

Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Daniel Moreira Cardoso
Gusttavo Nunes Gomes
Jeferson Rossini Ferreira Lourenço
Paulo Henrique Rodrigues Araujo
Warley Rodrigues de Andrade

JDBC com transações

Outubro
2017

Sumário

1	O que é JDBC?	2
2	O que é transação?	2
3	Referências Bibliográfica	5

JDBC com transações

1 O que é JDBC?

Java Database Connectivity (JDBC) é um API (Conjunto de classes e interfaces) que permite a comunicação da linguagem JAVA com qualquer banco de dados relacional. Porém é necessário um driver específico para cada SGBD. [1]

2 O que é transação?

Transação é o nome que é dado à unidade de trabalho (dentro de um procedimento ou função) única, lógica e indivisível dentro de um SGBD. O código da transação é criado e disponibilizado dentro de um procedimento ou uma função: se o processamento não for realizado de forma completa ele será totalmente cancelado. Isto é feito para manter a consistência dos dados e das operações dentro do banco. [2]

Um bom exemplo de transação é a transferência bancária entre contas. Onde o valor é debitado da conta de um cliente e depois adicionado na conta de outro.

Para fazer esta transação é muito simples, conforme o código abaixo:

id	nome	saldo
1	Daniel	900
2	Paulo	200

```
1 update clientes set saldo = saldo-200 where id = 1;  
2 update clientes set saldo = saldo+200 where id = 2;
```

recursos/codigo/01.sql

Para a transação ocorrer os dois comandos devem ser válidos, se a transação for válida os dados são enviados ao banco de dados (*commit*). Caso não seja válida, ou se algo der errado o procedimento deve ser abortado (*rollback*).

Quando se fala em *commit* e *rollback* é importante falar de *autocommit*, que permite que as operações sejam commitadas automaticamente. A maioria dos SGBDS usam por modo padrão o *Autocommit*, ou seja, quando são executados comandos de UPDATE, INSERT ou DELETE sem iniciar uma transação explícita, essas operações são commitadas de forma automática.

Porém as transações normalmente são feitas a nível de programação e não a nível de banco de dados (*procedures*) para maior facilidade caso seja necessário migrar a base de dados para um outro SGBD.

Agora que já vimos o que é o JDBC e transações, vamos entender como fazer transações com JDBC.

Para fazer commit com java basta chamar o método `commit()` da interface `connection`.

```
1 connection.commit();
```

O método `rollback` também está na interface `connection`

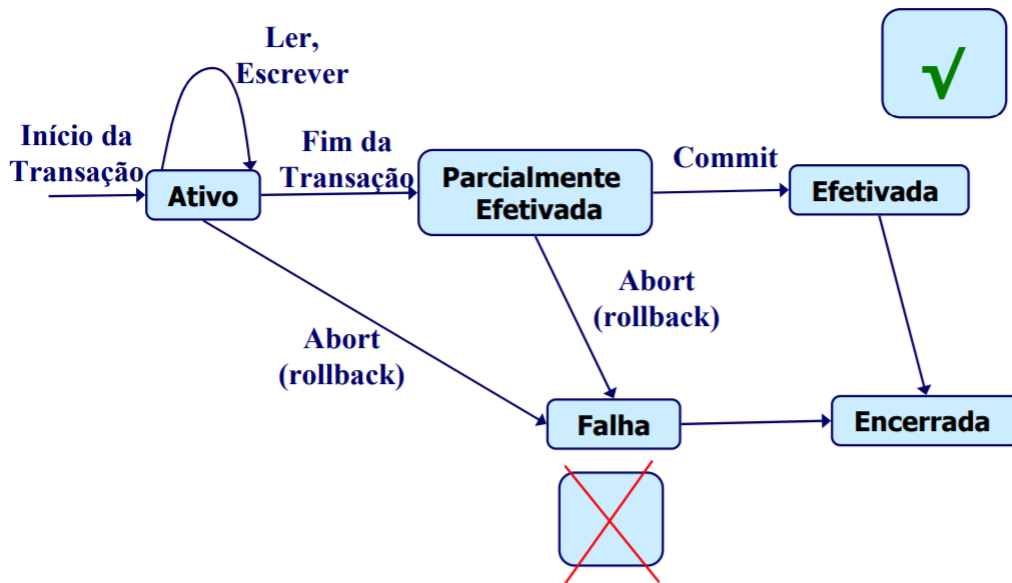


Figura 1: Fonte: <https://www.ime.usp.br/~jef/TransacoesControleConcorrencia.pdf>

```
1 connection.rollback();
```

O auto commit também pode ser feito em java e permite gerenciar automaticamente se será feito o commit, basta no início do método chamar o método passando como argumento um valor lógico (true ou false);

```
1 connection.setAutoCommit(true);
```

Então vejamos um exemplo bastante genérico de uma transação em JAVA:

```
1 connection.setAutoCommit(false);
2 // Digite o sql aqui
3
4 connection.commit();
5 // se der errado
6
7 connection.rollback();
```

recursos/codigo/02.java

Implementando um exemplo com uma estrutura de controle.

```
1 // Considere que já tenha sido declarado e instanciados todos os objetos
  // necessários
2 try {
3     connection.setAutoCommit(false);
4     PreparedStatement = connection.prepareStatement("Um SQL qualquer
    aqui");
5
6     PreparedStatement.executeUpdate();
7
8     connection.commit();
9 } catch (Exception e) {
```

```
10 connection.rollback();  
11 }
```

recursos/codigo/03.java

3 Referências Bibliográfica

[1] <https://pt.wikipedia.org/wiki/JDBC>

[2] REVISTABW. Transações em Bancos de Dados.Revista Brasileira de Web: Tecnologia. Disponível em: <<http://www.revistabw.com.br/revistabw/transacoes-em-bancos-de>>
Criado em: 03/01/2013. Última atualização: 24/07/2015. Visitado em: 31/10/2017

[3] <https://www.devmedia.com.br/aprendendo-java-com-jdbc/29116>

[4] <http://www.guj.com.br/t/transacoes-em-java/29674>