

## Program 1

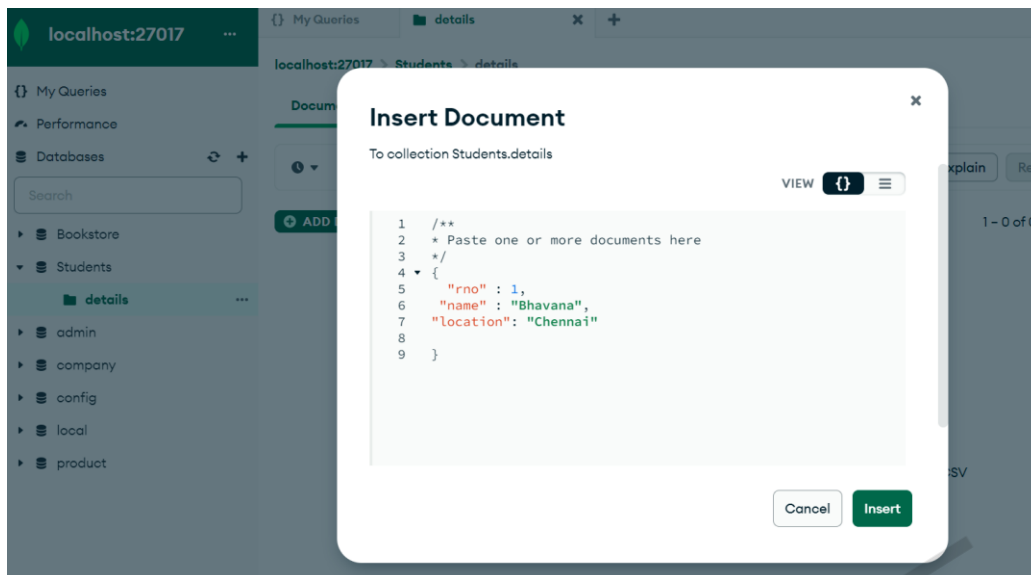
**a. Illustration of Where Clause, AND, OR operations in MongoDB.**

Create a database **Students** and collection **details** in Mongo DB IDE.

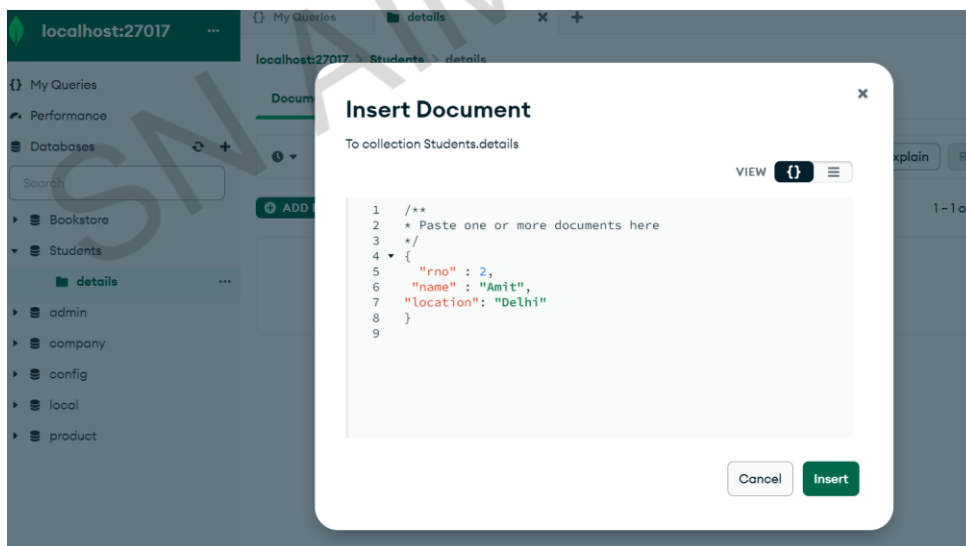
The screenshot shows the 'Create Database' dialog in MongoDB Atlas. The 'Database Name' field contains 'Students' and the 'Collection Name' field contains 'details'. The 'Time-Series' checkbox is unchecked. The 'Additional preferences' section is collapsed. The 'Create Database' button is highlighted in green.

Add the following documents in the **details** collection in MongoDB IDE.

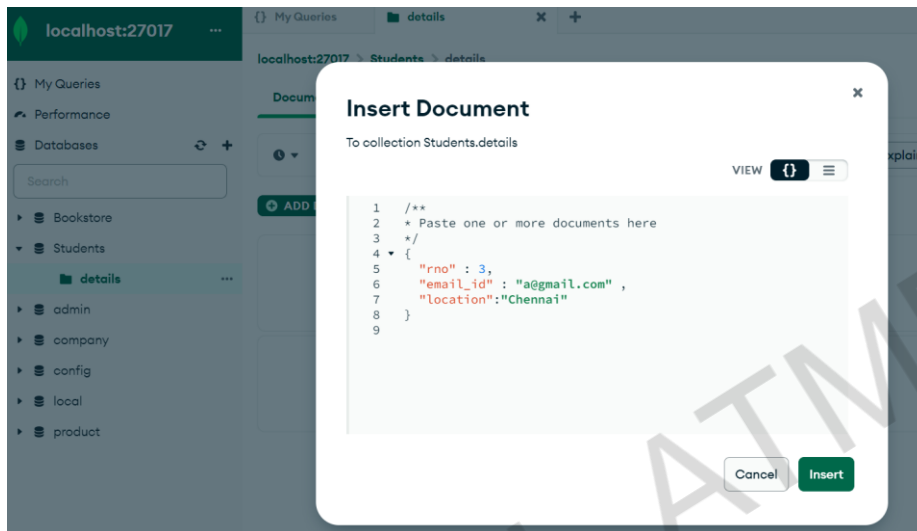
```
{
  "rno" : 1,
  "name" : "Bhavana",
  "location": "Chennai"
}
```



