

# Sistemas de Banco de Dados Projeto, implementação e gerenciamento

## *Capítulo 10*

Gerenciamento de transações e controle  
de concorrência

# Objetivos

- **Neste capítulo, você aprenderá:**
  - Sobre transações de bancos de dados e suas propriedades
  - O que é controle de concorrência e que papel executa na manutenção da integridade dos bancos de dados
  - O que são métodos de bloqueio e como funcionam

# Objetivos (cont.)

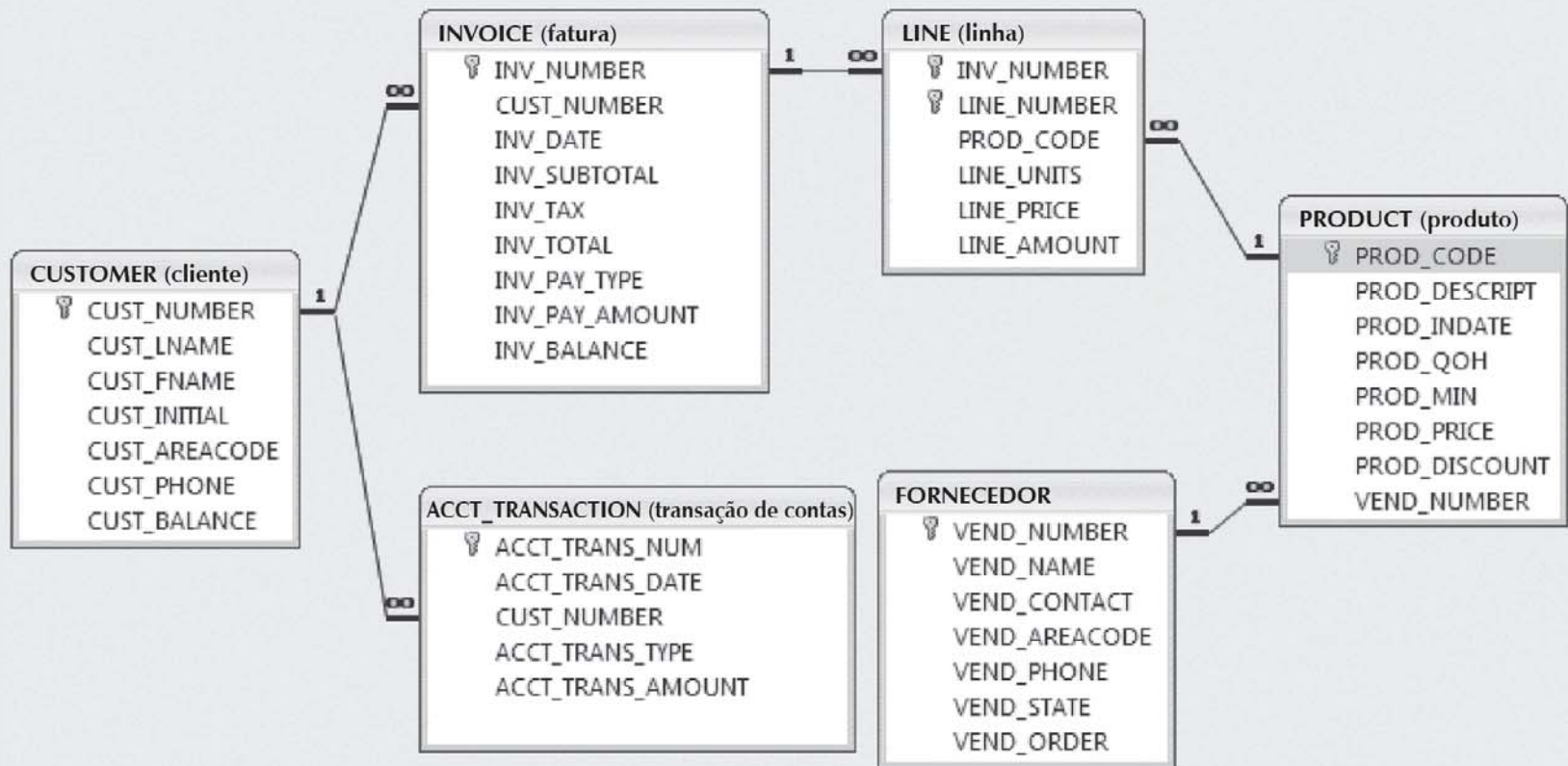
- Como os métodos de time stamping são utilizados para controle de concorrência
- Como os métodos otimistas são utilizados para controle de concorrência
- Como o gerenciamento de recuperação de bancos de dados é utilizado para manter a integridade

# O que é uma Transação?

- Unidade *lógica* de trabalho que deve ser concluída ou abortada inteiramente
- Uma transação bem-sucedida altera o banco de dados de um estado consistente para outro
- Um **banco de dados em estado consistente** é aquele em que são satisfeitas as restrições de integridade de todos os dados
- A maioria das transações reais é formada por duas ou mais solicitações
  - A **solicitação de banco de dados** é o equivalente a um único comando de SQL em um aplicativo ou transação

**FIGURA 10.1**

**Diagrama relacional do banco de dados**



# Avaliação dos Resultados de Transações

- Nem todas as transações atualizam o banco de dados
- O código de SQL representa uma transação, pois *acessa* o banco de dados
- Transações inadequadas ou incompletas podem ter efeito devastador sobre a integridade do banco de dados
  - Alguns SGBDs fornecem meios pelos quais os usuários podem definir restrições aplicáveis
  - Outras regras são aplicadas automaticamente pelo SGBD

