QUESTION SET-6

**QUESTION-1: Movies Collection**

**a) Retrieve all movies whose ratings include at least one rating greater than 8.0.**

db.Movies.find({ ratings: { $elemMatch: { $gt: 8.0 } } })

**b) Find movies that belong to either the "Action" or "Comedy" genre.**

db.Movies.find({ genres: { $in: ["Action", "Comedy"] } })

**c) Identify movies directed by "Christopher Nolan" and released after 2010.**

db.Movies.find({

director: "Christopher Nolan",

"details.releaseYear": { $gt: 2010 }

})

**d) Sort movies in descending order of release year and skip the first 5 records.**

db.Movies.find()

.sort({ "details.releaseYear": -1 })

.skip(5)

**e) Group movies by their director and count the total movies per director.**

db.Movies.aggregate([

{ $group: { \_id: "$director", totalMovies: { $sum: 1 } } }

])

**QUESTION-2: Users Collection**

**a) Retrieve users who have watched more than 10 movies but do not prefer "Horror".**

db.Users.find({

watchedMovies: { $size: { $gt: 10 } },

preferences: { $ne: "Horror" }

})

**b) Find users who have watched both "Inception" and "Interstellar".**

db.Users.find({ watchedMovies: { $all: ["Inception", "Interstellar"] } })

**c) Display names and contact details of users with a "Premium" subscription.**

db.Users.find(

{ subscriptionType: "Premium" },

{ \_id: 0, name: 1, contact: 1 }

)

**d) Create a sparse index on watchedMovies for optimizing queries.**

db.Users.createIndex(

{ watchedMovies: 1 },

{ sparse: true }

)

**e) Aggregate to project user’s name, subscription type, total watched movies, and filter out users with fewer than 5 movies.**

db.Users.aggregate([

{ $project: { name: 1, subscriptionType: 1, totalMovies: { $size: "$watchedMovies" } } },

{ $match: { totalMovies: { $gte: 5 } } }

])