Exercicio 1

```
db.jogadores.insertMany([
  { nome: "Adson", pais: "Brasil", idade: 21, nick: "adson21" },
  { nome: "Pedro", pais: "Brasil", idade: 19 },
  {  nome: "Maria", pais: "Brasil", idade: 25, nick: "mariag" },
  { nome: "Lucas", pais: "Portugal", idade: 23 },
  { nome: "Fernanda", pais: "Brasil", idade: 30, nick: "fer" }
]);
db.torneios.insertMany([
  { nome: "Copa CS:GO", tipo: "presencial", premio: 5000, local: "São Paulo", modalidade: "CS:GO" },
  { nome: "Torneio Valorant - Norte", tipo: "online", premio: 3000, local: "Online", modalidade: "Valorant" },
  { nome: "MegaArena Battle", tipo: "online", premio: 8000, local: "Online", modalidade: "Multi" },
  { nome: "Copa Indie", tipo: "presencial", premio: 1500, local: "Curitiba", modalidade: "Indie" }
]);
db.partidas.insertMany([
  { torneio: "Copa CS:GO", jogadores: ["Adson","Lucas"], resultado: { vencedor: "Adson" }, data: new Date("2025-08-01") },
  {  torneio: "MegaArena Battle", jogadores: ["Maria","Fernanda"], resultado: { vencedor: "Maria" }, data: new Date("2025-08-02") }
]);


const jogadores = db.jogadores.find().toArray()


// QUESTÃO 2 — EMBEDDING
db.torneios_embedded.insertOne({
  nome: "Copa Embedded",
  premio: 4500,
  tipo: "presencial",
  inscritos: [
```

{ jogador_id: jogadores[0]._id, nome: jogadores[0].nome, nick: jogadores[0].nick, pais: jogadores[0].pais },

{ jogador_id: jogadores[2]._id, nome: jogadores[2].nome, nick: jogadores[2].nick, pais: jogadores[2].pais }

]

});

db.torneios_embedded.findOne({ nome: "Copa Embedded" });

```
  }
> db.torneios_embedded.findOne({ nome: "Copa Embedded" });
< {
    _id: ObjectId('68c8a129e546a390ea0db180'),
    nome: 'Copa Embedded',
    premio: 4500,
    tipo: 'presencial',
    inscritos: [
      {
        jogador_id: ObjectId('68c8a024e546a390ea0db175'),
        nome: 'Adson',
        nick: 'adson21',
        pais: 'Brasil'
      },
      {
        jogador_id: ObjectId('68c8a024e546a390ea0db177'),
        nome: 'Maria',
        nick: 'mariag',
        pais: 'Brasil'
      }
    ]
```

// QUESTÃO 3 — REFERENCING

db.torneios_ref.insertOne({

nome: "Copa Referenced",

```
  premio: 6000,
  tipo: "online",
  inscritosIds: [ jogadores[0]._id, jogadores[3]._id ] // apenas ObjectId references
});
```

db.torneios_ref.find()

// QUESTÃO 4 — JUSTIFICATIVA - EMBEDDING é preferível quando os dados “filhos” (ex.: informações dos jogadores inscritos) são lidos junto com o documento pai com alta frequência, e o conjunto não cresce indefinidamente — ex.: torneios pequenos, leitura rápida dos inscritos, consistência local.

- REFERENCING é preferível quando jogadores participam de muitos torneios (dados grandes / compartilhados), quando é necessário evitar duplicação / garantir que

