

MAS DSE 250 Beyond Relational Data

Models Homework 3

Due Tue, Nov 20 by midnight

This homework is a programming assignment.

Please turn it in electronically, by github, in form of a single text file containing your queries.

In this assignment, we will develop some Mongo queries over a movie graph of the following type (we are using GraphQL's type language for denoting types of objects).

We have a collection “persons” containing objects structured conforming to the type Person defined below. Within these objects, data are structured as follows:

For simplicity, assume that the name of a person and that the title of a movie are unique identifiers.

1. Find persons who served as actors or directors (not mutually exclusive).

Output just the persons:

```
db.movies.find({$or: [{acted_in: {$exists:1}}, {directed: {$exists:1}}]}, {name:1, _id:0}).sort({name:1})
```

Output persons and the movie:

```
db.movies.find({$or: [{acted_in: {$exists:1}}, {directed: {$exists:1}}]}, {name:1, "acted_in.in.title":1, "directed.title":1, _id:0}).sort({name:1})
```

2. Find persons who served as both actor and director in the same movie (display person name and movie title).

```
db.movies.aggregate({$unwind:"$acted_in", {$unwind:"$directed"}, {$match:{$expr: {$eq: ["$acted_in.in.title", "$directed.title"]}}}, {$project: {_id:0, name:1, "acted_in.in.title":1}})
```

3. For each director, find the number of roles they have directed over their entire career (sum over all movies the count of roles acted in those movies). Display director name and role count).

```
db.movies.aggregate([{$unwind: "$directed"}, {$lookup: { from:
"movies", localField: "directed.title", foreignField:
"acted_in.in.title", as: "join"}}, {$unwind: "$join"}, {$unwind:
"$join.acted_in"}, {$match: {$expr: {$eq: ["$join.acted_in.in.title",
"$directed.title"]}}}}, {$group: {_id: "$name", role_count: {$sum:
{$size: "$join.acted_in.roles"}}}}, {$sort: {_id: 1}}])
```

4. Find the names of actors who have worked on the same movie.

```
db.movies.aggregate([{$unwind: "$acted_in"}, {$lookup: { from:
"movies", localField: "acted_in.in.title", foreignField:
"acted_in.in.title", as: "join"}}, {$unwind: "$join"}, {$unwind:
"$join.acted_in"}, {$match: {$expr: {$eq: ["$join.acted_in.in.title",
"$acted_in.in.title"]}}}}, {$group: {_id: "$name", co_actors:
{$addToSet: "$join.name"}}}, {$unwind: "$co_actors"}, {$match: {$expr:
{$ne: ["$_id", "$co_actors"]}}}}, {$group: {_id: "$_id", co_actors:
{$addToSet: "$co_actors"}}}, {$out: "first_degree"}])
```

```
db.first_degree.find().pretty()
```

5. Find the names of actors connected to Kevin Bacon via at most 6 degrees of separation via the relationship acted_in_same_movie defined in Part 4.

```
db.first_degree.aggregate([{$unwind: "$co_actors"}, {$lookup: { from:
"first_degree", localField: "co_actors", foreignField: "_id", as:
"join"}}, {$unwind: "$join"}, {$unwind: "$join.co_actors"}, {$match:
{$expr: {$ne: ["$join.co_actors", "$_id"]}}}}, {$group: {_id: "$_id",
co_actors: {$addToSet: "$join.co_actors"}}}, {$out: "second_degree"}])
```

```
db.second_degree.aggregate([{$unwind: "$co_actors"}, {$lookup: { from:
"first_degree", localField: "co_actors", foreignField: "_id", as:
"join"}}, {$unwind: "$join"}, {$unwind: "$join.co_actors"}, {$match:
{$expr: {$ne: ["$join.co_actors", "$_id"]}}}}, {$group: {_id: "$_id",
co_actors: {$addToSet: "$join.co_actors"}}}, {$out: "third_degree"}])
```

```
db.third_degree.aggregate([{$unwind: "$co_actors"}, {$lookup: { from:
"first_degree", localField: "co_actors", foreignField: "_id", as:
"join"}}, {$unwind: "$join"}, {$unwind: "$join.co_actors"}, {$match:
{$expr: {$ne: ["$join.co_actors", "$_id"]}}}}, {$group: {_id: "$_id",
co_actors: {$addToSet: "$join.co_actors"}}}, {$out: "fourth_degree"}])
```

```
db.fourth_degree.aggregate([{$unwind: "$co_actors"}, {$lookup: { from:
"first_degree", localField: "co_actors", foreignField: "_id", as:
"join"}}, {$unwind: "$join"}, {$unwind: "$join.co_actors"}, {$match:
{$expr: {$ne: ["$join.co_actors", "$_id"]}}}}, {$group: {_id: "$_id",
co_actors: {$addToSet: "$join.co_actors"}}}, {$out: "fifth_degree"}])
```

```
db.fifth_degree.aggregate([{$unwind: "$co_actors"}, {$lookup: { from:
"first_degree", localField: "co_actors", foreignField: "_id", as:
"join"}}, {$unwind: "$join"}, {$unwind: "$join.co_actors"}, {$match:
{$expr: {$ne: ["$join.co_actors", "$_id"]}}}}, {$group: {_id: "$_id",
co_actors: {$addToSet: "$join.co_actors"}}}, {$out: "sixth_degree"}])
```

#1st degree separation

```
db.first_degree.find({_id:"Kevin Bacon"})
```

#2nd degree separation

```
db.second_degree.find({_id:"Kevin Bacon"})
```

#3rd degree separation

```
db.third_degree.find({_id:"Kevin Bacon"})
```

#4th degree separation

```
db.fourth_degree.find({_id:"Kevin Bacon"})
```

#5th degree separation

```
db.fifth_degree.find({_id:"Kevin Bacon"})
```

#6th degree separation

```
db.sixth_degree.find({_id:"Kevin Bacon"})
```