DOTEE6751BB

Database and Big Data Management

2025R2 Project

Total marks (100 +5 [capped at 100])

Submission should be one .pdf document detailing the answers.

See Submission Instructions Later,

Q1. [45 Marks] Assume that we have a collection of documents for course records on an online learning platform. fields include: course ID (_id), title, Instructors, Year, and Category. Below shows an example document and its fields.

Requirements: Please download courses.json file and load collection. Run query on collection. For every query, submit both: (i) query you wrote, and (ii) result you obtained (e.g., a screenshot after executing query).

```
{
      _id: "cs101",
      title: "Introduction to Computer Science",
      instructors: ["Prof. Alice", "Dr. Bob"],
      year: 2023,
      category: "Computer Science",
      averageRating: 4.6
}
```

Use courses collection and answer following using MongoDB queries. Submit both queries and screenshots of results.

a. Count number of courses created before year 2020. [5 Marks]

b. List all courses taught by “Prof. Alice" but not in “Computer Science". [5 Marks]

c. Find all courses where at least one instructor is “Prof. Alice" or “Prof. Carol". [5 Marks]

d. Find all distinct categories of courses. [5 Marks]

e. List courses with more than one instructor created in or after 2020. [5 Marks]

f. Get titles and years of 5 “Data Science" courses with highest average rating. [5 Marks]

g. Compute average rating of courses for each category. [5 Marks]

h. Count total number of courses taught by each instructor. [5 Marks]

i. Group courses by decade and category, showing how many courses are in each group. [5 Marks]

!End of Question One!

!Start of Context for Questions Two to Five!

In this question, we build a database for **TaxiNow**, a ride-sharing platform. Primary details are as follows:

- **Customer Information**: Each user has a unique ID, name, gender, phone number, email, and registration date.
- **Driver Profiles**: Drivers have a unique ID, name, gender, phone number, license number, background check status, number of completed rides, average rating, signup date, and current status (e.g., active, suspended, off-duty, banned).
- **Driver Badge**: Drivers may earn one or more badges as recognition for their conduct or performance. There at most three badges: "Honest Driver," "Safe Driver," and "Excellent Service." Each badge has a name and description.
- **Booking Details**: Each booking is identified by a unique ID and includes details such as pickup and drop-off locations, timestamp when booking was made, time driver accepted booking, and fare. It also records assigned driver and vehicle, ride status (requested, accepted, cancelled, completed), as well as trip date. Let's assume each booking can only be made by one person.
- **Payment Details**: Each booking is associated with some payments: PaymentID, date, amount, payment method, and status (paid, failed, refunded).
- **User Ratings & Reviews**: After each trip, users have option to rate driver on a scale from 1 to 5; this is not mandatory. Users may also provide optional written reviews, such as comments about driver's professionalism or complaints about issues like smoke odours in vehicle.

Q2. [15 Marks] Familiar Territory (ER Modelling!) Checking Understanding

a. Create an ER diagram. This diagram should detail entities, their attributes, relationships among them, and any constraints that apply. [10 Marks]
b. Explain why you chose specific relationship types (one-to-one, many-to-one, many-to-many) between entities, and the participation constraints [5 Marks]

Q3. [15 Marks] Now Suppose your client suddenly learns of mongoDB; you are asked to now express the business application as one or more collections that retains all the information as above!!!

a. Provide commands that create the collections that you are expecting to use — your MongoDB code for these should be included in your final report. [5 Marks]
b. For each collection, insert some documents into your collection —your mongoDB code for these should be included in your final report. [10 Marks]
    - (You may think of what you need for a basic semi-real scenario just enough sample to answer the following questions below)

- Note: from lecturer:
    - **Try your best, creatively, if absolutely unable, please explain briefly what you have tried, and why you do not think they work.**
        - Please ignore any mongoDB related schema validation until the last question
    - Unlike Relational Mode, we must accept data to now be denormalized (many duplications)
        - Worst case scenario, you might put all columns from Q1 into one document? Not ideal, but workable, there are better ways for this question!

- **TIP1:** Use Embedding (Single Collection) whenever possible
    - i.e., for one-to-one or one-to-many relationships where the related data is frequently accessed together, embed the related data within a single document to reduce read operations!
    - **Example: Instead of separate Booking and Payment tables, a single Booking document can contain an array of embedded Payments.**

- **TIP2:** Reference (Multiple Collection) when necessary:
    - i.e., for complex many-to-many relationships, try use references
    - **Example: storing the _id of one document within another.**
    - **TIP: mongoDB provides the $lookup aggregation stage for performing joins when needed**

Q4. [25 Marks.] Below are some queries for 5 possible scenarios. Please write corresponding query and show a picture of your outputs. Iff you tried and find impossible, explain why with respect to your model, and how you can make it possible.

a. List top 5 most frequently requested (pickup, drop off) combination in the database.
b. List all bookings from past week where payment failed at least once. *(Tip: see the $$NOW aggregation variable)*
c. Count how many bookings has been completed in last 30 days. *(Tip: see the $dateDiff operator)*
d. For a given customer, list all his/her completed rides, including driver's name, car model, and the number of reviews he/she has given.
e. For each driver, calculate total of trips completed, number of those trips that received reviews, and the percentage of trips with reviews.

BonusQ5. [5 Marks] (Overall marks for project capped at 100 marks)

It is well known that mongoDB had an unpredictable Data Structure: If not strictly managed, flexible schemas can lead to data chaos… But there are some in-schema validation mechanism involved that we didn't cover in class.

- Take a look at in-schema validation in mongoDB, that resembles column constraints in relational database.
  - https://www.mongodb.com/docs/manual/core/schema-validation/

- Rewrite your command from Q2a. creating the collection, apply this validation rule to your collection. Your mongoDB code for these should be included in your final report!
  - Enforce the payment status information to be included, and must be exactly one of the few options (paid, failed, refunded)
  - Enforce that time the booking is made is on or before today.

!End of Group Project, Hope you had fun?