Cloud Computing Homework 1: Databases

Task 1 Detailed Report

Task 1 Specifications:

Being a consultant at a company called Aggies Forever, I was asked to create a fitness aware environment in the workplace and to work specifically on their fitness portal called AggieFit. The company wanted to design a system that will track the fitness activities of the employees and reward them for maintaining their health.

To maintain some anonymity, there is a separate portal in which employees create a Unique ID for AggieFit. The map between the unique ID for the fitness portal and employee records are hidden from the AggieFit team. The employees can, however, choose to expose their data. The employees can also choose to add their personal details (e.g., height, weight, age etc.) to AggieFit. Employees must choose from a list of fitness goals. They will then report their activities by integrating their fitness devices.

The task was to assist the AggieFit team in designing their system by providing prototypes and comments. As they designed their system, I was asked for help in various aspects of their system.

As their resident database expert, I was asked to design a database for AggieFit. I was provided with some dummy fitness data. Each employee used a different device to track their activities, and some employees used multiple devices. The employees might have shared their personal information. Some employees may also have decided to make their employment information available to AggieFit.

Approximate Time Taken to Complete the Task: 3 Hours

The time includes connection, configuration and integration of MongoDB using Python and also the queries to be performed as per the client instructions.

Task1a) Why MongoDB is a better fit in comparison to any Relational Database Model for AggieFit?

MongoDB, being a Collection and Key-Value based Document Data Model has quite a lot of advantages over a Relational Data Model. MongoDB offers Dynamic Schemas which makes it very easy to add objects and fields at the same time, especially unstructured data, into the database unlike traditional relational databases where schema needs to be pre-defined before loading any data which makes it difficult and time-taking process to load data when any alterations are needed or new fields are added in the tabular structures. So, clearly MongoDB gives a better performance than a Relational Database.

Also, while people experience difficulties with large data volumes in Relational Databases, MongoDB offers better storage capacity and speed with documents along with greater efficiency and reliability on handling large volumes of data.

Unlike some relational databases, MongoDB is built on a distributed systems architecture, rather than a monolithic, single node design. As a result, MongoDB offers out-of-the-box scale-out and data localization with automatic sharding, and replica sets to maintain always-on availability.

Task1b) Execute the given 6 sample queries on your database. Provide any suggestions.

WQ1: 6 new documents with respective fields were added from dummy-fitness.json file into the database.

WQ2: The database was updated with new fields and records of User 1001.

RQ1: Number of employees whose data was in the AggieFit database were counted.

- 9

RQ2: Employees who have been tagged as 'active' were retrieved.

- {u'_id': ObjectId('5a972cb60df3a315b6e27bcf'), u'uid': 1003}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc5'), u'uid': 1005}

RQ3: Employees that have a goal step count greater than 5000 steps were retrieved

- {u'_id': ObjectId('5a972cb60df3a315b6e27bcd'), u'uid': 1001}
- {u'_id': ObjectId('5a972cb60df3a315b6e27bcf'), u'uid': 1003}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc5'), u'uid': 1005}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc8'), u'uid': 1008}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc9'), u'uid': 1009}

RQ4: The total activity duration for each employee was aggregated.

- {u'_id': ObjectId('5a972cb60df3a315b6e27bcd'), u'activityDurationTotal': 140}
- {u'_id': ObjectId('5a972cb60df3a315b6e27bce'), u'activityDurationTotal': 276}
- {u'_id': ObjectId('5a972cb60df3a315b6e27bcf'), u'activityDurationTotal': 0}
- {u'_id': ObjectId('5a972ceb0df3a315f5887fc4'), u'activityDurationTotal': 501}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc5'), u'activityDurationTotal': 220}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc6'), u'activityDurationTotal': 0}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc7'), u'activityDurationTotal': 132}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc8'), u'activityDurationTotal': 435}
- {u'_id': ObjectId('5a972cec0df3a315f5887fc9'), u'activityDurationTotal': 423}

Few Suggestions in Improving the Design of the AggieFit Database:

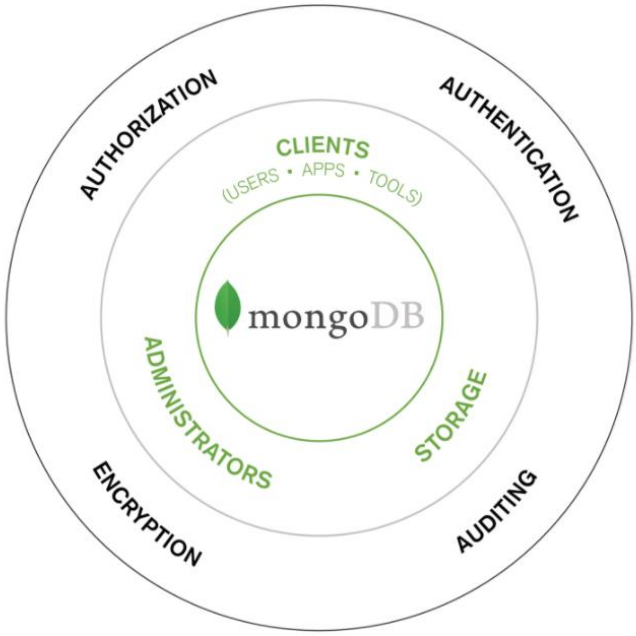
Requirements for Database Security & Compliance

In light of increasing threats to data security with heightened concern for individual privacy, a series of initiatives can be designed to increase security, reduce fraud and protect personally identifiable information. There are common foundational requirements, including:

- Restricting access to data, enforced via predefined privileges and security levels.
- Measures to protect against the accidental or malicious disclosure, loss, destruction or damage of sensitive data.
- The separation of duties when running applications and accessing data.
- Recording the activities of users, administrative staff, and applications in accessing and processing data.

A Holistic MongoDB Database Security Architecture must cover:

- User access management to restrict access to sensitive data, implemented through authentication and authorization controls.
- Logging operations against the database in an audit trail for forensic analysis.
- Data protection via encryption of data in-motion over the network and at-rest in persistent storage.
- Environmental and process controls.



MongoDB Database Security Architecture Sample