Subject: BUAN 6320.003 – Database Foundations – F17

Assignment: mySQL and MongoDB

Write a 5-page paper to compare mySQL and Mongo with regards to how queries are prepared and executed, and the reasons why. Include a section to compare and contrast the two technologies with regards to how they foster data integrity, performance and security.

Submitted by: Group 35

1. Introduction to MySQL and MongoDB

Database technology comprises two major types, namely SQL (relational databases) or NoSQL (non-relational databases).

The main feature of relational databases is the highly structured nature where the data is stored in multiple tables, in rows and columns. The columns contain the attributes whilst the rows contain the entries. The interconnections among the tables are determined by a schema, which is designed in a manner to keep the tables synchronized and minimize any data redundancy. The process of predefining this relation is termed as 'Data Modelling'. In data modelling, the designer needs to specify the 'Entities' (object in the model that we want to store information about) and populate them with identifiers. After defining the entities, the designer specifies the kind of relation between each of them based on the rules on cardinality. This is a time-consuming process and is not a feasible option for volatile data.

SQL is used to design relational databases (RDBMS) and gets its name from the term 'Structured Query Language'. It is like a script on the server-side end to execute queries to retrieve data or edit data (update, delete or create new records). MySQL is the most widely used open source database on which SQL executes its queries.

The focal attribute of a non-relational databases is the flexibility it offers in terms of speed and scalability. This is achieved by storing the data in a document-oriented way, where all sorts of related data are stored together, without being categorized. It is a schema-less method and is also known as NoSQL (Not Only SQL). Since the data is stored in bulk, this needs more processing power compared to structured data.

MongoDB is the most common open source (free) non-relational database. MongoDB derives its name from the word ‘Mongo’ which means big or huge. It uses JSON (Java Script Object Notation) documents for data storage. An important property of MongoDB is that storing data does not require Object Relational Mapping. Due to its unstructured format of storing data, Mongo is used for manipulating huge amounts of data, and is used widely in the field of Big Data. Companies that need to store large amount of data that have no defined structure, that are moving to cloud based storage (data spread across many servers) prefer MongoDB that is, the NoSQL format due to these features.

2. MySQL vs MongoDB - Query preparation and execution

- Create table or collections – MySQL has a defined schema for the database. Hence each column of a table will have a specific data type and all the rows in the table will adhere to the specific structure. When creating a table, the user should specify the columns and the data types as per the schema. The primary key should also be implicitly stated, and the user should input the values accordingly. In contrast, MongoDB stores files in a JSON format that is in the form of a document which accommodates a variety of structures. This document comprising of data is termed as a ‘Collection’ and it can have its own unique structure with fields containing any data type. The data to be stored in this collection is in the form of arrays. A collection does not necessarily have to be a part of one schema. MongoDB supports a dynamic schema method, making it easier to update data within it. A collection can be created with the first “insertOne()” or “insertMany()” command. The data types of each document need not be pre-specified. MongoDB adds the primary key “_id” automatically if the user does not specify it.
- Alter table queries – MySQL, the table can be altered in a structured form using the “ADD” or “DROP COLUMN” commands. When adding the column, it is vital the data type be defined. In MongoDB, at the collections level, since it comprises of data of any type, it doesn’t require any structured alterations. But at the document level, fields can be added or deleted from the document using the “$set” and the “$unset” operations respectively along with the “updateMany()” function. Further, any amount of data can be stored in any document and it is not necessary for all the documents to follow a single format.
- Reading records – In MySQL, the user has to specify the columns that needs to be retrieved using the “Select” statement in the query. The corresponding MongoDB

statement is “find()”. This method always outputs the object id, whether it is specified in the query or not. If the user wished to exclude it, it needs to specified in the query. This is also known as “projection”.

