

Ubuntu Commands for Installing and setup of mongosh terminal ;

Step 1: Import the MongoDB Public Key

```
curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo gpg -o  
/usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
```

Step 2: Create the MongoDB Source List

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ]  
https://repo.mongodb.org/apt/ubuntu $(lsb_release -cs)/mongodb-org/7.0 multiverse"  
| sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
```

Step 3: Update Package Lists

```
sudo apt update
```

Step 4: Install MongoDB

```
sudo apt install -y mongodb-org
```

Step 5: Start and Enable the MongoDB Service

```
sudo systemctl start mongod  
sudo systemctl enable mongod
```

Step 6: Verify MongoDB Installation

```
sudo systemctl status mongod
```

Step 7: Access MongoDB Shell (mongosh)

```
mongosh
```

If mongosh isn't installed, you can install it separately:

```
sudo apt install -y mongodb-mongosh
```

Step 8: Configure MongoDB (Optional)

```
sudo nano /etc/mongod.conf
```

After making changes, restart MongoDB:

```
sudo systemctl restart mongod
```

Step 9: Creating a Database

```
use your_database_name
```

////////////////////////////////////

Problem Statement 19 (Aggregation & Indexing)

Create the Collection Movies_Data(Movie_ID, Movie_Name, Director, Genre,
BoxOfficeCollection) and
solve the following:

1. Display a list stating how many Movies are directed by each "Director".
2. Display list of Movies with the highest BoxOfficeCollection in each Genre.
3. Display list of Movies with the highest BoxOfficeCollection in each Genre in ascending order of BoxOfficeCollection.
4. Create an index on field Movie_ID.
5. Create an index on fields " Movie_Name" and " Director".
6. Drop an index on field Movie_ID.
7. Drop an index on fields " Movie_Name" and " Director".

----- starts from here-----

```

use Movie
.....

db.createCollection("Movies_Data")

.....

db.Movies_Data.insertMany([
  { Movie_ID: 1, Movie_Name: "Movie A", Director: "Director X", Genre: "Action",
BoxOfficeCollection: 100 },
  { Movie_ID: 2, Movie_Name: "Movie B", Director: "Director Y", Genre: "Action",
BoxOfficeCollection: 150 },
  { Movie_ID: 3, Movie_Name: "Movie C", Director: "Director X", Genre: "Drama",
BoxOfficeCollection: 200 },
  { Movie_ID: 4, Movie_Name: "Movie D", Director: "Director Z", Genre: "Drama",
BoxOfficeCollection: 250 },
  { Movie_ID: 5, Movie_Name: "Movie E", Director: "Director Y", Genre: "Comedy",
BoxOfficeCollection: 300 }
])

.....task 1

db.Movies_Data.aggregate([
  { $group: { _id: "$Director", TotalMovies: { $sum: 1 } } }
])

..... task 2

db.Movies_Data.aggregate([
  { $group: { _id: "$Genre", HighestBoxOffice: { $max:
"$BoxOfficeCollection" } } },
  {
    $lookup: {
      from: "Movies_Data",
      let: { genre: "$_id", collection: "$HighestBoxOffice" },
      pipeline: [
        { $match: { $expr: { $and: [ { $eq: ["$Genre", "$$genre"] }, { $eq:
["$BoxOfficeCollection", "$$collection"] } ] } } }
      ],
      as: "HighestMovie"
    }
  },
  { $unwind: "$HighestMovie" },
  { $replaceRoot: { newRoot: "$HighestMovie" } }
])

..... task 3

db.Movies_Data.aggregate([
  { $group: { _id: "$Genre", HighestBoxOffice: { $max:
"$BoxOfficeCollection" } } },
  {
    $lookup: {
      from: "Movies_Data",
      let: { genre: "$_id", collection: "$HighestBoxOffice" },

```

```

        pipeline: [
          { $match: { $expr: { $and: [ { $eq: ["$Genre", "$$genre"] }, { $eq:
["$BoxOfficeCollection", "$$collection"] } ] } } }
        ],
        as: "HighestMovie"
      }
    },
    { $unwind: "$HighestMovie" },
    { $replaceRoot: { newRoot: "$HighestMovie" } },
    { $sort: { BoxOfficeCollection: 1 } }
  ])

```

..... task 4

```
db.Movies_Data.createIndex({ Movie_ID: 1 })
```

```
db.Movies_Data.find().pretty()
```

.....task 5

```
db.Movies_Data.createIndex({ Movie_Name: 1, Director: 1 })
```

```
db.Movies_Data.find().pretty()
```

.....task 6

```
db.Movies_Data.dropIndex("Movie_ID_1")
```

```
db.Movies_Data.find().pretty()
```

.....task 7

```
db.Movies_Data.dropIndex("Movie_Name_1_Director_1")
```

```
db.Movies_Data.find().pretty()
```

.....

```
// to see the database changes
```

```
db.Movies_Data.find().pretty()
```