|  |  |
| --- | --- |
|  | **Banco de Dados**  **Administração de Banco de Dados**  **AC3 – Isolamento de processos** |

*Nome do Aluno: Eugenio Diniz RA: 2100334*

*Turma:* BD *Série:* 4 Semestre *Data: Professor(a) Responsável:* Gustavo Bianchi Maia

Nota: ( ) Visto do Prof.

**Exercícios**

>>> Abra o SSMS ( SQL Server Management Studio ) e estabeleça 1 conexão com o servidor. <<< 1. **Detalhe os comandos executados para:**

a) Ajustar o nível de isolamento para SERIALIZABLE

b) Abrir uma transação

c) Executar um CREATE TABLE

d) Executar alguns INSERTs

e) Conferir se os dados foram inseridos ( SELECT )

f) Dar ROLLBACK de toda a operação

g) Conferir se os dados ainda estão lá ( SELECT )

Escreva abaixo todos os comandos e explique o que aconteceu após o resultado da letra G. <ALÉM DOS COMANDOS SOLICITADOS - TIRE UM PRINT DE SUA TELA E COLE AQUI>

**Código utilizado**

use impacta

--mudar nível de segurança da transação

SET TRANSACTION ISOLATION LEVEL Serializable;

--criar tabela

create table Eugenio\_2100334 (

PersonID int,

LastName varchar(50),

FirstName varchar(50),

Address varchar(50),

City varchar(50)

);

--select puro na tabela limpa

select \* from Eugenio\_2100334

begin transaction

--inserts na tabela

INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)

VALUES (1, 'Lenine', 'Eugenio', 'Paulista', 'brasil');

INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)

VALUES (1, 'Silva', 'Eduardo', 'Caracascity', 'brasil');

INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)

VALUES (1, 'Augusto', 'Renato', 'enoneo', 'brasil');

INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)

VALUES (1, 'Dias', 'Bruno', 'bemlonge321', 'brasil');

--select pra verificar os commits

select \* from Eugenio\_2100334

rollback

--select pra verificar o rollback

select \* from Eugenio\_2100334

> Abra o SSMS ( SQL Server Management Studio ) e estabeleça 2 conexões com o servidor. <<< Indicando apenas **UMA** instrução ou comando por linha, preenha os quadros das questões abaixo.

**2. Detalhe os comandos executados em cada conexão para simular um Bloqueio, ou seja, um dos processo terá que ficar esperando o outro acabar.** Preencha a seguinte tabela com os comandos executados em qual transação e sua ordem.

|  |  |  |
| --- | --- | --- |
| Ordem | Transação 1 | Transação 2 |
| 1 | SET TRANSACTION ISOLATION LEVEL Serializable |  |
| 2 | Create table... |  |
| 3 | Begin transaction |  |
| 4 | Insert into .... |  |
| 5 |  | INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)  VALUES (1, 'Lenine', 'Eugenio', 'Paulista', 'brasil');  INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)  VALUES (1, 'Silva', 'Eduardo', 'Caracascity', 'brasil');  INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)  VALUES (1, 'Augusto', 'Renato', 'enoneo', 'brasil');  INSERT INTO Dbo. Eugenio\_2100334 (PersonID, LastName, FirstName, Address, City)  VALUES (1, 'Dias', 'Bruno', 'bemlonge321', 'brasil');  --select pra verificar os commits  select \* from Eugenio\_2100334  rollback  --select pra verificar o rollback  select \* from Eugenio\_2100334 |

<ALÉM DOS COMANDOS SOLICITADOS - TIRE UM PRINT DE SUA TELA E COLE AQUI>

**3. Detalhe os comandos executados em cada conexão para simular um DEADLOCK, ou seja, um dos processos terá que ser derrubado do servidor.**

Preencha a seguinte tabela com os comandos executados em qual transação e sua ordem.

|  |  |  |
| --- | --- | --- |
| Ordem | Transação 1 | Transação 2 |
| 1 | Begin transaction |  |
| 2 | Create table a (a int) |  |
| 3 |  | Begin transaction |
| 4 |  | Create table b (b int) |
| 5 | select \* from b | select \* from a |

<ALÉM DOS COMANDOS SOLICITADOS - TIRE UM PRINT DE SUA TELA E COLE AQUI> >>> Abra o SSMS ( SQL Server Management Studio ) e estabeleça 2 conexões com o servidor. <<<

**4. Detalhe o nível de isolamento utilizado/necessário e os comandos necessários em cada conexão para:**

a) Simular um erro de **leitura suja**.

Preencha a seguinte tabela com os comandos executados em qual transação e sua ordem.

