UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO DI INFORMATICA
Corso di laurea in Data Science

PROGETTO IN GESTIONE DATI STRUTTURATI E NON STRUTTURATI

Studenti:

- ***Alaimo Giuseppe***
- ***Lovecchio Daniele***

INTRO

Il progetto, realizzato dagli studenti Alaimo Giuseppe e Lovecchio Daniele, ha come soggetto d'analisi la regione Basilicata.

L'obiettivo è analizzare diversi temi, tra cui turismo, infrastrutture, aziende multifunzionali (Fattorie e Industrie sul territorio) etc.

L'indagine è stata condotta raccogliendo ed elaborando dati di tipo strutturato e di tipo spaziale.

INDICE

RACCOLTA DATI

DATI NON STRUTTURATI

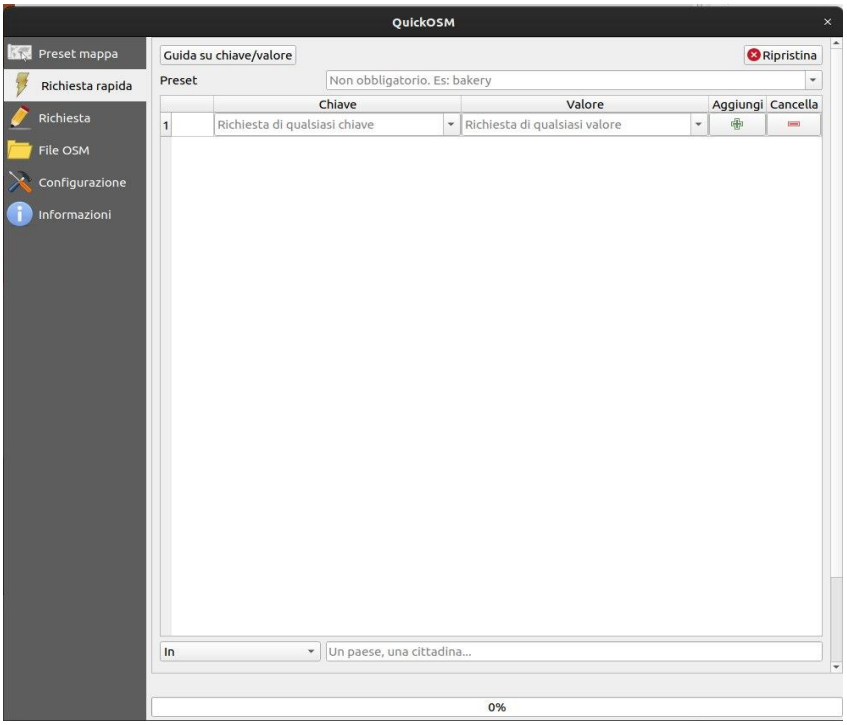
La raccolta di dati non strutturati, e quindi spaziali, è stata condotta su ***OpenStreetMap*** (**OSM**), ovvero un progetto open source che raccoglie dati geografici mondiali con lo scopo di creare mappe e cartografie.

OSM, quindi, ci ha permesso di ottenere dati inerenti a:

- Città della Basilicata (punti)
- Confini amministrativi (poligoni)
- Fiumi (linee)
- Stazioni ferroviarie (punti)
- Ferrovie (linee)
- Strade (linee)
- Monti e colline (punti)
- Fattorie (punti)
- Industrie (poligoni)
- Musei (punti)
- Chiese (poligoni)
- Attrazioni turistiche (punti)
- Hotel (punti)

Sfruttando il plugin **Quick Osm**, messo a disposizione da OpenStreetMap per QGis, è stato possibile ottenere i layer di nostro interesse, ovvero quelli sopra elencati.

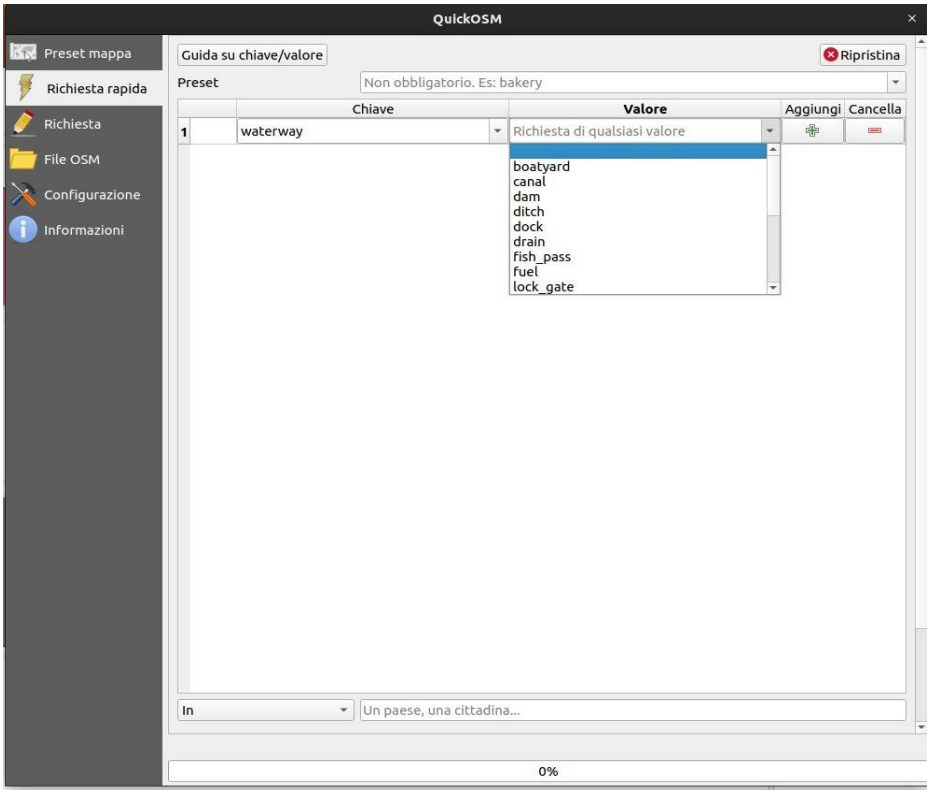
L'interfaccia con il plugin si presenta in questo modo:



Come si può notare, ci viene richiesto di inserire una coppia chiave-valore che identificano un determinato layer d'interesse e il territorio associato ad esso.

La comunità di OSM ha concordato sul determinare la combinazione di chiavi e valori per poter estrapolare dei layer ben precisi.

Nell’esempio sotto mostrato, si utilizza come chiave “waterway” e vediamo che ci vengono mostrati i diversi valori associati a quella chiave.



Una volta scelto il valore, otterremo il layer di nostro interesse. Per l’elenco completo di chiavi-valori disponibili basta visitare il sito https://wiki.openstreetmap.org/wiki/Map_features.