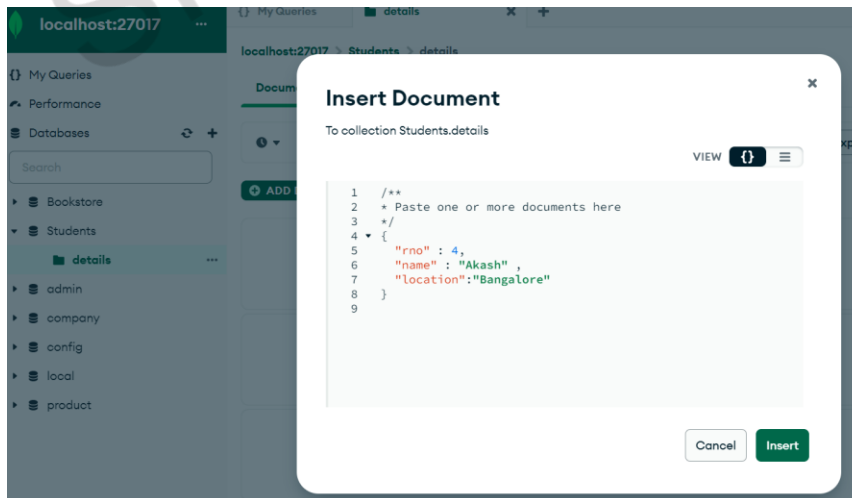
```
{
  "rno" : 2,
  "name" : "Amit",
  "location": "Delhi"
}
```



```
{  
  "rno" : 3,  
  "email_id" : "a@gmail.com",  
  "location": "Chennai"  
}
```



```
{  
  "rno" : 4,  
  "name" : "Akash",  
  "location": "Bangalore"  
}
```



1. **Where Clause in MongoDB:** In MongoDB, the find() method is used to query documents from a collection. The find() method can accept a query document as a parameter which acts as a "WHERE" clause.

**Syntax:** db.collection.find({ field: value })

**In MongoDB shell, execute the following code:**

**> use Students**

**> db.details.find()**

**Output:**

```
< {
  _id: ObjectId('665366c71d397fe133a7ade9'),
  rno: 1,
  name: 'Bhavana'
}
{
  _id: ObjectId('665367841d397fe133a7adeb'),
  rno: 3,
  email_id: 'a@gmail.com'
}
{
  _id: ObjectId('665367991d397fe133a7aded'),
  name: 'Amit',
  rno: 2
}
{
  _id: ObjectId('665367e81d397fe133a7adef'),
  rno: 4,
  name: 'Akash'
}
```

**//findOne to show only first record**

**> db. details.findOne()**

**Output:**

```
< {
  _id: ObjectId('665366c71d397fe133a7ade9'),
  rno: 1,
  name: 'Bhavana'
}
```

---

2. **AND Operation in MongoDB:** MongoDB provides the \$and operator to perform logical AND operation between multiple conditions in a query.

**Syntax:** db.collection.find({ \$and: [ { field1: value1 }, { field2: value2 } ] })

>db.details.find({\$and: [{"location": "Chennai"},{rno:1}] })

**Output:**

```
< {
  _id: ObjectId('66537e4f1d397fe133a7adf1'),
  rno: 1,
  name: 'Bhavana',
  location: 'Chennai'
}
```

---

3. **OR Operation in MongoDB:** Similarly, MongoDB provides the \$or operator to perform logical OR operation between multiple conditions in a query.

**Syntax:** db.collection.find({ \$or: [ { field1: value1 }, { field2: value2 } ] })

>db.details.find({\$or: [{"location": "Chennai"}, {"location": "Delhi"}] })

**Output:**

```
< {
  _id: ObjectId('66537e4f1d397fe133a7adf1'),
  rno: 1,
  name: 'Bhavana',
  location: 'Chennai'
}
{
  _id: ObjectId('66537e741d397fe133a7adf3'),
  rno: 2,
  name: 'Amit',
  location: 'Delhi'
}
{
  _id: ObjectId('6653824c1d397fe133a7adf5'),
  rno: 3,
  email_id: 'a@gmail.com',
  location: 'Chennai'
}
```

---

**b. Execute the Commands of MongoDB and operations in MongoDB: Insert, Query, Update, Delete and Projection. (Note: use any collection).**

- 1. Insert Operation:** Use the insertOne() method to insert a single document into a collection.

