HW-2A Report

TASK-1

Name	UIN
Venkata Koustubha Krishna Pydimarri	526005194
Ravi Teja Muppala	526003004

Time Tracking :

Total time : 4hrs

No.	Day	Time	Duration/hr	Tasks
1	10/15/2018	1:30pm-2:00pm	30min	Installing MongoDB and setting up Python in the desired path
2	09/22/2018	2:00pm-3:30pm	2hrs	Developing code to execute the queries
3	09/22/2018	3:30pm-4:30pm	1hr 30min	Writing report for Task 1

Why MongoDB is more advantageous than RDBMS for the application ?

For this task, there should be a comparison made to decide which database is better to run an application among MongoDB and RDBMS. Before comparing them, we will know what MongoDB and RDBMS are exactly.

MongoDB is a **NoSQL** database. **NoSQL** stands for “Not Only SQL”. It is an open-source, cross-platform, **document-oriented** database written in C++.

RDBMS stands for Relational Database Management System. It is a collection of programs and capabilities that enable one to create, update, administer and interact with the relational database. SQL is used to access the database.

RDBMS has some advantages as it is supported by an extensive ecosystem of tools and it underpins many legacy applications. However, in this writeup we will show that MongoDB can offer more and better advantages compared to RDBMS.

Advantages of MongoDB over RDBMS :

1. **No need of schema :**

MongoDB is a document-oriented database where each document is considered as a separate object. It is also schema-less. However, in RDBMS, tables, data structure, relations are to be designed (a schema) to run queries. Executing a query without designing the schema will result in an error. In MongoDB, each document can have varied entries differing in certain parameters. In RDBMS, there has to be one particular schema that has to be followed.

2. **Improves Performance & Availability :**

When we use MongoDB, we can add more servers i.e do sharding which increases the performance and availability of the systems. In RDBMS, one has the facility to increase RAM. MongoDB proves beneficial in terms of Performance, Consistency, Availability and Partitioning which are very much essential for an application to be running even when in trouble. Although RDBMS tries to provide these features through ACID(Atomicity, Consistency, Isolation, Durability), MongoDB can compete with the needs of the company in terms of speed of development of new technologies in the market.

3. **Ease of Use :**

MongoDB is easy to setup, configure and run in comparison to RDBMS. MongoDB supports JSON query language along with SQL but RDBMS supports SQL query language only. Using JSON notation makes it sink well with object oriented programming approach where objects are used. There is no need of complex joins in case of MongoDB. MongoDB uses internal memory cache for storing working sets resulting in faster access time. MongoDB is faster than traditional database system like RDBMS, which is slower in comparison with the NoSQL databases.

4. Advanced Features :

MongoDB supports deep query-ability i.e we can perform dynamic queries on documents using the document-based query language that's nearly as powerful as SQL. In MongoDB, mapping of application objects to database objects is not needed.

Having discussed about the advantages of MongoDB over RDBMS, here is the practical demonstration of the benefit that MongoDB offers as it **does not need any schema**.

For this, a table Inventory is being created using both SQL and MongoDB. Now here is the comparison of the performance of both.

MongoDB :

MongoDB does not need any schema. As we can see in the figure below, we are trying to insert an entry into the table with entries like item, qty, tags and size. The row gets inserted. Now in the second query, one more entity named **price** is added. Even then, the query gets executed updating the existing row.

```
>>> db.inventory.insertOne(
...    { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }
... )
{
    "acknowledged" : true,
    "insertedId" : ObjectId("5bc50ff6f80090066288a1b0")
}
>>> db.inventory.insertOne( { item: "canvas", qty: 100, tags: ["cotton"], price : 1000, size: { h: 28, w: 35.5, uom: "cm" } } )
{
    "acknowledged" : true,
    "insertedId" : ObjectId("5bc5100ef80090066288a1b1")
}
>>>
```

Figure-1 : MongoDB Performance

RDBMS (Using SQL) :

First we will try to create a table called inventory with just two entries quantity(qty) and the name of the item as shown in the figure below.

SQL Statement:

```
CREATE TABLE inventory(
qty bigint PRIMARY KEY NOT NULL,
name TEXT)
```

Fig-2: Creating table using SQL

Now we will insert one row with the appropriate quantity and item name with the command in the figure.

SQL Statement:

```
Insert into inventory values (1000,"canvas");
```

Fig-3: Inserting row to table

Here, we will also add the price option by putting number 1000 beside canvas.

SQL Statement:

```
Insert into inventory values (1000,"canvas",1000);
```

Fig-4: Adding additional entry to the command

But this resulted in a SQL error as the command is not following a particular schema in which it should be written.

Error 1: could not prepare statement (1 table inventory has 2 columns but 3 values were supplied)

Hence, it can be fairly concluded that MongoDB proves beneficial when compared to RDBMS as thought by AggieFit team.

Procedure for executing queries :

1. In order to complete the task, installing MongoDB in the local system is necessary.

2. After installing MongoDB, download the files given for Task 1. Then, it is essential to install Python in the path where files are located. Otherwise, code can't get executed.

3. Add path of Python in the "**Environment Variables**" dialog box of the local system.

4. Later, install pip from the command prompt. After installing pip, install **pymongo**. At this point, the initial setup for running the code would be done.

5. The link for our TAMU GitHub Repository is :-

   https://github.tamu.edu/Ravi/HW2.git

6. Run the queries.py file with the command "**python queries.py**". Then, the output would be displayed corresponding to a UIN in the **constants.py** file.

7. The collection from which entries are to be retrieved can be altered in **mongo_connect.py** file.

How data performance and storage features can be improved in MongoDB ?

Performance :

Performance can be improved by altering the database design. The design of the database could be improved by consolidating more of the data into separate groups. One can keep data that belongs to a particular logical category in each of the groups. This will make it easier to understand how the data is used. For instance, the information about height, weight can be categorized under a specific group called "**physical attributes**". The goal types can be categorized under **"goals"** group. The performance improvement could also be achieved through **aggregation** as data from multiple documents could be retrieved using a single query.

Storage :

In MongoDB, data is usually stored in the form of documents in JSON format. Whenever, the dataset is larger, it could be better to place items in an array because they take less disk space when compared to keys and values. This helps in better data storage and retrieval. Nested subdocuments can also be used for data storage. This method improves **performance** by dramatically decreasing the amount of MongoDB storage space needed for the data.