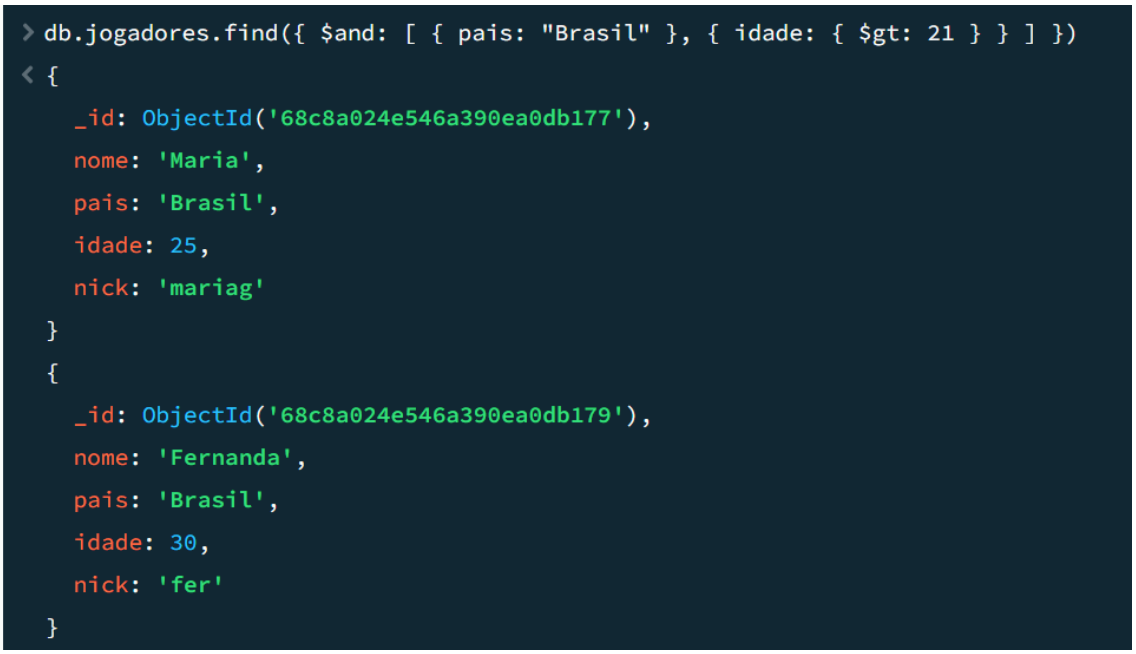
atualizações nos dados de jogador (ex.: nick) reflitam em todos os lugares sem atualização de múltiplos documentos, ou quando a lista de inscritos pode ficar muito grande.