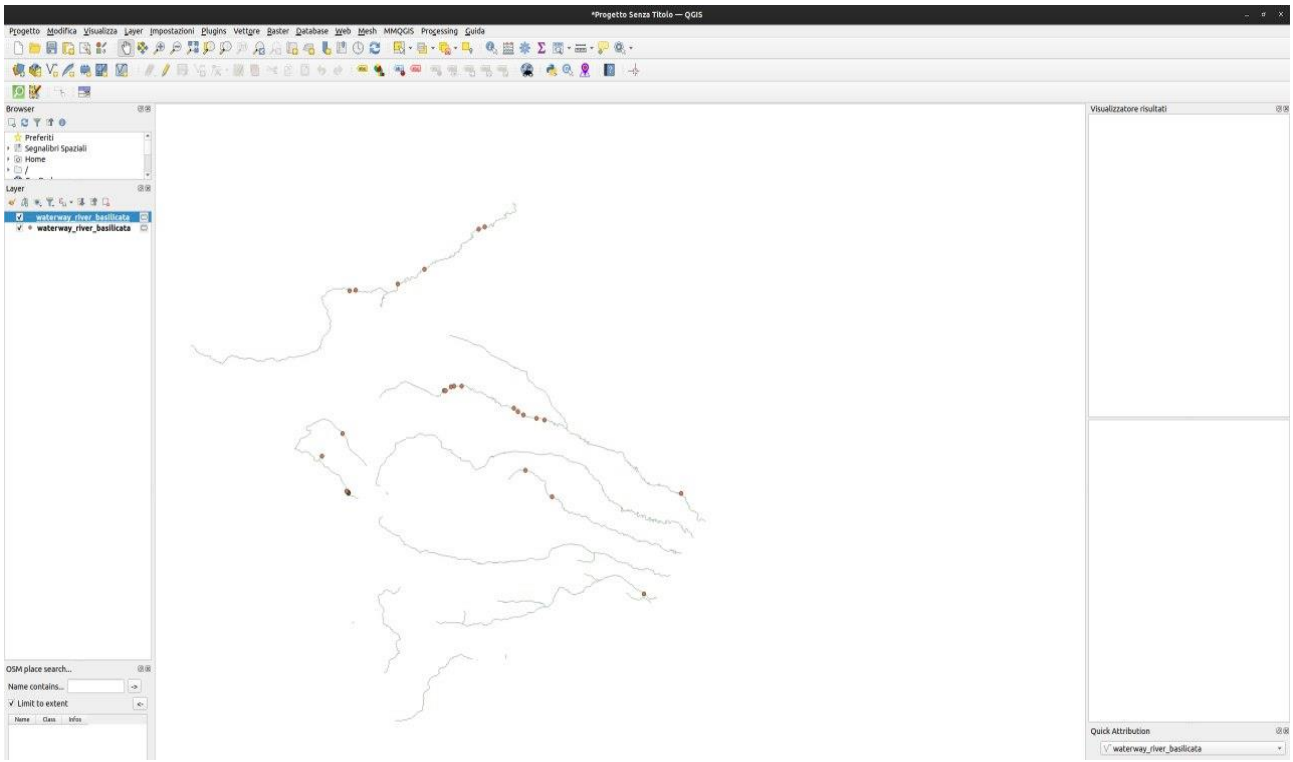
Waterway

This is used to described different types of waterways. When mapping the way of a river, stream, drain, canal, etc. these need to be aligned in the direction of the water flow. See the page titled **Waterways** for an introduction on its usage.

Natural watercourses

Key	Value	Element	Description	Map rendering	Image	Count
waterway	river		The linear flow of a river, in flow direction.			2 331 1 591 217 24 158
waterway	riverbank		The water-covered area of a river			0 4 058 4 385
waterway	stream		A naturally-forming waterway that is too narrow to be classed as a river.			6 699 15 736 902 32 880
waterway	tidal_channel		A natural intertidal waterway in mangroves, salt marshes and tidal flats with water flow in the direction of the tide			1 5 531 9

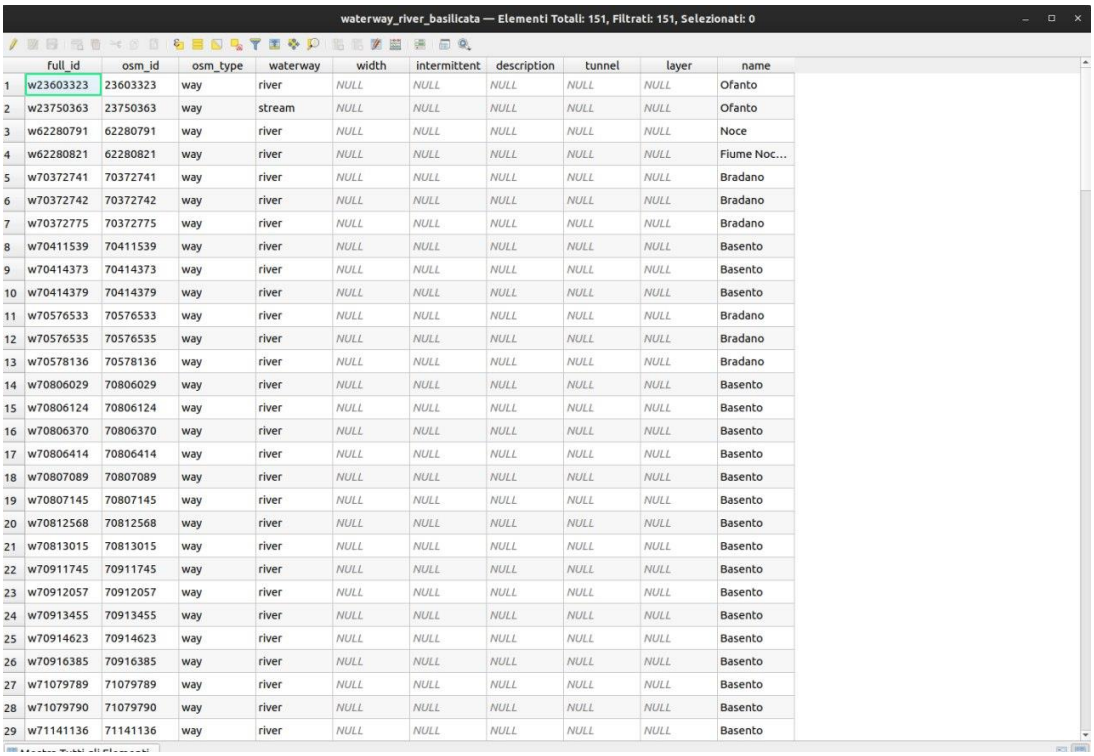
Di seguito viene mostrato come viene visualizzato su QGis il layer selezionato, in questo caso, il valore scelto è “river”, che identifica il layer dei fiumi.



Una volta importato il layer, sarà possibile visualizzare la tabella degli attributi associata ad esso.

La tabella è composta da una serie di attributi strutturati e un attributo di tipo non strutturato.

Prendendo come esempio sempre il layer dei fiumi, la tabella presenterà attributi che riguardano dati come il nome, la larghezza o la descrizione e un attributo che identifica la sua posizione sulla mappa e il tipo di geometria, in questo caso “LINESTRING”.



	full_id	osm_id	osm_type	waterway	width	intermittent	description	tunnel	layer	name
1	w23603323	23603323	way	river	NULL	NULL	NULL	NULL	NULL	Ofanto
2	w23750363	23750363	way	stream	NULL	NULL	NULL	NULL	NULL	Ofanto
3	w62280791	62280791	way	river	NULL	NULL	NULL	NULL	NULL	Noce
4	w62280821	62280821	way	river	NULL	NULL	NULL	NULL	NULL	Fiume Noc…
5	w70372741	70372741	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
6	w70372742	70372742	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
7	w70372775	70372775	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
8	w70411539	70411539	way	river	NULL	NULL	NULL	NULL	NULL	Basento
9	w70414373	70414373	way	river	NULL	NULL	NULL	NULL	NULL	Basento
10	w70414379	70414379	way	river	NULL	NULL	NULL	NULL	NULL	Basento
11	w70576533	70576533	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
12	w70576535	70576535	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
13	w70578136	70578136	way	river	NULL	NULL	NULL	NULL	NULL	Bradano
14	w70806029	70806029	way	river	NULL	NULL	NULL	NULL	NULL	Basento
15	w70806124	70806124	way	river	NULL	NULL	NULL	NULL	NULL	Basento
16	w70806370	70806370	way	river	NULL	NULL	NULL	NULL	NULL	Basento
17	w70806414	70806414	way	river	NULL	NULL	NULL	NULL	NULL	Basento
18	w70807089	70807089	way	river	NULL	NULL	NULL	NULL	NULL	Basento
19	w70807145	70807145	way	river	NULL	NULL	NULL	NULL	NULL	Basento
20	w70812568	70812568	way	river	NULL	NULL	NULL	NULL	NULL	Basento
21	w70813015	70813015	way	river	NULL	NULL	NULL	NULL	NULL	Basento
22	w70911745	70911745	way	river	NULL	NULL	NULL	NULL	NULL	Basento
23	w70912057	70912057	way	river	NULL	NULL	NULL	NULL	NULL	Basento
24	w70913455	70913455	way	river	NULL	NULL	NULL	NULL	NULL	Basento
25	w70914623	70914623	way	river	NULL	NULL	NULL	NULL	NULL	Basento
26	w70916385	70916385	way	river	NULL	NULL	NULL	NULL	NULL	Basento
27	w71079789	71079789	way	river	NULL	NULL	NULL	NULL	NULL	Basento
28	w71079790	71079790	way	river	NULL	NULL	NULL	NULL	NULL	Basento
29	w71141136	71141136	way	river	NULL	NULL	NULL	NULL	NULL	Basento

Anche se QGis non permette di visualizzare l'attributo geometrico, questo potrà essere visualizzato copiando la tabella in un file Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	wkt_geom	full_id	osm_id	osm_type	waterway	width	intermitte	descriptio	tunnel	layer	name		
2	LineString (15.3269943000000	w23603323	23603323	way	river						Ofanto		
3	LineString (15.0932300000000	w23750363	23750363	way	stream						Ofanto		
4	LineString (15.8099054999999	w62280791	62280791	way	river						Noce		
5	LineString (15.8055967000000	w62280821	62280821	way	river						Fiume Noce o Castrocucco		
6	LineString (16.8250118999999	w70372741	70372741	way	river						Bradano		
7	LineString (16.7919443999999	w70372742	70372742	way	river						Bradano		

