



# *Tuning* de Banco de Dados Relacionais

PPGCC – DC – UFSCar

CCO–620 – Modelagem e Projeto de Banco de Dados

Aluno: Cesar Joaquim Neto



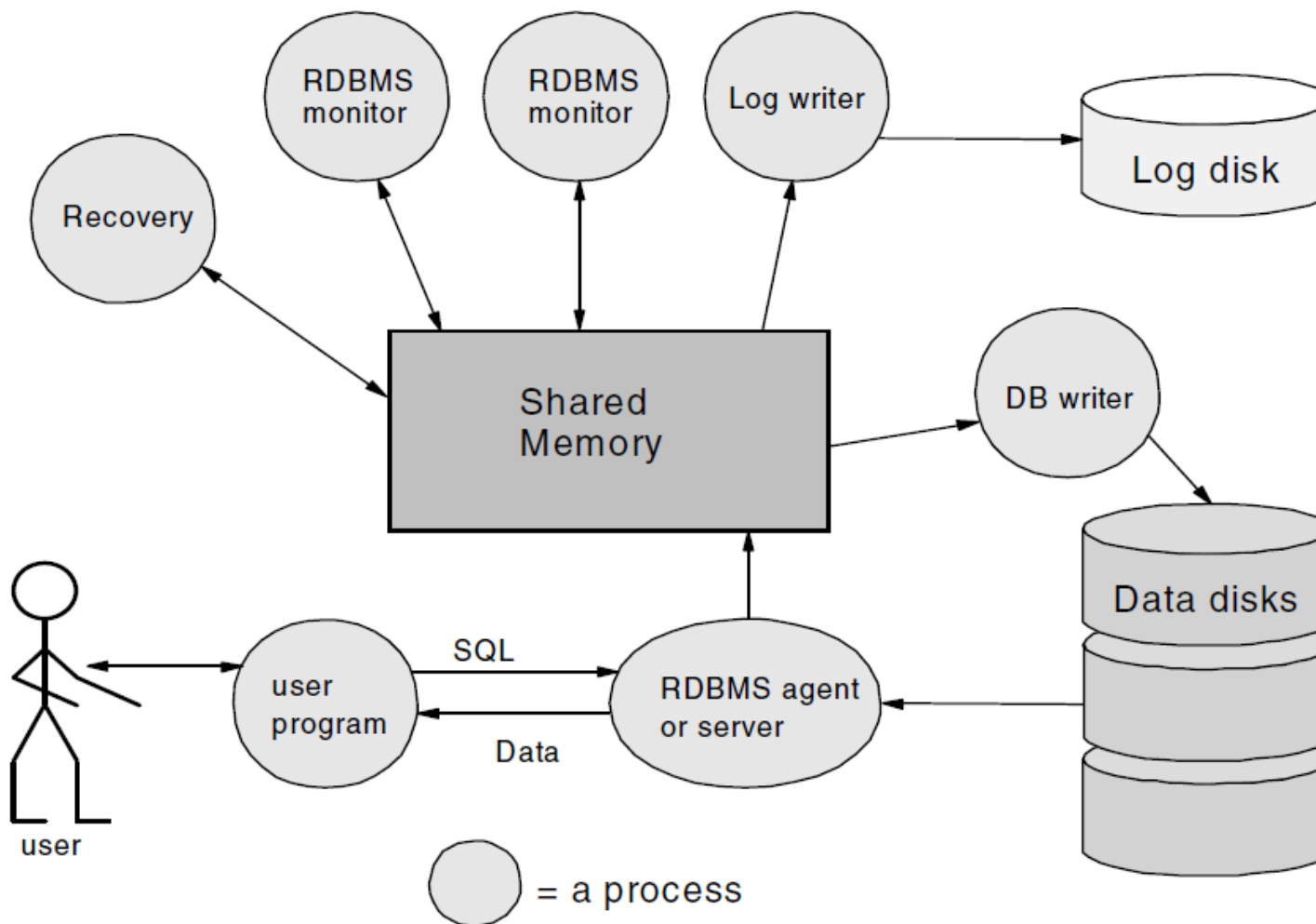


# Tópicos

- Organização Geral de um Banco de Dados Relacional ➡
- OLTP X OLAP ➡
- Gargalos no banco de dados ➡
- Manutenções de rotina ➡
- Melhores práticas nas queries ➡
- Referências ➡



