NAME: NEHA K REG NO: 230701209

# EX N0:14

MongoDB Queries for Restaurants and Movies Collections

# Restaurants Collection Queries

1. Find restaurants that don't serve 'American' or 'Chinese' or whose names start with 'Wil': db.restaurants.find({

$or: [

{ cuisine: { $nin: ["American", "Chinese"] } },

{ name: { $regex: /^Wil/, $options: 'i' } }

]

}, { restaurant\_id: 1, name: 1, borough: 1, cuisine: 1 });

1. Find restaurants with a grade of 'A' and score 11 on '2014-08-11':

db.restaurants.find({

"grades": { $elemMatch: { "grade": "A", "score": 11, "date": ISODate("2014-08-11T00:00:00Z") } }

}, { restaurant\_id: 1, name: 1, grades: 1 });

1. Find restaurants where the 2nd element of grades contains a grade 'A' and score 9 on '2014- 0811':

db.restaurants.find({

"grades.1.grade": "A",

"grades.1.score": 9,

"grades.1.date": ISODate("2014-08-11T00:00:00Z")

}, { restaurant\_id: 1, name: 1, grades: 1 });

1. Find restaurants where the 2nd element of `coord` array contains a value more than 42 and up to 52:

db.restaurants.find({

"address.coord.1": { $gt: 42, $lte: 52 }

}, { restaurant\_id: 1, name: 1, address: 1, "address.coord": 1 });

1. Arrange restaurant names in ascending order with all columns: db.restaurants.find().sort({ name: 1 });
2. Arrange restaurant names in descending order with all columns: db.restaurants.find().sort({ name: -1 });
3. Arrange cuisine in ascending order and borough in descending order: db.restaurants.find().sort({ cuisine: 1, borough: -1 });
4. Check if all addresses contain the street field:

db.restaurants.find({ "address.street": { $exists: true } });

1. Select all documents where the `coord` field value is of type Double: db.restaurants.find({ "address.coord": { $type: "double" } });
2. Find restaurants where score is divisible by 7: db.restaurants.find({ "grades": { $elemMatch: { "score": { $mod: [7, 0] } } }

}, { restaurant\_id: 1, name: 1, grades: 1 });

1. Find restaurants where name contains 'mon': db.restaurants.find({ name: {

$regex: "mon", $options: "i" }

}, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 });

1. Find restaurants where name starts with 'Mad': db.restaurants.find({ name: { $regex:

"^Mad", $options: "i" }

}, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 });

1. Find restaurants with at least one grade with a score less than 5:

db.restaurants.find({

"grades": { $elemMatch: { "score": { $lt: 5 } } }

});

1. Find restaurants with at least one grade with a score less than 5 and located in Manhattan: db.restaurants.find({

"grades": { $elemMatch: { "score": { $lt: 5 } } }, borough: "Manhattan"

});

1. Find restaurants with at least one grade with a score less than 5 in Manhattan or Brooklyn: db.restaurants.find({

"grades": { $elemMatch: { "score": { $lt: 5 } } }, borough: { $in: ["Manhattan", "Brooklyn"] }

});

1. Find restaurants with at least one grade with a score less than 5 in Manhattan or Brooklyn and not American cuisine: db.restaurants.find({

"grades": { $elemMatch: { "score": { $lt: 5 } } },

borough: { $in: ["Manhattan", "Brooklyn"] }, cuisine: { $ne: "American" }

});

1. Find restaurants with at least one grade with a score less than 5 in Manhattan or Brooklyn and not American or Chinese cuisine:

db.restaurants.find({

"grades": { $elemMatch: { "score": { $lt: 5 } } },

borough: { $in: ["Manhattan", "Brooklyn"] }, cuisine: { $nin: ["American", "Chinese"] }

});

1. Find restaurants with grades having a score of 2 and a score of 6:

db.restaurants.find({

grades: { $all: [{ $elemMatch: { score: 2 } }, { $elemMatch: { score: 6 } }] }

});

1. Find restaurants with grades having a score of 2 and a score of 6 in Manhattan: db.restaurants.find({ grades: { $all: [{ $elemMatch: { score: 2 } }, {

$elemMatch: { score: 6 } }] }, borough: "Manhattan"

});

1. Find restaurants with grades having a score of 2 and a score of 6 in Manhattan or Brooklyn: db.restaurants.find({ grades: { $all: [{ $elemMatch: { score: 2 } }, {

$elemMatch: { score: 6 } }] }, borough: { $in: ["Manhattan", "Brooklyn"] }

});

1. Find restaurants with grades having a score of 2 and a score of 6 in Manhattan or Brooklyn and not American cuisine: db.restaurants.find({ grades: { $all: [{ $elemMatch: { score: 2 } }, {

$elemMatch: { score: 6 } }] }, borough: { $in: ["Manhattan", "Brooklyn"] }, cuisine: { $ne: "American" }

});

1. Find restaurants with grades having a score of 2 and a score of 6 in Manhattan or Brooklyn and not American or Chinese cuisine:

db.restaurants.find({ grades: { $all: [{ $elemMatch: { score: 2 } }, {

$elemMatch: { score: 6 } }] }, borough: { $in: ["Manhattan", "Brooklyn"]

}, cuisine: { $nin: ["American", "Chinese"] }

});

1. Find restaurants with a grade of 2 or a grade of 6: db.restaurants.find({

$or: [{ "grades.score": 2 }, { "grades.score": 6 }]

});

# Movies Collection Queries

1. Find movies released in 1893:

db.movies.find({ year: 1893

});

1. Find movies with runtime greater than 120 minutes:

db.movies.find({ runtime: { $gt: 120 }

});

1. Find movies with genre 'Short':

db.movies.find({ genres: "Short"

});

1. Find movies directed by 'William K.L. Dickson': db.movies.find({ directors: "William K.L. Dickson"

});

# 5) Find movies released in the USA:

db.movies.find({ countries: "USA"

});

1. Find movies released in the USA: db.movies.find({

countries: "USA"

});

1. Find movies rated as 'UNRATED':

db.movies.find({ rated: "UNRATED"

});

1. Find movies with more than 1000 votes on IMDb:

db.movies.find({

"imdb.votes": { $gt: 1000 }

});

1. Find movies with IMDb rating higher than 7: db.movies.find({ "imdb.rating": { $gt: 7 }

});

1. Find movies with viewer rating higher than 4 on Tomatoes:

db.movies.find({

"tomatoes.viewer.rating": { $gt: 4 }

});

1. Find movies that have received an award:

db.movies.find({ "awards.wins": { $gt: 0 }

});

1. Find movies with at least one nomination:

db.movies.find({ "awards.nominations": { $gt: 0 }

}, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime:

1, cast: 1, countries: 1 });

1. Find movies with cast including 'Charles Kayser': db.movies.find({ cast: "Charles Kayser"

}, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime:

1, cast: 1, countries: 1 });

1. Find movies released on May 9, 1893:

db.movies.find({ released:

ISODate("1893-05-09T00:00:00Z")

}, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 });

1. Find movies with 'scene' in the title: db.movies.find({ title: { $regex: "scene", $options: "i"

}

}, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 });