

TAREFA ORIENTADA S18

Controlo de Concorrência

Objetivos:

- Objetivos do controlo de concorrência
- Otimização do nível de concorrência
- Noção de impasse (*deadlock*) e formas para o tentar impedir

O controlo de concorrência consiste na capacidade do sistema suportar várias transações a trabalhar com os mesmos dados ao mesmo tempo, otimizando o desempenho do sistema e evitando problemas que comprometam a integridade dos dados. A concorrência poderá ser um problema quando os dados são alterados. A consulta dos mesmos dados por várias transações não causa problemas, pois não existe interferência.

Existem várias técnicas para controlo de concorrência. Uma das mais usadas baseia-se em protocolos de bloqueio. As técnicas de bloqueio impedem a execução de outras transações quando existe conflito com uma transação que já está a decorrer. Quando essa transação desbloquear os dados, outra transação já poderá usar os dados.

Os problemas que as técnicas de controlo de concorrência visam evitar são os seguintes:

- **Leituras erróneas (*Dirty reads*)** - Este problema pode surgir quando numa transação são lidos dados que foram alterados, mas não foram confirmados, por outra transação que posteriormente é abortada.
- **Perda de atualizações (*Lost update*)** - Este problema pode ocorrer quando várias transações selecionam os mesmos dados, alterando-os com base no valor original. A última transação (confirmada) anula as atualizações efetuadas pelas outras transações (confirmadas anteriormente).

- **Leituras não reprodutíveis** (*Nonrepeatable reads*) - Problema que ocorre quando, numa mesma transação, duas consultas aos mesmos dados mostram informação diferente, devido a alterações entretanto confirmadas por outras transações.
- **Leituras fantasmas** - (*Phantom reads*) - Este problema sucede quando ações de remoção e/ou inserção são realizadas sobre dados que satisfazem as condições de seleção de registos lidos por outra transação T; A transação T conterá registos que já não existem e/ou não conterá registos que passaram a existir (pois seriam selecionados inicialmente se já existissem).

Apesar de os bloqueios serem geridos automaticamente pelo sistema, pode-se otimizar o comportamento recorrendo a duas abordagens:

- Utilizando opções em instruções;
- Especificando o nível de isolamento pretendido para a sessão ou conjunto de instruções (via SET TRANSACTION ISOLATION LEVEL)

Os níveis de isolamento disponíveis no SQL Server são apresentados seguidamente, indicando os problemas que resolvem (V) ou que não resolvem(X).

Nível de isolamento	Leituras erróneas	Perda de atualizações	Leituras não reprodutíveis	Leituras fantasmas
READ UNCOMMITTED	X	X	X	X
READ COMMITTED	V	X	X	X
REPEATABLE READ	V	V	V	X
SNAPSHOT	V	V	V	V
SERIALIZABLE	V	V	V	V

O nível de isolamento por omissão do SQL Server é READ COMMITTED. Para alterar o nível de isolamento, usa-se o comando apresentado seguidamente. Esta alteração é aplicada no âmbito de uma sessão ou conexão.

```
SET TRANSACTION ISOLATION LEVEL
```

{ READ UNCOMMITTED READ COMMITTED REPEATABLE READ SNAPSHOT SERIALIZABLE }
--

Em sistemas com elevado número de utilizadores e transações poderá ser necessário gerir e otimizar este aspeto e alterar o comportamento por omissão também no âmbito de um comando de consulta. Existe a possibilidade de desativar bloqueios, usando a opção (table hint) READ UNCOMMITTED (READUNCOMMITTED) que é equivalente a usar NOLOCK. Esta opção pode ser usada com instruções SELECT, mas não com instruções DELETE ou UPDATE.

Outra opção para alterar o comportamento por omissão do sistema é bloqueio de toda a tabela durante uma instrução SQL via TABLOCK (ou TABLOCKX). Por exemplo, durante uma instrução SQL tal como UPDATE podemos evitar que alguém acesse a tabela, bloqueando o seu acesso e aumentando o desempenho de uma funcionalidade mais crítica.

