



## PLANEJAMENTO DE AULA EM SALA

Curso: Engenharia da Computação

Disciplina: Banco de Dados  
Prof. Eduardo Chagas de Oliveira

### ROTEIRO DE AULA PRÁTICA

#### TÍTULO DA AULA PRÁTICA: Transações

##### Objetivos

- Aplicar na prática o funcionamento do controle transacional no banco de dados.
- Capacitar a resolução de problemas que requerem o uso de controle de concorrência no banco de dados.
- Realizar operações utilizando uma mesma transação.

##### Transação

É uma unidade lógica de processamento que garante a atomicidade das informações e tem por objetivo preservar a integridade e a consistência dos dados, quando existe a presença de concorrência.

Sintaxe básica:

```
Begin Transaction  
-- corpo do comando  
Commit/Rollback
```

Onde:

- **Begin Transaction:** tag que delimita o início de uma transação
- **Corpo do comando:** conjunto de comandos a serem executados dentro da transação, operações como de inserção, seleção, alteração e exclusão.
- **Commit ou Rollback:** comandos que finalizam a transação. Commit confirma sucesso realizado pelo conjunto de comandos e Rollback desfaz o processo executado até o início transacional.

No SQL Server é possível realizar um processo de decisão se houve erro através da chamada @@ERROR com o retorno 0 (zero) caso não ocorra erros e 1 (um) para quando houver um erro. Segue um exemplo, considerando o diagrama da Figura 1.

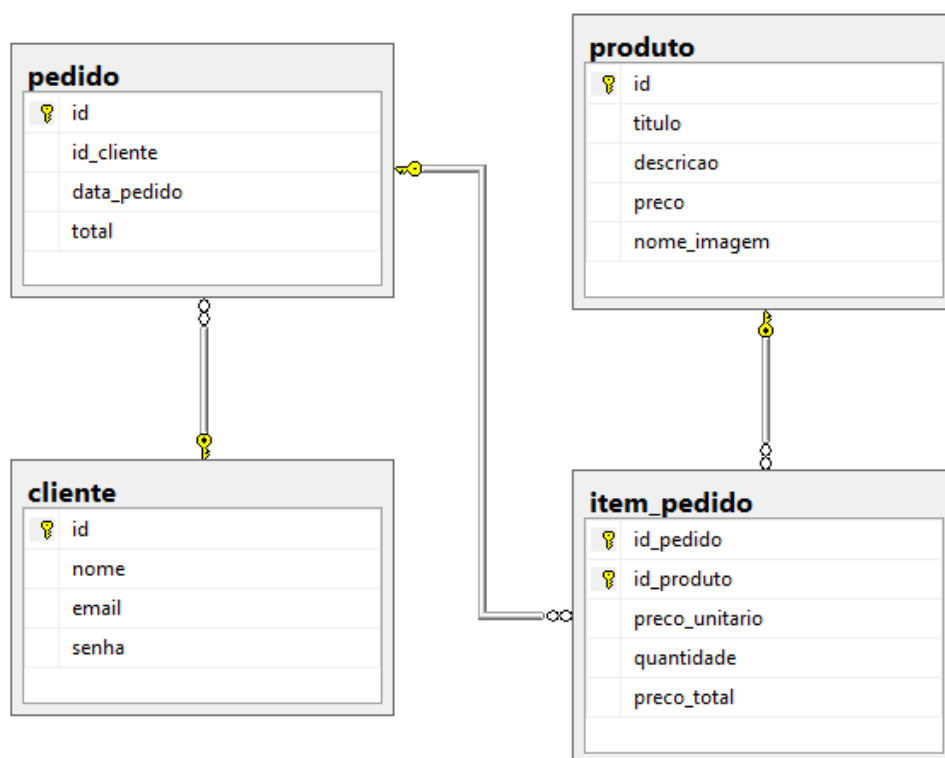


Figura 1: Diagrama de entidade e relacionamento de um sistema para *ecommerce*.

```

(...)
BEGIN TRANSACTION

update from pedido
set preco_total = 1300000.00      --- Pode ocasionar um erro de casas decimais
where id_pedido = 3

IF @@ERROR = 0
    COMMIT
ELSE
    ROLLBACK
END
  
```

Esta transação pode ser adicionada no corpo de uma procedure e junto a uma função PRINT para imprimir de forma detalhada o comportamento de erro e facilitar a rastreabilidade.

```

CREATE PROCEDURE sp_aditivo_valor_pedidos
@valor DECIMAL(11, 2)
AS

BEGIN TRANSACTION;

update pedido
set total = total + @valor;

IF @@ERROR = 0
    COMMIT
ELSE
    ROLLBACK
    PRINT 'Erro no aditamento dos valores dos pedidos'
  
```

Para executar a procedure com a transação basta passar um parâmetro decimal, por exemplo 10.

```
EXECUTE sp_aditivo_valor_pedidos 10
```

Existem ainda outras formas de se gerenciar as transações. Uma forma é através de blocos de tratamento TRY/CATCH e o uso de @@TRANCOUNT para verificar se há transação corrente executando.

```
CREATE PROCEDURE sp_aditivo_valor_pedidos
@valor DECIMAL(11, 2)
AS
BEGIN
    BEGIN TRANSACTION;
    SAVE TRANSACTION pontoRetorno;

    BEGIN TRY
        update pedido
        set total = total + @valor;

        COMMIT;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            ROLLBACK TRANSACTION pontoRetorno;
            PRINT 'Erro no aditamento dos valores dos pedidos';
        END
    END CATCH
END
```

## Exercícios

1. Utilizando o DER da Figura 1:
  - a) Execute a procedure *sp\_aditivo\_valor\_pedidos* com um valor que possa causar um erro em algum registro na tabela de *pedido*, com estouro de valor. Mostre que nenhum outro registro é modificado e a transação é garantida.
  - b) Crie uma *procedure* chamada *sp\_poc\_delecao\_registros* para deletar um pedido que não tenha integridade na chave estrangeira e, logo em seguida, um registro na *pedido* que possua referência na chave estrangeira. Verifique que, como a política de delação é RESTRICT, não haverá nenhuma delação, dado a atomicidade transacional.

## Referência Bibliográfica

FEITOSA, M. P. **Fundamentos de Banco de Dados**. Uma abordagem prático-didática. ed. 1, São Paulo: Independente, 2013.

**Introdução aos Stored Procedures no SQL Server**. Disponível em:

<<https://www.devmedia.com.br/introducao-aos-stored-procedures-no-sql-server/7904>>. Acessado em: 06 nov. 2017.