DATI STRUTTURATI

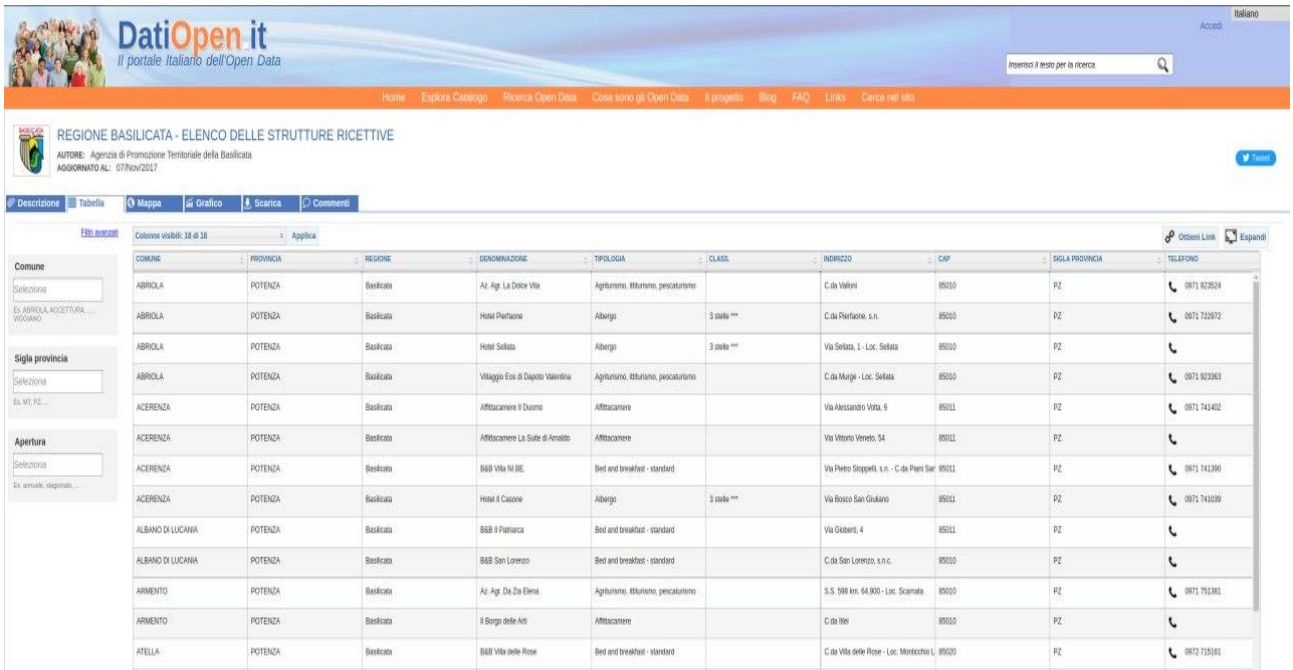
La raccolta di dati strutturati è stata condotta in parte su OpenStreetMap, poiché, come abbiamo visto, oltre ai dati spaziali ci vengono forniti anche alcuni dati strutturati, e in parte su altre fonti più specifiche.

Le altre fonti sono:

- Open Data Regione Basilicata (http://dati.regione.basilicata.it/catalog/);
- DatiOpen.it (http://www.datiopen.it/it/opendata/);
- Recensioni Google.

Dai primi due siti sono state ricavate informazioni aggiuntive riguardanti hotel, attrazioni turistiche, chiese e musei.

Di seguito viene riportato, come esempio, l'elenco delle strutture ricettive della regione Basilicata, facilmente reperibile in formato CSV (Comma-Separated Values).



Ottenuto un numero sufficiente di dati, questi sono stati accorpati con quelli già reperiti tramite OSM.

Dalle recensioni Google, invece, sono stati ricavati, tramite web scraping, i dati relativi a 2368 turisti e alle loro recensioni inerenti a hotel, chiese, attrazioni turistiche e musei visitati.

La procedura di web scraping è stata svolta in ambiente Python, tramite le librerie:

- "***Requests***", una libreria standard Python che mette a disposizione quasi tutte le funzionalità HTTP;

- ” ***Selenium***”, una libreria esterna che fornisce API, tramite quale si può accedere a tutte le funzionalità dei WebDrivers forniti da Google, Firefox…;

- ” ***Beautiful Soup***”, una libreria esterna utile ad estrarre dati da file HTML e XML.

PRE-PROCESSING DEI DATI RACCOLTI

La fase di pre-processing è consistita nell’aggregare e pulire i dati raccolti.

In particolare, ci si è occupati di eliminare dati poco significativi o ridondanti.

Per dati poco significativi si intendono quelli che, come si vede dall’immagine sottostante, indicano informazioni futili all’analisi proposta, come denominazione in diverse lingue di una città o colonne con solo valori nulli.

full_id	osm_id	osm_type	boundary	official_name:a	name:ur	name:fa	name:ar	ref:nuts:2	name:sl	name:sc	name:pl
r40232	40232	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40233	40233	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40243	40243	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40258	40258	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40259	40259	relation	administra…	مقاطعة ماتيرا	ماتيرا	ماترا	ماتيرا	NULL	NULL	NULL	NULL
r40279	40279	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40289	40289	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40292	40292	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40296	40296	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40297	40297	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40303	40303	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
r40306	40306	relation	administra…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Per dati ridondanti, invece, si intendono quelle colonne che indicano la stessa informazione.

Queste si presentano quando si aggregano dati provenienti da diverse fonti.

Ad esempio, i dati raccolti sugli hotel della Basilicata sono stati ottenuti tramite OSM e Open Data Basilicata, ed in entrambe le fonti si riportavano dati come la città, la provincia, la classe e il nome.

Un altro lavoro svolto è stato quello di sostituire, laddove si presentasse, in tutte le colonne di ogni tabella, il valore “null” con ‘ ’ per le colonne di tipo alfanumerico, mentre con il valore “-1” per le colonne di tipo numerico.

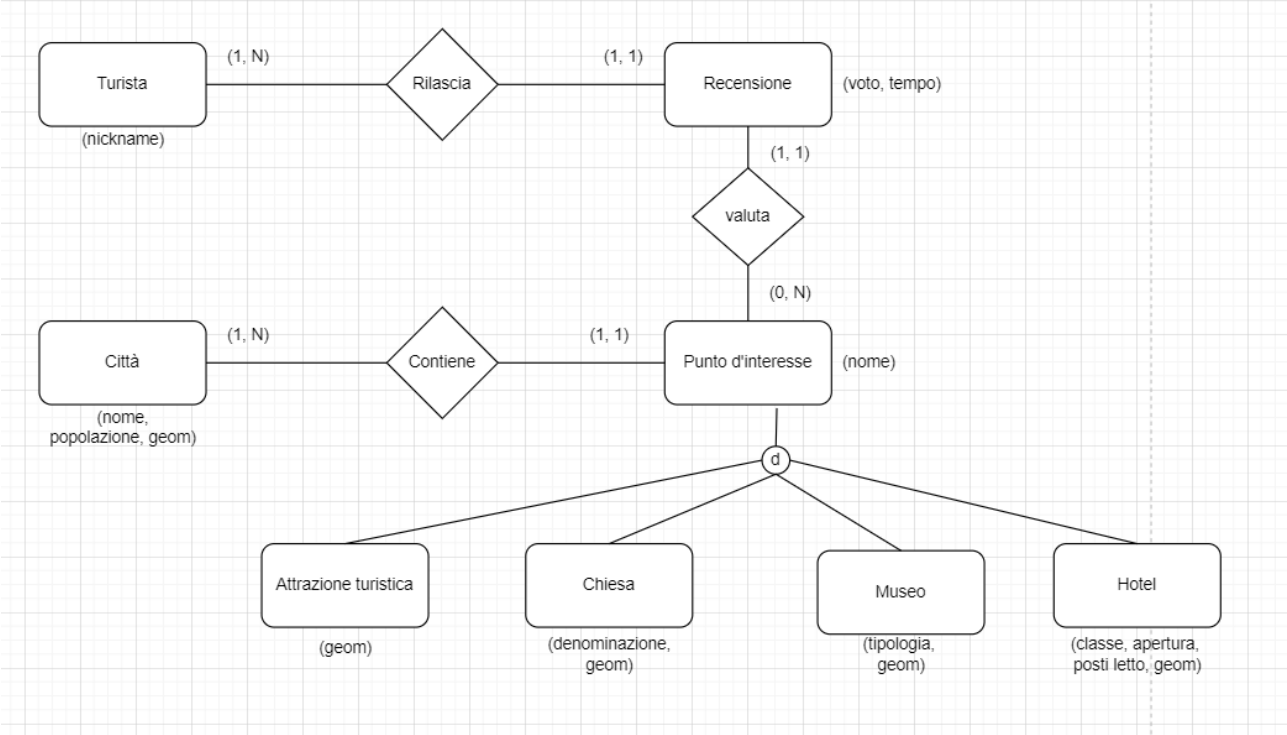
MODELLO CONCETTUALE

La modellazione concettuale consiste nel creare un diagramma **Entità-Relazioni**, ovvero un tipo di diagramma di flusso che illustra come le "entità" entrano in gioco nel sistema e come esse si relazionano tra loro.

