Database Comparison

Anamika Chatterjee (202004102)
Hemanth chekka (202006238), Ruthvik Jaini (202002268)
October 12, 2023

Abstract

Over the years the technology to store data has been developing. At first the evolution data storage technology led us to have relational databases which were widely used in all fields. However, further evolution and research has introduced many new forms of storage technologies and nowadays one of the most prominent of them is non-relational databases. This survey is conducted to achieve complete understanding of the differences between relational and non-relational databases and will provide a quantitative and qualitative analysis to highlight those differences.

1 Introduction

The aim of this project is to compare the MySQL relational database and No-SQL databases like Cassandra and MongoDB. It is important to know the differences between the three databases so that one can make informed decision while choosing database for an application.

1.1 SQL Databases

SQL (Structured Query Language) is based on tuple calculus and relational algebra. Columns and tables are used to store data. Databases have several qualities like atomicity, consistency, isolation, and durability:

- **Atomicity** - It aids in comprehending that transactions must be completed effectively or they should be rolled back totally.
- **Consistency** – We may argue that in terms of consistency. Before and after a transaction, the database should be consistent.
- **Isolation** - Each transaction should be treated as a distinct entity, with no overlap between two transactions. We comprehend solitude in this way.

- **Durability** – The database is also long-lasting since all transaction modifications are permanent.

1.2 No-SQL Databases

No-SQL databases are non-relational because they do not follow the relational model. They are schema-less because they are column records without tables. As a result, data movement is simple. Document stores, graph databases, broad columns, and database key value stores are among the SQL database models that they are familiar with. No-SQL databases follows the CAP theorem (Consistency, Availability, Partition Tolerance)

1.3 CAP Theorem

[1] It states that a distributed store cannot provide more than two of the three desirable features (Example: cp or ap):

Consistency: It means that no matter which node a client connects to, they all see the same data at the same time.

Availability: This term refers to the fact that something is available. Even if one or more nodes are offline, any client making a data request receives a response.

Partition tolerance: A communication is a division. A lost or temporally delayed connection between two nodes in a distributed system is known as a break. It indicates that, despite this, the cluster must continue to function. Any number of communications are possible.

2 Advantages of SQL Database

[7] Some of the advantages of SQL database are:

- It is quite interactive and there are multiple data views
- No major coding skills required
- It is a standardized language.
- Faster query processing
- It is portable

3 Advantages of No-SQL

Advantages of No-SQL include:

- Can store unstructured, semi-structured and structured data
- Highly Scalable and available
- High flexibility
- Open Source

4 Graph Database

Further navigation through relationships. We use graph databases. We can also store the entities in nodes and the relations on the edges. It can be used for social media, recommendation engines and fraud detection.

5 Key-Value Store

[6] The key-value store is a type of non-relational database which stores the data in a simple key value method. The data is fetched by a unique key to retrieve the values. Each key has a value associated with it and all the keys are unique. So, whenever the key is called the associated value is retrieved.

6 Comparison of Databases

6.1 My-SQL

MySQL is an open-source relational database management system. Over the years MySQL has become popular since it is open source and the ease of use it gives while working with java.

6.1.1 Architecture of MySQL

The essential components of MySQL are as follows:

Connectors - Connectors allow us to talk to the database when we are making an application using Java, Python, C, C++, etc.

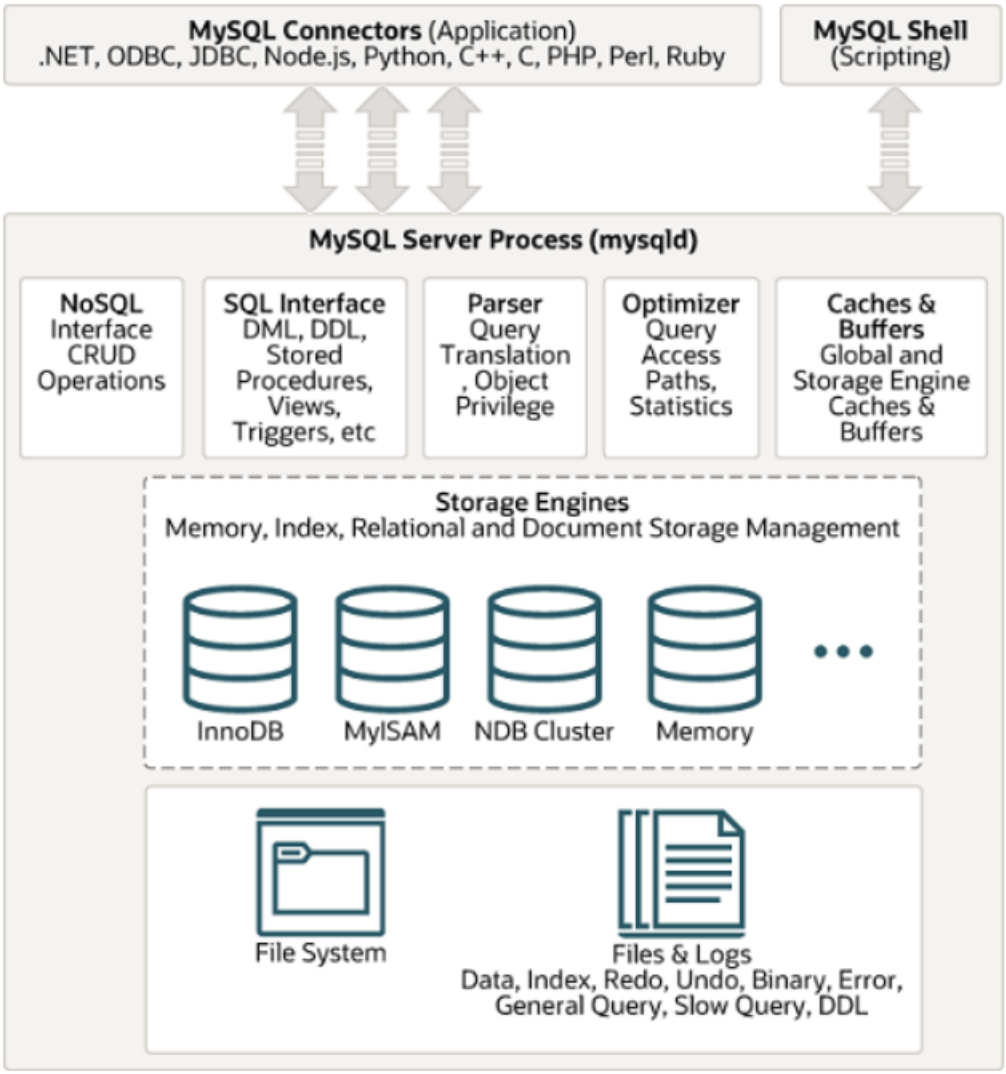

Figure 1: Architecture of MySQL [2]

Client – MYSQL workbench is a form of client. It has mysql shell with a GUI.

