UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN DATA SCIENCE A.A. 2023/2024

ISTRUZIONE DI PROGETTO

CASO DI STUDIO

CORSO DI GESTIONE DI DATI STRUTTURATI E NON STRUTTURATI

DOCENTI

Prof. Mario Alessandro Bochicchio

Prof. Corrado Loglisci

STUDENTI

Detomaso Giacomo

(e-mail: g.detomaso7@studenti.uniba.it)

Detomaso Gabriele

(e-mail: g.detomaso6@studenti.uniba.it)

Sommario

Indicazioni generali sul progetto

Premessa

Questo documento riporta alcune informazioni di tipo descrittivo sul progetto di "Gestione dati strutturati e non strutturati".

Il tema di progetto, fornito da uno stakeholder operante nel campo B&B, è il medesimo per le due parti di esame. La nostra idea è stata quella di fare due analisi che non fossero indipendenti, ma che, ad un certo punto, potessero "incontrarsi".

Ad esempio, un caso emblematico è quello in cui, in fase di **analisi non strutturata**, è emerso che i neighborhoods che lo stakeholder dovrebbe considerare al fine di poter aprire la propria attività a NYC, sono ubicati nei boroughs di Manhattan, Queens e Brooklyn, i quali, in fase di **analisi strutturata**, erano emersi come i distretti chiave sui quali concentrarsi nella ricerca di un neighborhood ideale (in base ai requisiti forniti per le analisi).

Le due relazioni, inoltre, terminano **con la medesima query conclusiva** la quale, nonostante sia "strutturata", sfrutta particolari risultati ottenute nelle due analisi.

Struttura del progetto

Il progetto è suddiviso in numerose cartelle:

- **DDL, DML, QL** contengono i codici relativi alle operazioni descritte dal nome della cartella
    - In QL in particolare i file structured.sql contengono queries solo strutturate, il non-structured.sql contengono queries solo non strutturate;
- **Datasets**: contiene i file ottenuti dalle fonti date (suddivisi in CSV e shapefiles, questi ultimi visti la grande dimensione sono zippati);
- **Schema**: contiene gli schemi prodotti in formato SVG;
- **Results sets**: contiene i risultati (CSV) per le varie lanciate
- **Result presentation**: contiene le presentazioni grafiche dei risultati

Per queste ultime due cartelle vi è una divisione tra strutturato e non.

Caricamento automatico del progetto

Il notebook project_loader.ipynb, una volta lanciato permette di caricare il progetto su Postgres in maniera automatizzata. E' necessario fornire semplicemente i propri parametri di connessione nei blocchi di codice della sezione iniziale.

Il notebook esegue automaticamente:

- Caricamento shapefiles in tabelle d'appoggio;
- Caricamento CSV in tabelle di appoggio;
- Esecuzione di script DDL e DML nell'ordine prestabilito.

Caricamento manuale del progetto

Di seguito sono forniti gli step da seguire per eseguire correttamente il progetto su Postgres:

- Caricare con la GUI di utilty di Postgres gli shapefiles;
- Lanciare lo script: DDL/ddl_shapefiles.sql;
- Lanciare lo script: DDL/ddl/csv_temporary_tables;
- Per poter popolare le tabelle CSV d'appoggio è necessario per forza usare il processo automatizzato fornito (o in alternativa usare comandi da terminale di postgres o utility pg admin). In particolare, è necessario eseguire tutti i blocchi di codice del notebook presenti nelle seguenti sezioni:
    - Sezione di introduzione per il setting della connessione;
    - Sezione dal titolo: "**Loading CSV into postgres**"
- Lanciare lo script: DDL/ddl_csv.sql;
- Lanciare gli script DML degli shapefiles (l'ordine non è importante);
- Lanciare gli script DML dei file CSV (l'ordine non è importante) tranne i seguenti: dm_function_make_point.sql e dm_function_find_neighborhood;
- Eseguire i due script menzionati al punto precedente nell'ordine desiderato;
- Lanciare script DDL/ddl_constraints.sql;
- Lanciare script DDL/ddl_drop_temp_tables.sql.

Indicazioni sulle query

La produzione di ogni query è stata fatta a seguito di richieste precise dello stakeholder al fine di raggiungere gli obiettivi di analisi. Pertanto, ogni query di analisi dati, per entrambe le parti di esame, comprende la creazione e gestione di numerosi semilavorati come temporary tables o view (anche view materializzate per prestazioni migliori delle query), cosi come funzioni, al fine di raggiungere, tramite una **select finale** i risultati voluti.

Pertanto, si fa presente che le queries strutturate in questa maniera non sono banali select.

Ogni query, è stata sviluppata con l'idea di fornire una presentazione chiara dei risultati in linea con quanto espresso dallo stakeholder, sfruttando i costrutti SQL standard e spaziali (visti a lezione e non) necessari per raggiungere gli obiettivi .

Acronimi comuni nelle query

In questa sezione si rende noto che:

- I termini rental units, unità di affito, B&B, bnb sono usati in maniera interscambiabile nel corso della relazione e sviluppo queries;
- Per questioni di analisi sono state considerate solo le rental units significative denominate significant rental units. È quindi comune trovare le seguenti abbreviazioni
    - SRU: significant_rental_units
    - RU: rental_units

In generale si è cercato di commentare in maniera adeguata il codice prodotto.

Note sul DML

Il mapping del dominio della tabella POI è stato effettuato con un documento presente nella relativa cartella zippata all'interno della cartella datasets/spatial_dataset_zipped/nyc_point_of_intereset.zip.