

BANCO DE DADOS E GESTÃO DA INFORMAÇÃO

Administração de Banco de Dados



SUMÁRIO

Administração de Banco de Dados	4
1. Técnicas de Armazenamento	4
Hierarquia de Dispositivos de Armazenamento	7
Storages	9
DAS ou Direct Attached Storage	9
NAS ou Network Attached Storage	10
SAN ou Storage Area Network	11
RAID	11
Acesso ao Bloco de Disco	17
Journaling	17
2. Organização de Arquivos	19
Registros de Tamanho Fixo e Variável	20
Alocação de Espaço em Disco	20
Cabeçalhos de Arquivo	22
Operações em Arquivos	22
Arquivos de Registros Desordenados (Arquivos Heap)	22
Arquivos de Registros Ordenados (Arquivos Classificados)	23
Técnicas de Hashing	24
3. Métodos de Acesso	25
Organização Sequencial	26
Organização Serial	27
Organização Indexada	28
Organização em Campos	29
Organização em Registros	29
4. Tipos de Banco de Dados	30
5. Bancos de Dados NoSQL	33
Bancos de Dados Colunares	33

Bancos de Dados Orientados a Grafos	34
Bancos de Dados Orientados a Documentos	35
Bancos de Dados do Tipo Chave-Valor	36
6. Arquitetura de Banco de Dados	36
Arquitetura Centralizada	37
Arquitetura Cliente-Servidor	37
Arquitetura Paralela	39
Arquitetura Distribuída	40
Resumo	42
Questões de Concurso	48
Gabarito	56
Gabarito Comentado	57

ADMINISTRAÇÃO DE BANCO DE DADOS

1. TÉCNICAS DE ARMAZENAMENTO

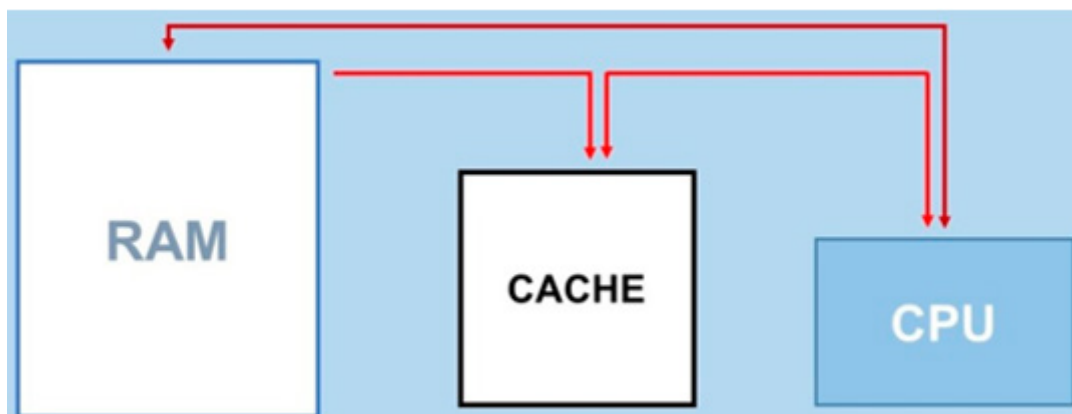
Para entender sobre **técnicas de armazenamento**, serão abordados conceitos de **armazenamento de dados**.

Obs.: **Armazenamento de dados** é a **coleta e a retenção de informações digitais** – os *bits* e *bytes* das aplicações, protocolos de rede, documentos, mídias, catálogos de endereços, preferências dos usuários e muito mais; no nosso caso, os **bancos de dados**.

Existem vários tipos de **armazenamento de dados** na maioria dos sistemas de computador. Esses **meios de armazenamento** são classificados pela velocidade com que os dados podem ser acessados, pelo custo por unidade de dados e por sua confiabilidade.

Entre os **meios de armazenamento de dados** disponíveis, podemos destacar:

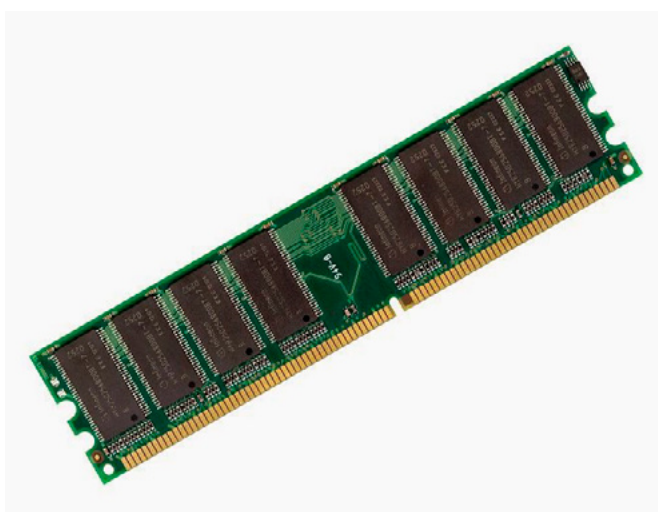
- **Memória Cache:** É a forma de armazenamento mais rápida e dispendiosa. A **memória cache** é **pequena** e seu uso é gerenciado pelo *hardware* do sistema de computador. A **memória cache** é uma pequena quantidade de memória **localizada perto do processador**. Surgiu quando a **memória RAM** não estava mais acompanhando o desenvolvimento do processador, atuando como **armazenamento intermediário entre os registradores do processador e a memória RAM**.



- **Memória Principal:** A **memória** trata de todos os dispositivos que permitem a um computador **guardar dados**, de forma **temporária** ou **permanente**. O termo **memória** é utilizado para designar componentes de um sistema capazes de **armazenar dados e programas**. A **memória principal** é um **depósito temporário de dados e instruções** dos programas de computador. A **memória principal** do computador é **volátil**, e seu tamanho é limitado pelo custo do *hardware*. Assim, os usuários necessitam de algum método para armazenar e recuperar informações de modo permanente (a **memória secundária**). **As instruções de máquina de uso geral operam sobre a memória principal**.

DICA

Embora a **memória principal** possa conter muitos *megabytes* de dados em um PC pessoal ou centenas de *gigabytes* de dados em grandes sistemas de servidor, ela geralmente é muito pequena (ou muito cara) para armazenar o banco de dados inteiro. A **memória principal** do computador, por ser **volátil**, precisa ser atualizada com dados e instruções cada vez que o computador é ligado.



- **Memória Flash:** A **memória flash** difere da **memória principal** porque **os dados armazenados sobrevivem à falta de energia** (ou a falhas), ou seja, trata de um **armazenamento não volátil**, o que significa que não precisa de energia para manter as informações armazenadas no *chip*. Além disso, a **memória flash** oferece um tempo de acesso rápido, embora não tão rápido como a **memória principal** (ou memória RAM), e melhor resistência do que discos rígidos. Uma limitação é que a **memória flash** tem um **número finito de modificações** (escrita/exclusão), ou seja, de operações de entrada e saída de dados (IOPS). Há dois tipos de **memória flash**, denominados **flash NAND** e **flash NOR**. A memória **flash NAND** propicia uma capacidade de armazenamento muito mais alta em relação a determinado custo, sendo muito utilizada para o armazenamento de dados em dispositivos como câmeras, *players* de música, celulares e *laptops*. A **memória flash** também é muito utilizada como **substituição dos discos magnéticos** para armazenar quantidade moderada de dados. Essas unidades de **memória flash** que substituem unidades de disco são chamadas de **unidades em estado sólido**, ou **SSD**.



- **Disco Magnético:** O principal meio para **armazenamento de dados a longo prazo** é o **disco magnético**. Normalmente, um banco de dados inteiro é armazenado em **disco magnético**. O sistema precisa **mover os dados do disco para a memória principal**, de modo que **possam ser acessados**. Depois que o sistema tiver realizado as operações designadas, **os dados que foram modificados precisam ser gravados em disco**. O **armazenamento em disco** sobrevive a faltas de energia e falhas do sistema. Os próprios dispositivos de **armazenamento em disco** às vezes podem falhar e, portanto, destruir dados, mas essas falhas normalmente ocorrem com muito menos frequência do que as falhas do sistema.



O PULO DO GATO

O disco rígido é uma **memória não volátil**, ou seja, as informações não são perdidas quando o computador é desligado, sendo considerado o principal meio de **armazenamento de dados em massa**. Por ser uma **memória não volátil**, serve para executar programas já rodados ou carregar arquivos contendo dados inseridos anteriormente.



- **Armazenamento Óptico:** As formas mais populares de **armazenamento óptico** são o **compact disk** (CD), que pode manter cerca de 700 *megabytes* de dados e possui um tempo de execução de cerca de 80 minutos; e o **digital video disk** (DVD), que pode manter 4,7 ou 8,5 *gigabytes* de dados por lado do disco (ou até 17 *gigabytes* em um disco com dois lados). Os dados são armazenados de forma óptica em um disco e lidos por um *laser*.

Os dispositivos de **armazenamento óptico** são os mais utilizados para o armazenamento de **informações multimídia**, sendo amplamente aplicados no armazenamento de filmes, música etc. Apesar disso, também são muito utilizados para o **armazenamento de informações e programas**, especialmente para instalação em um computador.



- **Armazenamento em Fita:** É usado principalmente para **backup** e **arquivamento de dados**. Embora a **fita magnética** seja mais barata do que os discos, o **acesso aos dados é muito mais lento**, pois a fita precisa ser **acessada sequencialmente desde o início**. Por esse motivo, o **armazenamento em fita** é chamado de **armazenamento por acesso sequencial**. Por sua vez, o **armazenamento em disco** é chamado de **armazenamento por acesso direto**, pois é possível ler dados em qualquer local no disco. As fitas possuem alta capacidade (atualmente, existem fitas de 40-300 *gigabytes*) e podem ser removidas da unidade de fita, de modo que são bastante adequadas para o arquivamento com baixo custo.

HIERARQUIA DE DISPOSITIVOS DE ARMAZENAMENTO

Os diversos meios de armazenamento podem ser organizados em uma **hierarquia**, conforme mostrado na figura a seguir, de acordo com sua velocidade e seu custo.

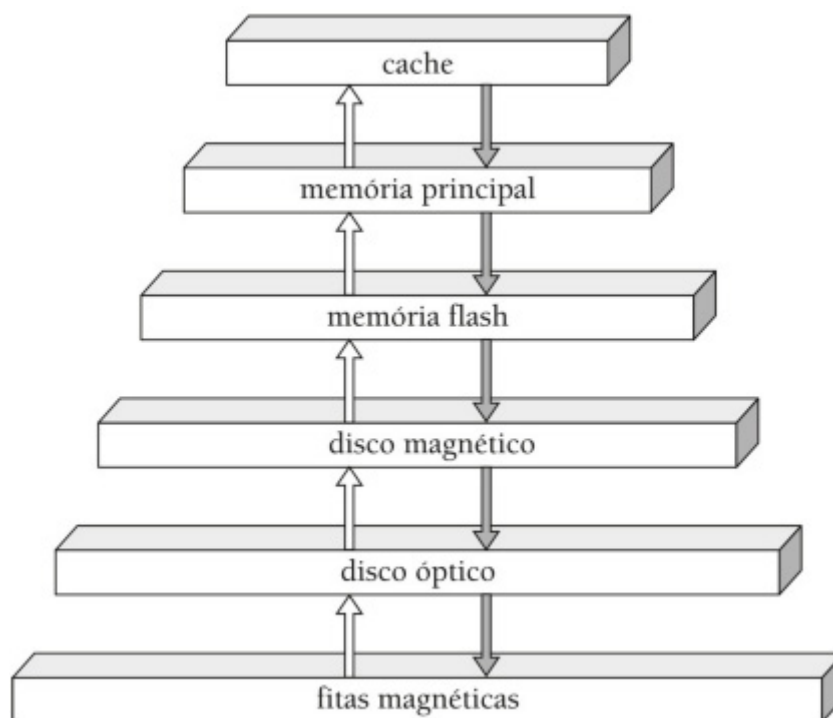


Figura 1 – Hierarquia de Dispositivos de Armazenamento

DICA

Os níveis mais altos são caros, mas são rápidos. Ao **descer-mos na hierarquia**, o **custo por bit diminui**, enquanto o **tempo de acesso aumenta**. Essa compensação (*trade-off*) é razoável; se determinado sistema de armazenamento fosse mais rápido e mais barato do que outro – outras propriedades sendo iguais –, não haveria motivo para usar a memória mais lenta, mais dispendiosa.

Os meios de armazenamento mais rápidos – por exemplo, **cache** e **memória principal** – são considerados **armazenamento primário**. Os meios no próximo nível da **hierarquia** – por exemplo, **discos magnéticos** – são considerados **armazenamento secundário**. Os níveis mais baixos da **hierarquia** – por exemplo, **fita magnética** e **discos ópticos** – são considerados **armazenamento terciário**.

Além da velocidade e do custo dos diversos sistemas de armazenamento, há também a questão do **armazenamento volátil**. O **armazenamento volátil** perde seu conteúdo quando a energia ao dispositivo é removida. Na hierarquia mostrada na figura anterior, os sistemas de armazenamento da memória principal para cima são **voláteis**, enquanto os sistemas de armazenamento abaixo da memória principal são **não voláteis**. Os dados precisam ser gravados no armazenamento não volátil por segurança.

STORAGES

O **storage** é um repositório onde **os dados da rede local de sua empresa são centralizados**, mas que também pode assumir outras funções e servir, por exemplo, como servidor de arquivos, *backup*, área de compartilhamento e colaboração — tudo que envolve a administração e o processamento de dados armazenados.

DICA

O termo **storage** significa **armazenamento**. É geralmente utilizado para **designar sistemas de armazenamento de dados** em servidores, computadores e eletrodomésticos. Um **storage** também pode ser traduzido como uma **área de armazenamento para pequenos objetos** ou um **espaço dentro de um armazém**.

Esses sistemas de armazenamento de grande porte são **divididos em três grandes grupos: DAS, NAS e SAN**.

Todos estes sistemas corporativos contam geralmente com tecnologias como a **montagem de arranjos RAID**, possibilitando o **agrupamento de vários discos** para maior proteção contra a perda de dados. Essa tecnologia proporciona maior tolerância a falhas e melhora a performance do global da solução. Existem diversos **níveis de arranjos de discos**, todos voltados para obter maior proteção ou performance do sistema de armazenamento.

DAS OU DIRECT ATTACHED STORAGE

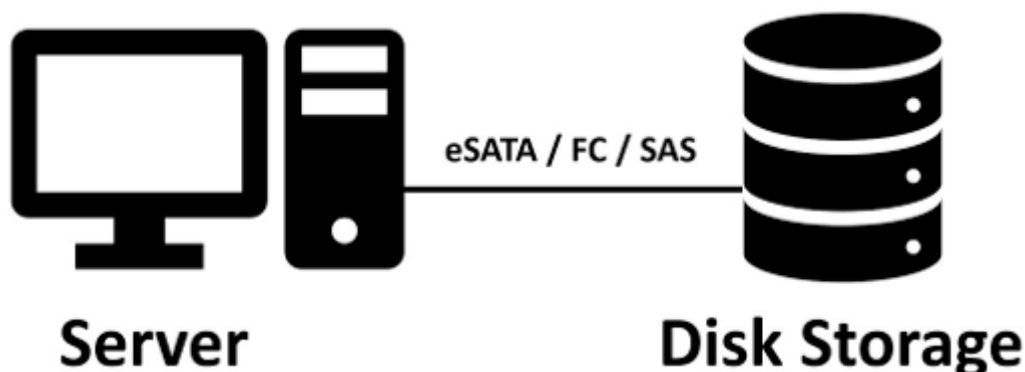
Os **DAS (Direct Attached Storage)** são quaisquer **unidades conectadas diretamente a computadores**, sejam eles servidores, *notebooks* ou estações de trabalho.

Ao ser instalada, uma **unidade de armazenamento direto** torna-se um disco adicional para seu *host*, seja ele um servidor, computador ou qualquer outro dispositivo. Como esses equipamentos não possuem sistema operacional próprio, **todo o gerenciamento de um storage DAS acontece no servidor hospedeiro**.

A principal vantagem desse tipo de equipamento é o sistema operacional do computador hospedeiro ter **comunicação direta com a área de dados**, sem a necessidade de adicionar e interpretar protocolos nas requisições para comunicação de dados.

Esse **acesso direto** acaba se traduzindo em maior performance, pois o servidor que possui um sistema de armazenamento direto não precisa encapsular dados e utilizar parte do sistema para controlar o tráfego de dados dentro da rede.

Como exemplo de armazenamento para conexão direta, inclui-se desde *pen drives* e *hard disks* baseados em portas USB até **storages** equipados com conexões **mini-SAS** ou **Fibre Channel** para grandes infraestruturas profissionais de TI, como *datacenters*.



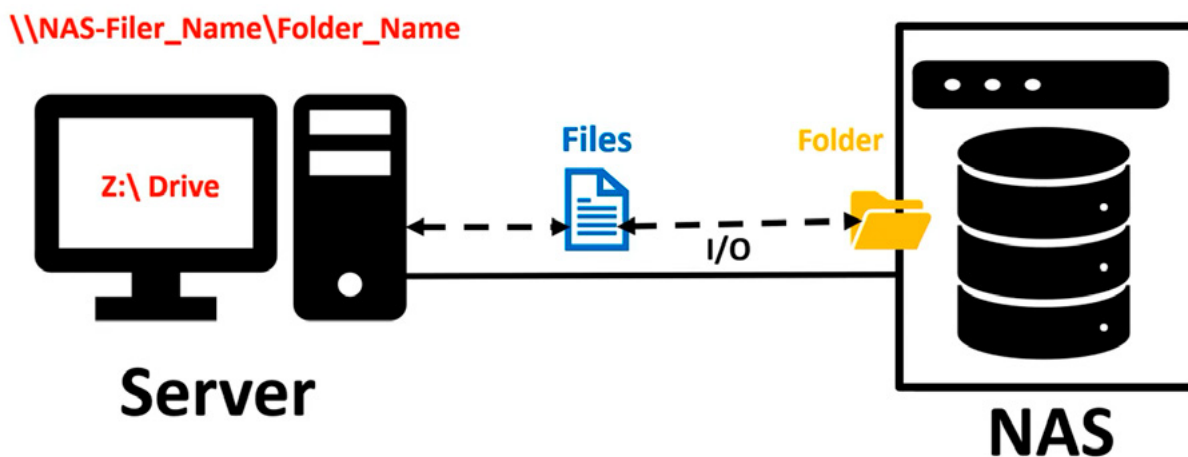
NAS ou NETWORK ATTACHED STORAGE

Diferentemente dos sistemas de armazenamento direto, os **Network Attached Storages** são equipamentos que possuem sistemas operacionais incorporados.

Obs.: Isso significa que esse tipo de **storage** não precisa de um servidor ou *host* para funcionar, sendo **conectado diretamente em qualquer rede local**.

Um **NAS** é basicamente um **grande disco rígido conectado à rede**, com recursos profissionais de gerenciamento, componentes redundantes para maior proteção dos dados e capaz de centralizar dados de forma organizada.

Também conhecidos como **network storages**, esses equipamentos permitem armazenar, compartilhar, acessar e gerenciar todos os dados armazenados por todos os usuários autorizados via rede, utilizando recursos como acesso protegido por *login*, senha e recursos para registrar as atividades de cada usuário.



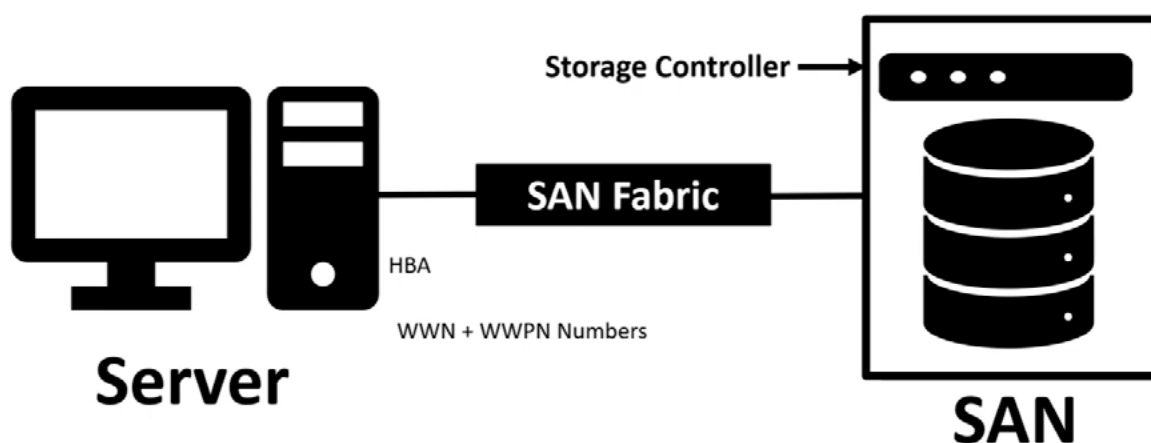
SAN OU STORAGE AREA NETWORK

SAN (Storage Area Network) é uma **rede corporativa de dados** voltada para **conectar e manter disponíveis servidores e storages via LAN** dentro de um ambiente seguro, redundante e de alta performance.

