

# Introdução

- A recuperação de um sistema de gestão de bases de dados, significa restaurar a base de dados a um estado consistente, depois de alguma falha o levar a um estado inconsistente, ou pelo menos suspeito.
- ▶ Um estado consistente significa que satisfaz todas as restrições de integridade conhecidas.

**3** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Introdução

- ▶ Os exemplos dados baseiam-se na base de dados habitual:
  - F < F#, FNOME, STATUS, CIDADE>
  - P < P#, PNOME, COR, PESO, CIDADE>
  - FP <F#, P#, QTD>
  - F# é FK de F; P# é FK de P
  - as PKs encontram-se sublinhadas.

2018/2019 - Bases de Dados - Fernanda Brito Correia

• 4

## Introdução

- Os princípios em que se baseia a recuperação são a:
  - Redundância
- A redundância a que nos referimos é ao nível físico e não ao nível lógico (normalização).
- A maneira de garantir que a base de dados é recuperável é garantir que toda a informação que ela contém possa ser reconstruída a partir de alguma outra informação armazenada de modo redundante num outro lugar do sistema.

**5** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Introdução

- ▶ Supomos que estamos num ambiente de base de dados de grande porte (concorrente e multi-utilizador).
- Nos sistemas de pequeno porte (de um só utilizador), normalmente a recuperação é da responsabilidade do utilizador (deve criar cópias periódicas do trabalho backups e refazer o trabalho manualmente).

2018/2019 - Bases de Dados - Fernanda Brito Correia

Transação é uma unidade lógica de trabalho.

2018/2019 - Bases de Dados - Fernanda Brito Correia

- Suponha que a tabela das peças P, inclui o atributo adicional QTDTOTAL, representando a quantidade total de um fornecimento da peça em questão.
- Ou seja, o valor de QTDTOTAL para qualquer peça deve ser igual à soma dos valores de QTD de todos os fornecimentos dessa peça (isso é uma restrição da base de dados).

> 7

## Transacções

- Vamos mostrar o pseudocódigo de uma transacção que consiste em adicionar um novo fornecimento do fornecedor F5 da peça P1 na quantidade 1000, à base de dados.
- Esta operação consiste num:
  - INSERT para adicionar o novo fornecimento à tabela FP;
  - ▶ UPDATE para actualizar o valor de QTDTOTAL da peça PI.

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Transacções Pseudo-código: BEGIN TRANSACTION: INSERT INTO FP 2018/2019 - Bases de Dados - Fernanda Brito Correia RELATION {TUPLO { $F\#=F\#(`F5'), P\#=P\#(`P1'), QTD=QTD(1000)}$ }; SE ocorreu algum erro THEN GO TO UNDO; END IF; UPDATE P WHERE P#=P#('P1') QTDTOTAL := QTDTOTAL+QTD(1000); SE ocorreu algum erro THEN GO TO UNDO; END IF; COMMIT; GO TO FINISH; UNDO: ROLLBACK; FINISH: RETURN; **9**

## Transacções

- ▶ Em vez de uma operação atómica, temos duas atualizações da base de dados, uma operação INSERT e uma operação UPDATE.
- A base de dados não é consistente entre essas duas atualizações.
- Viola temporariamente a restrição de que o valor de QTDTOTAL para a peça PI dever ser igual à soma de todos os valores de QTD para a peça PI.

2018/2019 - Bases de Dados - Fernanda Brito Correia

Uma transacção em geral é uma sequência de várias operações, a qual transforma um estado consistente da base de dados noutro estado consistente.

Não podemos deixar que ocorra uma das operações e as outras não, pois deixaria a base de dados inconsistente. Temos de ter a garantia de que todas as actualizações sejam executadas.

2018/2019 - Bases de Dados - Fernanda Brito Correia

▶ 11

### Transacções

- O que pode ocorrer para que as actualizações não sejam todas executadas:
  - Queda do sistema entre as operações INSERT e UPDATE;
  - Overflow aritmético na operação UPDATE;

2018/2019 - Bases de Dados - Fernanda Brito Correia

▶ Um sistema que admita a **gestão de transações**, garante que se a transação executar atualizações e ocorrer uma falha (por qualquer motivo) antes de a transação ter sido concluída como planeado, então **essas atualizações serão desfeitas**.

