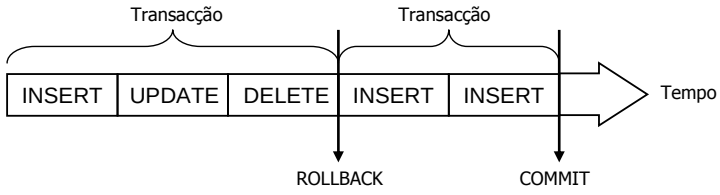
Gestão de Transacções

- Conceitos e Propriedades
- Controlo de Concorrência
- Recuperação
- Transacções no SGBD *Oracle*

Conceitos

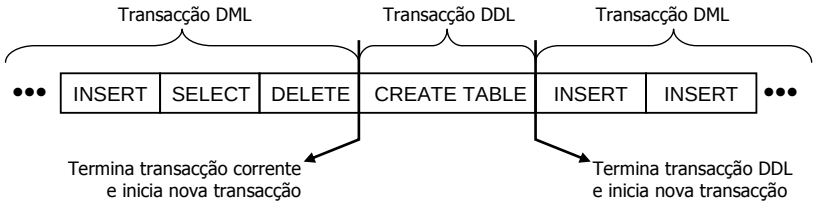
- Transacção
  - Uma acção ou um conjunto de acções, realizadas por um único utilizador ou programa de aplicação que acedem ou alteram os conteúdos da BD
  - Unidade lógica de trabalho numa BD
  - Só tem um único resultado
    - COMMIT
    - ROLLBACK



- Propriedades de uma Transacção
  - Atomicidade
  - Consistência
  - Independência
  - Durabilidade

Transacções em Oracle

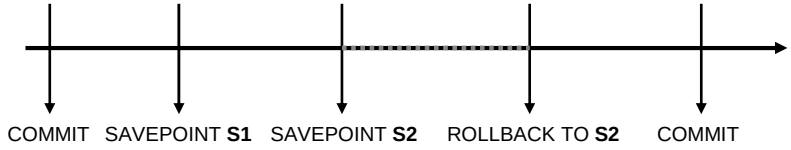
- Transacção DDL
  - Contém 1 único comando DDL
- Transacção DML
  - Contém 1 ou mais comandos DML



- Processamento de Transacções DML
  - Início
    - Primeiro comando DML
  - Fim
    - COMMIT
    - ROLLBACK
    - Comando DDL
    - Detecção de qualquer ERRO
    - Saída da sessão SQL

Transacções em Oracle

- *Savepoint*
  - Permite guardar todas as alterações efectuadas desde o início da transacção corrente, ou desde o último *Savepoint*
  - Não termina uma transacção
  - Úteis em transacções muito longas
  - Permite anular apenas as alterações introduzidas após se ter realizado um *savepoint*
  - Quando a transacção termina todos os *savepoints* são eliminados



Transacções em Oracle

- Comandos SQL
  - COMMIT ;
  - SAVEPOINT <nome_savepoint> ;
  - ROLLBACK [TO <nome_savepoint>] ;

- Exemplo

⋮

```
INSERT INTO TCliente (BI, Nome, Morada, NTelemovel)
VALUES (112212121, 'José Pereira', 'Leiria', 911234567);

SAVEPOINT fim_inserção;

UPDATE TCliente     SET nome = 'José Pereira Silva';

ROLLBACK  TO  fim_inserção;

UPDATE TCliente     SET nome = 'José Pereira Silva'
WHERE BI = 112212121;

COMMIT;
```

