

O Balanceamento de Runeterra (CRUD com Campeões de LoL)

Saudações, invocadores e desenvolvedores! Hoje, vocês foram recrutados para a equipe de balanceamento de League of Legends. Nossa principal ferramenta não será o martelo do Ornn, mas sim o **MongoDB**. Nossa missão é usar o poder do CRUD para lançar novos campeões, ajustar seus poderes (buffs e nerfs), catalogar skins e garantir que o meta de Summoner's Rift permaneça justo e divertido.

Abram seus **MongoDB Compass**. A arena nos aguarda!

Passo 1: Carregando os Campeões no Nexus (Setup do Banco de Dados)

Todo jogo precisa de seus personagens. Vamos carregar os dados iniciais dos nossos campeões.

1. Selecione o Banco de Dados do Jogo:

use summoners_rift

Crie a Coleção campeoes com os Primeiros Lutadores:

Download do arquivo campeoes.json no Studeo.

Copiem e cole este comando no Shell para popular nossa coleção.

Reparem na estrutura de cada documento!

Passo 2: Aprofundando em Consultas: Operadores Lógicos e Condicionais

Operadores de Comparação (`$gt`, `$gte`, `$lt`, `$lte`, `$ne`)

Solicitação da Equipe de Balanceamento: "Precisamos de uma lista de campeões mais frágeis, com vida base menor ou igual a 560, mas que não sejam da função 'Suporte'."

Comando: `$lte` , `$ne`

Operadores de Inclusão (`$in`, `$nin`)

Solicitação da Equipe de E-sports: "Para analisar a diversidade do meta, liste todos os campeões que são de 'Ionia', 'Noxus' ou 'Zaun'."

Comando (`$in`):

Solicitação da Equipe de Design: "Estamos planejando um evento para as regiões menores. Mostre-me todos os campeões que NÃO são de 'Demacia' nem de 'Ionia'."

Comando (`$nin`):

Operadores Lógicos (\$and, \$or, \$nor, \$not)

Solicitação da Equipe de Balanceamento: "Liste os campeões que são 'Assassinos' OU que possuem velocidade de movimento maior que 340."

Comando (\$or):

Solicitação da Equipe de Balanceamento (Desafio): "Quero uma lista de campeões que atendam a dois critérios simultaneamente: o dano de ataque deve ser maior ou igual a 60 E a vida base deve ser maior ou igual a 590."

Comando (\$and):

Solicitação da Equipe de Design (Caso Inverso): "Mostre-me todos os campeões que NÃO são 'Magos' E NEM 'Atiradores'."

Comando (\$nor):

Solicitação da Equipe de Balanceamento: "Preciso de uma lista de campeões cujo recurso NÃO seja 'Mana'."

Comando (\$not):

Passo 3: Manipulação Avançada de Arrays

a) Pegar uma sequência específica de valores (\$slice)

Solicitação da Equipe de Conteúdo: "Para um post no blog, mostre as duas primeiras skins listadas para a campeã 'Lux'."

Comando: \$slice

b) Pegar somente se o array contiver os valores exatos

Solicitação da Equipe de Balanceamento: "Existe algum campeão que tenha exatamente as skins 'PROJETO' e 'Florescer Espiritual', e nenhuma outra, nessa ordem?"

Comando: find

c) Alterar o 3º valor da lista (Atualização por Posição)

Solicitação da Equipe de Conteúdo: "A skin 'Academia de Batalha' da Lux será renomeada para 'Academia de Batalha (Prestígio)'. Altere o terceiro item da lista de skins dela."

Comando:\$set (coluna.posicao)

d) Retirar o 3º valor da lista (Remoção por Posição)

Solicitação da Equipe de Conteúdo: "Decidimos remover a skin 'Guardiã Estelar' da Jinx temporariamente. Remova o terceiro item da lista de skins dela."

// Passo 1: Definir o valor como nulo

// Passo 2: Remover todos os nulos do array

Outros Comandos Úteis para Arrays (\$addToSet, \$pop, \$pullAll)

Solicitação: "Adicione a skin 'PROJETO' ao Yasuo, mas apenas se ele ainda não a tiver na lista."

Comando: \$addToSet (Adicionar sem duplicar)

Solicitação: "Remova a skin mais antiga (a primeira da lista) do Blitzcrank."

Comando: \$pop (Remover o primeiro ou último item)

Solicitação: "As skins 'Odisseia' e 'Arcane' da Jinx serão movidas para um cofre legado. Remova ambas da lista dela."

Comando: \$pullAll (Remover múltiplos itens)

Passo 4: Missões de CRUD (Create, Read, Update, Delete)

Missão 1 (Create): "Lançamento do novo campeão 'Ahri', uma maga de Ionia."

```
db.campeoes.insertOne({
  nome: "Ahri",
  titulo: "a Raposa de Nove Caudas",
  funcao: "Mago",
  regioao: "Ionia",
  recurso: "Mana",
  estatisticas: { vida: 570, dano_ataque: 55,
  velocidade_movimento: 330 },
  habilidades: [],
  skins: []
})
```

Missão 2 (Update): "Nerf na Jinx! Reduza seu dano de ataque base em 2 pontos."

