

## **HW2: Databases (Task – 1 Report)**

Himanshu Gupta (UIN: 326003459)

Manish Singhal (UIN: 326007129)

**Time to complete the task:** Roughly 4 hours

**Task1a:** A section on the features of MongoDB that make it suitable for the application. If you disagree with your client, you will have to provide a thorough explanation.

The greatest advantage of using MongoDB is the flexibility in schema design. We do not need to pre-define the fields of the data and can be modified at any point. There are many other performance benefits for using MongoDB like it can handle a very high throughput and handle more queries than a standard SQL database. It makes sure that the amount of data does not create a performance impact. The aggiefit database will have a huge amount of entries for as the users grow, the application will track their fitness history, so it can get very big with time.

MongoDB can be easily scaled up with cloud-based services. MongoDB is built to provide scalability using multiple instances and replicas. Since they do not need any schema for the data, it saves a lot of software development effort. The current data provided is a simply json formatted unstructured data, which can be stored as it is in the database. MongoDB uses key-value pairs to track the entries and perform search and update operations. As it does not need to support strong consistency in the data, it can be easily sharded.

**Task1b:** Include any comments you have about improving the design of the database.

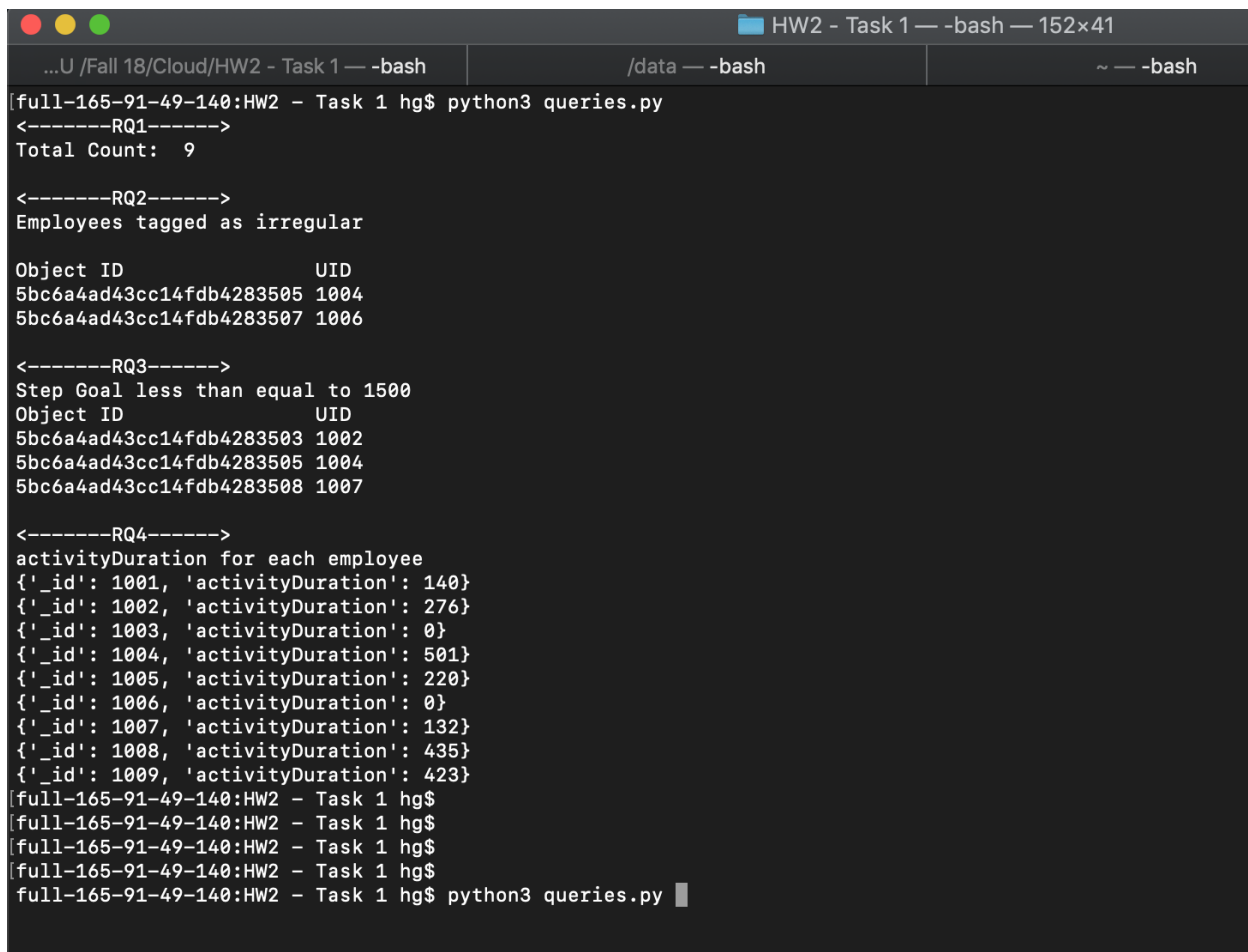
1. **Limited access of read and write:** It is required to give limited access of read and write operation to the database. It will help in keeping the database secure and keeping the data uncorrupted.
2. **Data Replication:** It is storing data in more than one site or node. It is useful in improving the availability of data. Copying data from a database from one server to another server helps in faster access of the same data without any inconsistency. The result is a **distributed database** which increase the availability and accessibility of data.
3. **Proper Logging:** Database should be designed in such a way that there is proper logging for each query/operation that is performed. It not only helps in easy debugging at the time error but also helps understand the working of the model.
4. **Caching:** Database caching improves scalability by distributing query workload from backend to multiple cheap front-end systems. It allows flexibility in the processing of data. Caching can improve availability of data, by providing continued service for applications that depend only on cached tables even if the backend server is unavailable. Another benefit is improved data access speeds brought about by locality of data and smoothing out load peaks by avoiding round-trips between middle-tier and data-tier.

