

Curso: Engenharia de Software		Série: 6S	Turma: A	Turno: Noturno	
Professor: Aparecido Vilela Junior		Horário: 19:00 - 22:30			
Acadêmico (a):				RA:	
Disciplina: Banco de Dados NoSql				Data:	
Prova	Prova Prática	Atividades de estudo programadas (AEP)	Prova integrada	Nota final do bimestre	
	8,0	1,0	1,0	10,0	
INSTRUÇÕES PARA REALIZAÇÃO DA PROVA: <ul style="list-style-type: none"> ⇒ Os dados do cabeçalho deverão ser preenchidos com letra maiúscula. E as questões deverão ser respondidas com letra legível. ⇒ É vedado, durante a prova, o porte e/ou o uso de aparelhos sonoros, fonográficos, de comunicação ou de registro eletrônico ou não, tais como: notebooks, celulares, tablets e similares. ⇒ A prova é individual e sem consulta, deverá ser respondida a caneta azul ou preta. Prova escrita a lápis não dá direito à revisão. Não é permitido o uso de corretivo. ⇒ É obrigatória a permanência do acadêmico 1 (uma) hora em sala de aula após o início da prova. ⇒ Não será permitida a entrada na sala de aula após 10 minutos do início da prova. ⇒ É obrigatória a assinatura da lista de presença impressa na qual constam RA, nome e curso. ⇒ O valor de cada questão está ao lado da mesma. ⇒ Todas as respostas devem constar no espaço destinado e autorizado pelo professor, à resposta. ⇒ Em caso de qualquer irregularidade comunicar ao Professor ou fiscal de sala. ⇒ Ao término da prova, levante o braço e aguarde atendimento do professor ou do fiscal. 					
1ºbim.		2ºbim.		1ªsub.	
				2ªsub.	

ENTREGA DA PROVA PRÁTICA

Gerar um pdf (Salvar como ou Exportar para Pdf) e enviar nesse link o arquivo anexado

https://docs.google.com/forms/d/e/1FAIpQLSfeKV-5GUvLWuqZ6Ozjezo-ngWwaYHEtx04QaLvGWzps05Ebw/viewform?usp=pp_url

Instrução: Copie e cole todo o comando abaixo no seu MongoDB Shell e execute-o.

```
db.campeoes.drop();
db.campeoes.insertMany([
  { nome: "Garen", regioao: "Demacia", funcao: "Lutador", titulo: "O Poder de Demacia", stats: { vida: 620, dano: 66 }, skins: ["Reinos Combatentes", "Deus-Rei"], recurso: "Nenhum" },
  { nome: "Lux", regioao: "Demacia", funcao: "Mago", titulo: "A Dama da Luz", stats: { vida: 560, dano: 54 }, skins: ["Elementalista", "Guardiã Estelar"], recurso: "Mana", habilidades: [{ nome: "Singularidade Lucente", tipo: "Mágico", dano_base: 280 }, { nome: "Centelha Final", tipo: "Mágico", dano_base: 600 } ] },
  { nome: "Ashe", regioao: "Freljord", funcao: "Atirador", titulo: "A Arqueira do Gelo", stats: { vida: 570, dano: 61 }, skins: ["PROJETO", "Campeonato"], recurso: "Mana" },
  { nome: "Tryndamere", regioao: "Freljord", funcao: "Lutador", titulo: "O Rei Bárbaro", stats: { vida: 625, dano: 68 }, skins: ["Sultan", "Reinos Combatentes"], recurso: "Fúria" },
  { nome: "Darius", regioao: "Noxus", funcao: "Lutador", titulo: "A Mão de Noxus", stats: { vida: 652, dano: 64 }, skins: ["Rei dos Bretões", "Deus-Rei"], recurso: "Mana" },
  { nome: "Katarina", regioao: "Noxus", funcao: "Assassino", titulo: "A Lâmina Sinistra", stats: { vida: 602, dano: 60 }, skins: ["Gatinha", "Reinos Combatentes"], recurso: "Nenhum" },
  { nome: "Jinx", regioao: "Zaun", funcao: "Atirador", titulo: "O Gatilho Desenfreado", stats: { vida: 610, dano: 57 }, skins: ["Guardiã Estelar", "Arcane"], recurso: "Mana", habilidades: [{ nome: "Zap!", tipo: "Físico", dano_base: 210 }, { nome: "Super Mega Míssil da Mortel", tipo: "Físico", dano_base: 450 } ] },
  { nome: "Ekko", regioao: "Zaun", funcao: "Assassino", titulo: "O Rapaz que Estilhou o Tempo", stats: { vida: 585, dano: 58 }, skins: ["PROJETO", "True Damage"], recurso: "Mana" },
  { nome: "Yasuo", regioao: "Ionia", funcao: "Lutador", titulo: "O Imperdoável", stats: { vida: 590, dano: 60 }, skins: ["PROJETO", "Florescer Espiritual"], recurso: "Fluxo" },
  { nome: "Ahri", regioao: "Ionia", funcao: "Mago", titulo: "A Raposa de Nove Caudas", stats: { vida: 570, dano: 53 }, skins: ["Guardiã Estelar", "K/DA"], recurso: "Mana" },
  { nome: "Leona", regioao: "Targon", funcao: "Tanque", titulo: "A Alvorada Radiante", stats: { vida: 646, dano: 60 }, skins: ["PROJETO", "Eclipse Solar"], recurso: "Mana" },
  { nome: "Thresh", regioao: "Ilhas das Sombras", funcao: "Suporte", titulo: "O Guardião das Correntes", stats: { vida: 561, dano: 56 }, skins: ["Florescer Espiritual", "Estrela Negra"], recurso: "Mana", build_sugerida: ["Medalhão dos Solari de Ferro", "Juramento do Cavaleiro", "Redenção"] },
  { nome: "Miss Fortune", regioao: "Águas de Sentina", funcao: "Atirador", titulo: "A Caçadora de Recompensas", stats: { vida: 570, dano: 52 }, skins: ["Fliperama", "Guardiã Estelar"], recurso: "Mana", build_sugerida: ["A Coletora", "Mata-Cráquens", "Lâmina da Fúria de Guinsoo"] },
  { nome: "Zed", regioao: "Ionia", funcao: "Assassino", titulo: "O Mestre das Sombras", stats: { vida: 584, dano: 63 }, skins: ["PROJETO", "Galáxias"], recurso: "Energia", habilidades: [{ nome: "Shuriken Lâmina Sombria", tipo: "Físico", dano_base: 270 }, { nome: "Marca Fatal", tipo: "Físico", dano_base: 500 } ] },
  { nome: "Vi", regioao: "Piltover", funcao: "Lutador", titulo: "A Defensora de Piltover", stats: { vida: 645, dano: 63 }, skins: ["Arcane", "Reinos Combatentes"], recurso: "Mana", build_sugerida: ["Ruptor Divino", "Cutelo Negro", "Hexdrinker"] }
]);
```

