04/12/2024

LAURA MATIOLE
SAMUEL SANTANA
LEONARDO FORMENTON

---

1) **Defina o que é TCL e sua importância em um banco de dados.**
   TCL (Transaction Control Language) gerencia transações em um banco de dados, garantindo que as operações sejam completadas corretamente ou revertidas em caso de erro.

   Liste e explique os principais comandos da TCL:
   - COMMIT: Confirma e torna as alterações permanentes.
   - ROLLBACK:Desfaz as alterações, retornando ao estado anterior
   - SAVEPOINT:Cria um ponto de salvamento para reverter parcialmente.
   - SET TRANSACTION:Define características da transação, como o nível de isolamento.

2) **Explique como as transações funcionam:**

   O que é uma transação em um banco de dados?

   -Transação pode ser definida como uma unidade de trabalho realizada dentro de um SGBD (Sistema de Gerenciamento de Banco de Dados). É o conjunto de uma ou mais operações que compõem uma tarefa com o objetivo de realizar uma alteração no estado final do banco de dados.

   Como o MySQL lida com transações de banco de dados?

   O MySQL lida com transações de banco de dados por meio do uso de comandos específicos que garantem a integridade e a consistência dos dados. Além disso, por padrão, o MySQL utiliza o modo autocommit, que significa que declarações INSERT, UPDATE ou DELETE sofrem commit automaticamente quando executadas. Vamos usar transações para que o commit (ou rollback) sejam executados quando nós o quisermos, de modo a ter maior controle sobre a execução das declarações DML.

O que acontece se a transação for bem-sucedida ou falhar?
Se a transação for bem-sucedida, o comando COMMIT é executado, o que faz com que todas as operações realizadas durante a transação sejam confirmadas e persistam no banco de dados. Isso significa que todas as alterações feitas se tornam permanentes e visíveis para outras transações.

Se algo der errado durante a execução de uma transação, o MySQL reverte todas as operações realizadas até o momento utilizando o comando ROLLBACK. Isso garante que o banco de dados não seja deixado em um estado inconsistente. A transação é "cancelada", e o banco de dados retorna ao estado em que estava antes do início da transação.

Como garantir a integridade dos dados usando transações?

Para garantir a integridade dos dados, usamos métodos comuns usados para verificação de integridade de dados que incluem:

-Limite o acesso aos dados e altere as permissões para restringir modificações nos dados por partes não aprovadas.

-Concentre-se na validação de dados para garantir a precisão dos dados quando coletados ou integrados.

-Mantenha um backup regular dos dados.

-Use logs para monitorar quando os dados são inseridos, alterados ou apagados.

-Realize auditorias internas sistemáticas para garantir que as informações estejam atualizadas.

3) Estudo de caso:

Pesquise um exemplo de quando a utilização de transações pode ser vantajosa em um sistema de banco de dados.

```
CREATE TABLE contas_bancarias (
    id INT AUTO_INCREMENT PRIMARY KEY,
    titular VARCHAR(100),
    saldo DECIMAL(10, 2)
```

```
);

CREATE TABLE transacoes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    conta_origem INT,
    conta_destino INT,
    valor DECIMAL(10, 2),
    data_transacao DATETIME
);

-- Inserir algumas contas com saldo inicial
INSERT INTO contas_bancarias (titular, saldo) VALUES
('João', 1000.00),
('Maria', 500.00);

-- Vamos verificar as contas antes da transação
SELECT * FROM contas_bancarias;

START TRANSACTION;

-- Operação 1: Retirar dinheiro da conta do João
UPDATE contas_bancarias
SET saldo = saldo - 200.00
WHERE titular = 'João' AND saldo >= 200.00;

-- Verificar se o saldo do João foi alterado com sucesso
SELECT * FROM contas_bancarias WHERE titular = 'João';

-- Operação 2: Adicionar dinheiro na conta da Maria
UPDATE contas_bancarias
SET saldo = saldo + 200.00
WHERE titular = 'Maria';

-- Verificar se o saldo da Maria foi alterado com sucesso
SELECT * FROM contas_bancarias WHERE titular = 'Maria';

-- Operação 3: Registrar a transação na tabela de transações
INSERT INTO transacoes (conta_origem, conta_destino, valor, data_transacao)
VALUES ((SELECT id FROM contas_bancarias WHERE titular = 'João'),
        (SELECT id FROM contas_bancarias WHERE titular = 'Maria'),
        200.00, NOW());

-- Se tudo ocorrer bem, confirmamos a transação
COMMIT;

START TRANSACTION;

-- Operação 1: Retirar dinheiro da conta do João
```

```
UPDATE contas_bancarias
SET saldo = saldo - 200.00
WHERE titular = 'João' AND saldo >= 200.00;

-- Simulando um erro (por exemplo, conta da Maria não encontrada)
-- Vamos forçar um erro ao tentar transferir para uma conta que não existe
UPDATE contas_bancarias
SET saldo = saldo + 200.00
WHERE titular = 'Não Existe';

-- Se algum erro ocorrer, a transação será revertida
ROLLBACK;
```

Explique como as transações podem ser usadas para garantir que todas as operações de uma tarefa sejam realizadas ou, caso algo falhe, nenhuma delas seja realizada.

