

# O que é um Deadlock?

## O que é um impasse (deadlock)?

Um deadlock ocorre quando dois ou mais processos estão aguardando um ao outro para liberar os recursos de que precisam para continuar. Isso resulta em uma situação em que nenhum dos processos pode prosseguir e eles acabam esperando indefinidamente.

## Condições de Coffman

As condições de Coffman, em homenagem a Edward G. Coffman Jr., que as descreveu pela primeira vez em 1971, descrevem quatro condições necessárias que devem estar presentes para que ocorra um impasse:

- **Exclusão mútua:** os recursos não podem ser compartilhados simultaneamente.
- **Hold and Wait:** Os processos retêm recursos enquanto aguardam outros.
- **Sem preempção:** os recursos não podem ser retirados à força.
- **Espera circular:** Uma cadeia circular de processos, cada um esperando por um recurso mantido pelo próximo.

## Prevenção de deadlock

- **Ordenação de recursos:** exija que os processos solicitem recursos em uma ordem específica.
- **Tempos limite:** defina limites de tempo para o uso de recursos para evitar retenções indefinidas.
- **Algoritmo do banqueiro:** Um método que verifica se a concessão de uma solicitação de recurso levaria a um estado seguro.

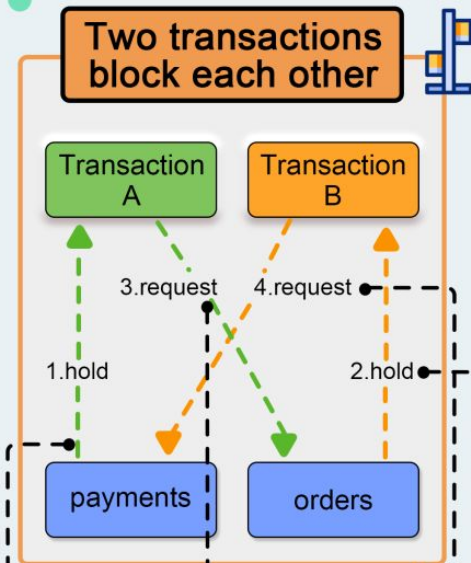
## Recuperação de deadlock

- **Selecionando uma vítima:** escolha um ou mais processos para encerrar ou reverter.
- **Reversão:** Desfaça as ações do(s) processo(s) selecionado(s) para liberar recursos.

# What is a Deadlock?



## Two transactions block each other



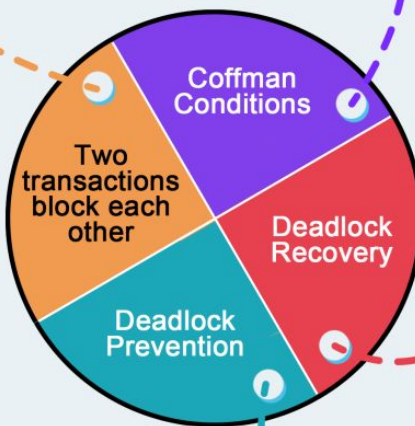
## Coffman Conditions

**Mutual Exclusion**  
at least one resource must be held in a non-shareable mode

**Hold and Wait**  
a process is currently holding at least one resource and requesting additional resources that are being held by other processes

**Circular Wait**  
there exists a set of processes waiting for resources held by others, forming a circular chain

**No Preemption**  
resources cannot be forcibly removed from the processes holding them



## Deadlock Recovery



Select a victim



Rollback

## Deadlock Prevention



each process requests resources in a strictly increasing order



a process that holds resources for too long can be rolled back



**Banker's Algorithm**  
A deadlock avoidance algorithm that simulates the allocation of resources

Step	Transaction A	Transaction B
1	UPDATE payments SET status = 'Done' WHERE order_id = 100	
2		UPDATE orders SET status = 'Paid' WHERE order_id = 100
3	SELECT* FROM orders WHERE order_id = 100	
4		SELECT* FROM payments WHERE order_id = 100

Revisão #: contagem de revisões  
Criado: duração de tempo por usuário  
Atualizado: duração de tempo por usuário