Transações em Banco de Dados

Definição
Início e Fim de uma Transação
Estados de Execução de uma Transação
Controle de Transações
Savepoint
Exemplo
Manipulação de Dados
Objetivos na Manipulação
Estado dos Dados
Consistência de Leitura
Exemplo

1 Definição

Transação é uma **unidade lógica**, que consiste de uma **série de operações** executadas, onde o **sistema de banco de dados** precisa **manter certas propriedades** para garantir o **sucesso** sem perda de informações lógicas.

1) Transação é uma unidade lógica de trabalho, envolvendo diversas operações de bancos dados (C. J. Date);

2) É uma unidade lógica de processamento no BD (Navathe);

3) É uma unidade de execução do programa que acessa e possivelmente atualiza vários itens de dados (Silberschatz).

Pode ser um programa inteiro, uma parte dele ou somente um comando. Deve-se especificar explicitamente o início da transação e o seu fim. Todas as operações entre estas cláusulas são consideradas parte de uma transação.

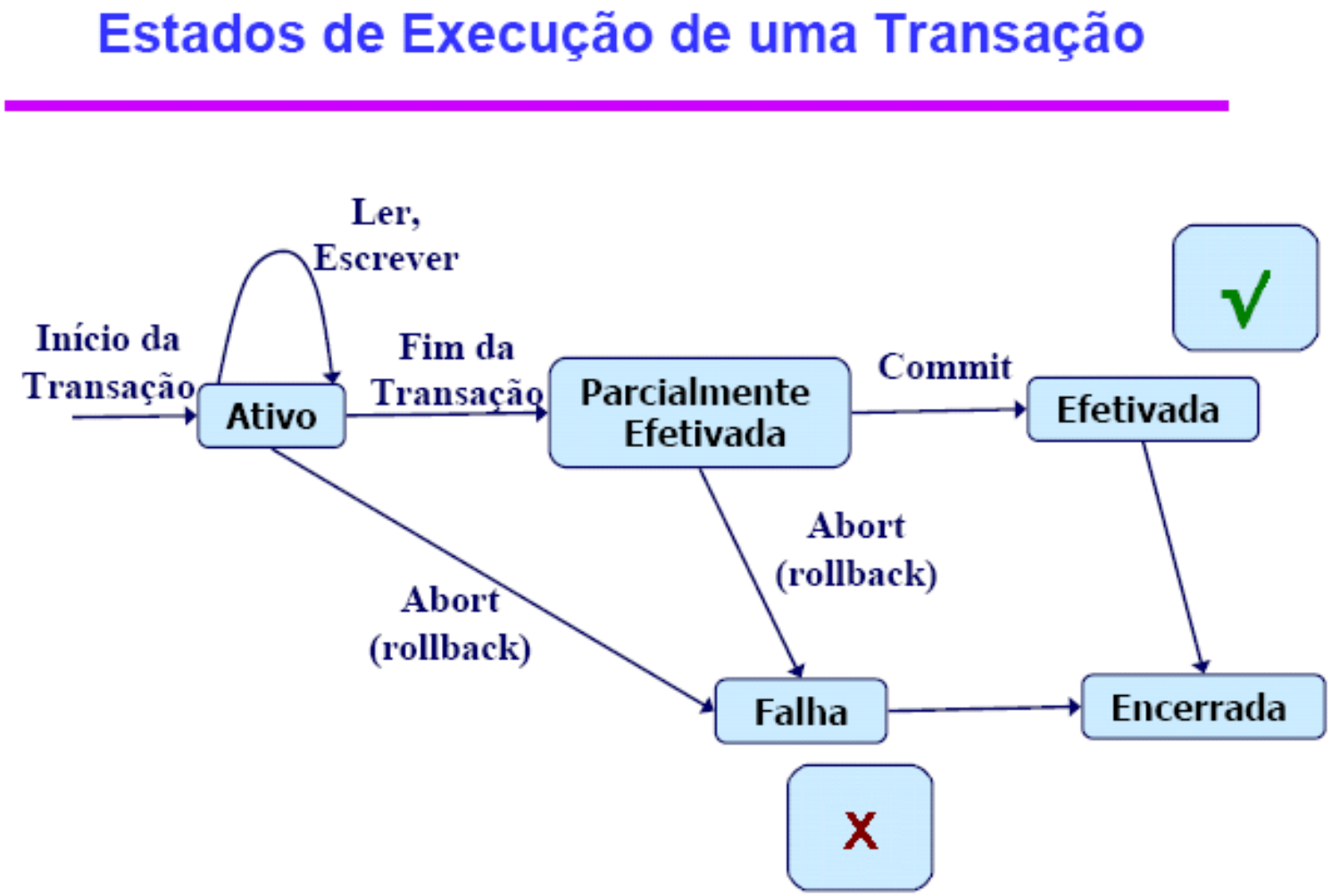
2 Início e Fim de uma Transação

Uma transação inicia-se, basicamente, com uma instrução BEGIN e termina com um COMMIT ou ROLLBACK.

COMMIT
Indica a conclusão de uma transação bem-sucedida (**parcialmente efetivada**), informando ao gerenciador de transações do SGBD que uma unidade lógica de trabalho foi concluída com sucesso. Com essa ação, o banco retoma o seu estado consistente e todas as atualizações feitas por essa unidade já podem se tornar permanentes (**efetivada**).

ROLLBACK
Indica a conclusão de uma transação mal sucedida (**em falha**). Ela informa ao gerenciador a ocorrência desta falha. Nesse momento o **banco** pode estar **inconsistente**, portanto todas as atualizações realizadas pela transação devem ser desfeitas (**abortada**).

3 Estados de Execução de uma Transação



4 Controle de Transações

Uma transação é uma série de instruções SQL que pode ser aplicada com sucesso, falhar, ou cancelar.

Os comandos para controlar transações são:

- COMMIT [WORK] para confirmar uma transação;
- ROLLBACK [WORK] [TO SAVEPOINT x] para cancelar parte ou toda uma transação;
- SAVEPOINT para criar um ponto de referência (“salvamento”) na transação;