As transações garantem que todas as operações de uma tarefa sejam realizadas ou, em caso de falha, nenhuma delas seja realizada. Isso é possível graças ao conceito **ACID**, que define quatro propriedades essenciais:

Atomicidade: Ou todas as operações são concluídas ou nenhuma delas é aplicada.
Consistência: O banco de dados permanece em um estado válido antes e depois da transação.
Isolamento: As operações de uma transação são isoladas das outras até sua conclusão.
Durabilidade: Uma vez concluída, a transação é permanente, mesmo em caso de falhas no sistema.
Em caso de erro, o sistema pode reverter (rollback) as mudanças feitas, garantindo a integridade dos dados. Isso é feito com comandos como BEGIN TRANSACTION, COMMIT (para confirmar) e ROLLBACK (para desfazer).

4) **Criação de um exemplo prático:**

```
create table cliente(
idCliente int primary key,
nome varchar(100)
);

create table pedido(
idPedido int primary key,
dataPedido date,
idCliente int,
foreign key (idCliente) references cliente(idCliente)
);
```

```
START TRANSACTION;

insert into cliente(idCliente, nome) values
(1, 'Laura'),
(2, 'Leonardo'),
(3, 'Samuel');
SAVEPOINT cliente_inserted;

INSERT INTO pedido (idPedido, idCliente, dataPedido) VALUES
(101, 1, NOW()),
(22, 2, NOW()),
(102, 3, NOW());

COMMIT;
ROLLBACK TO SAVEPOINT cliente_inserted;
```

No MySQL Workbench

5. Conclusão:
Resuma a importância da TCL na gestão de transações dentro de sistemas de banco de dados.

O TCL é crucial no gerenciamento de banco de dados, pois garante a integridade e a consistência dos dados dentro das transações. Ele permite que o banco de dados permaneça em um estado consistente mesmo em caso de erros durante uma transação.

Quais são as vantagens de usar transações em sistemas críticos, como sistemas bancários, de vendas online e outros sistemas de grande porte?

Resposta Rápida: O TPS deve processar transações com baixo tempo de resposta, geralmente abaixo de um segundo.
Inflexibilidade: O processamento de transações é padronizado para garantir consistência e evitar erros.
Confiabilidade: O sistema possui backup e recuperação robustos para garantir continuidade e proteção de dados sensíveis.
Processamento Controlado: As transações seguem uma ordem específica, alinhada à estrutura da empresa, preservando a integridade dos dados.
Processamento em Tempo Real: As transações são processadas instantaneamente, garantindo tempo de resposta rápido.
Controle de Simultaneidade: Mecanismos garantem que transações simultâneas sejam processadas corretamente, evitando inconsistências.

Ademais, temos também o:

Processamento em Tempo Real (TPS): Processa transações à medida que ocorrem, oferecendo atualizações e respostas quase imediatas. É usado quando dados atualizados são necessários rapidamente, como em transações bancárias online, bolsas de valores e sistemas de reserva.

Processamento em Lote (TPS): Agrupa transações em lotes de tamanho predefinido e as processa periodicamente. É utilizado para tarefas que não exigem resposta imediata, como geração de relatórios, processamento de folha de pagamento e envio de faturas.

FONTE:

https://pt.linkedin.com/pulse/o-que-%C3%A9-transa%C3%A7%C3%A3o-em-banco-de-dados-controle-e-deadlock-ladeira

https://www.bosontreinamentos.com.br/mysql/como-trabalhar-com-transacoes-em-mysql/#:~:text=Por%20padr%C3%A3o%2C%20o%20MySQL%20utiliza,a%20execu%C3%A7%C3%A3o%20das%20declara%C3%A7%C3%B5es%20DML.

https://www.astera.com/pt/type/blog/data-integrity-in-a-database/#:~:text=Como%20garantir%20a%20integridade%20dos%20dados%20em%20um%20banco%20de%20dados&text=Concentre%2Dse%20na%20valida%C3%A7%C3%A3o%20de,s%C3%A3o%20inseridos%2C%20alterados%20ou%20apagados.

https://medium.com/@alrazak/sql-transaction-control-language-tcl-3fde7f979625

https://dashdevs.com/blog/what-is-a-transaction-processing-system-tps-types-and-usages/