

Database - The Magical Marvels of MongoDB Questions

Conjuring MongoDB (Level 1)

01. How would you set the current database to “tapePlayer”?
02. How would you find all the documents inside the “tapes” collection?
03. How would you add a document to the “tapes” collection whose “name” is “Rockin 70s” and whose “creator” is “Big Rig Joe”?
04. How would you query for documents whose “name” is “Cool 60s” in the “tapes” collection?
05. How would you query for documents whose “creator” is “Slim Jim Johnson” in the “tapes” collection?
06. How would you insert the following tape into the “tapes” collection after adding the following name-value pairs?
 - ```
{
 "name": "Best of 90s",
 "creator": "Sweet Kim Kahn"
}
```
  - You need an “age” for this tape, whose value is 20.
  - You need a “price” for this tape, whose value is 19.99.
  - You need a “themes” for this tapes, whose values are “Rock”, “Jazz” and “Hip Hop”, stored as an array.
  - You need a “critics” value, where critics are “official” and “unofficial”. The value of “official” is 6, while the value of “unofficial” is 9.
07. How would you query your “tapes” collection to find documents who have “Rock” in their “themes”?
08. Which validation does the following bad data break?

```
{ "_id": ObjectId("1234567"), "name": "Best of 90s", "creator": "Sweet Kim Kahn", "age": 20, "price": 19.99, "themes": ["Rock", "Jazz", "Hip Hop"], "critics": {"official": 6, "unofficial": 9}},
{ "_id": ObjectId("1234567"), "name": "Best of 2000s", "creator": "Salty Jim Boon", "age": 10, "price": 24.99, "themes": ["Rock", "Hip Hop", "Metal"], "critics": {"official": 7, "unofficial": 8}}
```

A. Unique\_id      B. Valid Syntax      C. Size less than 16 MB

### Mystical Modifications (Level 2)

09. How would you remove the tape with “name” of “Rockin 70s” from our “tapes” collection?
10. How would you remove all tapes that have “Rock” in their “themes” from our “tapes” collection?
11. How would you set the “price” of the tape with “name” of “Rockin 70s” in the “tapes” collection to 14.99?
12. How would you increase the “age” by 2 for all tapes with “Jazz” in their “themes” in the “tapes” collection?
13. How would you increase the view “count” of the tape with “name” of “Cool 60s” in the “logs” collection, and create the document if it doesn’t exist?

14. How would you remove the "taste" field from all documents in the "tapes" collection?
15. How would you change the "creator" field to "writer" for all documents in the "tapes" collection?
16. In the following document, how would you update the value of "Rock" to "Rock and Roll" in the "themes" array, assuming this document is in the "tapes" collection.

```
{
 "name": "Best of 90s",
 "creator": "Sweet Kim Kahn",
 "age": 20,
 "price": 19.99,
 "themes": ["Rock", "Jazz", "Hip Hop"],
 "critics": {"official": 6, "unofficial": 9}
}
```
17. Using the updated document from question 16, how would you update the value of "Jazz" to "Smooth Jazz" in the "themes" field using the positional operator?
18. Using the updated document from question 17, how would you add "Disco" to the end of the "themes" field?
19. How would you add "Disco" to "themes" field in every document in the "tapes" collection if it isn't present already?
20. Using the updated document from question 18, how would you multiply the "official" value by 2?

### **Materializing Potions (Level 3)**

21. How would you find all tapes where the "maker" is "Slim Jim Johnson" and the "age" is 10 that are in the "tapes" collection?
22. How would you find all tapes in the "tapes" collection where "age" is less than or equal to 15?
23. How would you find all tapes in the "tapes" collection that don't have "Rock and Roll" in the "themes" field?
24. How would you find tapes in the "tapes" collection that have "critics.official" value greater than or equal to 2 and less than or equal to 5?
25. All tapes now have a "speeds" field showing a list of speeds the tapes play in, as shown below. How would you find all tapes that contains "speeds" that are greater than or equal to 1.5 but less than 2.5?

```
{
 "name": "Best of 90s",
 "creator": "Sweet Kim Kahn",
 "age": 20,
 "price": 19.99,
```

```

 "themes": ["Rock", "Jazz", "Hip Hop"],
 "critics": {"official": 6, "unofficial": 9}
 "speeds": [1, 1.5, 2, 2.5, 3]
 }

```

26. How would you find tapes in the "tapes" collection whose "maker" is not "Slim Jim Johnson", whose "age" is less than or equal to 10, whose "price" is less than 14.99, and whose "speeds" are 1.5 or 2?
27. How would you query all tapes in the "tapes" collection, but project on the "\_id" and "name" fields, and sort them alphabetically?
28. How would you query all tapes in the "tapes" collection, and project everything except the "\_id", "price", and "speeds" fields?
29. How would you query all tapes in the "tapes" collection, and project the "name" and "themes" fields, but exclude the "\_id" field?
30. How would you count the number of tapes in the "tapes" collection that have an "age" of 5?
31. Finish the following query so that only 5 tapes are returned by the cursor for each page from the "tapes" collection:  
db.find({}).skip(0)
32. How would you write a query to match all tapes in the "tapes" collection, then sort them in ascending order by their "price", and add a cursor method to limit the results to 4 documents?

