Tarefas

**Questão 1** – Criação do Banco Crie o banco de dados megaArena com as coleções jogadores, torneios e partidas.

use megaArena;

db.createCollection('jogadores')

db.createCollection('torneios')

db.createCollection('partidas')

**Questão 2** – Embedding Modele uma versão alternativa onde os jogadores inscritos em um torneio fiquem incorporados diretamente dentro do documento torneio.

db.torneios.insertOne({

"nome": "Copa Mundial de E-sports 2024",

"data\_inicio": new Date("2024-05-10T09:00:00Z"),

"data\_fim": new Date("2024-05-15T18:00:00Z"),

"jogo": "Mega Arena",

"premio": 100000,

"local": {

"cidade": "São Paulo",

"estado": "SP",

"pais": "Brasil"

},

"jogadores\_inscritos": [

{

"nome\_completo": "Ana Silva",

"nick": "AninhaBR",

"pais": "Brasil",

"idade": 23,

"email": "ana.silva@email.com"

},

{

"nome\_completo": "Carlos Santos",

"nick": "C-S\_Pro",

"pais": "Portugal",

"idade": 28,

"email": "carlos.santos@email.com"

},

{

"nome\_completo": "Emily Chen",

"nick": "EmiChen",

"pais": "China",

"idade": 25,

"email": "emily.chen@email.com"

}

]

});

**Questão 3** – Referencing Modele uma versão alternativa onde cada torneio guarda apenas os IDs dos jogadores inscritos, mantendo-os em coleção separada.

db.jogadores.insertMany([

{

"\_id": ObjectId("60c72b2f9f1b2c001f3b3b2c"),

"nome\_completo": "Ana Silva",

"nick": "AninhaBR",

"pais": "Brasil",

"idade": 23

},

{

"\_id": ObjectId("60c72b2f9f1b2c001f3b3b2d"),

"nome\_completo": "Carlos Santos",

"nick": "C-S\_Pro",

"pais": "Portugal",

"idade": 28

},

{

"\_id": ObjectId("60c72b2f9f1b2c001f3b3b2e"),

"nome\_completo": "Emily Chen",

"nick": "EmiChen",

"pais": "China",

"idade": 25

}

]);

db.torneios.insertOne({

"nome": "Copa Mundial de E-sports 2024",

"data\_inicio": new Date("2024-05-10T09:00:00Z"),

"data\_fim": new Date("2024-05-15T18:00:00Z"),

"jogo": "Mega Arena",

"premio": 100000,

"local": {

"cidade": "São Paulo",

"estado": "SP",

"pais": "Brasil"

},

"jogadores\_inscritos": [

ObjectId("60c72b2f9f1b2c001f3b3b2c"),

ObjectId("60c72b2f9f1b2c001f3b3b2d"),

ObjectId("60c72b2f9f1b2c001f3b3b2e")

]

});

**Questão 4** – Justificativa de Modelagem Explique em qual situação é melhor usar embedding e em qual é melhor usar referencing no cenário da MegaArena.

* **Embedding:**
  + Os dados do jogador são pequenos e quase sempre consultados junto com os dados do torneio.
  + Não há necessidade de consultar os jogadores de forma independente dos torneios.
  + O número de jogadores por torneio é limitado, evitando que o documento do torneio se torne excessivamente grande.
  + A integridade dos dados (manter o jogador "dentro" do torneio) é mais importante do que a flexibilidade de dados.
* **Referencing:**
  + Os dados do jogador são grandes ou acessados frequentemente de forma independente dos torneios.
  + Um jogador pode participar de muitos torneios, evitando a duplicação de dados e tornando as atualizações mais eficientes (por exemplo, atualizar a idade de um jogador em um único lugar).
  + O número de jogadores por torneio pode ser muito grande, mantendo os documentos do torneio menores.
  + A flexibilidade e a normalização dos dados são prioridades.

No cenário da **MegaArena**, o modelo de **referencing** é geralmente mais adequado. Um jogador pode participar de vários torneios e é provável que suas informações sejam acessadas de forma independente. Atualizar os dados de um jogador (como seu nome de usuário ou país) seria muito mais simples, pois a alteração precisaria ser feita apenas uma vez.

**Questão 5** – Consulta Liste todos os torneios cujo prêmio seja maior ou igual a 4000.

db.torneios.find({ premio: { $gte: 4000 } });

**Questão 6** – Consulta Avançada com $and Liste os jogadores do Brasil com idade maior que 21 anos.

db.jogadores.find({ $and: [ { pais: "Brasil" }, { idade: { $gt: 21 } } ] });

**Questão 7** – Operador $exists Liste todos os jogadores que já possuem o campo nick.

db.jogadores.find({ nick: { $exists: true } });

**Questão 8** – Atualização com $set Atualize o torneio “Copa CS:GO” adicionando o campo status: "ativo".

db.torneios.updateOne(

{ nome: "Copa CS:GO" },

{ $set: { status: "ativo" } }

);

**Questão 9** – Exclusão Remova o jogador com nome Pedro.

db.jogadores.deleteOne({ nome: "Pedro" });

**Questão 10** – Paginação Liste os torneios ordenados pela premiação (decrescente), exibindo apenas o segundo torneio mais bem pago.

db.torneios.find().sort({ premio: -1 }).skip(1).limit(1);