



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

Desenvolvimento WEB II

Banco de Dados e SQL: Transações.



Roteiro

- Banco de Dados e SQL: Transações:
 - O que é uma Transação?
 - Propriedades ACID;
 - Implementando Transações em SQL;
 - Controle de Concorrência;
 - Atividade Prática.



O que é uma Transação?

Uma transação em banco de dados é uma sequência de operações realizadas como uma única unidade de trabalho.

Isso significa que **todas as operações dentro da transação devem ser concluídas com sucesso**; caso contrário, todas as alterações feitas devem ser desfeitas (rollback).



O que é uma Transação?

Exemplo prático:

Imagine que você esteja transferindo dinheiro de uma conta bancária para outra. A operação de débito na conta de origem e a operação de crédito na conta de destino devem ser tratadas como uma única transação. Se qualquer uma dessas operações falhar, nenhuma alteração deve ser feita, garantindo que o dinheiro não desapareça no processo.



Propriedades ACID

As transações devem cumprir quatro propriedades conhecidas como ACID:

- **Atomicidade (Atomicity):** Garante que todas as operações dentro da transação sejam completadas com sucesso ou que nenhuma alteração seja feita.
- **Consistência (Consistency):** Garante que a transação leve o banco de dados de um estado consistente para outro estado consistente.



Propriedades ACID

As transações devem cumprir quatro propriedades conhecidas como ACID:

- **Isolamento (Isolation):** Garante que as operações de transações concorrentes não interfiram umas nas outras.
- **Durabilidade (Durability):** Garante que, uma vez que a transação seja concluída, as alterações sejam permanentes no banco de dados, mesmo em caso de falha do sistema.



Propriedades ACID

Exemplo prático:

Imagine que duas pessoas tentem sacar o último dinheiro disponível em uma conta ao mesmo tempo. O isolamento garante que apenas uma dessas operações seja realizada com sucesso, evitando inconsistências.



Implementando Transações em SQL

As transações em SQL são gerenciadas usando comandos específicos:

- **BEGIN TRANSACTION** ou **START TRANSACTION**: Inicia uma nova transação.
- **COMMIT**: Confirma todas as alterações feitas durante a transação, tornando-as permanentes.
- **ROLLBACK**: Desfaz todas as alterações feitas durante a transação, retornando o banco de dados ao estado original.



Implementando Transações em SQL

Exemplo SQL:

```
1  START TRANSACTION;
2
3  UPDATE contas SET saldo = saldo - 100 WHERE id = 1;
4  UPDATE contas SET saldo = saldo + 100 WHERE id = 2;
5
6  COMMIT;
```



Implementando Transações em SQL

Neste exemplo, os **comandos de atualização** (UPDATE) são realizados dentro de uma transação. Se ambos os comandos forem bem-sucedidos, o **COMMIT** será executado, e as mudanças serão salvas. Se qualquer comando falhar, o **ROLLBACK** deve ser executado para desfazer as alterações.



Controle de Concorrência

Quando **várias transações são executadas simultaneamente**, pode haver conflitos. Por exemplo, duas transações podem tentar modificar os mesmos dados ao mesmo tempo.

Para gerenciar isso, os **bancos de dados** usam **técnicas** como:



Controle de Concorrência

- **Bloqueios (Locks):** Asseguram que uma transação tenha acesso exclusivo a certos dados até que seja concluída.
- **Controle de Versão (MVCC - Multiversion Concurrency Control):** Permite que várias transações leiam e escrevam dados ao mesmo tempo, sem bloqueios, mantendo diferentes versões dos dados.



Controle de Concorrência

Exemplo prático:

Se uma **transação A** estiver atualizando o saldo de uma conta, a **transação B** terá que esperar até que a **transação A** termine para acessar esse saldo, *garantindo que os dados não sejam corrompidos.*



Conclusão

Transações são fundamentais para garantir a integridade e consistência dos dados em um banco de dados, especialmente em sistemas que realizam múltiplas operações simultâneas. Entender e implementar corretamente as propriedades ACID e o controle de concorrência é crucial para o desenvolvimento de aplicações robustas e seguras.



Atividade Prática

Questão:

1. Implemente uma transação em SQL que realize a transferência de \$500 da conta 1 para a conta 2, e garanta que em caso de qualquer erro durante o processo, nenhuma alteração seja feita no banco de dados.



Exercício Prático

Resposta:

```
1  START TRANSACTION;
2
3  UPDATE contas SET saldo = saldo - 500 WHERE id = 1;
4  UPDATE contas SET saldo = saldo + 500 WHERE id = 2;
5
6  --ROLLBACK;
7
8  COMMIT;
```

Se houver algum erro, o comando ROLLBACK; deve ser utilizado para reverter todas as operações.



Obrigado!

Questões?