## Morphing Models (Level 4)

33. Fill in the blank. When we take the data from one document and place it inside another one, that's called an \_\_\_\_\_ document.
34. Fill in the blank. If we take a unique value like an \_id from one document and store it as a value within a related document, we have just created a \_\_\_\_\_ document.
35. Consider the following documents in the "tapes" collection. For what reason might we want to consider referencing "maker" information instead of embedding it within each wand?

|                                                                                         |                                                                                         |                                                                                       |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| {<br>"name": "Rockin 70s",<br>"maker": ["Slim Jim Johnson",<br>"Sweet Timmy Tim:"]<br>} | {<br>"name": "Awsome 80s",<br>"maker": ["Slim Jim Johnson",<br>"Tall Momma June:"]<br>} | {<br>"name": "Cool 90s",<br>"maker": ["Slim Jim Johnson",<br>"Sweet Timmy Tim:"]<br>} |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|

- A. Atomic Writes      B. Duplicate Date      C. Single Query Access
36. What's the minimum number of queries we'd have to write in order to retrieve a document and its referenced data?  
A. 1      B. 2      C. 3

37. Which modeling option would give us all the data we need with a single query, support for atomic writes, and is great for data that is strongly related?  
A. Embedding      B. Referencing
38. Which data modeling decision doesn't have default support for atomic writes across multiple document and be utilized with care?  
A. Embedded      B. Referenced
39. In general, what's the best option to first consider for modeling your related data?  
A. Embedding      B. Referencing
40. Which data modeling option would be the best fit for storing users and their addresses when we know that the data is used together often, won't be changing regularly, and each user will only be storing a few addresses?  
A. Embedding      B. Referencing
41. We'd like to store information about computers, and each computer can have a few hundred parts. Most of the time, we won't be needing specific information about each part. Which data modeling route should we take?  
A. Embedding      B. Referencing
42. Which modeling route is best when we have data that is constantly changing and will help prevent data inconsistencies from arising?  
A. Embedding      B. Referencing
43. Which data modeling route allows us to access our data independently instead of having to use something like dot notation to get information?  
A. Embedding      B. Referencing

### **Aggregation Apparitions (Level 5)**

44. How would you write an aggregate to group "tapes" by their "maker" so we get a list of unique makers?
45. How would you write an aggregate that groups "tapes" by their "critics.unofficial", and add an accumulator with a "tapes\_critiqued" to count the number of critiques per "critics.unofficial" score?
46. How would you write an aggregate that groups our "tapes" by the "maker" field, and add an accumulator with the "total\_cost" field that sums the "price" for each tape per "maker"?
47. How would you write an aggregate that groups "tapes" by their "age", and add an accumulator with a field "price\_averaged" to get the average "price" for the tapes per age?
48. How would you write an aggregate that groups "tapes" by their "maker", add an accumulator with the field "total\_tapes" to sum the number of tapes each "maker" has, add an accumulator with the field "max\_critique" that finds the greatest "critics.official" per "maker", and add an accumulator with the field "lowest\_price" that finds the lowest tapes "price" per maker?

49. How would you write an aggregate that will only match tapes that have "Rock and Roll" in their "themes", then another stage to group the data by their "maker", then an accumulator with a field name "lowest\_age" that finds the lowest "age" per "maker"?
50. How would you write an aggregate that matches tapes that have a "price" that is less than 50, then a stage to group the tapes by their "maker", then a accumulator with a field name "average\_critique" to find the average "critics.official" per "maker", then after the existing stages, add another match stage that retrieves results with an "average\_critique" that is greater than 40?
51. How would you write an aggregate that finds tapes that have an "age" that's less than or equal to 5, then add a stage to group tapes by their "maker", then add an accumulator with the field "max\_critique" that finds the max "critics.unofficial" per "maker", then add a stage to sort the "max\_critique" in descending order, the add a stage to limit only the first 4 results, then add a stage in the correct place to project only the "maker" and "critics.unofficial" fields?