Faculty of Media Engineering and Technology
German University in Cairo
Dr. Nourhan Ehab

Milestone 2: International Hotel Booking Analytics *(10%)*

Deadline: 2nd of December at 23:59

Introduction

After completing Milestone 1, you have gained a comprehensive understanding of the hotel booking dataset, including its structure, feature types, interconnections between different entities, and the critical issue of data leakage that can affect model performance.

Building upon this foundation, Milestone 2 focuses on constructing a Knowledge Graph (KG) database using Neo4j that semantically represents the relationships and entities within the hotel booking ecosystem.
The primary objective of this milestone is to transform the tabular hotel booking data into a rich, interconnected knowledge graph that captures the semantic relationships between travellers, hotels, reviews, cities, and countries. This knowledge graph will serve as the foundation for an intelligent recommendation system or travel assistant application that can help travellers make informed decisions about their travel destinations and accommodations based on their personal preferences, demographic characteristics, and travel history.

A key enhancement in Milestone 2 is the introduction of visa requirement data, which adds an important dimension to the travel planning process. This new file enables the system to consider visa requirements when recommending destinations, making the recommendations more practical and actionable for international travellers. The visa dataset captures country-to-country visa requirements, allowing the system to identify destinations that are accessible without visa complications or to inform travellers about necessary visa arrangements.

The Dataset:

1. **Hotels Dataset (25 hotels)**

The *hotels.csv* file*,* each row represents a unique property with key identifiers such as *hotel_id* and *hotel_name*, alongside geographical details like *city*, *country*, *lat*, and *lon*. These attributes are indispensable for answering the city-level question.

A hotel's baseline quality is represented by fields such as *star_rating*, *cleanliness_base*, *comfort_base*, and *facilities_base*. These base metrics can be contrasted with customer review scores to evaluate where hotels meet or deviate from expectations.

2. **Reviews Dataset (50,000 reviews)**

The file contains unique identifiers like *review_id*, *user_id*, and *hotel_id*, allowing seamless joining with the other tables to connect reviews with specific customers and hotels.
The scoring system, with columns including *score_overall*, *score_cleanliness*, *score_comfort*, *score_facilities*, *score_location*, *score_staff*, and *score_value_for_money*.

3. **Users Dataset (2,000 users)**

The *users.csv* file provides a list of unique customers, offering essential demographic insights with columns *user_id*, *country*, *age*, *gender*, and *Traveller_type* [*Solo, Business, Family, Couple*].

4. **Visa Dataset (381 rows)**

The visa.csv file (available on the CMS inside a zip file) indicates which travellers require a visa to be able to enter another country, using the Traveller's country.
Each row contains the following: the *from:* indicating the country that the user is from, the *to:* indicating the country to travel to, *requires:* a boolean indicating the need for a visa, and *visa_type:* indicating the type of visa if needed.

Milestone 2 Requirements:

This milestone requires you to construct a comprehensive Knowledge Graph database using Neo4j that represents the hotel booking domain according to the specified schema. The knowledge graph will model the complex relationships between travellers, their reviews, hotels, geographic locations, and visa requirements, creating a rich semantic network that enables sophisticated querying and analysis.

Your primary tasks include:

1. **Knowledge Graph Construction**

Implement a script (using Python and the Neo4j driver) that reads the provided CSV files and constructs the knowledge graph according to the exact schema specification provided below. This involves creating nodes for Travellers, Hotels, Reviews, Cities, and Countries, and establishing the relationships between them as defined in the schema.

2. Schema Compliance

It is critical that you strictly adhere to the provided schema. You are **not permitted** to modify the schema in any way, including:

- Changing the type of any node (*e.g., you cannot rename "Traveller" to "User" or "Customer"*)
- Changing the type of any relationship *(e.g., you cannot rename ":wrote" to ":created" or ":authored"*)
- Removing or modifying any properties specified for nodes
- Removing any node types or relationship types from the schema

3. Implementing The Given Scoring Rule

You must implement the "Exceed Expectations" scoring rule as a core analytical capability in your knowledge graph. This rule evaluates whether hotels meet or exceed the expectations of specific traveller demographics by comparing advertised base quality scores against actual review scores.

