

# Bases de Dados 2007/2008

## Aula 6

---

### Sumário

0. Transacções e Concorrência

---

### Referências

<http://www.di.ubi.pt/~pprata/bd/BD0506-Transaccoes.pdf>  
<http://www.di.ubi.pt/~agomes/bd2/teoric/08-transaccoes.pdf>  
<http://www.di.ubi.pt/~agomes/bd2/teoric/09-concurrencia.pdf>

SQL Server 2000 Para Profissionais, Orlando Belo, FCA ISBN 972-722-505-5

SQL - Structured Query Language, Luís Manuel Dias Damas, FCA ISBN 972-722-443-1

---

1. Iniciar duas instâncias do SQL Query Analyzer e execute uma das queries indicadas em cada sessão.

O SQL possui 4 níveis de isolamento para tratar os problemas que podem ocorrer devido à concorrência. Os níveis existentes são: READ COMMITTED (opção por defeito), REPEATABLE READ, READ UNCOMMITTED e SERIALIZABLE.

- READ COMMITTED, este nível de isolamento permite apenas visualizar dados que já estejam inseridos.
- REPEATABLE READ é igual ao READ COMMITTED, apenas com a diferença de que se um registo é lido uma vez, então terá de ser sempre devolvido se a leitura for repetida.
- Se uma transacção executar-se com a opção READ UNCOMMITTED, Podemos ler informação temporária de outras transacções.
- Uma transacção SERIALIZABLE, apenas vê o estado da base de dados antes ou depois de esta ser executada.

2. Exemplo 1 READ COMMITTED

Query 1

```
select top 5 nome, D.local from departamento as D order by nome
begin tran
update departamento set Local = 'UBI'

select top 5 nome, D.local
from departamento as D order by nome

waitfor delay '00:00:05'
Rollback tran
select top 5 nome, D.local
from departamento as D order by nome
```

Query 2

```
Set transaction Isolation level read uncommitted
Select 'Agora está a vê-los...'

select top 5 nome, D.local from departamento as D
where local = 'UBI' order by nome

if @@RowCount > 0
```

```

Begin
waitfor delay '00:00:05'

Select 'Agora não...'

select top 5 nome, D.local
from departamento as D
where local = 'UBI' order by nome
end

```

Nota: enquanto o primeiro query está a executar (tem-se 5 segundos) se executarmos o segundo query podemos aceder a dados alterados pelo primeiro query mas ainda não confirmados (committed). Após esperar 5 segundos tenta-se ler novamente os mesmos dados. Contudo, como as alterações foram descartadas (rolled back), os dados desapareceram, provocando no segundo query uma leitura que não se pode repetir (nonrepeatable read).

Query 1		Query 2	
nome	local	-----	
-----	-----	Agora está a vê-los...	
Camarote	Camarate	(1 row(s) affected)	
Comercial	Lisboa	nome	local
Informática	Covilhã	-----	-----
Produção	Guarda	Camarote	UBI
(4 row(s) affected)		Comercial	UBI
(4 row(s) affected)		Informática	UBI
nome	local	Produção	UBI
-----	-----	(4 row(s) affected)	
Camarote	UBI	-----	
Comercial	UBI	Agora não...	
Informática	UBI	(1 row(s) affected)	
Produção	UBI	nome	local
(4 row(s) affected)		-----	-----
nome	local	Camarote	Camarate
-----	-----	Comercial	Lisboa
Camarote	Camarate	Informática	Covilhã
Comercial	Lisboa	Produção	Guarda
Informática	Covilhã	(4 row(s) affected)	
Produção	Guarda	(0 row(s) affected)	
(4 row(s) affected)			

### 3. Exemplo 2 READ COMMITTED

#### Query 1

```

Set transaction Isolation level read committed
begin tran
Select 'Agora está a vê-los...'

select top 5 nome, D.local
from departamento as D
order by nome

waitfor delay '00:00:05'

Select 'Agora não...'

select top 5 nome, D.local
from departamento as D
order by nome
rollback tran

```

#### Query 2

```

Set transaction Isolation level read committed

```

```

update departamento
set Nome = 'Camarata'
where nome = 'Camarote'

```

Nota: o nome do departamento foi alterado entre as leituras efectuadas durante o primeiro query – a nonrepeatable read!

Query 1	Query 2
<pre> ----- Agora está a vê-los... (1 row(s) affected)  nome                local ----- Camarote            Camarate Comercial            Lisboa Informática         Covilhã Produção            Guarda  (4 row(s) affected)  ----- Agora não... (1 row(s) affected)  nome                local ----- Camarata            Camarate Comercial            Lisboa Informática         Covilhã Produção            Guarda  (4 row(s) affected) </pre>	<pre> (1 row(s) affected) </pre>

#### 4. Exemplo 3 REPEATABLE READ

REPEATABLE READ impede outros utilizadores de alterarem a informação acedida pela transacção não impedindo a inserção de novos registos.

