



Database Applications ISYS1101/1102 | Semester 2 2024 Assignment 2: MongoDB

Changelog:

A minor correction was made to Task 1, Question (e):

Old: Display the listings within a \$10-range from the average price you found in Part (e). i.e.

New: Display the listings within a \$10-range from the average price you found in Part (d). i.e.

Assessment Type	Series, and
Due Date	
Demonstrations	
Silence Period	
Weight	
Submission	provided on Canvas.

1 Overview

1.1 Assessment Criteria

This assessment will determine:

1. analyse the requirements outlined in the problem description;
2. write MongoDB queries required for CRUD (create, read, update and delete) operations on the Airbnb database available on MongoDB Atlas;
3. expand the Airbnb database with additional document collections to accommodate bookings;
4. write MongoDB queries to manage bookings;
5. be able to demonstrate the functionality of your database and answer questions posed by the marker

1.2 Learning Outcomes

This assessment will assess how you attained the following course learning outcomes:

CLO 1: apply advanced data analysis and modeling concepts, physical design, integrity, security and transaction management.

CLO 4: build an efficient database application with an emphasis on storage management, indexing, and query optimization;

CLO 5: use MongoDB for store and retrieve non-relational data;

2 Assessment Details

2.1 Preparation Work

1. You must have deployed your MongoDB database on MongoDB Atlas, the cloud-based MongoDB database platform (cloud.mongodb.com). You must have populated your database with the sample database called "sample_airbnb" with "listingsAndReviews" documents collection;
2. You are required to have the Mongo Compass client application installed on your laptop, having it properly configured to access your MongoDB database running on MongoDB Atlas (see above);

Week 9 lab sheet walks you through, in step-by-step detail, on how to set up MongoDB Atlas and MongoCompass. If you haven't completed Week 9 lab, you must complete it before attempting the assignment tasks outlined below. Apart from Week 9 Lab, no further instructional support provided for setting up your database. You can acquire these skills through your consultation sessions to get any teething issues resolved.

You must be very familiar with MongoDB query language, able to write simple queries, and aggregation pipelines on it. You can acquire these skills through your consultation sessions.

2.2 Assignment Task

Milestone 1: Written Submission

Task 1: CRUD tasks on listings

In this task, you are required to write MongoDB queries in JavaScript language. Some of the queries may involve multi-stage aggregation pipelines. Importantly, you must write the queries in MongoDB Shell format; and (2) you must provide the queries as inputs into Mongo Compass.

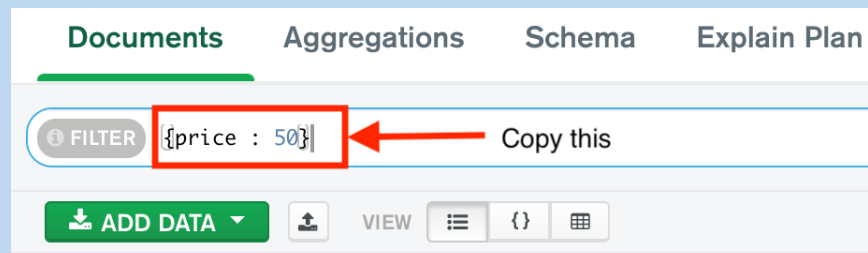
Do not submit any screenshots of Mongo Compass, only statements in plain text are to be submitted.

Sample Question: Display listings with a daily rent (price) of \$50.00

You will try it out on both interactive Compass window and Mongo Shell

```
>_MONGOSH
Atlas atlas-ezxrzs-shard-0 [primary] sample_airbnb> db.listingsAndReviews.find( {price : 50} )
```

Copy this



Sample Answer to be submitted:

Mongo Shell: `db.listingsAndReviews.find ({price : 50})`

Mongo Compass: On Filter field: `{price : 50}`

Write MongoDB statements to extract data on each of the following filtering conditions. For the questions below, use the 'market' sub-field of the Address field. In the tourism industry, it is a commonly used term to determine specific tourist destinations.