▶ Uma transação **ou** é executada na sua totalidade **ou** será totalmente cancelada (como se nunca tivesse sido executada).

2018/2019 - Bases de Dados - Fernanda Brito Correia

**1**3

## Transacções

- ▶ Com as transações, uma sequência de operações que não é atómica, pode parecer atómica do ponto de vista externo.
- O componente do sistema que fornece essa atomicidade é o gestor de transações, também conhecido por monitor de processamento de transações ou monitor TP e utiliza as operações COMMIT e ROLLBACK.

2018/2019 - Bases de Dados - Fernanda Brito Correia

#### **▶** COMMIT

indica o fim de uma transação bem sucedida.

- Informa o gestor de transações que foi concluída uma unidade lógica de trabalho com sucesso.
- A base de dados está (ou deveria estar) novamente num estado consistente e todas as atualizações podem ser tornadas permanentes.

#### ▶ ROLLBACK

- indica o fim de uma transação mal sucedida
- Informa o gestor de transações que ocorreu algum erro e que a base de dados pode estar num estado inconsistente e que todas as atualizações feitas pela transação devem ser desfeitas.

**1**5

2018/2019 - Bases de Dados - Fernanda Brito Correia

### Transacções

### ▶ Como é possível desfazer uma atualização?

- O sistema mantém um log ou diário em disco, no qual são registados os detalhes de todas as operações de atualização, em particular imagens do objeto atualizado antes e depois das operações
- Se for necessário desfazer alguma atualização, o sistema poderá usar a entrada de log correspondente para restaurar o objeto atualizado com o seu valor anterior.

2018/2019 - Bases de Dados - Fernanda Brito Correia

- Um sistema deve garantir que instruções individuais sejam atómicas.
- Como as instruções, num sistema relacional, são ao nível de conjuntos e operam sobre muitos tuplos ao mesmo tempo, não deve ser possível que uma determinada instrução falhe durante o processo e deixe a base de dados inconsistente (por exemplo com alguns tuplos actualizados e outros não).
- Se ocorrer um erro no meio de uma dada instrução a base de dados deve permanecer totalmente inalterada.

**17** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação de transacções

- ▶ Uma transação começa com a execução bem sucedida de BEGIN TRANSACTION e termina com a execução bem sucedida de uma instrução COMMIT ou de uma instrução ROLLBACK.
- ▶ Um COMMIT estabelece um **ponto de COMMIT** ou **validação** ou **ponto de sincronização**.
- ▶ O COMMIT e o ROLLBACK terminam a transação e não o programa. Em geral um único programa consiste numa sequência de várias transações executadas uma após a outra.

# Recuperação de transacções

Quando um ponto de COMMIT é estabelecido:

- ▶ Todas as actualizações feitas pelo programa em execução desde o ponto de COMMIT anterior são validadas, isto é tornam-se permanentes.
- Antes do ponto de COMMIT todas as actualizações devem ser consideradas apenas tentativas (podem ser desfeitas).
- Uma vez validada uma actualização tem a garantia de nunca será desfeita.

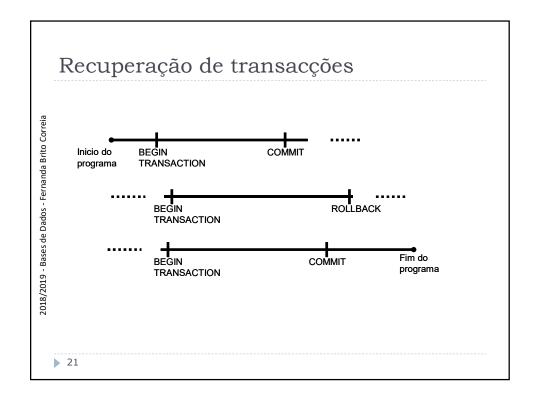
2018/2019 - Bases de Dados - Fernanda Brito Correia

**19** 

## Recuperação de transacções

- Quando um ponto de COMMIT ou de ROLLBACK é estabelecido:
  - Todos os posicionamentos da base de dados (por exemplo cursores) serão perdidos e todos os bloqueios de tuplos serão libertados.
  - Alguns sistemas oferecem uma opção pela qual o programa pode ser capaz de conservar a capacidade de endereçamento para determinados tuplos ( e reter certos bloqueios de tuplos) de uma transação para a seguinte.

