

LAB-BD - 03/03/22

bit.ly/PPZNO SQL

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.
2. Como é feita a persistência de hierarquias de clientes corporativos e atributos multivalorados em bancos de dados relacionais.
3. Cite alguns motivos pelos quais é difícil a persistência de dados de redes sociais em bancos de dados relacionais.
4. Cite dois objetivos pelos quais bancos de dados semi-estruturados são procurados pelas empresas nos dias atuais.
5. Diferencie escalabilidade horizontal e vertical para bancos de dados. Exemplifique.
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.
7. O que é consistência eventual. Qual é uma tradução errada em pt-br que deve ser evitada.
8. Enuncie o teorema CAP.
9. Porque é difícil comparar bancos NoSQL entre si.
10. Dizemos que a fazemos a modelagem no banco relacional Bottom Up e no MongoDB Top Down. Explique.
11. O MongoDB não aceita "transações", porém ele é muito utilizado. Em que cenários ele é competitivo e em quais ele não é recomendável.
12. O que é um upsert.
13. Disserte sobre Schema Design no MongoDB. 1:1, 1:N e N:N.
14. O que é o sharding no MongoDB. Isso pressupõe algum gargalo no banco de dados.
15. Cite duas vantagens de um banco colunar para agregações massivas de um sistema de BI.
16. Fale sobre as opções de persistência do Redis. O que significa in memory database.
17. Qual é uma grande desvantagem do MongoDB em relação ao CouchDB. Explique.
18. No Redis explique a vantagem de inserts TTL. Como é feito o namespace dos databases.
19. No MongoDB qual é a função do _id. Vantagem da collection ser schema free.
20. No MongoDB o que são upserts. O que são "fire-and-forget functions" e sua vantagem.
21. No MongoDB exemplifique a vantagem de findOne. O que são hint e explain.
22. Explique o que significa ser "linearmente escalável" no caso do Cassandra. Qual é a limitação deste NoSQL.
23. O que é BASE e qual sua relação com ACID.
24. Exemplifique uma situação onde bancos NoSQL não são adequados.
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 criteria: 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)
```

Prova P1:

1. Para otimizar agregações tirando a maior quantidade de joins. Porque o sistema de BI é read-only. Para não ter dados duplicados no sistema.
2. Tabela Ponte.
3. Schema flexível e dinâmico.
4. Schema flexível e dinâmico, escalabilidade horizontal.
5. Escalabilidade horizontal: BD distribuído em unidades baratas (like PC)
Escalabilidade vertical: servidor monolítico.

Facebook, Google, Amazon, etc - Horizontal.

Mainframe de instituição financeira tradicional: servidor monolítico vertical.

6. - Schema Flexível e Dinâmico;
- Escalabilidade Horizontal;
- Open Source;
- Atendem necessidades específicas.
7. 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. Consistency, Availability, Partition.
Não é possível, num BD distribuído, no mesmo instante, termos as 3 características.
9. Porque atendem necessidades muito específicas.
10. 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 é recomendável para aplicações pequenas ou com dados bem definidos, não é recomendável para sistemas grandes e genéricos, como sistemas de cartão de crédito.
12. Update ou insert. Se o documento já existe, atualiza, se não existe, então insere.
13. 1:1
RH : CV
Se o CV é grande ou pouco usado, então fazemos documentos separados, caso contrário insiro em RH.

1:N Temos dois casos: N grande ou pequeno

Post : Comentários

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

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. É o particionamento horizontal da collection do MongoDB. Tem um gargalo que é o MongoS, que distribui as requisições.
15. Não preciso fazer o FULL SCAN da linha toda, a taxa de compactação de dados é maior. Posso ter índices das colunas mais usadas.

16. 1. Sem persistência, o que você grava não espera as respostas.
2. A cada x writes ou y milissegundos eu atualizo a base.
3. Guarda cópia de todo write num append log
In memory database é sua base toda na memória.
17. MongoDB não tem consistência forte, como o CouchDB, que é ACID. É uma escolha ter consistência forte, ao custo de ser mais lento.
18. TTL = Time To Live, consigo gravar algo com tempo de vida pré-definido. O namespace é apenas um índice inteiro: 0 1 2 ...
19. _id é a chave do documento no cluster. Flexibilidade e rapidez no desenvolvimento.
20. upsert = update insert, se o doc existe, atualiza, caso contrário insere. O server não retorna se gravou, muito rápido.
21. findOne encontra o primeiro documento na rápido no cluster, hint = dá a sequência de índices a serem usados na próxima query, explain mostra o log da última query executada.
22. A disponibilidade aumenta proporcionalmente ao número de nós, se eu dobrar, dobra a disponibilidade. Cassandra é eventualmente consistente (o que grava no primário demora uns instantes para chegar em todos os secundários).
23. Base é a relaxação do ACID para ter mais disponibilidade.
24. Sistemas grandes e genéricos.