

Nome: Simão Lavarinhas Amaro  
Número estudante: 2200937

**1) (0,5 valores) Explique o que entende por atomicidade de transações no contexto das bases de dados. Justifique porque é importante que os SGBD cumpram este requisito e quais os problemas caso contrário. Dê um exemplo..**

Dos slides da cadeira, em “Introdução às bases de dados relacionais”, slide 1.6:

Atomicidade das atualizações

- Falhas podem deixar base de dados num estado inconsistente com atualizações parciais realizadas
- Exemplo: Transferência de dinheiro de uma conta para outra deve ter sido completada ou então não acontecer!

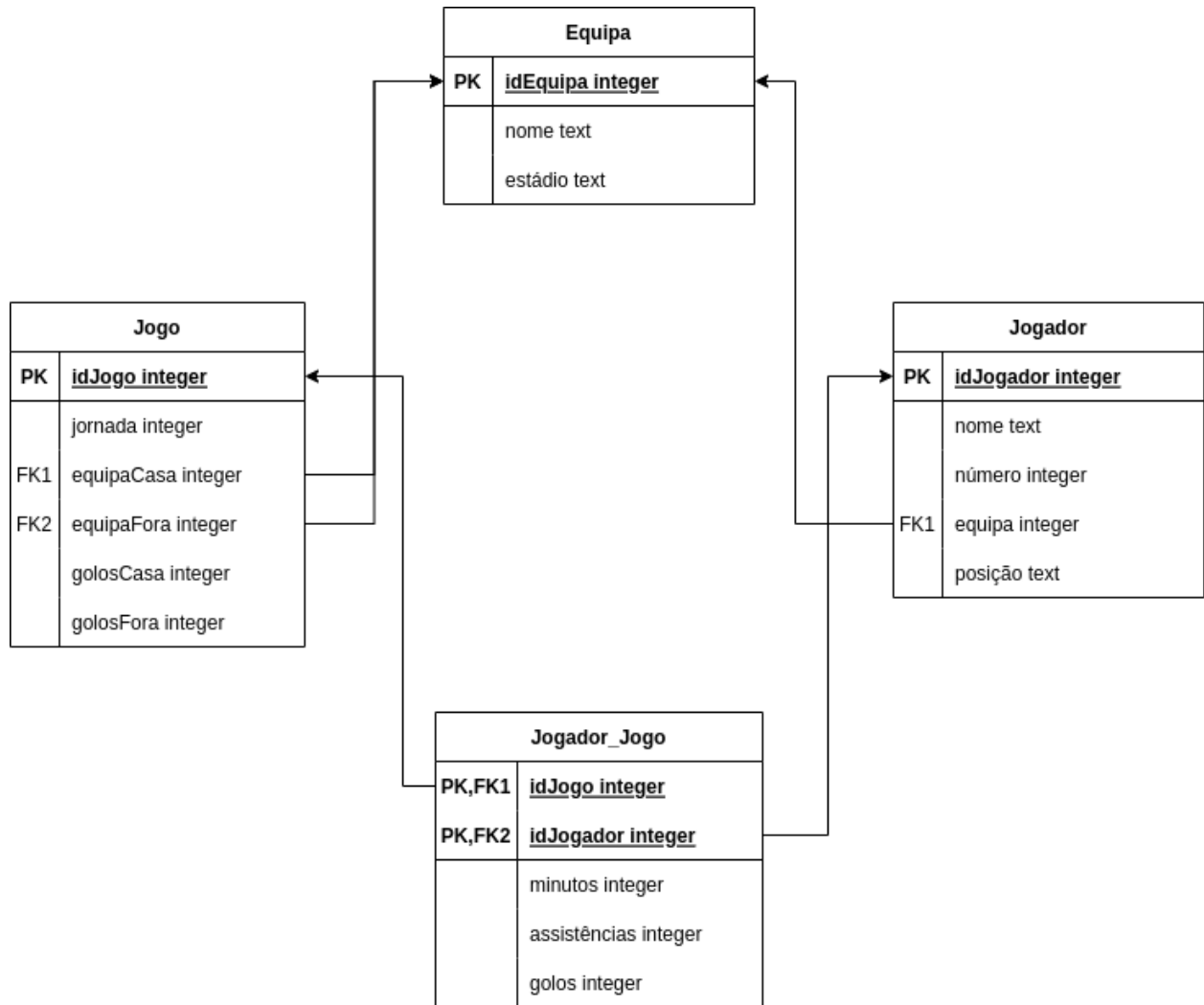
Por outras palavras, a atomicidade de transações é muito importante, visto que pode-se estar a comprometer a base de dados com estados indesejados. Por exemplo, pretendendo ir do estado A -> C, pode ser necessário passar por B, formando A -> B -> C. Caso haja algum problema na passagem B -> C, a base de dados pode ficar presa no estado B, completamente diferente do C esperado.

Para além disto, poderia gerar problemas de concurrency, em que várias chamadas estão a ser efetuadas simultaneamente, o que, no caso das transações não serem atômicas, poderia alterar o resultado da transação.

Como exemplo, supõe-se uma base de dados com uma lista de pessoas que têm direito a receber algum dinheiro. Para pagar, faz-se um query à base de dados e envia-se o dinheiro para essas pessoas. De seguida, apaga-se as linhas cujas pessoas já receberam o dinheiro. Não sendo atômica, pode acontecer apenas algumas pessoas serem apagadas, ficando ainda algumas pessoas que já receberam o dinheiro na base de dados. Deste modo, o operador poderia não se aperceber do sucedido e a mesma pessoa podia receber o dinheiro duplamente.

2) (0,5 valores) Considere uma base de dados que armazena dados sobre uma competição de futebol, com as seguintes tabelas:

- Equipa (idEquipa -> nome, estádio)
- Jogo (idJogo -> jornada, equipaCasa, equipaFora, GolosCasa, GolosFora)
- Jogador (idJogador -> nome, numero, equipa, posicao)
- Jogador\_Jogo (idJogo, idJogador -> minutos, assistencias, golos)



**a) (0,5 valores) Obtenha todos os nomes das equipas que jogaram em casa e ganharam o jogo, nas jornadas 1 ou 17.**

**i)**

```
SELECT nome
FROM   Equipa,
       Jogo
WHERE  idEquipa == equipaCasa
      AND ( jornada == 1 )
      AND ( golosCasa > golosFora )
UNION
SELECT nome
FROM   Equipa,
       Jogo
WHERE  idEquipa == equipaCasa
      AND ( golosCasa > golosFora )
      AND ( jornada == 17 )
```

**ii)**

```
SELECT nome
FROM   Equipa,
       Jogo
WHERE  idEquipa == equipaCasa
      AND ( jornada == 1
          OR jornada == 17 )
      AND ( golosCasa > golosFora )
```

**b) (0,5 valores) Apresente o nome dos jogadores e o seu maior número de golos marcados num jogo, ordenados de forma descendente.**

```
SELECT nome,
       Max(golos)
FROM   Jogador,
       Jogador_Jogo
WHERE  Jogador.idJogador == Jogador_Jogo.idJogador
GROUP BY nome
ORDER BY Max(golos) DESC
```