4. Cypher Query Development

After constructing the knowledge graph, you will develop Cypher queries to answer a comprehensive set of analytical questions. These queries will test your understanding of graph traversal, aggregation, filtering, and pattern matching in Cypher. The queries range from simple filtering operations to complex multi-hop traversals that require careful consideration of relationship directions and node properties.

.

The Knowledge Graph Schema:

1. Nodes:

- Traveller: *user_id* (unique identifier), age, type, gender
- Hotel: *hotel_id (*unique identifier), name, star_rating, cleanliness_base, comfort_base, and facilities_base, average_reviews_score
- City: name (unique identifier)
- Country: name (unique identifier)
- Review: *review_id* (unique identifier), text, date, score_overall, score_cleanliness, score_comfort, score_facilities, score_location, score_staff, and score_value_for_money.

2. **Relationships:**
    - (Traveller) - [:WROTE]-> (Review)
    - (Traveller) -[:FROM_COUNTRY] -> (Country)
    - (Traveller) - [:STAYED_AT] ->(Hotel)
    - (Review) -[:REVIEWED]-> (Hotel)
    - (Hotel) -[:LOCATED_IN]-> (City)
    - (City) - [:LOCATED_IN]->(Country)
    - (Country) -[:NEEDS_VISA]-> (Country);
        - Property: *visa_type*

Knowledge Graph Verification Queries:

NOTE ON VERIFICATION: These are example questions & answers for initial structure confirmation. Create queries that answer the following questions. If your queries return the same answers as below, then the structure of your knowledge graph is sound. The final assessment of the graph's design and robustness will involve a separate set of private queries to ensure the model is flexible and not hardcoded to these specific examples.

1. Retrieve the count of travellers who have visited a country where they don't need a visa. Include domestic travel (travellers visiting their own country) in the results.
   Answer:1999
2. Retrieve the top 3 hotels with the highest average ratings for *Business* travellers. The average rating should be calculated as the average of score_overall from all reviews written by *Business* travellers for each hotel.

3. Return the number of *couple* travellers who stayed at each hotel. Include all hotels in the results, even those with 0 *couple* travellers (show count as 0). Sort by Hotel name.

4. Return the top 3 hotels with an average review cleanliness below 8.8. Sort by the average review cleanliness in ascending order.

5. Return all hotels with the highest scores for locations for women, using the average score_location from reviews (include all ties).

Exceeds Expectations Rule:

A hotel is said to “exceed expectations” when the sum of its base quality scores is greater than or equal to the average of the review scores from a specific demographic group of travellers.

For each demographic group (e.g., each age group), focus on *Solo Female* travellers. while noting that for women travelling alone, the main concern would be the cleanliness and comfort of the hotel and the facilities they provide.
The following statistics should be calculated for hotels that exceed expectations:

- Minimum Improvement Percentage: The lowest improvement percentage among all hotels that exceed expectations for this demographic

- Maximum Improvement Percentage: The highest improvement percentage among all hotels that exceed expectations for this demographic

- Average Improvement Percentage: The mean improvement percentage across all hotels that exceed expectations for this demographic

These statistics help identify:
- Which demographic groups have hotels that consistently exceed expectations
- The range of performance (min to max) for hotels exceeding expectations
- The average level of expectation-exceeding performance per demographic

First row of the answer:

age_group	min_improve	max_improve	avg_improve
‘18-24’	1.7	4.8	3.51

Deliverables:

You are required to submit the following by filling out the form:

1. **Create_kg.py**
    - Python script that creates the Knowledge Graph in Neo4j.
    - Should read CSV files located in the same directory
    - Should follow the exact schema specification.
    - The script should use the config.txt file with Neo4j credentials. Not doing so might result in 0.
2. **config.txt**

```
URI=neo4j://localhost:7687

USERNAME=neo4j

PASSWORD=your_password
```

3. **rule.txt**

A text file containing the implemented rule for the scoring task.

4. **queries.txt**

A text file containing the queries for the above questions with their numbers.

```
//1. Query 1

Match(...)

Return (....)

//2. Query 2

Match(..)

Return (....)

Etc..
```