Esso rappresenta, a livello concettuale, la struttura relazionale del database dell'applicazione.

Nel progetto proposto, la modellazione concettuale è stata sviluppata solo su una frazione delle tematiche affrontate, ovvero solo sulle entità riguardanti il turismo.

Di seguito troviamo la versione finale del diagramma, frutto di numerosi accorgimenti che hanno costituito le precedenti versioni.



Nel diagramma vediamo le seguenti entità che si rapportano nel seguente modo:

Il turista, che possiede un proprio nickname, rilascia una recensione ad un determinato punto d'interesse, definendo un voto. L'entità recensione tiene conto di quanti anni sono passati dal suo rilascio.

L'entità punto d'interesse, di cui si riporta solo il nome, è una generalizzazione disgiunta di altre quattro entità:

- Attrazione turistica, che possiede un attributo di tipo geometrico;
- Chiesa, che possiede un attributo di tipo geometrico e una denominazione (cattolica, evangelista…);
- Museo che possiede un attributo di tipo geometrico e la sua tipologia (archeologico, naturale…);
- Hotel, che possiede un attributo geometrico, la sua classe (numero di stelle), il periodo d'apertura e il numero totale di posti letto.

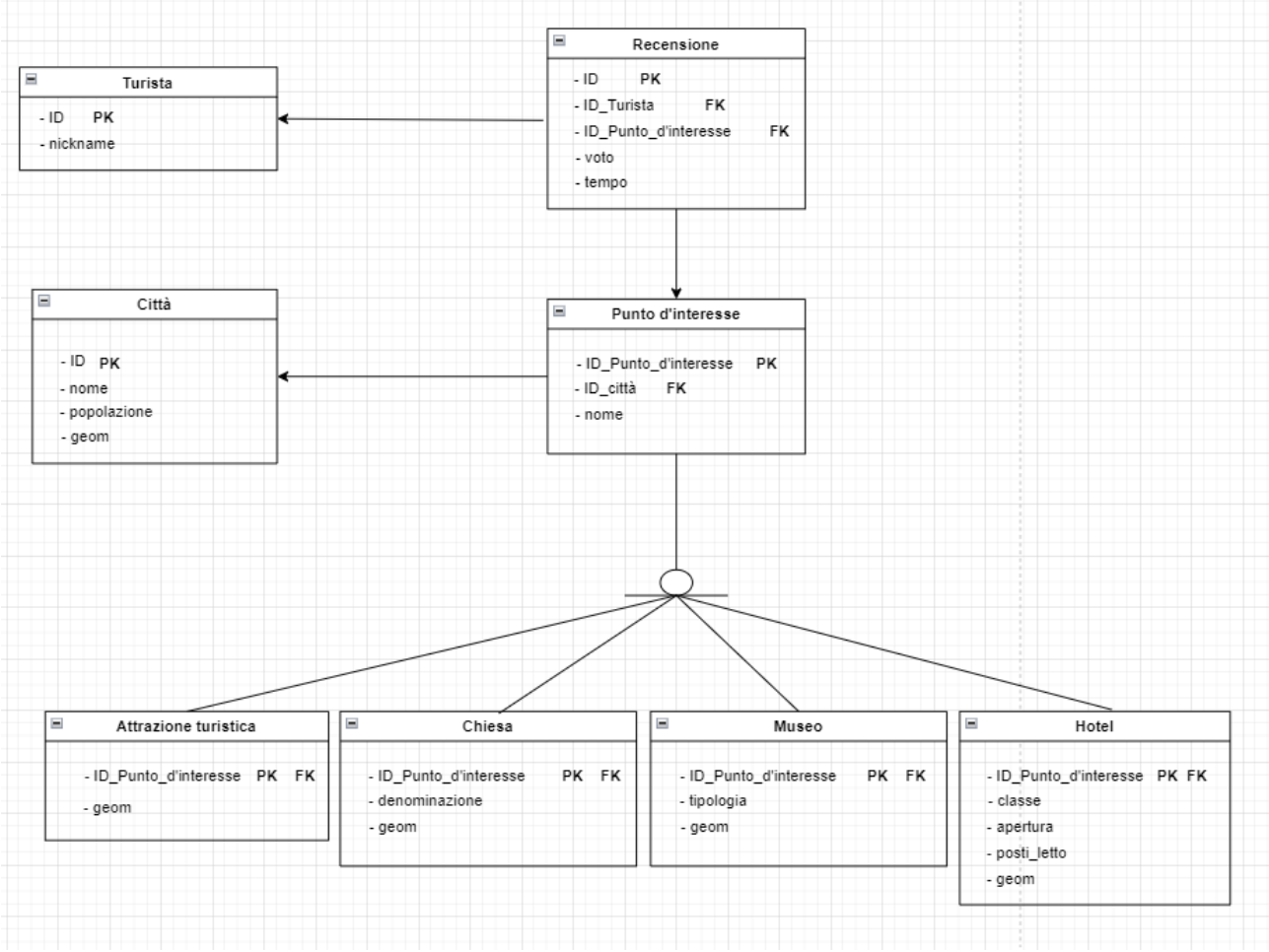
Ogni punto d'interesse è contenuto in una città, di cui si riporta il nome, la popolazione e un attributo geometrico.

MODELLO LOGICO

La modellazione logica consiste nel creare uno **schema relazionale**, che rappresenta un passo intermedio tra il livello concettuale e il livello fisico dei dati.

Basandosi sul modello E-R e applicando l'algoritmo di mapping si ottiene la traduzione da schema concettuale a schema logico con la conseguente costruzione delle tabelle.

Di seguito troviamo lo schema risultante.



Essendo la relazione "rilascia", una relazione del tipo "Un turista può rilasciare da una ad N recensioni e una recensione può essere rilasciata da uno ed un solo turista", nello schema logico la tabella Recensione incorporerà un identificativo del turista.

Il medesimo discorso viene fatto sulla relazione "valuta", infatti Recensione incorporerà un identificativo del punto d'interesse al quale il turista ha assegnato una valutazione.

Per quanto riguarda la generalizzazione, questa viene gestita costruendo cinque tabelle, ognuna per ogni entità, con l'accorgimento che le tabelle figlie della tabella Punto_interesse dovranno avere la stessa chiave primaria della tabella padre.

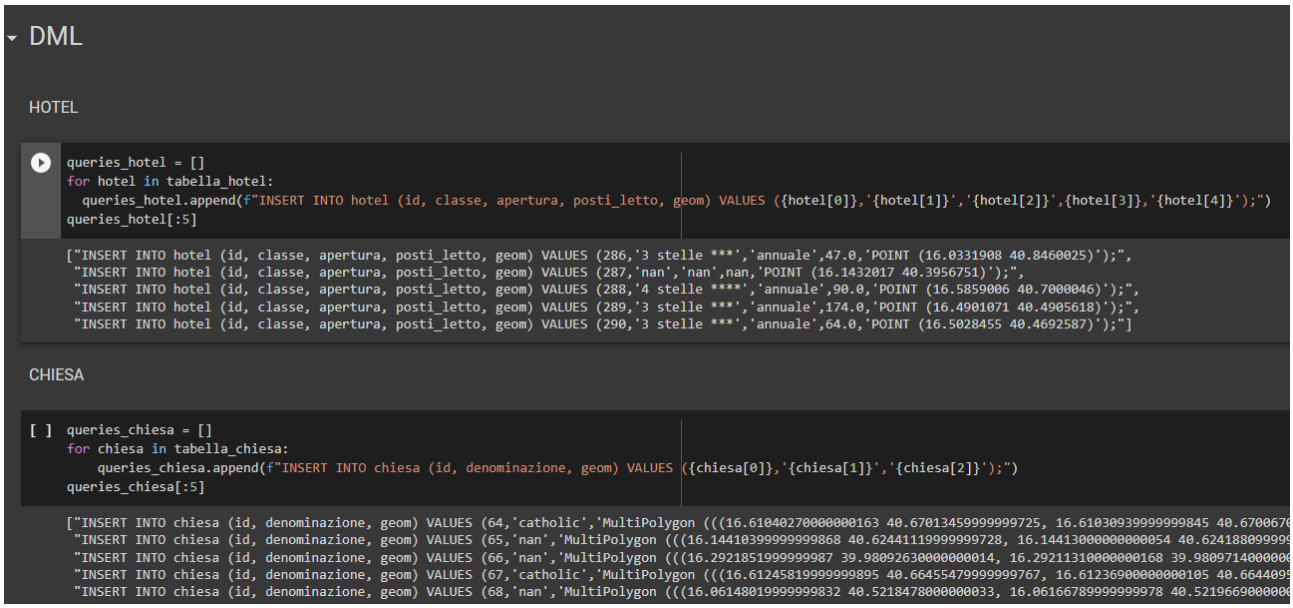
La tabella Punto_interesse avrà un identificativo della città in cui si trova, poiché la relazione "contiene" è del tipo "Un punto d'interesse è contenuto in una ed in una sola città e una città contiene da uno ad N punti d'interesse".