**Syntax:** db.collection.insertOne({ field1: value1, field2: value2, field3: value3 })

**Every row/document can be different than other**

**> db.details.insertOne({name:'Amar',rno:5},{name:'Ajay',rno:10})**

Output:

```
< {
  acknowledged: true,
  insertedId: ObjectId('66580922f7e76d265c992a34')
}
```

**Verification Code:**

**>db.details.find({name:'Amar',rno:5})**

```
< {
  _id: ObjectId('665386afafe50186baf8fd4b'),
  name: 'Amar',
  rno: 5
}
```

>db.details.find({name:'Ajay',rno:10})

```
> db.details.find({name:'Ajay',rno:10})
<
```

> db.details.insert({rno:6, email\_id:'d@gmail.com'})

Output:

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66538753afe50186baf8fd4c')
  }
}
```

Verification Code:

>db.details.find({rno:6, email\_id:'d@gmail.com'})

```
< {
  _id: ObjectId('66538753afe50186baf8fd4c'),
  rno: 6,
  email_id: 'd@gmail.com'
}
```

// To insert date use ISODate function

> db.details.insert({rno:15, name:'Ravina', dob: ISODate("2019-09-14")})

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66538842afe50186baf8fd4d')
  }
}
```

### Verification Code:

```
> db.details.find({rno:15, name:'Ravina', dob: ISODate("2019-09-14")})
< {
  _id: ObjectId('66538842afe50186baf8fd4d'),
  rno: 15,
  name: 'Ravina',
  dob: 2019-09-14T00:00:00.000Z
}
```

---

//Insert multiple documents at once

```
> db.details.insert([{rno:7,name:'a'},{rno:8,name:'b'},{rno:8,name:'c'}])
```

### Output:

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66538970afe50186baf8fd4e'),
    '1': ObjectId('66538970afe50186baf8fd4f'),
    '2': ObjectId('66538970afe50186baf8fd50')
  }
}
```

### Verification Code:

```
> db.details.find({rno:7,name:'a'})
< {
  _id: ObjectId('66538970afe50186baf8fd4e'),
  rno: 7,
  name: 'a'
}
```

// to insert multiple values for one key using []

```
>db.details.insert({rno:10,name:'Ankit',hobbies:['singing','cricket','swimming',
'music'],age:21})
```

### Output:



```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66538a35afe50186baf8fd51')
  }
}
```

## Verification Code:

```
> db.details.find({rno:10,name:'Ankit',hobbies:['singing','cricket','swimming','music'],age:21})
< {
  _id: ObjectId('66538a35afe50186baf8fd51'),
  rno: 10,
  name: 'Ankit',
  hobbies: [
    'singing',
    'cricket',
    'swimming',
    'music'
  ],
  age: 21
}
```

---

**2. Query Operation:** Use the find() method to query documents from a collection.

**Syntax:** db.collection.find({ field: value })

**>db.details.find({rno:1})**

**Output:**

```
< {
  _id: ObjectId('66537e4f1d397fe133a7adf1'),
  rno: 1,
  name: 'Bhavana',
  location: 'Chennai'
}
```

---

**3.Delete Operation:** Use the deleteOne() method to delete a single document from a collection.

**Syntax:** db.collection.deleteOne({ field: value })

**>db.details.deleteOne({rno:1})**

**Output:**

```
< {  
  acknowledged: true,  
  deletedCount: 1  
}
```

**Verification Code:**

```
> db.details.find({rno:1})  
<
```

---

**>db. details.deleteMany( { location: "Chennai" } )**

**Output:**

```
< {  
  acknowledged: true,  
  deletedCount: 2  
}
```

**Verification Code:**

```
> db.details.find( { location: "Chennai" } )  
<
```

---

**Update Operation:** Use the updateOne() method to update a single document in a collection.

**Syntax:** db.collection.updateOne( { field: value }, { \$set: { fieldToUpdate: newValue } })

**> db.details.updateOne({ rno: 2 }, {\$set: { location: "Mysore" } })**

**Output:**

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

**Verification Code:**

```
> db.details.find({ rno: 2 })
< {
  _id: ObjectId('66537e741d397fe133a7adf3'),
  rno: 2,
  name: 'Amit',
  location: 'Mysore'
}
```

---

**4.Projection Operation:** Use the second parameter of the find() method to specify which fields to include or exclude in the query result.

**Syntax:** db.collection.find({}, { field1: 1, field2: 1, \_id: 0 })

**// Find command to show only names without condition**

**> db. details.find({}, {name:1,\_id:0})**

```
< {  
  name: 'Amit'  
}  
{  
  name: 'Amit'  
}  
{ }  
{  
  name: 'Amar'  
}  
{ }  
{  
  name: 'Ravina'  
}  
{  
  name: 'Reena'  
}
```

```
{  
  name: 'a'  
}  
{  
  name: 'b'  
}  
{  
  name: 'c'  
}  
{  
  name: 'Ankit'  
}
```