Essa **rede de armazenamento exclusiva para dados** proporciona alta velocidade e segurança para as redes corporativas, separando logicamente o tráfego gerado pelos usuários dos sistemas de armazenamento e servidores.

Utilizada em infraestruturas de TI corporativas, uma rede **SAN** é uma **rede de armazenamento composta exclusivamente por servidores e storages**, interligados através de **conexões IP (iSCSI)** ou **Fibre Channel (FC)**.

Buscando simplificar e consolidar a produção de dados, uma rede **SAN** centraliza e melhora o gerenciamento das informações, proporcionando mais segurança e velocidade no acesso aos dados em grandes infraestruturas de TI e *datacenters*.



RAID

Os requisitos de armazenamento de dados de algumas aplicações (em particular, aplicações Web, banco de dados e multimídia) têm crescido tão rapidamente que um número maior de discos é necessário para armazenar seus dados, mesmo com as capacidades da unidade de disco crescendo muito rapidamente.

Ter grande quantidade de discos em um sistema apresenta oportunidades para melhorar a velocidade em que os dados podem ser lidos ou escritos, se os discos forem operados em paralelo. **Várias leituras ou escritas independentes também podem ser realizadas em paralelo.** Além do mais, essa configuração oferece o potencial para melhorar a confiabilidade do armazenamento de dados, pois as **informações redundantes podem ser armazenadas em vários discos.** Assim, a **falha de um disco não leva à perda de dados.**

Diversas técnicas de organização de disco, coletivamente chamadas de **redundant arrays of independent disks (RAID)**, têm sido propostas para alcançar melhor desempenho e confiabilidade.

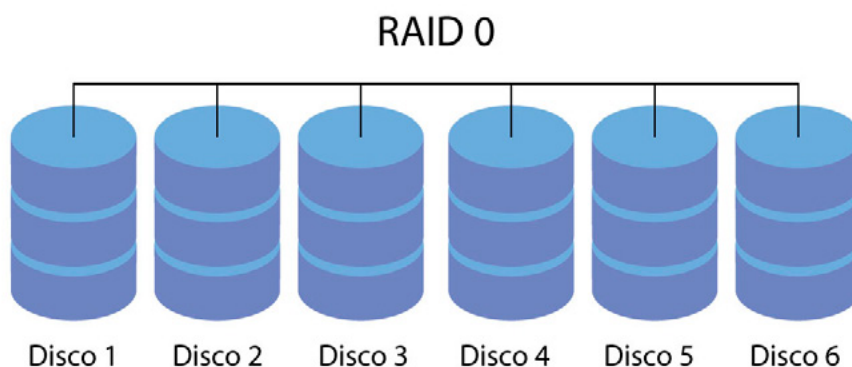
Existem vários tipos de **RAID** que podem ser realizados com **discos de armazenamento**. Os arranjos de disco **RAID** podem ser criados e gerenciados por meio de um *software* ou de um *hardware* próprio para esse fim. Em linhas gerais, temos os seguintes **RAIDs**:

RAID 0

Esse arranjo de discos utiliza dois ou mais discos rígidos com finalidade da maximização do desempenho ao armazenar e acessar informações. A técnica também é conhecida como **"Stripping Array"** ou fracionamento e é considerada o nível de **RAID** mais rápido, porém o menos seguro.

A lógica do **RAID 0** consiste em **distribuir os dados a serem armazenados no sistema de armazenamento**, gravando a informação particionada em diversos discos rígidos de forma simultânea. Dessa forma, a gravação (ou leitura) dos dados utiliza todos os discos do arranjo, sem necessidade de nenhum cálculo para geração de paridade.

Em consequência do alto desempenho conseguido, o **RAID 0** perde em questões de segurança, não proporcionando nenhuma tolerância a falhas. Caso apenas um dos discos apresente problemas, todos os dados do sistema estarão comprometidos.

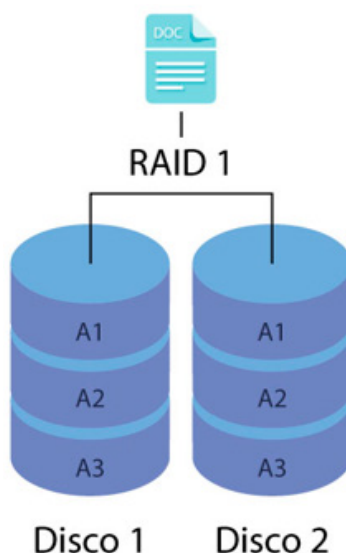


RAID 1

Também conhecido como **disk mirror**, replicação ou espelhamento de disco, o arranjo **RAID 1** é ideal para pequenas empresas e residências, sendo implementado com apenas dois discos rígidos. Esse sistema faz uma cópia dos aplicativos e dados em tempo real, **escrevendo as informações de forma simultânea nos discos**, sem a intervenção do usuário.

Obs.: A desvantagem desse tipo de implementação é a **redução de capacidade bruta**, pois como as informações serão escritas integralmente duas vezes, somente a metade da capacidade total será disponibilizada pelo sistema.

O principal atributo desse mecanismo é a segurança gerada para os dados armazenados, pois mesmo que um dos discos falhe, as informações estarão seguras e on-line no outro disco, possibilitando que o disco defeituoso seja substituído ou que o *backup* dos dados seja realizado sem que nada seja perdido.



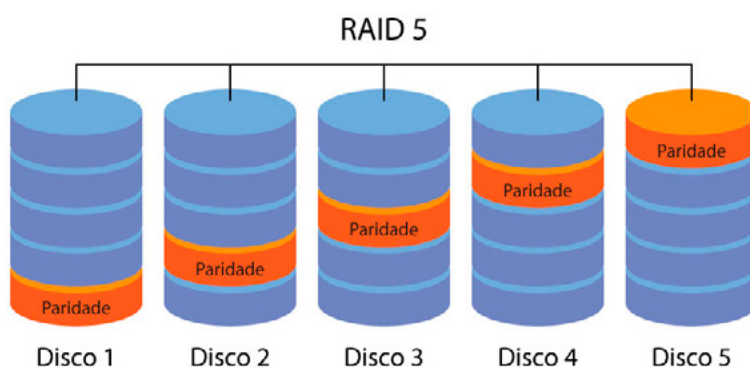
RAID 5

O modelo **RAID 5** é conhecido como “**Strip Set com Paridade**”. O método é muito utilizado em servidores ou *storages* com pelo menos três discos rígidos instalados e cria uma camada de redundância, sacrificando parte da capacidade de armazenamento do sistema para proporcionar maior segurança aos dados.

No **RAID 5**, **bits de paridade são criados e acrescentados aos dados, escritos de forma alternada em todos os discos**.



Caso um dos discos falhe enquanto o sistema estiver funcionando, nenhum dado será perdido. A **paridade** é a segurança do sistema que possibilita a **reconstrução dos dados sem perda de informação**.



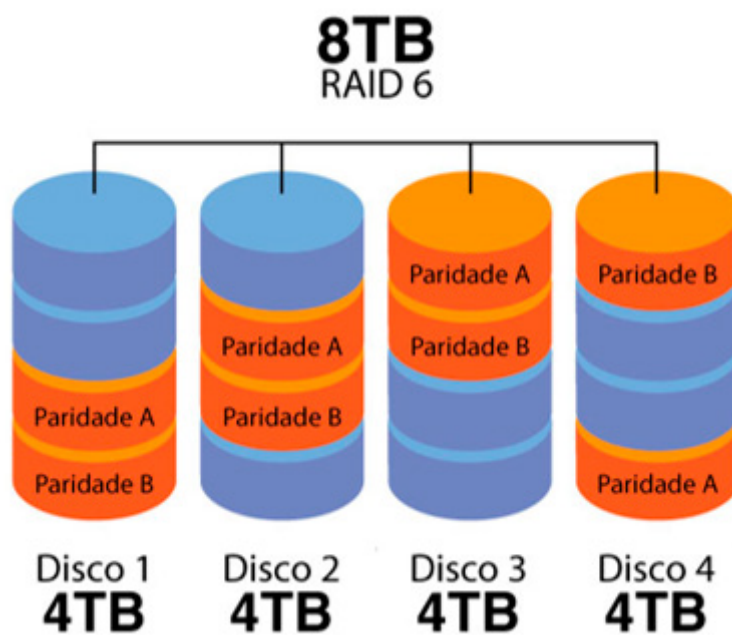
RAID 6

O **RAID 6** é um arranjo com características próximas ao **RAID 5**, mas **acrescentando dupla paridade às informações gravadas**.

Obs.: Isso significa que, no **RAID 6**, até dois discos rígidos podem falhar sem perda de dados, porém o dobro do espaço será utilizado para gravar a paridade e manter a redundância do sistema. Assim, numa implantação em sistema **RAID 6** com oito discos, a capacidade total disponível será a multiplicação do valor de seis discos rígidos, sendo que os outros dois servirão para redundância.

DICA

À medida que aumentarmos o número de discos nesse arranjo, também minimizaremos o espaço proporcional sacrificado. Esse tipo de sistema pode ser implementado a partir de quatro discos e é usado em situações que demandam maior segurança para as informações armazenadas.

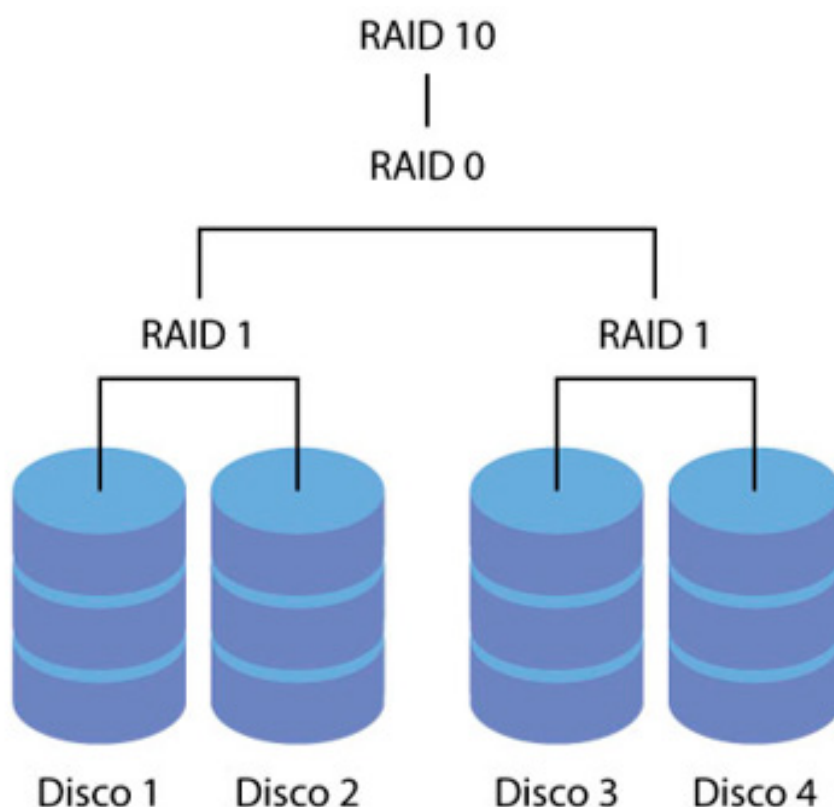


RAID 10 ou RAID 1+0

Nesse arranjo, é feita a **combinação de dois ou mais subgrupos de espelhamentos agrupados numa única matriz**. Para implementação de um arranjo **RAID 10**, são utilizados pelo menos 4 discos rígidos, nos quais são criadas combinações de dois ou mais grupos **RAID 1** para escrita simultânea (**RAID 0**).

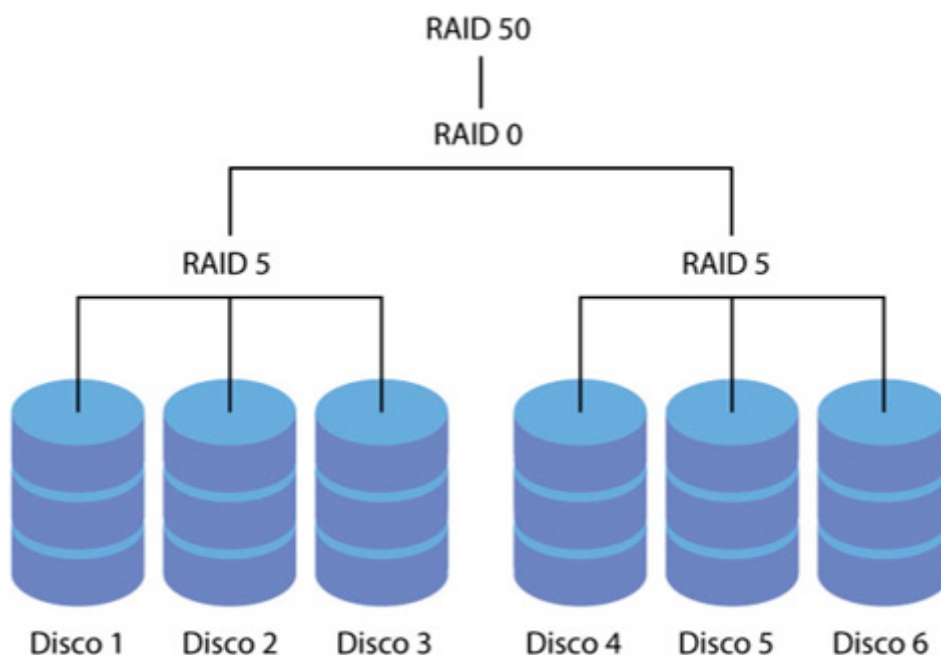
Sua principal característica é que ela une desempenho e segurança em um único agrupamento de discos. Além de oferecer um desempenho superior no momento de transferência dos dados, ela não compromete a integridade das informações caso ocorra algum problema em um dos discos rígidos.

Obs.: O maior inconveniente para usuários que fazem uso dos arranjos **RAID 10** é o grande número de discos rígidos utilizados para segurança dos dados em relação aos demais, tornando-se economicamente inviável para algumas aplicações.



RAID 50 ou RAID 5+0

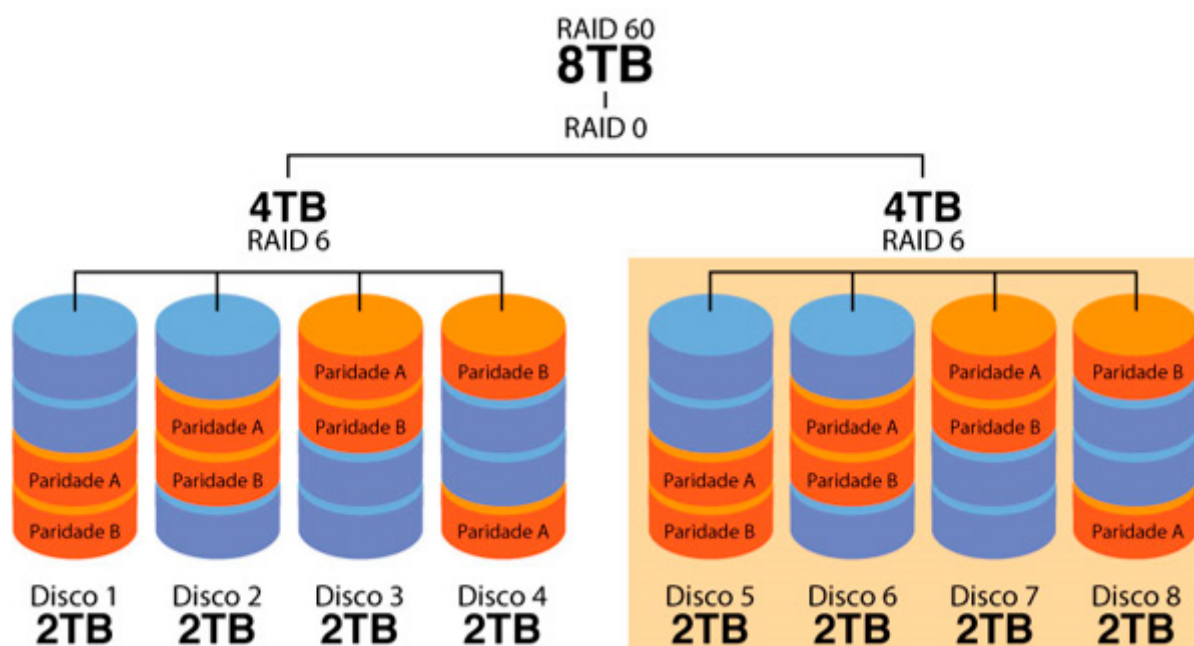
O **RAID 50**, também conhecido como **RAID 5 + 0**, combina a **paridade distribuída (RAID 5)** com **striping (RAID 0)** e requer um mínimo de seis unidades. Esse nível de **RAID** oferece melhor desempenho de gravação, maior proteção de dados e recriações mais rápidas do que o **RAID 5**. O desempenho não diminui tanto quanto em um **array RAID 5**, porque uma única falha afeta apenas um **array**. Até quatro falhas de discos podem ser superadas, desde que cada unidade com falha ocorra em um **array RAID 5** diferente.



RAID 60 ou RAID 6+0

Combinando dois arranjos já abordados, o **RAID 60 (6+0)** também é um híbrido que combina duas configurações RAID num único *pool* de discos: o **RAID 6** e o **RAID 0**.

Com exigência de pelo menos 8 discos rígidos e de dois subgrupos **RAID 6** montados no mesmo *pool*, esse sistema mantém **dupla paridade por subgrupo**. Isso garante que o sistema continuará funcionando mesmo que quatro discos falhem simultaneamente, sem que haja perda de informações.



ACESSO AO BLOCO DE DISCO

As **solicitações de E/S** (entrada e saída) de disco são geradas pelo sistema de arquivos e pelo gerenciador de memória virtual, encontrado na maioria dos sistemas operacionais. Cada solicitação **especifica o endereço no disco a ser referenciado**. Esse endereço está na forma de um número de bloco.

Obs.: Um **bloco** é uma **unidade lógica consistindo em um número fixo de setores contíguos**. Os tamanhos de bloco **variam de 512 bytes a vários kilobytes**. Os dados são **transferidos entre o disco e a memória principal** em unidades de blocos.

Uma sequência de solicitações de blocos do disco pode ser classificada como um padrão de **acesso sequencial** ou como um padrão de **acesso aleatório**.

Em um padrão de **acesso sequencial**, ocorrem **solicitações sucessivas dos números de blocos sucessivos**, que se **encontram na mesma trilha ou em trilhas adjacentes**. Para ler blocos no **acesso sequencial**, é necessário buscar o primeiro bloco do disco, mas as solicitações sucessivas não exigiriam uma busca, ou exigiriam uma busca em uma trilha adjacente, o que é mais rápido do que uma busca a uma trilha muito mais afastada.

Ao contrário do que ocorreria em um padrão de **acesso aleatório**, em que são realizadas **buscas sucessivas dos blocos localizados aleatoriamente no disco**. Toda solicitação desse tipo exigiria uma busca.



O número de **acessos a blocos aleatórios**, que pode ser atendido por um único disco em um segundo, depende do **tempo de busca** e geralmente compreende 100 a 200 acessos por segundo. Considerando que apenas uma pequena quantidade (um bloco) de dados é lida por busca, a **taxa de transferência** é muito mais lenta com um padrão de **acesso aleatório** do que com um padrão de **acesso sequencial**.

JOURNALING

Um sistema de arquivos com **journaling** é aquele que **rastreia as alterações ainda não confirmadas na parte principal do sistema de arquivos**, registrando as intenções de tais alterações em uma estrutura de dados conhecida como **log de dados** (diário), que geralmente é um **log circular**.

Obs.: No caso de uma falha do sistema ou falha de energia, esses sistemas de arquivos podem ser colocados on-line novamente mais rapidamente com uma probabilidade menor de serem corrompidos.

Dependendo da implementação real, um sistema de arquivos com **journaling** pode apenas **controlar os metadados armazenados**, resultando em melhor desempenho às custas de maior possibilidade de corrupção de dados. Alternativamente, um sistema de arquivos com **journaling** pode **rastrear os dados armazenados e os metadados relacionados**, enquanto algumas implementações permitem um comportamento selecionável a este respeito.

Exemplos de sistemas de arquivos que suportam **journaling**: Ext3, Ext4, JFS, JFFS, JFFS2, LogFS, NTFS, Reiser4, ReiserFS e XFS.