2018/2019 - Bases de Dados - Fernanda Brito Correia



## Recuperação de transacções

- No exemplo em pseudo-código foram incluídos testes explícitos de erros e foi emitida uma instrução explícita de ROLLBACK, no caso de ser detectado qualquer erro.
- Mas, o sistema não pode supor que um programa inclui testes explícitos para todos os erros possíveis.
- Por isso, o sistema emitirá uma instrução de ROLLBACK implícita para todo o programa que falhar por qualquer motivo sem atingir o seu fim planeado (uma instrução COMMIT ou ROLLBACK explícita).

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação de transacções

- Pode acontecer que o sistema caia depois de ter sido aceite a instrução COMMIT, mas antes de terem sido gravadas fisicamente na base de dados.
  - Podem estar num buffer da memória principal e serem perdidas no instante da queda do sistema.
- Mesmo que isso aconteça, o procedimento de reiniciar o sistema ainda instalará essas atualizações na base de dados.
- Descobre os valores que devem ser gravados através do exame das entradas no log.
- O log deve ser gravado fisicamente antes de se completar o processamento de COMMIT.
- Regra de gravação antes do registo no log

Write-ahead logging

23

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação de transacções

- As transacções são uma:
  - Unidade de trabalho
  - Unidade de recuperação
  - Unidade de concorrência
  - Unidade de integridade (transformam um estado consistente da base de dados noutro estado consistente)

2018/2019 - Bases de Dados - Fernanda Brito Correia

# As Propriedades ACID

- ▶ Atomicidade
- Consistência
- ▶ Isolamento
- Durabilidade

2018/2019 - Bases de Dados - Fernanda Brito Correia

**>** 25

# As Propriedades ACID

### Atomicidade

As transações são atómicas, tudo feito ou nada feito.

### ▶ Consistência

- ▶ Transações preservam a consistência da base de dados.
- ▶ Transformam um estado consistente noutro estado consistente, sem necessidade de preservar a consistência em todos os pontos intermédios.

2018/2019 - Bases de Dados - Fernanda Brito Correia

**≥** 26

## As Propriedades ACID

#### Isolamento

▶ Transações são isoladas umas das outras.

▶ Há muitas transações a serem executadas de um modo concorrente, mas as atualizações de qualquer transação são ocultas das outras até o commit dessa transação.

### Durabilidade

Uma vez terminada a operação commit de uma transação, as suas atualizações sobrevivem na base de dados mesmo que haja uma queda posterior do sistema.

**27** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação do Sistema

O sistema deve estar preparado para recuperar de:

#### ▶ Falhas locais

- ex: ocorrência de uma condição de overflow dentro de uma transação individual
- Só afetam a transação em que a falha ocorreu.

### ▶ Falhas globais

- ex: queda de energia
- Afeta todas as transações a decorrer no momento em que a falha ocorreu, e portanto tem implicações em todo o sistema.

As falhas globais enquadram-se em duas grandes categorias:

- ▶ Falhas do sistema (ex: queda de energia). Afectam todas as transacções em curso no momento, mas não danificam fisicamente a base de dados. Chamada soft crash.
- ▶ Falhas de hardware (ex: disco estragado). Causam danos na base de dados ou pelo menos em parte e afectam pelo menos todas as transacções que no momento estão a usar essa parte. Chamada hard crash.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 29

## Recuperação do Sistema

### ▶ Falhas do sistema:

- O conteúdo da memória principal perde-se, em particular a informação da base de dados que está nos buffers.
- O estado exato de qualquer transação em curso no momento da falha deixa de ser conhecido.
- A transação não pode ser concluída com sucesso e deverá ser desfeita.
- Pode ser necessário refazer, no momento de reiniciar, certas transações concluídas com sucesso antes da queda do sistema, mas que não conseguiram ter as suas atualizações transferidas dos buffers da base de dados para a base de dados física.