# FIGURA 10.2

## Rastreado a transação no banco de dados

Nome da tabela: INVOICE (fatura)

INV_NUMBER	CUST_NUMBER	INV_DATE	INV_SUBTOTAL	INV_TAX	INV_TOTAL	INV_PAY_TYPE	INV_PAY_AMOUNT	INV_BALANCE
1001	10014	16-Jan-08	54.92	4.39	59.31	cc	59.31	0.00
1002	10011	16-Jan-08	9.98	0.80	10.78	cash	10.78	0.00
1003	10012	16-Jan-08	270.70	21.66	292.36	cc	292.36	0.00
1004	10011	17-Jan-08	34.87	2.79	37.66	cc	37.66	0.00
1005	10018	17-Jan-08	70.44	5.64	76.08	cc	76.08	0.00
1006	10014	17-Jan-08	397.83	31.83	429.66	cred	100.00	329.66
1007	10015	17-Jan-08	34.97	2.80	37.77	chk	37.77	0.00
1008	10011	17-Jan-08	1033.08	82.65	1115.73	cred	500.00	615.73
1009	10016	18-Jan-08	256.99	20.56	277.55	cred	0.00	277.55

Nome da tabela: LINE (linha)

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE	LINE_AMOUNT
1001	1	13-Q2/P2	3	14.99	44.97
1001	2	23109-HB	1	9.95	9.95
1002	1	54778-2T	2	4.99	9.98
1003	1	2238/QPD	4	38.95	155.80
1003	2	1546-QQ2	1	39.95	39.95
1003	3	13-Q2/P2	5	14.99	74.95
1004	1	54778-2T	3	4.99	14.97
1004	2	23109-HB	2	9.95	19.90
1005	1	PVC23DRT	12	5.87	70.44
1006	1	SM-18277	3	6.99	20.97
1006	2	2232/QTY	1	109.92	109.92
1006	3	23109-HB	1	9.95	9.95
1006	4	89-WRE-Q	1	256.99	256.99
1007	1	13-Q2/P2	2	14.99	29.98
1007	2	54778-2T	1	4.99	4.99
1008	1	PVC23DRT	5	5.87	29.35
1008	2	WR3/TT3	4	119.95	479.80
1008	3	23109-HB	1	9.95	9.95
1008	4	89-WRE-Q	2	256.99	513.98
1009	1	89-WRE-Q	1	256.99	256.99

Nome da tabela: PRODUCT (produto)

PROD_CODE	PROD_DESCRIPTOR	PROD_INDATE	PROD_QOH	PROD_MIN	PROD_PRICE	PROD_DISCOUNT	VEND_NUMBER
11GER/31	Power painter, 15 psi., 3-nozzle	03-Nov-07	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-07	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-07	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-08	15	8	39.95	0.00	23119
1558-QMVI	Hrd. cloth, 1/2-in., 3x50	15-Jan-08	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-07	8	5	109.92	0.05	24288
2232/QMVE	B&D jigsaw, 8-in. blade	24-Dec-07	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-08	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-08	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-08	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-07	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Jan-08	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	06-Jan-08	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-08	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-08	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/8", 5" mesh	17-Jan-08	18	5	119.95	0.10	25595

Nome da tabela: CUSTOMER (cliente)

CUST_NUMBER	CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE	CUST_BALANCE
10010	Ramas	Alfred	A	615	844-2573	0.00
10011	Dunne	Leona	K	713	894-1238	615.73
10012	Smith	Kathy	vv	615	894-2285	0.00
10013	Olowski	Paul	F	615	894-2180	0.00
10014	Orlando	Myron		615	222-1672	0.00
10015	O'Brian	Arvy	B	713	442-3381	0.00
10016	Brown	James	G	615	297-1228	277.55
10017	Williams	George		615	290-2556	0.00
10018	Farriss	Anne	G	713	382-7185	0.00
10019	Smith	Olette	K	615	297-3809	0.00

Nome da tabela: ACCT TRANSACTION (transação de conta)

ACCT_TRANS_NUM	ACCT_TRANS_DATE	CUST_NUMBER	ACCT_TRANS_TYPE	ACCT_TRANS_AMOUNT
10003	17-Jan-08	10014	charge	329.66
10004	17-Jan-08	10011	charge	615.73
10006	29-Jan-08	10014	payment	329.66
10007	18-Jan-08	10016	charge	277.55

# Propriedades das Transações

- **Indivisibilidade**
  - Todas as operações de uma transação estejam concluídas
- **Consistência**
  - Indica a permanência do estado consistente do banco de dados
- **Isolamento**
  - Os dados utilizados durante a execução de uma transação não podem ser utilizados por uma segunda transação até que a primeira seja concluída



# Propriedades das Transações (cont.)

- **Durabilidade**

- Uma vez feitas alterações pelas transações, elas não podem ser desfeitas ou perdidas

- **Serializável**

- O escalonador da execução atual das transações produza resultados consistentes
- É importante em bancos de dados distribuídos e de multiusuário, em que várias transações provavelmente serão executadas de modo simultâneo

# Gerenciamento de Transações com SQL

- O Instituto Nacional Americano de Padrões (ANSI) definiu os padrões que determinam as transações de bancos de dados em SQL
- O suporte a transações é fornecido por dois comandos de SQL: COMMIT e ROLLBACK
- A sequência de transação deve continuar até que:
  - Chegar a um comando COMMIT
  - Chegar a um comando ROLLBACK
  - Chegar com sucesso ao fim de um programa
  - O programa seja encerrado de modo anormal

