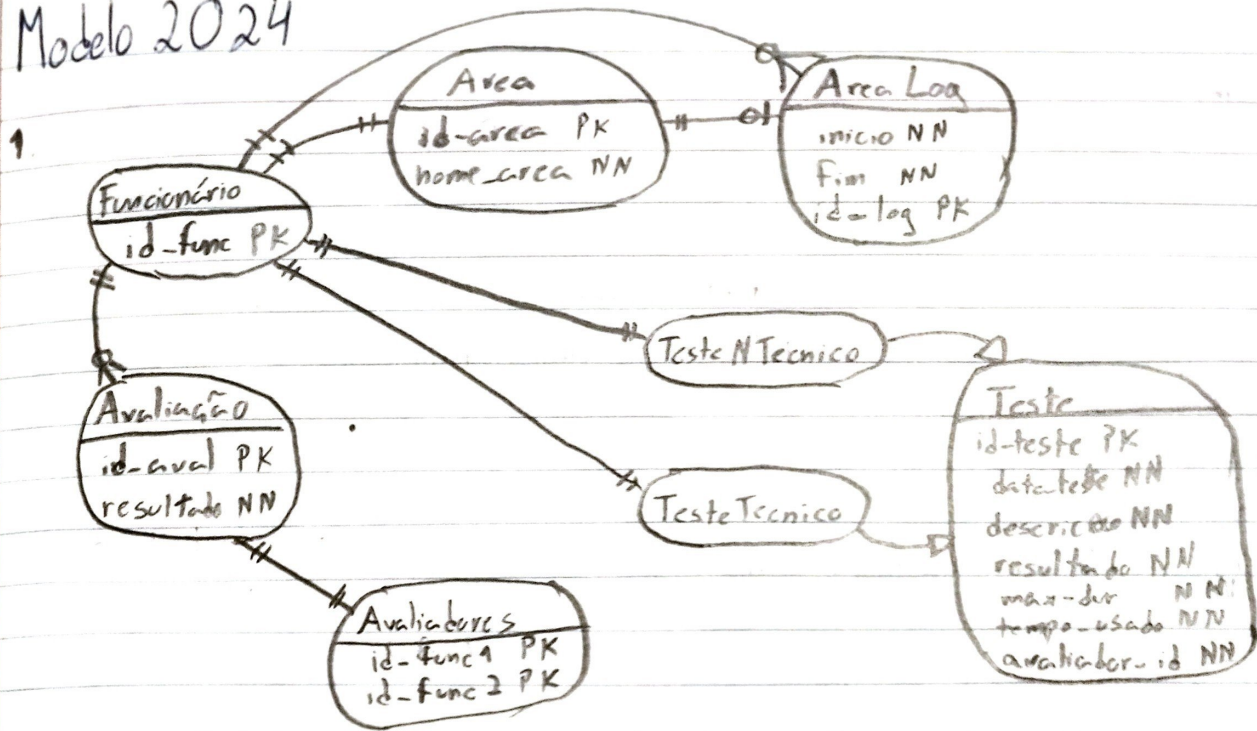


Modelo 2024



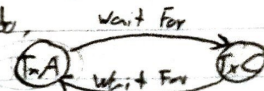
2a) Genericamente, deveriam ser adicionados índices às primary Keys e às foreign Keys, tendo em conta que estas vão ser, garantidamente, muito usadas em cláusulas Where, Group By e join, para além desses índices, sem saber mais informação sobre as queries específicas, não há nenhum índice que seria benéfico de adicionar em qualquer caso.

b) Neste caso, adicionar índices aos atributos que estão a ser usados em cláusulas where, de modo a tornar o acesso a esses dados mais rápido, ou seja: r.id-veículo, v.id-veículo, r.id-reparação, s.id-reparação, v.marca e v.modelo.

3. T2: NIB balance	T6: NIB balance	Isolação nível repeatable read
a)		garante que os resultados são consistentes com o estado do DB após a 1ª instrução
100 10000	100 10000	
200 10000	200 10000	

b) Vai ser 10000, pois na perspetiva de TxC, o estado do DB permanece igual ao estado em T2, uma vez que TxC está no nível de isolação repeatable read.

c) Dá-se um deadlock, pois TxC está à espera que o row exclusive lock seja libertado da linha com NIB=100, esse lock foi adquirido por Tx A, que, por sua vez, está à espera que o lock da linha com NIB=200 seja libertado, no entanto, este está na posse de TxC.



