

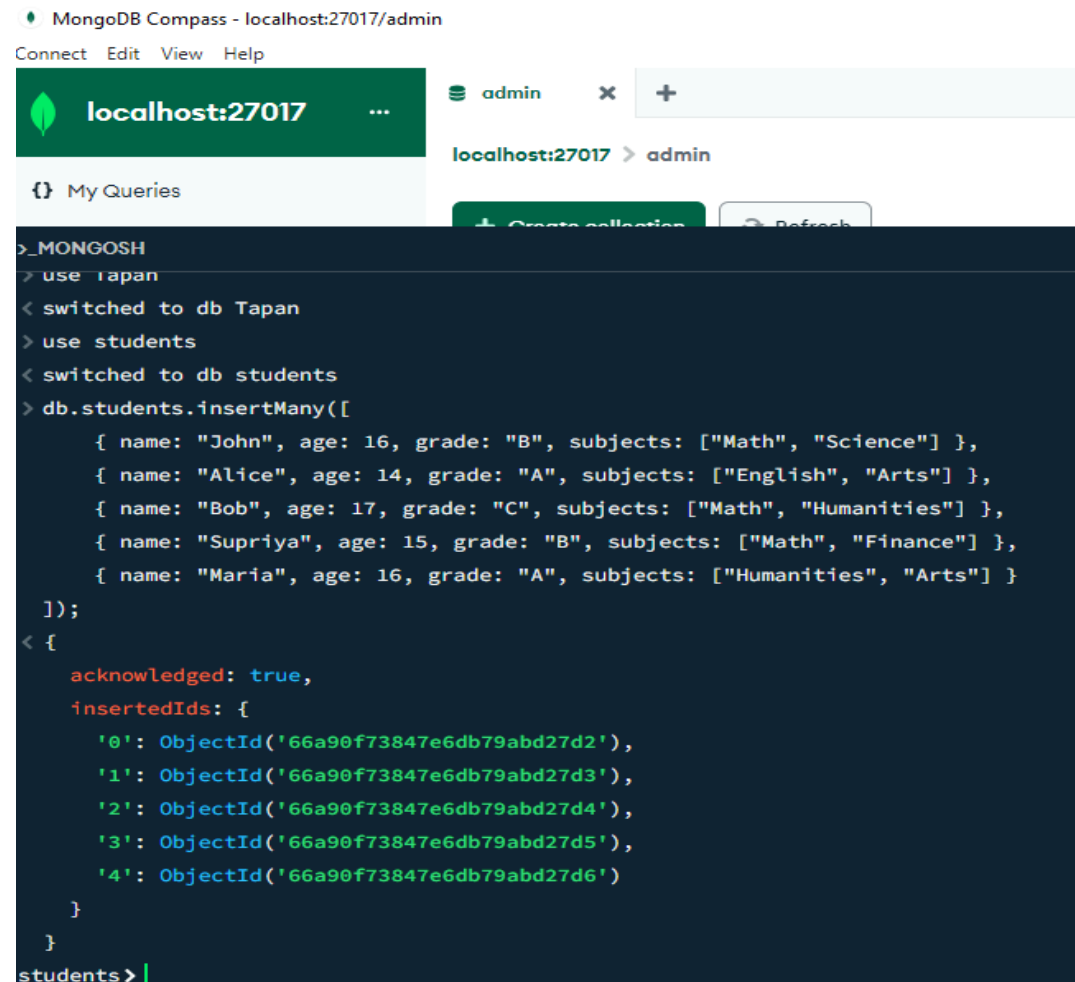
## Mongo DB

1)Create: Insert multiple documents into a collection called students with fields such as name, age, grade, and subjects.