- SET TRANSACTION para alterar o modo de consistência de leitura em uma transação.

5 Savepoint

Especifica um ou mais pontos de referência (savepoint ou salvamento) a serem criados. Estes nomes devem ser distintos dentro de uma mesma transação. É uma marca intermediária criada pelo usuário no contexto de uma transação.

Se um segundo savepoint é criado utilizando-se um mesmo identificador (já usado previamente), então o savepoint anterior é perdido (cancelado).

Depois que um savepoint é declarado explicitamente, as seguintes alternativas são possíveis:

1) Tornar permanente as mudanças em uma transação (COMMIT);
2) Desfazer as mudanças em uma transação, desde seu início (ROLLBACK);
3) Desfazer as mudanças realizadas até uma determinada “marca” (ROLLBACK TO SAVEPOINT).

Importante:

- ROLLBACK finaliza uma transação;
- ROLLBACK TO SAVEPOINT não finaliza uma transação.

6 Exemplo

```
CREATE TABLE Funcionario
(matricula      NUMBER(5),
 nome           VARCHAR2(30),
 depto          NUMBER(2),
 salario        NUMBER(10));

INSERT INTO funcionario VALUES (1, 'Daniel', 10, 100000);
INSERT INTO funcionario VALUES (2, 'Helena', 20, 100000);
INSERT INTO funcionario VALUES (3, 'Akito', 20, 50000);
INSERT INTO funcionario VALUES (4, 'Jaqueline', 20, 40000);
INSERT INTO funcionario VALUES (5, 'Ricardo', 20, 70000);
INSERT INTO funcionario VALUES (6, 'Joao', 20, 30000);
INSERT INTO funcionario VALUES (7, 'Clark', 20, 90000);
COMMIT;
```

6 Exemplo

```
SELECT nome, salario FROM funcionario WHERE nome in (‘Akito’,’Daniel’);

UPDATE funcionario
   SET salario = 7000
   WHERE nome = 'Daniel';

SAVEPOINT daniel_savepoint;

UPDATE funcionario
   SET salario = 12000
   WHERE nome = 'Akito';

SAVEPOINT akito_savepoint;

SELECT nome, salario FROM funcionario WHERE nome in (‘Akito’,’Daniel’);

ROLLBACK TO SAVEPOINT daniel_savepoint;

SELECT nome, salario FROM funcionario WHERE nome in (‘Akito’,’Daniel’);

UPDATE funcionario
   SET salario = 11000
   WHERE nome = 'Joao';

COMMIT;
```

7 Manipulação de Dados

DML (Data Manipulation Language) - Uma instrução DML é executada quando é feita:

- Adição de novas linhas em uma tabela;
- Modificação de linhas existentes em uma tabela;
- Remoção de linhas existentes em uma tabela.