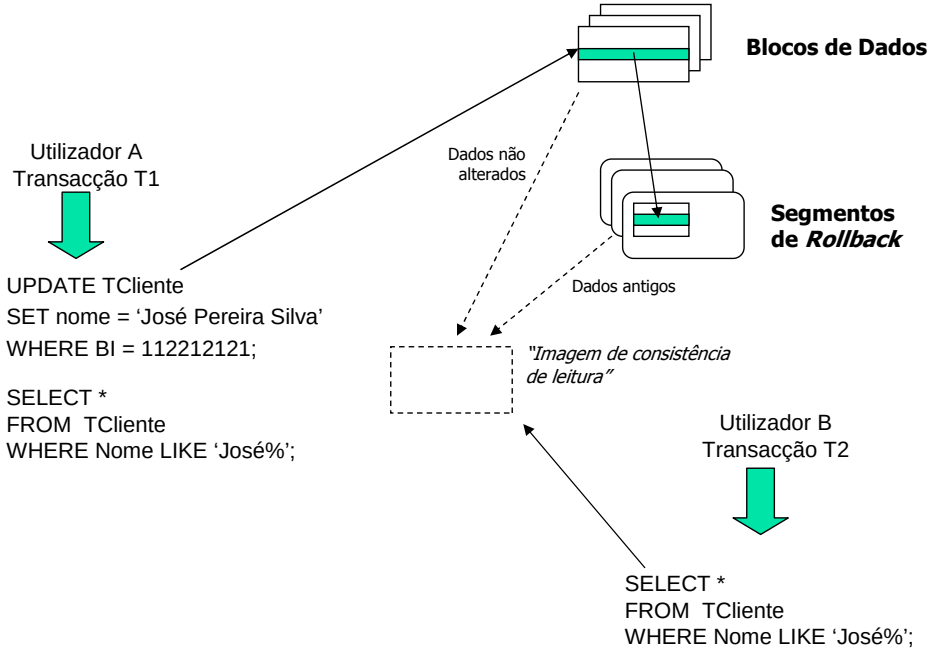

Transacções em Oracle

- Segmentos de *Rollback*
  - Componentes da estrutura interna da BD
  - Manipulados pelo SGBD
  - Contêm os valores antigos relativos a todas as alterações efectuadas aos dados durante uma transacção
  - Efectuam o cancelamento das transacções
  - Asseguram a consistência de leitura

- Consistência de leitura
  - Garantir que cada utilizador vê os dados conforme eles existiam no momento da última confirmação

Transacções em Oracle

- Utilização de Segmentos de *Rollback*



Transacções em Oracle

- Segmentos de *Rollback*
  - Sempre que começa uma transacção é atribuído um segmento de *rollback*, e podem ser atribuídos mais se necessário
  - Cada segmento de *rollback* pode tratar mais do que uma transacção
  - Contêm os valores antigos relativos a todas as alterações efectuadas aos dados durante uma transacção
  - COMMIT
    - A informação do segmento de *rollback* é libertada
    - Caso existam consultas (de outros utilizadores) sobre os dados, iniciadas antes do COMMIT, a informação do segmento de *rollback* só é libertada quando já não for necessária
  - ROLLBACK
    - A informação do segmento de *rollback* é reposta nos "blocos de dados" e depois libertada

Controlo de Concorrência

- Objectivos
  - Garantir a consistência dos dados
  - Maximizar a concorrência
- Problemas inerentes à concorrência
  - Leituras inconsistentes
  - Leituras não reprodutíveis
  - Leituras erróneas
  - Perda de actualizações
- Técnicas
  - Bloqueio (*Locking*)
  - Bloqueio de 2 fases (*Two-phase locking*)

Controlo de Concorrência

- Técnica de Bloqueio
  - Mecanismo utilizado para controlar o acesso concorrente aos dados
  - Impede a actualização simultânea dos mesmos dados por 2 ou mais utilizadores
  - Impede alterações à estrutura de uma tabela enquanto decorrerem transacções que actualizem dados dessa tabela
  - Utilizada sempre que um utilizador tenta introduzir alterações na BD (acessos de escrita)
  - Funcionamento:
    - Durante uma transacção, todas as linhas (tabelas) alteradas são automaticamente bloqueadas
    - Se numa transacção há a tentativa de alterar linhas previamente bloqueadas, esta transacção fica suspensa em lista de espera
    - Todas as linhas bloqueadas numa transacção são automaticamente desbloqueadas quando a transacção termina

