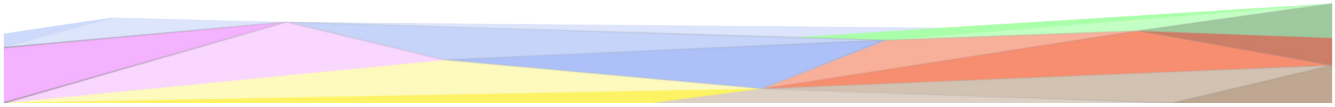
UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KDI 2021 - Project Report Template Trentino territory

Document Data:	Reference Persons:
December 24, 2021	Ludovic CHEVALLIER, Andrea FRIGO



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	20.04.2020	Fausto Giunchiglia	Document created

1 Introduction

Reusability is one of the main principles in the Data Integration (DI) process defined by iTelos. The data integration project documentation plays an important role in order to enhance the reusability of the resources handled during the methodology, as well as for the resources produced by the data integration process. A clear description of the resources and the process that has to manage them, provides a clear understanding of the information handled in the DI project, allowing external readers to exploit the same resources in different projects.

The current document aims to provide a detailed report of the DI project ”Trentino Territory” developed following the iTelos methodology. The report is structured, on top, to describe:

- Section 1: The project’s purpose and the resources involved (both schema and data resources) in the integration process.
- Section 2, 3, 4, 5: The integration process along the iTelos phases.
- Section 6: How the result of the integration process (KGs) can be exploited.

2 Purpose and project’s resources

2.1 Purpose Formalization

The purpose of our project is the development of an application that provides natural places in function of their category, name or location. The application will provide some information as name, description, category, company, location. With the usage of third-party services, the application will also be capable of calculating a route to arrive at the attraction from the user’s position. It will be possible to order the results by distance or name. The application will also suggest some interesting activities or places near the point of interest (for example a ”malga” in the mountain where the user wants to hike).

2.2 Domain of Interest

We identified two DoI:

- Geospace domain of the Trentino region territory: contains all the cities, roads, provinces, CAP and coordinates.
- Naturalistic information domain of the region: contains information about the different naturalistic attractions we can find, just like mountains, lakes, rivers and the companies that own them. The information we are interested in are the description of the attraction, its location, its type and, if presents, other attributes related to a specific type of attraction (for example, for a mountain its altitude).

2.3 Personas and scenarios

We created 3 different personas and scenarios. The personas are the following:

- P1: Sara, she is a 20 years old student living in Trento, she likes discovering new naturalistic point of interest. She likes high places and also goes by bike.
- P2: Micheal, he is a 30 years old tourist from France, his friends told him that in Trentino there are a lot of high mountains to visit, but he does not know their locations and how to reach them.
- P3: Vincenzo, he is a 45 years old who likes to ski in the mountains, but he does not know the Trentino territory.

The scenarios are instead the following:

- S1: Sara wants to have some advice to find an interesting location, possibly near her house, having the possibility to read a brief description about it. She tried to search on Google, but the results were too many and she lost an hour navigating without having any good results.
- S2: Micheal heard about the “Rifugio Pradidali”, but he has no idea on where it is and how to reach it, he would like to have a service to do so without any difficulties.
- S3: Vincenzo is looking for a ski run, but he does not know the different locations that Trentino has. He also wants to know when the ski run is open to rent skis and the telephone number to call and ask for specific information.
- S4: Vincenzo liked the place he found (owned by a company), he would like to go to other attractions owned by that company.

We think that these personas and scenarios cover all the main aspects of the application, simulating the principal user needs.

2.4 Metadata

Properties	Opendata Trentino	Schema.org
Description	Opendata Trentino is a data catalog containing public data about Trentino	Schema.org is a shared vocabulary for structured data on the Internet
Title	Opendata Trentino	Schema.org vocabulary
Dataset distribution	.	lov_schema.ttl
Keywork/tag	.	.
Theme/category	Distribution collection	General and Upper Ontologies
Version	.	v7-0
Update/modification date	.	31 March 2021, 15:08 (UTC+02:00)
Landing page	https://dati.trentino.it/	http://schema.org/

2.4.1 Data resources

From the different CSV datasets we selected these different attributes:

- Location: lat, lon, comune, address, indirizzo.
- Information: remoteId, titolo, informazioni, tipologia di luogo (category), leggi le informazioni dettagliate, descrizione breve, descrizione, impianto gestito da (company).
- Contact: indirizzo web, email, telefono.

For the JSON files instead, we selected these:

- Location: latitude, longitude, region, city, street.
- Information: _id, name, description, longDescription, serviceDescription, category.
- Contact: phone, mail, url.