# Log de Transações

- O log de transações armazena:
  - Um registro do início da transação
  - Para cada componente de transação (comando de SQL):
    - Tipo de operação sendo executada (atualização, exclusão, inserção)
    - Nomes dos objetos afetados pela transação (nome da tabela)
    - Valores de “antes” e “depois” nos campos sendo atualizados
    - Ponteiros para as entradas anteriores e posteriores da mesma transação no log
  - O fim (COMMIT) da transação

**TABELA 10.1 Log de transações**

TRL_ID	TRX_NUM	PTR ANTERIOR	PTR SEGUINTE	OPERAÇÃO	TABELA	ID DE LINHA	ATRIBUTO	VALOR ANTES	VALOR DEPOIS
341	101	Nulo	352	START	****Início da Transação				
352	101	341	363	UPDATE	PRODUCT	1558-QW1	PROD_QOH	25	23
363	101	352	365	UPDATE	CLIENTE	10011	CUST_BALANCE	525,75	615,73
365	101	363	Nulo	COMMIT	**** Fim da Transação				



**TRL\_ID** = ID de registro no log de transações

**PTR** = Ponteiro para uma ID de registro no log de transações

**TRX\_NUM** = Número de transação

**(Observação:** O número de transação é atribuído automaticamente pelo SGBD.)

# Controle de Concorrência

- A coordenação da execução simultânea de transações em um sistema de banco de dados multiusuário
- O objetivo desse controle é garantir a serialização das transações nesse tipo de ambiente

# Atualizações Perdidas

- O problema das **atualizações perdidas**:
  - Ocorre quando duas transações concorrentes, estão atualizando o mesmo elemento de dados
  - Uma das atualizações é perdida
    - Sobrescrita por outra aplicação

**TABELA 10.2 Duas transações concorrentes para atualizar a quantidade disponível**

TRANSAÇÃO	COMPUTAÇÃO
T1: compra 100 unidades	$PROD\_QOH = PROD\_QOH + 100$
T2: vende 30 unidades	$PROD\_QOH = PROD\_QOH - 30$

**TABELA 10.3 Execução em série das duas transações**

TEMPO	TRANSAÇÃO	ETAPA	VALOR ARMAZENADO
1	T1	Leitura de PROD_QOH	35
2	T1	$PROD\_QOH = 35 + 100$	
3	T1	Gravação de PROD_QOH	135
4	T2	Leitura de PROD_QOH	135
5	T2	$PROD\_QOH = 135 - 30$	
6	T2	Gravação de PROD_QOH	105

# Dados não Consolidados

- O fenômeno de **dados não consolidados (não salvos)**:
  - Ocorre quando duas transações são executadas de modo concorrente
  - A primeira é desfeita após a segunda ter acessado os dados não consolidados



**TABELA 10.4 Atualizações perdidas**

TEMPO	TRANSAÇÃO	ETAPA	VALOR ARMAZENADO
1	T1	Leitura de PROD_QOH	35
2	T2	Leitura de PROD_QOH	35
3	T1	$PROD\_QOH = 35 + 100$	
4	T2	$PROD\_QOH = 35 - 30$	
5	T1	Gravação PROD_QOH ( <b>Atualização perdida</b> )	135
6	T2	Gravação de PROD_QOH	5

**TABELA 10.5 Transações que criam o problema de dados não consolidados**

TRANSAÇÃO	COMPUTAÇÃO
T1: compra 100 unidades	$PROD\_QOH = PROD\_QOH + 100$ ( <b>Desfeita</b> )
T2: vende 30 unidades	$PROD\_QOH = PROD\_QOH - 30$

# Recuperações Inconsistentes

- **Recuperações inconsistentes:**
  - Ocorrem quando uma transação acessa dados antes e após outras transações terminarem de trabalhar com esses dados
- A transação pode ler alguns dados antes de serem alterados e outros *após* a alteração, produzindo, assim, resultados inconsistentes

**TABELA 10.8 Recuperação durante atualização**

TRANSAÇÃO T1		TRANSAÇÃO T2	
SELECT	SUM(PROD_QOH)	UPDATE	PRODUCT
FROM	PRODUCT	SET	PROD_QOH = PROD_QOH + 10
		WHERE	PROD_CODE = '1546-QQ2'
		UPDATE	PRODUCT
		SET	PROD_QOH = PROD_QOH - 10
		WHERE	PROD_CODE = '1558-QW1'
		COMMIT;	

**TABELA 10.9 Resultados da transação: correção de entrada de dados**

	ANTES	DEPOIS
PROD_CODE	PROD_QOH	PROD_QOH
11QER/31	8	8
13-Q2/P2	32	32
1546-QQ2	15	(15 + 10) → 25
1558-QW1	23	(23 - 10) → 13
2232-QTY	8	8
2232-QWE	6	6
Total	92	92

**TABELA 10.10 Recuperações inconsistentes**

TEMPO	TRANSAÇÃO	AÇÃO	VALOR	TOTAL
1	T1	Leitura de PROD_QOH para PROD_CODE = '11QER/31'	8	8
2	T1	Leitura de PROD_QOH para PROD_CODE = '13-Q2/P2'	32	40
3	T2	Leitura de PROD_QOH para PROD_CODE = '1546-QQ2'	15	
4	T2	PROD_QOH = 15 + 10		
5	T2	Gravação de PROD_QOH para PROD_CODE = '1546-QQ2'	25	
6	T1	Leitura de PROD_QOH para PROD_CODE = '1546-QQ2'	25	(Depois) 65
7	T1	Leitura de PROD_QOH para PROD_CODE = '1558-QW1'	23	(Antes) 88
8	T2	Leitura de PROD_QOH para PROD_CODE = '1558-QW1'	23	
9	T2	PROD_QOH = 23 – 10		
10	T2	Gravação de PROD_QOH para PROD_CODE = '1558-QW1'	13	
11	T2	***** <b>COMMIT</b> *****		
12	T1	Leitura de PROD_QOH para PROD_CODE = '2232-QTY'	8	96
13	T1	Leitura de PROD_QOH para PROD_CODE = '2232-QWE'	6	102

