**Business Case: Dubai Tourism InsightHub**

**Context**:  
Dubai Tourism InsightHub is an initiative by the Department of Economy and Tourism to enhance Dubai's tourism offerings and provide a data-driven approach to decision-making. Dubai is one of the most visited cities in the world, attracting millions of tourists each year. With a diverse range of attractions, events, and experiences, the city aims to better understand tourist behaviors, preferences, and trends to optimize its tourism strategy.

**Objective**:  
To collect, process, and analyze data from various tourism-related activities in Dubai to provide actionable insights that guide decision-making, improve visitor experiences, and drive revenue growth.

**Datasets**:

1. **Attractions.txt**: Contains details of major tourist attractions in Dubai.
2. **Visitors.txt**: Includes information about tourists visiting Dubai, their demographics, and preferences.
3. **Spending.txt**: Records visitor spending across various categories (accommodation, shopping, entertainment, etc.).

**Sample Data Structure**:

1. **Attractions.txt**
   * aID (integer): Attraction ID
   * aName (string): Name of the attraction
   * aType (string): Type of attraction (e.g., Cultural, Adventure, Shopping)
   * aLocation (string): Area or district of the attraction in Dubai

**Example:**

101, Burj Khalifa, Cultural, Downtown

102, Dubai Mall, Shopping, Downtown

103, Desert Safari, Adventure, Al Awir

1. **Visitors.txt**
   * vID (integer): Visitor ID
   * vName (string): Name of the visitor
   * vCountry (string): Country of origin
   * vAge (integer): Age of the visitor

**Example**:

201, John, USA, 30

202, Sarah, UK, 25

203, Rashid, Saudi Arabia, 45

1. **Spending.txt**
   * sID (integer): Spending ID
   * vID (integer): Visitor ID
   * aID (integer): Attraction ID
   * sAmount (integer): Amount spent in AED
   * sCategory (string): Category of spending (e.g., Accommodation, Tickets, Shopping)

**Example:**

301 201 101 200 Tickets

302 202 102 500 Shopping

303 203 103 1000 Adventure

Part -A

Questions:

**Question 1**: **Create a database named tourismDB. Then, create a collection named visitors with the following document structure and data:**

vID: 201,

vName: "John",

vCountry: "USA",

vAge: 30

**(4 marks)**

**Model Answer and Marking Scheme**

Create a database named tourismDB

**use tourismDB**

Create a collection named visitors

**db.createCollection("visitors")**

Insert a document into the collection

**db.visitors.insertOne({**

**vID: 201,**

**vName: "John",**

**vCountry: "USA",**

**vAge: 30**

**})**

Marking Guide:

* 2 marks for correctly creating the database and collection.
* 2 marks for inserting the document correctly.

**Question 2: Insert multiple records into the attractions collection in the tourismDB database. The data should include:**

aID: 101, aName: "Burj Khalifa", aType: "Cultural", aLocation: "Downtown"

aID: 102, aName: "Dubai Mall", aType: "Shopping", aLocation: "Downtown"

**(4 marks)**

**Model Answer and Marking Scheme**

**db.attractions.insertMany([**

**{ aID: 101, aName: "Burj Khalifa", aType: "Cultural", aLocation: "Downtown" },**

**{ aID: 102, aName: "Dubai Mall", aType: "Shopping", aLocation: "Downtown" }**

**])**

Marking Guide:

* 2 marks for using insertMany correctly.
* 2 marks for inserting the documents correctly.

**Question 3: Write a MongoDB query to find all visitors from the country "USA" in the visitors collection. (4 marks)**

**Model Answer and Marking Scheme**

**db.visitors.find({ vCountry: "USA" })**

Marking Guide:

* 4 marks for using the correct find query with the appropriate filter.

**Question 4: Update the aLocation of the attraction with aID 101 to "Uptown". (4 marks)**

**Model Answer and Marking Scheme**

**db.attractions.updateOne(**

**{ aID: 101 },**

**{ $set: { aLocation: "Uptown" } }**

**)**

Marking Guide:

* 2 marks for correctly filtering the document with aID: 101.
* 2 marks for using $set correctly to update the location.

**Question 5: Write a MongoDB query to display all attractions of type "Shopping" located in "Downtown". (4 marks)**

**Model Answer and Marking Scheme**

**db.attractions.find({ aType: "Shopping", aLocation: "Downtown" })**

Marking Guide:

* 2 marks for correctly using multiple conditions in the query.
* 2 marks for using the correct syntax for the find command.

**Question 6: Write a MongoDB query to find all visitors who are above the age of 40. (4 marks)**

**db.visitors.find({ vAge: { $gt: 40 } })**

Marking Guide:

* 2 marks for using the $gt (greater than) operator correctly.
* 2 marks for using the correct command find.

Part – B

**Question 7:**

**Identify and explain any two key limitations of MongoDB in handling large-scale applications.** **(2 marks)**

**Model Answer:**

Some key limitations of MongoDB include:

1. **Limited Support for Complex Joins**: MongoDB struggles with executing complex joins efficiently, which can affect performance when dealing with highly relational datasets.
2. **Lack of Strong ACID Compliance for Multi-Document Transactions**: MongoDB does not offer strong ACID compliance across multiple documents, potentially leading to data inconsistencies during complex operations.
3. **High Memory Usage Due to Data Redundancy**: The document-based structure can result in data duplication, leading to higher memory consumption and increased storage costs.
4. **Scalability Challenges in Write-Heavy Workloads**: MongoDB's horizontal scaling may face performance bottlenecks in write-heavy applications due to limited write concurrency.

**Note**: Students may answer any two of the above limitations.

**Marking Scheme:**

* **1 mark** for identifying and explaining the first limitation.
* **1 mark** for identifying and explaining the second limitation.