

INF / UFG

Disciplina
Banco de Dados

Conteúdo

**Controle de Concorrência:
Granularidade de Itens de Dados**



Preâmbulo

Todas as técnicas de controle de concorrência assumem que o **banco de dados é formado por itens de dados**.

Um item de banco de dados pode ser:

- >> um registro do banco de dados
- >> um campo de um registro do banco de dados
- >> um bloco de disco
- >> um arquivo inteiro
- >> um banco de dados inteiro

A granularidade pode afetar a performance de controle de concorrência e recuperação.



Impacto de Granularidade no Bloqueio de Dados

Granularidade

O tamanho de itens de dados é chamado de **granularidade**.

Granularidade fina refere-se a itens de tamanho pequeno.

Granularidade grossa refere-se a itens de tamanho grande.

Qual a granularidade a ser adotada ?

No contexto de bloqueio ...



Impacto de Granularidade no Bloqueio de Dados

Granularidade no contexto de bloqueio ...

Quanto maior a granularidade, menor a taxa de concorrência.

Exemplo 1:

- >> se a granularidade for **bloco de disco**, uma Transação **T** que necessita bloquear um Registro **R** deverá bloquear o Bloco **X** inteiro que contém o registro **R**.
- >> se outra transação **T*** precisar bloquear o Registro **S**, que também reside em **X**, deverá aguardar o desbloqueio de **X** por parte de **T**.

Exemplo 2:

- >> se no exemplo anterior, a granularidade for **registro do banco de dados**, **T** e **T*** poderiam prosseguir sem **T*** aguardar por **T**.



Impacto de Granularidade no Bloqueio de Dados

Granularidade no contexto de bloqueio ...

Quanto menor a granularidade, maior a taxa de concorrência.

Cada item de dado é associado a uma operação de bloqueio.

O sistema terá um grande número de **bloqueio ativos**:

- >> mais operações de bloqueio e de desbloqueio,
- >> mais *overhead*;
- >> mais espaço de armazenamento é requerido para a tabela de bloqueios.

Analogamente, em protocolos baseados em *timestamps*, uma granularidade mais fina requer um maior número de itens de dados para o controle de **read_TS** e **write_TS**, resultando em maior *overhead*.



Impacto de Granularidade no Bloqueio de Dados

Então, qual a granularidade a ser adotada ?

Depende do tipo de transações envolvidas.

Se uma transação acessa um pequeno número de registros, é vantajoso ter granularidade em nível de registro (granularidade fina).

Se a transação tipicamente acessa muitos registros no mesmo arquivo, pode ser melhor adotar granularidade em nível de bloco ou arquivo (granularidade grossa), visto que a transação considerará todos os registros como poucos (ou apenas um único) itens de dados.



Bloqueio em Múltiplos Níveis de Granularidade

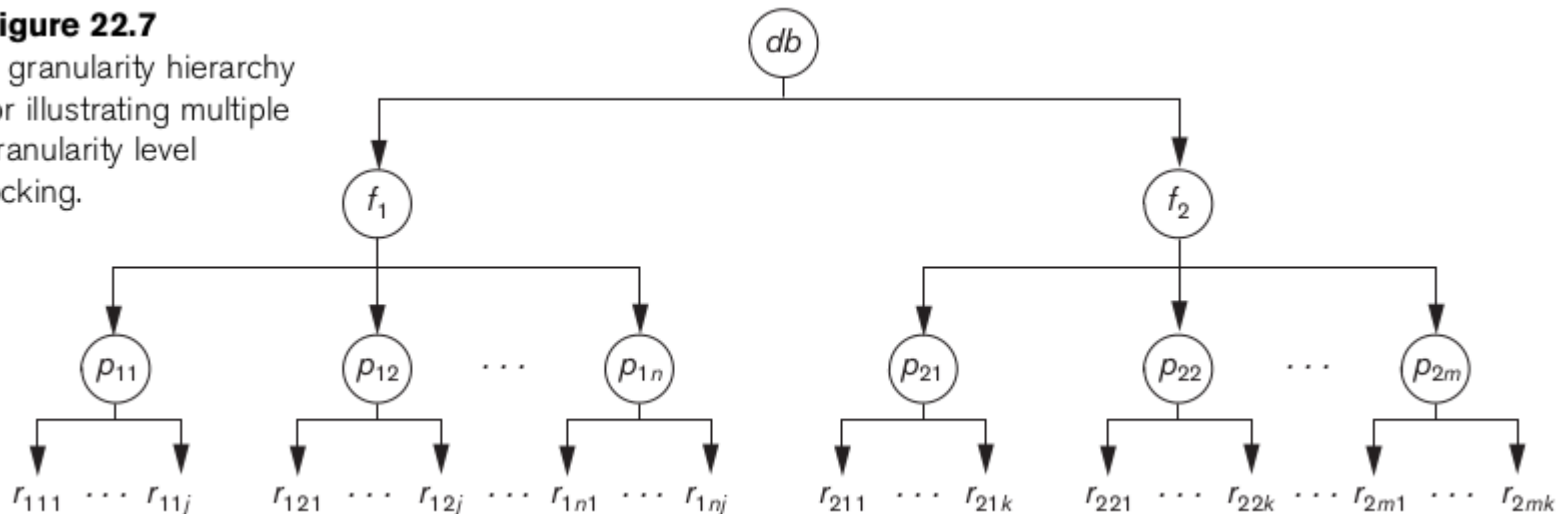
Um SGBD pode suportar múltiplos níveis de granularidade ...