Transações em Banco de Dados - Consiste de uma das seguintes instruções:

- Instrução DML que faz uma alteração;
- Instrução DDL;
- Instrução DCL (GRANT e REVOKE).

8 Objetivos na Manipulação

1. Garantir a consistência de dados;
2. Visualizar as alterações nos dados antes de torná-las permanentes;
3. Agrupar operações que se relacionam logicamente.

COMMIT Automático - Ocorre sob as seguintes circunstâncias:

- A instrução DDL é emitida;
- A instrução DCL é emitida;
- A saída normal do SQL*Plus, sem emitir explicitamente COMMIT ou ROLLBACK.

ROLLBACK Automático - Ocorre sob as seguintes circunstâncias:

- Finalização anormal do SQL*Plus;
- Queda do sistema.

9 Estado dos Dados

Antes do COMMIT/ROLLBACK:

1. O estado anterior dos dados pode ser recuperado;

2. O usuário atual pode revisar os resultados das operações DML usando a instrução SELECT;

3. Outros usuários não verão os resultados das instruções DML do usuário atual;

4. As linhas afetadas pelo comando são bloqueadas pelo usuário atual e outros usuários não podem modificá-las até que o usuário atual faça um COMMIT ou ROLLBACK.

9 Estado dos Dados

Após o COMMIT:

1. As alterações nos dados são feitas permanentemente no banco de dados;
2. O estado anterior dos dados é perdido permanentemente;
3. Todos os usuários passam a ver os resultados;
4. As linhas afetadas são desbloqueadas, estando disponíveis para manipulação por outros usuários;
5. Todos os savepoints são apagados.

10 Consistência de Leitura

A consistência na leitura garante sempre uma exibição consistente dos dados;

As alterações feitas por um usuário não entram em conflito com as alterações feitas por outro usuário;

A consistência na leitura garante que nos mesmos dados:

- Os leitores não esperem pelos autores;
- Os autores não esperem pelos leitores.

10 Consistência de Leitura

Durante todo o processamento de um comando SQL, o ORACLE mantém a consistência dos dados a partir do momento em que o comando **foi iniciado**.

Para o comando SELECT, o ORACLE marca o início da sua execução como o instante (timestamp) a partir do qual a consistência de leitura deve ser mantida.

A partir deste momento, quaisquer alterações feitas em uma tabela por outros usuários não serão vistas por quem emite um comando SELECT. Isto só ocorrerá quando os outros usuários terminarem suas transações, com os comandos COMMIT ou ROLLBACK.

Todas as alterações de dados são mantidas em um local no ORACLE chamado **Segmento de Rollback**. Na execução de um SELECT em uma tabela, este será o local onde o ORACLE lerá os valores antigos, e não nos blocos de dados originais (alterados).