Connections- For every request an application makes to the server, one thread will be made and the response will be sent back.

Query Cache - Some form of cache will be saved for frequently made queries. This is done so that database doesn't have to make the same query again and again.

Parser – Parser understands the query and converts it into system understandable language.

Optimizer – Optimizer's job is to optimize the query. It tries to shorten the query. It uses relation algebra to optimize.

Storage Engine - My SQL gives the option to the developer regarding how the data should be stored. A day to a developer can go with the default storage style. Or can store according to their needs. This is actually responsible for all the I/O operations for the application. By catering the needs of our application we can reduce the overhead on the database.

File system level storage - If we break down the database concept, we are just storing the data which we want in the form Of B trees or B plus trees. We do that so that we can retrieve the data faster. So, all the data is stored in the file system for faster retrieval through database.

6.1.2 Backup and recovery in MySQL

Following are the backup and recovery options in MySQL:

Backup and recovery in MySQLI - It is important to back up our databases so that we can recover our data and be up and running again in case problems occur, such as system crashes, hardware failures, or users deleting data by mistake [4].

Backup and Recovery Types - There are two types of backups. One is physical backup and other is logical backup. In physical backup we are copying the database file. In this we are copying everything. Everything includes tables and records present in them and all triggers etc. In logical backup we are running the script so that we retain all the tables. In physical backup we will have all the data, but in logical backup we are just making the tables again without the data.

Database Backup Methods - We can choose database backups as in we can choose what all we want to backup. We can either choose to backup the whole database or we can choose only few tables and backup them.

Using mysqldump for Backups - One of the database backup command is by using **mysqldump**. With this we can dump the data file. This comes under logical database backup method.This will backup With multiple threads and will provide better file compression. It dumps one or more MySQL databases for backup or transfer to another SQL server. This command can generate output in CSV, other delimited text, or XML format.

Point-in-Time (Incremental) Recovery - With Point-in-time recovery we can do re-

covery of data changes up to a given point in time. Typically, this type of recovery is performed after restoring a full backup that brings the server to its state as of the time the backup was made. Point-in-time recovery then brings the server up to date incrementally from the time of the full backup to a more recent time. It will have binary logs. It will have system images with the time stamps.

6.1.3 Security in MySQL

General Security Issues - From the perspective of a DBA they must keep passwords secure. Identify unrecognized requests to the database or in the network. They also must be aware there are any SQL injections possible. DBA is responsible for access controls. They are responsible for giving restricted access to people [3].

Using Encrypted Connections - When you must move information over a network in a secure fashion, an unencrypted connection is unacceptable. To make any kind of data unreadable, use encryption. Encryption algorithms must include security elements to resist many kinds of known attacks such as changing the order of encrypted messages or replaying data twice. Unencrypted connection between the MySQL client and the server ca be dangerous because someone with access to the network can watch all the traffic and inspect the data being sent or received between client and server. Hence unencrypted connections are unacceptable.To make any kind of data unreadable, use encryption. Encryption algorithms must include security elements to resist many kinds of known attacks such as changing the order of encrypted messages or replaying data twice. MySQL supports encrypted connections between clients and the server using the TLS (Transport Layer Security) protocol. TLS is sometimes referred to as SSL (Secure Sockets Layer) but MySQL does not actually use the SSL protocol for encrypted connections because its encryption is weak.

Security Components and Plugins –Plugins for authenticating attempts by clients to connect to MySQL Server. Plugins are available for several authentication protocols. These can mean OTP verification for DBA while entering into database.The **validate_password** component can be used to improve security by requiring account passwords and enabling strength testing of potential passwords. This component exposes system variables that enable you to configure password policy, and status variables for component monitoring.MySQL Server supports a keyring that enables internal server components and plugins to securely store sensitive information for later retrieval. Keyring components and plugins that manage a backing store or communicate with a storage back end. Keyring use involves installing one from among the available com-

ponents and plugins. Keyring components and plugins both manage keyring data but are configured differently and have operational differences.

MySQL Enterprise Data Masking and De-Identification – MySQL also takes care of sensitive information.In case there is any sensitive data we can store it in unreadable format. It hides the data even from developers. Transformation of existing data to mask it and remove identifying characteristics, such as changing all digits of a credit card number but the last four to 'X' characters.

MySQL Enterprise Encryption - MySQL Enterprise Edition includes a set of encryption functions. This will give added data protection using public-key asymmetric cryptography. We create public and private keys and digital signatures. MySQL allows us to perform asymmetric encryption and decryption. It also gives cryptographic hashing for digital signing and data verification and validation. MySQL Enterprise Encryption supports the RSA, DSA, and DH cryptographic algorithms. MySQL Enterprise Encryption is supplied as a library of loadable functions, from which individual functions can be installed individually.

6.1.4 Advantages of MySQL

Supports all major platforms - It can run on any platform like windows, Linux, Unix. But others like SQL server is in developing stages for Linux OS.The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (API's). MySQL also provides MySQL Server as an embedded multi-threaded library.

Easy to administer / Manage – SQL Workbench is a very handy GUI tool.It is made to make the life of a developer very easy. It comes with very easy to get around GUI which can be even used by a person who is in earlier stages of using a database tool.

Open source – It is free to purchase. Open Source also means that we can modify the software according to out needs. We can also go through the source code of the MySQL and change it.

ACID Compliant - For any transaction MySQL will follow the ACID properties. Atomicity, Consistency, Isolation and Durability.

Multiple HA technique – MySQL has High Availability features.It has the option

of adding backup servers. Disaster recovery in MySQL is very efficient as we have have seen in Backup and Recovery techniques. My SQL also supports replication from another server. If primary goes down backup server will support in the time of disaster.

6.1.5 Disadvantages of MySQL

There are fewer features in MySQL. MySQL holds some features back for its paid Enterprise Version eg. using a cluster database behind a paywall: MySQL-Enterprise Version. One of the advantage of MySQL is that if we try to make a text search for a huge amount of data it cannot search efficiently thought it has parallel processing. People are nowadays trying to mine data a lot. MySQL is not made for data analytics. MySQL is not made for running a lot of complex queries on huge amount of data.

