

Concorrência

- ▶ **Gestão de transações** engloba concorrência e recuperação.
- ▶ **Concorrência** refere-se ao facto dos SGBDs em geral permitirem que muitas transações tenham acesso à mesma base de dados ao mesmo tempo.
- Num sistema deste tipo é necessário algum tipo de mecanismo de controlo de concorrência para assegurar que transações concorrentes não interfiram umas com as outras.
- ► Uma transação mesmo correta pode fornecer uma resposta errada se sofrer alguma forma de interferência de outra transação

3



Concorrência

- Os três tipos principais de problemas que qualquer mecanismo de concorrência tem de tratar são:
 - **LOST UPDATE**
 - ▶ UNCOMMITED DEPENDENCY
 - **INCONSISTENT ANALYSIS**



Lost Update Problem

2018/2019 - Bases de Dados - Fernanda Brito Correia

Transação A	Tempo	Transação B
Lê record R	T1	
	T2	Lê record R
Update record R	Т3	
	T4	Update record R

5

Uncommitted Dependency Problem

2018/2019 - Bases de Dados - Fernanda Brito Correia

Transação A	Tempo	Transação B
	T1	Update record R
Lê record R	T2	
	Т3	Rollback

Unc	ommitted	Depender	ncy Problem	
2018/2019 - Bases de Dados - Fernanda Brito Correia	Transação A	Tempo	Transação B	
Dados - Fe		T1	Update record R	
- Bases de	Update record R	T2		
2018/2019		Т3	Rollback	
7				

Inconsistent Analysis Problem					
ACC	ACC1 = 40 ACC2 = 50 ACC3 = 30				
Correia	Transação A	Tempo	Transação B		
A transação A está a somar três registos	Lê ACC1(40); soma = 40;	T1	a trans	sação B está sferir 10 do	
cuja soma daria 120	Lê ACC2(50); soma = 90;	Т2		ACC3 para	
ados		Т3	Lê ACC3(30)		
Bases de Dados		T4	Update ACC3: 30→20	-	
- 6		Т5	Lê ACC1(40);	_	
Total dado pela transa errado, será 110 em v		Т6	Update ACC1: 40→50	-	
120		T7	COMMIT		
Lê	ACC3(20); soma = 110;	Т8		-	
▶ 8					

Concorrência

Os três problemas podem ser todos resolvidos por meio de uma técnica de concorrência chamada LOCKs.

- ▶ Supondo que o sistema só admite dois tipos de locks:
 - ▶ Locks Exclusivos X (write locks);
 - ▶ Locks Partilhados S (read locks);

2018/2019 - Bases de Dados - Fernanda Brito Correia

9

Locks

Matriz de compatibilidade dos 2 tipos de locks

TA / TB	X	S	-
X	N	N	S
S	N	S	S
-	S	S	S

2018/2019 - Bases de Dados - Fernanda Brito Correia

Locks

Protocolo de acesso a dados ou protocolo de locks, que usa os locks X e S para resolver os problemas já descritos:

- Uma transação que deseje ler um tuplo, primeiro tem de adquirir um lock S sobre esse tuplo.
- ▶ Uma transação que deseje atualizar um tuplo deve adquirir um lock X sobre esse tuplo.
- Se a transação já tiver um lock S sobre o tuplo (como acontece numa sequência READ, UPDATE) a alternativa é promover o lock S a X.

11

2018/2019 - Bases de Dados - Fernanda Brito Correia

Locks

- Protocolo de acesso a dados ou protocolo de locks, que usa os locks X e S para resolver os problemas já descritos:
 - Se um pedido de lock da transação B for negada devido a um conflito com um lock já obtido pela transação A, a transação B entrará num estado de espera.
 - B esperará até que o lock de A seja libertado.
 - Os locks X são mantidos até ao fim da transação COMMIT ou ROLLBACK).
 - Normalmente os locks S também são mantidos até essa altura.