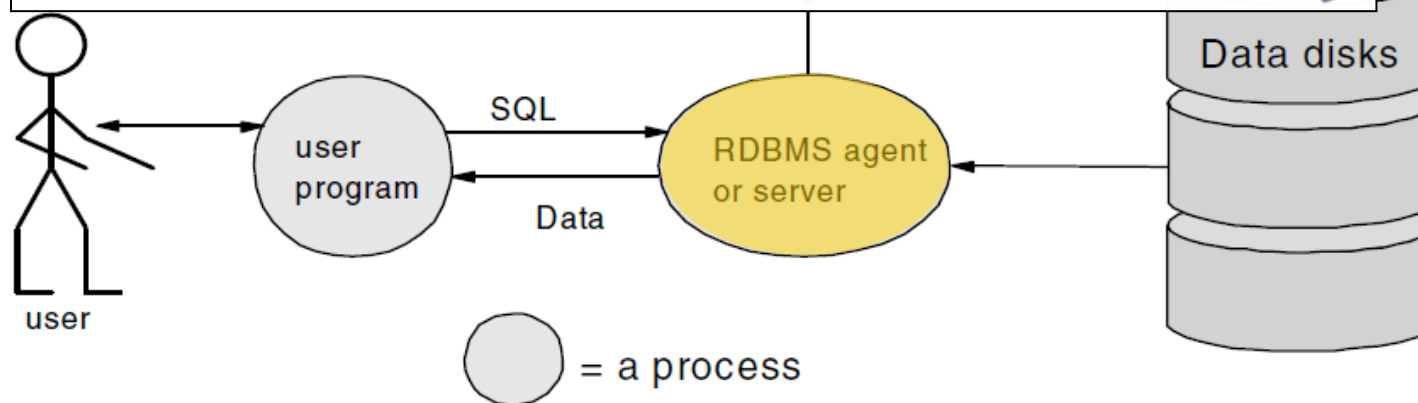
# Organização Geral de um RDBMS ↩



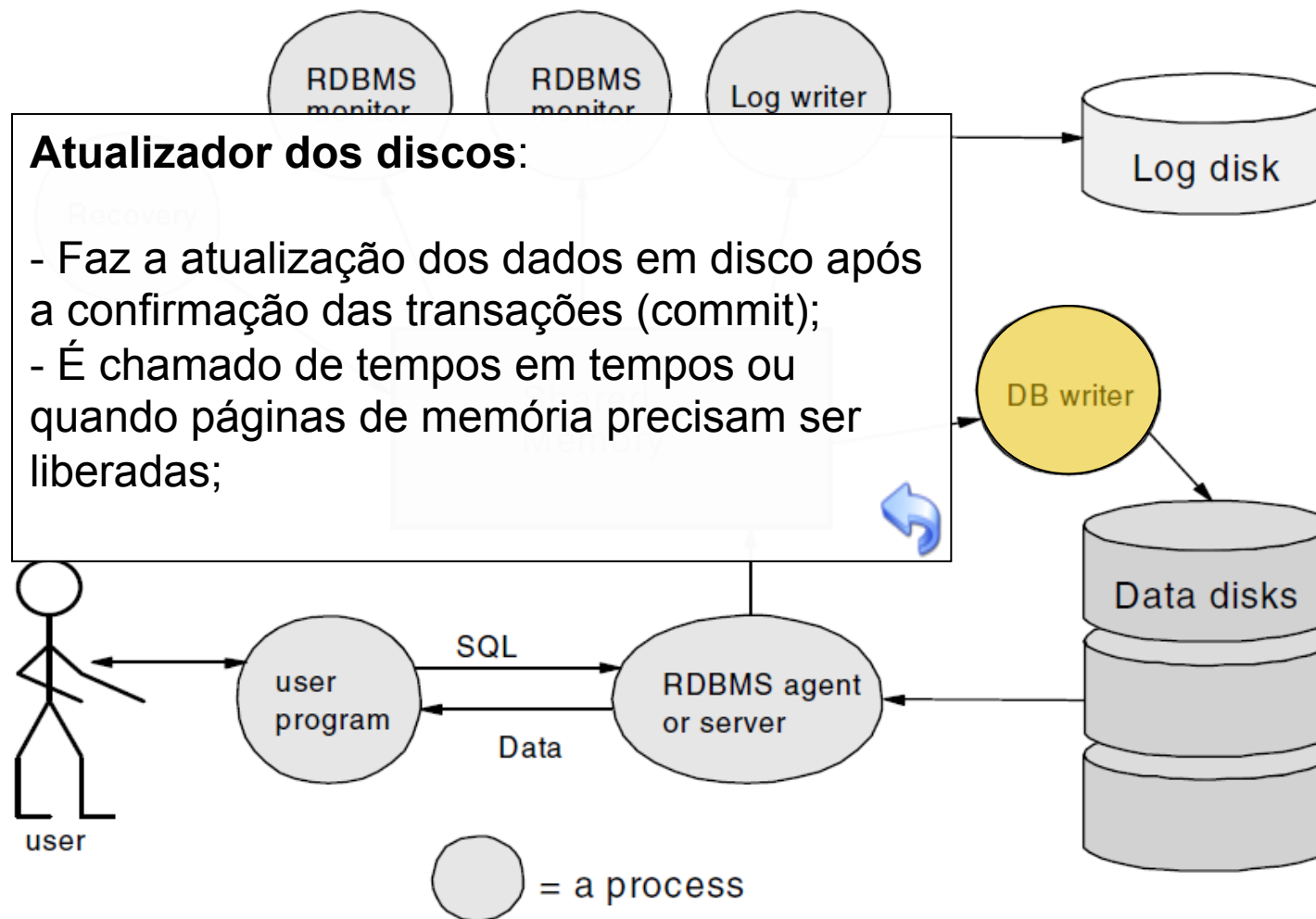
# Organização Geral de um RDBMS

## Processo agente ou servidor:

- Faz a interação entre o usuário e o RDBMS executando os comandos pedidos e retornando os resultados;
- Normalmente possui uma pequena área de memória própria (não mostrada na figura);
- Atende a conexões variadas e utilizando vários protocolos de comunicação (usualment IPC no Unix e TCP/IP para acesso remoto).



