CRUD Operations using NoSQL (MongoDB)

1. Create a Database

In MongoDB, databases are created automatically when you insert data into them.

Command: `use studentDB`

2. Create a Collection

Collections are created automatically when you insert the first document.

Command: `db.createCollection('students')`

3. Insert Data

Insert one document:

```
db.students.insertOne({ student_id: 101, name: 'Alice Johnson', course: 'Data
Science', marks: 88 })
```

Insert multiple documents:

```
db.students.insertMany([{ student_id: 102, name: 'Bob Smith', course: 'IoT', marks:
92 }, { student_id: 103, name: 'Charlie Brown', course: 'AI', marks: 79 }])
```

4. Select Data

```
Display all documents: db.students.find()
Display specific fields: db.students.find({}, { name: 1, course: 1 })
Apply condition: db.students.find({ course: 'IoT' })
Example with comparison: db.students.find({ marks: { $gt: 80 } })
```

5. Update Data

```
Update one document: db.students.updateOne({ student_id: 103 }, { $set: { marks: 85
} })
Update multiple documents: db.students.updateMany({ course: 'IoT' }, { $set: {
course: 'Internet of Things' } })
```

6. Delete Data

```
Delete one document: db.students.deleteOne({ student_id: 101 })
Delete multiple documents: db.students.deleteMany({ course: 'AI' })
Delete all: db.students.deleteMany({})
```

7. Drop Collection or Database

```
Drop collection: db.students.drop()
Drop database: db.dropDatabase()
```

Summary of Operations

Operation	MongoDB Command Example
Create Database	use studentDB

Create Collection	db.createCollection('students')
Insert Data	db.students.insertOne({...})
Read Data	db.students.find()
Update Data	db.students.updateOne({...})
Delete Data	db.students.deleteOne({...})