11 Exemplo

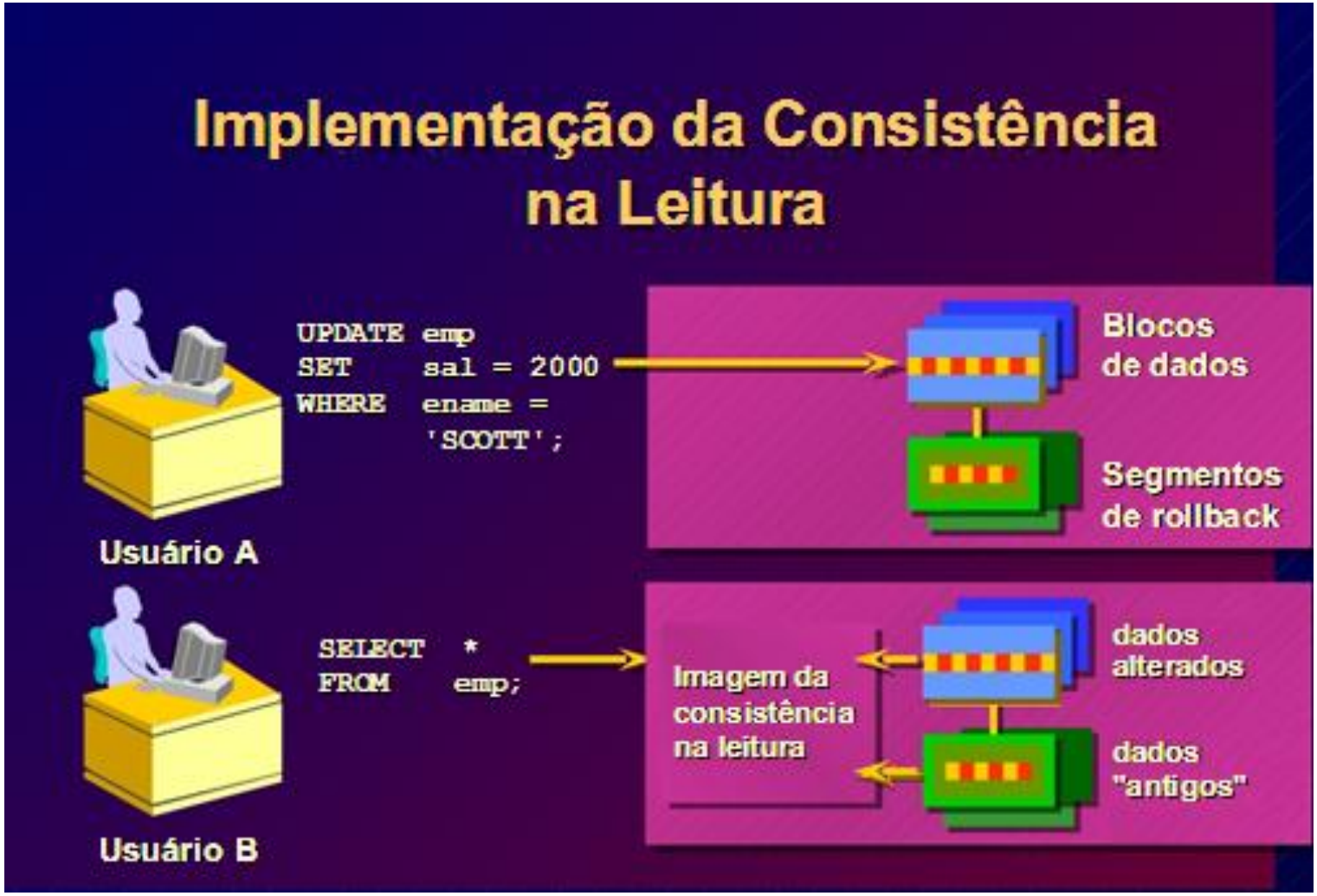
Às dez horas o usuário A executa o comando UPDATE, mas não efetiva a alterações.
10 h 00 min SQL> UPDATE EMP ...;

Às dez horas e um minuto o usuário B pesquisa a tabela EMP. Ele não enxerga as alterações feitas pelo usuário A. A partir do segmento de rollback (que registrou a alteração do usuário A) será trazido o valor antigo dos dados, ocorrendo a **consistência da leitura**.
10 h 01 min SQL> SELECT ... FROM emp;

Às dez horas e dois minutos o usuário A efetiva sua transação. Neste momento os segmentos de rollback que estavam alocados são liberados para uso em novas transações (sem apagar seu conteúdo).
10 h 02 min SQL> COMMIT;

Finalmente, às dez horas e três minutos o usuário B passa a enxergar as alterações feitas na tabela EMP (pelo comando UPDATE) do usuário A, pois a transação foi terminada e efetivada com o comando COMMIT.
10 h 03 min SQL> SELECT ... FROM emp;

11 Exemplo


Implementação da Consistência na Leitura
UPDATE emp
SET sal = 2000
WHERE ename = 'SCOTT';
Usuário A
SELECT *
FROM emp;
Usuário B
Blocos de dados
Segmentos de rollback
Imagem da consistência na leitura
dados alterados
dados "antigos"