6.1.6 CRUD operations in MySQL

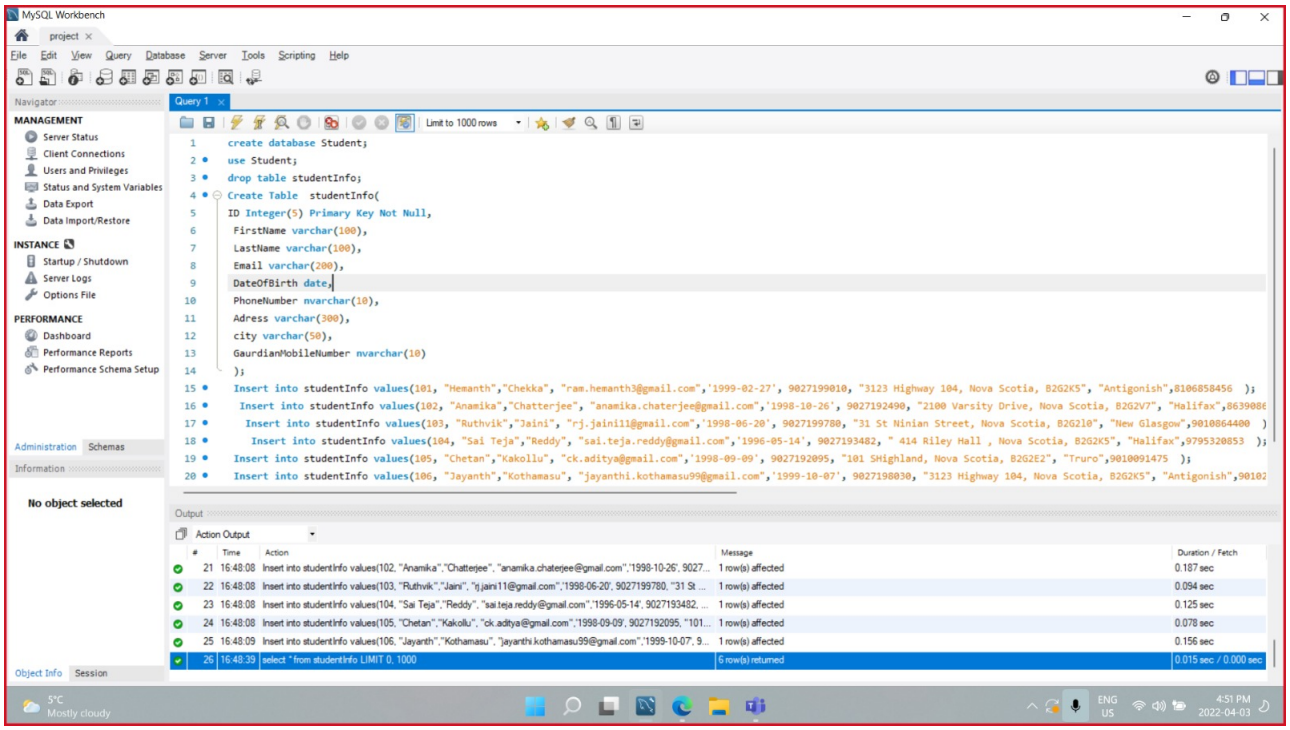

Figure 2: Create operation in MySQL

The screenshot shows the Create operation performed in MySQL in MySQL Work Bench. Here we have created a table studentinfo. Here 'ID' is the primary key. We have added 6 records using insert operation in MySQL.

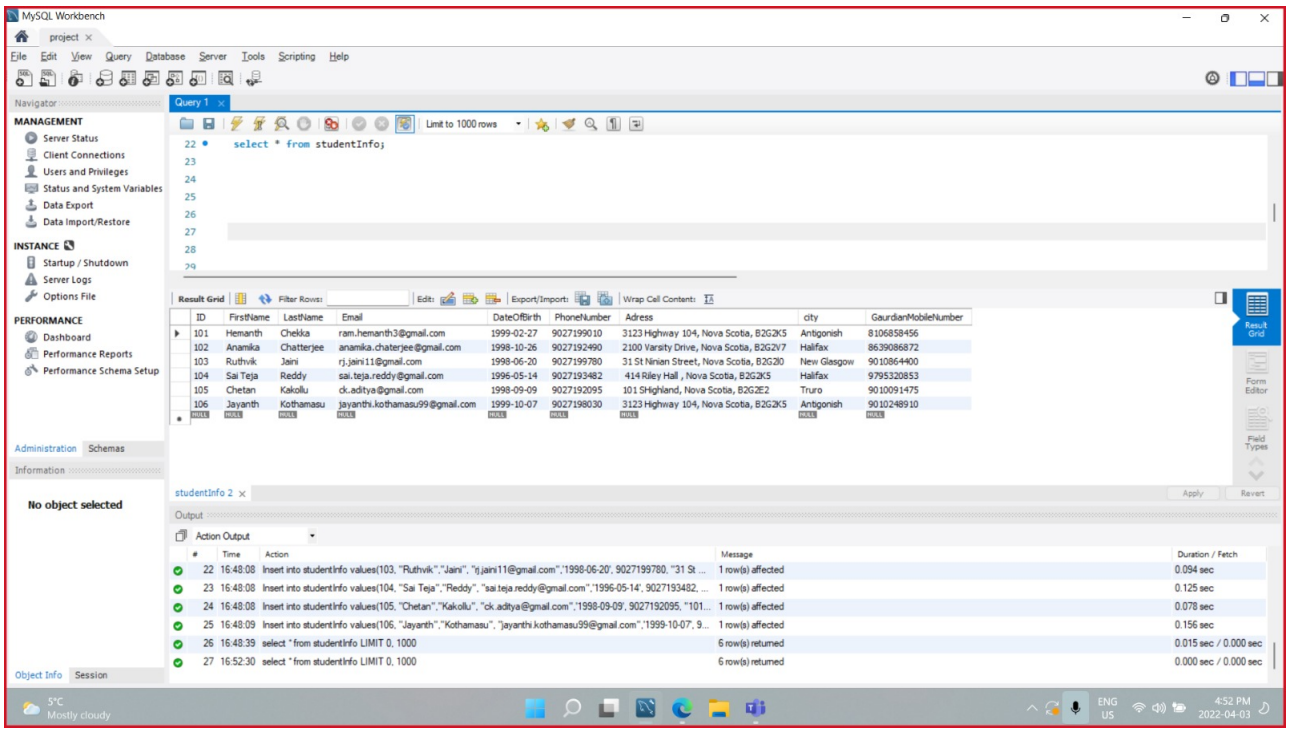

Figure 3: Read operation in MySQL

This screenshot illustrates read operations in MySQL. 6 records are read from the table. Every valid query in relational database should return another relation. Using Select * query we got the whole table back.