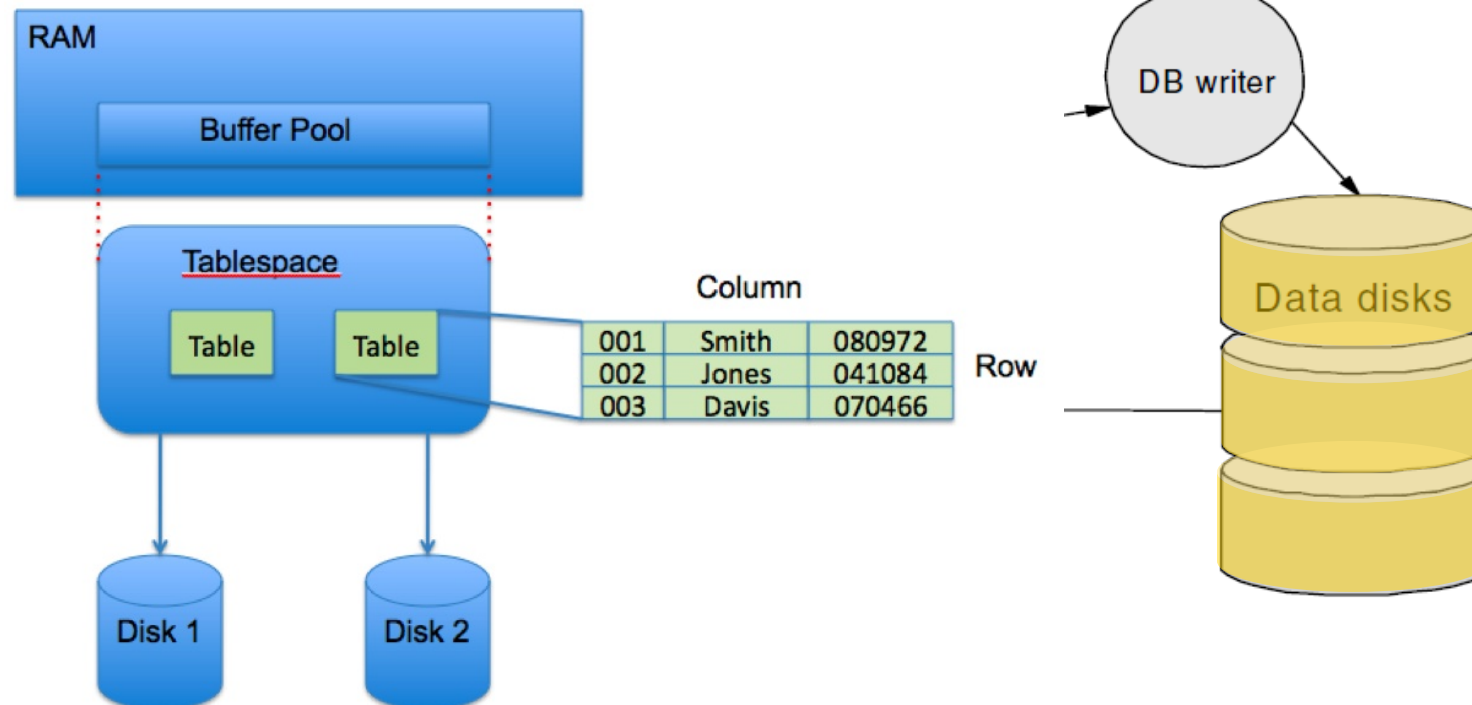
# Organização Geral de um RDBMS



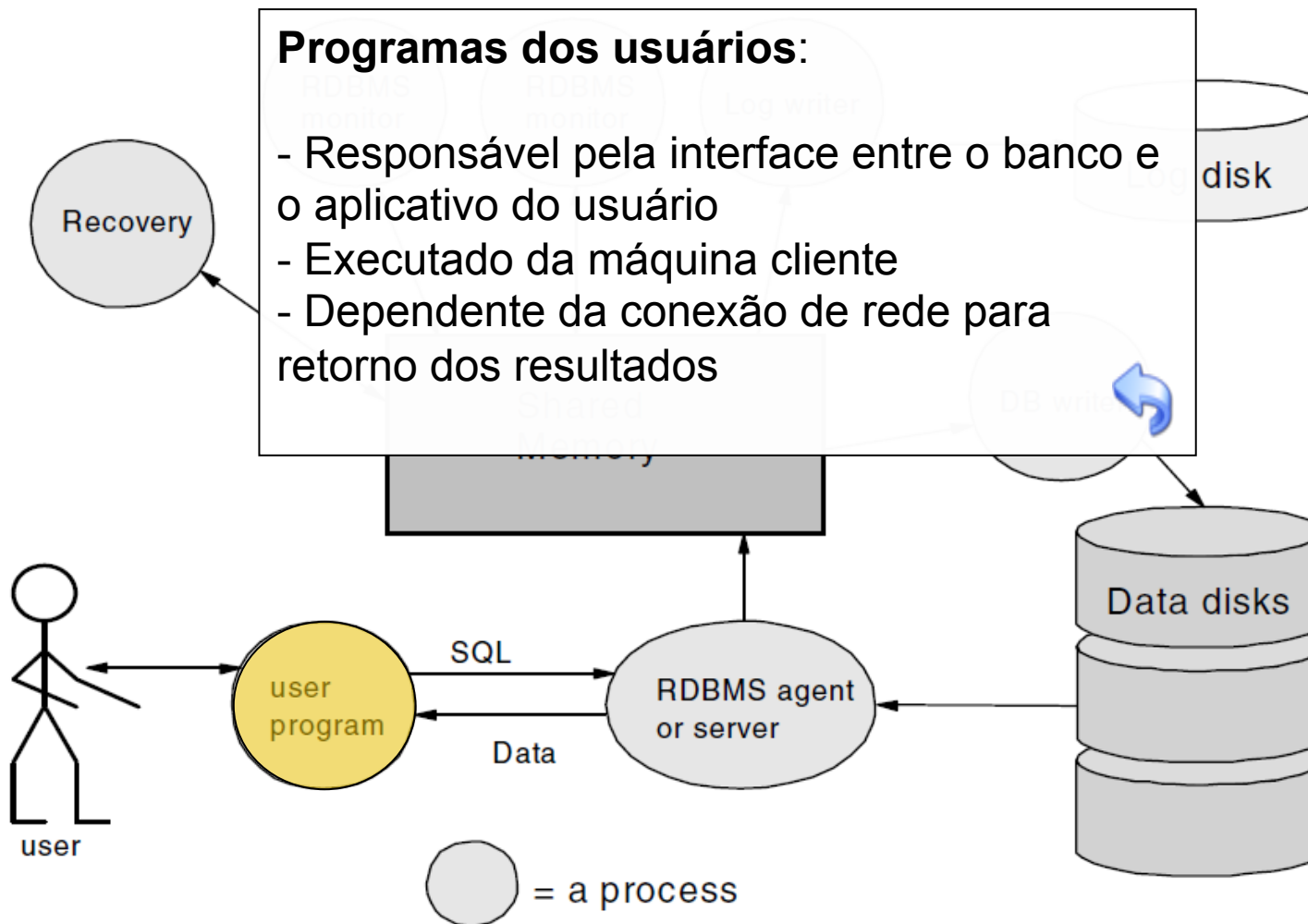
# Organização Geral de um RDBMS

## Organização dos arquivos:

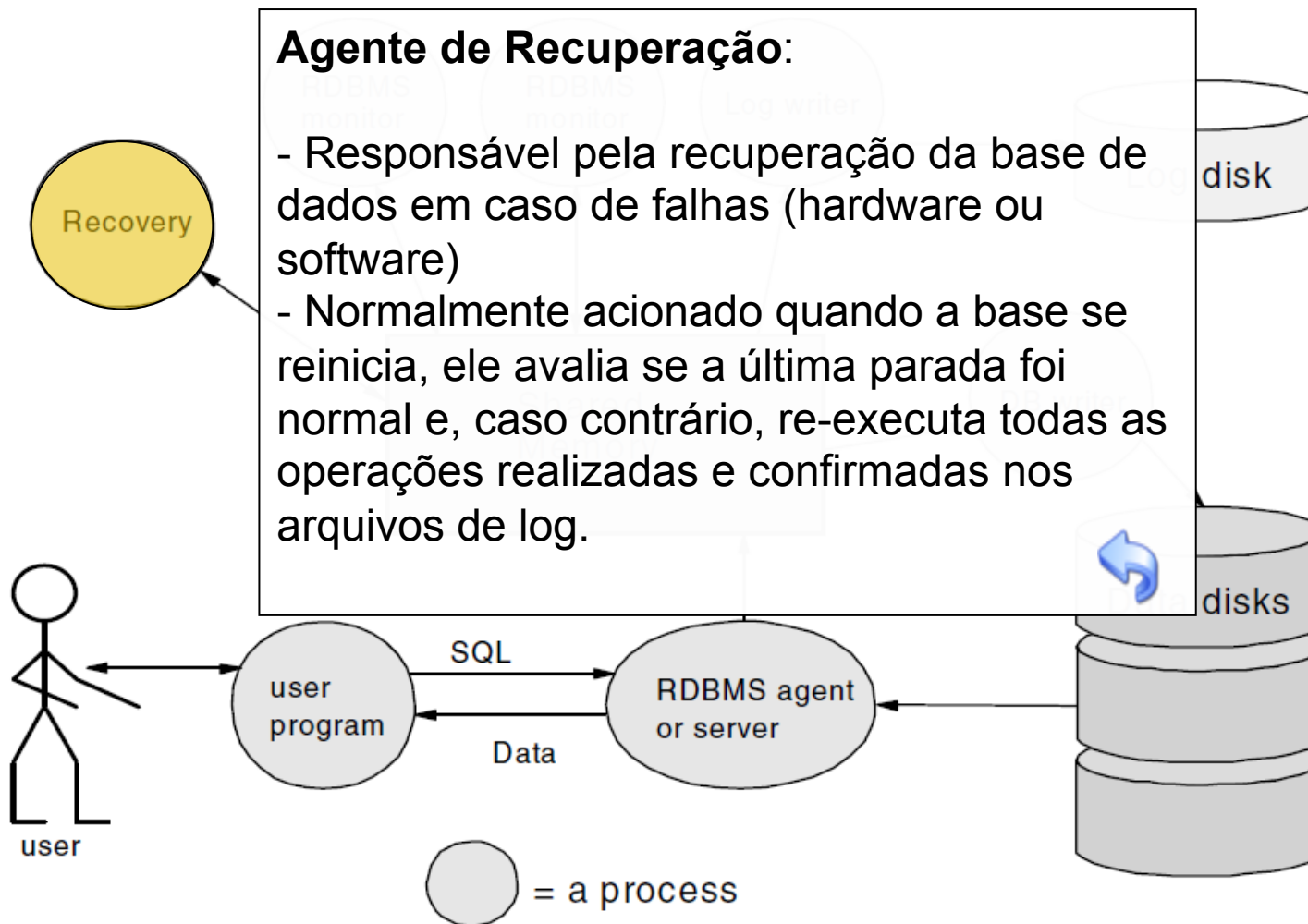
- Tablespaces / Tabelas / Registros
- Podem ser automáticas (SMS) ou manuais (DMS)
- Associadas a uma parte da memória para melhoria da performance