1) (Pontos 0,5). O comando find() é a principal forma de consultar dados em uma coleção. Um filtro simples, { campo: valor }, permite localizar documentos com base numa correspondência exata, sendo a operação de leitura mais fundamental em qualquer sistema de gestão de dados (CHODOROW, 2019). A equipa de narrativa do jogo precisa frequentemente de rever os detalhes completos de um campeão específico para garantir a consistência da sua história.

Com base nessas informações, **ESCREVA** o comando que encontra o documento completo do campeão cujo nome é "Yasuo".

Resposta:

```
db.campeoes.findOne(  
  
  {nome: "Yasuo"}  
  
)
```

2) (Pontos 1,0). O comando deleteMany() remove todos os documentos de uma coleção que correspondem a um filtro específico. É uma operação poderosa para limpezas de dados em massa, sendo essencial para a manutenção do ciclo de vida dos dados (CHODOROW, 2019). No universo do jogo, foi decidido que os campeões que usam "Fúria" como recurso serão redesenhados e movidos para um novo sistema.

Com base no exposto, **FORMULE** o comando para remover todos os campeões cujo campo recurso seja "Fúria" E que possuem no seu array de skins pelo menos uma das seguintes skins: "Guardiã Estelar" ou "PROJETO: Yasuo". **O campo _id não deve ser incluído no resultado.**

Resposta:

```
db.campeoes.find(  
  {recurso:"Fúria", skins: {$in:["Guardiã Estelar", "PROJETO: Yasuo"]}},  
  {_id: 0}  
)
```

3.(Pontos 1,0). Para realizar consultas mais complexas, o MongoDB oferece operadores como o \$in, que seleciona documentos cujo valor de um campo corresponde a qualquer valor num array especificado, e o \$gt (greater than), que encontra documentos onde um valor numérico é maior que o especificado. A combinação destes operadores permite criar filtros poderosos (MONGODB, 2024).

A equipe de balanceamento precisa de um relatório sobre magos de alto dano no final do jogo. O relatório deve listar apenas os campeões que são magos de uma lista específica e que possuem um poder de habilidade final muito elevado, para análise de possíveis ajustes.

De posse do conhecimento obtido, **DEMONSTRE** a consulta que retorna **apenas o nome** e o **título** dos campeões (sem o _id) e que o comando find **retorne** os campeões cujos nomes estão na lista ["Ahri", "Lux", "Veigar"] E que possuem um poder de habilidade final (ap_final) superior a 900.

Resposta:

```
db.campeoes.find(  
  
  {nome: {$in: ["Ahri", "Lux", "Veigar"]}, ap_final:{$gt:900}},  
  
  {nome: 1, titulo: 1, _id: 0}  
  
)
```

4) (**Pontos -> 0,5**) Para modificar campos dentro de um subdocumento, utiliza-se a "dot notation" ("campo_pai.campo_filho"). O operador de atualização \$inc é usado para incrementar (ou decrementar) um valor numérico atômicamente, sendo a forma recomendada para atualizar contadores e estatísticas. A campeã "Jinx" recebeu um buff (melhoria) no seu dano de ataque base.