Aba 1

--Criando a Tabela Empregados —

Create Table Empregados

(CodigoEmpregado Int Identity(1,1),

NomeEmpregado Varchar(100),

SalarioEmpregado Money,

Constraint PK\_CodigoEmpregado PRIMARY KEY(CodigoEmpregado) )

Go

--Inserindo os dados, populando a tabela Empregados —

Insert Into Empregados (NomeEmpregado, SalarioEmpregado)

Values ('Pedro Galvão', 2100),

('Fernanda Galvão', 2200),

('Eduardo Galvão', 4000),

('João Pedro Galvão', 1950),

('Maria Luíza Galvão', 6500)

Go

Begin Transaction

Update Empregados

Set SalarioEmpregado = 5000

Where CodigoEmpregado = 5

Go

Set Transaction Isolation Level Read Uncommitted

Set NoCount On

Go

Select CodigoEmpregado, NomeEmpregado, SalarioEmpregado

From Empregados

Where CodigoEmpregado = 5

Go

rollback

aba2

Select CodigoEmpregado, NomeEmpregado, SalarioEmpregado

From Empregados

Where CodigoEmpregado = 5

Go

Update Empregados

Set SalarioEmpregado = 7000

Where CodigoEmpregado = 5

Go

|  |  |  |
| --- | --- | --- |
| Ordem | Transação 1 | Transação 2 |
| 1 | Begin Transaction |  |
| 2 |  | Begin Transaction |
| 3 | Update |  |
| 4 |  | Select + update |
| 5 | Commit ou rollback |  |
| 6 |  | Commit |

b) Simular um erro de **leitura não repetível**

Preencha a seguinte tabela com os comandos executados em qual transação e sua ordem.

|  |  |  |
| --- | --- | --- |
| Ordem | Transação 1 | Transação 2 |
| 1 | BEGIN TRAN;  UPDATE BankDetailTbl  SET      Balance=Balance-45  WHERE AccountNumber = '7Y290394';  WAITFOR DELAY '00:00:20';  ROLLBACK TRAN;    SELECT \*  FROM BankDetailTbl  WHERE AccountNumber = '7Y290394'; |  |
| 2 |  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;  BEGIN TRAN;  SELECT \*  FROM BankDetailTbl  WHERE AccountNumber = '7Y290394';  COMMIT TRAN; |
| 3 |  |  |

<ALÉM DOS COMANDOS SOLICITADOS - TIRE UM PRINT DE SUA TELA E COLE AQUI> c) Simular um erro de **leitura fantasma**.

Preencha a seguinte tabela com os comandos executados em qual transação e sua ordem.

|  |  |  |
| --- | --- | --- |
| Ordem | Transação 1 | Transação 2 |
| 1 | CREATE TABLE Employees  (  Id INT PRIMARY KEY,  Name VARCHAR(100),  Gender VARCHAR(10)  )  Go  INSERT INTO Employees VALUES(1001,'Christian', 'Male')  INSERT INTO Employees VALUES(1002,'Carollina', 'Female')  INSERT INTO Employees VALUES(1003,'Cookie', 'Male')  INSERT INTO Employees VALUES(1004,'Clarinha', 'Female') |  |
| 2 | Select \* from Employees  Where Gender = ‘Male’ |  |
| 3 |  | INSERT INTO Employees VALUES(1003,'Madara', 'Male') |
| 4 | Select \* from Employees  Where Gender = ‘Male’ |  |

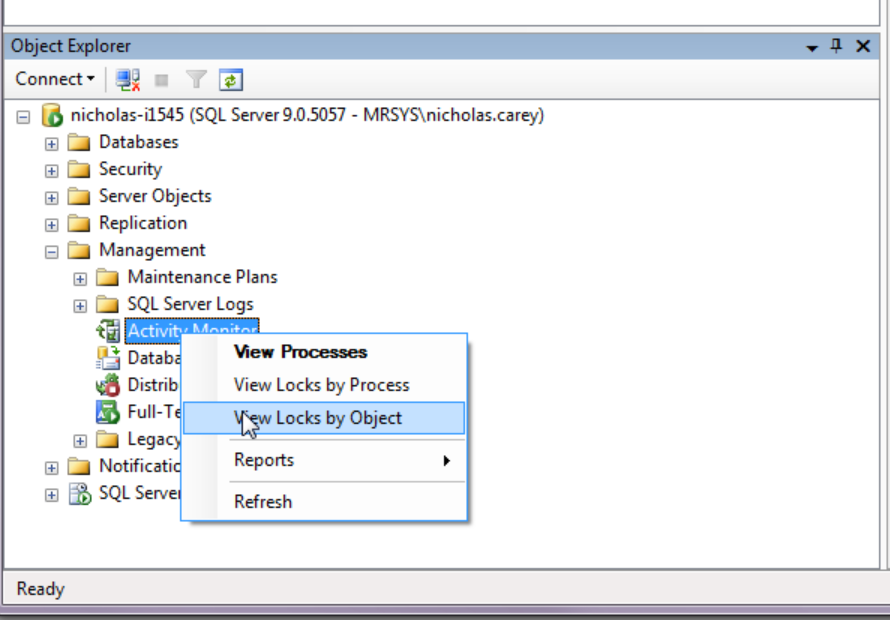
**5. Seja um ambiente conforme o simulado na questão 2, ou seja, uma conexão está bloqueando outra.**