# Organização Geral de um RDBMS

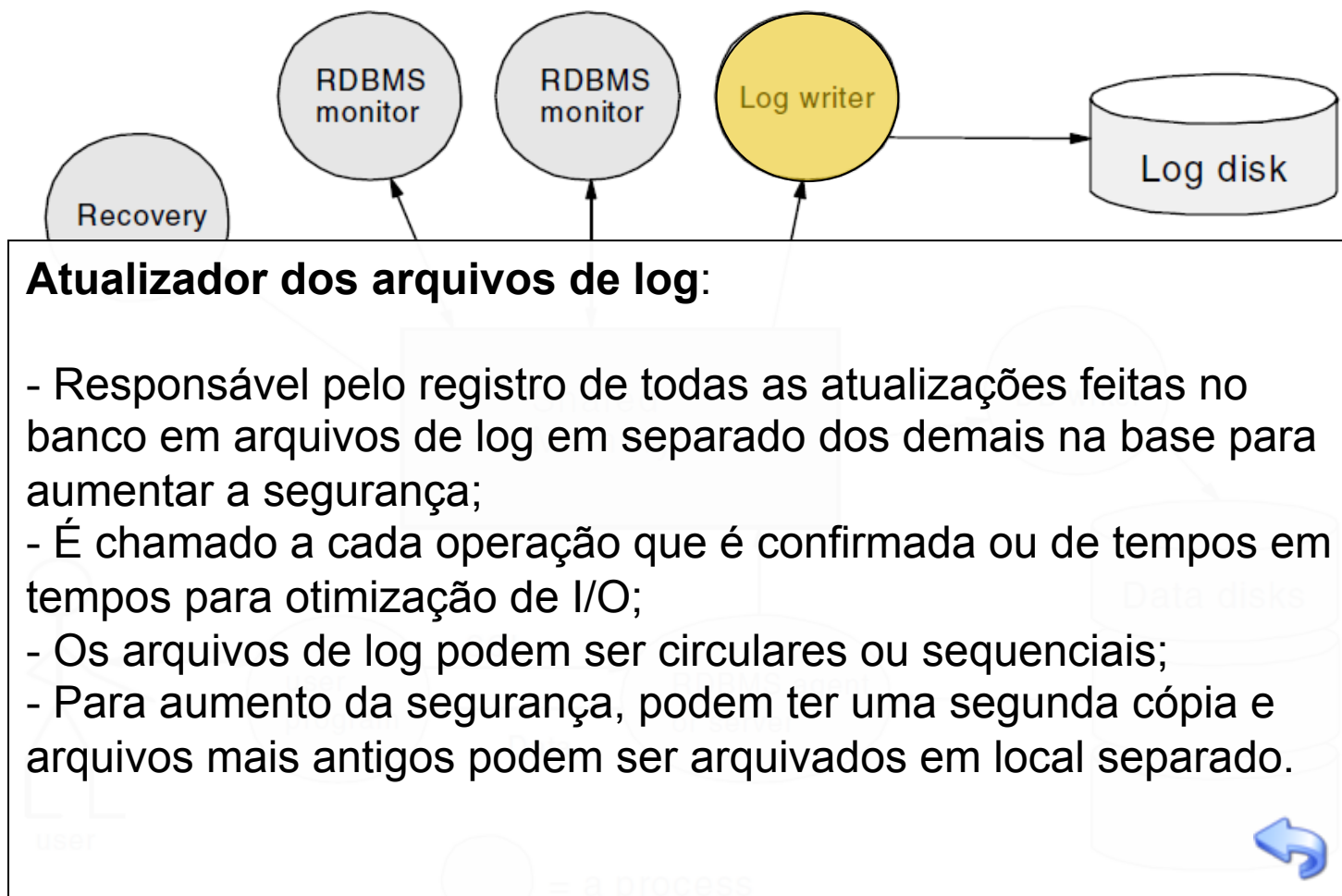


# Organização Geral de um RDBMS

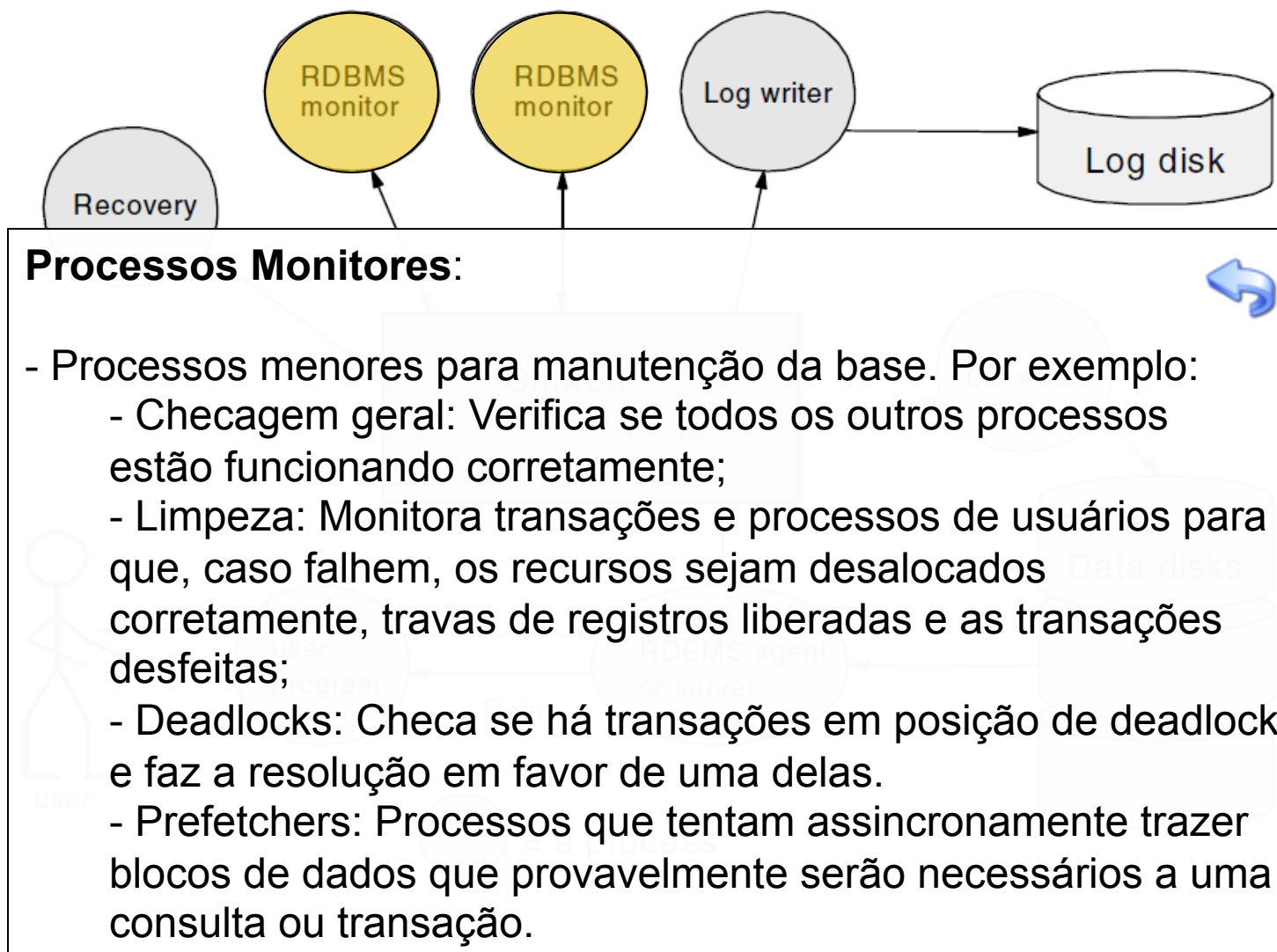




# Organização Geral de um RDBMS



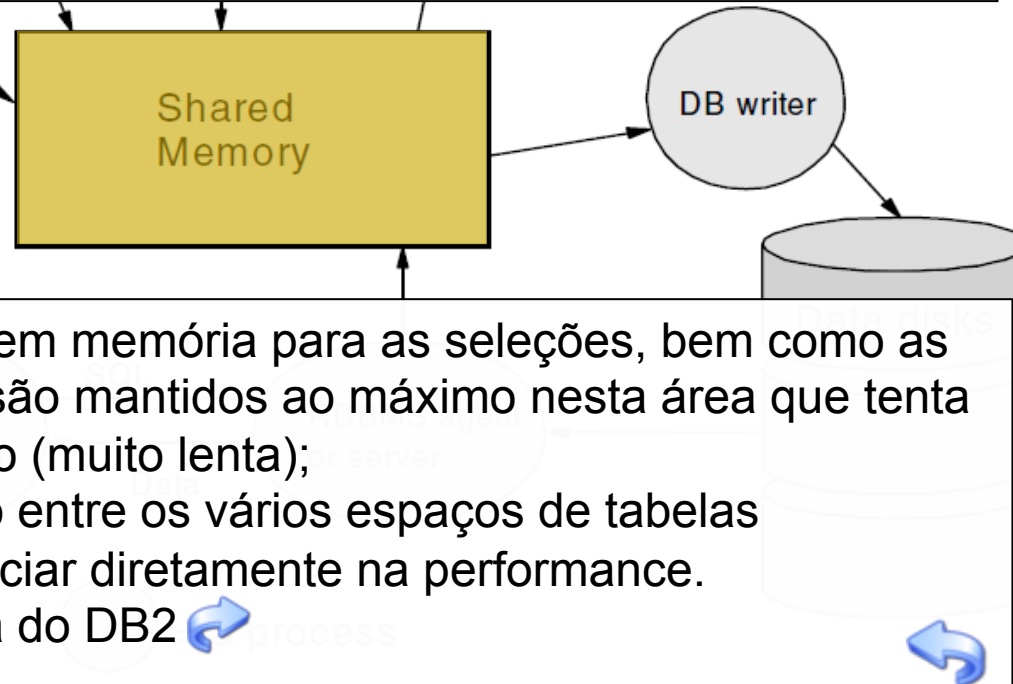
# Organização Geral de um RDBMS



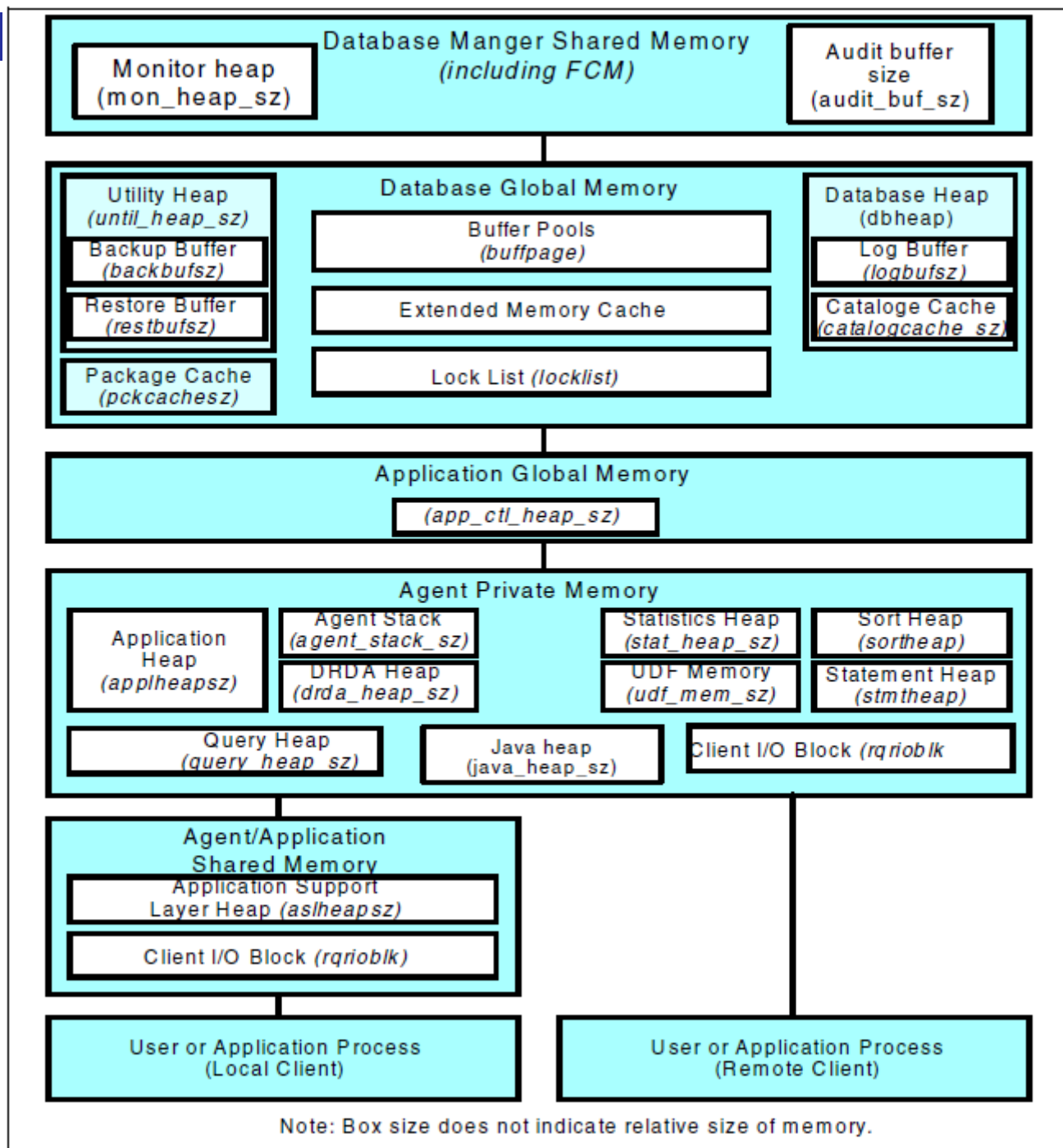
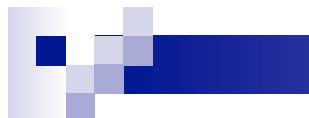
# Organização Geral de um RDBMS