- a. Display 3-bedroom apartments in Barcelona.
 - b. Display listings in Barcelona that has both Wifi and Cable TV.
 - c. Compute and display the average daily rate for Barcelona properties.
 - d. Display the average price for 3-bedroom apartments in Barcelona.
 - e. Display the listings within a \$10-range from the average price you found in Part (d)(e). i.e. 3-bedroom apartments in Barcelona with price in the range $[\text{avgPrice} - 5 \rightarrow \text{avgPrice} + 5]$.
 - f. Display 10 "least-expensive" tourist destinations for travelers. The "least expensive" is determined by averaging the prices of all properties in each destination.
2. The owner of the listing with ID 123456789 wants to add "Pet-friendly" as a new amenity. Write a MongoDB statement to update the listing.
 3. A recent guest has left a review for the listing with ID 123456789. The review text is "This holiday accommodation was amazing. The location is perfect for those who want to watch bull-fighting from the balcony." Write a MongoDB statement to insert this review into the reviews array of the listing.
 4. The current daily rate for the listing with ID 123456789 is 100. Fabio, the owner, computes the average daily rate for all listings in the same market, i.e. Barcelona. He only considers listings with 3 bedrooms and 2 bathrooms. He finds that the average daily rate is 110. Write a MongoDB statement to change the daily rate of the afore-mentioned listing to 110% of the current rate.
 5. Fabio, the owner of the above-mentioned listing, didn't like the recent review given by the guest mentioned in the question 3. He wants to delete this feedback. Write a MongoDB statement to delete the feedback currently stored in the reviews array.

Task 2: Extend the AirBnB database

The current sample database consists of only one document collection, which comprises of information on listings and reviews. It is built on "embedded" data model, where each document outlines information on **listings** and for each listing, additional information such as user **reviews** and hosts' details are "**embedded**".

In this task, you are required to extend the database to allow taking bookings of listings. The specifications for the extended system is outlined below.

For simplicity, assume that listings can only be booked for whole days, and no hourly bookings are accepted. At the moment, listings can be booked into the future, as far as 31 December 2026.

Bookings are made by registered clients of the AirBnB system. Each client has a unique client_ID generated by the system. In addition, they use an email address for login.

Bookings are for individual listings, i.e. you cannot book multiple listings in a single booking. Each booking is for a specific period, defined by the arrival date and departure date.

In addition to the listing name, arrival and departure dates, each booking will capture the identification details of the client, such as name, email address, daytime phone number, mobile number, postal address and home address. The other information included in a booking are: the deposit paid at the booking, the balance due, the due date for the balance payment, and number of guests, and for each guest the names and their ages. These other guests are not required to be registered clients of the Airbnb system.

Your tasks:

1. Under **embedder** in the **models** folder, create a new file named **embedder.py**. This file will demonstrate how these additional requirements are used in a MongoDB database. You must identify the data fields that are required and state whether they are unique (e.g., `unique=True` in `fieldsAndReviews`) or not. Attach the `embedder.py` file to the `models` folder.
 2. Provide MongoDB queries to carry out the tasks listed below. Some queries may be simple while some of them may require a multi-step query. Provide the queries for the both models you proposed in the previous task.
 - a. Add the `embedder.py` file to the `models` folder.
 - b. Display the client name, email address, booking start date, and booking end date for all bookings made so far for the listing named “Be Happy in Porto”.

Do not submit any screenshots of Mongo Compass, only statements in plain text are to be submitted.

The answers to the questions in Task 1 and Task 2 must be compiled into a single pdf document. You may use a plain text file or a word document to compile the answers, however, the final submission must be in pdf format. Only one file must be submitted.

Milestone 2: Demonstrate your work.

You are required to do a demo of your complete application hosted on MongoDB Atlas. These demos will be conducted during the week of 14th October. You should have some booking data stored in the application and the tutor will test out the functionality by using a test bed of data.

Tutors will use a standard testing plan for all submissions and you will only receive your Task 1 and Task 2 marks if you can demonstrate the functionality of your application during the demo.