a) Supondo que você não seja o causador destes bloqueios (ou seja, você é o DBA), como você verificaria que existem bloqueios no sistema? (detalhe as ferramentas utilizadas, os comandos executados para esta verificação, opcionalmente tire um ‘print’ da tela e cole aqui)

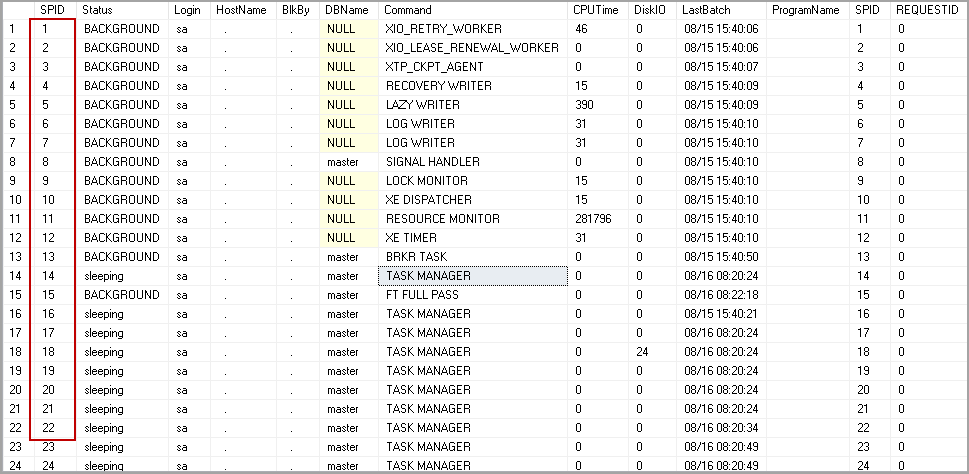
-No *Log de Transações,* verificar os Blocks e Locks ocasionados, identificar o usuário e/ou transação. Dependendo da situação, alterar o nível de isolamento poderia resolver:

SET TRANSACTION ISOLATION LEVEL {READ COMMITTED | READ UNCOMMITTED |

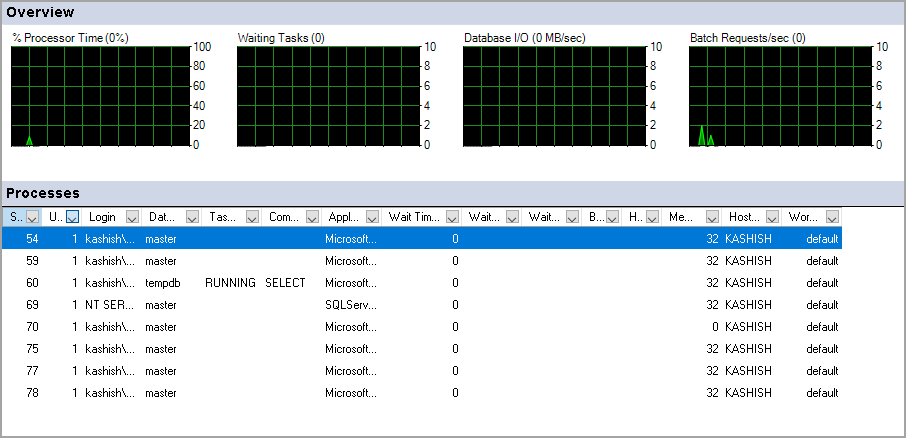
REPEATABLE READ | SERIALIZABLE | SNAPSHOT}

**

-Também é possível verificar usando o comando SP\_who2, que abrirá uma lista de todos os processos executados no servidor.



-Também é possível utilizar o monitor de atividade, podendo observar por usuário, o tempo de processamento e outras informações que poderiam atrapalhar as transações de outros usuários.



b) Como você resolveria esta situação encerrando o processo bloqueante (detalhe suas verificações e códigos executados )

-Após identificar o user que está causando a lentidão e travamento do sistema, basta executar o comando KILL (numero do user/sessão) e em alguns momentos o servidor fará o rollback de suas transações e liberará o sistema.