QUESTION SET-8

**QUESTION-1: Songs Collection**

**a) Retrieve all songs whose ratings include at least one rating greater than 4.0.**

db.Songs.find({ ratings: { $elemMatch: { $gt: 4.0 } } })

**b) Find songs that belong to either the "Pop" or "Rock" genre.**

db.Songs.find({ genres: { $in: ["Pop", "Rock"] } })

**c) Identify songs by the artist "Adele" and released after 2015.**

db.Songs.find({

artist: "Adele",

"details.releaseYear": { $gt: 2015 }

})

**d) Sort songs in ascending order of release year and skip the first 3 records.**

db.Songs.find()

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

.skip(3)

**e) Group songs by their genre and count the total songs in each genre.**

db.Songs.aggregate([

{ $unwind: "$genres" },

{ $group: { \_id: "$genres", totalSongs: { $sum: 1 } } }

])

**QUESTION-2: Users Collection**

**a) Retrieve users who have more than 5 songs in their playlist but do not prefer "Classical".**

db.Users.find({

"playlists": { $size: { $gt: 5 } },

"preferences": { $ne: "Classical" }

})

**b) Find users who have added both "Shape of You" and "Blinding Lights" to their playlists.**

db.Users.find({ playlists: { $all: ["Shape of You", "Blinding Lights"] } })

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

db.Users.find(

{ subscriptionType: "Free" },

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

)

**d) Create a unique index on subscriptionType to ensure no duplicates.**

db.Users.createIndex(

{ subscriptionType: 1 },

{ unique: true }

)

**e) Aggregate to project user’s name, subscription type, total playlist songs, and filter users with fewer than 3 songs.**

db.Users.aggregate([

{ $project: { name: 1, subscriptionType: 1, totalSongs: { $size: "$playlists" } } },

{ $match: { totalSongs: { $gte: 3 } } }

])