values are mapped on the application teleontology, generating the final **Entity Graph**.
- The definition of the Entity Graph compose **the entity language, knowledge and data layers**, thus concretely defining an entity by its three diversity layers/components.

Entity Graph (2)

- An example of Entity Graph definition.



Departments			
name	address	type	close_to
DISI	via x 3	INF	stazione Trento
Sociology	via y 2	SOC	stazione Rovereto

Train Stations		
name	address	area
stazione Trento	p. dante	150
stazione Rovereto	via stazione	80

Part 4.5
Entity Base

Entity Base definition

- We defined an entity as the composition of its language, knowledge and data layer, represented by the its definition into:
  - Lexical Teleontology (LT)
  - Application Teleontology (AT)
  - Entity Graph (EG)

- We can define now an Entity Base (EB) as a set of entiy, logically defined into an input DB and then represent through the three diversity KGs.

$$\textbf{EB} := <\textbf{LT, AT, EG}>$$

Entity Base - Example