2018/2019 - Bases de Dados - Fernanda Brito Correia

Locks

- ▶ Pedidos de locks de tuplos feitos por transações, normalmente são implícitos.
- ▶ Vamos ver como resolver os três problemas de concorrência com o protocolo anterior:

13

I	ost Update Pr	oblem	
to Correia	Transação A	Tempo	Transação B
nanda Brit	Lê record R Obtém um lock S sobre R	T1	
ados - Feri		T2	Lê record R Obtém um lock S sobre R
- Bases de Dados - Fernanda Brito Correia	Update record R	Т3	
2018/2019 - E	Pedido de lock X sobre R, negado Wait	T4	Update record R Pedido de lock X sobre R,
2		DEADLOCK	negado Wait
>	14		

	Unc	ommitted	Dependen	ncy Problem
2018/2019 - Bases de Dados - Fernanda Brito Correia		Transação A	Tempo	Transação B
ernanda E	•		T1	Update record
Dados - F	Ped	Lê record R	T2	Obtém um lock X sobre R
Bases de	nega	ado Wait	T3	Rollback (synchpoint)
2018/2019 -	Obtér	Lê record R n um lock S sobre R	T4	Liberta lock X sobre R
•	1 5			

	I	-	ency Problem
	Transação A	Tempo	Transação в
		T1	Update record R Obtém um lock X sobre R
P	Update record R edido de lock X sobre R, negado	T2	
	Wait	Т3	Rollback (synchpoint)
	Update record R Obtém um lock X sobre R	T4	Liberta lock X sobre R

Inconsistent Analysis Problem					
	Transação A	Tempo	Transação B		
A transação A está a somar três registos	Lê ACC1(40); soma = 40;	T1	a trans	ação B está ferir 10 do	
cuja soma daria 120	Lê ACC2(50); soma = 90;	Т2	registo o registo	ACC3 para o ACC1	
ados		Т3	Lê ACC3(30)		
Bases de Dados		T4	Update ACC3: 30→20		
- 6 - 83		Т5	Lê ACC1(40);	•	
Total dado pela trans- errado, será 110 em	,	Т6	Update ACC1: 40→50		
120		Т7	СОММІТ	·	
L	ê ACC3(20); soma = 110;	Т8			

	nconsistent	Analy	vsis Probl	em A transação B está
A transação A está a somar três registos	$\Delta (T) = \Delta (T) \Delta (T) = 5$	50 ACC3 =30		a transferir 10 do
cuja soma daria 120		Tempo	Transação	B registo ACC3 para o registo ACC1
Obtém um lock S sobre ACC1	Lê ACC1(40); soma = 40;	T1		
Obtém um lock S sobre ACC2	Lê ACC2(50); soma = 90;	T2		
dos - Fe		Т3	Lê ACC3(30);	Obtém um lock S sobre ACC3
- Bases de Dados		T4	Update ACC3: 30→20	Obtém um lock X sobre ACC3
		T5	Lê ACC1(40);	Obtém um lock S sobre ACC1
2018/2019		Т6	Update ACC1;	Pedido de lock X sobre ACC1, negado
		T7		Wait
Pedido de lock sobre ACC3, negado	Le ACC3(20),	Т8		
▶ 18 Wai		DEADLOCK		

Deadlock

- > situação em que duas ou mais transações estão em estado de espera simultaneamente, cada uma à espera que a outra liberte os locks, antes de prosseguir.
 - Na prática quase nunca envolvem mais que duas transacções.
- É desejável que o sistema detete o DEADLOCK e o interrompa.
 - Detetar um ciclo no grafo de espera (Wait-For Graph), o grafo de quem está à espera de quem.

19

Deadlock

- Na prática usam um mecanismo de espera (time out)
- Assumem que uma transação que não tenha realizado qualquer trabalho durante esse período de tempo está em DEADLOCK
- Alguns sistemas reinicializarão essa transação de forma automática desde o início
 - supondo que as condições que causaram o DEADLOCK não se repetirão
- Outros sistemas enviarão um código de exceção de volta à aplicação e cabe ao programa lidar com a situação.
 - Ainda que o programador tenha de se envolver, é sempre desejável ocultar do utilizador final, por razões óbvias.