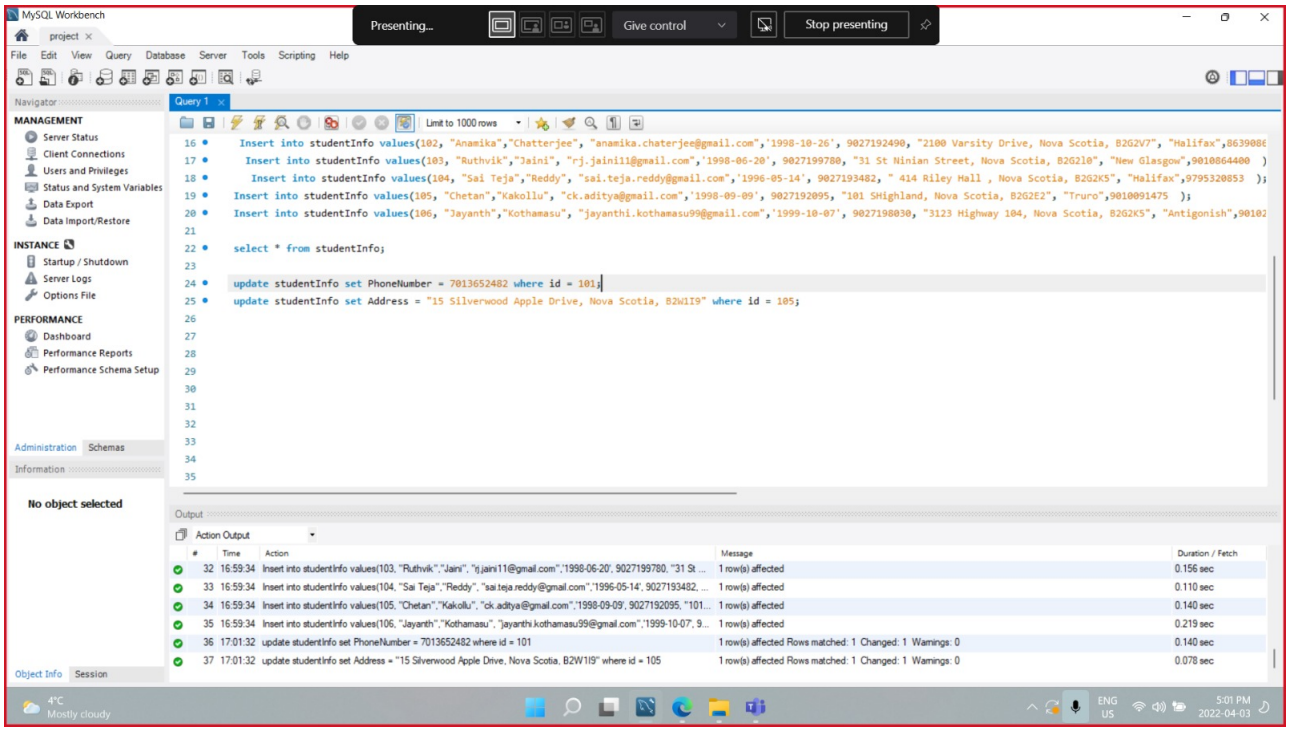

Figure 4: Update operation in MySQL

In this screenshot we can observe that two records are updated. We observe the changes in the records once we read the records.

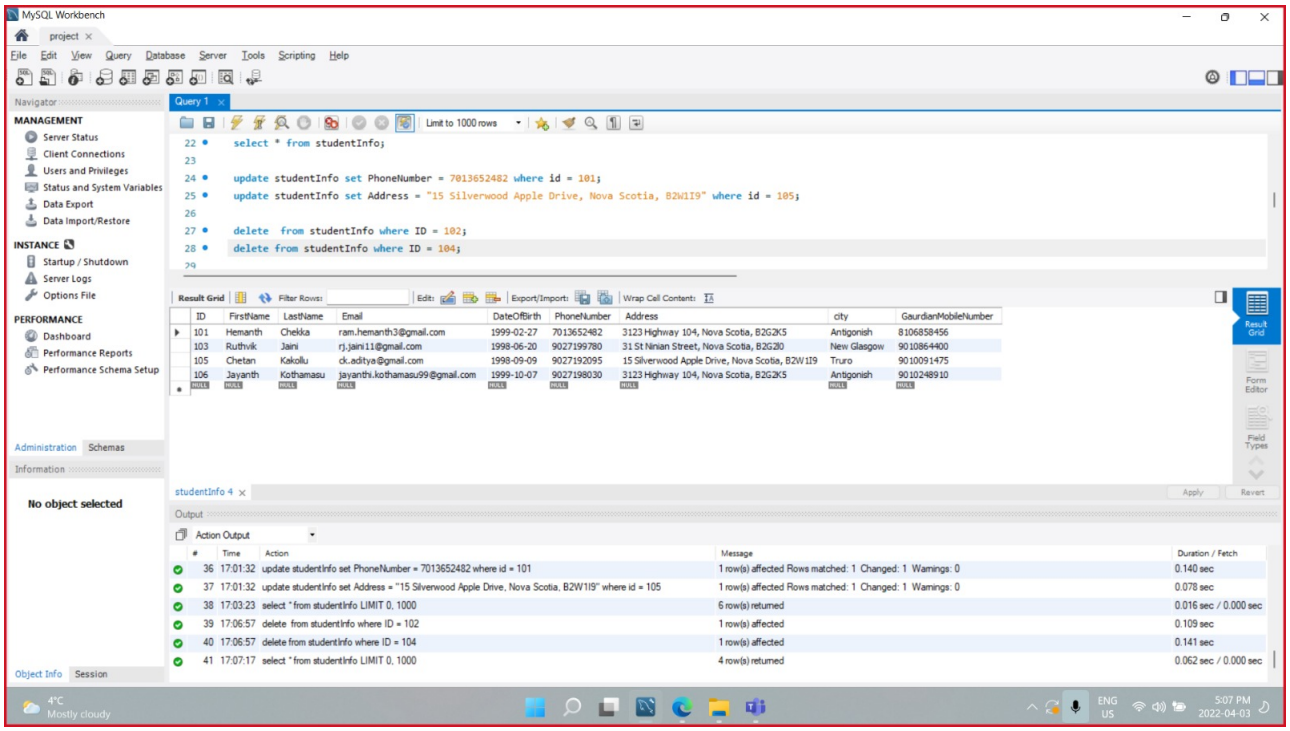

Figure 5: Delete operation in MySQL

The delete operation is shown in this screenshot. We can see that two of the records(rows) have been deleted and there are only 4 records left. For Delete operation too we need to mention primary key value of the records we want to delete. We can either choose primary key to delete a single row or we can give a condition in the delete query by which we can delete all the rows which satisfy that query.

