**Junaid Girkar**

**60004190057**

**TE COMPS A4**

**Experiment No. 8**

**Aim:** Data handling using JSON.

**Theory:**

**Documents**

A document is a record in a document database. A document typically stores information about one object and any of its related metadata. Documents store data in field-value pairs. The values can be a variety of types and structures, including strings, numbers, dates, arrays, or objects.

Documents can be stored in formats like JSON, BSON, and XML.

**Collections**

A collection is a group of documents. Collections typically store documents that have similar contents. Not all documents in a collection are required to have the same fields, because document databases have a flexible schema.

**Document Database:** A document database is a NoSQL data store that is designed to store and query data as JSON-like documents. The data in document databases is stored as documents with their metadata. The document stored is in a key/value pair where the key is the unique identifier of the document. Unlike relational databases, document databases are faster to load, access, and parse.

Document databases are also referred to as document database management systems, document-oriented databases, or document store databases.

Here are the key characteristics of document databases:

1. Document DBMSs are NoSQL databases.
2. Document DBMSs use key/value to store and access documents data.
3. Document DBMSs have a flexible schema that can be different for each document. For example, one document can be an Author profile, while another document can be a blog.
4. Common examples of document DBMS include JSON, XML docs, Catalogs, serialized PDFs and Excel docs, Profile data, and serialized objects.

Traditional relational DBMSs are not designed to provide efficient access to large documents or unstructured data. In case of catalogs, or profiles, or document storages, we don’t need structured design. For example, storing a document in a CMS does not require a structured format.

Document databases are designed to store large documents in a key/value store that are easy to search and access. The entire document is read into a memory object that is easy to read and present. User profiles, content management systems, and catalogs are some common use cases of document DBMS. One of the perfect examples of use of a document database is storing C#

Corner articles in a document DB, rather than a DBMS.

**MongoDB**

MongoDB is one of the most popular document databases. MongoDB is a free, distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. The document model maps to the objects in your application code, making data easy to work with.

**Implementation:**

1. **Importing a JSON file into Mongo**

Data:

| {  "quiz": {  "sport": {  "q1": {  "question": "Which one is correct team name in NBA?",  "options": [  "New York Bulls",  "Los Angeles Kings",  "Golden State Warriros",  "Huston Rocket"  ],  "answer": "Huston Rocket"  }  },  "maths": {  "q1": {  "question": "5 + 7 = ?",  "options": [  "10",  "11",  "12",  "13"  ],  "answer": "12"  },  "q2": {  "question": "12 - 8 = ?",  "options": [  "1",  "2",  "3",  "4"  ],  "answer": "4"  }  }  } } |
| --- |

Code:

| C: \Junaid\SEM5\ADBMS\Experiment 8>mongoimport --uri mongodb://localhost:27017/ADBMS --collection EXP8 --type json --file file.json |
| --- |

Output:

| 2022-01-28T12:52:29.15.0530 connected to: mongodb://localhost:27017/ADBMS  2022-01-28T12:52:29.174.0530 1 document(s) imported successfully. 0 document(s) failed to import. |
| --- |

1. **Printing the imported data in MongoDB shell:**

**Code:**

| > > use ADBMS switched to db ADBMS > db.EXP8.find() |
| --- |

**Output:**

| { "\_id" : ObjectId("61f399b5082f327f97e133d8"), "quiz" : { "sport" : { "q1" : { "question" : "Which one is correct team name in NBA?", "options" : [ "New York Bulls", "Los Angeles Kings", "Golden State Warriros", "Huston Rocket" ], "answer" : "Huston Rocket" } }, "maths" : { "q1" : { "question" : "5 + 7 = ?", "options" : [ "10", "11", "12", "13" ], "answer" : "12" }, "q2" : { "question" : "12 - 8 = ?", "options" : [ "1", "2", "3", "4" ], "answer" : "4" } } } } > |
| --- |

**Conclusion:** Hence, in this way we can import any JSON file into MongoDB and perform operations on it. This is how we handle JSON data by implementing it in document Databases.