2.4.2 Knowledge resources

For the knowledge resources files, we have these information:

- Place: address, geo (coordinates), lat,long, telephone, tourBookingPage (web sites), description, url, name, geoContains (Region).
- TourisiticDestination: touristType, Place properties.
- TouristAttraction: availableLanguage, touristType, Place properties.
- LandmarksOrHistoricalBuildings: Place properties.
- CivicStructure: openingHours, Place properties.
- Landform:Place properties.
- Organization:email,location, telephone, url,name

2.5 Knowledge resources

The knowledge resources we identified are taken from `schema.org` and are the following:

- Place: `https://schema.org/Place`
- TouristicDestination: `https://schema.org/TouristDestination`
- TouristicAttraction: `https://schema.org/TouristAttraction`
- LandmarksOrHistoricalBuildings: `https://schema.org/LandmarksOrHistoricalBuildings`
- CivicStructure: `https://schema.org/CivicStructure`
- Landform: `https://schema.org/Landform`
- Organization: `https://schema.org/Organization`

2.6 Data resources

The data resources we selected are taken from `dati.trentino.it` and are the following, in JSON or CSV format:

- Trentino-JSON: `https://dati.trentino.it/en_GB/dataset/punti-di-interesse-del-trentino`
- CSV: all the datasets from `https://dati.trentino.it/en_GB/dataset?tags=luoghi+e+punti+di+interesse`
- Valsugana-JSON:`https://dati.trentino.it/en_GB/dataset/punti-di-interesse-valsugana`

3 Inception

3.1 Purpose formalization and inception sheet description

From our purpose formalization, we extracted these competency questions:

Persona	Scenario	CQ	Kernel CQ
Sara	S1	List of all naturalistic attractions in Trentino	attractions, name
Sara	S1	Which are the attractions near my house?	attractions, location, latitude, longitude
Sara	S1	Read a description of one specific attraction	description, attraction
Micheal	S2	Where is the specific attraction x?	attractions, address, province, city, CAP, street, street number, commune
Micheal	S2	How much time do I need to reach it?	location
Vincenzo	S3	When is the attraction open?	attraction schedule
Vincenzo	S3	What is the price?	price
Vincenzo	S3	Is there a web site?	web site
Vincenzo	S3	Phone number for the attraction	phone number
Vincenzo	S3	Is there a parking area?	parking area
Vincenzo	S3	List of all places which are ski places	category
Vincenzo	S3	Which company owns the attraction?	company
Vincenzo	S3	Which attractions are owned by a company?	company, attraction

- List of all naturalistic attractions in Trentino
- Which are the attractions near my house?
- Read a description of one specific attraction
- Where is the specific attraction x?
- How much time do I need to reach it?
- When is the attraction open?
- What is the price?
- Is there a web site?

- Phone number for the attraction
- Is there a parking area?
- List of all places which are ski places
- Which company owns the attraction?
- Which attractions are owned by a company?

From the CQs we extracted these kernel concepts:

- Common: location, latitude, longitude, category, address, province, city, CAP, street, street number, commune, name.
- Core: attraction, company.
- Contextual: description, schedule, price, web site, phone number, parking area.

Location, latitude, longitude, address, province, city, CAP, street, street number, commune and category are common, because they are in all the DoI that handle geo-space data. Attraction and company are core, because they are the main topic of our service. All the other one are contextual, because they give information that do not belong directly to the DoI, but are still useful for giving a better and unique service.

3.2 Data resources

For the data collection we searched for different datasets in Open Data Trentino and on the web. Our criteria was to find datasets containing some of the categories from the CQ.

It has not been easy to find many datasets, because the information about naturalistic places are not as many as for example restaurants and similar point of interest. Talking about the different datasets listed in the previous chapter, we found out that the JSON file (Trentino) is well structured and has a lot of information, but it does not contain many objects (145) which match interesting categories (for example: 'rifugio','attrazione naturale'...).

For the csv files, each one of them has been defined by a commune, the information that they contain are good for our purpose. Most of them share the same schema. The problem we encountered with these datasets is that some of them have different schema, some have also missing fields in some objects or are encoded in a different way, making it difficult to integrate the data from these different datasets.

The last JSON (Valsugana) has the same structure as the first JSON (trentino) but it seems to have data that are present also in the other JSON file, we are keeping it for now, but it could be useless if it does not contain any more information than the other datasets.

One weakness of this process is the number of different datasets that we collected, like said before it is difficult to find information about naturalistic point of interest. From the datasets collected we can find almost all the core information that we need, but for some objects, contextual categories are missing (many are blank).

3.3 Knowledge resources

This is how we process our knowledge resources.

From our CQs we define these categories: name, address, location, category, description, url, telephone number, Region. We use these categories and the data collected as our criteria to define the usefulness of the schema. We only use the knowledge resource from one source because the other sources (link open data, Dbpedia home, Datascienta Home) didn't provide us better resources.

We chose the schema Place and its children because they are used in the geospace domain. Place has a lot of categories which feats our needs like address, latitude, longitude, telephone number…. The other schemas have these properties but also other one like touristType or openingHours which could be needed for our database. Landform and LandmarksOrHistoricalBuildings don't have any other properties but could be still interesting since they inherited from Place and are linked to our DoI. We also added a schema called organization, because we think that splitting the database in two tables, one for the point of interest and the other one for organizations, could be a good idea. Using this schema, we could add other useful information like the email or the name of the company, but for now we are not certain of this.

The weakness of this process is since we base our knowledge resource on one source, we don't have a lot of variety of information. But still, the resources are well documented, have a lot of other properties and fit our DoI pretty well, so we don't think we will have any problems with our schema.