Uma hierarquia de granularidade – banco de dados com 2 arquivos:

- >> cada arquivo possui várias páginas em disco;
- >> cada página contém vários registros.

Figure 22.7

A granularity hierarchy for illustrating multiple granularity level locking.



Bloqueio em Múltiplos Níveis de Granularidade

Suponha que a Transação **T1** precisa atualizar todos os registros do Arquivo **f1**, ou seja, **T1** solicita bloqueio exclusivo de **f1** :

>> requer o bloqueio de todas as páginas e registros de **f1**.

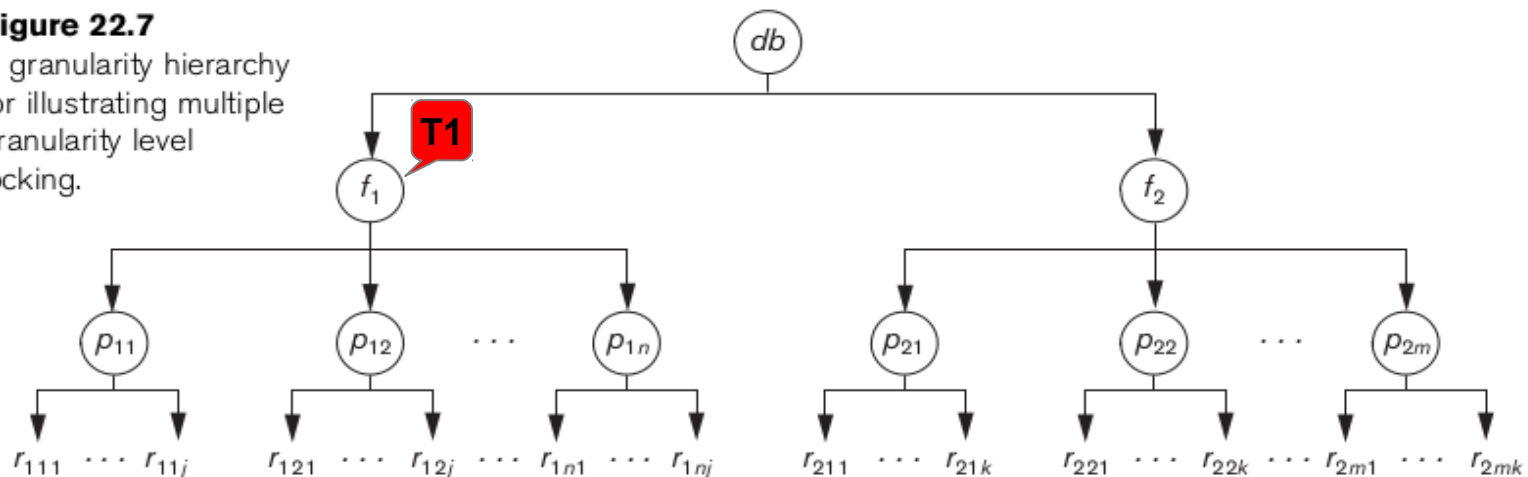
T1 alcança o bloqueio de **f1** (um único bloqueio com granularidade grossa):

>> bloqueio em nível de arquivo é mais eficiente que:

bloqueio de **n** páginas ou bloqueio de registros individuais.

Figure 22.7

A granularity hierarchy for illustrating multiple granularity level locking.



