MongoDB Cheat Sheet in Simple Terms

MongoDB Get Started

Purpose: Initial setup and basic operations to start using MongoDB. Example Usage: Download MongoDB, set up the MongoDB environment, and connect to the database using MongoDB Compass or CLI.

MongoDB Create DB

Purpose: Creating a new database in MongoDB.
Example Usage:

```
db = client.getDB("myNewDB");
```

MongoDB Collection

Purpose: Manage collections, which are groups of MongoDB documents, similar to tables in relational databases.
Example Usage:

```
db.createCollection("users");
```

MongoDB Insert

Purpose: Insert documents into a collection.
Example Usage:

```
db.users.insertOne({ name: "John Doe", age: 30 });
```

MongoDB Find

Purpose: Retrieve documents from a collection.
Example Usage:

```
const cursor = db.collection('users').find({ name: "John Doe" });
```

MongoDB Query

Purpose: Use query operators to find documents within a collection based on specific criteria.
Example Usage:

```

```

MongoDB Sort

Purpose: Sort the order of documents returned from a query.
Example Usage:

```
db.users.find().sort({ name: 1 }); // Ascending order
```

MongoDB Limit

Purpose: Limit the number of documents returned by a query.
Example Usage:

```
db.users.find().limit(5);
```

MongoDB Update

Purpose: Update documents in a collection.
Example Usage:

```
db.users.updateOne({ name: "John Doe" }, { $set: { age: 31 } });
```

MongoDB Delete

Purpose: Remove documents from a collection.
Example Usage:

```
db.users.deleteOne({ name: "John Doe" });
```

MongoDB Drop Collection

Purpose: Completely removes a collection and its indexes.
Example Usage:

```
db.collection('users').drop();
```

MongoDB Join

Purpose: Join data from multiple collections using lookup operations.
Example Usage:

```
db.orders.aggregate([
  {
    $lookup:
      {
        from: "users",
        localField: "user_id",
        foreignField: "_id",
        as: "orderdetails"
      }
  }
]);
```

MongoDB Aggregations

Purpose: Perform complex data processing and return computed results.
Example : Access query parameters and URL parameters.

```
db.users.aggregate([
  { $match: { status: "Active" } },
  { $group: { _id: "$age", total: { $sum: 1 } } }
]);
```

MongoDB Indexing/Search

Purpose: Improve the performance of search operations in MongoDB using indexes.
Example Usage :

```
db.users.createIndex({ name: 1 });
```

MongoDB Validation

Purpose: Ensures data consistency by validating documents before insertion or update.
Example Usage :

```
db.createCollection("users", {
  validator: { $jsonSchema: {
    bsonType: "object",
    required: [ "name", "email" ],
    properties: {
      name: {
        bsonType: "string",
        description: "must be a string and is required"
      },
      email: {
        bsonType: "string",
        pattern: "@mongodb\.com$",
        description: "must be a string and match the regular
expression pattern"
      }
    }
  }}
});
```

MongoDB Query Operators

Purpose: Operators used in queries to specify conditions in the find() method and other query operations.
Example Usage:

```
{ age: { $gt: 18 } }
```

MongoDB Update Operators

Purpose: Operators used to specify how the documents in a collection should be updated.
Example Usage:

```
{ $set: { status: "Active" } }
```

MongoDB Data API

Purpose: Allows applications to interact with your MongoDB data without using the MongoDB database drivers.
Example Usage: Useful for serverless applications where maintaining a persistent connection to the database is not feasible.

MongoDB Drivers

Purpose: Libraries available for different programming languages to interact with MongoDB.
Example Usage: Install the MongoDB driver for Node.js using **npm install mongodb.**

MongoDB Node.js Driver

Purpose: Official MongoDB driver designed for Node.js applications.
Example Usage: : Allows Node.js applications to connect to MongoDB and work with data programmatically.

MongoDB Charts

Purpose: Visualize MongoDB data directly from the database.
Example Usage: Enables users to create, share, and embed visual representations of MongoDB data on websites and applications.
Example Usage: Use MongoDB Charts to connect directly to your data, create visual representations using a drag-and-drop interface, and embed these charts into web applications for real-time data visualization.