**Q 18)**

**a. What are MongoDB documents and collections. b. What syntax is used to create and drop a collection in MongoDB?**

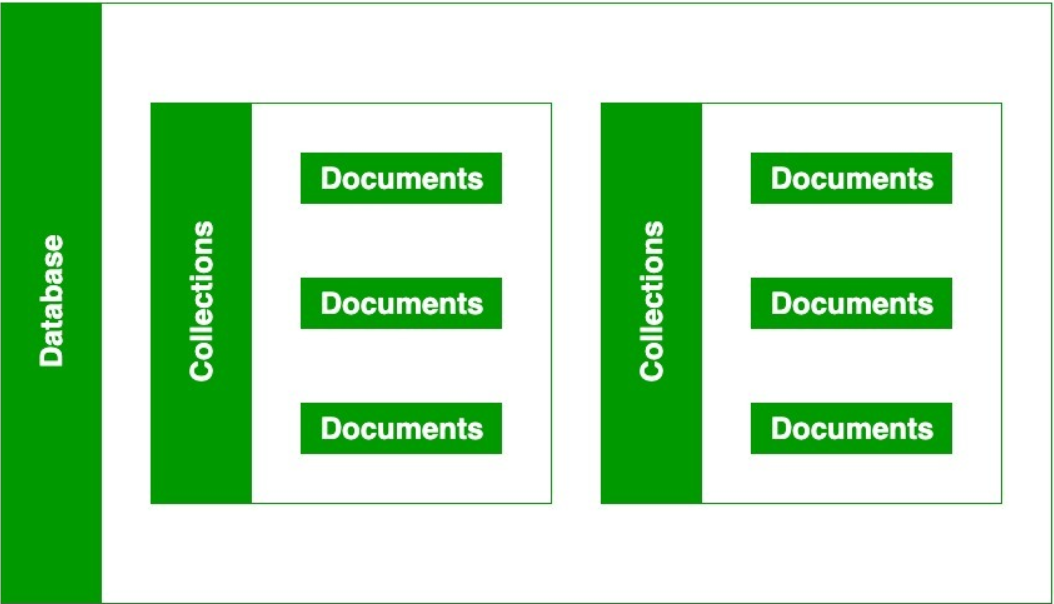
**Ans)**

MongoDB is a document-oriented NoSQL database system that provides high scalability, flexibility, and performance. Unlike standard relational databases, MongoDB stores data in a JSON document structure form. This makes it easy to operate with dynamic and unstructured data and MongoDB is an open-source and cross-platform database System.

**Database**

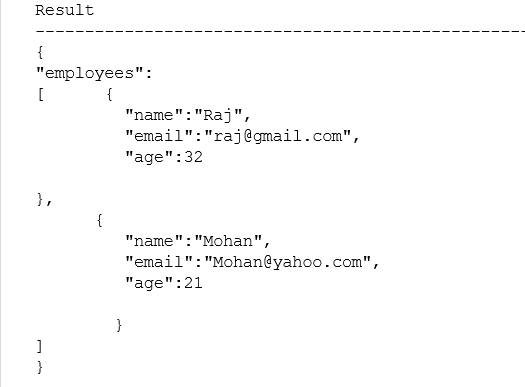
* Database is a container for collections.
* Each database gets its own set of files.
* A single MongoDB server can have multiple databases.

Databases, collections, documents are important parts of MongoDB without them you are not able to store data on the MongoDB server. A Database contains a collection, and a collection contains documents and the documents contain data, they are related to each other.



In MongoDB, a database contains the collections of documents. One can create multiple databases on the MongoDB server.

**Documents:**

* The document is the unit of storing data in a MongoDB database.
* document use JSON (JavaScript Object Notation, is a lightweight, thoroughly explorable format used to interchange data between various applications) style for storing data.
* A simple example of a JSON document is as follows:   
  **Employee.json**  
    
  
* Often, the term "object" is used to refer a document.
* Documents are analogous to the records of an RDBMS. Insert, update, and delete operations can be performed on a collection. The following table will help you to understand the concept more easily:

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB** |
| Table | Collection |
| Column | Key |
| Value | Value |
| Records / Rows | Document / Object |

* The following table shows the various datatypes which may be used in MongoDB.

|  |  |
| --- | --- |
| **Data Types** | **Description** |
| string | May be an empty string or a combination of characters. |
| integer | Digits. |
| boolean | Logical values True or False. |
| double | A type of floating point number. |
| null | Not zero, not empty. |
| array | A list of values. |
| object | An entity which can be used in programming. May be a value, variable, function, or data structure. |
| timestamp | A 64 bit value referring to a time and unique on a single "mongod" instance. The first 32 bit of this value refers to seconds since the UTC January 1, 1970. And last 32 bits refer to the incrementing ordinal for operations within a given second. |
| Internationalized Strings | UTF-8 for strings. |
| Object IDs | Every MongoDB object or document must have an Object ID which is unique. This is a BSON(Binary JavaScript Object Notation, which is the binary interpretation of JSON) object id, a 12-byte binary value which has a very rare chance of getting duplicated. This id consists of a 4-byte timestamp (seconds since epoch), a 3-byte machine id, a 2-byte process id, and a 3-byte counter. |

**Collections:**

* A collection may store a number of documents. A collection is analogous to a table of an RDBMS.
* A collection may store documents those who are not same in structure. This is possible because MongoDB is a Schema-free database. In a relational database like MySQL, a schema defines the organization / structure of data in a database. MongoDB does not require such a set of formula defining structure of data. So, it is quite possible to store documents of varying structures in a collection. Practically, you don't need to define a column and it's datatype unlike in RDBMS, while working with MongoDB.
* In the following code, it is shown that two MongoDB documents, belongs to same collection, storing data of different structures.



* A collection is created, when the first document is inserted.
* Collection names must begin with letters or an underscore.
* A Collection name may contain numbers.
* You can't use "$" character within the name of a collection. "$" is reserved.
* A Collection name must not exceed 128 characters. It will be nice if you keep it within 80/90 characters.
* Using a "." (dot) notation, collections can be organized in named groups. For example, tutorials.php and tutorials.javascript both belong to tutorials. This mechanism is called as collection namespace which is for user primarily. Databases don't have much to do with it.
* Following is how to use it programmatically:



**Creating a Collection:**

To create a collection in MongoDB, you use the “db.createCollection()” method.

Syntax:

db.createCollection(

"collectionName", // Name of the collection you want to create

options // Optional parameter for additional configuration options

)

* **"collectionName":** This is a string parameter representing the name of the collection you want to create. You can choose any valid string as the name for your collection. For example, "users", "products", "orders", etc.
* **options (optional):** This parameter allows you to specify additional configuration options for the collection being created. These options can include settings related to storage engine, validation rules, indexing options, etc. If you don't need to specify any options, you can omit this parameter.

**Dropping a Collection:**

To drop (delete) a collection in MongoDB, we use the db.collectionName.drop() method.

Syntax:

db.collectionName.drop()

* **"collectionName":** This is the name of the collection you want to drop. After connecting to a specific database, you can reference collections within that database directly using the db object, followed by the name of the collection. For example, if you have a collection named "users", you would reference it as db.users.
* **.drop():** This method is called on the collection object and tells MongoDB to delete the entire collection along with all its documents and indexes.