3.4 Resource classification

As written before, we have taken two different types of datasets: the JSON and CSV. We think that first type is contextual, because, even if we don't have a lot of objects, each one has a lot of attributes, and all of these are complete. Since we have all these contextual information and even more, we think that these datasets are contextual. However the CSV type have a lot of objects, but their quality is not high, because almost all the common attributes have complete data, but the contextual attributes are often left blank. That's why we think that these datasets are common.

For the knowledge classification, we think that all the schema contain common and contextual elements. Since they contain all the information and represents our attraction and company we think they are core.

All these resources represent the core of our DoI (attraction and company), but there are no real attributes that represent these core concepts.

3.5 Evaluation

For the coverage, the values that we obtained are:

- Dataset: for ETypes 1, for properties 0.75
- Ontology: for ETypes 1, for properties 0.8

The values obtained for Etypes, both for dataset and ontology, means that we cover all the Etypes that we defined. For the properties we have some missing elements like parking area and price, but we think that dataset and schema cover well the CQ. For the extensivness, the values that we obtained are for ETypes 0.57 and for properties 0.25. For Etypes we found a lot of schemas regarding our domain of interest, but most of the schemas share the same properties, so the value for properties is lower. For the sparsity, the values that we obtained are for ETypes 0 and for properties 0.42. For ETypes there are no differences between CQs and datasets, so the value obtained is 0, while for the properties, as said before, they are quite the same for different schemas, this affected the value that we found. In general we found out that the datasets we have provide only information in the CQs, but not other information. We think that for our domain and purpose this is fine and no other information are needed.

4 Informal Modeling

4.1 Purpose formalization and informal sheet description

From the inception phase we obtained the following kernel competency questions:

- Common: location, latitude, longitude, category, address, province, city, CAP, street, street number, commune, name.
- Core: attraction, company.
- Contextual: description, schedule, price, web site, phone number, parking area.

We classify only 4 concepts:

- Object: attraction, company, location, address.
  These kernel CQ are objects because they are real physical concepts or need to be detailed with data properties.

  In our opinion our other concept don't need more information or can't be added in the one of the 3 categories.

We selected the following foundational primitives:

- EType: attraction, location, address, company.
  These are ETypes, because they are concepts having attributes, they can not be defined by them self for our purpose.
- Object property: has location, has address, has company, has attraction.
  We created these properties to represent the connection between an attraction and a company (an attraction is owned by a company, a company owns one or more attractions) and the link between address and location with company and attraction.
- Data property: latitude, longitude, description, province, city, CAP, street, street number, commune, schedule, price, web site, phone number, parking area, category.
  These are data properties, because they define the ETypes.

The EType attraction represents the naturalistic point of interest, the most important element of our purpose, it includes the following properties:

- name: the attraction name
- description: the description of that point of interest, a text containing information about the place
- category: the type of the attraction, it tells the user what kind of attraction it is
- parking area: it tells whether there is a parking area near the attraction
- has location: it gives the attraction location
- has address: it gives the attraction address
- has company: it tells which company owns the attraction

The EType company represents the company that owns an attraction, it includes the following properties:

- name: the company name
- schedule: the opening hours
- price: the price given by the company
- web site: the url to the company web site
- phone number: the company phone number
- has location: it gives the company location
- has address: it gives the company address
- has attraction: it gives the attractions owned by that company

The EType location represents the geo-coordinates of a place, it has no information about the place itself but only its geographic coordinates, it includes the following properties:

- latitude: the latitude of the place
- longitude: the longitude of the place

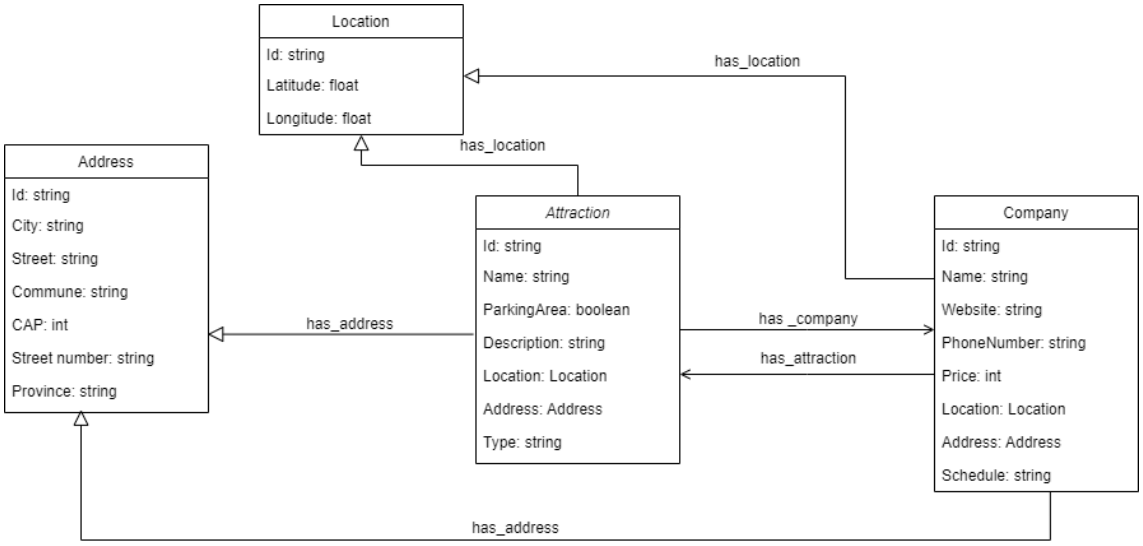
The EType address represents a place location, but in a human understandable way, its properties are the following:

