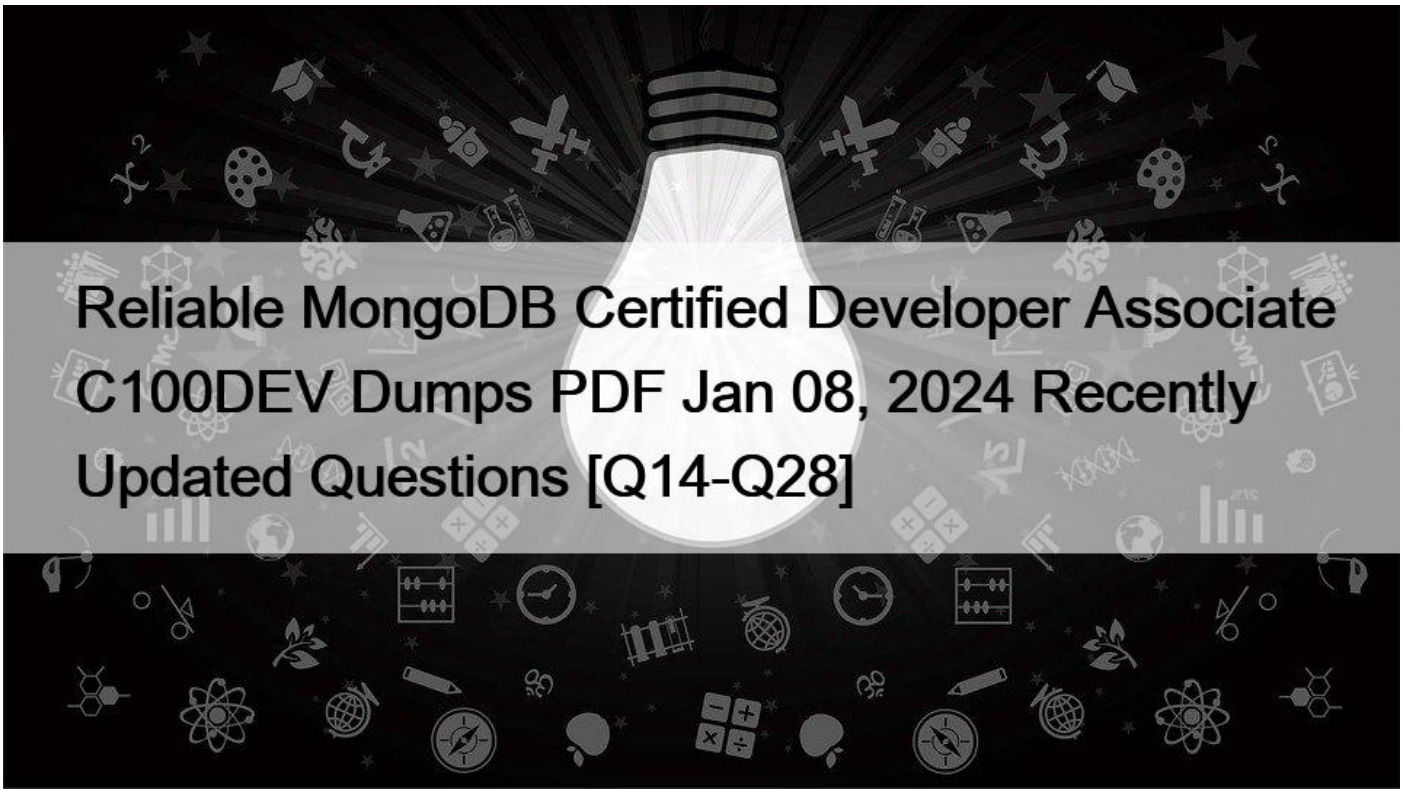
Reliable MongoDB Certified Developer Associate C100DEV Dumps PDF Jan 08, 2024 Recently Updated Questions [Q14-Q28



Reliable MongoDB Certified Developer Associate C100DEV Dumps PDF Jan 08, 2024 Recently Updated Questions
Pass Your MongoDB C100DEV Exam with Correct 253 Questions and Answers

QUESTION 14

We have the following schema for a movies collection: { _id: ObjectId, title: String, genres: Array, languages: Array, year: 32-bit integer } And the following index on the movies collection: { title: 1 } Which of the following queries will use the given index to perform the sorting stage?
* db.movies.find({ genres: “Drama” }).sort({ year: 1 })
* db.movies.find({}).sort({ title: -1 })
* db.movies.find({}).sort({ title: 1 })
* db.movies.find({ genres: “Drama” }).sort({ title: 1 })
db.movies.find({ genres: “Drama” }).sort({ title: 1 }) Yes, in this case the index will be used to retrieve the sorted documents and then it will filter the movies matching the genre. https://docs.mongodb.com/manual/indexes/

QUESTION 15

Data modeling. You are considering the use of nesting or references in your data model. Documents can grow rapidly. Which option would be more appropriate?
* Use of nesting.
* Use of reference.

https://docs.mongodb.com/manual/core/data-modeling-introduction/

QUESTION 16

In which situations can we consider sharding?
* We want to improve read performance for our application.
* It takes too long to backup and restore the data set.
* A new developer will join your development team soon.
* Our dataset is too big to fit in one MongoDB instance.
https://docs.mongodb.com/manual/sharding/

QUESTION 17

Select all true statements regarding to transactions in MongoDB.
* Transactions are associated with a session.
* We cannot read/write to collections in the config, admin, or local databases.
* The collections used in a transaction can be in different databases.
https://docs.mongodb.com/manual/core/transactions/