##### Query 1

```

Set transaction Isolation level repeatable read
begin tran
Select 'Veja bem...'

select top 5 nome, D.local
from departamento as D
order by nome

waitfor delay '00:00:05'

Select 'Veja melhor...'

select top 5 nome, D.local
from departamento as D
order by nome
rollback tran

```

##### Query 2

```

Set transaction Isolation level repeatable read
Insert departamento
Values( 50, 'YYYYY', 'XXXXX')

```

Query 1	Query 2
<pre> ----- Veja bem... </pre>	<pre> (1 row(s) affected) </pre>

(1 row(s) affected)	
nome	local
-----	-----
Camarata	Camarate
Comercial	Lisboa
Informática	Covilhã
Produção	Guarda
(4 row(s) affected)	
-----	
Veja melhor...	
(1 row(s) affected)	
nome	local
-----	-----
Camarata	Camarate
Comercial	Lisboa
Informática	Covilhã
Produção	Guarda
<b>YYYYY</b>	<b>XXXXX</b>
(5 row(s) affected)	

## 5. Exemplo 3 SERIALIZABLE

### Query 1

```

Set transaction Isolation level serializable
begin tran
Select 'Veja bem...'

select top 5 nome, D.local
from departamento as D
order by nome

waitfor delay '00:00:05'

Select 'Veja melhor...'

select top 5 nome, D.local
from departamento as D
order by nome
rollback tran

```

### Query 2

```

Set transaction Isolation level serializable
Insert departamento
Values(49, 'ZZZZZ', 'XXXXX')

```

Query 1	Query 2
-----	(1 row(s) affected)
Veja bem...	
(1 row(s) affected)	
nome	local
-----	-----
Camarate	Camarata
Comercial	Lisboa
Informática	Covilhã
Produção	Guarda
YYYYY	XXXXX
(5 row(s) affected)	
-----	

Veja melhor...	
(1 row(s) affected)	
nome	local
-----	-----
Camarata	Camarate
Comercial	Lisboa
Informática	Covilhã
Produção	Guarda
YYYYY	XXXXX
(5 row(s) affected)	

6. Criar a base de dados Conta (bancária) e introduzir uma conta nova com o saldo inicial de 55,4€.
7. Aceda à conta criada utilizando dois clientes distintos em “simultâneo” e efectue um depósito de 100€ numa das transacções e um levantamento de 60€na outra.
8. Que problemas podem ocorrer neste tipo de operação. Que procedimentos considera adequados para manter a consistência e coerência dos dados na base de dados.
9. Efectue as operações referidas utilizando transacções.
10. Crie uma nova conta com o saldo de 1000€.
11. Calcular o valor total dos depósitos.
12. Qual o resultado obtido se a operação 6 fosse efectuada durante a operação 2.
13. Dependendo do tipo de transacção definida para as operações 2 5 e 6 podemos obter resultados diferentes consoantes as transacções sejam do tipo Read committed, Read uncommitted, Repeatable read ou Serializable. Indique quais os possíveis resultados.
14. Calcular qual o saldo da conta com o menor saldo.
15. Efectue uma transferência desta conta para a conta inicial de 500€.
16. Que resultado se obtém no caso de existir uma falha no sistema depois da execução do primeiro UPDATE. Como solucionar estes problemas.
17. Quais as propriedades ACID é que são violadas na situação anterior.
18. Efectuar uma transacção para inserir quatro novas contas com os respectivos saldos.
19. Efectue uma nova transacção para introduzir 5 novas contas incluindo um SAVEPOINTS após as duas primeiras, depois introduzir um registo com uma chave primária duplicada. Usar a instrução IF (@@ERROR <> 0) BEGIN (...) END para efectuar o ROLLBACK. Verificar o resultado solucionar o problema.
20. Crie duas transacções, uma para alterar o número da última conta inserido e outra para visualizar as contas existentes. Pretende-se que nesta situação se as transacções forem executadas concorrentemente não seja possível à segunda transacção visualizar os dados inseridos pela primeira e ainda não salvaguardados na base de dados.
21. Reescreva a segunda transacção do ponto anterior de modo que numa situação de concorrência podemos garantir que a segunda instrução visualiza sempre o mesmo resultado.
22. Execute em SQL uma transacção para repor a situação existente no ponto 2.
23. Efectue novamente as todas as operações definidas sobre a tabela conta utilizando as transacções explícitas adequadas.