20

2018/2019 - Bases de Dados - Fernanda Brito Correia

 Serialização – critério de correção geralmente aceite para a execução de um dado conjunto de transações.

Uma dada execução de um conjunto de transações é considerada correta se for serializável.



▶ Se produz o mesmo resultado de alguma execução **serial** das mesmas transações(executadas uma a seguir à outra).

21

2018/2019 - Bases de Dados - Fernanda Brito Correia

Serialização

▶ Porquê? Porque:

- I- As transações individuais são consideradas corretas transformam um estado correto da base de dados noutro estado correto.
- 2- Assim a execução das transações em qualquer ordem serial também é correta.
- 3- Logo, uma transação intercalada é correta se é equivalente a alguma execução serial isto é se é serializável.

2018/2019 - Bases de Dados - Fernanda Brito Correia

- Dado um conjunto de transações:
- ► **Escalonamento** é qualquer execução dessas transacções intercaladas ou não.
- ▶ **Escalonamento serial** é a execução das transacções uma de cada vez sem intercalação.
- ▶ Escalonamento não serial é um escalonamento intercalado (ou simplesmente não serial).

23

2018/2019 - Bases de Dados - Fernanda Brito Correia

Serialização

- Dois escalonamentos são equivalentes se produzem o mesmo resultado, qualquer que seja o estado inicial da base de dados
- ► Um escalonamento é correto (isto é, serializável) se é equivalente a algum escalonamento serial

2018/2019 - Bases de Dados - Fernanda Brito Correia

 Dois escalonamentos seriais diferentes envolvendo o mesmo conjunto de transações poderiam produzir resultados diferentes.

Dais assalanamantas interna

- Dois escalonamentos intercalados diferentes envolvendo as mesmas transações poderiam produzir resultados diferentes e serem ambos considerados corretos.
- ▶ Dois escalonamentos de 2 transações A e B são corretos se forem equivalente a executar A depois de B ou B depois de A.

> 25

2018/2019 - Bases de Dados - Fernanda Brito Correia

Serialização

▶ Exemplo:

2018/2019 - Bases de Dados - Fernanda Brito Correia

Transação A: Somar I a x.

Transação B: duplicar x

com x algum item da base de dados, com valor inicial de x, 10

- Escalonamento serial A depois de B X = 21
- 2) Escalonamento serial **B** depois de A X = 22

▶ Teorema de bloqueio de duas fases (Eswaran et al.) (não tem nada a ver com o COMMIT de duas fases) — se duas transações obedecerem ao protocolo de bloqueio de duas fases então todos os escalonamentos intercalados possíveis são seriais.

2018/2019 - Bases de Dados - Fernanda Brito Correia

27

Serialização

Protocolo de bloqueio de duas fases:

- Antes de operar sobre qualquer objeto (ex. tuplo da base de dados) uma transação deve adquirir um bloqueio sobre esse objeto.
- 2. Depois de libertar um bloqueio, uma transação nunca deve adquirir outros bloqueios.
- Uma transação que obedece a este protocolo tem duas fases:
 - Fase da aquisição de bloqueio fase de "crescimento".
 - Fase de libertação de bloqueio fase de "encolhimento".

2018/2019 - Bases de Dados - Fernanda Brito Correia

▶ O protocolo de acesso a dados, que descrevemos atrás para os LOCKs X e S, pode ser considerado como uma forma robusta de protocolo de bloqueio de duas fases.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 29

Serialização

▶ Para melhorar o desempenho e o rendimento, os sistemas reais permitem a construção de transações que não tenham as duas fases (níveis de isolamento inferiores ao nível máximo de isolamento).