6.2 Apache Cassandra

Apache Cassandra is a wide-column store non-relational distributed database. It was developed at Facebook first. It was developed for the inbox search feature of Facebook. Later Facebook released it as an open source project later. Cassandra is also highly scalable because it has a peer-to-peer distribution model. To understand Cassandra fully we need to understand the architecture and the protocols and mechanisms it uses. Cassandra focuses on Availability and Partition Tolerance.

6.2.1 Cassandra Architecture

It's quite easy to understand the architecture of Cassandra. Since Cassandra is supposed to be used for systems that are distributed over physically separate location therefore, Cassandra architecture consists of clusters which contain nodes and nodes contain keyspaces. We can think of keyspaces as databases. The nodes in Cassandra can be in different locations physically but will contain the same data if they are the replicated nodes. Keyspaces consist something called column-families. Column-family can be compared to a table in relational database however column-families are stored in seperate files on the disk. To understand column-families we need to understand how wide-column store works.

6.2.2 Keyspace

A keyspace in Cassandra is nothing but a container of column families. It can be compared to databases in RDBMS. In Cassandra keyspaces have a set of attributes that decide keyspace wide behavior. The set of attributes are:

Replication factor: This attributes allows us to decide the number of nodes that will act as copies of each row of data. For instance, if we set replication factor to 3 then three nodes in the cluster will have copies of each row.

Replication placement strategy: This attribute allows you to decide how the replicas will be placed in the ring. There are different strategies that Cassandra provides for determining which node will get copies of which keys.

Keyspaces also let us create column families and set primary keys for the each of those column families. Cassandra also lets us set something called a partition key. Partition key works the same way as a group-by function in RDBMS however we don't need to write a new line of query for it.

6.2.3 Wide-Column Store

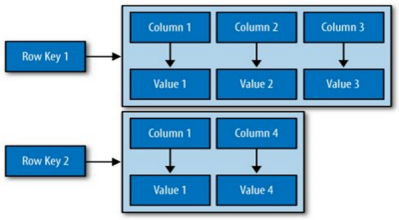

Figure 6: Wide-Column Store

Similar to relational databases wide-column store databases have tables, rows and columns in which data is stored. However, unlike relational databases wide column stores don't have fixed column names and format which essentially makes it schema free. This means that the list of columns can vary from row to row and every row doesn't have to include all the column values.Each row can have one or more columns but it does not need to have the values for all the columns. Therefore, instead of storing null for columns we don't know the values for we don't store that column for that row saving disk space.

The figure above illustrates how a wide column store works, columns are nothing but a bunch of name-value pairs. The row-keys just help group bunch of columns for a single instance of an entity.So, a row-key with name-value pairs represent a single row. Row-keys can be compared to primary keys in RDBMS. Therefore, we can see the wide-column store as key-value store too.For example, if entity is Employee which will be the "table-name" then the row-key will the employee-id and columns can be Name and Salary with individual values.

6.2.4 Peer-to-Peer Distribution

Unlike RDBMS Cassandra has a peer to peer distribution model which supports high scalability. In such a model the nodes do not follow a master and slave roles. So all the nodes get changes written in them at the same time. This helps decentralizing the system. A decentralized system lowers the risk for system failures because it does not depend on a master node. In master-slave model all the all writes directly go to the master node and slave nodes synchronize their data. This means that if the master node fails due to network failure or hardware problems, then all the other nodes do not get the recent written data. Therefore Cassandra distribution model also support partition tolerance and high availability because if one node fails the data is available on other nodes this way it doesn't lead to a whole system failure [10].

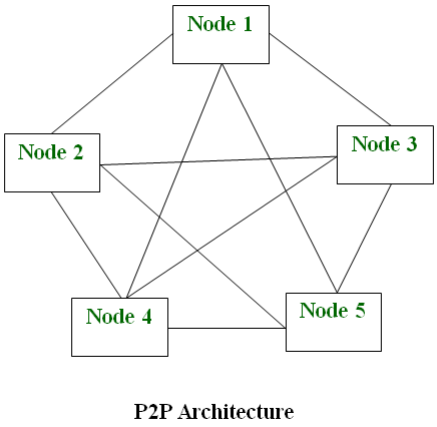

Figure 7: Peer-to-Peer Distribution

6.2.5 Gossip Protocol

Gossip protocol is a failure detection protocol which Cassandra uses to further support partition tolerance. In this protocol nodes communicate which each other to know their status information. The gossiper node will send the first message to the other node and wait for an acknowledgement. If the other node is not available then the gossiper will not get a reply which will inform the gossiper that the node is down. After some time the gossiper can send another message and if it gets a reply back then a it can trigger the process of hinted handoff which is data synchronization method. The given figure

Figure 8: Three-way communication in Gossip protocol

describes the gossip messages exchanged between two nodes when both the nodes are up and running after the last message is sent the hinted handoff is triggered.

6.2.6 Anti Entropy Read-Repair

Anti entropy read repair is also a data synchronization mechanism however it is manual unlike hinted handoff which is triggered by the gossip protocol. In this two nodes compare their "Merkle trees". Merkle tree is nothing but hash of the column family. A node initiated the conversation between two nodes and the Merkle trees are compared if any data is missing or not the same in both trees then they are repaired and the recent

version of data is updated. This too supports partition tolerance and high scalability.

6.2.7 CRUD operations in Cassandra

Create

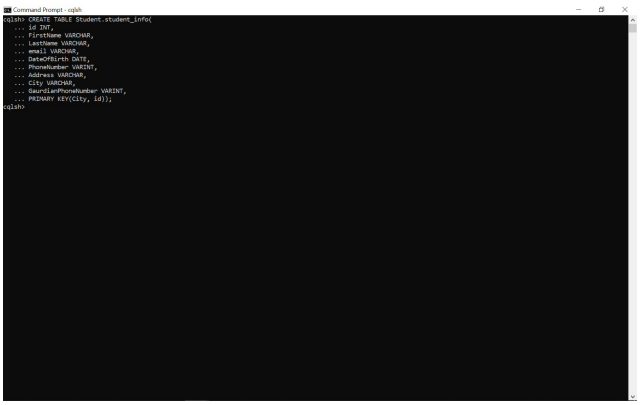

Figure 9: Create operation in CQL

The screenshot shows the Create operation perform in Cassandra in CQL(Cassandra Query Language). Here we have just created a table/column family named studentinfo in keyspace called Student. Here 'id' is the primary key and city is the partition key hence student records will be grouped by city names.