## Memória principal:

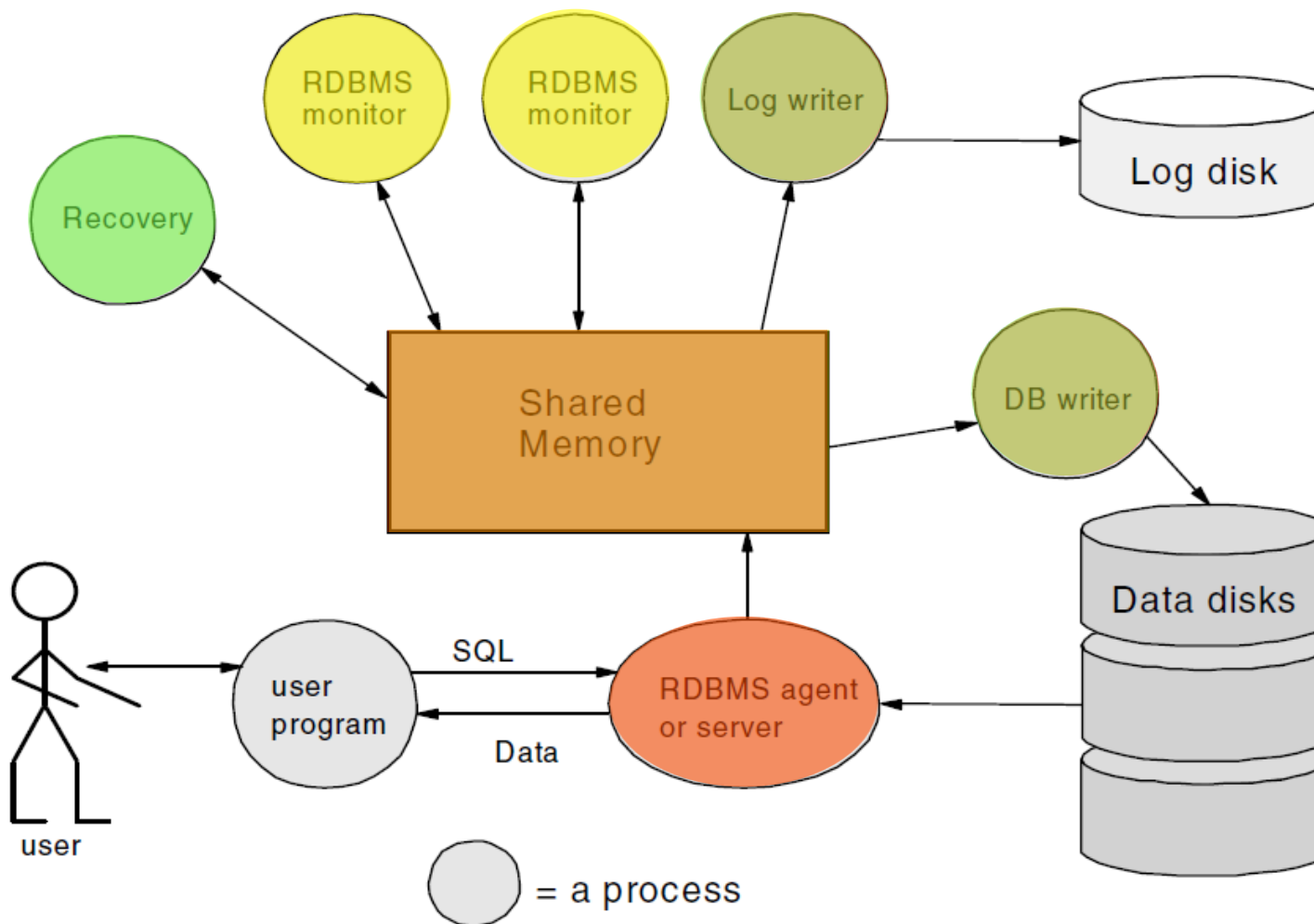
- Memória RAM usada pelo RDBMS para processar queries e manter o banco de dados funcionando com boa performance;
- Principais áreas alocadas: ordenação (sort), buffer de dados, informações de gerenciamento da instância e das bases, entre outras.



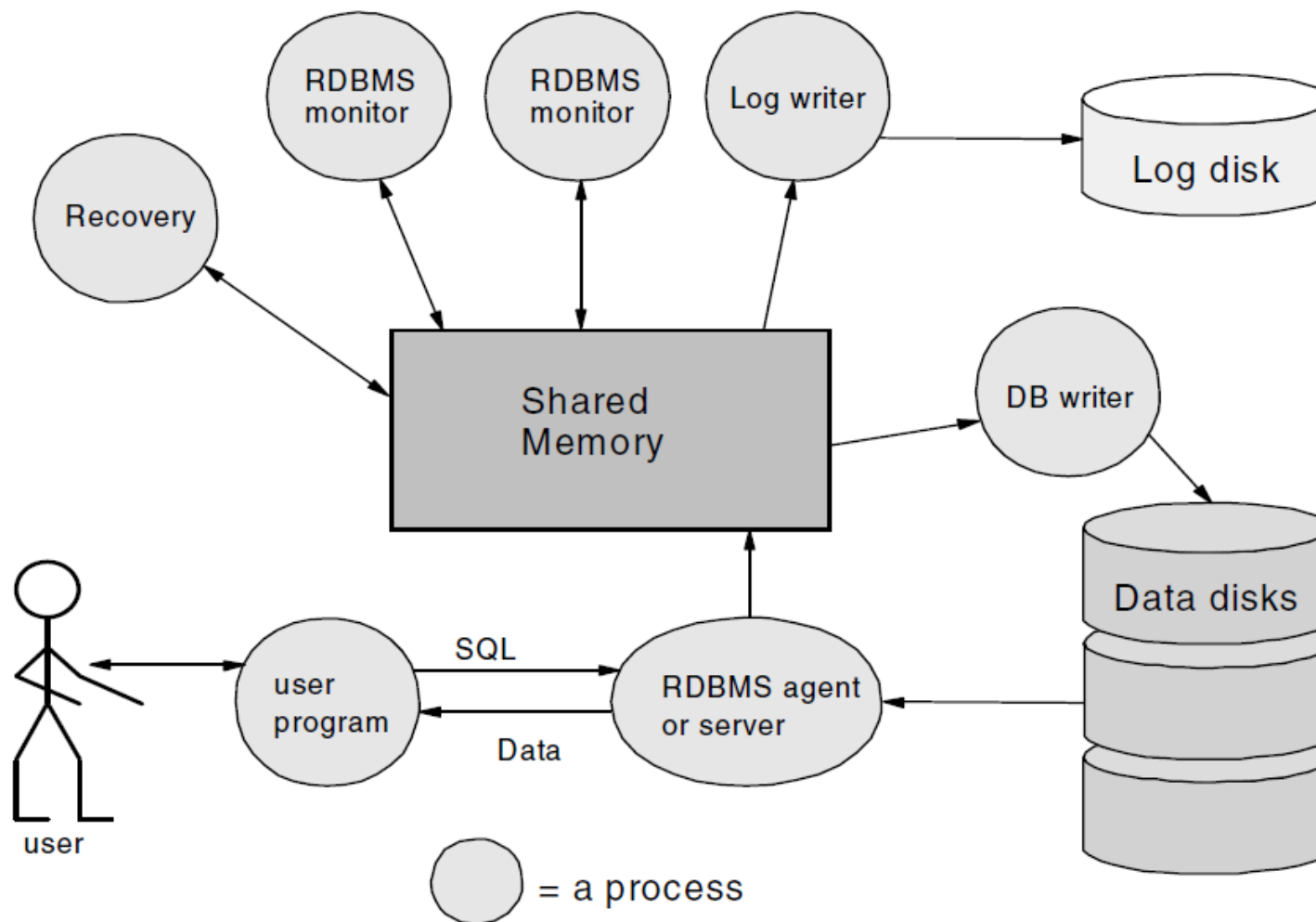
- Os dados carregados em memória para as seleções, bem como as transações realizadas, são mantidos ao máximo nesta área que tenta minimizar a E/S de disco (muito lenta);
- Distribuição de espaço entre os vários espaços de tabelas (tablespaces) irá influenciar diretamente na performance.
- Estruturas de memória do DB2