Estas transações são uma proposta arriscada.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 30

Nível de Isolamento

- ▶ **Nível de isolamento** é o grau de interferência que uma dada transação está preparada para tolerar por parte de transações concorrentes.
- Para garantir a serialização, não pode ser aceite nenhuma interferência
- O nível de isolamento tem de ser o máximo.
- Por questões práticas os sistemas reais em geral admitem que as transações operem em níveis de isolamento abaixo do máximo.

31

2018/2019 - Bases de Dados - Fernanda Brito Correia

Nível de Isolamento

- Pensando no **nível de isolamento** como uma propriedade de uma transação, podem ser definidos muitos níveis de isolamento diferentes
- ▶ O padrão SQL define apenas quatro.
- ▶ O DB2 tem atualmente dois:
 - ► REPEATABLE READ (RR)
 - ▶ CURSOR STABILITY (CS)
- Quanto mais alto o nível de isolamento, menor a interferência e mais baixo a concorrência.
- Quanto mais baixo o nível de isolamento, maior a interferência e mais alta a concorrência.

> 32

Nível de Isolamento

▶ REPEATABLE READ (RR):

- RR é o nível máximo.
- ▶ Se todas as transações operarem neste nível isso implica que todos os escalonamentos são serializáveis.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 33

Nível de Isolamento

Pontos importantes:

- RR é sempre uma escolha melhor que CS.
- Uma transação sob CS não é de duas fases e assim a serialização não pode ser garantida.
- No entanto, CS oferece mais concorrência que RR.
- Uma implementação que admite um nível de isolamento menor que o máximo, normalmente oferece recursos explícitos de controlo de concorrência – LOCKs explícitos – para os utilizadores poderem escrever as suas aplicações de modo que a segurança seja garantida na ausência de tal garantia do sistema.

Nível de Isolamento

O DB2 oferece LOCK TABLE explícito.

Os utilizadores que operem no nível CS podem adquirir LOCKs explícitos, além dos que o DB2 adquire de forma automática para impor esse nível.

▶ A caracterização RR – nível máximo – refere-se ao DB2.

O padrão SQL usa o mesmo termo RR para indicar um nível de isolamento estritamente mais baixo que o nível máximo (explicado mais à frente).

> 35

2018/2019 - Bases de Dados - Fernanda Brito Correia

Intenção de LOCK

Supondo que a unidade para fins de LOCK é o tuplo (registo)

▶ **Granularidade dos LOCKs** – é o tamanho dos objetos que podem ser bloqueados.

- Muitos sistemas oferecem várias opções de granularidade.
- Quanto mais fina a granularidade maior a concorrência.
- Quanto mais grosseira a granularidade menor o número de bloqueios que precisam de ser definidos e testados e mais baixa a sobrecarga.

▶ Protocolo de intenção de bloqueio (Intent lock):

▶ Nenhuma transação está autorizada a adquirir um bloqueio sobre um tuplo sem primeiro adquirir um bloqueio — uma intenção de bloqueio, provavelmente — sobre a variável de relação que o contém.

2018/2019 - Bases de Dados - Fernanda Brito Correia

37

Intenção de LOCK

- ▶ Os LOCKs X e S fazem sentido para:
 - Variáveis de relação inteiras
 - ▶ Tuplos individuais

Além destes LOCKs:

- ▶ Intenções de LOCKs (intent LOCK)
- Fazem sentido para variáveis de relação mas não para tuplos individuais.



- As intenções de LOCKs são:
 - ▶ Intent Share (IS)
 - ▶ Intent Exclusive (IX)
 - ▶ Share Intent Exclusive (SIX)

38

Intent Share (IS):

A transação T tem a intenção de definir LOCKs S sobre tuplos individuais de R para garantir a estabilidade desses tuplos quando estiverem a ser processados.

▶ Intent Exclusive (IX):

Igual a IS, só que a transação T pode atualizar tuplos individuais de R e assim definir LOCKs X sobre esses tuplos.

