Software-Lab IV
Assignment-2 MongoDB

Submitted by:- **Hasan Koser**
Roll no:- **BT19CSE036**

Short Theory on the given Topic

MongoDB details and its main features and characteristics.

MongoDB is a scalable, flexible NoSQL document database platform designed to overcome the relational databases approach and the limitations of other NoSQL solutions. MongoDB is well known for its horizontal scaling and load balancing capabilities, which have given application developers an unprecedented level of flexibility and scalability.

MongoDB's few top technical features:

1. Ad-hoc queries for optimized, real-time analytics

When designing the schema of a database, it is impossible to know in advance all the queries that will be performed by end-users. An ad hoc query is a short-lived command whose value depends on a variable. Each time an ad hoc query is executed, the result may be different, depending on the variables in question.

Optimizing the way in which ad-hoc queries are handled can make a significant difference at scale, when thousands to millions of variables may need to be considered.

2. Indexing appropriately for better query executions

In our experience, the number one issue that many technical support teams fail to address with their users is indexing. Done right, indexes are intended to improve search speed and performance. A failure to properly define appropriate indices can and usually will lead to a myriad of accessibility issues, such as problems with query execution and load balancing.

3. Replication for better data availability and stability

When your data only resides in a single database, it is exposed to multiple potential points of failure, such as a server crash, service interruptions, or even good old

hardware failure. Any of these events would make accessing your data nearly impossible.

Replication allows you to sidestep these vulnerabilities by deploying multiple servers for disaster recovery and backup. Horizontal scaling across multiple servers that house the same data (or shards of that same data) means greatly increased data availability and stability. Naturally, replication also helps with load balancing. When multiple users access the same data, the load can be distributed evenly across servers.

4. Sharding
When dealing with particularly large datasets, sharding—the process of splitting larger datasets across multiple distributed collections, or “shards”—helps the database distribute and better execute what might otherwise be problematic and cumbersome queries. Without sharding, scaling a growing web application with millions of daily users is nearly impossible.

5. Load balancing
Fortunately, via horizontal scaling features like replication and sharding, MongoDB supports large-scale load balancing. The platform can handle multiple concurrent read and write requests for the same data with best-in-class concurrency control and locking protocols that ensure data consistency. There’s no need to add an external load balancer—MongoDB ensures that each and every user has a consistent view and quality experience with the data they need to access