Read

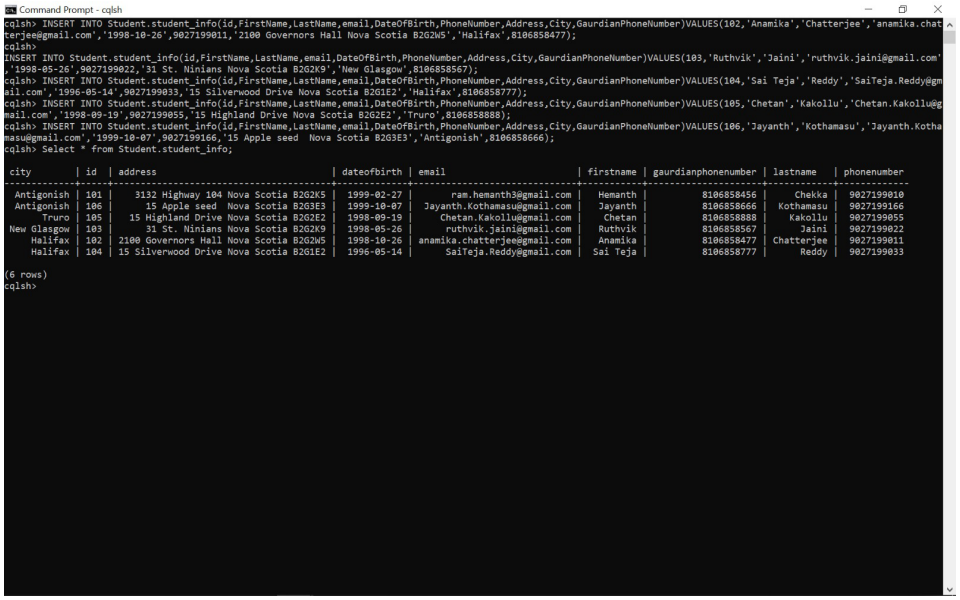

Figure 10: Insert and read operations in CQL

This screenshot illustrates the insert and read operations in Cassandra. 6 records are inserted and 6 records are read. It is observed that the records are grouped by the

partition key which is the city name and ordered alphabetically according to city names. The column are ordered alphabetically too unlike RDBMS.

Update

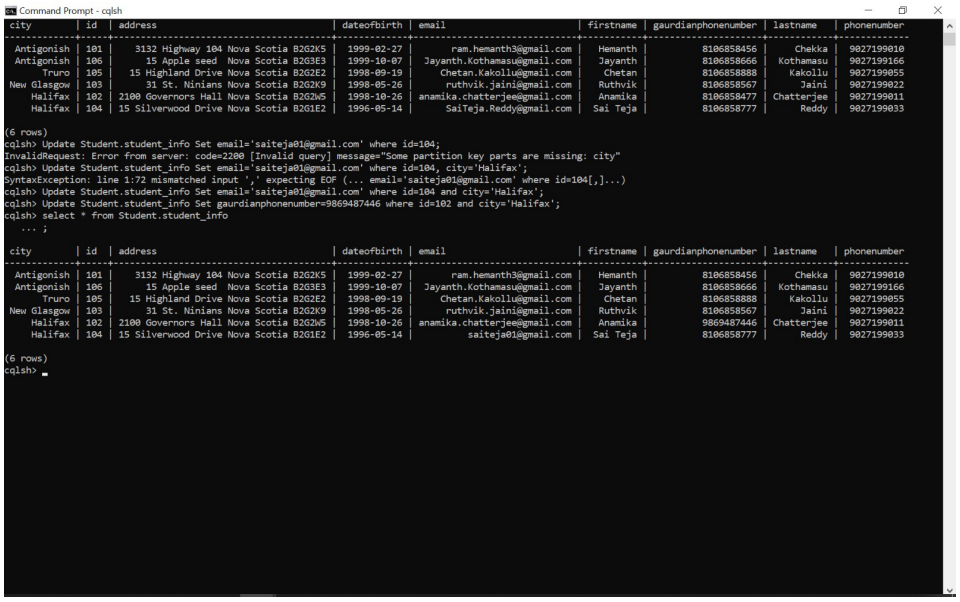

Figure 11: Update operation in CQL

In this screenshot we can observe that two records are updated. We observe the changes in the records once we read the records. The query for update is almost same as that of SQL however here we need to mention both the primary and partition key value of the records that we want to update.

Delete

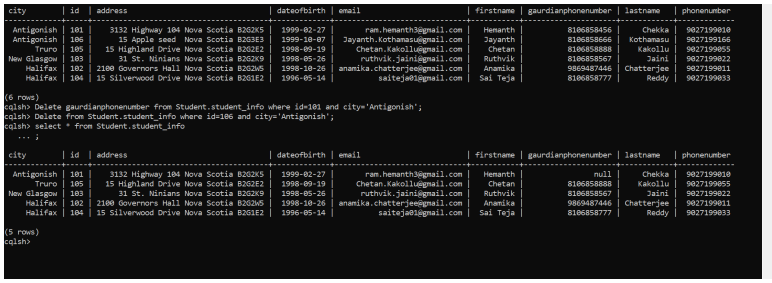

Figure 12: Delete operation in CQL

The delete operation is shown in this screenshot. We can see that one of the records(rows) have been deleted and there are only 5 records left. For Delete operation too we need to mention both primary and partition key values of the records we want to delete.

6.2.8 Advantages and Disadvantages of Cassandra

To summarise the concepts of Cassandra let us look at some advantages and disadvantages of using Cassandra DB

Advantages

- Cassandra is schema free: Since it is wide-column store it can be schema free which means we do not need to describe a fixed schema during creation.
- Highly Scalable: Due to peer-to-peer distribution it is scalable.
- Highly Available and Partition Tolerant: Due to mechanisms and protocols like Gossip and Anti Entropy read repair.
- It is open source.
- CQL: It uses Cassandra Query Language which really similar to SQL so there is no need to learn a new language.

