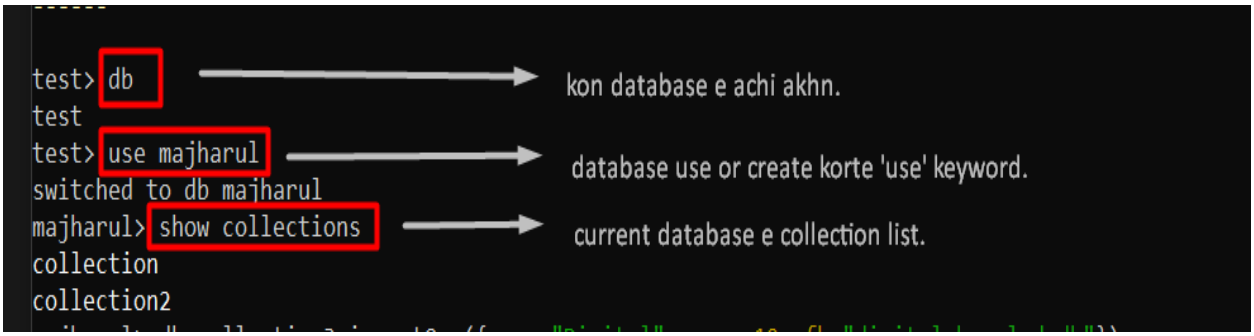
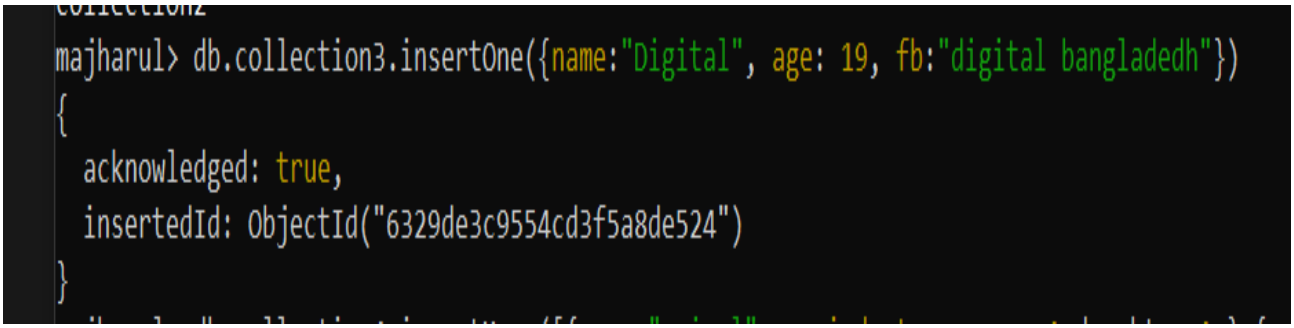
Create or Insert Operation:

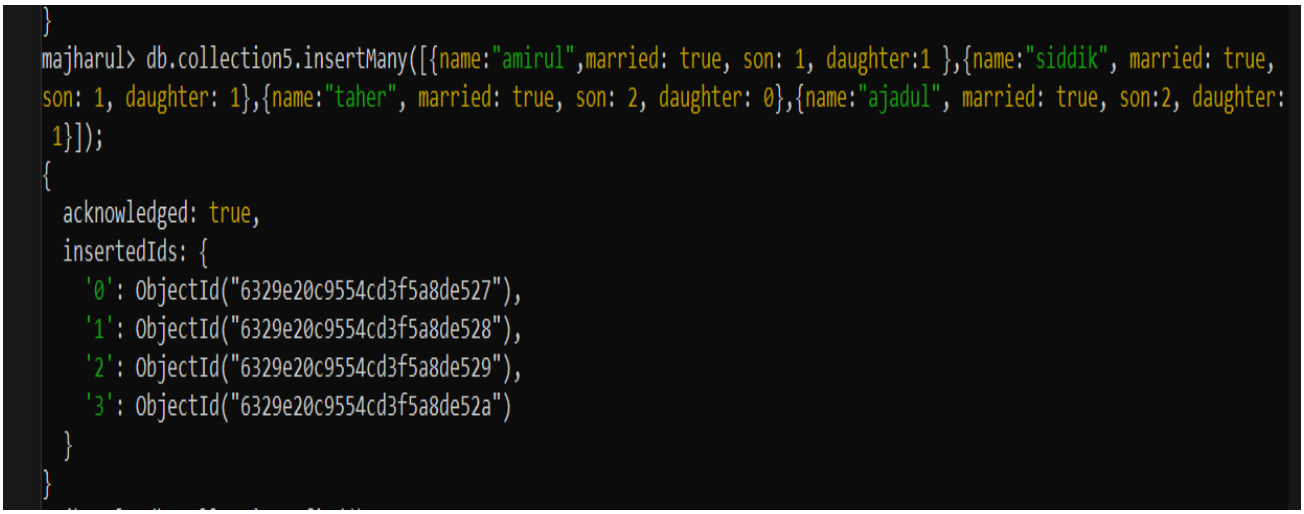
Database intro:



For creating collections(table) with One document(row):

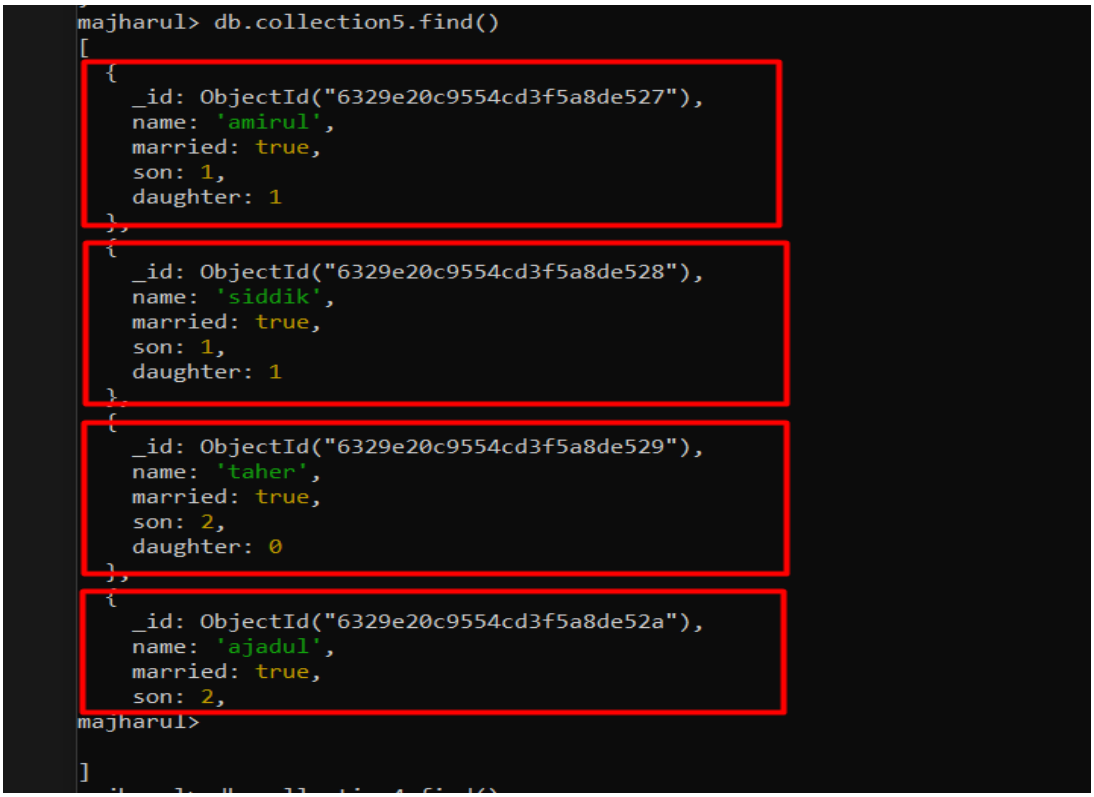


For creating collections(table) with Many document(row):



To show document(row) in collection(table) :

→ [db.collection name.find()]



Read or Queries the Documents :