▶ Share (S):

- A transação T pode tolerar leituras concorrentes mas não atualizações concorrentes.
- A própria transação T não atualizará qualquer tuplo em R.

> 39

2018/2019 - Bases de Dados - Fernanda Brito Correia

Intenção de LOCK

Share Intent Exclusive (SIX):

- Combina S e IX, isto é a transação T pode tolerar leituras concorrentes, mas não atualizações concorrentes.
- ▶ Além disso a transação T tem intenção de pedir LOCKs X sobre esses tuplos individuais.

Exclusive (X):

A transação T não pode tolerar qualquer acesso concorrente a R.

2018/2019 - Bases de Dados - Fernanda Brito Correia

Matriz de compatibilidade estendida para incluir intent LOCKs:

orreia A	iau iz ue	Compaul	ollidade e	steriulua	рага птс	iuir <i>ii iiei i</i>	LUCKS:
- Bases de Dados - Fernanda Brito Correia		Х	SIX	IX	S	IS	-
rnanda	Х	N	N	N	N	N	S
dos - Fe	SIX	N	N	N	N	S	S
s de Da	IX	N	N	S	N	S	S
9 - Base	S	N	N	N	S	S	S
2018/2019	IS	N	S	S	S	S	S
50	_	S	S	S	S	S	S

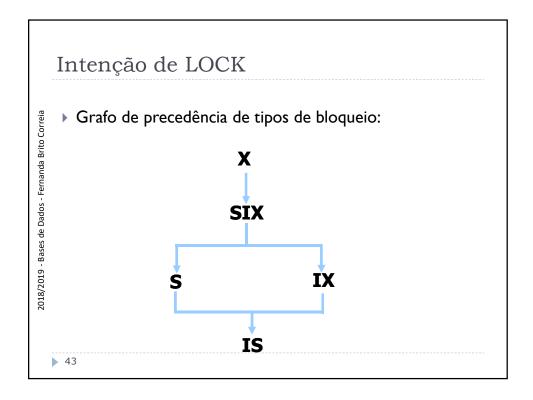
41

Intenção de LOCK

 Protocolo de intenção de LOCKs (enunciado mais preciso, mas ainda não completo)

- I. Antes de uma dada transação poder adquirir um LOCK S sobre um dado tuplo, primeiro deve adquiri um LOCK IS ou mais forte (ver à frente) sobre a variável de relação que contém esse tuplo.
- 2. Antes de uma dada transação poder adquirir um LOCK X sobre um dado tuplo, primeiro deve adquirir um LOCK IX ou mais forte (ver à frente) sobre a variável de relação que contém esse tuplo.

2018/2019 - Bases de Dados - Fernanda Brito Correia



▶ Um tipo de LOCK L2 é mais forte - isto é está numa posição mais alta do grafo – que o tipo de LOCK L1 se e só se sempre que houver um "N" na coluna L1 da matriz de compatibilidade para uma determinada linha também haverá um "N" na coluna L2 para a mesma linha.

44

▶ Um pedido de LOCK que falhe para um dado tipo de LOCK, certamente falhará para um tipo de LOCK mais forte.



• É mais seguro utilizar um tipo de LOCK que seja mais forte do que o estritamente necessário.

2018/2019 - Bases de Dados - Fernanda Brito Correia

45

Intenção de LOCK

- A escalada de bloqueios é implementada em muitos sistemas e parece funcionar bem na prática.
- ▶ Escalada de bloqueios é quando um sistema atinge um limiar predefinido, automaticamente ele substitui um conjunto de LOCKs de granularidade fina, por um conjunto de LOCKs de granularidade mais grossa.
 - Ex. trocando um conjunto de LOCKs S ao nível de tuplos individuais e convertendo um LOCK IS sobre a variável de relação que contém os tuplos, num bloqueio S.

2018/2019 - Bases de Dados - Fernanda Brito Correia

Recursos de SQL

O SQL padrão não fornece qualquer recurso explícito de bloqueio.