- province: the province that contains the place
- city: the city in which the place is
- CAP: the postal code of the place
- street: the street where the place is located
- street number: the place building's number
- commune: the commune that contains the place

We did not define more general properties like nation and region, because for our purpose we consider only places in the Trentino territory.

4.2 ER model description

The ER model that we created is the following:



From our propose we defined the two main ETypes: attraction and company. We did not define any subclass, the only doubt we had was about the attraction's type, which could become a subclass. We decided to keep it as a simple attribute because, at least for now, we didn't have specific attributes to assign to the specific subclass. This choice allows us to create a more general dataset, that could be shared to other projects.

4.3 Dataset selection

For the different CSV datasets we made the selection looking for files that contain at least one object that contains an "interesting" type (the attribute "Tipologia di luogo" represents the type of attraction, we selected some of them according to our purpose). The result that we obtained is 83 interesting datasets over 135, consisting of 908 objects.

For the JSON datasets we kept only the "Trentino" one and not the "Valsugana", because we found out that the second is a subpart of the first and does not give any additional information.

The number of interesting type that we defined is 63, these are stored in the file "categories.txt" inside our repository, they are simple strings like "Ghiacciaio", "Noleggio attrezzature sportive", "Parco naturale", these interesting types represent the kind/type of attraction. We decided to use this information also to select the useful datasets and objects in order to provide as result only information of naturalistic point of interest.

4.4 Evaluation

4.4.1 ER vs CQ

For the coverage, the values that we obtained are:

- ETypes: 1
- Properties: 1

The ER model matches perfectly the CQ list, so the value obtained is 1. For the extensivness, the values that we obtained are:

- ETypes: 0
- Properties: 0

Since the matching between CQ and ER model is perfect, the ER model doesn't try to add additional information.

4.4.2 ER vs Dataset

For the coverage, the values that we obtained are:

- ETypes: 1
- Properties: 0.71

Our datasets do not cover all the CQ, in case of the Etypes the match is perfect, but some properties are missing, so the value obtained for them is not 1. The result is still acceptable, because 0.71 means that many properties are in the datasets. For the sparsity, the values that we obtained are:

- ETypes: 0
- Properties: 0.28

The values obtained were expected, that is because there are no significant differences between Etypes and properties of ER model with respect to the other in the datasets.

5 Formal Modeling

5.1 ETG generation

To create our ETG we used our ER MODEL and the software protégé to define for each class the object and data properties associated. During this process we also tried to modify the name of our data properties by the names inside of the ontology as much as we could and without modifying the meaning that we gave them. For example we modified website by url or PhoneNumber by telephone or CAP by postal code which comes from the schema PLACE which comes from our knowledge resources (look part 2.4). But we didn't modify location by geocoordinates because for us it's too precise and not clear. We did this because it improves the reusability and sharebility.

For our classes Attraction and Company we defined a new section called non living being which could be defined as things which are lifeless. In fact we thought that our classes were real kind but were not living being since we talk about places.

For our object property ”has company” we indicate that it is the inverse of ”has attraction” so that from a company we can get all its attractions and from an attraction get its company.

For each classes that we have created we had to define it’s data properties and the meanings of each data property (Language alignment) thanks to UKC.

Here what we have in the end:

Address: the place where a person or organization can be found or communicated with

- ID: a symbol that establishes the identity of the one bearing it
- city: people living in a large densely populated municipality
- street: the address where a person or organization can be found
- commune: the smallest administrative district of several European countries
- postal code: a code of letters and digits added to a postal address to aid in the sorting of mail
- street number: number of the street
- province: the territory occupied by one of the constituent administrative districts of a nation

Location: A determination of the place where something is

- ID: a symbol that establishes the identity of the one bearing it
- latitude: a number that identifies a position relative to an axis
- longitude: a number that identifies a position relative to an axis

Attraction: An entertainment that is offered to the public

- ID: a symbol that establishes the identity of the one bearing it
- name: a language unit by which a person or thing is known
- parkingArea: space in which vehicles can be parked
- description: a statement that represents something in words
- type: identify as belonging to a certain type

Company: An institution created to conduct business

- ID: a symbol that establishes the identity of the one bearing it
- name: a language unit by which a person or thing is known
- url: the address of a web page on the world wide web
- telephone: the number is used in calling a particular telephone
- price: the amount of money needed to purchase something
- schedule: an ordered list of times at which things are planned to occur