Outro aspeto importante e relacionado trata-se do conceito de impasse (*deadlock*) que ocorre quando duas transações mantêm e solicitam recursos detidos por ambas. Os impasses tendem a ocorrer mais frequentemente em níveis de isolamento superior. Cabe ao sistema detetar a ocorrência de um impasse e selecionar a transação que será cancelada, sendo esta designada a “vítima”. Já ao programador cabe a tarefa de tentar evitar situações conducentes a impasses, tendo em conta indicações como as apresentadas mais à frente, depois dos exemplos.

EXEMPLOS

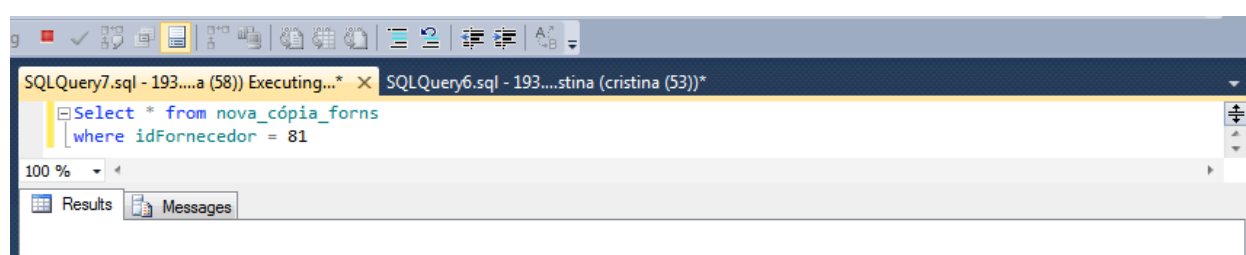
- 1 Formule, analise e execute as instruções a seguir apresentadas.
 - 1.1 Use o botão *NewQuery* (se necessário) para abrir uma janela (que designaremos NQ1) e crie uma nova cópia da tabela Fornecedores.

```
select *  
into nova_cópia_forns  
from fornecedores
```

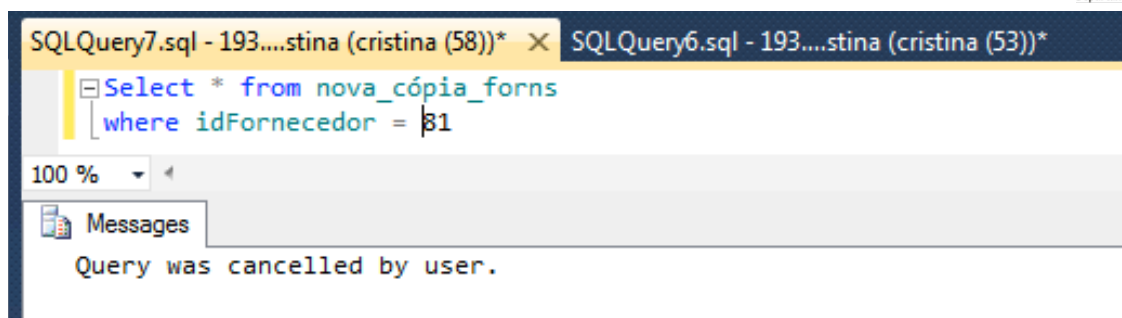
- 1.2 Use novamente o botão *NewQuery* para abrir outra janela (que designaremos NQ2) e execute a seguinte instrução

```
Begin tran  
Update nova_cópia_forns  
set Nome='alterado' where idFornecedor = 81  
-- commit
```

- 1.3 Volte para a primeira janela (NQ1) e execute a instrução seguinte.



- 1.4 Repare que a consulta fica bloqueada. Cancele a execução da consulta, usando o botão para o efeito que aparece assinalado a vermelho (ou Alt+Break).