▶ O SQL padrão exige que a implementação dê as garantias habituais quanto à interferência ou ausência de interferência entre transações em execução concorrente.

- ▶ O SQL padrão exige que as atualizações feitas por uma determinada transação TI não sejam visíveis para qualquer transação T2 distinta, a não ser que TI termine com COMMIT.
 - Pressupõe que as transações são executadas num dos níveis de isolamento:
 - ▶ READ UNCOMMITED
 - READ COMMITED
 - ▶ REPEATABLE READ
 - ▶ SERIALIZABLE

47

2018/2019 - Bases de Dados - Fernanda Brito Correia

Níveis de Isolamento no SQL

▶ O SQL inclui uma instrução SET TRANSACTION que define certas características da próxima transação a ser iniciada.

- ▶ Uma delas é o nível de isolamento:
 - ▶ Padrão : SERIALIZABLE
 - ▶ Se algum dos outros três níveis for especificado a implementação será livre de atribuir algum nível mais alto, de acordo com a ordem seguinte:



48

- Se todas as transações forem executadas no nível de isolamento SERIALIZABLE (por defeito) então a execução intercalada de qualquer conjunto de transações concorrente terá a garantia de ser serializável.
- Se a transação for executada num nível de isolamento menor então a serializabilidade poderá ser violada de várias maneiras diferentes.
- O SQL padrão define três maneiras especificas, nas quais a serializabilidade poderá ser violada:
 - DIRTY READ
 - NON REPEATABLE READ
 - PHANTOM

49

2018/2019 - Bases de Dados - Fernanda Brito Correia

Níveis de Isolamento no SQL

DIRTY READ:

- ▶ Supondo que a transação TI executa uma atualização numa linha.
- ▶ Então a transação T2 lê essa linha e a transação T1 termina em seguida, com um ROLLBACK.
- A transação T2 enviou uma linha que já não existe, e que de alguma maneira nunca existiu, porque a transação T1 nunca foi executada.

T1	Tempo	T2
Update record R	T1	
	T2	Lê record R
ROLLBACK	T3	

DIRTY READ

NON REPEATABLE READ:

- ▶ Supõe-se que a transação TI lê uma linha,
- depois a transação T2 atualiza essa linha,
- e a transação TI lê novamente a mesma linha.
- A transação TI leu duas vezes a mesma linha, mas viu duas vezes a "mesma" linha, mas viu dois valores diferentes para ela.

T1	Tempo	T2
Lê record R	T1	
	T2	Update record R
Lê record R	T3	

NON REPEATABLE READ

51

2018/2019 - Bases de Dados - Fernanda Brito Correia

Níveis de Isolamento no SQL

▶ PHANTOM

- Supõe-se que a transação TI leu o conjunto de todas as linhas que satisfazem a alguma condição (por exemplo, todas as linhas do departamento 10).
- Supõe-se que a transação T2 insere em seguida uma nova linha que satisfaz à mesma condição.
- Se TI repetir a sua solicitação de leitura, verá uma nova linha que não existia antes – uma linha "fantasma".

T1	Tempo	T2
Lê records NDEP=10	T1	
	T2	Insert record NDEP=10
Lê records NDEP=10	Т3	

Os vários níveis de isolamento são definidos, em termos de quais as violações de serializabilidade descritas, eles permitem.

Nível de Isolame		DIRTY READ	NONREPEATABLE READ	PHANTOM
READ UI	NCOMMITTED	S	S	S
READ CO	DMMITTED	N	S	S
REPEATA	BLE READ	N	N	S
SERIALI	ZABLE	N	N	N

- "S" significa que a violação indicada pode ocorrer.
- "N" − significa que não pode.

53

2018/2019 - Bases de Dados - Fernanda Brito Correia

Níveis de Isolamento no SQL

Como pode o sistema impedir a ocorrência de PHANTOM?

Precisa de bloquear o caminho de acesso usado para obter os dados em questão.