# Escalonador

- Pprocesso especial do SGBD
  - Estabelece a ordem em que são executadas operações de transações concorrentes
- Intercala a execução de operações de banco de dados, garantindo a serialização e o isolamento das transações
- **Escala serializável** é uma escala de operações transacionais na qual a execução intercalada das transações produz o mesmo resultado

# Controle de Concorrência com Métodos de Bloqueio

- **Bloqueio**
  - Garante a utilização exclusiva de um item de dados por uma transação atual
  - São necessários para evitar que outras transações leiam dados inconsistentes
- **Gerente de bloqueio**
  - Responsável por atribuir e inspecionar os bloqueios utilizados pelas transações

# Granularidade de Bloqueio

- Indica o nível de utilização de bloqueio
- Pode ocorrer nos seguintes níveis:
  - Banco de dados
  - Tabela
  - Página
  - Linha
  - Campo (atributo)

# Granularidade de Bloqueio (cont.)

- **Bloqueio no nível de banco de dados**
  - O banco inteiro é bloqueado
- **Bloqueio no nível de tabela**
  - A tabela inteira é bloqueada
- **Bloqueio no nível de página**
  - O SGBD bloqueia uma página de disco inteira

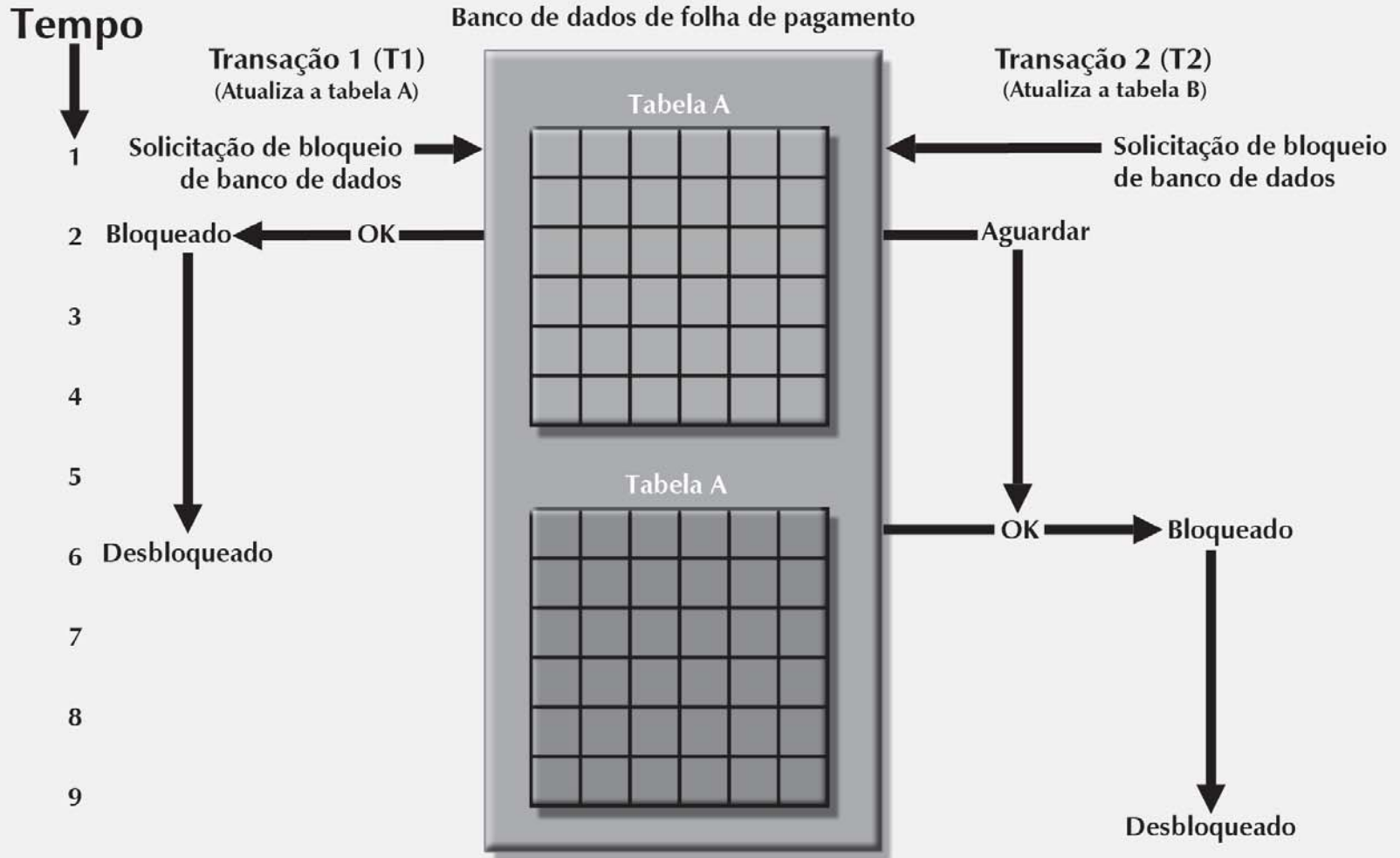


# Granularidade de Bloqueio (cont.)

- **Bloqueio no nível de linha**
  - Permite que transações concorrentes acessem linhas diferentes da mesma tabela
    - Mesmo quando estas estiverem localizadas na mesma página
- **Bloqueio no nível de campo**
  - Permite que transações concorrentes acessem a mesma linha
    - Contanto que solicitem a utilização de campos (atributos) diferentes nessa linha