Queries supported by Screenshots along with questions

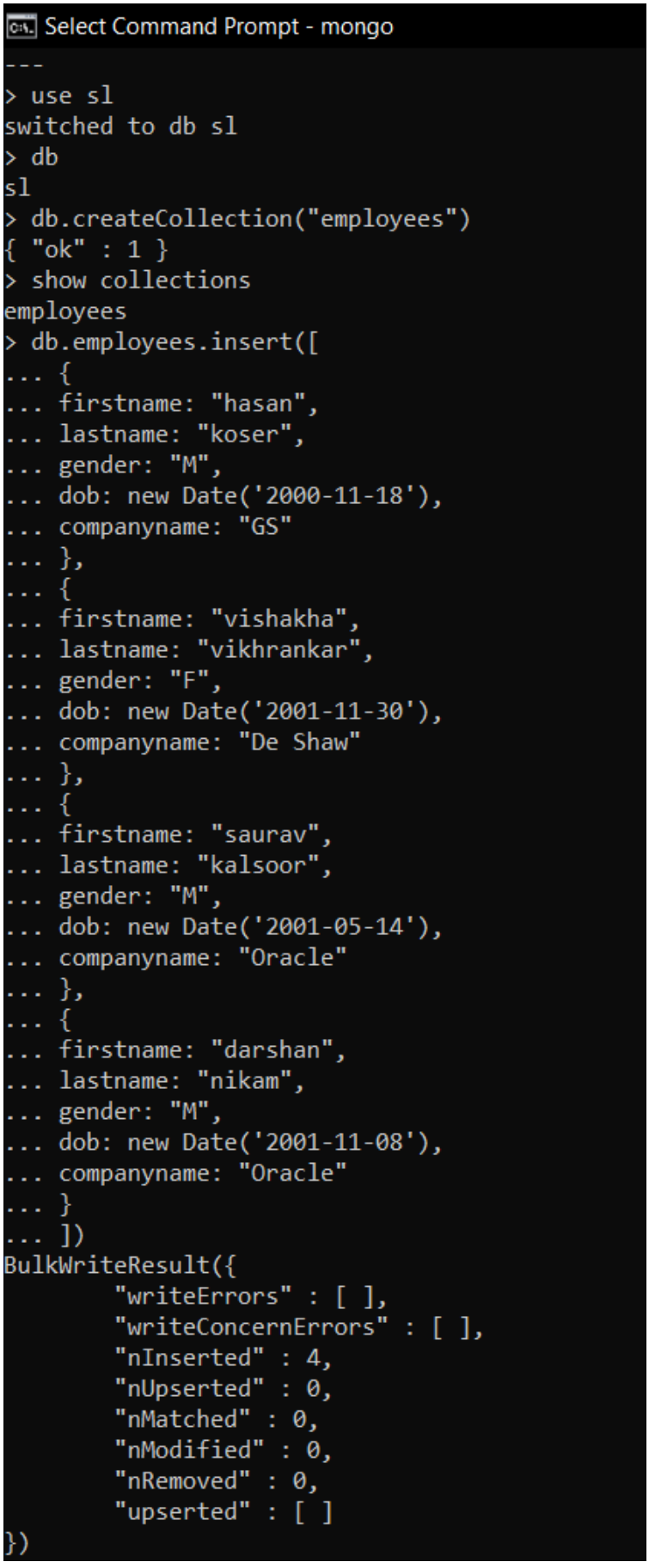
1. Create a collection with a document.

Code Snippet:-

```
use sl;

db;

db.createCollection("employees");

show collections;

db.employees.insert([
```

```
    {
        firstname: "hasan",
        lastname: "koser",
        gender: "M",
        dob: new Date('2000-11-18'),
        companyname: "GS"
    },
    {
        firstname: "vishakha",
        lastname: "vikhrankar",
        gender: "F",
        dob: new Date('2001-11-30'),
        companyname: "De Shaw"
    },
    {
        firstname: "saurav",
        lastname: "kalsoor",
        gender: "M",
        dob: new Date('2001-05-14'),
        companyname: "Oracle"
    },
    {
        firstname: "darshan",
        lastname: "nikam",
        gender: "M",
        dob: new Date('2001-11-08'),
        companyname: "Oracle"
    }
]);
```

