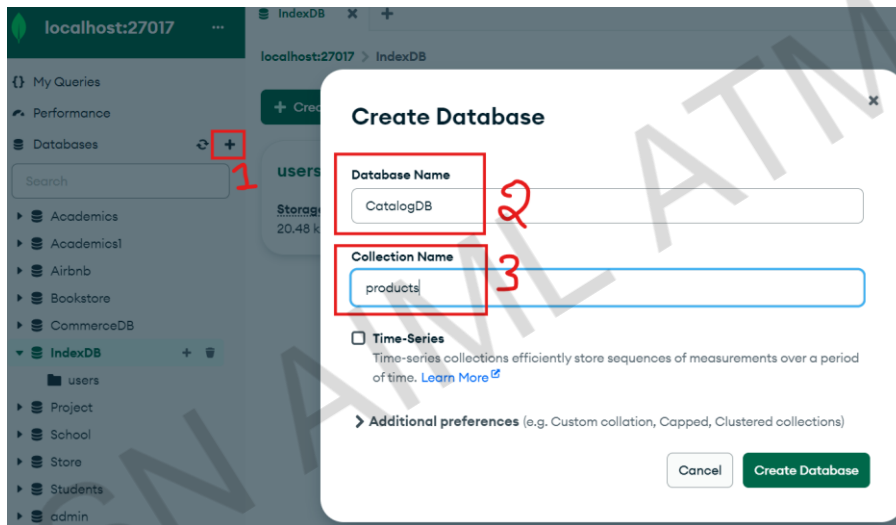


Program 9

- 9. a. Develop a query to demonstrate Text search using catalog data collection for a given word**
b. Develop queries to illustrate excluding documents with certain words and phrases.

Create a database **CatalogDB** and collection **products** in Mongo IDE.



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

```
{
  "name": "Apple iPhone 14",
  "description": "Latest model of iPhone with advanced features",
  "category": "Electronics"
}
```

Insert Document

To collection CatalogDB.products

VIEW { } ≡

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "name": "Apple iPhone 14",
6    "description": "Latest model of iPhone with advanced featu
7    "category": "Electronics"
8  }
```

Cancel

Insert

```
{
  "name": "Samsung Galaxy S21",
  "description": "Newest Samsung smartphone with great camera",
  "category": "Electronics"
}
```

Insert Document

To collection CatalogDB.products

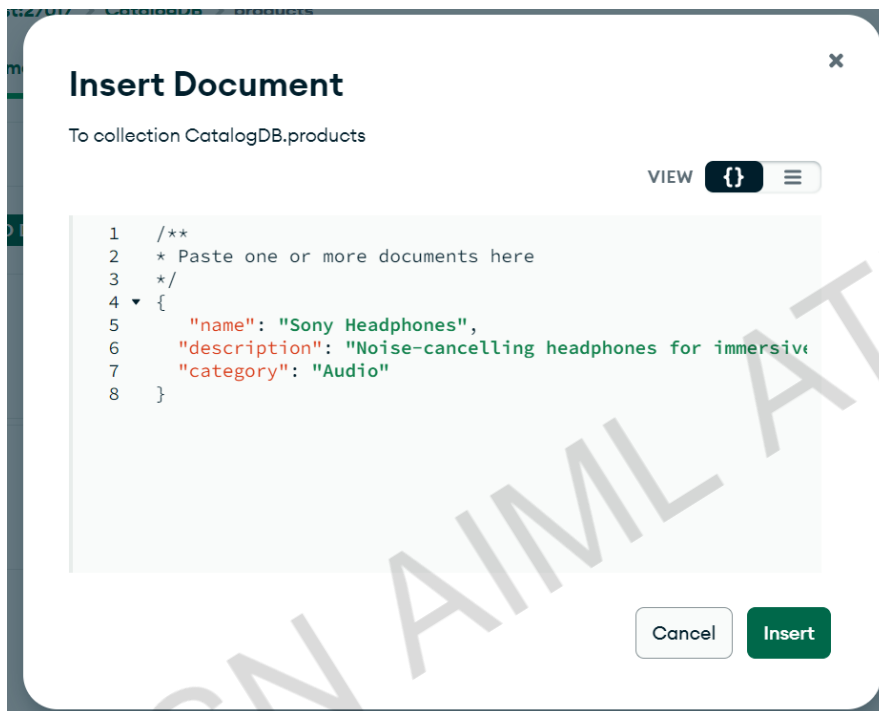
VIEW { } ≡

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "name": "Samsung Galaxy S21",
6    "description": "Newest Samsung smartphone with great camera",
7    "category": "Electronics"
8  }
```

Cancel

Insert

```
{  
  
  "name": "Sony Headphones",  
  
  "description": "Noise-cancelling headphones for immersive sound",  
  
  "category": "Audio"  
}
```



```
{  
  
  "name": "Dell Laptop",  
  
  "description": "High performance laptop for work and play",  
  
  "category": "Computers"  
}
```

