===================== 1 =======================

db.createCollection("books", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["title"],

properties: {

title: {

bsonType: "string",

minLength: 1,

description: "Title is a required non-empty string"

}

}

}

}

});

{ ok: 1 }

================================ 2 =============================

db.authors.insertOne({

"name": "Author1",

"nationality": "British"

})  
{

acknowledged: true,

insertedId: ObjectId('6766e00a73a1b109001e4cdb')

}

==================================== 3 ========================================

db.createCollection("logs", {

capped: true,

size: 1024 \* 1024

});

{ ok: 1 }

=================================== 4 ==========================================

db.books.createIndex({"title": 1});

title\_1

=================================== 5 ==========================================

db.books.insertOne({

"title": "book1",

"auther": "ali",

"year": 1937,

"genre": ["fantasy", "Adventure"]

});

{

acknowledged: true,

insertedId: ObjectId('6766ec9f73a1b109001e4cdc')

}

================================== 6 =============================================  
db.books.insertMany([

{

"title": "Future",

"auther": "George Ore",

"year": 1937,

"genre": ["fantasy", "Adventure"]

},{

"title": "To Kill a Mockingbird",

"auther": "Harper Lee",

"year": 1960,

" genre": ["Classic", "Fiction"]

},{

"title": "Brave New world",  
 "auther": "Aldous Huxley",

"year": 2006,

" genre": ["Cystopian", "Science Fiction"]

}  
]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6766f03e73a1b109001e4cde'),

'1': ObjectId('6766f03e73a1b109001e4cdf'),

'2': ObjectId('6766f03e73a1b109001e4ce0')

}

}

===================================== 7 =============================================

db.logs.insertOne({ message: "New log entry", createdAt: new Date() });

===================================== 8 =============================================

db.books.updateOne({ title: "Future" }, { $set: { year: 2022 } });

===================================== 9 =============================================

db.books.find({ title: "Brave New World" });

===================================== 10 =============================================

db.books.find({ year: { $gte: 1990, $lte: 2010 } });

===================================== 11 =============================================

db.books.find({ genre: "Science Fiction" });

===================================== 12 =============================================

db.books.find({}).sort({ year: -1 }).skip(2).limit(3);

===================================== 13 =============================================

db.books.find({ year: { $type: "int" } });

===================================== 14 =============================================

db.books.find({ genre: { $nin: ["Horror", "Science Fiction"] } });

===================================== 15 =============================================

db.books.deleteMany({ year: { $lt: 2000 } });

===================================== 16 =============================================

db.books.aggregate([

{ $match: { year: { $gt: 2000 } } },

{ $sort: { year: -1 } }

]);

===================================== 17 =============================================

db.books.aggregate([

{ $match: { year: { $gt: 2000 } } },

{ $project: { \_id: 0, title: 1, author: 1, year: 1 } }

]);

===================================== 18 =============================================

db.books.aggregate([

{ $unwind: "$genre" }

]);

===================================== 19 =============================================

db.books.aggregate([

{ $lookup: {

from: "logs",

localField: "\_id",

foreignField: "bookId",

as: "logDetails"

}

}

]);