2018/2019 - Bases de Dados - Fernanda Brito Correia

▶ Como é que o sistema sabe no momento de reiniciar quais as transações que devem ser desfeitas e quais as que devem ser refeitas?

▶ Resposta:

▶ Em certos intervalos predeterminados, em geral quando um número preestabelecido de entradas é gravado no log, o sistema automaticamente **marca um checkpoint**.

**31** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação do Sistema

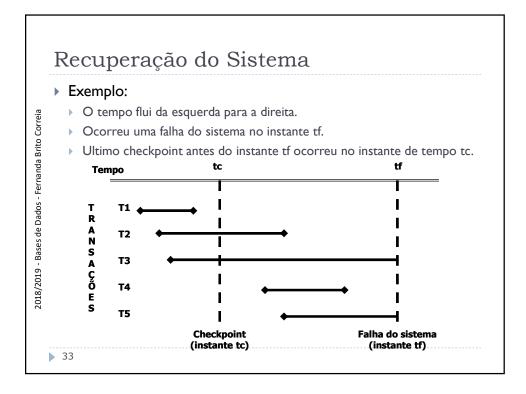
▶ O que é marcar um checkpoint?

### ▶ Envolve:

- Gravar fisicamente (gravação forçada) o conteúdo dos buffers da base de dados na base de dados física.
- Gravar fisicamente um registo de checkpoint especial no log físico.
- O registo de checkpoint fornece uma lista de todas as transações que estavam a decorrer no momento em que o checkpoint foi marcado.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 32



- As transações do tipo TI foram concluídas com sucesso antes do instante tc.
- As transações do tipo T2 foram iniciadas antes do instante to e concluídas com sucesso após o instante to e antes do instante tf.
- As transações do tipo T3 também foram iniciadas antes do instante tc, mas não foram concluídas antes do instante tf.
- As transações do tipo T4 começaram após o instante to e foram concluídas com sucesso antes do instante tf.
- As transações do tipo T5 também foram iniciadas após o instante tc mas não foram concluídas até ao instante tf.

**34** 

### O que acontece quando o sistema é reiniciado?

- As transações do tipo T3 e T5 devem ser desfeitas.
- As transações do tipo T2 e T4 devem ser refeitas.
- As transações do tipo TI não entram no processo de reinicialização, porque as suas atualizações foram forçadas na base de dados no instante tc, como parte do processo de checkpoint.
- As transações concluídas sem sucesso (isto é com um rollback) antes do instante tf também não entram no processo de reiniciar.

> 35

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação do Sistema

No momento da reinicialização o sistema executa o procedimento para identificar as transações do tipo T2 a T5.

- Utiliza 2 listas de transações, a lista UNDO e a lista REDO.
- ▶ Define a lista UNDO como igual à lista de todas as transações dadas no registo do checkpoint mais recente;
- Define a lista REDO como vazia.
- Pesquisa o log a partir do registo do checkpoint.
- Se for encontrada uma entrada BEGIN TRANSACTION para a transação T, acrescentar T à lista de UNDO.
- Se for encontrada uma entrada de log COMMIT para a transação T, mover T da lista de UNDO para a lista REDO.
- Quando for alcançado o final do log, as listas UNDO e REDO identificarão respetivamente as transações dos tipos T3 e T5 e transações dos tipos T2 e T4.

> 36

- Após essa identificação, o sistema percorre o log do fim para o inicio, desfazendo as transacções da lista UNDO (também chamada recuperação inversa).
- Em seguida percorre o log de novo para a frente, refazendo as transacções da lista REDO (também chamada recuperação directa).
- ▶ Só quando toda a actividade de recuperação estiver concluída é que o sistema está pronto para aceitar um novo trabalho.