Insert Document

To collection CatalogDB.products

VIEW { } ≡

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "name": "Dell Laptop",
6    "description": "High performance laptop for work and play",
7    "category": "Computers"
8  }
```

Cancel Insert

In MongoShell

>use CatalogDB

a. 1. Create a Text Index

To enable text search, you need to create a text index on the fields you want to search. Here, we'll create a text index on the `name` and `description` fields:

```
db.products.createIndex({ name: "text", description: "text" })
```

Output:

```
name_text_description_text
```

2. Perform a Text Search

Now, let's perform a text search. Suppose you want to search for products related to the word "latest":

```
db.products.find({ $text: { $search: "latest" } })
```

Output:

```
< {
  _id: ObjectId('6683756d028a2202a8cb7087'),
  name: 'Apple iPhone 14',
  description: 'Latest model of iPhone with advanced features',
  category: 'Electronics'
}
```

db.products.find({ \$text: { \$search: "High performance" } })

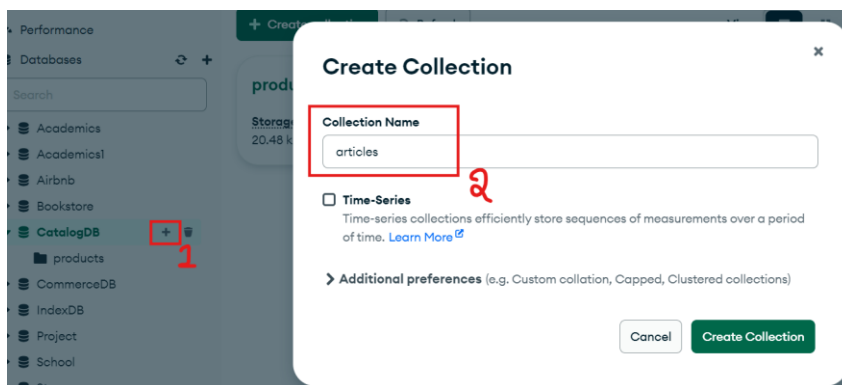
Output:

```
< {
  _id: ObjectId('66837a7a028a2202a8cb708d'),
  name: 'Dell Laptop',
  description: 'High performance laptop for work and play',
  category: 'Computers'
}
```

b. Develop queries to illustrate excluding documents with certain words and phrases.

In MongoDB, you can use the `$not` operator combined with the `$regex` operator to exclude documents that contain certain words or phrases. Below are some examples of queries to illustrate this.

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



```
{  
  
  "_id": 1,  
  
  "title": "MongoDB Basics",  
  
  "content": "This article explains the basics of MongoDB."  
  
}
```



Insert Document ✕

To collection CatalogDB.articles

VIEW { } ≡

```
1  /**  
2  * Paste one or more documents here  
3  */  
4  {  
5    "_id": 1,  
6    "title": "MongoDB Basics",  
7    "content": "This article explains the basics of MongoDB."  
8  }
```

Cancel Insert

```
{  
  
  "_id": 2,  
  
  "title": "Advanced MongoDB",  
  
  "content": "This article covers advanced MongoDB topics."  
  
}
```

Insert Document

To collection CatalogDB.articles

VIEW



```
1  /**
2  * Paste one or more documents here
3  */
4  ▼ {
5      "_id": 2,
6      "title": "Advanced MongoDB",
7      "content": "This article covers advanced MongoDB topics."
8  }
```

Cancel

Insert

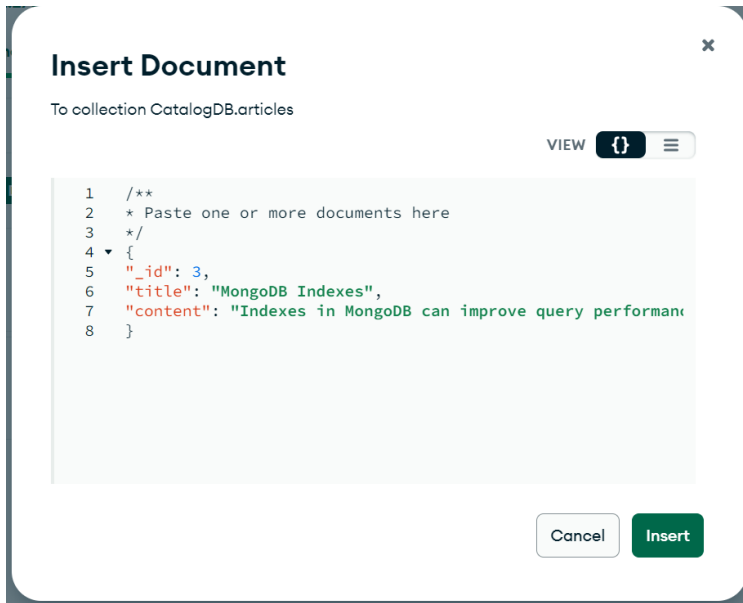
```
{
```