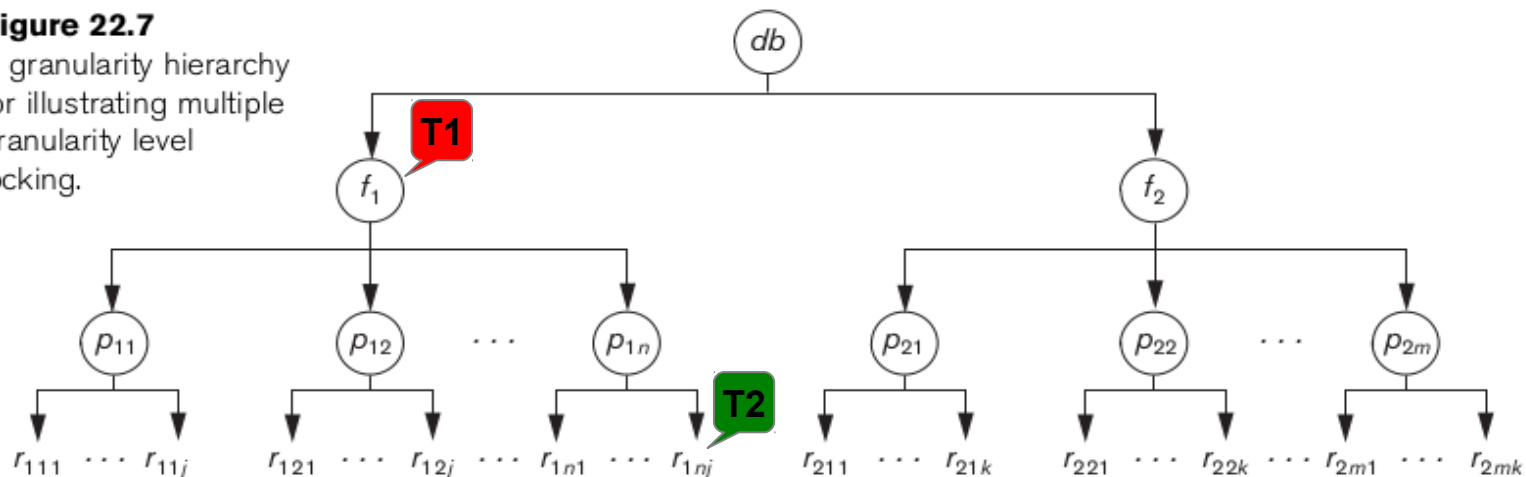
Bloqueio em Múltiplos Níveis de Granularidade

Ainda, outra Transação T2 precisa ler o Registro r_{1nj} da Página p_{1n} do Arquivo f_1 :
 >> T2 solicita bloqueio compartilhado do Registro r_{1nj} .

Como checar a compatibilidade da requisição com bloqueios já existentes ?
 >> no exemplo, atravessar a hierarquia, da folha até a raiz;
 >> a requisição de T2 é negada, pois T1 já bloqueia f_1 de forma exclusiva;
 >> T2 precisará esperar.

Figure 22.7

A granularity hierarchy for illustrating multiple granularity level locking.



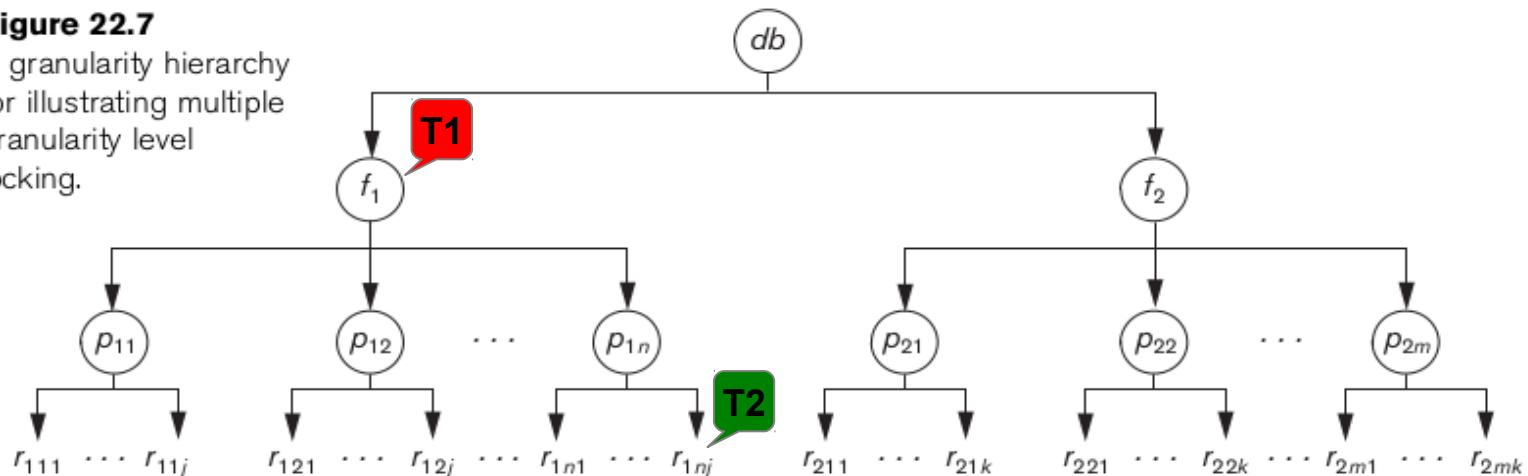
Bloqueio em Múltiplos Níveis de Granularidade

Em outro cenário, T2 solicitou o bloqueio antes de T1:

- >> T2 alcança o bloqueio compartilhado do Registro r_{1nj} ;
- >> T1 alcançará o bloqueio exclusivo de Arquivo f_1 ?
- >> para responder, é preciso checar todos os nós descendentes de f_1 ;
- >> é uma tarefa ineficiente ...