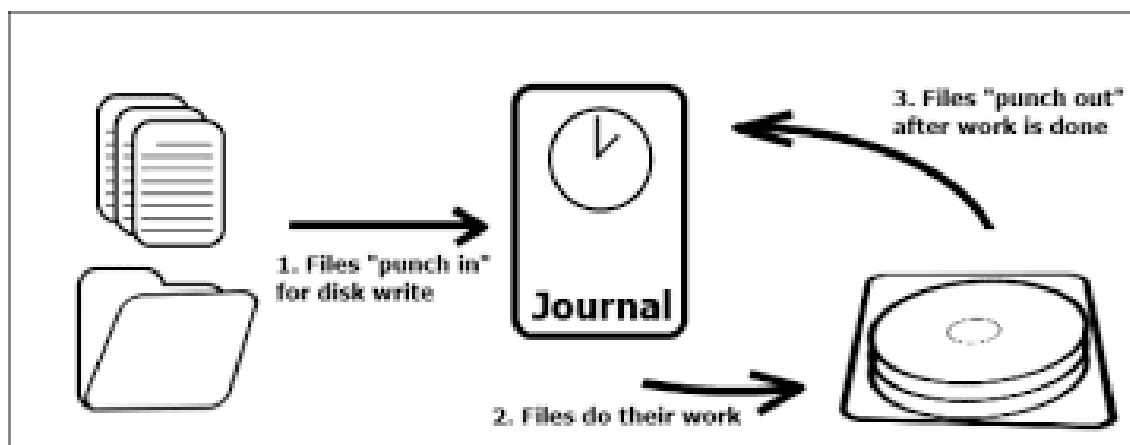


Figura 2 – Funcionamento do journal em um sistema de arquivos

Para refletir mudanças em arquivos e diretórios, a atualização de sistemas de arquivos no geral requer muitas operações de escrita separadas. Isso possibilita que uma interrupção (como uma queda do fornecimento de energia ou um *crash* do sistema) entre as operações de escritas deixem as estruturas de dados em um estado intermediário inválido.

Detectar e recuperar essas inconsistências normalmente exige uma **checagem completa das suas estruturas de dados**. Um exemplo de ferramenta adequada para tal ação é o **fsck** (o verificador do sistema de arquivos). O padrão é que isso ocorra antes que o sistema de arquivos seja novamente montado para acesso de leitura-escrita (*read-write*). Se o sistema de arquivos for grande e se houver pequena largura de banda de E/S, pode fazer com que demore muito tempo e resulte em um tempo de parada mais longo se houver o bloqueio do restante do sistema de voltar à ativa.

Para evitar essa demora, um sistema de arquivo com **journaling** aloca uma área especial – o **journal** – no qual registra com antecedência as mudanças que fará. Após um acidente, a recuperação simplesmente envolve a **leitura do journal do sistema de arquivos e a repetição das mudanças deste journal até o sistema de arquivos voltar a ser consistente**. Essas mudanças são, portanto, atômicas (indivisíveis) na medida em que todas as mudanças são realizadas (repetidas completamente durante a recuperação), ou não são reproduzidas de forma alguma (são ignoradas porque ainda não tinham sido completamente escritas no *journal* antes da ocorrência do *crash*).

2. ORGANIZAÇÃO DE ARQUIVOS

Um banco de dados é **mapeado em diversos arquivos diferentes**, mantidos pelo sistema operacional básico. Esses arquivos residem permanentemente nos discos. Um **arquivo é organizado logicamente como uma sequência de registros, mapeados para blocos de disco**.

Os **arquivos** representam uma construção básica nos sistemas operacionais, de modo que devemos pressupor a existência de um sistema de arquivos básico.

Cada **arquivo** também é **particionado logicamente em unidades de armazenamento de tamanho fixo**, chamados de **blocos**, que são as **unidades de alocação de armazenamento e transferência de dados**.

Um **bloco** pode **conter vários registros**. O conjunto exato de registros contidos em um bloco é determinado pela **forma de organização dos dados físicos em uso**. Devemos presumir que **nenhum registro é maior do que um bloco**.

Em um **banco de dados relacional**, as tuplas de relações distintas têm geralmente tamanhos diferentes. Uma abordagem para o **mapeamento do banco de dados para arquivos é utilizar vários arquivos e armazenar registros de apenas um tamanho fixo em um arquivo específico**.

Os dados costumam ser armazenados na forma de **registros**. Cada registro **contém uma coleção de valores ou itens de dados relacionados**, em que cada valor é formado por um ou mais *bytes* e **corresponde a um campo em particular do registro**. Os **registros** normalmente **descrevem entidades e seus atributos**.

Por exemplo, um registro de FUNCIONARIO representa uma entidade de funcionário e o valor de cada campo no registro especifica algum atributo desse funcionário, como Nome, Data_nascimento, Salario ou Supervisor.

Uma **coleção de nomes de campo e seus tipos de dados correspondentes** constituem uma definição de **tipo de registro** ou **formato de registro**. Um **tipo de dado**, associado a cada campo, **especifica os tipos de valores que um campo pode assumir**. O que pode incluir os tipos de dados **numéricos** (inteiro, inteiro longo ou ponto flutuante), **cadeia de caracteres** (tamanho fixo ou variável), **booleanos** (tendo apenas valores 0 e 1, ou *TRUE* e *FALSE*) e, às vezes, tipos **data e hora** especialmente codificados.

Em algumas aplicações de banco de dados, pode haver necessidade de **armazenar itens de dados que consistem em grandes objetos não estruturados**, que representam imagens, vídeos digitalizados, *streams* de áudio ou texto livre. Estes são conhecidos como **BLOB (objetos binários grandes)**. Um item de dados **BLOB** costuma ser **armazenado separadamente de seu registro**, em um **conjunto de blocos de disco**, e um **ponteiro para o BLOB é incluído no registro**. Para **armazenar texto livre**, alguns **SGBDs** (por exemplo, Oracle, DB2 etc.) oferecem um tipo de dado chamado **CLOB (objeto grande de caracteres)**, alguns **SGBDs** chamam esse tipo de dado de **texto**.

REGISTROS DE TAMANHO FIXO E VARIÁVEL

Um **arquivo** é uma **sequência de registros**. Em muitos casos, **todos os registros em um arquivo são do mesmo tipo de registro**.

Se cada registro no arquivo tem **exatamente o mesmo tamanho** (em *bytes*), o arquivo é considerado composto de **registros de tamanho fixo**.

Se diferentes registros no arquivo **possuem diversos tamanhos**, o arquivo é considerado composto de **registros de tamanho variável**.

Os registros de FUNCIONARIO de **tamanho fixo** na figura a seguir têm um tamanho de registro de 71 *bytes*.



Figura 3 – Registro de Tamanho Fixo

Cada registro tem os mesmos campos, e os **tamanhos de campo são fixos**, de modo que o sistema pode identificar a posição de *byte* inicial de cada campo em relação à posição inicial do registro. Isso facilita a localização de valores de campo pelos programas que acessam tais arquivos.

Para campos de **tamanho variável**, cada registro **tem um valor para cada campo**, mas **não é conhecido o tamanho exato de alguns valores de campo**. Em um registro em particular, para determinar os *bytes* que representam cada campo, é possível usar **caracteres separadores especiais** (como? ou % ou \$) – que não aparecem em qualquer valor do campo – para **terminar os campos de tamanho variável**, como mostra a figura a seguir, ou é possível armazenar o tamanho do campo em *bytes* no próprio registro, antes do valor do campo.

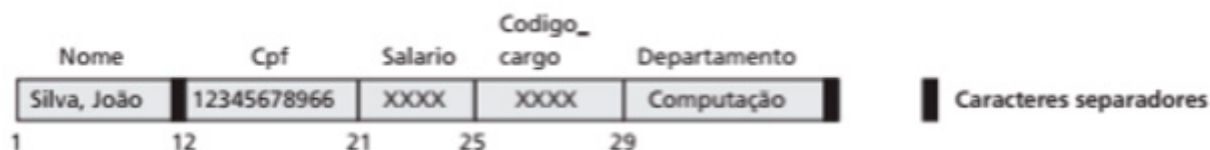


Figura 4 – Registro de Tamanho Variável

ALOCÇÃO DE ESPAÇO EM DISCO

O sistema operacional possui uma estrutura de dados que armazena informações que possibilitam ao sistema de arquivos gerenciar as áreas ou blocos livres. Nessa estrutura, geralmente uma lista ou tabela, são identificados **blocos livres** que poderão ser **alocados por um novo arquivo**. Quando um **arquivo é eliminado, todos os seus blocos são liberados para a estrutura de espaços livres**.

Existem várias técnicas-padrão para **alocar os blocos de um arquivo no disco**.

Alocação Contígua

A **alocação contígua** consiste em **armazenar um arquivo em blocos sequencialmente dispostos**, permitindo ao sistema localizar um arquivo por meio do **endereço do primeiro bloco** e da sua **extensão em blocos**.

Obs.: Um problema desse tipo de alocação é que quando um arquivo é criado com n blocos, é necessário que exista uma cadeia de n blocos livres disposto sequencialmente. Nesse tipo de alocação, o disco é visto como um grande vetor, com segmentos ocupados e livres.

Na **alocação contígua**, os blocos de arquivo são alocados em blocos de disco consecutivos.

Alocação Encadeada

Na **alocação encadeada**, um arquivo pode ser organizado como um **conjunto de blocos ligados logicamente no disco, independente da sua localização física**, sendo que **cada bloco possui um ponteiro para o bloco seguinte do arquivo**, e assim sucessivamente.

Neste tipo de alocação, ocorre grande fragmentação dos arquivos devido aos blocos livres dos arquivos **não precisarem ser contíguos**. Aos **diversos pedaços derivados da quebra do arquivo**, dá-se o nome de **extents**. Essa fragmentação aumenta o tempo de acesso aos arquivos, pois exige que o mecanismo de leitura/gravação se desloque diversas vezes sob sua superfície. Dessa forma, se torna necessária a **execução da operação de desfragmentação periodicamente**.

Outro problema na **alocação encadeada** é que ela só permite o **acesso sequencial aos blocos dos arquivos**, não possuindo acesso direto aos blocos, e **desperdiça espaço nos blocos com o armazenamento de ponteiros**.

Obs.: Na **alocação encadeada**, cada bloco de arquivo contém um ponteiro para o próximo bloco. Isso facilita a expansão do arquivo, mas torna sua leitura mais lenta.

Alocação Indexada

A **alocação indexada** soluciona o problema da alocação encadeada referente ao acesso direto aos blocos dos arquivos, pois **mantém os ponteiros de todos os blocos do arquivo em uma única estrutura denominada bloco de índice**.

CABEÇALHOS DE ARQUIVO

Um **cabeçalho de arquivo** ou descritor de arquivo **contém informações sobre um arquivo**, que são exigidas pelos programas do sistema que acessam os registros do arquivo.

O **cabeçalho** inclui informações para **determinar os endereços de disco dos blocos de arquivo**, bem como para **registrar descrições de formato**, que podem incluir **tamanhos de campo** e a **ordem dos campos em um registro**, para registros de **tamanho fixo**, e **códigos de tipo de campo**, **caracteres separadores** e **códigos de tipo de registro**, para registros de **tamanho variável**.

OPERAÇÕES EM ARQUIVOS

As **operações em arquivos** costumam ser agrupadas em **operações de recuperação** e **operações de atualização**.

As **operações de recuperação** não mudam quaisquer dados no arquivo, apenas **localizam certos registros** de modo que seus valores de campo possam ser examinados e processados.



As **operações de atualização** mudam o arquivo **pela inserção ou exclusão de registros**, ou pela **modificação dos valores de campo**. De qualquer forma, podemos ter **de selecionar um ou mais registros para recuperação, exclusão ou modificação** com base em **uma condição de seleção** (ou **condição de filtragem**), que especifica **critérios que os registros desejados devem satisfazer**.

Obs.: Considere um arquivo FUNCIONARIO com os campos Nome, Cpf, Salario, Codigo_cargo e Departamento. Uma condição de seleção simples pode envolver uma **comparação de igualdade em algum valor de campo** — por exemplo, Cpf = '12345678966' ou Departamento = 'Pesquisa'. O caso geral é ter uma **expressão booleana** qualquer nos campos do arquivo como **condição de seleção**.

Quando vários registros de arquivo satisfazem uma **condição de pesquisa**, o **primeiro registro** — em relação à sequência física de registros de arquivo — é inicialmente **localizado** e **designado** como o **registro atual**. As operações de busca subsequentes **começam desse registro** e **localizam o próximo registro** no arquivo que satisfaz a condição.

ARQUIVOS DE REGISTROS DESORDENADOS (ARQUIVOS HEAP)

Neste tipo de organização mais simples e mais básico, os **registros são arquivados na ordem em que são inseridos**, de modo que **novos registros são inseridos ao final do arquivo**. Essa organização é chamada **arquivo de heap ou pilha**. Normalmente, ela é usada com caminhos de acesso adicionais, como os índices secundários. Ela também é usada para coletar e armazenar registros de dados para uso futuro.

A **inserção de um novo registro** é muito **eficiente**.

Um registro pode ser **colocado em qualquer lugar no arquivo onde exista espaço para acomodá-lo**. Não existe ordenação de registros. Normalmente, **existe um único arquivo para cada relação**.

ARQUIVOS DE REGISTROS ORDENADOS (ARQUIVOS CLASSIFICADOS)

Podemos **ordenar fisicamente os registros de um arquivo no disco** com base nos valores de um de seus campos – chamado de **campo de ordenação**. Isso leva a um **arquivo ordenado ou sequencial**. Se o **campo de ordenação** também for um **campo-chave do arquivo** – um campo com garantias de ter um valor exclusivo em cada registro –, o campo é chamado de **chave de ordenação para o arquivo**.

A figura seguinte mostra um **arquivo ordenado** com Nome como **campo-chave de ordenação** (supondo que os funcionários tenham nomes distintos):

	Nome	Cpf	Data_nascimento	Cargo	Salario	Sexo
Bloco 1	Aaron, Eduardo					
	Abilio, Diana					
	⋮					
	Acosta, Marcos					
Bloco 2	Adams, João					
	Adams, Roberto					
	⋮					
	Akers, Janete					
Bloco 3	Alexandre, Eduardo					
	Alfredo, Roberto					
	⋮					
	Allen, Samuel					
Bloco 4	Allen, Tiago					
	Anderson, Kely					
	⋮					
	Anderson, Joel					
Bloco 5	Anderson, Isaac					
	Angeli, José					
	⋮					
	Anita, Sueli					
Bloco 6	Arnold, Marcelo					
	Arnold, Estêvão					
	⋮					
	Atkins, Timóteo					

Bloco $n-1$	Wong, Jaime					
	Wood, Ronaldo					
	⋮					
	Woods, Manuel					
Bloco n	Wright, Pâmela					
	Wyatt, Charles					
	⋮					
	Zimmer, André					

Obs.: Os **registros ordenados** têm algumas vantagens em relação aos arquivos desordenados. Por exemplo, a **leitura dos registros na ordem dos valores da chave de ordenação é extremamente eficiente** porque **nenhuma classificação é necessária**. Além disso, os registros são armazenados em **ordem sequencial**, de acordo com o valor de uma “**chave de busca**” de cada registro.

TÉCNICAS DE HASHING

Outro tipo de organização de arquivo principal é **baseado no hashing**, que oferece **acesso muito rápido aos registros** sob certas condições de pesquisa. Essa organização costuma ser chamada de **arquivo de hash**.

A **condição de pesquisa** precisa ser uma **condição de igualdade em um único campo**, chamado **campo de hash**. Na maior parte dos casos, o **campo de hash** também é um **campo-chave do arquivo**, quando é chamado de **chave hash**.

DICA

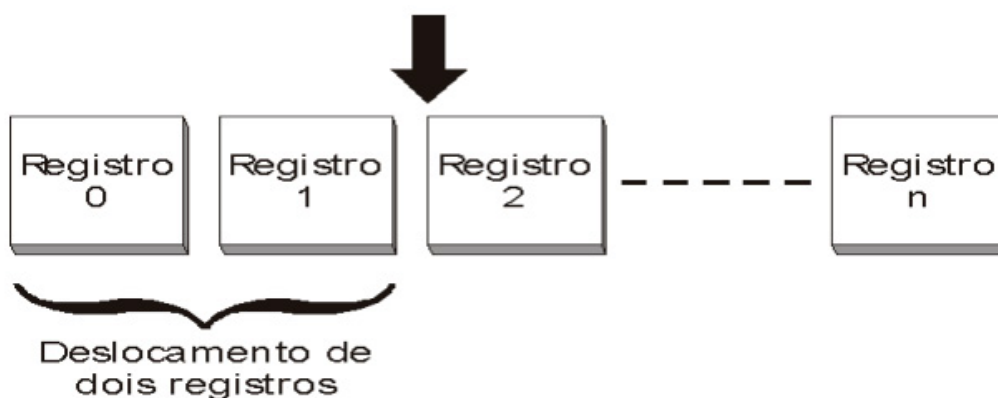
A ideia por trás do **hashing** é oferecer uma função h , chamada **função de hash** ou **função de randomização**, que é aplicada ao **valor do campo de hash** de um registro e **gera o endereço do bloco de disco em que o registro está armazenado**.

Uma **função de hash** é calculada sobre algum atributo de cada registro. O **resultado da função de hash** determina **em que bloco do arquivo o registro deve ser colocado**. Está bastante relacionada às **estruturas de indexação**.

3. MÉTODOS DE ACESSO

Em função de como o arquivo está organizado, o sistema de arquivos pode recuperar registros de diferentes maneiras. Inicialmente, os primeiros sistemas operacionais só **armazenavam arquivos em fitas magnéticas**. Com isso, o **acesso era restrito à leitura dos registros na ordem em que eram gravados**, e a **gravação de novos registros só era possível no final do arquivo**. Este tipo de acesso, chamado de **acesso sequencial**, era próprio da fita magnética, que, como meio de armazenamento, possuía esta limitação.

Com o advento dos discos magnéticos, foi possível a introdução de métodos de acesso mais eficientes. O primeiro a surgir foi o **acesso direto**, que **permite a leitura/gravação de um registro diretamente na sua posição**. Este método é realizado por meio do **número do registro**, que é a sua **posição relativa ao início do arquivo**.



Obs.: O **método sequencial de acesso** é o mais tradicional e consiste em efetuar a **leitura dos registros, um após o outro**, comparando o **argumento de pesquisa** com o valor do campo-chave (primária ou secundária) no registro corrente, até encontrar os registros desejados ou o final do arquivo.

PROGRAMA Y
INÍCIO....

Repita até fim
ler registro

chave secundária (campo chave)

SE NOME = "JOSÉ"
ENTÃO IMPRIMIR

argumento de pesquisa

Fim repita (volte a ler)

FIM DO PROGRAMA

Obs.: O **método direto de acesso** consiste em **recuperar os registros** desejados, **sem a necessidade de efetuar a leitura dos registros que os antecedem**, o que pode ser feito **por meio de um índice ou** com o auxílio de um **algoritmo de randomização que localiza o registro**, calculando a posição ocupada pelo registro no disco **com base no valor do argumento de pesquisa**, que deve ser um campo numérico.



O PULO DO GATO

Em ambos os casos, a localização do registro ocorre a cargo do gerenciador de arquivos, de maneira transparente para o programador, que só precisa escolher a organização adequada para o arquivo e fornecer no programa o **argumento de pesquisa**.

PROGRAMA Z
INÍCIO....

.

ABRIR ARQUIVO **ALUNO INDEXADO POR NOME**

.

NOME="JOSÉ" → argumento de pesquisa

LOCALIZAR REGISTRO → acesso direto (indexado)

SE ENCONTROU REGISTRO

ENTÃO IMPRIMIR

.

.

FIM DO PROGRAMA

ORGANIZAÇÃO SEQUENCIAL

A **organização sequencial** caracteriza-se pela existência de uma **chave de ordenação**. Essa chave determina a **ordem em que os registros são armazenados**, e pode ser simples ou composta por dois ou mais campos. Geralmente, coincide com a **chave primária**, mas não obrigatoriamente.

A **organização sequencial** somente permite o **acesso sequencial**.

Obs.: Na figura a seguir, temos um arquivo com **organização sequencial** e **chave primária** (MATRICULA), distinta da **chave de ordenação** (NOME).

ARQUIVO ALUNO

↗ **chave primária**
↗ **chave de ordenação**

MATRICULA	NOME	ENDEREÇO	DT_NASC
001	Ana	SQS 308 ...	23/08/78
003	José	QND 14	25/09/70
002	José	SQN 410 ...	10/08/85
005	Maria	GAMA	05/04/76
.	.	.	.
.	.	.	.

ORGANIZAÇÃO SERIAL

Nesta forma de organização, os registros são **armazenados de acordo com a ordem de inclusão**.

Obs.: O arquivo **não possui chave de ordenação**, portanto, não existe preocupação com a ordem de armazenamento dos registros. No entanto, é sempre recomendável o arquivo possua uma **chave primária**.

DICA

A **organização serial** somente permite o **acesso sequencial**. Não deve ser utilizada em processos de exclusão e alteração de registros na modalidade *batch* (atualização em lote), pois degrada a performance.

É muito utilizada em processos de inclusão de registros em que não haja preocupação de manter a sequência dos mesmos. É também empregada no arquivo de dados que serve de base para a **organização indexada**.

