

Prova P1 2019-1

1.

Porque é necessário denormalizar as tabelas de dimensão num sistema de BI e porque isto não fere a integridade referencial do sistema.

Porque normalizamos um sistema operacional tradicional.

Para otimizar agregações tirando maior quantidade de joins, porque o sistema de BI é Ready Only, para não ter dados duplicados no sistema.

2. Como é feita a persistência de hierarquias de clientes corporativos e atributos multivalorados em bancos de dados relacionais.

Tabela ponte.

3. Cite alguns motivos pelos quais é difícil a persistência de dados de redes sociais em bancos de dados relacionais.

Schema flexível e dinâmico. Totalmente aleatório.

4. Cite dois objetivos pelos quais bancos de dados semi-estruturados são procurados pelas empresas nos dias atuais.

Schema flexível e dinâmico e escalabilidade horizontal.

5. Diferencie escalabilidade horizontal e vertical para bancos de dados. Exemplifique.

Escalabilidade horizontal: BD distribuído em unidades; baratas, custo da aplicação (PC commodity) Facebook, Google, Amazon, etc.

Escalabilidade vertical: servidor monolítico (banco tradicional). Banco do Brasil, Itaú

6. Bancos não relacionais são antigos, citando como exemplo IMS da década de 60. Cite alguns motivos pelos quais bancos NoSQL ganharam tanto destaque nos dias de hoje.

Schema Flexível e dinâmico.

Escalabilidade horizontal.

Open Source

Atendem necessidades específicas.

7. O que é consistência eventual. Qual é uma tradução errada em pt-br que deve ser evitada.

O que eu gravo num nó primário, em instantes será replicado nos nós secundários. Com certeza será replicado.

A tradução errada em pt-br é PODE NÃO GRAVAR

8. Enuncie o teorema CAP.

Consistência

Disponibilidade

Particionamento

Não é possível, num BD distribuído, no mesmo instante, termos as 3 características.

9. Porque é difícil comparar bancos NoSQL entre si.

Porque atendem necessidades muito específicas.

10. Dizemos que a fazemos a modelagem no banco relacional Bottom Up e no MongoDB Top Down. Explique

Bottom UP no banco relacional partimos das tuplas em direção ao objeto usando ORM (Mapeamento Objeto Relacional)

Top Down porque a partir do uso do dado você define o documento (JSON)

11. O MongoDB não aceita “transações”, porém ele é muito utilizado. Em que cenários ele é competitivo e em quais ele não é recomendável.

MongoDB é recomendável para aplicações pequenas ou com dados bem definidos, não é recomendável para sistemas grandes e genericos, com sistemas de cartão de crédito.

12. O que é um upsert.

Update ou inserte. Se o documento já existe, atualiza, se não existe, então insere.

13. Disserte sobre Schema Design no MongoDB. 1:1, 1:N e N:N.

1:1

RH:CV

Se o CV (currículo) é grande ou pouco usado, então fazemos documentos separados, caso contrário insiro em RH (Recursos Humanos).

1:N – temos dois casos N grande ou pequeno

Post:comentários

Como a cardinalidade dos comentários é baixa, então inserimos os comentários como uma lista dentro do Posto.

SJC:habitantes

Nesse caso é melhor separar, porque não vou inserir 700K na cidade.

N:N

Livros : Autores

Como as cardinalidades são baixas, insiro um dentro do outro.

14. O que é o sharding no MongoDB. Isso pressupõe algum gargalo no banco de dados.

É o particionamento horizontal da collection do MongoDB. Tem um gargalo que é o MongoS, que distribui as requisições.

15. Cite duas vantagens de um banco colunar para agregações massivas de um sistema de BI.

1) Não preciso fazer FULL SCAN da linha toda.

2) A taxa de compactação de dados é maior.

3) Posso ter índices nas colunas mais usadas.

16. Fale sobre as opções de persistência do Redis. O que significa in memory database.

1) Sem persistencia, o que você grava não espera a respostas.

2) A cada x writes ou y milisegundos eu atualizo a base.

3) Guarda cópia de todo write num append log

In memory database é sua base toda na memória

17. Qual é uma grande desvantagem do MongoDB em relação ao CouchDB. Explique.

MongoDB não tem consistencia forte, como o CouchDB que é ACID. É uma escolha ter consistência forte, ao custo de ser mais lento.

18. No Redis explique a vantagem de inserts TTL. Como é feito o namespace dos databases.

TTL = Time To Live, consigo gravar algo com tempo de vida pré-definido. O namespace é apenas um índice inteiro: 0, 1, 2 ...