Exemplo:

- ▶ No exemplo mencionado acima sobre os fornecedores de Paris, se acontecer que o caminho de acesso seja um índice sobre as cidade dos fornecedores, então o sistema deve bloquear a entrada Paris nesse índice.
- O que impedirá a criação de fantasmas uma vez que a inserção exigiria que o caminho de acesso fosse atualizado.

2018/2019 - Bases de Dados - Fernanda Brito Correia

- Não esquecer que REPEATABLE READ (RR) em SQL e REPEATABLE READ (RR) em DB2 não são a mesma coisa.
- ▶ A operação RR do DB2 é igual à operação SERIALIZABLE do SQL.

55

2018/2019 - Bases de Dados - Fernanda Brito Correia

Bloqueamento (Locking) no ORACLE

- ▶ O Oracle Server bloqueia as linhas que são atualizadas pelos comandos DML (UPDATE, DELETE e INSERT) enquanto a transação não terminar.
- ▶ Todas as operações serão permitidas nas linhas bloqueadas, exceto novamente operações de DML.
- As operações de leitura sobre linhas que foram bloqueadas fornecem uma visão do seu estado antes da alteração.
- Depois de terminada a transação, todos os LOCKS são removidos das linhas
- ▶ Caso as atualizações tenham sido confirmadas (COMMIT), então as operações de leitura vão fornecer a imagem modificada das linhas outrora bloqueadas.

> 56

O Oracle garante a consistência de leitura

- garantindo que os dados visíveis por uma transação são consistentes com a informação confirmada num determinado instante no tempo
- garantindo que leituras dos dados da base de dados, não esperam por processos que estejam a alterar os mesmos dados
- garantindo que a escrita dos dados na base de dados, não esperam por processos que estejam a ler os mesmos dados
- garantindo que a escrita dos dados na base de dados, apenas esperam por processos que estejam a alterar os mesmos dados

57

2018/2019 - Bases de Dados - Fernanda Brito Correia

Bloqueamento (Locking) no ORACLE Transacção A (Utilizador U1) Transacção B (Utilizador U2) Update livro A linha de código 10 está bloqueada. O 2018/2019 - Bases de Dados - Fernanda Brito Correia valor que o utilizador U2 vê é o valor Set titulo = 'Olhos de Cão' anterior à alteração. Where codigo = 10; Se a linha com o código 10 for referenciada por algum comando UPDATE ou DELETE a sessão deste utilizador fica bloqueada. Delete UI.livro A linha de código 10 continua bloqueada pelo utilizador UI, mas agora a linha de Where codigo = 20 código 20 ficou bloqueada pelo utilizador U2. Se algum outro utilizador U3 fizer uma consulta à tabela LIVRO, não vai ver nenhuma das alterações e só poderá efetuar operações de consulta nas linhas bloqueadas. 58

 O utilizador pode alterar o comportamento anterior, bloqueando explicitamente uma tabela com o comando LOCK TABLE

```
LOCK TABLE {<nome_da_tabela>|<nome_da_vista>}

[@<database_link>]

IN { ROW SHARE | SHARE UPDATE |

ROW EXCLUSIVE | SHARE |

SHARE ROW EXCLUSIVE |

EXCLUSIVE } MODE [NOWAIT];
```

Deve ser emitido no inicio da transacção de modo a sobrepor-se ao mecanismo de bloqueamento interno implementado pelo *Oracle Server*. O lock é libertado quando a transacção terminar.

> 59

2018/2019 - Bases de Dados - Fernanda Brito Correia

Bloqueamento (Locking) no ORACLE

ROW EXCLUSIVE:

- ▶ É o mecanismo de LOCK implementado, por defeito, pelo Oracle Server, quando são emitidos comandos de DML.
- Impede que outro utilizador possa obter qualquer tipo de LOCK sobre as mesmas linhas, onde se encontra este LOCK.

