

### Questão 1 – Criação do Banco

Crie o banco de dados megaArena com as coleções jogadores, torneios e partidas

Resposta:

```
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.

Resposta:

```
db.torneios.insertOne({
  nome: "Torneio de Verão",
  local: "São Paulo",
  data: ISODate("2025-09-20"),
  jogadores: [
    "Carlos",
    "Ana",
    "João"
  ]
})
-----
```

### Questão 3 – Referencing

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

Resposta:

```
db.jogadores.insertMany([
  {
    _id: ObjectId("64fa1a1a1a1a1a1a1a1a1a1a"),
    nome: "Carlos",
    ranking: 3,
    pais: "Brasil"
  },
  {
    _id: ObjectId("64fa1a1a1a1a1a1a1a1a1a1b"),
    nome: "Ana",
    ranking: 2,
    pais: "Brasil"
  },
  {
    _id: ObjectId("64fa1a1a1a1a1a1a1a1a1a1c"),
    nome: "João",
    ranking: 1,
    pais: "Portugal"
  }
])
```

```
])
```

-----

#### Questão 4 – Justificativa de Modelagem

Explique em qual situação é melhor usar embedding e em qual é melhor usar referencing no cenário da MegaArena.

Resposta:

No cenário da MegaArena, embedding é melhor quando os dados são usados exclusivamente dentro de um contexto, como partidas embutidas no documento de um torneio, isso facilita leitura rápida e evita múltiplas consultas. Já referencing é ideal para dados reutilizáveis e dinâmicos, como jogadores que participam de vários torneios, mantendo seus dados centralizados e fáceis de atualizar. Essa combinação garante eficiência e flexibilidade no banco.

-----

#### Questão 5 – Consulta

Liste todos os torneios cujo prêmio seja maior ou igual a 4000.

Resposta:

```
db.torneios.find(
  { premio: { $gte: 4000 } },
  { nome: 1, premio: 1, _id: 0 }
)
```

-----

#### Questão 6 – Consulta Avançada com \$and

Liste os jogadores do Brasil com idade maior que 21 anos.

Resposta:

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

-----

#### Questão 7 – Operador \$exists

Liste todos os jogadores que já possuem o campo nick.

Resposta:

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

-----

#### Questão 8 – Atualização com \$set

Atualize o torneio “Copa CS:GO” adicionando o campo status: "ativo".

Respsota:

```
db.torneios.updateOne(
  { nome: "Copa CS:GO" },
  { $set: { status: "ativo" } }
```

)

-----

#### Questão 9 – Exclusão

Remova o jogador com nome Pedro

Resposta:

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

-----

#### Questão 10 – Paginação

Liste os torneios ordenados pela premiação (decrecente), exibindo apenas o segundo torneio mais bem pago.

Resposta:

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