



Transações



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



O que são transações?

Uma transação é um conjunto de operações em um banco de dados que deve ser executado como uma unidade.





O que são transações?

Há ocasiões em que é necessário a realização de várias operações no banco de dados como um único bloco, ou seja, que sejam executadas todas as operações ou nenhuma, mas que não realize apenas algumas dessas operações.

Se forem executadas parcialmente até o momento em que uma operação falhar, o estado dos dados no banco de dados podem ficar inconsistentes. Neste caso precisamos de um mecanismo para devolvê-lo ao estado anterior, podendo desfazer todas as operações realizadas.





Exemplo

Suponha que vamos reservar um voo de São Paulo para Sydney.

Para chegar a Sydney, temos que fazer várias paradas. Teremos que reservar um de São Paulo a Madrid, outro de Madrid a Dubai e, finalmente, voar de Dubai a Sydney. Se representarmos cada reserva como uma inserção no banco de dados, teremos as seguintes instruções:





```
try {  
    s.executeUpdate("INSERT INTO RESERVAS (passageiro, origem, destino) VALUES ('Felipe',  
        'São Paulo', 'Madrid')");  
    s.executeUpdate("INSERT INTO RESERVAS (passageiro, origem, destino) VALUES ('Felipe',  
        'Madrid', 'Dubai')");  
    s.executeUpdate("INSERT INTO RESERVAS (passageiro, origem, destino) VALUES ('Felipe',  
        'Dubai', 'Sydney')");  
} catch(SQLException e) {  
    // Ocorreu um erro, mas não sabemos onde  
    // Quais operações foram realizadas? Como desfazê-la?  
}
```



Um objeto **Connection** por padrão executa automaticamente todas as operações no banco de dados. Isso significa que toda vez que uma instrução é executada, ela é refletida no banco de dados e não pode ser desfeita. Por padrão, o **modo auto-commit** na conexão está habilitado.

Os seguintes métodos na interface de conexão (Connection) são usados para gerenciar transações no banco de dados:

```
void setAutoCommit(boolean valor)
void commit()
void rollback()
```



Para iniciar uma transação, desabilitamos o modo auto-commit usando o método **setAutoCommit(false)**. Isso nos dá o controle sobre o que é realizado e quando é realizado.

Um chamada para o método **commit()** realizará todas as instruções emitidas desde a última chamada do método commit().

Uma chamada para **rollback()** irá desfazer todas as mudanças realizadas desde o último commit(). **Depois que uma instrução commit() é emitida, essas transações não podem ser revertidas com o rollback().**



DigitalHouse>
Coding School