QUESTION 18

Select all true statements about query plans in MongoDB. (select 2)
* For a given query, each index in the collection generates at least one query plan.
* Query plans are cached so that plans don’t have to be generated and compared with each other each time a query is executed.
* If there are no indexes for the query, the main stage in the query plan will be the COLLSCAN stage.
https://docs.mongodb.com/manual/core/query-plans/

QUESTION 19

We have a movies collection with the following document structure: { _id: ObjectId(“573a1390f29313caabcd6223”), genres: [‘Comedy’, ‘Drama’, ‘Family’], title: ‘The Poor Little Rich Girl’, released: ISODate(“1917-03-05T00:00:00.000Z”), year: 1917, imdb: { rating: 6.9, votes: 884, id: 8443 } } We need to use Aggregation Framework to fetch all movies from this collection where ‘Drama’ is in genres list and the minimum ‘imdb.votes’ is at least 100. Additionally, in the projection stage, we want to leave only the following fields: -> title -> genres -> imdb.votes We also want to sort the result set by decreasing imdb votes. Example output: [{ imdb: { votes: 1521105 }, genres: [‘Crime’, ‘Drama’], title: ‘The Shawshank Redemption’ }, { imdb: { votes: 1513145 }, genres: [‘Crime’, ‘Drama’], title: ‘The Shawshank Redemption’ }, { imdb: { votes: 1495351 }, genres: [‘Action’, ‘Crime’, ‘Drama’], title: ‘The Dark Knight’ }, … Which pipeline should you use?
* [{ $match: { genres: { $in: [‘Drama’] }, ‘imdb.votes’: { $gte: 100 } } }, { $project: { _id: 0, title: 1, genres: 1, ‘imdb.votes’: 1 } }, { $sort: { ‘imdb.votes’: -1 } }]
* [{ $match: { genres: { $in: [‘Drama’] }, ‘imdb.votes’: { $gte: 100 } } }, { $project: { _id: 0, title: 1, genres: 1, ‘imdb.votes’: 1 } }, { $sort: { ‘imdb.votes’: 1 } }]
* [{ $match: { genres: { $in: [‘Drama’] }, ‘imdb.votes’: { $gte: 100 } } }, { $project: { _id: 0, title: 1, genres: 1, ‘imdb.votes’: 1 } }, { $limit: { ‘imdb.votes’: -1 } }]
https://docs.mongodb.com/manual/reference/method/db.collection.aggregate/

QUESTION 20

Select all valid BSON types in MongoDB. (select 4)
* ObjectId
* Boolean
* String
* Array
* Dictionary
https://docs.mongodb.com/manual/reference/bson-types/

QUESTION 21

Select true statements about sorting & indexing performance.
* Index prefixes cannot be used in query predicates to increase index utilization.
* Indexes can be traversed both forward and backward.
* Index prefixes can be used in sort predicates to prevent sort in memory.
* It’s possible to have a sorted query use an index for both sorting and filtering.
https://docs.mongodb.com/manual/tutorial/sort-results-with-indexes/

QUESTION 22

Can arrays in MongoDB store values of different data types?
* No
* Yes

QUESTION 23

Select all true statements about data modeling in MongoDB.
* MongoDB can easily handle both unstructured and structured data sets.
* Unlike SQL databases, where the table schema must be specified and declared before inserting the data, MongoDB collections don’t, by default, require documents to have the same schema.
* The field set and data type for each field can differ across documents within a collection.
* In MongoDB we can enforce document validation rules for a collection during update and insert operations.
https://docs.mongodb.com/manual/core/data-modeling-introduction/

QUESTION 24

Suppose you have a products collection with an index: { product_category: 1 } For which of the following queries can MongoDB look at only a subset of the index entries, rather than all of the index entries?
* db.products.find({ product_category: /A./ })
* db.products.find({ product_category: /S./ })
* db.products.find({ product_category: /A./ })
https://docs.mongodb.com/manual/reference/operator/query/regex/

QUESTION 25

Select all true statement regarding to the MongoDB (BSON, JSON). (select 3)
* MongoDB stores data in BSON, and we can view it in JSON.
* MongoDB stores data in JSON, and we can view it in BSON.
* BSON supports as many data types as JSON.
* BSON supports more data types than JSON.
* BSON is faster to parse and lighter to store than JSON.

https://www.mongodb.com/json-and-bson

QUESTION 26

Which of the following built-in roles provides the greatest cluster management access?
* clusterManager
* clusterMonitor
* clusterAdmin

Explanation: https://docs.mongodb.com/manual/reference/built-in-roles/#mongodb-authrole-clusterAdmin

QUESTION 27

It’s very important to picking a good shard key. If we don’t choose a good shard key, then we won’t be able to see the benefits of horizontal scaling.
* False
* True

https://docs.mongodb.com/manual/core/sharding-shard-key/

QUESTION 28

Which cursor method should you use to specify the maximum number of documents returned?
* cursor.map()
* cursor.limit()
* cursor.skip()
* cursor.hint()
* cursor.count()

https://docs.mongodb.com/manual/reference/method/cursor.limit/

MongoDB C100DEV Certification Exam is a valuable resource for developers looking to showcase their skills and expertise in database development using MongoDB. C100DEV exam covers a variety of topics, ranging from basic to advanced, and it requires a specific level of knowledge to complete successfully. Individuals who pass the exam receive certification, which is valid for two years, and it is an excellent way to show potential employers that the developer has the skills and knowledge needed to work with MongoDB effectively. C100DEV exam is available worldwide, and it is administered through the MongoDB certification portal.

Latest 2024 Realistic Verified C100DEV Dumps: https://www.vceprep.com/C100DEV-latest-vce-prep.html]