Università degli Studi di Salerno

Dipartimento di Informatica
Corso di Laurea Magistrale in Informatica

Progetto di Basi Dati 2

Gestione degli AirB&B di Milano, tramite MongoDB

Costante Marco 0522501330

Anno Accademico 2021-2022

Capitolo 1

Introduzione

Lo scopo di questo progetto è stato la realizzazione di un sistema web-based che permettesse di interagire con un database di tipo NoSQL. In particolare, in questo lavoro progettuale è stato utilizzato un database orientato ai documenti, utilizzando il tool **MongoDB**. In questo lavoro è stato realizzato un sistema che permette

l'interazione e la gestione degli AirB&B di Milano. I dati utilizzati sono stati ottenuti dalla piattaforma **Kaggle**, che offre una serie di dataset open-source con dati strutturati e non.
Ai fini di questo lavoro si è resa necessaria un'approfondita fase di pre-elaborazione dei dati originali, al fine di disporre delle sole informazioni relative ai fini del progetto, che fossero leggibili e consistenti.
Successivamente è stato possibile effettuare il popolamento della base di dati, utilizzando file in formato **JSON**, e in fine è stata realizzata l'interfaccia che permette l'esecuzione di classiche operazioni CRUD.

In questo documento verranno descritte tutte le fasi del lavoro, analizzando nel dettaglio i problemi affrontanti, le soluzioni ideate e le tecnologie utilizzate.

Capitolo 2

Fase di pre-processing

Come già anticipato, i dataset di riferimento sono stati ottenuti attraverso la piattaforma Kaggle, in particolare i dati fanno riferimento agli AirB&B di Milano nell'anno 2020.

La fase di pre-processing è stata realizzata in linguaggio **Python**, utilizzando la libreria **Pandas**, la quale si è resa un ottimo strumento per la gestione e preparazione dei dati.

2.1 Data cleaning

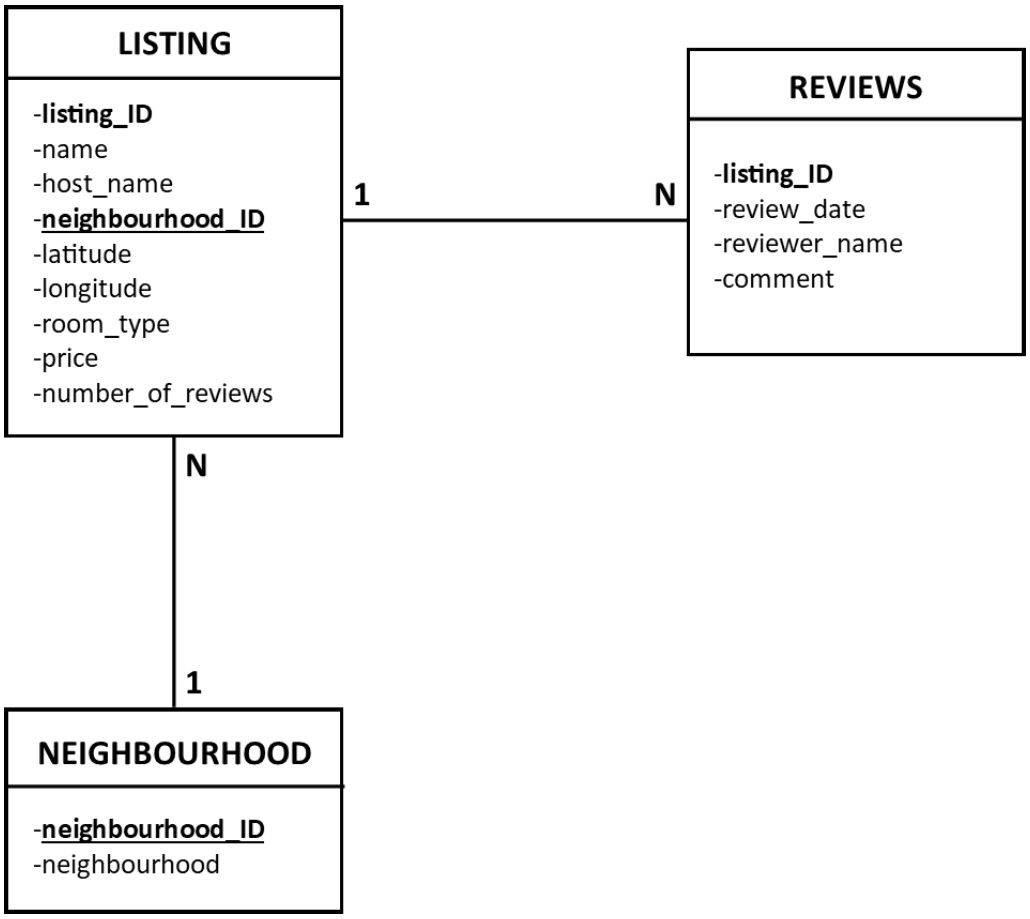
Una prima fase affrontata in questo lavoro è stata la pulizia di dati.
Il dataset conteneva in origine tre file CSV:

- **Listing:** contenente tutte le informazioni riguardanti 18830 B&B. Inizialmente il file conteneva 16 colonne, delle quali sono state mantenute solo quelle rilevanti:
  - listing id;
  - name;
  - host name;
  - neighbourhood;
  - latitude;
  - longitude;
  - room type;
  - price;

  - number of reviews;

- **Reviews:** contenente tutte le recensioni di ogni B&B. In particolare, per poter associare ad ogni recensione il B&B di riferimento, veniva utilizzato, come chiave esterna, il listing id. I campi mantenuti sono stati:
  - review date;
  - reviewer name;
  - comments;
- **Neighbourhood:** contenente la lista di tutti gli 88 quartieri, compreso un id rappresentante la chiave esterna nella tabella Listing (successivamente rimosso).

Inizialmente le tabelle si presentavano con una classica struttura relazionale. Sfruttando i vantaggi offerti dall'uso di un database NoSQL, è nata l'idea di fondere le tre tabelle in un'unica **collection**, costituita da una serie di **document** contenenti il B&B, il quartiere in cui si trova e la lista delle recensioni ad esso associato.



Per poter procedere alla creazione dei file JSON che avrebbero popolato il database, si è resa necessaria una fase di analisi dei valori nulli presenti all'interno delle tabelle.

La tabella contenente i quartieri risultava avere una colonna costituita interamente da valori nulli, la quale è stata rimossa. Diverse recensioni presentavano valori nulli in corrispondenza del campo *comments*, chiaramente disporre di recensioni di cui si conosce solo la data e il nome non sarebbe stato utile, pertanto sono state rimosse tutte le tuple di questo tipo.

2.2 Data preparation e population

Al termine della fase di cleaning, avendo a disposizione i tre file csv di base, è stato possibile fonderli per creare dei file formato **JSON** per la creazione della base di dati.

Sono stati creati due file json per il popolamento:

- **MilanoTotalListing.json**: contenente una lista di oggetti ognuno rappresentante un diverso B&B;
- **Location.json**: creato al fine di costruire oggetti **GeoJSON**, dato che avendo a disposizione la *latitudine* e la *longitudine* di ogni B&B, è stata introdotta la possibilità di eseguire **query geospaziali**;

I file creati sono stati quindi importati in MongoDB, creando un database **Milano** con le due collection **listing** e **location**.

Il database è stato popolato importando i file JSON precedentemente costruiti utilizzando la shell MongoDB, attraverso il comando *mongoimport*.

Per poter velocizzare le operazioni di ricerca sono stati costruiti una serie di indici sugli attributi principali della collection LISTING.

Esempio di oggetto **Listing**:


```
{
"listing_id": 6400,
"name": "The Studio Milan",
"host_name": "Francesca",
"neighbourhood": "TIBALDI",
"latitude": 45.44195,
"longitude": 9.17797,
"room_type": "Private room",
"price": 100,
"number_of_reviews": 12,
"reviews": [
{
        "review_date": "2010-04-19",
        "reviewer_name": "Hyun",
        "comments": "I had such a great
                                stay at 'the studio."
},
{
        "review_date": "2011-04-16",
        "reviewer_name": "Tej",
        "comments": "Staying at Francesca's and
                        Alberto's place was a pleasure."
}]}
```

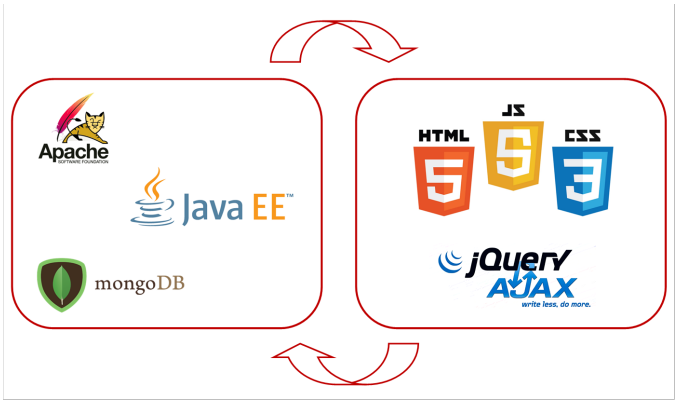

Esempio di oggetto **Location**


```
{
"id": 6400,
"location": {
        "coordinates": [
        9.17797,
        45.44195
        ],
        "type": "Point"
}}
```

Capitolo 3

WebApp

La web application è stata creata attraverso le classiche tecnologie software per il web, come mostrato in figura.



I requisiti funzionali definiti sono stati:

- Inserimento di un nuovo B&B;
- Rimozione di B&B;
- Modifica dei dati dei B&B;
- Visualizzazione delle recensioni di ogni B&B;
- Ricerca per quartiere;
- Ricerca per numero di recensioni;
- Ricerca per prezzo;
- Ricerca per tipo di camera;
- Visualizzazione dei 15 B&B più vicini ad un luogo d'interesse;