Obs.: Na figura a seguir, temos um arquivo com **organização serial**. Note que ele não possui **chave de ordenação**, apenas **chave primária** (MATRICULA).

ARQUIVO ALUNO

↗ **chave primária**

MATRICULA	NOME	ENDEREÇO	DT_NASC
005	Maria	SQS 308 ...	23/08/78
003	José	QND 14	25/09/70
002	Ana	SQN 410 ...	10/08/85
001	José	GAMA	05/04/76
.	.	.	.

ORGANIZAÇÃO INDEXADA

Nesta forma de organização, os registros são armazenados em um arquivo de dados com **organização serial**. Para cada campo (ou combinação deles), por meio do qual se deseja obter **acesso direto (indexado)**, deve-se criar um **arquivo de índice (processo de indexação)**. Um mesmo arquivo de dados pode possuir diversos arquivos de **índice** a ele associados.

O **arquivo de índice** é composto basicamente por **duas colunas**. A primeira corresponde ao campo utilizado no **processo de indexação (endereço lógico)** e a segunda **armazena um valor (endereço físico)** que serve como referência, a fim de que o gerenciador de arquivos localize o registro no disco magnético.

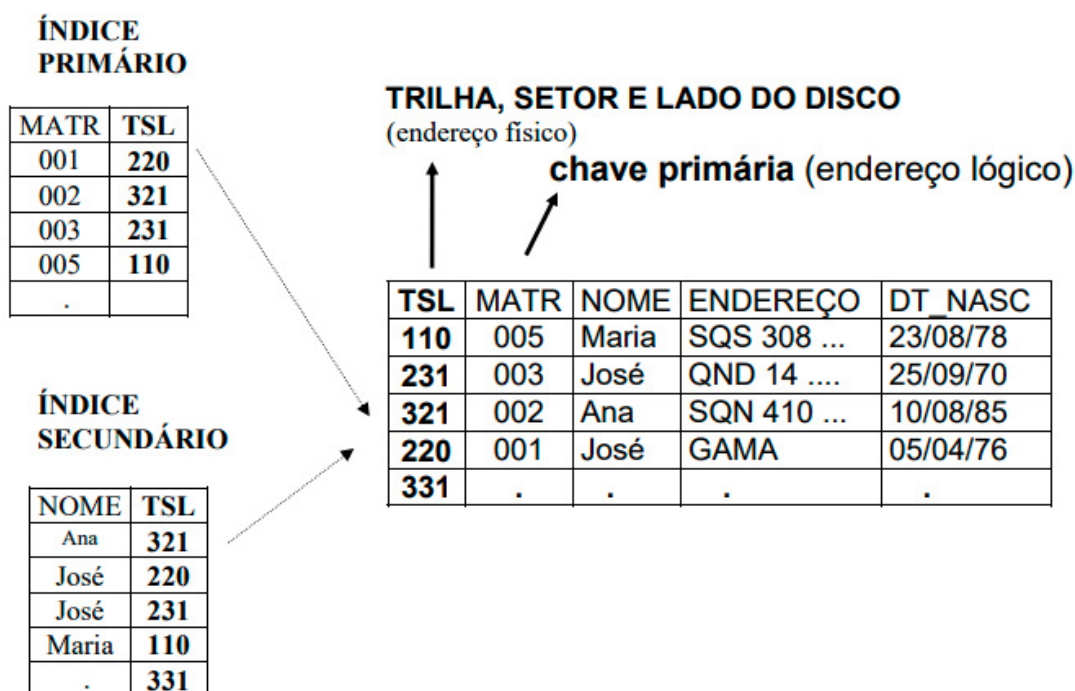
DICA

Os **registros dos arquivos de índice** são **ordenados pelo endereço lógico**. Portanto, se utilizarmos um algoritmo de leitura sequencial em um arquivo indexado por nome, por exemplo, obtaremos os registros em ordem alfabética, mesmo sendo o arquivo de dados um arquivo serial. Ou seja, prevalece a ordem do índice.

Sempre que um arquivo índice for referenciado por um programa, ele será carregado para memória principal, o que torna desprezível o tempo de busca dos registros nesse arquivo. Além disso, o algoritmo utilizado na busca é o de **pesquisa binária**, o que reduz ainda mais o tempo.

Obs.: Os índices constituídos com base no valor da **chave primária** ou **chave candidata** são conhecidos como **índices primários** e os demais como **índices secundários**.

ARQUIVO ALUNO



ORGANIZAÇÃO EM CAMPOS

Um **campo** é a **menor unidade lógica de informação em um arquivo**.

Existem várias maneiras de organizar um arquivo mantendo a identidade dos campos. Um desses métodos é **forçar todos os campos para um tamanho fixo**. Dessa maneira, cada campo ocupa no arquivo um **tamanho fixo**, pré-determinado.

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

Nos dados dispostos anteriormente, cada campo possui um **tamanho fixo**, que foi definido no momento da criação do campo. Veja a seguir um exemplo de declaração desses campos:

```
char last[10];
char first[10];
char city[15];
char state[2];
char zip[9];
```

Obs.: Um dos problemas desse método é que o espaço alocado (e não usado) aumenta desnecessariamente o tamanho do arquivo, gerando desperdício de espaço em disco.

Outro método seria começar cada campo com um **indicador de tamanho** desse campo. Dessa forma, o tamanho de cada campo é armazenado imediatamente antes do dado. Observe o exemplo:

```
05Maria05Rua 10312310São Carlos
04João05Rua A0325509Rio Claro
05Pedro06Rua 10025610Rib. Preto
```

ORGANIZAÇÃO EM REGISTROS

Um **registro** é um **conjunto de campos agrupados**, os quais estão **logicamente associados a uma mesma entidade**. Permite a representação de um arquivo em um nível de organização mais alto.

Obs.: Também podemos aplicar diferentes métodos para organizar os registros. Um deles seria **forçar os registros para um tamanho fixo**. Assim, todos os registros teriam o mesmo número de *bytes*.

DICA

Podemos ter **registros de tamanho fixo com campos de tamanho fixo** e **registros de tamanho fixo com campos de tamanho variável**.

Registro de tamanho fixo e campos de tamanho fixo:			
Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

Registro de tamanho fixo e campos de tamanho variável:				
Maria	Rua 1	123	São Carlos	← Espaço vazio →
João	Rua A	255	Rio Claro	← Espaço vazio →
Pedro	Rua 10	56	Rib. Preto	← Espaço vazio →

Outro método é **forçar todos os registros a conterem um número fixo de campos**. Aqui, cada registro contém um **número fixo de campos** e o tamanho do registro pode ser de **tamanho variável**. Neste caso, os campos seriam **separados por delimitadores**.

Registro com número fixo de campos:

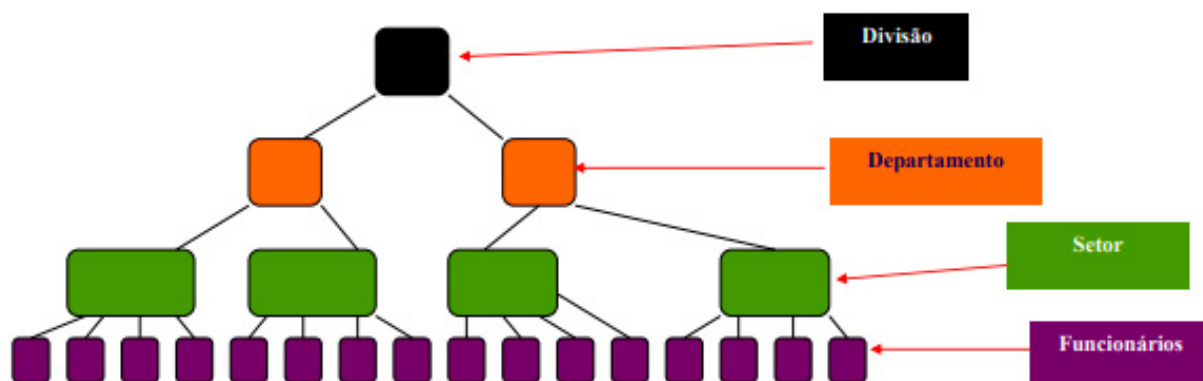
Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua 10|56|Rib. Preto|

4. TIPOS DE BANCO DE DADOS

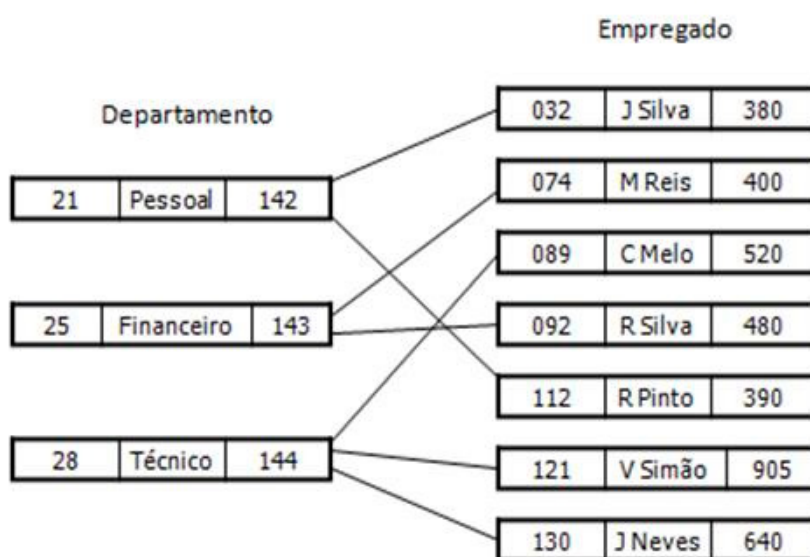
Um **modelo de banco de dados** mostra a **estrutura lógica de um banco de dados**, incluindo as relações e restrições que determinam como os dados podem ser armazenados e acessados. A maioria dos modelos de dados pode ser representada por um **diagrama de banco de dados**.

Há vários tipos de **modelos de dados**:

- **Hierárquico** – Os **dados são organizados hierarquicamente em uma estrutura de árvore**, de modo que cada dado só possui apenas um possuidor (dono). **Os nós das hierarquias contêm ocorrências de registros**, em que **cada registro é uma coleção de campos** (atributos), **cada um contendo apenas uma informação**. O registro da hierarquia que precede a outros é o **registro-pai**, os outros são chamados de **registros-filhos**. **O relacionamento entre um registro-pai e vários registros-filhos possui cardinalidade 1:N**. Este modelo entrou em desuso. Um exemplo fácil de assimilar é a árvore genealógica familiar.



- **Redes** – Os dados são organizados em uma estrutura de grafos direcionados, permitindo que várias tabelas sejam usadas por meio de **apontadores** (referências). A principal diferença entre o modelo de redes e o hierárquico é que, **no modelo de redes, não existe restrição hierárquica**, ou seja, **um mesmo registro pode estar envolvido em várias associações**. Os registros são **organizados em grafos** em que aparece **um único tipo de associação** (set) que **define uma relação 1:N** entre dois tipos de registros: **proprietário** e **membro**.

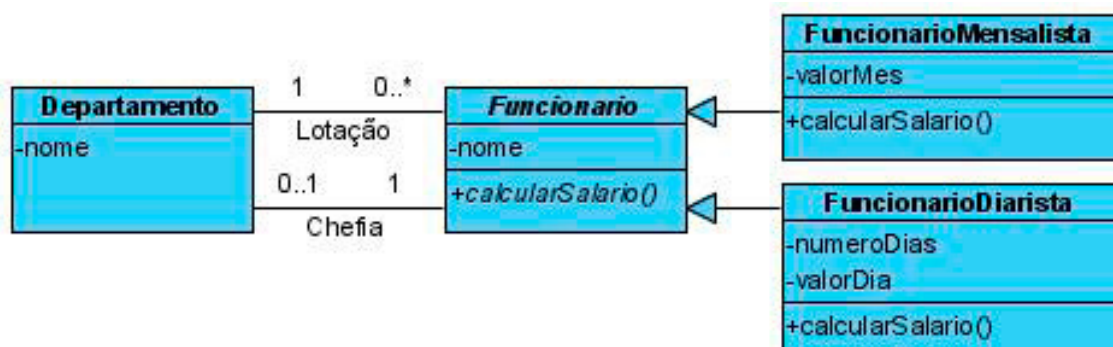


- **Relacional** – Caracterizado por **organizar informações em tabelas** e **depende da interação entre colunas e linhas**. O tipo de dado (nome, ID, data de nascimento etc.) fica na coluna, e a descrição das informações fica contida nas linhas. É ideal para utilizar quando precisa-se **armazenar informações tabulares** de pouca complexidade e que precisam ser recuperadas rapidamente. Essa é a forma de armazenamento mais utilizada, pois oferece alta confiabilidade. Formalmente definido por Edgard Frank Codd em 1970, o objeto básico tratado pelo modelo relacional é a **“entidade”** ou **“relação”**. **Uma entidade equivale ao conceito matemático de conjunto**, ou seja, um **agrupamento de elementos**.

Porém, é necessária uma **estrutura de relacionamento entre tabelas** para que as informações possam ser recuperadas. A linguagem usada para manipular essas informações é o **SQL (Structured Query Language)**.

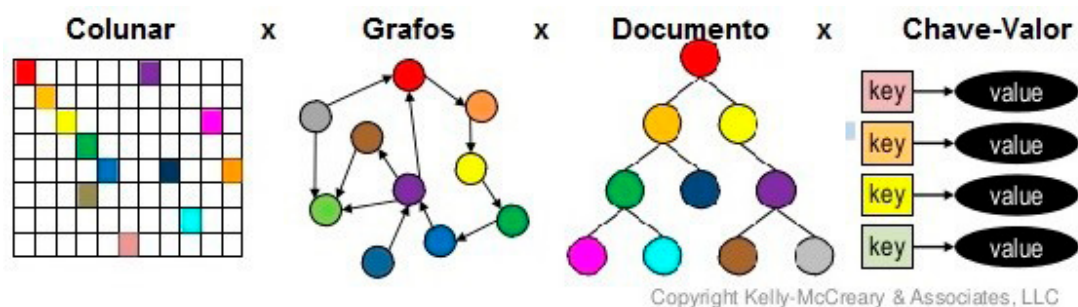


- **Orientado a Objetos** – É baseado no paradigma da **programação orientada a objetos**. Nesse modelo de banco de dados, as funcionalidades de **orientação a objetos** são **integradas às do banco de dados**. Cada informação é **armazenada na forma de objetos**, ou seja, utiliza a estrutura de dados denominada **orientação a objetos**, a qual permeia as linguagens mais modernas. O gerenciador do banco de dados para um **orientado a objeto** é referenciado por vários como **ODBMS** ou **OODBMS**. Este modelo define o banco de dados como uma **coleção de objetos**, ou **elementos de software reutilizáveis**, com **recursos e métodos associados**.



- **NoSQL** – O termo **NoSQL** foi primeiramente utilizado em 1998 como o nome de um **banco de dados não relacional de código aberto**. O movimento **NoSQL** é **completamente distinto do modelo relacional** e, portanto, deveria ser mais apropriadamente chamado **"NoREL"** ou algo que produzisse o mesmo efeito. Toda a informação necessária estará **agrupada no mesmo registro**, ou seja, em vez de você ter o relacionamento entre várias tabelas para formar uma informação, **o dado estará armazenado no mesmo registro**, em

sua totalidade. Os **bancos de dados relacionais** escalam, mas quanto maior o tamanho, mais custoso se torna essa escalabilidade, seja pelo custo de novas máquinas, seja pelo aumento de especialistas nos bancos de dados utilizados. Por sua vez, os **bancos de dados não relacionais** permitem uma escalabilidade mais barata e menos trabalhosa, pois não exigem máquinas extremamente poderosas. Basicamente, os **tipos de bancos de dados NoSQL** são: **Colunares**; **Documentos**; **Grafos**; **Chave-valor**.



5. BANCOS DE DADOS NoSQL

O formato não relacional atende uma parcela de dados que **não podem ser inseridos e acessados por meio de tabela**, como imagens. É muito valorizado pela sua alta performance e por manter todos os registros em um único lugar.

Portanto, **não é necessário criar um sistema de relacionamento entre informações**, como os bancos de dados relacionais. Como as informações não estão associadas entre si, é mais fácil fazer alterações e exclusões no conteúdo. A linguagem utilizada é o **NoSQL (Not Only SQL)**.

BANCOS DE DADOS COLUNARES

Os **bancos de dados colunares** possuem o **armazenamento orientado a colunas**, o que influencia significativamente na sua performance, já que diminuem a quantidade de dados que você precisará carregar no disco.

DICA

Esses bancos de dados utilizam-se de **tabelas para representar entidades** e seus dados são gravados em discos e agrupados por colunas separadas.

Os bancos de dados relacionais normalmente **guardam os registros das tabelas contigualmente no disco**.

Por exemplo, caso se queira guardar id, nome e endereço de usuários em um sistema de cadastro, os registros seriam: Id1, Nome1, Endereço1; Id2, Nome2, Endereço2.

Essa estrutura torna a escrita muito rápida, pois todos os dados de um registro são colocados no disco com uma única escrita no banco. Essa estrutura também é eficiente caso se queira ler registros inteiros. Mas para situações em que se quer **ler algumas poucas colunas de muitos registros**, essa estrutura é pouco eficiente, pois muitos blocos do disco terão de ser lidos.

DICA

Para esses casos em que se quer **otimizar a leitura de dados estruturados**, os **bancos de dados colunares** são mais interessantes, pois eles **guardam os dados contiguamente por coluna**.



Podemos listar os seguintes bancos de dados que são orientados a coluna: Hadoop, Cassandra, Hypertable, Amazon SimpleDB etc.

BANCOS DE DADOS ORIENTADOS A GRAFOS

Com o **armazenamento de documentos em forma de grafos**, os dados são predispostos no **formato de arcos conectados por arestas**.

A ideia do **banco de dados orientados a grafos** é **representar os dados e/ou o esquema dos dados como grafos dirigidos**, ou como **estruturas que generalizem a noção de grafos**.

O **modelo orientado a grafos** possui **três componentes básicos**: os **nós** (são os vértices do grafo), os **relacionamentos** (são as arestas) e as **propriedades** (ou atributos) **dos nós e relacionamentos**.



Criado especificamente para possibilitar o **armazenamento de relacionamentos e a navegação**, o modelo utiliza **nós para armazenar entidades de dados e bordas para armazenar os relacionamentos entre estas entidades**.

DICA

O **banco de dados orientado a grafos** é voltado para trabalho com **dados altamente conectados** e com **relacionamentos dinâmicos em grandes volumes**, sendo bastante utilizados para aplicações como redes sociais, sistema de recomendações e outros tipos de dados mais complexos.

BANCOS DE DADOS ORIENTADOS A DOCUMENTOS

Os documentos dos **bancos de dados orientados a documentos** são **coleções de atributos e valores** em que **um atributo pode ser multivalorado**. Em geral, os bancos de dados orientados a documento não possuem esquema, ou seja, os documentos armazenados não precisam possuir estrutura em comum.

Também conhecido como **armazenamento por documentos**, os **bancos de dados orientados a documentos** tratam de um modelo de banco de dados projetado para **gerenciar, armazenar e recuperar informações orientadas a documentos, não carecendo de colunas pré-montadas**, e é um modelo eficiente para o **trabalho com dados não estruturados** (que não podem ser organizados em tabelas).

Alguns bancos que utilizam esse padrão são: MongoDB, CouchDB, RavenDB etc.

Document 1	Document 2	Document 3
<pre>{ "id": "1", "name": "John Smith", "isActive": true, "dob": "1964-30-08" }</pre>	<pre>{ "id": "2", "fullName": "Sarah Jones", "isActive": false, "dob": "2002-02-18" }</pre>	<pre>{ "id": "3", "fullName": { "first": "Adam", "last": "Stark" }, "isActive": true, "dob": "2015-04-19" }</pre>

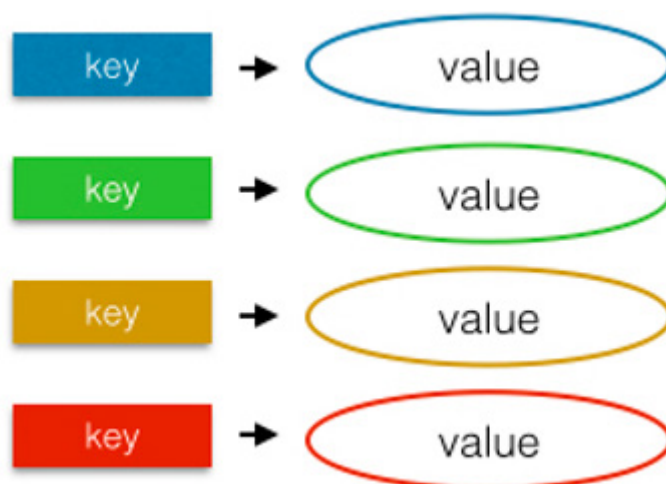
Essa estrutura torna a **escrita muito rápida**, pois **todos os dados de um registro são colocados no disco com uma única escrita no banco**. Essa estrutura também é eficiente caso se queira **ler registros inteiros**. Mas para situações em que se quer ler algumas poucas colunas

de muitos registros, essa estrutura é pouco eficiente, pois muitos blocos do disco terão de ser lidos.