Pautando-se nessas informações, **APLIQUE** o comando que atualiza o documento da "Jinx", incrementando o valor do campo dano (dentro do subdocumento stats) em 3 pontos.

Resposta:

```
db.campeoes.updateOne(  
  
  {nome: "Jinx"},  
  
  {$inc:{"stats.dano": 3}}  
  
)
```

5) (**Pontos 1,5**). O comando `createCollection` permite definir um validator usando a sintaxe `$jsonSchema`. Esta funcionalidade é crucial para garantir a integridade e a consistência dos dados, agindo como as constraints de um banco de dados relacional. A equipa de desenvolvimento precisa de criar uma nova coleção chamada `itens` para armazenar os itens do jogo, e é fundamental que cada item inserido siga um padrão rigoroso para evitar bugs.

A partir dessas informações, **ELABORE** o comando para criar a coleção `itens`, aplicando as seguintes regras de validação:

O documento deve ser um objeto.

Os campos `e` (string e de tamanho mínimo com 3 caracteres e no máximo 80) e `custo` (int) são obrigatórios.

O campo `tipo` (string) é opcional, mas se existir, só pode ter os valores "Ataque", "Defesa" ou "Utilidade".

O campo custo deve ser um número inteiro com valor mínimo de 50.

Deve ter um campo tags do tipo array, com numero de itens máximo 3.

Resposta:

```
db.createCollection("itens",{
    validator:{
        $jsonSchema:{
            bsonType: "object",
            required:["nome_item", "custo"],
            properties:{
                nome_item:{bsonType:"string", minLength:3, maxLength:80},
                custo: {bsonType: "int", minimum:50},
                tipo: {bsonType: "string", enum:["Ataque", "Defesa",
"Utilidade"]},
                tags: {bsonType: "array", maxItems: 3}
            }
        }
    }
})
```

6(Pontos1,0). O operador de atualização \$pull remove de um array todos os elementos que correspondem a uma condição especificada. É a ferramenta ideal para remover itens de uma lista dentro de um documento com base em critérios, sendo essencial para manter a consistência de dados embutidos (BANKER, 2016). O item "Hexdrinker" foi considerado muito fraco e será removido do jogo.

Considerando o cenário, **PRODUZA** um comando que remova o item "Hexdrinker" da lista build_sugerida de todos os campeões que o possuem.

Resposta:

```
db.campeoes.updateMany(  
  
  {},  
  
  {$pull: {build_sugerida: "Hexdrinker"}}  
  
)
```

7) (**Pontos 1,0**). O operador de consulta \$elemMatch é usado para encontrar documentos que contêm um campo de array onde pelo menos um subdocumento corresponde a múltiplas condições simultaneamente. É essencial para consultas precisas em arrays de objetos, evitando falsos positivos que poderiam ocorrer ao consultar os campos do subdocumento separadamente (CHODOROW, 2019).

Partindo deste entendimento, **CONSTRUA** uma consulta find que utilize \$elemMatch para encontrar todos os campeões que possuem uma habilidade de dano "Físico" com um dano_base superior a 400.

Resposta:

```
db.campeoes.find(  
  
  {habilidades: {$elemMatch: {tipo: "Físico", dano_base: {$gt: 400}}}}  
  
)
```

8) (**Pontos 1,5**) A combinação de operadores lógicos como \$or (para condições alternativas) e \$and (para condições simultâneas, muitas vezes implícito) permite construir consultas complexas para relatórios analíticos, que são a base da tomada de decisão em sistemas de BI e balanceamento de jogos.

Para construir consultas altamente específicas, é comum aninhar operadores lógicos. O operador \$and pode ser usado explicitamente para garantir que um conjunto de condições de alto nível seja satisfeito, enquanto um operador \$or, aninhado dentro dele, pode ser usado para especificar um subconjunto de condições alternativas (MONGODB, 2024).

A equipe de balanceamento precisa de analisar os campeões de elite da região de Noxus. Especificamente, eles procuram por campeões que são de "Noxus" E que atendem a um dos seguintes perfis de combate: são "Assassinos" com mais de 55 de dano base OU são "Lutadores" com mais de 650 de vida base.

De posse do conhecimento obtido, **FORMULE** a consulta que encontra todos os campeões que são da região de "Noxus" e que satisfazem uma das duas condições de função/stats mencionadas.

Resposta:

```
db.campeoes.find(
```

```
  {regiao:"Noxus", $or:[{funcao: "Assassino", "stats.dano":{$gt:55}},{funcao:  
"Lutador", "stats.vida":{$gt:650}}]}
```

```
)
```