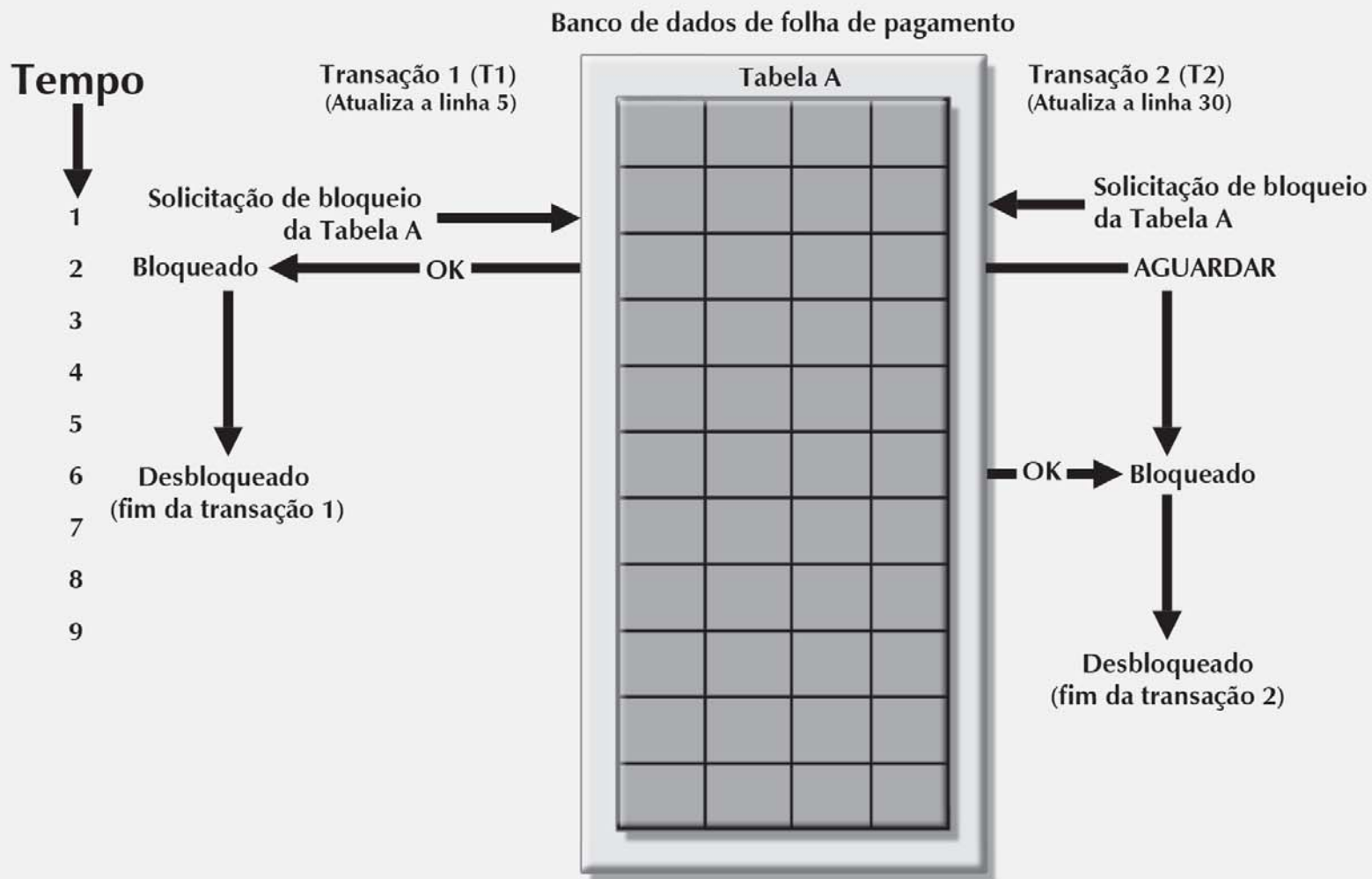
**FIGURA 10.3**

**Sequência de bloqueio no nível de banco de dados**



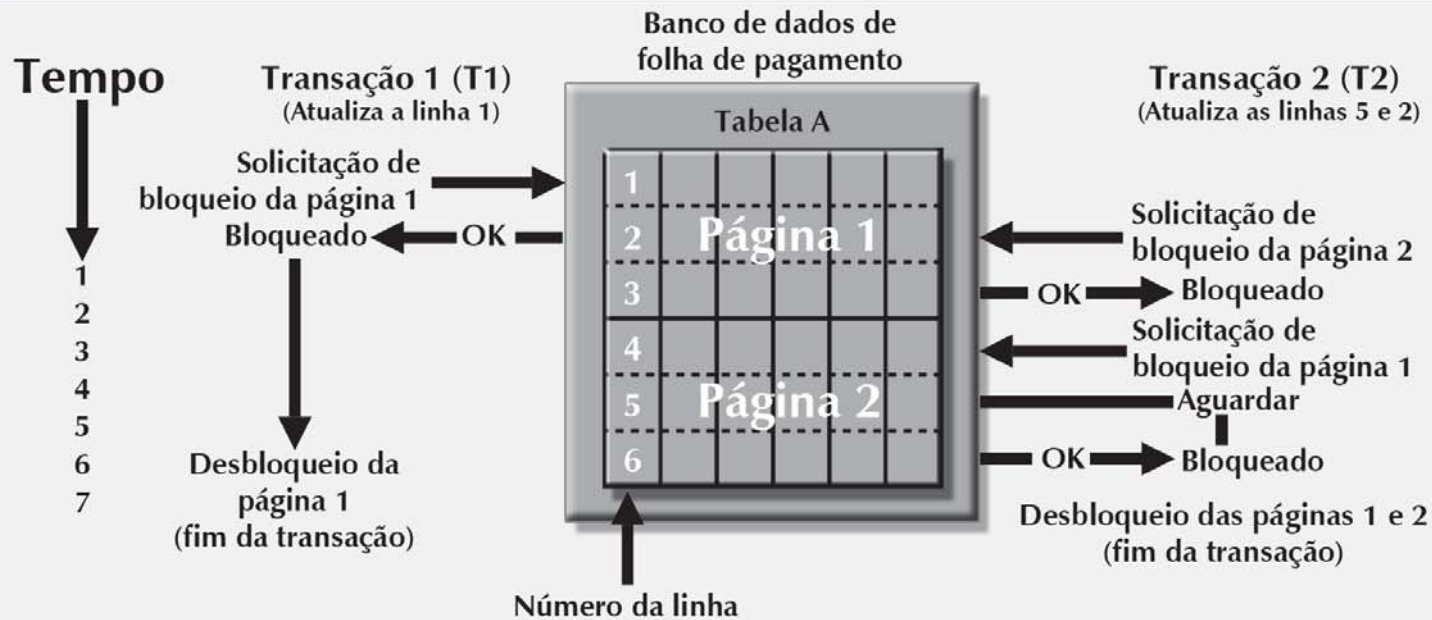
**FIGURA 10.4**

**Exemplo de bloqueio no nível de tabela**



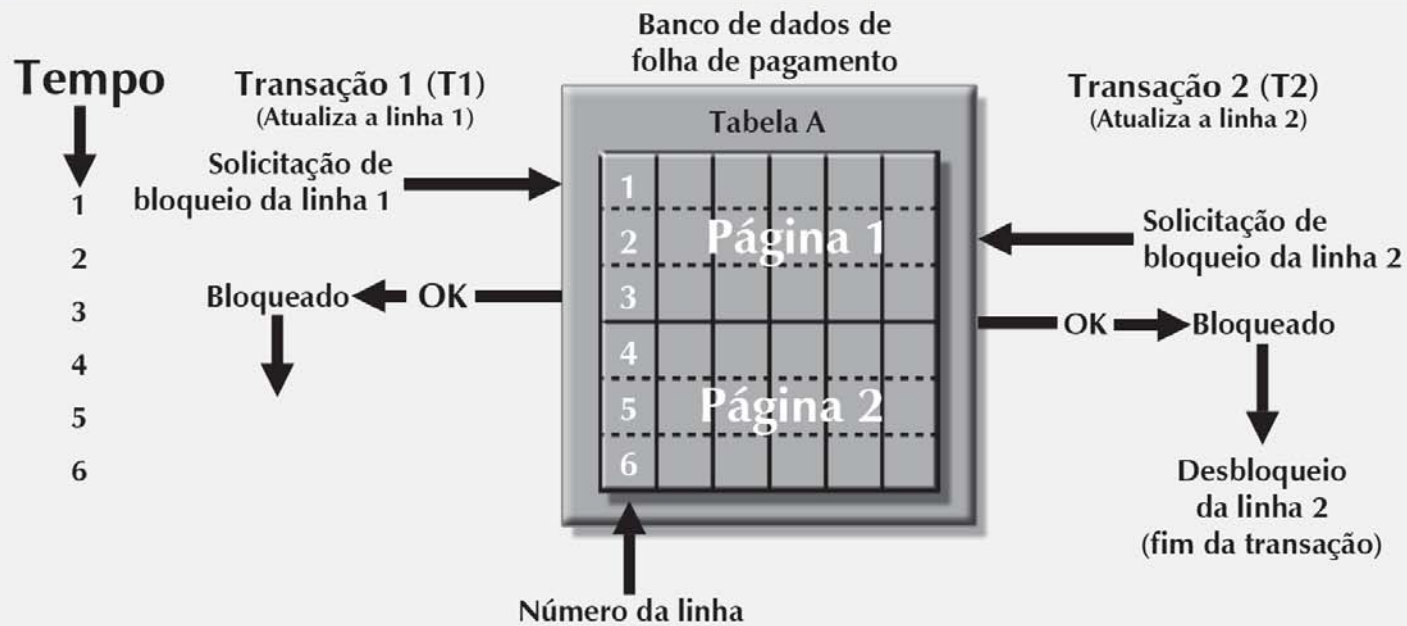
**FIGURA 10.5**

**Exemplo de bloqueio no nível de página**



**FIGURA 10.6**

**Exemplo de bloqueio no nível de linha**



# Tipos de Bloqueio

- **Bloqueios binários**
  - Dois estados: bloqueado (1) e desbloqueado (0)
- **Bloqueios exclusivos**
  - O acesso é reservado especificamente para a transação que bloqueou o objeto
  - Deve ser utilizado quando houver potencial para conflitos
- **Bloqueio compartilhado**
  - Ocorre quando transações concorrentes recebem acesso de leitura com base em um bloqueio comum

**TABELA 10.12 Exemplo de bloqueio binário**

TEMPO	TRANSAÇÃO	ETAPA	VALOR ARMAZENADO
1	T1	Bloqueio de PRODUCT	
2	T1	Leitura de PROD_QOH	15
3	T1	$PROD\_QOH = 15 + 10$	
4	T1	Gravação de PROD_QOH	25
5	T1	Desbloqueio de PRODUCT	
6	T2	Bloqueio de PRODUCT	
7	T2	Leitura de PROD_QOH	23
8	T2	$PROD\_QOH = 23 - 10$	
9	T2	Gravação de PROD_QOH	13
10	T2	Desbloqueio de PRODUCT	

