1. Find movies from 2010 that either won at least 5 awards or belong to the Drama genre

MongoDB Query:

```
db.movies.find( {
   year: 2010,
   $or: [ { "awards.wins": { $gte: 5 } }, { genres: "Drama" } ]
} )
```

SQL Equivalent:

```
SELECT *
FROM movies
WHERE year = 2010
AND (awards_wins >= 5 OR 'Drama' IN (SELECT genre FROM movie_genres WHERE
movie_genres.movie_id = movies.id));
```

(Assumes a separate `movie_genres` table storing genres per movie.)

---

2. Find movies produced in Mexico with an IMDb rating of at least 7

MongoDB Query:

```
db.movies.find( { countries: "Mexico", "imdb.rating": { $gte: 7 } } )
```

SQL Equivalent:

```
SELECT *
FROM movies
WHERE 'Mexico' IN (SELECT country FROM movie_countries WHERE
movie_countries.movie_id = movies.id)
AND imdb_rating >= 7;
```

(Assumes a separate `movie_countries` table storing country information.)

---

3. Get theater addresses in Massachusetts, formatted with specific fields

MongoDB Aggregation Query:

```
c = mflixdb.theaters.aggregate([
    {"$match": {"location.address.state": "MA"}},
    {"$project": {
        "_id": 0,
        "street": "$location.address.street1",
        "city": "$location.address.city",
        "zipcode": "$location.address.zipcode"
    }},
])
```

SQL Equivalent:

```
SELECT location_address_street1 AS street,
       location_address_city AS city,
       location_address_zipcode AS zipcode
FROM theaters
WHERE location_address_state = 'MA';
```

---

4. Count the number of theaters in each state and sort by state name

MongoDB Aggregation Query:

```
c = mflixdb.theaters.aggregate([
    {"$group": {"_id": "$location.address.state", "count": {"$sum": 1}}},
    {"$sort": {"_id": 1}}
])
```

SQL Equivalent:

```
SELECT location_address_state AS state, COUNT(*) AS count
FROM theaters
GROUP BY location_address_state
ORDER BY location_address_state ASC;
```

---

5. Find the longest movie and return its title and genres

MongoDB Aggregation Query:

```
longest_movie = mflixdb.movies.aggregate([
    {"$sort": {"runtime": -1}},
    {"$limit": 1},
    {"$project": {"_id": 0, "title": 1, "genres": 1}}
])
```

SQL Equivalent:

```
SELECT title, genres
FROM movies
ORDER BY runtime DESC
LIMIT 1;
```

6. Find movies released after 2010 with a viewer rating of at least 3, sorted by rating

MongoDB Aggregation Query:

```
matching_movies = mflixdb.movies.aggregate([
    {
        "$match": {
            "year": {"$gt": 2010},
            "tomatoes.viewer.rating": {"$gte": 3}
        }
    },
    {
        "$project": {
            "_id": 0,
            "title": 1,
            "viewer_rating": "$tomatoes.viewer.rating"
        }
    },
    {
        "$sort": {"viewer_rating": -1}
    }
])
```

SQL Equivalent:

```
SELECT title, tomatoes_viewer_rating AS viewer_rating
FROM movies
WHERE year > 2010
AND tomatoes_viewer_rating >= 3
ORDER BY viewer_rating DESC;
```