BANCOS DE DADOS DO TIPO CHAVE-VALOR

O **banco de dados do tipo chave-valor** é formado apenas por **pares de chaves com valores associados**, permitindo **obter os valores quando uma consulta é realizada a uma chave**. A **chave** funciona como um **identificador exclusivo**. Por conta de sua simplicidade, não é empregado em aplicações mais complexas.

Uma de suas principais aplicações são os **sistemas de armazenamento embarcados e dados em cache**.



Alguns bancos que utilizam esse padrão são: DynamoDb, Couchbase, Riak, Azure Table Storage, Redis, Tokyo Cabinet, Berkeley DB etc.

Um **banco de dados de chave-valor**, ou **armazenamento de chave-valor**, é um paradigma de armazenamento de dados projetado para **armazenar, recuperar e gerenciar matrizes associativas** e uma estrutura de dados mais comumente conhecida hoje como **dicionário** ou **tabela hash**.

6. ARQUITETURA DE BANCO DE DADOS

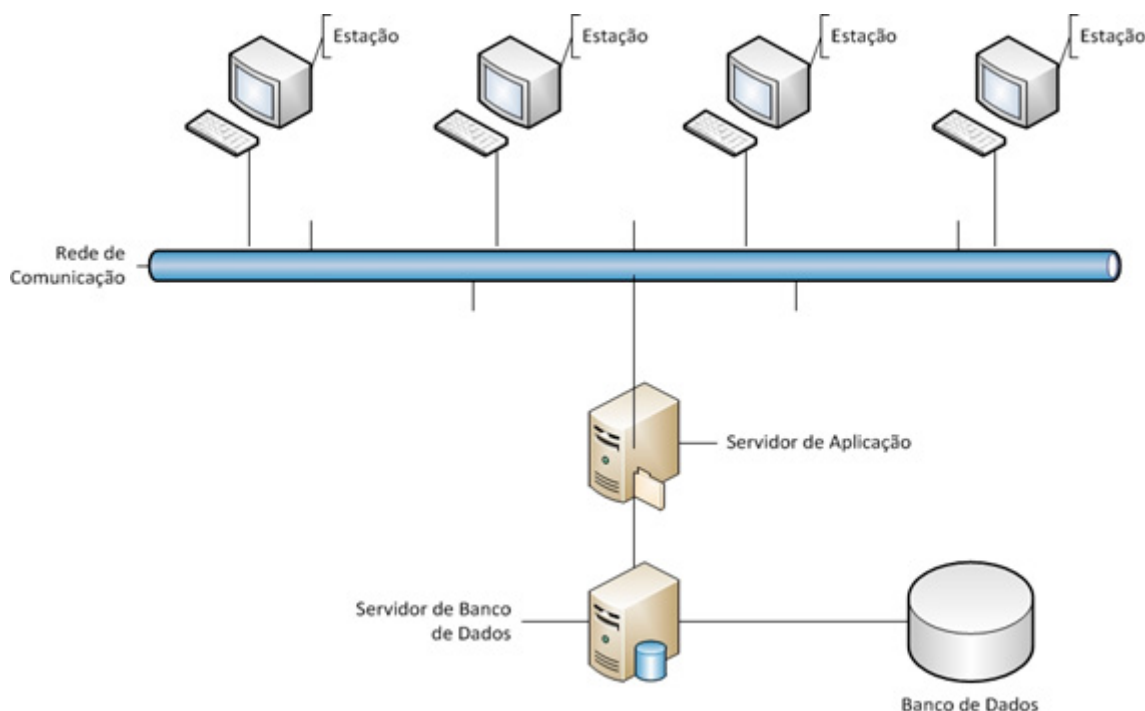
A **arquitetura de um banco de dados** está diretamente ligada ao *hardware* e ao sistema computacional existentes na estrutura em questão. O poder de processamento, a estrutura de rede, a memória e diversos outros componentes, são características que fazem parte da análise de toda a **arquitetura de um banco de dados**.

Sendo assim, serão abordados **quatro tipos de arquitetura: centralizada, cliente-servidor, paralela e distribuída**. Cada um destes tipos teve grande importância na história da evolução dos bancos de dados.

ARQUITETURA CENTRALIZADA

Os primeiros bancos de dados foram implementados baseados na **arquitetura centralizada**. Na década de 1970, eles tinham a implementação centralizada, normalmente em **mainframes**. Para os usuários acessarem os dados, eram utilizados terminais sem capacidade de processamento. Esses computadores se conectavam em *mainframes* (nos quais era feito todo o processamento) e ali obtinham os dados que necessitavam.

Na **arquitetura centralizada**, toda a massa de dados fica em um **único local**, em um único servidor, e é acessado pelas demais aplicações ou clientes. Neste caso, ele assume toda a capacidade de armazenamento e resposta. É um modelo que exige grande poder de processamento do servidor e excelente desempenho do banco sistema gerenciador de banco de dados.



ARQUITETURA CLIENTE-SERVIDOR

Conforme os computadores pessoais foram evoluindo e o seu custo se tornou menor, a arquitetura centralizada, como era conhecida, teve o seu uso diminuído. Os computadores pessoais substituíram os terminais de acesso e começaram a desempenhar novos papéis. O gerenciamento da interface que antes era responsabilidade do servidor, agora passou a ser gerenciado por clientes.

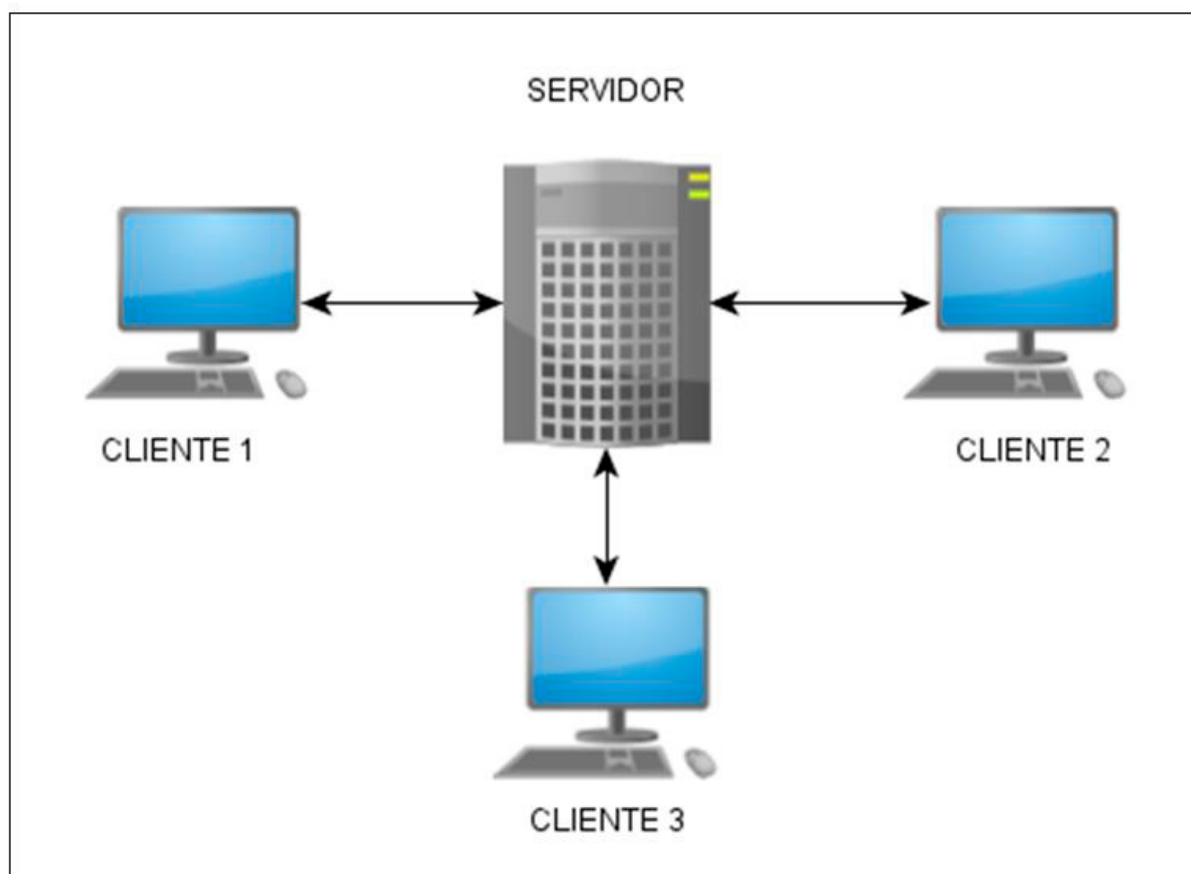


O PULO DO GATO

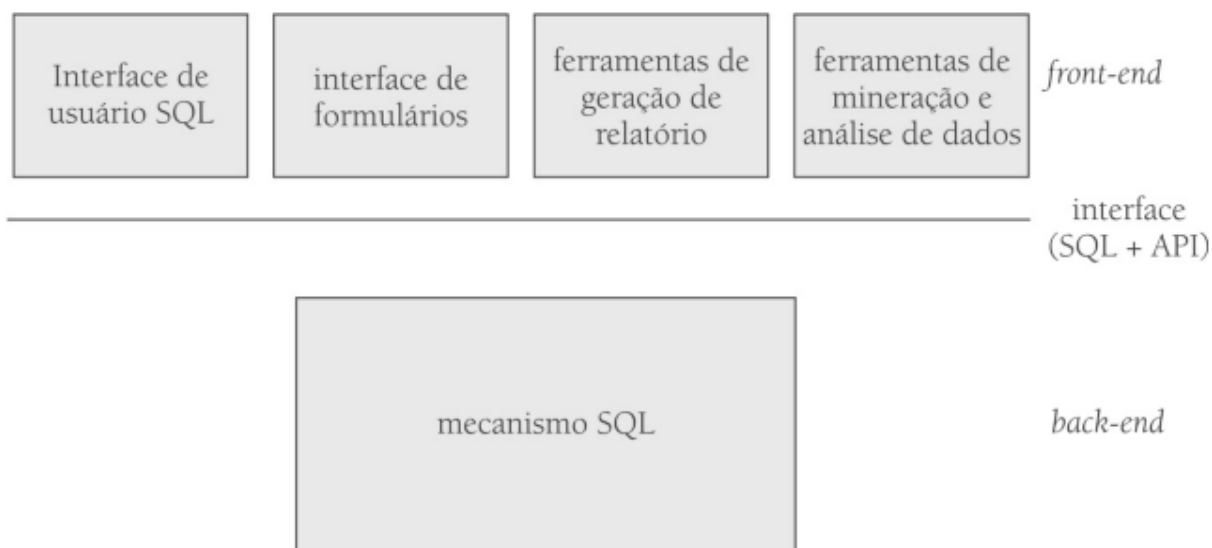
Em resumo, os **sistemas centralizados** atualmente respondem requisições que são feitas **pe-
los sistemas clientes**. Esse é o conceito de **cliente-servidor**.

Em sua forma mais simples, um **banco de dados cliente-servidor (C/S)** **divide o proces-
samento do banco de dados entre dois sistemas**: o cliente (geralmente um PC) executando
a aplicação do banco de dados, e o servidor do banco de dados que executa todo o SGBD ou
parte dele.

Na **arquitetura cliente-servidor**, o **cliente executa as tarefas do aplicativo**, ou seja, fornece
a interface do usuário (tela, e processamento de entrada e saída). O **servidor de banco de da-
dos executa as consultas no SGBD e retorna os resultados ao cliente**.



A funcionalidade oferecida pelos servidores de **bancos de dados cliente-servidor** pode ser
dividida em duas partes: o **front-end** e o **back-end**. O **back-end** basicamente **controla as estru-
turas de acesso ao dado**, enquanto o **front-end** faz a **interface com o usuário final**. A figura a
seguir expõe esse cenário:



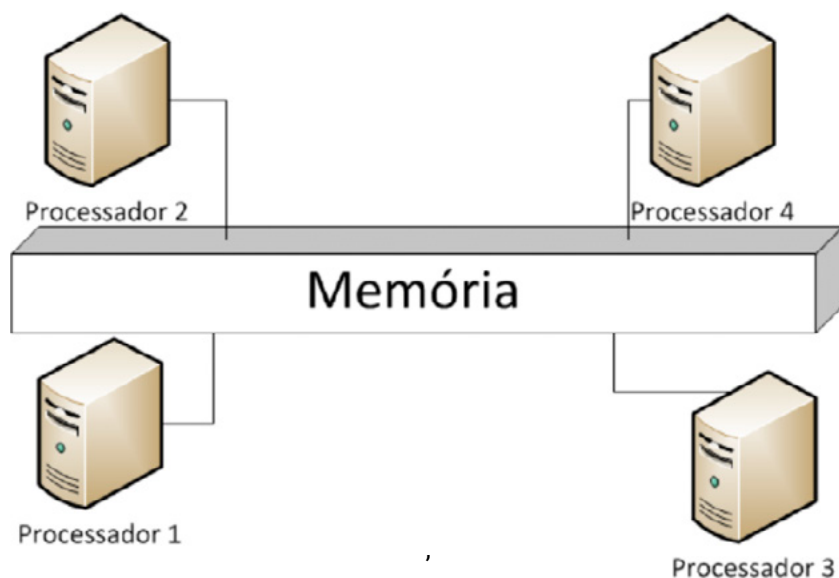
ARQUITETURA PARALELA

É uma arquitetura que tem como objetivo o **compartilhamento de recursos para processamento de dados**.

Os **sistemas paralelos** melhoram as **velocidades de processamento e E/S**, usando **várias CPUs e discos em paralelo**.

Obs.: As máquinas paralelas estão se tornando cada vez mais comuns, tornando o estudo de sistemas de banco de dados paralelos cada dia mais importante.

No **processamento paralelo**, muitas **operações são realizadas simultaneamente**, ao contrário do processamento serial, em que as etapas computacionais são realizadas sequencialmente.



Tradicionalmente, o *software* é escrito para ser **executado sequencialmente**. Para resolver um problema específico, um algoritmo é construído e implementado como um **fluxo serial de instruções**. Tais instruções são então executadas por uma **unidade central de processamento** (CPU) de um computador. **Somente uma instrução pode ser executada por vez**; é preciso sua finalização para que a próxima inicie.

Por outro lado, a **computação paralela** faz uso de **múltiplos elementos de processamento simultaneamente** para resolver um problema. Isso é possível ao **quebrar um problema em partes independentes**, de forma que **cada elemento de processamento possa executar sua parte do algoritmo simultaneamente com outros**.

Obs.: Os elementos de processamento podem ser diversos e incluir recursos como um único computador com múltiplos processadores, diversos computadores em rede, *hardware* especializado ou qualquer combinação dos anteriores.

ARQUITETURA DISTRIBUÍDA

Em um **sistema de banco de dados distribuído**, o **banco de dados é armazenado em vários computadores**. Os computadores em um **sistema distribuído** se **comunicam entre si através de diversos meios de comunicação**, como redes de alta velocidade ou linhas telefônicas. Eles **não compartilham memória principal ou discos**. Os computadores em um **sistema distribuído** podem variar em tamanho e função, desde estações de trabalho até sistemas de *mainframe*.

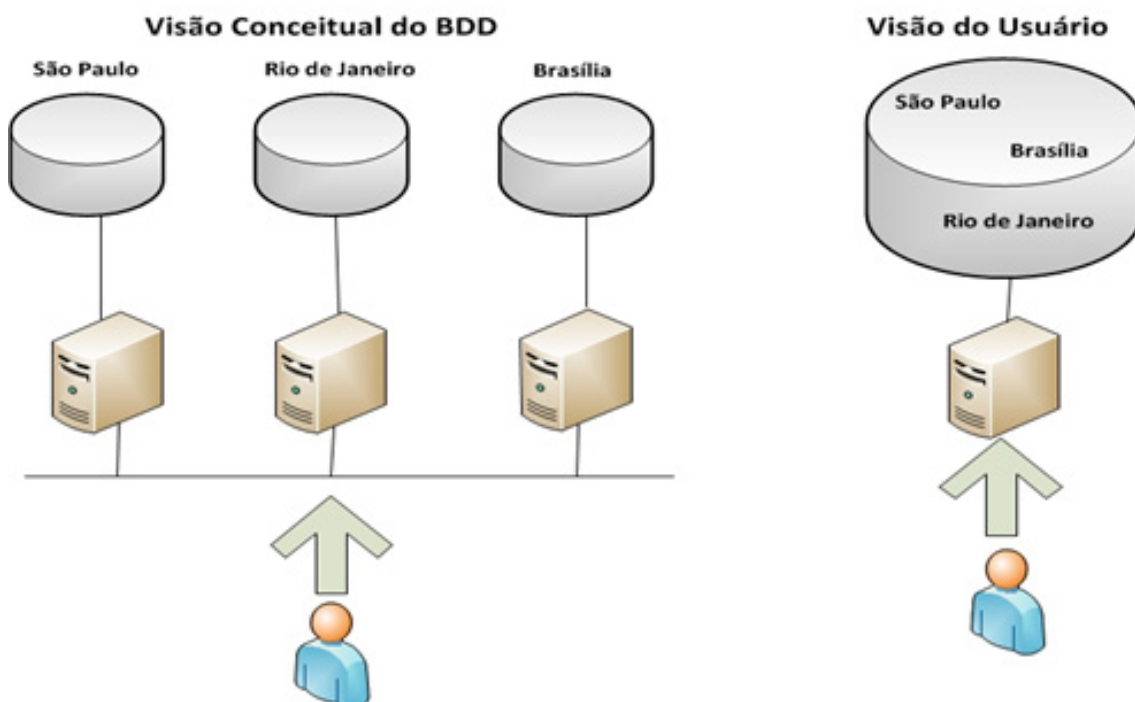


A definição mais básica de **sistemas distribuídos** é que **são máquinas diferentes que podem estar conectadas entre si através de uma rede de comunicações**.

DICA

Em um **sistema paralelo**, a tendência é que as **máquinas** fiquem **mais próximas**. No **sistema distribuído**, elas podem estar **geograficamente distantes**.

A **arquitetura de um banco de dados distribuído** é **totalmente transparente para os usuários** que utilizam aplicações nessa arquitetura distribuída, ou seja, **para o usuário, os dados estão centralizados em um único servidor**, mas, na verdade, eles podem estar “**espalhados**”, **distribuídos em vários locais fisicamente separados**.



RESUMO

Armazenamento de Dados

Os diversos meios de armazenamento podem ser organizados em uma **hierarquia**, conforme mostrado na figura, de acordo com sua velocidade e seu custo.

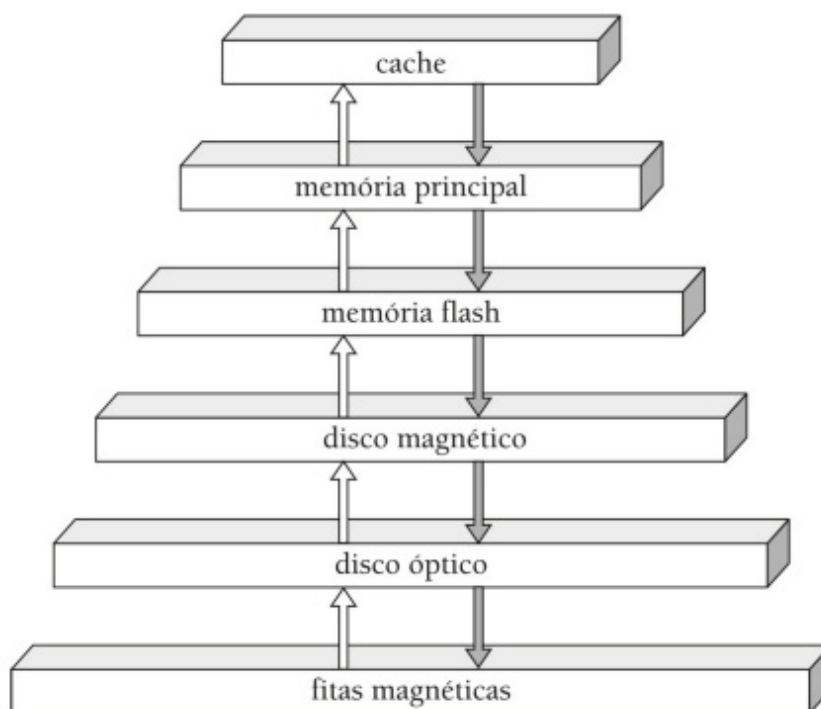


Figura 5 – Hierarquia de meios de armazenamento segundo custo-benefício

DICA

Os níveis mais altos são caros, mas são rápidos. Ao **descer-mos na hierarquia**, o **custo por bit diminui**, enquanto o **tempo de acesso aumenta**. Essa compensação (*trade-off*) é razoável; se determinado sistema de armazenamento fosse mais rápido e mais barato do que outro – outras propriedades sendo iguais –, não haveria motivo para usar a memória mais lenta, mais dispendiosa.