EXCLUSIVE:

- permite apenas que sejam feitas consultas na tabela, deixando para a transação que bloqueou a tabela, as permissões para emitir todo o tipo de operações.
- Exemplo:
 - ▶ LOCK TABLE LIVRO IN EXCLUSIVE MODE;

▶ ROW SHARE:

Permite que todos os utilizadores partilhem a informação da tabela, mas impede qualquer utilizador de adquirir um LOCK em modo EXCLUSIVE para a mesma tabela.

▶ SHARE UPDATE:

O modo de bloqueamento SHARE UPDATE é mantido apenas para compatibilização e é o mesmo que o modo ROW SHARE.

61

2018/2019 - Bases de Dados - Fernanda Brito Correia

Bloqueamento (Locking) no ORACLE

▶ SHARE:

O modo de bloqueamento SHARE permite que mais do que um utilizador adquira o LOCK SHARE sobre a mesma tabela, mas impede-os de emitirem comandos DML sobre a tabela.

SHARE ROW EXCLUSIVE:

Permite que os outros utilizadores consultem a tabela, mas impede-os de adquirirem outros LOCKs explicitamente ou de emitirem comandos DML.

62

_		Tipo	Locks Existentes				
Correia	Comando SQL	Lock	RS	RX	S	SRX	Χ
	SELECTFROM table	nenhum	Y	Y	Y	Y	Y
Brito	INSERT INTO table	RX	Υ	Υ	N	N	N
_	UPDATE table	RX	Y*	Y*	N	N	N
ernanda	DELETE FROM table	RX	Y*	Y*	Ν	N	N
Fer	SELECT FROM FOR UPDATE OF	RS	Y*	Y*	Y*	Y*	N
- SO	LOCK TABLE table IN						
Dados	ROW SHARE MODE	RS	Υ	Υ	Υ	Υ	N
de	SHARE MODE	S	Υ	N	Y	N	N
Bases	ROW EXCLUSIVE MODE	RX	Υ	Υ	Ν	N	N
-Ba	SHARE ROW EXCLUSIVE MODE	SRX	Y	Ν	N	N	N
119	EXCLUSIVE MODE	X	N	N	N	N	N

X: exclusive

RX: row exclusive

RS: row share

S: share

SRX: share row exclusive

* se os registos não estiverem a ser bloqueados por outra transacção, senão espera

63

Bloqueamento (Locking) no ORACLE

Cláusula NOWAIT

- destina-se a evitar que o pedido de obtenção do lock fique à espera que outros locks eventualmente sejam libertados da tabela
- ▶ Com esta clausula o pedido falha e o utilizador pode tentar mais tarde sem ficar "pendurado".
 - Exemplo de pedido de lock a uma tabela que já estava bloqueada (implícita ou explicitamente):

SQL> lock table livro in share row exclusive mode nowait; lock table livro in share row exclusive mode nowait

*
ERROR at line 1:
ORA-00054: resourse busy and acquire with NOWAIT specified

O comando SELECT apresenta ainda uma cláusula que permite bloquear a tabela, durante a operação de pesquisa.

A clausula chama-se FOR UPDATE e é usada assim:

65

2018/2019 - Bases de Dados - Fernanda Brito Correia

Bloqueamento (Locking) no ORACLE

- Com a cláusula FOR UPDATE são bloqueadas as linhas afetadas pelo comando SELECT.
- No caso de existir um JOIN podem ser especificadas quais as tabelas que se desejam bloquear, indicando também as colunas.
- A clausula NOWAIT tem exatamente o mesmo comportamento do comando LOCK TABLE,
 - ou seja, seleciona as linhas, mas se as linhas já tiverem sido bloqueadas num modo que não seja SHARE,
 - então não bloqueia as linhas selecionadas.

2018/2019 - Bases de Dados - Fernanda Brito Correia

Exemplo:

2018/2019 - Bases de Dados - Fernanda Brito Correia

Útil para implementar um sistema de acesso síncrono a uma tabela, permitindo apenas que os dados sejam acedidos e alterados por um utilizador de cada vez.