# Bloqueio de Duas Fases para Assegurar a Serialização

- Define como as transações obtêm e liberam bloqueios
- Garante a serialização, mas não evita deadlocks
  - Fase de crescimento
    - Uma transação adquire todos os bloqueios solicitados sem desbloquear nenhum dado
  - Fase de redução
    - Uma transação libera todos os bloqueios e não pode obter novos

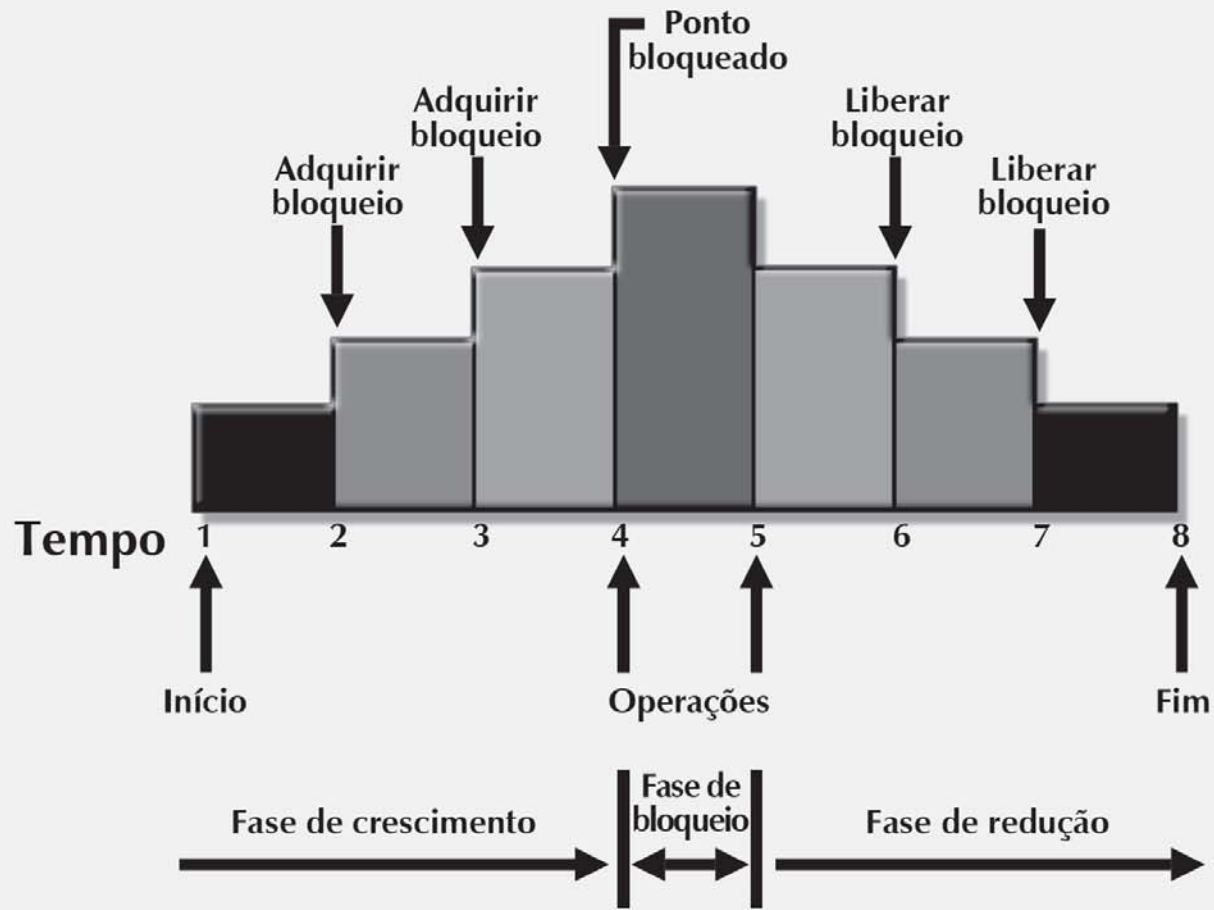


# Bloqueio de Duas Fases para Assegurar a Serialização (cont.)

- O protocolo de bloqueio de duas fases é determinado pelas seguintes regras:
  - Duas transações não podem apresentar bloqueios conflitantes
  - Nenhuma operação de desbloqueio pode preceder uma operação de bloqueio na mesma transação
  - Nenhum dado é afetado até que todos os bloqueios sejam obtidos

**FIGURA  
10.7**

**Protocolo de bloqueio de duas fases**



# Deadlocks

- Ocorre quando duas transações aguardam indefinidamente que a outra desbloqueie dados
- São possíveis apenas quando uma das transações obtiver um bloqueio exclusivo sobre o item de dados
  - Não pode haver situação de deadlock entre bloqueios compartilhados

# Deadlocks (cont.)

- As três técnicas básicas de controle são:
  - Prevenção
  - Detecção
  - Evasão
- A escolha do melhor método de controle de deadlocks depende do ambiente de banco de dados
  - Se a probabilidade de deadlocks for baixa, recomenda-se a detecção
  - Se a probabilidade for alta, a prevenção é mais adequada

**TABELA 10.13 Criação de uma condição de deadlock**

TEMPO	TRANSAÇÃO	RESPOSTA	STATUS DE BLOQUEIO	
0			Dado X	Dado Y
1	T1:LOCK(X)	OK	Desbloqueado	Desbloqueado
2	T2:LOCK(Y)	OK	Bloqueado	Desbloqueado
3	T1:LOCK(Y)	Aguardar	Bloqueado	Bloqueado
4	T2:LOCK(X)	Aguardar	Bloqueado	Bloqueado
5	T1:LOCK(Y)	Aguardar	Bloqueado	Bloqueado
6	T2:LOCK(X)	Aguardar	Bloqueado	Bloqueado
7	T1:LOCK(Y)	Aguardar	Bloqueado	Bloqueado
8	T2:LOCK(X)	Aguardar	Bloqueado	Bloqueado
9	T1:LOCK(Y)	Aguardar	Bloqueado	Bloqueado
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....