Find by ‘name’: it will give total documents.

```
majharul> db.collection5.find({name:"ajadul"})
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de52a"),
    name: 'ajadul',
    married: true,
    son: 2,
    daughter: 1
  }
]
```

```
majharul> db.collection5.find({name:"ajadul"},{name:1})
[ { _id: ObjectId("6329e20c9554cd3f5a8de52a"), name: 'ajadul' } ]
```

```
majharul> db.collection5.find({name:"ajadul"},{_id:0,name:1})
[ { name: 'ajadul' } ]
```

Another Example:

```
majharul> db.collection5.find({son: 1})
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de527"),
    name: 'amirul',
    married: true,
    son: 1,
    daughter: 1
  },
  {
    _id: ObjectId("6329e20c9554cd3f5a8de528"),
    name: 'siddik',
    married: true,
    son: 1,
    daughter: 1
  }
]
```

```
majharul> db.collection5.find({son: 1}).limit(1)
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de527"),
    name: 'amirul',
    married: true,
    son: 1,
    daughter: 1
  }
]
```

Two way -> using 'limit'/'findOne'

```
majharul> db.collection5.findOne({son: 1})
{
  _id: ObjectId("6329e20c9554cd3f5a8de527"),
  name: 'amirul',
  married: true,
  son: 1,
  daughter: 1
}
```

```
majharul> db.collection5.find({son: 1}).limit(2)
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de527"),
    name: 'amirul',
    married: true,
    son: 1,
    daughter: 1
  },
  {
    _id: ObjectId("6329e20c9554cd3f5a8de528"),
    name: 'siddik',
    married: true,
    son: 1,
    daughter: 1
  }
]
```

//It means find and limit er first output ta skip korbe, porer ta show korbe.

```
majharul> db.collection5.find({son: 1}).limit(1).skip(1)
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de528"),
    name: 'siddik',
    married: true,
    son: 1,
    daughter: 1
  }
]
```

Update the Documents:

updateOne:

```
]
majharul> db.collection5.updateOne({name: "taher"},{$set: {son:3}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
{
  _id: ObjectId("6329e20c9554cd3f5a8de529"),
  name: 'taher',
  married: true,
  son: 3,
  daughter: 0
},
```

updateMany:

```
majharul> db.collection5.updateMany({son:1},{$set:{daughter: 0}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

```
majharul> db.collection5.find({son:1})
[
  {
    _id: ObjectId("6329e20c9554cd3f5a8de527"),
    name: 'amirul',
    married: true,
    son: 1,
    daughter: 0
  },
  {
    _id: ObjectId("6329e20c9554cd3f5a8de528"),
    name: 'siddik',
    married: true,
    son: 1,
    daughter: 0
  }
]
```

Delete the Documents:

```
  {
    _id: ObjectId("632abb92b8daf74d83e90a46"),
    name: 'null',
    married: null,
    son: null,
    daughter: null
  }
]
majharul> db.collection5.deleteMany({name:"null"})
{ acknowledged: true, deletedCount: 1 }
```

after calling deleteMany({name:"nul"})

```
//crate collection manually

db.createCollection('collection_name');

//delete collection from command

db.collection_name.drop()

//delete database from command

Db.dropDatabase();
```

More Info:

Documents — MongoDB Manual : https://www.mongodb.com/docs/manual/core/document/

Delete the Documents (mongodb/doc):

findOneAndDelete() provides a sort option. The option allows for the deletion of the first document sorted by the specified order.

`db.collection.findAndModify()` provides a sort option. The option allows for the deletion of the first document sorted by the specified order.

`db.collection.bulkWrite()` method provides the ability to perform bulk insert, update, and delete operations.

`bulkWrite()` supports the following write operations:

- insertOne
- updateOne
- updateMany
- replaceOne
- deleteOne
- deleteMany

```
try {
   db.pizzas.bulkWrite( [
      { insertOne: { document: { _id: 3, type: "beef", size: "medium", price: 6 } } },
      { insertOne: { document: { _id: 4, type: "sausage", size: "large", price: 10 } } },
      { updateOne: {
         filter: { type: "cheese" },
         update: { $set: { price: 8 } }
      } },
      { deleteOne: { filter: { type: "pepperoni"} } },
      { replaceOne: {
         filter: { type: "vegan" },
         replacement: { type: "tofu", size: "small", price: 4 }
      } }
   ] )
} catch( error ) {
   print( error )
}
```