Curso SQL07
Laboratório – Módulo 03

Questão 01 – Deadlock

Utilize o script abaixo para criar a tabela T1 no banco de dados Tempdb. No script "Lab03 – Aluno.sql" o script equivale aos passos 1.1 a 1.2.

```
USE tempdb
GO

IF OBJECT_ID('dbo.T1') IS NOT NULL
  DROP TABLE dbo.T1

CREATE TABLE dbo.T1
(
  keycol INT         NOT NULL PRIMARY KEY,
  col1   INT         NOT NULL,
  col2   VARCHAR(50) NOT NULL
)

INSERT INTO dbo.T1(keycol, col1, col2) VALUES(1, 101, 'A')
INSERT INTO dbo.T1(keycol, col1, col2) VALUES(2, 102, 'B')
INSERT INTO dbo.T1(keycol, col1, col2) VALUES(3, 103, 'C')

CREATE INDEX idx_col1 ON dbo.T1(col1)
GO
```

Em conexões distintas, execute os batches abaixo.

```
-- Conexão 01
USE tempdb
GO
WHILE 1 = 1
  UPDATE dbo.T1 SET col1 = 203 - col1 WHERE keycol = 2
GO

-- Conexão 02
USE tempdb
GO
DECLARE @i AS VARCHAR(10)
WHILE 1 = 1
  SET @i = (SELECT col2 FROM dbo.T1 WITH (index = idx_col1)
            WHERE col1 = 102)
GO
```

Depois de certo tempo, você verá que uma das conexões terá seu batch cancelado devido a um deadlock encontrado pelo SQL Server, sendo exibida a mensagem 1205. Seu trabalho nesse exercício é:

- Identificar qual a causa do deadlock.
- Propor uma solução para o problema acima (existe mais de uma solução, se você encontrar as duas, excelente)

Dica: utilize o SQL Server Profiler, monitorando o evento Deadlock Graph para ajudá-lo a encontrar a causa raiz do problema.

Questão 02 – Locking

Utilizando o SQLCMD abra uma conexão com o seu SQL Server e execute o script "Lab03 – Setup.sql", **sem analisar nesse momento o conteúdo do script**. Um exemplo de utilização do comando SQLCMD pode ser:

SQLCMD –E –i "Lab03 – Setup.sql"

Abra **duas conexões** com o seu servidor utilizando o SQLCMD (ex.: SQLCMD –E ou SQLCMD –Usa –Psenha) e execute os comandos abaixos:

Na primeira conexão:

```
1> Use tempdb
2> go
1> exec Simulacao01
2> go
```

Na segunda conexão:

```
1> Use tempdb
2> go
1> exec Simulacao02
2> GO
```

Você vai reparar que a conexão 02 é responsável por simular uma série de autenticações no sistema, que vai informando na tela quando um usuário é autenticado. Porém em determinados momentos o usuário final informa que não consegue se autenticar no sistema e, depois de algum tempo, o sistema volta a funcionar corretamente. Você deve:

03.a Identificar a causa raiz para essas paradas intermitentes e explicá-la.

03.b Propor ao menos duas soluções diferentes para o problema.

Questão 03 – Análise de bloqueios

03.a Usando o banco de dados AdventureWorks2008, analise a situação dos bloqueios armazenados pelo SQL Server após a execução de cada conjunto de instruções abaixo (cada número representa uma análise).

Existem várias maneiras de analisar os bloqueios, escolha aquela que mais lhe agrada. Depois de cada execução, execute a instrução ROLLBACK para cada transação aberta, com o intuito de liberar os bloqueios, e passe para análise do próximo batch.

O que o SQL Server fez durante a execução do batch? Quais bloqueios foram pegos? Algum foi liberado? O que muda de uma instrução para outra?

```
USE AdventureWorks2008
go

-- (1)
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail
-- ANALISE OS BLOQUEIOS....
ROLLBACK


-- (2)
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail
WHERE ProductID = 710
-- ANALISE OS BLOQUEIOS....
ROLLBACK


-- (3)
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail  WITH (HOLDLOCK, XLOCK)
-- ANALISE OS BLOQUEIOS....
ROLLBACK


-- (4)
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail  WITH (HOLDLOCK, XLOCK)
```

```
WHERE SalesOrderID BETWEEN 72400 AND 75000
-- ANALISE OS BLOQUEIOS....
ROLLBACK


-- (5)
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail  WITH (HOLDLOCK, PAGLOCK, XLOCK)
WHERE SalesOrderID BETWEEN 72400 AND 75000
-- ANALISE OS BLOQUEIOS....
ROLLBACK


-- (6)
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRANSACTION
SELECT *
FROM Sales.SalesOrderDetail  WITH (HOLDLOCK, XLOCK)
WHERE SalesOrderID BETWEEN 72500 AND 75000
-- ANALISE OS BLOQUEIOS....
ROLLBACK
```

03.b Usando o script e a sessão da questão, utilize o Profiler para visualizar o impacto de cada um dos batches acima no banco de dados.

Abra a ferramenta SQL Server Profiler e monitore todas as colunas dos seguintes eventos: “lock acquired”, “lock released” e “lock escalation”, colocando um filtro sobre o campo SPID, indicando no **Equals** o identificador da sua sessão aberta.

Entre cada execução, lembre-se de fazer o rollback da transação e usar o botão “clear trace window”, para limpar os eventos já capturados pelo Profiler.

O que mudou da sua análise anterior? Você estava correto?