Os meios de armazenamento mais rápidos – por exemplo, **cache** e **memória principal** – são considerados **armazenamento primário**. Os meios no próximo nível da **hierarquia** – por exemplo, **discos magnéticos** – são considerados **armazenamento secundário**. Os níveis mais baixos da **hierarquia** – por exemplo, **fita magnética** e **discos ópticos** – são considerados **armazenamento terciário**.

Além da velocidade e do custo dos diversos sistemas de armazenamento, há também a questão do **armazenamento volátil**. O **armazenamento volátil** perde seu conteúdo quando a energia ao dispositivo é removida. Na hierarquia mostrada na figura anterior, os sistemas de armazenamento da memória principal para cima são **voláteis**, enquanto os sistemas de armazenamento abaixo da memória principal são **não voláteis**.

Obs.: | Os dados precisam ser gravados no armazenamento não volátil por segurança.

Storage

Storage é o termo em inglês para **armazenamento**. Geralmente utilizado para designar **sistemas de armazenamento de dados** em servidores, computadores e eletrodomésticos, um **storage** também pode ser traduzido como uma **área de armazenamento para pequenos objetos** ou um **espaço dentro de um armazém**.

Esses sistemas de armazenamento de grande porte são divididos em três grandes grupos: **DAS, NAS e SAN**.

Os **DAS (Direct Attached Storage)** são quaisquer **unidades conectadas diretamente a computadores**, sejam eles servidores, *notebooks* ou estações de trabalho.

Diferentemente dos sistemas de armazenamento direto, os **NAS (Network Attached Storage)** são equipamentos que possuem sistemas operacionais incorporados.

Obs.: | Isso significa que esse tipo de **storage** não precisa de um servidor ou *host* para funcionar, sendo **conectado diretamente em qualquer rede local**.

SAN (Storage Area Network) é uma **rede corporativa de dados** voltada para **conectar e manter disponíveis servidores e storages via LAN** dentro de um ambiente seguro, redundante e de alta performance.

RAID

Diversas técnicas de organização de disco, coletivamente chamadas de **redundant arrays of independent disks (RAID)**, têm sido propostas para alcançar melhor desempenho e confiabilidade.

Existem vários tipos de **RAID** que podem ser realizados com **discos de armazenamento**. Os arranjos de disco **RAID** podem ser criados e gerenciados por meio de um *software* ou de um *hardware* próprio para esse fim. Em linhas gerais, temos os seguintes **RAIDs**:

- **RAID 0:** consiste em **distribuir os dados a serem armazenados no sistema de armazenamento**, gravando a informação particionada em diversos discos rígidos de forma simultânea.

- **RAID 1:** também conhecido como **disk mirror**, replicação ou espelhamento de disco. **Escreve as informações de forma simultânea nos discos.**
- **RAID 5:** é conhecido como **“Strip Set com Paridade”**. **Bits de paridade são criados e acrescentados aos dados, escritos de forma alternada em todos os discos.**
- **RAID 6:** é um arranjo com características próximas ao **RAID 5**, mas **acrescentando dupla paridade às informações gravadas.**

Organização de Arquivo

A organização do arquivo determina os métodos de acesso que podem ser utilizados na recuperação dos registros.

Um banco de dados é **mapeado em diversos arquivos diferentes**, mantidos pelo sistema operacional básico. **Esses arquivos residem permanentemente nos discos.**

Um **arquivo** é **organizado logicamente como uma sequência de registros, mapeados para blocos de disco.**

Dado um conjunto de registros, a próxima pergunta é: como organizá-los em um arquivo?

São possíveis maneiras de **organizar registros em arquivos**:

- **Organização de arquivos em heap** (pilha): um registro pode ser colocado em qualquer lugar no arquivo onde exista espaço para acomodá-lo. Não existe ordenação de registros. Normalmente, existe um único arquivo para cada relação;
- **Organização sequencial de arquivos:** os registros são armazenados em ordem sequencial, de acordo com o valor de uma “chave de busca” de cada registro;
- **Organização de arquivos com hashing:** uma função de *hash* é calculada sobre algum atributo de cada registro. O resultado da função de *hash* especifica em que bloco do arquivo o registro deve ser colocado. Está bastante relacionada às estruturas de indexação.

Métodos de Acesso

Para recuperarmos um registro em um arquivo, podemos utilizar acesso **sequencial** ou **direto**.

O **método sequencial de acesso** é o mais tradicional e consiste em efetuar a **leitura dos registros, um após o outro**, comparando o **argumento de pesquisa** com o valor do campo chave (primária ou secundária) no registro corrente, até encontrar os registros desejados ou o final do arquivo.

O **método direto de acesso** consiste em recuperar o(s) registro(s) desejado(s), **sem a necessidade de efetuar a leitura dos registros que o(s) antecede(m)**, o que pode ser feito por meio de um **índice** ou com o auxílio de um **algoritmo de randomização que localiza o registro**, calculando a posição ocupada pelo registro no disco com base no valor do argumento de pesquisa, que deve ser um campo numérico.

Organização Sequencial

A **organização sequencial** caracteriza-se pela existência de uma **chave de ordenação**. Essa chave determina a **ordem em que os registros são armazenados**, e pode ser simples ou composta por dois ou mais campos. Geralmente, coincide com a **chave primária**, mas não obrigatoriamente.

Obs.: | A **organização sequencial** somente permite o **acesso sequencial**.

Organização Serial

Os registros são **armazenados de acordo com a ordem de inclusão**. O arquivo **não possui chave de ordenação**, portanto, não existe preocupação com a ordem de armazenamento dos registros.

Obs.: | A **organização serial** somente permite o **acesso sequencial**.

Organização Indexada

Os registros são armazenados em um arquivo de dados com **organização serial**, e para cada campo (ou combinação deles) por meio do qual se deseja obter **acesso direto (indexado)**, deve-se criar um **arquivo de índice (processo de indexação)**. Um mesmo arquivo de dados pode possuir diversos arquivos de **índice** a ele associados.

O **arquivo de índice** é composto basicamente por **duas colunas**. A primeira corresponde ao campo utilizado no **processo de indexação (endereço lógico)** e a segunda **armazena um valor (endereço físico)**, que serve como referência para que o gerenciador de arquivos localize o registro no disco magnético.

Tipos de Banco de Dados

Há vários tipos de **modelos de dados**:

- **Hierárquico** – Os **dados são organizados hierarquicamente em uma estrutura de árvore**, de modo que cada dado só possui apenas um possuidor (dono);
- **Redes** – Os **dados são organizados em uma estrutura de grafos direcionados**, permitindo que várias tabelas sejam usadas por meio de **apontadores** (referências);
- **Relacional** – Caracterizado por **organizar informações em tabelas e depender da integração entre colunas e linhas**;
- **Orientado a Objetos** – É baseado no paradigma da **programação orientada a objetos**. Nesse modelo de banco de dados, as funcionalidades de **orientação a objetos** são **integradas às do banco de dados**;

- **NoSQL** – O termo **NoSQL** foi primeiramente utilizado em 1998 como o nome de um **banco de dados não relacional de código aberto**. O movimento **NoSQL** é **completamente distinto do modelo relacional** e, portanto, deveria ser mais apropriadamente chamado “**NoREL**” ou algo que produzisse o mesmo efeito. Basicamente, os **tipos de bancos de dados NoSQL** são: **Colunares; Documentos; Grafos; Chave-valor**.



O PULO DO GATO

O formato não relacional atende uma parcela de dados que **não podem ser inseridos e acessados por meio de tabela**, como imagens.

- **Bancos de Dados Colunares:** Possuem o **armazenamento orientado a colunas**. Esses bancos de dados utilizam-se de **tabelas para representar entidades** e seus **dados são gravados em discos e agrupados por colunas separadas**;
- **Bancos de Dados Orientados a Grafos:** Os dados são predispostos no **formato de arcos conectados por arestas**. A ideia é **representar os dados e/ou o esquema dos dados como grafos dirigidos** ou como **estruturas que generalizem a noção de grafos**;
- **Bancos de Dados Orientados a Documentos:** São **coleções de atributos e valores** em que **um atributo pode ser multivalorado**. Trata-se de um modelo projetado para **gerenciar, armazenar e recuperar informações orientadas a documentos, não carecendo de colunas pré-montadas**, e é um modelo eficiente para o **trabalho com dados não estruturados** (que **não podem ser organizados em tabelas**);
- **Bancos de Dados do Tipo Chave-Valor:** É formado apenas por **pares de chaves com valores associados**, permitindo **obter os valores quando uma consulta é realizada a uma chave**. A **chave** funciona como um **identificador exclusivo**.

Arquitetura de Banco de Dados

A **arquitetura de um banco de dados** está diretamente ligada ao *hardware* e ao sistema computacional existentes na estrutura em questão. Sendo assim, serão abordados **quatro tipos de arquitetura**:

- **Arquitetura Centralizada:** Toda a massa de dados fica em um **único local**, em um único servidor, e é acessado pelas demais aplicações ou clientes;
- **Arquitetura Cliente-Servidor:** Um **banco de dados cliente-servidor (C/S)** **divide o processamento do banco de dados entre dois sistemas**: o **cliente** (geralmente um PC) executando a aplicação do banco de dados, e o **servidor** do banco de dados que executa todo o **SGBD** ou parte dele;

- **Arquitetura Paralela:** É uma arquitetura que tem como objetivo o **compartilhamento de recursos para processamento de dados**. No **processamento paralelo**, muitas **operações são realizadas simultaneamente**, ao contrário do processamento serial, em que as etapas computacionais são realizadas sequencialmente;
- **Arquitetura Distribuída:** O **banco de dados é armazenado em vários computadores**. Os computadores se **comunicam entre si através de diversos meios de comunicação**, como redes de alta velocidade ou linhas telefônicas. Eles **não compartilham memória principal ou discos**.

QUESTÕES DE CONCURSO

001. (FGV/SEDUC-AM/ASSISTENTE TÉCNICO/2014) Sobre os componentes de *hardware* de um microcomputador, assinale a opção que indica a principal função da memória cache.

- a) Agilizar o processamento.
- b) Converter formatos de dados.
- c) Executar a multitarefa preemptiva.
- d) Armazenar dados de forma permanente.
- e) Implementar o mecanismo da memória virtual.

002. (FGV/DETRAN-RN/PROGRAMADOR DE COMPUTADOR/2010) Assinale a alternativa que, de forma usual, NÃO condiz com uma característica da Memória Principal (MP) de um computador:

- a) Tempo de acesso mais rápido que o dos discos rígidos.
- b) Capacidade menor que a dos discos rígidos.
- c) É composta, predominantemente, de memória volátil.
- d) Armazena a instrução que será acessada pela UCP.
- e) Armazena as informações de forma permanente, podendo estas serem recuperadas após a interrupção da alimentação de energia elétrica.

003. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Na comparação entre *drives* HD e SSD, considere os seguintes fatores:

- I – custo por GB armazenado;
- II – velocidade de leitura;
- III – capacidade máxima de armazenagem;
- IV – resistência a choques e durabilidade.

Está correto concluir que os drives com tecnologia HD apresentam vantagens:

- a) somente no fator I.
- b) somente nos fatores I e III.
- c) somente no fator IV.
- d) somente nos fatores I, II e III.
- e) em todos os fatores.

004. (FGV/ALERJ/ESPECIALISTA LEGISLATIVO – TECNOLOGIA DA INFORMAÇÃO/2018) Sobre tecnologias de armazenagem, analise as afirmativas a seguir:

- I – A arquitetura de armazenamento DAS requer uma infraestrutura de rede ligando os servidores aos dispositivos de armazenamento de dados.
- II – As redes usadas na arquitetura SAN são as que permitem trocas de arquivos entre seus nós, como é o caso das redes locais (LANs).

III – A tecnologia NAS permite o compartilhamento de arquivos entre servidores, mesmo com diferentes sistemas operacionais.

Está correto o que se afirma em:

- a) somente I.
- b) somente II.
- c) somente III.
- d) somente I e II.
- e) I, II e III.

005. (FGV/COMPESA/ANALISTA DE GESTÃO – ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) Sobre as formas de organização de um sistema de armazenamento de dados corporativos, analise as afirmativas a seguir.

I – Um DAS (*Direct Attached Storage*) é um sistema de armazenamento que não é compartilhado e nenhum outro dispositivo na rede pode ter acesso direto aos dados.

II – Um NAS (*Network Attached Storage*) apresenta como principal desvantagem a dependência do modelo cliente-servidor para comunicação e troca de dados, o que pode criar uma sobrecarga no compartilhamento de arquivos com vários servidores.

III – Um SAN (*Storage Area Network*) é projetado para conectar elementos de armazenamento, como conjuntos de discos e fitas, a múltiplos servidores por meio de dispositivos de rede de alta disponibilidade, confiabilidade e tolerância a falhas.

Está correto o que se afirma em:

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e III, somente.
- e) I, II e III.

006. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Um equipamento tem vários discos iguais. Para obter redundância dos dados, os discos sofrem um espelhamento, em que todos os dados são gravados em dois discos diferentes. Se um disco falhar ou for removido, os dados preservados no outro disco permitem a não descontinuidade da operação do sistema. Para obter esse esquema, a controladora de discos deve ser configurada para RAID:

- a) 0
- b) 1
- c) 4
- d) 5
- e) 6

007. (FGV/DPE-MT/ANALISTA DE SISTEMAS/2015) Relacione os diferentes níveis padrão de arranjo redundante de discos independentes (RAID) às respectivas características.

1. RAID 0
2. RAID 1
3. RAID 5
4. RAID 6

() Mantém o arranjo funcionando mesmo que dois dos discos falhem, mas requer pelo menos 4 discos.

() Tem como prioridade o desempenho de transferência. O arranjo fica indisponível se apenas um dos discos falhar.

() Balanceia desempenho e redundância, permitindo que o arranjo fique disponível mesmo que um dos discos falhe.

() Tem como prioridade a redundância de dados. O arranjo fica indisponível apenas se todos os discos falharem.

Assinale a opção que indica a relação correta, de cima para baixo.

- a) 3 – 2 – 4 – 1.
- b) 4 – 2 – 3 – 1.
- c) 1 – 4 – 2 – 3.
- d) 3 – 1 – 4 – 2.
- e) 4 – 1 – 3 – 2.

008. (FGV/DPE-MT/ASSISTENTE ADMINISTRATIVO/2015) Relacione os métodos de organizações de dados no armazenamento secundário às respectivas definições.

1. Organização sequencial de arquivos.
2. Organização direta de arquivos.
3. Organização indexada de arquivos.

() Os registros são armazenados fisicamente em uma determinada ordem.

() Os registros são armazenados por meio de um identificador exclusivo.

() Os registros podem ser acessados sequencialmente ou diretamente.

Assinale a opção que indica a relação correta, de cima para baixo.

- a) 1, 2 e 3.
- b) 1, 3 e 2.
- c) 2, 1 e 3.
- d) 2, 3 e 1.
- e) 3, 1 e 2.

009. (FGV/AL-MT/ANALISTA DE SISTEMAS – PROGRAMADOR/2013) As alternativas a seguir apresentam características dos arquivos *Heap*, à exceção de uma. Assinale-a.

- a) Inclusão de novos registros é muito eficiente.

- b) Organização de arquivos é a mais simples e básica.
- c) Pesquisa de registros envolve busca sequencial bloco a bloco.
- d) Para um arquivo com b blocos, uma busca envolve em média $b/2$ blocos.
- e) Os registros são posicionados de modo ordenado através da chave de classificação.

010. (FGV/TJ-GO/ANALISTA JUDICIÁRIO – ANÁLISE DE SISTEMAS – BANCO DE DADOS/2014) Considere as seguintes afirmativas comparativas entre métodos de busca baseados em árvores B e funções de *hashing*:

- I – A inserção de chaves não ordenadas é geralmente mais rápida em métodos de *hashing*.
- II – O número médio de acessos para localização de registros tende a ser menor para métodos baseados em *hashing*.
- III – Métodos de *hashing* não disponibilizam acesso sequencial às chaves em ordem crescente ou decrescente.

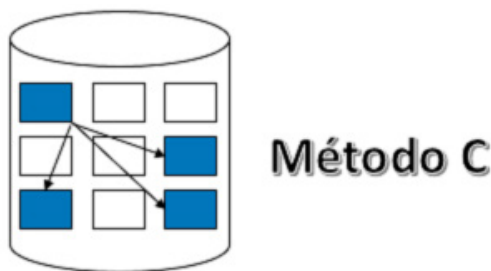
É correto concluir que:

- a) nenhuma está correta.
- b) somente I está correta.
- c) somente I e II estão corretas.
- d) somente II e III estão corretas.
- e) todas estão corretas.

011. (FGV/IBGE/ANALISTA – ANÁLISE DE SISTEMAS – SUPORTE OPERACIONAL/2016)

Em um sistema computacional, o sistema de arquivos possui diferentes estratégias para superar o problema de alocação de espaço em disco, de uma maneira em que ele possa ser explorado de forma eficiente e os arquivos nele contidos acessados rapidamente. Considere as imagens a seguir que representam 3 métodos de alocação utilizados pelo sistema de arquivos.





As ilustrações dos métodos A, B e C representam, respectivamente, os métodos de alocação:

- a) fila, encadeada, contígua.
- b) fila, indexada, direta.
- c) contígua, indexada, direta.
- d) contígua, encadeada, indexada.
- e) fila, sequencial, indexada.

012. (FGV/COMPESA/ANALISTA DE GESTÃO – ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) Em relação aos componentes do custo de execução de uma consulta SQL, analise as afirmativas a seguir.

I – O custo de acesso ao armazenamento secundário considera os custos de transferência (ler e gravar) blocos entre o armazenamento secundário e os buffers de memória principal.

II – O custo de memória diz respeito ao número de *buffers* de memória principal necessário para a execução de uma consulta.

III – O custo de comunicação diz respeito ao tempo de realizar as operações na memória em registros dentro dos buffers de dados.

Está correto o que se afirma em:

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e II, somente.
- e) I e III, somente.

013. (FGV/MPE-AL/TÉCNICO DO MINISTÉRIO PÚBLICO – GERAL/2018) O conjunto de programas responsável pelo gerenciamento de uma base de dados e que, entre outras funções, suporta uma linguagem de consulta, gera relatórios e disponibiliza uma interface para que os seus clientes possam incluir, alterar ou consultar dados, é chamado de:

- a) Banco de Dados Relacional (BDR).
- b) Dicionário de Dados (DD).
- c) Modelo Entidade Relacionamento (MER).
- d) Sistema de Suporte à Decisão (SSD).
- e) Sistema Gerenciador de Bancos de Dados (SGBD).

014. (FGV/TJ-RO/ANALISTA DE SISTEMAS/2015) Ao criar um banco de dados, o administrador (DBA) João analisa o perfil dos dados que serão gerados pela aplicação e percebe que eles devem ser agrupados no disco de acordo com as suas características, como:

- os maiores índices, que demandam mais recursos computacionais, podem ser localizados em discos mais velozes, como *Solid State Device* (SSD); e
- o armazenamento de dados históricos, que raramente são usados e não são muito críticos, pode ser realizado em discos mais lentos.

Em um Sistema Gerenciador de Banco de Dados (SGBD), para agrupar os dados em unidades lógicas de armazenamento, permitindo a definição da localização física dos arquivos correspondentes no disco, João deve definir:

- a) *schemas*.
- b) *data files*.
- c) *tablespaces*.
- d) *extents*.
- e) *data blocks*.

015. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Analise as afirmativas a respeito da classe de gerenciadores de bancos de dados, surgida em anos recentes, conhecida como NoSQL.

I – Mesmo sem suportar tabelas relacionais, baseiam-se em esquemas de dados previamente definidos;

II – Suas estruturas não permitem o uso de linguagens do tipo do SQL para recuperação de dados;

III – Garantem operações com as propriedades conhecidas pela sigla ACID;

IV – Privilegiam a rapidez de acesso e a disponibilidade dos dados em detrimento das regras de consistência das transações.

O número de afirmativas corretas é:

- a) uma
- b) duas
- c) três
- d) quatro
- e) cinco

016. (FGV/FIOCRUZ/TECNOLOGISTA EM SAÚDE – GEOPROCESSAMENTO/2010) Sobre banco de dados orientado a objeto, pode-se dizer que:

- a) é um banco de dados no qual os dados são guardados como propriedades de objetos, e os objetos são valores definidos segundo classes ou tipos de dados complexos.
- b) é formado por tabelas e registros de dados inseridos nas tabelas que estão diretamente vinculados ao dado primário.

- c) é um banco de dados que opera em rede e permite que várias tabelas sejam usadas simultaneamente por meio do uso de referências.
- d) diferencia-se de outros modelos de bancos de dados no sentido em que especifica a forma do resultado, e não o caminho para chegar a ele.
- e) é um conjunto de bancos de dados logicamente inter-relacionados, distribuídos em rede.