# Controle de Concorrência com Métodos de Time Stamping

- A abordagem de **time stamping** para escalonar transações correntes atribui um registro global e exclusivo a cada transação
- Esse valor produz uma ordem explícita em que as transações são enviadas ao SGBD
- A **exclusividade** garante que não existam valores iguais do registro
- A **monotonicidade** assegura que os valores sejam sempre maiores

# Esquemas Wait/die e Wound/wait

- **Wait/die**
  - A transação mais antiga aguarda que a mais nova seja concluída e libere seus bloqueios
- **Wound/wait**
  - A transação mais antiga desfaz a mais nova e a reescalona

# Controle de Concorrência com Métodos Otimistas

- **Abordagem otimista**
  - Baseia-se no pressuposto de que a maioria das operações de bancos de dados não entre em conflito
  - Não exige nem as técnicas de bloqueio nem as de time stamping
  - A transação é executada sem restrições até ser consolidada
  - Fases: leitura, validação e gravação



# Gerenciamento de Recuperação de Banco de Dados

- Restaura o banco de determinado estado (normalmente inconsistente) para o estado consistente anterior
- Baseiam-se na **propriedade de transações indivisíveis**
  - Todas as partes da transação devem ser tratadas como única unidade lógica de trabalho em que todas as operações são aplicadas e concluídas para produzir um banco consistente
- Se uma operação de transação não puder ser concluída
  - A transação deve ser abortada
  - Quaisquer mudanças no banco de dados devem ser desfeitas

# Recuperação de Transações

- O **protocolo de log gravação direta** garante que os logs de transações sejam sempre escritos *antes* de os dados do banco serem atualizados
- Os **logs de transações redundantes** garantem que a falha física de um disco não prejudique a capacidade do SGBD de recuperar dados
- Os **buffers** de bancos de dados são áreas de armazenamento temporário na memória principal
- Os **pontos de verificação** do banco de dados são operações em que o SGBD grava todos os buffers atualizados no disco

# Recuperação de Transações (cont.)

- **Técnica de gravação protelada**
  - Apenas o log de transações é atualizado
- Processo de recuperação:
  - Identificação do último ponto de verificação
  - Para uma transação iniciada e consolidada antes do último ponto de verificação, nada precisa ser feito
  - Para uma transação que executou operação de consolidação (COMMIT) após o último ponto de verificação, o SGBD utiliza os registros do log para refazer a transação e atualizar o banco de dados
  - Para qualquer transação que tenha uma operação ROLLBACK após o último ponto de verificação, nada precisa ser feito

# Recuperação de Transações (cont.)

- **Técnica de gravação indireta**
  - As operações de transação são atualizadas imediatamente no banco de dados durante a sua execução
- **Processo de recuperação:**
  - Identificação do último ponto de verificação
  - Para uma transação iniciada e consolidada antes do último ponto de verificação, nada precisa ser feito
  - Para uma transação consolidada após o último ponto de verificação, o SGBD refaz a transação utilizando os valores de “Depois” no log de transações
  - Para qualquer transação que tenha uma operação ROLLBACK após o último ponto de verificação ou que tenha sido deixada ativa, o SGBD utiliza os registros do log de transações para desfazer as operações, utilizando os valores de “Antes” nesse log

# Resumo

- A transação é uma sequência de operações que acessam o banco de dados
  - Deve ser uma unidade lógica de trabalho
    - Nenhuma parte dela pode existir por si mesma
  - Possuem cinco propriedades principais: indivisibilidade, consistência, isolamento, durabilidade e serialização
- COMMIT: salva alterações no disco
- ROLLBACK: restaura o estado anterior do banco de dados
- As transações de SQL são formadas por vários comandos e solicitações de bancos de dados

# Resumo (cont.)

- O log de transações rastreia todas as transações que modificam o banco
- O controle de concorrência coordena a execução simultânea de transações
- O escalonador é responsável por estabelecer a ordem em que operações de transações concorrentes são executadas
- O bloqueio garante o acesso exclusivo a um item de dados por uma transação
- Dois tipos de bloqueios em sistemas de bancos de dados: binários e compartilhados/exclusivos

# Resumo (cont.)

- A serialização de escalonamentos é garantida por meio da utilização de bloqueio de duas fases
- Deadlock: quando duas ou mais transações aguardam indefinidamente que a outra libere um bloqueio
- Três técnicas de controle de deadlocks: prevenção, detecção e evasão
- O controle de concorrência com time stamping atribui um registro exclusivo a cada transação
  - Escala a execução de transações conflitantes por ordem desse registro

## Resumo (cont.)

- O controle de concorrência com métodos otimistas assume que a maioria das transações de bancos de dados não entra em conflito
  - As transações são executadas de modo concorrente, utilizando cópias privadas temporárias dos dados
- A recuperação de bancos de dados restaura o banco de determinado estado para um estado consistente anterior