5. **Consistency:** Data replication encompasses duplication of data in other nodes, so it is required that the replicate is in a consistently updated state and synchronized with the source. We need to make sure data is consistent at all the nodes.

**GitHub link:** <https://github.tamu.edu/manish-singhal/CSCE678-HW2>

### Steps to Run the code:

1. Unzip the folder, it contains the json files (input), queries.py and its dependencies.
2. Run command: **python queries.py**
3. We will be able to see the output as shown below.



```

[full-165-91-49-140:HW2 - Task 1 hg$ python3 queries.py
<-----RQ1----->
Total Count: 9

<-----RQ2----->
Employees tagged as irregular

Object ID          UID
5bc6a4ad43cc14fdb4283505 1004
5bc6a4ad43cc14fdb4283507 1006

<-----RQ3----->
Step Goal less than equal to 1500
Object ID          UID
5bc6a4ad43cc14fdb4283503 1002
5bc6a4ad43cc14fdb4283505 1004
5bc6a4ad43cc14fdb4283508 1007

<-----RQ4----->
activityDuration for each employee
{'_id': 1001, 'activityDuration': 140}
{'_id': 1002, 'activityDuration': 276}
{'_id': 1003, 'activityDuration': 0}
{'_id': 1004, 'activityDuration': 501}
{'_id': 1005, 'activityDuration': 220}
{'_id': 1006, 'activityDuration': 0}
{'_id': 1007, 'activityDuration': 132}
{'_id': 1008, 'activityDuration': 435}
{'_id': 1009, 'activityDuration': 423}
[full-165-91-49-140:HW2 - Task 1 hg$
[full-165-91-49-140:HW2 - Task 1 hg$
[full-165-91-49-140:HW2 - Task 1 hg$
[full-165-91-49-140:HW2 - Task 1 hg$
[full-165-91-49-140:HW2 - Task 1 hg$ python3 queries.py

```