5.2 Data management

We had 2 objectives: First split the different value for each Etypes: for example for the Etype address all the information were in one single line. So we had to split that in order to fill our data properties and also have one value per cell. To do this we had to analyze the address string that we have in our datasets (it is a big string containing several information about the address). We had to find a way to split the different information of the string and classify them. We split the different elements checking where there is a coma or space, then we identified which kind of information that part of the string contains. We did it with different techniques:

- For Postal code and Street number we searched for number inside the address string, if the number is greater than 38000 it is a postal code (all the Trentino's postal code are 5 digits number starting by 38), if instead it is lower than 1000 it is a street number.
- For the Street information we check whether there are some common words like "Via" or "Piazza".
- For the City we used the complete list of all the cities in Trentino and searched for them inside the address string.
- For the Commune we simply used the name of the different CSV files, because the CSV are created by each commune, so this information can be used to assign a commune to each attraction of a CSV file.

The second objective is to do the syntactic heterogeneity : The main problem was latitude and longitude. All the other information were already syntactically hetero-gene. We had to fix the type since some of them were integer (0) and other were float but there were no problem of format. All the CSV elements were stored as strings, so we had only to check the elements that must be integer or floats in our ETG, for example latitude, longitude, street number and postal code (CAP). For empty elements of type string we decided to put a 'none', to show when we don't have some informations, while for floats we put -1.0 and for integers -1. We decided these values because they are not possible values for our elements (for example latitude and longitude that are float elements can not be -1.0, this applies to all the float and integer elements).

5.3 Evaluation

5.3.1 Sparsity

We obtained the following values for sparsity:

- Etype: 0
- Properties: 0.46

All the Etypes of the ETG are inside the selected ontologies, so the value obtained is supposed to be 0. For the properties instead the situation is different, only half of the properties from the ontologies are inside our ETG, because most of them were not easily accessible/found (example: logo and review were part of the ontologies that we found, but they are not useful in our domain). So we did not add them in the ETG.

5.3.2 Cue validity

For CUE validity the value obtained can be seen in the following table:

Property	Attraction	Company	Location	Address
ID	0.25	0.25	0.25	0.25
Latitude	0	0	1	0
Longitude	0	0	1	0
Telephone	0.5	0.5	0	0
Description	1	0	0	0
Name	0.5	0.5	0	0
URL	0	1	0	0
PostalCode	0	0	0	1
Street	0	0	0	1
OpeningHours	0	1	0	0
Type	1	0	0	0
Price	0	1	0	0
City	0	0	0	1
Commune	0	0	0	1
Street number	0	0	0	1
Province	0	0	0	1
Parking area	1	0	0	0
Schedule	0	1	0	0
Cue	0.71	0.75	0.75	0.89

The Cue value for ETG is 3.1.

5.3.3 Cue interpretation

The Cue value of the table (last row) has been obtained summing up the values of each column and normalizing the result dividing it by the number of properties of the column Etype. The normalized Cue that we got for each Etype tells us that each Etype does not share a lot of properties with the other, which is normal because each one is unique.

6 Data Integration

6.1 Data management

6.1.1 Entity Alignment

For the entity alignment we get all the different name of the columns in our csv and json and we map this list of names to a word. For example ”desc” and ”description breve” are merged and mapped to ”description”. This is the only case where we had to modify the columns, all the other were fine (after changing their name according to the schema that we have decided).

6.1.2 Entity matching

To identify if two attractions were the same we use their names and coordinates:
For names we used two algorithms :
the first one look for the 2 objects that have the same exact name and delete the second one. When we found the duplicates (around 61) we check by hands that they didn't have any different information from one to another. In here we delete the second without a specific reason, that is because the elements that have exactly the same name are exactly the same, so if I find two times "ATTRACTION1" then I'm sure that all the information related to that attraction are exactly the same (we found this out checking by hand the duplicates), this is due because some CSV files contain the same attraction.
For the second one we used the Jaro similarity to compute the percentage of similarity between two words. To have the similarity between two attraction names s1 and s2 it uses this formula:

$$JaroSim(s1, s2) = (1/3) * (C/|s1| + C/|s2| + (C - T)/C)$$

where C is the number of characters in common between s1 and s2.
T is the number of characters different between s1 and s2 divided by 2
Example: "DEIS" vs "DESI"
C=4(we find all characters from s1 in s2)
S=2/2 (the two last characters in the two words are not in the same position and need to be replaced)
|s1|=4
|s2|=4
JaroSim(s1,s2)=0,91
We used this formula and took a threshold of 90 percents of similarity because we think it gave words which are similar enough to be consider duplicates and not taking in account the false positives (found out checking by hand). Also we added a security, for example we deleted stop words. If you have two phrases :"Lago di Trento" and "Lago di Roma" the words "Lago" and "di" are not really interesting and the jaro formula will think that they are similar because of these two words, so when we have a word in the two names we delete it. Also we delete some words even if they appear only in one name like "di" "del" etc...

For coordinates we compute the distances between 2 geo coordinates. Unfortunately the minimun distances between 2 attractions was 200 meters and for us our threshold to decide that 2 attractions were similar was 50m.This was due to the fact that not all the objects had coordinates (around 15 percents) and we only used the one that have more than 6 digits after the coma to compute the distances, because we wanted to be precise, transforming coordinates into distance in meters requires the coordinates to be really precise. So we abandoned this idea, since we had a lot of coordinates that were not really precise (like 46.5 we had a lot of duplicates).