// QUESTÃO 5 — Consulta: torneios com prêmio >= 4000

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

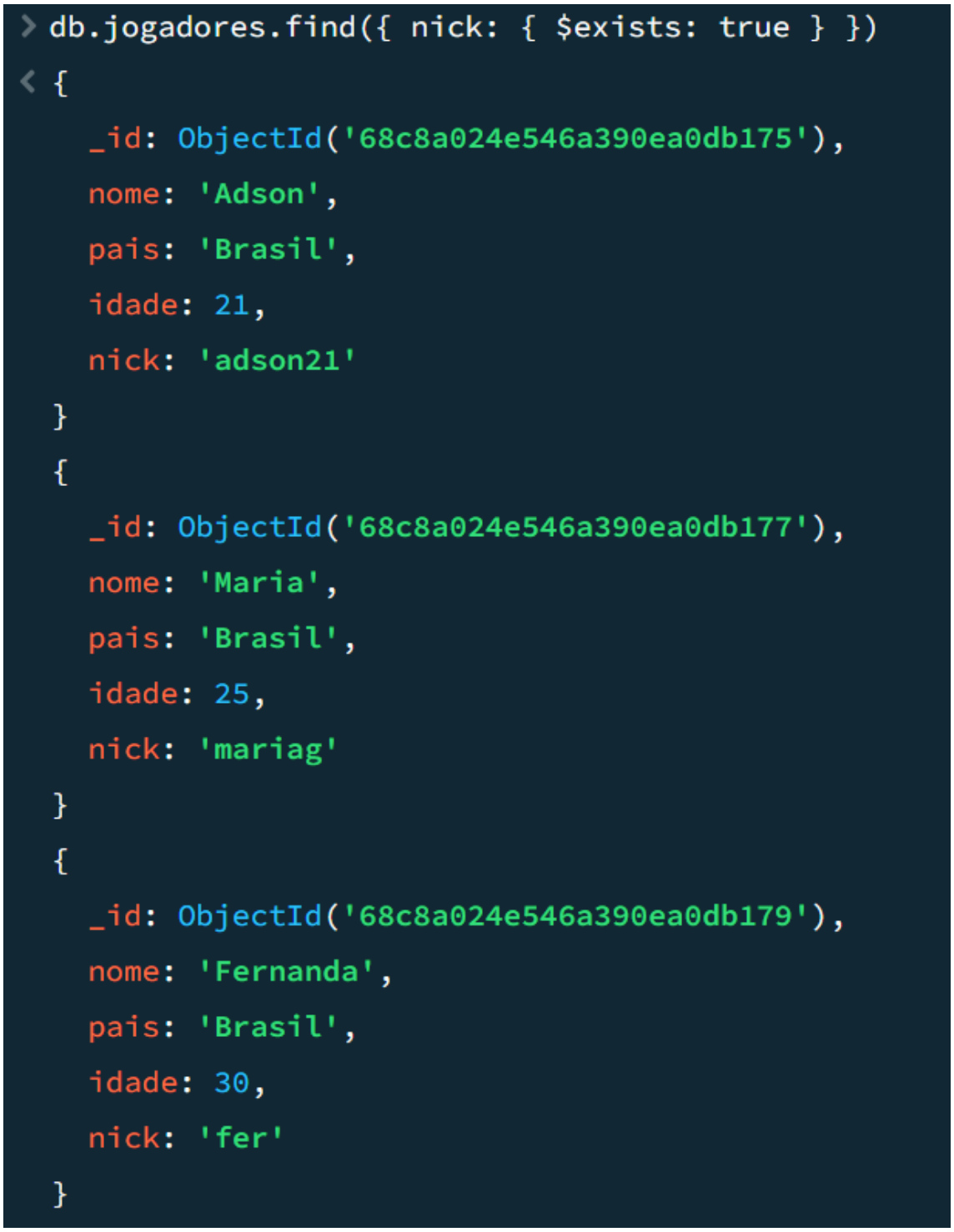
// QUESTÃO 6 — Consulta avançada com $and:

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



// QUESTÃO 7 — Operador $exists:

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



```
> db.jogadores.find({ nick: { $exists: true } })
< {
    _id: ObjectId('68c8a024e546a390ea0db175'),
    nome: 'Adson',
    pais: 'Brasil',
    idade: 21,
    nick: 'adson21'
  }
  {
    _id: ObjectId('68c8a024e546a390ea0db177'),
    nome: 'Maria',
    pais: 'Brasil',
    idade: 25,
    nick: 'mariag'
  }
  {
    _id: ObjectId('68c8a024e546a390ea0db179'),
    nome: 'Fernanda',
    pais: 'Brasil',
    idade: 30,
    nick: 'fer'
  }
```

// QUESTÃO 8 — Atualização com $set:

db.torneios.updateOne(

{ nome: "Copa CS:GO" },

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

);

db.torneios.findOne({ nome: "Copa CS:GO" });

```
> db.torneios.findOne({ nome: "Copa CS:GO" });
< {
    _id: ObjectId('68c8a033e546a390ea0db17a'),
    nome: 'Copa CS:GO',
    tipo: 'presencial',
    premio: 5000,
    local: 'São Paulo',
    modalidade: 'CS:GO',
    status: 'ativo'
  }
```

// QUESTÃO 9 — Exclusão:

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

db.jogadores.find({ nome: "Pedro" })

```
> db.jogadores.deleteOne({ nome: "Pedro" });
  db.jogadores.find({ nome: "Pedro" })
<
```

// ----------

// QUESTÃO 10 — Paginação:

db.torneios.find()

.sort({ premio: -1 })

.skip(1)

.limit(1)


>_MONGOSH
db.torneios.find()
.sort({ premio: -1 })
.skip(1)
.limit(1)
{
_id: ObjectId('68c8a033e546a390ea0db17a'),
nome: 'Copa CS:GO',
tipo: 'presencial',
premio: 5000,
local: 'São Paulo',
modalidade: 'CS:GO',
status: 'ativo'
}