CREAZIONE DELLE TABELLE ED INSERIMENTI

Sviluppati gli schemi precedenti, ottenuti ed elaborati i dati grezzi, si è passati alla creazione delle tabelle fisiche attraverso il ***DDL*** (Data Definition Language).

Concluso il passaggio di creazione, attraverso il ***DML*** (Data Manipulation Language), si è passati alla fase di inserimento dati.

I comandi d'inserimento sono stati creati attraverso script Python che leggono da file CSV pre-elaborati, come mostrato nello screenshot sottostante.



ESEMPI DI QUERIES SUL DATABASE

Ottenuta la struttura fisica del DB, è possibile interrogarlo per estrarre le informazioni di nostro interesse.

Tali interrogazioni vengono svolte utilizzando il linguaggio ***QL*** (Query Language).

Di seguito si troveranno quattro esempi di query incentrate su dati strutturati e quattro esempi di query incentrate su dati non strutturati.

I file contenenti tutte le query sono: "QL_dati_strutturati.sql" e "QL_dati_non_strutturati.sql", contenuti nella cartella del progetto assieme al file "DDL_DML.sql" che invece contiene ciò che è stato descritto nel capitolo precedente.

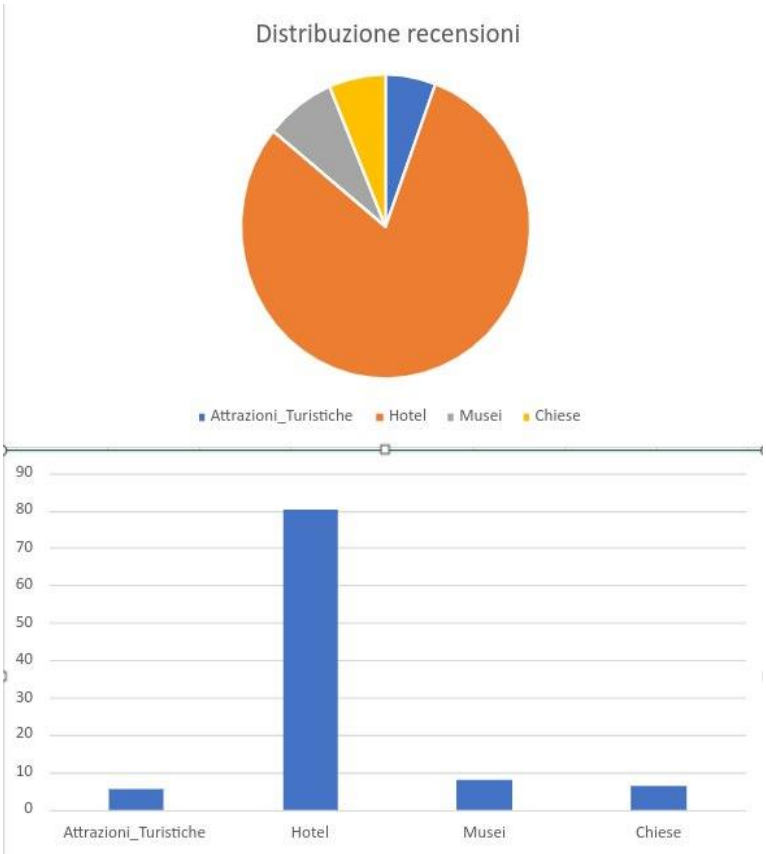
QUERIES DATI STRUTTURATI

Percentuale del numero di recensioni ottenute da ogni categoria di punto d'interesse.

```
SELECT CAST(tab_attr.numero_attrazioni_recensite as float)/CAST(tab_rec.numero_recensioni as float)*100 AS perc_attrazioni_turistiche,
       CAST(tab_hotel.numero_hotel_recensiti as float)/CAST(tab_rec.numero_recensioni as float)*100 AS perc_hotel,
       CAST(tab_mus.numero_musei_recensiti as float)/CAST(tab_rec.numero_recensioni as float)*100 AS perc_musei,
       CAST(tab_chiese.numero_chiese_recensite as float)/CAST(tab_rec.numero_recensioni as float)*100 AS perc_chiese
FROM (SELECT COUNT(*) AS numero_recensioni
        FROM recensione) AS tab_rec,
     (SELECT COUNT(*) AS numero_attrazioni_recensite
        FROM attrazione_turistica att INNER JOIN punto_interesse pi ON pi.id = att.id INNER JOIN recensione r ON r.id_punto_interesse = pi.id) AS tab_attr,
     (SELECT COUNT(*) AS numero_hotel_recensiti
        FROM hotel h INNER JOIN punto_interesse pi ON pi.id = h.id INNER JOIN recensione r ON r.id_punto_interesse = pi.id) AS tab_hotel,
     (SELECT COUNT(*) AS numero_musei_recensiti
        FROM museo mus INNER JOIN punto_interesse pi ON pi.id = mus.id INNER JOIN recensione r ON r.id_punto_interesse = pi.id) AS tab_mus,
     (SELECT COUNT(*) AS numero_chiese_recensite
        FROM chiesa ch INNER JOIN punto_interesse pi ON pi.id = ch.id INNER JOIN recensione r ON r.id_punto_interesse = pi.id) AS tab_chiese
```

Risultato | Explain | Messaggi | Notifiche

	perc_attrazioni_turistiche double precision	perc_hotel double precision	perc_musei double precision	perc_chiese double precision
1	5.616554054054054	80.19425675675676	7.85472972972973	6.33445945945946



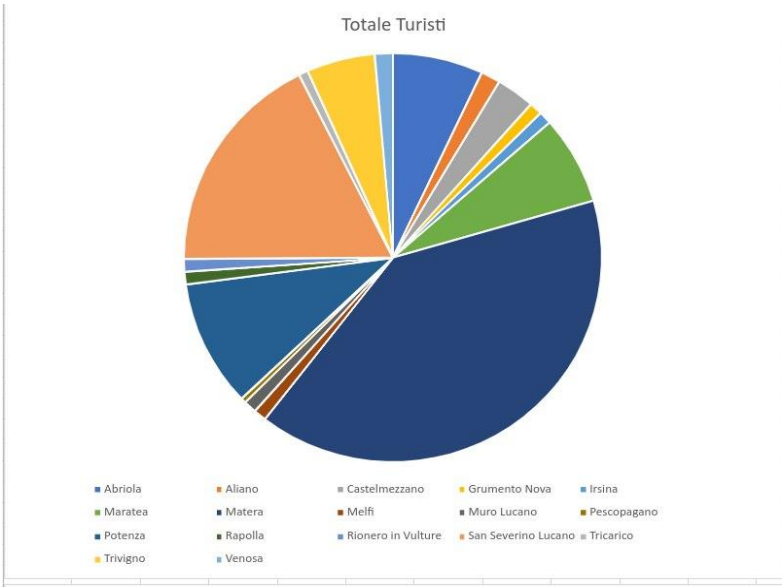
L'interrogazione mostra come si distribuiscono le recensioni che sono state raccolte; il 5.6% delle recensioni è dedicata ad attrazioni turistiche, l'80.2% è dedicata agli hotel, il 7.8% è dedicato ai musei ed il 6.3% è dedicato alle chiese.

Numero di turisti per ogni città, suddivisi in anni.

