Introduction to MongoDB

- MongoDB is a source-available cross-platform document-oriented database program.
- It is a NoSQL database.

DBMS

Database: A database is a collection of related data, that can be processed to produce information.

Relationship in DB

A relationship is an association between tables. Those associations create using join statements to retrieve data. It is a condition that exists between two database tables in which one table contains a foreign key that references the primary key of the other tables. Relationships enable relational databases to divide and store data in separate tables while connecting disparate data items.

MongoDB Query

1. Create Database

Syntax: use database_name

```
Use seu
```

2. Show databases

Syntax: show dbs

3. Create table

Syntax:

```
db.createCollection('Student_table')
db.createCollection('Course_table')
 db.createCollection('Reg_table')
```

4. Show tables

Syntax: show collections

5. Insert student table

Syntax: db.collection_name.insertMany([{ }])

```
db.Student_table is not a function
seu>
db.Student_table.insertMany([{ID:101,NAME:'Sifat',Address:'Dhaka',
Blood:'O'},{ID:102,NAME:'Nazmul',Address:'Dhaka',Blood:'A+'},{ID:10
3,NAME:'Nayeem',Address:'CTG',Blood:'A-'}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63dd1c274a959f56062a147f"),
    '1': ObjectId("63dd1c274a959f56062a1480"),
    '2': ObjectId("63dd1c274a959f56062a1481")
  }
}
```

6. Read the table

Syntax : db.collection_name.find()

```
 db.Student_table.find()

Output:
[
  {
    _id: ObjectId("63dd1c274a959f56062a147f"),
    ID: 101,
    NAME: 'Sifat',
    Address: 'Dhaka',
    Blood: 'O'
  },
  {
    _id: ObjectId("63dd1c274a959f56062a1480"),
    ID: 102,
    NAME: 'Nazmul',
    Address: 'Dhaka',
    Blood: 'A+'
  },
  {
    _id: ObjectId("63dd1c274a959f56062a1481"),
    ID: 103,
    NAME: 'Nayeem',
    Address: 'CTG',
    Blood: 'A-'
  }
]
```

7. Update operation

Syntax:

```
db.collection_name.updateOne({ Document_name: }, {$set: {
Document_name: }})
```

```
db.Student_table.updateOne({Blood:'O'},{$set:{Blood:'o+'}})
```

Output:

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. sort()

Syntax:

```
db.collection_name.find().sort({ ID: -1 })    // -1 mean Des
```

```
db.Student_table.find().sort({ID:-1})
[ncaught:
  {taxError: Invalid shorthand property initializer. (1:23)
    _id: ObjectId("63dd1c274a959f56062a1481"),
    ID: 103,o.find().sort({ID=-1})
    NAME: 'Nayeem',          ^
    Address: 'CTG',
    Blood: 'A-'
  },| ss
  {
    _id: ObjectId("63dd1c274a959f56062a1480"),
    ID: 102,
    NAME: 'Nazmul',
    Address: 'Dhaka',
    Blood: 'A+'
  },
  {
    _id: ObjectId("63dd1c274a959f56062a147f"),
    ID: 101,
    NAME: 'Sifat',
    Address: 'Dhaka',
    Blood: 'o+'
  }
]
```

9. Hide operation

Syntax :

```
db.collection_name.find({},{document_name:0})
```

```
db.Student_table.find({},{_id:0})
[
  { ID: 101, NAME: 'Sifat', Address: 'Dhaka', Blood: 'o+' },
  { ID: 102, NAME: 'Nazmul', Address: 'Dhaka', Blood: 'A+' },
  { ID: 103, NAME: 'Nayeem', Address: 'CTG', Blood: 'A-' }
]
```