

# ORACLE

## Academy

# Database Programming with SQL

18-1

## Transações de Banco de Dados

**ORACLE**  
Academy



# Objetivos

- Nesta lição, você aprenderá a:
  - Definir os termos COMMIT, ROLLBACK e SAVEPOINT e sua relação com as transações de dados
  - Listar três vantagens das instruções COMMIT, ROLLBACK e SAVEPOINT
  - Explicar por que é importante, da perspectiva dos negócios, ser capaz de controlar o fluxo de processamento de transações

# Finalidade

- E se um banco não tivesse um processo sistemático de registro de depósitos e saques?
- Como você saberia se um depósito foi creditado na sua conta antes que precisasse sacar dinheiro?
- Dá para imaginar a confusão que isso causaria

# Finalidade

- Felizmente, os bancos controlam os processos de transação para garantir a consistência dos dados
- Nesta lição, você aprenderá como o processo de alteração de dados é gerenciado e como as alterações em um banco de dados são submetidas a commit ou canceladas
- COMMIT, ROLLBACK e SAVEPOINT não são suportados no Oracle Application Express devido à maneira como ele gerencia as conexões ao banco de dados

# Transações

- As transações são um conceito fundamental de todos os sistemas de banco de dados
- As transações permitem aos usuários fazer alterações nos dados e depois decidir se vão salvar ou descartar o trabalho
- As transações de banco de dados unem várias etapas em uma unidade lógica de trabalho

# Transações

- Uma transação consiste em um dos seguintes elementos:
  - Instruções DML que representam uma alteração consistente dos dados
  - As instruções DML incluem INSERT, UPDATE, DELETE e MERGE
  - Uma instrução DDL, como CREATE, ALTER, DROP, RENAME ou TRUNCATE
  - Uma instrução DCL, como GRANT ou REVOKE

# Analogia das Transações

- O banco de dados de um banco contém os saldos de várias contas de clientes e os saldos totais de depósito de outras agências
- Vamos supor que um cliente queira sacar e transferir dinheiro de sua conta para a conta de outro cliente em uma agência diferente



# Analogia das Transações

- Existem várias etapas envolvidas na realização dessa operação relativamente simples
- Ambas as agências bancárias querem uma garantia de que todas as etapas da transação ocorram, ou que nenhuma ocorra, e que, se houver uma falha no sistema, a transação não fique parcialmente completa
- O agrupamento das etapas de saque e depósito em uma transação fornece essa garantia
- Uma transação acontece completamente ou não acontece

# Controlando Transações

- Use as seguintes instruções para controlar transações:
  - COMMIT: representa o ponto no tempo em que o usuário fez todas as alterações que deseja agrupar logicamente. Como não houve erros, ele está pronto para salvar o trabalho
    - Quando uma instrução COMMIT é executada, a transação atual é encerrada, tornando permanentes todas as alterações pendentes.
  - ROLLBACK: permite que o usuário descarte as mudanças feitas no banco de dados
    - Quando uma instrução ROLLBACK é executada, todas as alterações pendentes são descartadas

# Controlando Transações

- Use as seguintes instruções para controlar transações:
  - **SAVEPOINT**: cria um marcador em uma transação que a divide em pedaços menores.
  - **ROLLBACK TO SAVEPOINT**: permite ao usuário fazer rollback da transação atual para um ponto de salvamento específico.
    - Se um erro tiver sido cometido, o usuário poderá executar a instrução **ROLLBACK TO SAVEPOINT**, descartando somente as alterações feitas depois que o ponto de salvamento foi estabelecido.

# Exemplo de Transação

- No exemplo mostrado, o usuário executou uma instrução UPDATE e imediatamente criou SAVEPOINT one

```
UPDATE copy_departments  
SET manager_id= 101  
WHERE department_id = 60;
```

```
SAVEPOINT one;
```

- Após uma instrução INSERT e uma instrução UPDATE (no próximo slide), o usuário percebeu que uma cláusula WHERE não foi incluída na segunda instrução UPDATE
- Para corrigir o erro, o usuário executou ROLLBACK TO SAVEPOINT one

# Exemplo de Transação

- Os dados foram restaurados para seu estado em SAVEPOINT one

```
INSERT INTO copy_departments(department_id, department_name,  
    manager_id, location_id)  
VALUES(130, 'Estate Management', 102, 1500);
```

```
UPDATE copy_departments      ← Cláusula WHERE omitida  
SET department_id = 140;
```

```
ROLLBACK TO SAVEPOINT one;
```

```
COMMIT;
```