017. (FGV/MEC/ANALISTA DE SISTEMAS – ESPECIALISTA/2009) O Diagrama Entidade-Relacionamento, proposto por P. Chen, é uma ferramenta tipicamente utilizada para a elaboração do seguinte modelo de dados:

- a) físico
- b) interno
- c) externo
- d) conceitual
- e) hierárquico

018. (FGV/MEC/ADMINISTRADOR DE BANCO DE DADOS/2009) Em um Sistema Gerenciador de Banco de Dados – SGBD, os elementos de dados possuem uma estrutura lógica visualizada na forma de tabelas.

Nesse sentido, existem, inclusive, pacotes nos SGBDs que podem vincular elementos oriundos de várias tabelas, desde que essas compartilhem elementos comuns de dados.

Essa estrutura lógica de dados é conhecida como do tipo:

- a) em rede.
- b) relacional.
- c) em pilha.
- d) hierárquica.
- e) em árvore.

019. (FGV/AL-MA/TÉCNICO DE GESTÃO ADMINISTRATIVA – ANALISTA DE SISTEMAS/2013) Sobre sistemas de gerenciamento de bancos de dados, assinale a afirmativa incorreta.

- a) Em uma tabela chamada Carro, com os atributos Modelo, Ano, Cor e Placa, o atributo mais apropriado para servir como chave primária é Placa.
- b) O processo de aplicação de regras ao projeto de um banco de dados, com o intuito de permitir um acesso eficiente e armazenamento consistente dos dados, é denominado normalização do banco de dados, ou apenas normalização.
- c) O modelo hierárquico de banco de dados é o mais apropriado para dados que são representados por um relacionamento pai-filho, no qual cada registro pai pode estar relacionado com vários registros filho.

d) Uma Visão (View) de um banco de dados contém todas as informações sobre as entidades, atributos e tabelas. Seu objetivo é padronizar e unificar os termos utilizados nas variáveis utilizadas nos bancos de dados.

e) O “varchar” é uma cadeia de caracteres de tamanho indeterminado. Campos do tipo “varchar” podem ter qualquer tamanho, até um limite que varia de acordo com o SGBD.

020. (FGV/BADESC/ANALISTA DE SISTEMAS – BANCO DE DADOS/2010) Com relação à especificação e à manipulação de um atributo multivalorado pelos SGBDs, analise as afirmativas a seguir.

I – O modelo de dados relacional (especificado por Codd), não aceita este tipo de atributo, multivalorado. O modelo de dados relacional preconiza que este atributo deve ser processado pela implementação de uma tabela que possa contê-lo.

II – O modelo de dados relacional (especificado por Codd) especifica que para processar um atributo multivalorado deve antes normalizar a tabela em que este se encontra.

III – O modelo de objetos determina que todos os atributos de um objeto, inclusive os multivalorados, sejam encapsulados no objeto, permitindo, desta forma, a sua implementação e processamento.

Assinale:

- a) se somente a afirmativa I estiver correta.
- b) se somente a afirmativa II estiver correta.
- c) se somente a afirmativa III estiver correta.
- d) se somente as afirmativas I e II estiverem corretas.
- e) se todas as afirmativas estiverem corretas.

GABARITO

1. a
2. e
3. b
4. c
5. e
6. b
7. e
8. c
9. e
10. e
11. d
12. d
13. d
14. c
15. a
16. a
17. d
18. b
19. d
20. e

GABARITO COMENTADO

001. (FGV/SEDUC-AM/ASSISTENTE TÉCNICO/2014) Sobre os componentes de *hardware* de um microcomputador, assinale a opção que indica a principal função da memória cache.

- a) Agilizar o processamento.
- b) Converter formatos de dados.
- c) Executar a multitarefa preemptiva.
- d) Armazenar dados de forma permanente.
- e) Implementar o mecanismo da memória virtual.



Cache é uma memória utilizada entre a DRAM e a CPU (fisicamente posicionada no *chip* da CPU) para aumentar a velocidade de execução de programas. É do tipo SRAM (*static RAM*), cuja tecnologia permite velocidades de leitura e gravação muito maiores do que a velocidade da memória DRAM (utilizada como memória principal nos computadores atuais). Como a memória cache localiza-se próxima ao(s) núcleo(s) de execução da CPU, sua tecnologia permite velocidades maiores de leitura e gravação de dados. Ela armazena os dados usados com maior frequência pelo processador, fazendo com que os dados sejam lidos e gravados mais rapidamente pela CPU, o que leva a uma velocidade maior na execução dos programas.

Letra a.

002. (FGV/DETRAN-RN/PROGRAMADOR DE COMPUTADOR/2010) Assinale a alternativa que, de forma usual, NÃO condiz com uma característica da Memória Principal (MP) de um computador:

- a) Tempo de acesso mais rápido que o dos discos rígidos.
- b) Capacidade menor que a dos discos rígidos.
- c) É composta, predominantemente, de memória volátil.
- d) Armazena a instrução que será acessada pela UCP.
- e) Armazena as informações de forma permanente, podendo estas serem recuperadas após a interrupção da alimentação de energia elétrica.



A memória primária fornece geralmente uma ponte para as secundárias, mas a sua função principal é a de conter a informação necessária para o processador num determinado momento. Esta informação pode ser, por exemplo, os programas em execução. A memória principal também é volátil, ou seja, não mantém informações caso o fornecimento de energia seja interrompido.

Letra e.

003. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Na comparação entre *drives* HD e SSD, considere os seguintes fatores:

- I – custo por GB armazenado;
- II – velocidade de leitura;
- III – capacidade máxima de armazenagem;
- IV – resistência a choques e durabilidade.

Está correto concluir que os drives com tecnologia HD apresentam vantagens:

- a) somente no fator I.
- b) somente nos fatores I e III.
- c) somente no fator IV.
- d) somente nos fatores I, II e III.
- e) em todos os fatores.



O alto custo do *hardware* e a capacidade de armazenamento reduzida são desvantagens dos SSDs em relação aos HDDs, mas os primeiros são resistentes a choques (mecânicos) e possuem alta velocidade de transferência de dados (leitura). Em contrapartida, o HDD tem maior capacidade de armazenamento nos discos e consequentemente, menor custo por GB armazenado.

Letra b.

004. (FGV/ALERJ/ESPECIALISTA LEGISLATIVO – TECNOLOGIA DA INFORMAÇÃO/2018) Sobre tecnologias de armazenagem, analise as afirmativas a seguir:

- I – A arquitetura de armazenamento DAS requer uma infraestrutura de rede ligando os servidores aos dispositivos de armazenamento de dados.
- II – As redes usadas na arquitetura SAN são as que permitem trocas de arquivos entre seus nós, como é o caso das redes locais (LANs).
- III – A tecnologia NAS permite o compartilhamento de arquivos entre servidores, mesmo com diferentes sistemas operacionais.

Está correto o que se afirma em:

- a) somente I.
- b) somente II.
- c) somente III.
- d) somente I e II.
- e) I, II e III.



A afirmativa I está errada, uma vez que o DAS (*Direct Attached Storage*) constitui-se de uma arquitetura em que a unidade de armazenamento é conectada diretamente aos servidores, por meio de um barramento ponto a ponto, não se tornando necessário uma infraestrutura de rede ligando os servidores aos dispositivos de armazenamento de dados.

A afirmativa II também está errada porque, na arquitetura SAN, a base são as redes de armazenamento dedicadas e escaláveis que conectam servidores a dispositivos de armazenamento (*storages*), consistindo em uma rede dedicada, geralmente segregada, e de alta velocidade de servidores e dispositivos de armazenamento compartilhados. As redes locais (LANs), por sua vez, são associadas à arquitetura NAS.

A afirmativa III está correta, visto que na arquitetura NAS o arquivo é o tipo de informação que nela trafega. O armazenamento é centralizado e o gerenciamento é realizado pelo próprio *storage*. O NAS trabalha como servidor dedicado, de alto desempenho, alta velocidade e único propósito, além de possuir seu próprio sistema operacional.

Letra c.

005. (FGV/COMPESA/ANALISTA DE GESTÃO – ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) Sobre as formas de organização de um sistema de armazenamento de dados corporativos, analise as afirmativas a seguir.

I – Um DAS (*Direct Attached Storage*) é um sistema de armazenamento que não é compartilhado e nenhum outro dispositivo na rede pode ter acesso direto aos dados.

II – Um NAS (*Network Attached Storage*) apresenta como principal desvantagem a dependência do modelo cliente-servidor para comunicação e troca de dados, o que pode criar uma sobrecarga no compartilhamento de arquivos com vários servidores.

III – Um SAN (*Storage Area Network*) é projetado para conectar elementos de armazenamento, como conjuntos de discos e fitas, a múltiplos servidores por meio de dispositivos de rede de alta disponibilidade, confiabilidade e tolerância a falhas.

Está correto o que se afirma em:

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e III, somente.
- e) I, II e III.



É importante saber a diferença entre DAS, NAS e SAN.

DAS é uma solução bastante básica, de baixa manutenção e baixo custo, em que o sistema de armazenamento faz parte do *host* real ou conectado diretamente ao *host*. Isso pode ser tão simples quanto um disco rígido externo e é o oposto do armazenamento em rede, no qual estações de trabalho e servidores se conectam por meio de uma rede.

NAS oferece serviço e compartilhamento de arquivos dedicados através de uma rede, aumentando o desempenho e a confiabilidade com recursos como RAID e unidades intercambiáveis projetadas para cargas de trabalho com mais unidades. O armazenamento NAS também é uma boa solução para consolidar os sistemas de armazenamento.

SAN é um sistema de armazenamento dedicado e de alto desempenho que transfere dados em nível de bloco entre servidores e dispositivos de armazenamento. A SAN é normalmente usada em *datacenters*, empresas ou ambientes de computação virtual.

A primeira coisa que podemos notar é que o DAS não envolve conexão via rede (*Networking*). Assim, a afirmativa I está correta.

O NAS normalmente trabalha sobre redes TCP/IP e pode ser facilmente integrado. Entretanto, justamente por isso uma sobrecarga é adicionada ao processo. Por exemplo, os protocolos NFS e CIFS utilizam o NAS. Trata-se de protocolos da camada de aplicação, e isso implica em maior processamento para atender a cada requisição. Como comparativo, a SAN tradicionalmente emprega protocolos de rede de baixo nível para transferir blocos de disco, diminuindo o tempo de resposta para acesso ao dado. Portanto, a afirmativa II está correta.

DICA

Observe que o NAS pode interconectar diversos sistemas de arquivo. Já a SAN interconecta diversos discos através de um sistema de arquivo em comum. Enquanto um NAS é um computador conectado a uma rede que fornece serviços de armazenamento de dados baseados em arquivos para outros dispositivos na rede, uma SAN é uma maneira de fornecer aos usuários acesso compartilhado ao armazenamento de dados consolidado em nível de bloco, permitindo até que vários clientes acessem arquivos ao mesmo tempo com desempenho muito alto.

Uma SAN aprimora a acessibilidade de dispositivos de armazenamento, como matrizes de disco e bibliotecas de fitas, fazendo com que pareçam aos usuários como se fossem discos rígidos externos em seu sistema local. Por isso, uma SAN é indicada para ambientes de alto desempenho, com requisitos de disponibilidade e tolerância a falhas. Dessa forma, a afirmativa III também está correta.

Letra e.

006. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Um equipamento tem vários discos iguais. Para obter redundância dos dados, os discos sofrem um espelhamento, em que todos os dados são gravados em dois discos diferentes. Se um disco falhar ou for removido, os dados preservados no outro disco permitem a não descontinuidade da operação do sistema. Para obter esse esquema, a controladora de discos deve ser configurada para RAID:

- a) 0
- b) 1
- c) 4
- d) 5
- e) 6



RAID 0 usa particionamento de disco para fornecer alta taxa de transferência de dados, especialmente para arquivos grandes em um ambiente que não requer redundância de dados.

RAID 1 usa espelhamento de disco para que os dados gravados em um disco físico sejam simultaneamente gravados em outro disco físico. RAID 1 é ideal para bancos de dados pequenos ou outras aplicações que requerem pouca capacidade, mas que também exigem redundância completa dos dados.

RAID 5 usa particionamento de disco e dados de paridade em todos os discos físicos (paridade distribuída) para fornecer alta taxa de transferência de dados e redundância de dados, especialmente para pequenos acessos aleatórios.

RAID 6 é uma extensão do RAID 5 e usa um bloco de paridade adicional. Utiliza particionamento de nível de bloco com dois blocos de paridade distribuída por todos os discos membros. Também oferece proteção contra falhas de disco duplo e falhas ocorridas enquanto um disco único está em reconstrução.

Obs.: Se você estiver usando apenas um *array*, implantar o RAID 6 será mais eficiente do que implantar um disco *hot spare*.

RAID 10, uma combinação de RAID 0 e RAID 1, usa particionamento de disco em discos espelhados. Fornece alta taxa de transferência de dados e redundância completa de dados. RAID 10 pode suportar até oito conexões e até 32 discos físicos por conexão.

RAID 50 é uma combinação de RAID 0 e RAID 5, em que um *array* RAID 0 é particionado em elementos de RAID 5. RAID 50 requer pelo menos seis discos.

RAID 60 é uma combinação de RAID 0 e RAID 6, em que um **array** RAID 0 é particionado em elementos de RAID 6. RAID 60 requer pelo menos oito discos.

Letra b.

007. (FGV/DPE-MT/ANALISTA DE SISTEMAS/2015) Relacione os diferentes níveis padrão de arranjo redundante de discos independentes (RAID) às respectivas características.

1. RAID 0
2. RAID 1
3. RAID 5
4. RAID 6

() Mantém o arranjo funcionando mesmo que dois dos discos falhem, mas requer pelo menos 4 discos.

() Tem como prioridade o desempenho de transferência. O arranjo fica indisponível se apenas um dos discos falhar.

() Balanceia desempenho e redundância, permitindo que o arranjo fique disponível mesmo que um dos discos falhe.

() Tem como prioridade a redundância de dados. O arranjo fica indisponível apenas se todos os discos falharem.

Assinale a opção que indica a relação correta, de cima para baixo.

a) 3 – 2 – 4 – 1.

b) 4 – 2 – 3 – 1.

c) 1 – 4 – 2 – 3.

d) 3 – 1 – 4 – 2.

e) 4 – 1 – 3 – 2.



RAID 0 - Um dado é gravado de forma segmentada em diferentes mídias, de forma que o dado original só pode ser recuperado se todas as mídias do volume RAID estiverem funcionando. A falha de uma das mídias do volume tornará o dado irrecuperável. A leitura e escrita ganha em performance, uma vez que só uma fração do dado é escrita em cada mídia, tornando a leitura também mais rápida.

RAID 1 - Um dado é gravado em todas as mídias do volume RAID, que se tornam um espelho uns dos outros, sem ganho de desempenho.

RAID 2 - Um dado é gravado separadamente em diferentes mídias, segmentados por *bit*. Uma mídia armazenará *bits* de paridade para correção de erros. As mídias precisam ter suas rotações sincronizadas (caso sejam discos rígidos). Este tipo de RAID era utilizado até a década de 1980, quando os HDs ainda não tinham seus próprios mecanismos de correção de erros.

RAID 3 - Semelhante ao RAID 2, porém utiliza a segmentação por *byte*. Também não é utilizado atualmente.

RAID 4 - Semelhante ao RAID 2 e 3, porém utilizando a segmentação pelo (tamanho do) bloco do sistema de arquivos.

RAID 5 - Semelhante ao RAID 4, porém os *bits* de paridade para correção de erros são armazenados em todas os discos do RAID, sem a necessidade de um disco exclusivo para armazenar tais dados. Apresenta ganho de desempenho na escrita, mas especialmente na leitura.

RAID 6 - Semelhante ao RAID 5, porém utiliza o dobro de *bits* de paridade para correção de erros, o que permite que até dois discos do RAID apresentem falhas, e, portanto, mais lento que o RAID 5 em operações de escrita.

Letra e.

008. (FGV/DPE-MT/ASSISTENTE ADMINISTRATIVO/2015) Relacione os métodos de organizações de dados no armazenamento secundário às respectivas definições.

1. Organização sequencial de arquivos.

2. Organização direta de arquivos.

3. Organização indexada de arquivos.

- () Os registros são armazenados fisicamente em uma determinada ordem.
- () Os registros são armazenados por meio de um identificador exclusivo.
- () Os registros podem ser acessados sequencialmente ou diretamente.

Assinale a opção que indica a relação correta, de cima para baixo.

- a) 1, 2 e 3.
- b) 1, 3 e 2.
- c) 2, 1 e 3.
- d) 2, 3 e 1.
- e) 3, 1 e 2.



O armazenamento de pequenos volumes de dados, geralmente, não representa grandes problemas no que diz respeito à distribuição dos registros dentro de um arquivo, desde que a frequência de acessos aleatórios a registros não seja muito elevada.

À medida que cresce o volume de dados e/ou a frequência e a complexidade dos acessos, crescem também os problemas de eficiência do armazenamento dos arquivos e do acesso a seus registros, sendo a sofisticação das técnicas de armazenamento e recuperação de dados uma consequência da necessidade de acesso rápido a registros pertencentes a grandes arquivos ou, simplesmente, arquivos muito solicitados.

A maneira intuitiva de armazenar um arquivo consiste na distribuição dos seus registros em uma ordem arbitrária, um após o outro, dentro da área destinada a contê-lo. Esta ordem pode ser, por exemplo, aquela na qual os registros são gerados. Isto causa uma dificuldade na localização dos registros e uma perda de eficiência, porém esta técnica intuitiva é bastante usada, principalmente durante as fases preliminares da geração de um arquivo.

Um arquivo é formado por uma coleção de registros lógicos, cada um deles representando um objeto ou entidade.

Um registro lógico, ou simplesmente registro, é formado por uma sequência de itens, sendo cada item chamado campo ou atributo. Cada item corresponde a uma característica ou propriedade do objeto representado.

Arquivo Sequencial: os registros são dispostos ordenadamente, obedecendo à sequência determinada por uma chave primária, chamada chave de ordenação. Esta organização, que representa um aperfeiçoamento em relação àquela na qual os registros são dispostos aleatoriamente, representa, também, uma perda de flexibilidade por não acomodar com simplicidade as operações de modificação do arquivo.

Arquivo Indexado: os registros são acessados sempre através de um ou mais índices, não havendo qualquer compromisso com a ordem física de instalação dos registros.

Arquivo Direto: a ideia básica de um arquivo direto consiste na instalação dos registros em endereços determinados com base no valor de uma chave primária, de modo que se tenha acesso rápido aos registros especificados por argumentos de pesquisa, sem que haja necessidade de percorrer uma estrutura auxiliar (índice). Um arquivo direto é semelhante a um arquivo indexado, no sentido de que, nos dois casos, o objetivo principal é a obtenção de acesso aleatório eficiente. Em um arquivo direto, ao invés do índice, é usada uma função que calcula o endereço do registro a partir do argumento de pesquisa.

Dessa forma:

Os registros são armazenados fisicamente em uma determinada ordem é uma referência ao arquivo direto (2), no qual os registros são gravados em endereços determinados com base no valor de uma chave primária.

DICA

Note que não se trata de um índice, mas de uma função que calcula o endereço do registro pesquisado.

Os registros são armazenados por meio de um identificador exclusivo é uma referência ao arquivo sequencial (1), no qual os registros são gravados de forma ordenada (ou seja, sequencial), e assim como o arquivo direto, a sequência também obedece a uma chave primária (identificador exclusivo), o que pode causar uma certa confusão com o arquivo direto, já que ambos utilizam uma chave primária para determinar os endereços nos quais os registros devem ser gravados.

Os registros podem ser acessados sequencialmente ou diretamente é uma alusão ao arquivo indexado (3), pois neste caso é usado um (ou mais) índice(s), não havendo qualquer compromisso com a ordem física de instalação de registros, o que significa que os registros podem ser acessados sequencial ou diretamente.

Letra c.

009. (FGV/AL-MT/ANALISTA DE SISTEMAS – PROGRAMADOR/2013) As alternativas a seguir apresentam características dos arquivos *Heap*, à exceção de uma. Assinale-a.

- a) Inclusão de novos registros é muito eficiente.
- b) Organização de arquivos é a mais simples e básica.
- c) Pesquisa de registros envolve busca sequencial bloco a bloco.
- d) Para um arquivo com b blocos, uma busca envolve em média $b/2$ blocos.
- e) Os registros são posicionados de modo ordenado através da chave de classificação.



Neste tipo de organização mais simples e mais básico, os registros são arquivados na ordem em que são inseridos, de modo que novos registros são inseridos ao final do arquivo. Essa organização é chamada arquivo de *heap* ou pilha. Normalmente, ela é usada com caminhos de acesso adicionais, como os índices secundários. Ela também é usada para coletar e armazenar registros de dados para uso futuro. A inserção de um novo registro é muito eficiente. Um registro pode ser colocado em qualquer lugar no arquivo onde exista espaço para acomodá-lo. Não existe ordenação de registros. Normalmente, existe um único arquivo para cada relação.