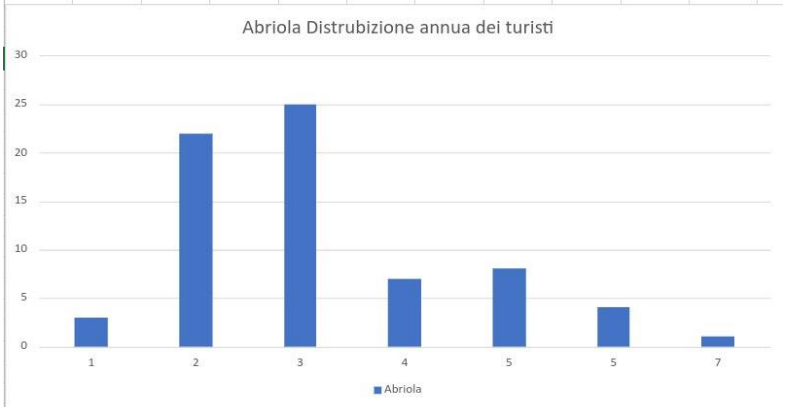
```
select tab3.nomecitta, tab3.totale as totaleturisti,string_agg(tab3.annofa::text,', ') as anni,string_agg(tab3.numeroInAnno::text,', ') as numeroTuristioAllanno from
(select tab2.nomecitta as nomecitta,tab2.totale as totale,tab2.annoFa as annofa ,count(r.tempo) as numeroInAnno
from recensione as r inner join punto_interesse on r.id_punto_interesse=punto_interesse.id inner join citta on citta.id=punto_interesse.id_citta,

(select tab.nomecitta as nomecitta, tab.totale as totale, recensione.tempo as annoFa from recensione,

 (select citta.nome as nomecitta, count(*) as totale
    from turista inner join recensione on recensione.id_turista=turista.id inner join punto_interesse on punto_interesse.id=recensione.id_punto_interesse
    inner join citta on citta.id=punto_interesse.id_citta
    group by citta.nome ) as tab
group by tab.nomecitta,tab.totale,recensione.tempo
order by tab.nomecitta)tab2

where r.tempo=tab2.annofa and citta.nome=tab2.nomecitta
group by tab2.nomecitta,tab2.totale,tab2.annoFa) tab3
group by tab3.nomecitta, tab3.totale
```

Risultato | Explain | Messaggi | Notifiche

	nomecitta character varying	totaleturisti bigint	anni text	numeroturistioallanno text
1	Abriola	70	1, 2, 3, 4, 5, 6, 7	3, 22, 25, 7, 8, 4, 1
2	Aliano	15	1, 2, 3	13, 1, 1
3	Castelmezzano	30	1	30
4	Grumento Nova	10	1, 4	9, 1
5	Irsina	10	1, 2, 3, 4, 5	1, 5, 2, 1, 1
6	Maratea	70	1, 2, 3, 4, 5, 6, 7	2, 1, 4, 19, 20, 15, 9
7	Matera	400	1, 2, 3, 4, 5, 6, 7, 8	124, 99, 76, 61, 29, 8, 2, 1



L'interrogazione mostra il numero totale di turisti ricevuti da ogni città e la distribuzione di tale numero nell'arco degli anni.



Per comprendere meglio il risultato dell'interrogazione prendiamo come esempio il comune di Abriola, che negli ultimi 7 anni ha ospitato 70 dei turisti registrati ed in particolare 3 nell'ultimo anno, 22 due anni fa, 25 tre anni fa, 7 quattro anni fa, 8 cinque anni fa, 4 sei anni fa e 1 sette anni fa.

Punti d'interesse migliori per ogni categoria, in base alla media delle recensioni.

```
SELECT hotel_migliore.nome AS hotel, chiesa_migliore.nome AS chiesa, museo_migliore.nome AS museo, attrazione_migliore.nome AS attrazione_turistica
FROM        (SELECT pi.nome AS nome, city.id AS id_comune, AVG(r.voto) AS media_voti
            FROM hotel h INNER JOIN punto_interesse pi ON pi.id = h.id
                        INNER JOIN recensione r ON pi.id = r.id_punto_interesse
                        INNER JOIN citta city ON city.id = pi.id_citta
            GROUP BY(pi.nome,id_comune)
            ORDER BY media_voti DESC
            LIMIT 1) AS hotel_migliore,

            (SELECT pi.nome AS nome, city.id AS id_comune, AVG(r.voto) AS media_voti
            FROM museo mus INNER JOIN punto_interesse pi ON pi.id = mus.id
                        INNER JOIN recensione r ON pi.id = r.id_punto_interesse
                        INNER JOIN citta city ON city.id = pi.id_citta
            GROUP BY (pi.nome, id_comune)
            ORDER BY media_voti DESC
            LIMIT 1) AS museo_migliore,


            (SELECT pi.nome AS nome, city.id AS id_comune, AVG(r.voto) AS media_voti
            FROM chiesa ch INNER JOIN punto_interesse pi ON pi.id = ch.id
                        INNER JOIN recensione r ON pi.id = r.id_punto_interesse
                        INNER JOIN citta city ON city.id = pi.id_citta
            GROUP BY (pi.nome, id_comune)
            ORDER BY media_voti DESC
            LIMIT 1) AS chiesa_migliore,


            (SELECT pi.nome AS nome, city.id AS id_comune, AVG(r.voto) AS media_voti
            FROM attrazione_turistica attr INNER JOIN punto_interesse pi ON pi.id = attr.id
                        INNER JOIN recensione r ON pi.id = r.id_punto_interesse
                        INNER JOIN citta city ON city.id = pi.id_citta
            GROUP BY (pi.nome, id_comune)
            ORDER BY media_voti DESC
            LIMIT 1) AS attrazione_migliore
```

Risultato | Explain | Messaggi | Notifiche

	hotel character varying	chiesa character varying	museo character varying	attrazione_turistica character varying
1	Sextantio - Le Grotte della Civita	Cappella di San Cataldo	Palafrido - Museo Laboratorio della fauna minore	San Nicola all Ofra

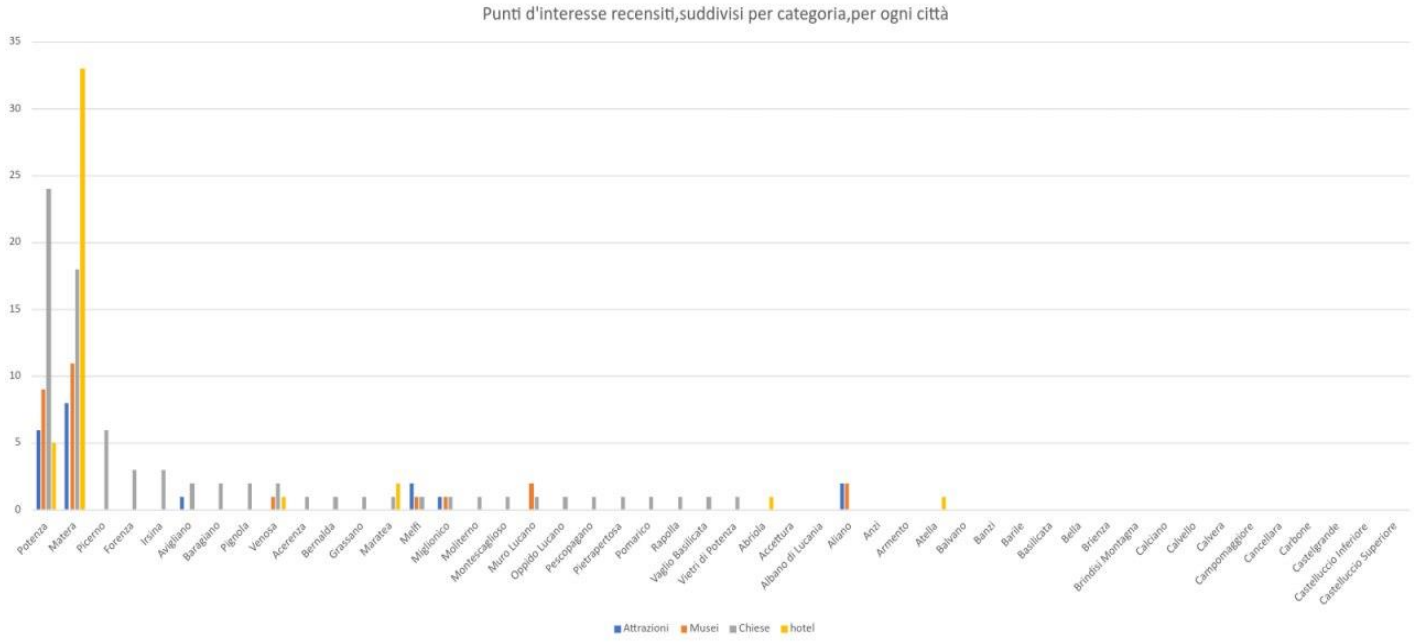
L'interrogazione mostra quale punto d'interesse ha una media aritmetica dei voti migliore rispetto a tutti gli altri che appartengono alla sua stessa categoria; L'hotel con la miglior media, la chiesa miglior media etc.