Answer:-

Use students

```
db.students.insertMany([
  { name: "John", age: 16, grade: "B", subjects: ["Math", "Science"] },
  { name: "Alice", age: 14, grade: "A", subjects: ["English", "Arts"] },
  { name: "Bob", age: 17, grade: "C", subjects: ["Math", "Humanities"] },
  { name: "Supriya", age: 15, grade: "B", subjects: ["Math", "Finance"] },
  { name: "Maria", age: 16, grade: "A", subjects: ["Humanities", "Arts"] }
]);
```



The screenshot shows the MongoDB Compass interface at the top, with a connection to localhost:27017/admin. Below it, a terminal window displays the execution of the same MongoDB command. The terminal output shows the command being executed, the switch to the 'students' collection, and the successful insertion of five documents, each with a unique ObjectId.

```
>_MONGOSH
>use tapan
< switched to db Tapan
>use students
< switched to db students
>db.students.insertMany([
  { name: "John", age: 16, grade: "B", subjects: ["Math", "Science"] },
  { name: "Alice", age: 14, grade: "A", subjects: ["English", "Arts"] },
  { name: "Bob", age: 17, grade: "C", subjects: ["Math", "Humanities"] },
  { name: "Supriya", age: 15, grade: "B", subjects: ["Math", "Finance"] },
  { name: "Maria", age: 16, grade: "A", subjects: ["Humanities", "Arts"] }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66a90f73847e6db79abd27d2'),
    '1': ObjectId('66a90f73847e6db79abd27d3'),
    '2': ObjectId('66a90f73847e6db79abd27d4'),
    '3': ObjectId('66a90f73847e6db79abd27d5'),
    '4': ObjectId('66a90f73847e6db79abd27d6')
  }
}
students>
```

## Read Operations

### 1.Find all students:

Answer:-

```
db.students.find({});
```

```
>_MONGOSH
    }
  }
> db.students.find({});
< {
  _id: ObjectId('66a90f73847e6db79abd27d2'),
  name: 'John',
  age: 16,
  grade: 'B',
  subjects: [
    'Math',
    'Science'
  ]
}
{
  _id: ObjectId('66a90f73847e6db79abd27d3'),
  name: 'Alice',
```

### 2.Find students who are older than 15 years:

Answer:-

```
db.students.find({ age: { $gt: 15 } });
```

```

>_MONGOSH

db.students.find({ age: { $gt: 15 } });
< {
  _id: ObjectId('66a90f73847e6db79abd27d2'),
  name: 'John',
  age: 16,
  grade: 'B',
  subjects: [
    'Math',
    'Science'
  ]
}
{
  _id: ObjectId('66a90f73847e6db79abd27d4'),
  name: 'Bob',
  age: 17,
  grade: 'C',

```

3.Find students who have "Math" as one of their subjects:

Answer

```
db.students.find({ subjects: "Math" });
```

```

>_MONGOSH

db.students.find({ subjects: "Math" });
< {
  _id: ObjectId('66a90f73847e6db79abd27d2'),
  name: 'John',
  age: 16,
  grade: 'B',
  subjects: [
    'Math',
    'Science'
  ]
}
{
  _id: ObjectId('66a90f73847e6db79abd27d4'),
  name: 'Bob',
  age: 17,
  grade: 'C',
  subjects: [
    'Math',

```

#### 4. Find students who have "Math" and "Humanities" subjects:

Answer:-

```
db.students.find({ subjects: { $all: ["Math", "Humanities"] } });
```

```
}
> db.students.find({ subjects: { $all: ["Math", "Humanities"] } });
< {
  _id: ObjectId('66a90f73847e6db79abd27d4'),
  name: 'Bob',
  age: 17,
  grade: 'C',
  subjects: [
    'Math',
    'Humanities'
  ]
}
students>
```

#### 5. Find students who do not have both "Finance" and "Arts" subjects:

Answer:-

```
db.students.find({ $nor: [{ subjects: "Finance" }, { subjects: "Arts" }] });
```

>\_MONGOSH

```
db.students.find({ $nor: [{ subjects: "Finance" }, { subjects: "Arts" }] });
```

```
< {
  _id: ObjectId('66a90f73847e6db79abd27d2'),
  name: 'John',
  age: 16,
  grade: 'B',
  subjects: [
    'Math',
    'Science'
  ]
}
{
  _id: ObjectId('66a90f73847e6db79abd27d4'),
  name: 'Bob',
  age: 17,
  grade: 'C',
  subjects: [
    'Math',
    'Humanities'
  ]
}
```