Comando : \$inc

Missão 3 (Delete): "O Teemo está sendo temporariamente removido para um rework. Delete seu documento."

Comando: delete

Desafio Final (Update Many e Find): "Marque todos os campeões com a skin 'PROJETO' como participantes de um evento e depois gere um relatório com o nome e as skins deles."

// Passo 1: Marcar os participantes

// Passo 2: Gerar o relatório

Resumo dos Comandos Utilizados

Aqui está uma tabela de consulta rápida com todos os comandos e operadores que usamos nesta aula.

Comando/Operador	Descrição/Aplicação	Exemplo de Uso na Aula
use	Seleciona o banco de dados atual ou o cria se não existir.	use summoners_rift
insertOne()	Adiciona um único documento a uma coleção.	db.campeoes.insertOne({ nome: "Ahri", ... })
insertMany()	Adiciona múltiplos documentos a uma coleção a partir de um array.	db.campeoes.insertMany([{ nome: "Thresh", ... }, { nome: "Leona", ... }])
find()	Recupera múltiplos documentos que correspondem a um critério de busca.	db.campeoes.find({ regioao: "Demacia" })
updateOne()	Atualiza o primeiro documento que corresponde a um critério.	db.campeoes.updateOne({ nome: "Jinx" }, { ... })
updateMany()	Atualiza todos os documentos que correspondem a um critério.	db.campeoes.updateMany({ recurso: "Nenhum" }, { ... })
deleteOne()	Remove o primeiro documento que corresponde a um critério.	db.campeoes.deleteOne({ nome: "Teemo" })
deleteMany()	Remove todos os documentos que correspondem a um critério.	db.campeoes.deleteMany({ recurso: "Nenhum" })
\$gt / \$gte	Maior que / Maior ou igual a. Usado para filtrar números ou datas.	db.campeoes.find({ "estatisticas.vida": { \$gt: 600 } })
\$lt / \$lte	Menor que / Menor ou igual a. Usado para filtrar números ou datas.	db.campeoes.find({ "estatisticas.vida": { \$lte: 560 } })
\$ne	Diferente de. Filtra documentos onde o valor do campo não é igual ao especificado.	db.campeoes.find({ funcao: { \$ne: "Suporte" } })
\$in / \$nin	Está em / Não está em. Filtra documentos onde o valor do campo está (ou não) em uma lista de valores.	db.campeoes.find({ funcao: { \$in: ["Lutador", "Mago"] } })
\$or	OU lógico. Retorna documentos que correspondem a pelo menos uma das condições.	db.campeoes.find({ \$or: [{ funcao: "Assassino" }, { ... }] })
\$and	E lógico. Retorna documentos que correspondem a todas as condições.	db.campeoes.find({ \$and: [{ "estatisticas.dano_ataque": ... }, { ... }] })
\$nor	NÃO OU lógico. Retorna documentos que não correspondem a nenhuma das condições.	db.campeoes.find({ \$nor: [{ funcao: "Mago" }, { funcao: "Atirador" }] })
\$not	NÃO lógico. Inverte o resultado de um operador de consulta.	db.campeoes.find({ recurso: { \$not: { \$eq: "Mana" } } })
\$slice	Em uma projeção, retorna um subconjunto de elementos de um array.	db.campeoes.find({ nome: "Lux" }, { skins: { \$slice: 2 } })
\$elemMatch	Corresponde a documentos que contêm um array com pelo menos um elemento que satisfaz todas as condições especificadas.	db.campeoes.find({ habilidades: { \$elemMatch: { nome: "Centelha Final", tipo: "R" } } })

\$set	Define o valor de um campo em um documento.	db.campeoes.updateOne({ nome: "Jinx" }, { \$set: { ... } })
\$inc	Incrementa (ou decrementa) o valor de um campo numérico.	db.campeoes.updateOne({ nome: "Garen" }, { \$inc: { "estatisticas.vida": 30 } })
\$push	Adiciona um elemento a um array.	db.campeoes.updateOne({ nome: "Lux" }, { \$push: { skins: "Lux Cósmica" } })
\$pull / \$pullAll	Remove todas as instâncias de um valor (ou valores) de um array.	db.campeoes.updateOne({ nome: "Jinx" }, { \$pullAll: { skins: ["Odisseia", "Arcane"] } })
\$pop	Remove o primeiro (-1) ou o último (1) elemento de um array.	db.campeoes.updateOne({ nome: "Blitzcrank" }, { \$pop: { skins: -1 } })
\$addToSet	Adiciona um elemento a um array, mas apenas se o elemento ainda não existir.	db.campeoes.updateOne({ nome: "Yasuo" }, { \$addToSet: { skins: "PROJETO" } })