Numero totale, suddiviso per categoria, dei punti d'interesse recensiti per ogni città.

```
select citta.nome, count(attrazione_turistica.id) as connteggioattrazione,tab3.conteggiomuseo,tab3.conteggiochiesa,tab3.conteggiohotel
from attrazione_turistica inner join punto_interesse on attrazione_turistica.id=punto_interesse.id right join citta on citta.id=punto_interesse.id_citta,
    (select citta.nome cittamuseo, count(museo.id) as conteggiomuseo, tab2.cittachiesa,tab2.conteggiochiesa,tab2.conteggiohotel,tab2.cittahotel
    from museo inner join punto_interesse on museo.id=punto_interesse.id right join citta on citta.id=punto_interesse.id_citta,

        (select citta.nome as cittachiesa,count(chiesa.id) conteggiochiesa,tab1.conteggiohotel,tab1.cittahotel
         from chiesa inner join punto_interesse on chiesa.id=punto_interesse.id right join citta on citta.id=punto_interesse.id_citta,

            (select count(hotel.id) conteggiohotel, citta.nome as cittahotel
             from hotel inner join punto_interesse on hotel.id=punto_interesse.id right join citta on citta.id=punto_interesse.id_citta
               group by citta.nome
               ) tab1

        where citta.nome=tab1.cittahotel
        group by cittachiesa,tab1.conteggiohotel,tab1.cittahotel) tab2

    where citta.nome=tab2.cittachiesa
    group by citta.nome,tab2.cittachiesa,tab2.conteggiochiesa,tab2.conteggiohotel,tab2.cittahotel) tab3

where citta.nome=tab3.cittamuseo
group by citta.nome,tab3.conteggiomuseo,tab3.conteggiochiesa,tab3.conteggiohotel
order by tab3.conteggiochiesa desc
```

Risultato | Explain | Messaggi | Notifiche

	nome character varying	connteggioattrazione bigint	conteggiomuseo bigint	conteggiochiesa bigint	conteggiohotel bigint
1	Potenza	6	9	24	5
2	Matera	8	11	18	33
3	Picerno	0	0	6	0
4	Forenza	0	0	3	0
5	Irsina	0	0	3	0
6	Avigliano	1	0	2	0



L'interrogazione mostra la distribuzione della categoria di punti d'interesse recensiti per ogni città.

Prendendo Potenza come esempio, questa ha 6 attrazioni turistiche recensite, 9 musei recensiti, 24 chiese recensite e 5 hotel recensiti.

QUERIES DATI NON STRUTTURATI

Nome dell'hotel più a ovest in Basilicata, comune d'appartenenza, chiesa più vicina, museo più vicino e attrazione turistica più vicina.

```
select hotel_ovest.nome_hotel ,area.nome as comune,  chiesa_vicina.nome as nomechiesa, museo_vicino.nome as nomemuseo, attrazione_vicina.nome as nomeattrazione
from (select st_x(hotel.geom), hotel.geom as pos_hotel, punto_interesse.nome as nome_hotel
   from hotel inner join punto_interesse on punto_interesse.id=hotel.id
   order by st_x(hotel.geom) limit 1) as hotel_ovest,

   (select punto_interesse.nome as nome
    from chiesa inner join punto_interesse on punto_interesse.id=chiesa.id,
    (select st_x(hotel.geom), hotel.geom as pos_hotel, punto_interesse.nome as nome_hotel
        from hotel inner join punto_interesse on punto_interesse.id=hotel.id
      order by st_x(hotel.geom) limit 1) as hotel_ovest
    where st_dwithin(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText(chiesa.geom)), 20000)
    order by st_distance(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText(chiesa.geom))) limit 1) as chiesa_vicina,


   (select punto_interesse.nome as nome
    from museo inner join punto_interesse on punto_interesse.id=museo.id,(
        select st_x(hotel.geom), hotel.geom as pos_hotel, punto_interesse.nome as nome_hotel
        from hotel inner join punto_interesse on punto_interesse.id=hotel.id
      order by st_x(hotel.geom) limit 1) as hotel_ovest
    where st_dwithin(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText( museo.geom)), 20000)
   order by st_distance(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText( museo.geom))) limit 1) as museo_vicino,

   (select punto_interesse.nome as nome
    from attrazione_turistica inner join punto_interesse on punto_interesse.id=attrazione_turistica.id,
    (select st_x(hotel.geom), hotel.geom as pos_hotel, punto_interesse.nome as nome_hotel
         from hotel inner join punto_interesse on punto_interesse.id=hotel.id
         order by st_x(hotel.geom) limit 1) as hotel_ovest
    where st_dwithin(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText( attrazione_turistica.geom)), 20000)
    order by st_distance(ST_GeographyFromText(st_AsText(hotel_ovest.pos_hotel)), ST_GeographyFromText(st_AsText( attrazione_turistica.geom))) limit 1) as attrazione_vicina,

   area
where st_contains( area.geom, hotel_ovest.pos_hotel) and
area.wikipedia!='it:Provincia di Potenza' and area.wikipedia!='it:Provincia di Matera' and area.wikipedia!='it:Basilicata'
```

Risultato Explain Messaggi Notifiche

	nome_hotel character varying	comune character varying	nomechiesa character varying	nomemuseo character varying	nomeattrazione character varying
1	Hotel delle Colline	Muro Lucano	Santa Maria delle Grazie	Museo Archeologico Nazionale di Muro Lucano	Cascate Vallone del Tuorno

Nome del comune d'appartenenza di una fattoria, nome del comune il quale centro urbano è il più vicino alla fattoria e la distanza in metri tra questo centro urbano e la fattoria.

```
select area.nome as nometerritorioapparteneza, tab6.nomefattoria, tab6.distanzaminima, tab6.nomecittavicino from area,
(select citta.nome as nomecittavicino , tab5.pos as posizione,tab5.nome as nomefattoria, tab2.minimo  as distanzaminima
from citta,(select fattoria.nome as nome, st_pointonsurface(fattoria.geom) as pos from fattoria) as tab5 ,
    (
    select tab2.nomefattoria, min(tab2.distanza) as minimo from
            (
            select citta.nome as nomecitta, tab.nome as nomefattoria, ST_Distance(ST_GeographyFromText(st_AsText(citta.geom)),ST_GeographyFromText(st_AsText(tab.pos)))
                as distanza
            from citta,
                (select fattoria.nome as nome, st_pointonsurface(fattoria.geom) as pos from fattoria) as tab
            )tab2
    group by tab2.nomefattoria
    )tab2

where ST_Distance(ST_GeographyFromText(st_AsText(citta.geom)),ST_GeographyFromText(st_AsText(tab5.pos))) = tab2.minimo)tab6
where st_contains(area.geom,tab6.posizione) and area.wikipedia!='it:Provincia di Potenza' and area.wikipedia!='it:Provincia di Matera' and area.wikipedia!='it:Basilicata'

```

Risultato Explain Messaggi Notifiche

	nometerritorioapparteneza character varying	nomefattoria character varying	distanzaminima double precision	nomecittavicino character varying
1	Stigliano	Mania di Vasti	5824.18150368	Craco
2	Stigliano	Masseria Mania Del Monte	7104.60643532	Craco
3	Stigliano	Masseria Caputo	9879.75520987	Craco
4	Stigliano	Gannano di Sopra	6638.83241717	Craco
5	Stigliano	Masseria Tempa Rossa	9132.19603928	Craco
6	Stigliano	Masseria Ursone	8824.92532905	Craco
7	Stigliano	Masseria Torre	8675.91244937	Craco
8	Stigliano	Mulino ad acqua Gannano	6929.03395998	Craco
9	Garaguso	Masseria La Maina	838.30303325	Garaguso
10	Garaguso	Masseria Don Paolo	1022.95375414	Garaguso
11	Garaguso	Masseria Spagna	941.37649492	Garaguso
12	Garaguso	Masseria Bosco del Duca	2811.4346033	Garaguso
13	Garaguso	Masseria Marra	2714.42479544	Garaguso
14	Garaguso	Cascina Dama	1323.08673986	Garaguso
15	Garaguso	Cascina De Luca	1168.93408138	Garaguso
16	Garaguso	Masseria del Carcerato	747.33225074	Garaguso
17	Garaguso	Masseria Boscone	2982.11859701	Garaguso
18	Garaguso	Cascina Barbarito	1546.88649333	Garaguso
19	Calciano	Masseria Molessa	1823.89091568	Garaguso

Per comprendere meglio il fine di questa interrogazione prendiamo come esempio la riga 1 del risultato della query, questa ci mostra che la fattoria “Mania di Vasti”, pur appartenendo all’area amministrativa di Stigliano ha come centro urbano più vicino, quello del comune di Craco.

