

Assignment

Q1. What is MongoDB? Explain non-relational databases in short. In which scenarios it is preferred to use MongoDB over SQL databases?

Ans:- MongoDB is a non-relational document database that provides support for JSON-like storage, MongoDB is a popular document-oriented NoSQL database that stores data in a flexible JSON-like format called BSON.

Non-relational databases, also known as NoSQL databases, Non-relational databases are often used when large quantities of complex and diverse data need to be organized. store data differently than traditional relational databases, which use tables with rows and columns to store data. Non-relational databases use different data models, such as key-value pairs, document-based, graph-based, or column-family based data models.

Lesser Server Cost and Open-Source NoSQL databases are cheap and open source. NoSQL database implementation is easy and typically uses cheap servers to manage the exploding data and transaction while RDBMS databases are expensive and it uses big servers and storage systems. MongoDB is particularly well-suited for scenarios where high write throughput is required, such as in real-time data analysis, logging, and social networking applications. It is also useful for applications that require flexible data modeling, such as e-commerce platforms, content management systems, and mobile and web applications. SQL databases may be more appropriate for applications that require complex transactions, strong data consistency, and well-defined relationships between data entities.

Q2. State and Explain the features of MongoDB.

- Technical features of MongoDB
- Ad-hoc queries for optimized, real-time analytics. ...
- Indexing appropriately for better query executions. ...
- Replication for better data availability and stability. ...
- Sharding. ...
- Load balancing.

MongoDB is a popular NoSQL database that is designed to be flexible, scalable, and easy to use. Here are some of the key features of MongoDB:

Document-oriented: MongoDB is a document-oriented database, which means that it stores data as JSON-like documents, rather than in tables with rows and columns.

Dynamic schema: MongoDB has a flexible schema, which means that it can handle a wide variety of data types and structures. This makes it easier to evolve your data model over time as your application changes.

Assignment

Scalability: MongoDB is designed to scale horizontally, which means that it can handle large amounts of data and high traffic volumes by adding more servers to a cluster.

High availability: MongoDB provides automatic sharding and replication, which ensures that your data is always available and reduces the risk of data loss in case of server failures.

Rich query language: MongoDB provides a powerful query language that supports complex queries and aggregation operations, making it easy to extract insights from your data.

Indexing: MongoDB supports a range of indexing options, including text search, geospatial indexing, and hashed indexes, which can help you optimize your queries for faster performance.

Document validation: MongoDB allows you to enforce document validation rules to ensure that your data meets specific criteria, such as data type and value range constraints.

Overall, MongoDB's flexibility, scalability, and rich feature set make it a popular choice for a wide range of applications, including web and mobile applications, content management systems, real-time analytics, and more.

Q3. Write a code to connect MongoDB to Python. Also, create a database and a collection in MongoDB.

Ans:-codes are in ipynb file

In this code, we first create a MongoClient instance without any arguments, which will connect to the default MongoDB server.

We then create a database called my_database using the client['my_database'] syntax.

If the database does not already exist, it will be created automatically.

Finally, we create a collection called my_collection within the my_database database, using the db['my_collection'].

Create collections explicitly using the create_collection () method.

Make sure you have installed pymongo library using **pip install pymongo** command

Q4. Using the database and the collection created in question number 3, write a code to insert one record, and insert many records. Use the find() and find_one() methods to print the inserted record.

Ans:- answer are in Ipynb file

Q5. Explain how you can use the find () method to query the MongoDB database. Write a simple code to demonstrate this.

Ans:- Code is in Ipynb File

Assignment

As per q4 code only we added query command

query with a search criteria to find documents where the city "bdr".

We pass the query dictionary as an argument to the find () method, which returns a cursor object that we can iterate over to get the matching documents.

Finally, we print the matching documents using a for loop.

Q6.Explain the sort () method. Give an example to demonstrate sorting in MongoDB.

Ans:- the sort () method is used to sort the documents in a collection based on one or more fields. The method takes one or more arguments, which can be either field names or tuples of field names and sort order.

The default sort order is ascending (i.e., from lowest to highest value).

To sort in descending order (i.e., from highest to lowest value),

You can use the pymongo. DESCENDING constant.

Example is in ipynb file

Q7. Explain why delete_one(), delete_many(), and drop() is used.

In MongoDB, delete_one(), delete_many(), and drop() are used for deleting collections from a database.

delete_one(): This method is used to delete a single document that matches a specified filter. If there are multiple documents that match the filter, only the first document that matches will be deleted.

delete_many(): This method is used to delete all the documents that match a specified filter. If there are no documents that match the filter, no documents will be deleted.

drop(): This method is used to delete an entire collection from a database. This is a more drastic action than deleting individual documents, as it deletes all the documents in the collection and removes the collection itself.

In general, these methods are used when you want to remove documents or collections that are no longer needed, or to clean up your database.

It's important to use these methods carefully, as deleting data from a database can be a permanent action.