Isolamento: Os dados de uma transação estão a ser usados noutra

d) Atomicidade: Uma transferência devia ser uma operação indivisível, se qualquer passo falhar, toda a transação devia ser abortada, isto não acontece, uma vez que a transferência está dividida em 2 transações.

Consistência: A quantidade de dinheiro total na base de dados devia manter-se sempre, no entanto, caso uma das duas operações de update falhe, a BD vai acabar com ± 100 € no total.

4 a)

```
Select QT20 As "Quantidade 2020",
       QT19 As "Quantidade 2019",
       i.num-item As "Código Produto
From (Select
      Sum(Case When date-part('year', e.data-enc) = 2020 Then
            i.quantidade Else 0)) As QT20
      Sum(Case When date-part('year', e.data-enc) = 2019 Then
            i.quantidade Else 0)) As QT19
      From encomendas As e
      Join itens As i
      On i.num-enc = e.num-enc
      Group By i.num-item;) As sub
Where i.num-item c in (Select i.num-item
                      From clientes As c
                      Join encomendas As e
                      On e.cod-cliente = c.cod-cliente
                      Join itens As i
                      On i.num-enc = e.num-enc
                      Where c.localidade = 'Coimbra';)
And QT20 > QT19
```

```
b) Select e.num-enc, e.data-enc, i.quantidade
      From encomendas As e
      Join itens As i On i.num-enc = e.num-enc
      Where i.quantidade > (Select Avg(i.quantidade)
                            From encomendas as e
                            Join itens as i2 On e.num-enc = i2.num-enc
                            Where date-part('year', e.data-enc) = 2019)
```


→

```
And i.num-item Not In (Select i.num-item
                        From items As i
                        Join encomendas As e
                        On e.num-item = i.num-item
                        Where date-part('year', e.data-enc) = 2020);
And date-part('year', e.data-enc) = 2019;
```

5

```
Create Or Replace Procedure proc ()
Language plpgsql (As $$)
Declare
```

```
cliente-id clientes.cod_cliente %Type;
```

```
c1 cursor for
```

```
Select c.cod_cliente, Sum(i.quantidade * i.preco) As gasto
```

```
From clientes As c
```

```
Join encomendas As e On c.cod_cliente = e.cod_cliente
```

```
Join itens As i On i.num-enc = e.num-enc
```

```
Group By c.cod_cliente
```

```
Order By gasto Desc
```

```
Limit 3;
```

And
date-part('year', e.data-enc) = 2022

```
Begin
```

```
open c1
```

```
Loop
```

```
Fetch c1 Into cliente-id;
```

```
Exit when not found
```

```
Update clientes
```

```
Set saldo = saldo + 50
```

```
Where c.cod_cliente = cliente-id;
```

```
End Loop;
```

```
Close c1;
```

```
End;
```

```
$$;
```

(OU)

```
Update clientes
```

```
Set saldo = saldo + 50
```

```
Where cod_cliente In (
```

```
Select c.cod_cliente, Sum(i.quantidade * i.preco) As
```

```
From encomendas as e
```

```
Join itens As i
```

```
On i.num-enc = e.num-enc
```

```
Where date-part('year', e.data-enc) = 2020
```

```
Group By c.cod_cliente
```

```
Order By gasto Desc
```

```
Limit 3
```

```
);
```

6 a) Sort cache, usada temporariamente para tornar operações de group by ou order by mais rápidas. SQL cache, guarda as queries mais recentes de modo as queries subsequentes e iguais não terem de passar pelo processo de parsing. Data cache, guarda os blocos recentemente acessados em RAM de modo a acelerar acessos a zonas de memória frequentemente utilizadas.

b)

Tablespace: define locais no file system nos quais são colocados os ficheiros relacionados a objetos e estruturas lógicas da base de dados.

Database: Coleção de dados organizada e agrupada por tópico ou propósito

Schema: Componente da database que contém outros tipos de objetos como tipos de dados, funções e operadores

c) OLTP usa uma estrutura de dados normalizada, um pequeno volume de dados e é usado para transações do dia a dia e consultas simples

OLAP usa uma estrutura de dados desnormalizada, geralmente em forma de cubo, lida com um grande volume de dados, é usado para grandes e complexas operações analíticas e consultas também complexas.

d) Podemos utilizar métodos de controlo de concorrência como locks na tabela de apostas para prevenir problemas derivados de escritas simultâneas