Nome del comune che contiene, all’interno della propria area amministrativa, un hotel che nel raggio di un kilometro ha una stazione e la distanza tra questi ultimi in metri.

```
1  select area.nome, tab.hotelnome, tab.stazionenome, tab.distanza from area,
2  (select hotel.geom as hotelpos, punto_interesse.nome as hotelnome, stazione.nome as stazionenome,
3   ST_Distance(ST_GeographyFromText(st_AsText(hotel.geom)),ST_GeographyFromText(st_AsText(stazione.geom))) as distanza
4   from hotel inner join punto_interesse on punto_interesse.id=hotel.id,stazione where ST_DWithin(hotel.geom,stazione.geom,0.01)
5  ) as tab
6   where ST_Contains(area.geom,tab.hotelpos) and  area.wikipedia!='it:Provincia di Potenza' and area.wikipedia!='it:Provincia di Matera' and area.wikipedia!='it:Basilicata'
7  group by area.nome, tab.hotelnome, tab.stazionenome,tab.distanza
```

Risultato | Explain | Messaggi | Notifiche

	nome character varying	hotelnome character varying	stazionenome character varying	distanza double precision
1	Bernalda	Alessidamo Club Metaponto		1033.42683704
2	Ferrandina	Hotel degli Ulivi		453.93047483
3	Ferrandina	Hotel Diamante		161.80442307
4	Garaguso	Hotel 407	Grassano	361.15894671

Percentuale dell'occupazione territoriale di ogni comune e la sua densità di popolazione.

```sql
select area.nome as comune, estensione_territoriale.perc_ext, densita_popolazione.dens as densita

from area, (select c.id as id_citta,(st_area(c.geom)/st_area(d.geom))*100 as perc_ext
   from area c, area d
   where d.nome = 'Basilicata') as estensione_territoriale,
   (select c.id as id_citta,(pt.popolazione/st_area(c.geom))*100 as dens
   from area c, citta pt
   where st_contains(c.geom, pt.geom) and c.nome = pt.nome) as densita_popolazione

where
area.id = estensione_territoriale.id_citta
and area.id = densita_popolazione.id_citta
and area.wikipedia!='it:Provincia di Potenza'
and area.wikipedia!='it:Provincia di Matera'
and area.wikipedia!='it:Basilicata'
```

Risultato | Explain | Messaggi | Notifiche

	comune character varying	perc_ext double precision	densita double precision
1	Rotonda	0.41691793076131195	87801916.79459913
2	Terranova di Pollino	1.1224539349395664	12867232.467169458
3	Trecchina	0.37225281387668846	60802969.297800586
4	Chiaromonte	0.7005674606258766	28867729.671027232
5	Cirigliano	0.14563623840175888	28768609.567583665
6	Viggianello	1.1905733012512334	25756683.049131684
7	Fardella	0.28037207326835123	25689461.246558864
8	Maratea	0.6879896812045523	71997048.27957621
9	San Severino Lucano	0.6039388423418951	29978827.03765286
10	San Paolo Albanese	0.2927716435973567	13378044.437839732
11	Castelluccio Inferiore	0.28162618140456896	78363332.2374008
12	Cersosimo	0.24557188197093838	32473798.101874005
13	Lauria	1.7380740210080439	69982418.07743612
14	Castelluccio Superiore	0.3309072642746847	28082719.747301318
15	Francavilla in Sinni	0.45290980369015066	90782008.03498726
16	San Costantino Albanese	0.42544782967369654	19562946.484005306
17	Latronico	0.7512820107323468	66157180.261759
18	Rivello	0.692490163390713	40924294.49651187
19	Nemoli	0.18802254469050655	78166612.49639982
20	Episcopia	0.2888656509426767	52964609.76705236

FUNZIONI

Al fine di interrogare il database in maniera più chiara è semplice, può essere utile implementare alcune funzioni in linguaggio ***plpgsql.***

Di seguito sono mostrate le funzioni implementate per questo progetto con le relative descrizioni.

La funzione "*conta_turisti*", che riceve come parametri il nome del punto d'interesse, la categoria (hotel, chiesa, museo, attrazione turistica) e un intero che indica il periodo da analizzare (ad esempio un anno fa, due anni fa etc.), esegue un conteggio sul numero di recensione inserite nel periodo indicato.

```
create or replace function conta_turisti(nome_struttura text,tipo_struttura text, anno integer)
returns int
as
$$
declare
num int;
begin
    if tipo_struttura='hotel' then

        select count(*) into num from recensione inner join punto_interesse on punto_interesse.id=recensione.id_punto_interesse inner join hotel
        on hotel.id=punto_interesse.id
        where punto_interesse.nome=nome_struttura and recensione.tempo=anno;
        return num;

    elsif tipo_struttura='museo' then

        select count(*) into num from recensione inner join punto_interesse on punto_interesse.id=recensione.id_punto_interesse inner join museo
        on museo.id=punto_interesse.id
        where punto_interesse.nome=nome_struttura and recensione.tempo=anno;
        return num;

    elsif tipo_struttura='chiesa' then
        select count(*) into num from recensione inner join punto_interesse on punto_interesse.id=recensione.id_punto_interesse inner join chiesa
        on chiesa.id=punto_interesse.id
        where punto_interesse.nome=nome_struttura and recensione.tempo=anno;
        return num;

    elsif tipo_struttura='attrazione_turistica' then
        select count(*) into num from recensione inner join punto_interesse on punto_interesse.id=recensione.id_punto_interesse inner join attrazione_turistica
        on attrazione_turistica.id=punto_interesse.id
        where punto_interesse.nome=nome_struttura and recensione.tempo=anno;
        return num;

    end if;

end;
$$ language plpgsql;
```

Esempi di risultati ottenuti inserendo come input prima un hotel e successivamente un'attrazione turistica:

```
1  select conta_turisti('Hotel Paradiso','hotel',1);
```

Risultato | Explain | Messaggi | Notifiche

	conta_turisti integer
1	9

```
select conta_turisti('Cristo La Gravinella','attrazione_turistica',2);
```

ltato | Explain | Messaggi | Notifiche

conta_turisti integer
5

La funzione “*priorita_manutenzione*”, esegue un conteggio sulle ferrovie e sui fiumi che ogni strada incrocia, determinando un indice di priorità di manutenzione, sebbene sarebbe più consono considerare la data dell’ultima manutenzione effettuata, questa non era presente nel set di dati.

```
create or replace function priorita_manutenzione()
returns table(nome character varying,conteggio bigint)
as
$func$
declare
nomi character varying[];
begin
    return query
    SELECT tab.nome_strada, (tab.numero_fiumi_incrociati + tab.numero_ferrovie_incrociate) AS total
    FROM (SELECT s.id AS id_strada, s.nome AS nome_strada,
 (SELECT COUNT(*) FROM fiume WHERE st_intersects(fiume.geom, s.geom)) AS numero_fiumi_incrociati,
 (SELECT COUNT(*) FROM ferrovia WHERE st_intersects(ferrovia.geom, s.geom)) AS numero_ferrovie_incrociate
 FROM strada s) AS tab
 where length(tab.nome_strada)>1
 order by total desc;



end;
$func$ language plpgsql;
```

L’esempio d’utilizzo della funzione mostra in output i nomi delle strade ordinati in base alla priorità in ordine decrescente e il numero totale di fiumi e ferrovie incrociate per ogni strada.

```
select priorita_manutenzione();
```

Risultato | Explain | Messaggi | Notifiche

	priorita_manutenzione record
1	("Ponte Musmeci",8)
2	("Ponte Musmeci",8)
3	("SP48 Basso Melfese",2)
4	(Appulo-Lucana,2)
5	("ex ss 169 di Genzano",1)
6	("Via del Palazzo",1)

SVILUPPI FUTURI

- ***Sviluppo di Bot Telegram in grado di fornire ai turisti informazioni utili come il miglior hotel in base alle recensioni, percorsi turistici vicino l'hotel scelto, elenco dei punti d'interesse più importanti da visitare basandosi sul numero di recensioni positive rilasciate… e che sia in grado di fornire ai gestori dei punti d'interesse informazioni utili diverse da quelle che si forniscono ai turisti come voti ottenuti dalle recensioni, possibilità di ottenere un confronto paritario con altri punti d'interesse della stessa categoria, possibilità di comprendere lo status di popolarità e appetibilità della zona in cui si è avviato o in cui si gestisce l'esercizio;***

- ***Sviluppo di un'app data centric capace di eseguire ciò che si è proposto in merito al Bot Telegram, ma con un'interfaccia più intuitiva e user friendly;***

- ***Ampliamento del set di dati strutturati, come ad esempio dati Istat o dati inerenti ad altre entità che non sono state considerate nella parte di progetto inerente e Dati Strutturai come ad esempio industrie e fattorie;***

- ***Possibilità di replicare l'idea di questo progetto in altre regioni;***

- ***Aggiornamento dei dati acquisiti tramite web scraping in tempo reale (già in fase di sviluppo)***.

CONCLUSIONI

In conclusione, nel seguente diagramma viene mostrato il workflow completo del lavoro proposto.