**37** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recuperação do Sistema

- A descrição do procedimento de recuperação do sistema mostra primeiro a execução de operações de desfazer e depois a execução das operações de refazer, que era a maneira como funcionavam os sistemas antigos.
- ▶ Por razões de eficiência, em geral, os sistemas modernos executam primeiro as operações de refazer e depois as operações de desfazer.

2018/2019 - Bases de Dados - Fernanda Brito Correia

> 38

# Recuperação do Hardware

- ▶ Este assunto é um pouco diferente dos assuntos de recuperação de transações e do sistema.
- Exemplos deste tipos de falhas são:
  - Queda da cabeça de disco
  - Falha no controlador do disco
- A recuperação de uma falha deste tipo envolve basicamente a restauração da base de dados a partir de uma cópia de backup (ou dump).
- Discos RAID.

> 39

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recursos de SQL

- ▶ O suporte de SQL para transacções admite as instruções usuais COMMIT e ROLLBACK, com uma palavra chave opcional WORK.
- ▶ Estas instruções forçam uma operação CLOSE de todo o cursor aberto, provocando a perda de todo o posicionamento da base de dados.
- Algumas implementações de SQL fornecem uma opção para impedir essa operação CLOSE automática e a perda de posicionamento em COMMIT (não em ROLLBACK).

**4**0

# Recursos de SQL

- ▶ O DB2 admite a opção WITH HOLD numa declaração de cursor.
- → O COMMIT não fecha esse cursor, mas deixa-o aberto e posicionado de tal forma que a próxima instrução FETCH desloca-o para a próxima linha em sequência.
- lsto evita o código de reposicionamento (provavelmente complexo) necessário na próxima instrução OPEN.
- ▶ Este recurso atualmente está incluído no SQL3.

2018/2019 - Bases de Dados - Fernanda Brito Correia

**41** 

## Recursos de SQL

- Uma instrução especial chamada SET TRANSACTION é usada para definir certas características da próxima transação a ser iniciada. Entre essas características temos:
  - Modo de acesso
  - Nível de isolamento
- A sintaxe é:

SET TRANSACTION < lista \_com\_virgulas de opções>

lista \_com\_virgulas de opções> especifica uma modo de acesso, um nível de isolamento, ou ambas as opções.

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recursos de SQL

- O modo de acesso é:
  - READ ONLY
  - READ WRITE
- Se nenhum deles for especificado, é pressuposto o modo READ WRITE, a menos que o nível de isolamento READ UNCOMMITED seja especificado.
- ▶ Neste caso será suposto o modo READ ONLY.
- ▶ Se READ WRITE for especificado o nível de isolamento não pode ser READ UNCOMMITED.

**4**3

2018/2019 - Bases de Dados - Fernanda Brito Correia

## Recursos de SQL

O nível de isolamento assume a forma,

ISOLATION LEVEL <isolamento>

#### onde <isolamento> é:

- ▶ READ UNCOMMITED
- ▶ READ COMMITED
- ▶ REPEATABLE READ
- ▶ SERIALIZABLE

(mais explicações no capítulo da concorrência)

2018/2019 - Bases de Dados - Fernanda Brito Correia

# Recursos de SQL

> SQL3 (SQL/99) inclui a totalidade do padrão SQL/92 (SQL)

> SQL3 inclui vários recursos novos de gestão de transações:

- ▶ Uma instrução START TRANSACTION explícita
  - > com os mesmos argumentos de SET TRANSACTION
- ▶ Uma opção WITH HOLD em DECLARE CURSOR
- Suporte para pontos de gravação

**45** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

### Transações no ORACLE

- Uma transação inicia-se depois da transação anterior ter terminado.
- ▶ Todos os comandos DML que se seguem fazem parte da mesma transação.
- ▶ São um conjunto de alterações feitas aos dados, que ainda não são definitivas e que apenas são visíveis ao utilizador que as está a efetuar.
- Independentemente destas operações DML virem a ser confirmadas, ou não, o final da transação marca também o início de outra nova.