Disadvantages

- Eventual Consistency: Since Cassandra focuses more on making data available and being partition tolerant therefore the consistency of data might suffer. Cassandra is considered an eventually consistent database.
- Since the column families and data is model based on queries a lot of the data might get repeated which might lead to redundancy and inconsistency.
- Cassandra documentation seems a bit lacking when compared to MYSQL

6.3 MongoDB

MongoDB is one of the leading no-sql database which is open-source that is written in C++. It is also known as performance-oriented database that is built for speed. The queries of MongoDB are easily readable as it is document based. This document-based data architecture makes it easier to process complicated, unstructured data without the need of creation of tables which are common in relational databases. When compared to other database systems, MongoDB is simple to use for its ability to store any form of data as the MongoDB data representation technique is pretty straightforward as it can store various forms of data which can be either structured or unstructured. The standard database system's concept of rows has been altered into a MongoDB document that is very dynamic, schema-free and conceivable to manipulate using a single record.

6.3.1 Data Model

MongoDB works on concept of collections and documents [5]:

Database: A database is a physical storage facility for data. On the file system, each database has its own collection of files. Multiple databases can usually be found on a single MongoDB server.

Collection: The term "collection" refers to a group of MongoDB documents. It's the same thing as an RDBMS table. Within a single database, there is a collection. A schema is not enforced by collections. Distinct fields can be found in different documents within a collection. In most cases, all of the documents in a collection serve the same or comparable purposes.

Document: A document consists of a collection of key-value pairs. The schema of documents is dynamic. Documents in the same collection don't have to have the same set of fields or structure, and common fields in a collection's documents can contain different types of data.

6.3.2 Terminologies comparison

The following table shows the relationship of RDBMS terminology with MongoDB [5]:

RDBMS	**MongoDB**
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded Documents

6.3.3 CRUD operations in MongoDB

Create

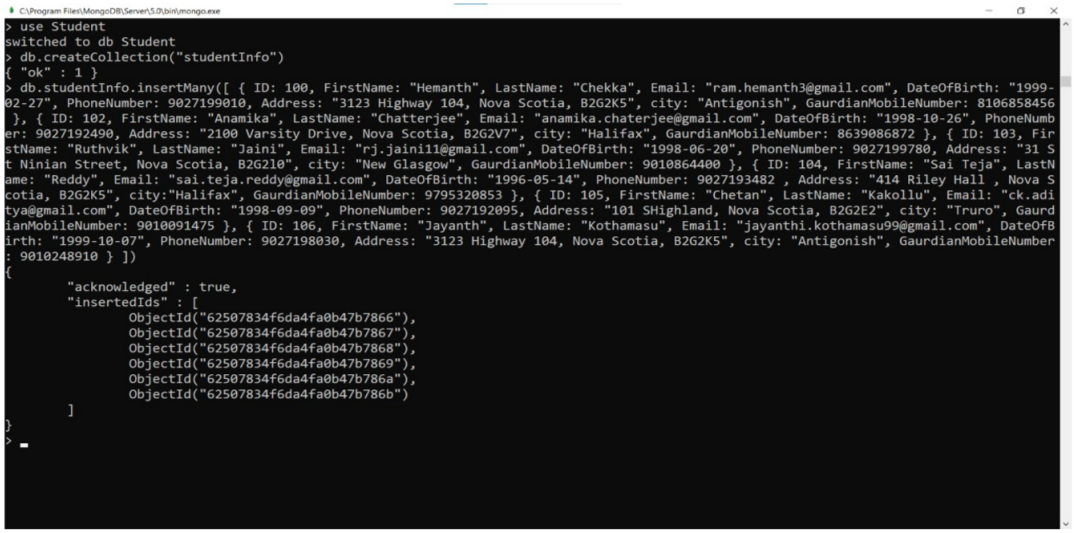

Figure 13: Create operation in MongoDB

The above figure shows the creation of database named "Student" followed by creating collection named "studentInfo" in the Student database to store Student details in the form of documents.

Read

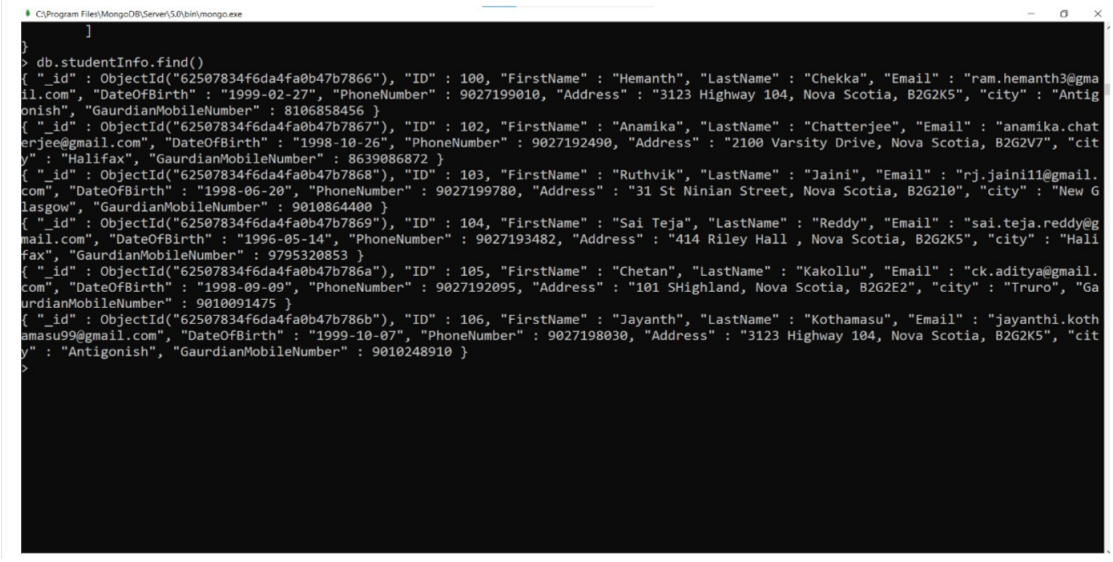

Figure 14: Read operation in MongoDB

This screenshot shows data retrieve operation in MongoDB where 6 records are read which are inserted in the above operation.