For the Etype location we could think from the distances algorithm that there is no duplicates but the thing is that there are a lot of coordinates that are not precise example : 46.5 and so we

have some duplicates caused by this imprecision. Due to this problem we decide to consider two locations the same only if they are exactly the same (latitude1 = latitude2, longitude1 = longitude2), we know that doing this we don't get all the duplicates, but it was the best idea that came in our minds.

For address we have to check that all its data properties are exactly the same to consider them as identical. So we don't have a lot of duplicates. In this case we did not use Jaro distance, because the attributes of address that are string are often very short, so the Jaro similarity would have provided us wrong results. As for location also in this case maybe we don't match all the duplicates, but we were able to find many of them.

For company we had to find a way to find duplicates also without having name, price and opening hours (for some companies we have their name, but not for all of them, instead for price ad opening ours we don't have any information). The only thing that we could do is the comparison by all elements, so two companies are the same only if their attributes are all the same. This brings us again the issue that we probably don't find all the duplicates.

After handling the duplicates we assign IDs using a string prefix (”ATT”, ”COM”, ”LOC”, ”ADD”) and a counter. For location and address we put the ID 0 to the ”empty element”, that is the entity for which we don't have any information (like lat:-1, lon:-1 for location), we did this in order to see easily if an attraction or company has a real address and location or if that information is missing. Choosing identifiers like these allows us to simply identify the entities.

6.1.3 Semantic heterogeneity

In case of conflicts between 2 entities here what we do (this control is done before dividing the row data, so the merge is done using lines that have information about all the 4 entities):

- Id: for the four different ids we keep the IDs of the first entity, this is because this information will not be used, because the ID will be regenerated.
- Name: we keep the name of the first attraction, this is because, if two lines are the same, this means that the name are equals or similar (by Jaro similarity), so both the names would be ok.
- ParkingArea: for parking area, that is a boolean, we keep true if at least one of the two entities have true, otherwise false. This is because we think that, if an elements have a parking area and the other one (that should represent the same attraction) has not, we can rely on the fact that a parking area exists. Unfortunately we don't have any information about parking areas in our datasets, so this control will always return false.
- Description: For example if we had 2 descriptions about the same attraction we keep the biggest one. Why ? It's because we found that most of the time the descriptions share the same information.
- Telephone: For telephone we keep both number.

- Longitude and latitude: We try to keep the one with more digits.
- type, name, opening hours, url: we take the first one. Why? Because we hadn't a clear way to decide which one to take in case of conflict so we take a deterministic decision.
- address: for all the address attribute we decide in block, this means that we will keep the address of the first line or the address of the second one, not some merge of the two, this in order to avoid the creation of a new, probably not existing, address. We simply decide based on the number of elements that contain useful informations (that are not blank). In the end we keep the address with more informations.

6.2 Data integration phase evaluation

6.2.1 Sparsity

We obtained the following values for sparsity between ETG and DS:

- Properties: 0

All the properties from our ETG are found in our DS but some of them have no values or few so if we wanted to be exact the value should be : 0.20 (since we don't have enough values for : schedules, price, parkingArea,name of company,province

6.2.2 CQ answered

This is CQ we could answered thanks to our EG:

- CQ1 List of all naturalistic attractions in Trentino: we can get all the attraction for a specific commune
- CQ2 Which are the attractions near my house? : we can get the location/address of an attraction and thanks to a third service compute the distances between the coordinates of the user and the attractions one
- CQ3 Read a description of one specific attraction: we can get the description of an attraction.
- CQ4 Where is the specific attraction x?: we can get the location and the address of an attraction
- CQ5 How much time do I need to reach it?: same thing as CQ2.
- CQ6 Is there a web site?: we can get the web site of a specific attraction
- CQ7 Phone number for the attraction: we can get the phone number of a specific attraction
- CQ8 List of all places which are ski places: we can get all the attraction of a specific type
- CQ9 Which company owns the attraction?: We can get the company of an attraction

- CQ10 Which attractions are owned by a company? : We can get all the attraction of a company.

We couldn't answered to this CQ: When is the attraction open?,What is the price?,Is there a parking area? Because we don't have this information in our Databases. We have implemented our system in order to be able to answer the questions, the problem is that the answer will always be the same ('none' for opening hours, '-1.0' for the price and 'false' for the parking area). So from our 13 CQ we can answer 10 CQ which mean that we have 76 percents of success.
For the last phase, we had one main issue: When we try to get the location and the address of an attraction by its name thanks to the object properties has location. It gave us two locations and addresses. Even if in our DB and rdf file we have only one location object link to the attraction. The problem was that the ID of the 4 etypes were the same : it was just a number. So we suppose that when we get the attraction by its name graphdb only return its id. and so when it looks for the location link to this ID it will return 2 objects. Why? Because, since the id is the same for all Etype,s he will get the location for the id attraction but also for the id company since its the same structure. Even when we try to only get the one with the type attraction thanks to rdf:type it returns both. So we decided to modify our ID by ATT1 for attraction COM1 for company LOC2 for location and ADD1 for address, doing this we solved the problem.