# Quando uma Transação Começa ou Termina?

- A transação começa com a primeira instrução DML (INSERT, UPDATE, DELETE ou MERGE)
  - Uma transação termina quando ocorre um dos seguintes eventos:
    - Uma instrução COMMIT ou ROLLBACK é executada
    - Uma instrução DDL(CREATE, ALTER, DROP, RENAME ou TRUNCATE) é executada
    - Uma instrução DCL(GRANT ou REVOKE) é executada
    - Um usuário sai normalmente do utilitário do Banco de Dados Oracle, fazendo a transação atual ser implicitamente submetida a commit

# Quando uma Transação Começa ou Termina?

- Depois que uma transação termina, a próxima instrução SQL executável inicia automaticamente a transação seguinte
- Uma instrução DDL ou DCL é submetida a commit automaticamente e, portanto, encerra implicitamente uma transação
- Todas as modificações de dados feitas durante uma transação são temporárias até que ela seja submetida a commit

# Consistência dos Dados

- Imagine passar horas fazendo alterações nos dados dos funcionários e descobrir que outra pessoa estava incluindo informações que conflitavam com as suas alterações!
- Para impedir interrupções ou conflitos assim e permitir que vários usuários accessem o banco de dados ao mesmo tempo, os sistemas de banco de dados empregam uma implementação automática chamada "consistência de leitura"

# Consistência de Leitura

- A consistência de leitura garante uma exibição consistente dos dados para todos os usuários o tempo todo
- Os leitores não visualizam dados que estejam em processo de modificação
- Os gravadores têm a certeza de que as alterações no banco de dados são feitas de maneira consistente
- As alterações feitas por um gravador não destroem ou entram em conflito com as alterações feitas por outro gravador



# Consistência de Leitura

- A consistência de leitura é uma implementação automática
- Uma cópia parcial do banco de dados é mantida em segmentos de undo. Quando o Usuário A executa uma operação de inserção, atualização ou exclusão no banco de dados, o servidor Oracle captura um instantâneo (cópia) dos dados antes de serem alterados e os grava em um segmento de undo (rollback)
- O Usuário B ainda vê o banco de dados como ele existia antes de as alterações começarem; ele vê o instantâneo do segmento de undo dos dados



# Consistência de Leitura

- Antes de as alterações serem submetidas a commit no banco de dados, somente o usuário que está modificando os dados vê; os outros usuários veem o instantâneo no segmento de undo
- Isso garante que os leitores vejam dados consistentes, que não estão passando por mudanças



# Mudanças Visíveis

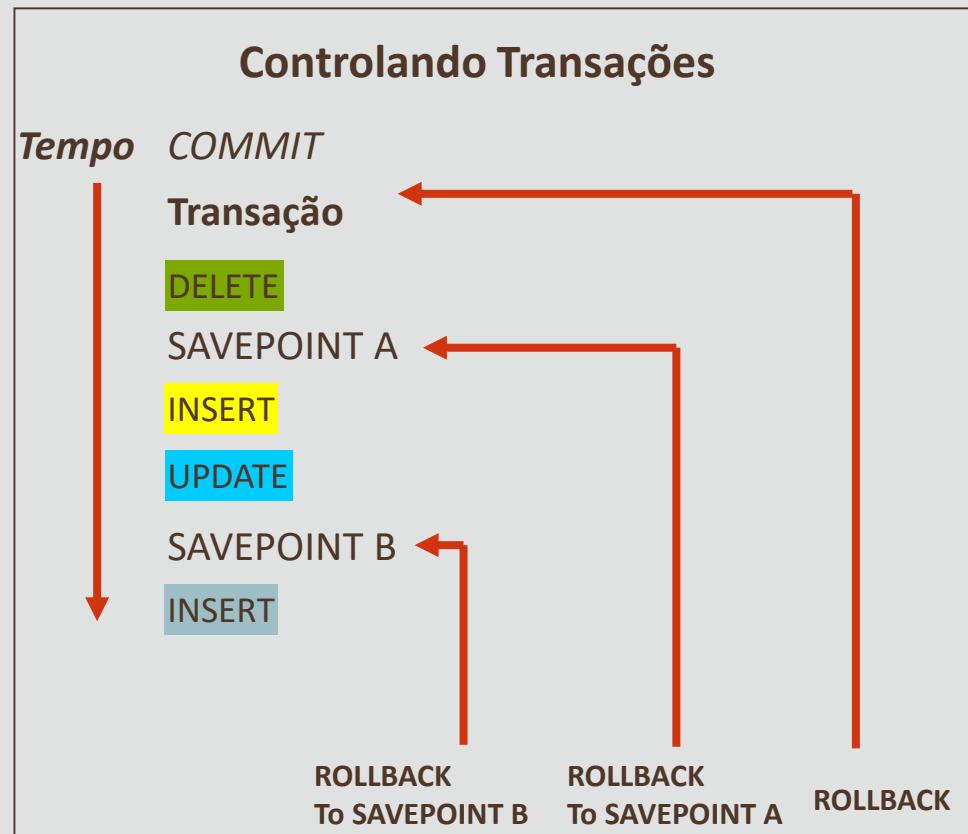
- Quando uma instrução DML é submetida a commit, a mudança feita no banco de dados se torna visível a qualquer um que esteja executando uma instrução SELECT
- Se a transação for submetida a rollback, as mudanças são desfeitas:
  - A versão mais antiga original dos dados no segmento undo é gravada de volta na tabela
  - Todos os usuários veem o banco de dados como ele existia antes de a transação começar

