Chit 24 –

use my\_db

db.createCollection("movies")

db.movies.insert([

... {

... "name": "KGF",

... "type": "action",

... "budget": 1000000,

... "producer": {

... "name": "Raju",

... "address": "Pune"

... }

... },

... {

... "name": "Bahubali",

... "type": "mythical",

... "budget": 2000000,

... "producer": {

... "name": "S.S. Rajamouli",

... "address": "Pune"

... }

... },

... {

... "name": "Avatar",

... "type": "Sci-fi",

... "budget": 1000000000,

... "producer": {

... "name": "James",

... "address": "Mumbai"

... }

... },

... {

... "name": "RRR",

... "type": "adventure",

... "budget": 3000000,

... "producer": {

... "name": "Danayya",

... "address": "Mumbai"

... }

... },

... {

... "name": "Kantara",

... "type": "thriller",

... "budget": 5000000,

... "producer": {

... "name": "Vijay",

... "address": "Pune"

... }

... }

... ])

QUERIES

• Find the name of the movie having budget greater than 1,00,000.

db.movies.find({budget: {$gt: 100000}}).pretty()

• Find the name of producer who lives in Pune

db.movies.find({"producer.address": "Pune"}, {"producer.name": 1, \_id: 0}).pretty()

• Update the type of movie “action” to “horror”

db.movies.update({type: 'action'}, {$set: {type: 'horror'}})

• Find all the documents produced by name “producer1” with their address

db.movies.find({"producer.name": "producer1"}).pretty()

• write any query using aggregate function – No of movies of each producer

db.movies.aggregate(

... [

... {

... $group: {

... \_id: "$producer.name",

... no\_of\_movies: {$sum: 1}

... }

... }

... ]).pretty()