- 1.5 Execute agora o seguinte comando. Vamos usar a opção READ UNCOMMITTED (READUNCOMMITTED) que é equivalente a usar NOLOCK. Esta opção pode ser usada com instruções SELECT, mas não com instruções DELETE ou UPDATE.

```
Select * from nova_cópia_forns (NOLOCK) -- ou with (NOLOCK)
where idFornecedor = 81
```

Results		Messages							
	IDFornecedor	Nome	Endereço	Localidade	CódigoPostal	Telefone	PNomeContacto	ÚNomeContacto	CondiçãoPagamento
1	81	alterado	Rua do Jardim das Patas, nº 1	Setubal	1500-111	212987666	Alexandre	Farinha	2

Os dados são mostrados alterados e trata-se de uma leitura errónea. Se a transação não for confirmada os dados não vão ser alterados. Este é o problema do uso de NOLOCK. Porém, a consulta não ficou bloqueada e foi possível aceder aos dados.

- 1.6 Seguidamente exemplifica-se a opção TABLOCK (ou TABLOCKX) no âmbito de uma operação de aumento de 100 do crédito de todos os clientes.

```
UPDATE Cópia_facturas
WITH (TABLOCKX) -- ou TABLOCK
SET crédito=crédito + 100;
```

- 1.7 Segue-se um exemplo de uma situação de impasse. Para que este exemplo funcione é necessário executar primeiro o conteúdo da janela NQ1 e, em 10 segundos no máximo (tempo de espera que se está a usar na simulação com WAITFOR DELAY), executar o conteúdo de NQ2.

NQ1	NQ2
<pre> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ DECLARE @totfact money BEGIN TRAN SELECT @totfact= SUM(MontanteItem) from cópia_itens_factura where IDfactura=19 WAITFOR DELAY '00:00:10' UPDATE cópia_facturas SET totalfactura=@totfact where IDfactura=19 COMMIT TRAN </pre>	<pre> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ DECLARE @totfact money BEGIN TRAN SELECT @totfact=totalfactura FROM cópia_facturas where IDfactura=19 UPDATE cópia_itens_factura SET MontanteItem = @totfact where IDfactura=19 and SequenciaItem=1 COMMIT TRAN </pre>

O resultado de NQ1 será (1 row(s) affected) e em NQ2 será o seguinte.

Messages
 Msg 1205, Level 13, State 45, Line 10
 Transaction (Process ID 60) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transacti

Seguidamente apresentam-se algumas indicações para evitar impasses:

- O nível de isolamento READ COMMITED é suficiente na maior parte das situações; Assim sendo, Devem-se reservar os níveis de isolamento superiores para transações curtas e que efetuam alterações a dados cuja integridade é vital;
- As transações devem ser curtas; os comando de consulta (SELECT) só devem fazer parte das transações quando for mesmo necessário; Nunca devem existir solicitações de dados aos utilizadores incluídas em transações;
- Devem ser evitadas operações de alteração intensiva de dados durante as horas de grande utilização dos dados; no caso de existirem transações que alteram os mesmos recursos, colocar as alterações pela mesma ordem para evitar situações como a exemplificada.

QUESTÕES

2 Crie as seguintes consultas.

2.1 Desfaça a instrução de alteração do nome do fornecedor com IdFornecedor 81 e verifique que o valor deste fornecedor não foi alterado.

Resultado

	IDFornecedor	Nome	Endereço	Localidade	Código Postal	Telefone	PNoneContacto	ÚNomeContacto	Condição Pagamento
1	81	Bell	Rua do Jardim das Patas, nº 1	Setubal	1500-111	212987666	Alexandre	Farinha	2

2.2 Altere o nível de isolamento da sessão ou conexão, de forma a não se garantir o tratamento do problema das leituras erróneas. Execute o passo 1.2 e 1.3 e verifique que é possível consultar os dados sem gerar bloqueio e que o seu conteúdo é o seguinte.

Resultado

	IDFornecedor	Nome	Endereço	Localidade	Código Postal	Telefone	PNoneContacto	ÚNomeContacto	Condição Pagamento
1	81	alterado	Rua do Jardim das Patas, nº 1	Setubal	1500-111	212987666	Alexandre	Farinha	2

2.3 Reponha o nível de isolamento que impede leituras erróneas.