# OLTP X OLAP



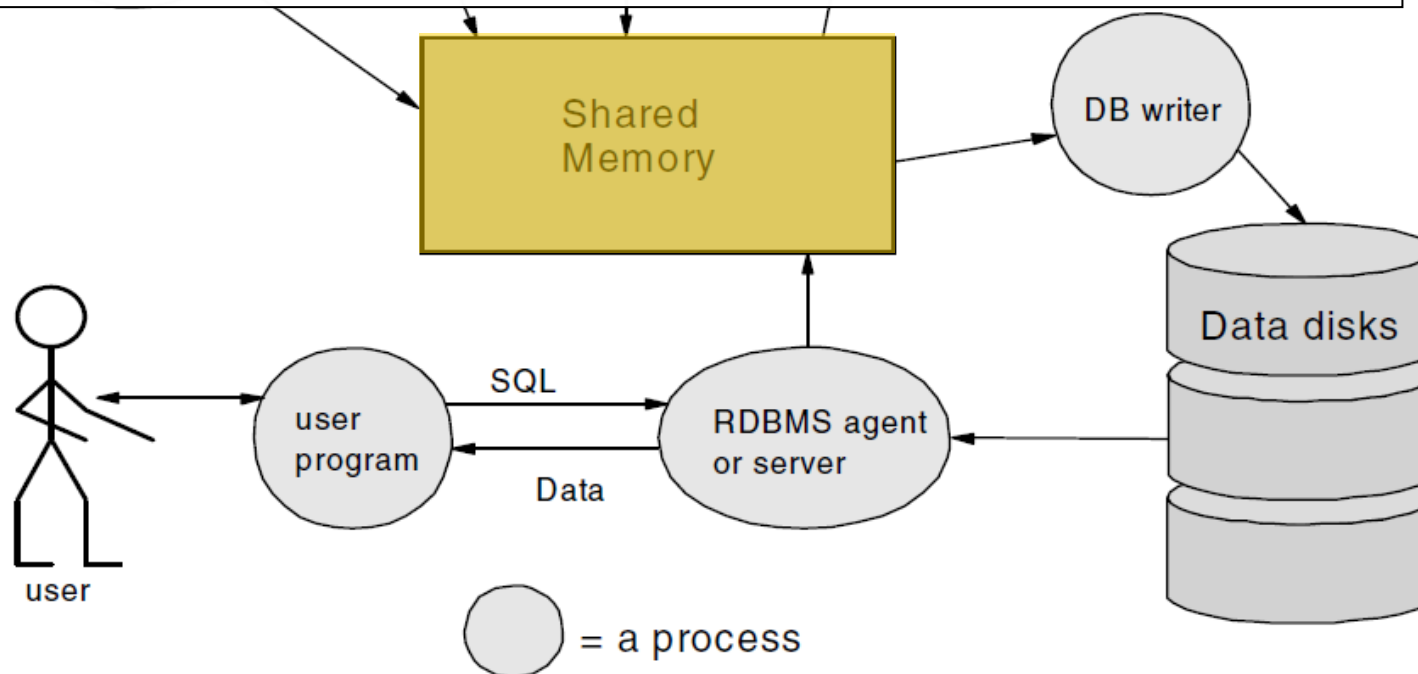
# Gargalos no Banco de Dados



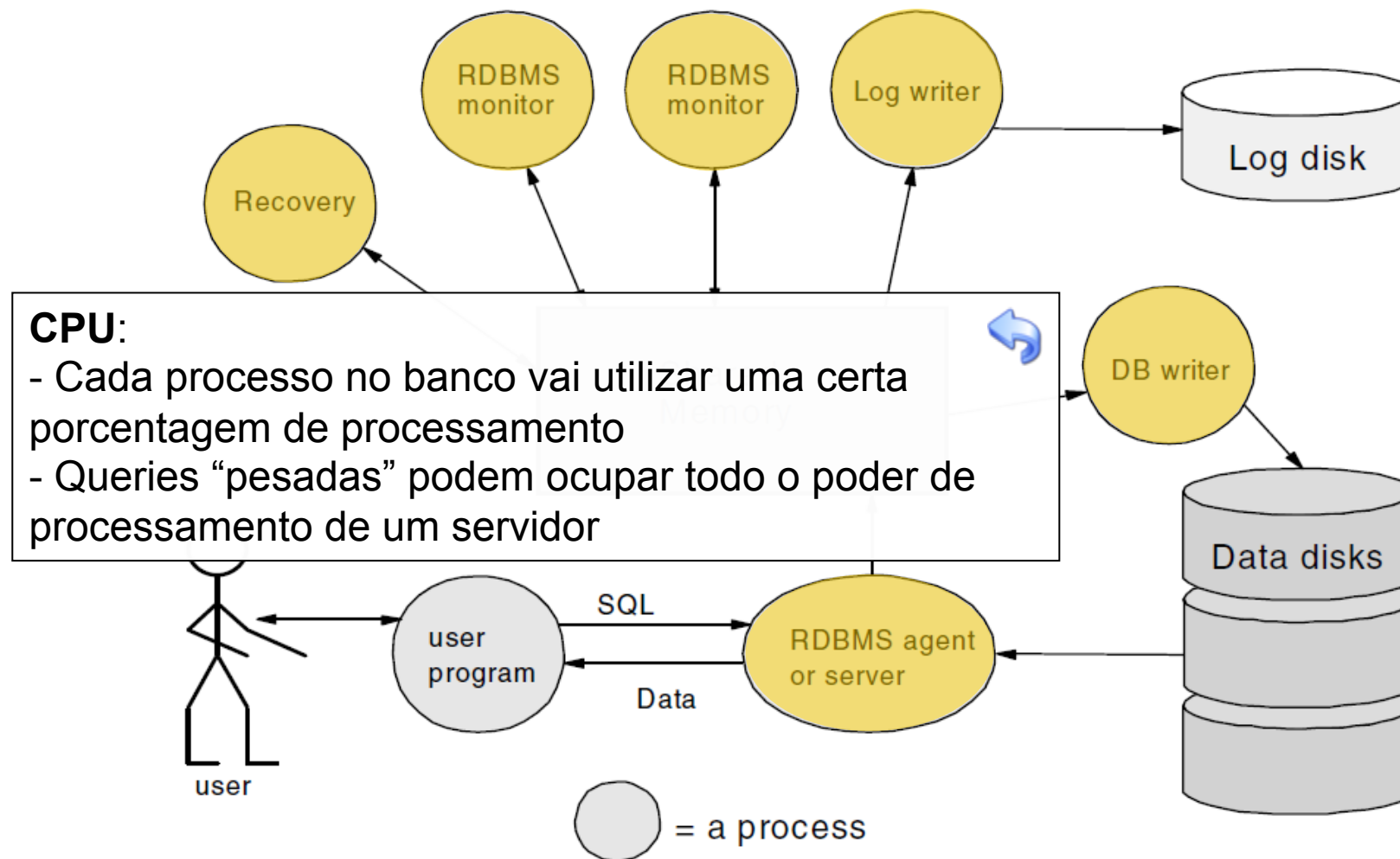
# Gargalos no Banco de Dados

## Memória:

- Cada tablespace é associada a uma área de buffer de dados
- Operações de ordenação (sort), *hash joins*, informações de gerenciamento da instância e das bases, entre outras.
- Na falta de espaço, o banco de dados vai buscar recursos na E/S de disco fazendo “despencar” a performance.