Letra e.

010. (FGV/TJ-GO/ANALISTA JUDICIÁRIO – ANÁLISE DE SISTEMAS – BANCO DE DADOS/2014) Considere as seguintes afirmativas comparativas entre métodos de busca baseados em árvores B e funções de *hashing*:

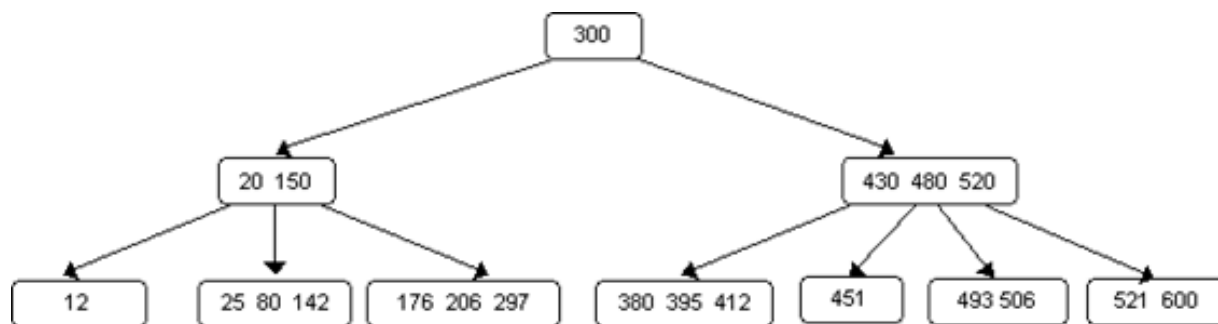
- I – A inserção de chaves não ordenadas é geralmente mais rápida em métodos de *hashing*.
- II – O número médio de acessos para localização de registros tende a ser menor para métodos baseados em *hashing*.
- III – Métodos de *hashing* não disponibilizam acesso sequencial às chaves em ordem crescente ou decrescente.

É correto concluir que:

- a) nenhuma está correta.
- b) somente I está correta.
- c) somente I e II estão corretas.
- d) somente II e III estão corretas.
- e) todas estão corretas.



O método de busca de *hashing* consiste em encontrar um elemento em uma tabela de dispersão baseado em uma chave de busca. Essa chave varre o vetor contendo os elementos de modo a localizá-los de maneira mais eficiente. Já as árvores b (B-Tree) são estruturas de dados aplicadas em buscas, nas quais os nós são ordenados de modo a facilitar a busca dentro da varredura.



Testa-se, então, um determinado valor na varredura: se esse valor for menor, toma-se o caminho A, se não, toma-se o caminho B ou C (ou tantos outros quanto existirem), e aí testa-se sucessivamente até que um valor seja encontrado, ou o último nível seja atingido e nenhum valor encontrado, retornando uma busca vazia.

EXEMPLO

Devemos encontrar o valor 80. 80 é maior que 300? Não. Toma-se o caminho da esquerda. 80 é maior que 20? É. É maior que 150? Não. Toma-se o caminho do meio. E assim sucessivamente...

Agora, vamos analisar cada afirmativa:

A afirmativa I está correta. A ordenação é essencial na árvore b, mas não é necessária no *hashing*, que varre a memória em busca de uma chave gerada por uma função de *hash* aleatória, não ordenada. Portanto, as chaves não ordenadas são mais eficientes em *hashing*.

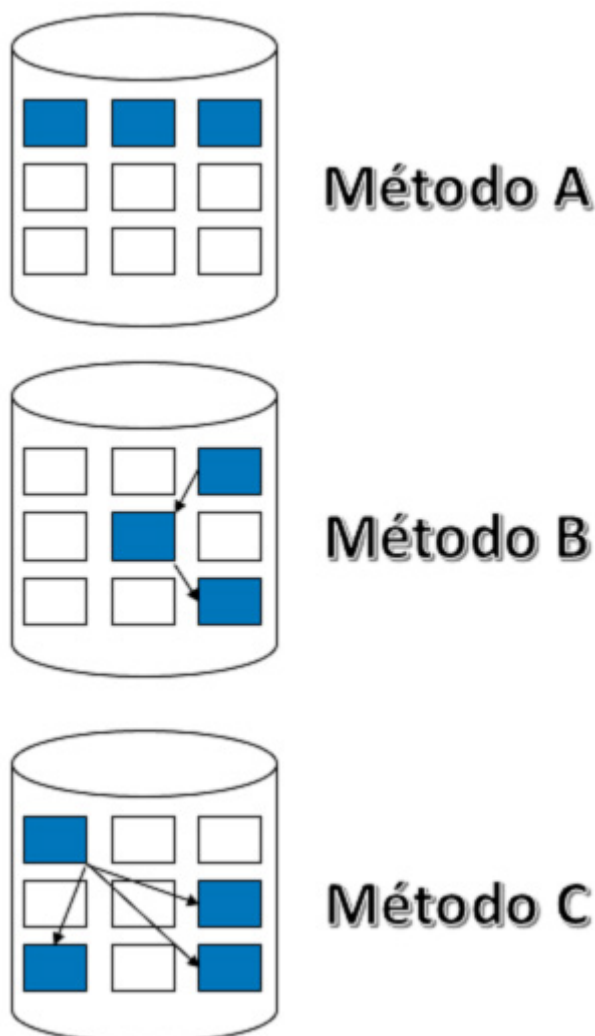
A afirmativa II está correta. Isso pode ser afirmado porque na árvore b provavelmente mais nós serão varridos (em termos de %), dado que um caminho inteiro será varrido em busca da chave. Assumindo que a árvore tenha n níveis, então, n células serão acessadas. Agora, se o vetor de *hash* tem n posições, nem todas elas serão acessadas.

A afirmativa III também está correta. Como dito, o *hash* não trabalha com ordenações, portanto, esse acesso não é permitido.

Letra e.

011. (FGV/IBGE/ANALISTA – ANÁLISE DE SISTEMAS – SUPORTE OPERACIONAL/2016)

Em um sistema computacional, o sistema de arquivos possui diferentes estratégias para superar o problema de alocação de espaço em disco, de uma maneira em que ele possa ser explorado de forma eficiente e os arquivos nele contidos acessados rapidamente. Considere as imagens a seguir que representam 3 métodos de alocação utilizados pelo sistema de arquivos.



As ilustrações dos métodos A, B e C representam, respectivamente, os métodos de alocação:

- a) fila, encadeada, contígua.
- b) fila, indexada, direta.
- c) contígua, indexada, direta.
- d) contígua, encadeada, indexada.
- e) fila, sequencial, indexada.



A alocação contígua consiste em armazenar um arquivo em blocos sequencialmente dispostos, permitindo ao sistema localizar um arquivo através do endereço do primeiro bloco e da sua extensão em blocos. O acesso é feito de maneira simples, tanto para a forma sequencial quanto para a direta.

Na alocação encadeada, um arquivo pode ser organizado como um conjunto de blocos ligados logicamente no disco, independente da sua localização física, sendo que cada bloco possui um ponteiro para o bloco seguinte do arquivo, e assim sucessivamente.

A alocação indexada soluciona o problema da alocação encadeada referente ao acesso direto aos blocos dos arquivos, pois mantém os ponteiros de todos os blocos do arquivo em uma única estrutura denominada bloco de índice.

Letra d.

012. (FGV/COMPESA/ANALISTA DE GESTÃO – ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) Em relação aos componentes do custo de execução de uma consulta SQL, analise as afirmativas a seguir.

I – O custo de acesso ao armazenamento secundário considera os custos de transferência (ler e gravar) blocos entre o armazenamento secundário e os buffers de memória principal.

II – O custo de memória diz respeito ao número de *buffers* de memória principal necessário para a execução de uma consulta.

III – O custo de comunicação diz respeito ao tempo de realizar as operações na memória em registros dentro dos buffers de dados.

Está correto o que se afirma em:

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e II, somente.
- e) I e III, somente.



A afirmativa I está correta. No armazenamento secundário e terciário, estão as mídias que não podem ser acessadas diretamente pela CPU, o que inclui unidades de disco rígido (secundário), unidades de mídia removível, discos ópticos e fitas. Estes níveis de armazenamento são mais baratos, com menor rapidez no acesso aos dados, mas com maior capacidade de armazenamento.

A afirmativa II também está correta. No armazenamento primário, estão as mídias que podem ser acessadas diretamente pela CPU (Unidade Central de Processamento), o que inclui a memória principal e a memória cache. As mídias de armazenamento primário são mais rápidas, mais caras e com menor capacidade de armazenamento do que as mídias de outros níveis de armazenamento.

A afirmativa III está errada. O custo da comunicação diz respeito à consulta de seus resultados.

Letra d.

013. (FGV/MPE-AL/TÉCNICO DO MINISTÉRIO PÚBLICO – GERAL/2018) O conjunto de programas responsável pelo gerenciamento de uma base de dados e que, entre outras funções, suporta uma linguagem de consulta, gera relatórios e disponibiliza uma interface para que os seus clientes possam incluir, alterar ou consultar dados, é chamado de:

- a) Banco de Dados Relacional (BDR).
- b) Dicionário de Dados (DD).
- c) Modelo Entidade Relacionamento (MER).
- d) Sistema de Suporte à Decisão (SSD).
- e) Sistema Gerenciador de Bancos de Dados (SGBD).



Um Sistema de Gerenciamento de Banco de Dados (SGBD) – do inglês *Data Base Management System* (DBMS) – é o conjunto de programas de computador (*softwares*) responsáveis pelo gerenciamento de uma base de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em bancos de dados relacionais, a interface é constituída pelas APIs (*Application Programming Interface*) ou *drivers* do SGBD, que executam comandos na linguagem SQL (*Structured Query Language*).

Letra e.

014. (FGV/TJ-RO/ANALISTA DE SISTEMAS/2015) Ao criar um banco de dados, o administrador (DBA) João analisa o perfil dos dados que serão gerados pela aplicação e percebe que eles devem ser agrupados no disco de acordo com as suas características, como:

- os maiores índices, que demandam mais recursos computacionais, podem ser localizados em discos mais velozes, como *Solid State Device* (SSD); e
- o armazenamento de dados históricos, que raramente são usados e não são muito críticos, pode ser realizado em discos mais lentos.

Em um Sistema Gerenciador de Banco de Dados (SGBD), para agrupar os dados em unidades lógicas de armazenamento, permitindo a definição da localização física dos arquivos correspondentes no disco, João deve definir:

- a) *schemas*.
- b) *data files*.
- c) *tablespaces*.
- d) *extents*.
- e) *data blocks*.



Tablespace é um termo em língua inglesa que designa uma subdivisão lógica de um banco de dados utilizado para agrupar estruturas lógicas relacionadas. As *tablespaces* apenas especificam a localização de armazenamento do banco de dados e são armazenadas fisicamente em *datafiles*, que alocam imediatamente o espaço especificado na sua criação.

Letra c.

015. (FGV/TJ-BA/TÉCNICO JUDICIÁRIO – TECNOLOGIA DA INFORMAÇÃO/2015) Analise as afirmativas a respeito da classe de gerenciadores de bancos de dados, surgida em anos recentes, conhecida como NoSQL.

- I – Mesmo sem suportar tabelas relacionais, baseiam-se em esquemas de dados previamente definidos;
- II – Suas estruturas não permitem o uso de linguagens do tipo do SQL para recuperação de dados;
- III – Garantem operações com as propriedades conhecidas pela sigla ACID;
- IV – Privilegiam a rapidez de acesso e a disponibilidade dos dados em detrimento das regras de consistência das transações.

O número de afirmativas corretas é:

- a) uma
- b) duas
- c) três
- d) quatro
- e) cinco



A afirmativa I está errada. Os modelos NoSQL se baseiam em modelos com ausência de esquema (*schema-free*) ou esquema flexível.

A afirmativa II está errada. NoSQL permite o uso de SQL em alguns tipos de bancos.

A afirmativa III está errada. NoSQL não garante operações com as propriedades conhecidas pela sigla ACID, e sim pelas propriedades BASE.

A afirmativa IV está certa. NoSQL privilegia a rapidez e disponibilidade em detrimento de consistência. Não quer dizer que ele não tem consistência em alguns momentos, e sim que privilegia as outras propriedades.

Letra a.

016. (FGV/FIOCRUZ/TECNOLOGISTA EM SAÚDE – GEOPROCESSAMENTO/2010) Sobre banco de dados orientado a objeto, pode-se dizer que:

- a) é um banco de dados no qual os dados são guardados como propriedades de objetos, e os objetos são valores definidos segundo classes ou tipos de dados complexos.
- b) é formado por tabelas e registros de dados inseridos nas tabelas que estão diretamente vinculados ao dado primário.
- c) é um banco de dados que opera em rede e permite que várias tabelas sejam usadas simultaneamente por meio do uso de referências.
- d) diferencia-se de outros modelos de bancos de dados no sentido em que especifica a forma do resultado, e não o caminho para chegar a ele.
- e) é um conjunto de bancos de dados logicamente inter-relacionados, distribuídos em rede.



Os bancos de dados orientado a objetos armazenam os dados como objetos, e não como tabela. É baseado no paradigma da programação orientada a objetos. Nesse modelo de banco de dados, as funcionalidades de orientação a objetos são integradas às do banco de dados. Cada informação é armazenada na forma de objetos, ou seja, utiliza a estrutura de dados denominada orientação a objetos, a qual permeia as linguagens mais modernas. O gerenciador do banco de dados para um orientado a objeto é referenciado por vários como ODBMS ou OODBMS. Este modelo define o banco de dados como uma coleção de objetos, ou elementos de *software* reutilizáveis, com recursos e métodos associados.

Letra a.

017. (FGV/MEC/ANALISTA DE SISTEMAS – ESPECIALISTA/2009) O Diagrama Entidade-Relacionamento, proposto por P. Chen, é uma ferramenta tipicamente utilizada para a elaboração do seguinte modelo de dados:

- a) físico
- b) interno
- c) externo
- d) conceitual
- e) hierárquico



O Modelo Entidade Relacionamento é uma ferramenta para modelagem de dados, utilizada durante a modelagem do projeto conceitual de banco de dados. A utilização do MER possibilita a criação de modelos na forma de diagramas, empregando para tanto o DER – Diagrama de Entidades e Relacionamento –, que permite representar as estruturas de dados referentes a uma parcela do mundo real (Domínio do Problema ou Minimundo) como resultado da abstração executada por um analista quando da realização do levantamento de requisitos do *software*.

Letra d.

018. (FGV/MEC/ADMINISTRADOR DE BANCO DE DADOS/2009) Em um Sistema Gerenciador de Banco de Dados – SGBD, os elementos de dados possuem uma estrutura lógica visualizada na forma de tabelas.

Nesse sentido, existem, inclusive, pacotes nos SGBDs que podem vincular elementos oriundos de várias tabelas, desde que essas compartilhem elementos comuns de dados.

Essa estrutura lógica de dados é conhecida como do tipo:

- a) em rede.
- b) relacional.
- c) em pilha.
- d) hierárquica.
- e) em árvore.



Bases de dados relacionais consistem, principalmente de três componentes: uma coleção de estruturas de dados, nomeadamente relações, ou informalmente tabelas; uma coleção dos operadores, a álgebra e o cálculo relacionais; e uma coleção de restrições da integridade, definindo o conjunto consistente de estados de base de dados e de alterações de estados. As restrições de integridade podem ser de quatro tipos: domínio (também conhecidas como type), atributo, relvar (variável relacional) e restrições de base de dados.

Letra b.

019. (FGV/AL-MA/TÉCNICO DE GESTÃO ADMINISTRATIVA – ANALISTA DE SISTEMAS/2013) Sobre sistemas de gerenciamento de bancos de dados, assinale a afirmativa incorreta.

- a) Em uma tabela chamada Carro, com os atributos Modelo, Ano, Cor e Placa, o atributo mais apropriado para servir como chave primária é Placa.
- b) O processo de aplicação de regras ao projeto de um banco de dados, com o intuito de permitir um acesso eficiente e armazenamento consistente dos dados, é denominado normalização do banco de dados, ou apenas normalização.
- c) O modelo hierárquico de banco de dados é o mais apropriado para dados que são representados por um relacionamento pai-filho, no qual cada registro pai pode estar relacionado com vários registros filho.
- d) Uma Visão (View) de um banco de dados contém todas as informações sobre as entidades, atributos e tabelas. Seu objetivo é padronizar e unificar os termos utilizados nas variáveis utilizadas nos bancos de dados.
- e) O “varchar” é uma cadeia de caracteres de tamanho indeterminado. Campos do tipo “varchar” podem ter qualquer tamanho, até um limite que varia de acordo com o SGBD.



Uma visão (View) é uma tabela virtual, resultado de uma consulta, que representa um subconjunto de outra(s) tabela(s). Dessa forma, ela não contém todas as informações referentes as entidades, atributos e tabelas de quem ela deriva.

Letra d.

020. (FGV/BADESC/ANALISTA DE SISTEMAS – BANCO DE DADOS/2010) Com relação à especificação e à manipulação de um atributo multivalorado pelos SGBDs, analise as afirmativas a seguir.

I – O modelo de dados relacional (especificado por Codd), não aceita este tipo de atributo, multivalorado. O modelo de dados relacional preconiza que este atributo deve ser processado pela implementação de uma tabela que possa contê-lo.

II – O modelo de dados relacional (especificado por Codd) especifica que para processar um atributo multivalorado deve antes normalizar a tabela em que este se encontra.

III – O modelo de objetos determina que todos os atributos de um objeto, inclusive os multivalorados, sejam encapsulados no objeto, permitindo, desta forma, a sua implementação e processamento.

Assinale:

- a) se somente a afirmativa I estiver correta.
- b) se somente a afirmativa II estiver correta.
- c) se somente a afirmativa III estiver correta.
- d) se somente as afirmativas I e II estiverem corretas.
- e) se todas as afirmativas estiverem corretas.



A afirmativa I está correta. O atributo multivalorado é trabalhado logo na 1FN (primeira forma normal), em que uma nova tabela é criada para armazenar esse atributo multivalorado.

A afirmativa II está correta. Normalizar a tabela significa colocá-la em alguma forma normal.

Obs.: Vamos relembrar aqui as formas normais:

Primeira Forma Normal (1FN): diz-se que uma tabela está na 1FN se ela não tem atributos multivalorados (ou seja, atributos cujos valores podem ser formados por conjunto de valores. Se a tabela tiver algum atributo multivalorado, podemos dizer que ela não está na 1FN;

Segunda Forma Normal (2FN): diz-se que uma tabela está na 2FN se ela está na 1FN e se todos os atributos não chave dependem unicamente da chave primária;

Terceira Forma Normal (3FN): diz-se que uma tabela está na 3FN quando ela também está na 2FN e quando todos os atributos são funcionalmente independentes entre si.

Forma Normal de Boyce Codd (BCNF): Para estar na BCNF, a tabela deve estar na 3FN e não deve existir nenhuma dependência funcional em superchave candidata.

DICA

Uma superchave é um conjunto de atributos de tabela para os quais não existem tuplas com a mesma combinação. Para que uma superchave seja candidata, ela precisa ter o menor número de atributos possível.

Obs.: Quarta Forma Normal (4FN): diz-se que uma tabela está na 4FN quando não há dependência multivalorada em chave candidata. As chaves candidatas da tabela Matrícula são (id_aluno, id_curso) e (id_curso, nome_aluno).

A afirmativa III está correta. O modelo de objetos é aquele do banco de dados orientado a objetos, em que há uma extensão dos tipos de dados permitidos pelo modelo relacional, de modo que objetos possam encapsular atributos que são outros objetos ou atributos de tipos primitivos, inclusive multivalorados.

Letra e.

Sérgio Sierral

Pós-Graduando em Business Intelligence pela Anhanguera, Engenheiro da Computação pelo Instituto de Estudos Superiores da Amazônia-IESAM. Atualmente é Técnico Judiciário no Tribunal Regional Federal da 1ª Região e professor no Gran Cursos Online. Certificado em diversos fabricantes, como Microsoft (MTA e MCSA) e VMWare (VSP, VTSP e VCA). Trabalhou na Oi S.A. como Business Development Manager-BDM, onde ficou responsável na disseminação e apresentação do portfólio de Soluções de TI e Segurança. Experiência como Consultor de Projetos e Pré-Vendas de Solução de TI como Datacenter, Virtualização, Armazenamento e Backup, apoiando as mais diversas empresas em sua Transformação Digital.

**NÃO SE ESQUEÇA DE
AVALIAR ESTA AULA!**

**SUA OPINIÃO É MUITO IMPORTANTE
PARA MELHORARMOS AINDA MAIS
NOSSOS MATERIAIS.**

**ESPERAMOS QUE TENHA GOSTADO
DESTA AULA!**

**PARA AVALIAR, BASTA CLICAR EM LER
A AULA E, DEPOIS, EM AVALIAR AULA.**

AVALIAR 