# COMMIT, ROLLBACK e SAVEPOINT

- COMMIT e ROLLBACK garantem a consistência dos dados, tornando possível ver as alterações dos dados antes de torná-las permanentes e agrupar operações logicamente relacionadas
- SAVEPOINT cria um ponto na transação para o qual podemos fazer rollback sem precisar desfazer toda a transação
- COMMIT, ROLLBACK e SAVEPOINT são conhecidos como Transaction Control Language, ou TCL

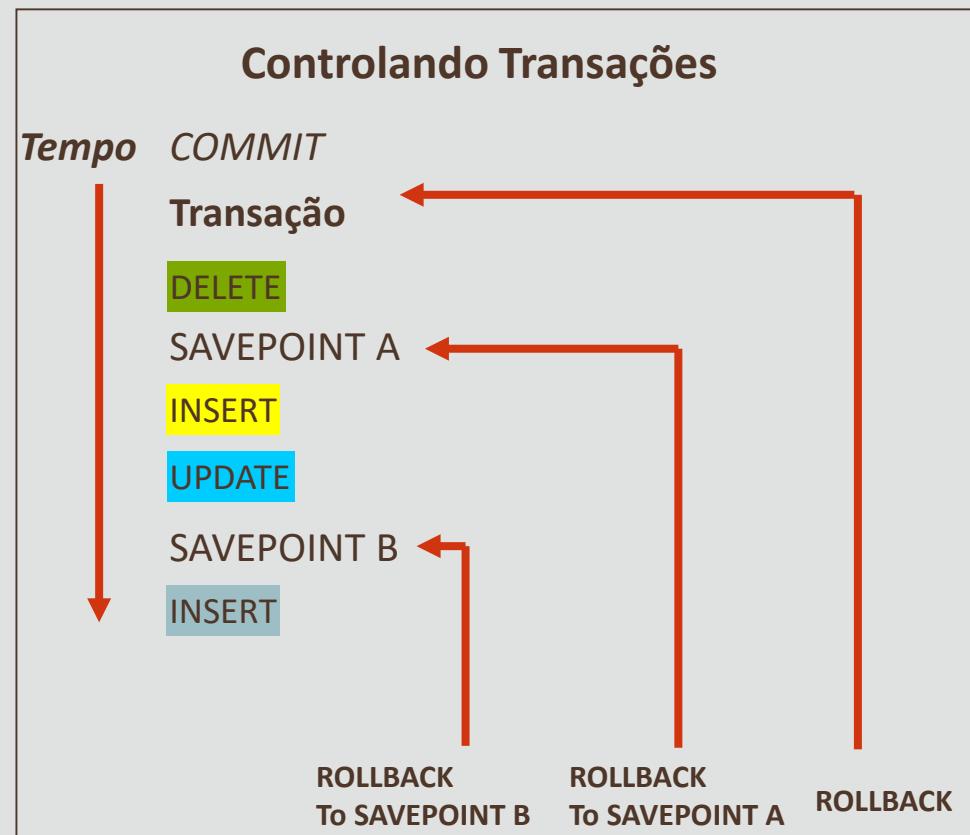
# COMMIT, ROLLBACK e SAVEPOINT

- Na transação mostrada no gráfico, uma instrução DELETE foi executada e o PONTO DE SALVAMENTO A foi estabelecido
- O ponto de salvamento age como um marcador que permitirá ao usuário fazer rollback de quaisquer alterações subsequentes feitas nos dados de volta para o estado em que estavam nesse ponto



