

EXP NO: 14

MONGO DB

Date:

## Restaurants Collection Queries

1. Find restaurants with cuisine not 'American' or 'Chinese' OR name starting with 'Wil':\*\*

```
```javascript
db.restaurants.find(
  {
    $or: [
      { cuisine: { $nin: ['American', 'Chinese'] } },
      { name: { $regex: '^Wil', $options: 'i' } }
    ]
  },
  { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 }
);
```
```

2. Find restaurants with grade 'A', score 11, and specific ISODate:\*\*

```
```javascript
db.restaurants.find(
  {
    grades: {
      $elemMatch: {
        grade: "A",
        score: 11,
        date: { $eq: ISODate("2014-08-11T00:00:00Z") }
      }
    }
  },
  { restaurant_id: 1, name: 1, grades: 1 }
);
```
```

3. \*\*Find restaurants where 2nd grade element is 'A', score 9, and specific ISODate:\*\*

```
```javascript
db.restaurants.find(
  {
    "grades.1": {
      grade: "A",
      score: 9,
      date: ISODate("2014-08-11T00:00:00Z")
    }
  }
);
```
```

```
    }  
  },  
  { restaurant_id: 1, name: 1, grades: 1 }  
);  
...
```

4. \*\*Find restaurants with 2nd `coord` array element between 42 and 52:\*\*

```
``javascript  
db.restaurants.find(  
  { "address.coord.1": { $gt: 42, $lte: 52 } },  
  { restaurant_id: 1, name: 1, address: 1, "address.coord": 1 }  
);  
...
```

5. \*\*Arrange restaurants in ascending order by name:\*\*

```
``javascript  
db.restaurants.find().sort({ name: 1 });  
...
```

6. \*\*Arrange restaurants in descending order by name:\*\*

```
``javascript  
db.restaurants.find().sort({ name: -1 });  
...
```

7. \*\*Arrange cuisine ascending and borough descending:\*\*

```
``javascript  
db.restaurants.find().sort({ cuisine: 1, borough: -1 });  
...
```

8. \*\*Check if all addresses contain a street field:\*\*

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

9. \*\*Select documents where `coord` field contains Double values:\*\*

```
``javascript  
db.restaurants.find(  
  { "address.coord": { $type: "double" } }  
);  
...
```

10. \*\*Find restaurants where score is divisible by 7:\*\*

```
``javascript  
db.restaurants.find(  
  { score: { $mod: 7, 0 } }  
);  
...
```

```

    {
      grades: { $elemMatch: { score: { $mod: [7, 0] } } }
    },
    { restaurant_id: 1, name: 1, grades: 1 }
  );
  ...

```

11. **\*\*Find restaurants with 'mon' in their name:\*\***

```

```javascript
db.restaurants.find(
  { name: { $regex: "mon", $options: "i" } },
  { name: 1, borough: 1, "address.coord": 1, cuisine: 1 }
);
...

```

12. **\*\*Find restaurants where name starts with 'Mad':\*\***

```

```javascript
db.restaurants.find(
  { name: { $regex: "^Mad", $options: "i" } },
  { name: 1, borough: 1, "address.coord": 1, cuisine: 1 }
);
...

```

13-17. **\*\*Find restaurants with grade score < 5 with borough or cuisine filters:\*\***

```

```javascript
// Example for Manhattan and Brooklyn, not American or Chinese cuisine:
db.restaurants.find(
  {
    grades: { $elemMatch: { score: { $lt: 5 } } },
    borough: { $in: ["Manhattan", "Brooklyn"] },
    cuisine: { $nin: ["American", "Chinese"] }
  }
);
...

```

18-22. **\*\*Find restaurants with specific grade scores and borough or cuisine filters:\*\***

```

```javascript
db.restaurants.find(
  {
    grades: {
      $all: [
        { $elemMatch: { score: 2 } },
        { $elemMatch: { score: 6 } }
      ]
    }
  }
);
...

```

```

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

```

23. **\*\*Find restaurants with grade score 2 OR 6:\*\***

```

```javascript
db.restaurants.find(
  {
    grades: { $elemMatch: { score: { $in: [2, 6] } } }
  }
);
...

```

---

### ### Movies Collection Queries

1. **\*\*Find movies released in 1893:\*\***

```

```javascript
db.movies.find({ year: 1893 });
...

```

2. **\*\*Find movies with runtime > 120 minutes:\*\***

```

```javascript
db.movies.find({ runtime: { $gt: 120 } });
...

```

3. **\*\*Find movies with "Short" genre:\*\***

```

```javascript
db.movies.find({ genres: "Short" });
...

```

4. **\*\*Find movies directed by "William K.L. Dickson":\*\***

```

```javascript
db.movies.find({ directors: "William K.L. Dickson" });
...

```

5. **\*\*Find movies released in the USA:\*\***

```

```javascript
db.movies.find({ countries: "USA" });
...

```

6. **\*\*Find movies rated as "UNRATED".\*\***

```
```javascript
db.movies.find({ rated: "UNRATED" });
```
```

7. **\*\*Find movies with more than 1000 IMDb votes.\*\***

```
```javascript
db.movies.find({ imdb.votes: { $gt: 1000 } });
```
```

8. **\*\*Find movies with IMDb rating > 7.\*\***

```
```javascript
db.movies.find({ imdb.rating: { $gt: 7 } });
```
```

9. **\*\*Find movies with viewer rating > 4 on Tomatoes.\*\***

```
```javascript
db.movies.find({ tomatoes.viewer.rating: { $gt: 4 } });
```
```

10. **\*\*Find movies that received awards.\*\***

```
```javascript
db.movies.find({ awards.wins: { $gt: 0 } });
```
```

11-13. **\*\*Find movies with nominations or specific cast.\*\***

```
```javascript
db.movies.find({ awards.nominations: { $gt: 0 } });
db.movies.find({ cast: "Charles Kayser" });
```
```

14. **\*\*Find movies released on May 9, 1893.\*\***

```
```javascript
db.movies.find({ released: ISODate("1893-05-09T00:00:00Z") });
```
```

15. **\*\*Find movies with 'scene' in the title.\*\***

```
```javascript
db.movies.find({ title: { $regex: "scene", $options: "i" } });
```
```