Unit 9: Databases & The Modern Web:

Lab 3 – People Pt 2 (Data Manipulation)

Assigned: 03/01/23

Due: 03/02/23

Completed: 03/02/23

**LMS:** <https://lms.grandcircus.co/mod/assign/view.php?id=22857>

**Google Doc:** <https://docs.google.com/document/d/1Y_nY0oDzPpndfAU1nAkTXwl1neTBNTehd6LIM4krwAw/preview>

**GitHub:** TBD

**Task:** Add, update, and delete documents as instructed below.

**Build Specifications:** Write MongoDB shell commands to do the following tasks. Record these commands in a text document and submit the document. You do not need to record the results of the commands.

In people collection

1. Add a person to the collection. You pick the data, but they should have an empty array for children.
   * db.people.insertOne({first\_name: ‘Kevin’, last\_name: ‘Durant’, email: ‘KD@phoenixsuns.com’, gender: ‘Male’, age: 37, state: ‘Arizona’, children: [] })
2. Add another person. They should have at least two children.
   * db.people.insertOne({first\_name: ‘Chris’, last\_name: ‘Paul’, email: ‘CP3@phoenixsuns.com’, gender: ‘Male’, age: 39, state: ‘Arizona’, children: [ {name: ‘Chris Jr’, age: 13}, {name: ‘Camryn’, age: 10} ]
3. Update one person named Clarence. He moved from North Dakota to South Dakota.
   * db.people.updateOne({ name: ‘Clarence’ }, { $set: {state: ‘South Dakota’} } )
4. Update Rebecca Hayes. Remove her email address.
   * db.people.updateOne({ name: ‘Rebecca’ }, { $set: {email: null} } })
5. Update everyone from Missouri. They all had a birthday today, so add one to their age. (expect 4 matches)
   * db.people.updateMany({ state: ‘Missouri’ }, { $inc: { age: 1} } )
6. Jerry Baker has updated information. Replace with a new document:  
   { first\_name: "Jerry", last\_name: "Baker-Mendez", email: "jerry@classic.ly", gender: "Male", age: 28, state: "Vermont", "children": [{name: "Alan", age: 18}, {name: "Jenny", age: 3}] }
   * db.people.replaceOne( { first\_name: ‘Jerry’ }, {first\_name: ‘Jerry’, last\_name: "Baker-Mendez", email: "jerry@classic.ly", gender: "Male", age: 28, state: "Vermont", "children": [{name: "Alan", age: 18}, {name: "Jenny", age: 3}] }
7. Delete Wanda Bowman.
   * db.people.deleteOne( { first\_name: ‘Wanda’ } )
8. Delete everyone who does not have an email address specified. (expect 36 matches - maybe more depending on what you added above)
   * db.people.deleteMany( { email: null } )

In submissions collection

1. Add several documents to a new submissions collection. Do it all in one command. (Remember, MongoDB will create the collection for you. Just start adding documents.)
   1. title: "The River Bend", upvotes: 10, downvotes: 2, artist: <ID of Anna Howard>
   2. title: "Nine Lives", upvotes: 7, downvotes: 0, artist: <ID of Scott Henderson>
   3. title: "Star Bright", upvotes: 19, downvotes: 3, artist: <ID of Andrea Burke>
   4. title: "Why Like This?", upvotes: 1, downvotes: 5, artist: <ID of Steven Marshall>
   5. title: "Non Sequitur", upvotes: 11, downvotes: 1, artist: <ID of Gerald Bailey>

db.submissions.insertMany( [

{ title: "The River Bend", upvotes: 10, downvotes: 2, artist: ObjectId:

‘60366e3a38a2df0ddbd794f8’ },

{ title: "Nine Lives", upvotes: 7, downvotes: 0, artist: ObjectId:

‘60366e3a38a2df0ddbd79526’ },

{ title: "Star Bright", upvotes: 19, downvotes: 3, artist: ObjectId:

‘60366e3a382df0ddbd795a9’ },

{ title: "Why Like This?", upvotes: 1, downvotes: 5, artist: ObjectId:

‘60366e3a38a2df0ddbd7952f’ },

{ title: "Non Sequitur", upvotes: 11, downvotes: 1, artist: ObjectId:

‘60366e3a38a2df0ddbd794f6’ }

] )

ALTERNATIVELY, you can use a sub-query to search for that and insertMany all in one query

db.submissions.insertMany( [

{ title: "The River Bend", upvotes: 10, downvotes: 2, artist:

db.people.findOne( { first\_name: ‘Anna’, last\_name: ‘Howard’ } ).\_id },

{ title: "Nine Lives", upvotes: 7, downvotes: 0, artist:

db.people.findOne( { first\_name: ‘Scott, last\_name: ‘Henderson’ } ).\_id },

{ title: "Star Bright", upvotes: 19, downvotes: 3, artist:

db.people.findOne( { first\_name: ‘Andrea, last\_name: ‘Burke’ } ).\_id },

{ title: "Why Like This?", upvotes: 1, downvotes: 5, artist:

db.people.findOne( { first\_name: ‘Steven, last\_name: ‘Marshall’ } ).\_id },

{ title: "Non Sequitur", upvotes: 11, downvotes: 1, artist:

db.people.findOne( { first\_name: ‘Gerald, last\_name: ‘Bailey’ } ).\_id },

] }

Graphical user interface, text, application

Description automatically generated

1. Add 2 upvotes for "The River Bend".
   * db.submissions.updateOne({ title: ‘The River Bend’ }, { $inc: { upvotes: 2} } )
2. Add a field round2 = true to all submissions with at least 10 upvotes. (expect 3 matches)
   * db.submissions.updateMany({ upvotes { $gte: 10 } }, { $set: { round2: true } } )

**Extended Challenges:**

1. Update Helen Clark. She had a baby! Add a child, name: Melanie, age: 0
   * db.people.updateOne( { first\_name: ‘Helen’, last\_name: ‘Clark’ }, { $push: { children: { name: ‘Melanie’, age: ‘0’ } } )
2. Joan Bishop has a child named Catherine. She just had a birthday and prefers to go by "Cat". In one query update the child's name to "Cat" and increment her age by one.
   * db.people.updateOne( { first\_name: ‘Joan’, last\_name: ‘Bishop’ }, { $set: { children.3.name: ‘Cat’ } )
   * db.people.updateOne( { first\_name: ‘Joan’, last\_name: ‘Bishop’ }, { $set: { children.3.name: ‘Cat’ }, $inc: { children.3.age: 1 )
3. List all submissions that have more downvotes than upvotes.
   * db.submissions.find( { $expr: { $gt: { ‘downvotes’, ‘upvotes’ } } } )

