

MongoDB Basics - Problem Statement 1 Solution

1. Installation of MongoDB:

Local Installation (Windows/Linux):

- **For Windows:**

1. Download the MongoDB installer from the official website: [MongoDB Download](#).
2. Run the installer and follow the on-screen instructions.
3. After installation, MongoDB will be installed in `C:\Program Files\MongoDB\Server\5.0\bin\` (or a similar path).
4. Add the MongoDB `bin` directory to your system's PATH environment variable.

- **For Linux:**

1. Update your package database:

```
sudo apt-get update
```

2. Install MongoDB:

```
sudo apt-get install -y mongodb
```

3. Start MongoDB service:

```
sudo service mongodb start
```

Cloud Installation (MongoDB Atlas):

1. Create an account on [MongoDB Atlas](#).
2. Follow the steps to create a cluster.

3. Once the cluster is created, get the connection string to connect to your cloud database using MongoDB shell or a GUI client.
-

2. Create a Sample Database and Collection

Creating Database and Collection via MongoDB Shell:

1. **Open the MongoDB shell** by typing `mongo` in the command prompt/terminal.
2. **Create a new database** named `ecommerce`:

```
use ecommerce;
```

3. **Create a collection** named `products` and insert sample data:

```
db.createCollection("products");
db.products.insertMany([
  { productName: "Laptop", price: 1200, category: "Ele
  { productName: "Smartphone", price: 800, category: "
  { productName: "Table", price: 150, category: "Furni
]);
```



3. Perform Basic Operations

Insert Data:

To insert documents into the `products` collection:

```
db.products.insertOne({ productName: "Headphones", price: 1
```



Retrieve Data:

To retrieve all documents in the `products` collection:

```
db.products.find();
```

To retrieve documents with a specific condition (e.g., products with price greater than 500):

```
db.products.find({ price: { $gt: 500 } });
```

Update Data:

To update a document (e.g., update the price of "Laptop"):

```
db.products.updateOne(  
  { productName: "Laptop" },  
  { $set: { price: 1100 } }  
);
```

Delete Data:

To delete a document (e.g., delete the product "Headphones"):

```
db.products.deleteOne({ productName: "Headphones" });
```

4. Sort Data:

To sort documents in the `products` collection by **price** in ascending order:

```
db.products.find().sort({ price: 1 });
```

To sort documents by **price** in descending order:

```
db.products.find().sort({ price: -1 });
```

5. Documenting the Installation and Operations:

Installation:

- **MongoDB Version:** MongoDB 5.0 (or the latest stable version).
- **Installation Method:** Installed locally on Windows or using MongoDB Atlas (cloud service).
- **MongoDB Version Installed:** Run `mongod --version` to verify.

Operations Performed:

1. **Created Database:** `ecommerce`

2. **Created Collection:** `products`

3. **Inserted Sample Data:**

- Product 1: `Laptop` , Price: 1200, Category: `Electronics`
- Product 2: `Smartphone` , Price: 800, Category: `Electronics`
- Product 3: `Table` , Price: 150, Category: `Furniture`

4. **Executed Queries:**

- Retrieve all products.
- Retrieve products with price greater than 500.
- Update the price of the `Laptop` product.
- Delete the `Headphones` product.

Queries Executed:

- **Insert Query:**

```
db.products.insertMany([...]);
```

- **Retrieve Query:**

```
db.products.find();  
db.products.find({ price: { $gt: 500 } });
```

- **Update Query:**

```
db.products.updateOne({ productName: "Laptop" }, { $set
```



- **Delete Query:**

```
db.products.deleteOne({ productName: "Headphones" });
```

Queries for Sorting:

```
db.products.find().sort({ price: 1 });  
db.products.find().sort({ price: -1 });
```

End of Document