Part 1 involves the design and implementation of a MongoDB application.

You have been commissioned to design and build a database for an airline company. The system is required to store information about the flights it runs, journey bookings, passengers and air crew. Specifically, the following details should be stored:

1)Pilots: including details of their address, their other contact details and information about their employment record with the Company, and the date of their last "fit for flying" test

2) Planes: details of their make, model, flying range (with a full fuel load), length of service, status (i.e. working, being repaired or upgraded etc), seating capacity

3) plane flights: including the plane making the flight, starting airport, destination airport, including the dates/times of departure and arrival, the pilot and co-pilot assigned to the flight, length of flight in kilometres. (Note that co-pilots must be drawn from the available qualified pilots).

4) Journey bookings: including date/time the booking was made, who made the booking, the passengers on the booking, flights used to get from the start to the final destination (note that a booking might involve the use of more than one plane flight), the total cost of the booking

5) Airports: including the name, location, cost of use (as a simplification, you can assume that each airport charges a fixed hourly rate for the length of time a plane stops at an airport, plus a further fixed charge for refueling, both of these charges will vary from airport to airport)

6) Airline employees: as a simplified staffing model, you can assume the airline employs booking clerks, maintenance staff, pilots and cabin staff.

7) Revenue: you need to be able to calculate the total revenue made or lost by the airline. The simplest model of this can be calculated as:

Revenue = total money from bookings – airport costs – total salary bill

Deliverables

Note that in all of the following, where MongoDB code is required, this code must be provided in the form of the text of the code itself, not as screen shots. However, if input data or output from queries are required, these can be shown as screen shots if you wish.

You are required to produce the following:

a) A list of any assumptions you have made during the development of your system

b) Details of your MongoDB schema. This includes details of the documents to be stored in your system and the collections used to store them. This should include identifying any array or embedded document structures in your design. You should also describe relationships between documents, e.g. one to many, many to many etc. This information should be provided in the form of text giving the details of documents and the fields to be stored. You can also supplement this text with a diagram if you find that useful. You should also provide a textual explanation of the choices and decisions you made during the design process.

c) A listing of the MongoDB code used to implement your MongoDB schema (please remember no screen shots here, just the text of the code).

d) You are then required to populate your schema with data. It is not necessary for you to enter a large amount of data, for most tables between 6-10 rows should be sufficient. However you should design your test data to make sure that it will demonstrate that your MongoDB queries will work robustly no matter what data is stored.

e) A set of 12 MongoDB queries to extract information from the system. You should aim to use a good range of MongoDB language constructs to demonstrate your knowledge of the language. These should include several examples of the use of the MongoDB Aggregate Framework. (please note again the text of query code must be provided, not screen shots, but screen shots of query output are acceptable).

Part 2 concerns the application of performance tuning tools provided in the MongoDB environment to your implemented application.

a) Demonstrate the use of indexes with the use of the Explain function. Do this by selecting two of your more complicated queries, preferably ones that access a significant proportion of your data. Run these queries with and without indexes, commenting on the output obtained from the Explain function. With the amount of data most groups are likely to have, you may not obtain differences in the execution times of queries with and without indexing, but in any case you should comment on the output produced by the Explain function in the presence and absence of indexes.

b) Show the use of the MongoDB profiler for examining the performance of a selection of your MongoDB queries. Illustrate the use of different values of the parameters to the MongoDB Profiler utility.