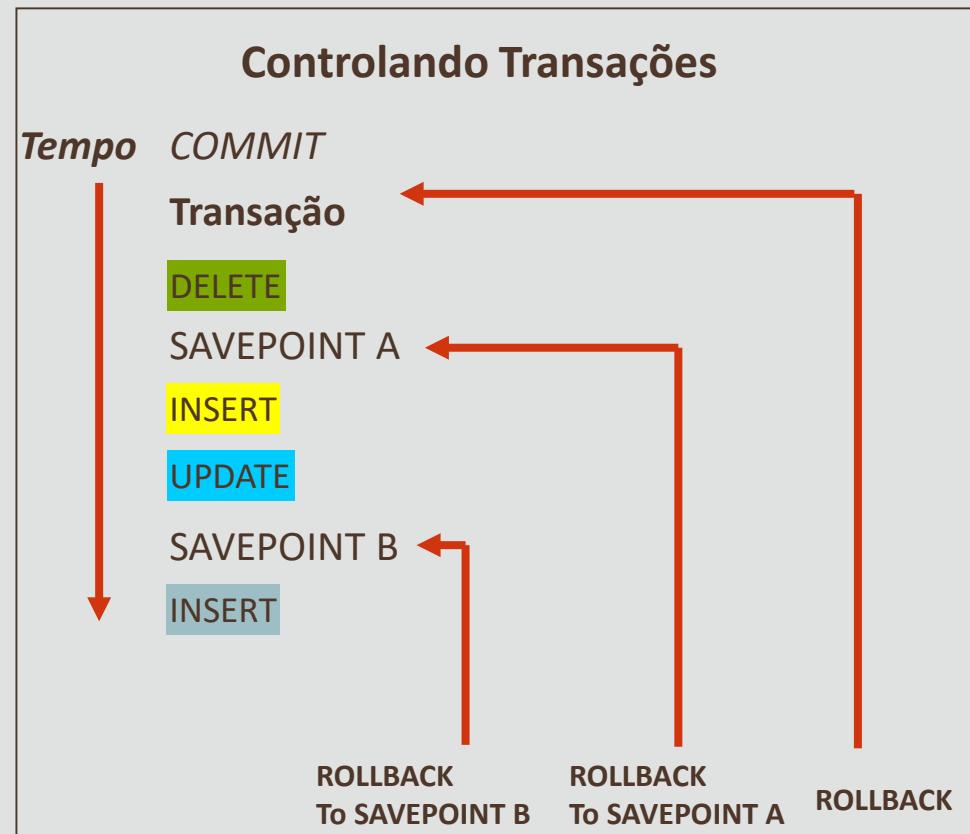
# COMMIT, ROLLBACK e SAVEPOINT

- No exemplo, após o PONTO DE SALVAMENTO A, o usuário executa as instruções INSERT e UPDATE e estabelece outro marcador de rollback no PONTO DE SALVAMENTO B