7 Open Issues

7.1 Missing data

The first problem, and maybe the most important, that we encountered is the missing data. All the datasets that we collected were good for our purpose, but they had many blanks inside them. For example in the JSON we had a field that told us the company that owns the attraction, but often this information is missing. This problem is of the data that we collected. Another problem related to the dataset that we found is that not all the columns/properties we were looking for are inside these datasets, so to complete our purpose we would have needed other datasets containing these information. For example the price and opening hours are contextual information that we wanted, but we couldn't find. This problem could have been partially solved by scraping the description, because sometimes, inside the description, we can find information related to opening hours for example, the problem is that this is a difficult thing to do and we decided to dedicate our time to the other phases, because this would have taken much effort and probably for not good results since the descriptions don't often contain the opening hour information and it is not easy to get this information from a long text with no predefined format.

7.2 More information

When we were creating the ER model during the Informal modeling phase, we thought about creating some sub-classes of attraction based on its type. The idea was to be more specific about some kind of attractions, giving some special attributes related to them. For example a sub-class could have been something like "mountain" with a new attribute indicating for example

the height. The problem that we encountered when thinking of this idea was the following: the mountain example is fine, but are there other type of attraction that could have interesting additional attributes? Also finding more specific information would not be easy (given also the difficulty to find the one that we decided from beginning, see section before). We decided then to keep it more general, also in order to provide a general service, hoping to give the user significant information instead of more specific ones but with less significant data.

7.3 Entity matching

As described in the Data integration phase, one challenge that we encountered was the entity matching. Understanding whether two entities are the same or not is not easy. The first approach that came in our minds was to do it using the coordinates of the attraction. The idea is the following, if two attractions are ”near” each other, then it is probable that the two are instead one. The ”near” thing in our mind was a range of 50m. In our opinion this control, combined with one for the name, could have been the ideal solution. The problem we encountered trying to do this is the precision of the location. Unfortunately, not many latitude and longitude values are precise enough to be used to compare two attractions. If for example we have a latitude value like 46.08 and a value like 46.0886, are those near? How can I know if the 46.08 is something like 46.0800 or 46.0899? This was a big problem that we could not solve, that is only a trivial example, but for computing distances in order of 50 meters, we would have needed more digits, the data that we have don’t provide us information that specific. Also for some attractions we don’t even have their location, we are talking about almost 15% attractions without location. Given that we could not use the location to find duplicates, we decided to use the name, to do this we used the Jaro distance, as we previously said in the data integration section. We still believe that a combined approach would have been the best to find only true duplicates (without having to check by hand).

8 Outcome exploitation

8.1 KG information

In the end we have 4 Etypes : company, attraction, location and address. In total we have 18 properties inside our KG.For company we have 243 entities, for attraction we have 956 entities, for location we have 505 entities, for address we have 221 entities. We have 4 object properties: has_location, has_address, has_company and has_attraction.

- has_location: can get the object location link to an attraction or a company (for now it will provide useful data only for attraction because we didn’t get any information about the location of company so they are all empty) (domain: attraction,company range: location)
- has_location: can get the object location link to an attraction or a company (for now it will only work with address because we didn’t get any information about the location of company so they are all empty) (domain: attraction,company range: location)

- has_address: can get the object address link to an attraction or a company (for now it will only work with address because we didn't get any information about the location of company so they are all empty) (domain: attraction,company range: address)
- has_attraction: can get all the attractions link to the company (domain: company range: attraction)
- has_company: can get the company link to the attraction (domain: attraction range: company)

Here we describe the type of each properties and assign it to its class.

- Address: ID(string), City(string), Street(string), Commune(string), Postal code(int), Street number(string), Province(string).
- Attraction: ID(string), Name(string), Parking area(Boolean), Description(string), Type(string)
- Location: ID(string), Latitude(float), Longitude(float)
- Company: ID(string), Name(string), URL(string), price(int), Opening hours(string)

8.2 KG exploitation

Our KG can be used for multiple cases:

- 1.If you want to have more information about an attraction (address, location, specific information)
- 2.If you want to look for some attraction near you (thanks to a third service)
- 3.If you want to look for an attraction by commune or city
- 4.If you need contact information about an attraction (web sites, phone number)
- 5.If you want to know other attractions owned by a company

