



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

Assessment Type	MongoDB Database Design, Implementation, Queries, and Demonstration Individual Assessment
Due Date	23:59 Sunday 13 October 2024
Demonstrations	Week starting Monday 14 October 2024
Silence Period	Starts at 5:00PM Friday 11 October 2024
Weight	25%
Submission	Online, via Canvas. Submission instructions are provided on Canvas.

1 Overview

1.1 Assessment Criteria

This assessment will determine your ability to:

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 labsheet 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 environment at this stage, however, you can use regular consultation sessions to get any teething issues resolved.

You must be very familiar with the document structure of the listingsAndReviews document collection, able to write simple queries on this document collection and be able to build aggregation pipelines on it. You can acquire these skills by completing Labsheets 9 – 11.

2.2 Assignment Task Description

Milestone 1: Written Submission

Task 1: CRUD tasks on listingsAndReviews Document Collection [Week 8 & 9. Week 10 Lab.](#)

In this task, you are required to provide statements written in MongoDB Query Language. Some of the statements can be written as a simple query, while some of them require a multi-stage aggregation pipeline. Importantly, you must provide your answers in **two** formats: (1) in Mongo Shell format; and (2) 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

Documents

Aggregations

Schema

Explain Plan

FILTER

{price : 50}

Copy this

ADD DATA

VIEW

VIEW

VIEW

VIEW

VIEW

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.

address.market

I am CONFUSED ->

- 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 (e). i.e. 3-bedroom apartments in Barcelona with price in the range $[avgPrice - 5 \rightarrow avgPrice + 5]$.
 - 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 named "Be Happy in Porto" wishes to add "Netflix" as a new amenity. Write a MongoDB statement to add this new amenity into the amenities list of the existing listing.
 3. A recent guest to the listing named "Be Happy in Porto" wishes to add the following comment as a guest review:
"This holiday accommodation did not meet my expectation. Being in Portugal, I wanted to watch bull-fighting from the balcony. But, neither balcony nor bull-fighting nearby are there."
 Write a MongoDB statement to add this comment into the reviews array of the existing listing of this property. Your new review must be of the same structure as any existing review.
 4. The current daily rate for the listing named "Be Happy in Porto" is \$30.00. Fabio, the owner, computes the average daily rate of similar properties in the area. For similarity comparison, he uses the property type and the number of bedrooms of the properties in the same market, i.e. he only considers the properties that are same type as his property and also having the same number of rooms as his. Then, he decides to set his property's daily rate 10% lower than the average daily rate. Area = address.market:"Porto", number of rooms = bedrooms
 Write a MongoDB statement to change the daily rate of the afore-mentioned listing.
<https://www.mongodb.com/docs/mongodb-shell/crud/update/>
 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.
<https://www.mongodb.com/docs/mongodb-shell/crud/delete/>

Task 2: Extend the AirBnB database Discussed in Week 10.

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". <- Documents embedded in documents.

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.

List of conditions in the sample data.

1. 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.
2. 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.
This is an object: Client {client id <field>, email <email details>}
3. 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.

A listing refers to a document in listingsAndReviews

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.

Just transform the sample data.

Your tasks: This is both an explanation and demonstrations

1. Under embedded and referenced approaches, using two examples, demonstrate how these additional requirements are accommodated in an extended version of this database.
You must identify in your model, the additional collections or fields required and state whether your new information are “embedded” within the current collection (listingsAndReviews) or build as separate collections using “reference” approach.
As sample data, you may use the new bookings data found in a text file attached to the assignment folder. <- Sample data will be posted on 1st of October
Reference will be a new collection.
2. Provide Mongo Shell statements, written in MongoDB Query Language, to carry out the tasks listed below. Some of the statements can be written as a simple query, while some of them may require a multi-stage aggregation pipeline. Importantly, you must provide queries for the both models you proposed in the answer for above question.
 - a. Add the first booking in the sample bookings file to your extended database.
 - 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.

You are not required to submit anything for the Milestone 2.

3 Submission

Follow the instructions on Canvas to complete your submission.

3.1 Milestone 1: Submission

You will need to submit:

- One pdf file;
- 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 editor such as Visual Studio Code to compile the answers, however, the final submission must be in pdf format;
- Do not submit data or screenshots;
- If you use any references to other resources (such as MongoDB website, StackOverflow, ChatGPT etc) they must be clearly identified so and follow the referencing guidelines stated here: <https://www.lib.rmit.edu.au/easy-cite>

3.2 Milestone 2: Demonstrating your work

No additional submissions required for this task. More details of the marking process, including how to book a marking session and also sample data requirements will be published later.

3.3 Assessment Declaration

When you submit work electronically, you agree to the [RMIT assessment declaration](#).

3.4 Late Submissions & Extensions

A penalty of 10% per day is applied to late submissions up to 5 business days, after which you will receive zero marks.

Short extensions may be granted by the course coordinator up to 1 business day *before* the due date in accordance with RMIT Assessment Adjustment process. However, extensions are not guaranteed and require suitable documentation. The course coordinator may refer requests to Special Considerations.

Special Consideration *may result in an equivalent assessment*, which may take the form of a timed assessment assessing the same knowledge and skills of the assignment and are generally granted on an individual basis. For more information refer to the [RMIT Special Consideration process](#).

3.5 Supported software for assessment and grading

All of the answers must be based on the sample database instance outlined in the Preparation Activities (sample AirBnB database on Mongo Atlas).

During the marking process, the marker will run your queries on an identical instance of the same database. As such, you should not make any changes to your copy of the database in whatsoever way. You won't be awarded marks if the marker cannot run your queries on the sample database instance.

4 Marking Guidelines

4.1 Task 1

- 15 marks for the statements required for tasks 1 – 6.

4.2 Task 2

The marks are divided into the following categories:

- 4 marks for the database design;
- 6 marks for the queries.

The detailed breakdown is provided on the marking Rubric available on Canvas.

5 Academic Integrity and Plagiarism (Standard Warning)

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e., directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites. If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [RMIT Academic Integrity Website](#).

The penalty for plagiarised assignments includes zero marks for that assignment, or failure for this course. Please keep in mind that RMIT University uses plagiarism detection software.