Figure 22.7

A granularity hierarchy for illustrating multiple granularity level locking.



Bloqueio em Múltiplos Níveis de Granularidade

Intenção de bloqueio ...

Uma transação pode indicar, ao longo do caminho do nó raiz até o nó desejado, que tipo de bloqueio (compartilhado ou exclusivo) pretende realizar.

Intention-shared (IS). Indica que um ou mais bloqueios compartilhados serão requisitados em nós descendentes.

Intention-exclusive (IX). Indica que um ou mais bloqueios exclusivos serão requisitados em nós descendentes.

Shared-intention-exclusive (SIX). Indica que o nó corrente é bloqueado em modo compartilhado, mas um ou mais bloqueios exclusivos serão requisitados em nós descendentes.



Bloqueio em Múltiplos Níveis de Granularidade

Exemplo:

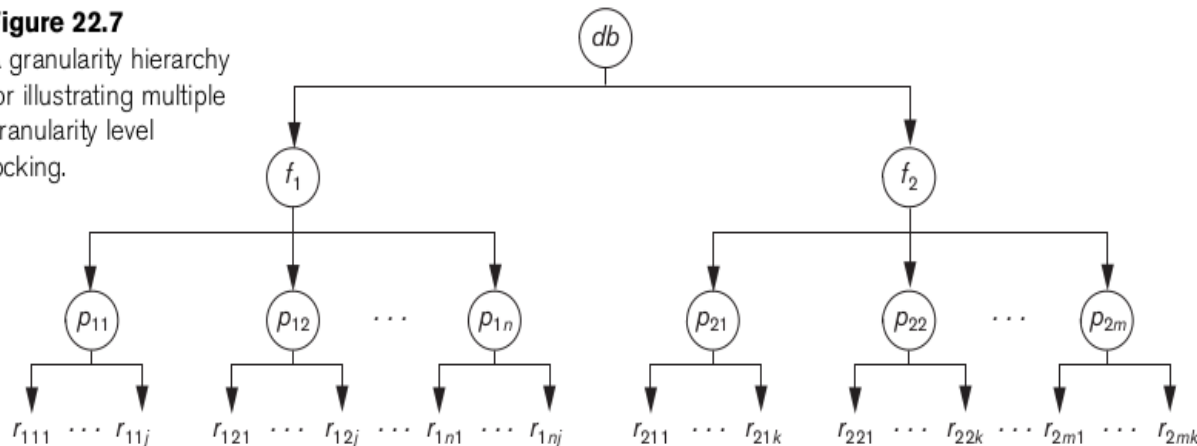
T1 precisa atualizar os Registros **r111** e **r211**.

T2 precisa atualizar todos os registros da Página **p12**.

T3 precisa ler o Registro **r11j** e o arquivo **f2** inteiro.

Figure 22.7

A granularity hierarchy for illustrating multiple granularity level locking.



T_1	T_2	T_3
IX(db) IX(f ₁)	IX(db)	IS(db) IS(f ₁) IS(p ₁₁)
IX(p ₁₁) X(r ₁₁₁)	IX(f ₁) X(p ₁₂)	S(r _{11j})
IX(f ₂) IX(p ₂₁) X(p ₂₁₁)		
unlock(r ₂₁₁) unlock(p ₂₁) unlock(f ₂)		S(f ₂)
	unlock(p ₁₂) unlock(f ₁) unlock(db)	
unlock(r ₁₁₁) unlock(p ₁₁) unlock(f ₁) unlock(db)		unlock(r _{11j}) unlock(p ₁₁) unlock(f ₁) unlock(f ₂) unlock(db)



Bloqueio em Múltiplos Níveis de Granularidade

Intenção de bloqueio ...

O protocolo de múltiplos níveis de granularidade é útil quando há:

- (1) transações curtas que acessam poucos itens de dados (registros ou campos), em conjunto com
- (2) transações longas que acessam arquivos inteiros.

Nesse cenário, o protocolo de múltiplos níveis é mais eficiente do que um protocolo de um único nível.