For the 1st case described above, an example using SPARQL can be the following:

```
#where is an attraction by its name
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
select ?lat ?lon ?street ?streetnum ?city ?commune ?postalcode ?province where {
    ?attraction prop:has_name_GID-2_Type-82091 "Passo San Pellegrino" ;
    prop:has_location_GID-93733_Type-82091 ?loc;
    prop:has_address_GID-93733_Type-82091 ?add.
    ?loc prop:has_latitude_GID-46263_Type-779 ?lat;
    prop:has_longitude_GID-46270_Type-779 ?lon.
    ?add prop:has_city_GID-45969_Type-45803 ?city;
    prop:has_street_GID-24034_Type-45803 ?street;
    prop:has_street_number_GID-300000_Type-45803 ?streetnum;
    prop:has_postal_code_GID-34110_Type-45803 ?postalcode;
    prop:has_commune_GID-45992_Type-45803 ?commune;
    prop:has_province_GID-46567_Type-45803 ?province.
}
```

	lat	lon	street	streetnum	city	commune	postalcode	province
1	"46.393068039941 84"^^xsd:float	"11.7996654119587 8"^^xsd:float	"Passo San Pellegri no"	"none"	"Moena"	"none"	"38035"^^xsd:integer	"none"

```
#get the description of an attraction by its name
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
select ?description where {
    ?attraction prop:has_name_GID-2_Type-82091 "Passo San Pellegrino" ;
    prop:has_description_GID-3_Type-35453 ?description.
}
```

	description
1	"Il Passo San Pellegrino è una delle località più spettacolari dell'Universiade Invernale Trentino 2013. Circondata dalle Dolomiti, questa località sciistica offre più di 60 km di piste da sci di lunghezza diversa e di livello di difficoltà variabile che si snodano su pedii tra i 1.918 e i 2.513 m.s.l.d.m.. Una di queste piste, la pista «Cima Uom o», ospiterà le gare di discesa libera e slalom gigante di Trentino 2013."

For the second one:

```
#list the 10 attraction nearest to us, ordered by distance
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
PREFIX f: <http://www.ontotext.com/sparql/functions/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?distance WHERE {
    BIND( 3963.0*f:acos(f:sin((46.067016619670284*f:pi())/180)*f:sin((?
lat*f:pi())/180)+f:cos((46.067016619670284*f:pi())/180)*f:cos((?lat*f:pi())/180)*
f:cos(((11.150280971722998-?lon)*f:pi())/180))  AS ?distance)
 ?attraction rdf:type prop:Attraction_GID-35453 ;
 prop:has_name_GID-2_Type-82091 ?name;
 prop:has_location_GID-93733_Type-82091 ?loc .
    ?loc prop:has_latitude_GID-46263_Type-779 ?lat;
 prop:has_longitude_GID-46270_Type-779 ?lon.
    FILTER(?lon !=-1.0)
}
ORDER BY ASC(?distance)
LIMIT 10
```

	name	distance
1	"Funivia Trento - Sardagna"	"1.6823682324036424"^^xsd:double
2	"Maranza"	"1.7613121958670652"^^xsd:double
3	"Lago di S. Colomba"	"2.0728351396172364"^^xsd:double
4	"Ecomuseo dell'Argentario"	"2.1387242051262616"^^xsd:double
5	"Palaghiaccio di Trento"	"2.242263798124643"^^xsd:double
6	"Forte di Civezzano"	"2.5668849772066356"^^xsd:double
7	"Civezzano"	"2.5668849772066356"^^xsd:double
8	"Trento Funivie S.P.A. - Monte Bondone"	"3.8872774292378773"^^xsd:double
9	"Maestri di Sci Azzurra Monte Bondone"	"3.8872774292378773"^^xsd:double
10	"Sport Nicolussi"	"3.8872774292378773"^^xsd:double

For the third one:

```
#list all the attraction of a specific type and commune
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
select ?attraction ?name where {
    ?attraction prop:has_type_GID-31834_Type-35453  "Località turistica" ;
    prop:has_address_GID-93733_Type-82091 ?add ;
    prop:has_name_GID-2_Type-82091 ?name .
    ?add prop:has_commune_GID-45992_Type-45803 "Comune_di_Pieve_di_Bono-Prezzo" .
}
```

	attraction	name
1	http://localhost:8080/source/ATT304	"Boniprati d'inverno"
2	http://localhost:8080/source/ATT305	"Pieve di Santa Giustina"

For the fourth one:

```
# give an attraction owner, website and phone number given its name
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
select ?name ?website ?telephone where {
    ?attraction prop:has_name_GID-2_Type-82091 "CAMPING RIVIERA" ;
    prop:has_company_GID-83703_Type-35453 ?com.
    ?com prop:has_url_GID-34123_Type-43715 ?website;
    prop:has_telephone_GID-34494_Type-43715 ?telephone;
    prop:has_name_GID-2_Type-82091 ?name.
}
```

	name	website	telephone
1	" Bellavista-Riviera snc"	"http://www.camping-riviera.net"	"(0039) 0461 724464"

For the fifth one:

```
#list all the attractions owned by a company, given the company name
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
select ?name where {
    ?company prop:has_name_GID-2_Type-82091  " SPORT CERMIS" ;
    prop:has_attraction_GID-111283_Type-43715 ?attraction.
    ?attraction prop:has_name_GID-2_Type-82091 ?name .
}
```

	name
1	"SPORT CERMIS FUNIVIA"
2	"SPORT CERMIS DOSS DEI LARESI"

To summarise our application can be used to get specific or geo-spatial information about an attraction in Trentino. It can be used by a third service for touristic purposes, to help the user to find its point of interest or get one.
There is one main issue with this applications: lake of information for the price, name of the company and opening hours of the attraction. That weren't in the original databases.