Supported Screenshot:-

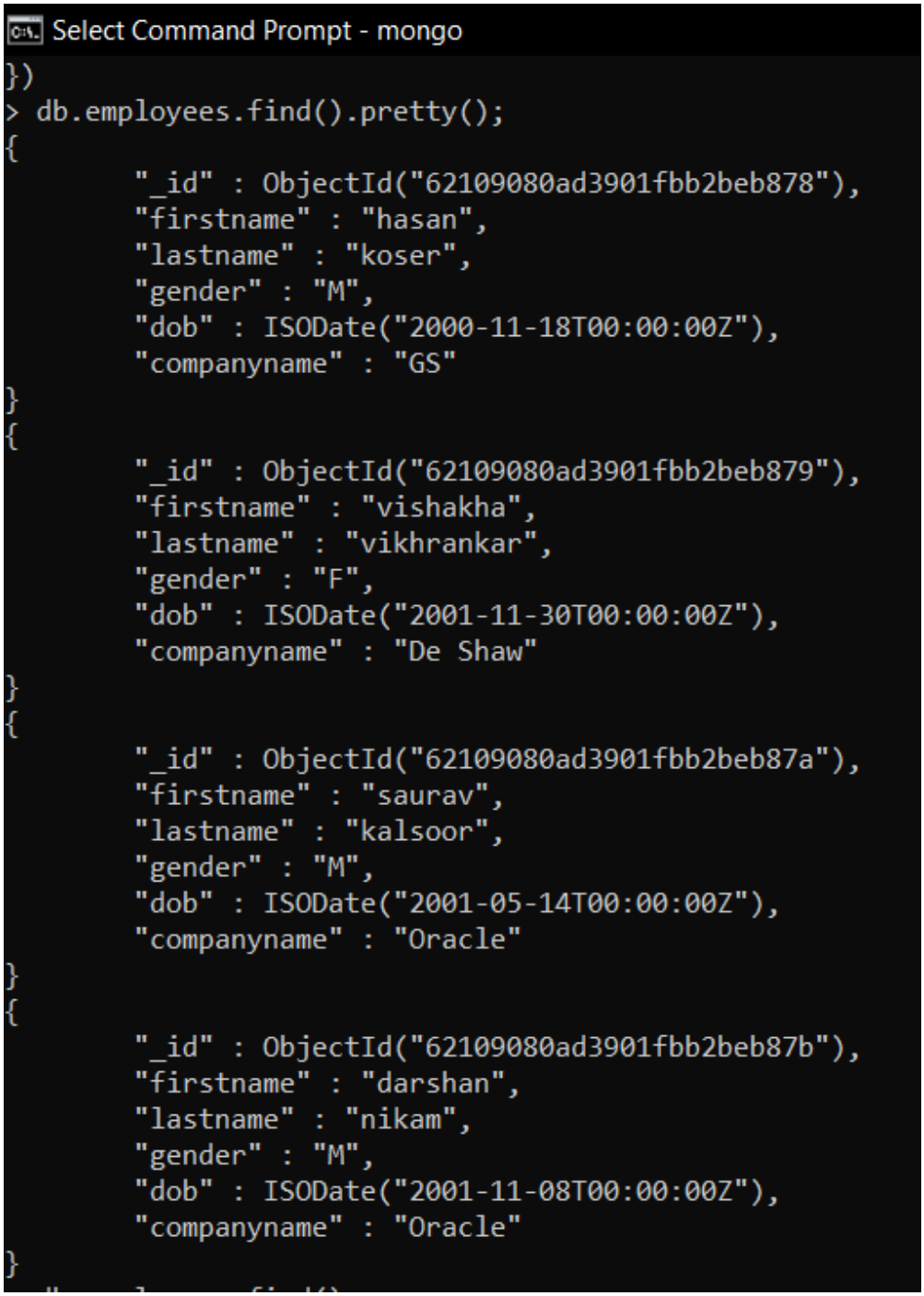


2. **Display all documents from a collection with the help of find() method.**

Code Snippet:-

```
db.employees.find().pretty();
```

Supported Screenshot:-

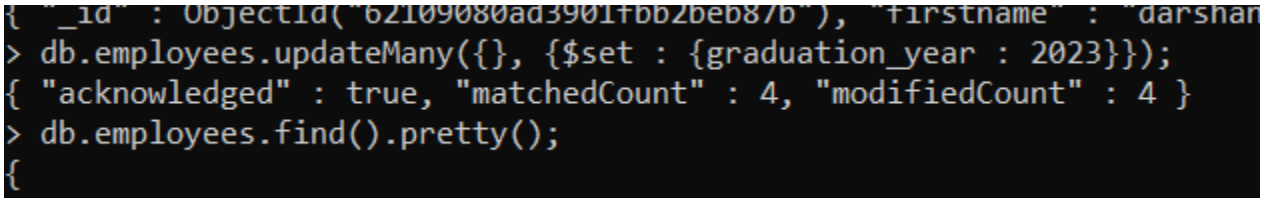


3. Write the query to update all documents.

Code Snippet:-

```
db.employees.updateMany({}, {$set : {graduation_year : 2023}});
```

Supported Screenshot:-



4. Now, check if the document has been updated or not.

Code Snippet:-

```
db.employees.find().pretty();
```

Supported Screenshot:-