- Indexing - Both the databases use indexes to find the data in quick time. If however the indexes are not defined, in MySQL, the database engine examines the whole table in order to locate all the applicable rows. In MongoDB, all the documents in a collection are examined to retrieve the documents that are called by the query statement.
- Merging Tables - The greatest benefit of MySQL is its ability to join different tables. This helps querying across different tables. MongoDB does not support JOIN operations but supports multidimensional data types like other documents and arrays. Since the data in MongoDB is de-normalized, merging tables may not be required either.
- Atomicity - MySQL supports atomicity, that is it can query single transactions. The Relational Database Structure supports ACID that stand for Atomicity, Consistency, Isolation and Durability, unlike MongoDB. This helps update a single row within a table of a RDBMS, faster and more efficiently. As the name ACID suggests, MySQL works through isolation of data into atomic structures that can be updated and queried independently. It maintains consistency among the database assigning a specific schema (entity) to each value entered.

3. MySQL vs MongoDB - Performance

- Speed – The data in MySQL is in normal form. Thus, to compile data from various sources, complex queries and JOINs may be required. MongoDB is de-normalized and hence it allows the user to retrieve the relevant information using a single query. Hence, it may be faster in executing queries.
- Scalability – MySQL has challenges in scaling as data is stored in specific tables in defined formats. MongoDB, on the other hand, supports “auto-sharding”. This implies that scaling up can be done across multiple distributed data centers as data volume grows with no impact on the application or any downtime.
- Redundancy – In MySQL, data is normalized. This means the same data is not duplicated across many tables. Instead, a parent table with the information is created and a reference (foreign key) is drawn from the parent to the other table (child). Hence,

any updates to the parent table can be done without disruptions to other tables. However, in MongoDB, documents are de-normalized. Hence redundancy exists.

- Availability of resources: - As compared to MySQL, MongoDB is relatively new and hence the community is not as strong as MySQL. There are more open source codes and ideas shared for MySQL as compared to MongoDB. There are no reporting tools for MongoDB like MySQL which makes it difficult to analyze the validity of the application in concern.

4. MySQL vs MongoDB - Data integrity

The accuracy, reliability and consistency of data stored in a database is referred to as “Data Integrity” which goes hand in hand with “Data Quality”. Data are said to be of high quality if they fit their intended use and performance is high. Data must also be guarded from unauthorized access, modification. To ensure data integrity and security database security professionals employ various measures like Integrity constraints, encryption, backup, access control, validation of data etc. Integrity constraints defined during data entry make certain that the data conforms to the corresponding rules.

MySQL: Data integrity in MySQL is enforced using various constraints. It consists of the following types:

- Row integrity: Primary key ensures that all rows have a unique identifier. Such an identifier ensures that the different rows in a particular column are assigned unique values.
- Column Integrity: All the data in a column have the same format, structure and definition. This constraint defines the data type, length and other attributes specific to the column
- Referential Integrity: Foreign keys ensure data integrity when updating data by creating relationship between tables.
- User-Defined Integrity: Specific business requirements which cannot be defined by the above integrity constraints are taken care of by User-Defined integrity constraints. This defines the user-defined constraints for data in a column.

MongoDB:

- At the document level MongoDB write operations are ACID (Atomicity, Consistency, Isolation and Durability) compliant – ensuring isolation as a document is modified during a transaction
- No built-in features to ensure consistency in MongoDB. The onus is on the engineer to not write inconsistent data to the database. It's hard to guarantee data integrity in MongoDB because of lack of server-side foreign keys relationships.

5. MySQL vs MongoDB - Security

MySQL – MySQL's high security standards for data comes from its privilege system. The primary function of the MySQL's privilege system is to authenticate a user and to associate that user with privileges on a particular database. The database is password protected and hence has a good authentication mechanism. The MySQL privilege system ensures that all users may perform only the operations permitted to them. Thus, authorization is controlled, giving access to the relevant users and databases only as required.

MongoDB - MongoDB uses a role-based access control with flexible set of privileges. The lack of encryption support for data at rest and weak authentication at database level may result to great security dangers. As a means to compensate for its weakness in security at database level, MongoDB creates copies of databases and make them available on multiple servers. Also, it can use TLS/SSL (Transport Layer Security/Secure Sockets Layer) to encrypt all of MongoDB's network traffic to ensure that it is only readable by the intended client.