## Transações no ORACLE

- □ Os comandos de DDL e os comandos de DCL (*Data Control Language*) são por si só uma transação.
- ☐ Isto é, um comando DDL encerra em si três ações:
  - A finalização da transação que estava a decorrer.
  - O início da transação que constitui o próprio comando de DDL.
  - A finalização da transação.
- □ CREATE SCHEMA é um comando DDL que permite juntar vários comandos de DDL na mesma transação

**47** 

2018/2019 - Bases de Dados - Fernanda Brito Correia

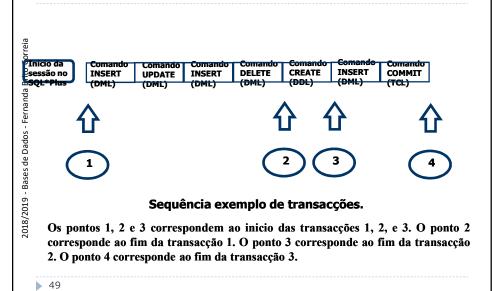
### Transações no ORACLE

O Oracle Server termina a transação quando:

- É emitido um comando DDL ou DCL
- É emitido um comando COMMIT
- É emitido um comando ROLLBACK
- O utilizador está conectado à base de dados, por exemplo através do SQL\*Plus e emite o comando EXIT.
  - É enviado para a base de dados um COMMIT implícito.
- Se a sessão do SQL\*Plus for terminada de forma brusca o Oracle Server vai tentar resolver a transação emitindo um ROLLBACK implícito.
  - Ex. se o utilizador matou o processo, ou devido a uma falha de energia elétrica, falha na rede, ...

**48** 

### Comandos de Gestão Transacional no ORACLE



### Comandos de Gestão Transacional no ORACLE

#### COMMIT

- Confirma todas as transações pendentes da transação atual
- O comando marca o fim da transação

#### SAVEPOINT <nome\_da\_marca>

Cria uma marca no meio da transação.

#### ROLLBACK

- Anula o efeito das alterações feitas pela transação atual.
- O comando marca o fim da transação

#### ROLLBACK TO <nome\_da\_marca>

todas as marcas posteriores são apagadas e a transação continua.

#### **SETTRANSACTION**

Altera as propriedades de uma transação.

**5**0

# Anulação de Transações no ORACLE

► SAVEPOINTS – são marcas na transação, que não confirmam a transação, mas permitem definir um ponto até ao qual a transação pode ser desfeita.

- ▶ Existem dois tipos de marcas:
  - Explicitas
    - ▶ definidas explicitamente pelo utilizador com o SAVEPOINT
  - Implícitas
    - Colocadas pelo Oracle Server no início de cada comando de DML.
    - Não podem ser usadas pelo utilizador.
    - São destinadas a implementar o mecanismo chamada de Anulação ao Nível do Comando DML.

**5**1

2018/2019 - Bases de Dados - Fernanda Brito Correi

# Anulação de Transações no ORACLE

- Exemplo:
  - Uma transação com 301 comandos, onde os primeiros 300 são comandos de INSERT e o último comando da transação é um UPDATE que afeta 500 linhas.
  - No início de cada comando o Oracle Server coloca uma marca que é limpa assim que se inicia o comando seguinte.
  - Se o comando de UPDATE falhar o *Oracle Server* vai implementar um ROLLBACK implícito até à marca que colocou no início do comando.
  - A falha no comando de UPDATE vai manter a transação ativa, com os 300 comandos de INSERT ainda prontos a serem confirmados.

SQL> INSERT ...
... (299 INSERTs mais tarde)
SQL> UPDATE ...
(falhou o comando de UPDATE)
SQL> --Ainda vai a tempo de confirmar os 300 INSERTs
SQL> COMMIT;
Commit complete.

**5**2

# Anulação de Transações no ORACLE

Marcas na transação explícitas:

▶ O comando ROLLBACK pode ser utilizado com referência às marcas.

**5**3