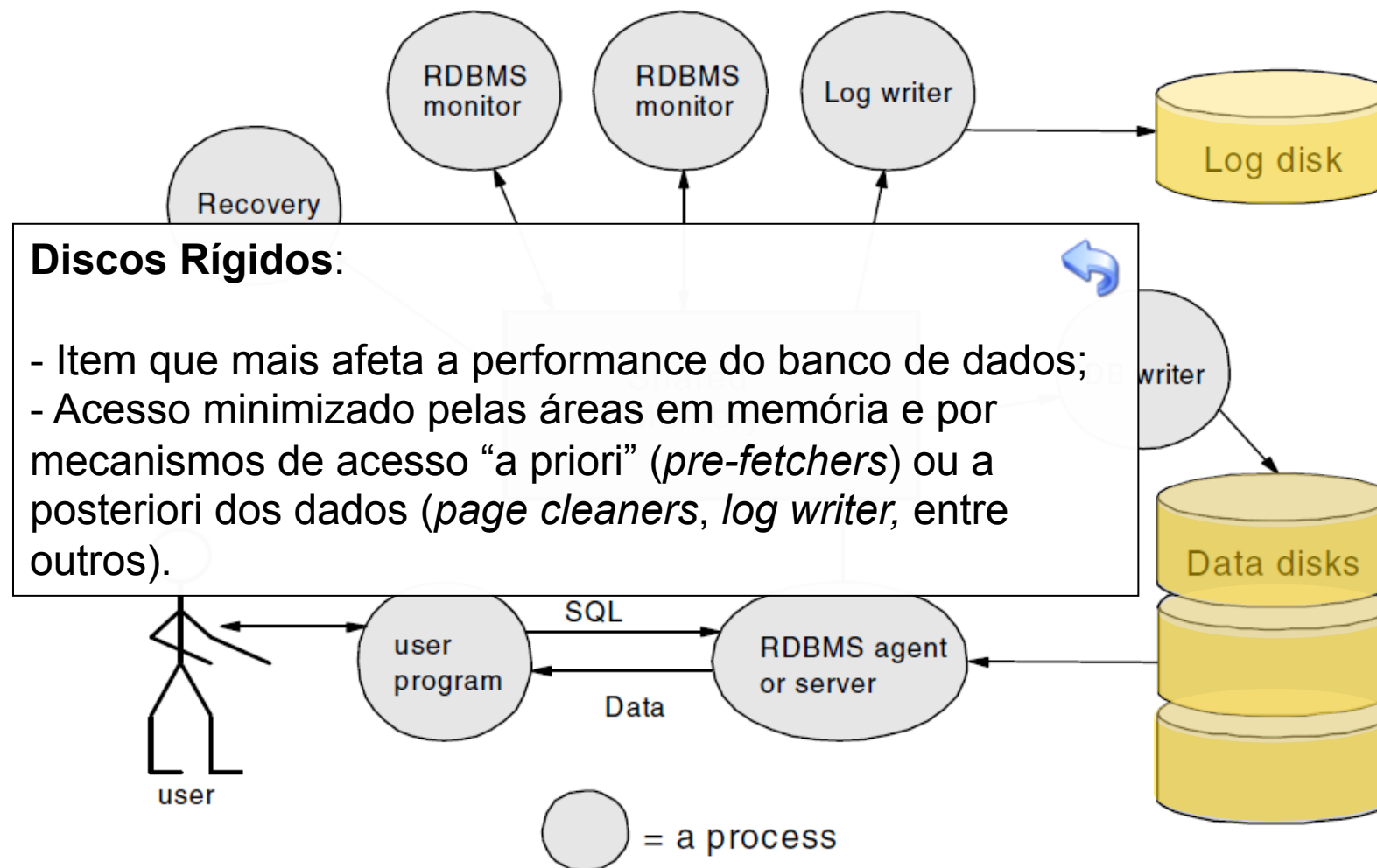


# Gargalos no Banco de Dados

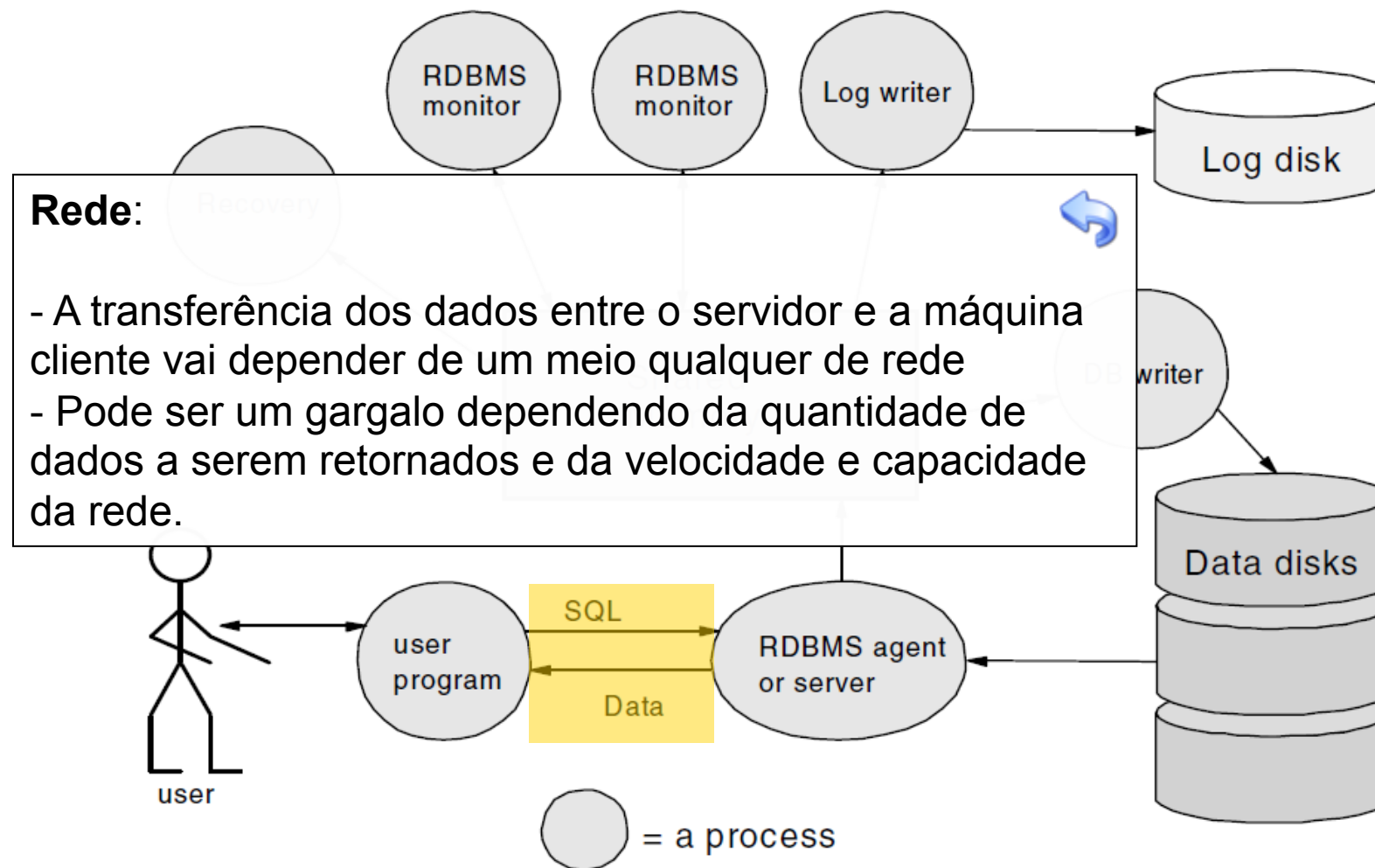




# Gargalos no Banco de Dados



# Gargalos no Banco de Dados



# Gargalos no Banco de Dados ➡

- Sistemas mal balanceados

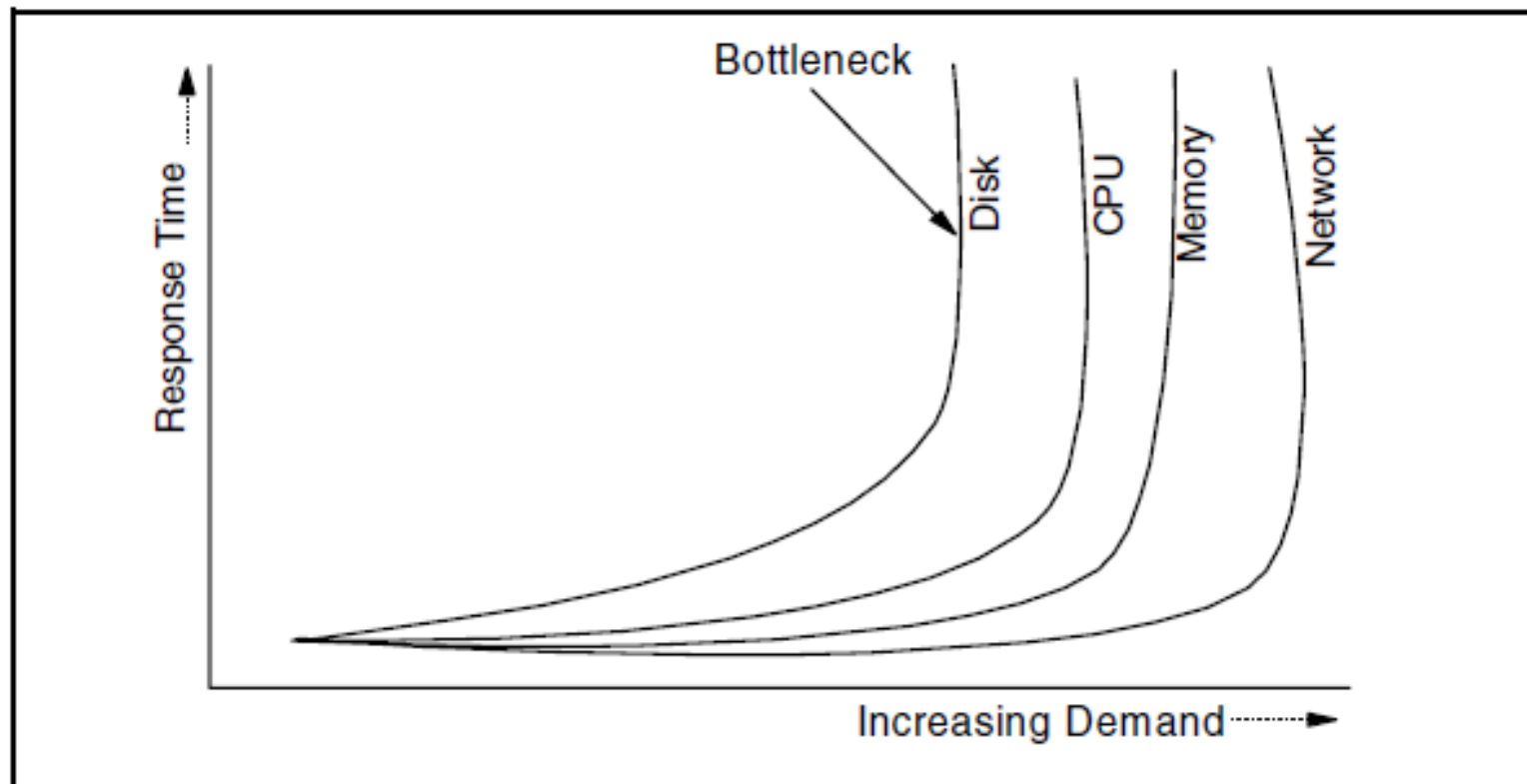


Figure 12-3 Poorly tuned means one bottleneck slows the system



# Gargalos no Banco de Dados ➡

## ■ Sistemas balanceados

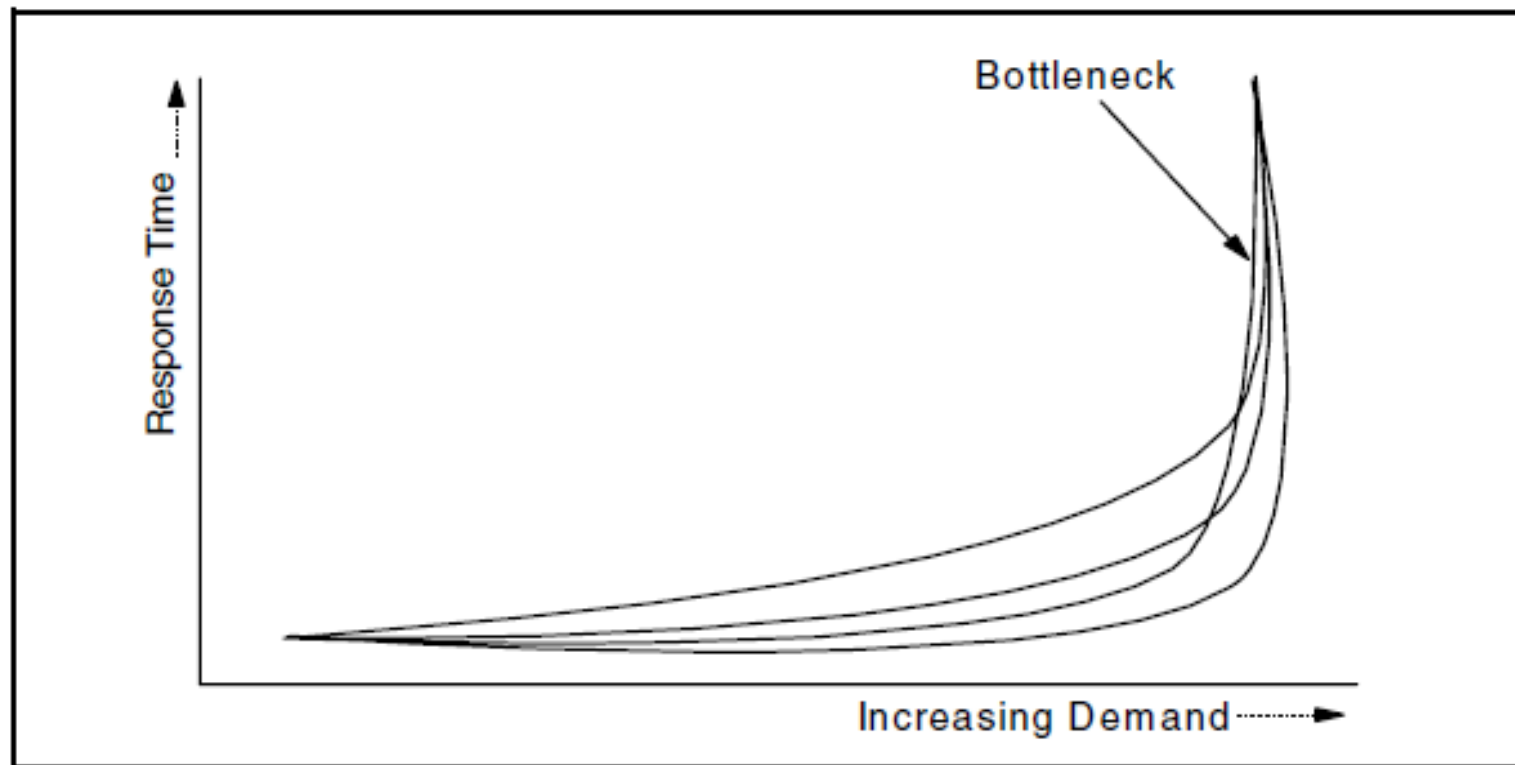


Figure 12-4 Well balanced systems postpone the bottleneck



# Gargalos no Banco de Dados ➡

## ■ Simulações

- ☐ **Memória**
- ☐ CPU
- ☐ Disco Rígido
- ☐ Rede

Estratégia:

- Diminuir as áreas em memória que servem as tablespaces;
- Diminuir as áreas de memória compartilhadas no banco

Verificação:

- Buffer Pool Hit Ratio
- Sort overflow
- Hash Join overflow





# Gargalos no Banco de Dados ➡

## ■ Simulações

- ☐ Memória
- ☐ **CPU**
- ☐ Disco Rígido
- ☐ Rede

Estratégia:

- Executar queries e transações de atualização dos dados ao mesmo tempo

Verificação:

- CPU execution time  
- Buffer Pool Hit Ratio perto de 100% sempre



# Gargalos no Banco de Dados

## ■ Simulações

- ☐ Memória
- ☐ CPU
- ☐ **Disco Rígido**
- ☐ Rede

### Estratégia:

- Aumentar prefetch size e número de prefetchers
- Aumentar número de page cleaners
- Diminuir tempo de espera entre limpezas
- Executar queries e transações de atualização dos dados ao mesmo tempo

### Verificação:

- Time waited for prefetch
- Buffer Pool physical reads



# Gargalos no Banco de Dados

## ■ Simulações

- ☐ Memória
- ☐ CPU
- ☐ Disco Rígido
- ☐ **Rede**

Estratégia:

- Executar a query remotamente de uma conexão precária como wireless (abaixo de alguns Kbps) ou celular

Verificação:

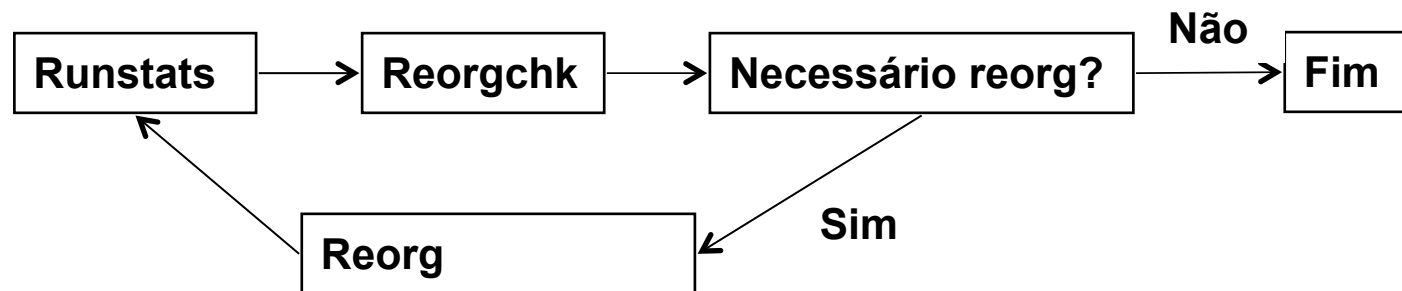
- Demora para transmissão dos dados enquanto a query já terminou de executar no banco de dados





# Manutenções de rotina ↻

- Coleta de estatísticas (runstats)
- Reorganização de tabelas (reorg)



# Melhores práticas nas queries

- Quebra de índices
  - ☐ Where rtrim(a.campo1)=rtrim(b.campo2)
- Seleções sem predicados - utilizar a cláusula "fetch first <n> rows only"
  - ☐ Select \* from employees fetch first 5 rows only
- Evitar seleção de colunas que não serão usadas
- Evitar select count(\*)
  - ☐ O "\*" faz com que o DB2 trabalhe na área temporária com todas as colunas de uma tabela (ou junção de tabelas) sendo que não vai utilizá-los depois;
  - ☐ Usar "select count(1)..."
- Table scans X Índices
  - ☐ Se ao fazer o explain de uma query você notar que há muitos table scans e, principalmente, nas suas maiores tabelas, verificar se não há possibilidade de colocar um índice para a junção da mesma com outras tabelas.

## Referências

- IBM. DB2 Database for Linux, UNIX, and Windows Information Center. Disponível em: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>. Acessado em 30/05/2011.
- DARMAWAN, B.; GROENEWALD, G.; IRVING, A.; MONTEIRO, S. H. S.; SNEDEKER, K. M. Database Performance Tuning on AIX. Segunda edição. IBM Redbooks. Código sg245511.
- IBM DEVELOPER WORKS. Best practices for tuning DB2 UDB v8.1 and its databases. Disponível em: <http://www.ibm.com/developerworks/data/library/techarticle/dm-0404mcarthur/index.html>. Acessado em 29/05/2011.

# Dúvidas?



*Cesar Joaquim Neto*  
**Database Administrator**  
**IBM Certified Database Administrator—DB2 9 LUW**  
**IBM Global Business Services**  
**[cjneto@yahoo.com](mailto:cjneto@yahoo.com)**