Update

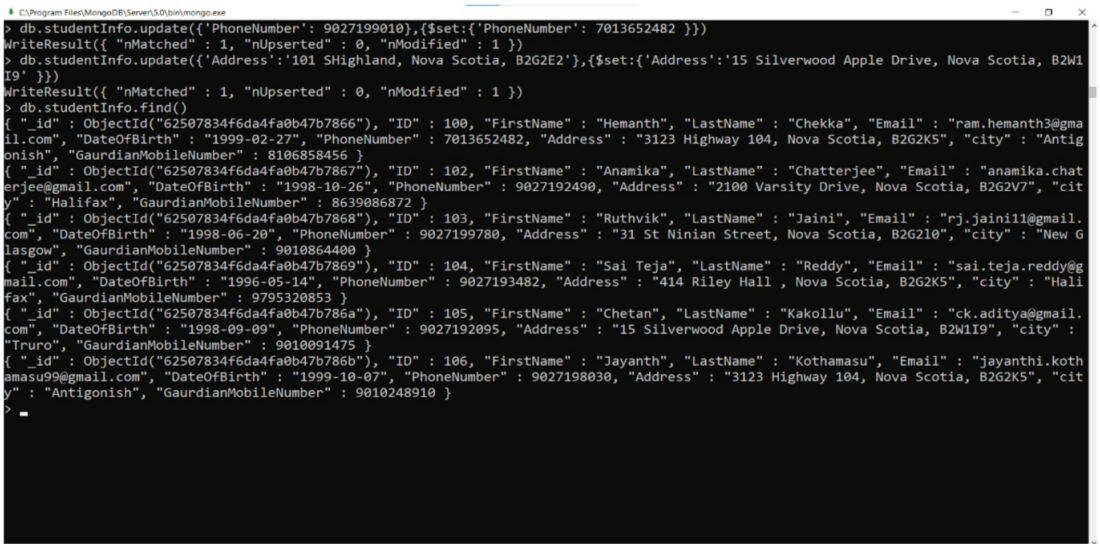

Figure 15: Update operation in MongoDB

In the above screenshot we can observe that two records are updated. We observe the changes in the records once we read the records.

Delete

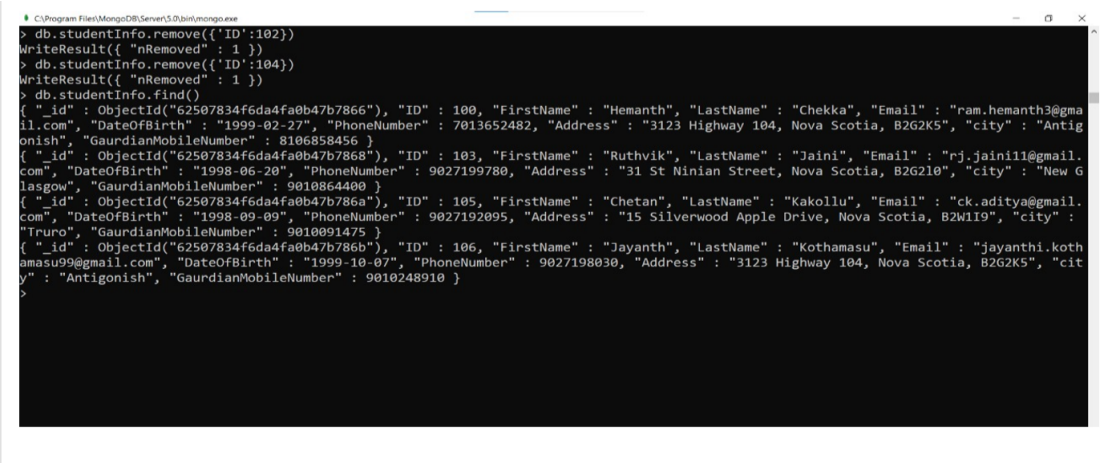

Figure 16: Delete operation in Mongodb

The delete operation is shown in this screenshot. We can see that two of the records have been deleted and there are only 4 records left.

6.3.4 Advantages of MongoDB

- Scalability: MongoDB has the ability to scale horizontally. Sharding of data over many servers is strongly related to scaling in MongoDB. Many databases employ sharding to spread data across connected servers and to encourage dynamic failover. Singular shards are made up of a copy set that consists of at least two nodes. MongoDB employs auto-sharding strategies to ensure that there is no single point of failure. MongoDB distributes read and write workloads across the cluster's shards, allowing each shard to handle a portion of the sharded cluster's activities [8].

- Availability: Data availability is supported by MongoDB through data replication procedures, which are always done in a master-slave setup, just like in traditional relational database management systems. MongoDB uses a replica set topology to manage database replication. For redundancy and dynamic availability of the replica data on the server, the data is replicated across all nodes in the network [8].

- MongoDB is a document database that stores several documents in a single collection. The number of fields, content, and size of a document can vary from one to the next.

- A typical schema design for a relational database displays the number of tables and the relationships between them but in MongoDB there is no concept of relationships.

- The ability to query deeply. MongoDB uses a document-based query language that is nearly as strong as SQL to support dynamic queries on documents.

- There are no complicated joins.

7 Conclusion

Things to consider when choosing between SQL and NO-SQL:

- Is your data structured or unstructured? We need to consider what kind of data will be stored in the database i.e., whether the data will structured, unstructured or semi-structured.
- How often do you need query the data? When, who and how will the data be queried since we all know if it is structured data the querying with SQL is the most efficient and easy. Although non-relational database provide flexibility in storing different kind of data querying might be far more complex than in SQL
- How much and what kind of scalability do you require? Finally, we need to considered how large the system is going to be and what kind of scaling would be most appropriate for the system [9].

References

[1] Cap-theorem https://www.ibm.com/cloud/learn/cap-theorem.

[2] Mysql 8.0 reference manual :: 16.11 overview of mysql storage engine architecture https://dev.mysql.com/doc/refman/8.0/en/pluggable-storage-overview.html.

[3] Mysql 8.0 reference manual ::chapter 6 security https://dev.mysql.com/doc/refman/8.0/en/security.html.

[4] Mysql 8.0 reference manual ::chapter 7 backup and recovery https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html.

[5] Tutorialspoint::mongodb tutorial https://www.tutorialspoint.com/mongodb/index.htm.

[6] What is a key-value database? https://www.mongodb.com/databases/key-value-database.

[7] Advantages and disadvantages of sql https://www.geeksforgeeks.org/advantages-and-disadvantages-of-sql/, Jul 2021.

[8] Wondwessen Haile Addal. *A Comparative Analysis of Relational and Non-relational Databases for web application*. PhD thesis, Near East University,2019, 2019.

[9] Mike Chan and 21. Sql vs. nosql - what's the best option for your database needs? https://www.thorntech.com/sql-vs-nosql/, Jan 2022.

[10] Eben Hewitt. *Cassandra: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2010.