Controlo de Concorrência

- Técnica de Bloqueio
  - Exemplo

Transacção T1

```
UPDATE TEmp
  SET nomeE = 'José'
  WHERE NEmp = 1002;

UPDATE TEmp
  SET Sal  = Sal * 1.05
  WHERE Dept = 10;
```

-
-
-

NEmp	NomeE	Dept	Sal
1002	Josssé	10	1000
1500	Maria	10	1200
1550	Luísa	20	600
1800	Luís	30	800

Transacção T2

```
SELECT * FROM TEmp;

UPDATE TEmp
  SET nomeE = 'Marie'
  WHERE NEmp = 1500;
```

Wait

Controlo de Concorrência

- Tipos de Bloqueio
  - Bloqueio do Dicionário de Dados
  - Bloqueio de Manipulação de Dados
- Níveis de Bloqueio
  - Bloqueio de tabela
  - Bloqueio de Linha
- Tipos de Bloqueio implícito (SGBD Oracle)
  - Exclusivo (X)
  - Exclusivo de registos (RX)

COMANDO	Bloqueio de Linha	Bloqueio de Tabela
SELECT	--	--
INSERT	X	RX
UPDATE	X	RX
DELETE	X	RX
DDL	--	X

Controlo de Concorrência

- Bloqueio Explícito (SGBD Oracle)
  - SELECT ... FOR UPDATE [OF ...] [NOWAIT]
    - Efectua uma consulta para identificar as linhas a actualizar e, de seguida, bloquei-as
    - NOWAIT previne impasses
    - Bloqueios são libertados com COMMIT ou ROLLBACK
  - LOCK TABLE <nome_tabela> IN <tipo_bloqueio>

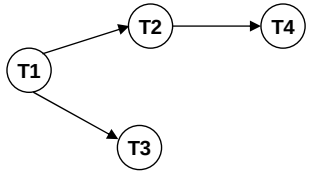
Controlo de Concorrência

- Impasse (*Deadlock*)
  - Ocorre quando 2 (ou mais) transacções tentam aceder aos mesmos dados e ficam à espera que sejam desbloqueados
  - Exemplo

Transacção T1	**Transacção T2**
UPDATE TEmp SET nomeE = 'José' WHERE NEmp = 1002;	UPDATE TEmp SET sal = sal * 1.10 WHERE NEmp = 1500;
UPDATE TEmp SET NomeE = 'Marie' WHERE NEmp = 1500;	UPDATE TEmp SET Sal = sal * 1.05 WHERE NEmp = 1002;
Wait *linha bloqueada pela T2*	***Wait*** *linha bloqueada pela T1*

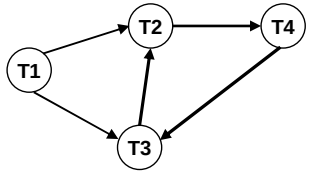
Controlo de Concorrência

- Impasse (*Deadlock*)
  - Prevenir
  - Detectar e recuperar
- Prevenir Impasses
  - No início de uma transacção, bloquear explicitamente todos os recursos a actualizar
  - Impor uma ordem para o bloqueio dos recursos e garantir que as transacções seguem essa ordem
  - *Timestamping*
- Detectar Impasses e Recuperar
  - Grafo de Esperas (*wait-for graph*)



Controlo de Concorrência

- Detectar Impasses e Recuperar
  - Grafo de Esperas (*wait-for graph*)
    - Exemplo de um impasse



    - Resolver um impasse
      - Fazer ROLLBACK a uma (ou mais) transacções
      - Critérios de escolha (minimizar os custos):
        - Transacção com *savepoints*
        - Transacção com menos alterações efectuadas até ao momento
        - Transacção que tenha mais dados ainda a manipular