19. No MongoDB qual é a função do `_id`. Vantagem da collection ser schema free.

`_id` é a chave única do documento no cluster.

Flexibilidade e rapidez no desenvolvimento.

20. No MongoDB o que são upserts. O que são “fire-and-forget functions” e sua vantagem.

Upsert update insert, se o documento existe atualiza, caso contrário insere o server não retorna se gravou, muito rápido.

21. No MongoDB exemplifique a vantagem de `findOne`. O que são `hint` e `explain`.

`Findone` encontra o primeiro documento rápido no cluster.

`Hint` = dá a sequência de índices a serem usados na próxima query.

`Explain` mostra o log da última query executada.

22. Explique o que significa ser “linearmente escalável” no caso do Cassandra. Qual é a limitação deste NoSQL.

A disponibilidade aumenta proporcionalmente ao número de nós, se eu dobrar, dobra a disponibilidade. Cassandra é eventualmente consistente (o que gravo no primário demora uns instantes para chegar em todos os secundários).

23. O que é BASE e qual sua relação com ACID.

BASE é a relaxação do ACID para ter mais disponibilidade.

24. Exemplifique uma situação onde bancos NoSQL não são adequados.

Sistemas grandes e genéricos.

Questões 25 a 41 são respondidas nos cursos M001 e M220P no MongoDB University

25. Seja um exemplo de documento da collection `username` do curso `{ "_id": "fmasanori", "password": "abacate" }`. Seguindo a lógica do exemplo, insira um usuário genérico `"username"`, com senha `"1234"`.

26. Seja a collection `username`, de acordo com o documento exemplo. Execute uma busca de todos os usuários em ordem crescente de `username`, sem mostrar o campo `"password"`.

27. When connecting to an Atlas cluster using the shell, why do we provide the hostnames for all nodes when we launch mongo?

28. O que esse comando faz? `db.movieDetails.find({"awards.wins": 2, "awards.nominations": 2}).count()`

29. Como será a query “How many movies in the `movieDetails` collection list “Western” second among its genres”?

30. In `ships.shipwrecks` collection, `watlev` describes the water level at the shipwreck site and `depth` describes how far below sea level the ship rests. How many documents in the `ships.shipwrecks` collection match either of the following criteria: `watlev` equal to `"always dry"` or `depth` equal to 0. Escreva com será a query.

31. 100YWeatherSmall.data collection. The sections field in this collection identifies supplementary readings available in a given document by a three-character code. How many documents list: "AG1", "MD1", and "OA1" among the codes in their sections array.
Your count should include all documents that include these three codes regardless of what other codes are also listed.

32. 100YWeatherSmall.data collection. How many documents in this collection contain exactly two elements in the sections array field?

33. How many documents in the citibike.trips collection have the key tripduration set to null? Ignore any documents that do not contain the tripduration key.

34. video.movies collection. How many movies match the following criteriaÇ: The cast includes either of the following actors: "Jack Nicholson", "John Huston". The viewerRating is greater than 7. The mpaaRating is "R".

35. Faça a query que procure os documentos na collection video.movies que possuem "Patty Jenkins" no campo director

36. Faça uma query que procure os documentos na collection citibike.trips que possuem tripduration >= 60 e <65

37. Dentro da collection video.movieDetails conte quantos documentos possuem genres do tipo "Comedy"

38. Dentro da collection video.movieDetails conte quantos documentos possuem exatamente 2 award wins e 2 award nominations

39. Dentro da collection video.movieDetails conte quantos documentos possuem rated "PG" e exatamente 10 award nominations

40. Dentro da collection video.movieDetails conte quantos documentos possuem "Western" como segundo na lista genres

41. Diga quais inserts serão executados nos comandos abaixo:

```
requests = [
InsertOne({ '_id': 11, 'name': 'Edgar Martinez', 'salary': "8.5M" }),
# Insert #1
InsertOne({ '_id': 3, 'name': 'Alex Rodriguez', 'salary': "18.3M" }),
# Insert #2
InsertOne({ '_id': 24, 'name': 'Ken Griffey Jr.', 'salary': "12.4M" }), # Insert #3
InsertOne({ '_id': 11, 'name': 'David Bell', 'salary': "2.5M" }),
# Insert #4
InsertOne({ '_id': 19, 'name': 'Jay Buhner', 'salary': "5.1M" })
# Insert #5
]
response = employees.bulk_write(requests)
```