```
  "_id": 3,
```

```
  "title": "MongoDB Indexes",
```



```
  "content": "Indexes in MongoDB can improve query performance."
```

```
}
```



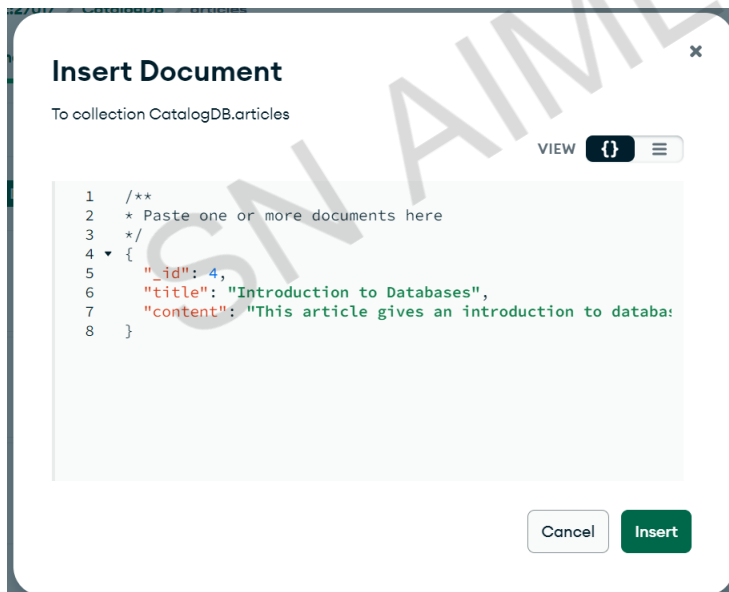
Insert Document

To collection CatalogDB.articles

VIEW  



```
1 /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": 3,
6    "title": "MongoDB Indexes",
7    "content": "Indexes in MongoDB can improve query performance"
8  }
```

```
{
  "_id": 4,
  "title": "Introduction to Databases",
  "content": "This article gives an introduction to databases in general."
}
```



Insert Document

To collection CatalogDB.articles

VIEW  

```
1 /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": 4,
6    "title": "Introduction to Databases",
7    "content": "This article gives an introduction to databases in general."
8  }
```

1. Exclude Documents Containing a Specific Word

To exclude documents that contain the word "advanced" in the 'content' field:


```
db.articles.find({
  "content": {
    $not: /advanced/
  }
})
```

Output:

```
< {
  _id: 1,
  title: 'MongoDB Basics',
  content: 'This article explains the basics of MongoDB.'
}
{
  _id: 3,
  title: 'MongoDB Indexes',
  content: 'Indexes in MongoDB can improve query performance.'
}
{
  _id: 4,
  title: 'Introduction to Databases',
  content: 'This article gives an introduction to databases in general.'
}
```

2. Exclude Documents Containing Any of Multiple Words

To exclude documents that contain either "improve" or "performance" in the 'content' field:

```
db.articles.find({
  "content": {
    $not: /(improve|performance)/
  }
})
```

Output:

```
< {
  _id: 1,
  title: 'MongoDB Basics',
  content: 'This article explains the basics of MongoDB.'
}
{
  _id: 2,
  title: 'Advanced MongoDB',
  content: 'This article covers advanced MongoDB topics.'
}
{
  _id: 4,
  title: 'Introduction to Databases',
  content: 'This article gives an introduction to databases in general.'
}
```

3. Exclude Documents Containing a Specific Phrase

To exclude documents that contain the phrase "MongoDB Basics" in the 'title' field:

```
db.articles.find({
```

```
  "title": {
```

```
    $not: /MongoDB Basics/
```

```
  }
```

```
})
```

Output:

```
< {
  _id: 2,
  title: 'Advanced MongoDB',
  content: 'This article covers advanced MongoDB topics.'
}
{
  _id: 3,
  title: 'MongoDB Indexes',
  content: 'Indexes in MongoDB can improve query performance.'
}
{
  _id: 4,
  title: 'Introduction to Databases',
  content: 'This article gives an introduction to databases in general.'
}
```

4. Exclude Documents Based on Multiple Fields

To exclude documents that contain "MongoDB" in the 'title' or "advanced" in the 'content':

```
db.articles.find({
  $and: [
    { "title": { $not: /MongoDB/ } },
    { "content": { $not: /advanced/ } }
  ]
})
```

Output:

```
< {
  _id: 4,
  title: 'Introduction to Databases',
  content: 'This article gives an introduction to databases in general.'
}
```

SN AIML ATME

SN AIML ATME