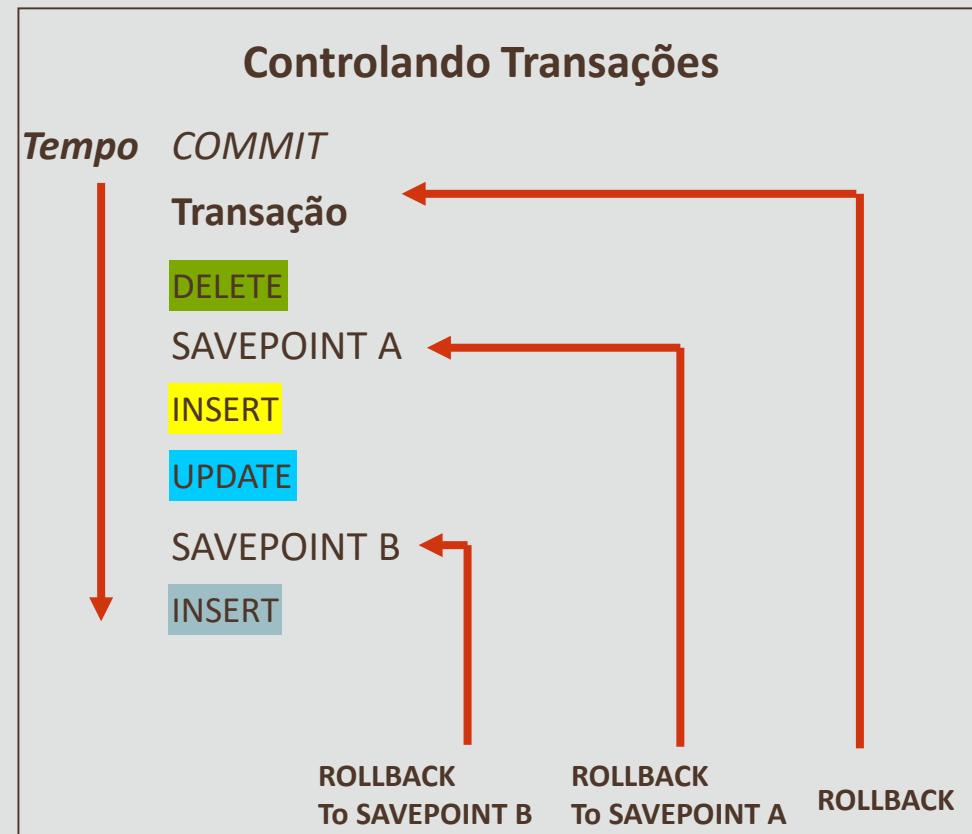
# COMMIT, ROLLBACK e SAVEPOINT

- Se, por algum motivo, não quiser que as instruções INSERT e/ou UPDATE ocorram, o usuário poderá executar a instrução ROLLBACK TO SAVEPOINT A
- Isso fará rollback para o estado em que os dados estavam no marcador SAVEPOINT A



# COMMIT, ROLLBACK e SAVEPOINT

- O acréscimo de outros pontos de salvamento cria mais marcadores para pontos de rollback
- Se um usuário executar ROLLBACK sem uma instrução ROLLBACK TO SAVEPOINT, toda a transação será encerrada e todas as alterações de dados pendentes serão descartadas



# Processamento de Transações Implícito

- O commit automático das alterações de dados ocorre nas seguintes circunstâncias:
  - uma instrução DDL é executada
  - uma instrução DCL é executada
  - um usuário sai normalmente do utilitário do Banco de Dados Oracle, fazendo a transação atual ser implicitamente submetida a commit
  - execução explícita de instruções COMMIT ou ROLLBACK

# Processamento de Transações Implícito

- O rollback automático ocorre após um encerramento anormal do utilitário do Banco de Dados Oracle ou quando há uma falha no sistema
- Isso impede que erros nos dados causem mudanças indesejadas nas tabelas subjacentes
- Portanto, a integridade dos dados é protegida

# Bloqueio

- É importante impedir que os dados sejam alterados por mais de um usuário por vez
- O Oracle usa bloqueios que impedem interação destrutiva entre as transações que acessam o mesmo recurso, seja um objeto do usuário (como tabelas ou linhas) ou um objeto do sistema não visível para os usuários (como estruturas de dados compartilhados e linhas do dicionário de dados)



# Como o Banco de Dados Oracle Bloqueia os Dados

- O bloqueio do Oracle é executado automaticamente e não requer ação do usuário
- O bloqueio implícito ocorre para as instruções SQL conforme necessário, dependendo da ação solicitada
- O bloqueio implícito ocorre para todas as instruções SQL, exceto SELECT
- Os usuários também podem bloquear os dados manualmente, o que é chamado de bloqueio explícito
- Quando uma instrução COMMIT ou ROLLBACK é executada, os bloqueios nas linhas afetadas são desfeitos

# Terminologia

- Estes são os principais termos usados nesta lição:
  - Transação
  - Commit
  - Ponto de salvamento
  - Rollback
  - Transaction Control Language
  - Consistência de leitura
  - Bloqueios

# Resumo

- Nesta lição, você aprendeu a:
  - Definir os termos COMMIT, ROLLBACK e SAVEPOINT e sua relação com as transações de dados
  - Listar três vantagens das instruções COMMIT, ROLLBACK e SAVEPOINT
  - Explicar por que é importante, da perspectiva dos negócios, ser capaz de controlar o fluxo de processamento de transações

# **ORACLE**

## Academy